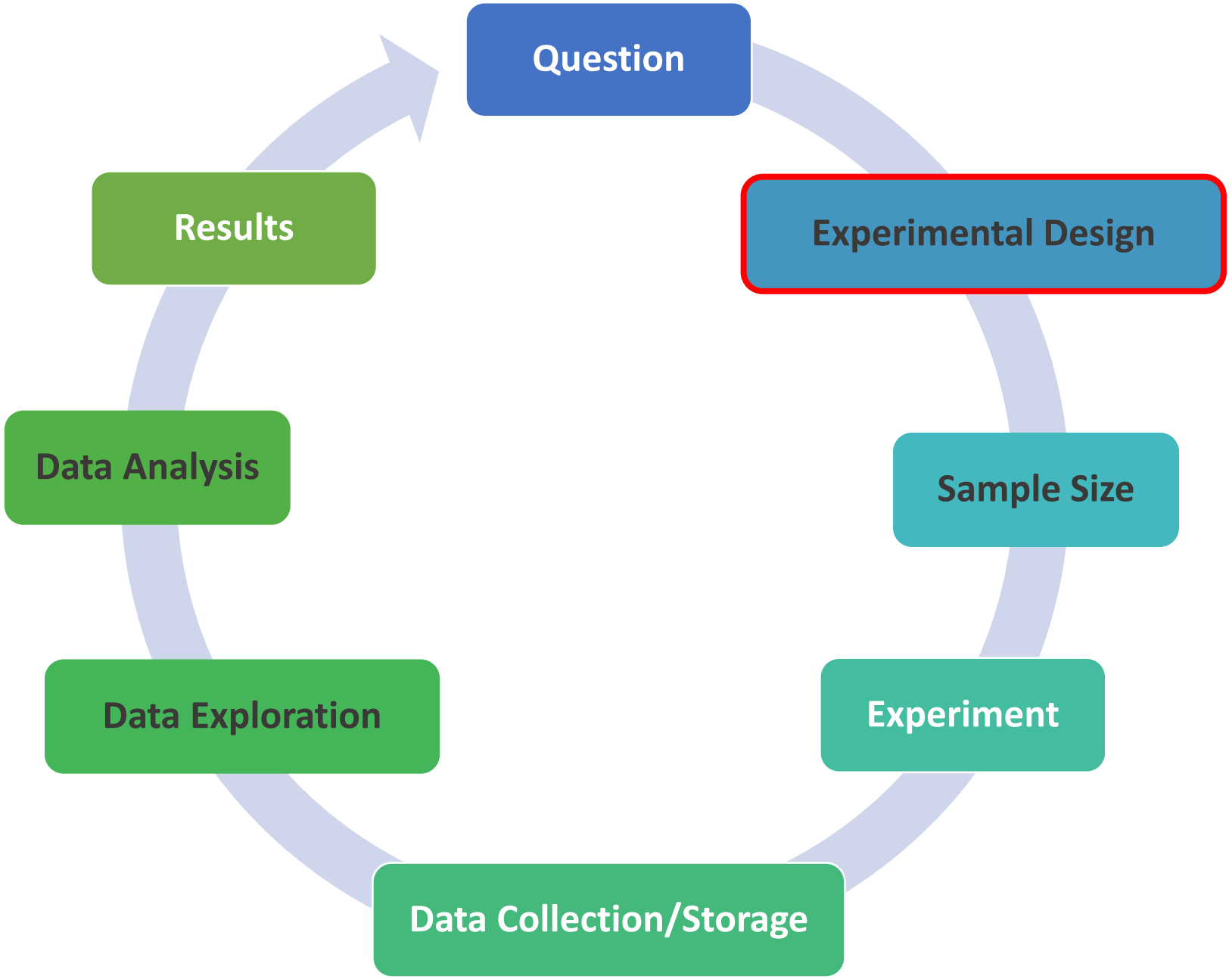
Babraham Institute

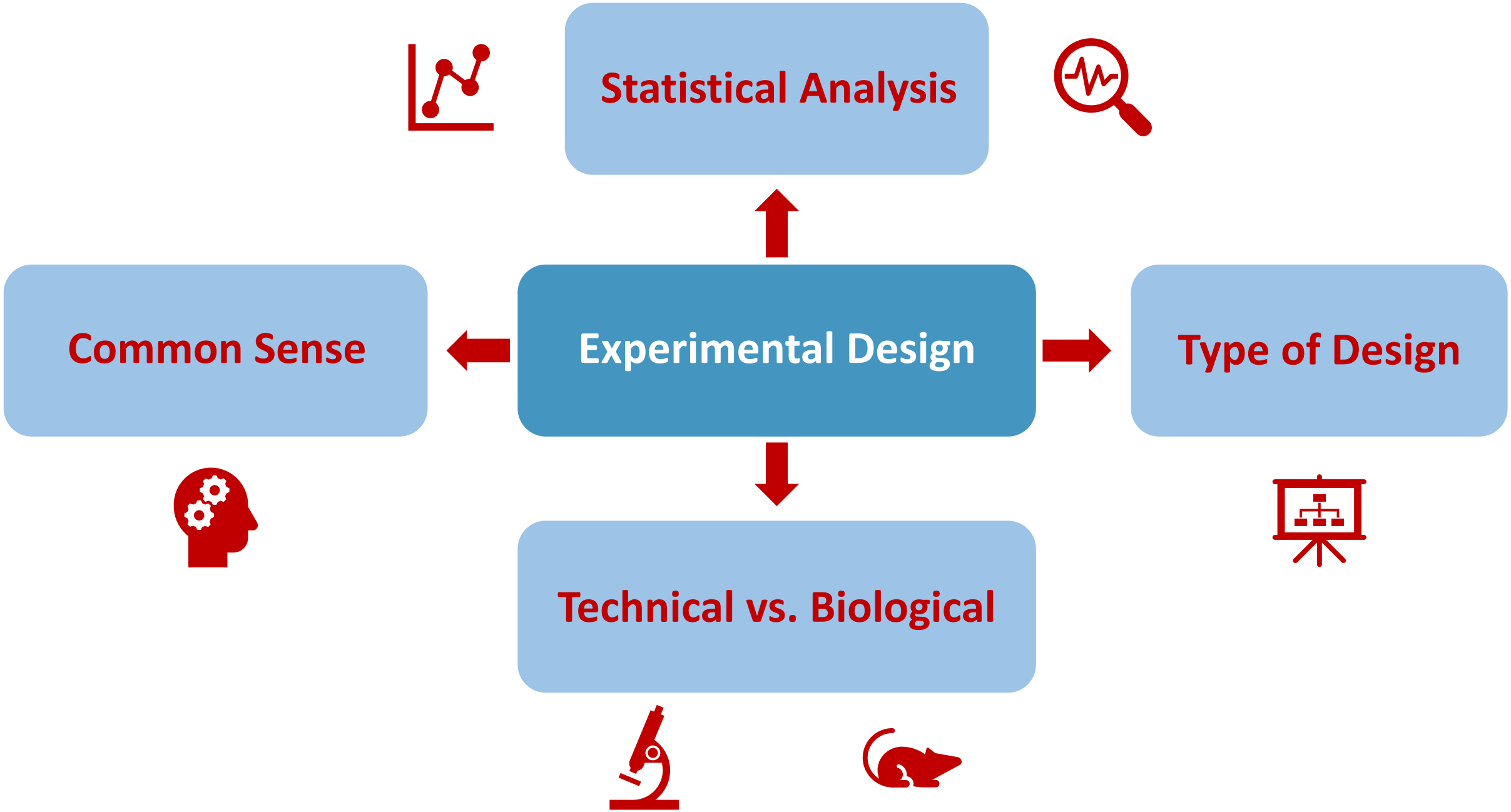
Experimental design

Hayley Carr & Anne Segonds-Pichon

v2025-01

Babraham Bioinformatics


Question
Experimental Design
Sample Size
Experiment
Data Collection/Storage
Data Exploration
Data Analysis
Results

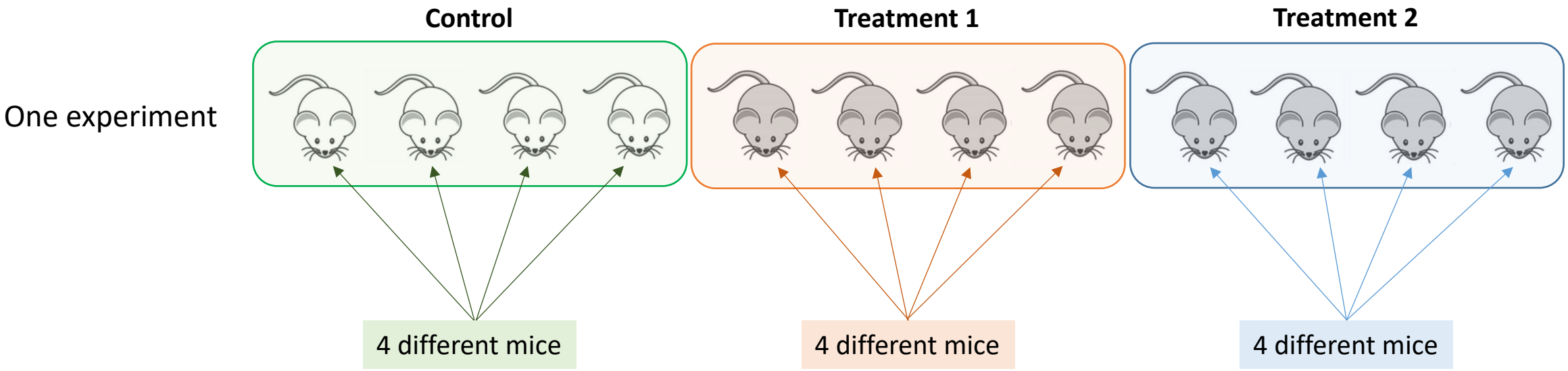

Statistical Analysis
Common Sense
Experimental Design
Type of Design
Technical vs. Biological

Experimental design
Independent versus matched design

Experimental design

Independent design

- 2 or more groups in an experiment with **independent** subjects
- **Example**: 3 groups with n=4 in the control group and n=4 in each treated group

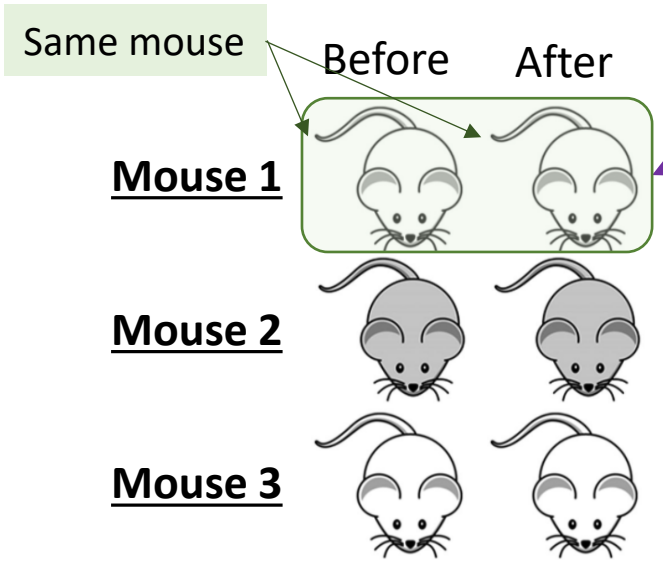


Experimental design

Matched design

- Also called repeated = dependent = paired (2 groups)
  - **Design 1**: ≥2 measures per animal/subject/petri dish

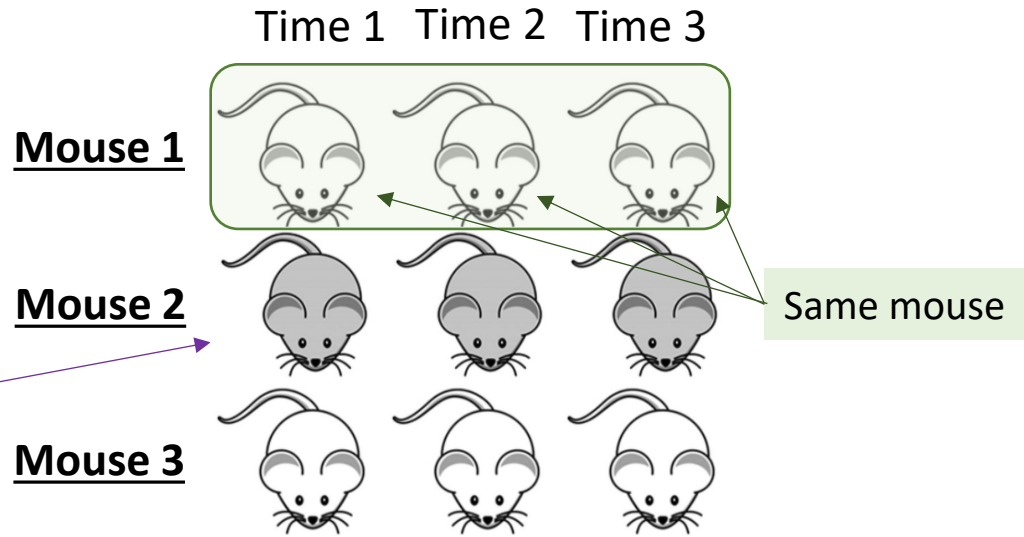
Example 1: before/after treatment measures



Paired/Matched values
Difference or ratio possible for each mouse

Matched/Repeated values
Ex: Difference or ratio vs. Time 1 for each mouse

Example 2: 3 time points

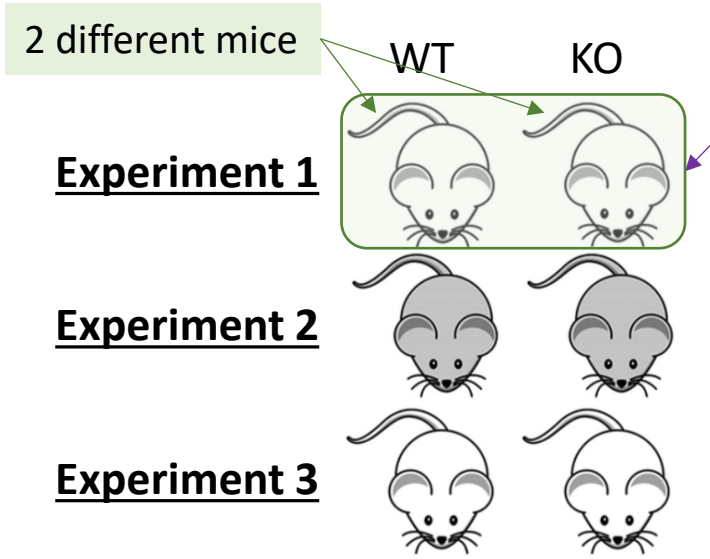


Experimental design

Matched design

- **Design 2**: experiment repeated independently

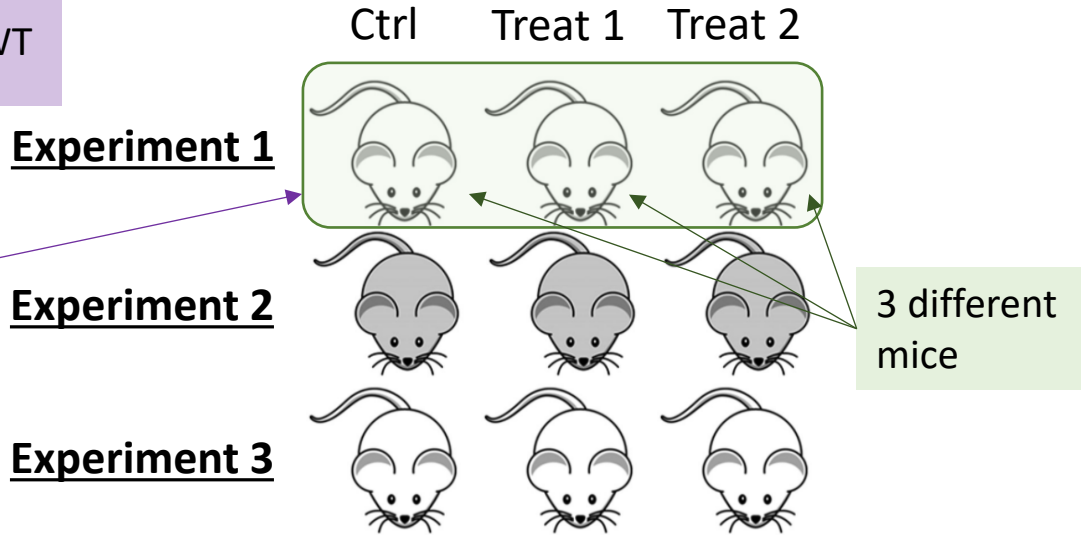
Example 1: 3 independent experiments
2 mice within each: WT and KO



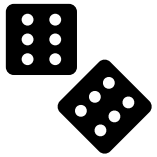
Matched/Paired values
Each KO mouse is matched with a WT
Difference or ratio possible
for each experiment

Matched values
Ex: Difference or ratio vs. Time 1
for each experiment

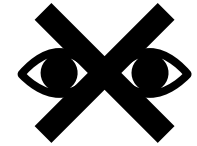
Example 2: 3 independent experiments
3 mice within each: control and 2 treatments



Experimental design



Other design considerations: bias



The lack of bias-reducing measures such as randomisation and blinding can contribute to as much as 30-45% inflation of effect sizes

- Simple **randomisation** or randomisation within blocks
- Example nuisance variables for blocking:
  - Time or day of experiment
  - Litter, cage, etc.
  - Person carrying out experiment
  - Sex, age, body weight, etc.
  - Another related measure (e.g. starting cell numbers, level of cytokine, or similar)
- Use random number generator, flip a coin, roll a dice

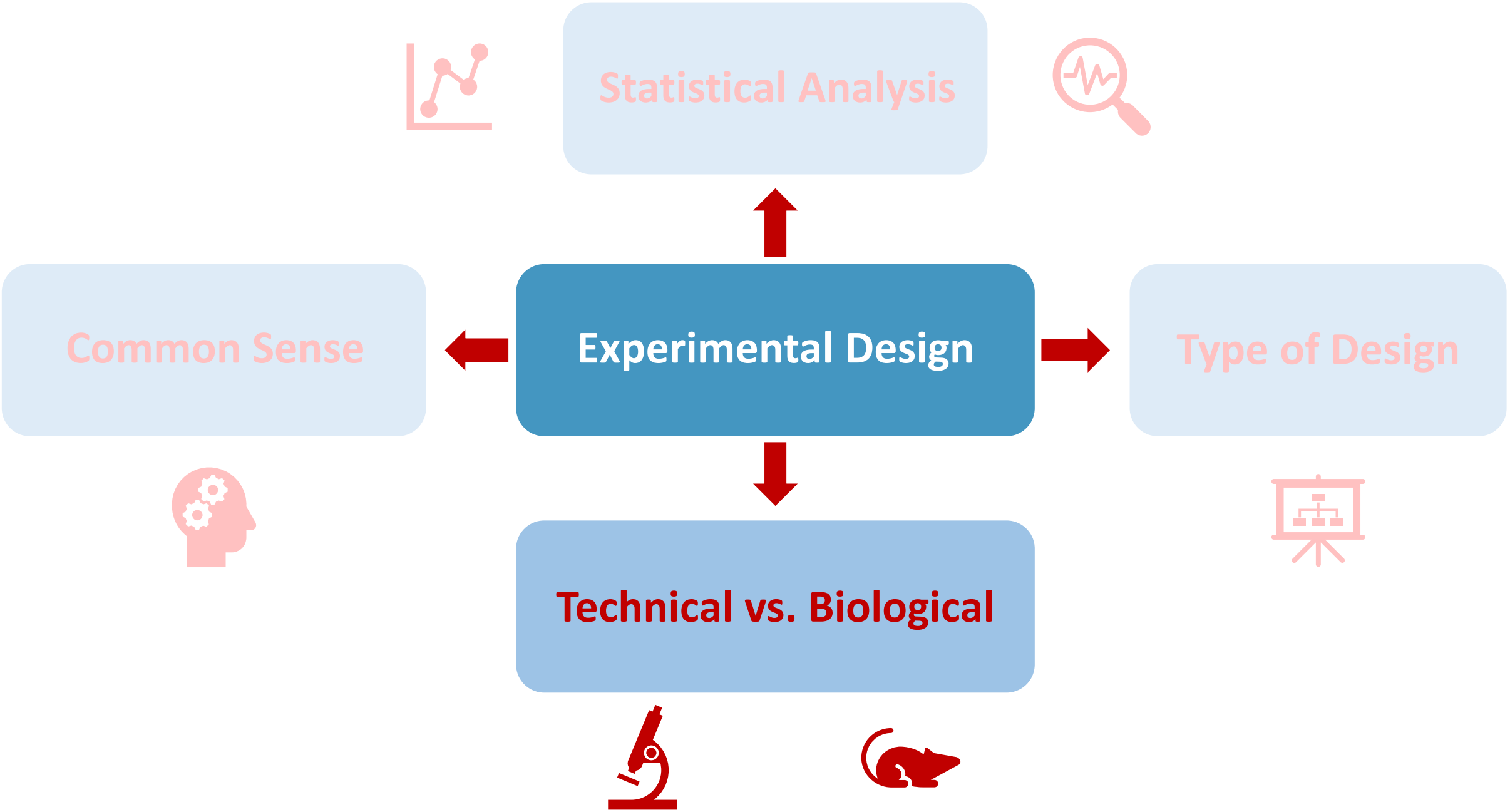
Blinding should be thought about:

- When allocating groups
- When doing the experiment
- When measuring outcomes
- When doing the analysis

The ARRIVE guidelines 2.0

1. Study design
2. Sample size
3. Inclusion and exclusion criteria
4. **Randomisation**
5. **Blinding/Masking**
6. Outcome measures
7. Statistical methods
8. Experimental animals
9. Results

https://arriveguidelines.org/arrive-guidelines

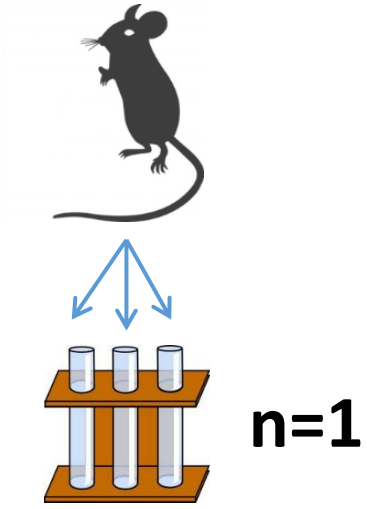

Statistical Analysis
Common Sense
Experimental Design
Type of Design
Technical vs. Biological

Technical/biological replicates

Not always easy

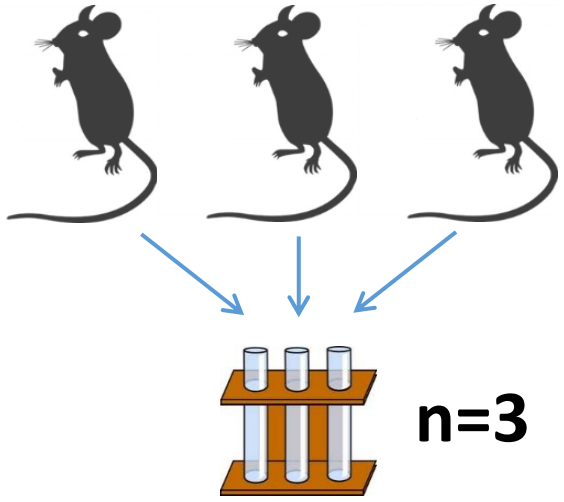
Technical versus biological replicates

- **Technical**: repeated measures of the same sample → variability in the protocol
- **Biological**: measures of biologically distinct samples → biological variation
- **Average** of technical replicates = 1 biological replicate → ↓measurement error



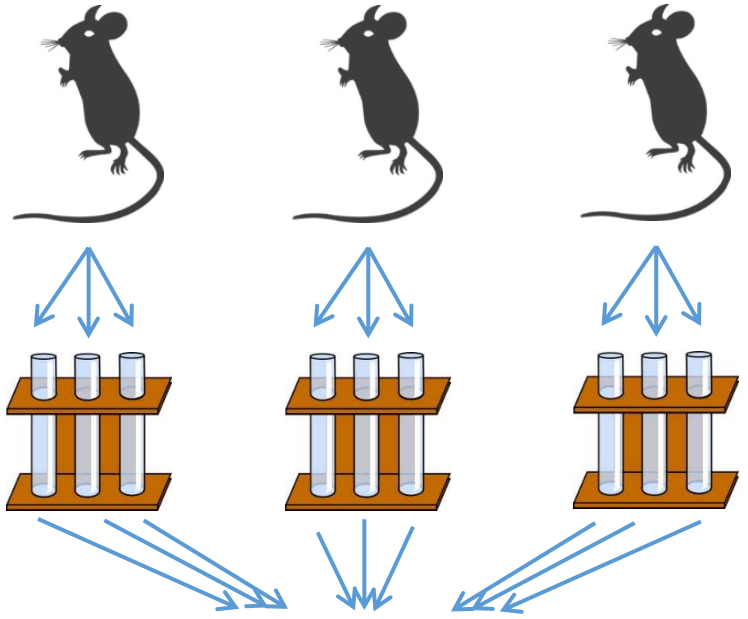
n=1

Technical



n=3

Biological



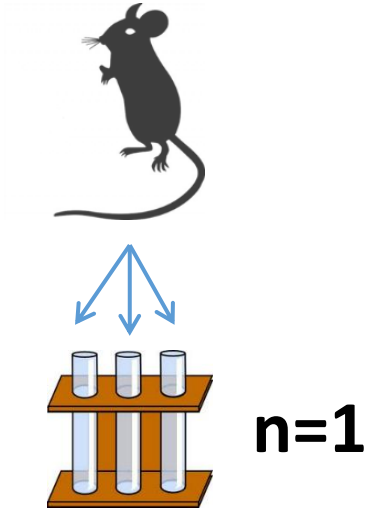
Technical & Biological

n=3

Technical versus biological replicates

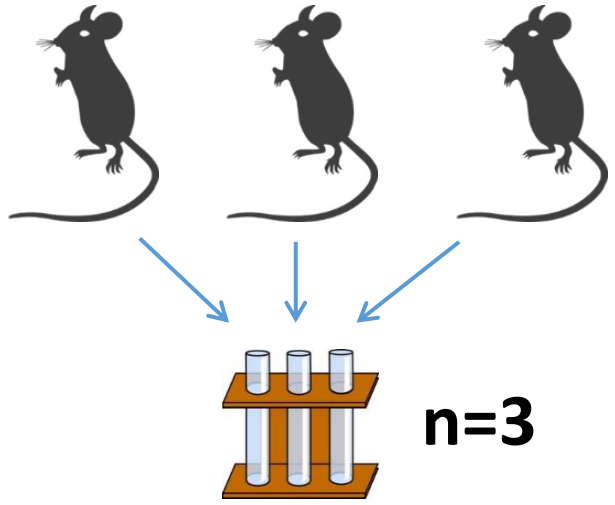
Not always easy to tell the difference

- Definition of technical and biological depends on the model
- Mouse, human, plant, or other complex organism
  - One value per individual organism = biological replicate



n=1

Technical



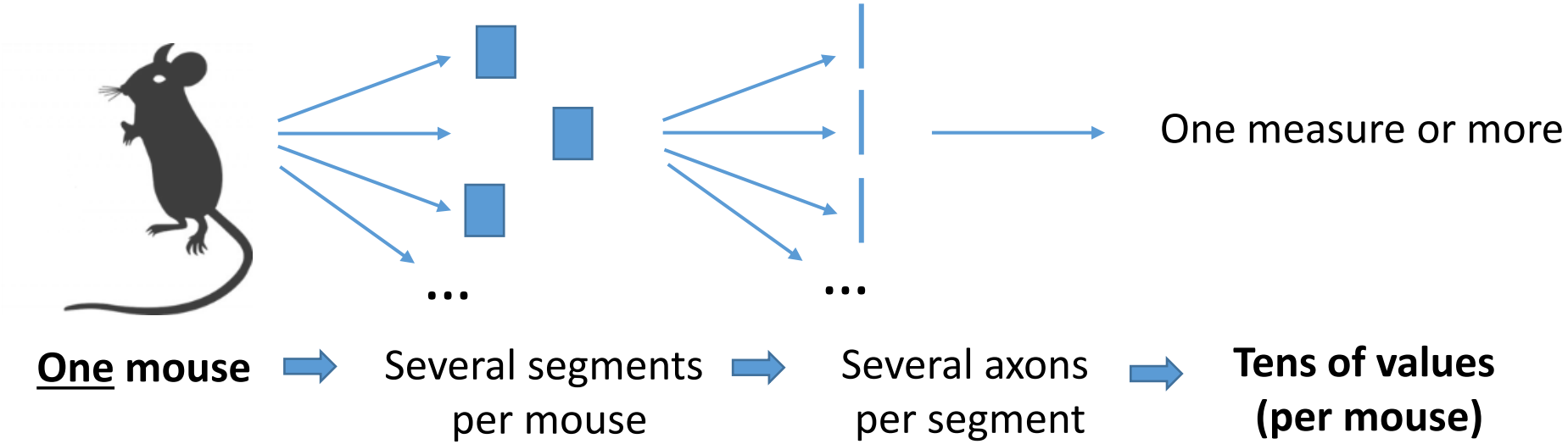
n=3

Biological

Technical versus biological replicates

Not always easy to tell the difference

- The model: mouse or other complex organism
  - >1 value per individual, e.g. axon degeneration

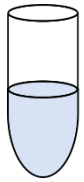


- **What to do**? **Not one good answer**
  - In this case: mouse = experiment unit, nerve segments = biological replicates, axons = technical replicates
  - But how generalisable to a wider population is this?

Technical versus biological replicates

Not always easy to tell the difference

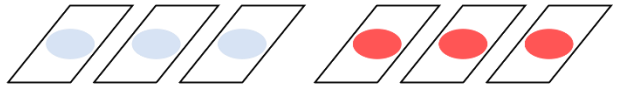
- Cells, worms, etc. = many ‘individuals’:
  - What is ‘n’ in cell culture experiments?



Vial of frozen cells

Control Treatment

Dishes, flasks, wells, …
Cells in culture
Point of Treatment

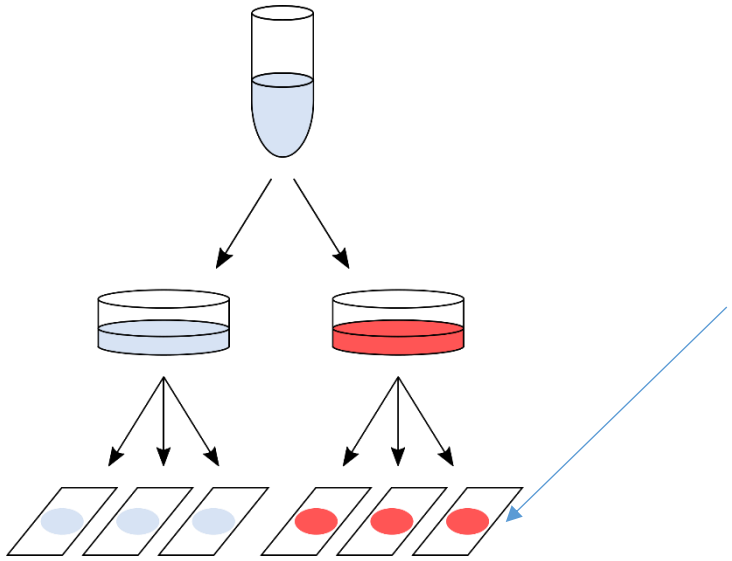


Glass slides, microarrays,
lanes in gel, wells in plate,
…
Point of Measurements

Technical versus biological replicates

Not always easy to tell the difference

- **Design 1**:

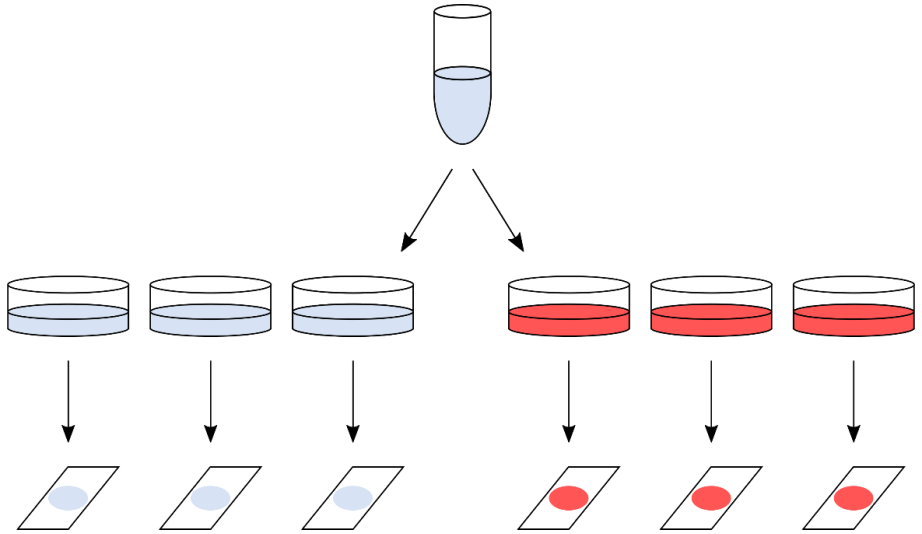


  - After quantification: 6 values
    - Sample size: **n = 1**
      - no independence between slides
      - variability = pipetting/measurement error

Technical versus biological replicates

Not always easy to tell the difference

- **Design 2**:



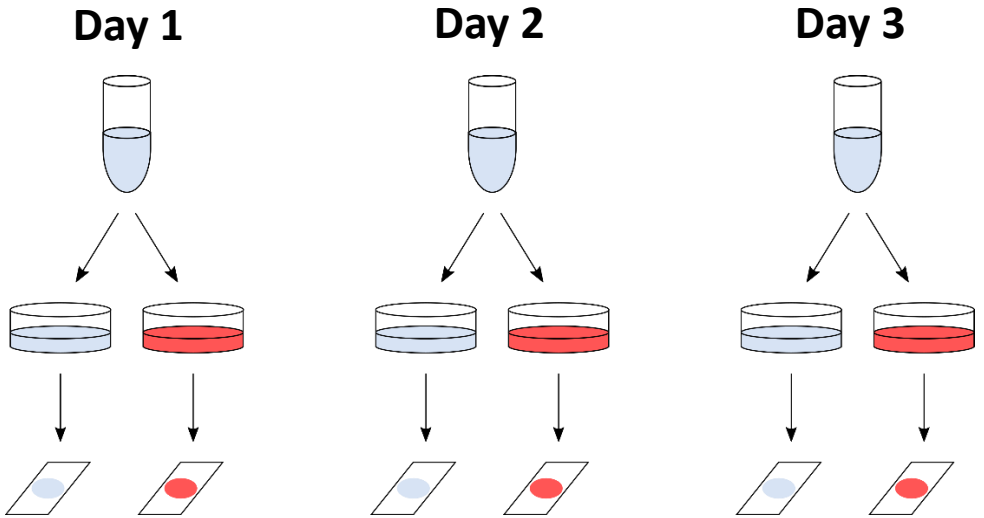
Everything processed on the same day

- After quantification: 6 values
  - Sample size: **n = 1**
    - no independence between plates
    - variability = bit better as sample split higher up in the hierarchy

Technical versus biological replicates

Not always easy to tell the difference

- **Design 3**: Often, as good as it can get

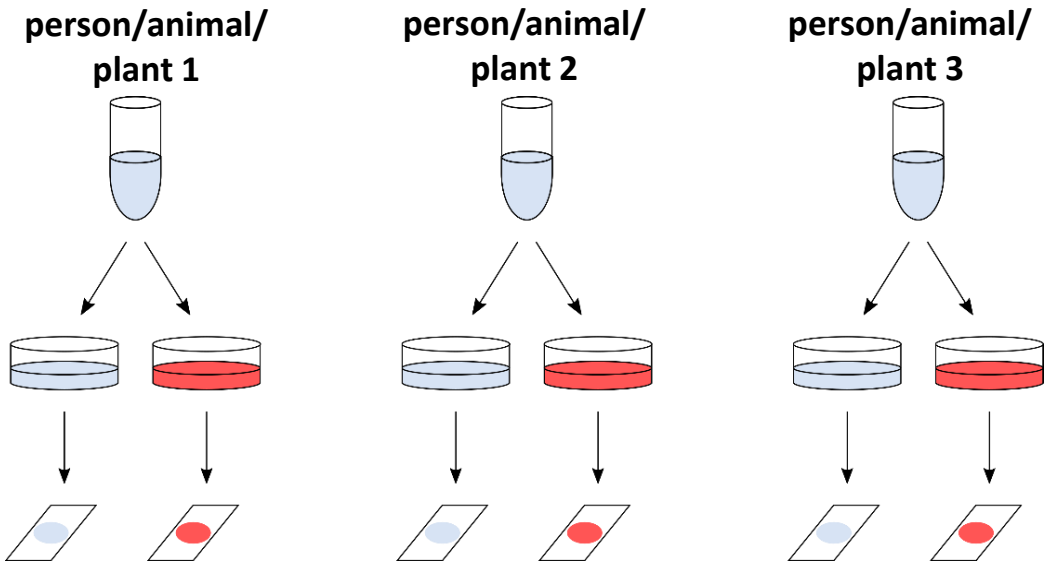


- After quantification: 6 values
  - Sample size: **n = 3**
    - **Whole procedure** repeated **3 separate times**
      - 3 days are (mostly) independent
    - Technical variability but at the highest hierarchical level
    - 2 glass slides = paired observations

Technical versus biological replicates

Not always easy to tell the difference

- Design 4: The ideal design



- After quantification: 6 values
  - Sample size: **n = 3**
    - Real biological replicates

Technical versus biological replicates

- Identify technical and biological replicates
- Make the replicates as independent as possible
- Consider wider factors, e.g. rarity of samples, cost and accuracy of measurements

- Never mix technical and biological replicates
- Do not generalise your results beyond what you are able to show

- How ‘good’ your biological replicates are determines how generalisable your results are
  - ↑ confidence if true biological replicates
  - ↓ confidence if single cell line

Statistical Analysis

Common Sense

Experimental Design

Type of Design

Technical vs. Biological

Experimental Design

Statistical analysis

- Think about the statistical analyses **before** you collect any data
  - **Translate the hypothesis into statistical questions**

What data will I collect?

How will it be recorded/produced?

Do I know enough stats to analyse my data?

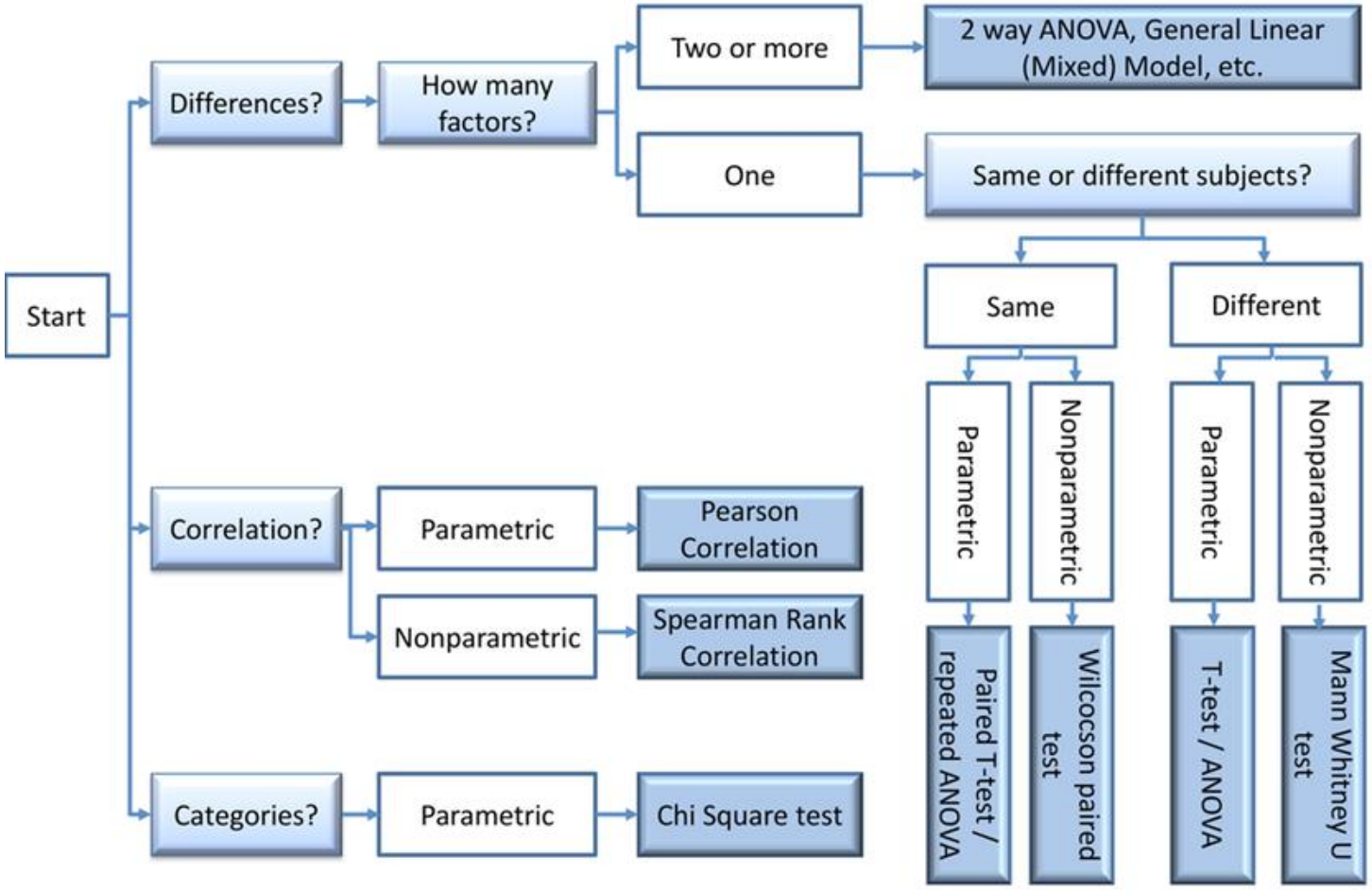
Will I have access to the raw data?

I have been told to do this test/use that template, is that right?

If not: ask for help!

Experimental Design

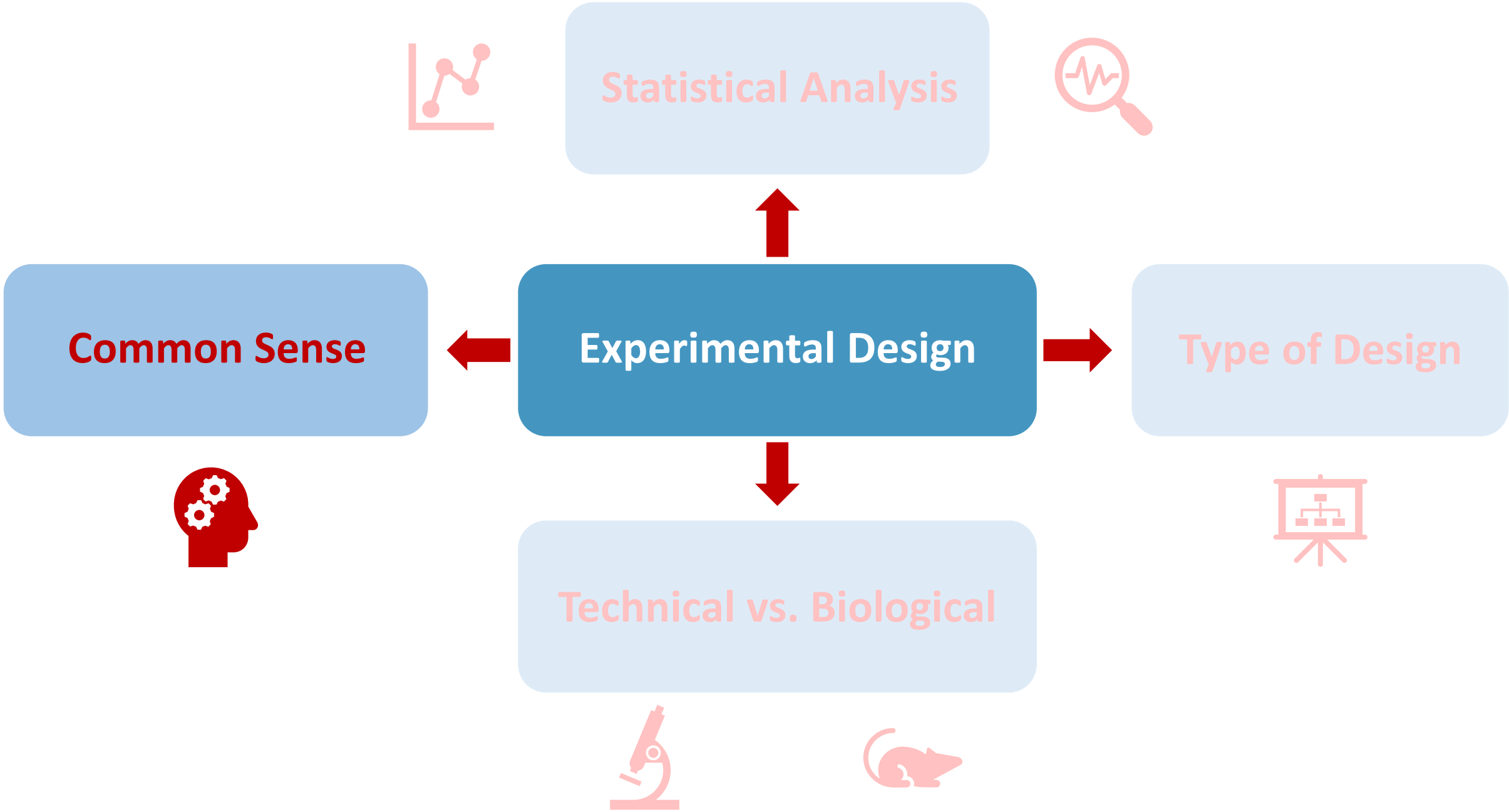
Statistical analysis



Experimental Design

Exploratory data analysis (EDA)

- Purpose of EDA = discovery
- Less confidence in results so follow up with confirmatory tests
- Confirmatory approaches (hypothesis testing) provide stronger statistical evidence
- EDA ≠ “p-hacking” but could be if reported as if confirmatory
- **Be clear about approach taken** – harm comes from misrepresenting processes


Statistical Analysis
Common Sense
Experimental Design
Type of Design
Technical vs. Biological

Experimental Design

Common sense

- Design your experiment to be analysable
- Imagine how your results will look
- Imagine what could go wrong at each step
- Accept limitations and account for them (be prepared for follow up experiments, if required)

- The gathering of results or carrying out of a procedure is not the end goal
- Don’t get fixated on being able to perform a cool technique or experimental protocol
- Don’t overcomplicate
- Don’t get overwhelmed (ask for help)

Will these results address your hypothesis?

Babraham Institute

Power Analysis

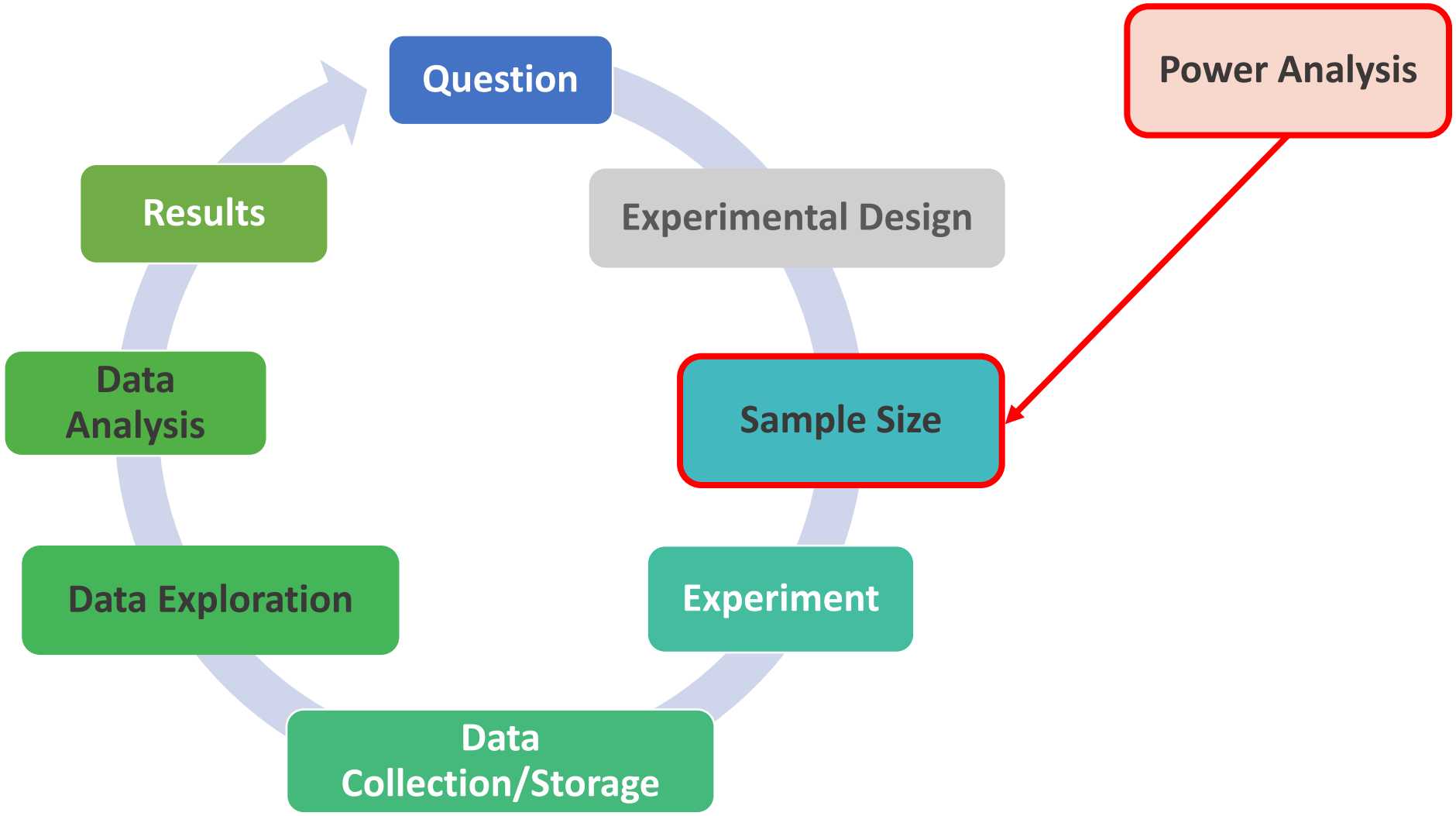
Hayley Carr & Anne Segonds-Pichon

v2023-09

Babraham Bioinformatics

Power analysis

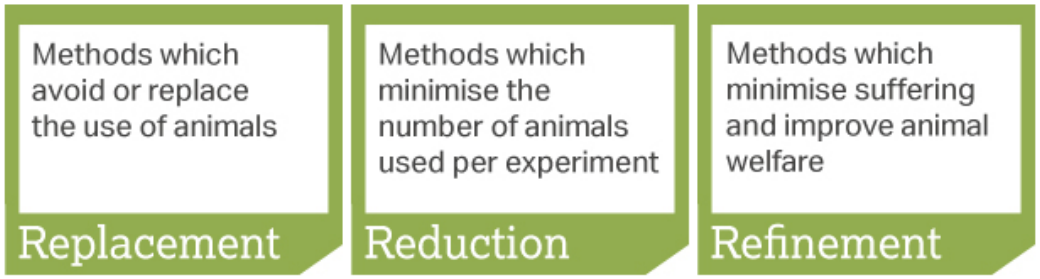
- **Power analysis** is about estimating the **appropriate sample size**.



Sample size

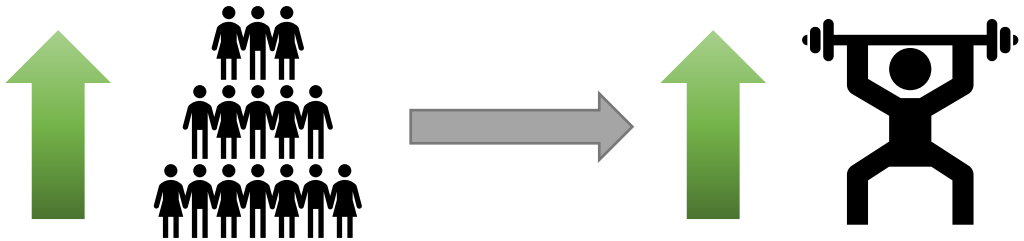
- **Too big**: waste of resources
- **Too small**: may miss the effect (p>0.05) + waste of resources
- **Grants**: justification of sample size
- **Publications:** reviewers ask for power calculation evidence
- **Home office (UK)**: the 3 Rs: Replacement, **Reduction** and Refinement

- To estimate an appropriate **sample size**, we need to do a **power analysis**



Statistical power

- In a nutshell: the bigger the experiment (bigger sample size), the bigger the power (more likely to pick up a difference)
- **Power** = probability of **detecting an effect**, given that the effect is really there
  - = the probability that a statistical test will reject a false null hypothesis (H0)
    - To really understand power, we first need to understand some statistical concepts…

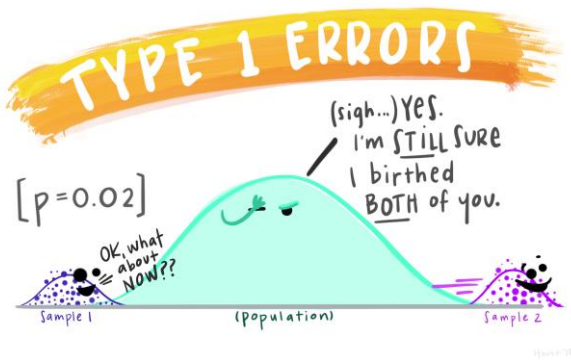


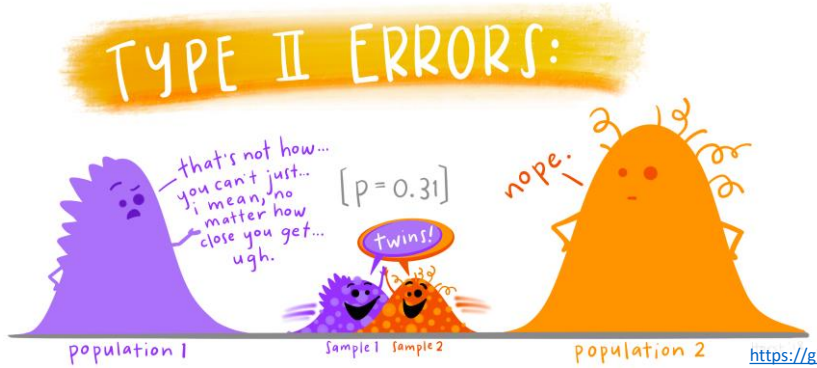
Hypothesis testing

- The null hypothesis: H_0 = no effect
- The aim of a statistical test is to reject or not H_0.

Statistical decision	True state of H_0	
	H_0 True (no effect)	**H_0 False (effect)**
Reject H_0	**Type I error α** **False Positive**	Correct **True Positive**
Do not reject H_0	Correct **True Negative**	**Type II error β** **False Negative**

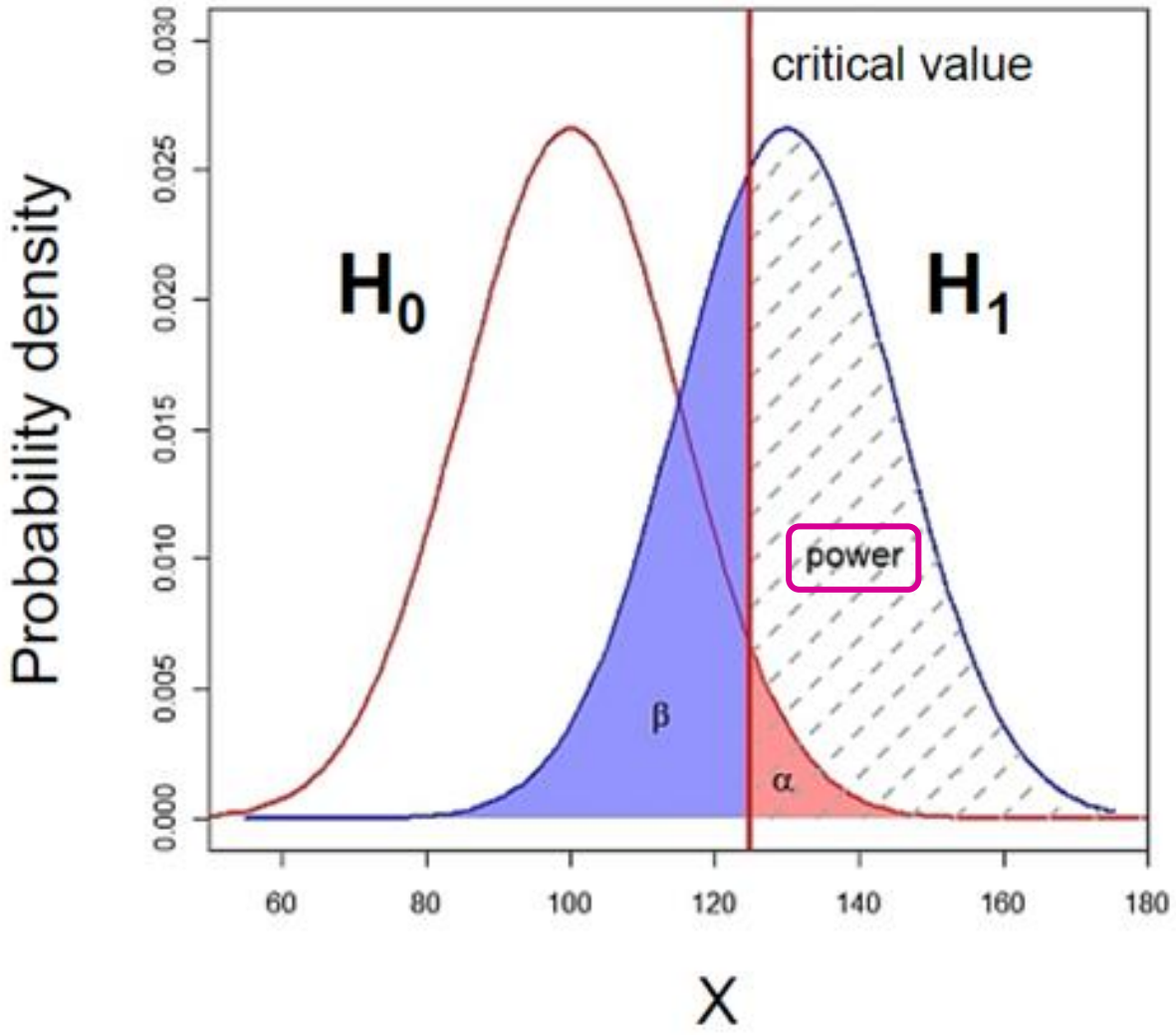
Properly **powered** studies minimise this



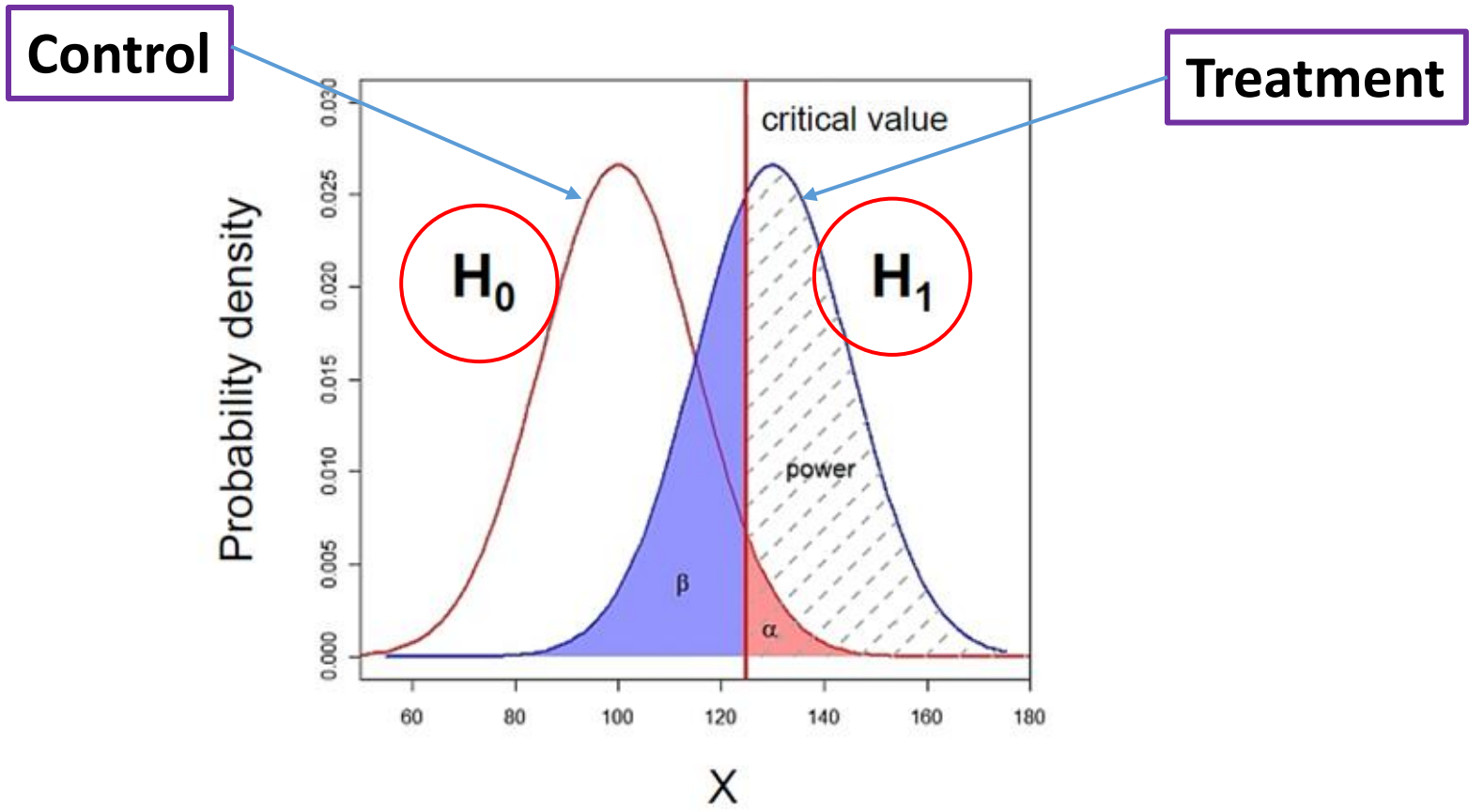


https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

What does Power look like?

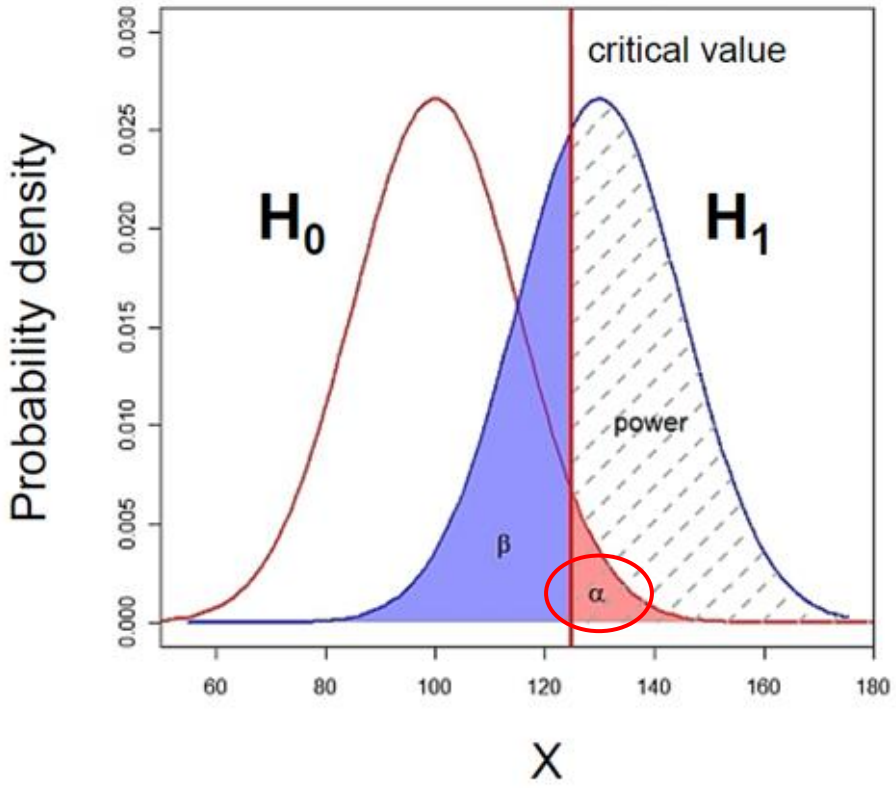


What does Power look like? Null and alternative hypotheses



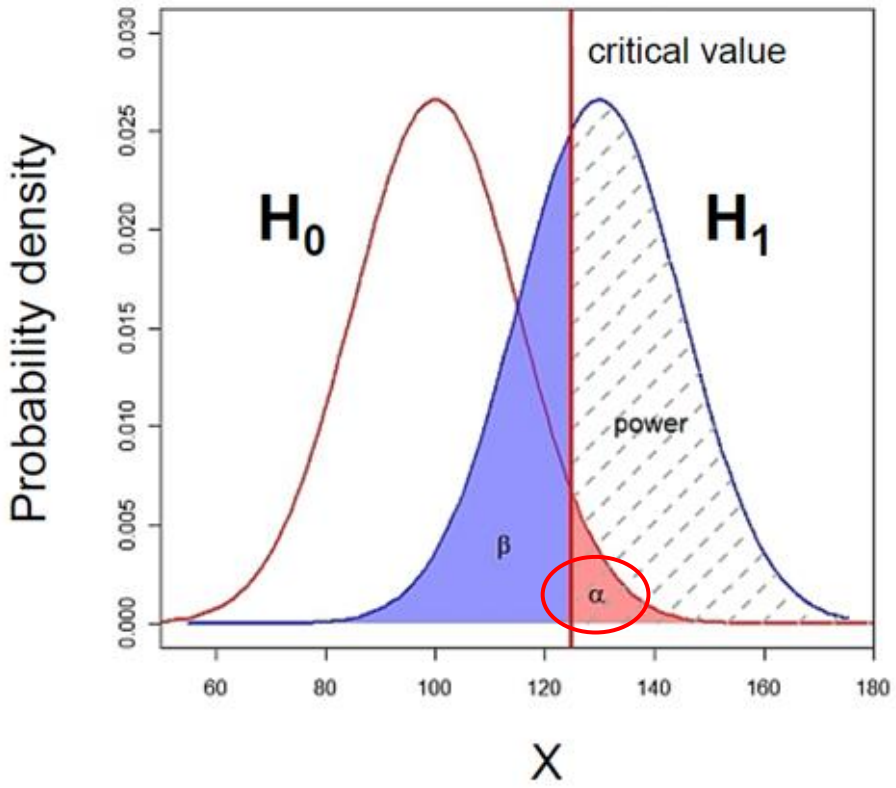
- Probability that the observed result occurs if H_0 is true
  - H_0 : **Null hypothesis** = absence of effect
  - H_1: **Alternative hypothesis** = presence of an effect
    - Statistics is all about rejecting the Null or not.

What does Power look like? Type I error (α)



- **Type I error** is the failure to reject a <u>true</u> H_0
  - <u>Claiming</u> an effect which is not there.
  - **α** : probability of making a Type I error
- **α** : the significance level, usually set at **0.05 or 5%**

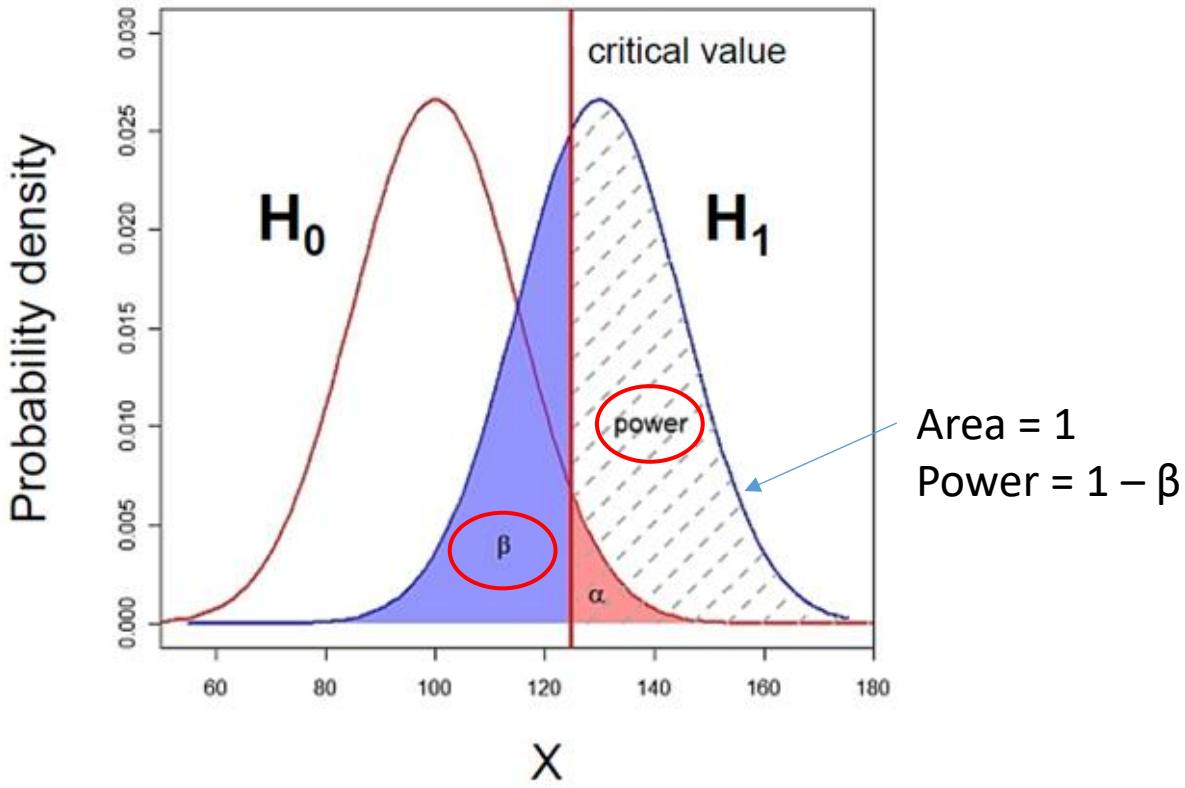
What does Power look like? Type I error (α) and the p-value



- **p-value**: probability that the observed statistic occurred by chance alone
  - probability that a difference as big as the one observed could be found even if there is no effect.

- **Statistical significance**: comparison between **α (=0.05)** and the **p-value**
  - **p-value < 0.05**: there is a significant difference ☺ (reject H_0)
  - **p-value > 0.05**: there is no significant difference ☹ (fail to reject H_0)

What does Power look like? Type II error (β) and Power

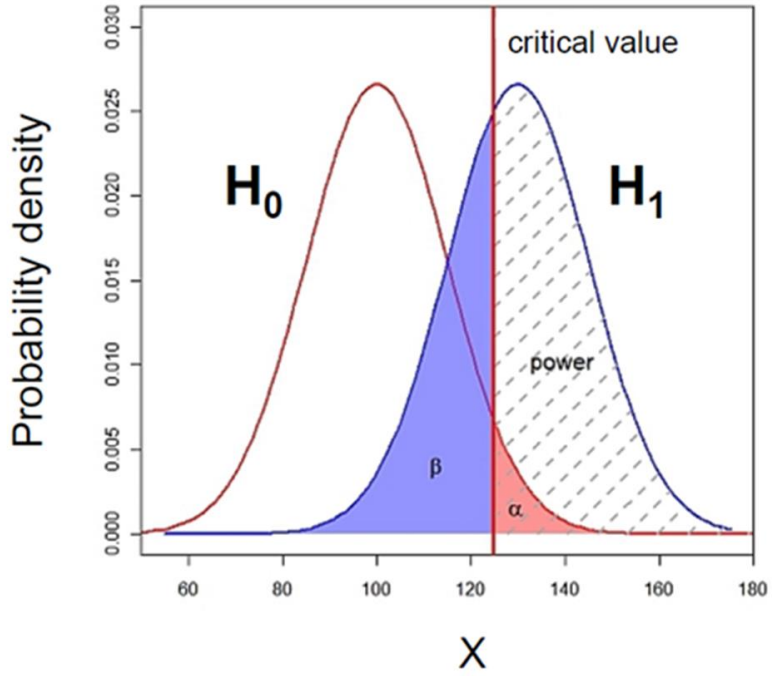


- **Type II error** (**β**) is the failure to reject a <u>false</u> H_0
  - <u>Missing</u> an effect which is really there
  - **β** : probability of making a Type II error

- **Power**: Probability of <u>detecting</u> an effect which is really there.
  = Probability of rejecting a false H_0
  - Direct relationship between **Power** and **Type II error**: **Power** = 1 – **β**

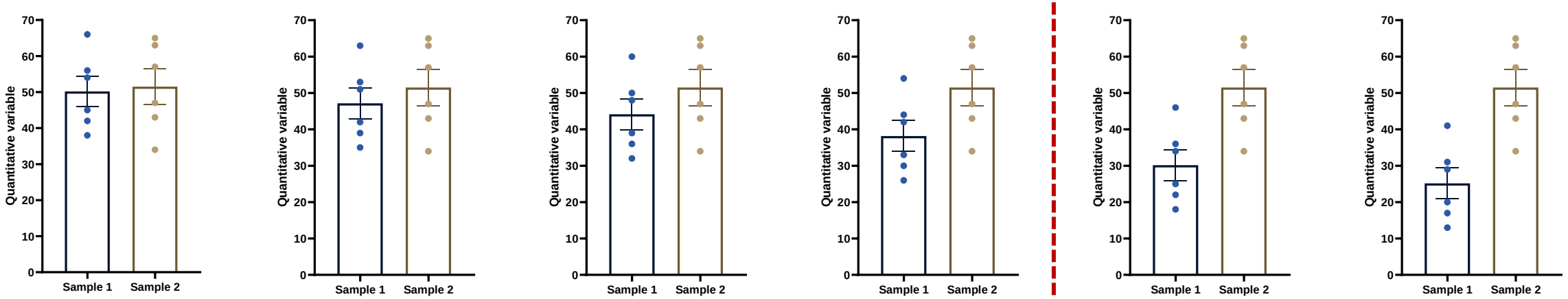
What does Power look like? Power = 80%

- General convention: 80% but could be more
- Means a true difference will be missed 20% of the time
  - If power = 0.8 then β = 1- power = 0.2 (20%)
- Jacob Cohen (1962):
  - Type I errors are 4x more serious than Type II errors:
    - **0.05 * 4 = 0.2**
- Compromise between power and sample size, e.g. for 2 group comparisons:
  - 90% power = +30% sample size
  - 95% power = +60% sample size



The critical value

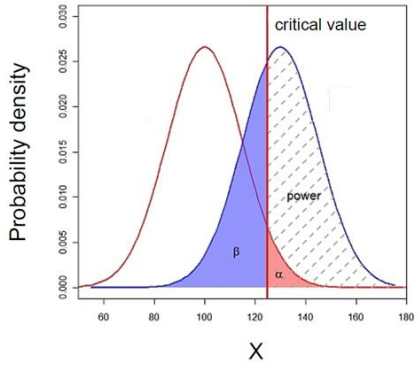
Critical value



Small difference → Big difference

Not significant: p>0.05

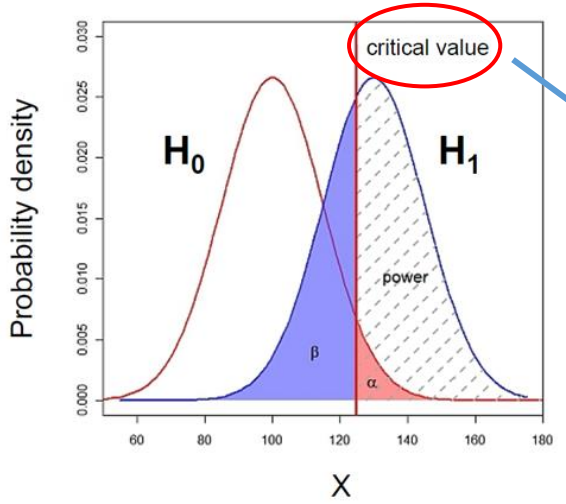
Significant: p<0.05



Critical value = size of difference + sample size + significance level

The critical value: size of difference + sample size + significance

Example with the *t*-test



Significance level

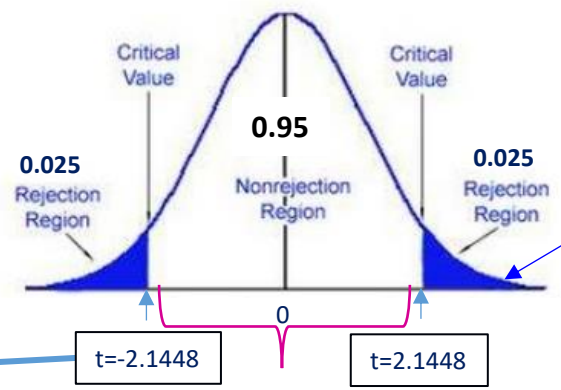
Degrees of freedom

df	0.20	0.10	0.05	0.02	0.01	0.001
1	3.0777	6.3138	12.7062	31.8205	63.6567	636.6192
2	1.8856	2.9200	4.3027	6.9646	9.9248	31.5991
3	1.6377	2.3534	3.1824	4.5407	5.8409	12.9240
4	1.5332	2.1318	2.7764	3.7469	4.6041	8.6103
5	1.4759	2.0150	2.5706	3.3649	4.0321	6.8688
6	1.4398	1.9432	2.4469	3.1427	3.7074	5.9588
7	1.4149	1.8946	2.3646	2.9980	3.4995	5.4079
8	1.3968	1.8595	2.3060	2.8965	3.3554	5.0413
9	1.3830	1.8331	2.2622	2.8214	3.2498	4.7809
10	1.3722	1.8125	2.2281	2.7638	3.1693	4.5869
11	1.3634	1.7959	2.2010	2.7181	3.1058	4.4370
12	1.3562	1.7823	2.1788	2.6810	3.0545	4.3178
13	1.3502	1.7709	2.1604	2.6503	3.0123	4.2208
14	1.3450	1.7613	2.1448	2.6245	2.9768	4.1405
15	1.3406	1.7531	2.1314	2.6025	2.9467	4.0728

Example: 2-tailed t-test with n=15 (df=14)

~ Difference

T Distribution



Diff "big enough"

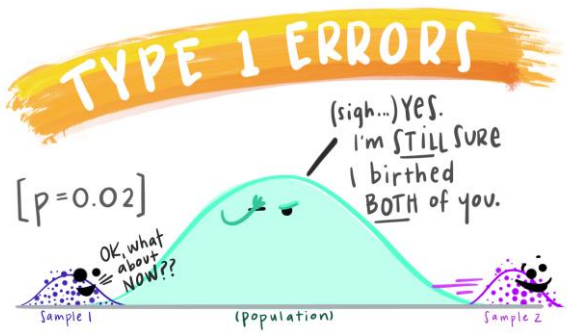
Difference "too small"

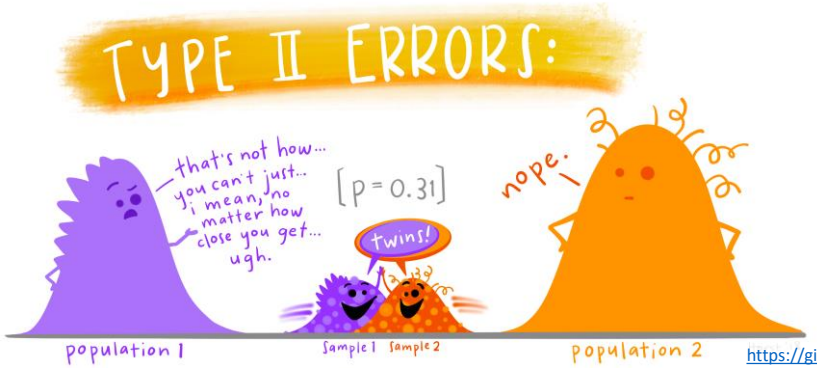
- In **hypothesis testing**:
  - **critical value** is compared to the **test statistic** to determine significance
  - Example of test statistic: t-value
- If **test statistic** > **critical value**: statistical significance and rejection of the null **hypothesis**
  - Example: t-value > critical t-value

To recap:

- The null hypothesis: H_0 = no effect
- The aim of a statistical test is to reject or not H_0.

Statistical decision	True state of H_0	
	H_0 True (no effect)	**H_0 False (effect)**
Reject H_0	**Type I error α** **False Positive**	Correct **True Positive**
Do not reject H_0	Correct **True Negative**	**Type II error β** **False Negative**





https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

Power analysis

The power analysis depends on the relationship between 6 variables:

- the significance level (5%)
- the desired power of the experiment (80%)
- the alternative hypothesis (i.e. one or two-sided test)
- the **difference** of biological interest
- the **variability** in the data (standard deviation)

} **Effect size**

- the **sample size**

Power analysis

The power analysis depends on the relationship between 6 variables:

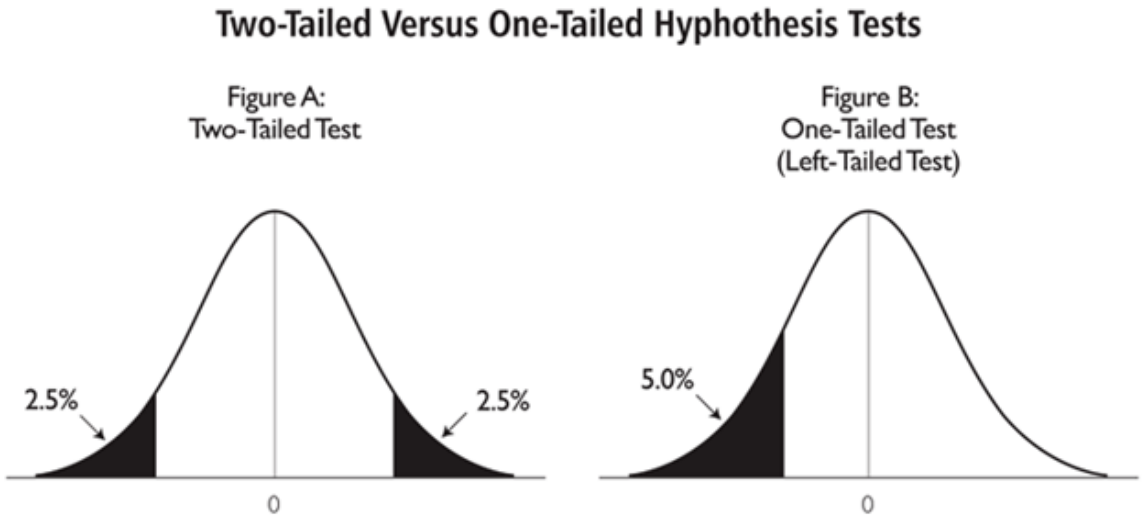
- the significance level (5%)
- the desired power of the experiment (80%)
- the alternative hypothesis (i.e. one or two-sided test)
- the **difference** of biological interest
- the **variability** in the data (standard deviation)

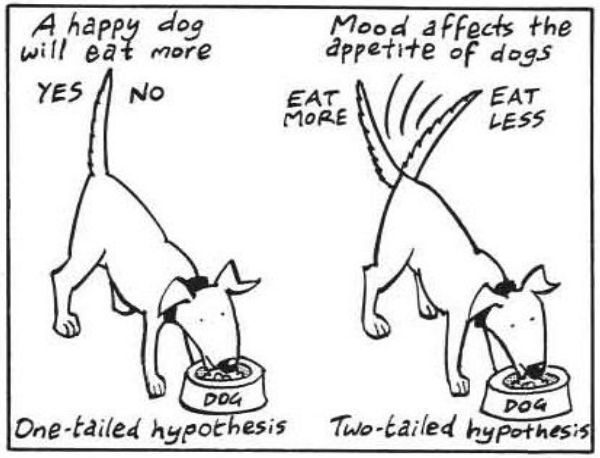
} **Effect size**

- the **sample size**

The alternative hypothesis: what is it?

- One-tailed or two-tailed test? One-sided or two-sided tests?





- Is the question:
  - Is the there a difference? → Two-tailed
  - Is it larger than or smaller than? → One-tailed

- Can rarely justify the use of a one-tailed test
- Two times easier to reach significance with one-tailed than two-tailed → suspicious reviewer!

Power analysis

The power analysis depends on the relationship between 6 variables:

- the significance level (5%)
- the desired power of the experiment (80%)
- the alternative hypothesis (i.e. one or two-sided test)
- the **difference** of biological interest
- the **variability** in the data (standard deviation)

} **Effect size**

- the **sample size**

The difference of biological interest

- Determined scientifically (not statistically)
  - Minimum meaningful effect of biological relevance (Minimum Effect of Interest, MEI)
- **How to determine it?**
  - Previous research, pilot study

The variability

- We need to have an idea of the standard deviation before we start the experiment
- **How to determine it?**
  - Data from previous research on WT, control or baseline

Effect size

Combination of absolute effect and variability

The effect size: how is it calculated?

- Depends on the type of difference and the data
  - Easy example: comparison between 2 means

$$\text{Effect Size} = \frac{[\text{Mean of experimental group}] - [\text{Mean of control group}]}{\text{Standard Deviation}}$$

Absolute difference

Variability

- Jacob Cohen defined small, medium and large effects for different tests – but **not recommended**

Test	Relevant effect size	Effect Size Threshold		
		Small	Medium	Large
t-test for means	d	0.2	0.5	0.8
F-test for ANOVA	f	0.1	0.25	0.4
t-test for correlation	r	0.1	0.3	0.5
Chi-square	w	0.1	0.3	0.5
2 proportions	h	0.2	0.5	0.8

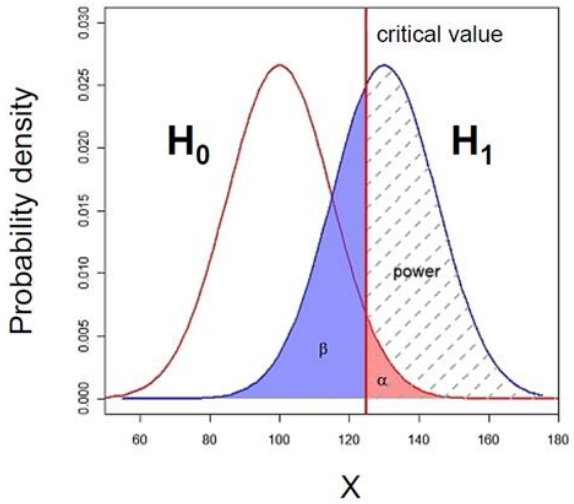
The effect size: how is it calculated?

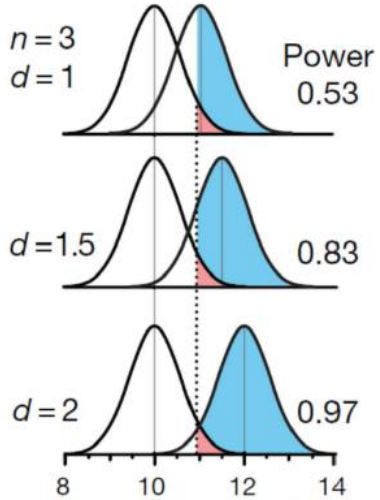
The absolute difference

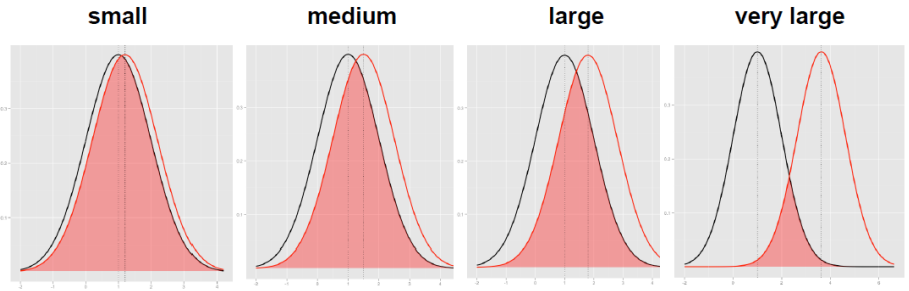
- The bigger the effect (the absolute difference), the bigger the power
  = the bigger the probability of picking up the difference

$$\text{Effect Size} = \frac{\text{[Mean of experimental group] – [Mean of control group]}}{\text{Standard Deviation}}$$

Absolute difference







http://rpsychologist.com/d3/cohend/

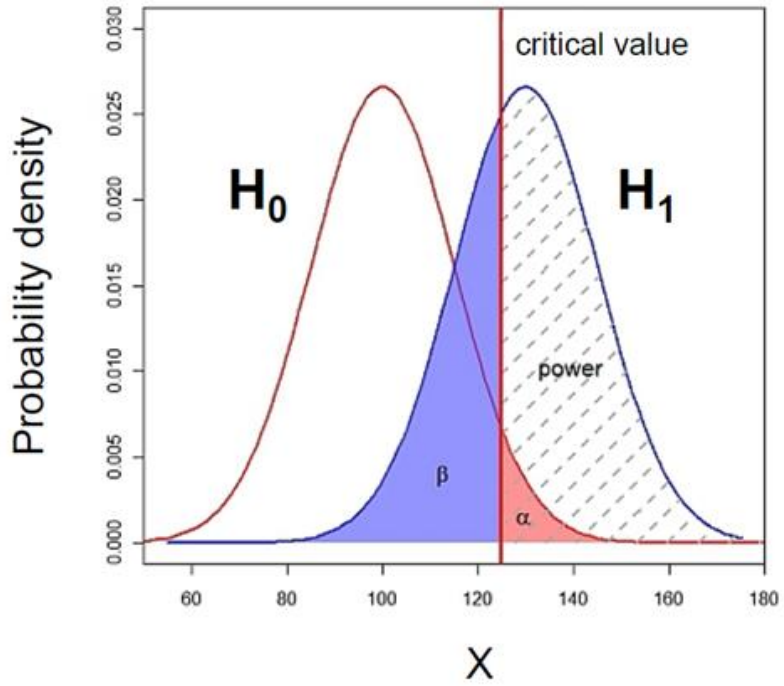
The effect size: how is it calculated?

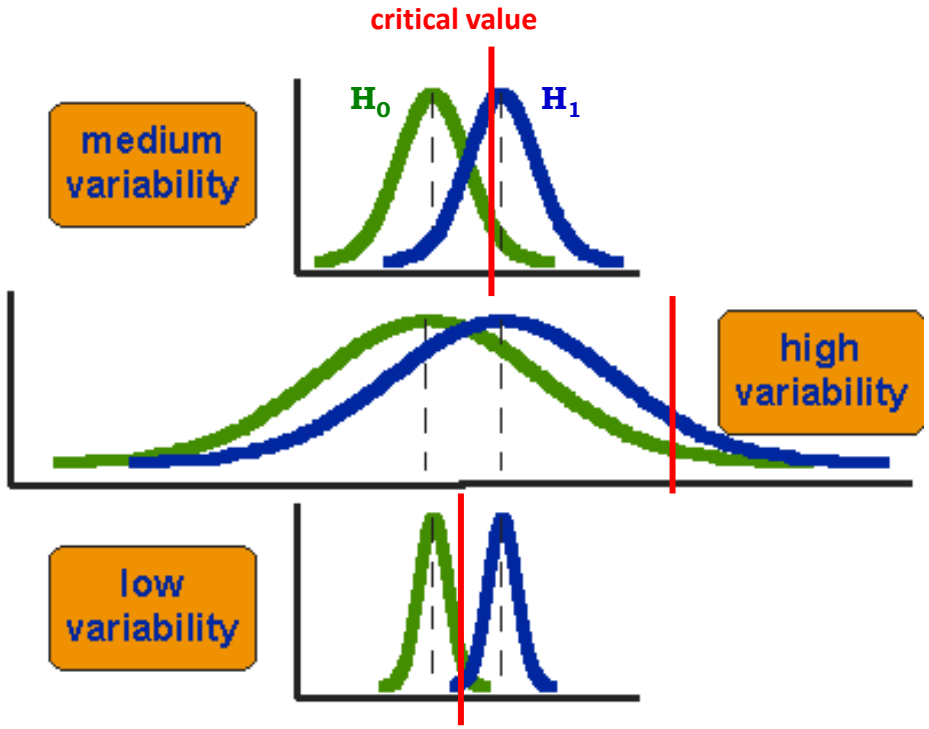
The standard deviation

- The bigger the variability of the data, the smaller the power

$$\text{Effect Size} = \frac{\text{[Mean of experimental group] – [Mean of control group]}}{\text{Standard Deviation}}$$

Variability





Power analysis

The power analysis depends on the relationship between 6 variables:

- the significance level (5%)
- the desired power of the experiment (80%)
- the alternative hypothesis (i.e. one or two-sided test)
- the **difference** of biological interest
- the **variability** in the data (standard deviation)

} **Effect size**

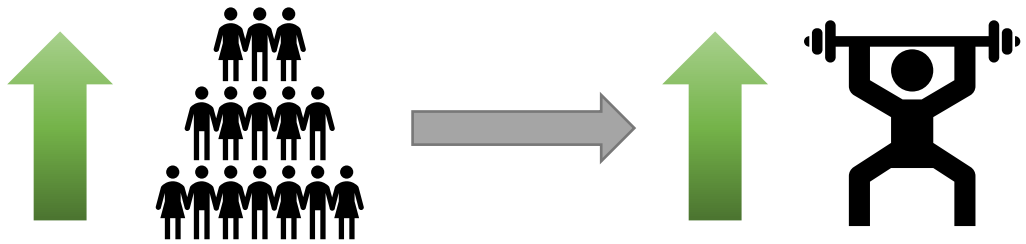
- the **sample size**

The sample size

- Most of the time, the output of a power calculation

- In reality it is difficult to reduce the variability in data, or the contrast between means
  - most effective way of improving power:
    - **increase the sample size**

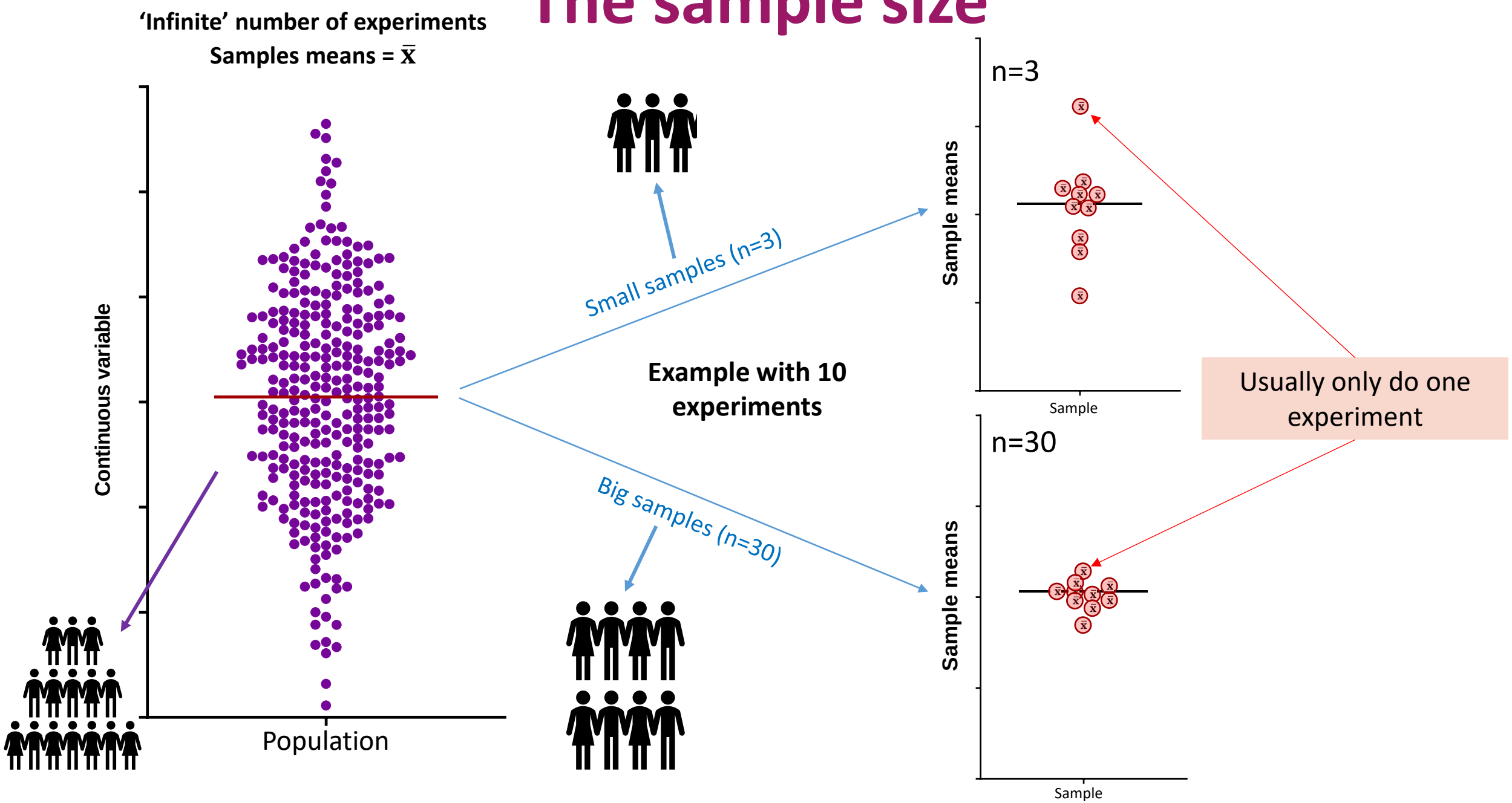
- **The bigger the sample, the bigger the power**
  - but how does it actually work?



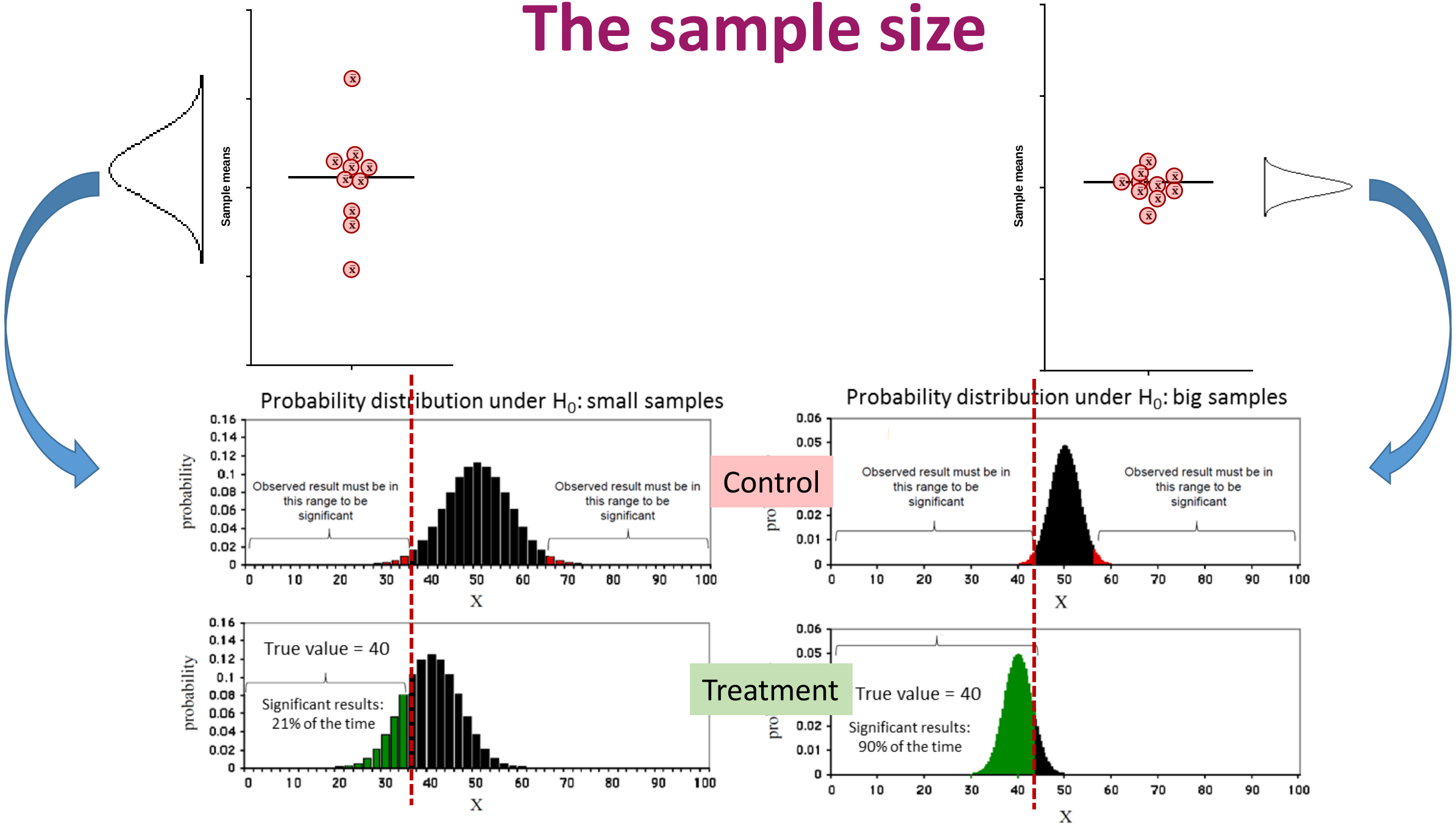
Samples and population

- We want to know about **whole population**
  - *All* people, *all* patients, *all* mice, *all* cells…
- Not possible to measure whole population
- So take a **representative sample**
- Make **inferences** about the population
- Larger samples more likely to be representative of the population

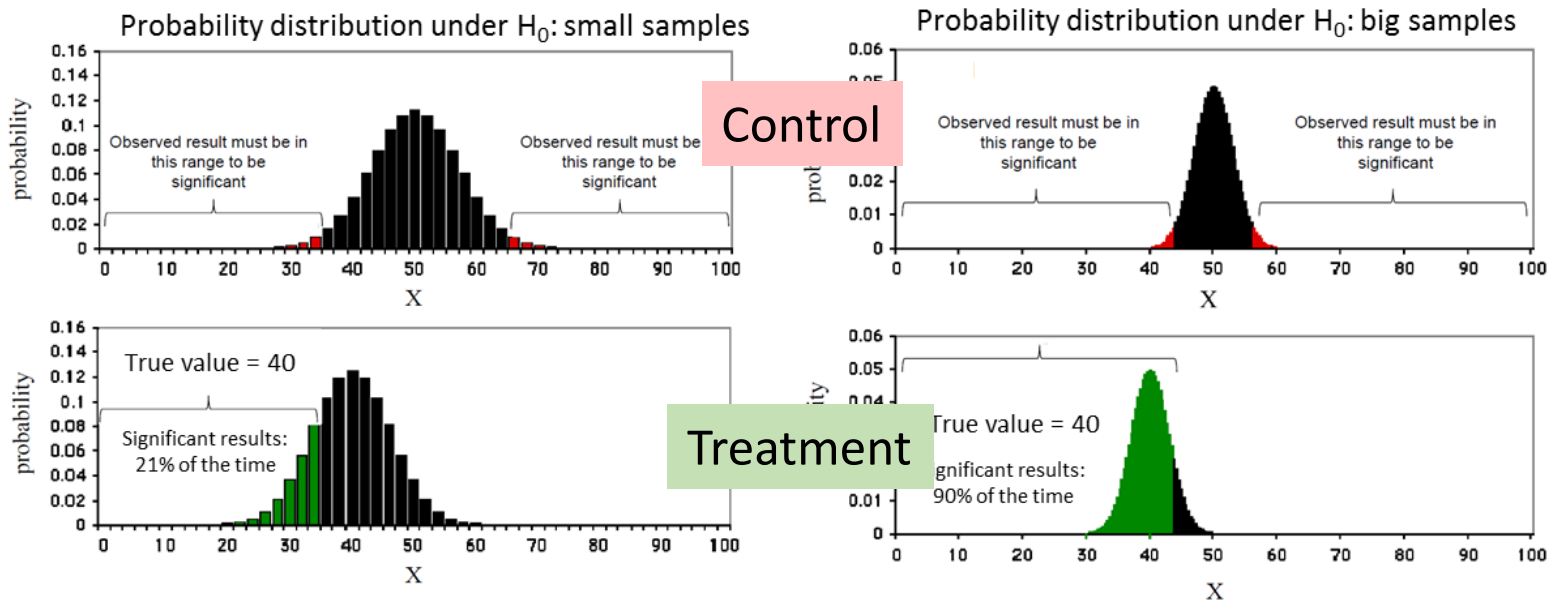
The sample size

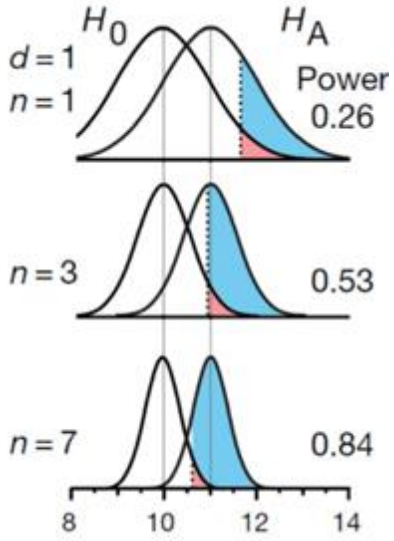


The sample size



The sample size



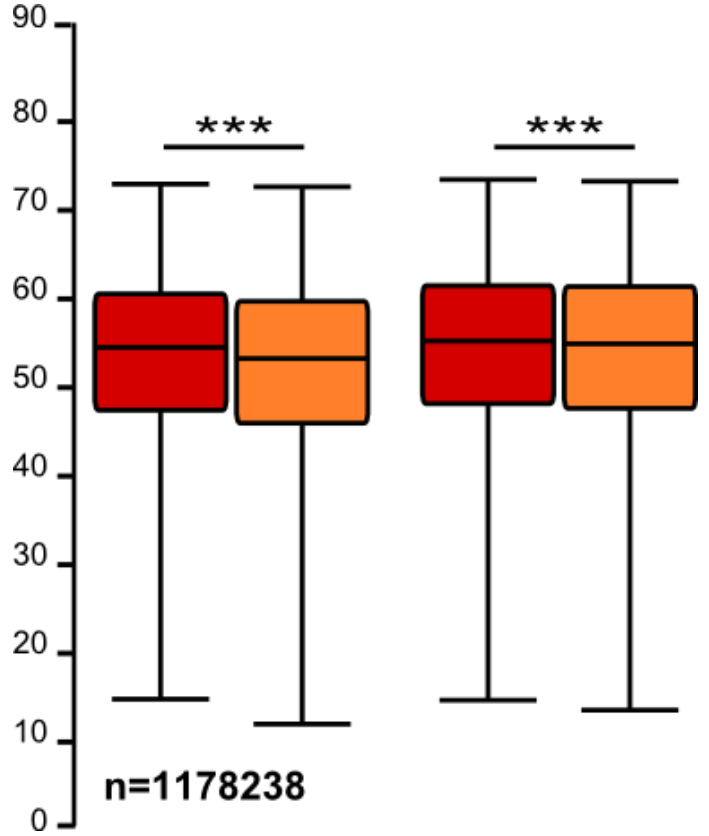


The sample size: the bigger the better?

- It takes huge samples to detect tiny differences but tiny samples to detect huge differences

- What if the tiny difference is meaningless?
  - Beware of **overpower**
  - Nothing wrong with the stats: it is all about interpretation of the results of the test

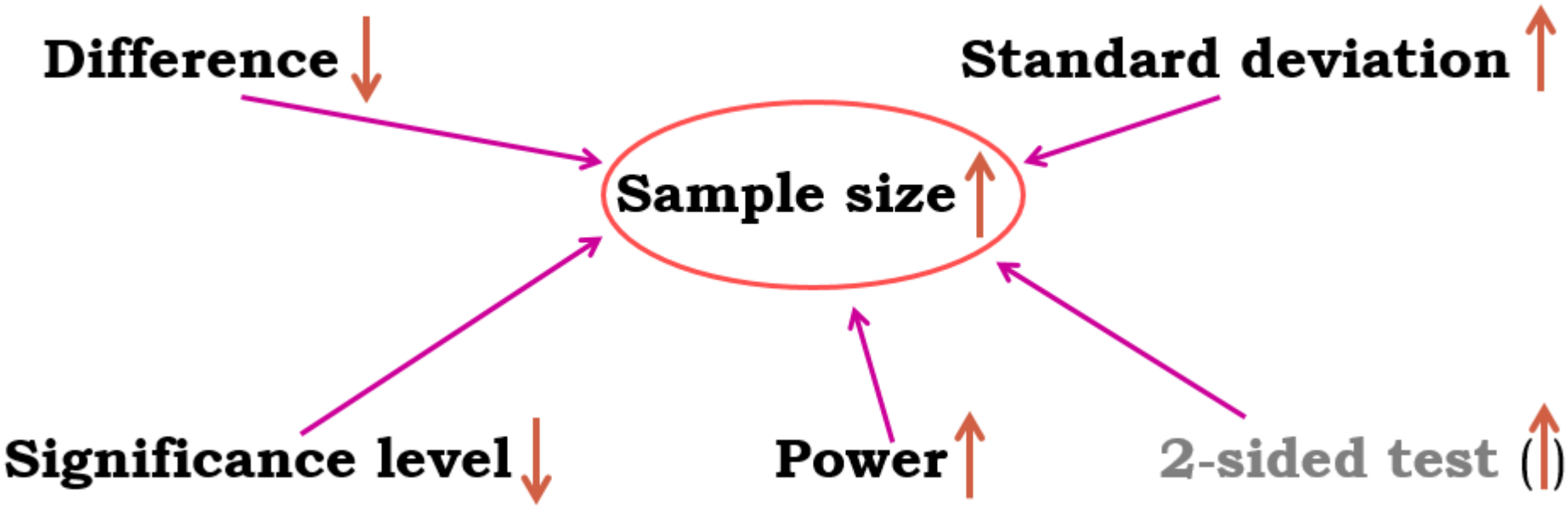
- Remember the important first step of power analysis:
  - **What is the effect size of biological interest?**



Power analysis

Typical question

What sample size do I need to have a 80% probability (**power**) to detect this particular effect (**difference** and **standard deviation**) at a 5% **significance level** using a **2-sided test**?



Power analysis

- Fix any **five of the variables**, a mathematical relationship is used to estimate the **sixth**

Difference of biological interest

+ Variability in the data (standard deviation)

+ Desired power of the experiment (80%)

+ Significance level (5%)

+ Alternative hypothesis (i.e. one or two-sided test)

Appropriate sample size

Power analysis

- **Good news**:

there are packages that can do the power analysis for you, providing you have some prior knowledge of the key parameters!

Use **R Help** to find out how to use the functions

- e.g. `?power.prop.test` in the console

- **R**
- G*Power

```
power.prop.test(n=NULL, p1=NULL, p2=NULL,
        sig.level=NULL, power=NULL,
        alternative=c("two.sided", "one.sided"))
power.t.test(n=NULL, delta=NULL, sd=1, sig.level=NULL,
        power=NULL,
        type=c("two.sample", "one.sample", "paired"),
        alternative=c("two.sided", "one.sided"))
```

Power Analysis

Comparing 2 proportions (Fisher's exact test)

Exercise:

- Scientists have come up with a solution that may reduce the number of lions being shot by farmers in Africa:
  - Painting eyes on cows' bottoms.
- Early trials suggest that lions are less likely to attack livestock when they think they're being watched
  - Fewer livestock attacks could help farmers and lions co-exist more peacefully.

- Pilot study over 6 weeks:
  - 3 out of 39 unpainted cows were killed by lions, none of the 23 painted cows from the same herd were killed.
- **Tasks**:
  - Do you think the observed effect is meaningful to the extent that such a 'treatment' should be applied?
    Consider ethics, economics, conservation …
  - Run a power calculation to find out how many cows should be included in the study.
    - Clue 1: `power.prop.test()`
    - Clue 2: exactly one of the parameters must be passed as NULL, and that parameter is determined by the others

http://www.sciencealert.com/scientists-are-painting-eyes-on-cows-butts-to-stop-lions-getting-shot

Power Analysis

Comparing 2 proportions

Exercise 1: *Answer*

- Pilot study over 6 weeks:
  - 3 out of 39 unpainted cows were killed by lions, none of the 23 painted cows from the same herd were killed.

```
power.prop.test(n=NULL,
                p1=3/39,
                p2=0,
                sig.level=0.05,
                power=0.8,
                alternative="two.sided")
```

```
     Two-sample comparison of proportions power calculation

              n = 96.92364
             p1 = 0.07692308
             p2 = 0
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

Providing the preliminary results are to be trusted, to be able to pick up such a difference between the 2 groups, with a **power of 80%** and a **significance level of 5%**, we will need at least **97 cows in each group**.

Article | Open Access | Published: 07 August 2020

Artificial eyespots on cattle reduce predation by large carnivores

Cameron Radford, John Weldon McNutt, Tracey Rogers, Ben Maslen & Neil Jordan

Communications Biology **3**, Article number: 430 (2020) | Cite this article

49k Accesses | **1** Citations | **2040** Altmetric | Metrics

Abstract

Eyespots evolved independently in many taxa as anti-predator signals. There remains debate regarding whether eyespots function as diversion targets, predator mimics, conspicuous startling signals, deceptive detection, or a combination. Although eye patterns and gaze modify human behaviour, anti-predator eyespots do not occur naturally in contemporary mammals. Here we show that eyespots painted on cattle rumps were associated with reduced attacks by ambush carnivores (lions and leopards). Cattle painted with eyespots were significantly more likely to survive than were cross-marked and unmarked cattle, despite all treatment groups being similarly exposed to predation risk. While higher survival of eyespot-painted cattle supports the detection hypothesis, increased survival of cross-marked cattle suggests an effect of novel and conspicuous marks more generally. To our knowledge, this is the first time eyespots have been shown to deter large mammalian predators. Applying artificial marks to high-value livestock may therefore represent a cost-effective tool to reduce livestock predation.

a artificial eyespots (bicolour as pictured, or white/yellow inner only, or black outer only, for maximum contrast depending on cattle coat colour). **b** cross-marked procedural control (black or white depending on coat colour for contrast). **c** unmarked control. Images provided by C.R.

https://www.nature.com/articles/s42003-020-01156-0.pdf

Power Analysis

Comparing 2 means (t-test)

Exercise:

- We want to know whether male and female coyotes differ in size
- No data from a pilot study but we have found some information in the literature:
  - In a study run in similar conditions as in the one we intend to run, **male coyotes** were found to measure: **92cm +/- 7cm (SD)**
- We expect a **5% difference** between sexes
  = smallest biologically meaningful difference
- **Task**:
  - Run a power calculation to find out how many coyotes should be included in the study
  - Using `power.t.test()`

Power Analysis

Comparing 2 means (t-test)

Independent t-test

***A priori* Power analysis**

Example case:

From a similar study, male coyotes were found to measure:
92cm+/- 7cm (SD)

You expect a **5% difference** between sexes with similar variability in the female sample

```
power.t.test(
   n = NULL, delta = NULL, sd = NULL,
   sig.level = NULL, power = NULL,
   type = "two.sample" or "one.sample"
      or "paired",
   alternative = "two.sided" or
      "one.sided")
```

↓

```
power.t.test(delta = 92-87.4, sd = 7,
      sig.level = 0.05, power = 0.8)
```

↓

```
     Two-sample t test power calculation

              n = 37.33624
          delta = 4.6
             sd = 7
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

You need a sample size of n=76 (2*38)

Unequal sample sizes

- No simple trade-off – if need 2 groups of 30 → 20 and 40 = decreased power
  - **Unbalanced design = bigger total sample**
- **Solution 1** *(old school)*:

  Step 1: power calculation for equal sample size

  Step 2: adjustment

Where:

- N = total sample size in unbalanced design
- n = total sample size in balanced design
- n_1 = group 1 size
- n_2 = group 2 size
- k = ratio n_2/n_1

$$N = \frac{2n(1+k)^2}{4k}$$

$$n_1 = \frac{N}{(1+k)}$$

$$n_2 = \frac{kN}{(1+k)}$$

Cow example:

- Balanced design: **n = 97**
- If unpainted group is 2 times bigger than painted (k=0.5):

$$N = \frac{2 \times 97 \times (1+0.5)^2}{4 \times 0.5} = 218.25 \approx 219$$

Unpainted butts **(n_1)=146** Painted butts **(n_2)=73**

- Balanced design: **n = 2*97 = 194**
- Unbalanced design: **n= 73+146 = 219**

Unequal sample sizes

- **Solution 2:** Use R
- In practice with R, `# MESS package #`
  - Comparing 2 proportions **with unequal n**: `power_prop_test()`
  - Comparing 2 means **with unequal n**: `power_t_test()`

```
power_prop_test(n=NULL,
      p1=3/39,  p2=0, sig.level=0.05,
      power =0.8, ratio = 0.5)
```

```
    Two-sample comparison of proportions power calculation with unequal sample sizes

              n = 160.01567, 80.00783
             p1 = 0.07692308
             p2 = 0
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is vector of number in each group
```

Also consider anything else that might impact final numbers, e.g. if likely to lose some samples during experiment

Different methods give slightly different sample sizes:

- Using adjustment
  - Unpainted (n_1) = 146
  - Painted (n_2) = 73
  - **Total sample = 219**
- Using R:
  - Unpainted (n_1) = 161
  - Painted (n_2) = 81
  - **Total sample = 242**

Non-parametric tests

- Do not assume data come from a Gaussian/normal distribution
  - Based on ranking values from low to high
  - Almost always less powerful
- Proper power calculations need to specify which kind of distribution we are dealing with – not easy
- Non-parametric usually do not require more than 15% additional subjects compared to parametric
- Crude rule of thumb:
  - **Compute sample size required for a parametric test and add 15%**

What happens if we ignore the power of a test?

- **Misinterpretation** of the results

- Never ever interpret p-values without context!
  - **Significant p-value (<0.05)**: but what is the difference?
    - >= smallest meaningful difference: **exciting effect**
    - < smallest meaningful difference: essentially a false positive/type 1 error
      - Too big sample, overpowered – difference has **no biological relevance**
  - **Not significant p-value (>0.05)**: but how big was the sample?
    - Big enough = enough power: **no effect**
    - Not big enough = underpowered: potentially a false negative/type 2 error
      - Too small sample – **potentially miss a meaningful difference**

Exercise 1

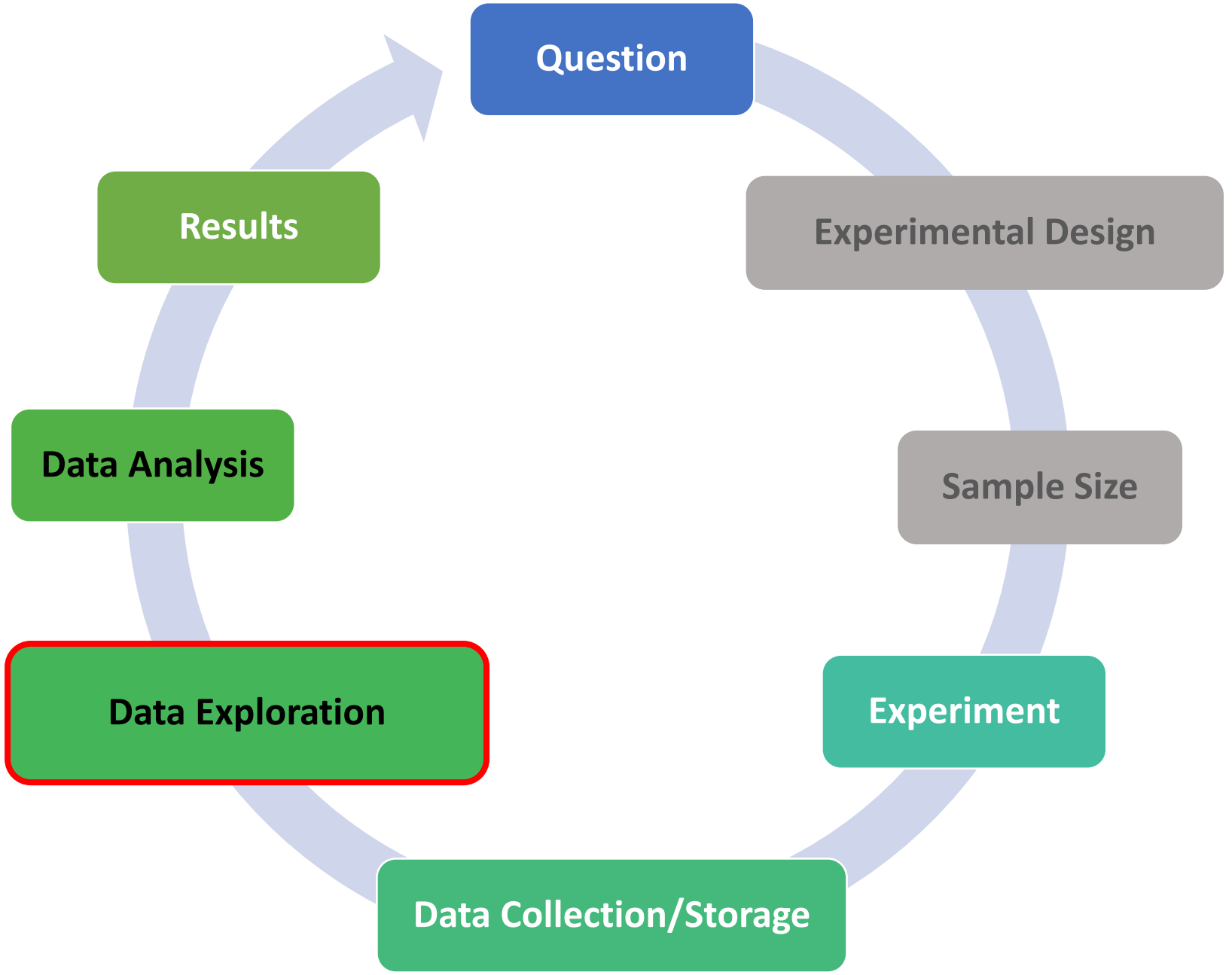
Babraham Institute

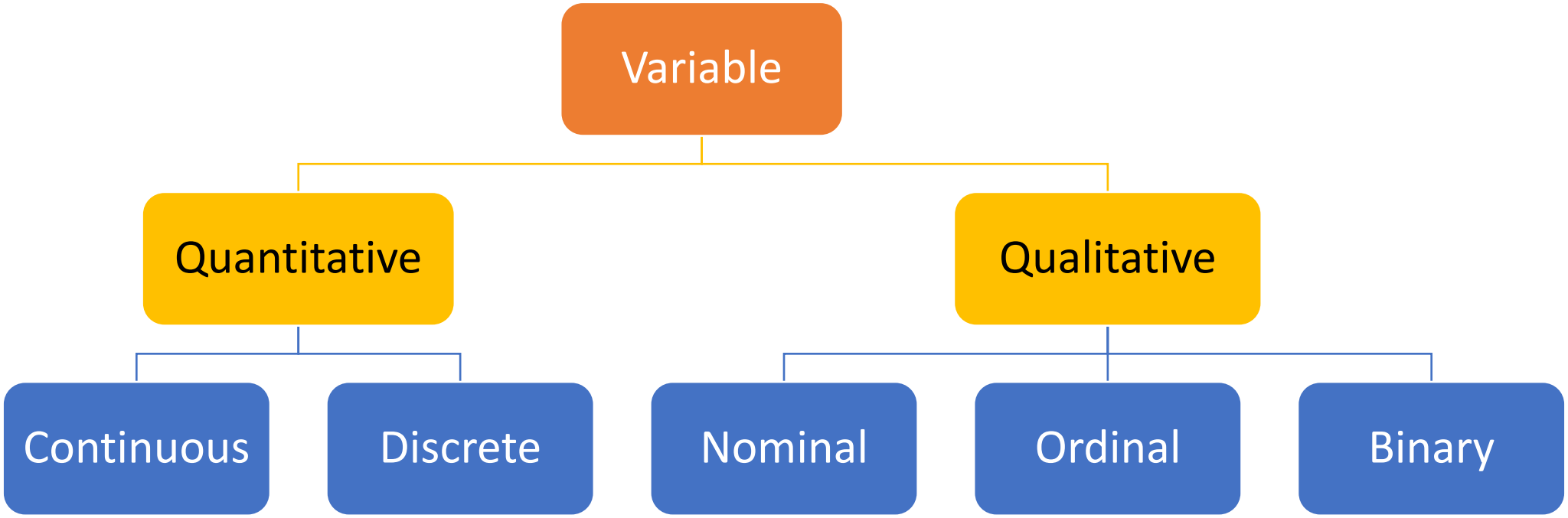
Descriptive Stats and Data Exploration

Hayley Carr & Anne Segonds-Pichon

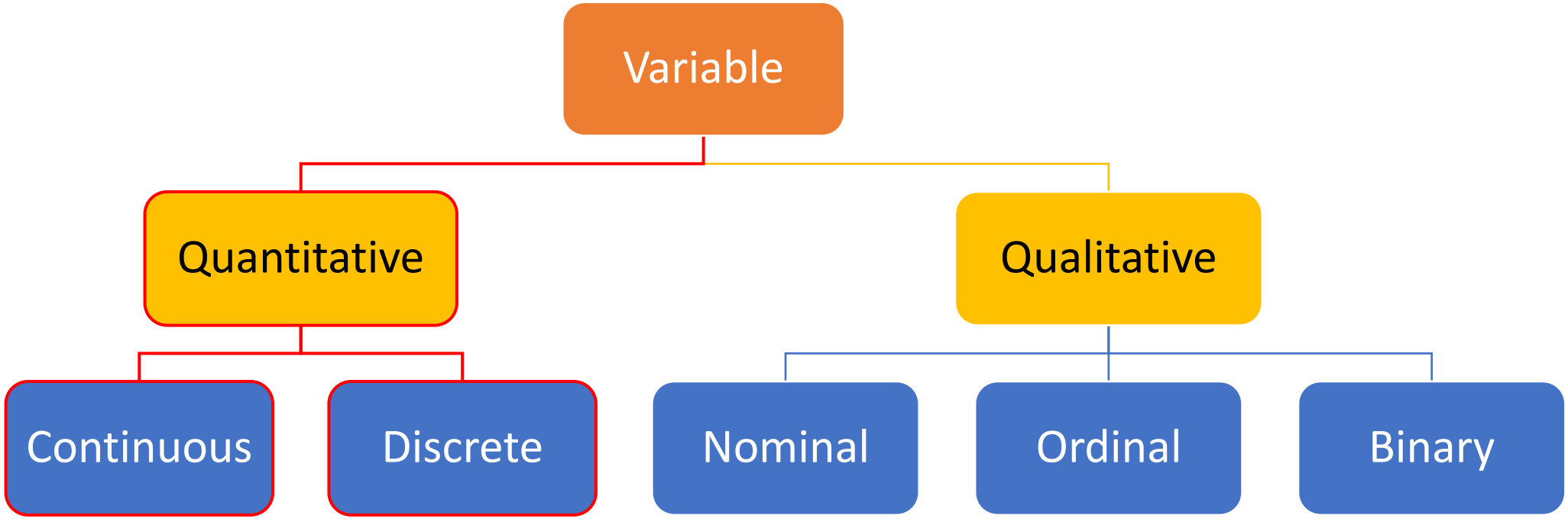
v2024-05

Babraham Bioinformatics


Question
Experimental Design
Sample Size
Experiment
Data Collection/Storage
Data Exploration
Data Analysis
Results



https://github.com/allisonhorst/stats-illustrations#other-stats-artwork



https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

Quantitative data

- They take numerical values (units of measurement)

- Discrete: obtained by counting
  - Example: number of students in a class
  - values vary by finite specific steps
- Continuous: obtained by measuring
  - Example: height of students in a class
  - any values – can have decimal places

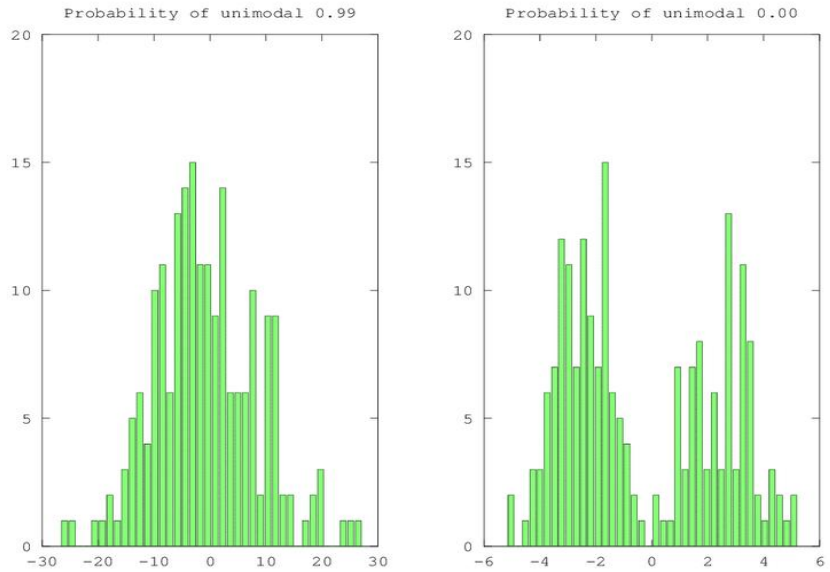
https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

- They can be described by a series of parameters:
  - Mean, variance, standard deviation, standard error and confidence interval

Measures of central tendency

Mode and Median

- **Mode:** most commonly occurring value in a distribution



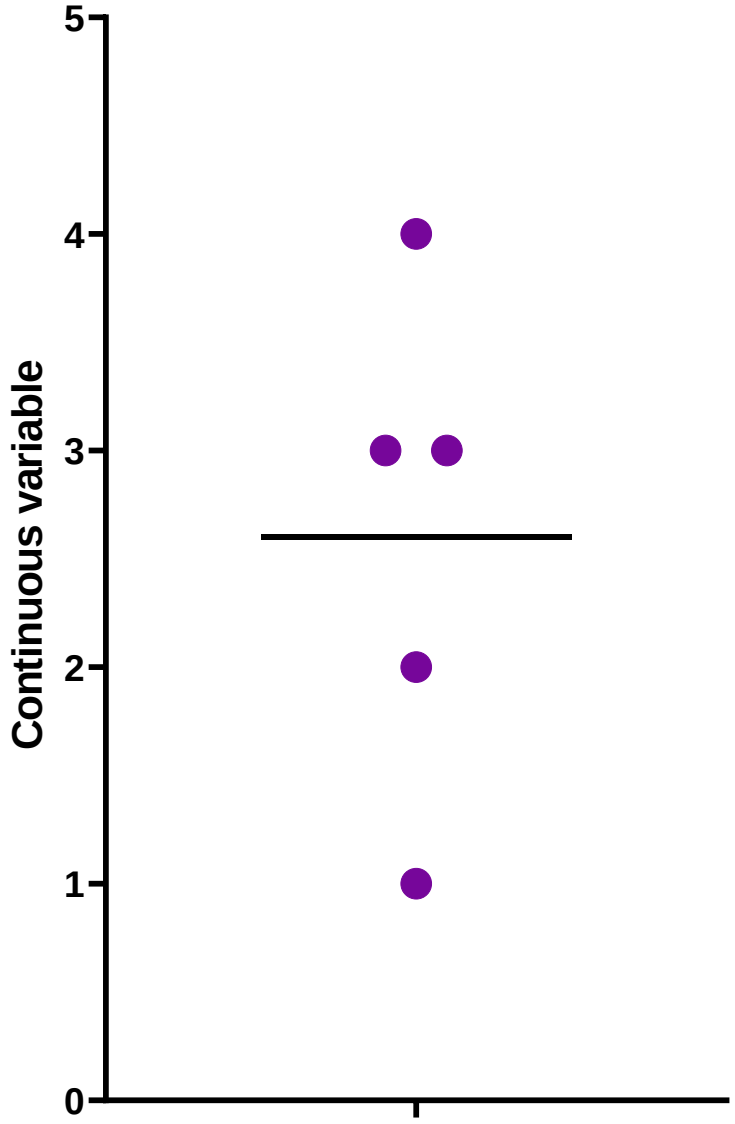
- **Median:** value exactly in the middle of an ordered set of numbers

Example 1: 18 27 34 52 54 59 61 68 78 82 85 87 91 93 100, Median = 68
Example 2: 18 27 27 34 52 52 59 61 68 68 85 85 85 90, Median = 60

Measures of central tendency

Mean

- Definition: average of all values in a column
- Example: mean of: 1, 2, 3, 3 and 4 = (1+2+3+3+4)/5 = 2.6
- The mean is a model because it summaries the data
- How do we know that it is an accurate model?
  - Difference between the real data and the model created



Measures of dispersion

- Calculate the magnitude of the differences between each data and the mean

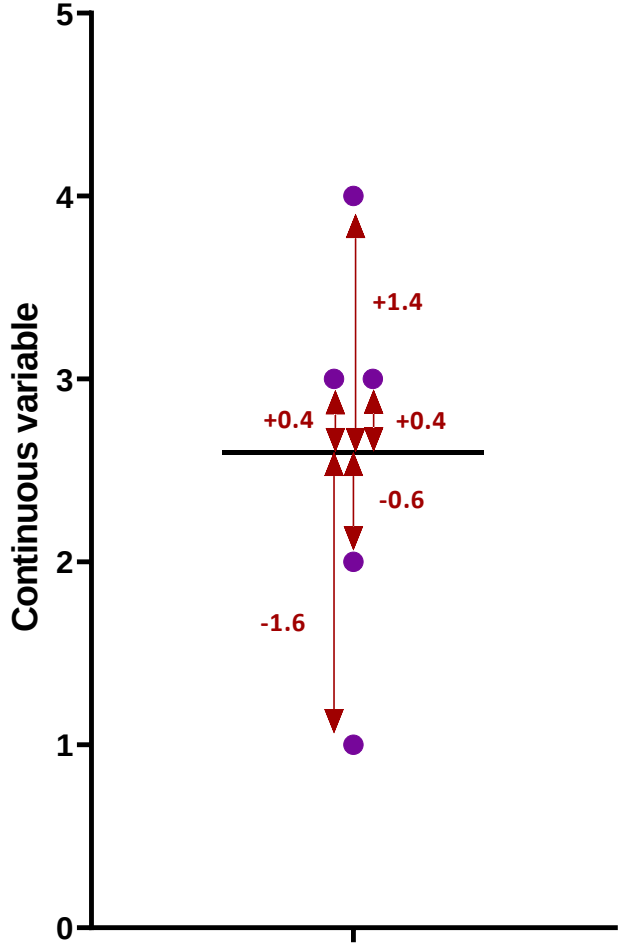
- Total error = sum of differences

  $= \Sigma(x_i - \overline{x})$ = -1.6 - 0.6 + 0.4 + 0.4 + 1.4 = 0

  No errors: positive and negative cancel each other out

- To solve that problem we square the errors

  → Sum of squared errors (SS)



Sum of Squared errors (SS)

- Sum of squared errors = $(SS) = \Sigma(x_i - \overline{x})^2$

  $= (-1.6)^2 + (-0.6)^2 + (0.4)^2 + (0.4)^2 + (1.4)^2$

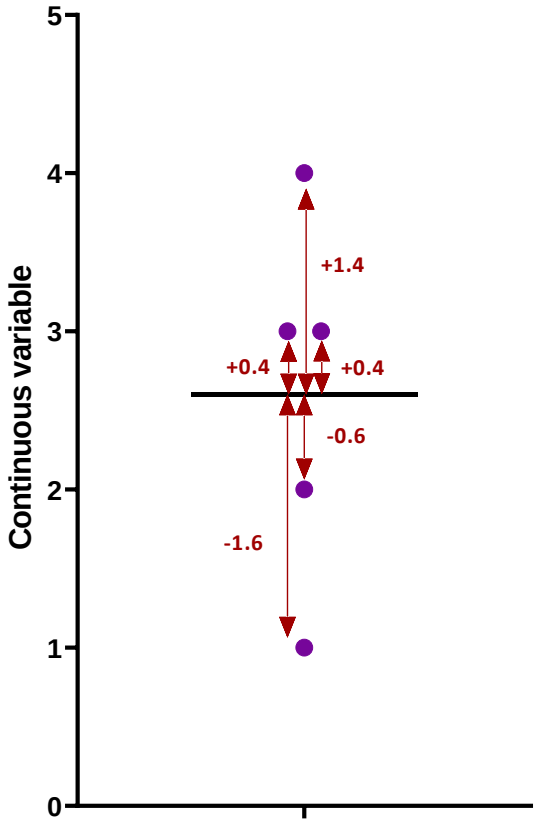
  $= 5.20$

- Good measure of the accuracy of the model

- Depends on amount of data: larger sample → larger SS

  - Account for number of observations (N) by dividing SS by N-1 (degrees of freedom)

    → the **variance** (S^2) = SS/N-1



Degrees of freedom

$$variance\ (s^2) = \frac{SS}{N-1} = \frac{\Sigma\ (x_i - \bar{x})^2}{N-1}$$

- To calculate the variance, we need the mean
- If we know the mean, we do not need all the values in the sample to calculate the variance
- Example: Sample: n = 5, Mean ($\bar{x}$) = 2.6
  - 2.6 x 5 – (1+2+3+3) = 4

$$\frac{1+2+3+3+4}{5} = 2.6$$

- Once we know the mean, we only need to know the first 4 numbers (N-1) and we can calculate the last number

Degrees of freedom

$$variance\ (s^2) = \frac{SS}{N-1} = \frac{\Sigma\ (x_i - \bar{x})^2}{N-1}$$

- The last (n^{th}) value in the sample is no longer independent, **is not free.**

n – 1 degrees of freedom

- Because we know the mean, the variance does not depend on all of the values of the sample, only on n-1 of the values

Variance and standard deviation

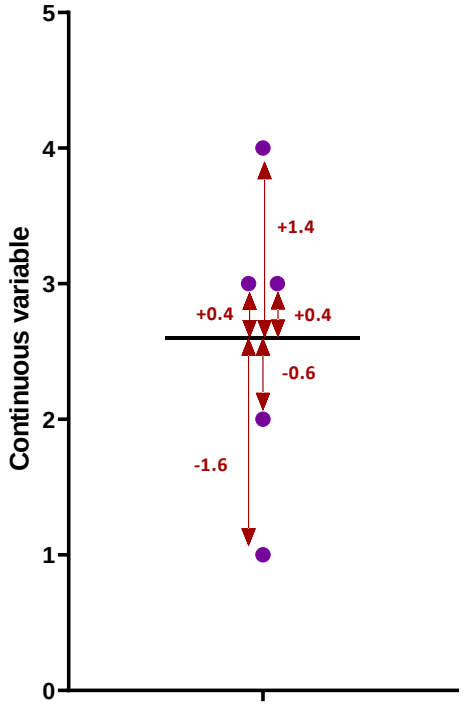
- $variance\ (s^2) = \frac{SS}{N-1} = \frac{\Sigma\,(x_i - \overline{x})^2}{N-1} = \frac{5.20}{4} = 1.3$

- Problem with variance: in squared units
  - Take the square root to get the same unit as the original measure

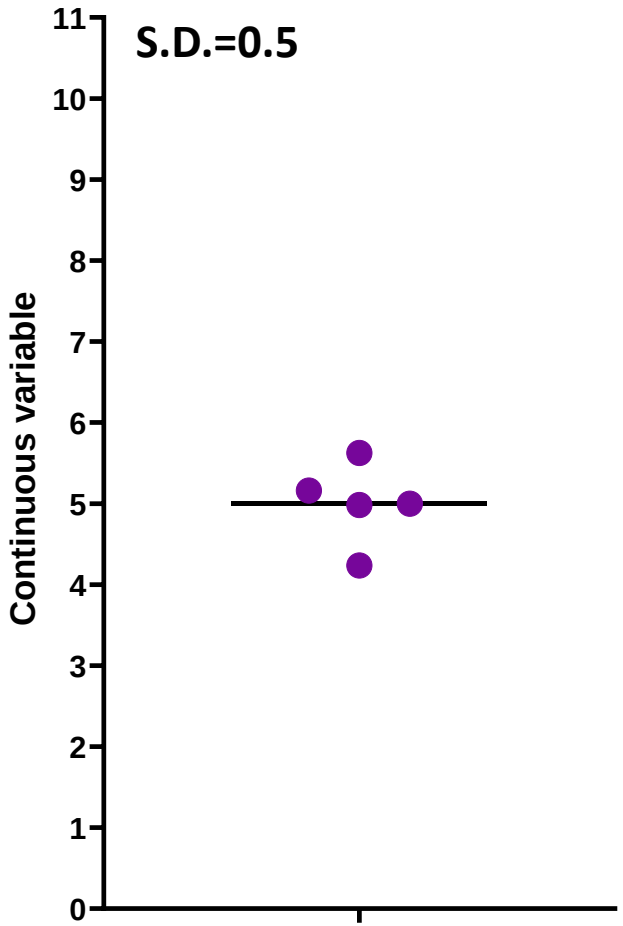
    → the **standard deviation**

    S.D. = √(SS/N-1) = √(s^2) = s = $\sqrt{1.3} = 1.14$

- SD = a measure of how well the mean represents the data.



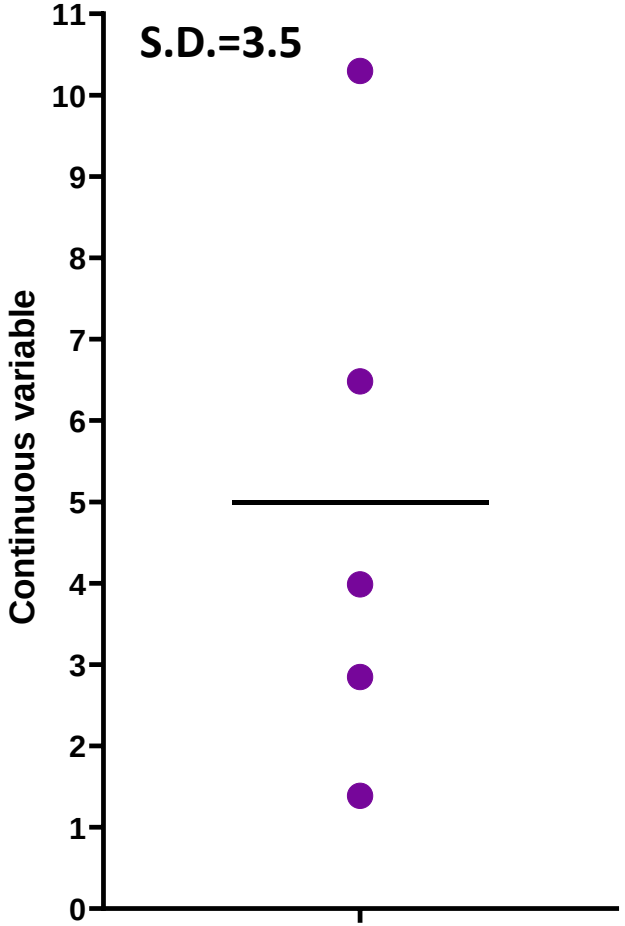
Standard deviation



Small S.D.

data close to the mean:

good fit of the data

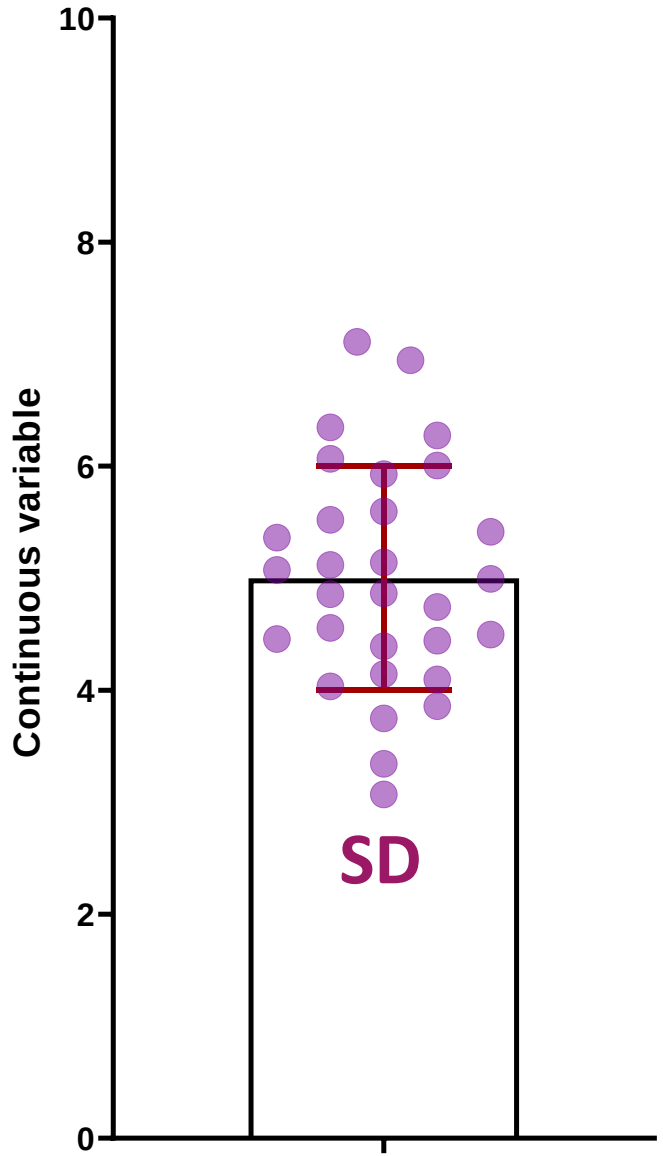


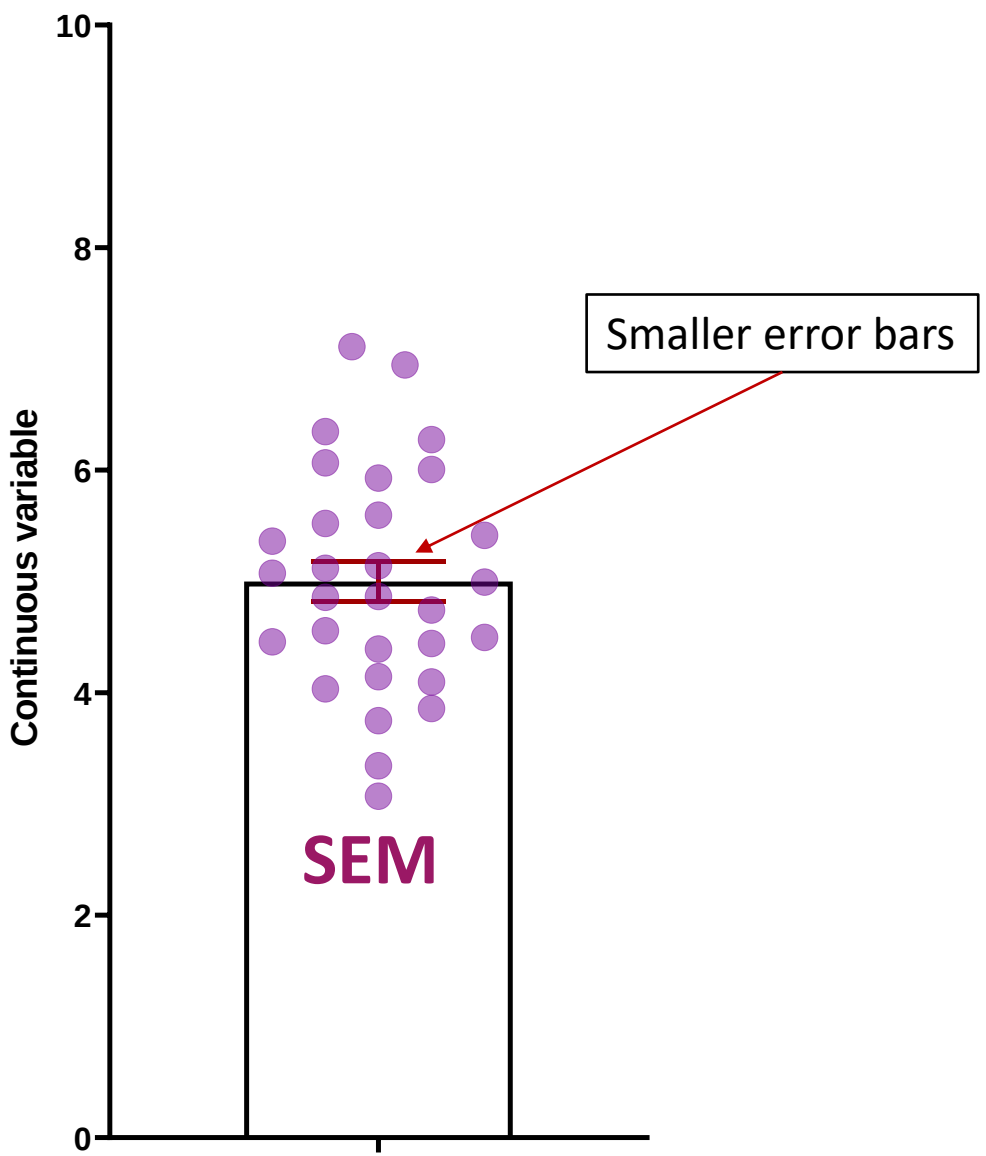
Large S.D.

data distant from the mean:

not an accurate representation

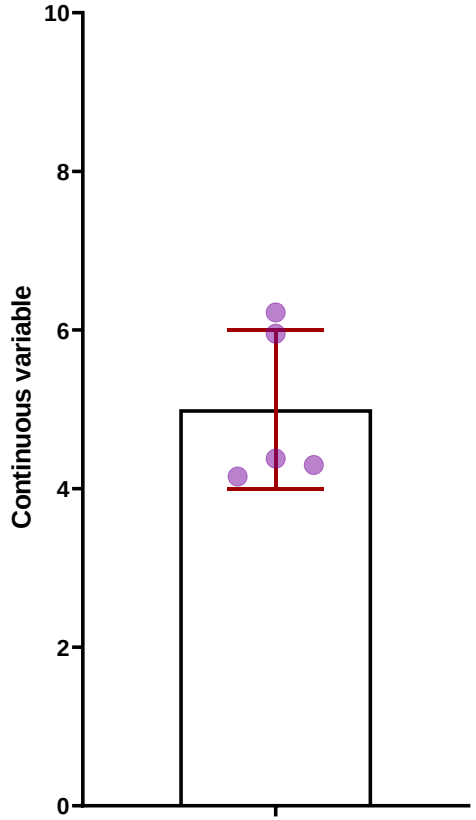
Standard Deviation (SD) or Standard Error Mean (SEM)?

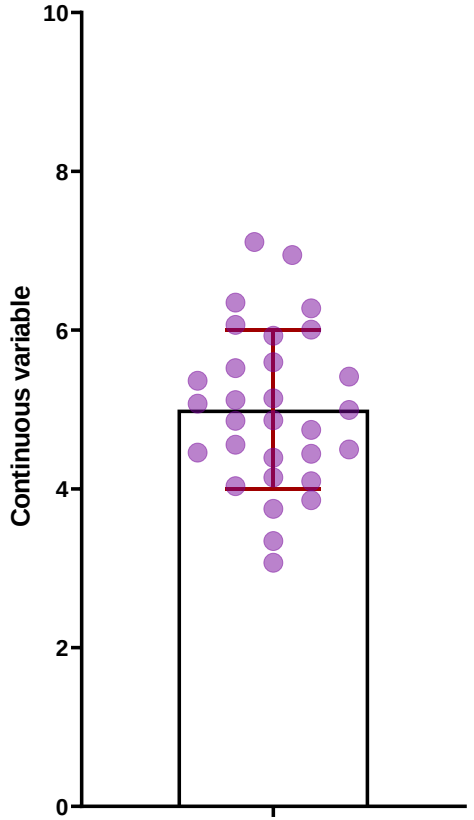




Standard Deviation

- The SD quantifies how much the values vary from one another
  → **scatter or spread**
- Does not change predictably as you acquire more data

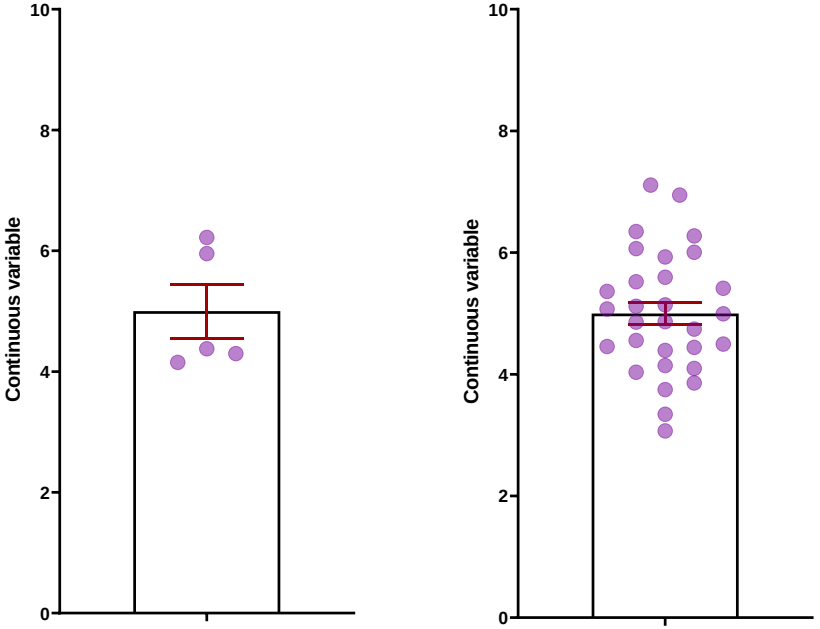




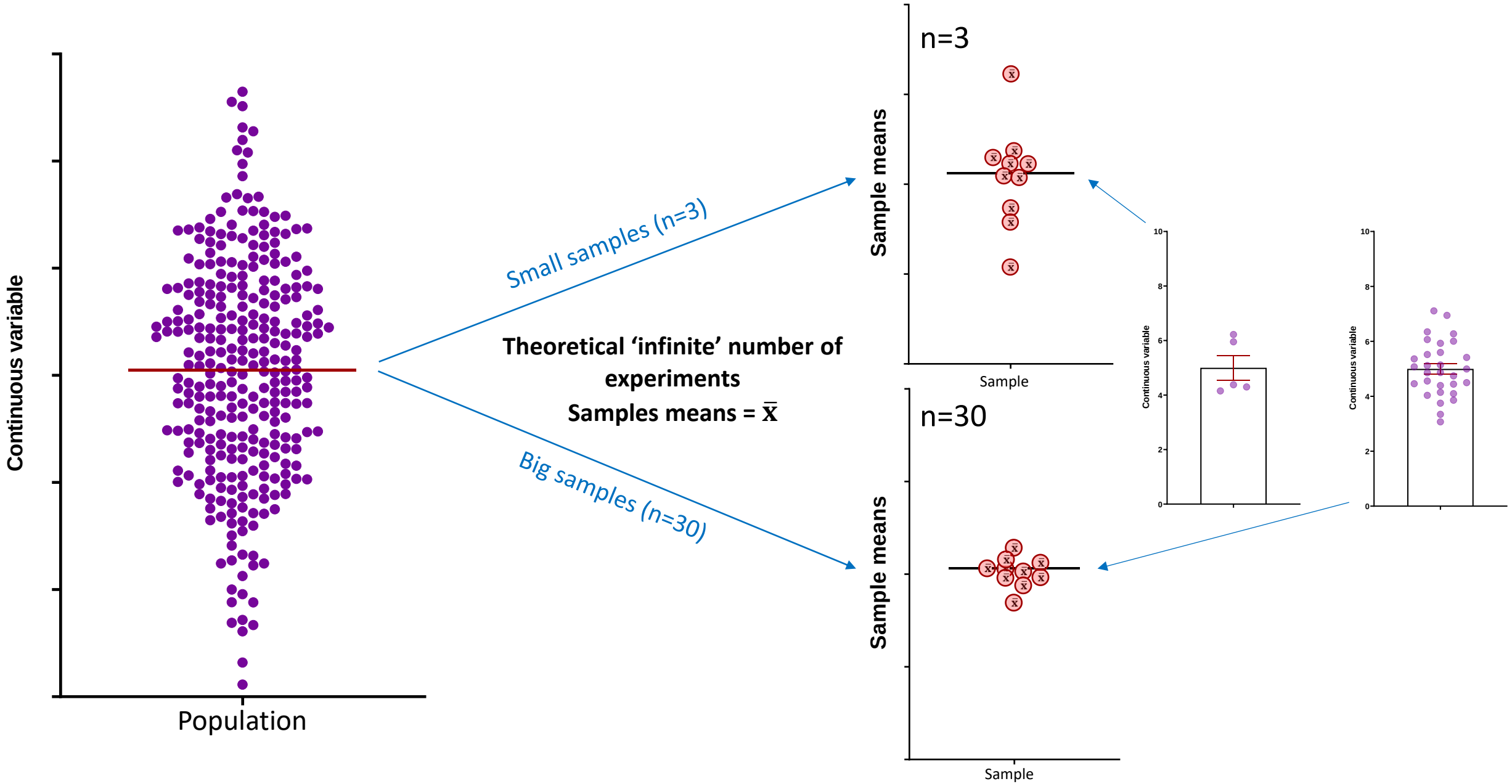
Standard Error of the Mean

$$\text{SEM} = \frac{\text{SD}}{\sqrt{N}}$$

- The SEM quantifies how accurately we know the **true mean** of the population
  - Takes into account: SD + sample size
  - Make **inferences** about the population
- The SEM gets smaller as the sample gets larger
  - Mean of a large sample likely closer to true mean than mean of a small sample



The SEM and the sample size



SD or SEM ?

Whichever you choose make sure to report it accurately!

- If we want to show the variation among values:

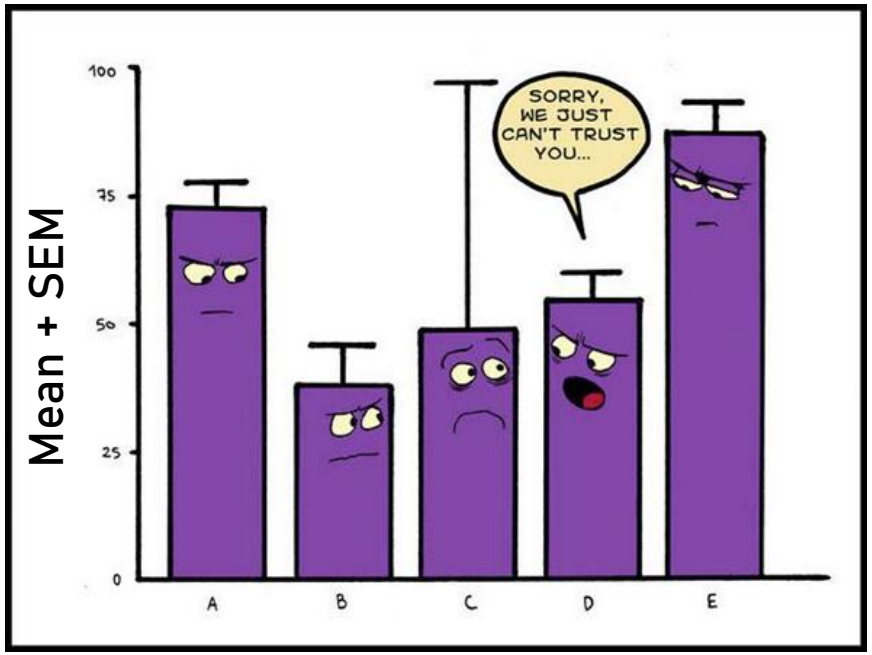
  → Report the **SD**

- If we want to show how precisely we have determined the population mean:

  → Report the **SEM**

- Preferably show all data points and the SEM

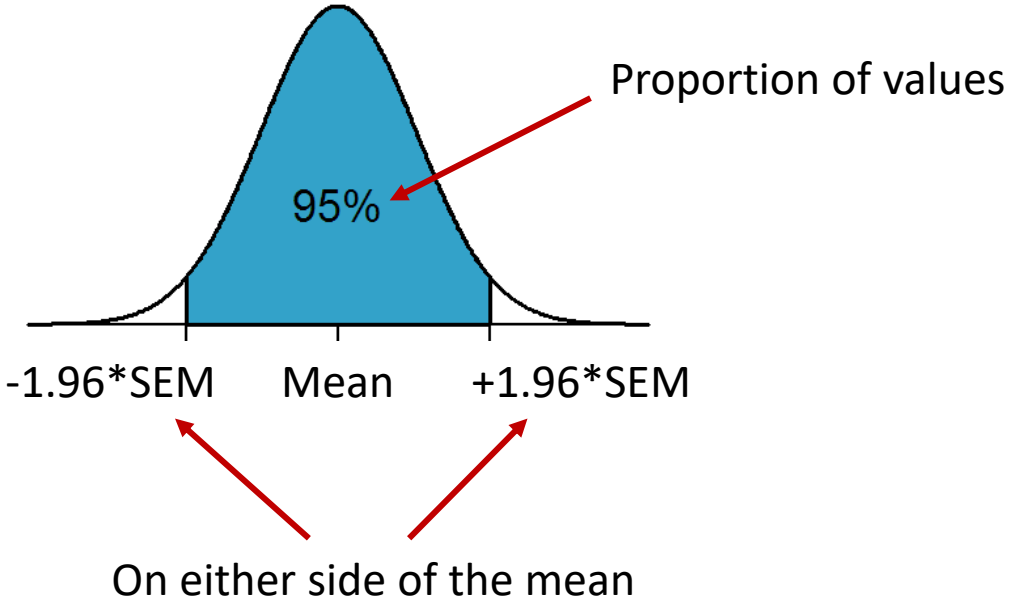
  → Both variation and precision

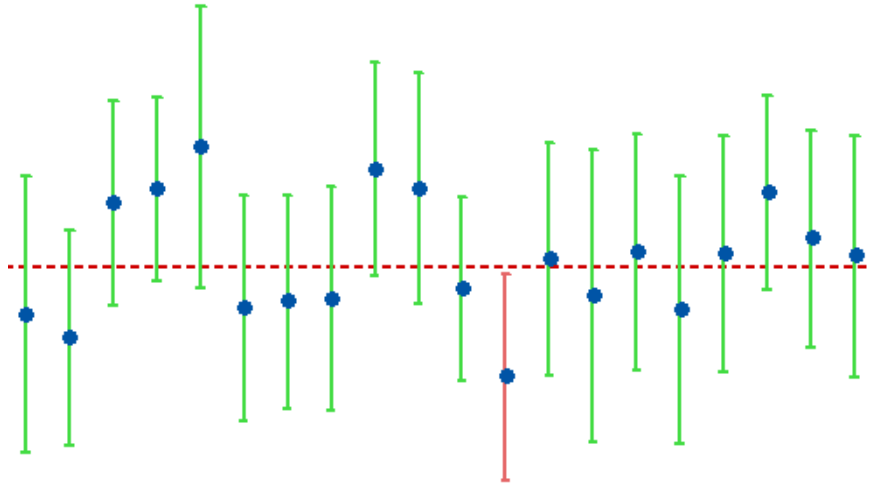


https://lymielynn.medium.com/a-little-closer-to-cooks-distance-e8cc923a3250

Confidence interval

- Range of values that we can be 95% confident contains the true mean of the population
  - Limits of 95% CI: [Mean - 1.96*SEM; Mean + 1.96*SEM] (SEM = SD/√N)
- On average 19/20 experiments include the population mean

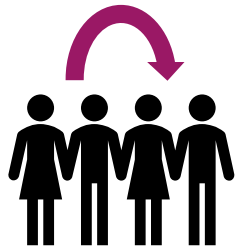




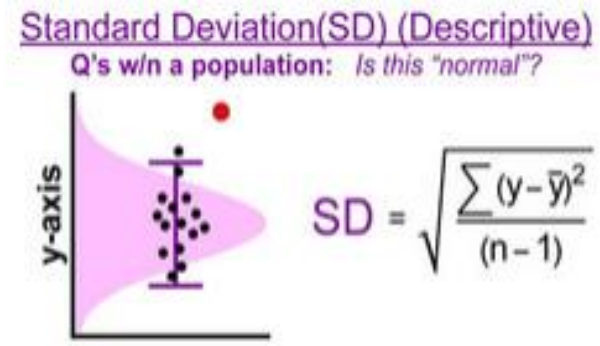
https://statisticsbyjim.com/hypothesis-testing/confidence-interval/

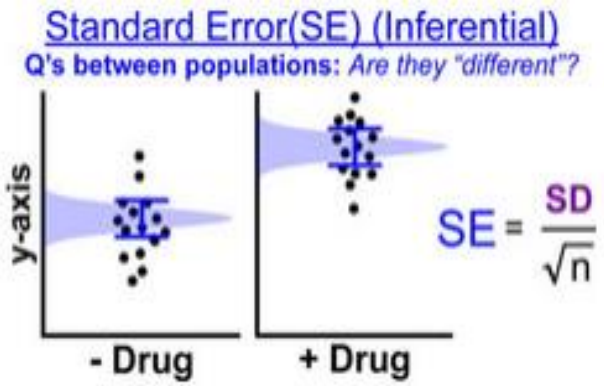
To recap

- The Standard Deviation is **descriptive**
  - Just about the sample

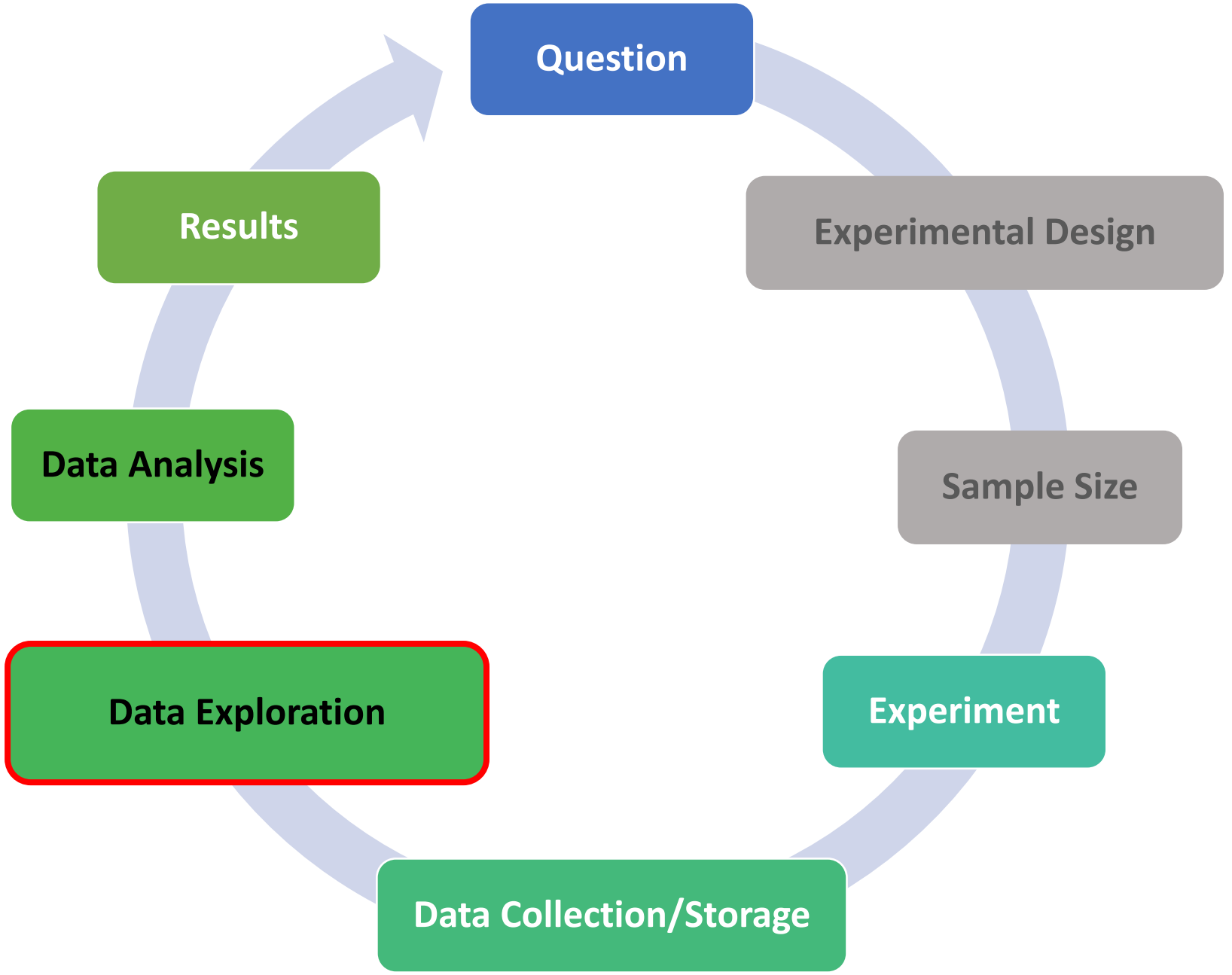


- The Standard Error and the Confidence Interval are **inferential**
  - Sample → General Population

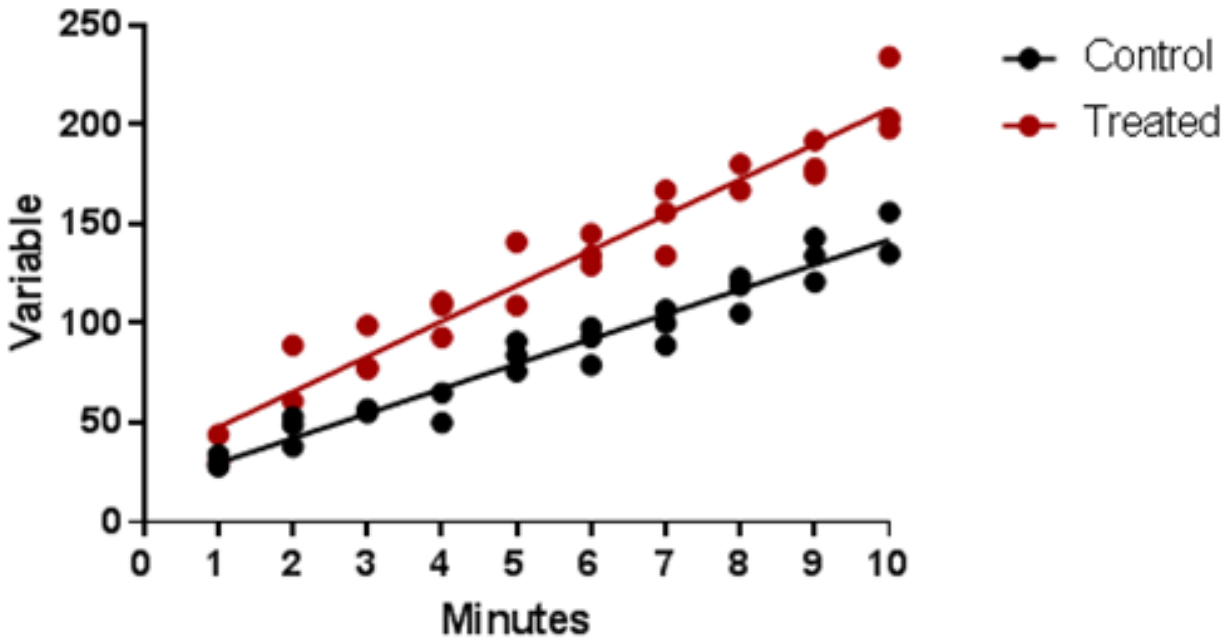


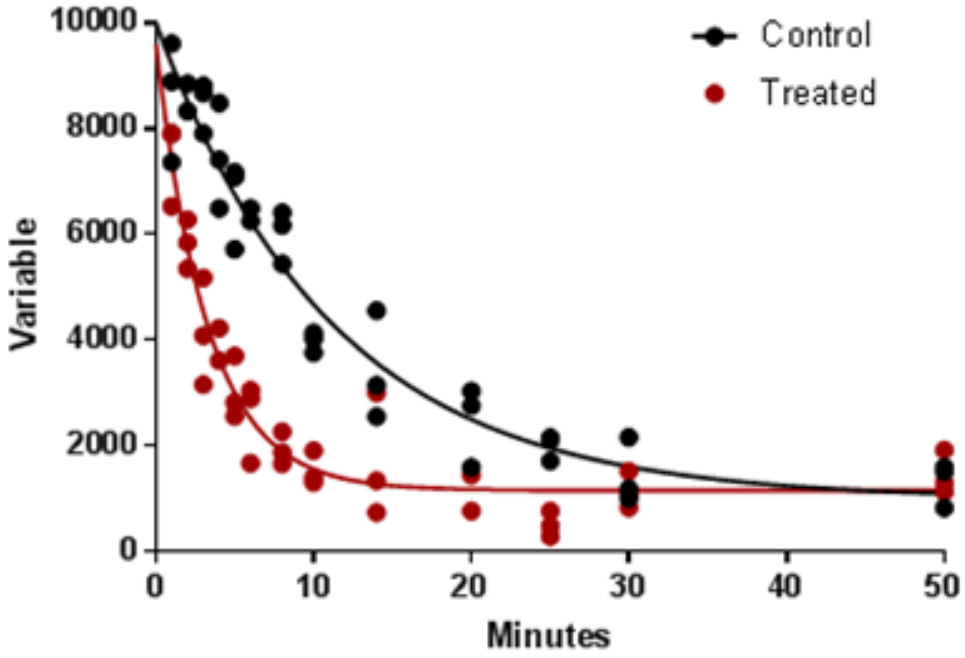


Statistical tests are also **inferential**

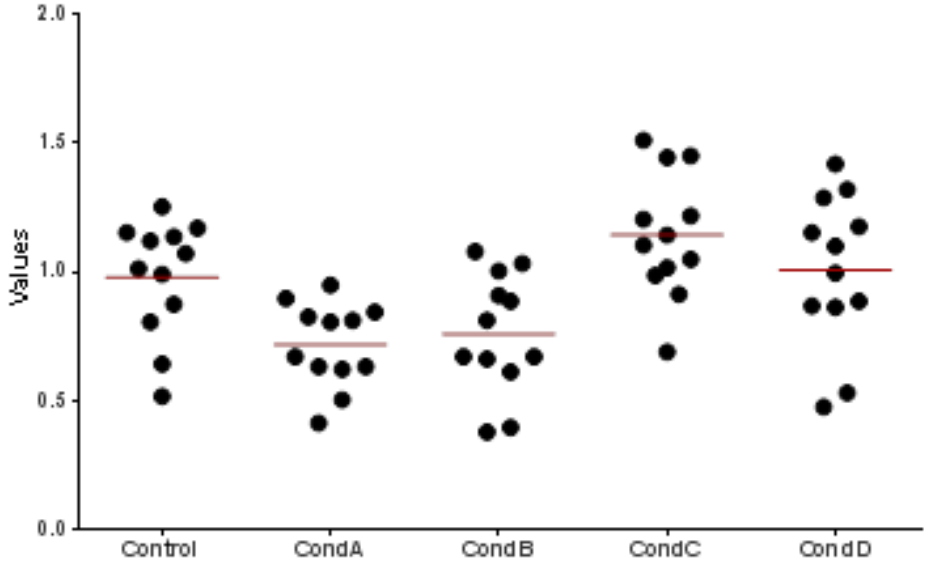

Question
Experimental Design
Sample Size
Experiment
Data Collection/Storage
Data Exploration
Data Analysis
Results

Quantitative data: Scatterplot

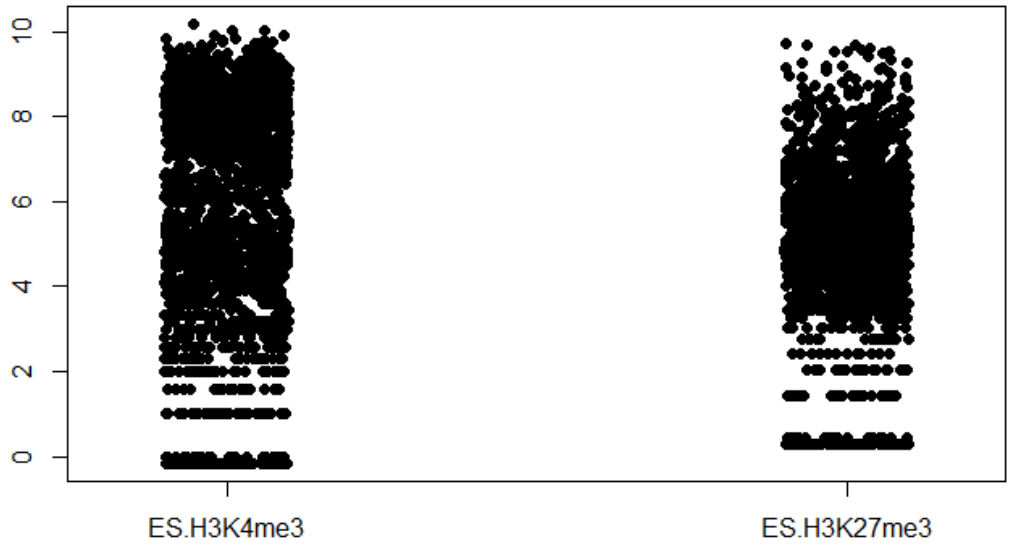




Quantitative data: Scatterplot/stripchart

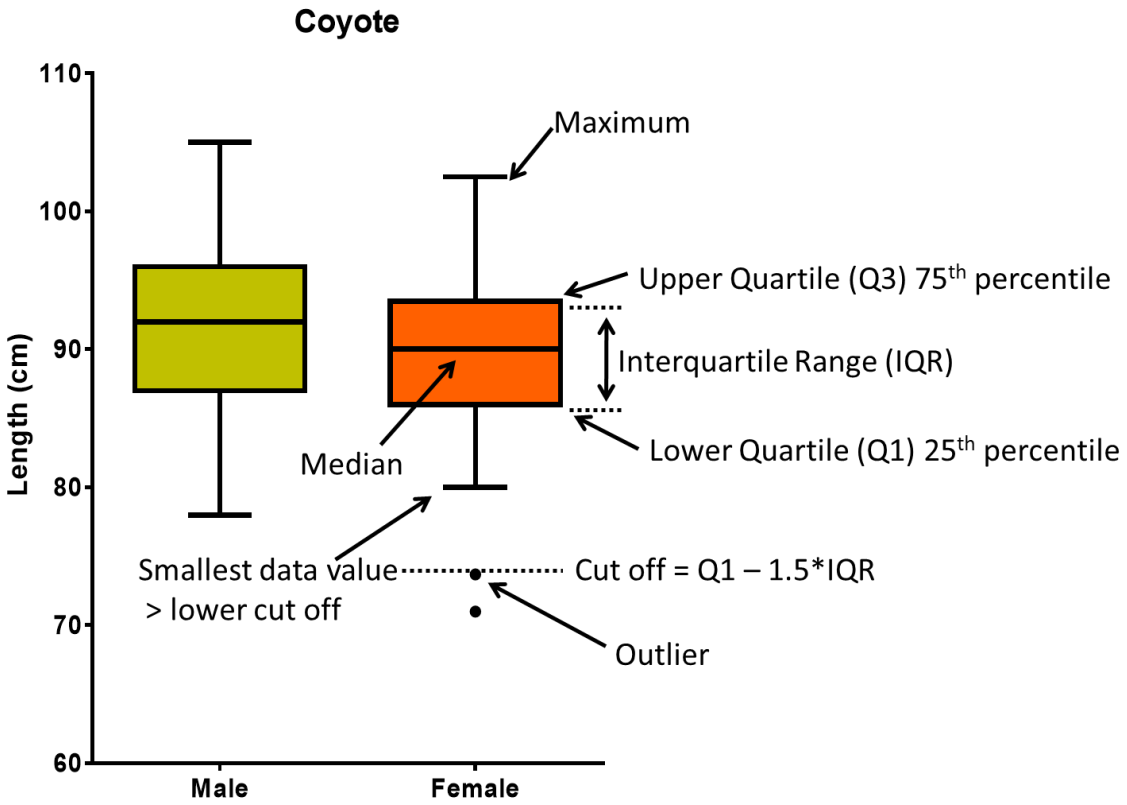


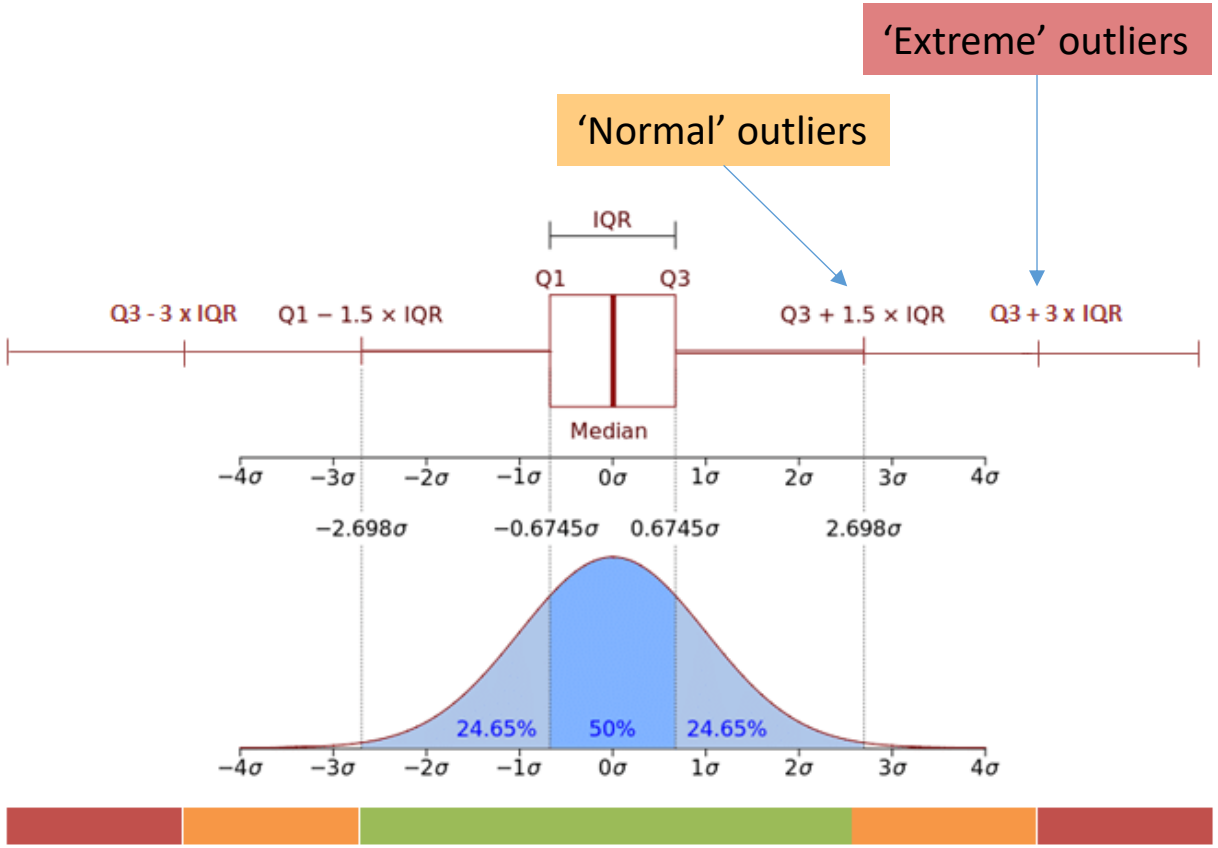
Small sample



Big sample

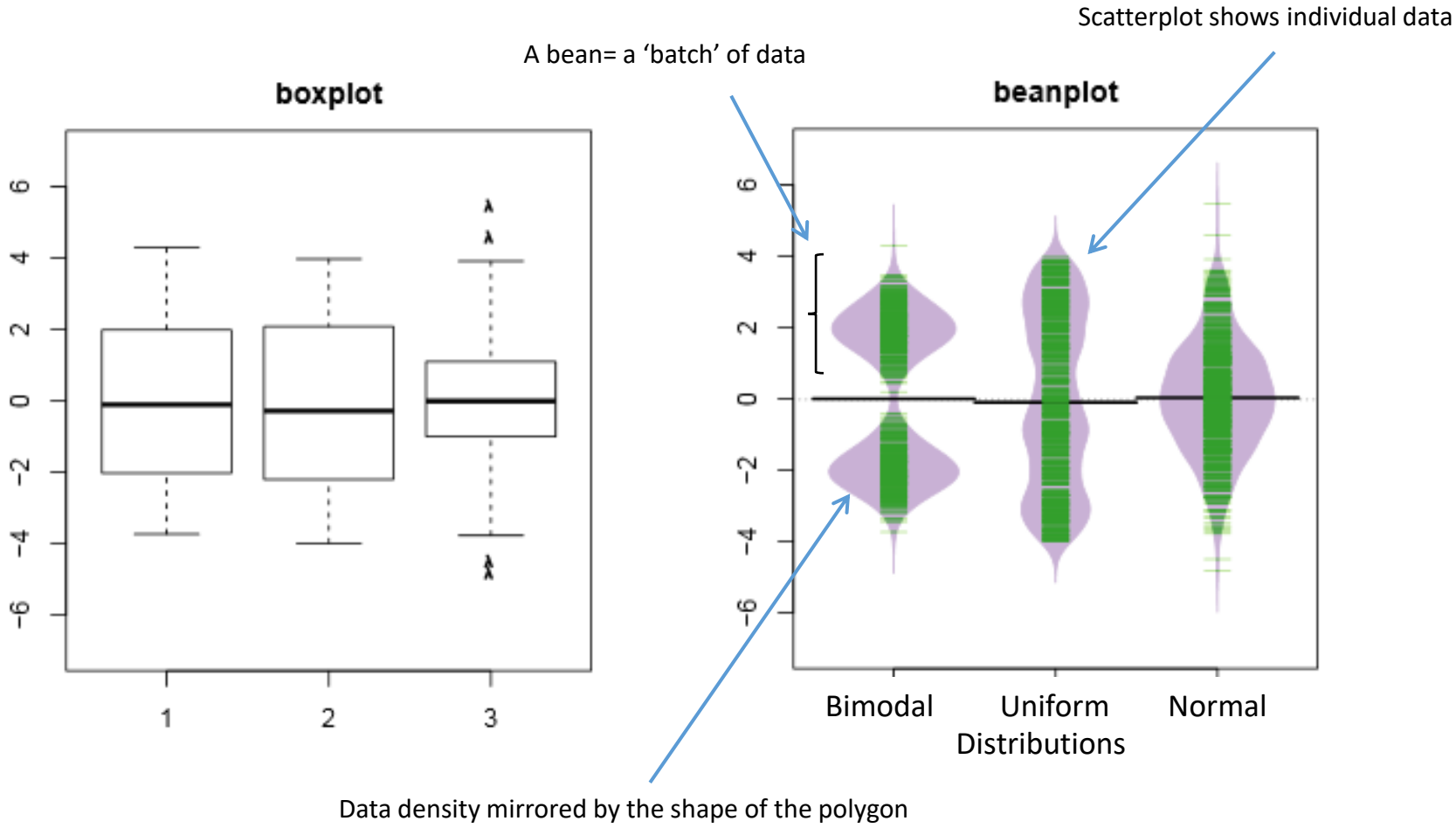
Quantitative data: Boxplot





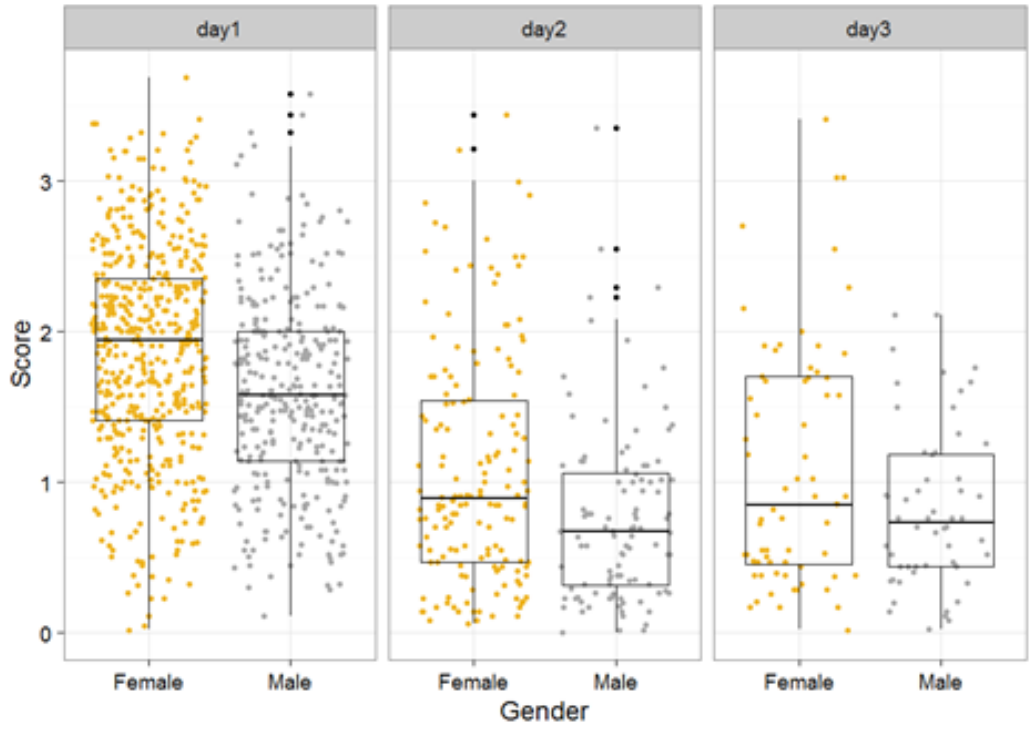
https://www.researchgate.net/publication/328818609_Outcomes_and_features_of_the_inspection_of_receiver_tubes_ITR_system_for_improved_OM_in_parabolic_trough_plants

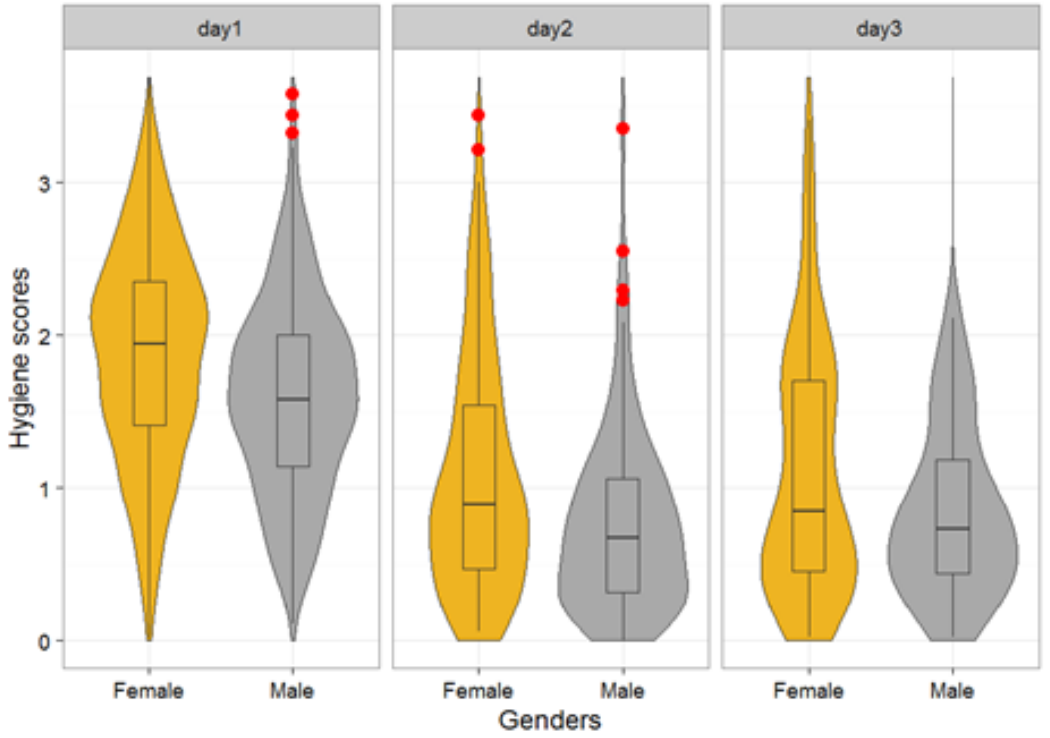
Quantitative data: Boxplot or Beanplot (aka Violinplot)



Quantitative data:

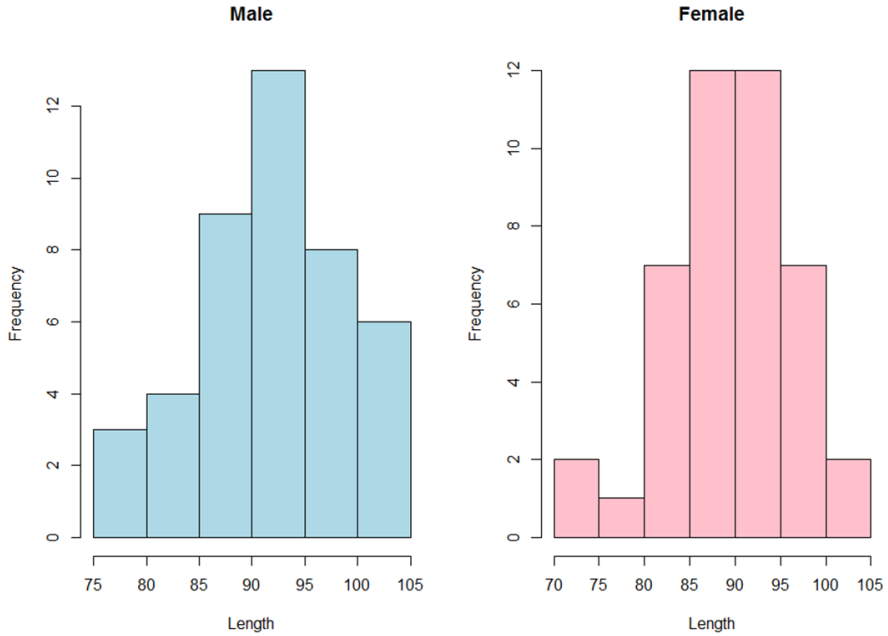
Boxplot and Violinplot and Scatterplot





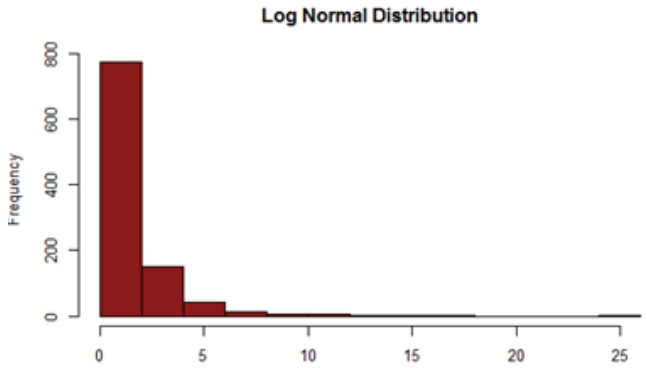
Quantitative data: Histogram

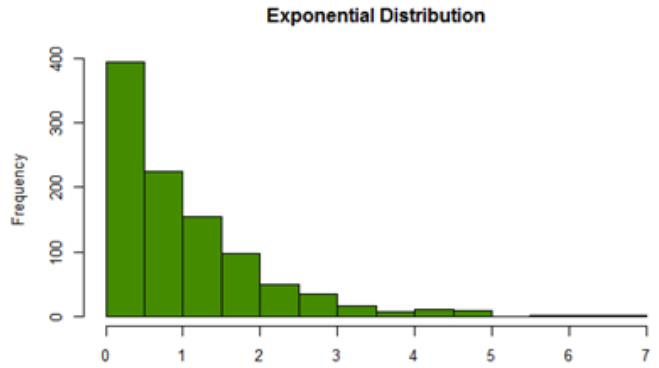
Big sample

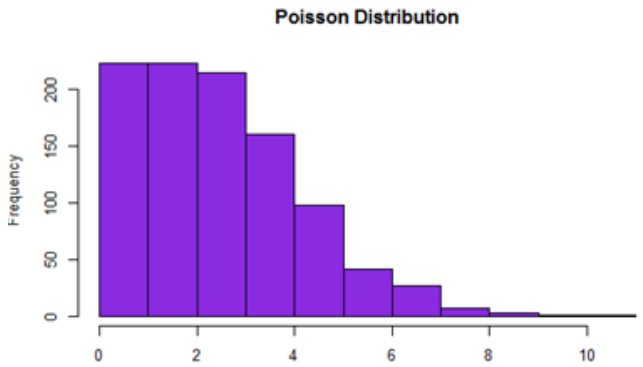


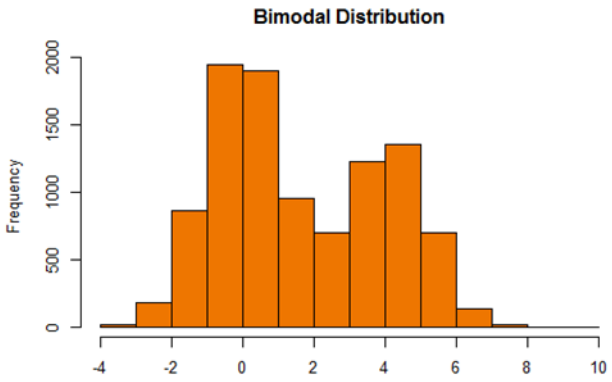
Small sample

Quantitative data: Histogram (distribution)







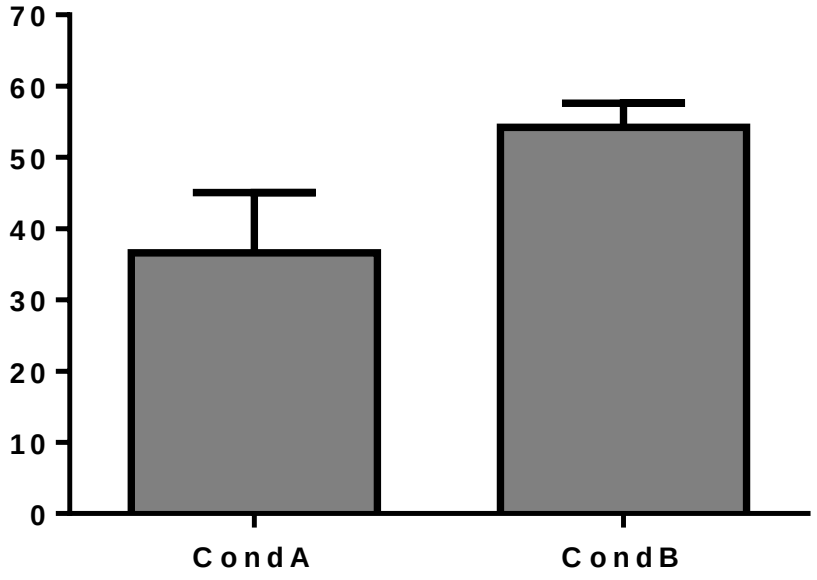


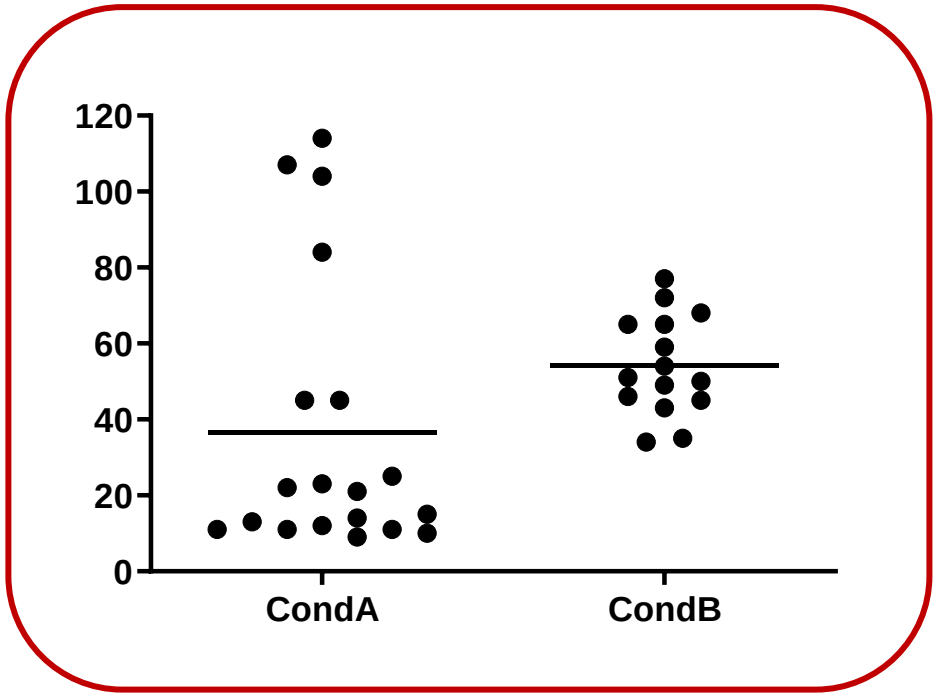
Data exploration ≠ plotting data

https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

Plotting is not inherently the same thing as exploring

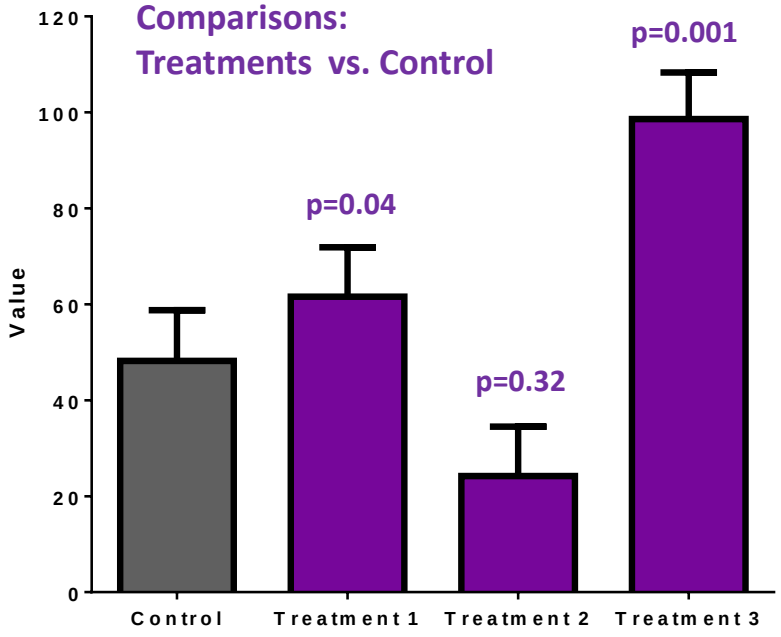
- One experiment: change in the variable of interest between CondA to CondB.
  - ❖ Data plotted as a **bar chart**.

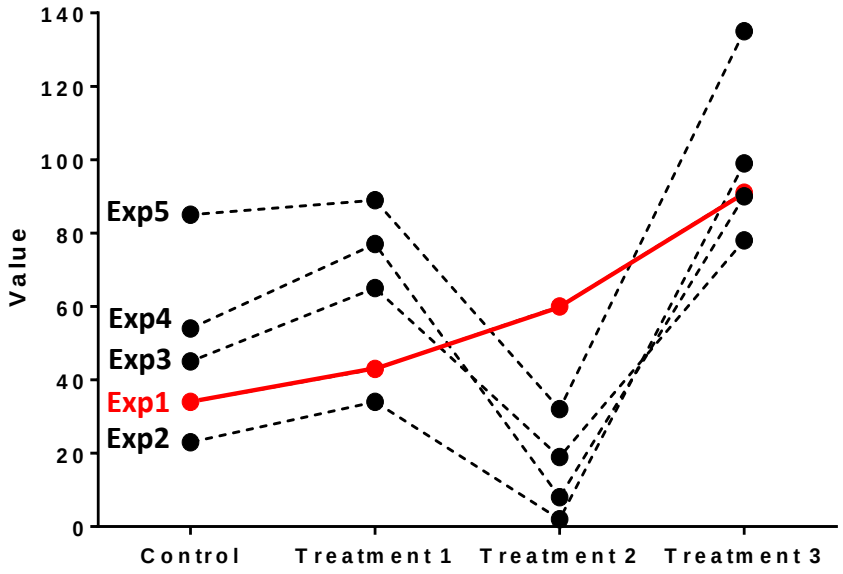




Plotting is not inherently the same thing as exploring

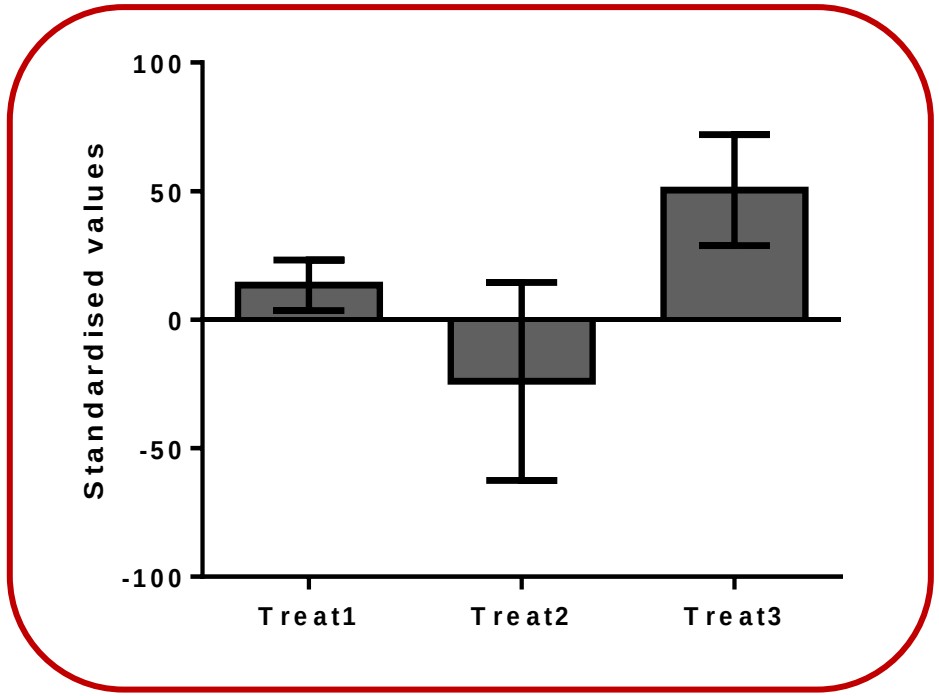
- Five experiments: change in the variable of interest between 3 treatments and a control.
  - ❖ Data plotted as a **bar chart**.

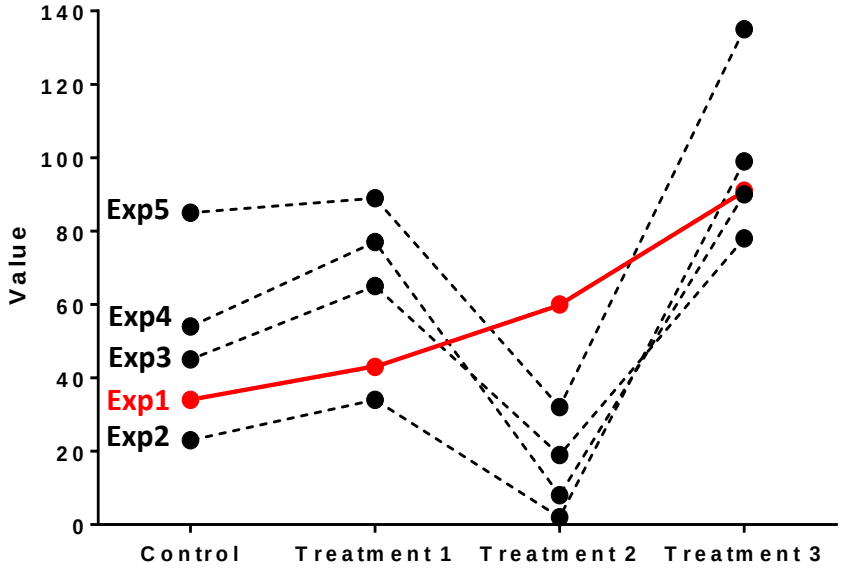




Plotting is not inherently the same thing as exploring

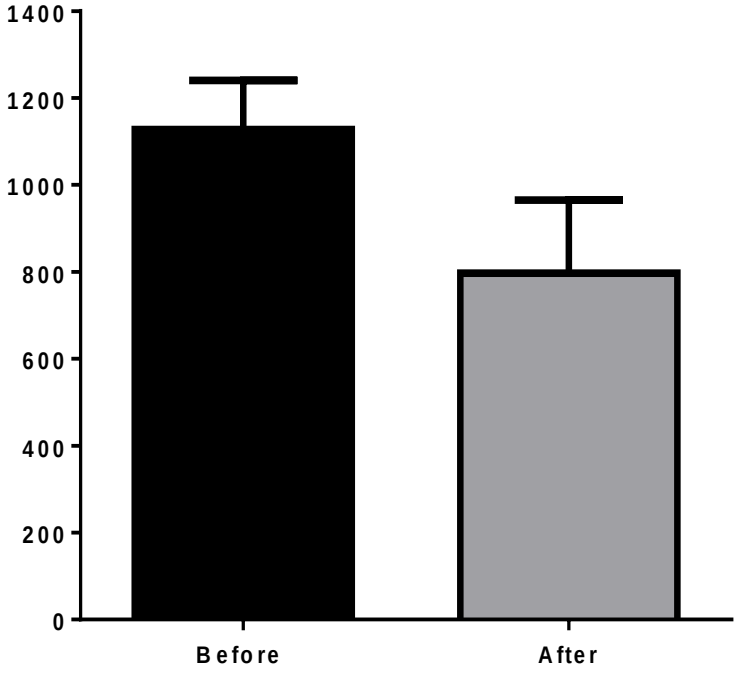
- Five experiments: change in the variable of interest between 3 treatments and a control.
  - ❖ Data plotted as a **bar chart**.

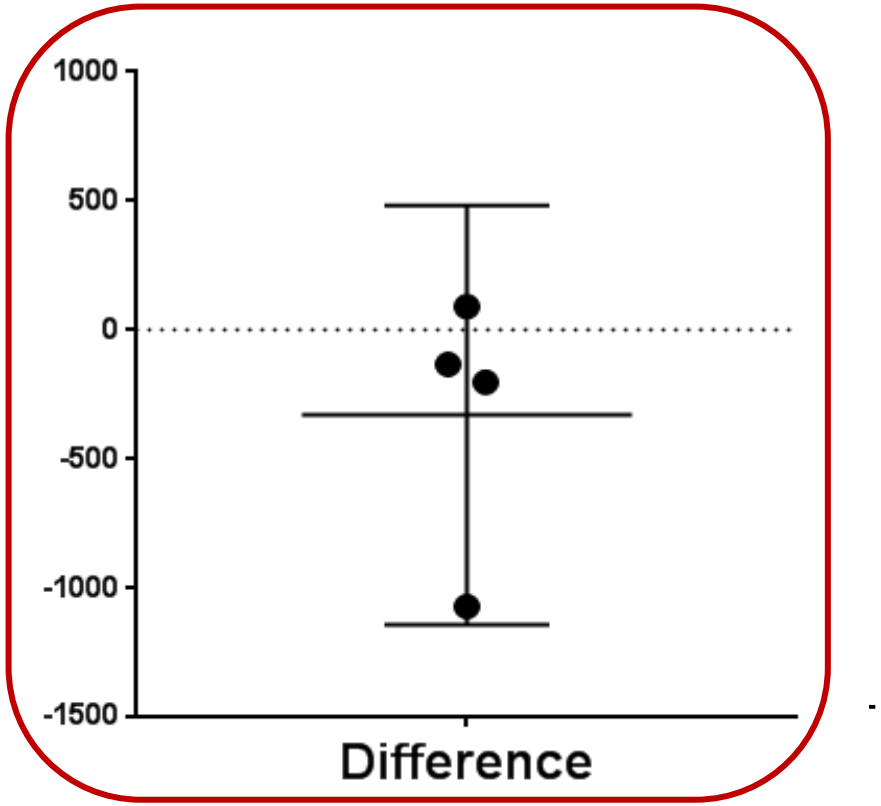




Plotting is not inherently the same thing as exploring

- Four experiments: Before-After treatment effect on a variable of interest.
- Hypothesis: Applying a treatment will decrease the levels of the variable of interest.
  - ❖ Data plotted as a **bar chart**.





Data exploration ≠ plotting data

https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

Babraham Institute

Key concepts and Assumptions

Analysis of Quantitative data

Choice of a statistical test

Hayley Carr & Anne Segonds-Pichon

v2024-05

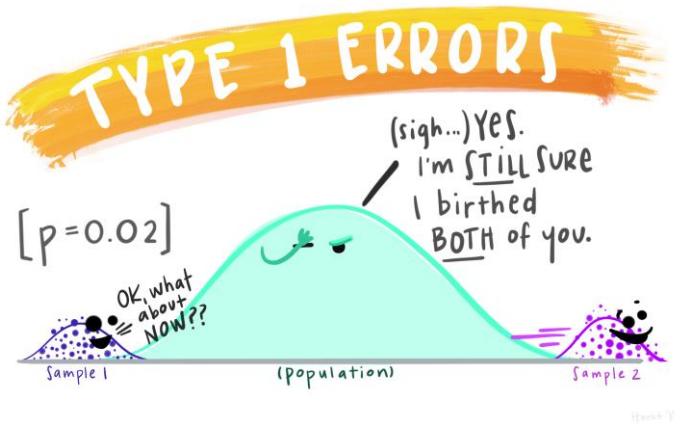
Babraham Bioinformatics

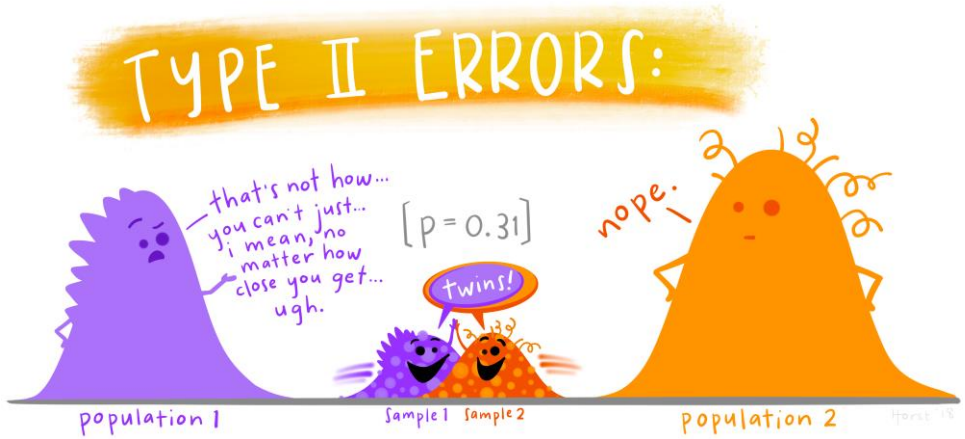
The null hypothesis and error types

- The null hypothesis (H_0): H_0 = no effect
- The aim of a statistical test is to reject or not H_0.

- **High specificity** = low **False Positives** = low **Type I error**
- **High sensitivity** = low **False Negatives** = low **Type II error**

Statistical decision	**True state of H_0**	
	H_0 true (no effect)	**H_0 false (effect)**
Reject H_0	Type I error α False positive	Correct True positive
Do not reject H_0	Correct True negative	Type II error β False negative





https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

Sample → **Statistical inference** → Population

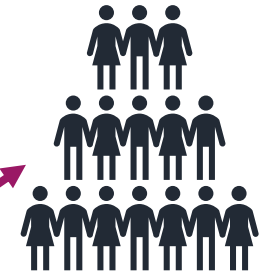
Difference

Is it meaningful? | Is it real?

Determined scientifically ← Is it meaningful?

Statistical test

Difference | Variation | Sample size



Result

Statistic (t, F, …) ← Is it difference big enough? → Significance

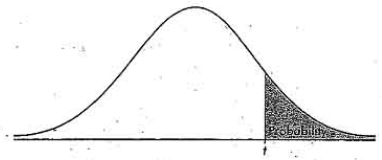
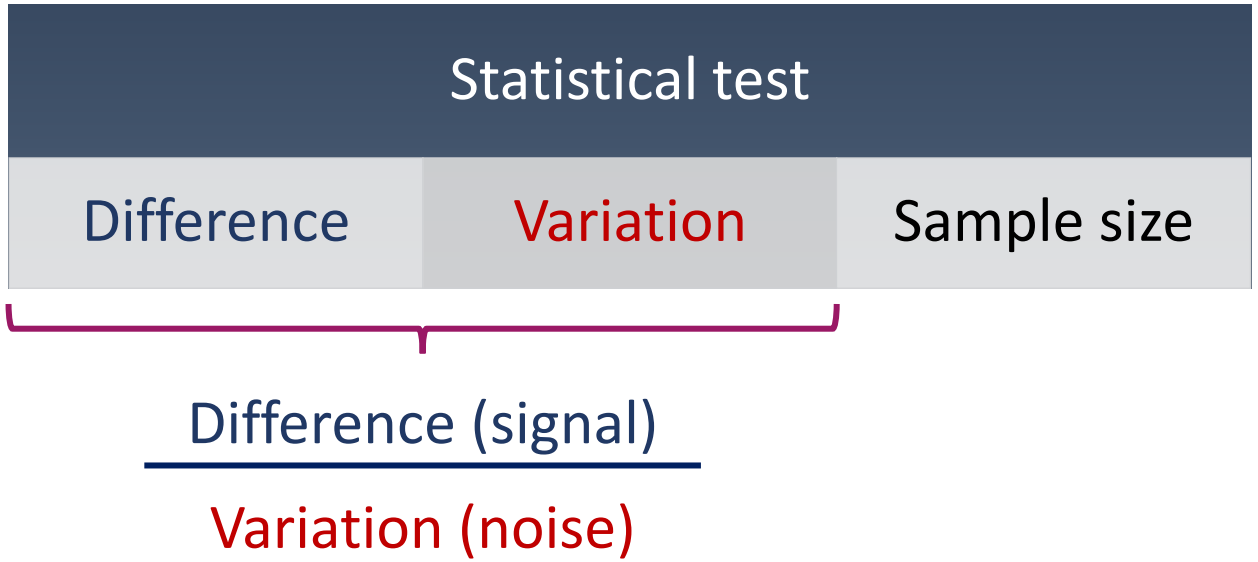


TABLE B: *t*-DISTRIBUTION CRITICAL VALUES

	Tail probability *p*											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792

Signal-to-noise ratio

- Stats are all about understanding and controlling variation
- The ratio of signal to noise determines the significance



$\frac{\text{Signal}}{\text{Noise}}$

If the **noise** (interindividual variation) **is low** then the **signal is detectable**

→ statistical significance

$\frac{\text{Signal}}{\text{Noise}}$

If the **noise is large** the **same signal will not be detected**

→ no statistical significance

Choice of a statistical test

There are many statistical tests. Which one we use depends on:

- What we want to do
  - The questions asked
  - Correct **statistical test** to answer our questions
- What sort of data we have
  - The type and behaviour
  - Correct **statistical family**
    - There are 2 families of statistical tests:
      - Parametric tests with 4 assumptions to be met
      - Non-parametric tests with no or few assumptions and/or for qualitative data

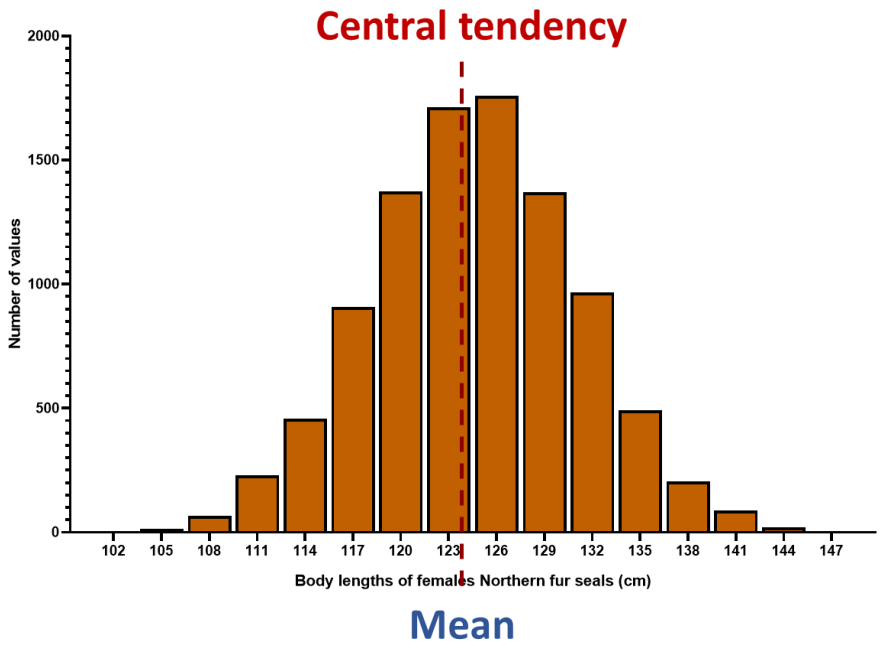
Assumptions of Parametric Data

What sort of data we have

- All parametric tests have 4 basic assumptions that must be met for the test to be accurate.

First assumption: Normality

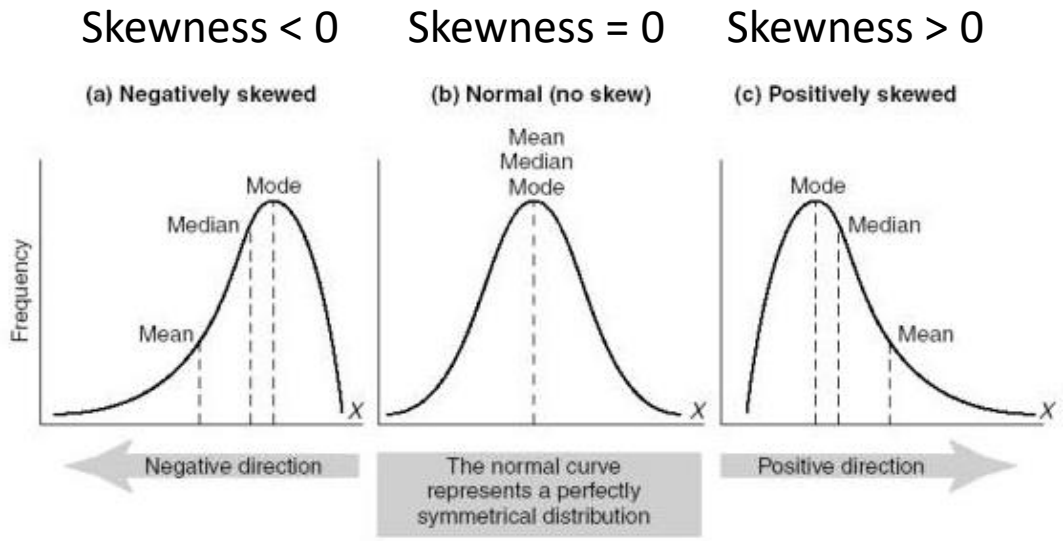
- Normal shape, bell shape, Gaussian shape



Assumptions of Parametric Data

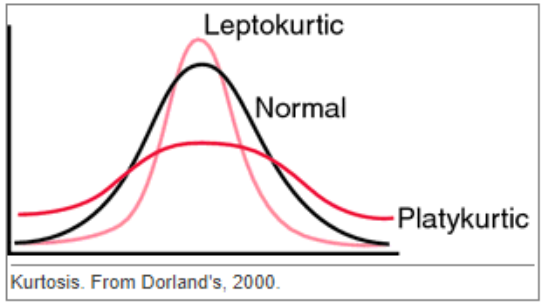
What sort of data we have

- Frequent departures from normality:
  - Skewness: lack of symmetry of a distribution



  - Kurtosis: measure of the degree of 'peakedness'
    - Same variance and same skew but differ markedly in kurtosis

More peaked distribution: kurtosis > 0



Flatter distribution: kurtosis < 0

Assumptions of Parametric Data

What sort of data we have

Second assumption: Homoscedasticity (Homogeneity in variance)

- The variance should not change systematically throughout the data

Third assumption: Interval data (linearity)

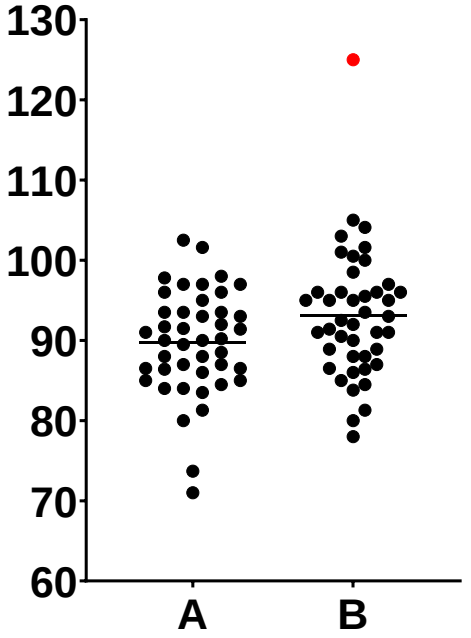
- The distance between points of the scale should be equal at all parts along the scale

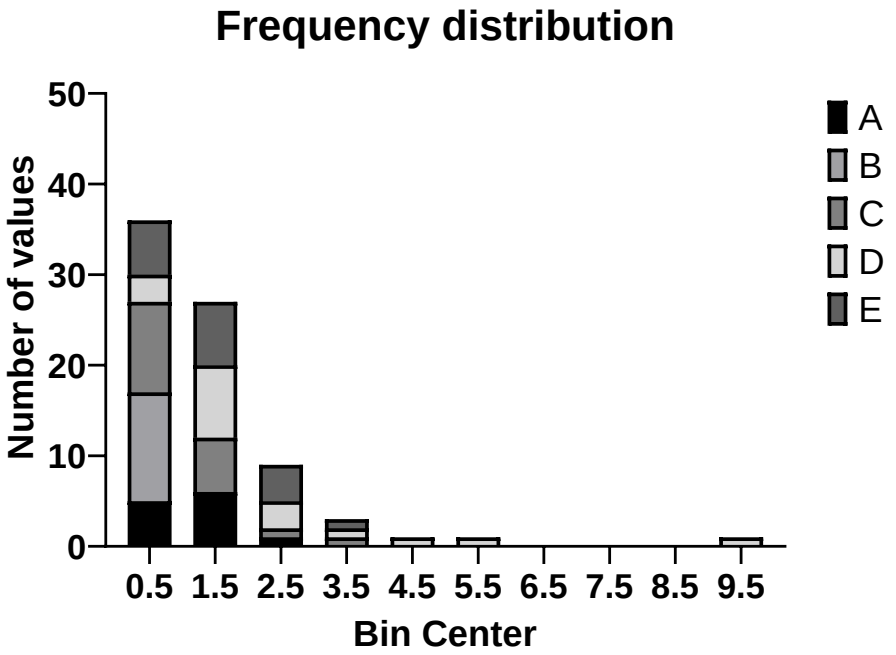
Fourth assumption: Independence

- Data from different subjects are independent
  - Each data point in the sample is independent from all the others = Values corresponding to one subject do not influence the values corresponding to another subject
  - Important in repeated measures experiments

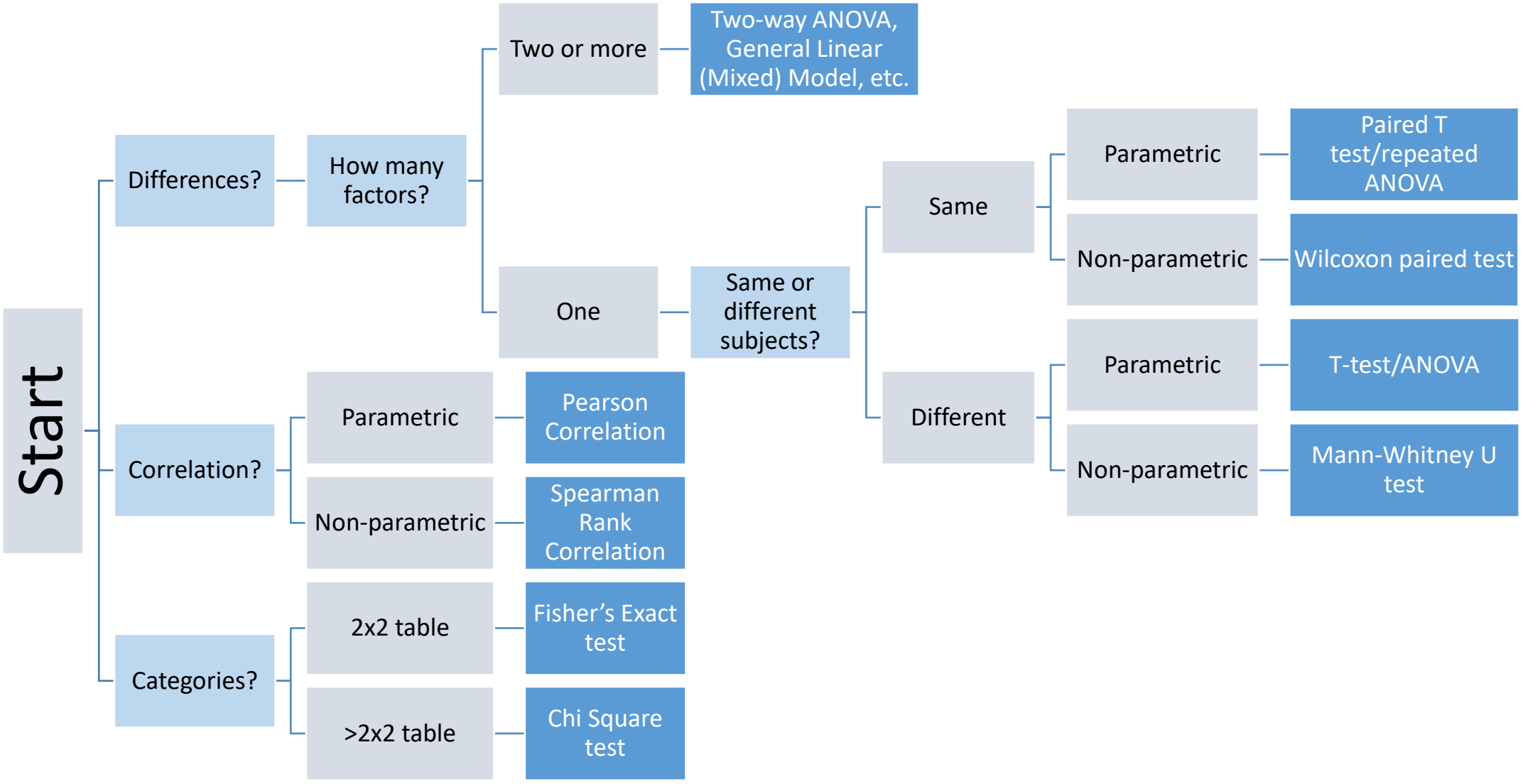
Non-parametric tests

- General principle: original data are transformed into **ranks**
- **Not meeting the assumptions for parametric tests is not enough** to switch to a non-parametric approach
- Data exploration is key:
  - Outliers?
  - Possible transformation?
  - Parametric with corrections?
- If outcome is a rank or a score with limited possible values: often non-parametric approach



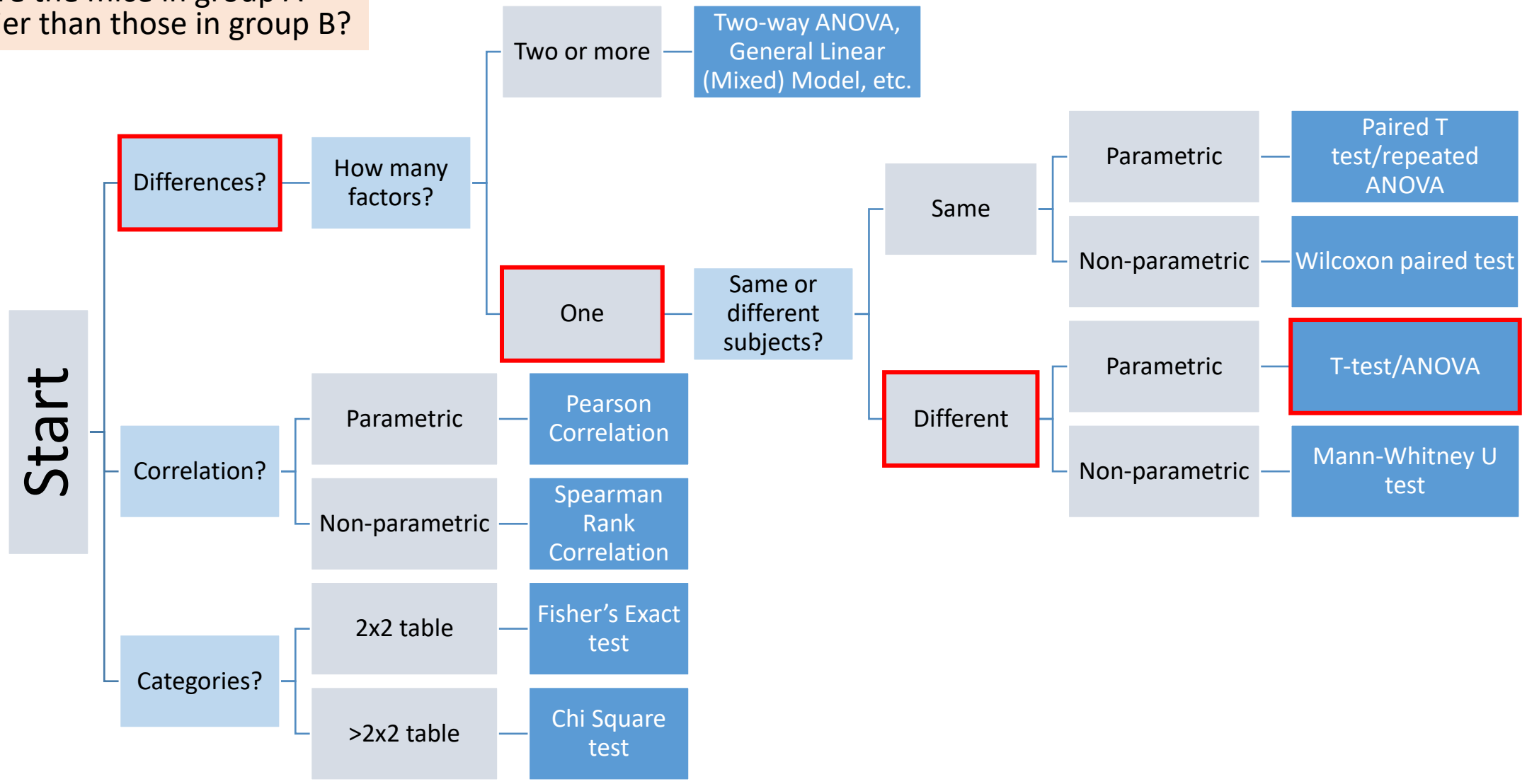


Choice of a statistical test



Choice of a statistical test

Are the mice in group A heavier than those in group B?



Babraham Institute

Analysis of Quantitative data

Student's *t*-test

Hayley Carr & Anne Segonds-Pichon

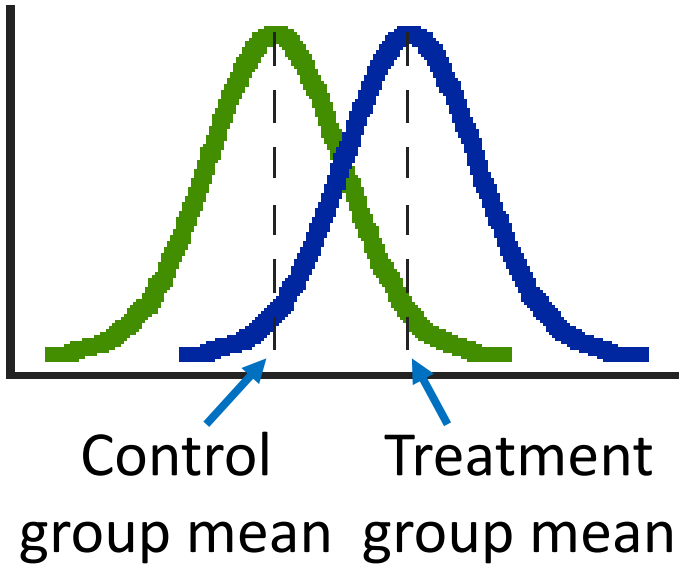
v2025-02

Babraham Bioinformatics

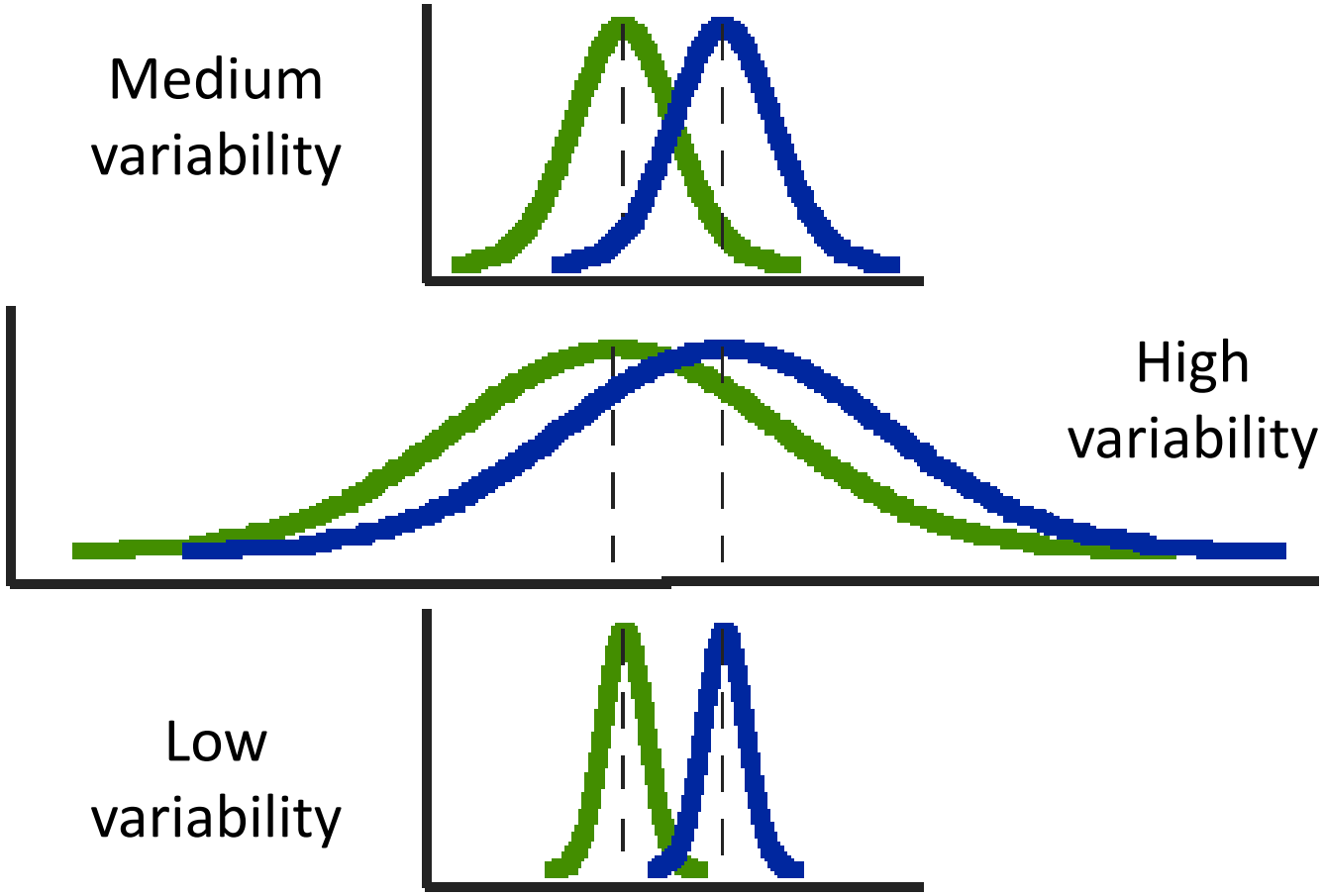
Comparison between 2 groups

Student's *t*-test

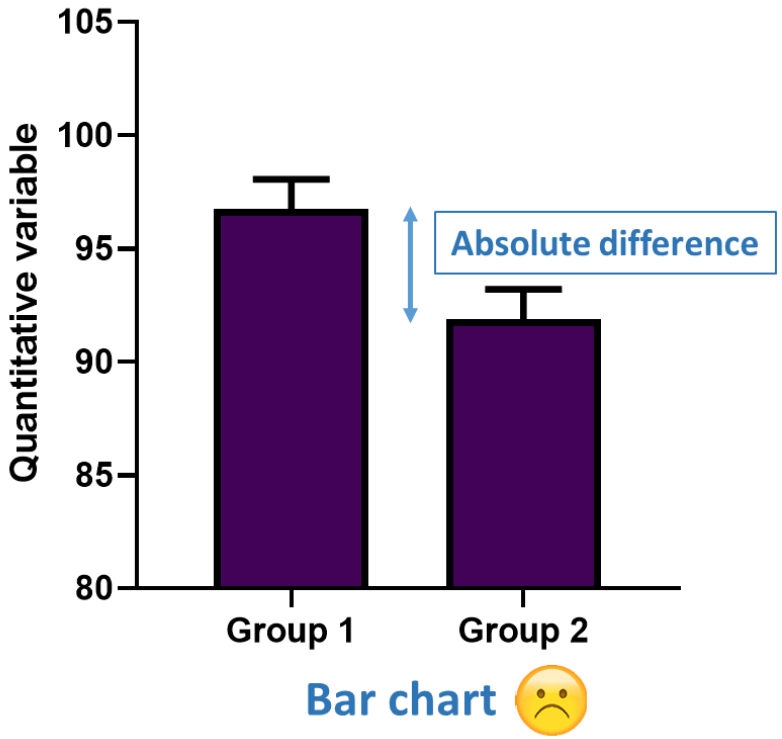
- **Basic idea**:
  - Comparison between 2 means accounting for variability
    - Absolute difference vs. variability



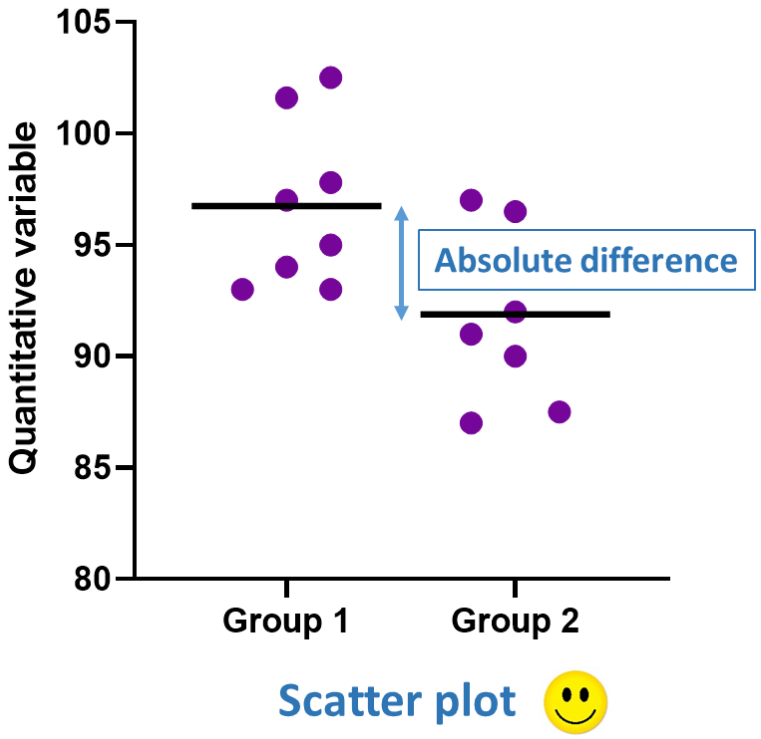
Variability does matter



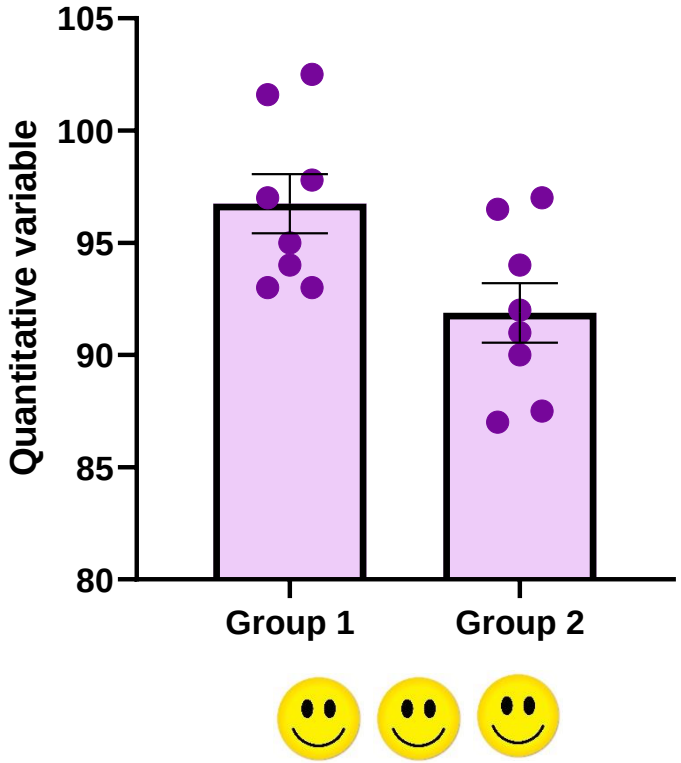
Variability does matter



Bar chart ☹️

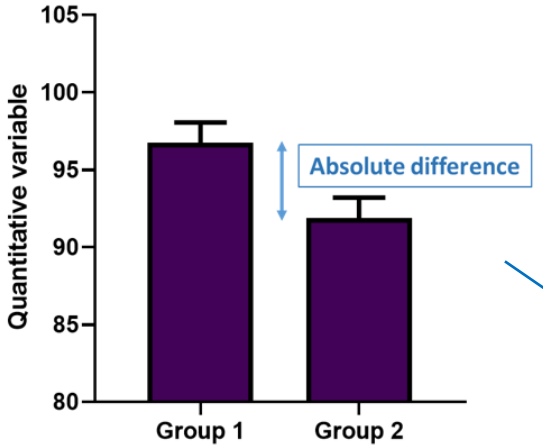


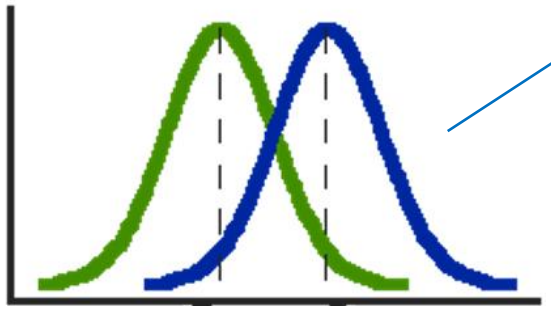
Scatter plot 🙂



🙂 🙂 🙂

Signal-to-noise ratio





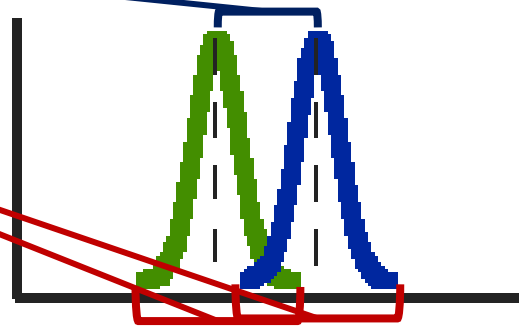
$$\frac{\text{Difference (signal)}}{\text{Variation (noise)}}$$

$$\frac{\text{Signal}}{\text{Noise}} = \textbf{statistical significance}$$

$$\frac{\text{Signal}}{\text{Noise}} = \textbf{no statistical significance}$$

Student's *t*-test

$$\frac{\text{Signal}}{\text{Noise}} = \frac{\text{Difference between group means}}{\text{Variability of groups}}$$



$$= \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

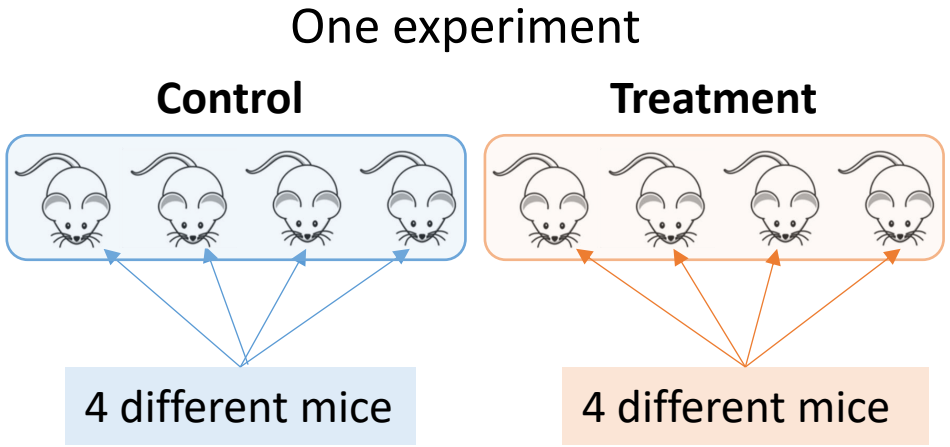
Basically the combination of the 2 SEM

$$=$$ **t-value**

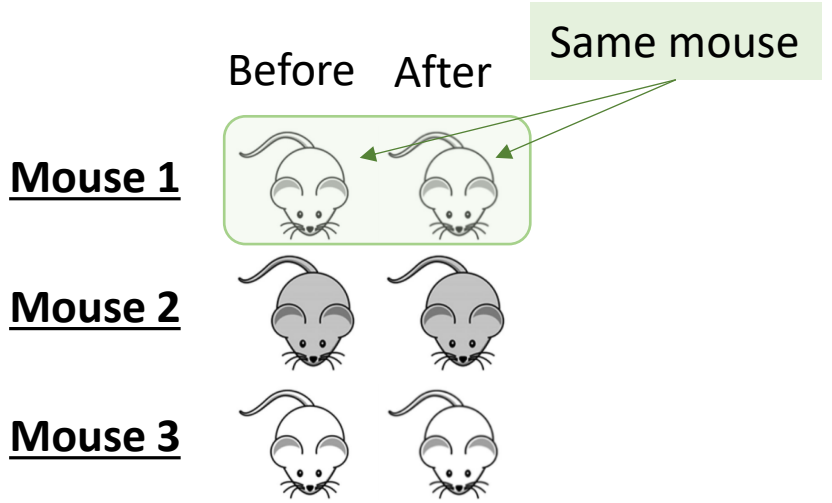
Student's *t*-test

- 3 types, depending on experimental design

- **Independent *t*-test**
  - Difference between 2 means of one variable for two independent groups



- **Paired *t*-test**
  - Difference between two measures of one variable for one group



- **One-Sample *t*-test**
  - Difference between the mean of a single variable and a specified constant

Example: coyotes.csv

- Do male and female coyotes differ in size?
- The file contains individual body length of male and female coyotes.
- Steps:
  - Load **coyote.csv**
  - Data exploration
    - Plot the data as **boxplot**, **violinplot**, **histogram** and **stripchart**
  - Check the assumptions for parametric test

Example: Load coyote.csv

- Read in the data using read_csv after loading tidyverse package
  - Use path to where your data is stored

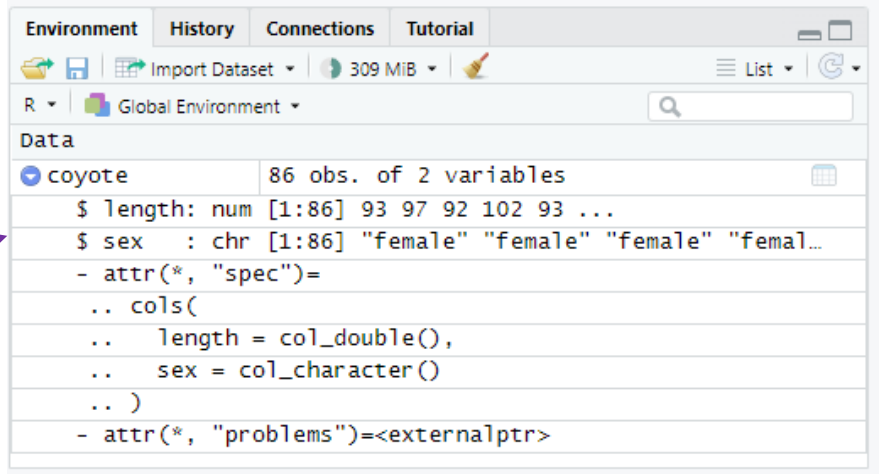
```
library(tidyverse)
coyote <- read_csv("Datasets to use/Coyotes.csv")
```

- View the data:

```
coyote

head(coyote)

View(coyote)
```

```
> coyote
# A tibble: 86 x 2
   length sex
    <dbl> <chr>
 1   93   female
 2   97   female
 3   92   female
 4  102.  female
 5   93   female
 6   84.5 female
 7  102.  female
 8   97.8 female
 9   91   female
10   98   female
# i 76 more rows
# i Use `print(n = ...)` to see more rows
```

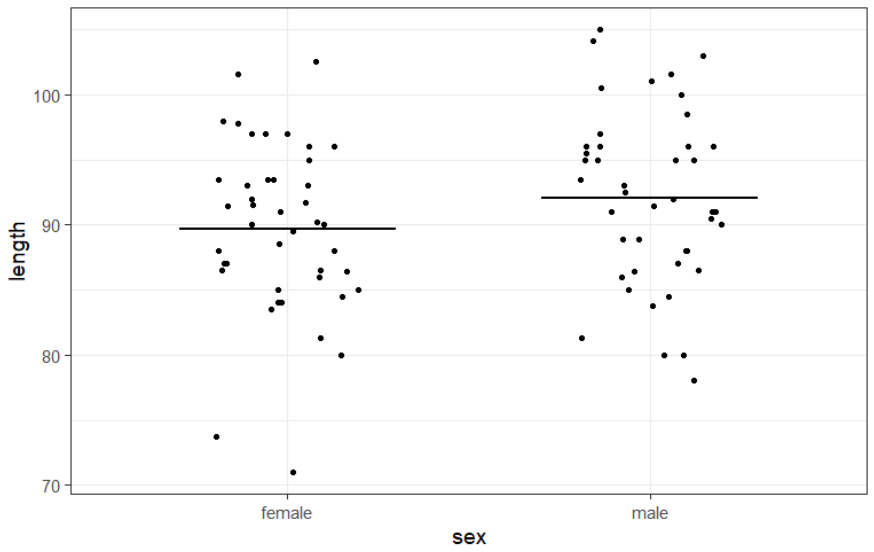
Or click on coyotes in Rstudio “Environment” tab

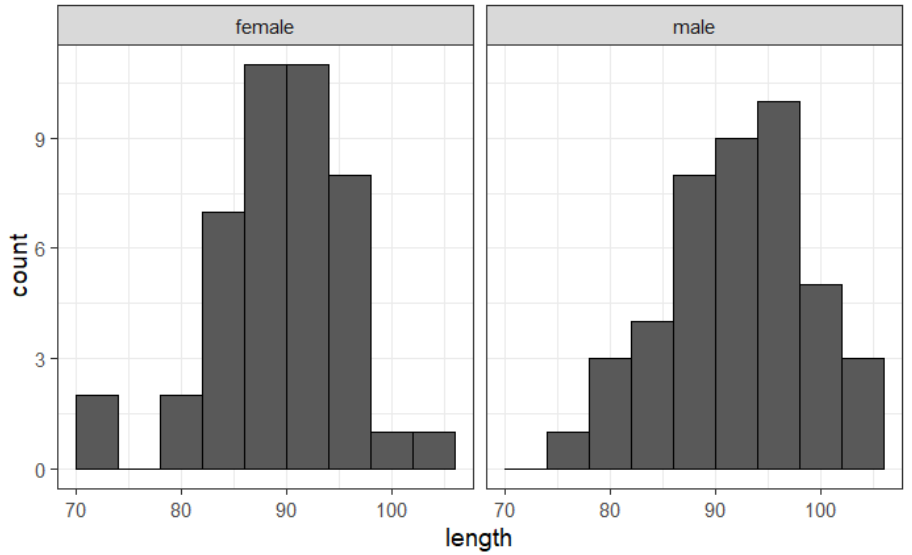


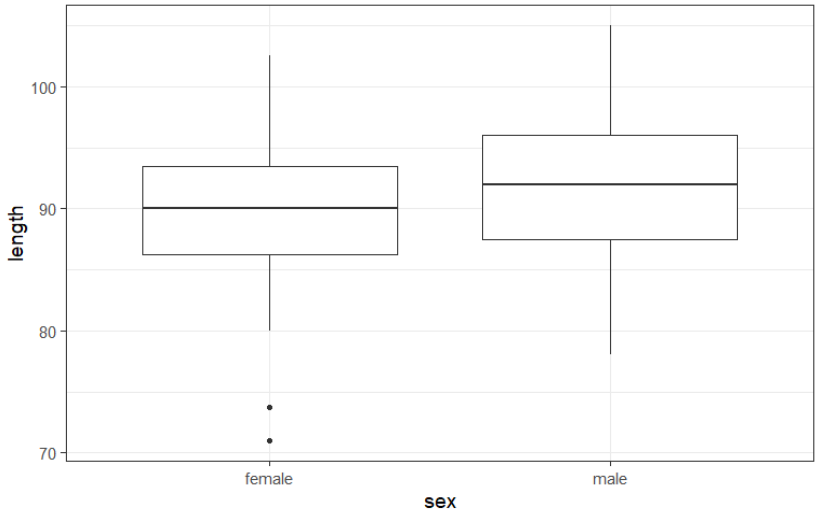
Example: Data exploration

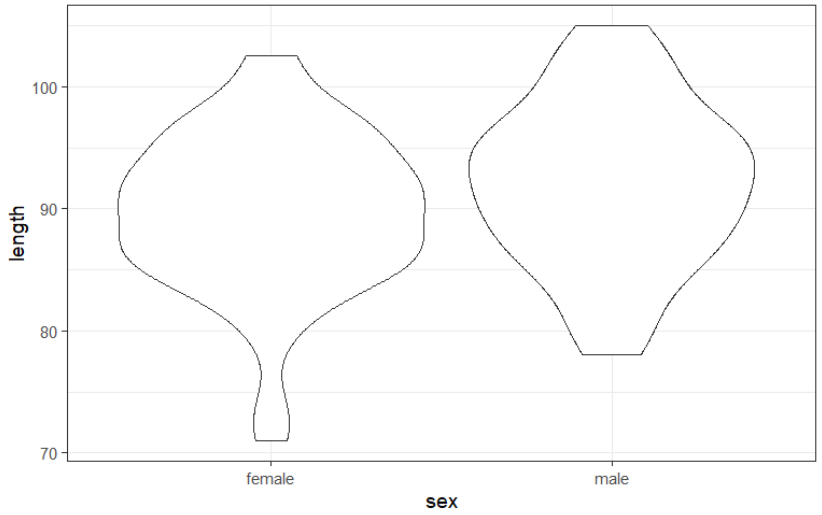
```
coyote %>%
  ggplot(aes(x=sex, y=length))+
  geom_...()
```

- Explore data using 4 different representations

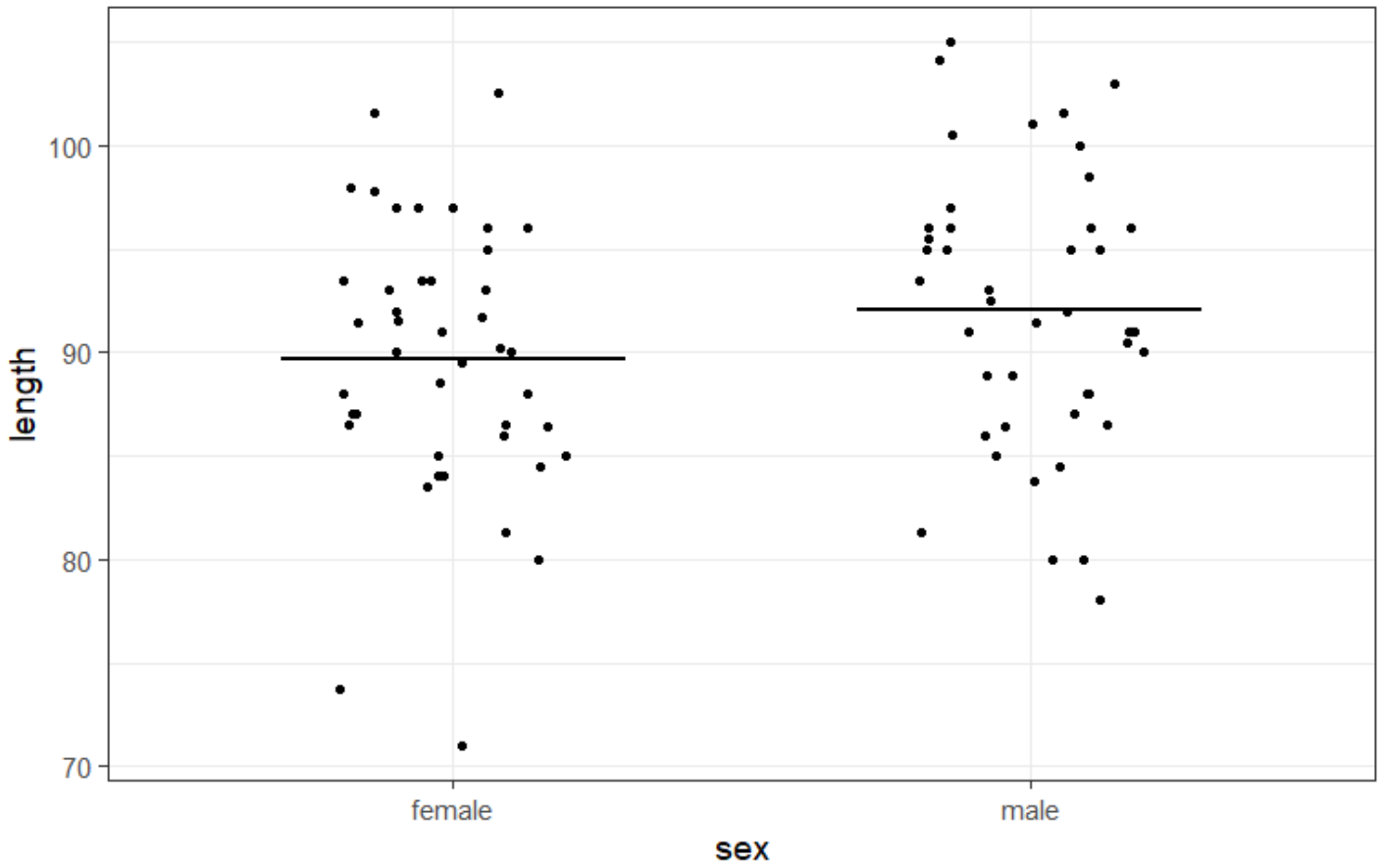








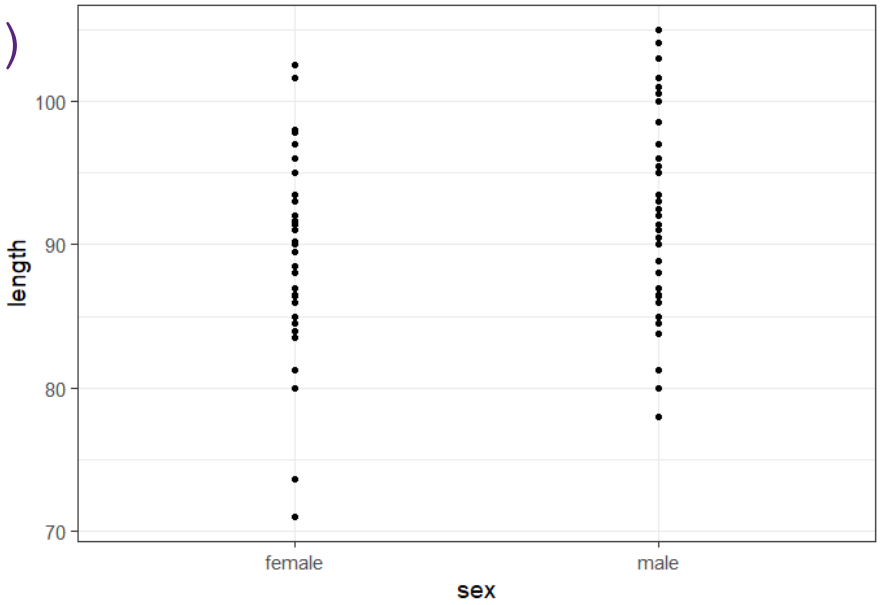
Example: Strip chart and line



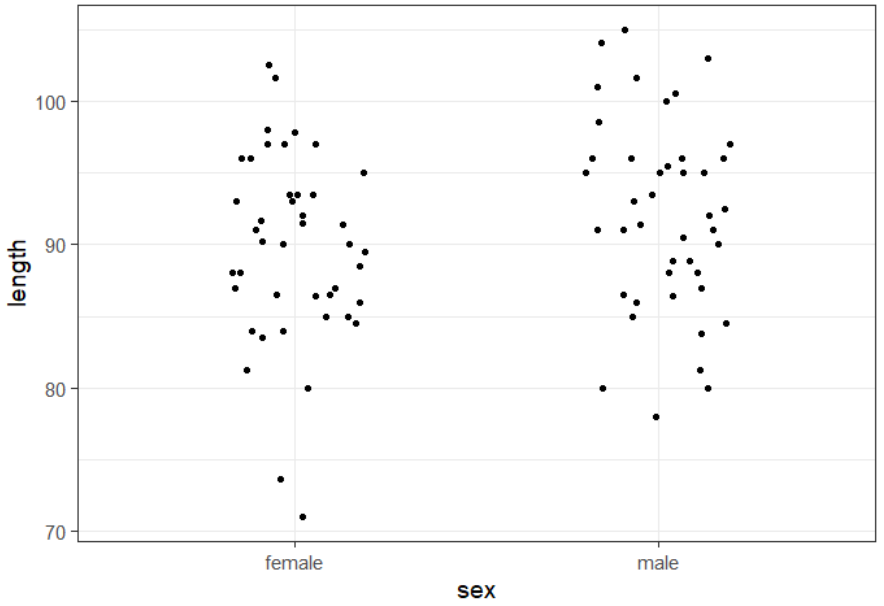
Example: Strip chart: geom_jitter()

- Variation of geom_point(): geom_jitter()

```
coyote %>%
  ggplot(aes(x=sex,y=length))+
  geom_point()
```




```
coyote %>%
  ggplot(aes(x=sex,y=length)) +
  geom_jitter(height=0, width=0.2)
```

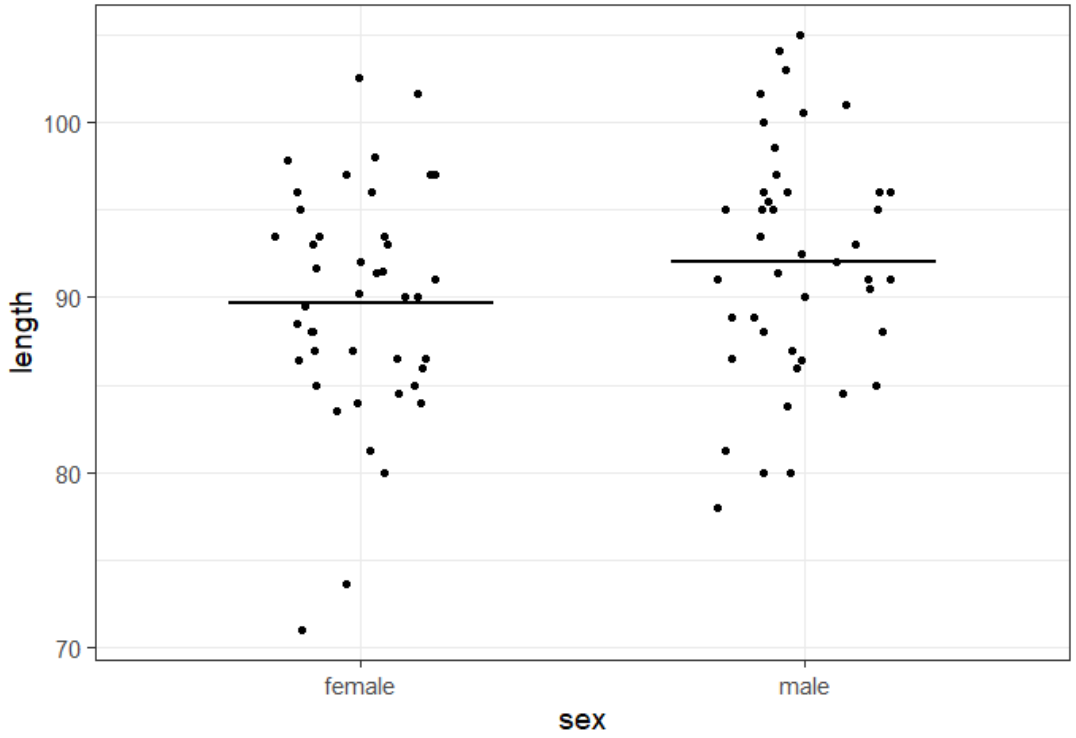


Example: Strip chart and line:

`stat_summary(geom=, fun=)`

- Graphical representation = a line: `geom="crossbar"`
- Statistical summary, given function: `fun = "mean" (or "median")`

```
coyote %>%
  ggplot(aes(sex,length)) +
  geom_jitter(height=0, width=0.2) +
  stat_summary(geom="crossbar",
      fun="mean", width=0.6,
      linewidth=0.3)+
```

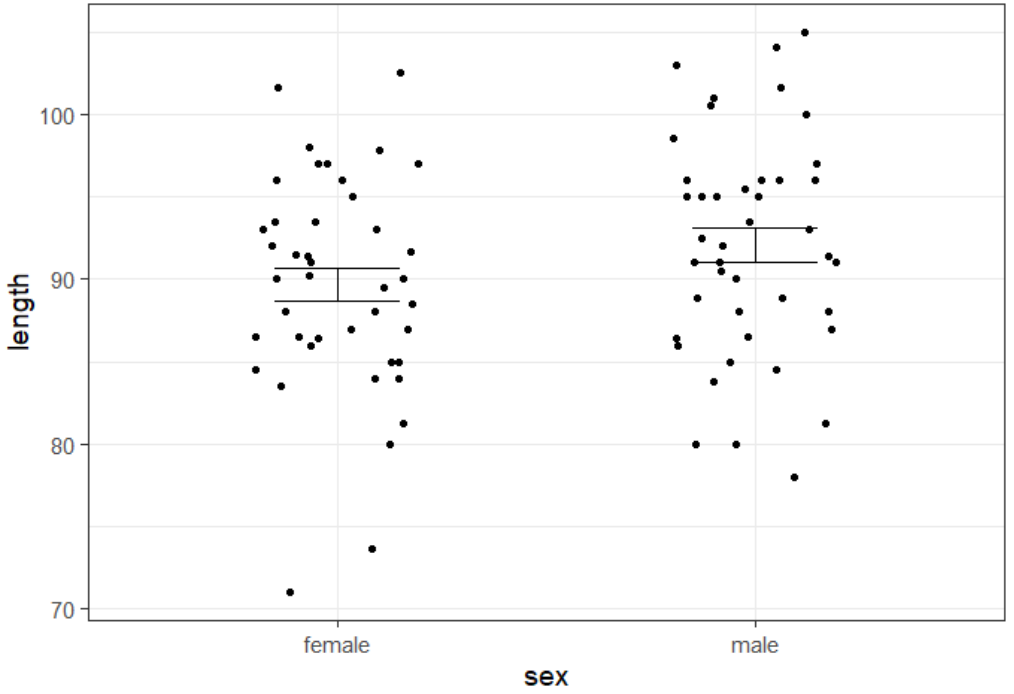


Example: Strip chart and line:

`stat_summary(geom=, fun.data=`

- Can alternatively add error bars: `geom="errorbar"`
- Now need function incl. error bars: `fun.data="mean_se"` (default)

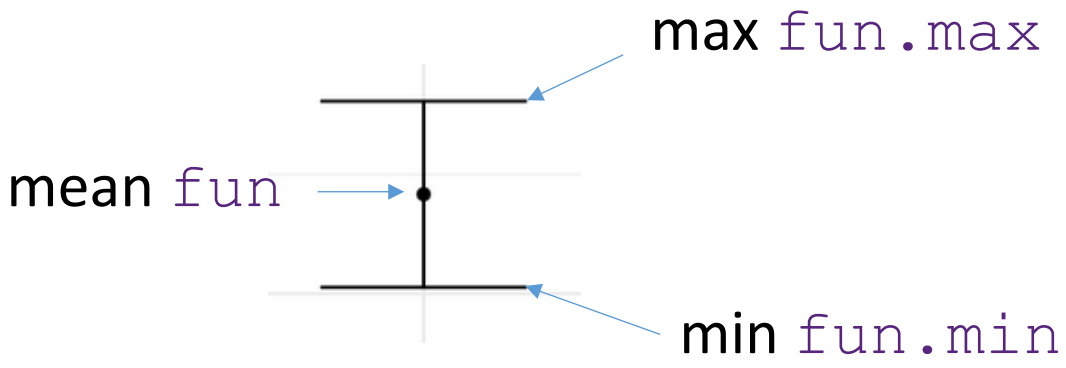
```
coyote %>%
  ggplot(aes(sex,length)) +
  geom_jitter(height=0, width=0.2) +
  stat_summary(geom="errorbar",
      fun.data="mean_se",
   width=0.3, linewidth=0.3)
```



Example: Strip chart: `stat_summary()`

```
stat_summary(geom=,
     fun=, fun.min=, fun.max=)
```

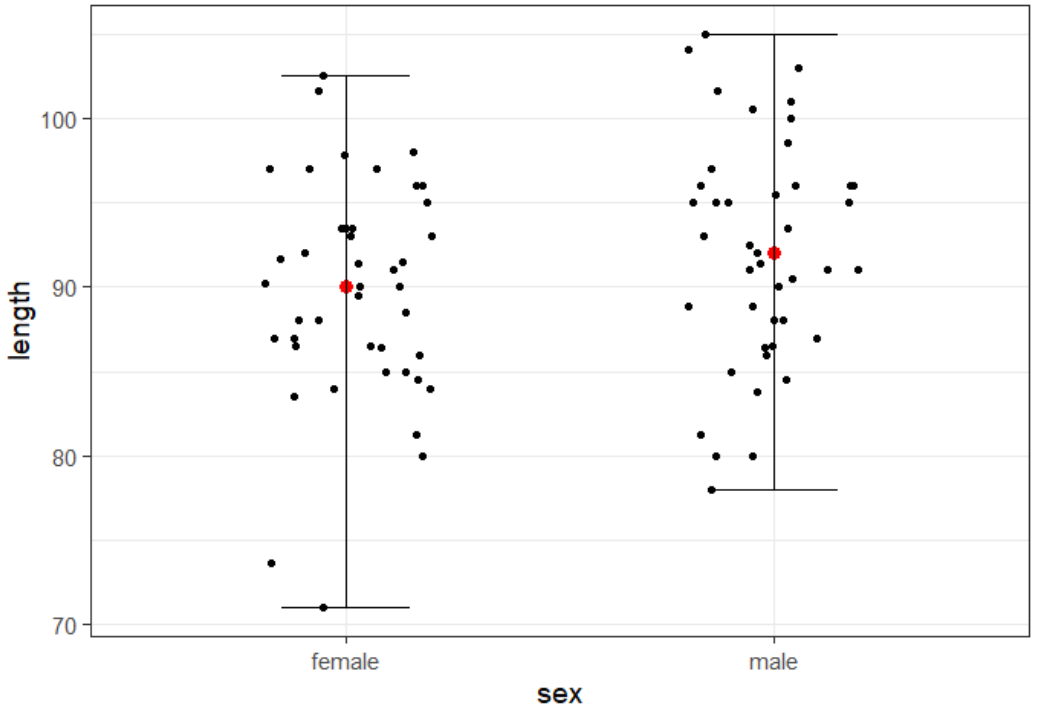
- Can manually add min/max




```
stat_summary(geom="point",
  fun=median, colour = "red",size = 3)+
stat_summary(geom="errorbar",
  fun=median, fun.min=min,
  fun.max=max)
```

ggpubr # has more functions that can be useful:

```
mean_sd()
mean_ci()
mean_range()
median_iqr()
median_q1q3()
median_range()
```



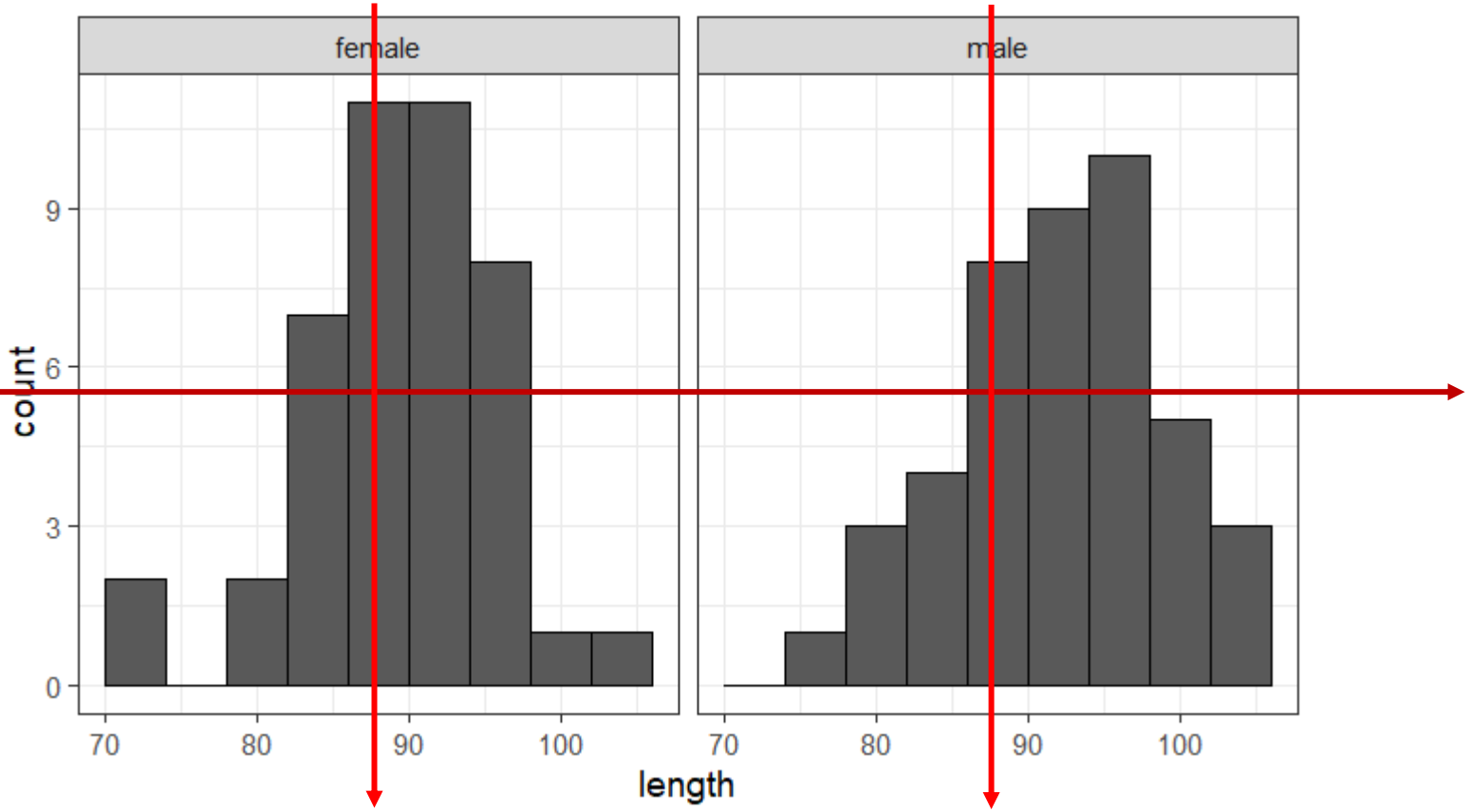
Example: Histogram

also works
`facet_wrap(~sex)`

```
geom_histogram() +
facet_grid(rows=vars(row),cols=vars(column))
```

2 columns: one per sex

One row

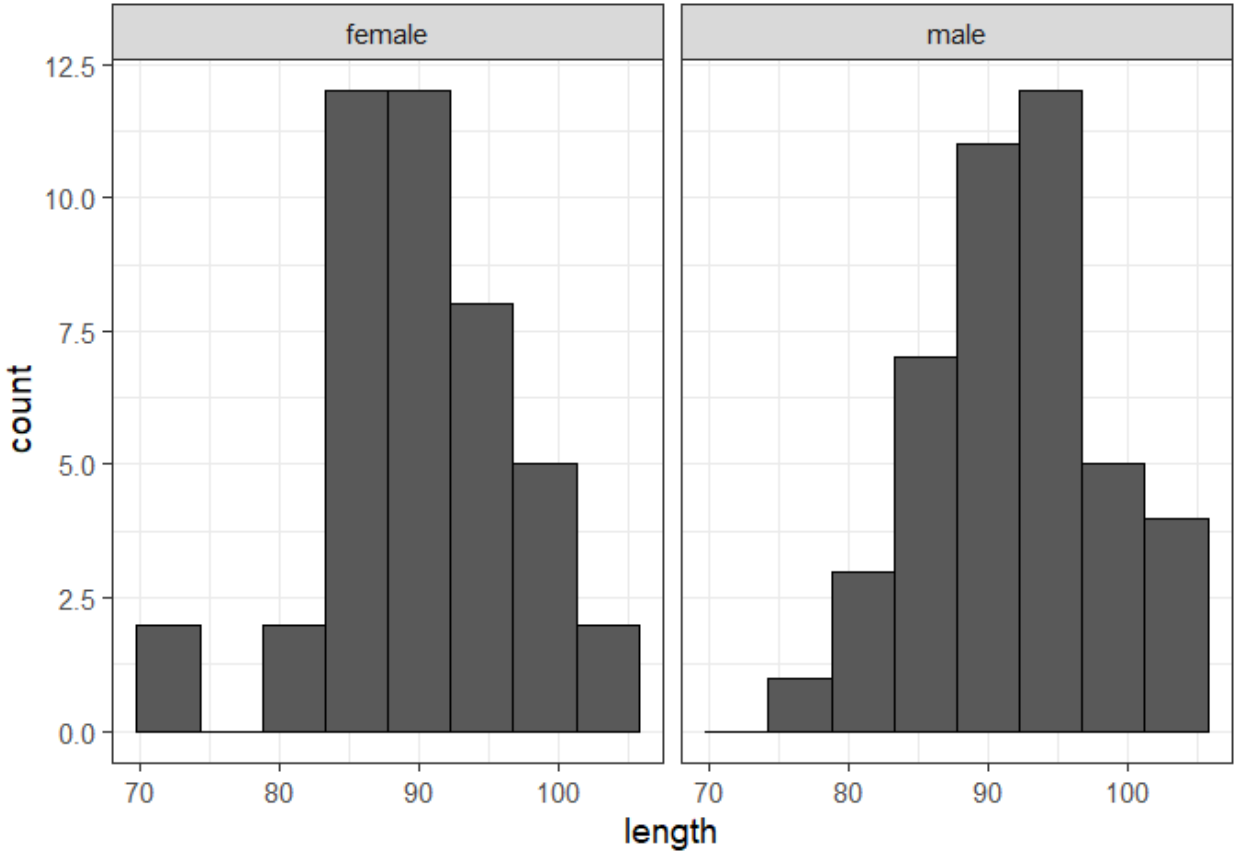



```
facet_grid(cols=vars(sex))
```

Example: Histogram

```
geom_histogram() +
facet_grid(rows=vars(row),cols=vars(column))
```

```
coyote %>%
  ggplot(aes(length)) +
  geom_histogram(binwidth = 4.5,
      colour="black",
      show.legend = FALSE) +
  facet_grid(cols=vars(sex))
```

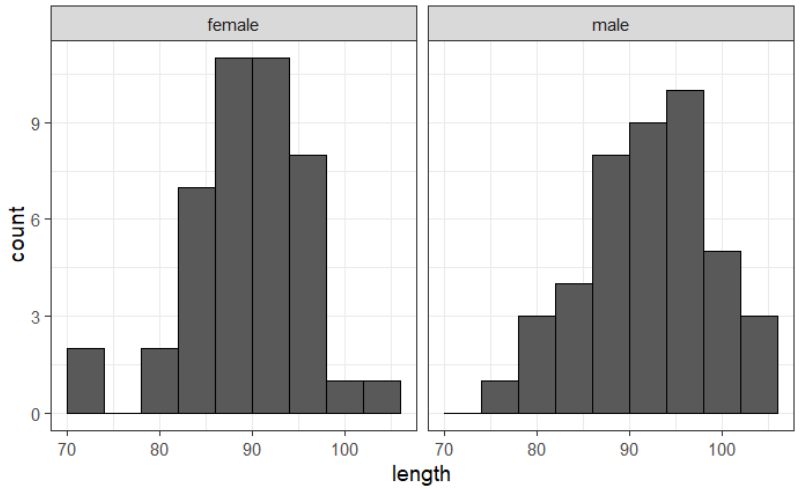


Example: Data exploration

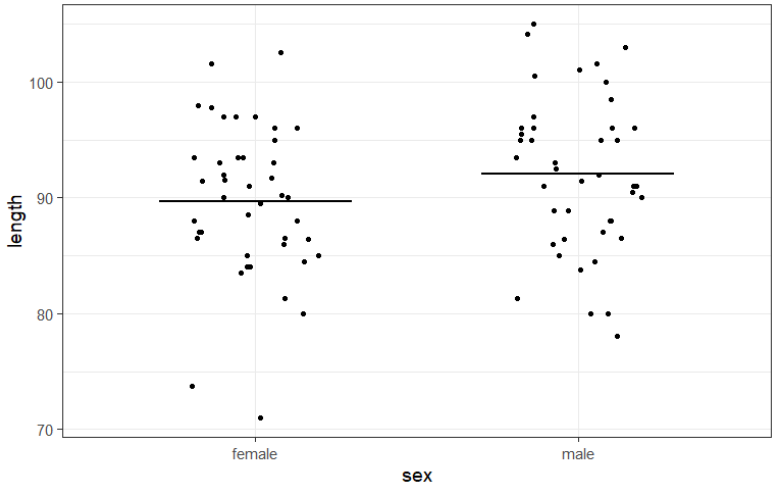
```
coyote %>%
  ggplot(aes(x=sex, y=length))+
  geom_...()
```

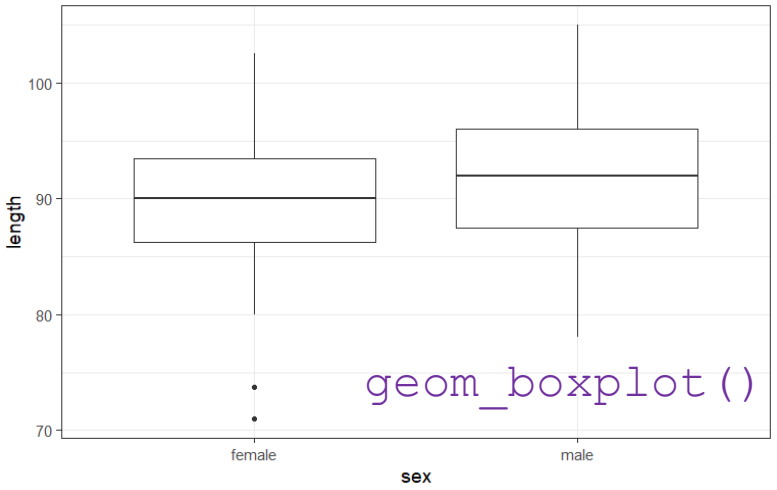
- Explore data using 4 different representations:

```
facet_grid(rows=vars(row),cols=vars(column))
geom_histogram()
```



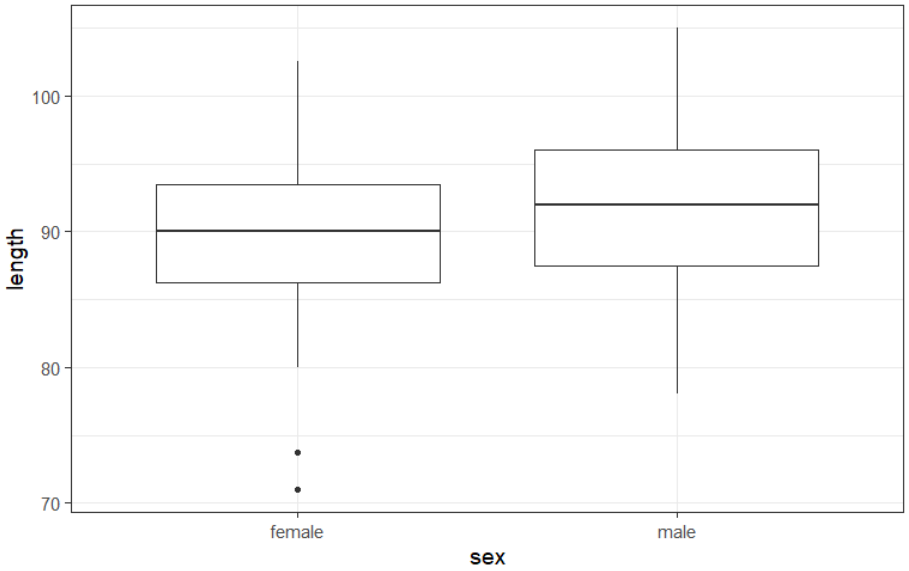

```
geom_jitter()
stat_summary(geom= "crossbar")
```

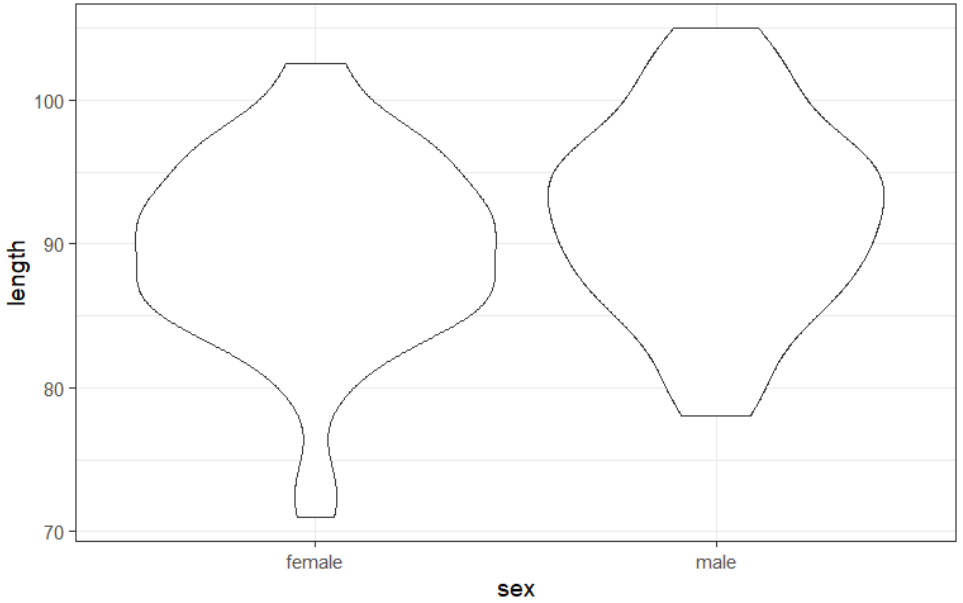




Example: Data exploration - Boxplots and violinplots

```
coyote %>%
  ggplot(aes(x=sex, y=length)) +
  geom_boxplot()
```

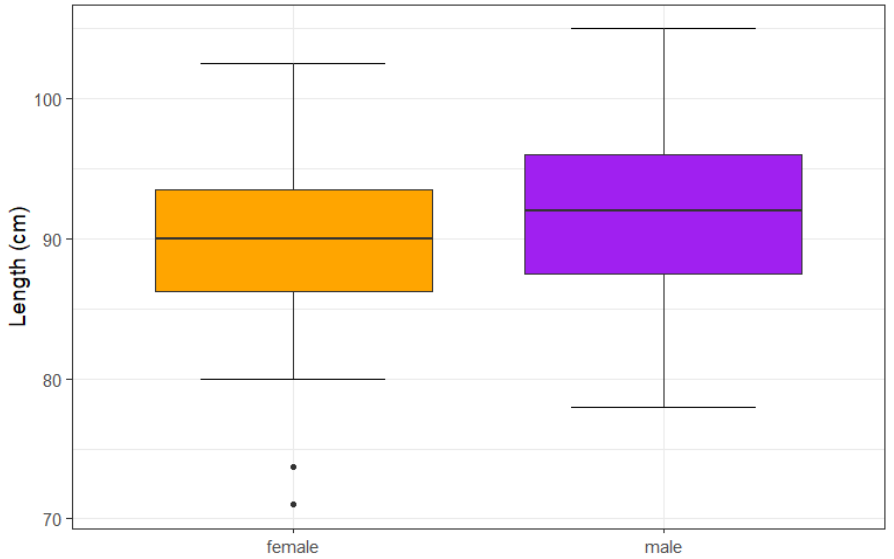


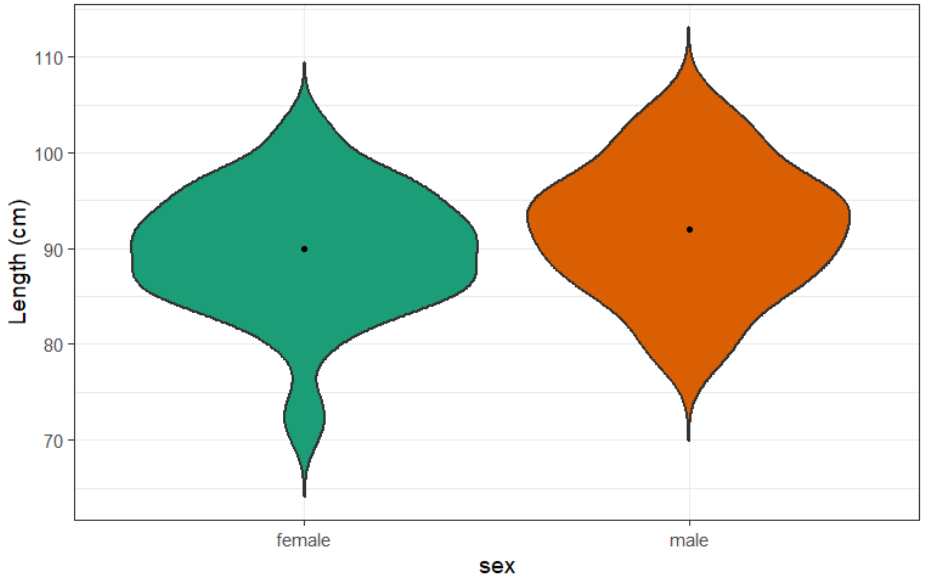



```
coyote %>%
  ggplot(aes(x=sex, y=length)) +
  geom_violin()
```

Example: Data exploration - Boxplots and violinplots

```
coyote %>%
  ggplot(aes(x=sex, y=length, fill=sex))+
        stat_boxplot(geom="errorbar", width=0.5)+
        geom_boxplot(show.legend=FALSE)+
        ylab("Length (cm)")+
        xlab(NULL)+
        scale_fill_manual(values = c("orange","purple"))
```

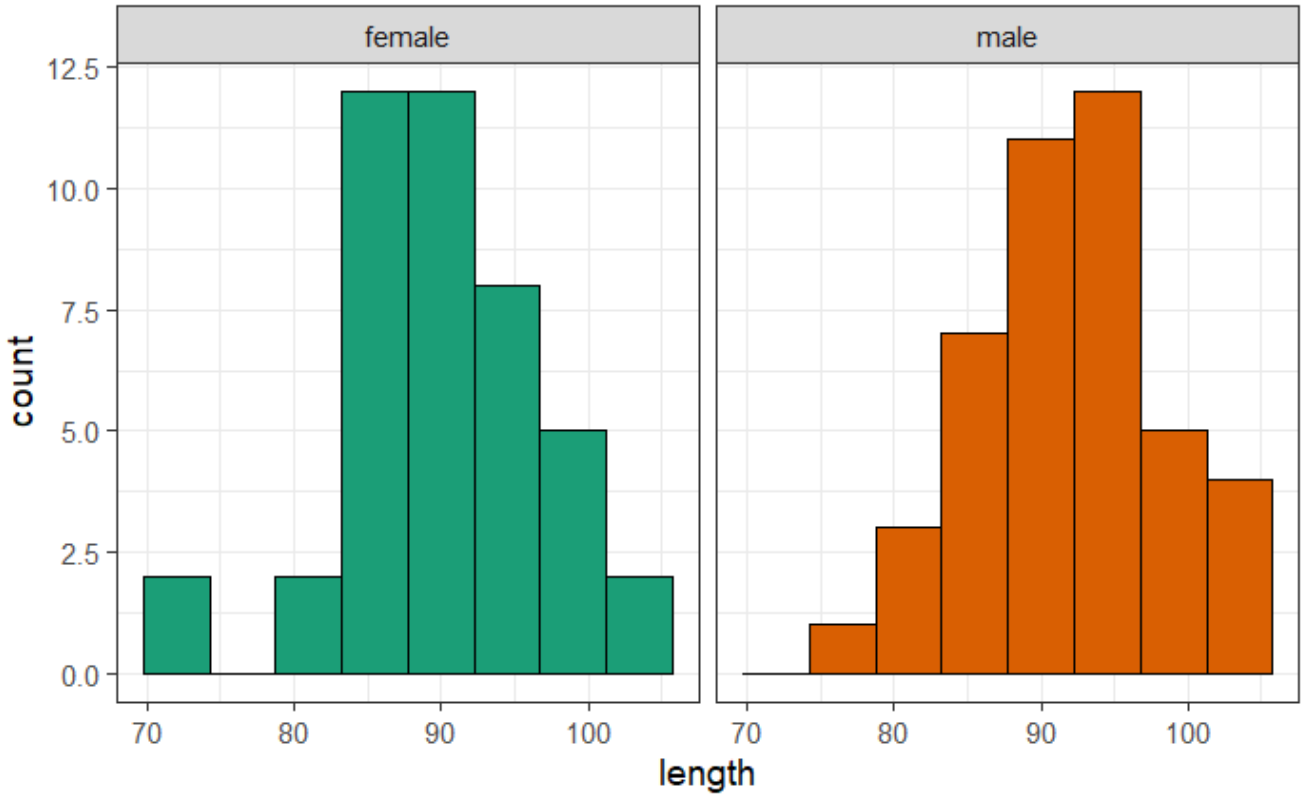





```
coyote %>%
  ggplot(aes(x=sex, y=length, fill=sex))+
    geom_violin(trim=FALSE, linewidth=1, show.legend=FALSE)+
    ylab("Length (cm)")+
    scale_fill_brewer(palette="Dark2")+
    stat_summary(geom = "point", fun = median, show.legend=FALSE)
```

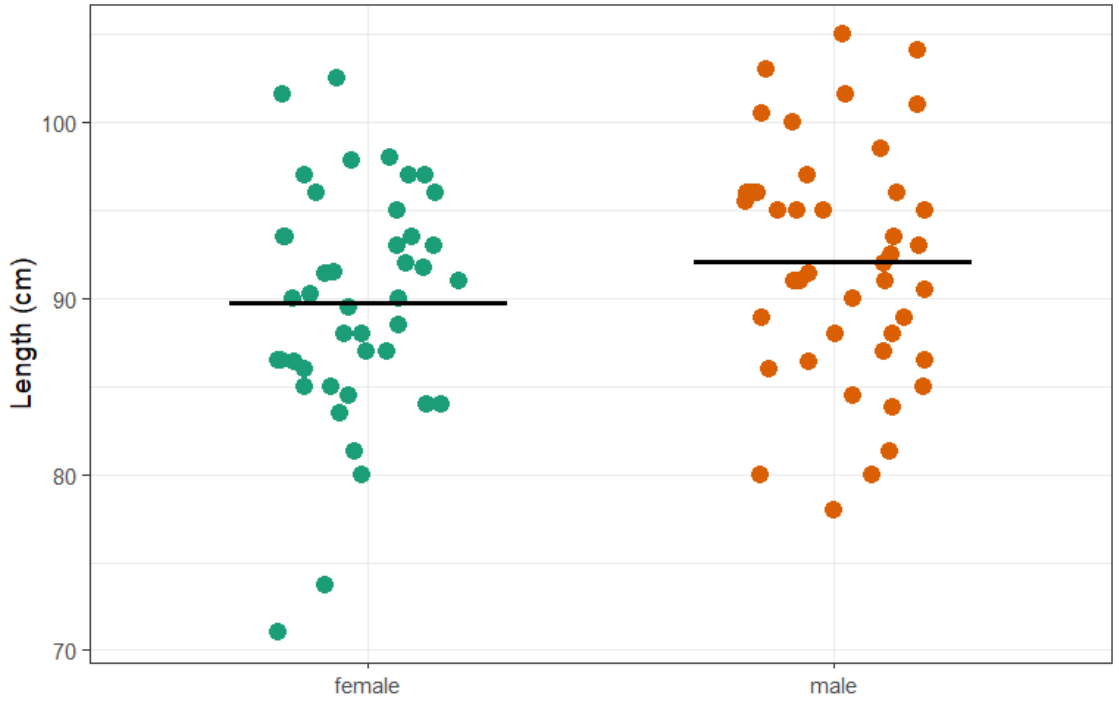
Example: Data exploration - Histograms

```
coyote %>%
  ggplot(aes(length, fill=sex))+
    geom_histogram(binwidth = 4.5, colour="black", show.legend = FALSE) +
    scale_fill_brewer(palette="Dark2")+
    facet_grid(cols=vars(sex))
```



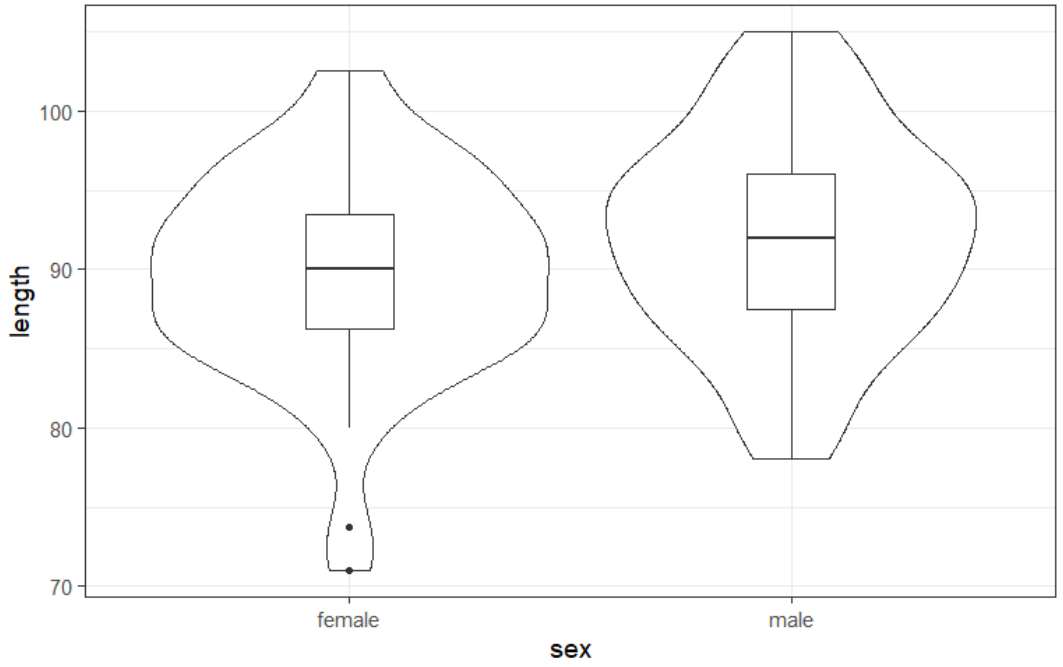
Example: Data exploration - Stripcharts

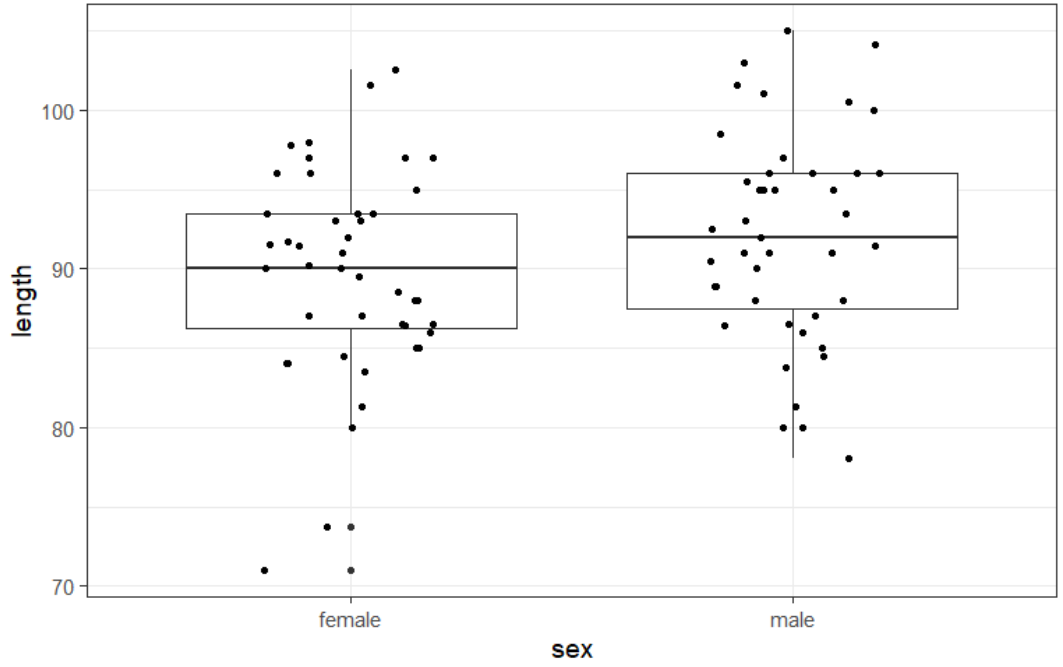
```
coyote %>%
  ggplot(aes(x=sex,y=length, colour=sex)) +
    geom_jitter(height=0, size=4, width=0.2, show.legend = FALSE) +
    ylab("Length (cm)")+
    scale_colour_brewer(palette="Dark2")+
    xlab(NULL)+
    stat_summary(geom="crossbar", fun=mean, colour="black", linewidth=0.5, width=0.6)
```



Example extra: Data exploration - Combinations/overlays

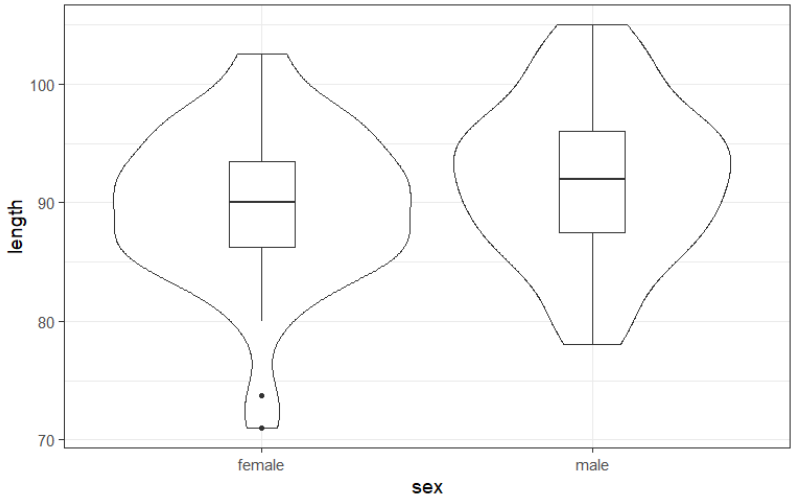
- Explore data using 2 different combinations/overlays of graphs

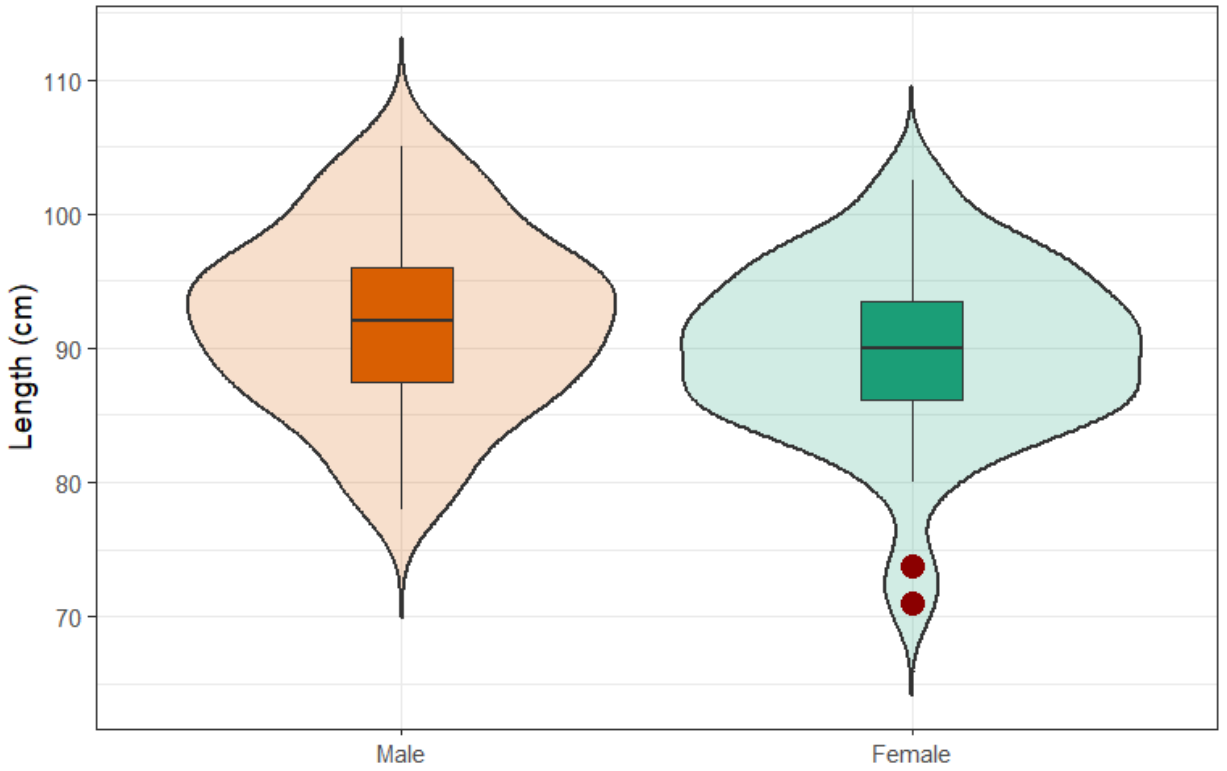




Example extra: Data exploration - Combinations/overlays

```
coyote %>%
  ggplot(aes(x=sex, y=length)) +
  geom_violin() +
  geom_boxplot(width=0.2)
```

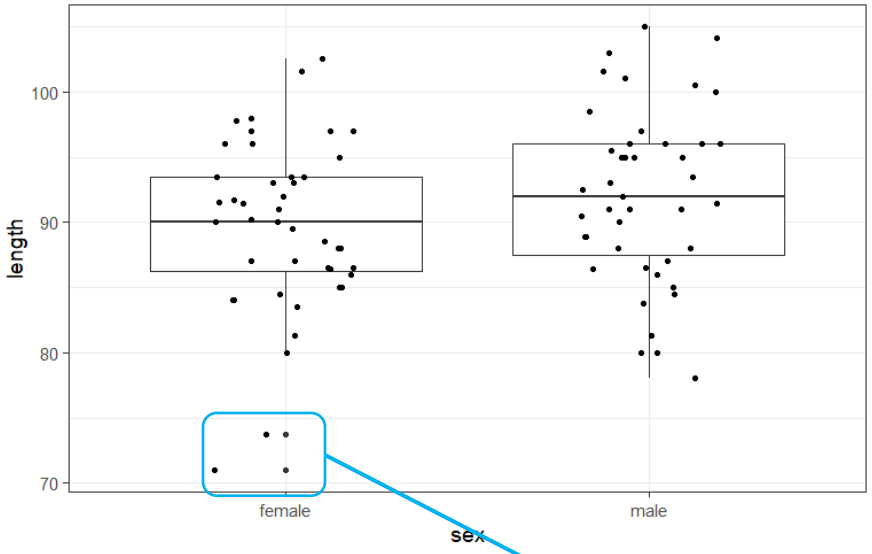


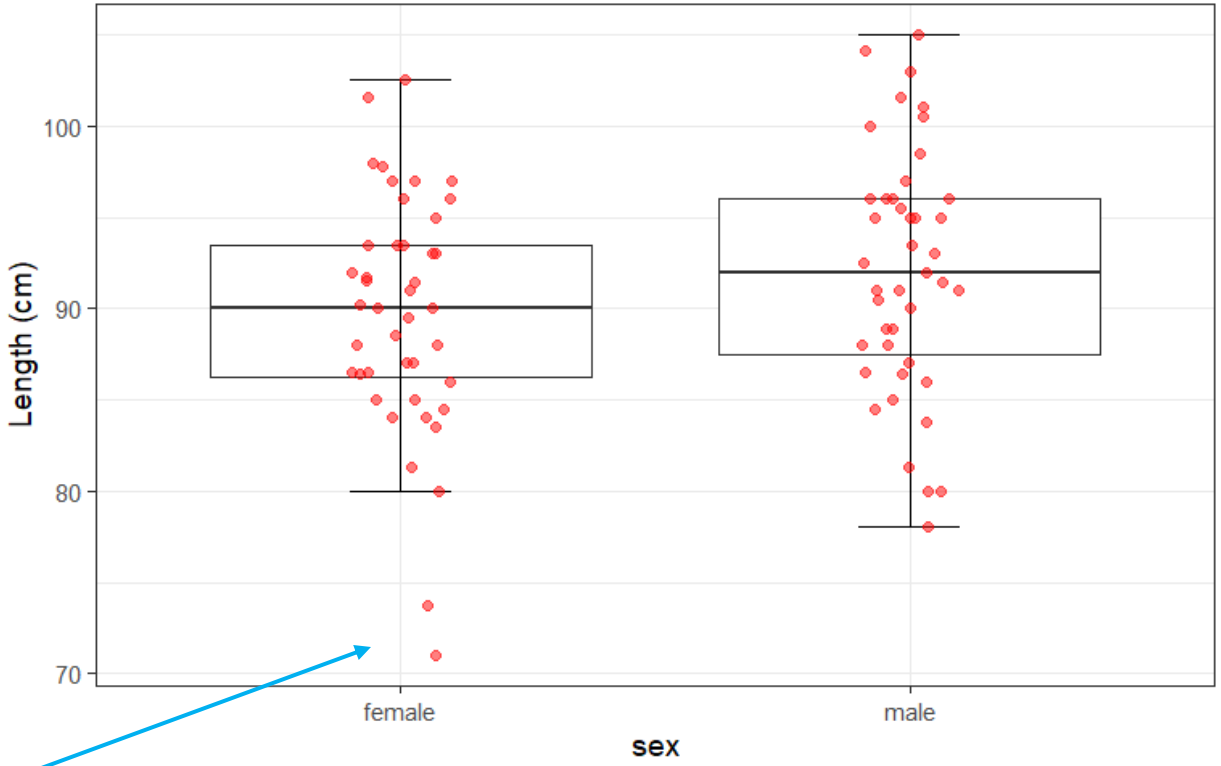



```
coyote %>%
  ggplot(aes(x=sex,y=length, fill=sex)) +
        geom_violin(linewidth=1, trim = FALSE, alpha=0.2, show.legend=FALSE) +
        geom_boxplot(width=0.2, outlier.size=5, outlier.colour = "darkred", show.legend=FALSE)+
        scale_fill_brewer(palette="Dark2")+
        ylab("Length (cm)")+
        xlab(NULL)+
        scale_x_discrete(labels=c("female"="Female", "male"="Male"), limits =c("male", "female"))
```

Example extra: Data exploration - Combinations/overlays

```
coyote %>%
  ggplot(aes(x=sex, y=length)) +
  geom_boxplot()+
  geom_jitter(height=0, width=0.2)
```





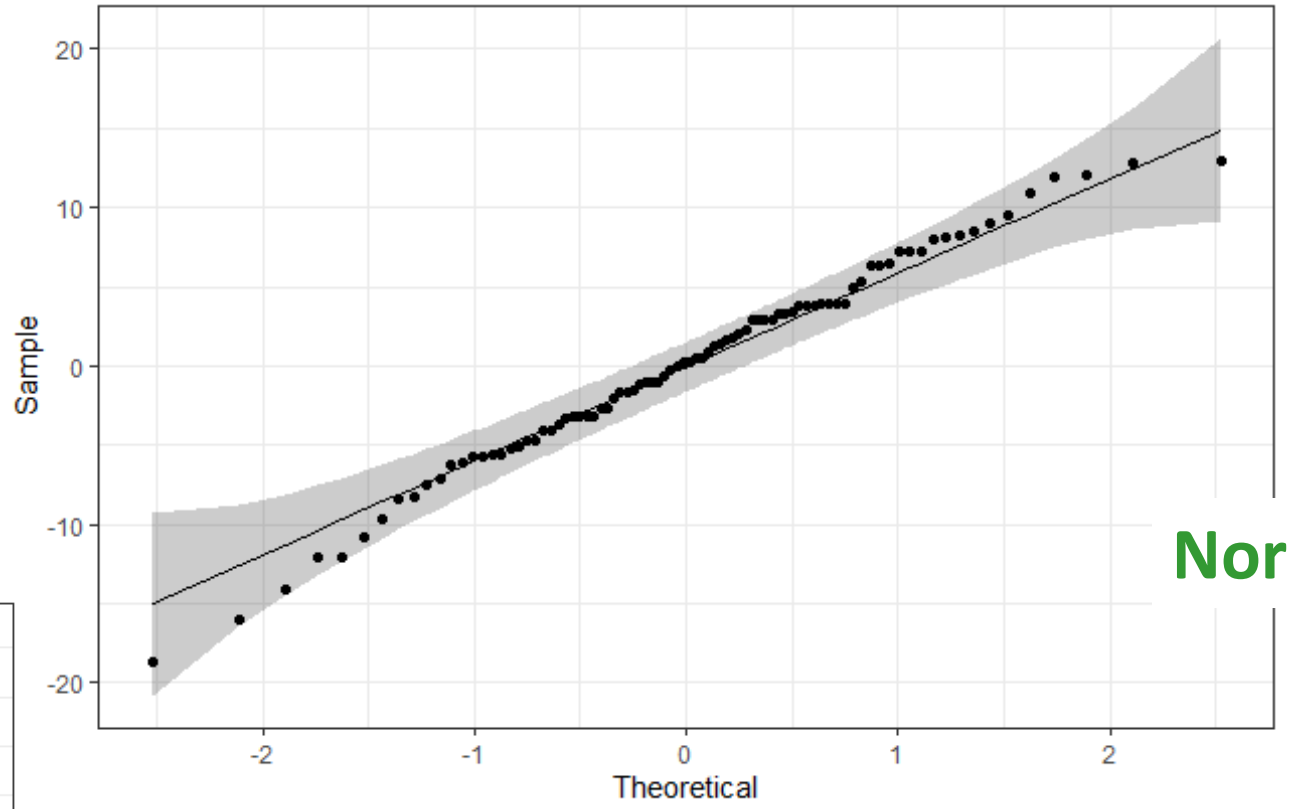

```
coyote %>%
  ggplot(aes(x=sex, y=length)) +
      geom_boxplot(outlier.shape=NA)+
      stat_boxplot(geom="errorbar", width=0.2)+
      geom_jitter(height=0, width=0.1, size=2, alpha=0.5, colour="red")+
      ylab("Length (cm)")
```

Checking the assumptions

Normality assumption: QQ Plot

QQ plot= Quantile – Quantile plot

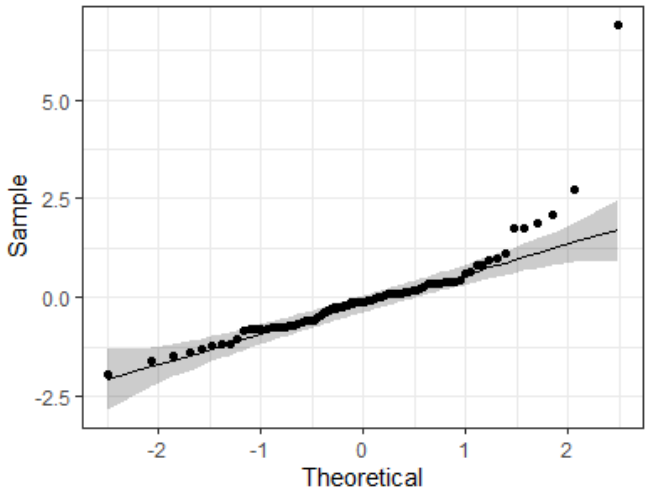
Our coyotes (residuals)



Theoretical normal distribution

Normality ☑ (ish)

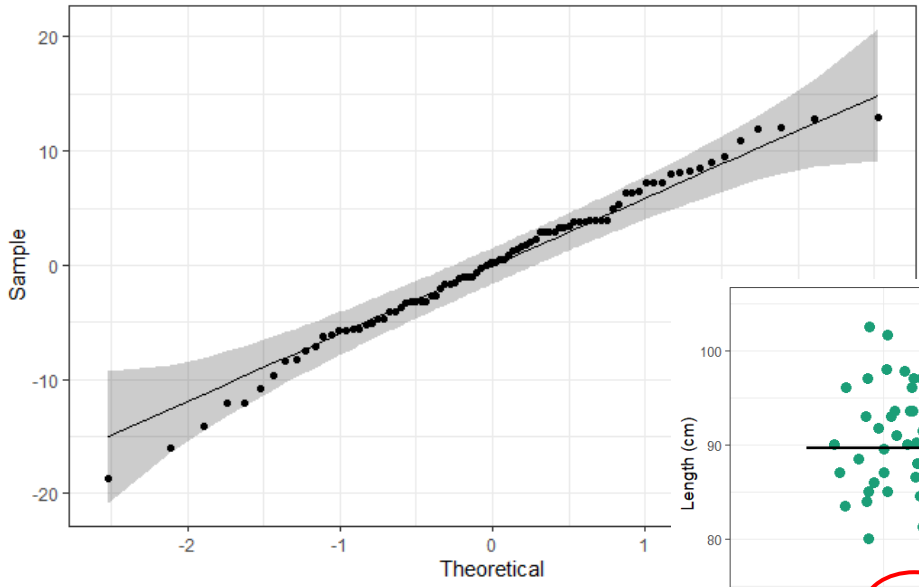
Poor QQ plot

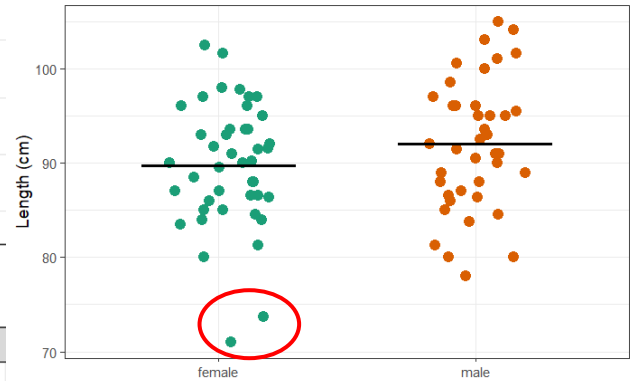


Equivalent dataset
Same sample size
Perfectly normal distribution

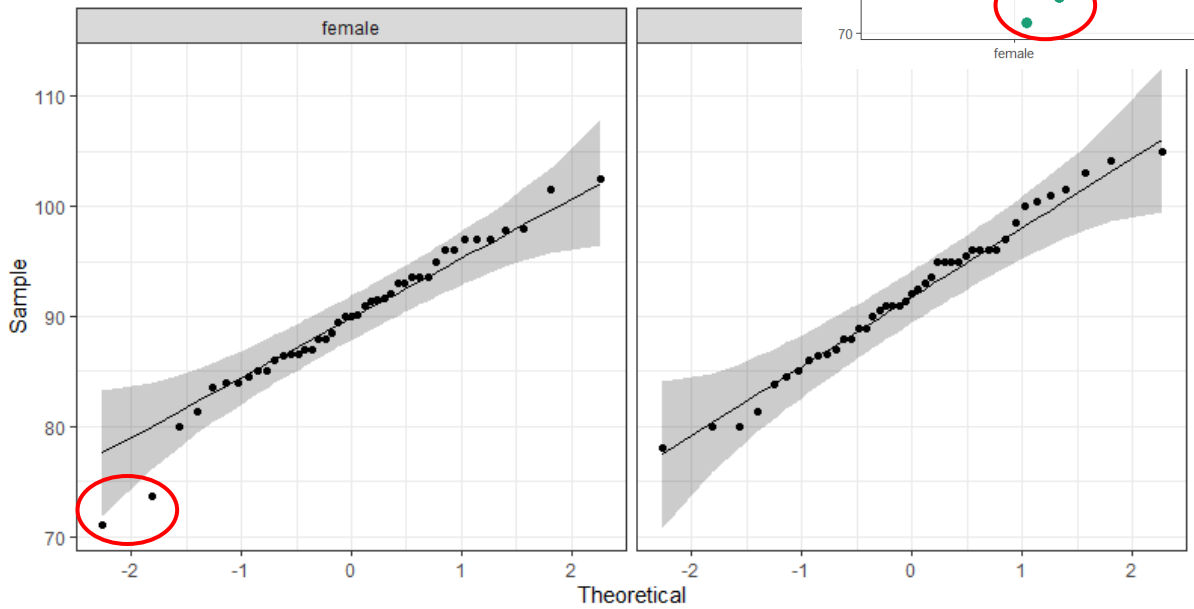
Normality assumption: QQ plot

```
model  <- aov(length ~ sex,
        data = coyote)
ggqqplot(residuals(model)) + theme_bw()
```






```
ggqqplot(coyote, x = "length",
        facet.by = "sex") +
   theme_bw()
```



Assumptions of Parametric Data

- First assumption: **Normality**
  - Shapiro-Wilk test `shapiro_test()` # rstatix package #
    - Based on the correlation between the data and the corresponding normal scores

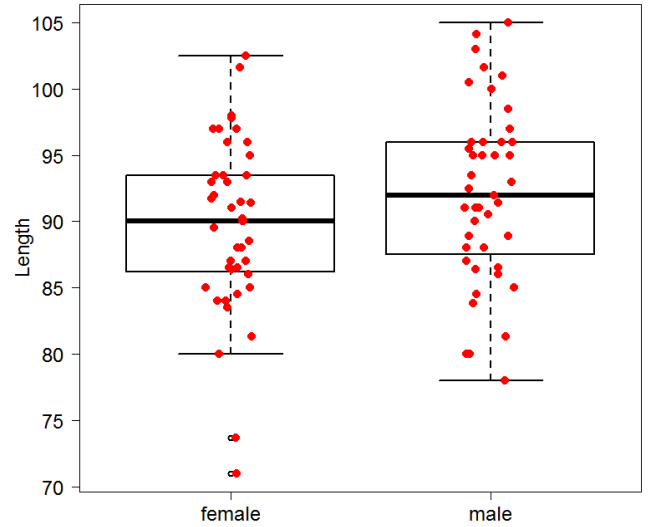
```
model  <- aov(length ~ sex,
              data = coyote)
shapiro_test(residuals(model))
```

variable	statistic	p.value
<chr>	*<dbl>*	*<dbl>*
residuals(model)	0.987	0.568

```
coyote %>%
  group_by(sex) %>%
   shapiro_test(length)
```

sex <chr>	variable <chr>	statistic <dbl>	p <dbl>
female	length	0.9700101	0.3164448
male	length	0.9844570	0.8189831

Normality ☑



Other options: Core R

- Second assumption: **Homoscedasticity**
  - Levene test `levene_test()`

```
coyote %>%
  levene_test(length ~ sex)
```

df1 <int>	df2 <int>	statistic <dbl>	p <dbl>
1	84	0.167929	0.6830022

Homogeneity in variance ☑

Normality

Other classic: D'Agostino-Pearson test

`dagoTest()` # fBasics package #

Homoscedasticity

More robust: Brown-Forsythe test

`bf.test()` # onewaytests package #

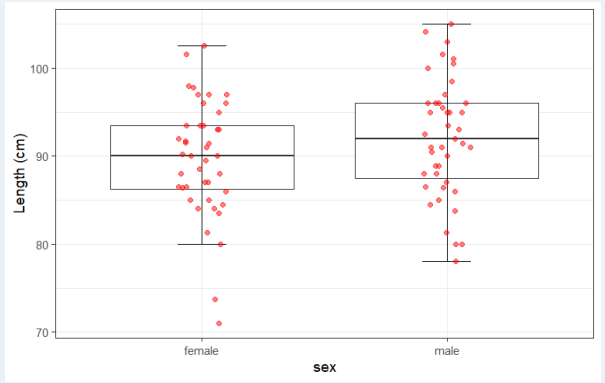
Other classic: Bartlett test

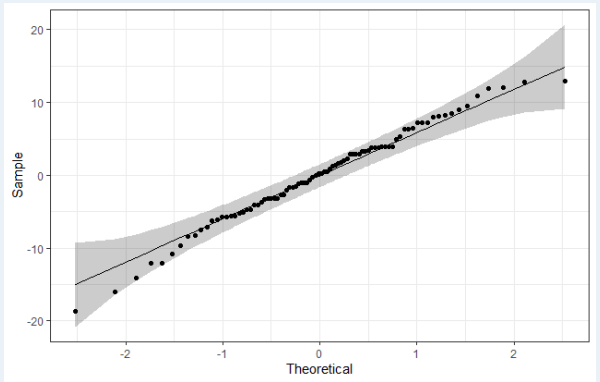
`bartlett.test()`

Independent Student's *t*-test

To recap

- **Data exploration and assumptions**






```
shapiro_test(residuals(model))
```

variable	statistic	p.value
<chr>	<dbl>	<dbl>
residuals(model)	0.987	0.568

Normality ☑

```
coyote %>%
  levene_test(length ~ sex)
```

df1 <int>	df2 <int>	statistic <dbl>	p <dbl>
1	84	0.167929	0.6830022

Homogeneity in variance ☑

Much more important/useful

- **Student's t-test** # rstatix package #

```
coyote %>%
       t_test(length ~ sex, var.equal = TRUE)
```

Independent Student's *t*-test: results

```
coyote %>%
       t_test(length~sex, var.equal = TRUE)
```

	.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>	statistic <dbl>	df <dbl>	p <dbl>
1	length	female	male	43	43	-1.641109	84	0.105

```
coyote %>%
  t_test(length~sex, var.equal = TRUE, detailed = TRUE)
```

estimate <dbl>	estimate1 <dbl>	estimate2 <dbl>	.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>	statistic <dbl>	p <dbl>	df <dbl>	conf.low <dbl>	conf.high <dbl>	method <chr>	alternative <chr>
-2.34	89.7	92.1	length	female	male	43	43	-1.64	0.105	84	-5.18	0.496	T-test	two.sided

```
coyote %>%
  group_by(sex) %>%
  get_summary_stats(length, type = "mean_se")
```

sex <chr>	variable <fct>	n <dbl>	mean <dbl>	se <dbl>
female	length	43	89.7	0.999
male	length	43	92.1	1.02

$$t = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{\left(s^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right)}}$$

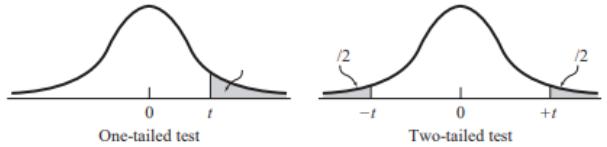
t = 89.7 – 92.1/SQRT(0.99^2+1.02^2)
= -1.64

- **Answer: Males tend to be longer than females but not significantly so (p=0.1045)**

Independent *t*-test: results

The old-fashion way

n1 <int>	n2 <int>	statistic <dbl>
43	43	-1.641109



	Level of Significance for One-Tailed Test								
	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.0005
	Level of Significance for Two-Tailed Test								
df	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.001
1	1.000	1.376	1.963	3.078	6.314	12.706	31.821	63.657	636.620
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	31.599
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	12.924
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	4.015
17	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.883
20	0.687	0.860	1.064	1.32					
21	0.686	0.859	1.063	1.32					
22	0.686	0.858	1.061	1.32					
23	0.685	0.858	1.060	1.31					
24	0.685	0.857	1.059	1.31					
25	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.725
26	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.707
27	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.690
28	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.674
29	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.659
30	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.646
40	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.551
50	0.679	0.849	1.047	1.299	1.676	2.009	2.403	2.678	3.496
100	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.390
∞	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.291

$t = 1.641 < 1.984$: not significant

Critical value

Independent *t*-test: results

Power!

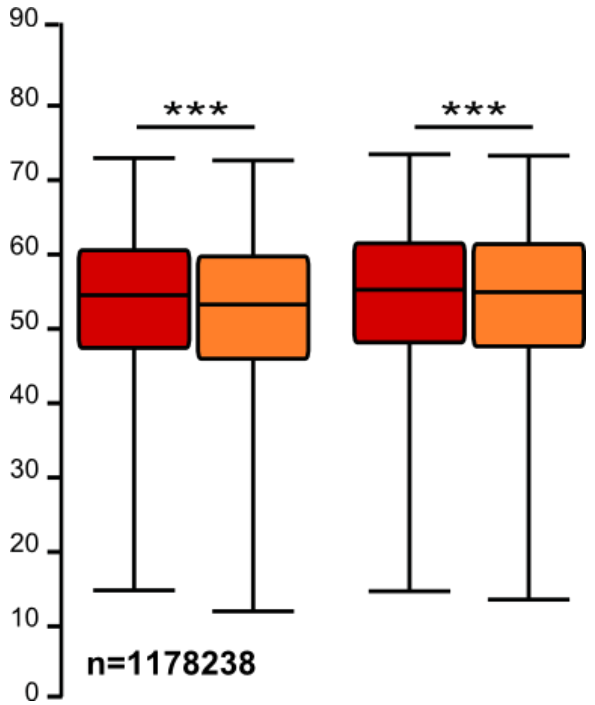
- Power: How many more coyotes to reach significance?

```
coyote %>%
        group_by(sex) %>%
        get_summary_stats(length, type = "mean_sd")
```

sex	variable	n	mean	sd
<chr>	<fct>	<dbl>	<dbl>	<dbl>
female	length	43	89.7	6.55
male	length	43	92.1	6.70

```
> power.t.test(delta=92.1-89.7, sd=6.7, sig.level=0.05, power=0.8)

     Two-sample t test power calculation

              n = 123.3067
          delta = 2.4
             sd = 6.7
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

With nearly 250 coyotes, we get a star

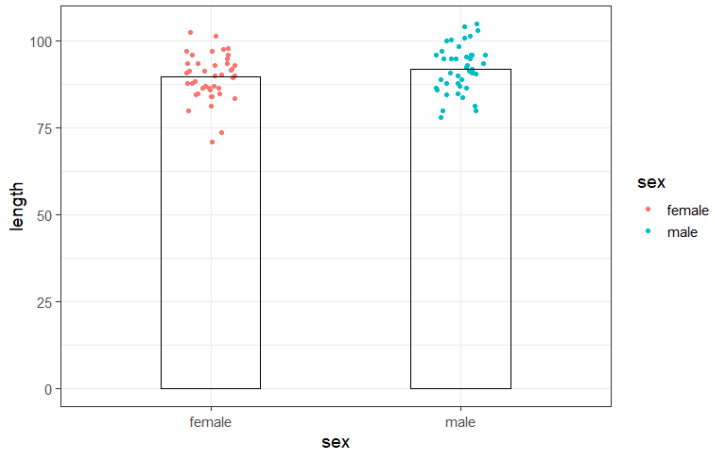


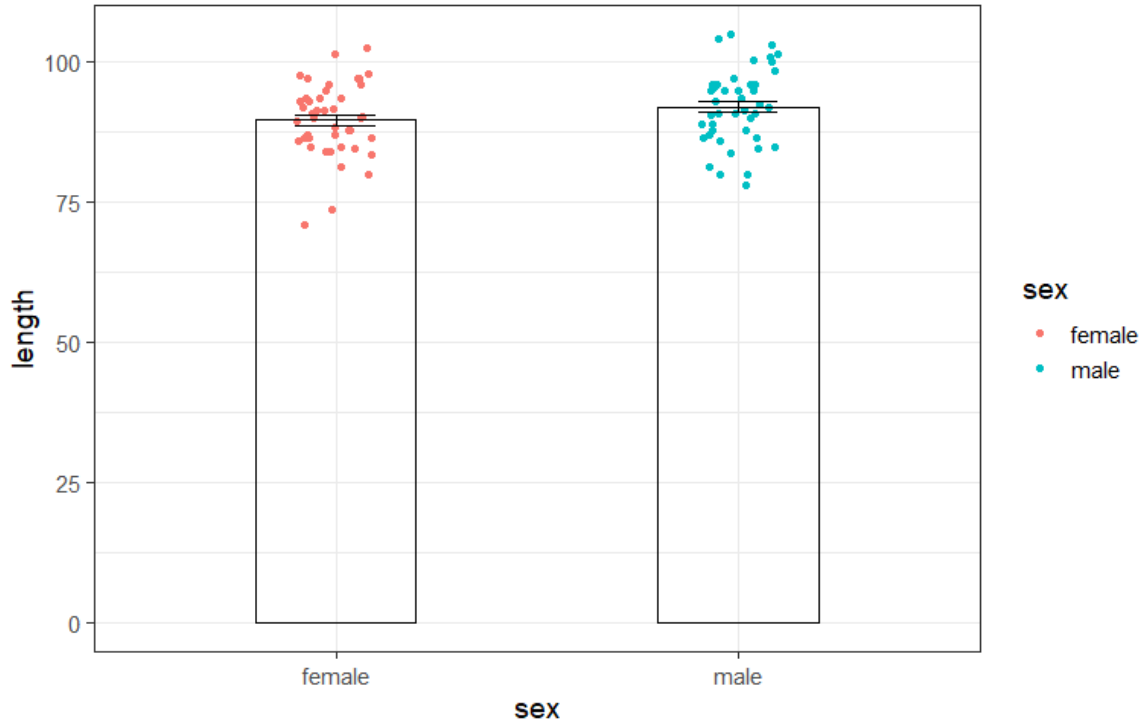
But does it make sense?

Independent *t*-test: results

Plotting the data

```
coyote %>%
  ggplot(aes(sex,length, colour=sex)) +
        geom_jitter(height=0, width=0.1)+
        geom_bar(stat = "summary", fun="mean", width=0.4, alpha=0, colour="black")
```





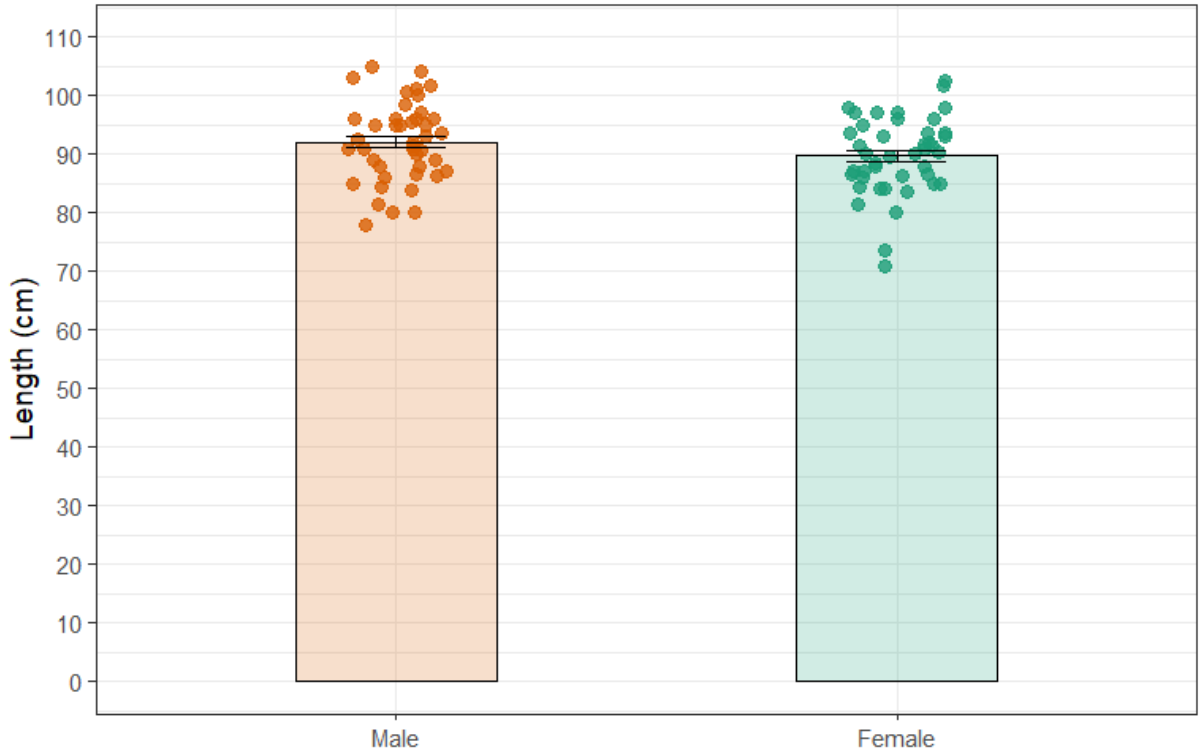
- Add error bars

```
coyote %>%
  ggplot(aes(sex,length, colour=sex)) +
    geom_jitter(height=0, width=0.1)+
    geom_bar(stat = "summary", fun="mean", width=0.4, alpha=0, colour="black")+
    stat_summary(geom="errorbar", colour="black", width=0.2)
```

Independent *t*-test: results

Plotting the data

- Prettier version




```
coyote %>%
  ggplot(aes(sex,length, colour=sex, fill=sex)) +
  geom_jitter(height=0, width=0.1, show.legend=FALSE, size=3, alpha=0.8)+
  geom_bar(stat="summary", fun="mean", width=0.4, alpha=0.2, colour="black", show.legend=FALSE)+
  stat_summary(geom="errorbar", colour="black", width=0.2)+
  scale_colour_brewer(palette="Dark2")+
  scale_fill_brewer(palette="Dark2")+
  theme(legend.position = "none")+
  scale_x_discrete(limits = c("male", "female"), labels = c("male"="Male", "female"="Female"))+
  scale_y_continuous(breaks=c(seq(0,110,10)), limits = c(0, 110))+
  xlab(NULL)+
  ylab("Length (cm)")
```

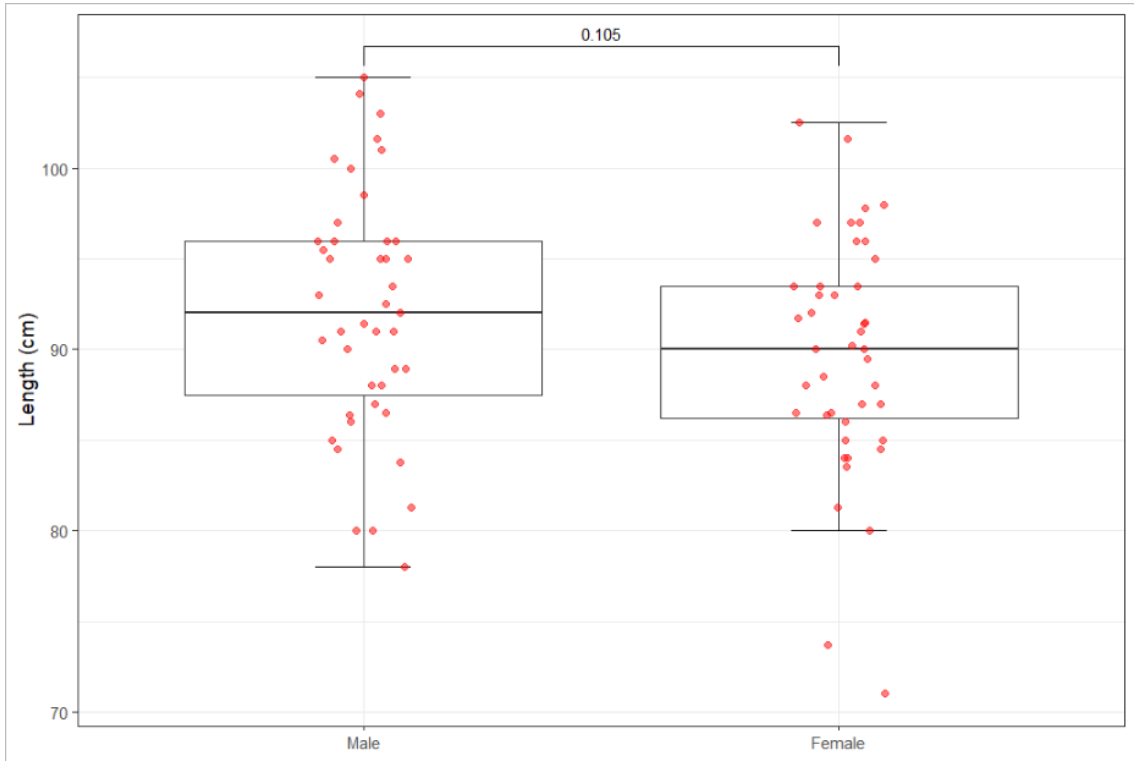
Independent *t*-test: results

Plotting the data

ggsignif package

```
t_results <- coyote %>%
       t_test(length~sex, var.equal = TRUE)
```

```
t_results                                    1 c
  $ .y.       : chr "length"
  $ group1    : chr "female"
  $ group2    : chr "male"
  $ n1        : int 43
  $ n2        : int 43
  $ statistic: Named num -1.64
   ..- attr(*, "names")= chr "t"
  $ df        : Named num 84
   ..- attr(*, "names")= chr "df"
  $ p         : num 0.105
```

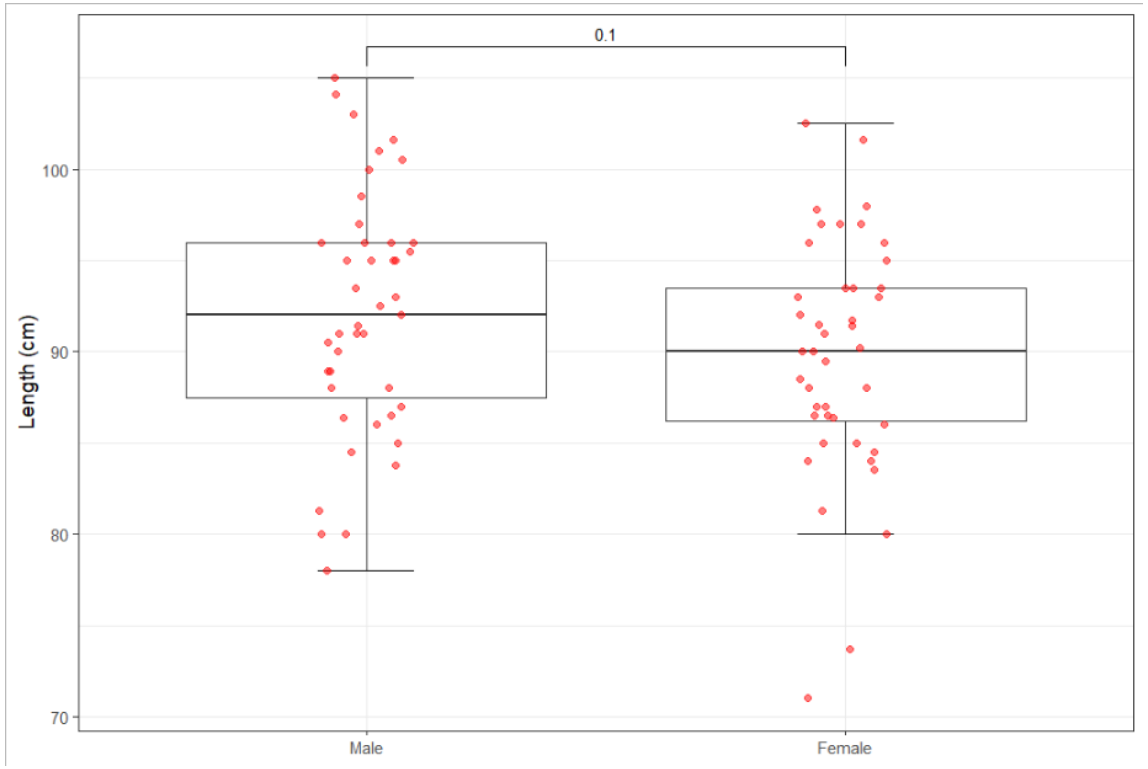



```
coyote %>%
  ggplot(aes(sex, length)) +
  stat_boxplot(geom="errorbar", width=0.2)+
  geom_boxplot(outlier.shape = NA)+
  geom_jitter(height=0, width=0.1, size = 2, alpha = 0.5, colour="red")+
  scale_x_discrete(limits = c("male", "female"), labels = c("male"="Male", "female"="Female"))+
  ylab("Length (cm)")+
  xlab(NULL)+
  geom_signif(comparisons = list(c("female", "male")), annotations = t_results$p)
```

Independent *t*-test: results

Plotting the data

ggsignif package



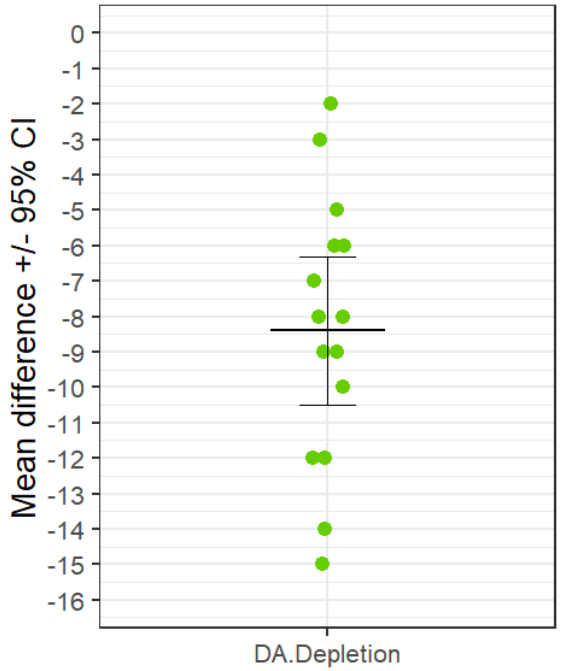
This also works but there is less control on the test.

```
coyote %>%
  ggplot(aes(sex, length)) +
  stat_boxplot(geom="errorbar", width=0.2)+
  geom_boxplot(outlier.shape = NA)+
  geom_jitter(height=0, width=0.1, size = 2, alpha = 0.5, colour="red")+
  scale_x_discrete(limits = c("male", "female"), labels = c("male"="Male", "female"="Female"))+
  ylab("Length (cm)")+
  xlab(NULL)+
  geom_signif(comparisons = list(c("female", "male")), test = "t.test")
```

Dependent or Paired *t*-test

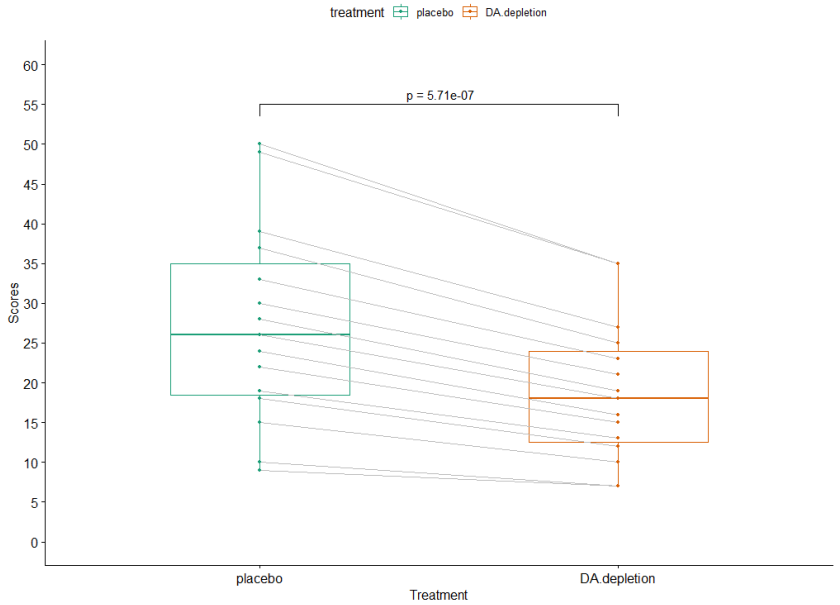
- For paired t-test there are 2 ways of approaching:
  - Calculate differences and use these as input to a one sample t-test

```
working.memory <- working.memory %>%
  mutate(difference = DA.depletion - placebo)
working.memory %>%
  t_test(difference ~ 1, mu=0, detailed = TRUE)
```



  - Using paired version of `t_test()` on long form data

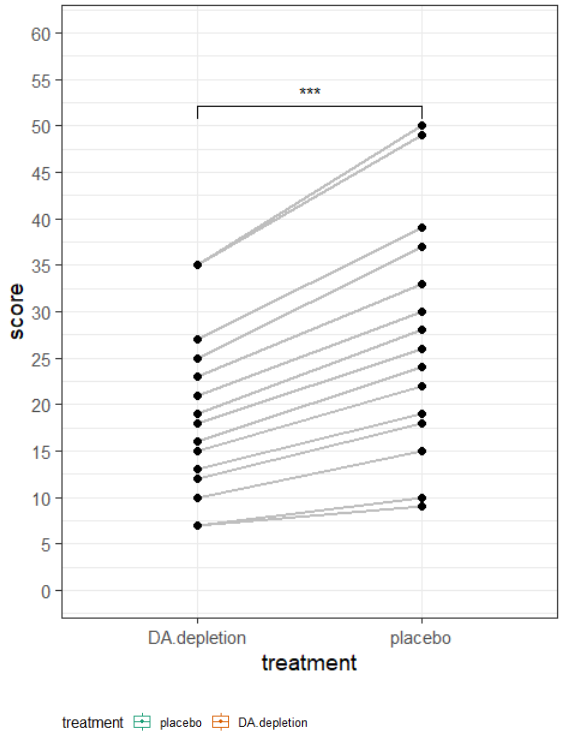
```
working.memory.long <- working.memory %>%
  pivot_longer(cols= 2:3, names_to = "treatment",
       values_to = "scores")
working.memory.long %>%
  arrange(Subject) %>%
  t_test(scores ~ treatment, paired = TRUE)-> stat.test
```



Dependent or Paired *t*-test

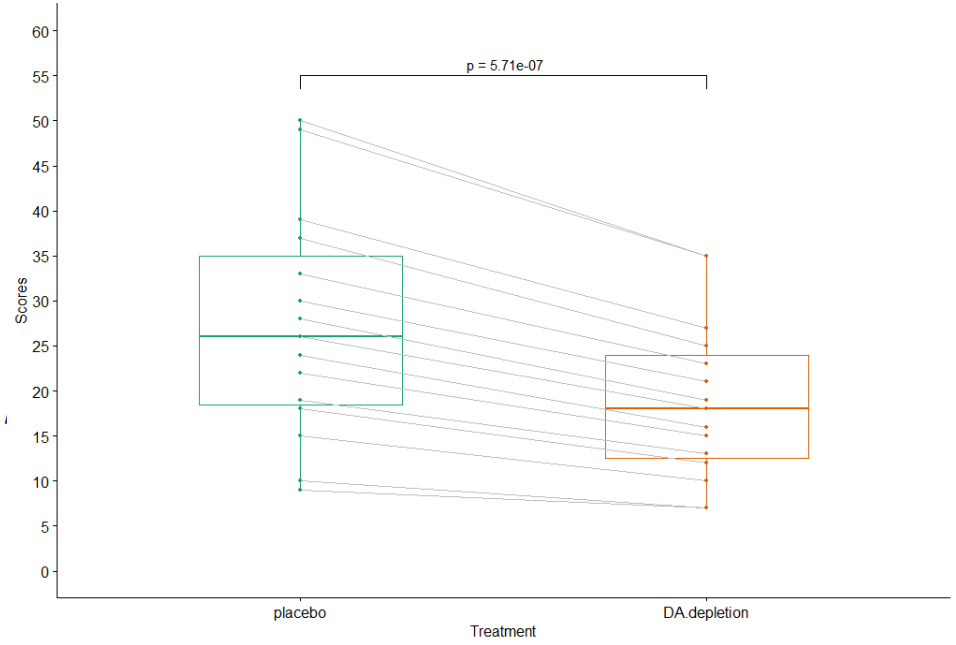
- Means also two ways of plotting – plotting differences or paired plots

```
working.memory.long %>%
  arrange(Subject) %>%
  ggplot(aes(x=treatment, y=score, group=Subject))+
  geom_line(linewidth=1, colour = "grey")+
  geom_point(colour= "black", size = 2) +
  scale_y_continuous(breaks=seq(from =0, by=5, to=60),
        limits = c(0,60)) +
  geom_signif(comparisons = list(c("placebo", "DA.depletion")),
        test = "t.test", test.args = list(paired=TRUE),
        map_signif_level = TRUE)
```



# ggpubr package #
ggline()

```
working.memory.long %>%
  ggpaired(x = "treatment", y = "scores",
           color = "treatment", id = "Subject",
           palette = "Dark2", line.color = "gray",
           line.size = 0.4,
           xlab = "Treatment", ylab = "Scores")+
         scale_y_continuous(breaks=seq(from =0, by=5, to=60),
                            limits = c(0,60))+
           stat_pvalue_manual(stat.test, label="p = {p}",
                              y.position = 55)
```



Extra R: changing format

Simon Andrews, Anne Segonds-Pichon

v2021-09

Data file format: Example

Wide format

3 WT mice

3 KO mice

	Gene	WT_1	WT_2	WT_3	KO_1	KO_2	KO_3
Gene 1	ABC1	8.86	4.18	8.90	4.00	14.52	13.39
Gene 2	DEF1	29.60	41.22	36.15	11.18	16.68	1.64

Long format

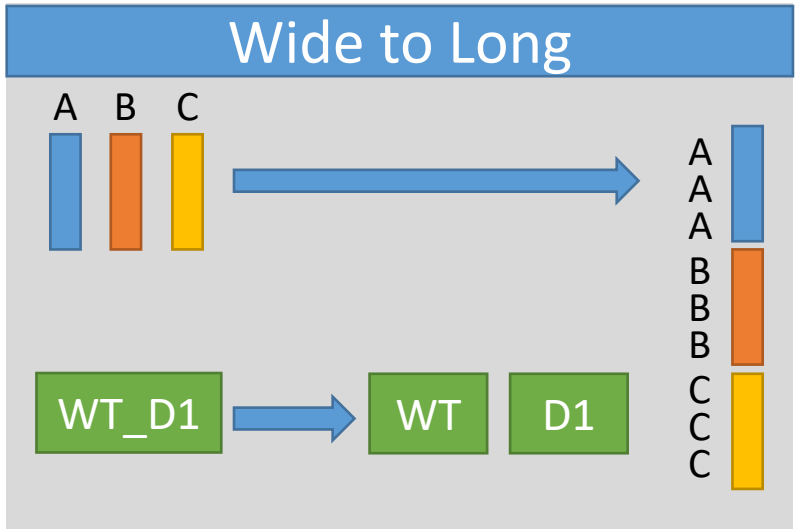
3 WT mice

	Gene	Genotype	Replicate	Value
Gene 1	ABC1	WT	1	8.86
	ABC1	WT	2	4.18
	ABC1	WT	3	8.90
	ABC1	KO	1	4.00
	ABC1	KO	2	14.52
	ABC1	KO	3	13.39
Gene 2	DEF1	WT	1	29.60
	DEF1	WT	2	41.22
	DEF1	WT	3	36.15
	DEF1	KO	1	11.18
	DEF1	KO	2	16.68
	DEF1	KO	3	1.64

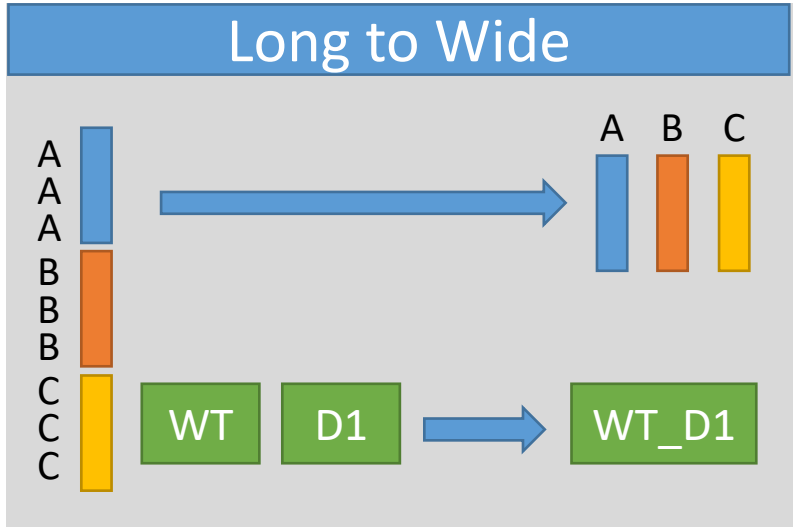
3 KO mice

Converting between formats: Tidying operations

- `pivot_longer()`
  - Takes multiple columns of the same type and puts them into a pair of key-value columns
- `separate`
  - Splits a delimited column into multiple columns



- `pivot_wider()`
  - Takes a key-value column pair and spreads them out to multiple columns of the same type
- `unite`
  - Combines multiple columns into one



Converting to ‘tidy’ format
wide to long

```
> working.memory
# A tibble: 15 x 4
   Subject placebo DA.depletion
   <chr>     <dbl>        <dbl>
 1 M1            9            7
 2 M2           10            7
 3 M3           15           10
 4 M4           18           12
 5 M5           19           13
 6 M6           22           15
 7 M7           24           16
 8 M8           26           18
 9 M9           28           19
10 M10          30           21
11 M11          33           23
12 M12          37           25
13 M13          39           27
14 M14          49           35
15 M15          50           35
```

```
# A tibble: 30 x 3
   Subject treatment    scores
   <chr>   <chr>         <dbl>
 1 M1      placebo           9
 2 M1      DA.depletion      7
 3 M2      placebo          10
 4 M2      DA.depletion      7
 5 M3      placebo          15
 6 M3      DA.depletion     10
 7 M4      placebo          18
 8 M4      DA.depletion     12
 9 M5      placebo          19
10 M5      DA.depletion     13
# ... with 20 more rows
```

```
working.memory %>%
  pivot_longer(cols= 2:3, names_to = "treatment", values_to = "scores")
```

Exercise 2

Babraham Institute

Analysis of Quantitative data

One-Way ANOVA

Hayley Carr & Anne Segonds-Pichon

v2025-02

Babraham Bioinformatics

Analysis of Quantitative data
One-Way + Two-Way ANOVA

- One-way ANOVA
  - Independent design
  - Repeated measures design

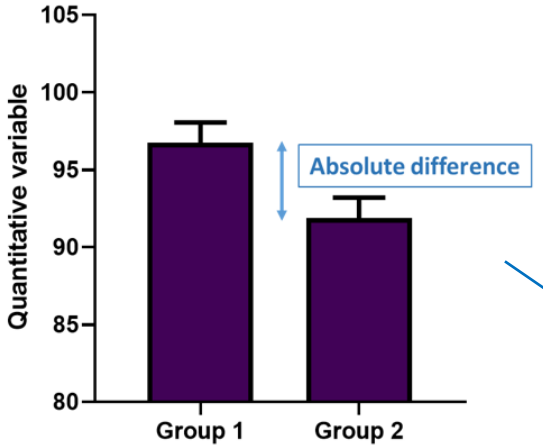
- Two-way ANOVA (two factors/predictors)
  - Tests **each factor** and **interactions** between them
  - Independent design
  - Repeated measures design (time series)

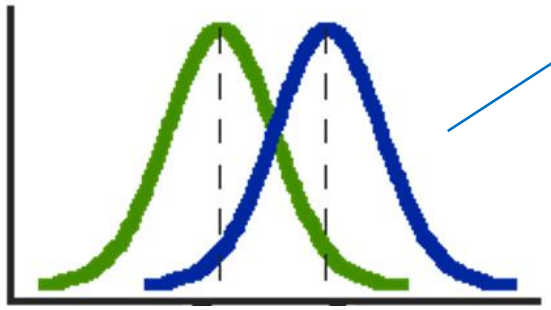
Comparison between more than 2 groups

One factor = One predictor

One-Way ANOVA

Signal-to-noise ratio





$$\frac{\text{Difference (signal)}}{\text{Variation (noise)}}$$

$$\frac{\text{Signal}}{\text{Noise}} = \textbf{statistical significance}$$

$$\frac{\text{Signal}}{\text{Noise}} = \textbf{no statistical significance}$$

Analysis of variance: how does it work?

$$\frac{\text{Signal}}{\text{Noise}} = \frac{\text{Difference between the means}}{\text{Variability in the groups}}$$

$$= \text{F ratio}$$

- If the variance amongst sample means is greater than the error/random variance, then F>1
  - In an ANOVA, we test whether F is significantly higher than 1 or not

One-Way Analysis of variance

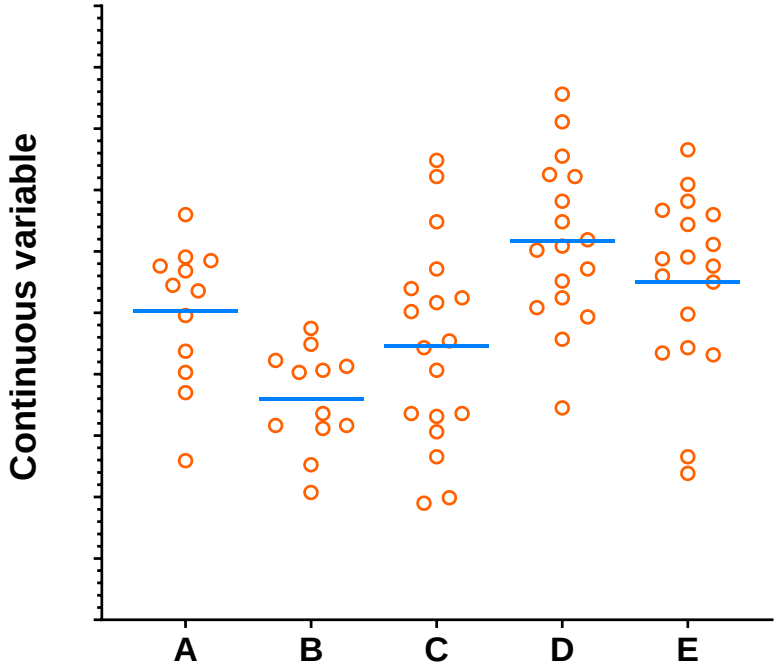
Step 1: Omnibus test

- It tells us if there is a difference between the means but not which means are significantly different from which other ones

Step 2: Post-hoc tests

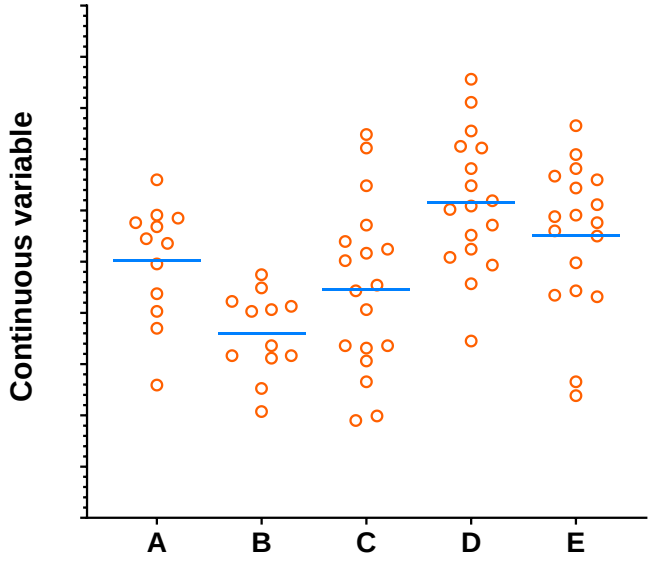
- Tell us if there are differences between the group means pairwise
- A correction for multiple comparisons will be applied on the p-values
- *Should only be used when the ANOVA finds a significant effect*

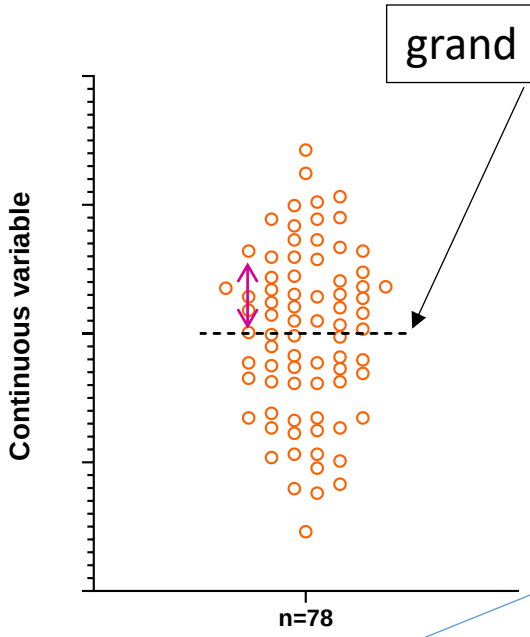
Analysis of variance: how does it work?



Source of variation	Sum of Squares	df	Mean Square	F	p-value
Between Groups	18.1	4	4.5	6.32	0.0002
Within Groups	51.8	73	0.71		
Total	69.9				

Analysis of variance: how does it work?





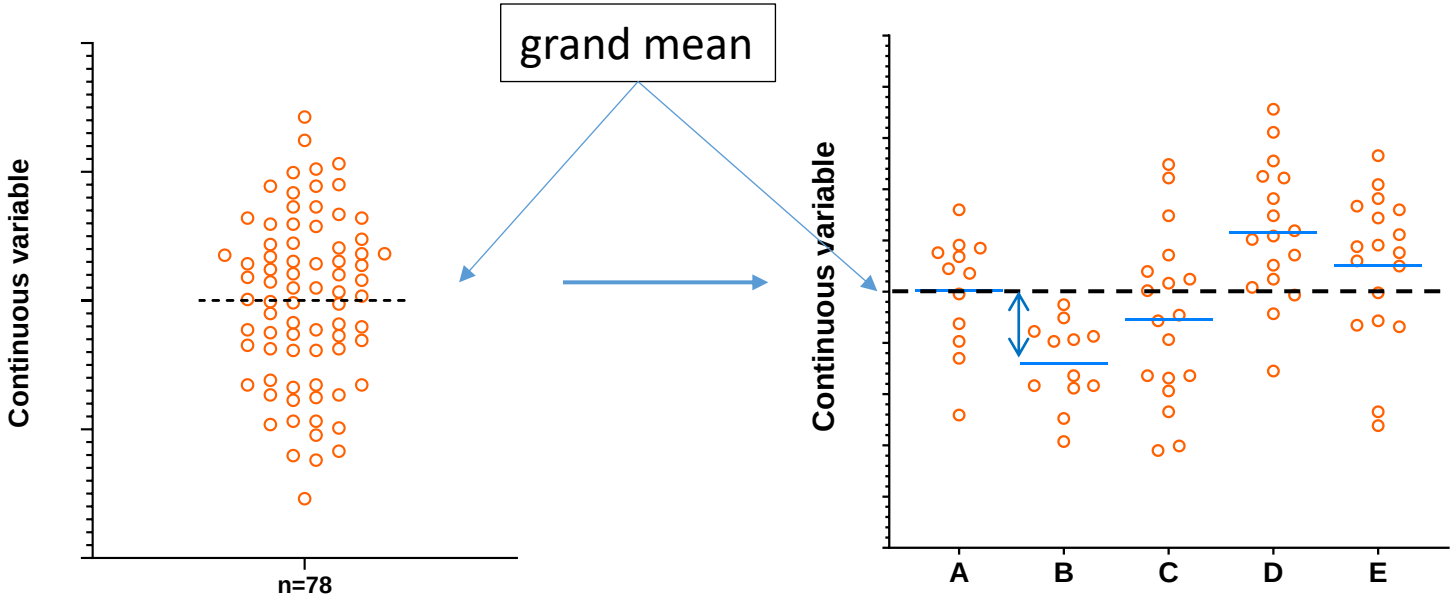
78 differences: $\sum_1^{78}(\text{value}_n - \text{grand mean})^2$

=

Sum of squared errors

Source of variation	Sum of Squares	df	Mean Square	F	p-value
Between Groups					
Within Groups					
Total	69.9				

Analysis of variance: how does it work?



5 differences: $\sum_1^5 (\text{mean}_n - \text{grand mean})^2$

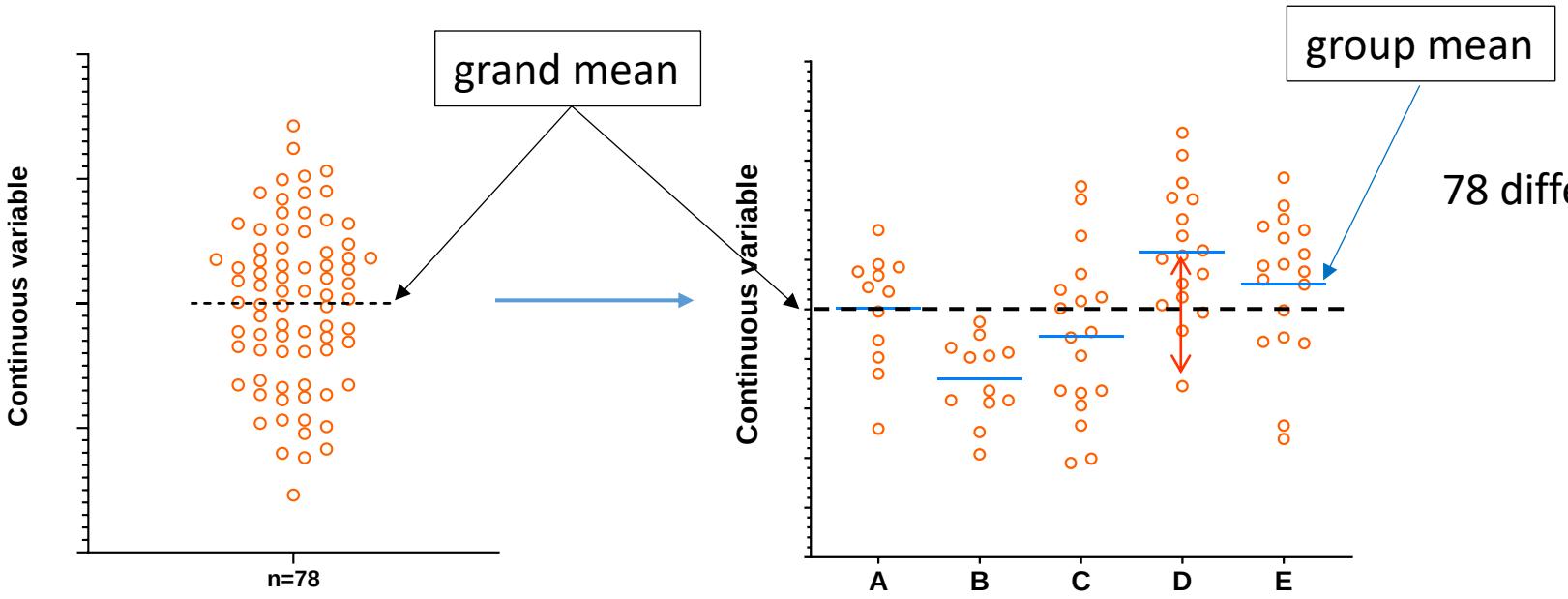
=

Sum of squared errors

Between the groups

Source of variation	Sum of Squares	df	Mean Square	F	p-value
Between Groups	18.1				
Within Groups					
Total	69.9				

Analysis of variance: how does it work?



78 differences: $\sum_1^{78}(\text{value}_n - \text{group mean})^2$

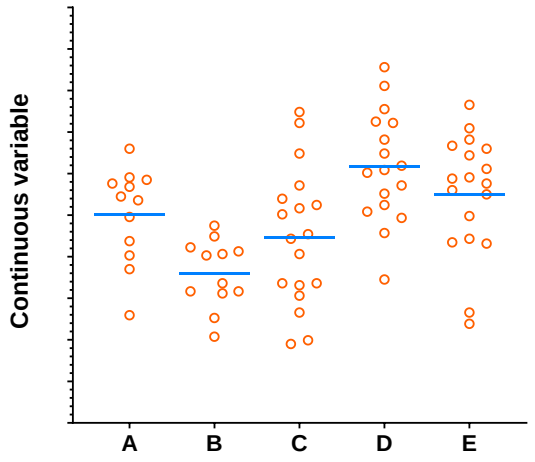
=

Sum of squared errors

Within the Groups

Source of variation	Sum of Squares	df	Mean Squares	F	p-value
Between Groups	18.1				
Within Groups	51.8				
Total	69.9				

Analysis of variance: how does it work?



	Source of variation	Sum of Squares	df	Mean Squares	F ratio	p-value
Signal	Between Groups	18.1	k-1			
Noise	Within Groups	51.8	n-k			
	Total	69.9				

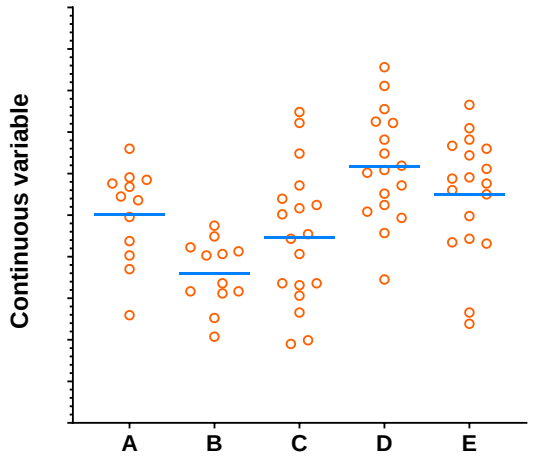
df: degree of freedom with df = n-1

n = number of values, k = number of groups

Between groups: df = 4 (k-1)

Within groups: df = 73 (n-k = n_1-1 + … + n_5-1)

Analysis of variance: how does it work?



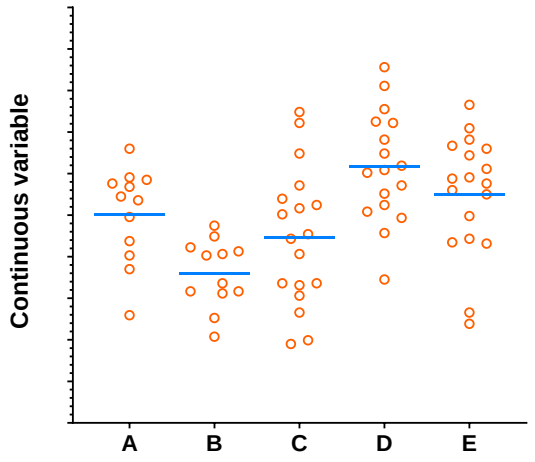
	Source of variation	Sum of Squares	df	Mean Squares	F ratio	p-value
Signal	Between Groups	18.1	4	4.5		
Noise	Within Groups	51.8	73	0.71		
	Total	69.9				

df: degree of freedom with df = n-1

18.2/4 = 4.5 **51.8/73 = 0.71**

Mean squares = **Sum of Squares / n-1 = Variance!**

Analysis of variance: how does it work?



Source of variation	Sum of Squares	df	Mean Squares	F ratio	p-value
Between Groups	18.1	4	4.5	6.34	0.0002
Within Groups	51.8	73	0.71		
Total	69.9				

Mean squares = **Sum of Squares / n-1 = Variance**

$$\text{F ratio} = \frac{\text{Variance between the groups}}{\text{Variance within the groups (individual variability)}} = \frac{4.5}{0.71} = 6.34$$

One-Way Analysis of variance

Step 1: Omnibus test

- It tells us if there is a difference between the means but not which means are significantly different from which other ones

Step 2: Post-hoc tests

- Tell us if there are differences between the group means pairwise
- A correction for multiple comparisons will be applied on the p-values
- *Should only be used when the ANOVA finds a significant effect*

Comparison of more than 2 means

- Running multiple tests on the same data increases the **familywise error rate**
  = error rate across tests on the same experimental data

- One of the basic rules ('laws') of probability:
  - The Multiplicative Rule: The probability of the joint occurrence of 2 or more independent events is the product of the individual probabilities

$$P(A,B) = P(A) \times P(B)$$

For example:

$$P(\text{2 heads}) = P(\text{head}) \times P(\text{head}) = 0.5 \times 0.5 = 0.25$$

Familywise error rate

- **Example**: All pairwise comparisons between 3 groups A, B and C:
  = A-B, A-C and B-C

- Probability of making the Type I Error: **5%**
  → probability of not making the Type I Error is 95% (= 1 – 0.05)

- Multiplicative Rule:
  - Overall probability of no Type I errors = 0.95*0.95*0.95 = 0.857

- Probability of making at least one Type I Error = 1 – 0.857 = 0.143 or **14.3%**
  - Probability has increased from **5% → 14.3%**

- For comparisons between 5 groups, the familywise error rate is **40%** (=1-$(0.95)^n$)

Familywise error rate

- Solution to increased familywise error rate = correction for multiple comparisons

  → post-hoc tests

- Many different approaches:
  - Different statisticians addressed different issues
    - e.g. unbalanced design, heterogeneity of variance, liberal vs conservative

- Two main ways to address the multiple testing problem:
  - **Familywise Error Rate** (**FWER**) and **False Discovery Rate** (**FDR**)

- In all cases:

  More tests → higher familywise error rate → more stringent correction

Multiple testing problem

- **Difference between FWER and FDR**:
  - FWER: a p-value of 0.05 implies that 5% of **all tests** will result in false positives
  - FDR: an adjusted p-value (or **q-value**) of 0.05 implies that 5% of **significant tests** will result in false positives
- **FWER**: **Bonferroni**: α_{adjust} = 0.05/n comparisons, e.g. 3 comparisons: 0.05/3=0.016
  - Problem: very conservative leading to **loss of power** (lots of false negative)
  - 10 comparisons: threshold for significance = 0.05/10 = 0.005
  - Pairwise comparisons across 20,000 genes = 0.05/20,000 = 2.5×10^{-6}
- **FDR**: **Benjamini-Hochberg**: controls the expected proportion of "discoveries" (significant tests) that are false (false positive)
  - Correction applied only on the significant tests
  - **More power** but increased Type I Errors

Repeated measures One-Way ANOVA

- **A new assumption:**
  - That the variances of the differences between all combinations of related conditions (or group levels) are equal – known as the **assumption of sphericity**
  - The **Mauchly's test of sphericity** is used to assess whether the assumption of sphericity is met
  - If the assumption of sphericity is not met, a correction is applied
    - Often the default as the assumption is seldom met
  - Most common correction: **Greenhouse-Geisser correction**

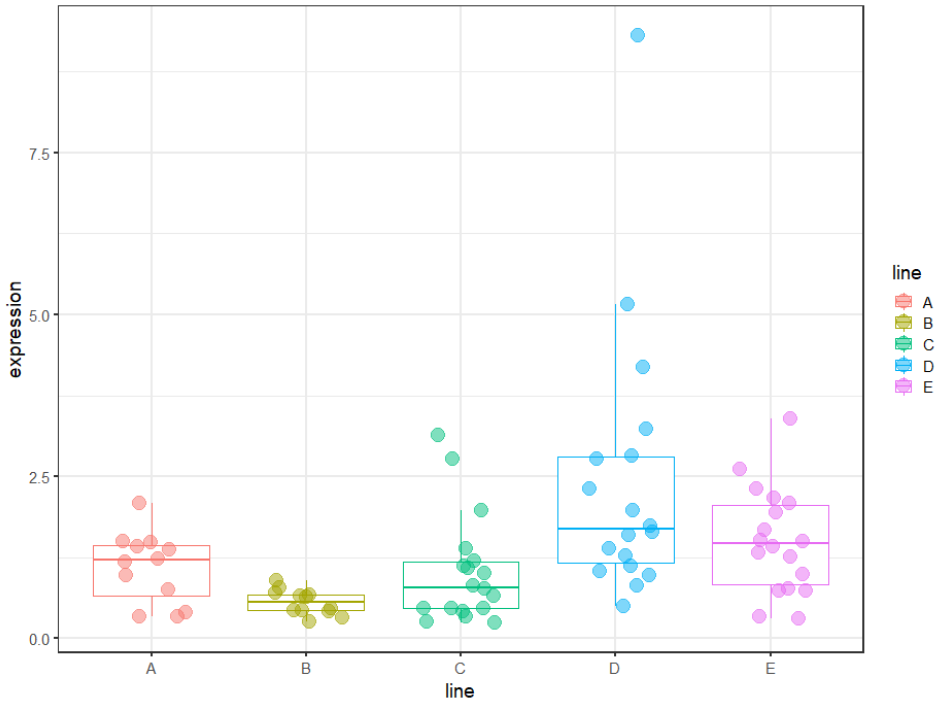
Exercise: One-way ANOVA: Data Exploration

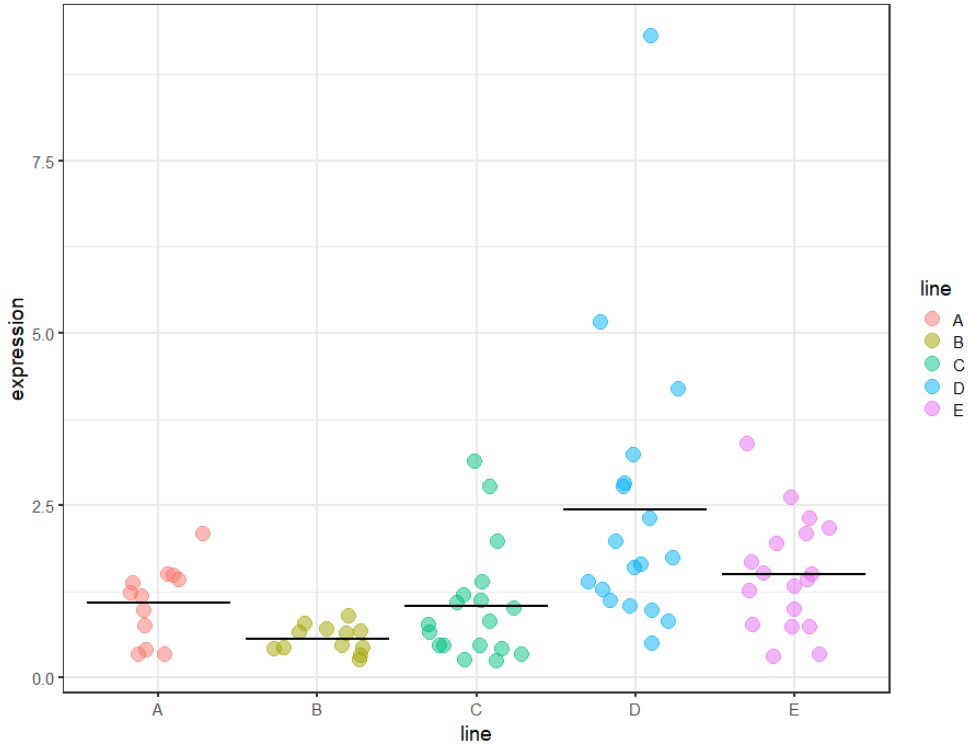
protein.expression.csv

- Question: is there a difference in protein expression between the 5 cell lines?
  - Load protein.expression.csv
  - Plot the data using at least 2 types of graph
    - `geom_boxplot(), geom_jitter(), geom_violin()`
  - Draw a QQplot
    - `ggqqplot()` #ggpubr package#
  - Check the first 2 assumptions with formal tests
    - `shapiro_test()` `levene_test()` # rstatix package #

Exercise: One-way ANOVA: Data Exploration

```
protein %>%
  ggplot(aes(x=line, y=expression, colour=line))+
  geom_boxplot(outlier.shape = NA)+
  geom_jitter(height=0, width=0.25, alpha=0.5, size=5)
```



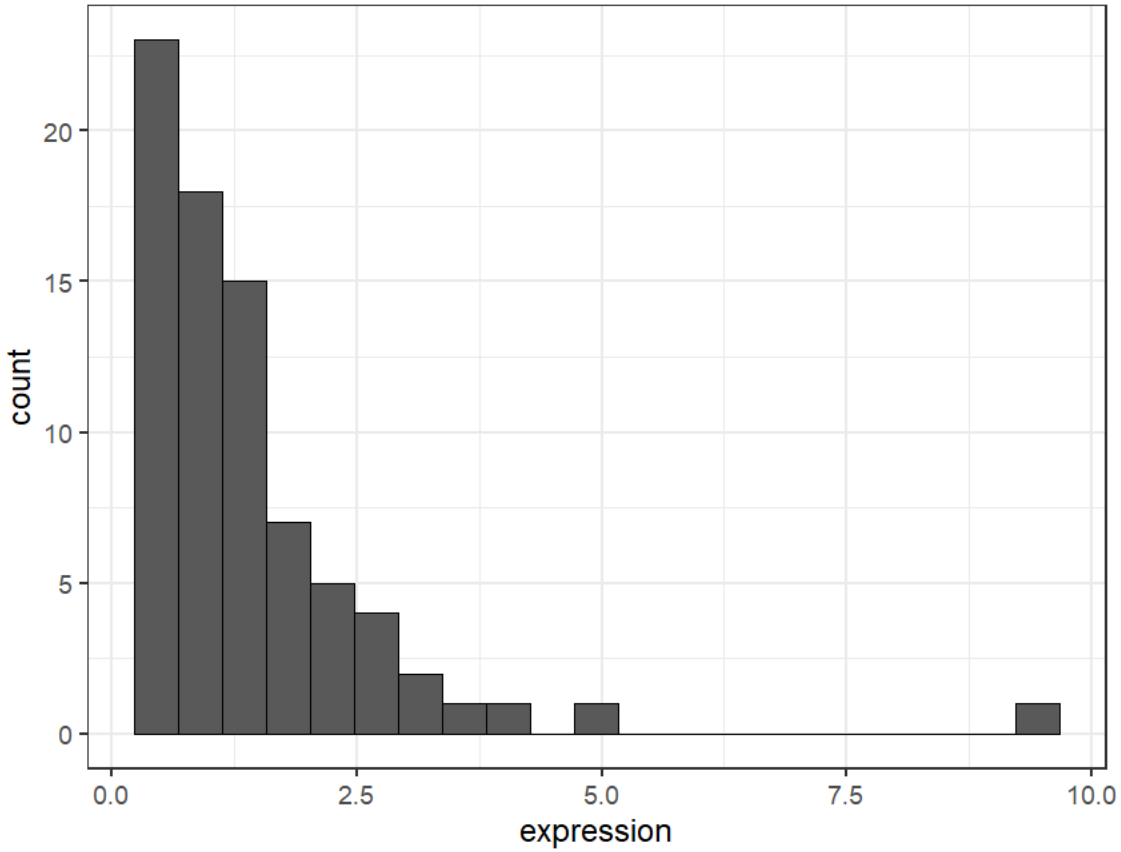


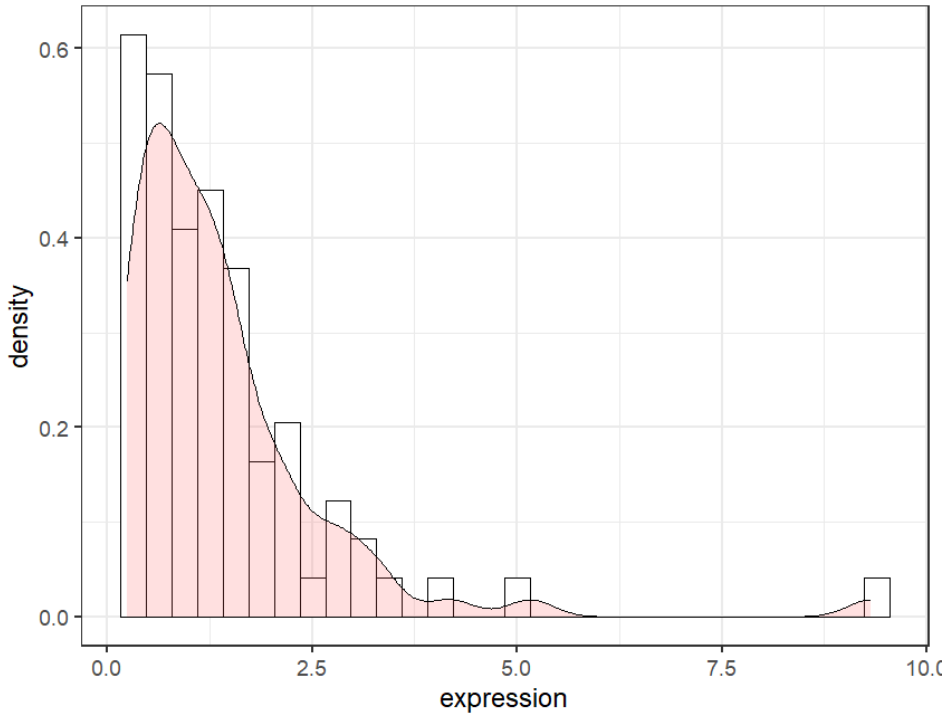

```
protein %>%
  ggplot(aes(x=line, y=expression, colour=line))+
  geom_jitter(height=0, width=0.3, alpha=0.5, size=5)+
 stat_summary(geom="crossbar", fun=mean, colour="black", linewidth=0.5)
```

Exercise: One-way ANOVA: Data Exploration

Histograms & density plots

```
protein %>%
  ggplot(aes(x=expression))+
  geom_histogram(binwidth = 0.45,
       colour="black")
```






```
protein %>%
  ggplot(aes(x=expression))+
  geom_histogram(aes(y=after_stat(density)),
       colour="black", fill="white")+
  geom_density(alpha=0.2, fill="#FF6666")
```

Exercise: One-way ANOVA: Data Exploration

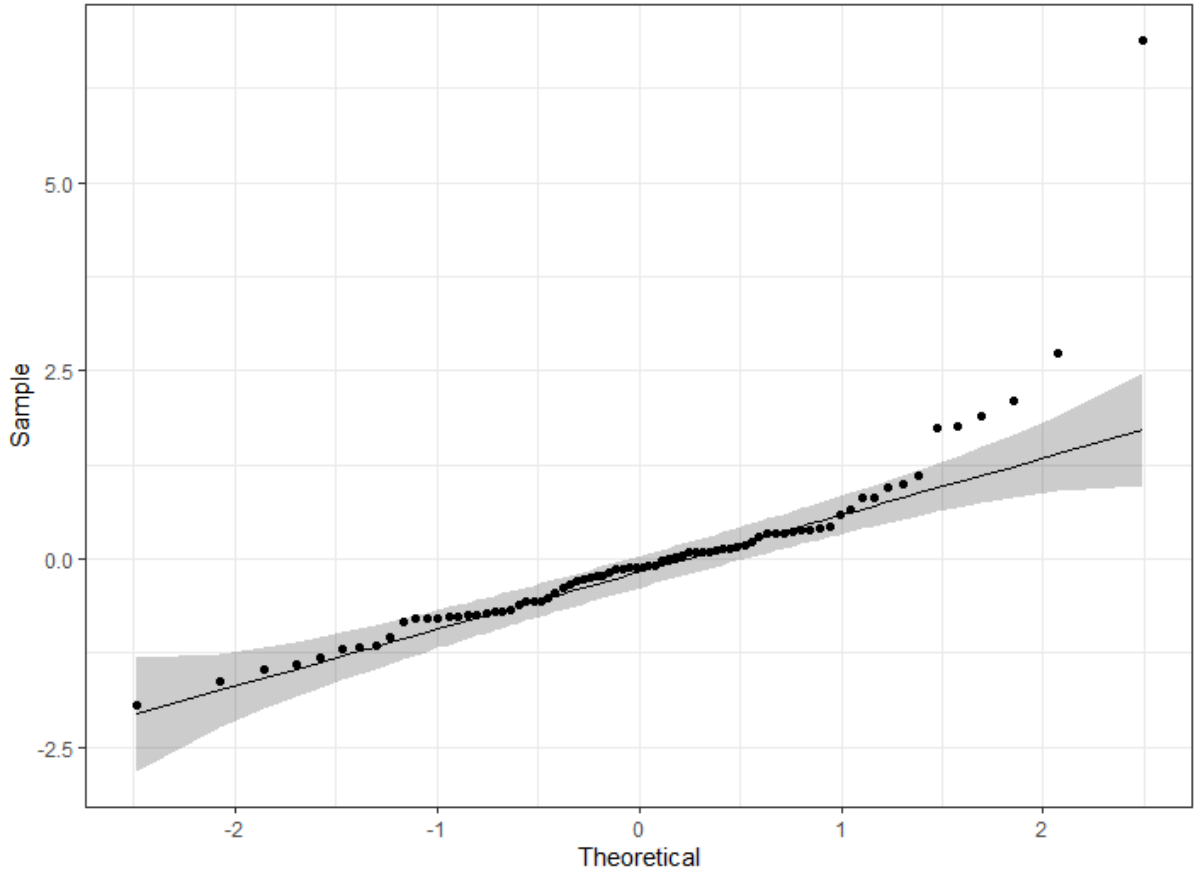
QQ plot

Build an anova model so can extract residuals

```
model <- aov(expression ~ line, data = protein)
```

Then draw the QQ plot

```
ggqqplot(residuals(model)) + theme_bw()
```



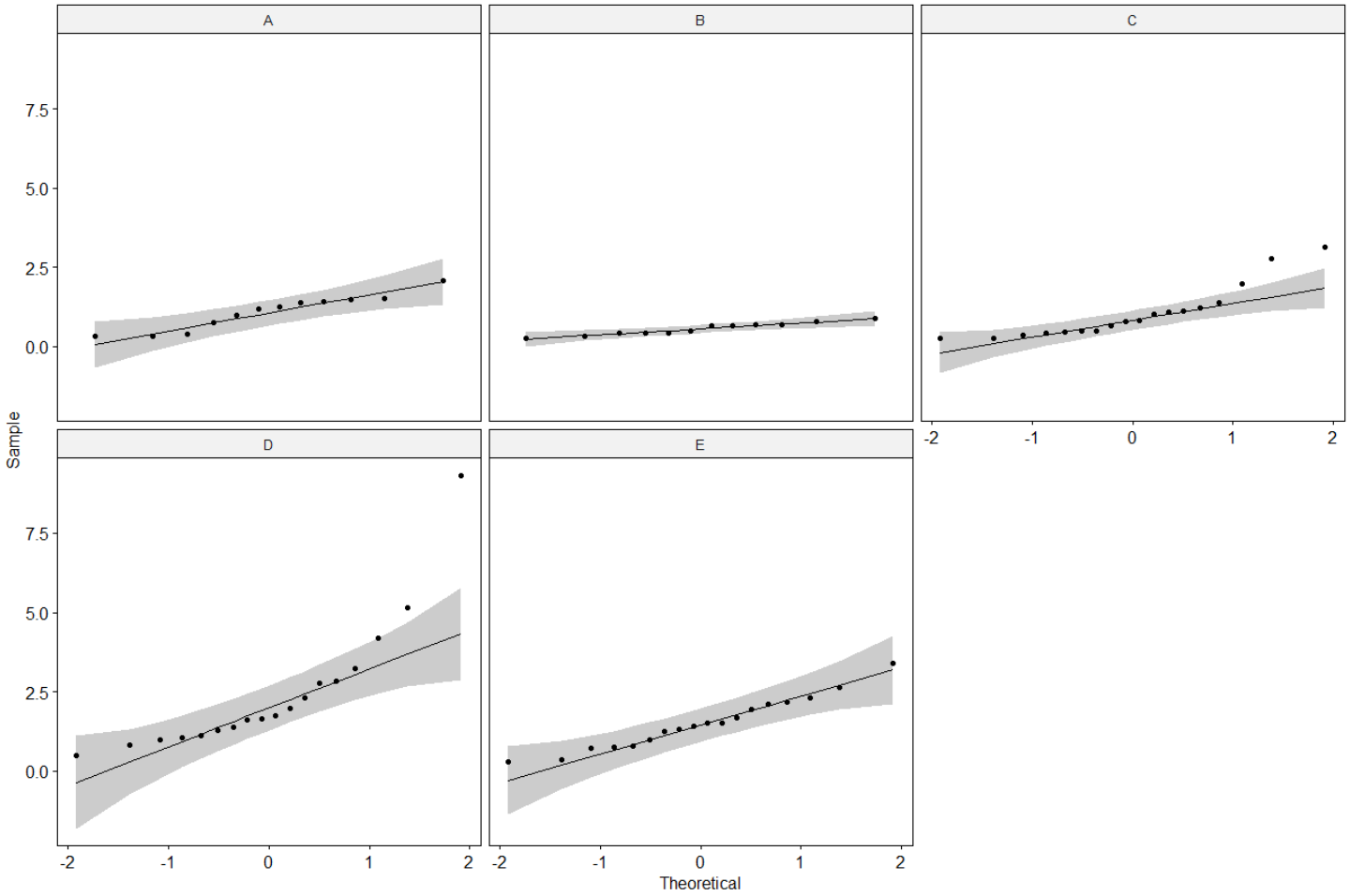

```
#ggpubr package#
```

Exercise: One-way ANOVA: Data Exploration

QQ plot

Or can look at groups individually

```
ggqqplot(protein, x = "expression", facet.by = "line")
```

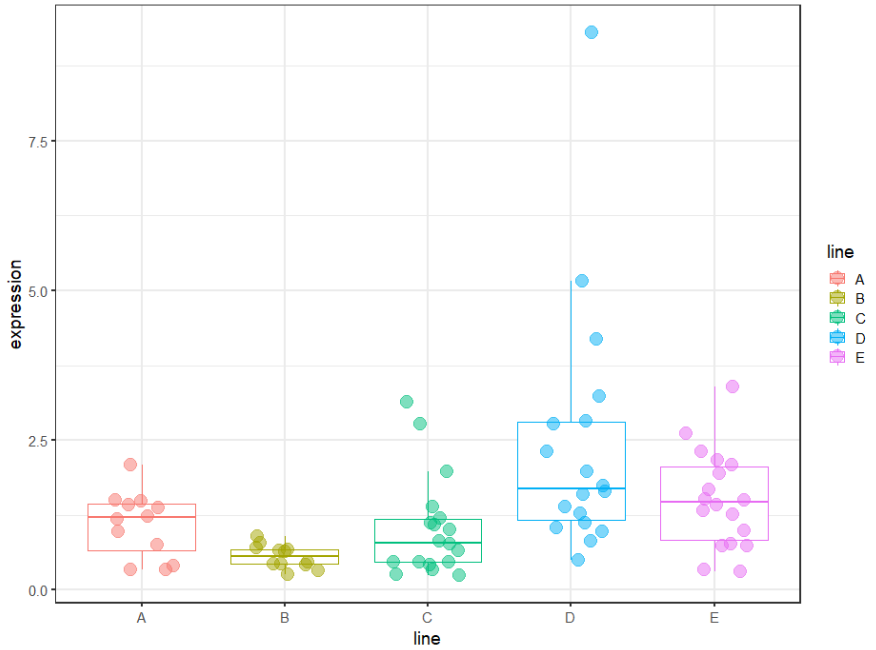


Exercise: One-way ANOVA: Data Exploration

```
protein %>%
  group_by(line) %>%
    identify_outliers(expression)
```

line <chr>	expression <dbl>	log10.expression <dbl>	is.outlier <lgl>	is.extreme <lgl>
C	3.14	0.4969296	TRUE	FALSE
C	2.78	0.4440448	TRUE	FALSE
D	9.32	0.9694159	TRUE	TRUE

3 rows



Exercise: One-way ANOVA: Data Exploration

```
model <- aov(expression ~ line,
             data = protein)
protein %>%
  shapiro_test(residuals(model))
```

variable	statistic	p.value
<chr>	<dbl>	<dbl>
residuals(model)	0.772	0.000000001 20

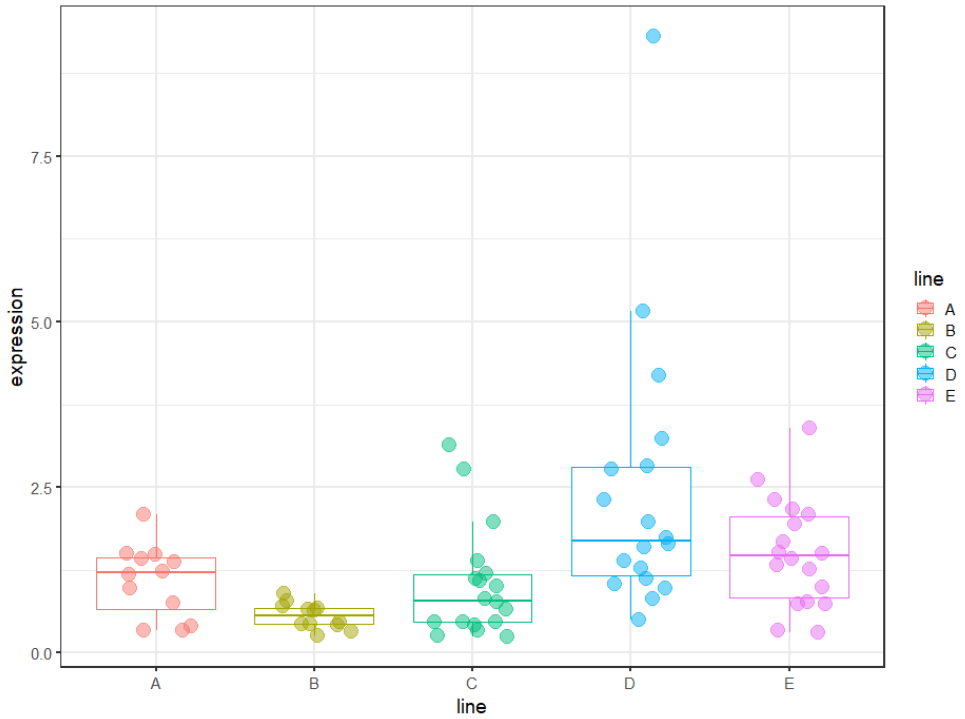
```
protein %>%
  group_by(line) %>%
    shapiro_test(expression)
```

line <chr>	variable <chr>	statistic <dbl>	p <dbl>
A	expression	0.9295671	0.3755460156
B	expression	0.9535144	0.6887867228
C	expression	0.8196840	0.0029210891
D	expression	0.7530720	0.0003548725
E	expression	0.9670693	0.7411280600

What do we do now?

```
protein %>%
  levene_test(expression ~ line)
```

	df1 <int>	df2 <int>	statistic <dbl>	p <dbl>
1	4	73	2.88	0.028 2

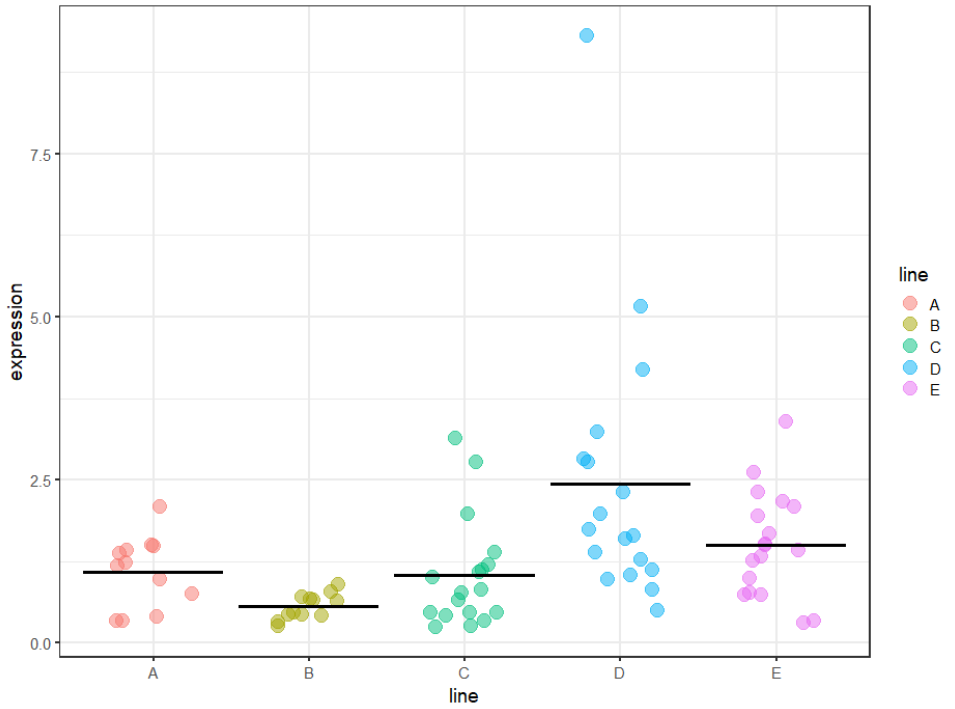


One-way ANOVA

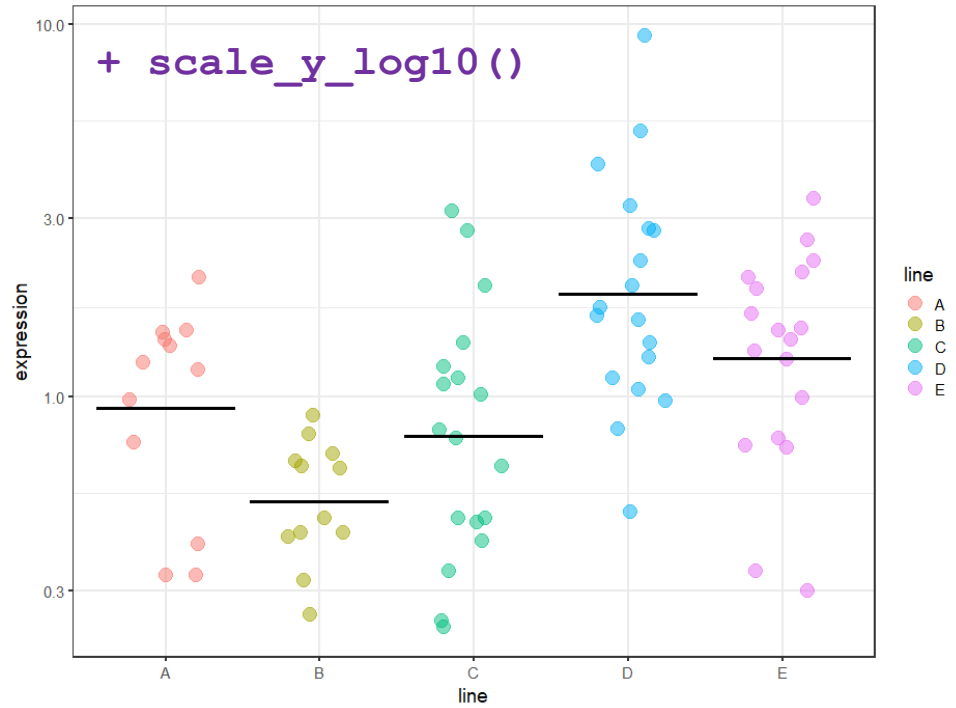
Change of scale

```
protein %>%
   ggplot(aes(x=line, y=expression, colour=line))+
     geom_jitter(height=0, width=0.2, size=3, show.legend=FALSE)+
     stat_summary(geom="crossbar", fun=mean, colour="black", linewidth=0.5) +
     scale_y_log10()
```

Linear scale



Log scale

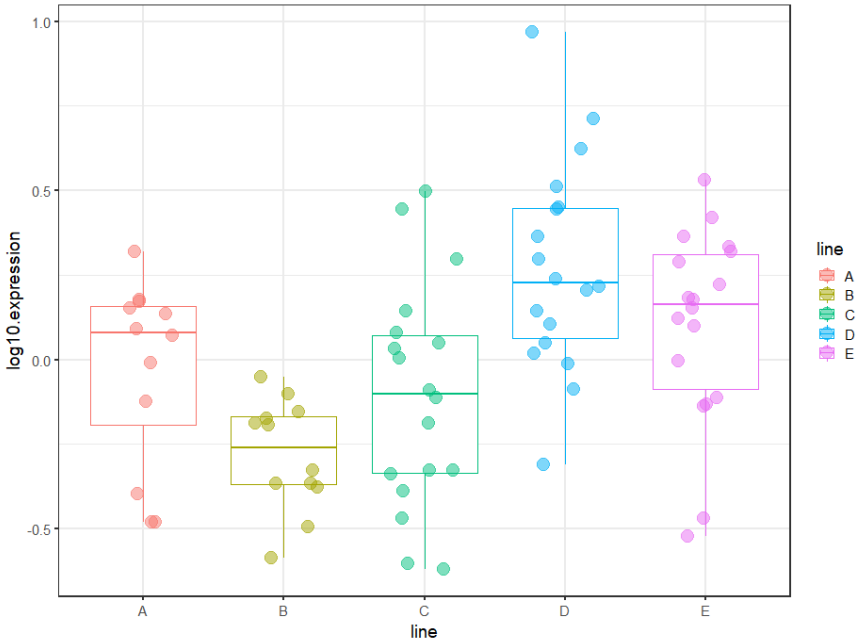


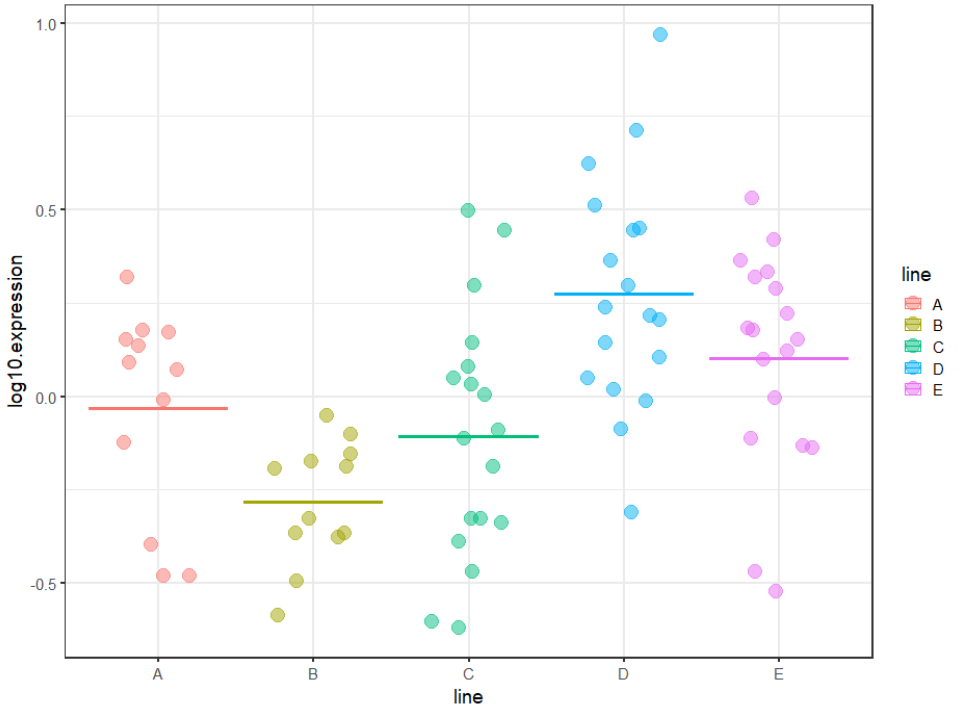

```
protein %>%
  mutate(log10.expression=log10(expression)) -> protein
```

One-way ANOVA

Log-transformed values

```
protein %>%
  ggplot(aes(x=line, y=log10.expression, colour=line))+
  geom_boxplot()+
  geom_jitter(height=0, width=0.25, alpha=0.5, size=5)
```



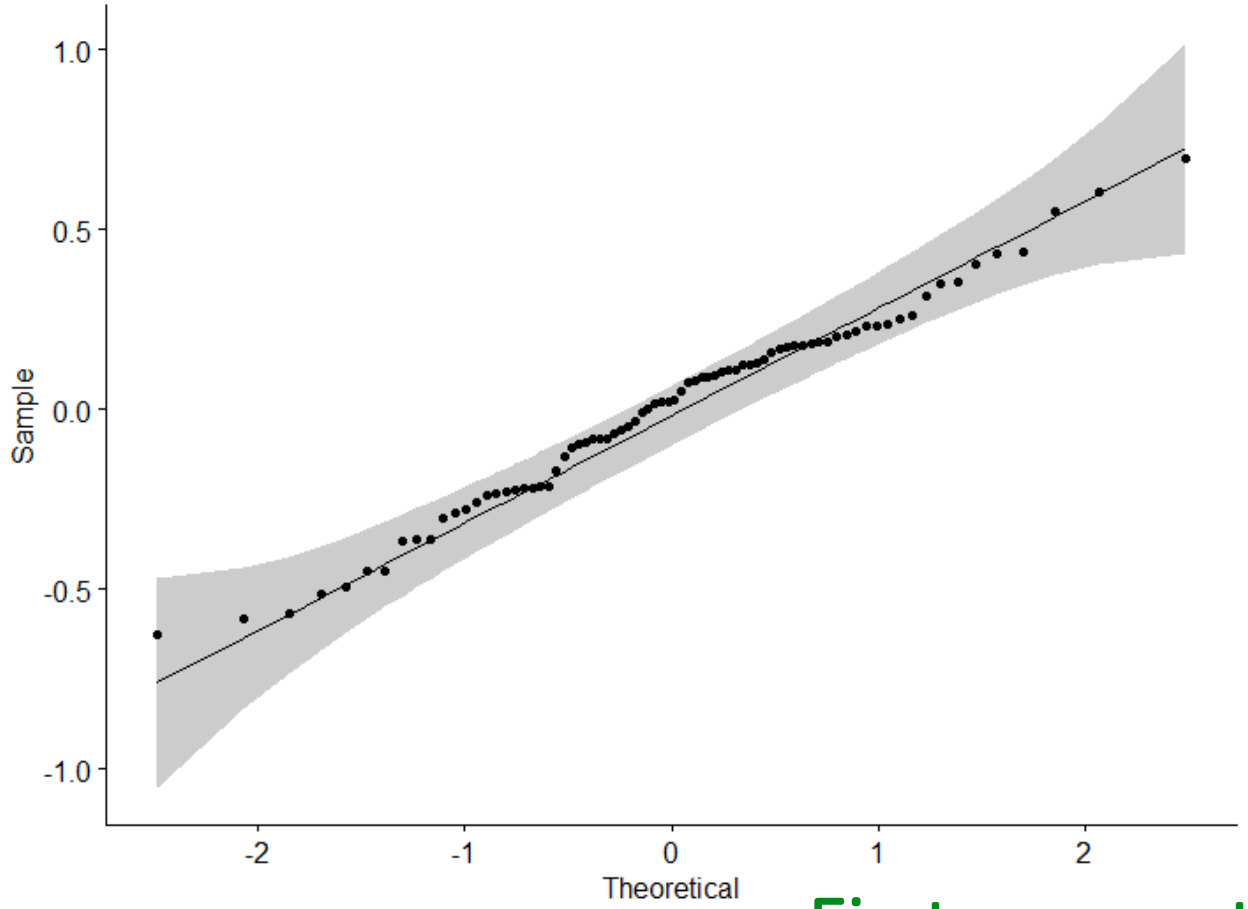



```
protein %>%
  ggplot(aes(x=line, y=log10.expression, colour=line))+
  geom_jitter(height=0, width=0.25, alpha=0.5, size=5)+
  stat_summary(geom="crossbar", fun=mean, linewidth=0.5)
```

One-way ANOVA

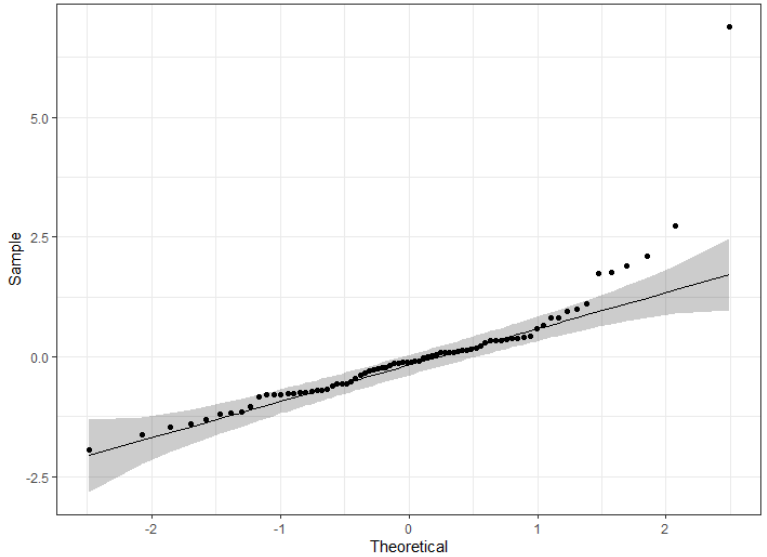
Log-transformed values

```
model  <- aov(log10.expression ~ line, data = protein)
ggqqplot(residuals(model))
```



First assumption ✓

Before log-transformation



Assumptions of Parametric Data

Formal tests

```
protein %>%
  group_by(line) %>%
    shapiro_test(log10.expression)
```

line <chr>	variable <chr>	statistic <dbl>	p <dbl>
A	log10.expression	0.8542464	0.04143953
B	log10.expression	0.9458450	0.57725321
C	log10.expression	0.9657060	0.71417958
D	log10.expression	0.9868425	0.99348831
E	log10.expression	0.9313425	0.20502703

First assumption ✓ish

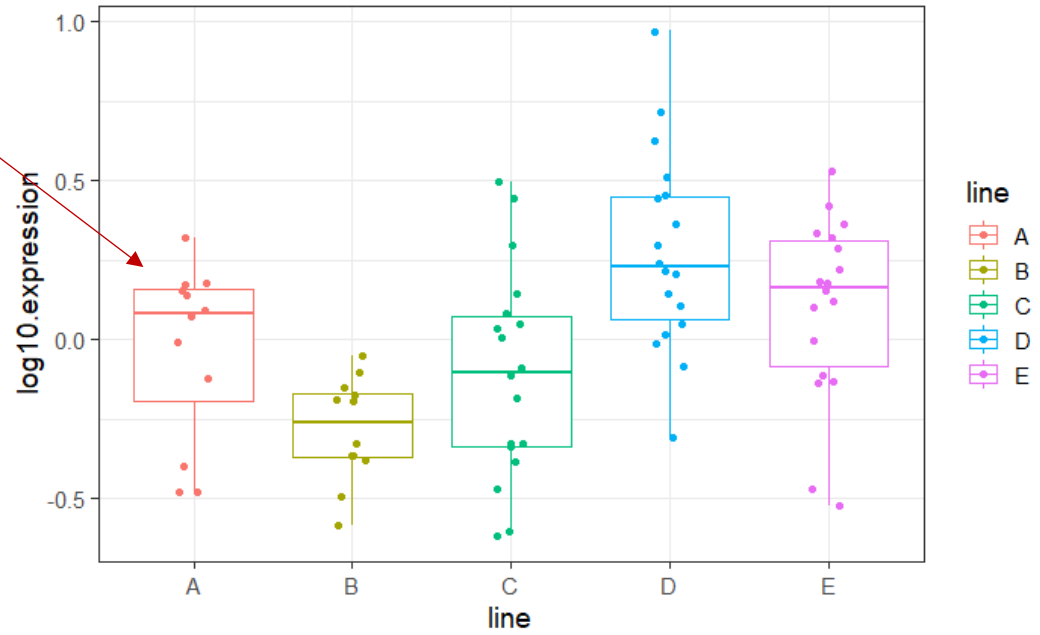
```
protein %>%
  levene_test(log10.expression ~ line)
```

df1 <int>	df2 <int>	statistic <dbl>	p <dbl>
4	73	0.982112	0.4227373

Second assumption ✓

```
model <- aov(log10.expression ~ line,
  data = protein)
protein %>%
  shapiro_test(residuals(model))
```

```
# A tibble: 1 x 3
  variable          statistic p.value
  <chr>                 <dbl>   <dbl>
1 residuals(model)      0.986   0.566
```



Analysis of variance

Let's do it

- Task 1: **omnibus test**

```
data %>%
  anova_test(y~x)
```

- Task 2: **post-hoc tests**

Tukey correction

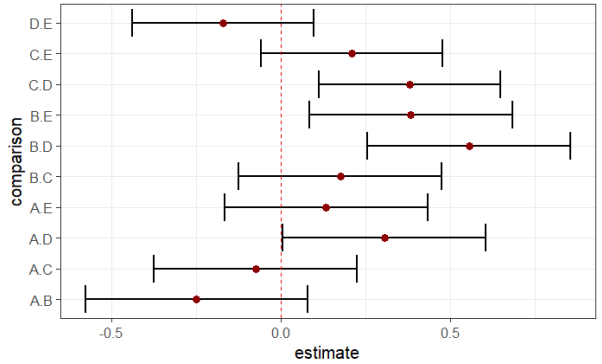
```
data %>%
  tukey_hsd(y~x)
```

Bonferroni correction # emmeans package #

Default

```
data %>%
 emmeans_test(y~x, p.adjust.method="bonferroni")
```

- **Extra task**: Plot confidence intervals



Analysis of variance

```
protein %>%
  anova_test(log10.expression~line)
```

```
ANOVA Table (type II tests)

  Effect DFn DFd     F        p p<.05   ges
1   line   4  73 8.123 1.78e-05     * 0.308
```

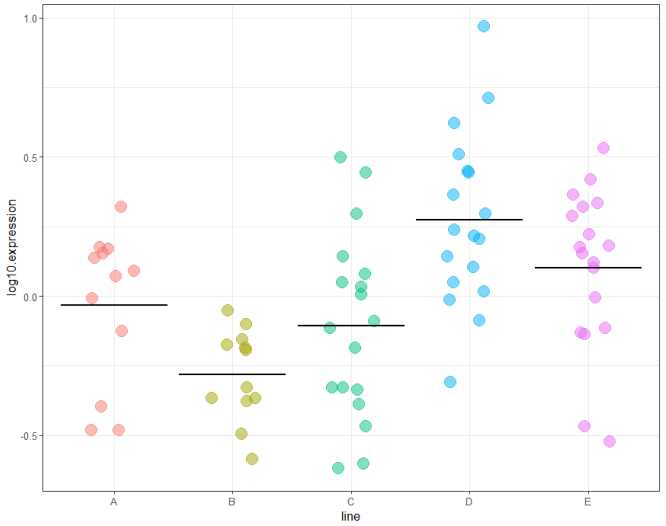
Not the p-value!

generalised **e**ffect **s**ize (Eta squared η^2) = R^2 ish

```
protein %>%
  tukey_hsd(log10.expression~line)
```

Tukey correction

	term <chr>	group1 <chr>	group2 <chr>	estimate <dbl>	conf.low <dbl>	conf.high <dbl>	p.adj <dbl>	p.adj.signif <chr>
1	line	A	B	-0.25024832	-0.578882494	0.07838585	2.19e-01	ns
2	line	A	C	-0.07499724	-0.374997820	0.22500335	9.56e-01	ns
3	line	A	D	0.30549397	0.005493391	0.60549456	4.39e-02	*
4	line	A	E	0.13327517	-0.166725416	0.43327575	7.27e-01	ns
5	line	B	C	0.17525108	-0.124749499	0.47525167	4.81e-01	ns
6	line	B	D	0.55574230	0.255741712	0.85574288	1.83e-05	****
7	line	B	E	0.38352349	0.083522904	0.68352407	5.48e-03	**
8	line	C	D	0.38049121	0.112162532	0.64881989	1.54e-03	**
9	line	C	E	0.20827240	-0.060056276	0.47660108	2.02e-01	ns
10	line	D	E	-0.17221881	-0.440547487	0.09610987	3.84e-01	ns



Analysis of variance

```
protein %>%
  anova_test(log10.expression~line)
```

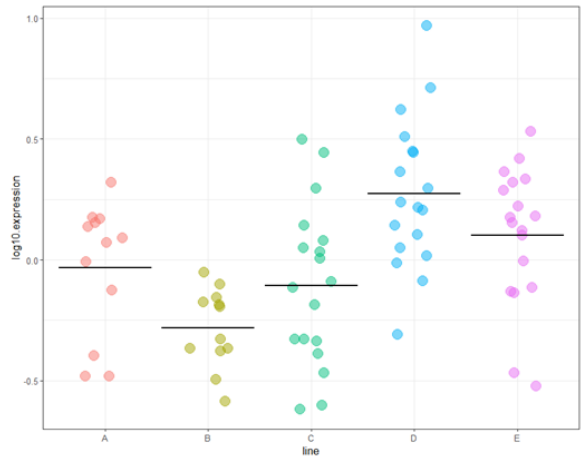
```
ANOVA Table (type II tests)

  Effect DFn DFd     F        p p<.05   ges
1   line   4  73 8.123 1.78e-05     * 0.308
```

generalised **e**ffect **s**ize (Eta squared η^2) = R^2 ish

```
protein %>%
  emmeans_test(log10.expression ~ line, p.adjust.method = "bonferroni") # emmeans package #
```

Bonferroni correction

	.y. <chr>	group1 <chr>	group2 <chr>	df <dbl>	statistic <dbl>	p <dbl>	p.adj <dbl>	p.adj.signif <chr>
1	log10.expression	A	B	73	2.1299578	3.654611e-02	3.654611e-01	ns
2	log10.expression	A	C	73	0.6992552	4.866147e-01	1.000000e+00	ns
3	log10.expression	A	D	73	-2.8483483	5.705474e-03	5.705474e-02	ns
4	log10.expression	A	E	73	-1.2426238	2.179833e-01	1.000000e+00	ns
5	log10.expression	B	C	73	-1.6339966	1.065653e-01	1.000000e+00	ns
6	log10.expression	B	D	73	-5.1816001	1.882302e-06	1.882302e-05	****
7	log10.expression	B	E	73	-3.5758757	6.238766e-04	6.238766e-03	**
8	log10.expression	C	D	73	-3.9663413	1.687079e-04	1.687079e-03	**
9	log10.expression	C	E	73	-2.1710868	3.317601e-02	3.317601e-01	ns
10	log10.expression	D	E	73	1.7952545	7.675206e-02	7.675206e-01	ns

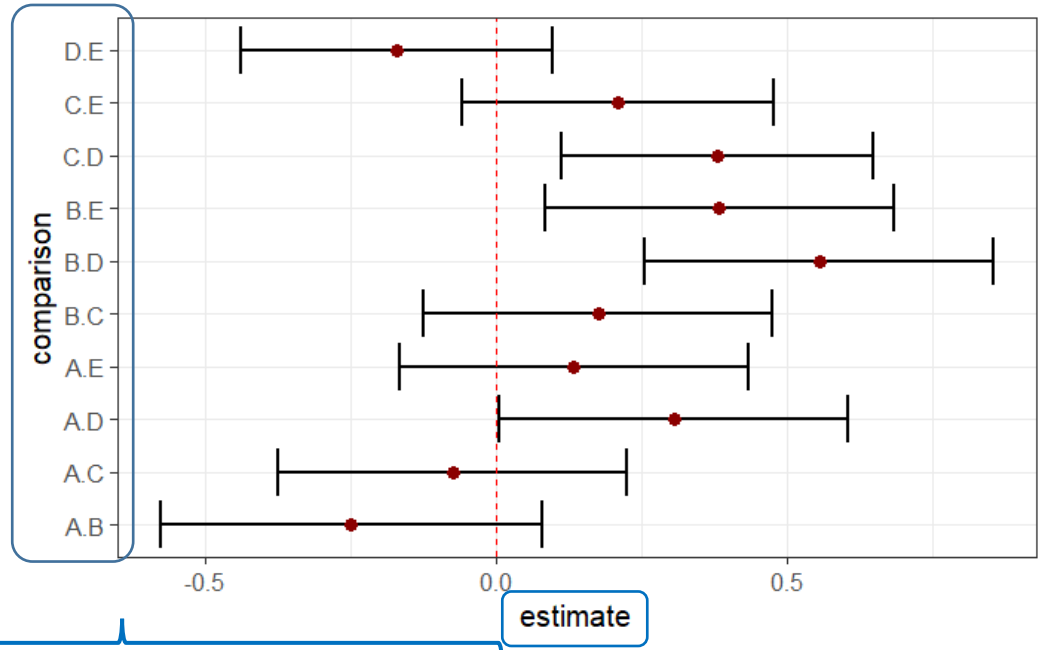


Analysis of variance

Plot confidence intervals (forest plots)

```
protein %>%
  tukey_hsd(log10.expression~line)%>%
  mutate(comparison = paste(group1, sep=".", group2)) -> tukey.conf
```

term	group1	group2	null.value	estimate	conf.low	conf.high	p.adj	p.adj.signif	comparison
line	A	B	0	-0.25024832	-0.578882494	0.07838585	2.19e-01	ns	A.B
line	A	C	0	-0.07499724	-0.374997820	0.22500335	9.56e-01	ns	A.C
line	A	D	0	0.30549397	0.005493391	0.60549456	4.39e-02	*	A.D
line	A	E	0	0.13327517	-0.166725416	0.43327575	7.27e-01	ns	A.E
line	B	C	0	0.17525108	-0.124749499	0.47525167	4.81e-01	ns	B.C
line	B	D	0	0.55574230	0.255741712	0.85574288	1.83e-05	****	B.D
line	B	E	0	0.38352349	0.083522904	0.68352407	5.48e-03	**	B.E
line	C	D	0	0.38049121	0.112162532	0.64881989	1.54e-03	**	C.D
line	C	E	0	0.20827240	-0.060056276	0.47660108	2.02e-01	ns	C.E
line	D	E	0	-0.17221881	-0.440547487	0.09610987	3.84e-01	ns	D.E

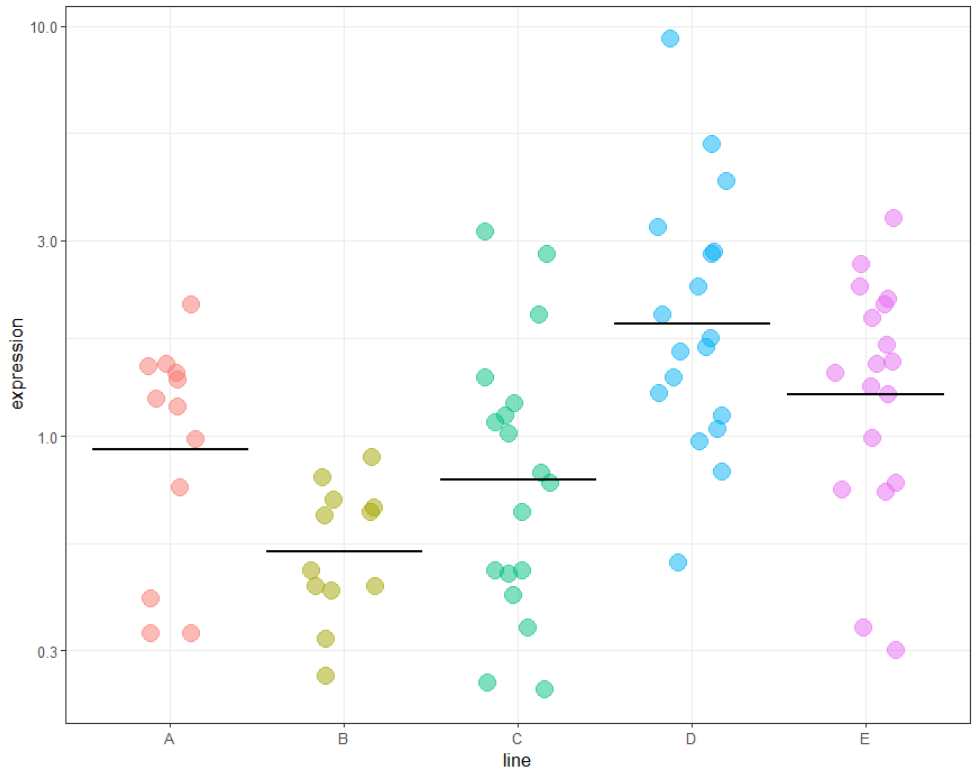



```
tukey.conf %>%
  ggplot(aes(x=comparison, y=estimate, ymin=conf.low, ymax=conf.high)) +
   geom_errorbar(colour="black", linewidth=1)+
   geom_point(size=3, colour="darkred")+
   coord_flip()+
   geom_hline(yintercept=0, linetype="dashed", color = "red")+
```

Analysis of variance

Stripchart

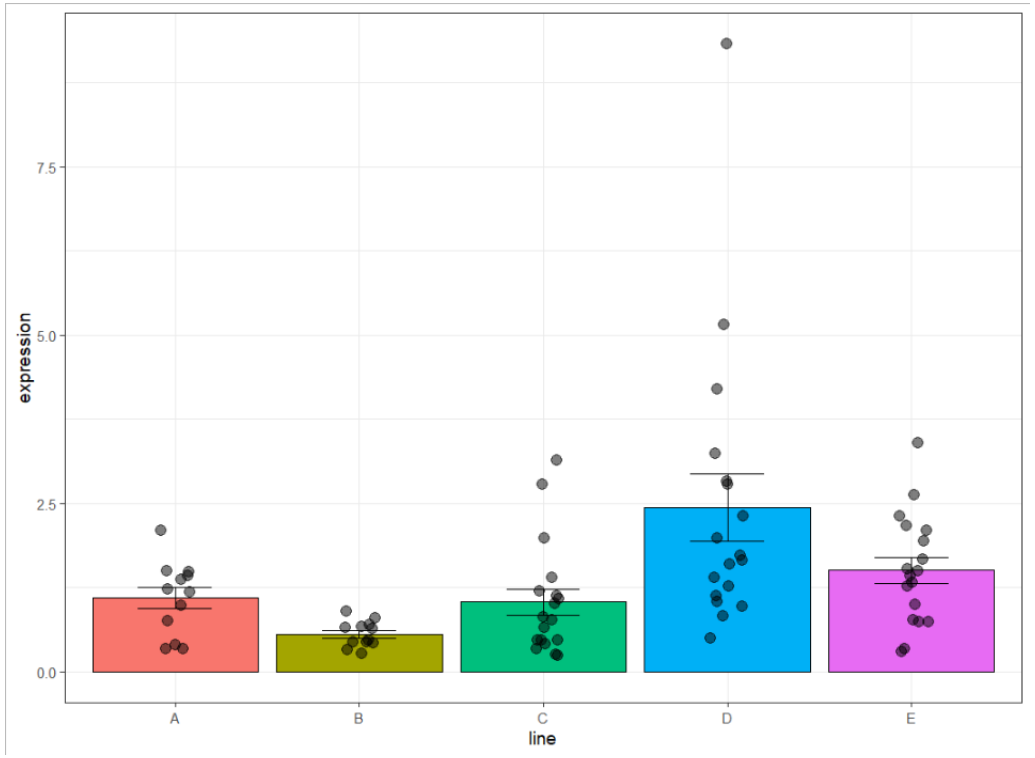
```
protein %>%
  ggplot(aes(x=line, y=expression, colour=line))+
    geom_jitter(height = 0, width=0.2, size=6, show.legend=FALSE, alpha=0.5)+
    stat_summary(geom="errorbar",fun=mean,fun.min=mean,fun.max = mean, colour="black",
    linewidth=1)+
    scale_y_log10()
```



Analysis of variance

Overlay: stripchart and barchart

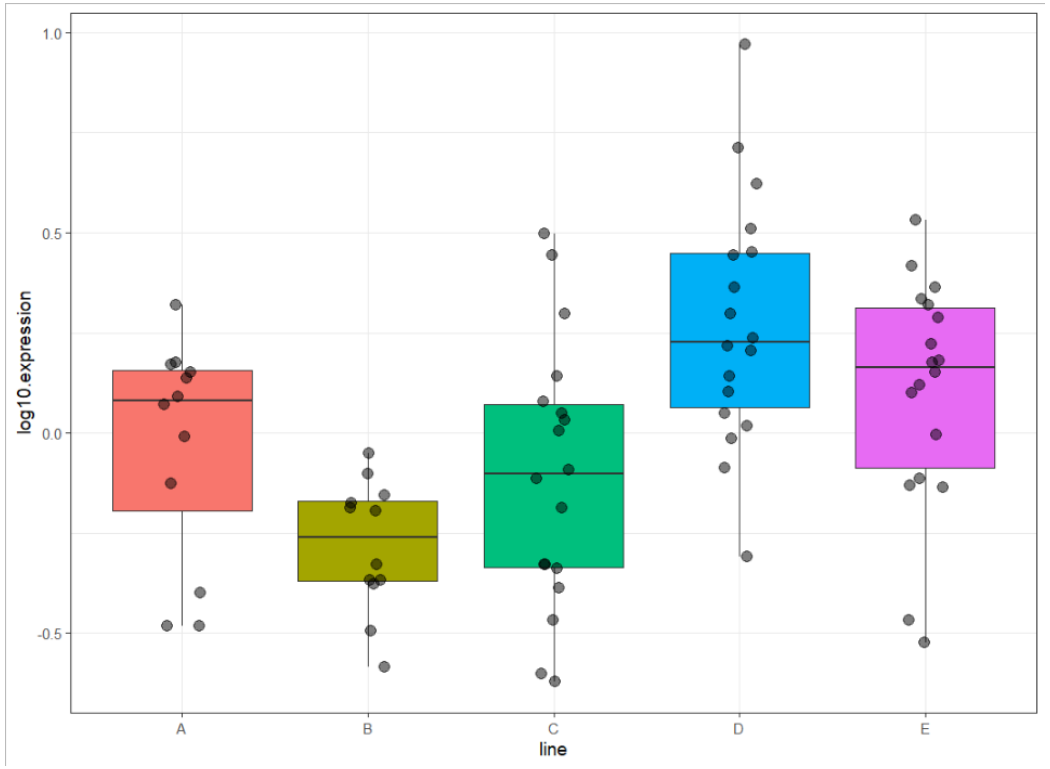
```
protein %>%
  ggplot(aes(x=line, y=expression, fill=line)) +
       geom_bar(stat="summary", fun="mean", colour="black", show.legend=FALSE)+
       stat_summary(geom="errorbar", colour="black", width=0.4)+
       geom_jitter(height=0, width=0.1, alpha=0.5, size=4,  show.legend=FALSE)
```



Analysis of variance

Overlay: boxplot and stripchart (log10 data)

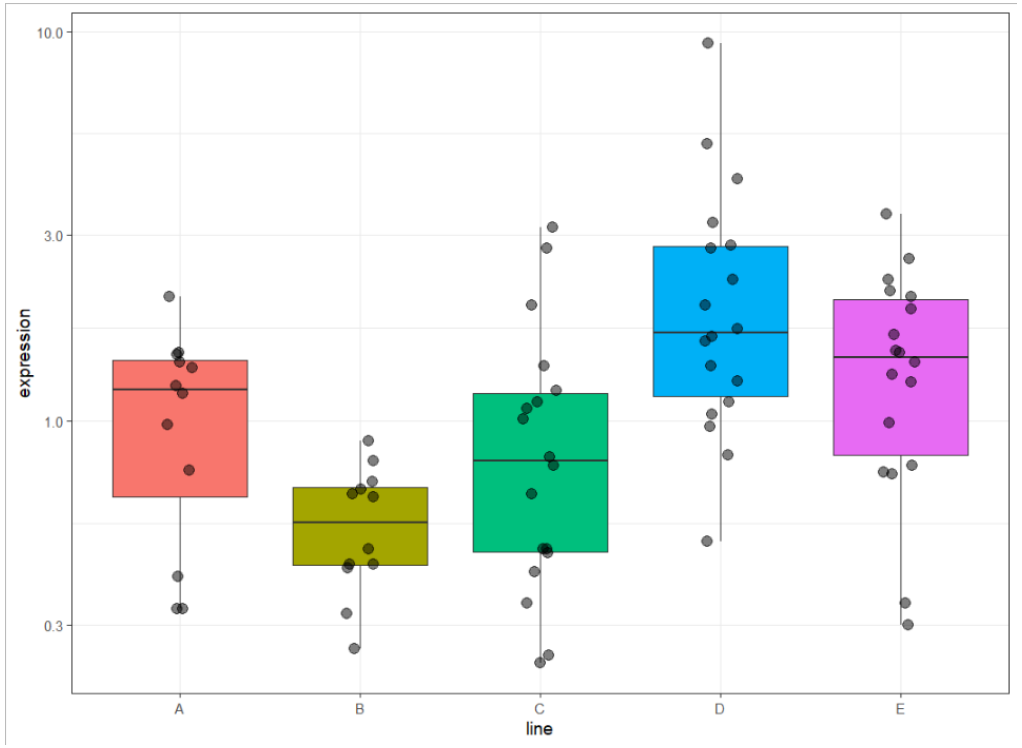
```
protein %>%
  ggplot(aes(x=line, y=log10.expression, fill=line)) +
       geom_boxplot(show.legend=FALSE)+
       geom_jitter(height=0, width=0.1, alpha=0.5, size=4, show.legend=FALSE)
```



Analysis of variance

Overlay: boxplot and stripchart (log scale)

```
protein %>%
  ggplot(aes(x=line, y=expression, fill=line)) +
      geom_boxplot(show.legend=FALSE)+
      geom_jitter(height=0, width=0.1, alpha=0.5, size=4, show.legend=FALSE)+
      scale_y_log10()
```

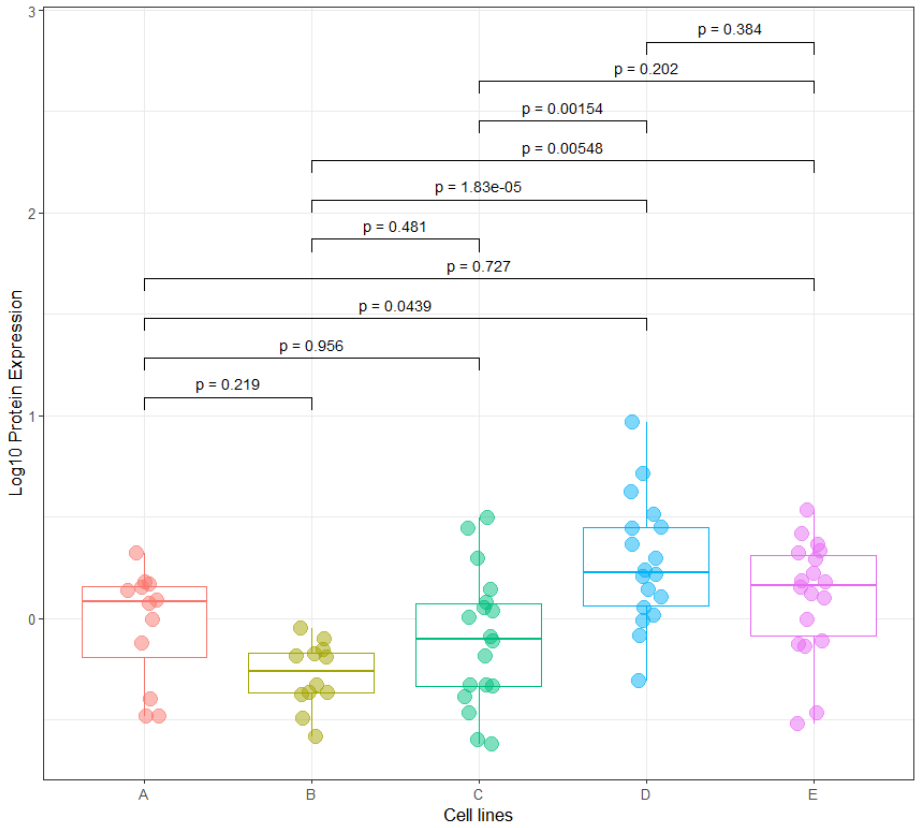


Analysis of variance

Graphical presentation with p-values

Approach 1: ggpubr

```
proteins.tukey <- protein %>%
  tukey_hsd(log10.expression~line) %>%
  add_xy_position()

protein %>%
  ggplot(aes(x=line, y=log10.expression, colour=line)) +
  geom_boxplot(show.legend = FALSE)+
  geom_jitter(height=0, width=0.1, alpha=0.5,
        size=5, show.legend = FALSE)+
  stat_pvalue_manual(proteins.tukey,label="p = {p.adj}",
        label.size=4,tip.length=0.02,step.increase=0.02)+
  xlab("Cell lines")+
  ylab("Log10 Protein Expression")
```



Analysis of variance

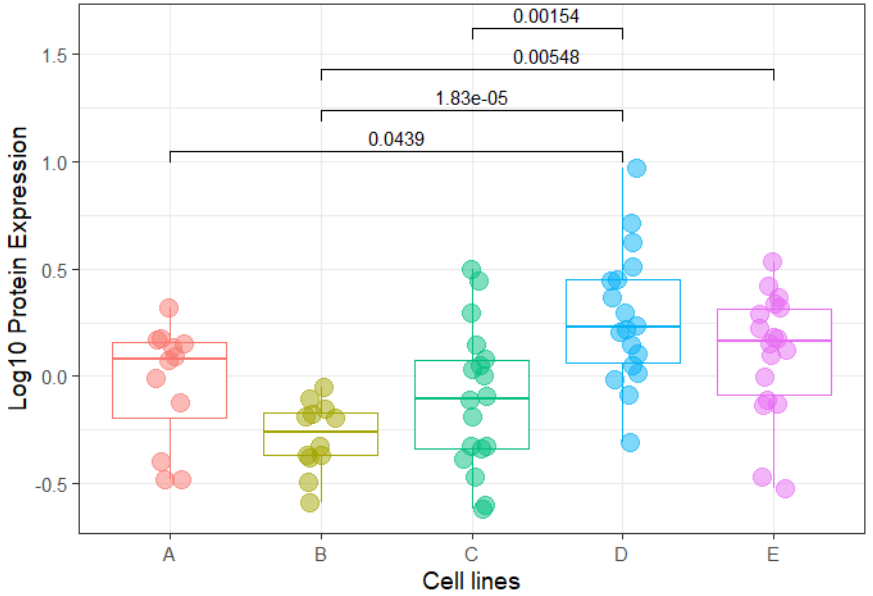
Graphical presentation with p-values

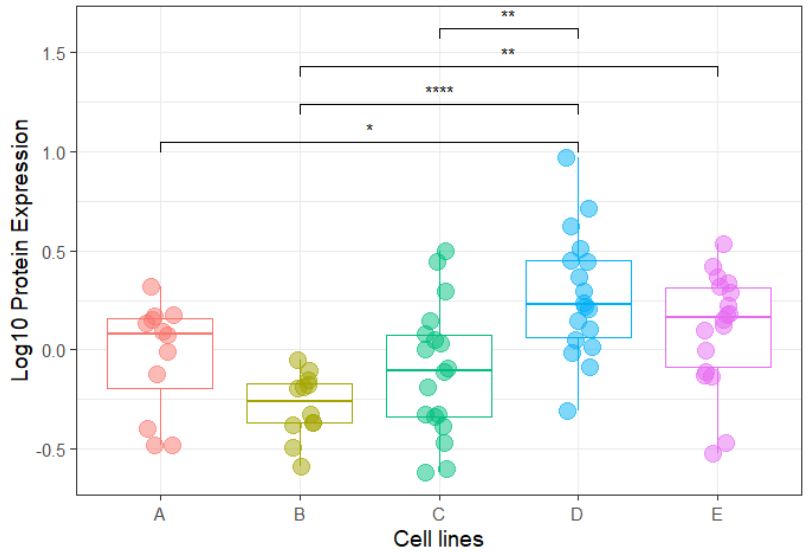
Approach 2: also ggpubr

```
protein %>%
  ggplot(aes(x=line, y=log10.expression, colour=line))+
  geom_boxplot(show.legend = FALSE)+
  geom_jitter(height=0, width=0.1, alpha=0.5,
        size=5, show.legend = FALSE)+
  stat_pwc(method = "tukey_hsd", label = "p.adj",
        hide.ns = TRUE, show.legend = FALSE)+
  xlab("Cell lines")+
  ylab("Log10 Protein Expression")
```

OR

```
protein %>%
  ggplot(aes(x=line, y=log10.expression, colour=line))+
  geom_boxplot(show.legend = FALSE)+
  geom_jitter(height=0, width=0.1, alpha=0.5,
        size=5, show.legend = FALSE)+
  stat_pwc(method = "tukey_hsd", label = "p.adj.signif",
        hide.ns = TRUE, show.legend = FALSE)+
  xlab("Cell lines")+
  ylab("Log10 Protein Expression")
```



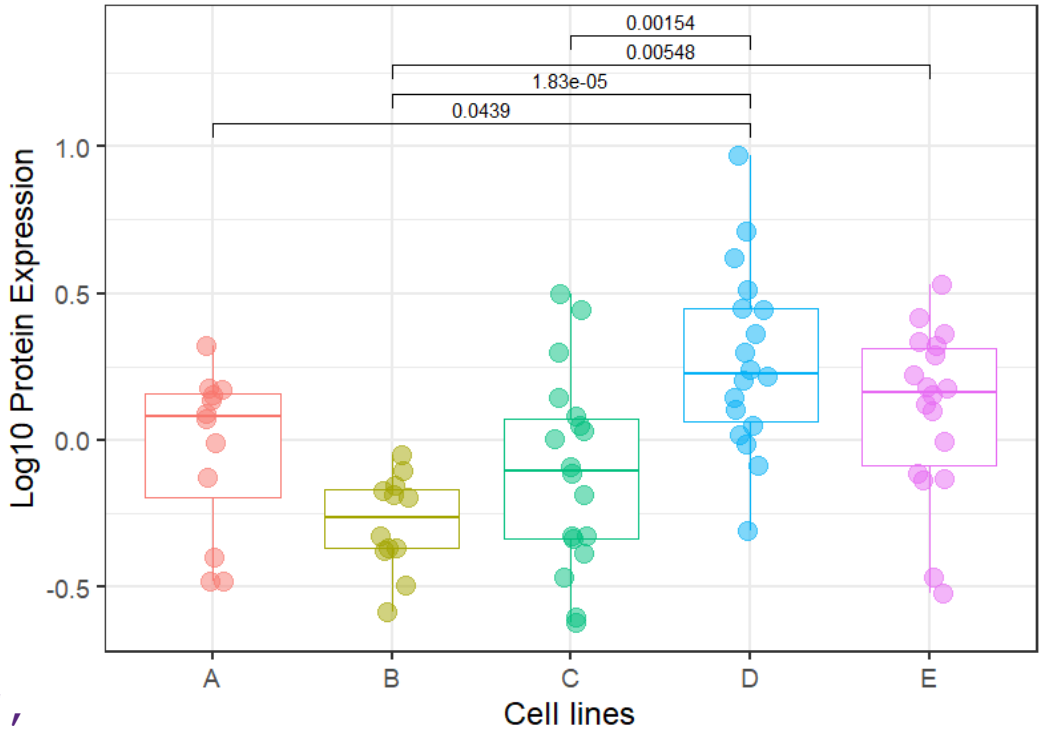


Analysis of variance

Graphical presentation with p-values

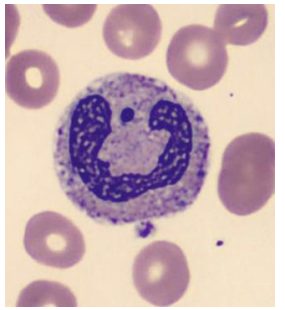
Approach 3: ggsignif

```
sig.comp <- proteins.tukey %>%
  filter(p.adj<0.05)

protein %>%
  ggplot(aes(x=line, y=log10.expression, colour=line))+
  geom_boxplot(show.legend = FALSE)+
  geom_jitter(height=0, width=0.1, alpha=0.5,
        size=5, show.legend = FALSE)+
  geom_signif(comparisons = list(c("A","D"), c("B","D"),
                c("B","E"), c("C","D")),
        annotations = sig.comp$p.adj,
        y_position = c(1, 1.1, 1.2, 1.3), colour = "black",

        show.legend = FALSE)+
  xlab("Cell lines")+
  ylab("Log10 Protein Expression")
```



Analysis of variance

Matched/repeated measures



- For repeated measures ANOVA and post-hoc tests need to specify matching:

Experiment identifier

```
anova_test(dv =, wid =, within =) -> res.aov
get_anova_table(res.aov)
pairwise_t_test(p.adjust.method =)
```

To choose the Reference group and account for the matched design

```
neutrophils.long %>%
  anova_test(dv = Values, wid = Experiment,
        within = Condition) -> res.aov
get_anova_table(res.aov)
# post-hoc test
neutrophils.long %>%
  pairwise_t_test(Values~Condition, paired=TRUE,
        ref.group = "WT", p.adjust.method = "holm")
```

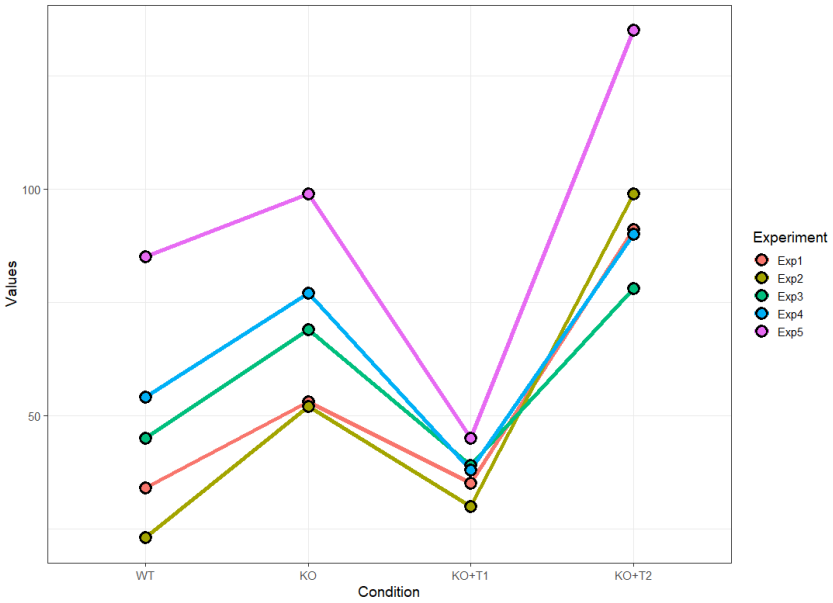
Table format: Column	Group A	Group B	Group C	Group D
	WT	KO	KO+T1	KO+T2
1 Exp1	34.00	53.00	35.00	91
2 Exp2	23.00	52.00	30.00	99
3 Exp3	45.00	69.00	39.00	78
4 Exp4	54.00	77.00	38.00	90
5 Exp5	85.00	99.00	45.00	135

Analysis of variance: Matched/repeated measures

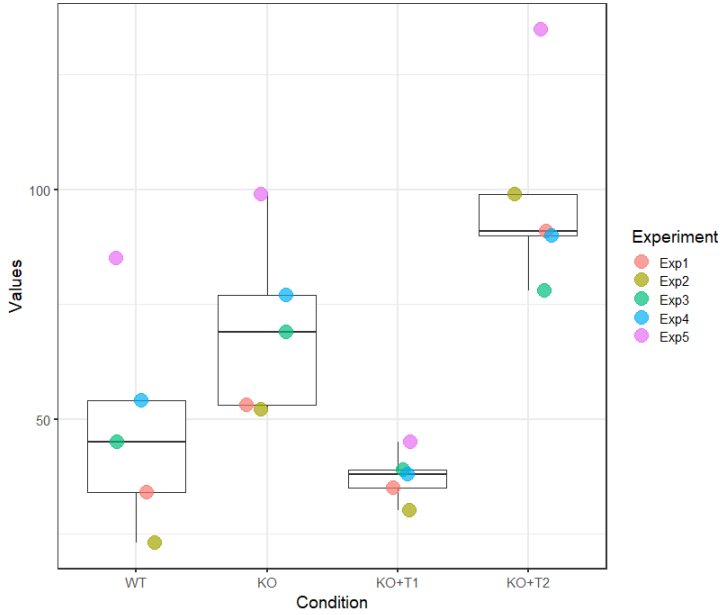
- Again, when plotting want to show matching

```
neutrophils.long %>%
  mutate(Condition=factor(Condition,
       levels = c("WT", "KO", "KO+T1", "KO+T2")))
```

```
neutrophils.long %>%
  ggplot(aes(x=Condition, y=Values, group=Experiment,
       colour=Experiment, fill=Experiment))+
  geom_line(linewidth=2)+
  geom_point(size=4, shape=21,
       colour="black", stroke=2)
```




```
neutrophils.long %>%
  ggplot(aes(x=Condition, y=Values, colour=Experiment))+
  geom_boxplot(outlier.shape = NA, colour="black")+
  geom_jitter(height=0, width=0.2,
       size=6, alpha=0.7)
```



Exercise 3

Babraham Institute

Analysis of Quantitative data

Two-way ANOVA

Hayley Carr & Anne Segonds-Pichon

v2025-02

Babraham Bioinformatics

Comparison between more than 2 groups

Two factors = Two predictors

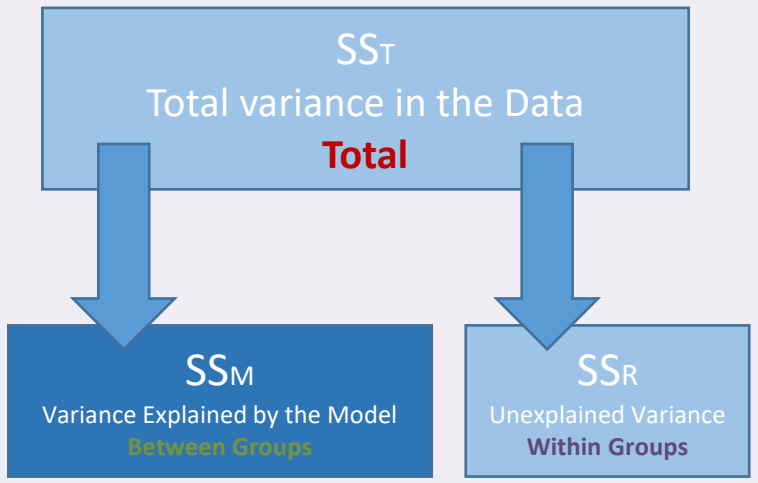
Two-Way ANOVA

Two-way Analysis of Variance (Factorial ANOVA)

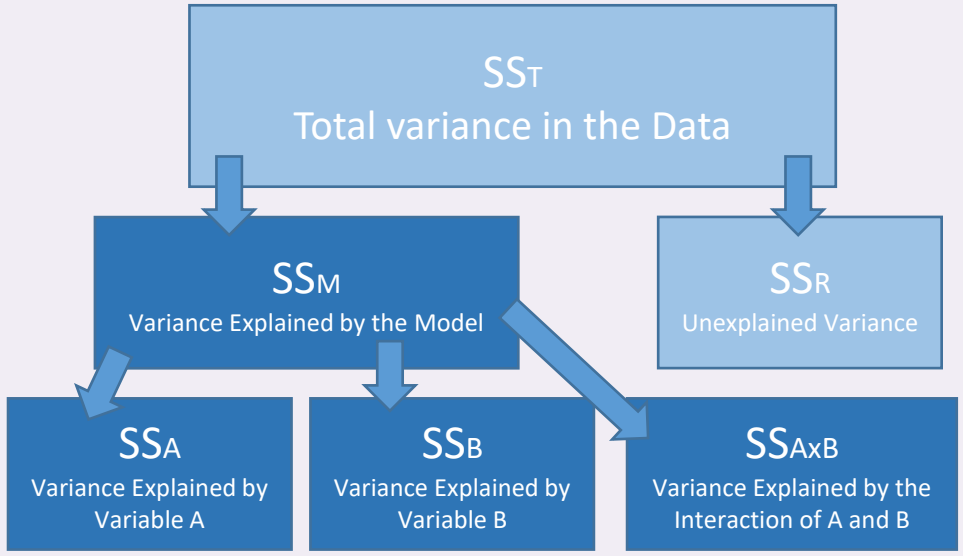
Source of variation	Sum of Squares	Df	Mean Square	F	p-value
Variable A (Between Groups)	2.665	4	0.6663	8.42	<0.0001
Within Groups (Residual)	5.775	73	0.0791		
Total	8.44	77			

Source of variation	Sum of Squares	Df	Mean Square	F	p-value
Variable A * Variable B	1978	2	989.1	F (2, 42) = 11.91	P < 0.0001
Variable B (Between groups)	3332	2	1666	F (2, 42) = 20.07	P < 0.0001
Variable A (Between groups)	168.8	1	168.8	F (1, 42) = 2.032	P = 0.1614
Residuals	3488	42	83.04		

One-way ANOVA= 1 predictor variable



2-way ANOVA= 2 predictor variables: A and B



Two-way Analysis of Variance

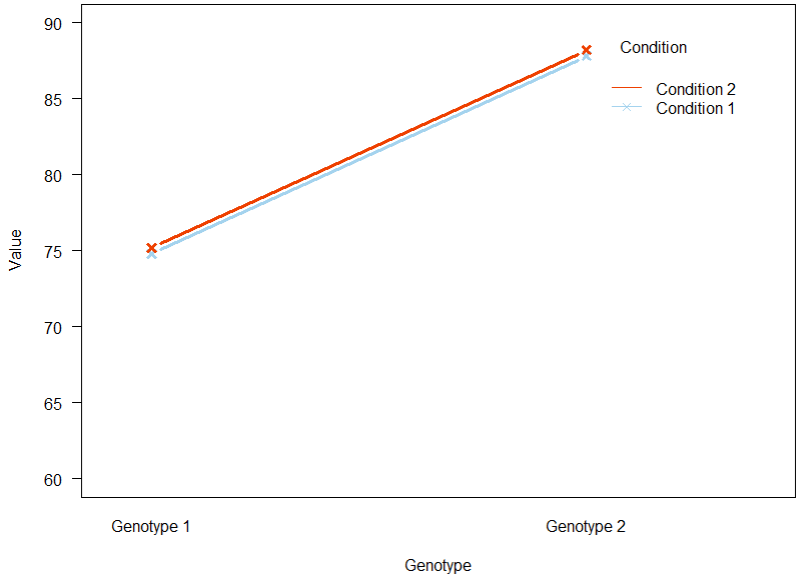
- **Interaction plots: Examples**
  - Fake dataset:
    - 2 factors: Genotype (2 levels) and Condition (2 levels)

Genotype	Condition	Value
Genotype 1	Condition 1	74.8
Genotype 1	Condition 1	65
Genotype 1	Condition 1	74.8
Genotype 1	Condition 2	75.2
Genotype 1	Condition 2	75
Genotype 1	Condition 2	75.2
Genotype 2	Condition 1	87.8
Genotype 2	Condition 1	65
Genotype 2	Condition 1	74.8
Genotype 2	Condition 2	88.2
Genotype 2	Condition 2	75
Genotype 2	Condition 2	75.2

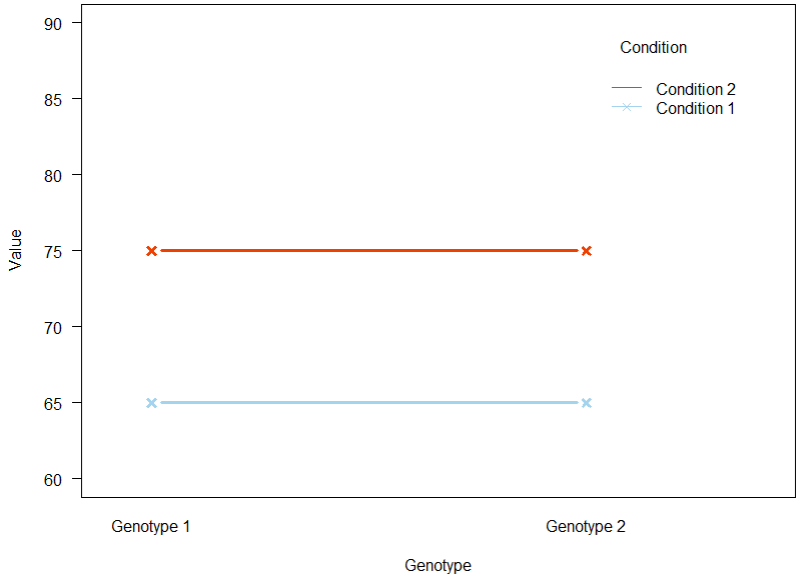
Two-way Analysis of Variance

- **Interaction plots: Examples**
  - 2 factors: Genotype (2 levels) and Condition (2 levels)

Single Effect



Genotype Effect

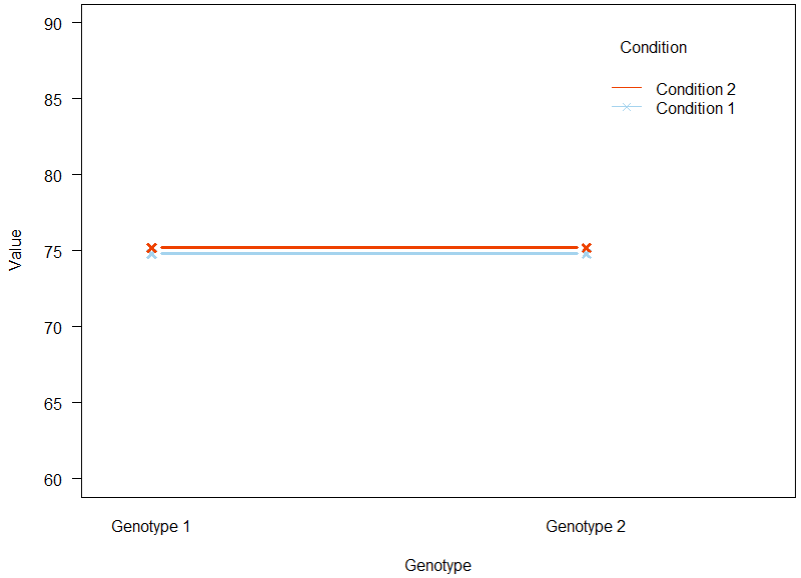


Condition Effect

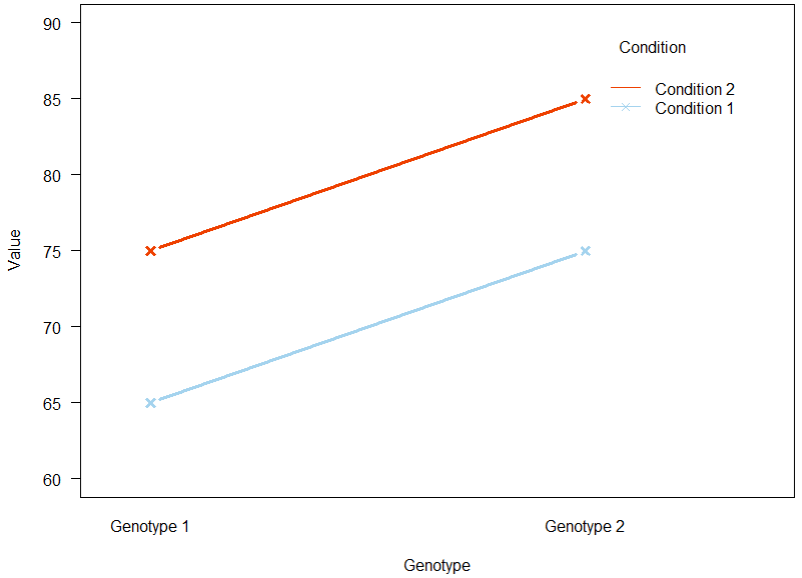
Two-way Analysis of Variance

- **Interaction plots: Examples**
  - 2 factors: Genotype (2 levels) and Condition (2 levels)

Zero or Both Effect



Zero Effect

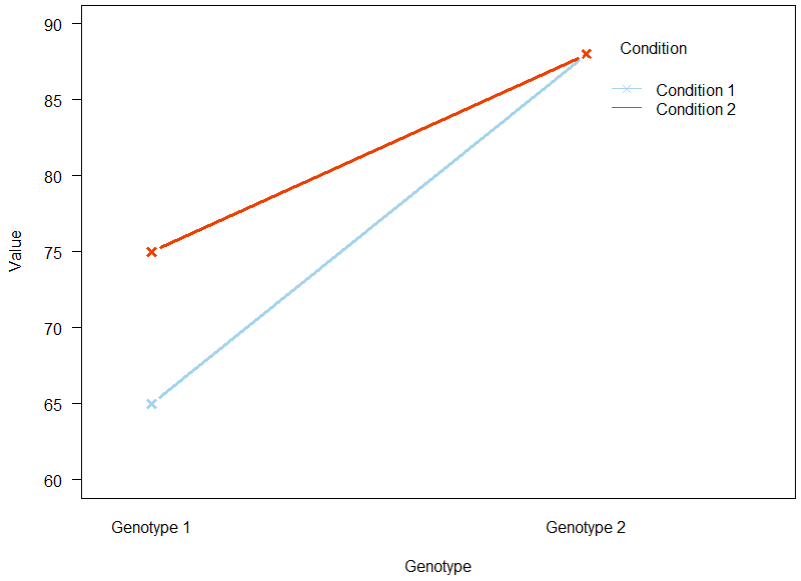


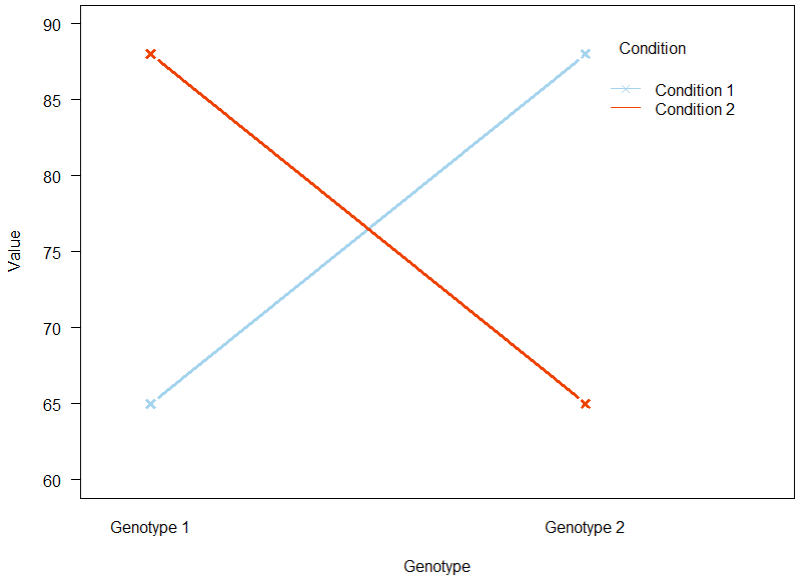
Both Effect

Two-way Analysis of Variance

- **Interaction plots: Examples**
  - 2 factors: Genotype (2 levels) and Condition (2 levels)

Interaction





Two-way Analysis of Variance

Example: crop.density.csv

- Want to know if planting density (1=low density, 2=high density) or fertiliser type (1, 2, or 3) have an impact on crop yield

- Three null hypotheses:
  - No difference in yield for any fertiliser type
  - No difference in yield for either planting density
  - Effect of fertiliser type or density on yield does not depend on the effect of the other variable

https://www.scribbr.com/statistics/two-way-anova/

Exercise: One-way ANOVA: Data Exploration

crop.density.csv

- Want to know if planting density (1=low density, 2=high density) or fertiliser type (1, 2, or 3) have an impact on crop yield

- Graphically explore the data
  - effect of density only
  - effect of fertiliser only
  - effect of both

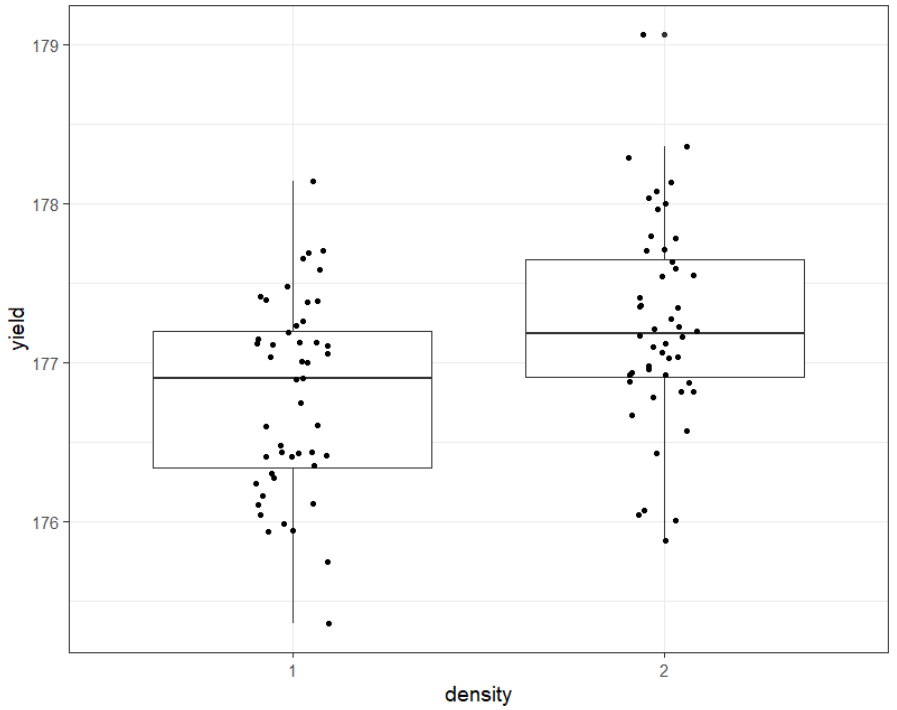
```
crop <- crop %>%
  mutate(density= factor(density),
         fertilizer = factor(fertilizer))
```

- Check the assumptions visually (plot+qqplot) and formally (test)

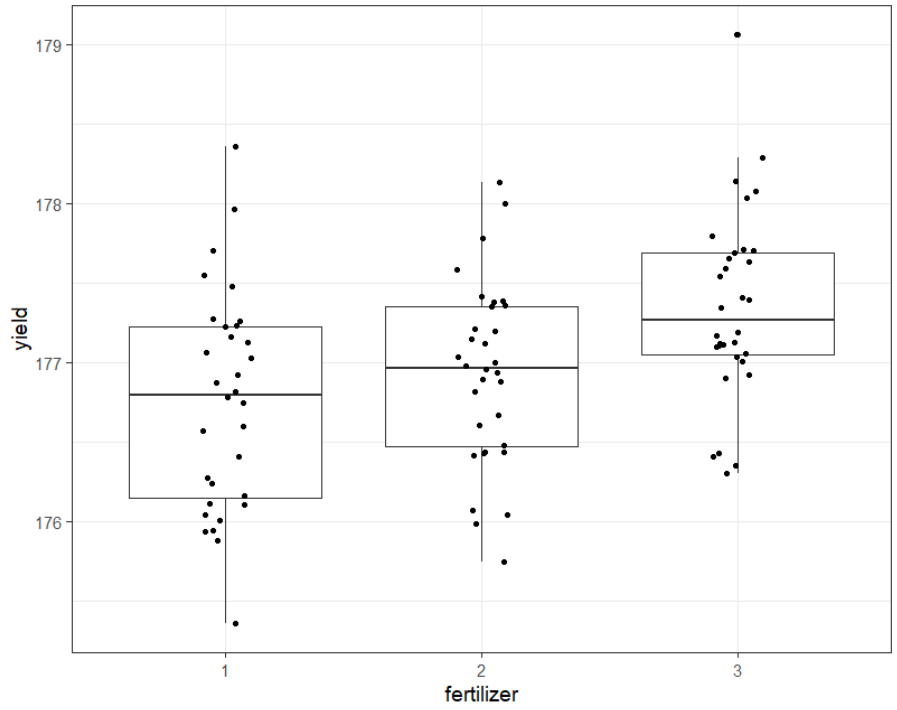
Two-way Analysis of Variance

- As always, first step: get to know the data

```
crop %>%
  ggplot(aes(density, yield))+
  geom_boxplot(outlier.shape = NA)+
  geom_jitter(height=0, width=0.1))
```



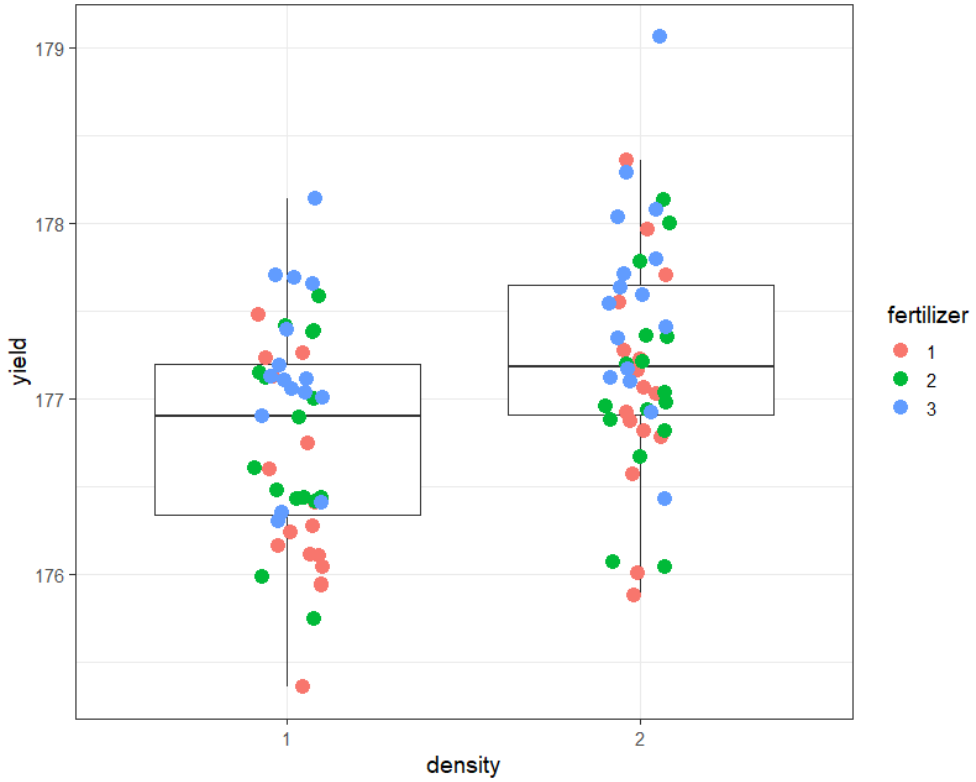

```
crop %>%
  ggplot(aes(fertilizer, yield))+
  geom_boxplot(outlier.shape = NA)+
  geom_jitter(height=0, width=0.1)
```

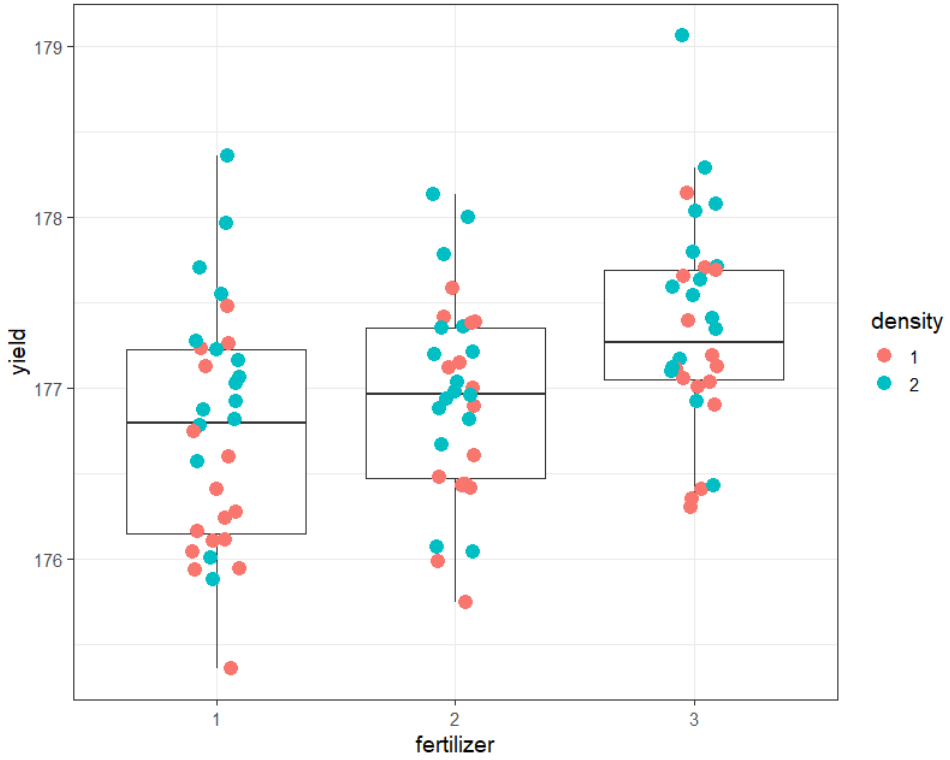


Two-way Analysis of Variance

- As always, first step: get to know the data

```
crop %>%  ggplot(aes(density, yield))+
  geom_boxplot(outlier.shape = NA)+
  geom_jitter(aes(density, yield, colour=fertilizer),
        height=0, width=0.1, size=4)
```

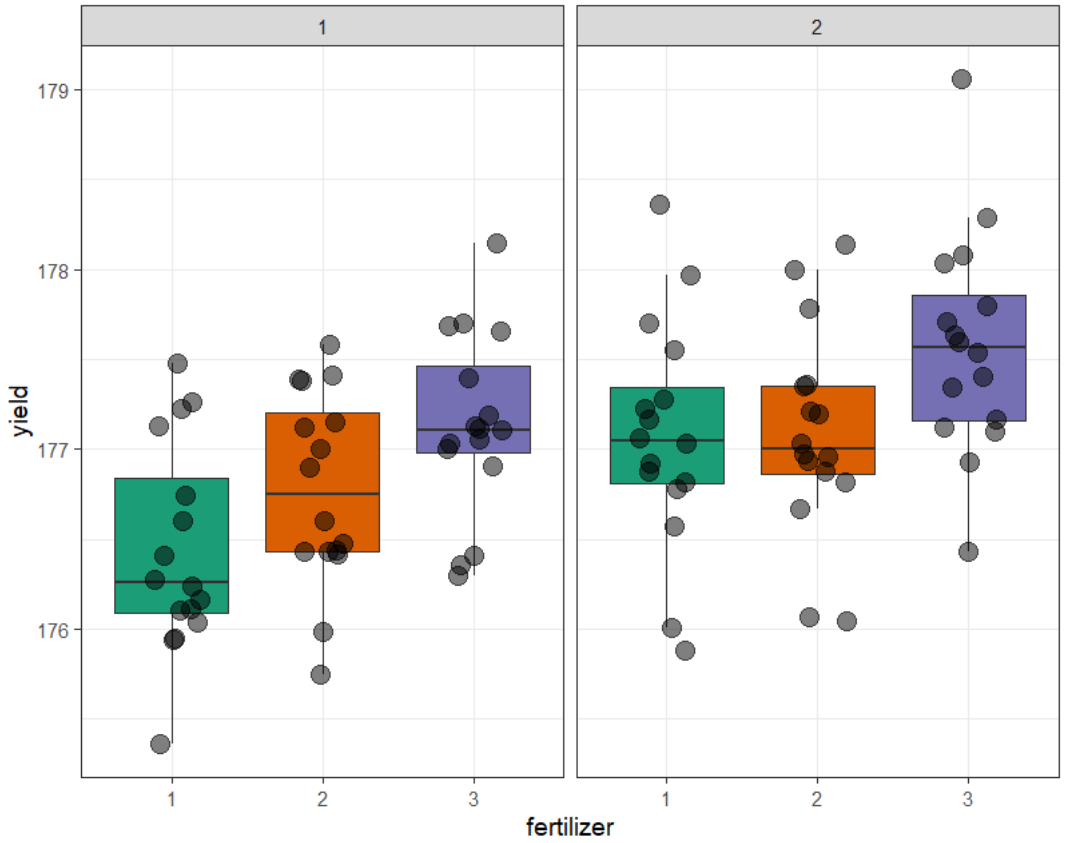





```
crop %>%  ggplot(aes(fertilizer, yield))+
  geom_boxplot(outlier.shape = NA)+
  geom_jitter(aes(fertilizer, yield, colour=density),
        height=0, width=0.1, size=4)
```

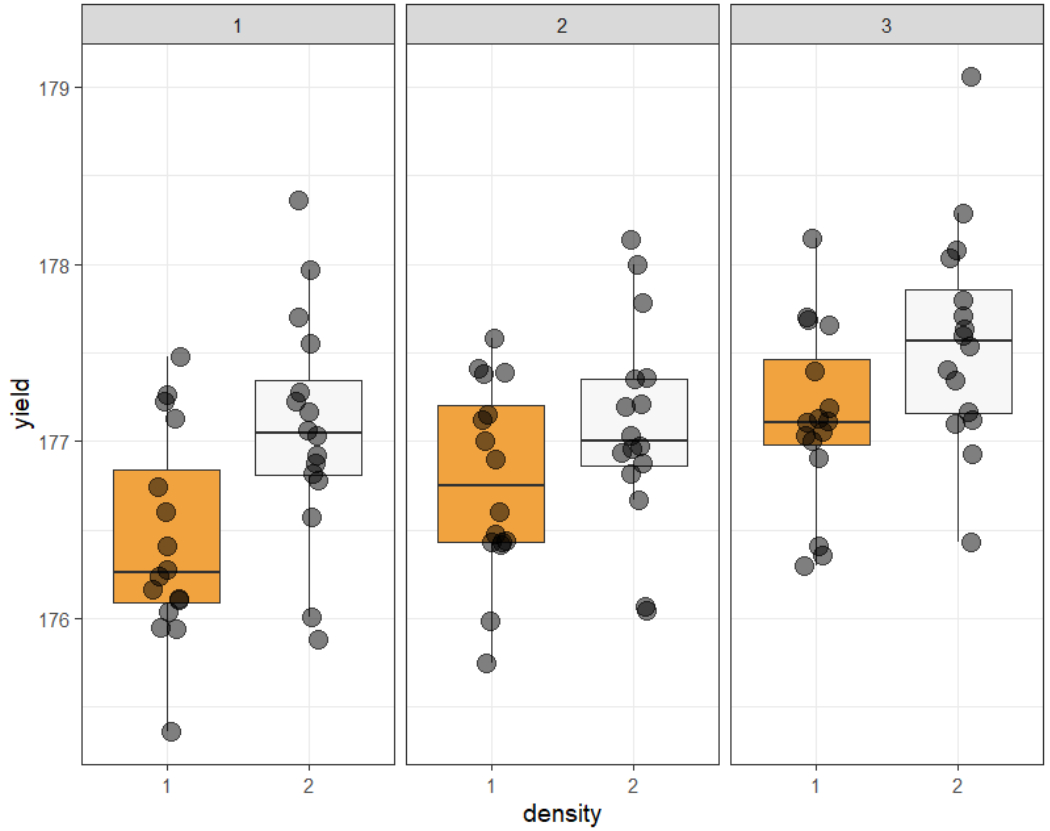
Two-way Analysis of Variance

```
crop %>%  ggplot(aes(x=fertilizer, y=yield, fill=fertilizer))+
  geom_boxplot(show.legend = FALSE, outlier.shape = NA)+
  geom_jitter(height=0, width=0.2, size=5, alpha=0.5, show.legend = FALSE)+
  facet_grid(cols=vars(density))+
  scale_fill_brewer(palette="Dark2")
```



Two-way Analysis of Variance

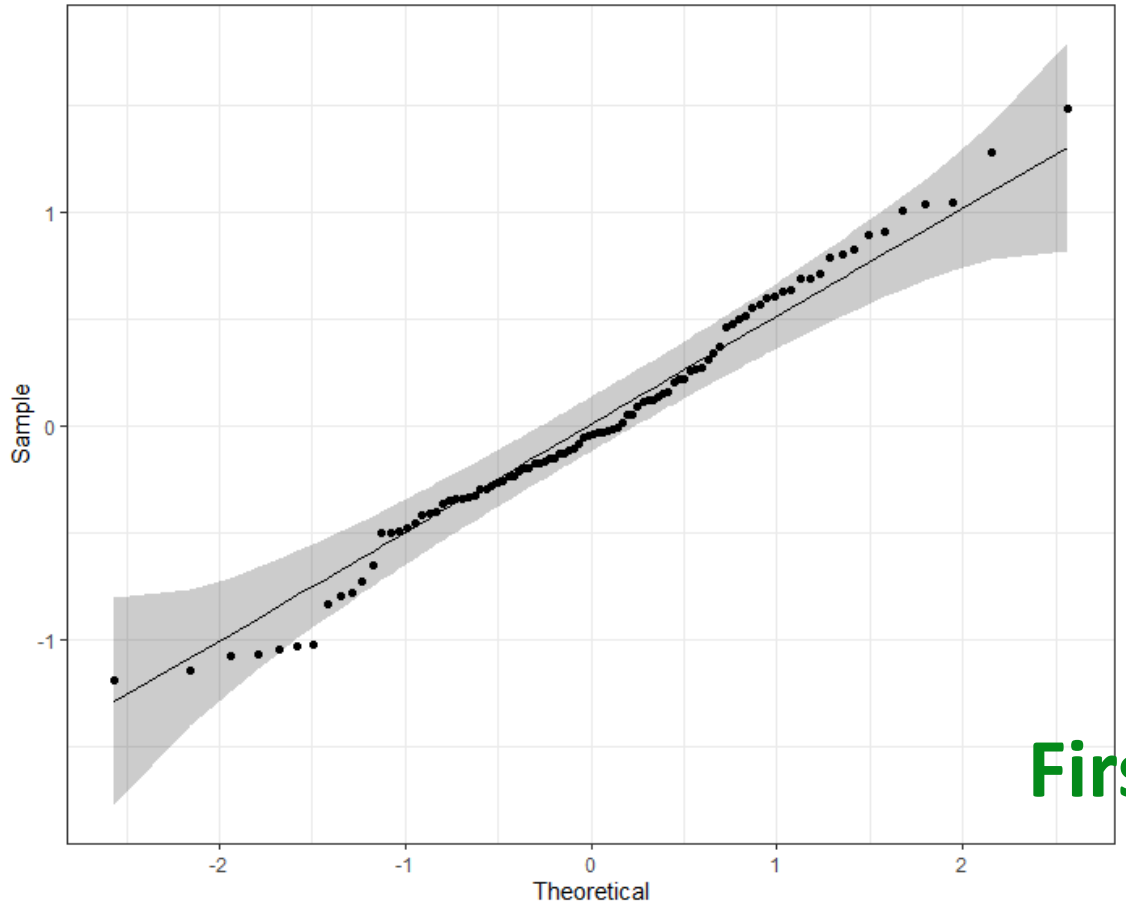
```
crop %>%  ggplot(aes(x=density, y=yield, fill=density))+
  geom_boxplot(show.legend = FALSE, outlier.shape = NA)+
  geom_jitter(height=0, width=0.1, size=5, alpha=0.5, show.legend = FALSE)+
  facet_grid(cols=vars(fertilizer))+
  scale_fill_brewer(palette="PuOr")
```



Two-way Analysis of Variance

Checking the assumptions

```
model  <- aov(yield ~ fertilizer*density, data = crop)
ggqqplot(residuals(model)) +  theme_bw()
```



First assumption ✓

Two-way Analysis of Variance

Checking the assumptions

```
model  <-
  aov(yield ~ fertilizer*density,
      data = crop)
shapiro_test(residuals(model))
```

```
# A tibble: 1 x 3
  variable          statistic p.value
  <chr>                 <dbl>   <dbl>
1 residuals(model)      0.985   0.360
```

First assumption ✓

```
crop %>%
  group_by(fertilizer, density) %>%
  shapiro_test(yield)
```

```
# A tibble: 6 x 5
  density fertilizer variable statistic     p
  <fct>   <fct>      <chr>         <dbl> <dbl>
1 1       1          yield         0.937 0.315
2 2       1          yield         0.972 0.865
3 1       2          yield         0.942 0.373
4 2       2          yield         0.948 0.466
5 1       3          yield         0.943 0.390
6 2       3          yield         0.970 0.842
```

```
crop %>%
  levene_test(yield ~ fertilizer*density)
```

```
    df1   df2 statistic     p
  <int> <int>     <dbl> <dbl>
      5    90     0.159 0.977
```

Second assumption ✓

Two-way Analysis of variance

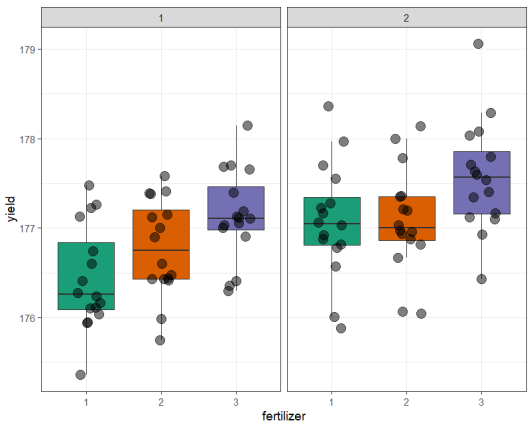
Let's do it

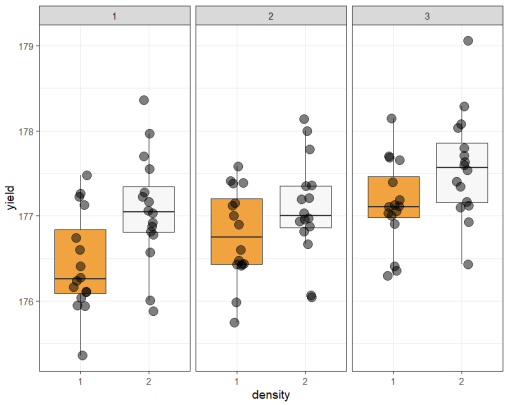
- Run the first step of the ANOVA

```
data %>%
  anova_test(y ~ factor1 + factor2 + factor1*factor2)
```

- Run the second step (post-hoc tests)

```
data %>%
  tukey_hsd(y ~ factor1*factor2)
```





- Run post-hoc tests by fertiliser and density

- **Extra task**: plot the stats results on the graphs

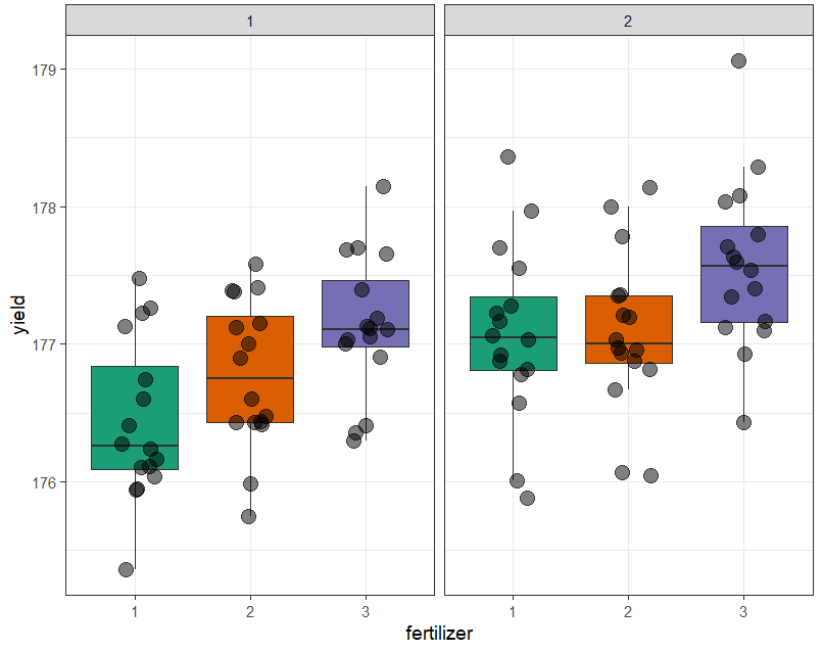
Two-way Analysis of Variance

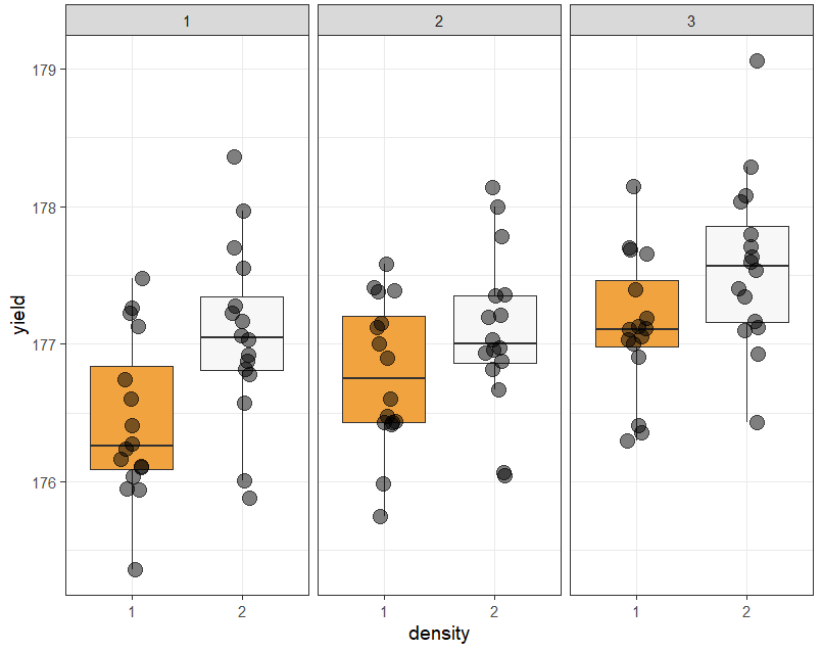
Omnibus test

```
crop %>%
  anova_test(yield ~ density + fertilizer + density*fertilizer)
```

```
ANOVA Table (type II tests)

             Effect DFn DFd      F        p p<.05   ges
1           density   1  90 15.195 0.000186     * 0.144
2        fertilizer   2  90  9.001 0.000273     * 0.167
3 density:fertilizer   2  90  0.635 0.533000       0.014
```

Gives same results as `density*fertilizer` but explicitly specifies





Two-way Analysis of Variance

Post-hoc tests

```
crop %>%
tukey_hsd(yield ~ fertilizer*density)
```

Gives all comparisons, can be too much: overcorrecting!

	term	group1	group2	null.value	estimate	conf.low	conf.high	p.adj	p.adj.signif
*	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	fertilizer	1	2	0	0.176	-0.170	0.522	0.448	ns
2	fertilizer	1	3	0	0.599	0.253	0.945	0.000239	***
3	fertilizer	2	3	0	0.423	0.0771	0.769	0.0124	*
4	density	1	2	0	0.462	0.227	0.697	0.000186	***
5	fertilizer:density	1:1	2:1	0	0.339	-0.259	0.936	0.568	ns
6	fertilizer:density	1:1	3:1	0	0.696	0.0983	1.29	0.0129	*
7	fertilizer:density	1:1	1:2	0	0.635	0.0372	1.23	0.0307	*
8	fertilizer:density	1:1	2:2	0	0.649	0.0508	1.25	0.0254	*
9	fertilizer:density	1:1	3:2	0	1.14	0.539	1.73	0.00000438	****
10	fertilizer:density	2:1	3:1	0	0.357	-0.240	0.955	0.509	ns
11	fertilizer:density	2:1	1:2	0	0.296	-0.302	0.894	0.701	ns
12	fertilizer:density	2:1	2:2	0	0.310	-0.288	0.908	0.659	ns
13	fertilizer:density	2:1	3:2	0	0.798	0.201	1.40	0.00257	**
14	fertilizer:density	3:1	1:2	0	-0.0611	-0.659	0.537	1	ns
15	fertilizer:density	3:1	2:2	0	-0.0475	-0.645	0.550	1	ns
16	fertilizer:density	3:1	3:2	0	0.441	-0.157	1.04	0.272	ns
17	fertilizer:density	1:2	2:2	0	0.0136	-0.584	0.611	1	ns
18	fertilizer:density	1:2	3:2	0	0.502	-0.0955	1.10	0.152	ns
19	fertilizer:density	2:2	3:2	0	0.489	-0.109	1.09	0.174	ns

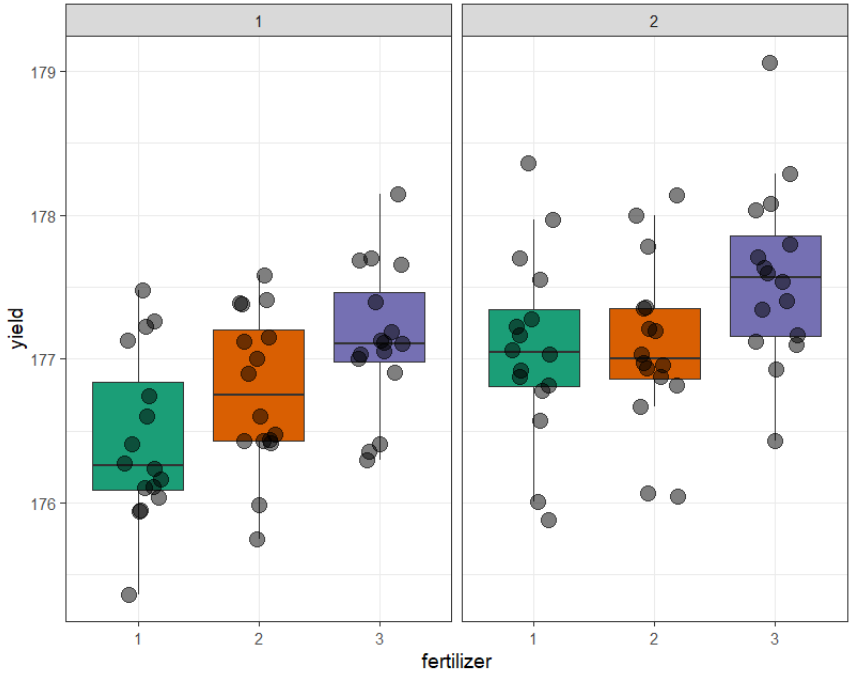
Two-way Analysis of Variance

Post-hoc tests by density level

More specific – fewer unnecessary comparisons

```
crop %>%
  group_by(density) %>%
  emmeans_test(yield ~ fertilizer,
               p.adjust.method = "bonferroni")
```

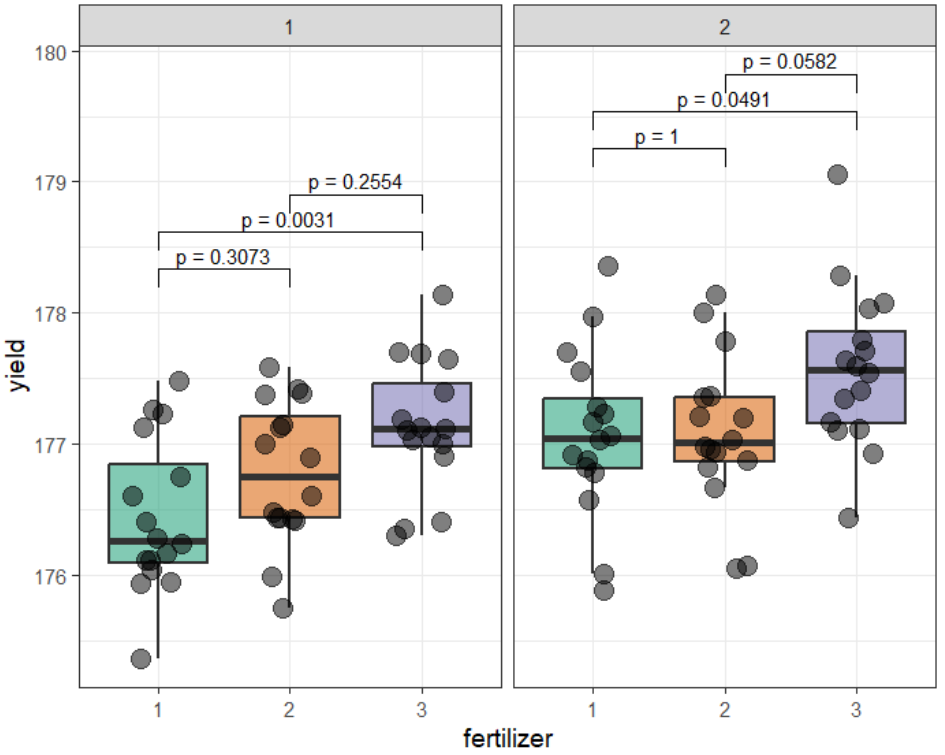
```
# A tibble: 6 × 10
  density term       .y.   group1 group2    df statistic       p   p.adj p.adj.signif
* <fct>   <chr>      <chr> <chr>  <chr>  <dbl>     <dbl>   <dbl>   <dbl> <chr>
1 1       fertilizer yield 1      2         90     -1.65   0.102   0.307   ns
2 1       fertilizer yield 1      3         90     -3.39   0.00104 0.00311 **
3 1       fertilizer yield 2      3         90     -1.74   0.0851  0.255   ns
4 2       fertilizer yield 1      2         90     -0.0665 0.947   1       ns
5 2       fertilizer yield 1      3         90     -2.45   0.0164  0.0491  *
6 2       fertilizer yield 2      3         90     -2.38   0.0194  0.0582  ns
```



Two-way Analysis of Variance

Post-hoc tests by density level: with p-values on graph

```
crop %>%
  group_by(density) %>%
  emmeans_test(yield ~ fertilizer,
       p.adjust.method = "bonferroni")%>%
  add_xy_position(x = "fertilizer") %>%
  ungroup()-> results.density
```




```
crop %>%
  ggplot(aes(x=fertilizer, y=yield))+
  geom_boxplot(linewidth=1, aes(fill = fertilizer, alpha=0.5), show.legend = FALSE,
       outlier.shape = NA)+
  geom_jitter(height=0, width=0.2, size=5, alpha=0.5, show.legend = FALSE)+
  facet_grid(cols=vars(density))+
  scale_fill_brewer(palette="Dark2")+
  stat_pvalue_manual(results.density, label = "p = {round(p.adj, digits=4)}")
```

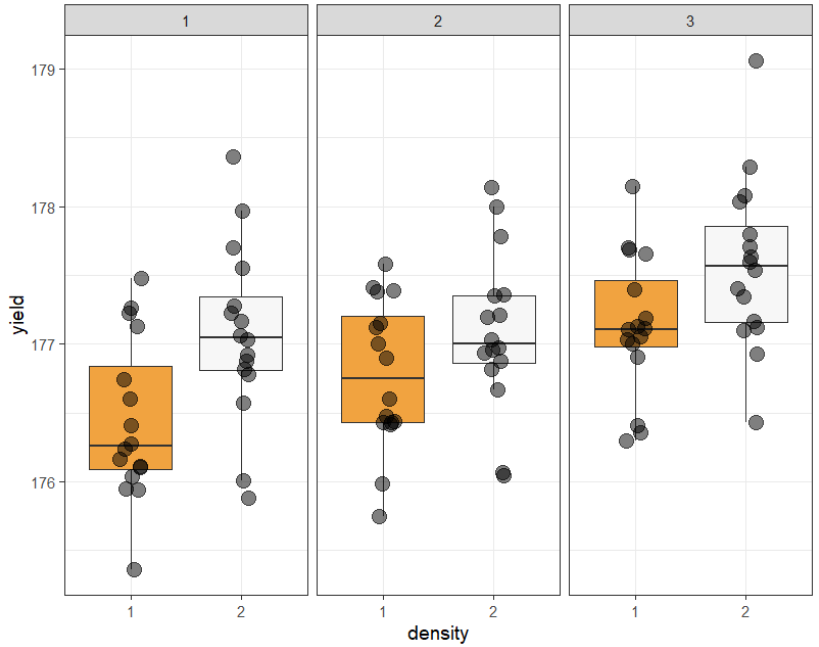
ggpubr package

Two-way Analysis of Variance

Post-hoc tests by fertilizer

```
crop %>%
  group_by(fertilizer) %>%
  emmeans_test(yield ~ density, p.adjust.method = "bonferroni")
```

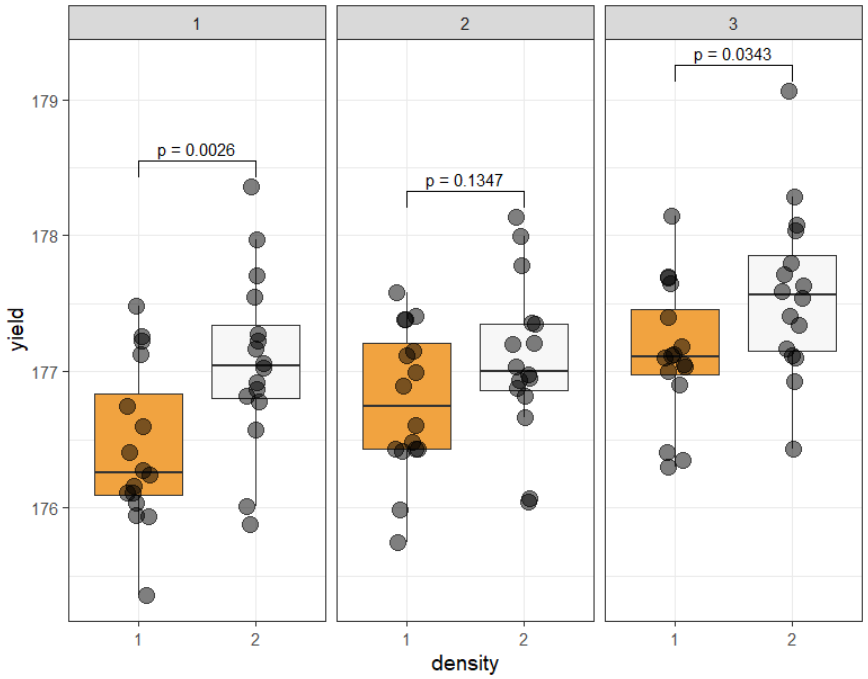
fertilizer	term	.y.	group1	group2	df	statistic	p	p.adj	p.adj.signif
<fct>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	density	yield	1	2	90	-3.09	0.00264	0.00264	**
2	density	yield	1	2	90	-1.51	0.135	0.135	ns
3	density	yield	1	2	90	-2.15	0.0343	0.0343	*



Two-way Analysis of Variance

Post-hoc tests by fertilizer with p-values on graph

```
crop %>%
  group_by(fertilizer) %>%
  emmeans_test(yield ~ density, p.adjust.method = "bonferroni")%>%
  add_xy_position(x = "density") %>%
  ungroup()-> results.fertilizer
```




```
crop %>%
  ggplot(aes(x=density, y=yield))+
  geom_boxplot(show.legend = FALSE, outlier.shape = NA, aes(fill=density))+
  geom_jitter(height=0, width=0.1, size=5, alpha=0.5, show.legend = FALSE)+
  facet_grid(cols=vars(fertilizer))+
  scale_fill_brewer(palette="PuOr")+
  stat_pvalue_manual(results.fertilizer, label = "p = {round(p.adj, digits=4)}")
```

ggpubr package

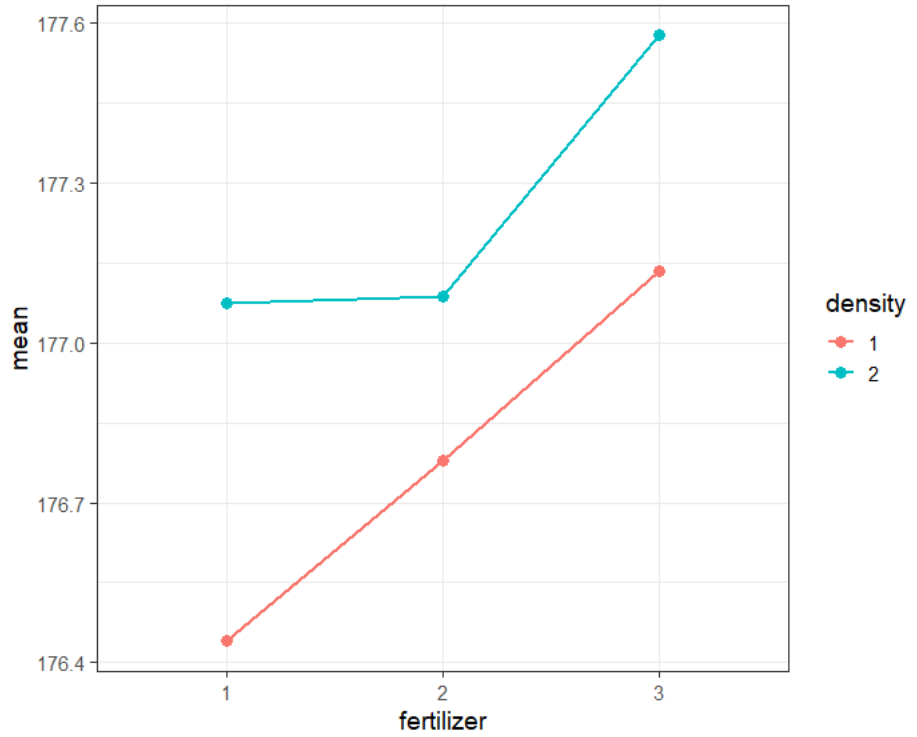
Two-way Analysis of Variance

- Now a quick way to have a look at the interaction

```
crop %>%
  group_by(fertilizer, density)%>%
    summarise(mean=mean(yield))
      -> crop.summary
```

fertilizer	density	mean
<fct>	<fct>	<dbl>
1	1	176.
1	2	177.
2	1	177.
2	2	177.
3	1	177.
3	2	178.

```
crop.summary %>%
  ggplot(aes(x=fertilizer, y= mean,
      colour=density, group=density))+
  geom_line(size = 1)+
  geom_point(size = 3)
```



Babraham Institute

Analysis of Quantitative data
Correlation & linear regression

Hayley Carr & Anne Segonds-Pichon
v2025-02

Babraham Bioinformatics

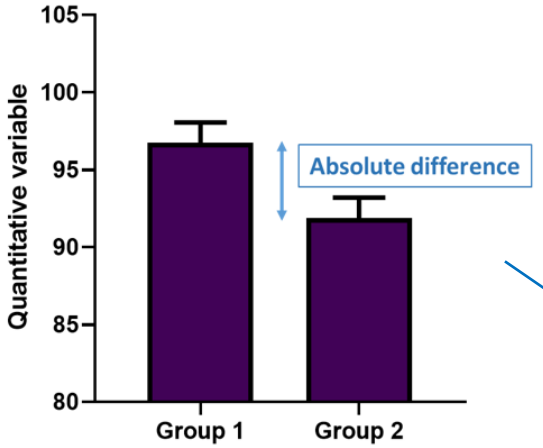
Association between 2 continuous variables

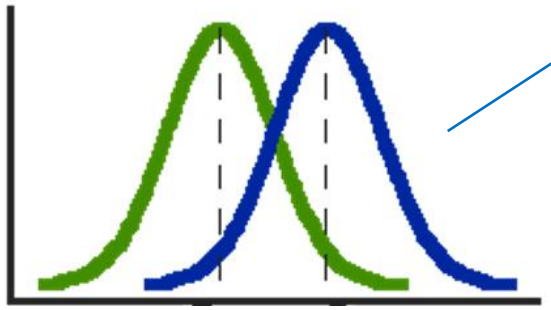
One variable X and One variable Y

Linear relationship

Correlation and Regression

Signal-to-noise ratio





$$\frac{\text{Difference (signal)}}{\text{Variation (noise)}}$$

$$\frac{\text{Signal}}{\text{Noise}} = \textbf{statistical significance}$$

$$\frac{\text{Signal}}{\text{Noise}} = \textbf{no statistical significance}$$

Signal-to-noise ratio and Correlation

$$\frac{\text{Difference (signal)}}{\text{Variation (noise)}}$$

- For correlation, signal is **similarity** of behaviour between variable x and variable y
- **Coefficient of correlation**: $r = \frac{\text{Similarity}}{\text{Variability}} = \frac{\text{Signal}}{\text{Noise}}$

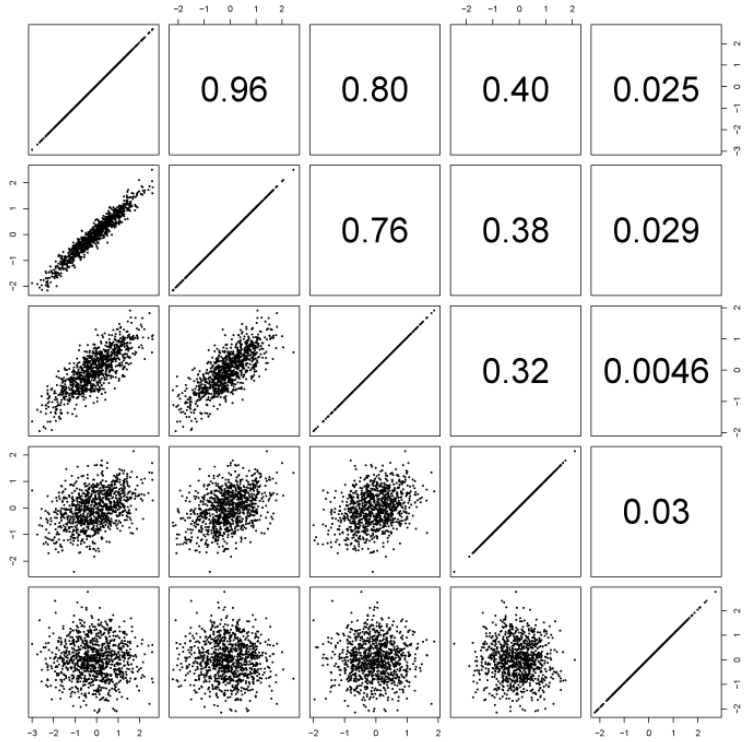
$$r = \frac{\text{Similarity}}{\text{Variability}} = \frac{COV_{xy}}{SD_x SD_y}$$

Covariance

Standard Deviation

Correlation

- Most widely-used correlation coefficient:
  - **Pearson product-moment correlation coefficient: r**
    - The **magnitude** and the **direction** of the relation between 2 variables
    - Designed to range in value between **-1 and +1**
    - Often look for **>|0.6|**
  - **Coefficient of determination: r^2**
    - Gives the proportion of variance in Y that can be explained by X
      - Helps with the interpretation of r
      - Basically the **effect size**



Coefficient (+ve or –ve)	Strength of relationship
0.0 to 0.2	Negligible
0.2 to 0.4	Weak
0.4 to 0.7	Moderate
0.7 to 0.9	Strong
0.9 to 1.0	Very strong

Correlation

p = 0.0002

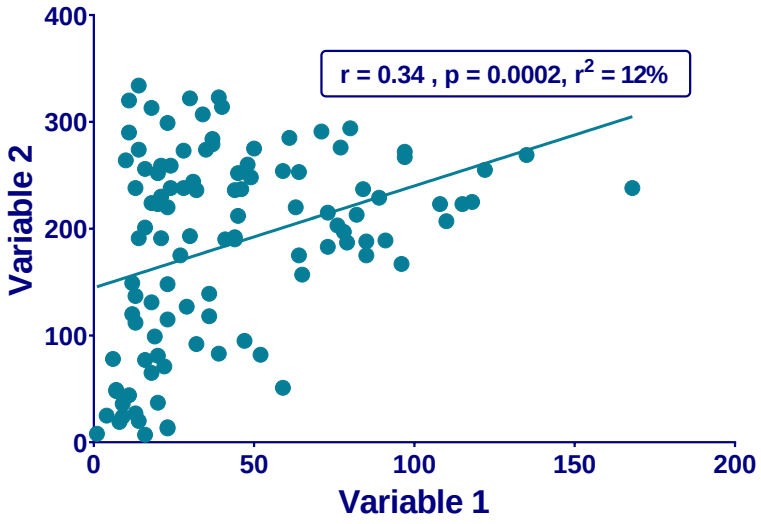
r = 0.34

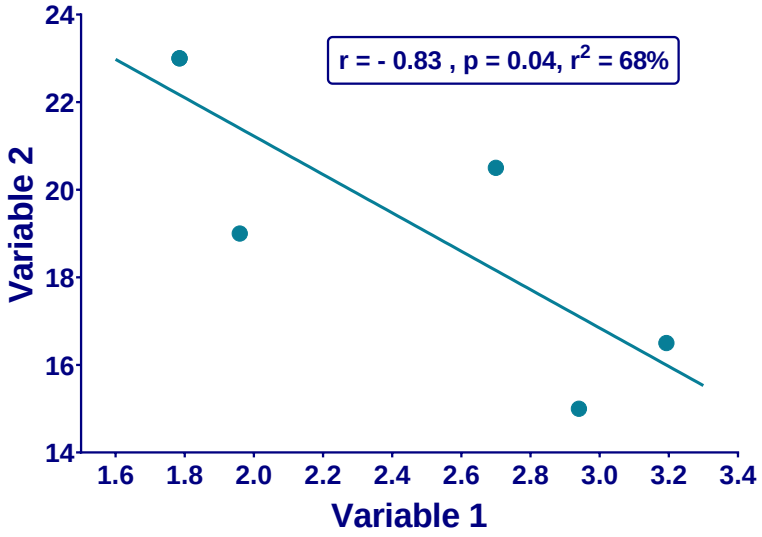
r^2 = 0.12

p = 0.04

r = - 0.83

r^2 = 0.68





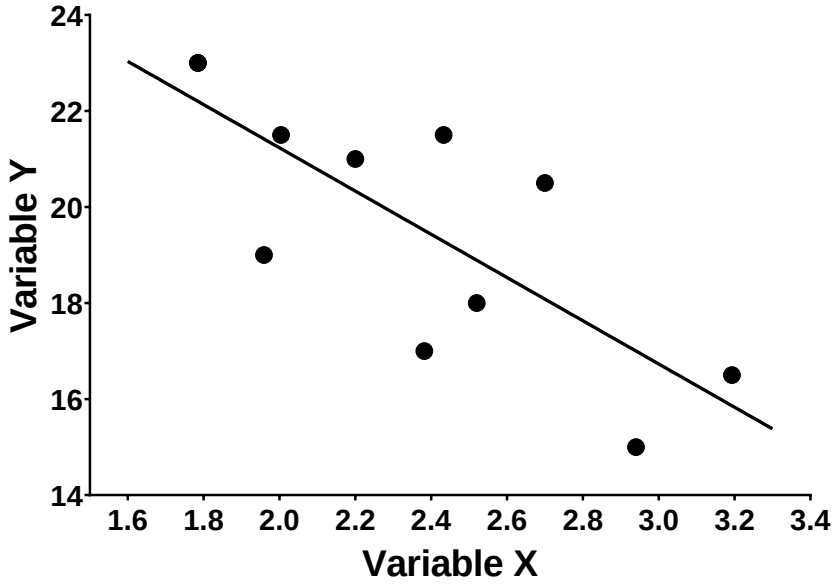
Power!!

Correlation: Assumptions

Pearson correlation is a parametric test for linear relationships

First assumption for parametric test: Normality

Correlation: bivariate Gaussian distribution



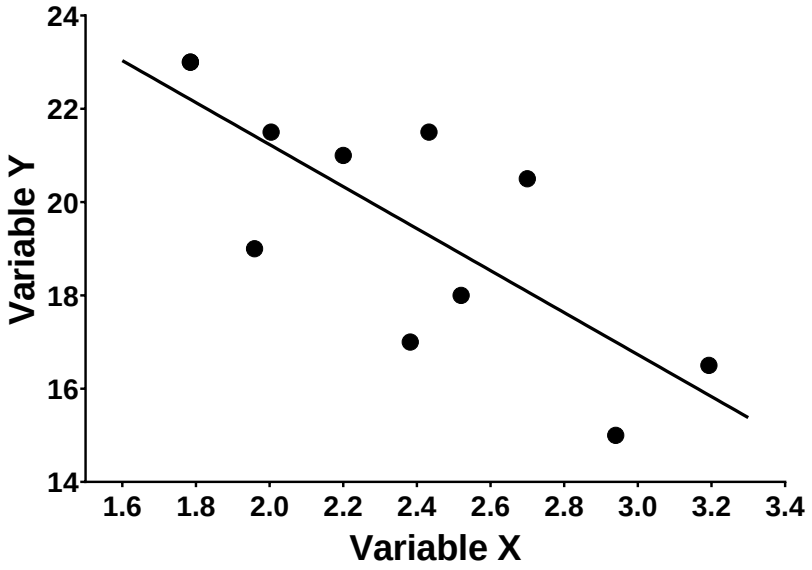
Symmetry of the values on either side of the line of best fit.

Correlation and regression

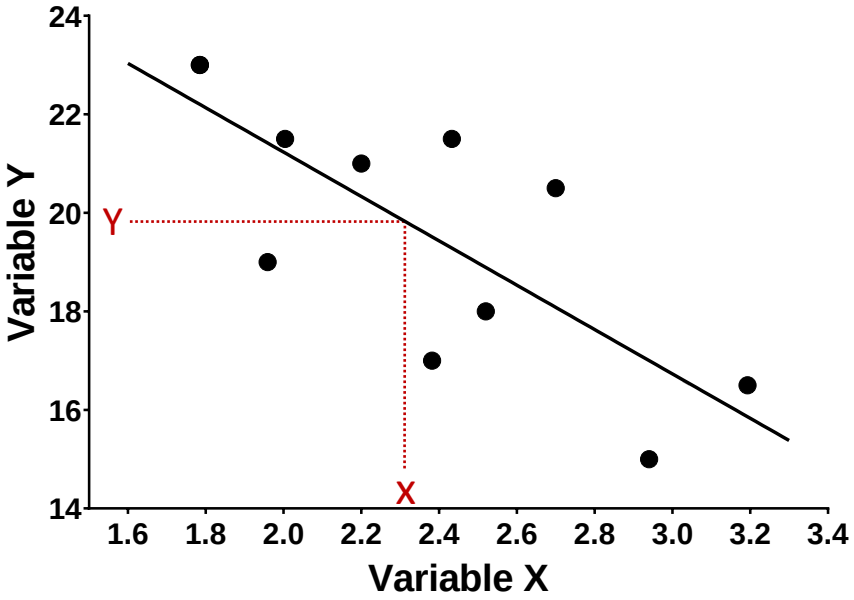
Line of best fit comes from a regression

Correlation: nature and strength of the association

Regression: nature and strength of the association and prediction



Correlation = Association



Regression = Prediction

Y = A*X + B

Correlation

treelight.csv

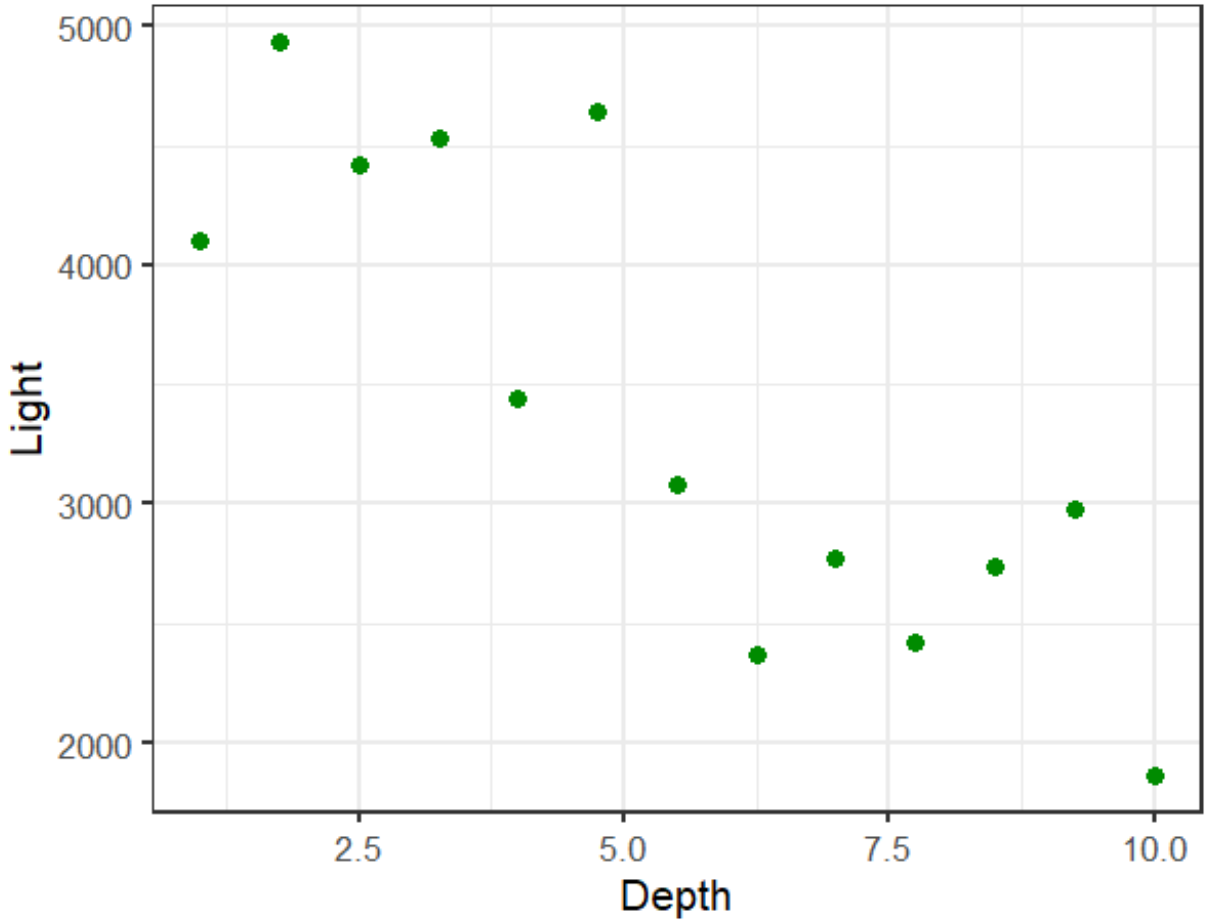
Amount of light in a tree

Correlation: treelight.csv

- **Question**:
  - What is the nature and the strength of the relationship between depth and light?

Read in the data and create initial plot

```
read_csv("treelight.csv") -> treelight

treelight %>% ggplot(aes(Depth, Light))+
  geom_point(size=3, colour="green4")
```



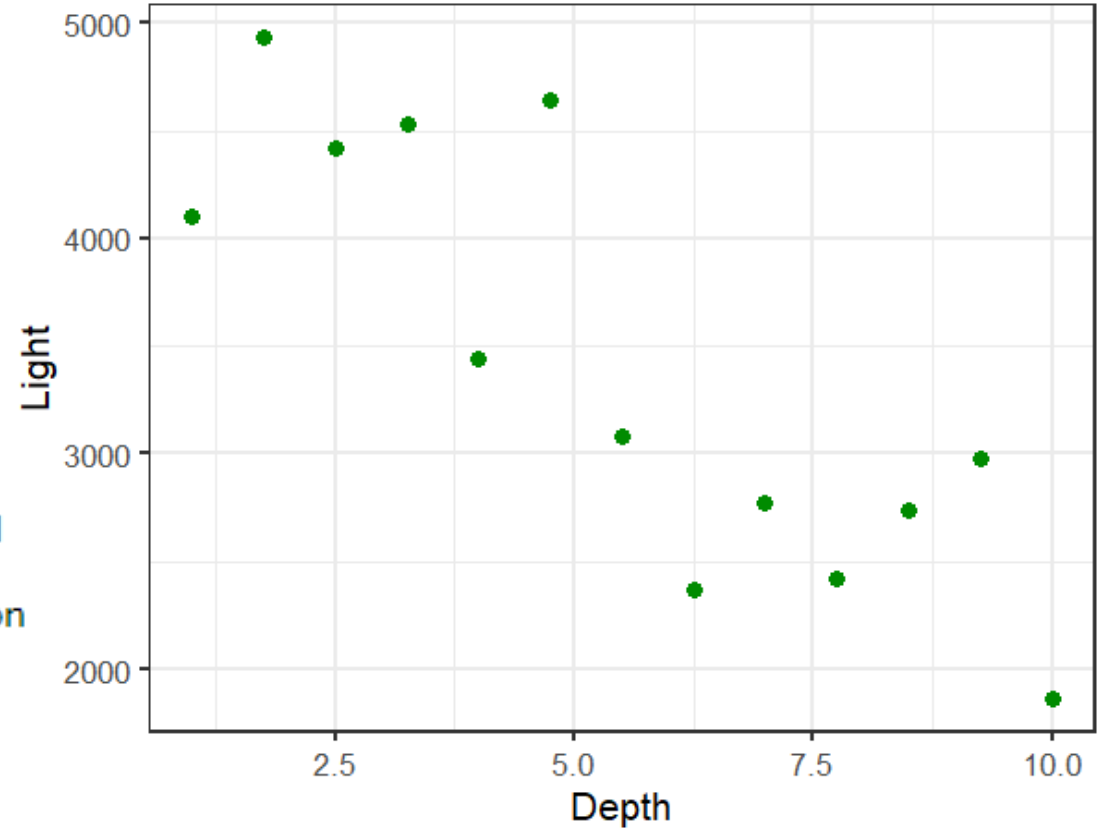
Correlation: treelight.csv

- **Question**:
  - What is the nature and the strength of the relationship between depth and light?

rstatix package

```
treelight %>%
   cor_test(Depth, Light)
```

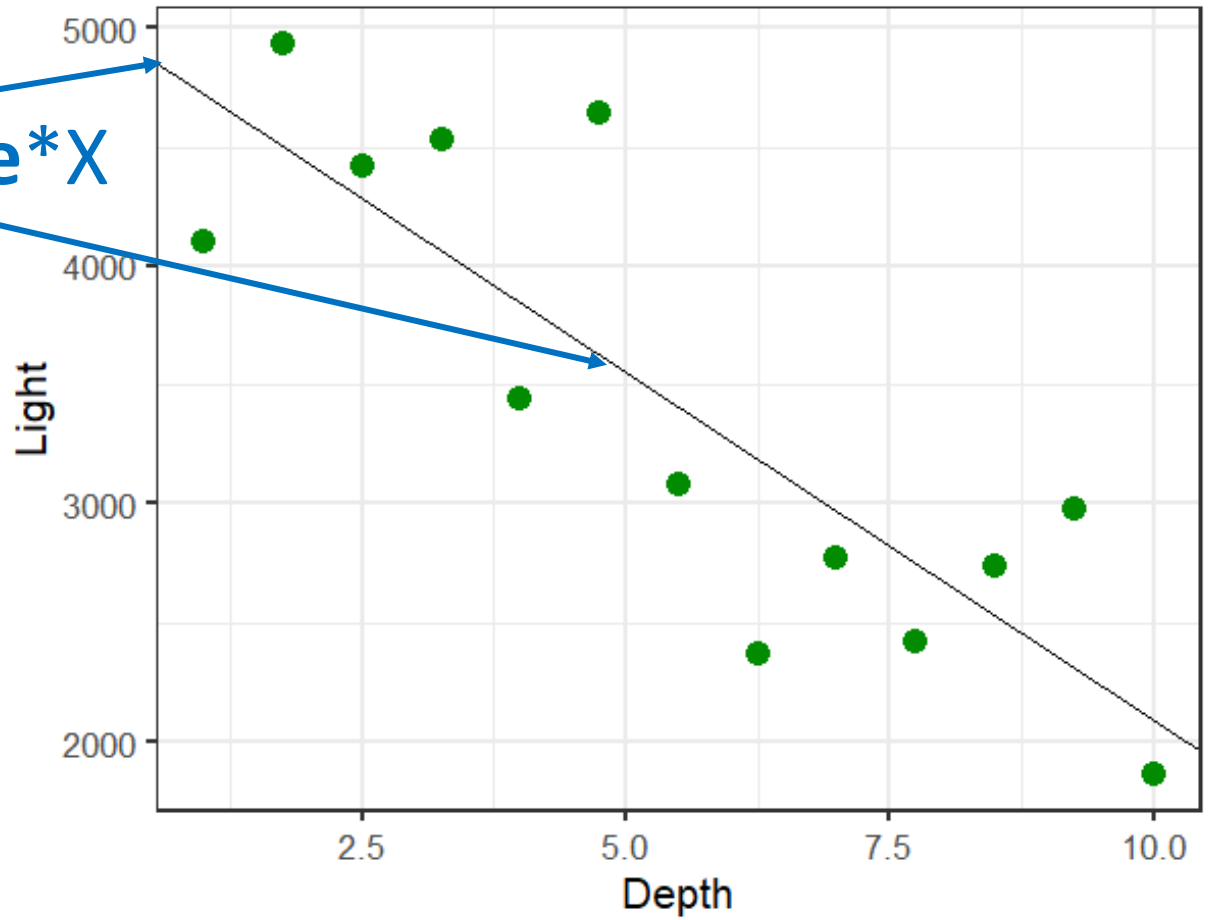
```
# A tibble: 1 × 8
  var1  var2    cor statistic        p conf.low conf.high method
  <chr> <chr> <dbl>     <dbl>    <dbl>    <dbl>     <dbl> <chr>
1 Depth Light -0.85     -5.27 0.000263   -0.953    -0.554 Pearson
```



Correlation: treelight.csv

- Next we want to add a **line of best-fit:** Y = A + B*X

Y=**intercept** + **slope***X



Correlation: treelight.csv

- For the line of best-fit: 3 new functions

```
lm(y~x, data=) -> fit
coefficients(fit) -> coef.fit
geom_abline(intercept=coef.fit[1], slope=coef.fit[2])
```

(vector of 2 values)

Core R

With the tree data:

```
lm(Light ~ Depth, data=treelight)-> fit.treelight
coefficients(fit.treelight) -> coef.treelight
coef.treelight
```

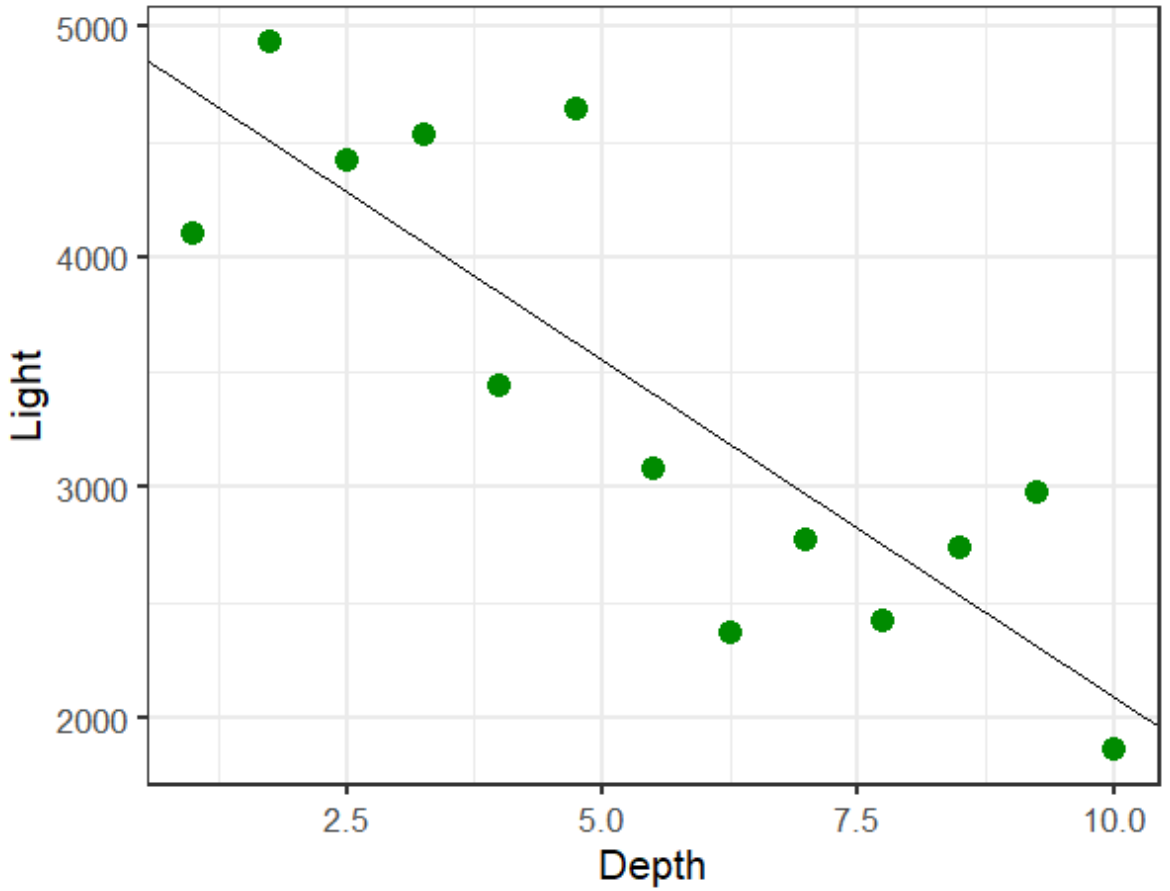
```
(Intercept)       Depth
  5013.9822   -292.1614
```

intercept slope

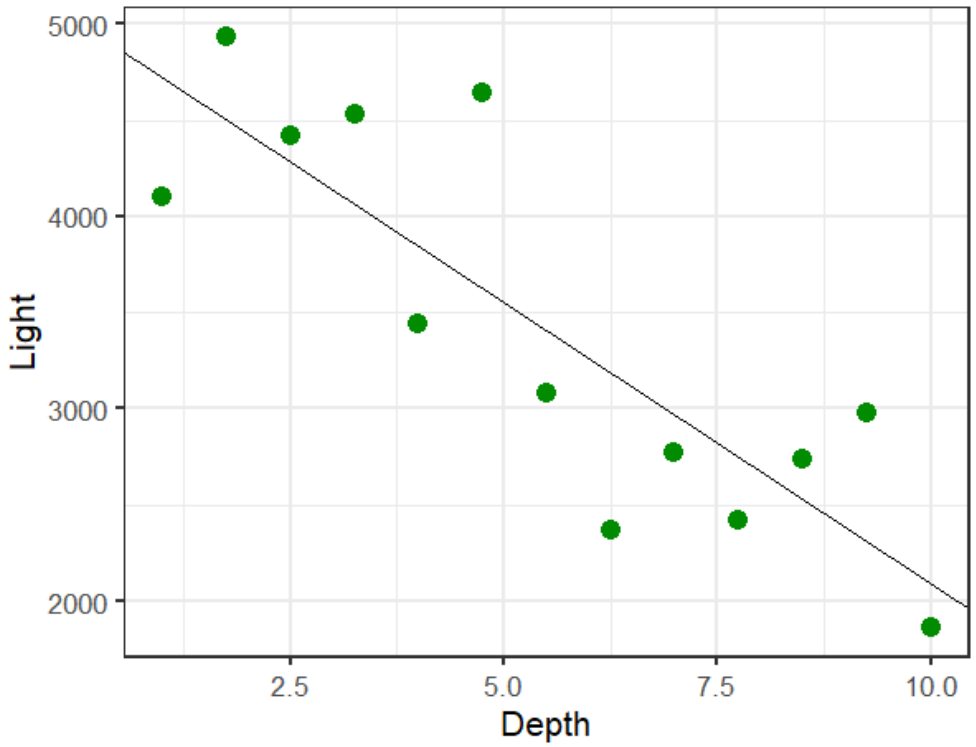
```
 coef.treelight[1]
(Intercept)
   5013.982
```

Correlation: treelight.csv

```
treelight %>%
  ggplot(aes(x=Depth, y=Light)) +
  geom_point(size=4, colour="green4") +
  geom_abline(intercept = coef.treelight[1], slope = coef.treelight[2])
```



Correlation: treelight.csv




```
summary(fit.treelight)
```

Line of best fit: Y=5013.98 – 292.16*X

```
Call:
lm(formula = Light ~ Depth, data = treelight)

Residuals:
   Min     1Q Median     3Q    Max
-819.9 -330.5 -192.3  431.2 1014.1

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5013.98     342.15  14.654 1.46e-08 ***
Depth        -292.16      55.41  -5.272 0.000263 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 560.7 on 11 degrees of freedom
Multiple R-squared:  0.7165,    Adjusted R-squared:  0.6907
F-statistic:  27.8 on 1 and 11 DF,  p-value: 0.0002633
```

```
treelight %>%
   cor_test(Depth, Light)
```

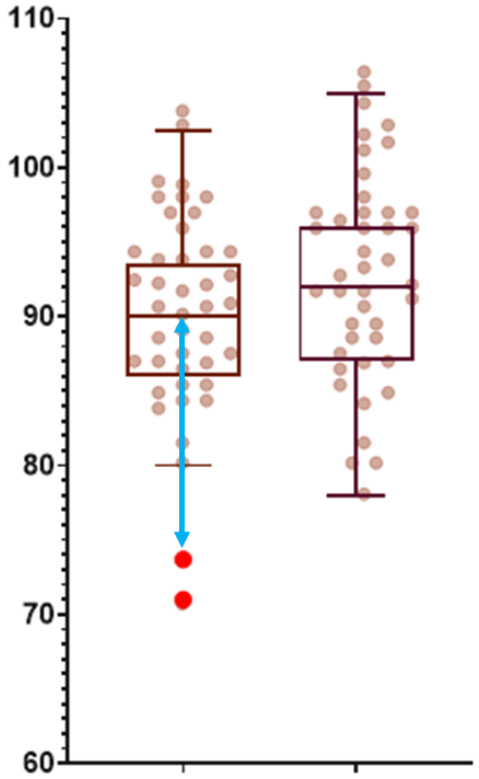
sqrt(0.7165)

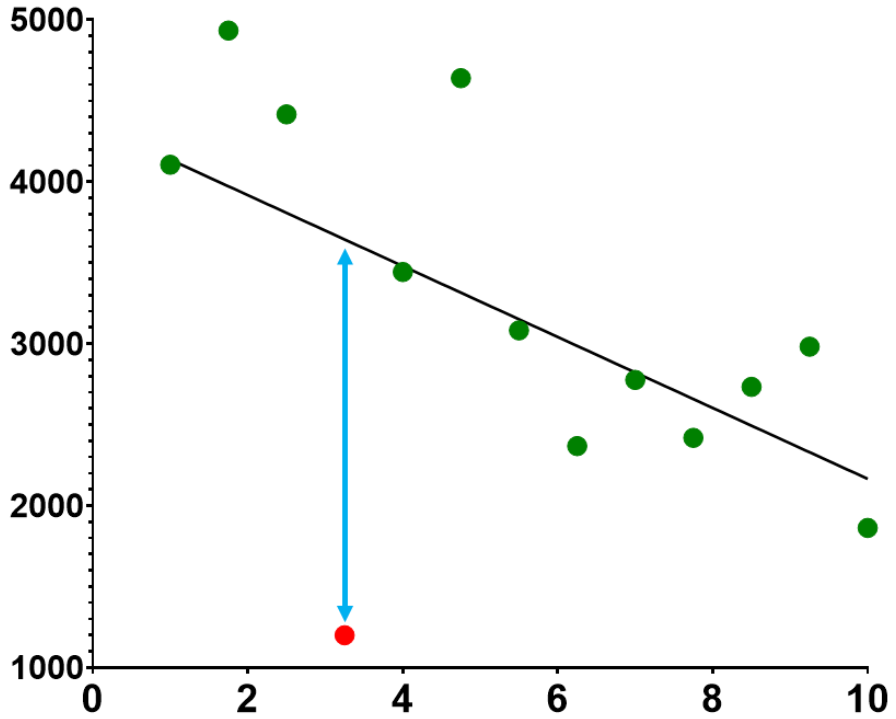
```
# A tibble: 1 × 8
  var1  var2    cor statistic        p conf.low conf.high method
  <chr> <chr> <dbl>     <dbl>    <dbl>    <dbl>     <dbl> <chr>
1 Depth Light -0.85      -5.27 0.000263   -0.953    -0.554 Pearson
```

Correlation: Other considerations

Outliers and High leverage points

- If have outlying points and/or you are interested in fitting the best line for your data overall, there are more considerations
- **Outliers**: the observed value for the point is very different from that predicted by the model.

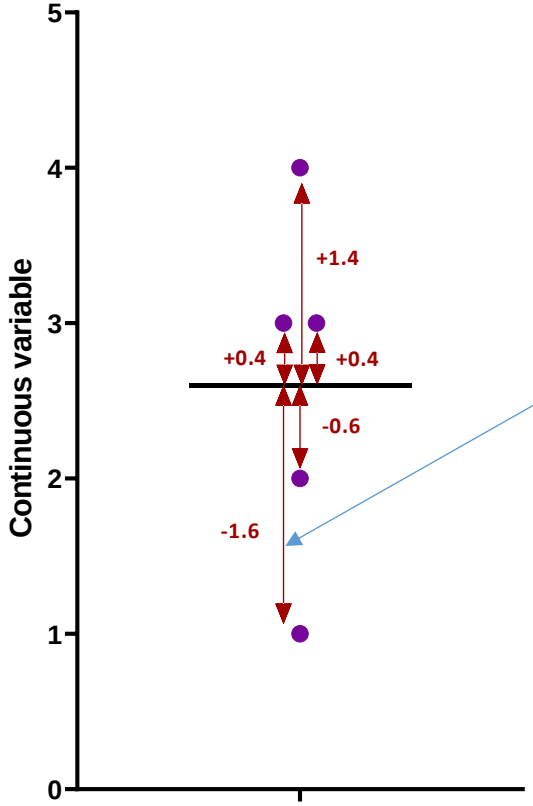




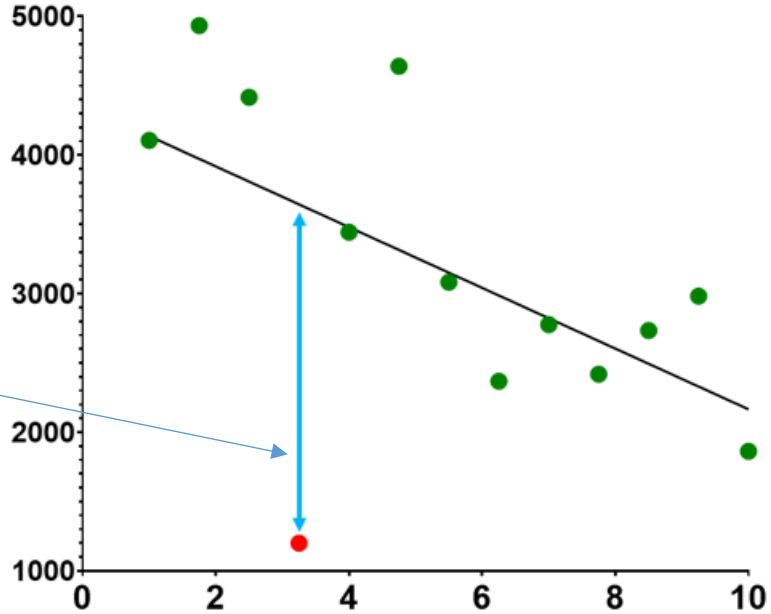
Correlation

Error a.k.a. Distance a.k.a. Residual

- **Outliers**: the observed value for the point is very different from that predicted by the model = big residual



Residual
=
Distance
=
Error

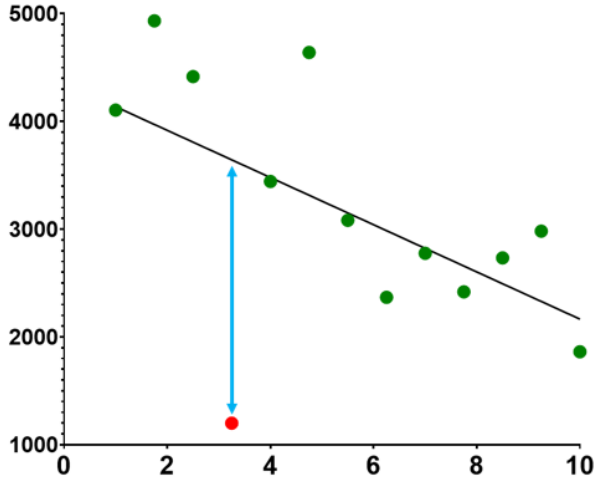


Correlation

Outliers and High leverage points

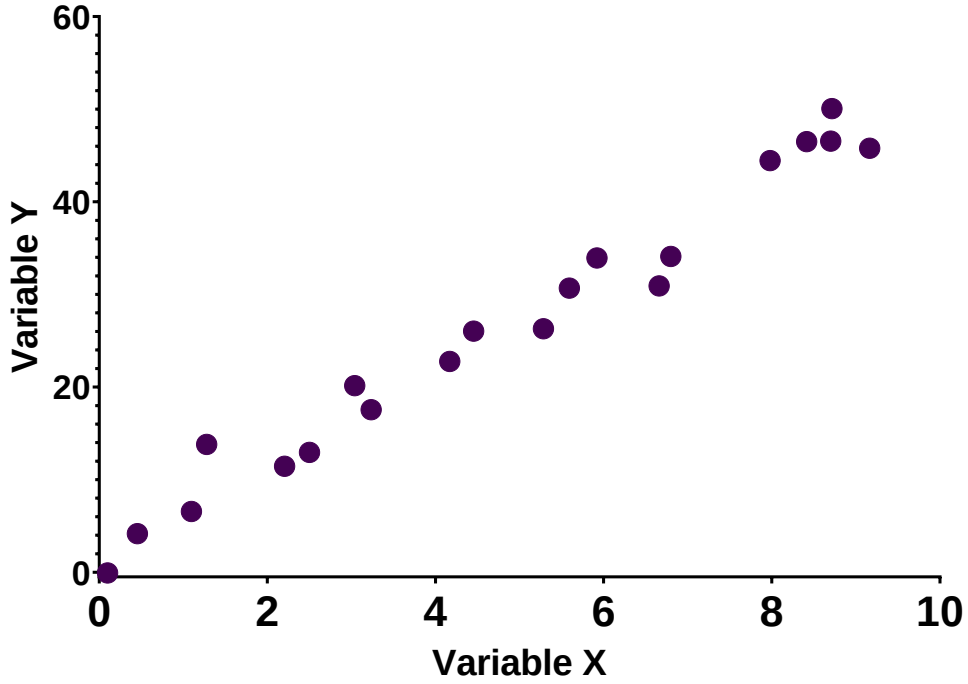
- **Leverage points**: A leverage point is defined as an observation that has **a value of x that is far away from the mean of x**. A point with high leverage has the potential to dramatically impact the model.

- **Outlier**: high discrepancy: **a point has an unusual y-value given its x-value**



Correlation

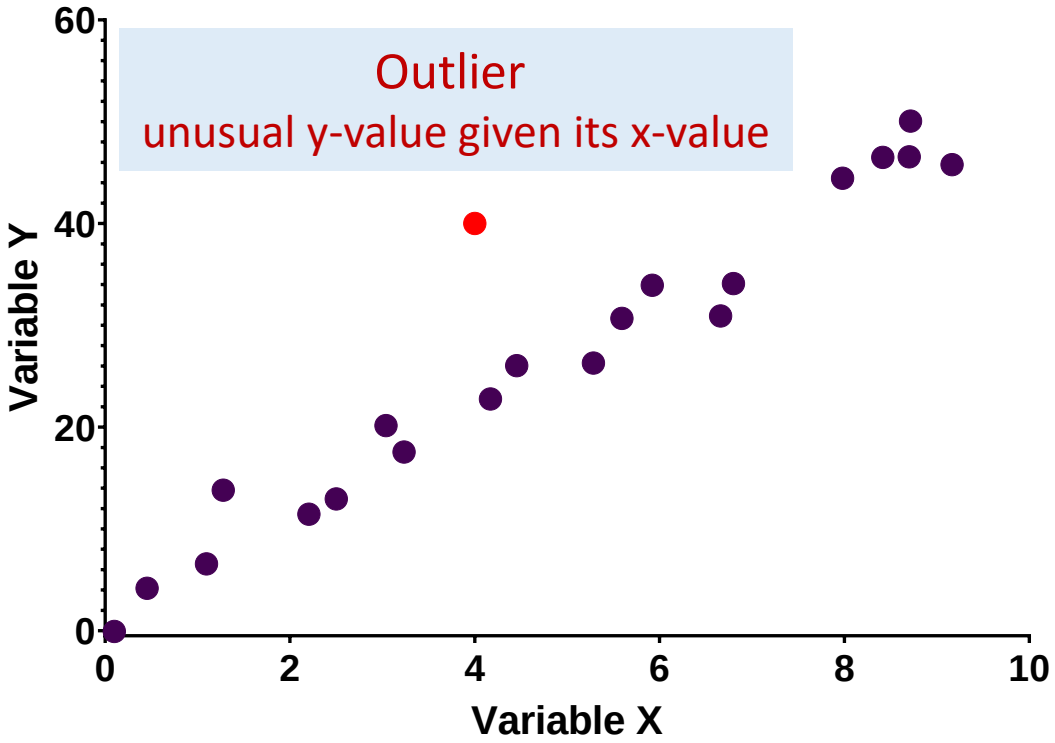
Outliers and High leverage points

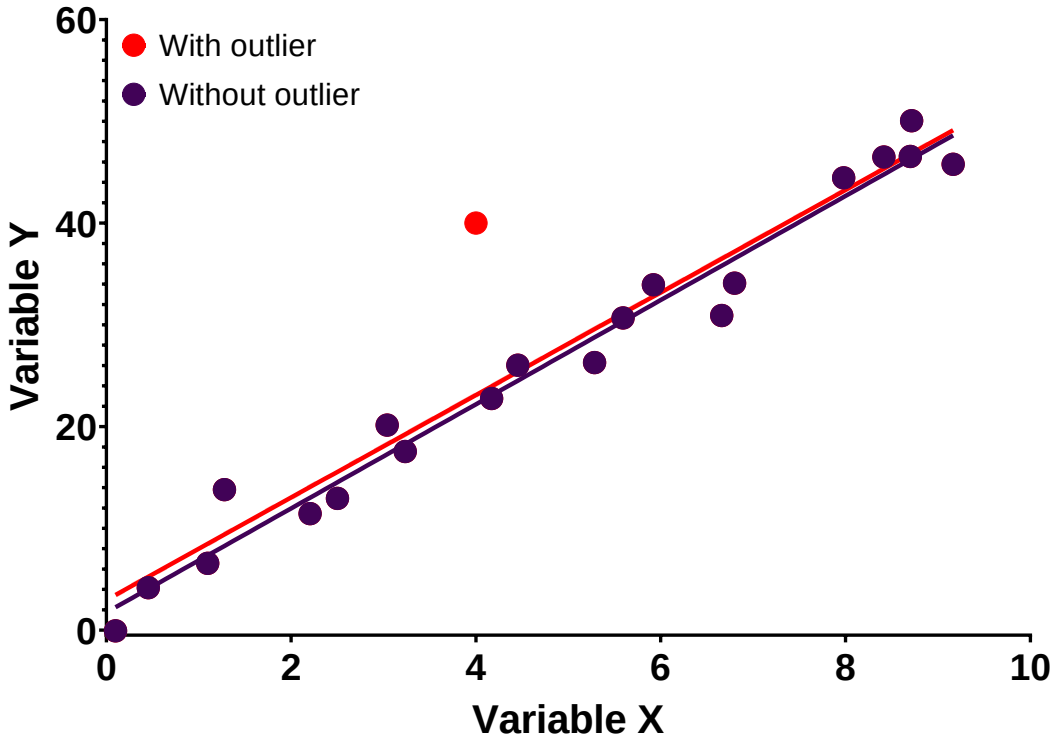


All good

Correlation

Outliers and High leverage points



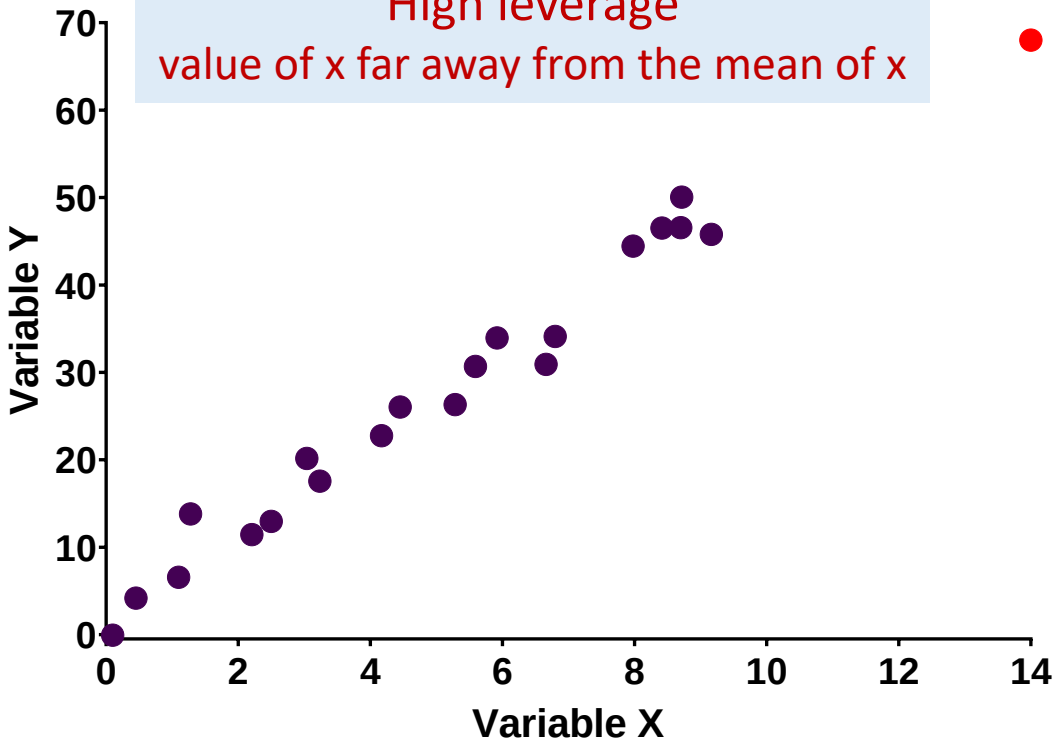


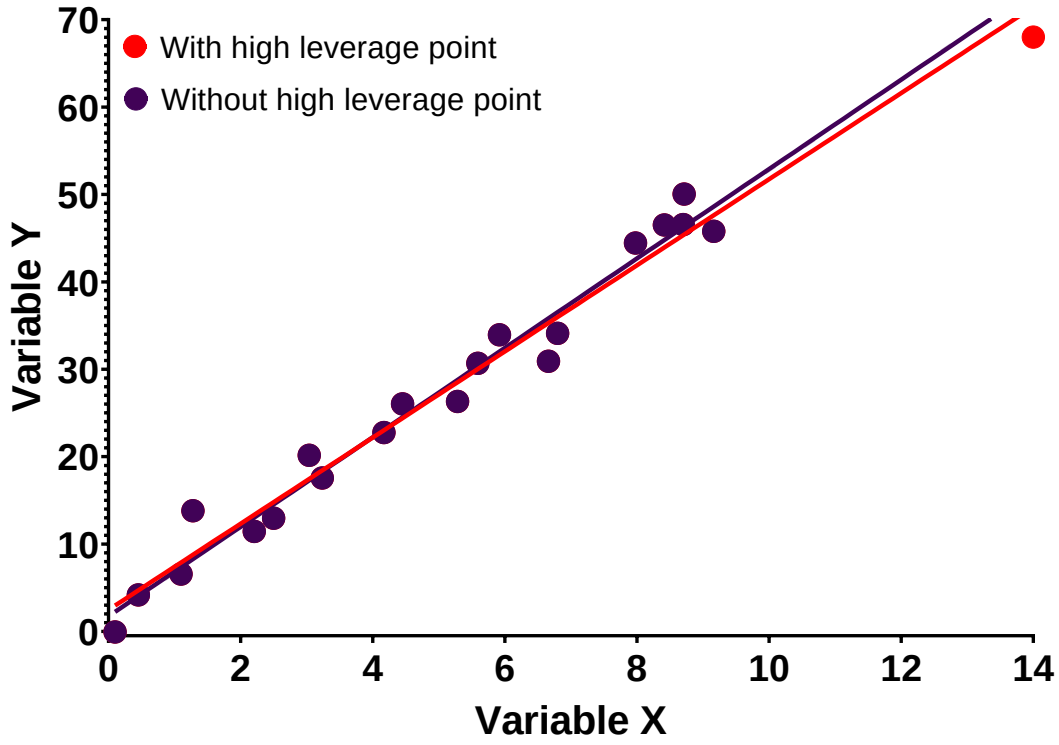
Outlier but not influential value

Correlation

Outliers and High leverage points

High leverage
value of x far away from the mean of x

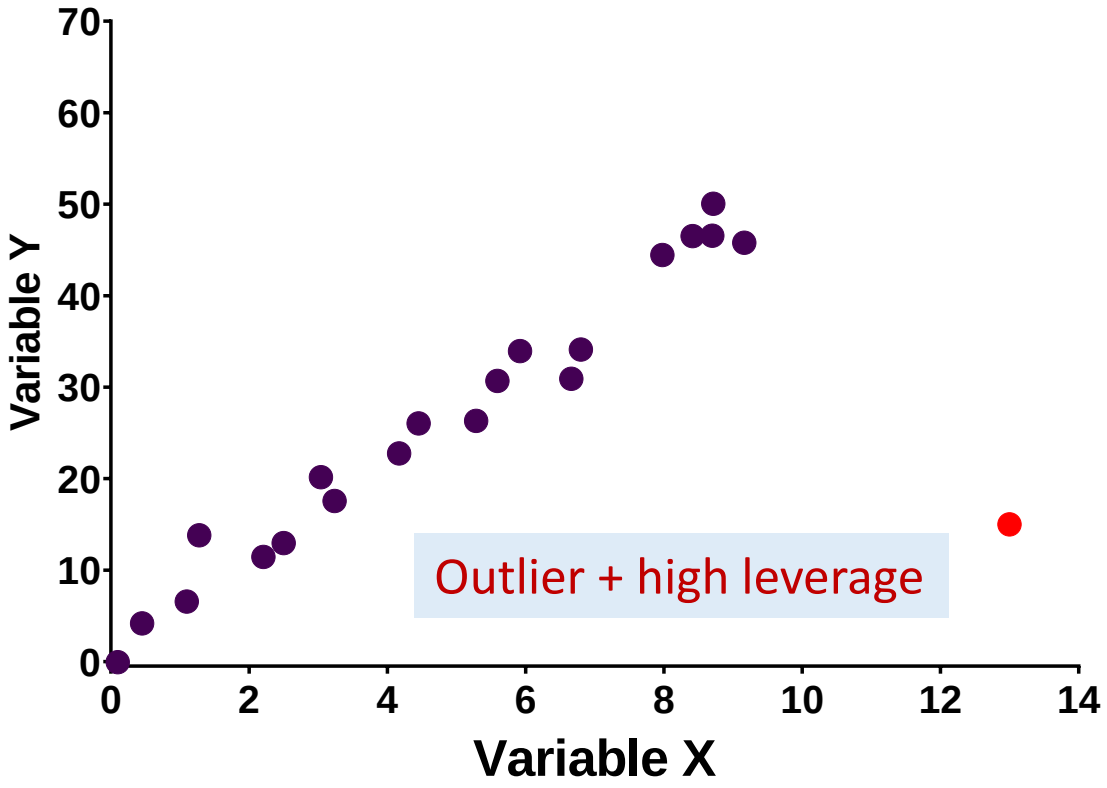


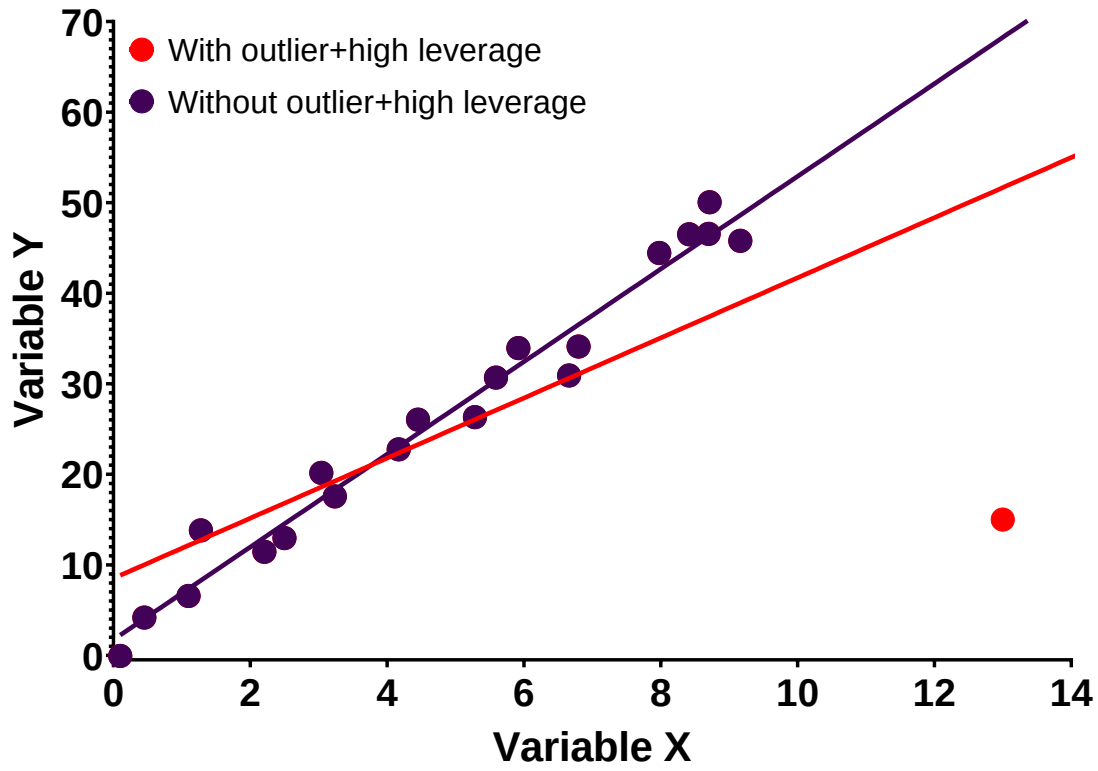


High leverage but not influential value

Correlation

Outliers and High leverage points





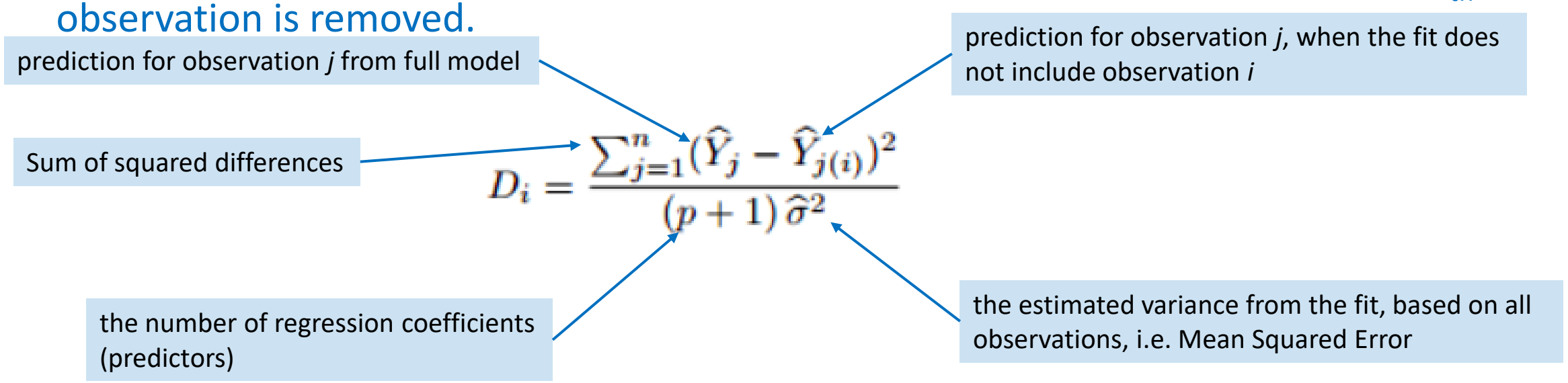
Outlier and High leverage: Influential value

Correlation

Outliers and High leverage points =

Influential observation

- One way to identify influential observations: the **Cook's distance**:
- Combination of each observation's **leverage** and **residual values**
- Higher leverage and residuals → higher Cook's distance = more likely an influential observation
  - Summarizes how much all the values in the regression model change when the i_{th} observation is removed.



Correlation

Outliers and High leverage points =

Influential observation

- **Consensus:** **Cook's distance D > 1** (0.5): likely to be an influential value
  - *"Observation which deviates so much from other observations as to arouse suspicion it was generated by a different mechanism" — Hawkins (1980)*

- Classic method to find influential points is to compare the fit of the model **with** and **without** the outlying point

https://lymilynn.medium.com/a-little-closer-to-cooks-distance-e8cc923a3250

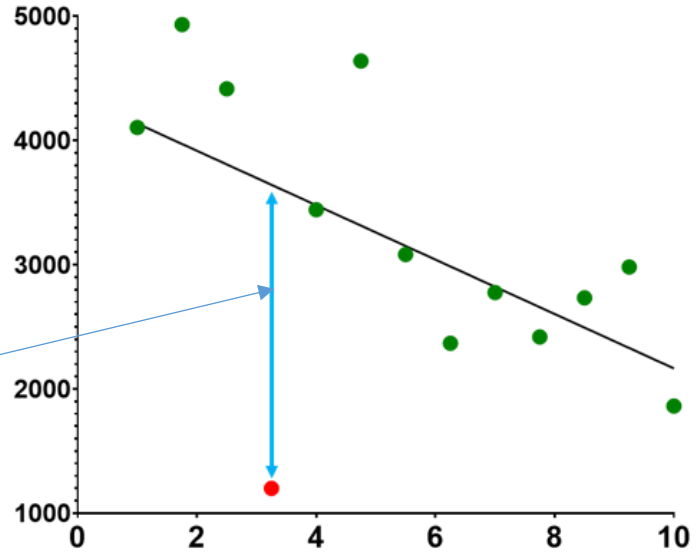
Correlation

Residuals to deal with dodgy values

- **Consensus:** **standardised residual > 3**: likely to be an outlier

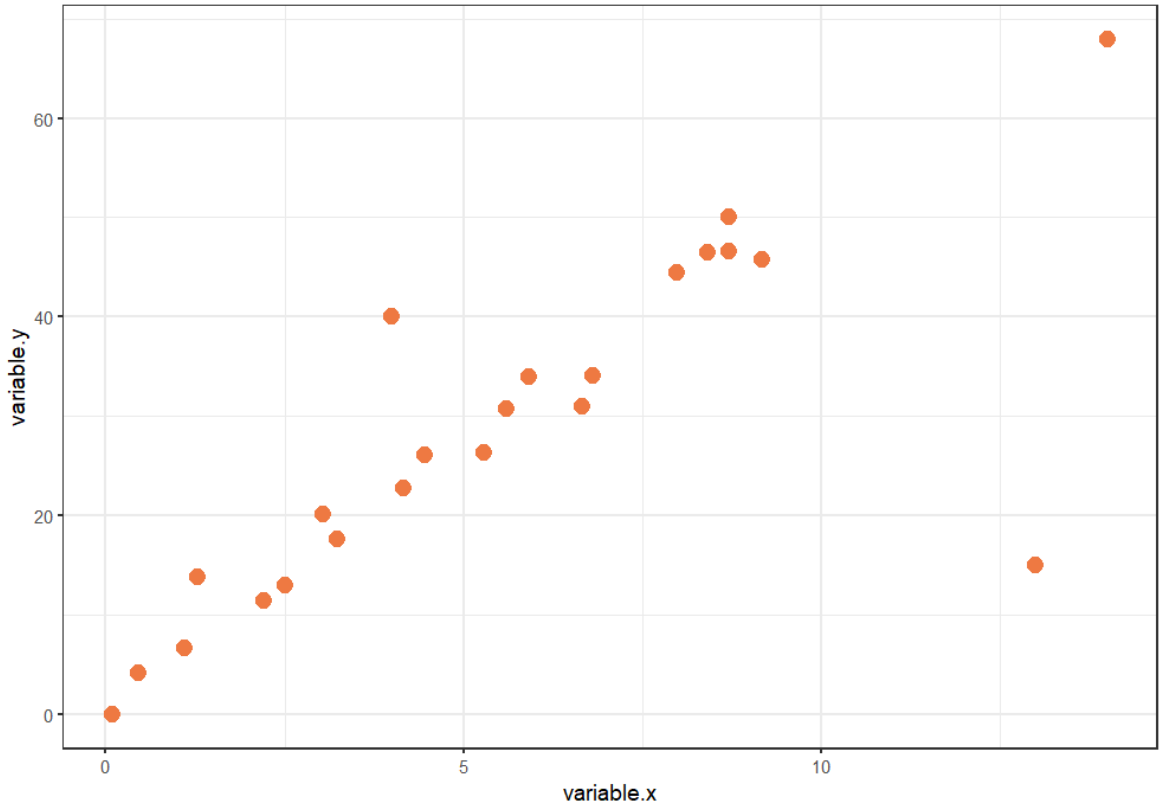
- Classical way to identify outliers is to look at the **residuals**
- A value with a big residual is poorly fitted by the model

Residual
=
Distance
=
Error



Correlation: correlation.csv

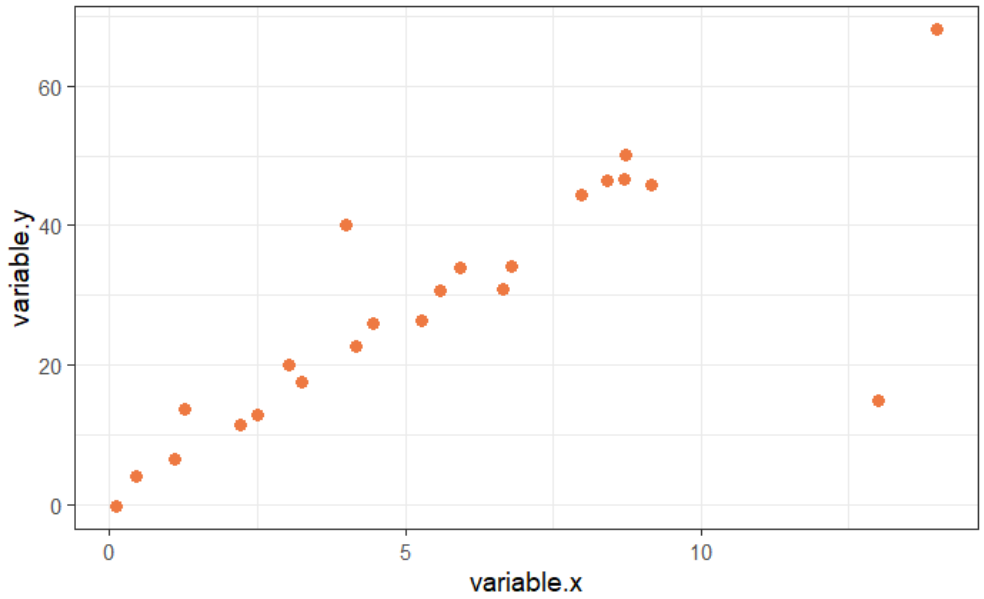
- **Questions**:
  - What is the nature and the strength of the relationship between X and Y?
  - Are there any dodgy points?



Correlation: correlation.csv

- **Question**: are there any dodgy points?

```
read_csv("correlation.csv") -> correlation

correlation %>%
  ggplot(aes(x=variable.x, y=variable.y)) +
  geom_point(size=5, colour="sienna2")
```

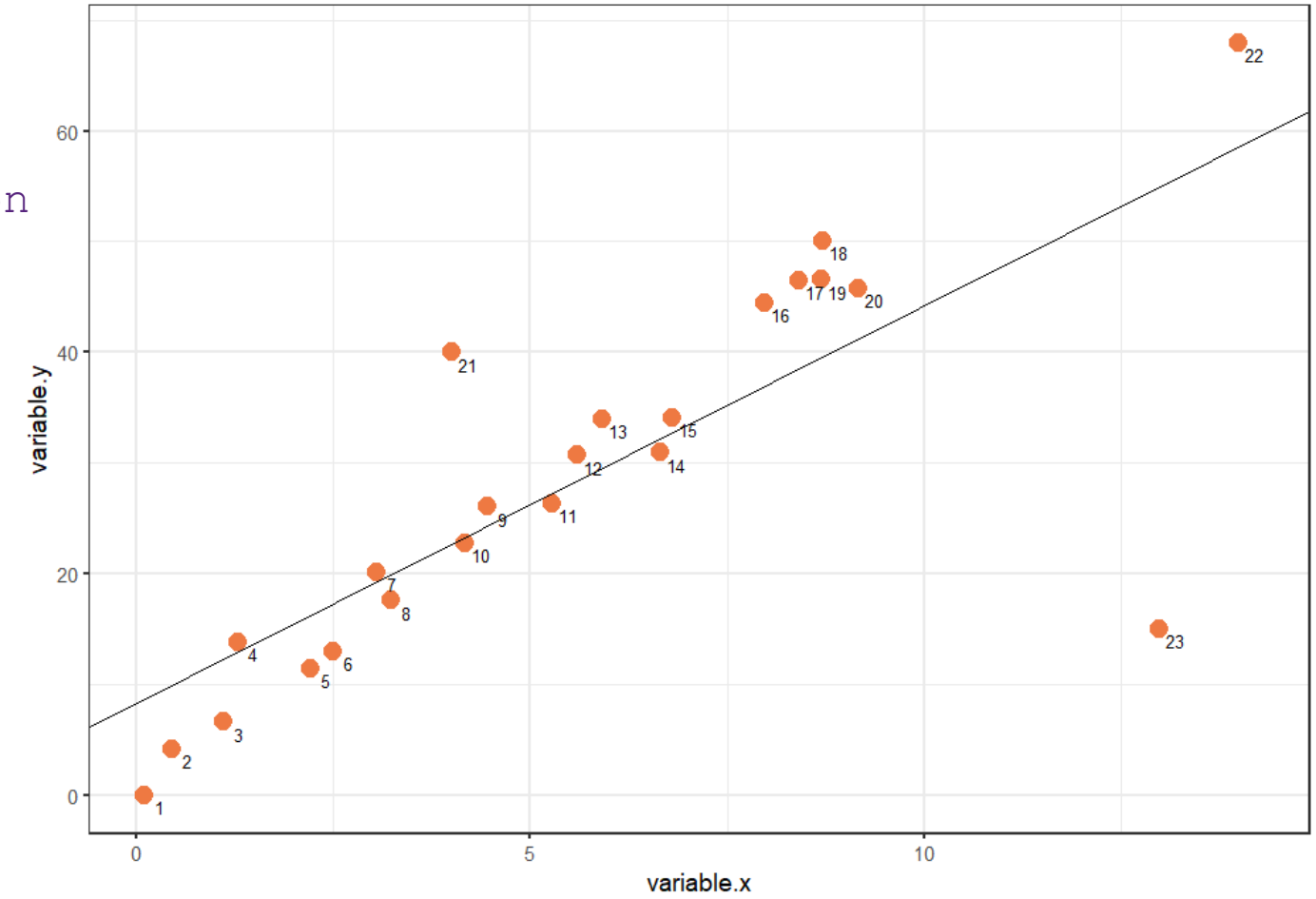


ID <dbl>	variable.x <dbl>	variable.y <dbl>
1	0.10000	-0.0716
2	0.45401	4.1673
3	1.09765	6.5703
4	1.27936	13.8150
5	2.20611	11.4501
6	2.50064	12.9554
7	3.04030	20.1575
8	3.23583	17.5633
9	4.45308	26.0317
10	4.16990	22.7573

1-10 of 23 rows

Correlation: correlation.csv

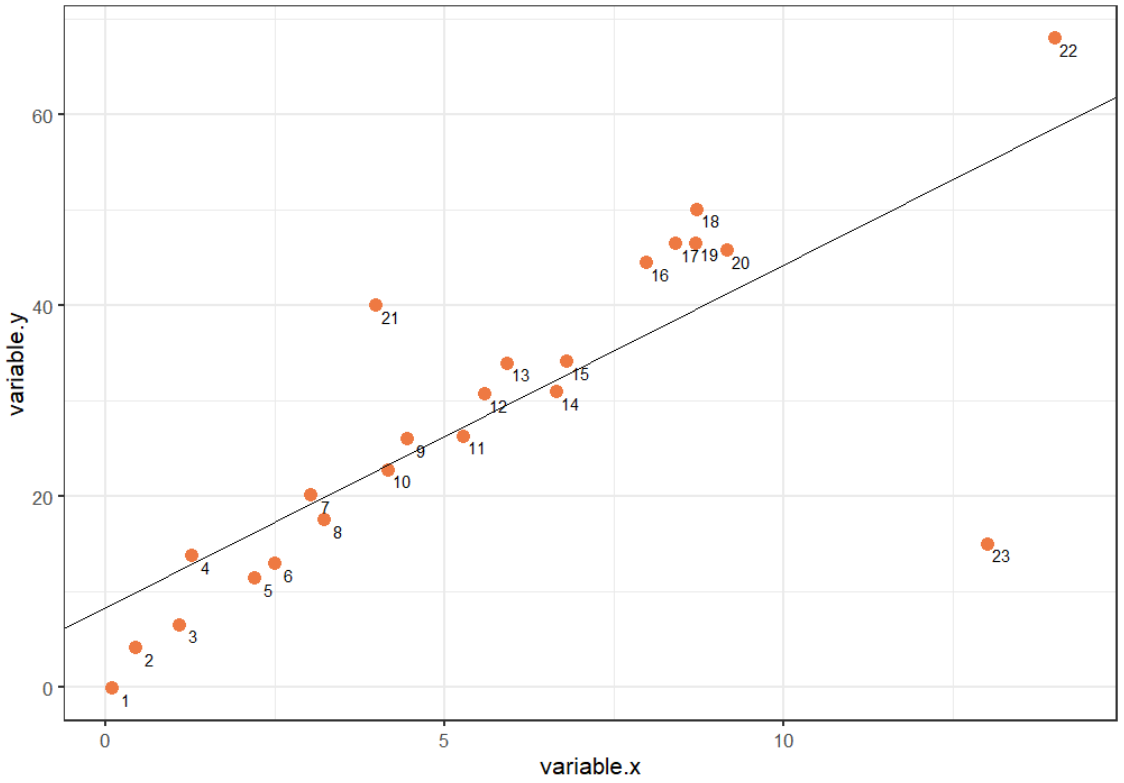
```
lm(variable.y ~ variable.x,
   data=correlation)-> fit.correlation
coefficients(fit.correlation) ->
   coef.correlation
```




```
correlation %>%
  ggplot(aes(x=variable.x, y=variable.y, label = ID)) +
  geom_point(size=3, colour="sienna2") +
  geom_abline(intercept = coef.correlation[1], slope = coef.correlation[2])+
  geom_text(vjust = 1.3, nudge_x = 0.2)
```

Correlation: correlation.csv

```
correlation %>%
  ggplot(aes(x=variable.x, y=variable.y, label = ID)) +
  geom_point(size=5, colour="sienna2") +
  geom_abline(intercept = coef.correlation[1], slope = coef.correlation[2])+
  geom_text(vjust = 1.3, nudge_x = 0.2)
```



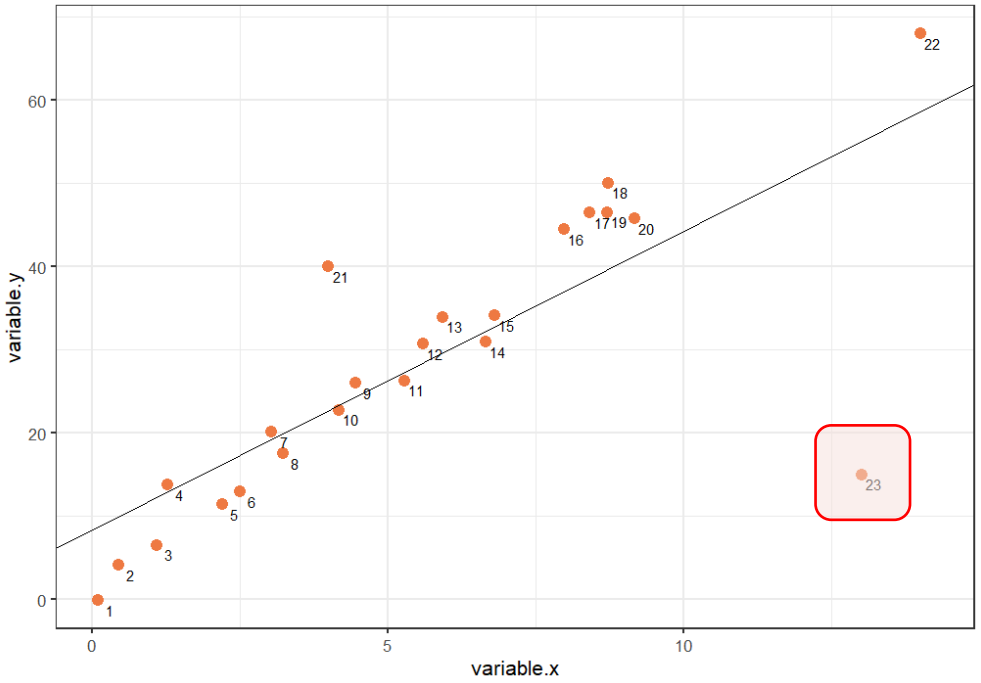
How good is the fit?

```
summary(fit.correlation)
```

Correlation?

```
correlation %>%
    cor_test(variable.x, variable.y)
```

Correlation: correlation.csv



How good is the fit?

```
summary(fit.correlation)
```

Line of best fit: Y=8.38 + 3.59*X

```
Call:
lm(formula = variable.y ~ variable.x, data = correlation)

Residuals:
    Min      1Q  Median      3Q     Max
-40.034  -3.414   0.867   5.723  17.265

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.3798     4.1195   2.034   0.0548 .
variable.x    3.5888     0.6225   5.765 1.01e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.93 on 21 degrees of freedom
Multiple R-squared:  0.6128,    Adjusted R-squared:  0.5943
F-statistic: 33.23 on 1 and 21 DF,  p-value: 1.01e-05
```

Correlation?

```
correlation %>%
   cor_test(variable.x, variable.y)
```

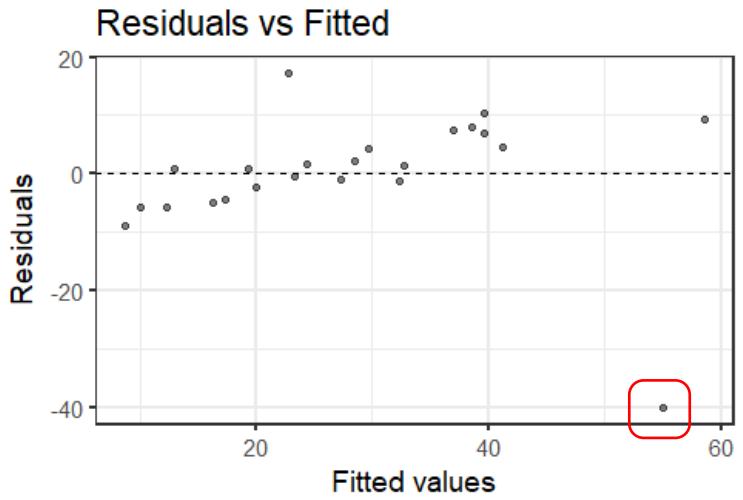
var1 <chr>	var2 <chr>	cor <dbl>	statistic <dbl>	p <dbl>	conf.low <dbl>	conf.high <dbl>	method <chr>
variable.x	variable.y	0.78	5.764871	1.01e-05	0.5471597	0.9034793	Pearson

Correlation: correlation.csv

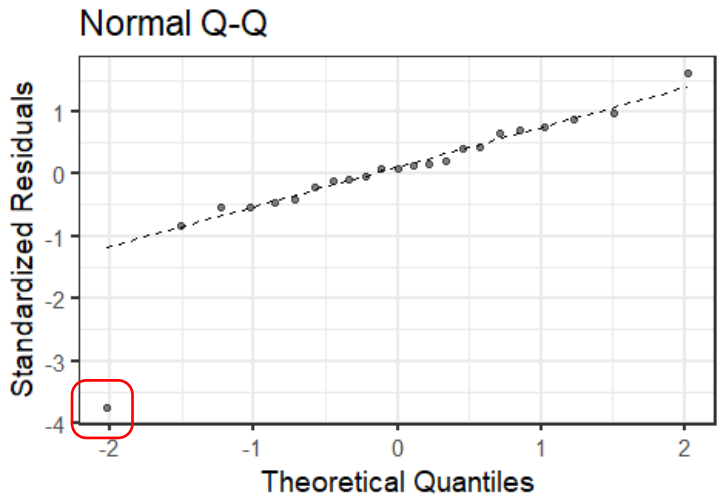
Assumptions, outliers and influential cases

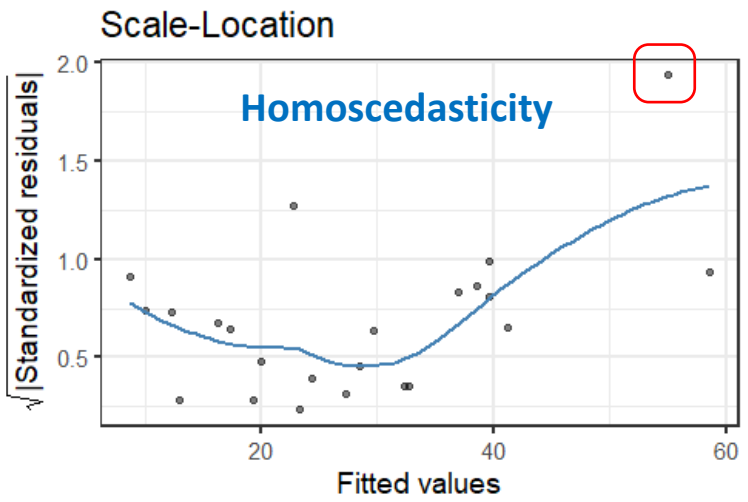
```
gglm(fit.correlation, theme = theme_bw(base_size = 16))
```

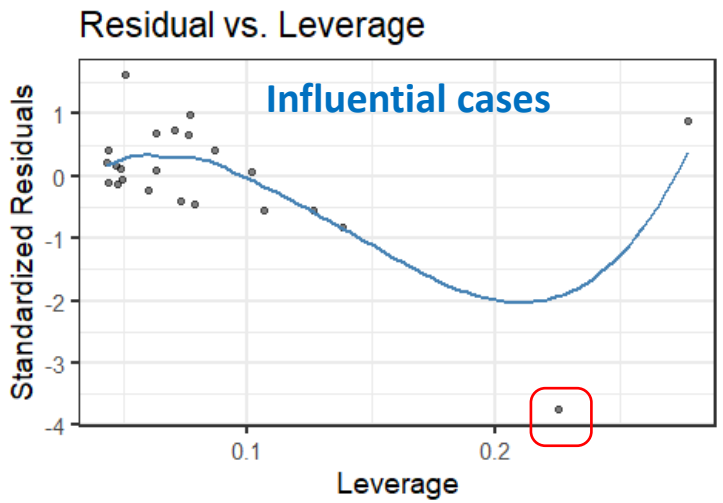
Linearity, homoscedasticity and outlier



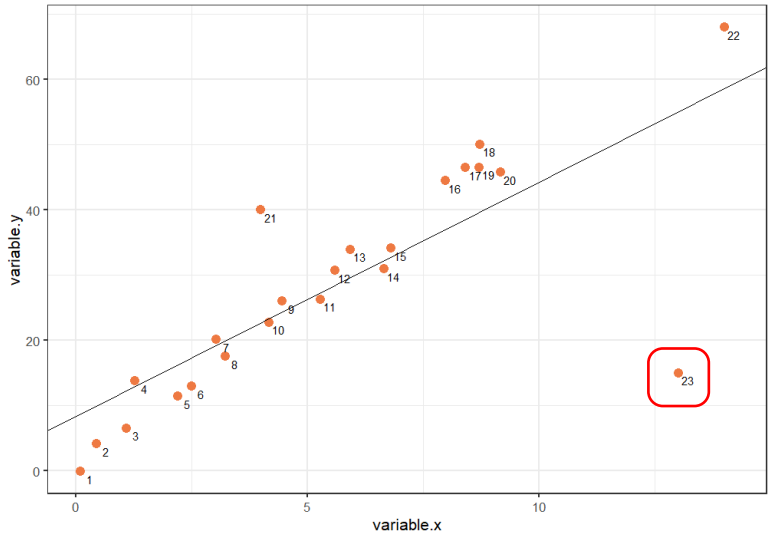
Normality and outlier








```
# gglm package #
```

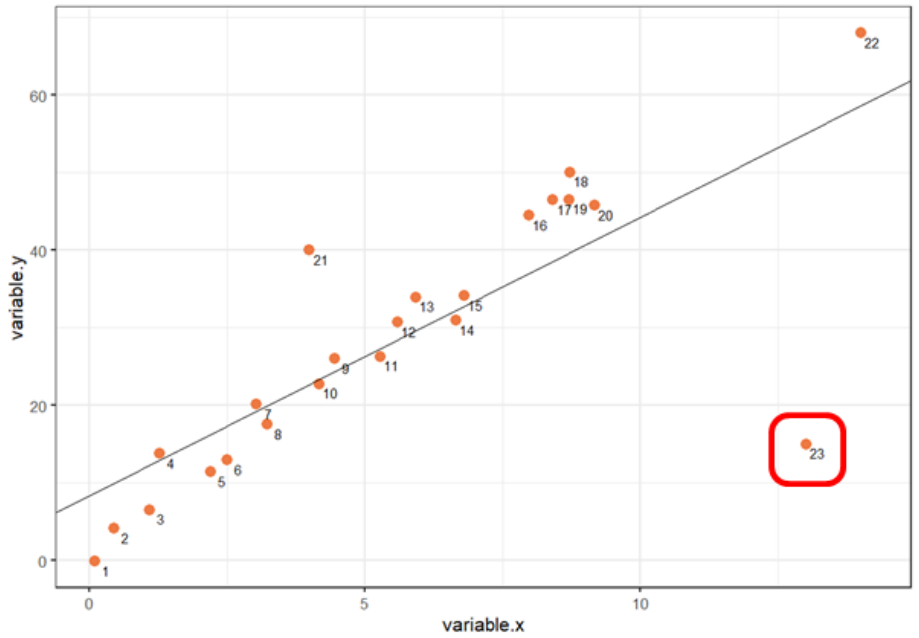


Correlation: correlation.csv

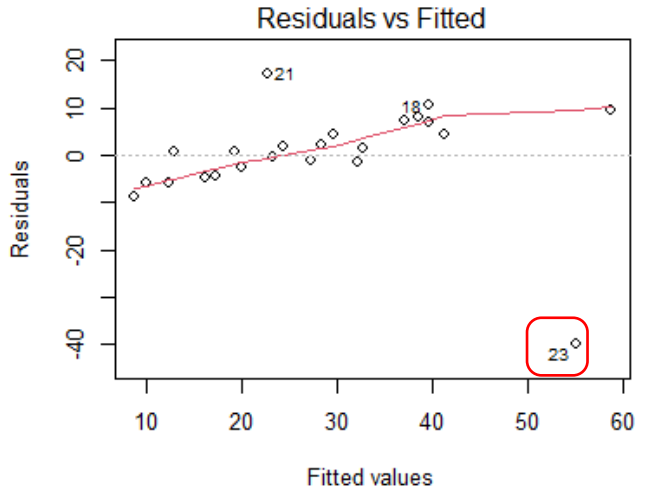
Assumptions, outliers and influential cases

```
par(mfrow=c(2,2))
plot(fit.correlation)
```

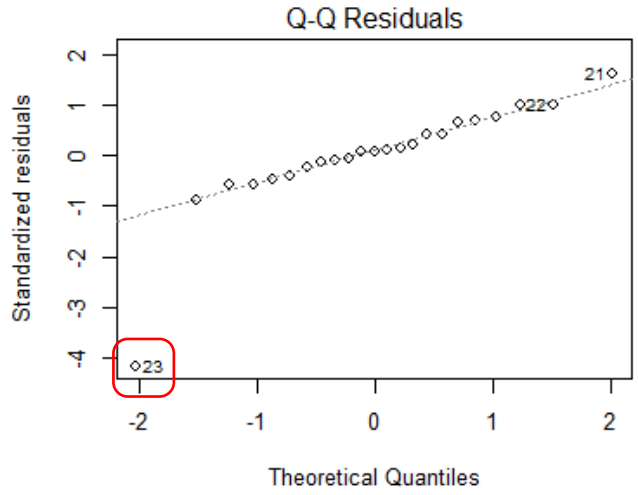
Core R



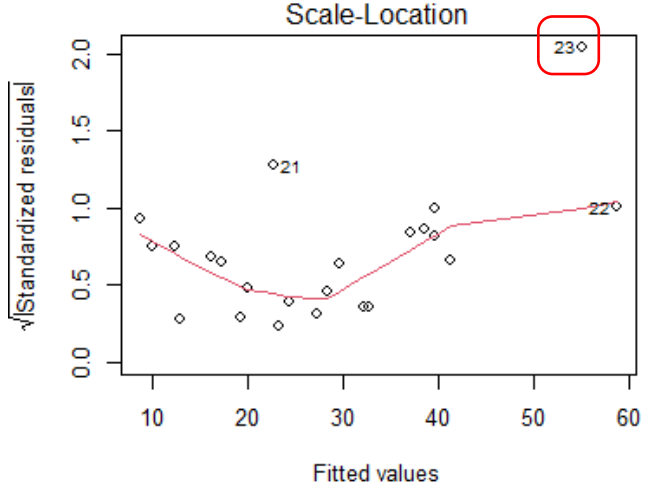
Linearity, homoscedasticity and outlier



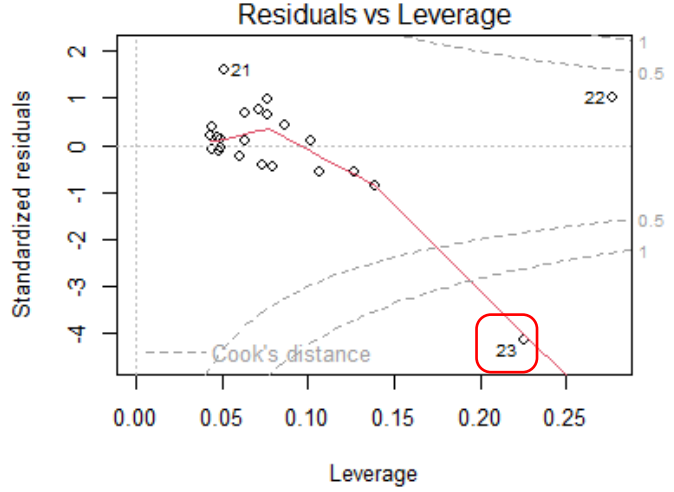
Normality and outlier



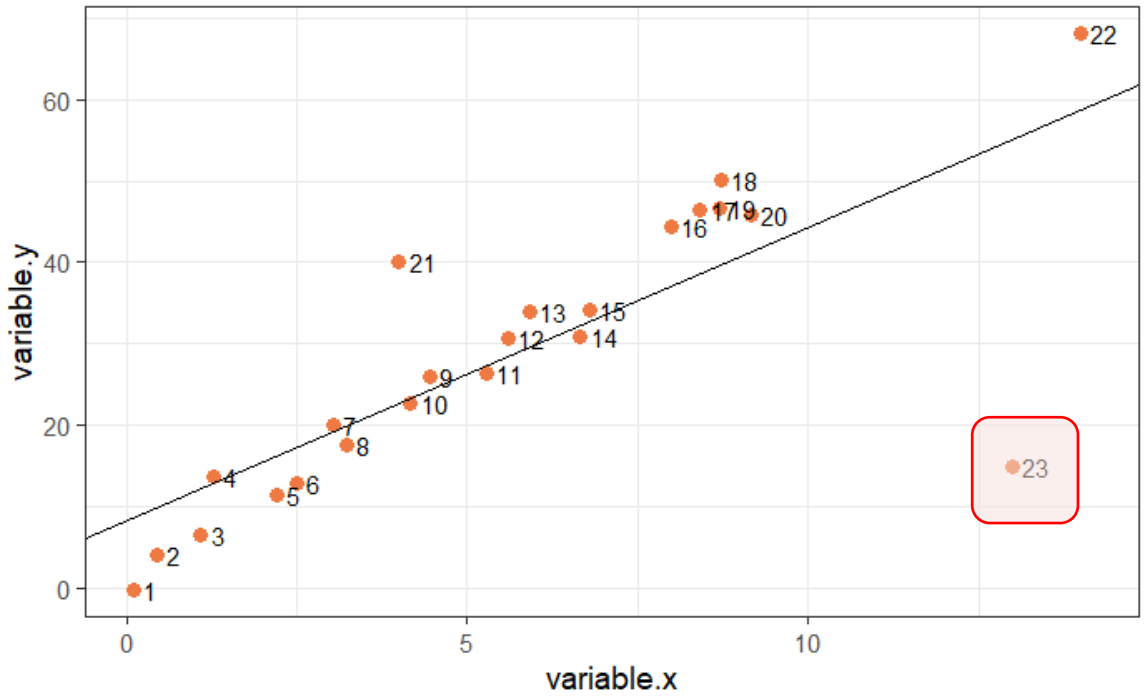
Homoscedasticity



Influential cases



Correlation: correlation.csv



Have a go: Remove ID 23, then re-run the model and plot the graph again.
Hint: you may need `cooks.distance()` `rstandard()` and `filter()`

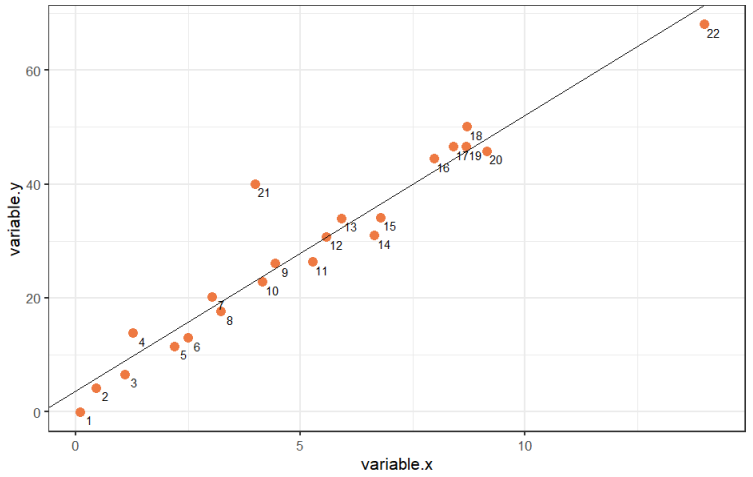
Correlation: correlation.csv

```
cooks.distance(fit.correlation)-> cook
rstandard(fit.correlation)-> residual

correlation %>%
  add_column(cook) %>%
  add_column(residual) -> correlation
```

Filter

	ID	variable.x	variable.y	cook	residual
1	23	13.00000	15.0000	2.517207e+00	-4.16060845
2	22	14.00000	68.0000	1.950580e-01	1.00880694
3	21	4.00000	40.0000	7.044938e-02	1.62105018
4	1	0.10000	-0.0716	6.057002e-02	-0.86823862
5	18	8.71607	50.0568	4.073005e-02	0.98975608

```
correlation %>%
  filter(cook<1) -> correlation.23

lm(variable.y ~ variable.x, correlation.23)-> fit.correlation.23
summary(fit.correlation.23)
```



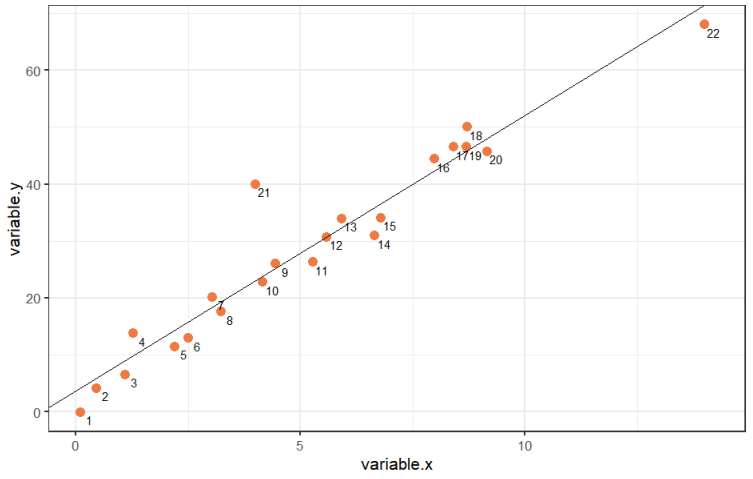

```
Call:
lm(formula = variable.y ~ variable.x, data = correlation.23)

Residuals:
    Min      1Q  Median      3Q     Max
-5.049 -2.784 -1.446   1.679 16.915

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.7103     1.8338   2.023   0.0566 .
variable.x     4.8436     0.2971  16.303 5.13e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.695 on 20 degrees of freedom
Multiple R-squared:  0.93,      Adjusted R-squared:  0.9265
F-statistic: 265.8 on 1 and 20 DF,  p-value: 5.13e-13
```

From $r^2 = 0.6128$

Correlation: correlation.csv

```
cooks.distance(fit.correlation)-> cook
rstandard(fit.correlation)-> residual

correlation %>%
  add_column(cook) %>%
  add_column(residual) -> correlation

correlation %>%
  filter(cook<1) -> correlation.23

lm(variable.y ~ variable.x, correlation.23)-> fit.correlation.23
summary(fit.correlation.23)
```

ID <dbl>	variable.x <dbl>	variable.y <dbl>	cook <dbl>	residual <dbl>
23	13.00000	15.0000	2.517207e+00	-4.16060845
22	14.00000	68.0000	1.950580e-01	1.00880694
21	4.00000	40.0000	7.044938e-02	1.62105018
1	0.10000	-0.0716	6.057002e-02	-0.86823862
18	8.71607	50.0568	4.073005e-02	0.98975608



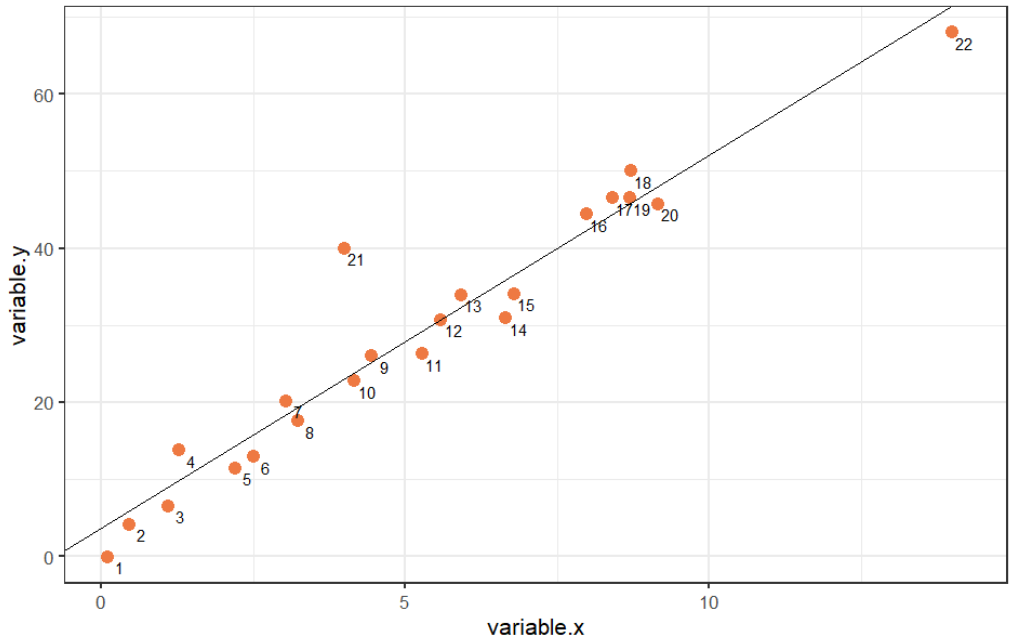

```
Call:
lm(formula = variable.y ~ variable.x, data = correlation.23)

Residuals:
   Min     1Q Median     3Q    Max
-5.049 -2.784 -1.446  1.679 16.915

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.7103     1.8338   2.023   0.0566 .
variable.x    4.8436     0.2971  16.303 5.13e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.695 on 20 degrees of freedom
Multiple R-squared:   0.93,     Adjusted R-squared:  0.9265
F-statistic: 265.8 on 1 and 20 DF,  p-value: 5.13e-13
```

From $r^2 = 0.6128$

Correlation: correlation.csv

```
coefficients(fit.correlation.23) -> coef.correlation.23
```

```
correlation.23 %>%
  ggplot(aes(x=variable.x, y=variable.y, label = ID)) +
  geom_point(size=, colour="sienna2") +
  geom_abline(intercept = coef.correlation.23[1], slope = coef.correlation.23[2])+
  geom_text(vjust = 1.3, nudge_x = 0.2)
```



From r = 0.78

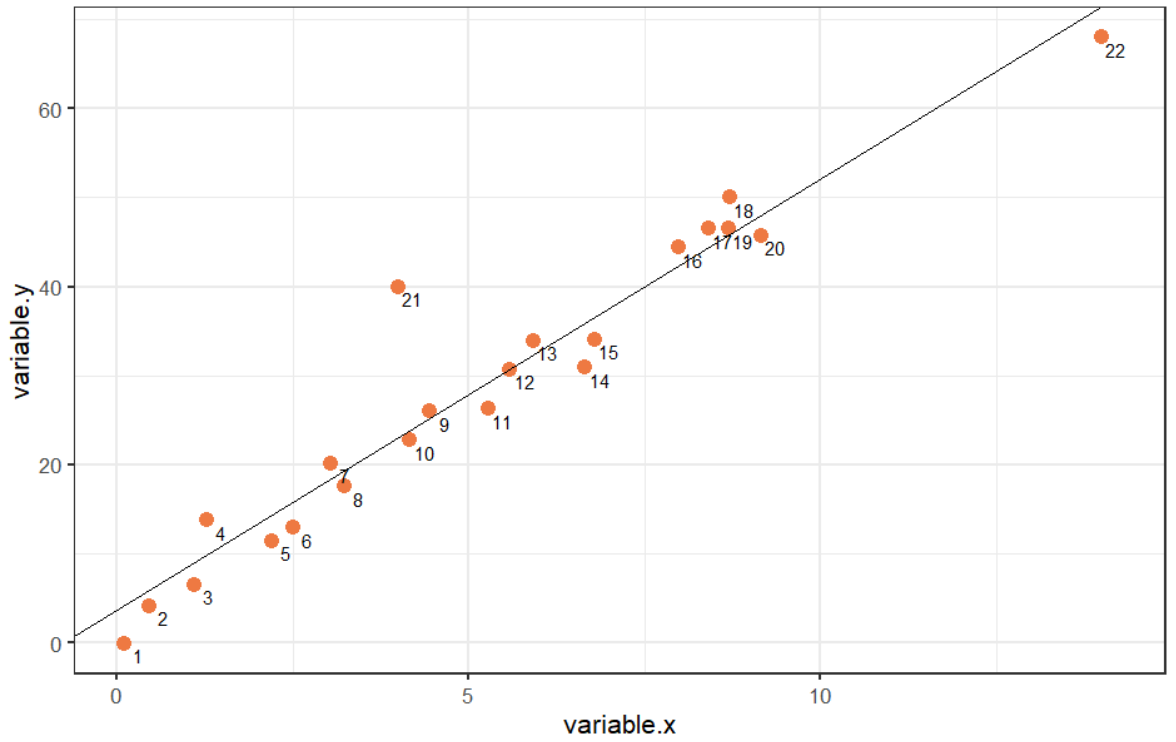
```
correlation.23 %>%
  cor_test(variable.x, variable.y)
```

var1	var2	cor	statistic	p	conf.low	conf.high	method
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
variable.x	variable.y	0.96	16.3	5.13e-13	0.915	0.985	Pearson

Correlation: correlation.csv

Let's add confidence bands to the graph

- Confidence interval → how well we have determined a particular parameter
  e.g. mean or coefficient of regression

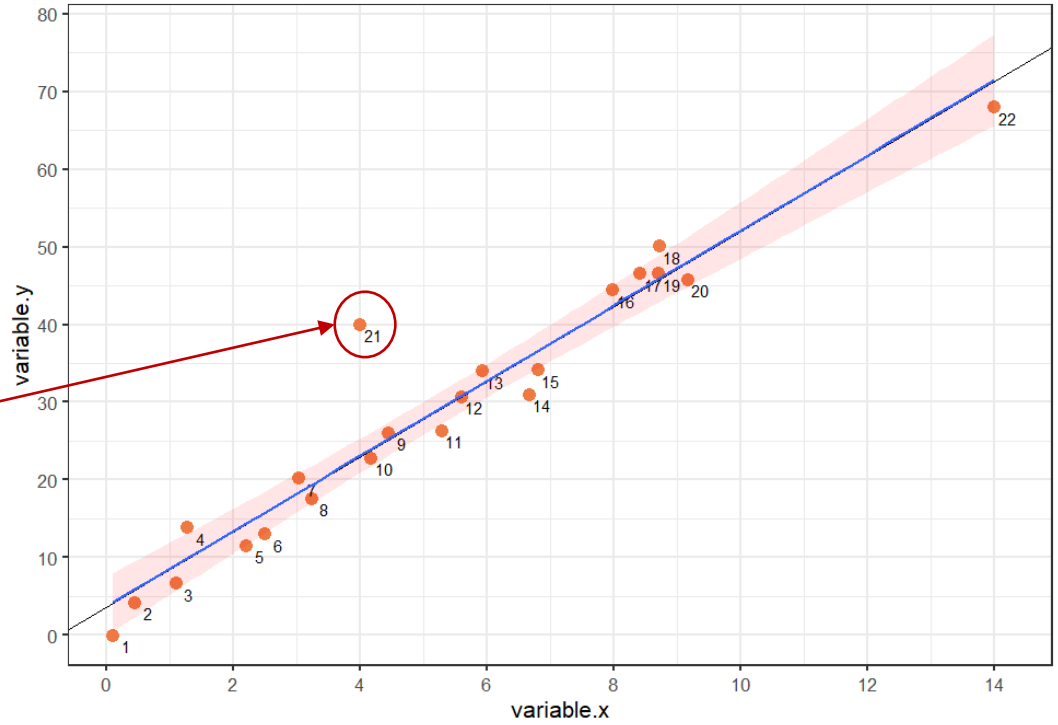


Correlation: correlation.csv

Let's add confidence bands to the graph

```
correlation.23 %>%
  ggplot(aes(x=variable.x, y=variable.y, label = ID)) +
  geom_point(size=4, colour="sienna2") +
  geom_abline(intercept = coef.correlation.23[1], slope = coef.correlation.23[2])+
  geom_text(vjust = 1.3, nudge_x = 0.2)+
  geom_smooth(method=lm, fill="red", alpha=0.1)+
  scale_x_continuous(breaks=seq(from=0, by=2, to=20))+
  scale_y_continuous(breaks=seq(from=0, by=10, to=80))
```

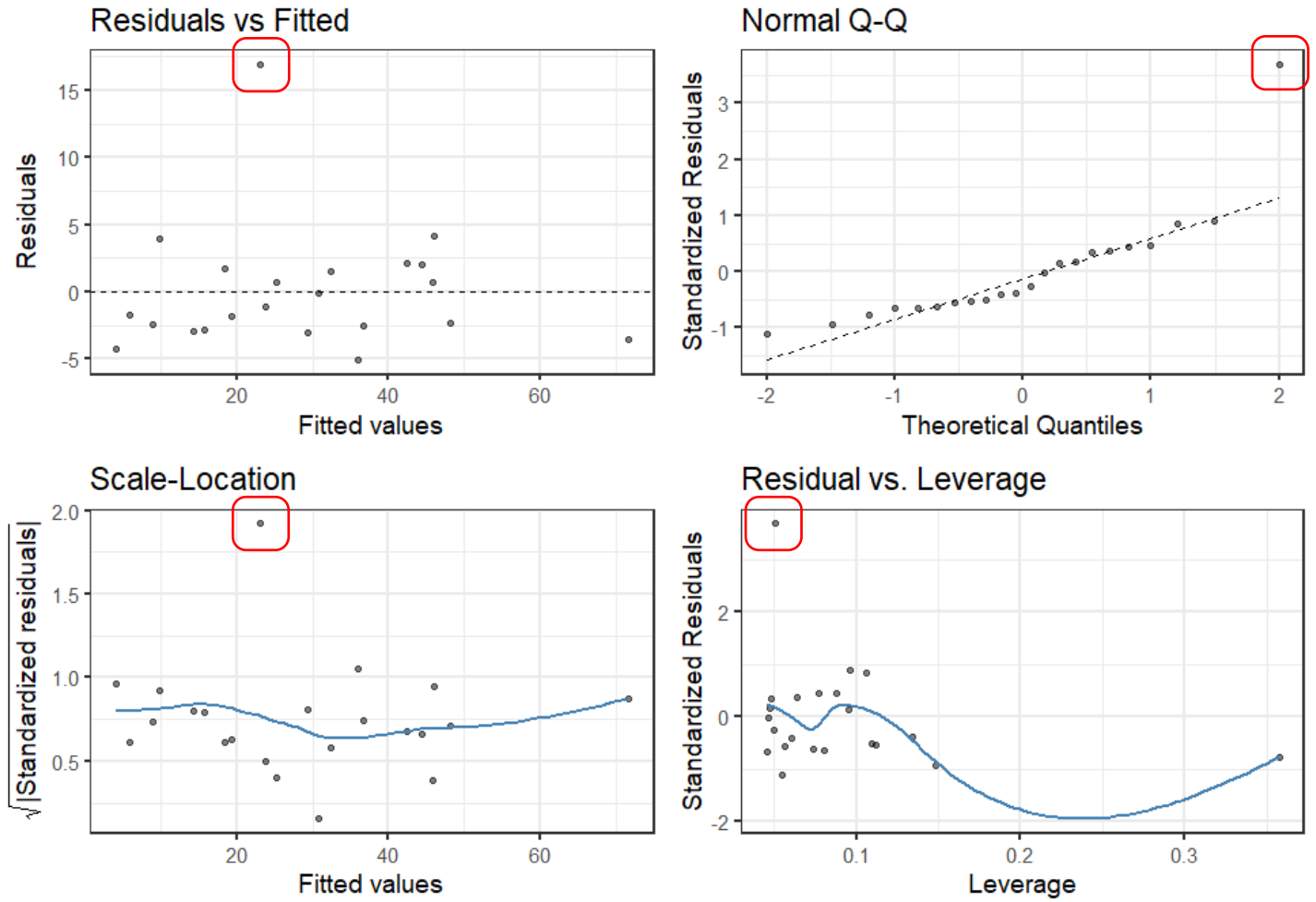
21 falls well outside confidence bands



Correlation: correlation.csv

Let's take care of ID 21

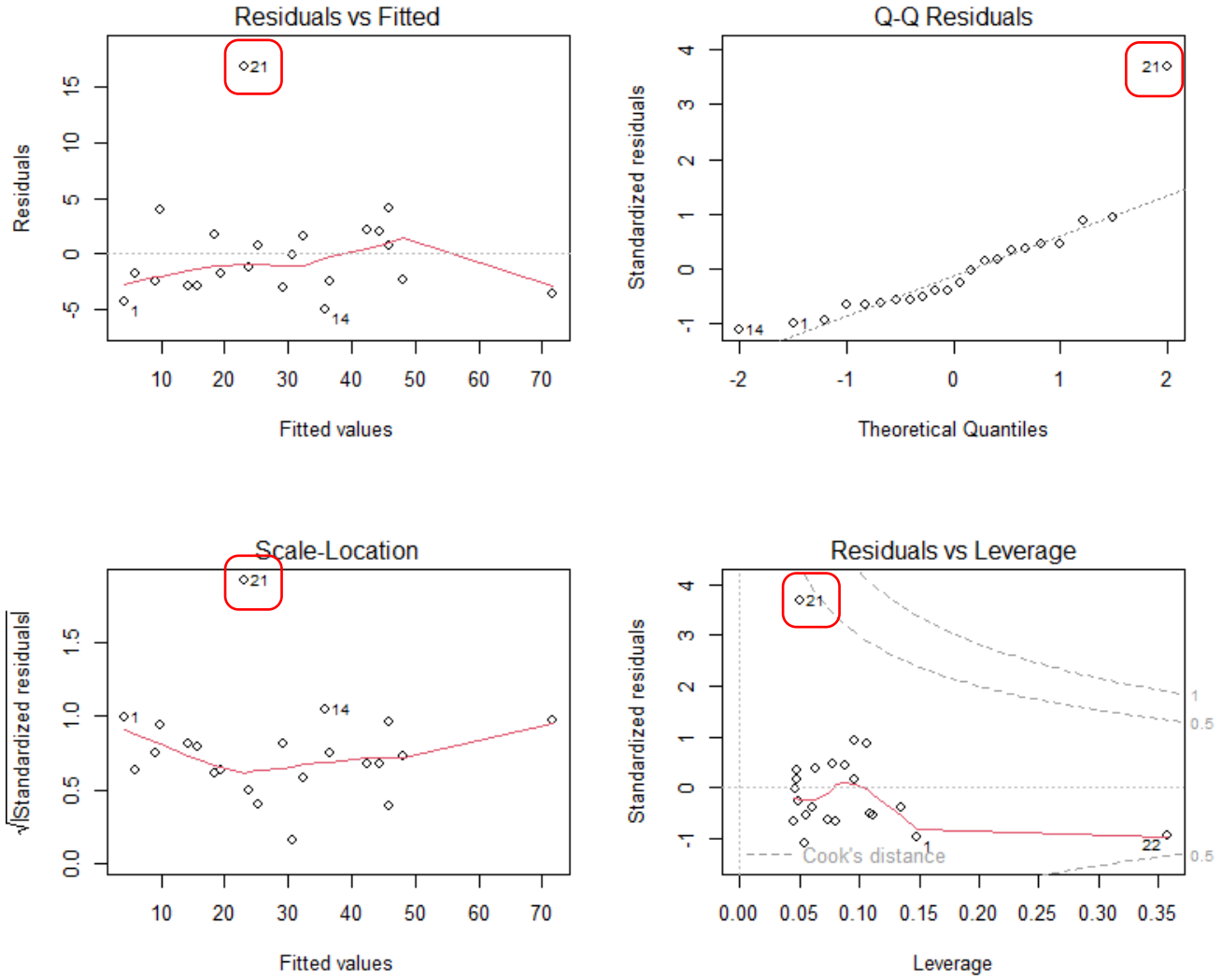
```
gglm(fit.correlation.23, theme = theme_bw(base_size = 16))
```



Correlation: correlation.csv

Let's take care of ID 21

```
par(mfrow=c(2,2))
plot(fit.correlation.23)
```



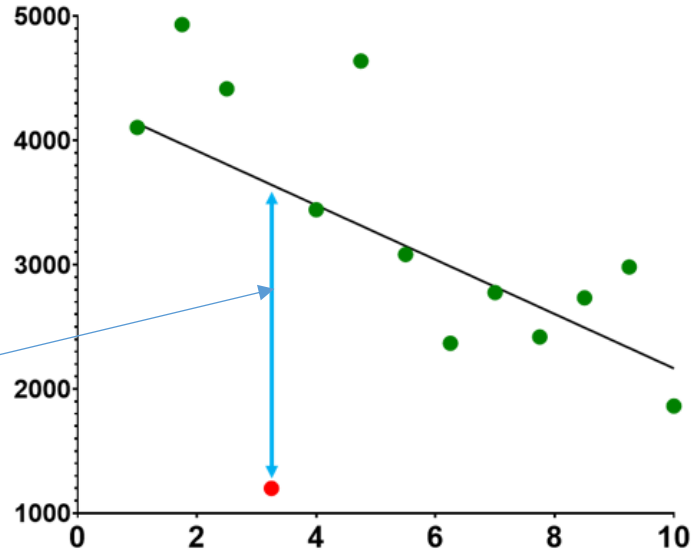
Correlation

Residuals to deal with dodgy values

- **Consensus:** **standardised residual > 3**: likely to be an outlier

- Classical way to identify outliers is to look at the **residuals**
- A value with a big residual is poorly fitted by the model
- Residuals can be positive or negative – look at absolute

Residual
=
Distance
=
Error



Correlation: correlation.csv

Let's take care of ID 21

```
rstandard(fit.correlation.23) -> residual23
cooks.distance(fit.correlation.23) -> cook23

correlation.23 %>%
  select(-cook, -residual) %>%
  add_column(cook23) %>%
  add_column(residual23) %>%
  filter(abs(residual23) < 3) -> correlation.23.21
```

	ID	variable.x	variable.y	residual23	cook23
1	21	4.00000	40.0000	3.69795678	3.670619e-01
2	22	14.00000	68.0000	-0.93557418	2.435359e-01
3	1	0.10000	-0.0716	-0.98462563	8.449122e-02
4	4	1.27936	13.8150	0.88030544	4.599235e-02
5	18	8.71607	50.0568	0.92478700	4.528413e-02
6	14	6.66066	30.9228	1.10583670	3.513040e-02

Correlation: correlation.csv

Let's remove ID 21 as well

```
lm(variable.y ~ variable.x, correlation.23.21) -> fit.correlation.23.21
summary(fit.correlation.23.21)

correlation.23.21 %>%
  cor_test(variable.x, variable.y)
```

var1	var2	cor	statistic	p	conf.low	conf.high	method
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
variable.x	variable.y	0.99	28.7	4.23e-17	0.972	0.995	Pearson

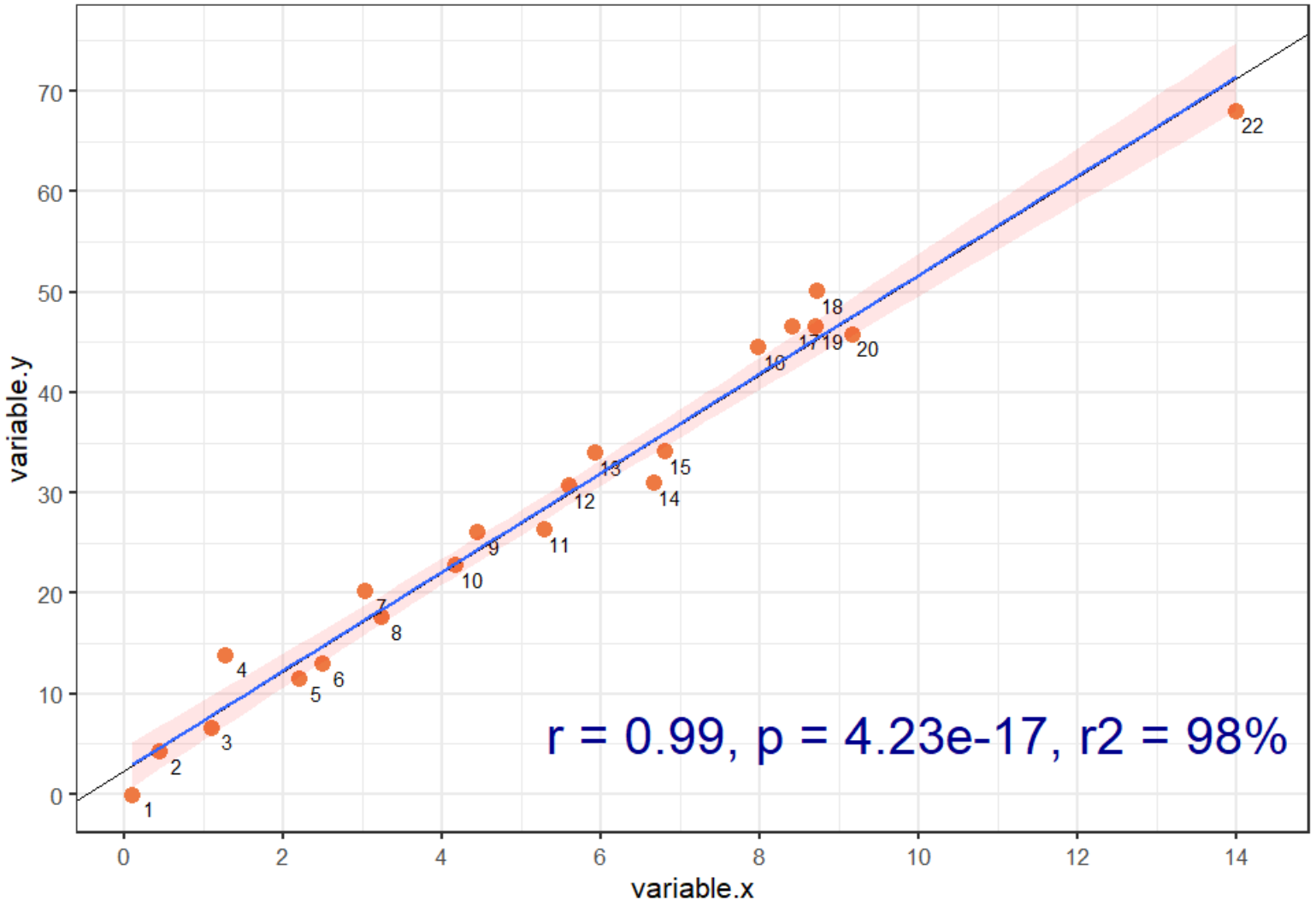
From r = 0.96

```
Call:
lm(formula = variable.y ~ variable.x, data = correlation.23.21)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3636 -1.8607 -0.5376  2.2987  5.0434

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4679     1.0757   2.294   0.0333 *
variable.x    4.9272     0.1719  28.661   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.709 on 19 degrees of freedom
Multiple R-squared:  0.9774,    Adjusted R-squared:  0.9762
F-statistic: 821.4 on 1 and 19 DF,  p-value: < 2.2e-16
```

From r^2 = 0.93

Correlation: correlation.csv

Finally



Correlation: correlation.csv

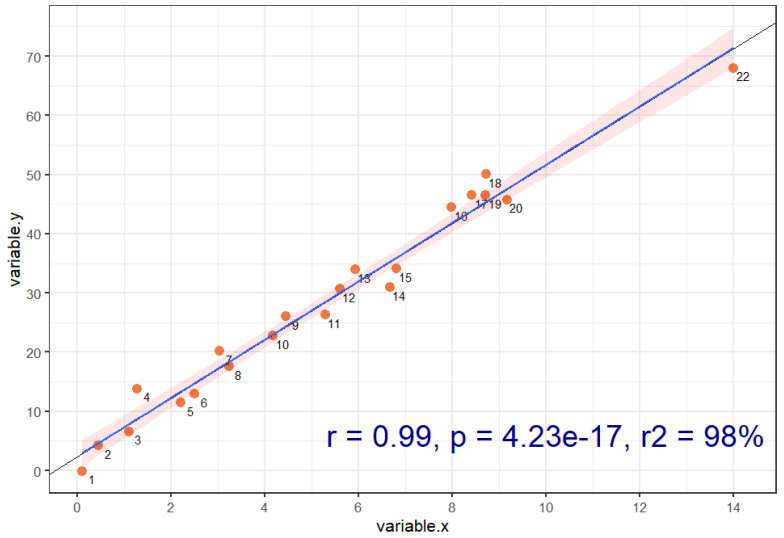
Final code for pretty graph

```
correlation.23.21%>%
  ggplot(aes(x=variable.x, y=variable.y, label = ID)) +
  geom_point(size=4, colour="sienna2") +
  geom_abline(intercept = coef.correlation.23.21[1], slope = coef.correlation.23.21[2])+
  geom_text(vjust = 1.3, nudge_x = 0.2)+
  geom_smooth(method=lm, se=TRUE, level=0.95, fill="red", alpha=0.1)+
scale_x_continuous(breaks=seq(from=0, by=2, to=20))+
  scale_y_continuous(breaks=seq(from=0, by=10, to=80))+
  annotate(geom="text", label="r = 0.99, p = 4.23e-17, r2 = 98%", x=10, y=6, size=10,colour="darkblue")
```

Depends on what your aim is:

- If want to predict, want the best model
- If want to best represent your data, might not want to exclude

Beware of overfitting



Exercise 4

Babraham Institute

Analysis of Quantitative data
Introduction to Linear Modelling

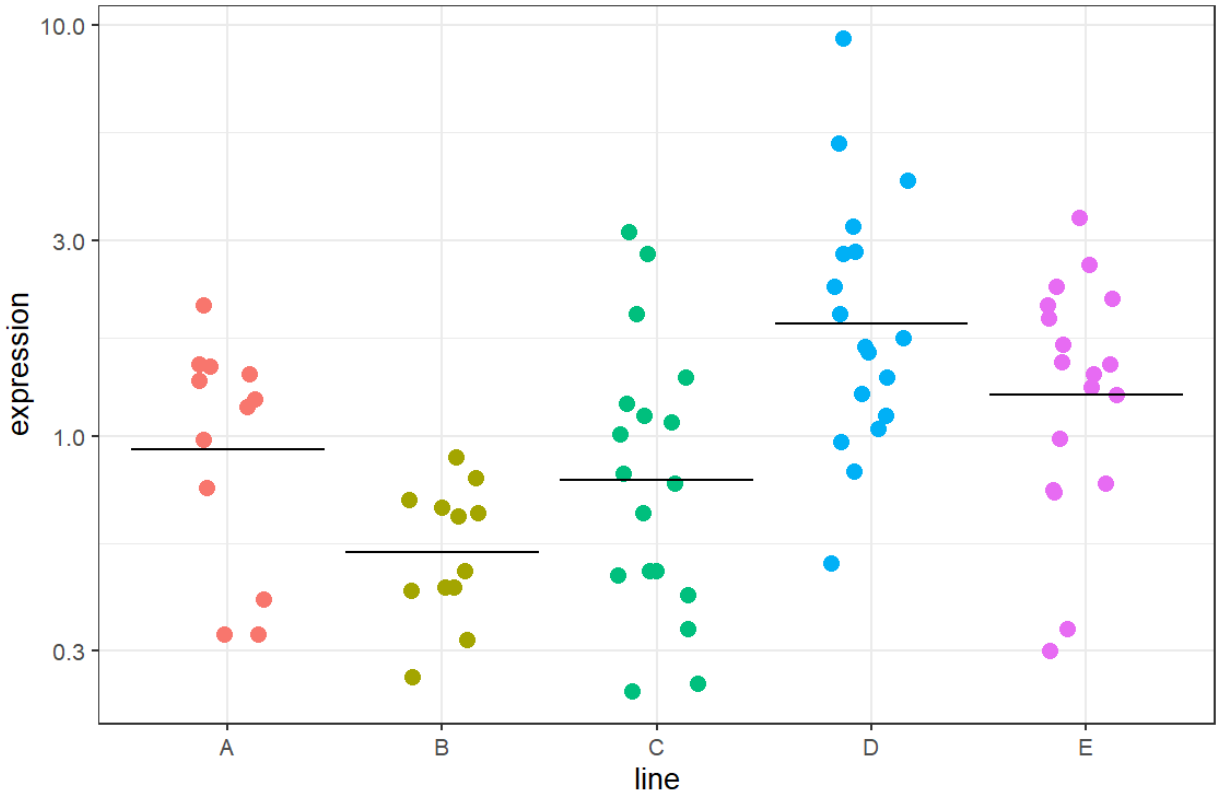
Hayley Carr & Anne Segonds-Pichon

v2025-02

Babraham Bioinformatics

Linear modelling is about language

Is there a difference between the cell lines?



Can cell line predict expression?

Model(line) = expression

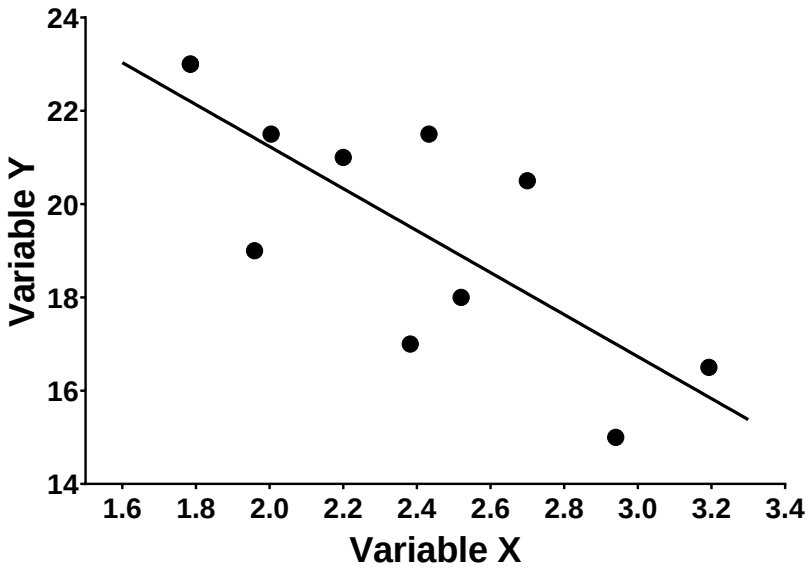
Simple linear model

Linear regression

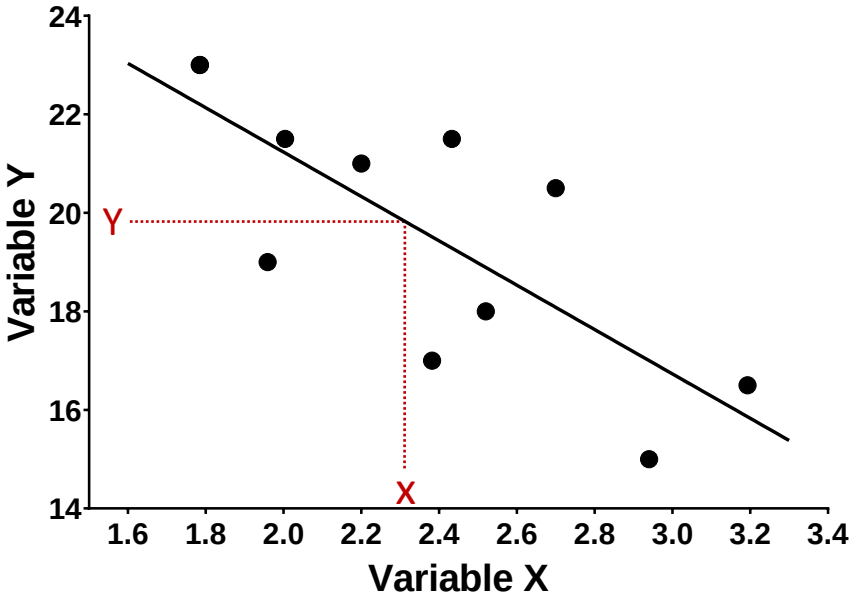
Correlation: is there an **association** between 2 variables?

Regression: is there an **association** and

can one variable be used to **predict** the values of the other?



Correlation = Association



Regression = Prediction

Y = A*X + B

Simple linear model

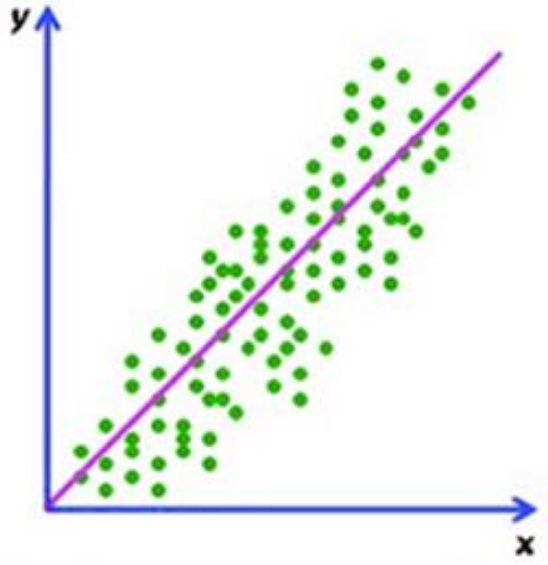
- Linear regression models the dependence between 2 variables:
  a **dependent y** and a **independent x**
  Model(x) = y

response

predictor

$$y = \beta_0 + \beta_1 * x$$

Model



- **In R:**

Linear regression: `lm()`

Linear regression

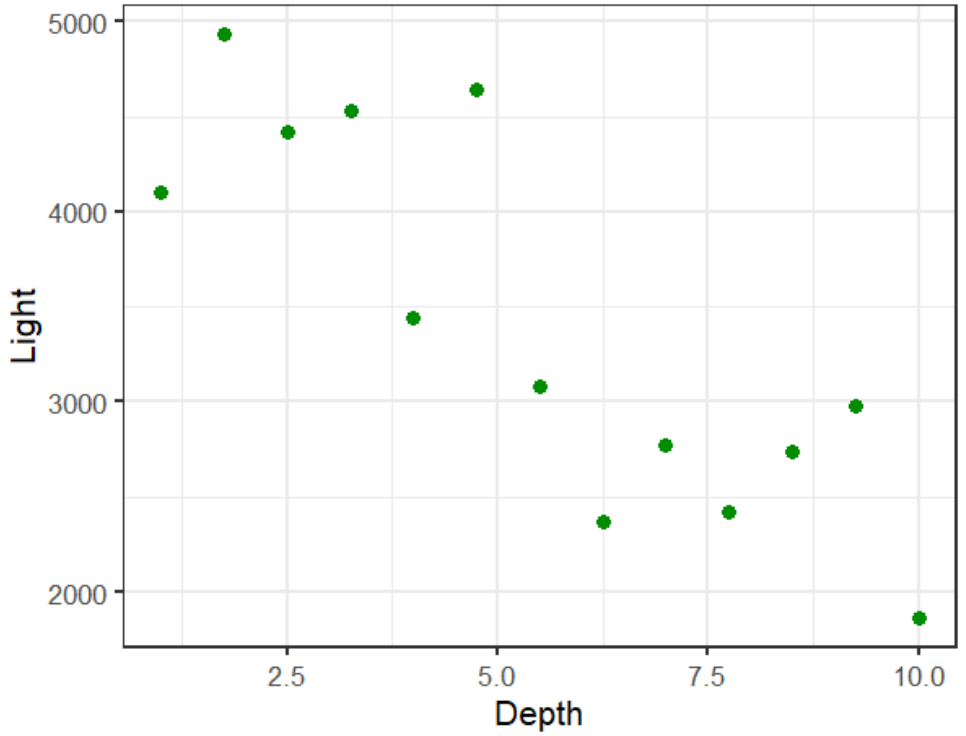
- **Example: treelight.csv**

```
treelight<-read_csv("treelight.csv")
```

Light <dbl>	Depth <dbl>
4105.646	1.00
4933.925	1.75
4416.527	2.50
4528.618	3.25
3442.610	4.00
4640.297	4.75
3081.990	5.50
2368.113	6.25
2776.557	7.00
2419.193	7.75

- Question: how is **light** affected by the **depth** at which it is measured?

$$\textbf{light} = \beta_0 + \beta_1 * \textbf{depth}$$



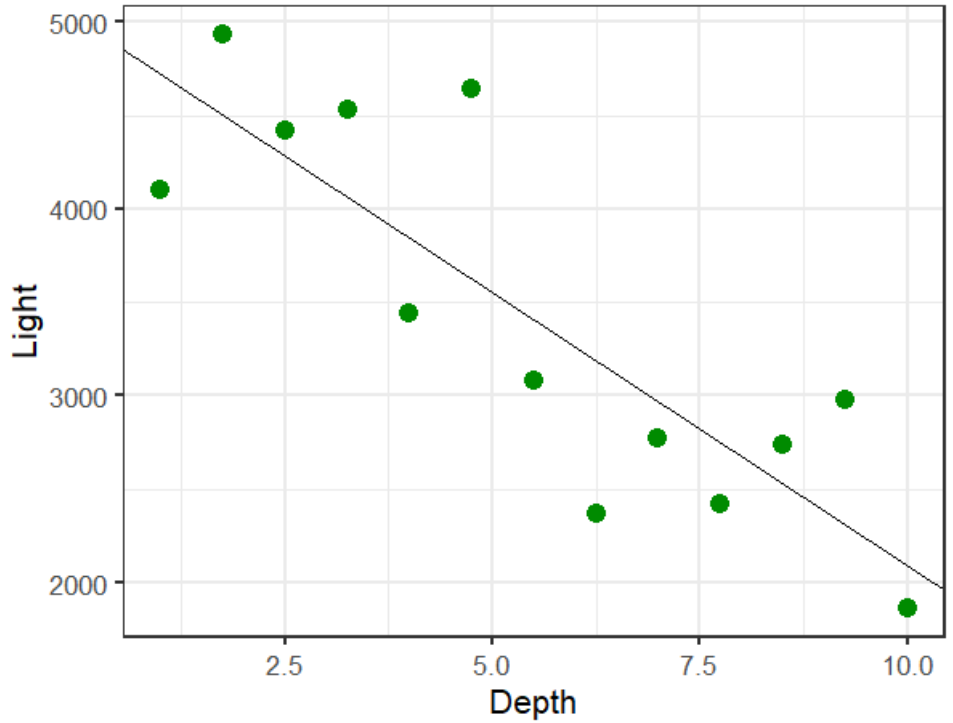

```
treelight %>%
  ggplot(aes(x=Depth, y=Light))+
  geom_point(colour="forestgreen", size=3)
```

Linear regression

```
coefficients(fit.treelight) -> coef.treelight
```

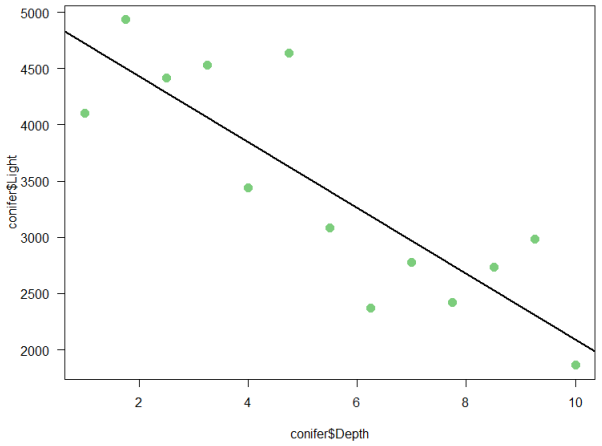
```
(Intercept)       Depth
  5013.9822   -292.1614
```

```
treelight %>%
  ggplot(aes(x=Depth, y=Light)) +
  geom_point(size=4, colour="green4") +
  geom_abline(intercept = coef.treelight[1], slope = coef.treelight[2])
```

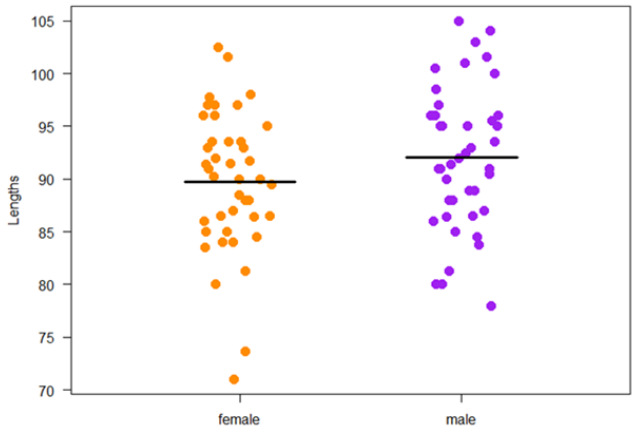


light = 5014 - 292*depth

The linear model perspective



Continuous predictor



Categorical predictor

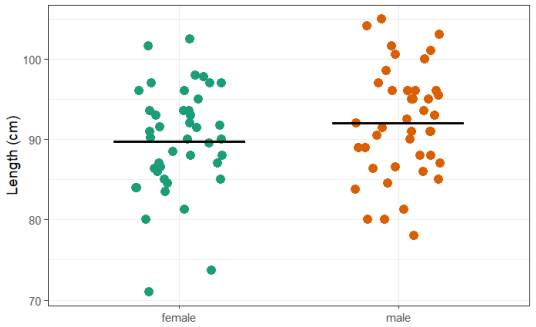
Coyotes body length

- Is there a difference between the 2 sexes?

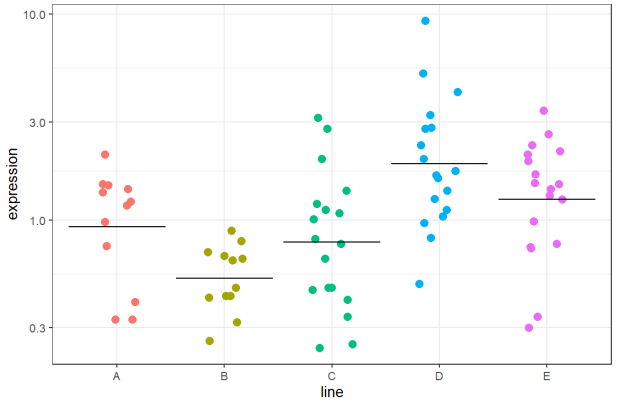
becomes

- **Does sex predict coyote body length?**

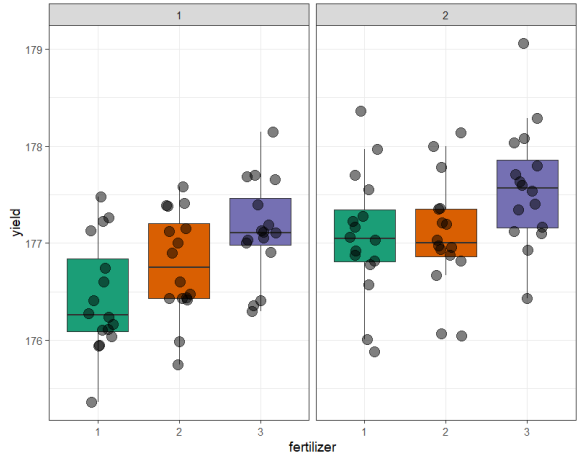
The linear model perspective



Coyotes = Body length~sex



Protein = Expression~Cell line



Crop = Yield~Fertiliser and Density

Example: coyotes

- **Questions**: *do male and female coyotes differ in size?*
  - Does sex predict coyote body length?
  - How much of body length is predicted by sex?

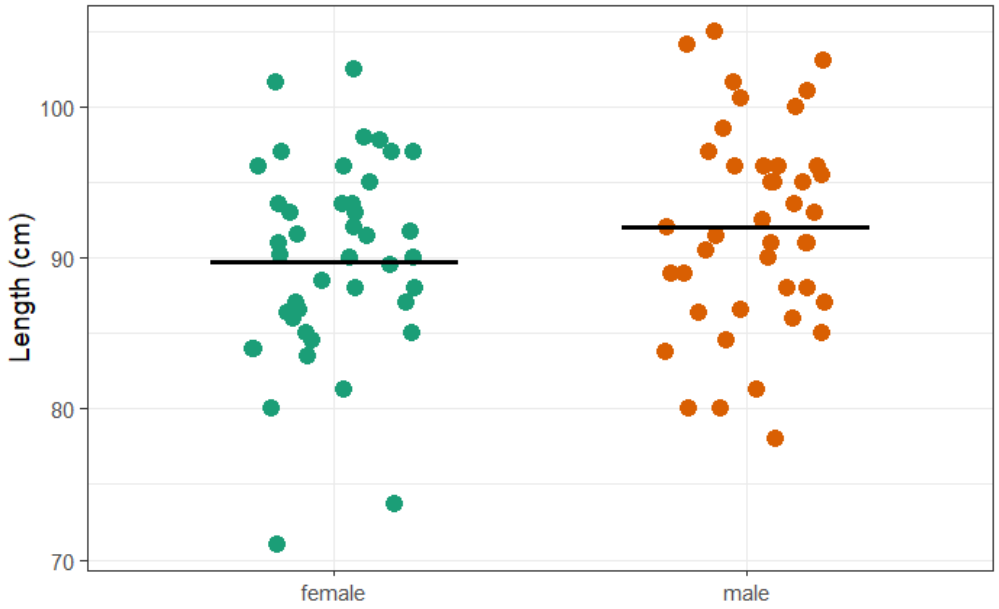
Exercises: coyotes

- **coyote.csv** `coyote <- read_csv("coyote.csv")`
  - Run the t-test again `t_test()`
  - Run the same analysis using a linear model approach `lm()`
  - Compare the outputs and understand the coefficients from `lm()`
  - Use `summary()` and `anova()` to explore further
  - Work out R^2 from the `anova()` output
  - *Don't forget to check the assumptions*

The linear model perspective

Comparing 2 groups

```
read_csv("coyote.csv") -> coyote

coyote %>%
  ggplot(aes(x=sex, y=length, colour=sex)) +
    geom_jitter(height=0, size=4, width=0.2) +
    theme(legend.position = "none")+
    ylab("Length (cm)")+
    scale_colour_brewer(palette="Dark2")+
    xlab(NULL)+
    stat_summary(fun=mean, fun.min=mean, fun.max=mean, geom="errorbar",colour="black", linewidth=1.2,
    width=0.6)
```



The linear model perspective

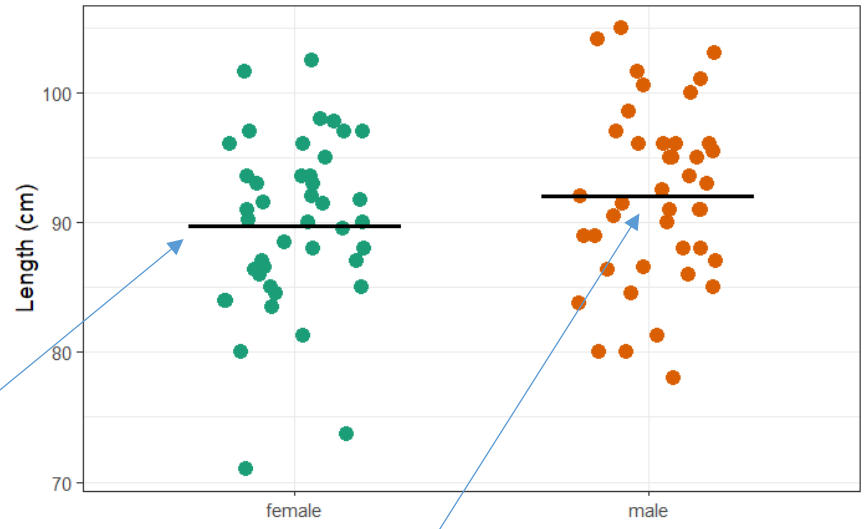
Comparing 2 groups

```
coyote %>%
  t_test(length~sex, var.equal=T)
```

.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>	statistic <dbl>	df <dbl>	p <dbl>
length	female	male	43	43	-1.641109	84	0.105

```
lm(length~sex, data=coyote)
```

```
Call:
lm(formula = length ~ sex, data = coyote)

Coefficients:
(Intercept)      sexmale
     89.712        2.344
```



Females=89.71 cm, Males=89.71 + 2.34=92.05

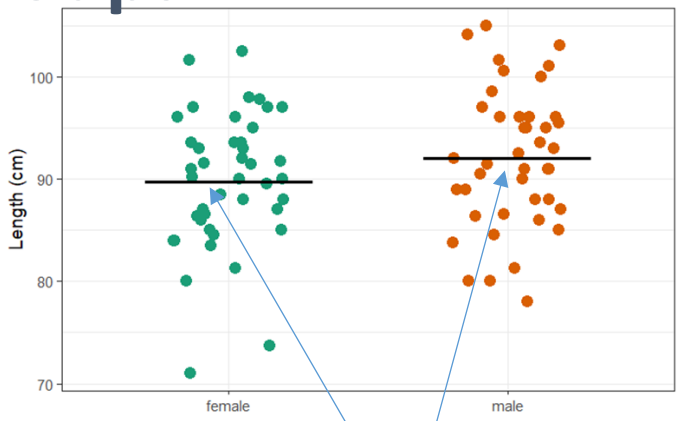
The linear model perspective

Comparing 2 groups

```
lm(length~sex, data=coyote)
```

```
Call:
lm(formula = length ~ sex, data = coyote)

Coefficients:
(Intercept)       sexmale
     89.712         2.344
```

Body length = $\beta_0 + \beta_1$*sex



Model

$$\text{Body Length} = \begin{pmatrix} 89.71 \\ 92.06 \end{pmatrix} \begin{pmatrix} \text{If Female} \\ \text{If Male} \end{pmatrix}$$

$$\text{Body Length} = 89.71 + \begin{pmatrix} 0 \\ 2.344 \end{pmatrix} \begin{pmatrix} \text{If Female} \\ \text{If Male} \end{pmatrix}$$

Body length = 89.712 + 2.344*sex

The linear model perspective

Comparing 2 groups

$$y = \beta_0 + \beta_1 * x$$

continuous

treelight.csv

light = 5014 - 292*depth

categorical

coyote.csv

Body length = 89.712 + 2.344*sex

$$\text{Body Length} = 89.71 + \begin{pmatrix} 0 \\ 2.344 \end{pmatrix} \begin{pmatrix} \text{If Female} \\ \text{If Male} \end{pmatrix}$$

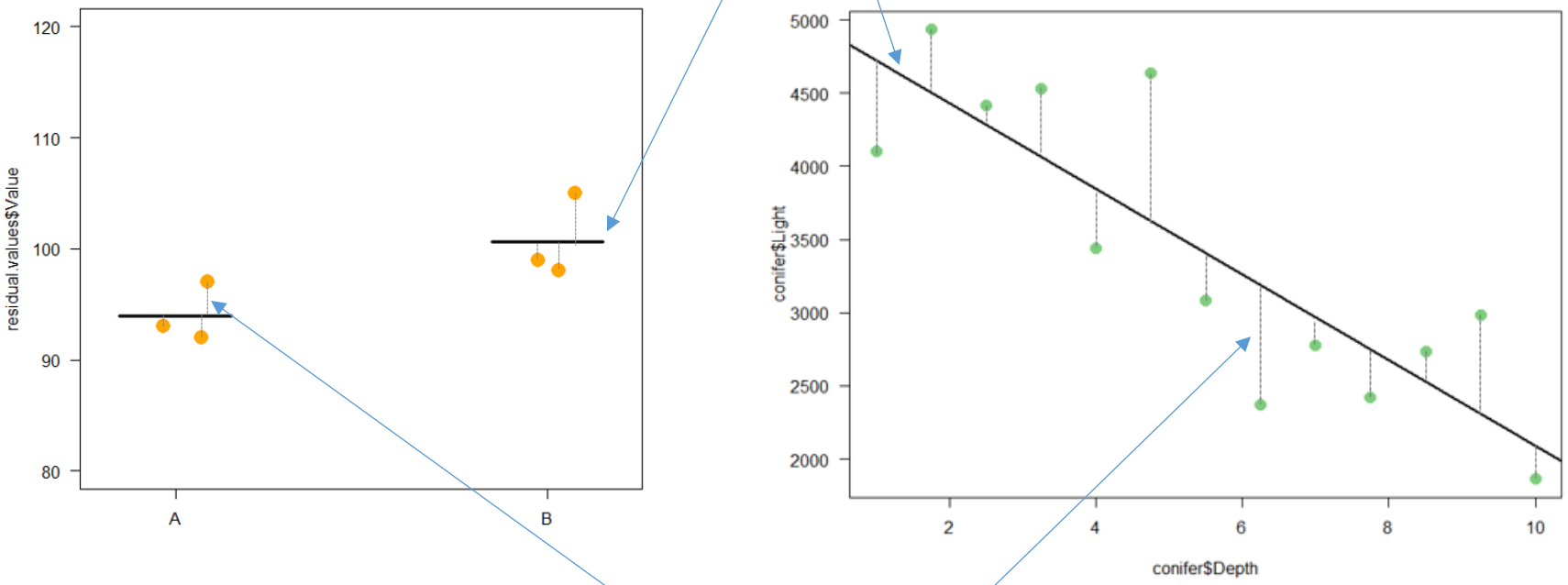
vector

$$y = \beta_0 + \beta_1 * x$$

The linear model perspective

Comparing 2 groups

Model



Residuals

The linear model perspective

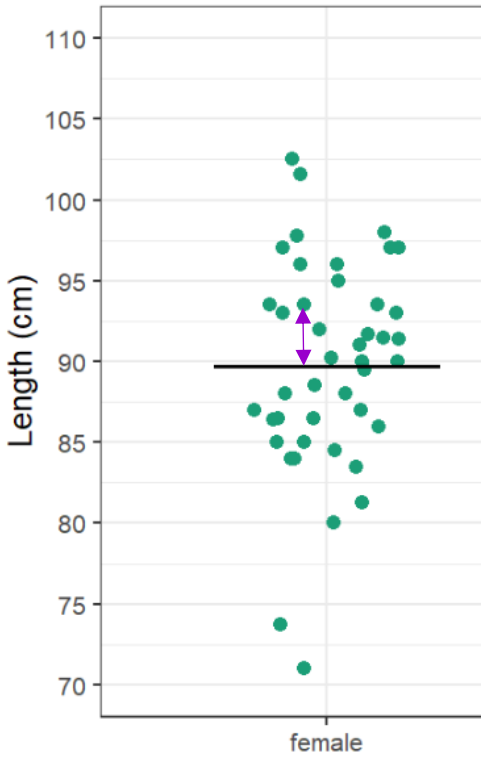
Comparing 2 groups

```
linear.coyote<-lm(length~sex, data=coyote)
```

```
linear.coyote

Coefficients:
(Intercept)      sexmale
     89.712        2.344
```

linear.coyote	list [13] (S3: lm)	List of length 13
coefficients	double [2]	89.71 2.34
(Intercept)	double [1]	89.71163
coyote$gendermale	double [1]	2.344186
residuals	double [86]	3.29 7.29 2.29 11.89 3.29 -5.21 ...

86 coyotes



$$\text{Body Length} = 89.71 + \begin{pmatrix} 0 \\ 2.344 \end{pmatrix} \begin{pmatrix} \text{If Female} \\ \text{If Male} \end{pmatrix}$$

Female 1: 89.71 + 3.29 = 93 cm

length	gender
93.0	female
97.0	female
92.0	female
101.6	female
93.0	female
84.5	female
102.5	female
97.8	female
91.0	female
98.0	female

The linear model perspective

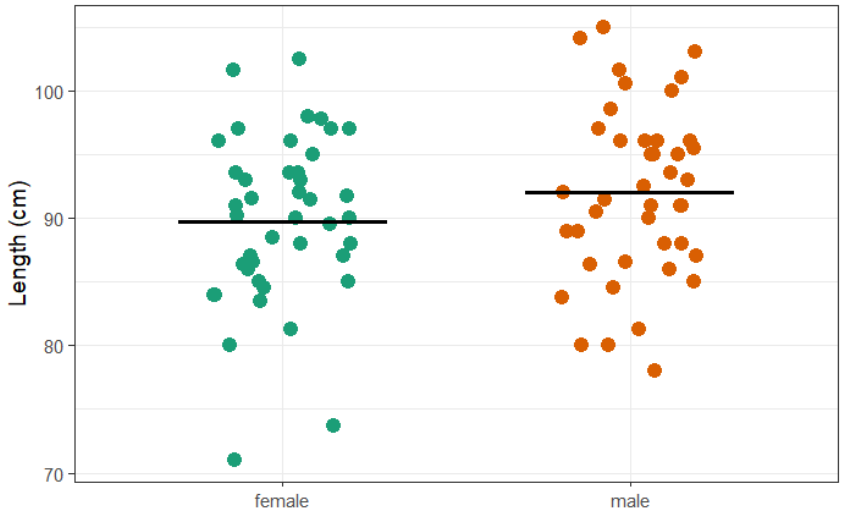
Comparing 2 groups

```
coyote %>%
  t_test(length~sex, var.equal=T)
```

.y. <chr>	group1 <chr>	group2 <chr>	n1 <int>	n2 <int>	statistic <dbl>	df <dbl>	p <dbl>
length	female	male	43	43	-1.641109	84	0.105

```
summary(linear.coyote)
```

```
Call:
lm(formula = length ~ sex, data = coyote)

Residuals:
     Min       1Q   Median       3Q      Max
-18.7116  -4.0558   0.2884   3.9442  12.9442

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   89.712      1.010  88.820   <2e-16 ***
sexmale        2.344      1.428   1.641    0.105
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.623 on 84 degrees of freedom
Multiple R-squared:  0.03107,   Adjusted R-squared:  0.01953
F-statistic: 2.693 on 1 and 84 DF,  p-value: 0.1045
```



The linear model perspective

Comparing 2 groups

```
anova(linear.coyote)
```

```
Analysis of Variance Table

Response: length
          Df  Sum Sq Mean Sq F value Pr(>F)
sex        1   118.1 118.147  2.6932 0.1045
Residuals 84  3684.9  43.868
```

118.1 + 3684.9 = 3803: total amount of variance in the data

Proportion explained by sex: 118.1/3803 = **0.031**

About 3% of the variability in body length is explained by sex

```
summary(linear.coyote)
```

```
Call:
lm(formula = length ~ sex, data = coyote)

Residuals:
     Min       1Q   Median       3Q      Max
-18.7116  -4.0558   0.2884   3.9442  12.9442

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   89.712      1.010  88.820   <2e-16 ***
sexmale        2.344      1.428   1.641    0.105
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.623 on 84 degrees of freedom
Multiple R-squared:  0.03107,   Adjusted R-squared:  0.01953
F-statistic: 2.693 on 1 and 84 DF,  p-value: 0.1045
```

R^2 = coefficient of determination

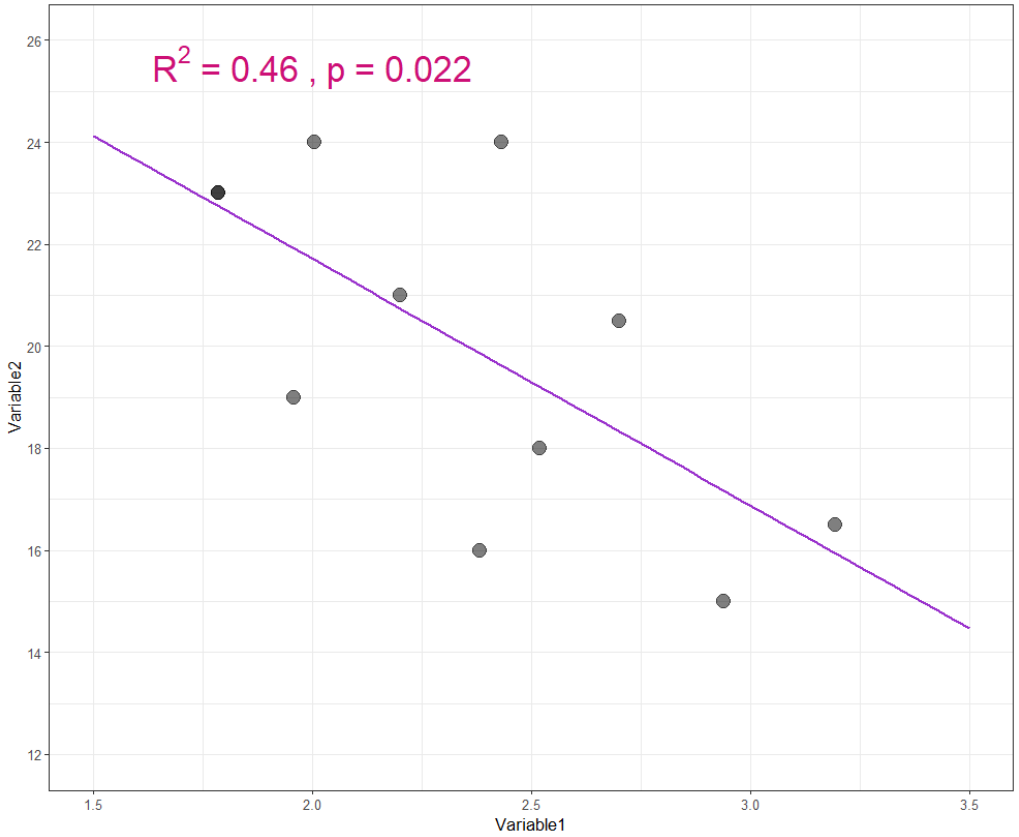
Same as in correlation/linear regression

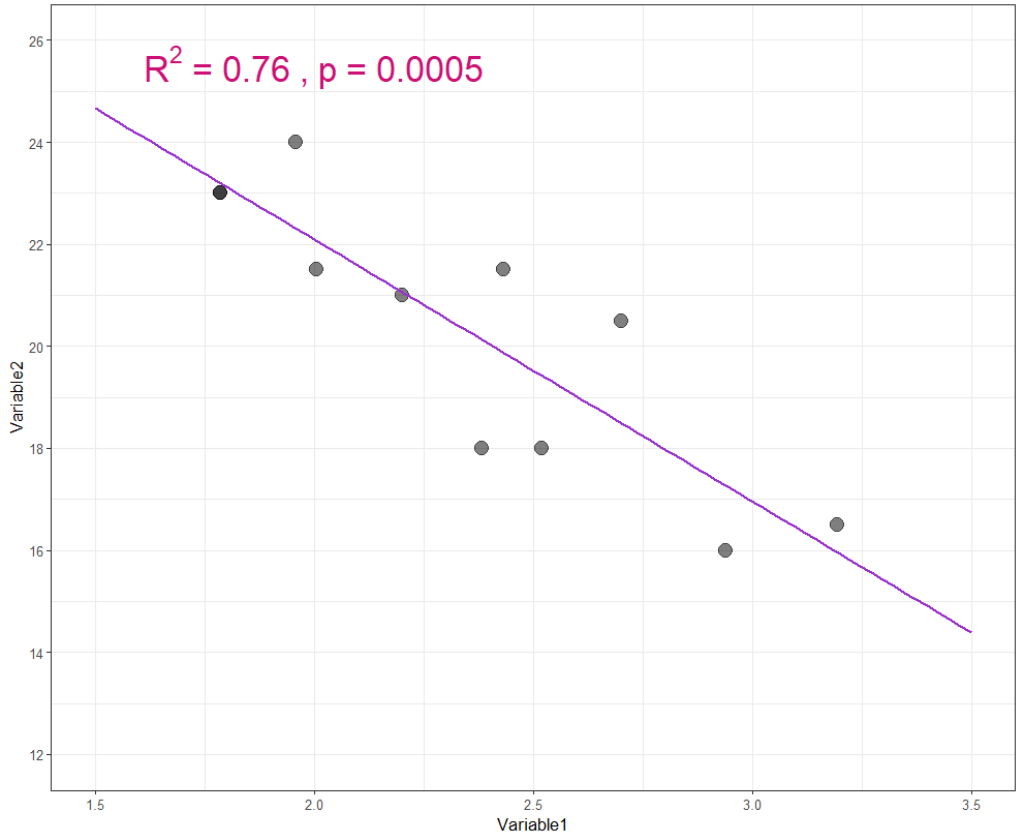
= Variability in y explained by x

Coefficient of determination

An illustration: change in variability

variability ↓ R^2 ↑ p-value ↓

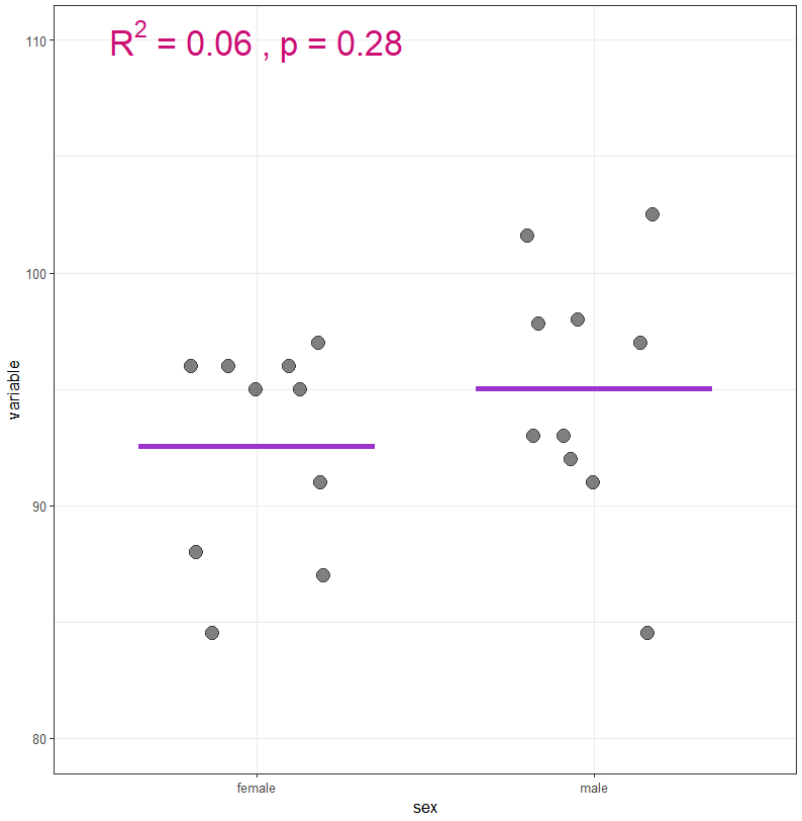


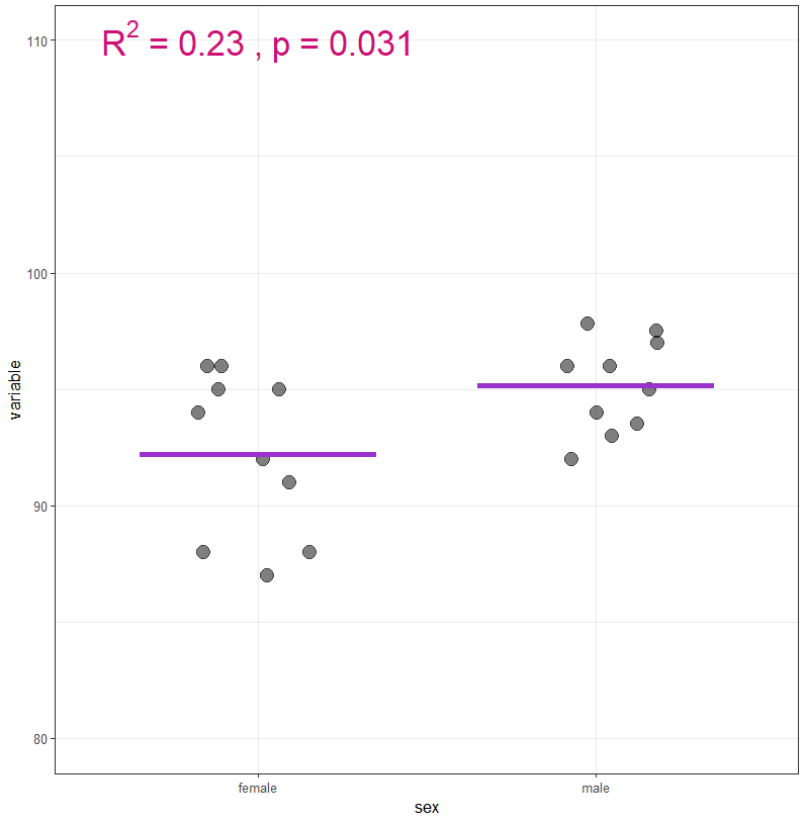


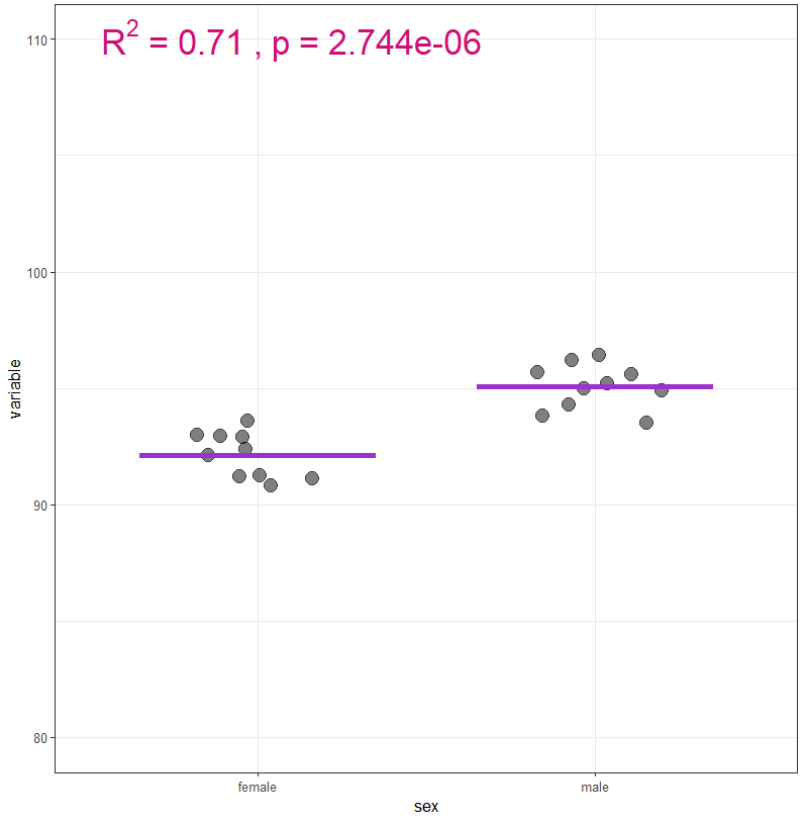
Coefficient of determination

An illustration: change in variability

variability ↓ R^2 ↑ p-value ↓

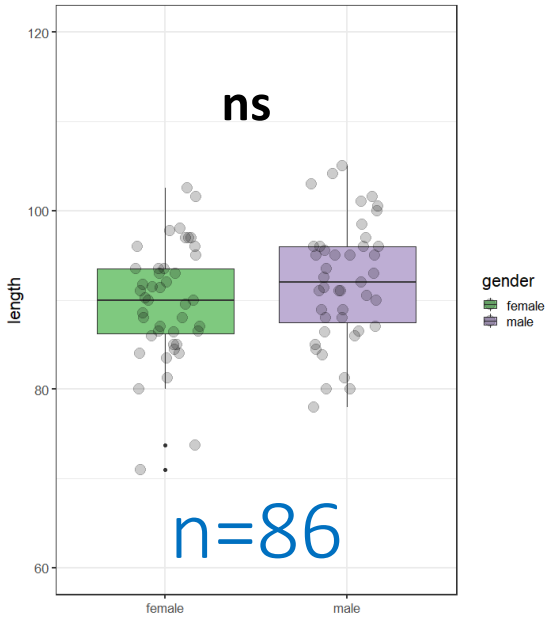






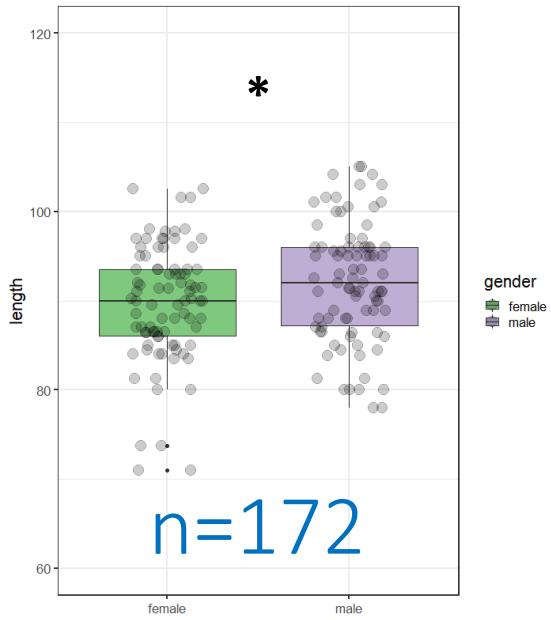
Coefficient of determination

An illustration: change in sample size



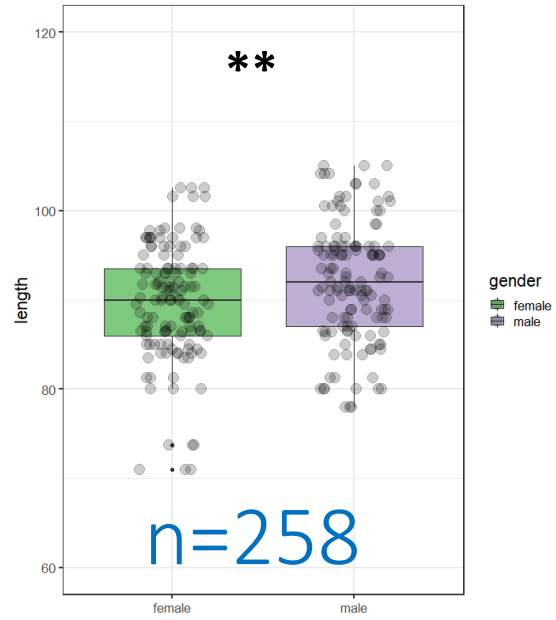
t-test

.y.	group1	group2	n1	n2	statistic	df	p
<chr>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>
length	female	male	43	43	-1.64	84.0	0.105



t-test

.y.	group1	group2	n1	n2	statistic	df	p
<chr>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>
length	female	male	86	86	-2.33	170.	0.0207



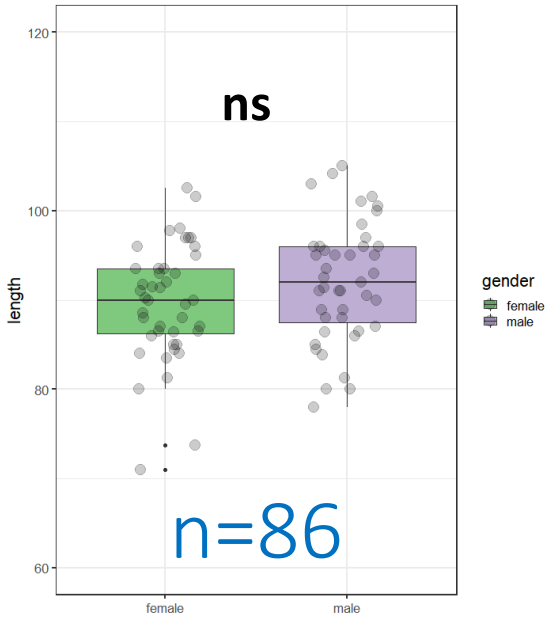
t-test

.y.	group1	group2	n1	n2	statistic	df	p
<chr>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>
length	female	male	129	129	-2.86	256.	0.00452

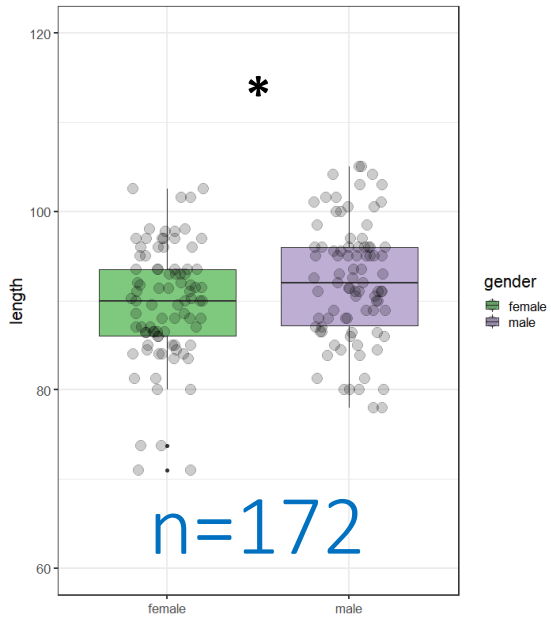
Coefficient of determination

An illustration: change in sample size

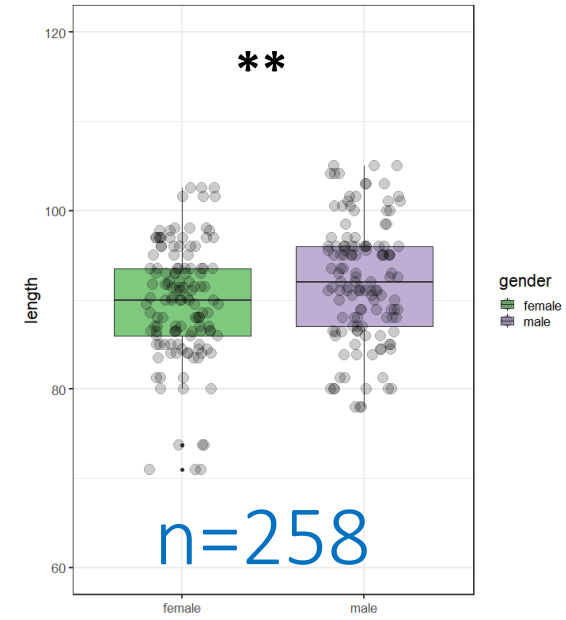
Sample ↑ = Power ↑ → R^2 does not change but p-value ↓




```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   89.712      1.010  88.820   <2e-16 ***
gendermale     2.344      1.428   1.641    0.105
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.623 on 84 degrees of freedom
Multiple R-squared:  0.03107,    Adjusted R-squared:  0.01953
F-statistic: 2.693 on 1 and 84 DF,  p-value: 0.1045
```




```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   89.712      0.710 126.355   <2e-16 ***
gendermale     2.344      1.004   2.335   0.0207 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.584 on 170 degrees of freedom
Multiple R-squared:  0.03107,    Adjusted R-squared:  0.02537
F-statistic: 5.451 on 1 and 170 DF,  p-value: 0.02073
```




```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  89.7116     0.5786 155.056  < 2e-16 ***
gendermale    2.3442     0.8182   2.865  0.00452 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.571 on 256 degrees of freedom
Multiple R-squared:  0.03107,    Adjusted R-squared:  0.02728
F-statistic: 8.208 on 1 and 256 DF,  p-value: 0.004517
```

The linear model perspective

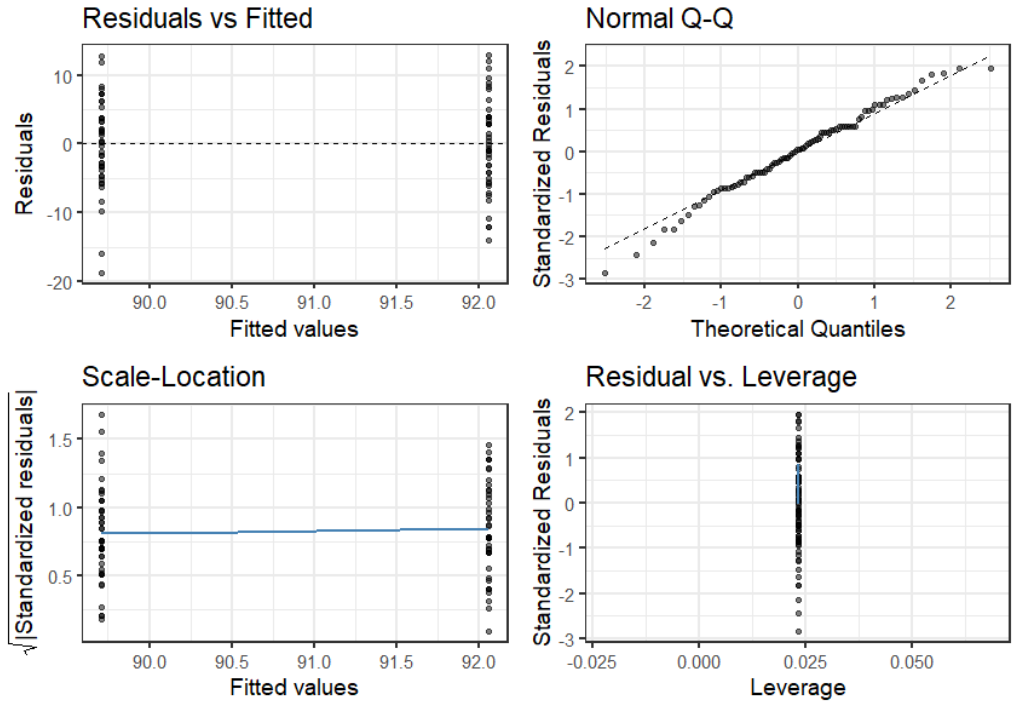
Comparing 2 groups

`linear.coyote`

Assumptions

```
linear.coyote                List of 13
 coefficients : Named num [1:2] 89.71 2.34
 ..- attr(*, "names")= chr [1:2] "(Intercept)" "coyote$gendermale"
 residuals : Named num [1:86] 3.29 7.29 2.29 11.89 3.29 ...
 - attr(*, "names")= chr [1:86] "1" "2" "3" "4"
```

```
gglm(linear.coyote, theme = theme_bw(base_size = 16))
```



Example: coyote.csv

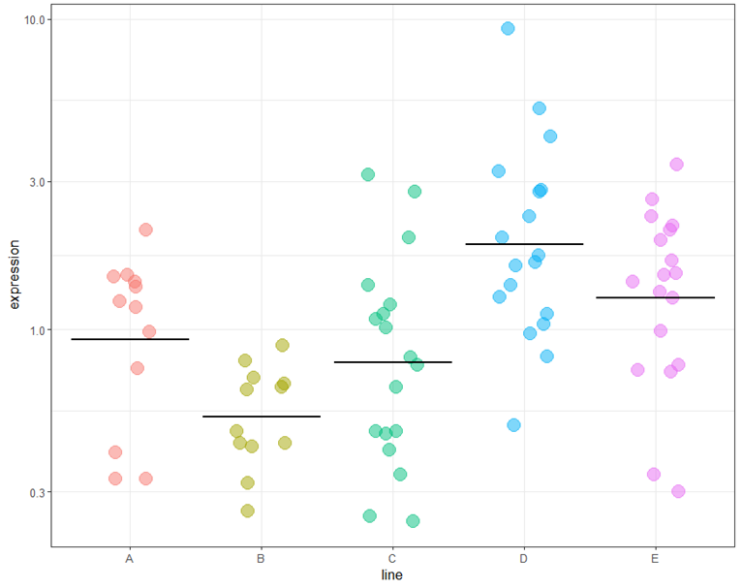
- **Questions**: *do male and female coyotes differ in size?*
  - Does sex predict body length?
    - **Answer**: Quite unlikely: p = 0.105
  - How much of body length is predicted by sex?
    - **Answer**: About 3% (R^2=0.031)

The linear model perspective

One factor with more than 2 levels

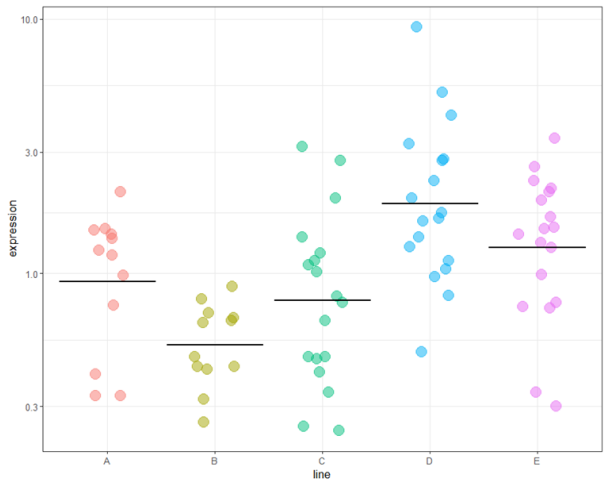
protein.expression.csv

- **Questions**: *is there a difference in protein expression between the 5 cell lines?*
  - Does cell line predict protein expression?
  - How much of the protein expression is predicted by the cell line?



Exercise: protein expression

- **protein.expression.csv** `protein<-read_csv("protein.expression.csv")`
  - Log-transformed the expression `log10()`
  - Run the ANOVA again using `anova_test()`
  - Use `lm()` and `summary()` for the linear model approach
  - Compare the 2 outputs
  - Work out the means log10.expression for the 5 cell lines
  - Compare the outputs and understand the coefficients from `lm()`
  - Work out R^2 from the `anova()` output
  - Don't forget to check out the assumptions



Analysis of variance

```
protein %>%
  anova_test(log10.expression~line)
```

```
ANOVA Table (type II tests)

  Effect DFn DFd     F        p p<.05   ges
1   line   4  73 8.123 1.78e-05     * 0.308
```

```
protein %>%
  tukey_hsd(log10.expression~line)
```

Tukey correction

	term <chr>	group1 <chr>	group2 <chr>	estimate <dbl>	conf.low <dbl>	conf.high <dbl>	p.adj <dbl>	p.adj.signif <chr>
1	line	A	B	-0.25024832	-0.578882494	0.07838585	2.19e-01	ns
2	line	A	C	-0.07499724	-0.374997820	0.22500335	9.56e-01	ns
3	line	A	D	0.30549397	0.005493391	0.60549456	4.39e-02	*
4	line	A	E	0.13327517	-0.166725416	0.43327575	7.27e-01	ns
5	line	B	C	0.17525108	-0.124749499	0.47525167	4.81e-01	ns
6	line	B	D	0.55574230	0.255741712	0.85574288	1.83e-05	****
7	line	B	E	0.38352349	0.083522904	0.68352407	5.48e-03	**
8	line	C	D	0.38049121	0.112162532	0.64881989	1.54e-03	**
9	line	C	E	0.20827240	-0.060056276	0.47660108	2.02e-01	ns
10	line	D	E	-0.17221881	-0.440547487	0.09610987	3.84e-01	ns

Analysis of variance: The Linear model perspective

```
linear.protein<-lm(log10.expression~line, data=protein)
```

```
anova(linear.protein)
```

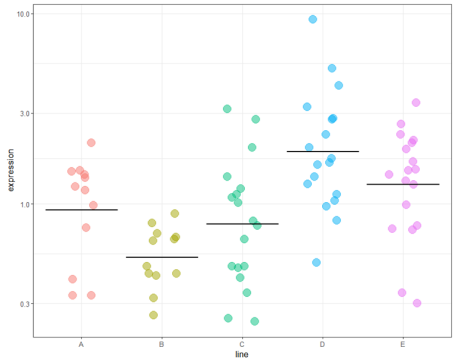
```
          Df Sum Sq Mean Sq F value    Pr(>F)
line       4  2.691  0.6728   8.123 1.78e-05 ***
Residuals 73  6.046  0.0828
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
protein %>%
  anova_test(log10.expression~line)
```

```
ANOVA Table (type II tests)

  Effect DFn DFd     F        p p<.05   ges
1   line   4  73 8.123 1.78e-05     * 0.308
```

```
summary(linear.protein)
```

```
Call:
lm(formula = log10.expression ~ line, data = protein.stack.clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.62471 -0.21993  0.02264  0.18263  0.69537

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03144    0.08308  -0.378  0.70617
lineB       -0.25025    0.11749  -2.130  0.03655 *
lineC       -0.07500    0.10725  -0.699  0.48661
lineD        0.30549    0.10725   2.848  0.00571 **
lineE        0.13328    0.10725   1.243  0.21798
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2878 on 73 degrees of freedom
Multiple R-squared:  0.308,     Adjusted R-squared:  0.2701
F-statistic: 8.123 on 4 and 73 DF,  p-value: 1.784e-05
```




```
lm(log10.expression~line,data=protein)
```

```
Call:
lm(formula = log10.expression ~ line, data = protein.stack.clean)

Coefficients:
(Intercept)        lineB        lineC        lineD        lineE
   -0.03144     -0.25025     -0.07500      0.30549      0.13328
```

Analysis of variance: The Linear model perspective

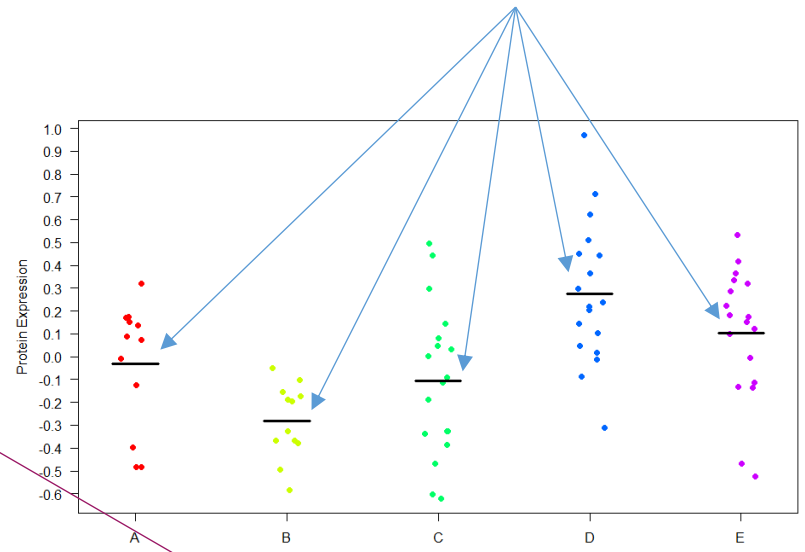
```
protein %>%
  group_by(line) %>%
    summarise(mean=mean(log10.expression))
```

line <chr>	mean <dbl>
A	-0.03144412
B	-0.28169245
C	-0.10644136
D	0.27404985
E	0.10183104

```
lm(log10.expression~line,data=protein)
```

```
Call:
lm(formula = log10.expression ~ line, data = protein.stack.clean)

Coefficients:
(Intercept)       lineB       lineC       lineD       lineE
    -0.03144    -0.25025    -0.07500     0.30549     0.13328
```

Model



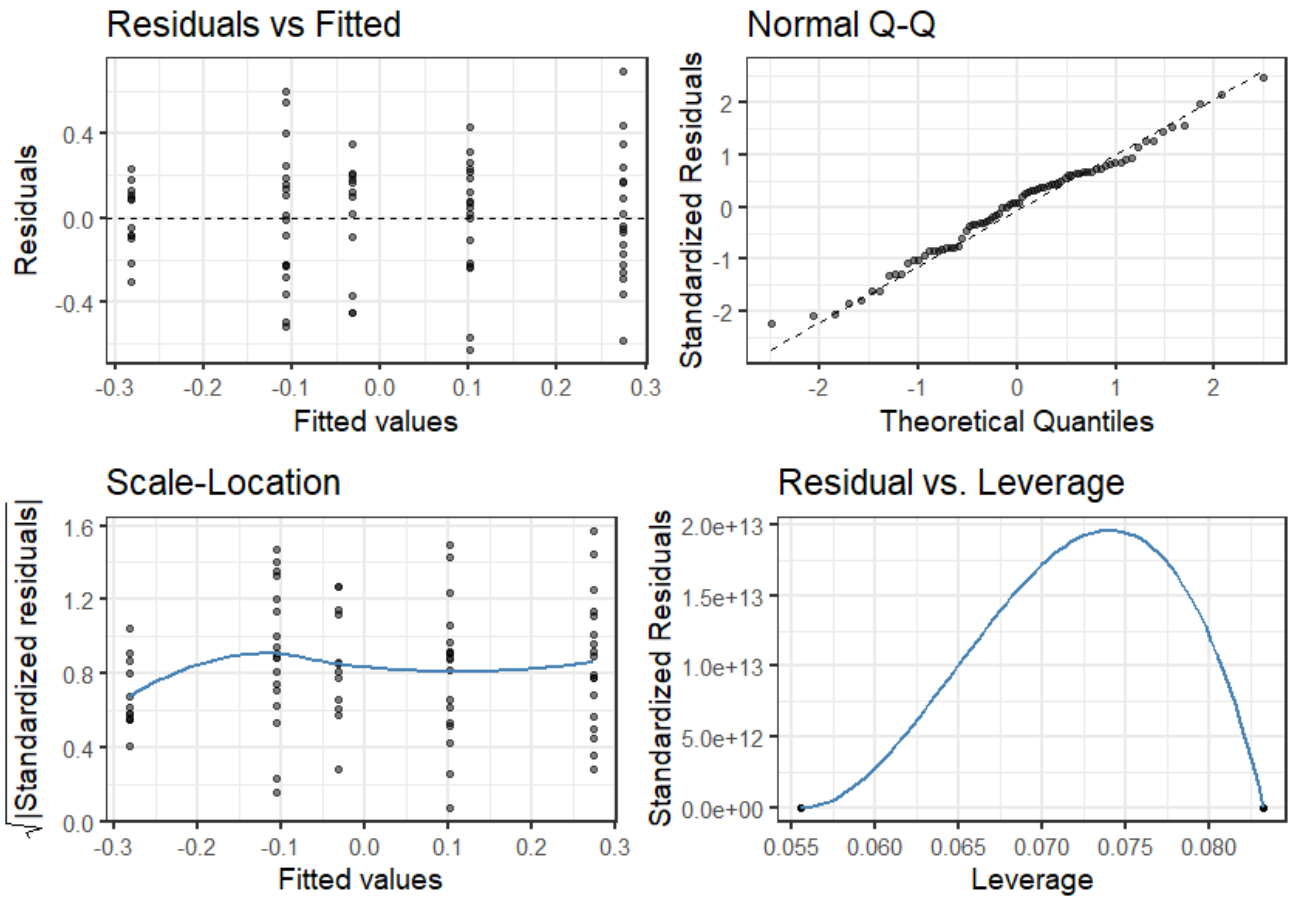
$$\text{Expression} = \beta_0 + \beta_1 * \text{Line}$$

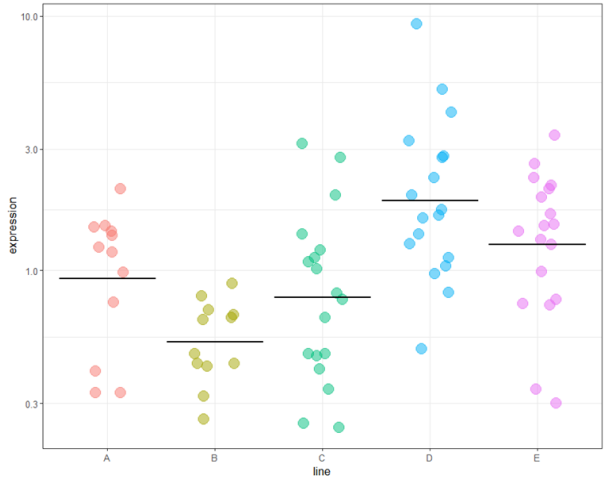
$$\text{Expression} = -0.03144 + \begin{pmatrix} 0 \\ -0.25025 \\ -0.07500 \\ 0.30549 \\ 0.13328 \end{pmatrix} \begin{pmatrix} \text{Line A} \\ \text{Line B} \\ \text{Line C} \\ \text{Line D} \\ \text{Line E} \end{pmatrix}$$

Example:
Line B = -0.03-0.25 = -0.28

Analysis of variance: The Linear model perspective

```
gglm(linear.protein, theme = theme_bw(base_size = 16))
```





Analysis of variance: The Linear model perspective

```
linear.protein<-lm(log10.expression~line,data=protein)
summary(linear.protein)
```

```
Call:
lm(formula = log10.expression ~ line, data = protein)

Residuals:
     Min       1Q   Median       3Q      Max
-0.62471 -0.21993  0.02264  0.18263  0.69537

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03144    0.08308  -0.378  0.70617
lineB       -0.25025    0.11749  -2.130  0.03655 *
lineC       -0.07500    0.10725  -0.699  0.48661
lineD        0.30549    0.10725   2.848  0.00571 **
lineE        0.13328    0.10725   1.243  0.21798
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2878 on 73 degrees of freedom
Multiple R-squared:  0.308,     Adjusted R-squared:  0.2701
F-statistic: 8.123 on 4 and 73 DF,  p-value: 1.784e-05
```

Proportion of variance explained by cell lines: 31%

```
protein %>%
  anova_test(log10.expression~line, detailed = TRUE)
```

```
  Effect   SSn   SSd DFn DFd     F        p p<.05   ges
1   line 2.691 6.046   4  73 8.123 1.78e-05     * 0.308
```

SSn

SSd

Source of variation	Sum of Squares	df	Mean Square	F	p-value
Between Groups	2.691	4	0.673	8.12	<0.0001
Within Groups	6.046	73	0.083		
Total	8.637				

2.691 + 6.046 = 8.737: total amount of variance in the data

Proportion explained by cell line: 2.691/8.737 = **0.308**

Analysis of variance: The Linear model perspective

- **Questions**: *is there a difference in protein expression between the 5 cell lines?*
  - Does cell line predict protein expression?
    - **Answer:** Yes p=1.78e-05
  - How much of the protein expression is predicted by the cell line?
    - **Answer:** About 31% (R^2=0.308)

Linear model: Additional customisation

Default reference group/level

```
linear.protein<-lm(log10.expression~line, data=protein)
summary(linear.protein)
```

Intercept =
Reference level = Line A

```
Call:
lm(formula = log10.expression ~ line, data = protein)

Residuals:
     Min       1Q   Median       3Q      Max
-0.62471 -0.21993  0.02264  0.18263  0.69537

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03144    0.08308  -0.378  0.70617
lineB       -0.25025    0.11749  -2.130  0.03655 *
lineC       -0.07500    0.10725  -0.699  0.48661
lineD        0.30549    0.10725   2.848  0.00571 **
lineE        0.13328    0.10725   1.243  0.21798
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2878 on 73 degrees of freedom
Multiple R-squared:  0.308,     Adjusted R-squared:  0.2701
F-statistic: 8.123 on 4 and 73 DF,  p-value: 1.784e-05
```

Linear model: Additional customisation

Choosing the reference group/level

```
protein %>%
  mutate(line = factor(line)) %>%
  mutate(line = relevel(line, ref = "B")) -> protein
```

```
linear.protein<-lm(log10.expression~line, data=protein)
summary(linear.protein)
```

Intercept =
Reference level = Line B

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.62471 -0.21993  0.02264  0.18263  0.69537

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.28169    0.08308  -3.391 0.001128 **
lineA        0.25025    0.11749   2.130 0.036546 *
lineC        0.17525    0.10725   1.634 0.106565
lineD        0.55574    0.10725   5.182 1.88e-06 ***
lineE        0.38352    0.10725   3.576 0.000624 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2878 on 73 degrees of freedom
Multiple R-squared:  0.308,     Adjusted R-squared:  0.2701
F-statistic: 8.123 on 4 and 73 DF,  p-value: 1.784e-05
```

Linear model

Simplest

$$y = \beta_0 + \beta_1 * x$$

With 2 factors

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_1 x_2$$

With n factors

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_1 x_2 + \ldots + \beta_n * x_n$$

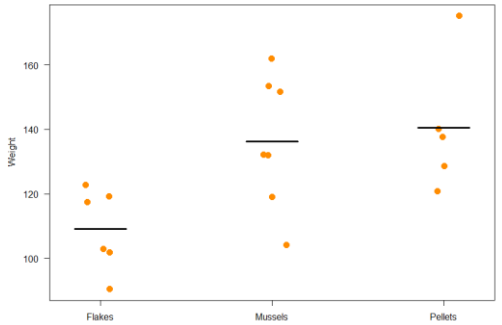
Let's not forget the error

$$y_i = (\beta_0 + \beta_1 * x_i) + \varepsilon_i$$

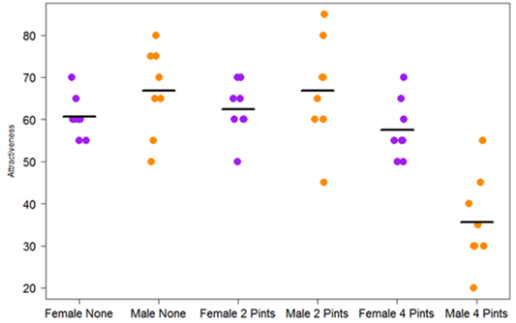
General formula

$$\mathbf{y_i = (model) + error_i}$$

One-way ANOVA

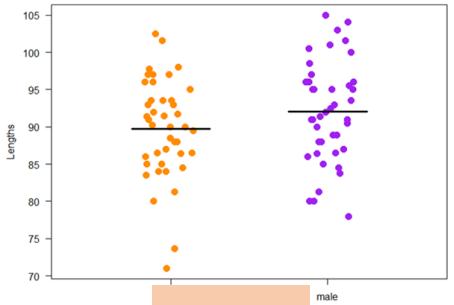


Two-way ANOVA

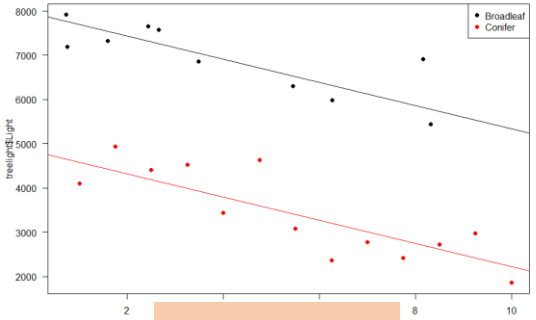


Linear model

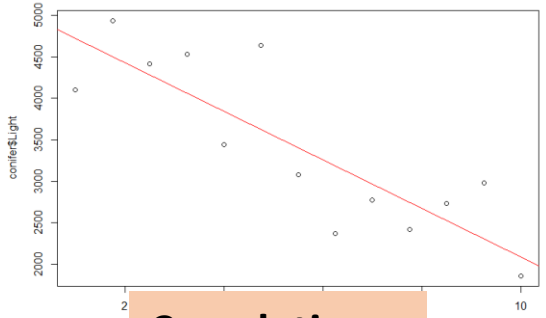
$$y_i = (\text{model}) + \text{error}_i$$



***t*-test**



ANCOVA



Correlation

Babraham Institute

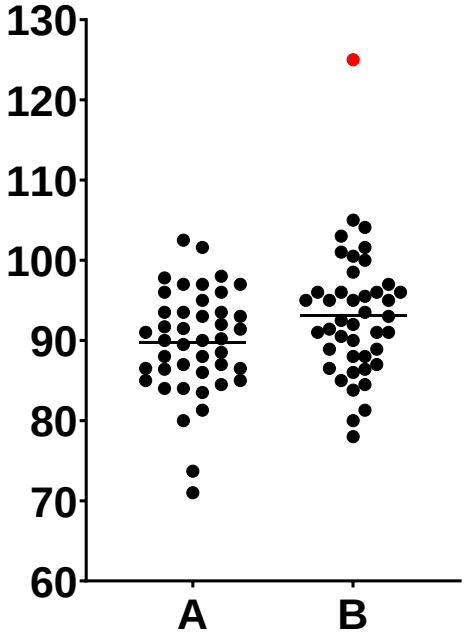
Analysis of Quantitative data
Non parametric statistics

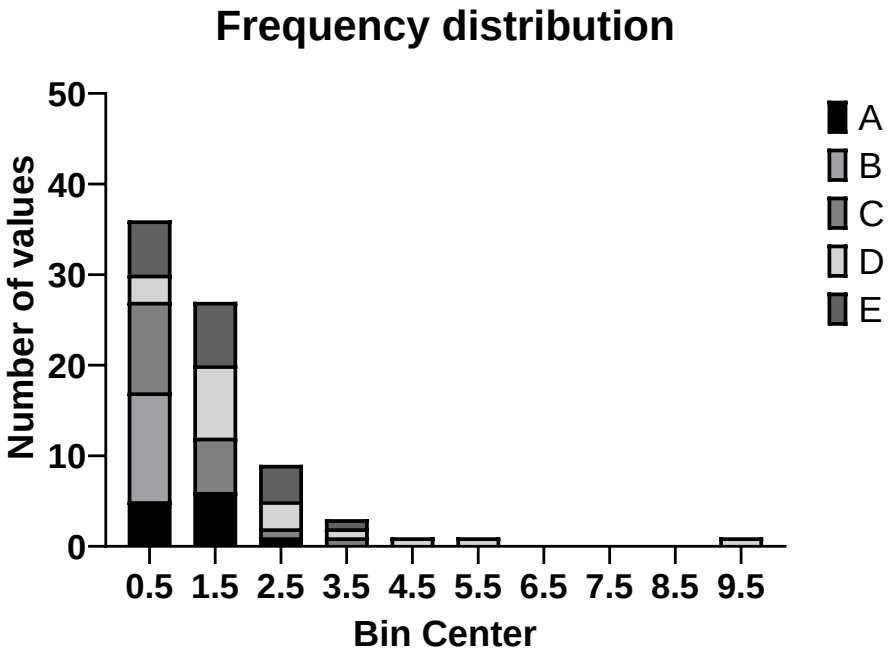
Hayley Carr & Anne Segonds-Pichon
v2025-02

Babraham Bioinformatics

Non-parametric tests

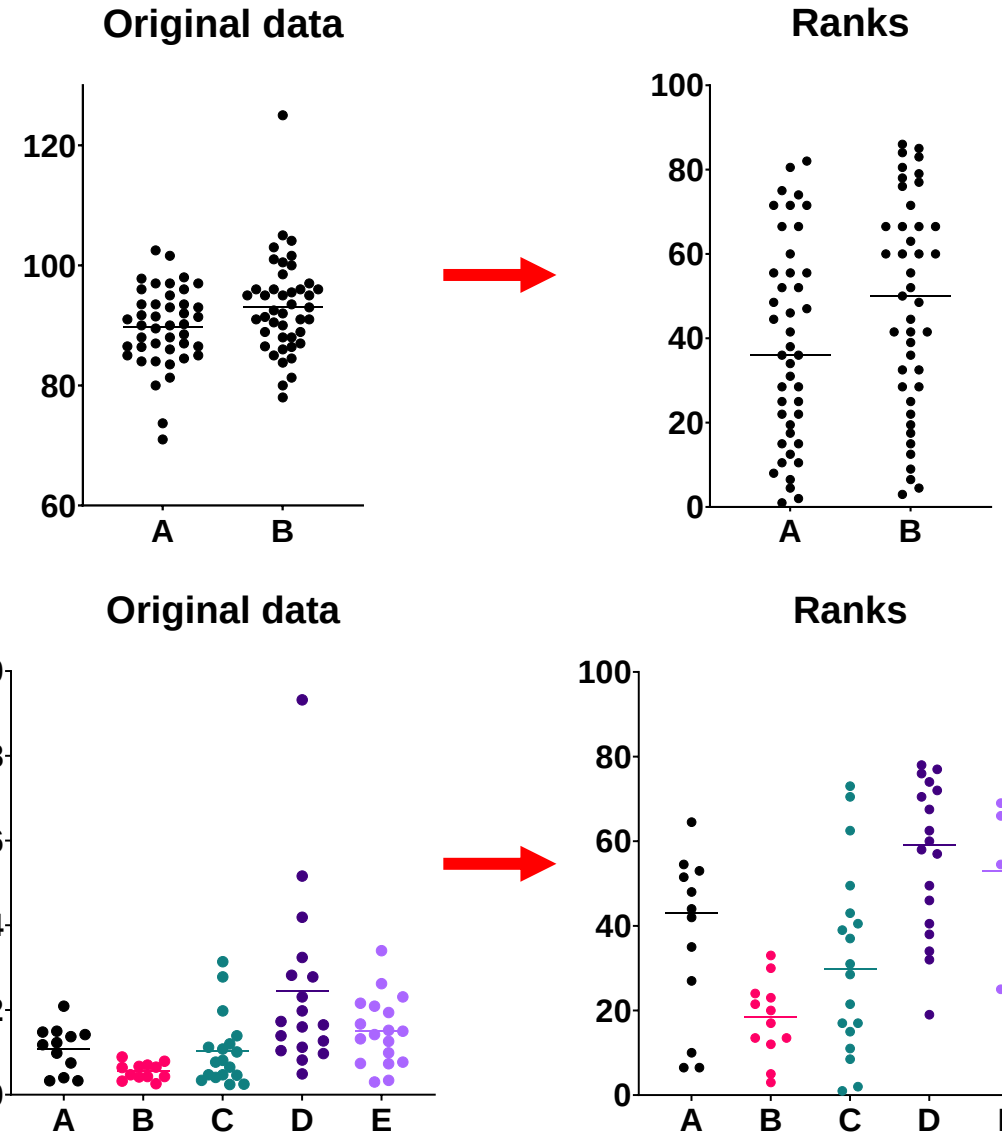
- General principle: original data are transformed into **ranks**
- **Not meeting the assumptions for parametric tests is not enough** to switch to a non-parametric approach
- Data exploration is key:
  - Outliers?
  - Possible transformation?
  - Parametric with corrections?
- If outcome is a rank or a score with limited possible values: often non-parametric approach





Non-parametric tests

- General principle: **original data** are transformed into **ranks**
- Beware of misinterpretation: distribution of the data
  - Distributions = symmetrical and similar → compares means
  - Distributions = similar → compares medians
  - Distributions = not similar → compares distributions (though not always)
- A correction is applied when there are ties



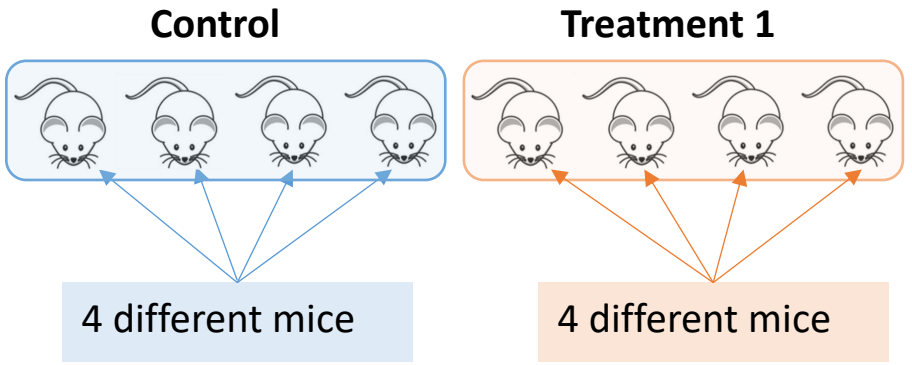
Comparison between 2 groups

Non-Parametric data

Comparison between 2 independent groups

Mann-Whitney *U* test

- Non-parametric equivalent of the *t*-test (and not)
- In the case of **inequality of variance** (violation of the homoscedasticity assumption), the 'unequal' version of the t-test is a possibility: **Welch's t-test**
- For a correct interpretation of the test: **Data exploration!**
- **Mann-Whitney *U* test** (Mann–Whitney–Wilcoxon, Wilcoxon rank-sum test or Wilcoxon–Mann–Whitney)
  - Wilcoxon: equal sample size
  - Mann and Whitney: different sample size



Comparison between 2 independent groups

Mann-Whitney *U* test

- **How does the Mann-Whitney *U* test work?**

Group 1	Group 2
5	8
7	9
3	6

→

Real values	Ranks
3	1
5	2
6	3
7	4
8	5
9	6

→

	Group 1	Group 2
	2	5
	4	6
	1	3
Sum	R_1=7	R_2=14

$$U_1 = R_1 - \frac{n_1(n_1+1)}{2}$$

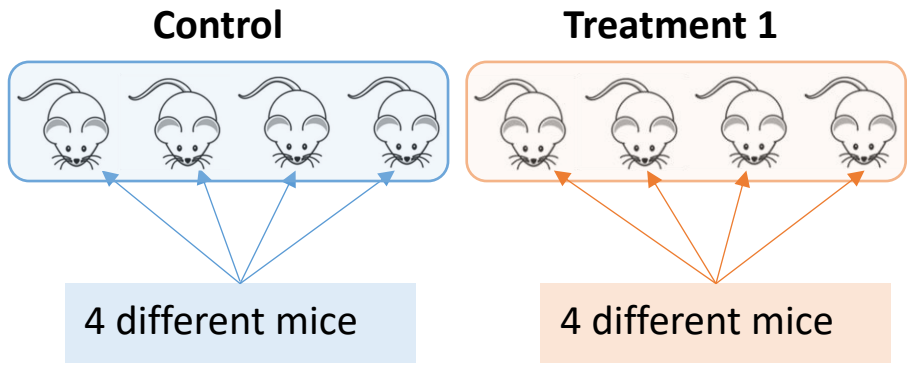
$$U_2 = R_2 - \frac{n_2(n_2+1)}{2}$$

Where:
- R = sum of ranks
- n = sample size

- Statistic of the Mann-Whitney test: **U (W)**

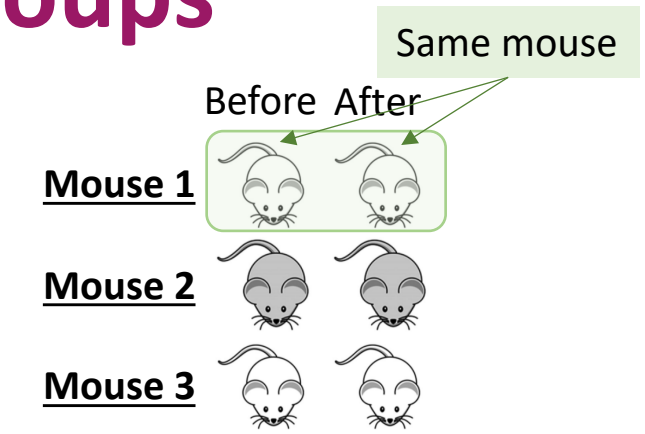
  $\mathbf{U_1}$ = 7-6 = 1 and $\mathbf{U_2}$ = 14-6 = 8
  - Smallest of the 2 Us: U_1
  - U1 comparison to critical value + sample size → **p-value**

- **R tidyverse:** `wilcox_test(y~x)`



Comparison between 2 paired groups

Wilcoxon's signed-rank test



- Non-parametric equivalent of the paired *t*-test (ish)
- Information about the Mann-Whitney test also applies
- **How does the Wilcoxon's signed-rank test work?**

2+3=5/2=2.5: average rank

Before	After	Differences
9	3	-6
7	4	-3
10	4	-6
8	5	-3
5	6	1
8	2	-6
7	7	0
9	4	-5
10	5	-5

Abs. Diff.	Ranking	Ranks
0		
1	1	1
3	2	2.5
3	3	2.5
5	4	4.5
5	5	4.5
6	6	7
6	7	7
6	8	7

	Negative ranks	Positives ranks
		1
	-2.5	
	-2.5	
	-4.5	
	-4.5	
	-7	
	-7	
	-7	
Sum	-35	1

- Statistic of the Wilcoxon's signed-rank test: Sum of signed ranks = **W**
  - Here: W = -35 + 1 = -34
  - Statistic W + sample size → **p-value**

R: `wilcox_test(y~x, paired = TRUE)`

Comparison between more than 2 groups

One factor

Non-Parametric data

Non-parametric tests

Kruskal-Wallis and Friedman tests

- Non-parametric equivalent of the One-Way ANOVA (ish)
  - Data replaced by ranks
  - **Data exploration**
    - If data represent different distributions: comparison of said distributions
    - If original data come from similar distributions: comparison of the medians

- **Kruskal-Wallis**: independent measures
  - Statistic = **H**
- **Friedman**: repeated measures
  - Statistic = **Q** or **T1** or **FM**

- Post-hoc test associated with Kruskal-Wallis and Friedman: **Dunn's test**
  - Works pretty much like the Mann-Whitney test

Comparison between more than 2 groups

Independent: Kruskal-Wallis test

Actual values: n=15

No	Once	Twice
63	0	2239
-261	-652	171
-153	4724	40
-13	-2	1395
965	0	
	-86	

Ranks: 15

No	Once	Twice
10	7.5	14
2	1	11
3	15	9
5	6	13
12	7.5	
	4	
32	**41**	**47**

$$\mathbf{H} = \left[\frac{12}{n(n+1)}\sum_{j=1}^{c}\frac{T_j^2}{n_j}\right] - 3(n+1) \qquad \mathbf{H} = \left[\frac{12}{15(15+1)}\left(\frac{32^2}{5}+\frac{41^2}{6}+\frac{47^2}{4}\right)\right] - 3(15+1) = 3.868$$

Where:

- n = total sample size across all groups
- c = number of groups
- T_j = sum of ranks in the j^{th} group
- n_j = size of the j^{th} group

Interpretation of the test: H + degrees of freedom = p-value

`kruskal_test(y~x)` produces omnibus part of the analysis

`dunn_test(y~x)` produces pairwise comparisons results

```
# dunn.test package #
```

Comparison between more than 2 groups

Matched/repeated: Friedman test

Actual values

Violinists	Violin A	Violin B	Violin C
1	9	7	6
2	9.5	6.5	8
3	5	7	4
4	7.5	7.5	6
5	9.5	5	7
6	7.5	8	6.5
7	8	6	6
8	7	6.5	4
9	8.5	7	6.5
10	6	7	3

Ranks

Matched set of values

Violinists	Violin A	Violin B	Violin C
1	3	2	1
2	3	1	2
3	2	3	1
4	2.5	2.5	1
5	3	1	2
6	2	3	1
7	3	1.5	1.5
8	3	2	1
9	3	2	1
10	2	3	1
Sum	R_A = 26.5	R_B=21	R_C=12.5

- Basic idea: if the sums are very different (here R_A R_B and R_C) the p-value will be small.

$$Q\ or\ T1\ or\ FM\ or\ F = \left[\frac{12}{N \times k \times (k+1)}\right] \times \sum R^2 - [3 \times N \times (k+1)]$$

$$F = \left[\frac{12}{10 \times 3 \times (3+1)}\right] \times [26.5^2 + 21^2 + 12.5^2] - [3 \times 10 \times (3+1)]$$

$$F = \left[\frac{12}{120}\right] \times [702.25 + 441 + 156.25] - 120 = 9.95$$

Where:

- N = the number of subjects (violinists)
- k = number of groups (violins)
- R = sum of ranks in the group (e.g. R_A)

```
friedman_test(y~x|id)

wilcox_test(y~x,
        paired = TRUE,
        p.adjust.method =
        "bonferroni")
```

Interpretation of the test: Q or T1 or FM + df= p-value

Association between 2 continuous variables

Linear relationship

Non-Parametric data

Non-parametric tests
Spearman Correlation Coefficient

- Similar concepts as for the other non-parametric tests
- **ρ (rho)** is the equivalent of r and calculated in a similar way
- **Spearman's ρ** is Pearson's **r** applied on ranks

$$\rho = r_s = \frac{\text{Similarity}}{\text{Variability}} = \frac{COV_{R(x)R(y)}}{SD_{R(x)}SD_{R(y)}}$$

```
cor_test(method = "spearman")
```

Exercise 5

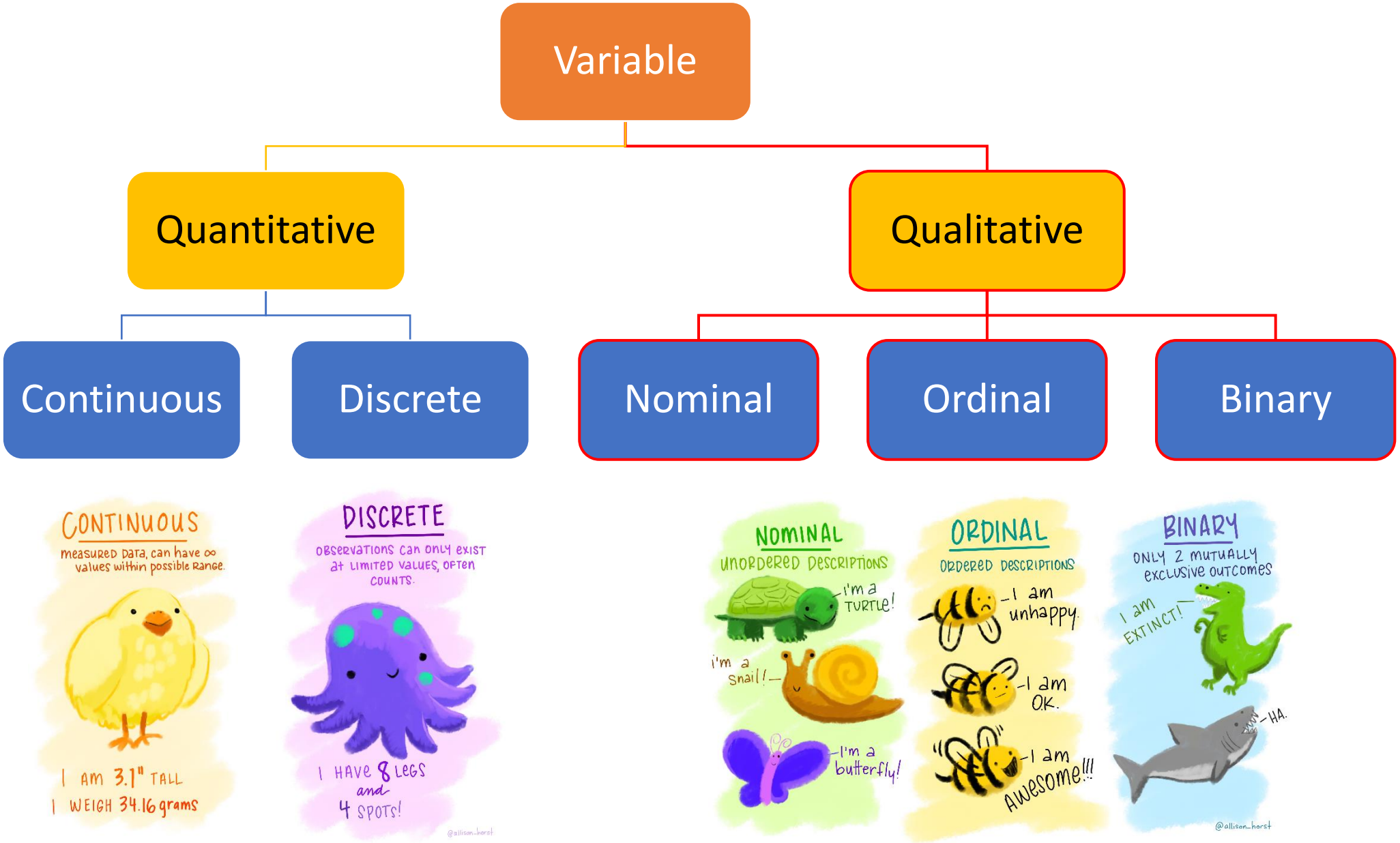
Babraham Institute

Analysis of Qualitative data

Hayley Carr & Anne Segonds-Pichon

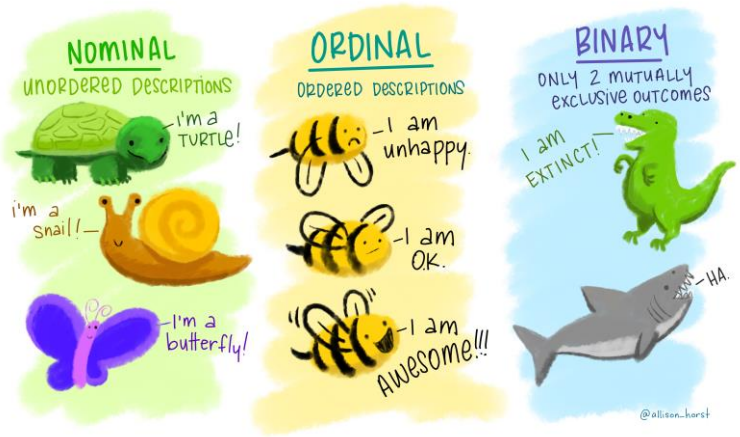
v2025-02

Babraham Bioinformatics



https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

Qualitative data



https://github.com/allisonhorst/stats-illustrations#other-stats-artwork

- Values taken = usually names (also nominal)
    - e.g. genotypes
- Can be numbers but not numerical
    - e.g. group number = numerical label but not unit of measurement
- Qualitative variable with intrinsic order in their categories = ordinal
    - e.g. low/medium/high
- Particular case: qualitative variable with 2 categories: binary or dichotomous
    - e.g. alive/dead or presence/absence

Comparison between 2 groups

Comparison between 2 proportions

Binary outcome

Chi-square and Fisher's tests

- Chi^2 is an approximation
- Chi^2 test very easy to calculate by hand but Fisher's very hard
- Often software will not perform a Fisher's test on tables > 2x2

- **Fisher's test** more accurate than Chi^2 test on **small samples**
- **Chi^2 test** generally preferable on **large samples**

- Chi^2 test assumptions:
  - 2x2 table: no expected count <5
  - Bigger tables: all expected > 1 and no more than 20% < 5

Chi-square test

- In a chi-square test, the observed frequencies for two or more groups are compared with expected frequencies by chance

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

  - O = Observed frequencies
  - E = Expected frequencies

Fisher's exact and Chi^2 tests

Example: cats and dogs.xlsx

- Cats and dogs trained to line dance
- 2 different rewards: food or affection
- **Question**: Is there a difference between the rewards?
  - Is there a significant relationship between the 2 variables?
  - Does the reward significantly affect the likelihood of dancing?

- To answer this type of question:
  - Contingency table
  - Fisher's exact or Chi^2 tests

	Food	Affection
Dance	?	?
No dance	?	?

- But first: **how many animals** do we need?
  - Power analysis

Exercise: Power calculation

- Preliminary results from a pilot study: **25%** of cats line-danced after having received affection as a reward vs. **70%** after having received food
  - **How many cats** do we need?

```
power.prop.test()
```

Exercise: Power calculation

- Preliminary results from a pilot study: **25%** line-danced after having received affection as a reward vs. **70%** after having received food.
- **How many cats** do we need?

```
power.prop.test(p1= 0.25, p2= 0.7, sig.level= 0.05, power= 0.8)
```

```
     Two-sample comparison of proportions power calculation

              n = 18.10585
             p1 = 0.25
             p2 = 0.7
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

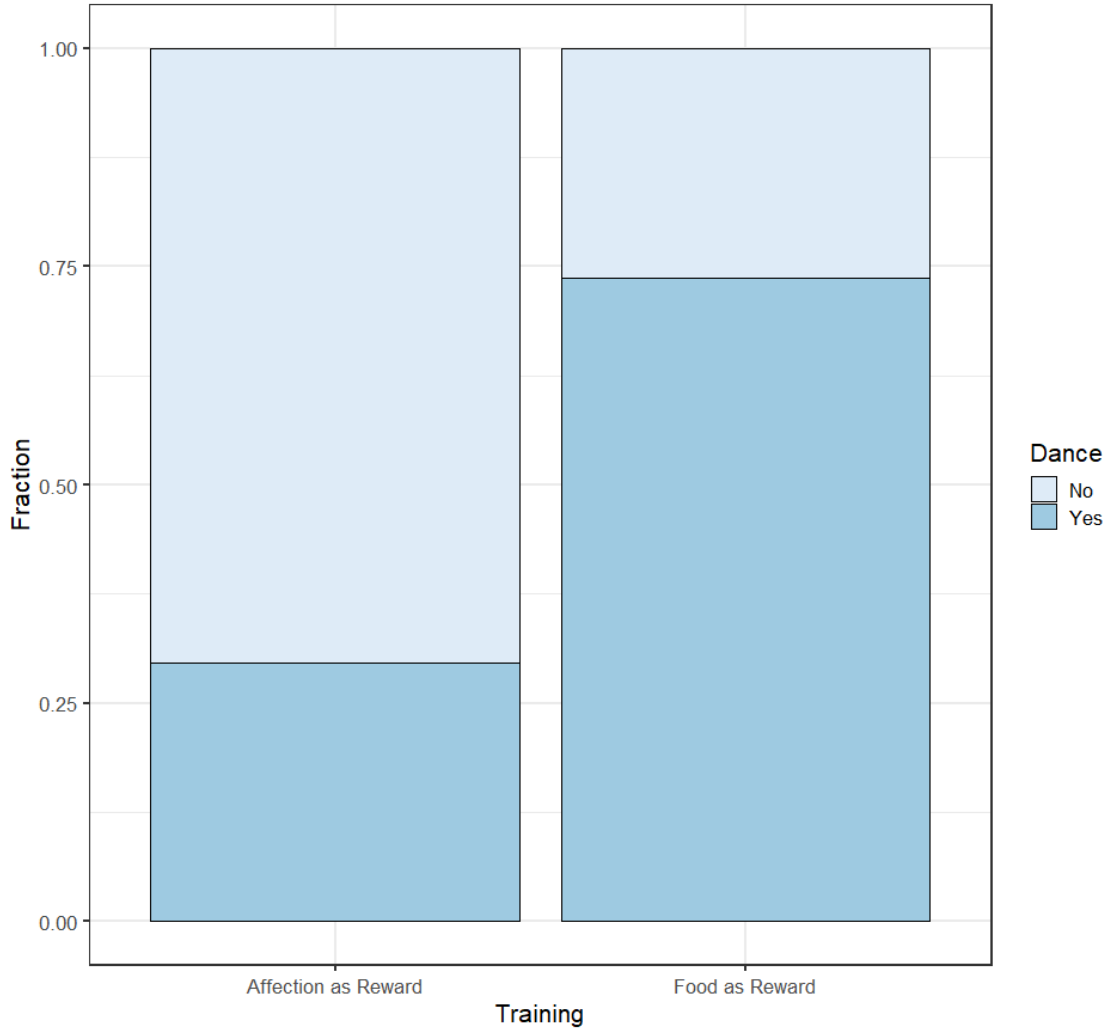
- Providing the effect size observed in the experiment is similar to the one observed in the pilot study, based on a significance threshold of 0.05, to achieve 80% power we will need **19 cats per group** (38 total) for a Fisher's exact test

Plot cats data

```
read_tsv("cats.dat") -> cats
cats
```

```
                Training Dance
1 Food as Reward   Yes
2 Food as Reward   Yes
3 Food as Reward   Yes
4 Food as Reward   Yes
5 Food as Reward   Yes
6 Food as Reward   Yes
```

```
cats %>%
  ggplot(aes(x=Training, fill=Dance))+
  geom_bar(position="fill", colour="black")+
  scale_fill_brewer(palette = 1)+
  ylab("Fraction")
```



How are the expected frequencies calculated?

$$\chi^2 = \sum \frac{(O - \boldsymbol{E})^2}{\boldsymbol{E}}$$

Example: expected frequency of cats line dancing after having received food as a reward

Direct counts approach:

Expected frequency

= (row total)*(column total)/grand total

= **32*32/68** = **15.1**

Probability approach: The Multiplicative Rule

Probability of line dancing: **32/68**

Probability of receiving food: **32/68**

Expected frequency:(**32/68**)*(**32/68**)=0.22: **22% of 68 = 15.1**

Observed frequencies

	Food	Affection	Total
Dance	**26**	6	**32**
No dance	6	30	36
Total	**32**	36	**68**

Expected frequencies

	Food	Affection
Dance	**15.1**	16.9
No dance	16.9	19.1

Chi² test

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Observed frequencies

	Food	Affection
Dance	26	6
No dance	6	30

Expected frequencies

	Food	Affection
Dance	15.1	16.9
No dance	16.9	19.1

$Chi^2 = (26-15.1)^2/15.1 + (6-16.9)^2/16.9 + (6-16.9)^2/16.9 + (30-19.1)^2/19.1 = \mathbf{28.4}$

Is 28.4 big enough for the test to be significant?

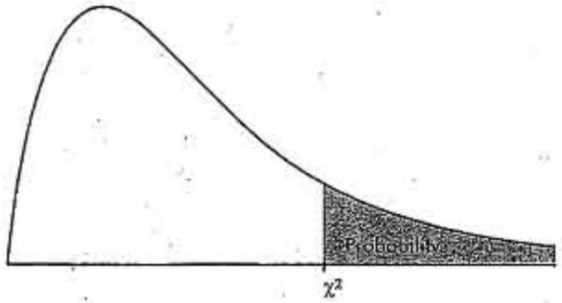
Is 28.4 big enough for the test to be significant?

The old fashion way

Degree of freedom: df

df = (row-1)(col-1)=1

	Food	Affection
Dance	26	6
No dance	6	30



Critical value

TABLE C: χ^2 CRITICAL VALUES

	Tail probability p								
df	.25	.20	.15	.10	.05	.025	.02	.01	.005
1	1.32	1.64	2.07	2.71	3.84	5.02	5.41	6.63	7.88
2	2.77	3.22	3.79	4.61	5.99	7.38	7.82	9.21	10.60
3	4.11	4.64	5.32	6.25	7.81	9.35	9.84	11.34	12.84
4	5.39	5.99	6.74	7.78	9.49	11.14	11.67	13.28	14.86
5	6.63	7.29	8.12	9.24	11.07	12.83	13.39	15.09	16.75
6	7.84	8.56	9.45	10.64	12.59	14.45	15.03	16.81	18.55
7	9.04	9.80	10.75	12.02	14.07	16.01	16.62	18.48	20.28
8	10.22	11.03	12.03	13.36	15.51	17.53	18.17	20.09	21.95
9	11.39	12.24	13.29	14.68	16.92	19.02	19.68	21.67	23.59
10	12.55	13.44	14.53	15.99	18.31	20.48	21.16	23.21	25.19

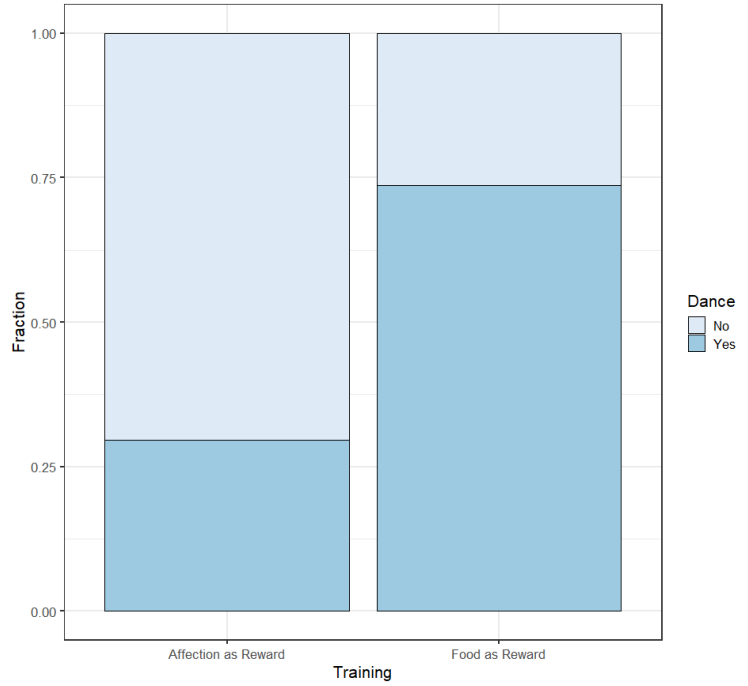
$\chi^2 = 28.4 > 3.84$ so Yes!

Prepare cats data for the stats

	Training	Dance
1	Food as Reward	Yes
2	Food as Reward	Yes
3	Food as Reward	Yes
4	Food as Reward	Yes
5	Food as Reward	Yes
6	Food as Reward	Yes

Training <chr>	No <int>	Yes <int>
Affection as Reward	114	48
Food as Reward	10	28

```
chisq_test()
fisher_test()
# rstatix package#
```




```
cats %>%
  group_by(Training, Dance) %>%
    count() %>%
    ungroup() %>%
    pivot_wider(names_from = Dance, values_from = n)-> cats.summary
```

Training <chr>	Dance <chr>	n <int>
Affection as Reward	No	114
Affection as Reward	Yes	48
Food as Reward	No	10
Food as Reward	Yes	28

Training <chr>	No <int>	Yes <int>
Affection as Reward	114	48
Food as Reward	10	28

```
cats.summary %>%
    select(No,Yes) %>%
    fisher_test()
```

n <int>	p <dbl>	p.signif <chr>
200	1.31e-06	****

Chi-square and Fisher's Exact tests

```
cats.summary %>%
    select(No,Yes) %>%
    fisher_test()
```

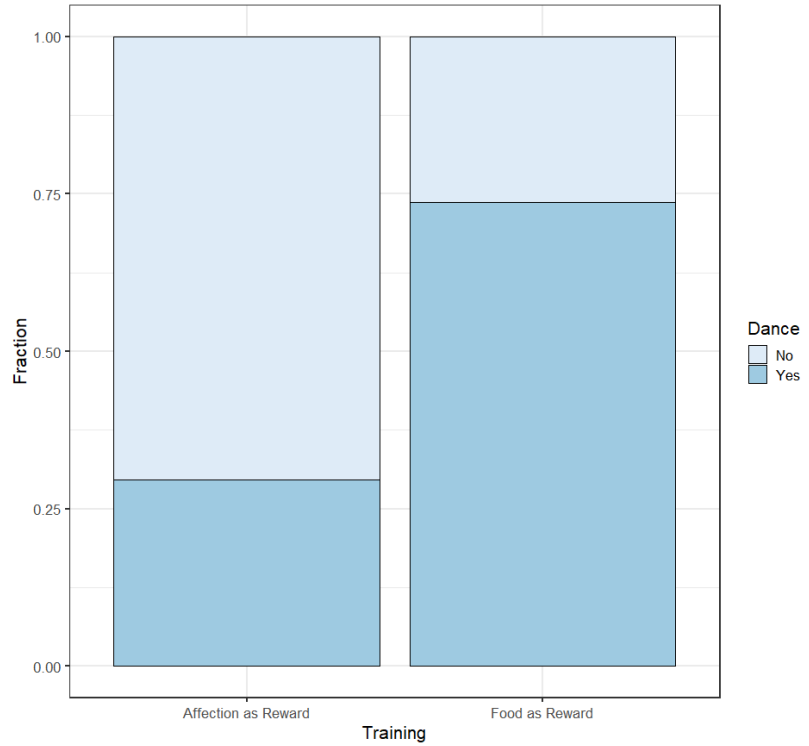
n <int>	p <dbl>	p.signif <chr>
200	1.31e-06	****

```
cats.summary %>%
    select(No,Yes) %>%
    chisq_test()
```

	n <int>	statistic <dbl>	p <dbl>	df <int>	method <chr>	p.signif <chr>
1	200	23.52028	1.24e-06	1	Chi-square test	****

```
cats.summary %>%
    select(No,Yes) %>%
    chisq_test(correct = FALSE)
```

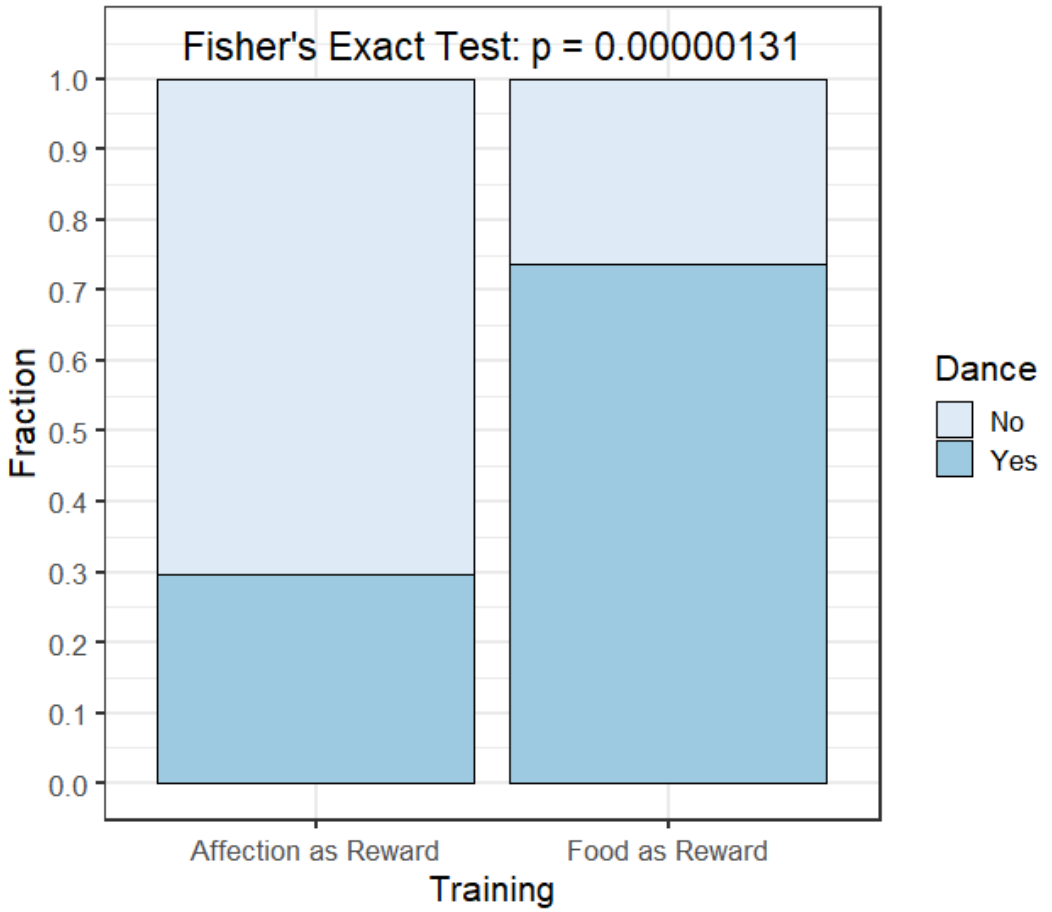
	n <int>	statistic <dbl>	p <dbl>	df <int>	method <chr>	p.signif <chr>
1	200	25.35569	4.77e-07	1	Chi-square test	****



Answer: Training significantly affects the likelihood of cats line dancing (p=4.8e-07).

Chi-square and Fisher's Exact tests

Stats on the graph

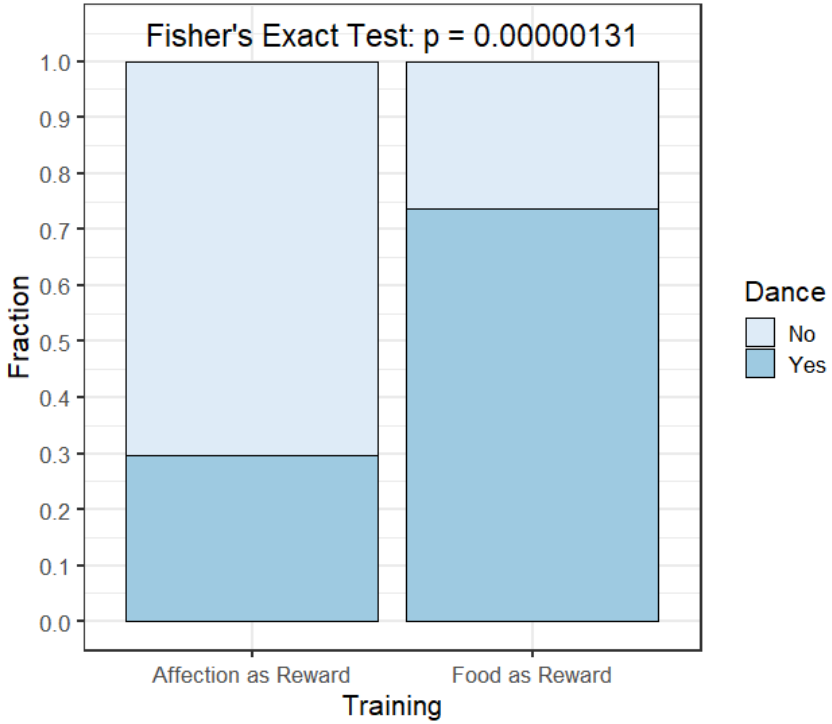



```
ggplot(cats, aes(x=Training, fill=Dance))+
  geom_bar(position="fill", colour="black")+
  scale_fill_brewer(palette = 1)+
  ylab("Fraction")+
  scale_y_continuous(breaks=seq(from =0, by=0.1, to=1.05), limits = c(0,1.05))+
  annotate("text", label="Fisher's Exact Test: p = 0.00000131", x=1.5, y=1.05, size=6)
```

Fisher's exact and Chi^2 tests

Beyond significance

- Important things to remember:
  - Qualitative data can be presented as percentages but the **tests should always be run on actual counts**
    - Power!
  - A p-value should always be interpreted in the **context of the experiment**
    - Power!



Exercise 6

Babraham Institute

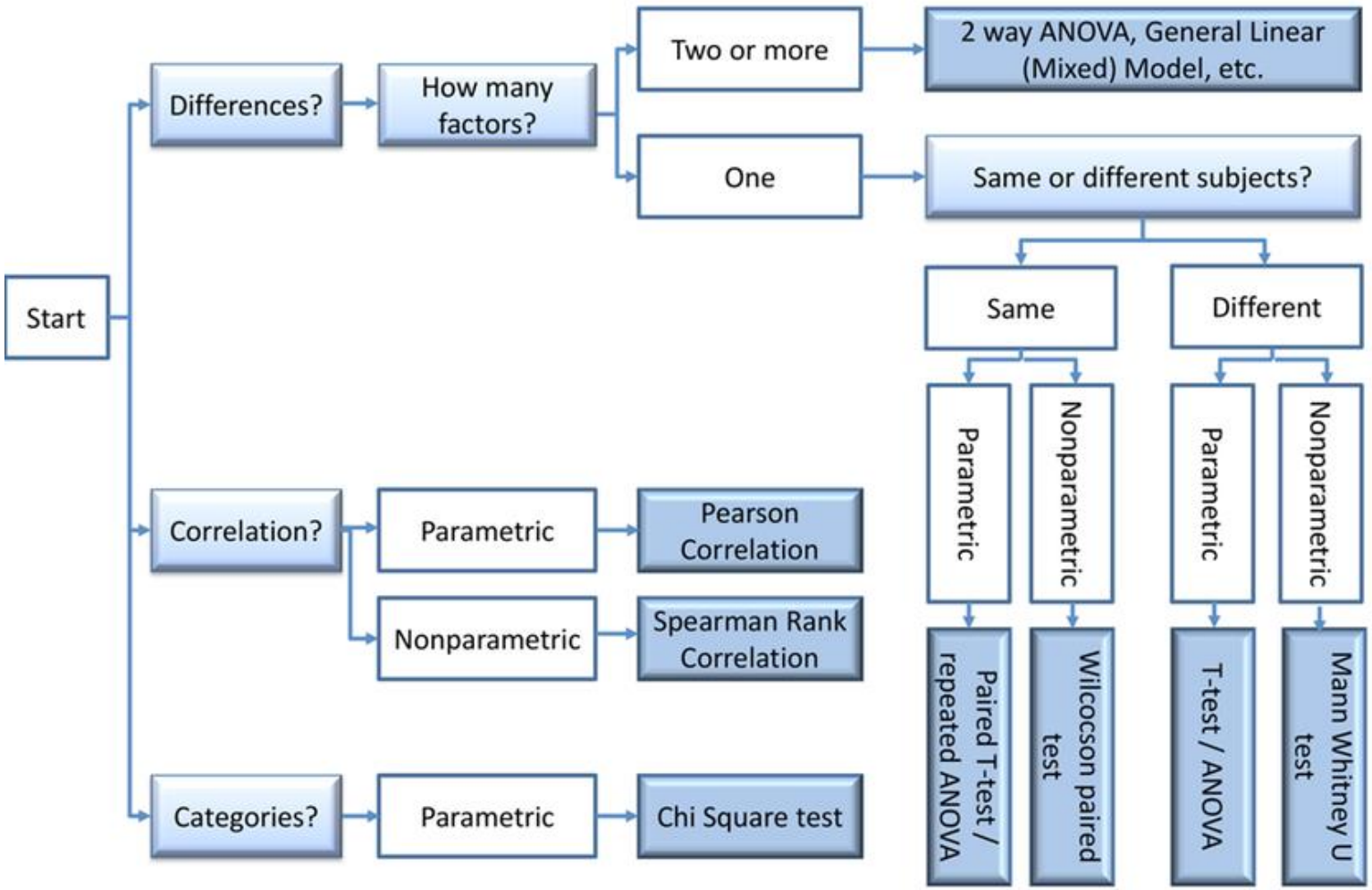
Decision trees & resources

Hayley Carr

v2024-05

Babraham Bioinformatics

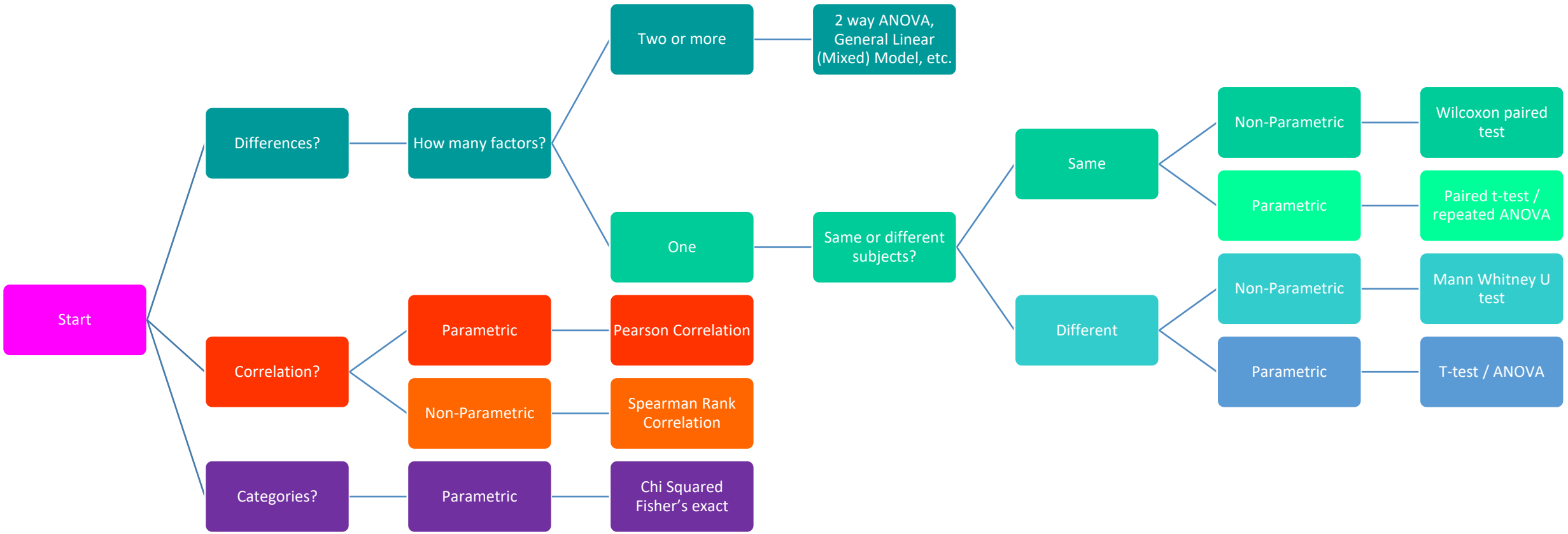
Choosing a test: Flow charts



Babraham Bioinformatics

Statistics Decision tree

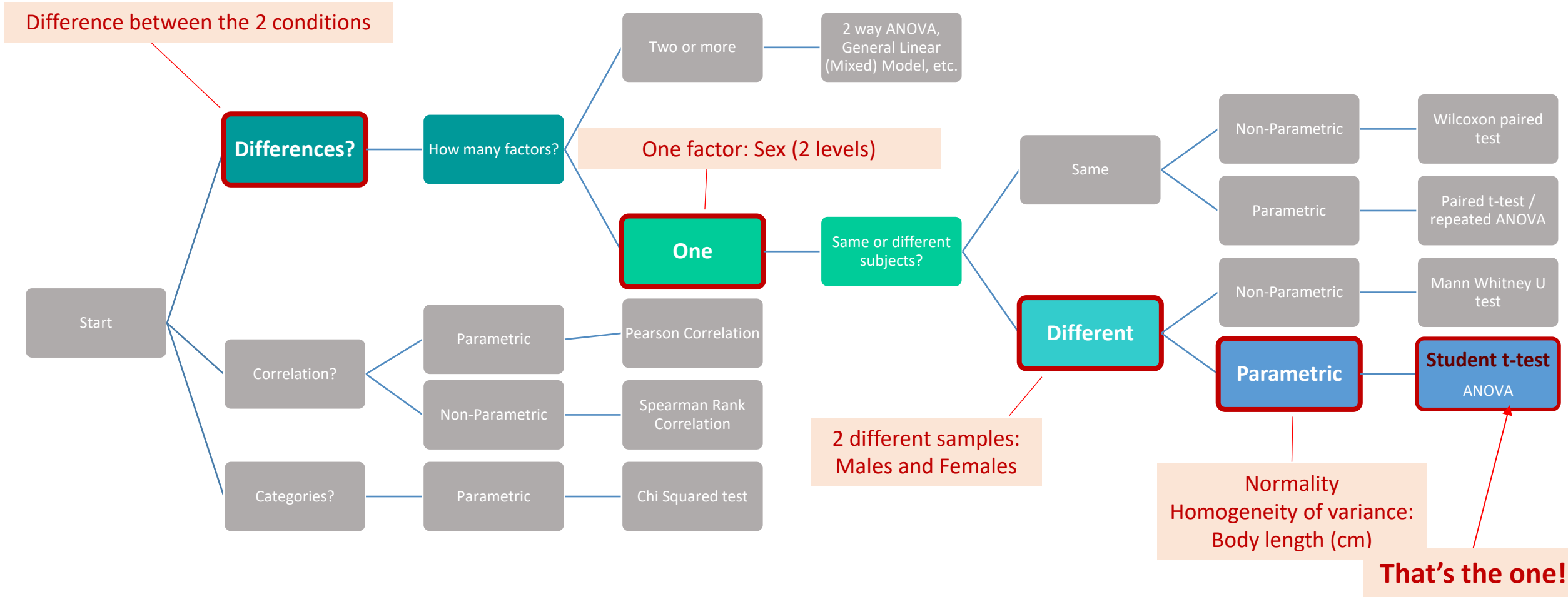
Anne Segonds-Pichon

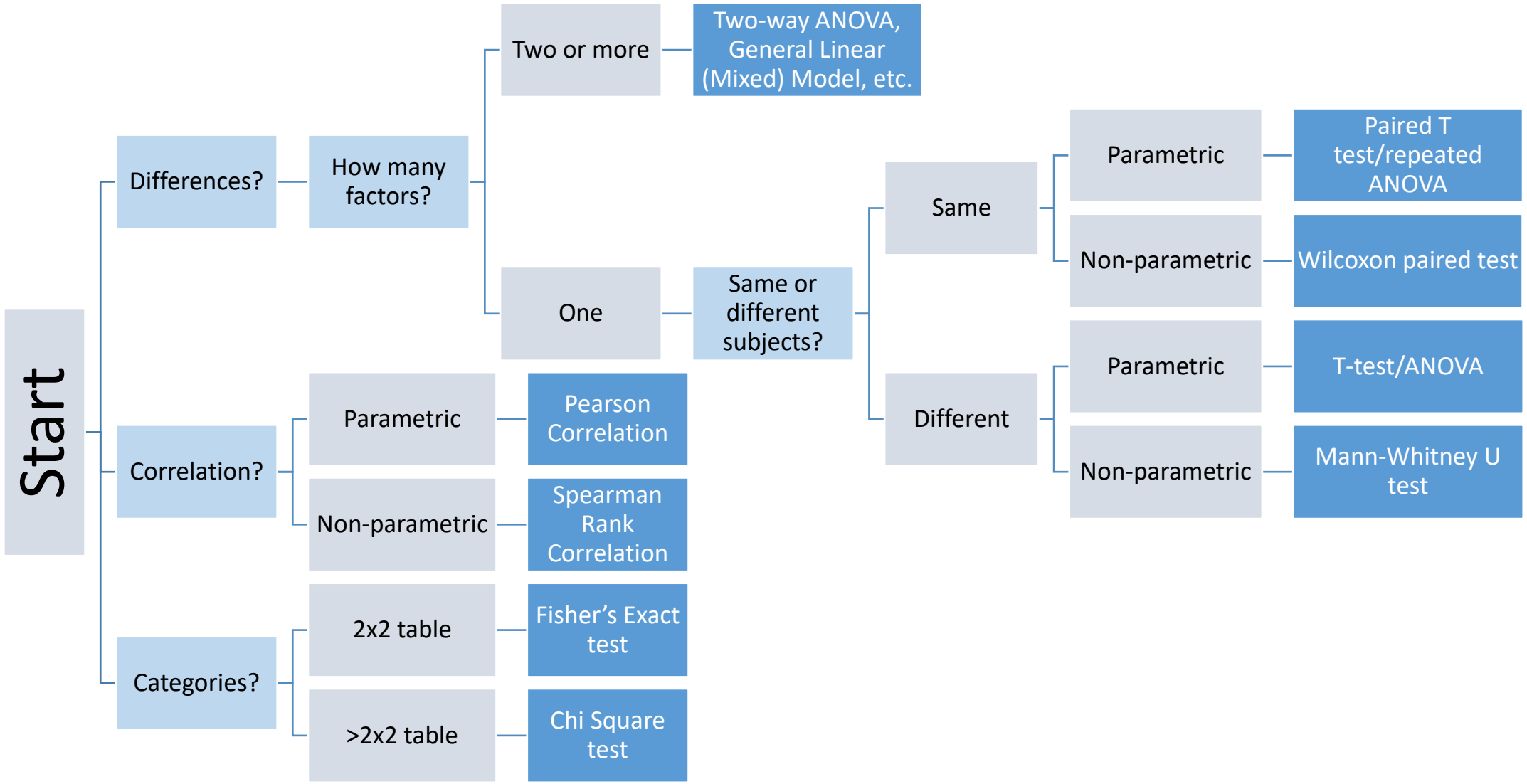


Statistics Decision tree

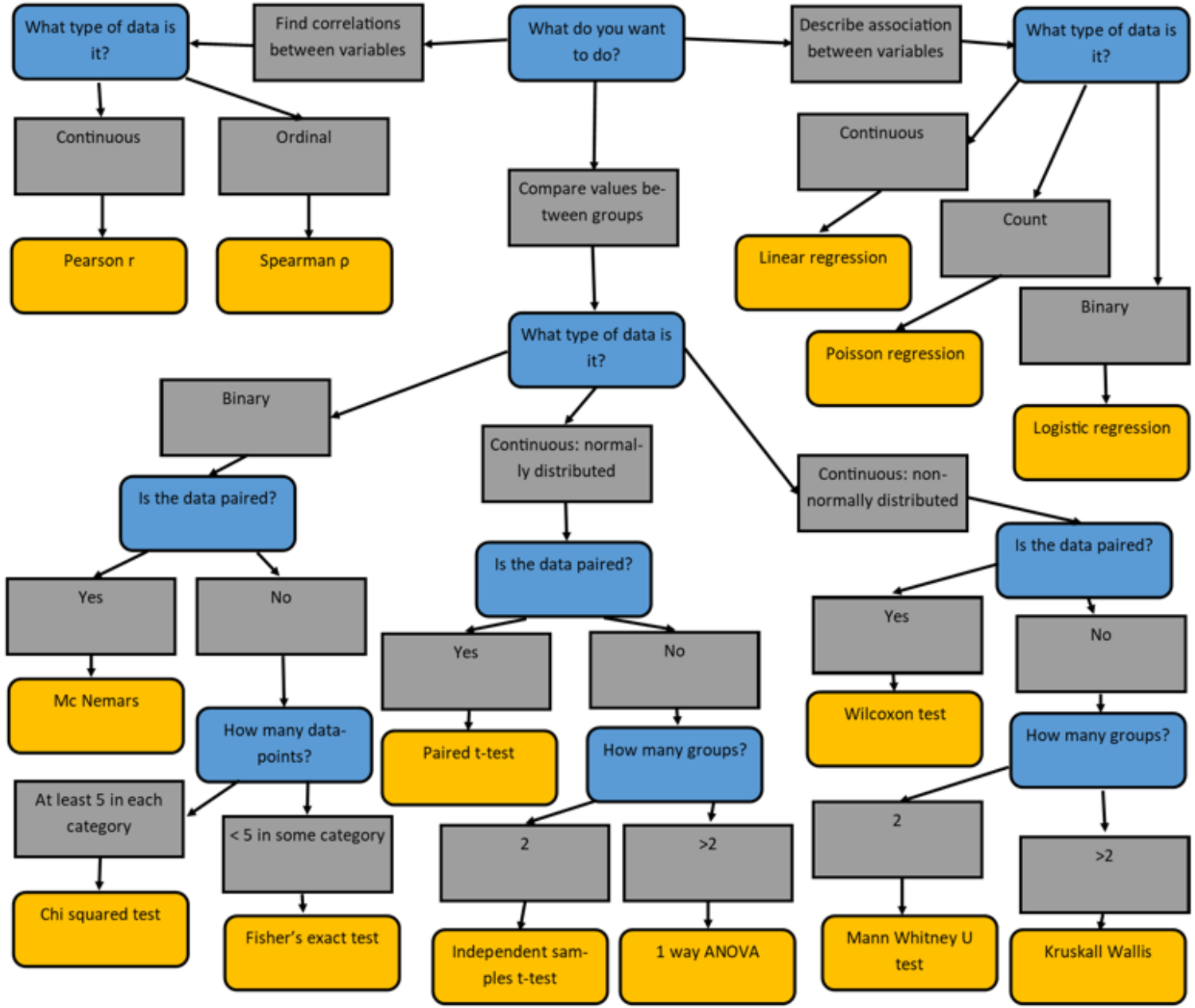
Anne Segonds-Pichon

Is there a difference between males and females coyotes in the body length?

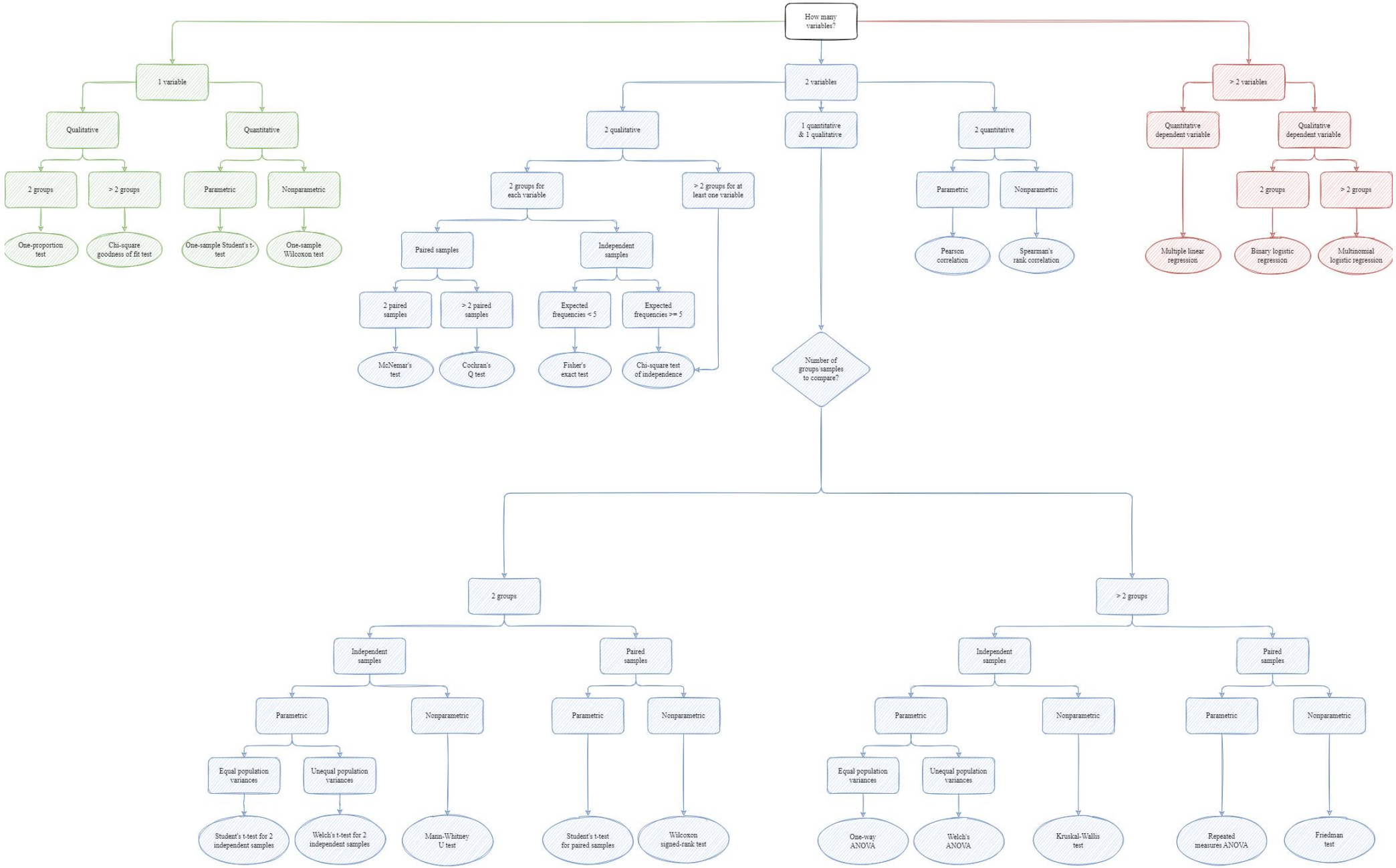



Start
Differences?
How many factors?
Two or more
Two-way ANOVA, General Linear (Mixed) Model, etc.
One
Same or different subjects?
Same
Parametric
Paired T test/repeated ANOVA
Non-parametric
Wilcoxon paired test
Different
Parametric
T-test/ANOVA
Non-parametric
Mann-Whitney U test
Correlation?
Parametric
Pearson Correlation
Non-parametric
Spearman Rank Correlation
Categories?
2x2 table
Fisher's Exact test
>2x2 table
Chi Square test

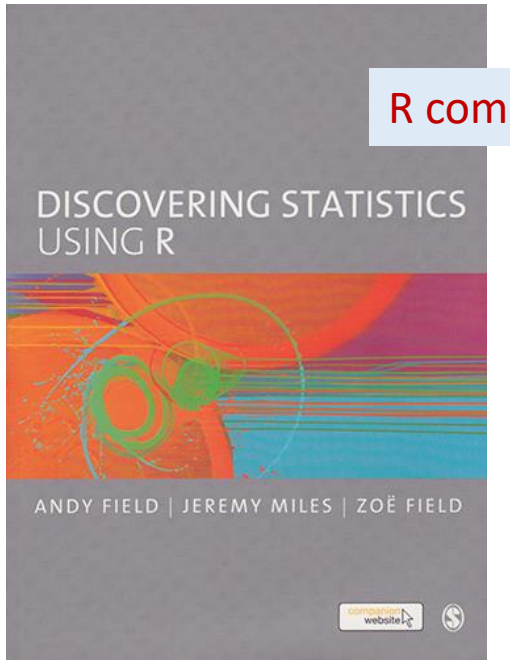
Which statistical test should I use?



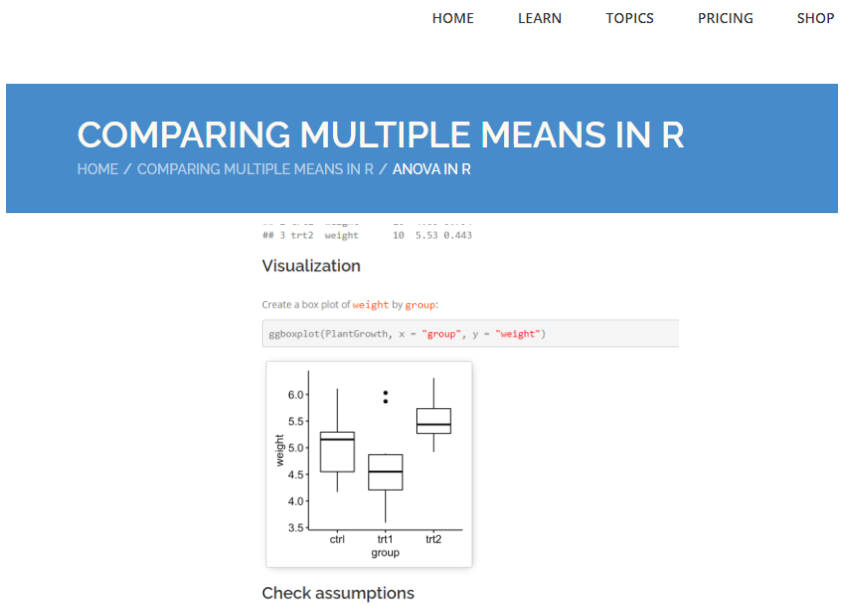
What statistical test should I do?



Statistics resources: R



R commander



https://www.datanovia.com/en/lessons/

https://rpkgs.datanovia.com/rstatix/

rstatix

Provides a simple and intuitive pipe-friendly framework, coherent with the 'tidyverse' design philosophy, for performing basic statistical tests, including t-test, Wilcoxon test, ANOVA, Kruskal-Wallis and correlation analyses.

The output of each test is automatically transformed into a tidy data frame to facilitate visualization.

Additional functions are available for reshaping, reordering, manipulating and visualizing correlation matrix. Functions are also included to facilitate the analysis of factorial experiments, including purely 'within-Ss' designs (repeated measures), purely 'between-Ss' designs, and mixed 'within-and-between-Ss' designs.

It's also possible to compute several effect size metrics, including "eta squared" for ANOVA, "Cohen's d" for t-test and "Cramer's V" for the association between categorical variables. The package contains helper functions for identifying univariate and multivariate outliers, assessing normality and homogeneity of variances.

Key functions

Descriptive statistics

- `get_summary_stats()` : Compute summary statistics for one or multiple numeric variables. Can handle grouped data.
- `freq_table()` : Compute frequency table of categorical variables.
- `get_mode()` : Compute the mode of a vector, that is the most frequent values.
- `identify_outliers()` : Detect univariate outliers using boxplot methods.
- `mahalanobis_distance()` : Compute Mahalanobis Distance and Flag Multivariate Outliers.
- `shapiro_test()` and `mshapiro_test()` : Univariate and multivariate Shapiro-Wilk normality test.

Comparing means

- `t_test()` : perform one-sample, two-sample and pairwise t-tests
- `wilcox_test()` : perform one-sample, two-sample and pairwise Wilcoxon tests
- `sign_test()` : perform sign test to determine whether there is a median difference between paired or matched observations.
- `anova_test()` : an easy-to-use wrapper around `car::Anova()` to perform different types of ANOVA tests, including **independent measures ANOVA**, **repeated measures ANOVA** and **mixed ANOVA**.
- `get_anova_test_table()` : extract ANOVA table from `anova_test()` results. Can apply sphericity correction automatically in the case of within-subject (repeated measures) designs. - `welch_anova_test()` : Welch one-Way ANOVA test. A pipe-friendly wrapper around the base function `stats::oneway.test()` . This is is an alternative to the standard one-way ANOVA in the situation where the homogeneity of variance assumption is violated.
- `kruskal_test()` : perform kruskal-wallis rank sum test
- `friedman_test()` : Provides a pipe-friendly framework to perform a Friedman rank sum test, which is the non-parametric alternative to the one-way repeated measures ANOVA test.
- `get_comparisons()` : Create a list of possible pairwise comparisons between groups.
- `get_pvalue_position` : autocompute p-value positions for plotting significance using ggplot2.

Core R

https://rcompanion.org/rcompanion/a_02.html

Statistics resources

https://www.nature.com/collections/qghhqm

Not always the friendliest, but covers lots of relevant topics

Hayley Carr (i.e. me!):
hayley.carr@babraham.ac.uk

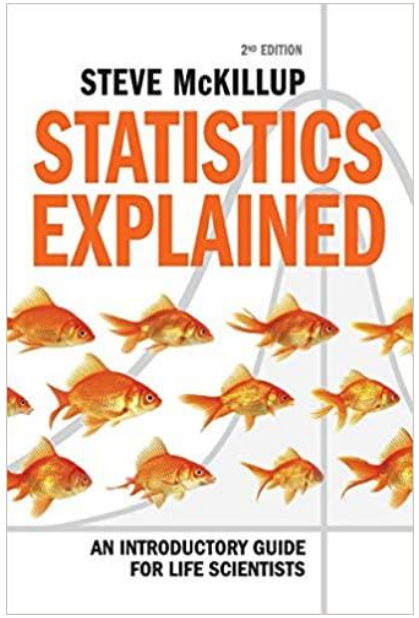
Collection | **09 May 2017**

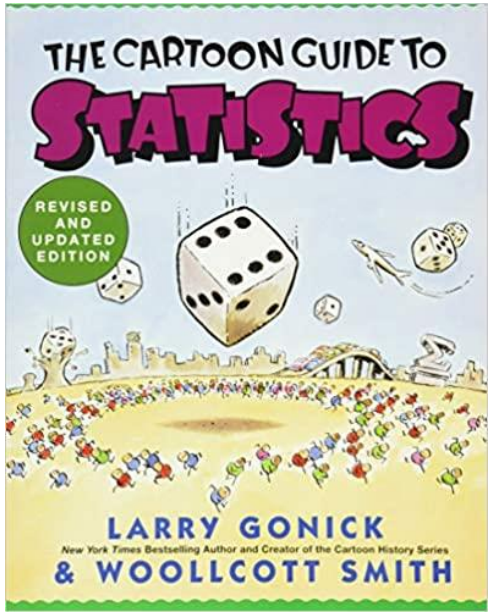
Statistics for Biologists

There is no disputing the importance of statistical analysis in biological research, but too often it is considered only after an experiment is completed, when it may be too late.

This collection highlights important statistical issues that biologists should be aware of and provides practical advice to help them improve the rigor of their work.

Nature Methods' **Points of Significance** column on statistics explains many key statistical and experimental design concepts. **Other resources** include an online plotting tool and links to statistics guides from other publishers.





Exercise 7