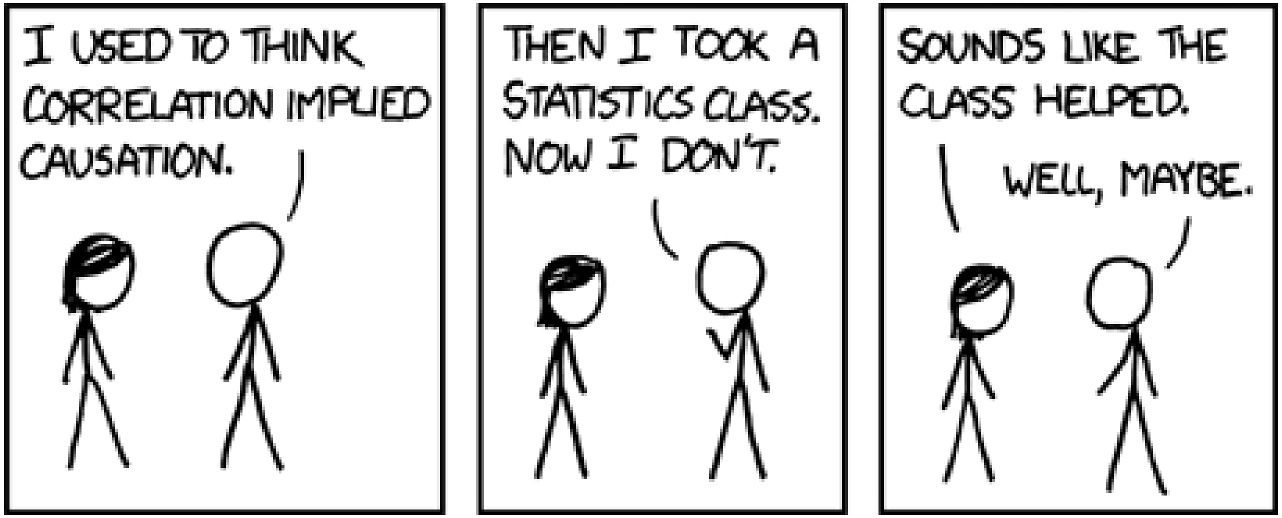
Statistical tests and effect size

Ivano Malavolta


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

VU VRIJE UNIVERSITEIT AMSTERDAM

LOOKING FURTHER

Roadmap

- Warm up
- Check for normality
- Main statistical tests
- p-value corrections
- Effect size

Context and assumptions for this course

- We focus on quantitative **variables** only
  - nominal
  - ordinal
  - interval
  - ratio
- **Factors** are nominal or ordinal
- Dependent variables typically ratios

  Our statistical tests detect differences between the **means** of the dependent variable

- Treatments are **fixed** a priori

Tasks for data analysis

1. Descriptive statistics
   - for understanding the “shape” of collected data
2. Select statistical test
   - according to collected metrics and data distribution
   - this might involve also data transformation
3. Hypothesis testing
   - for providing evidence about your findings
     i. statistical significance
4. Effect size calculation
   - for understanding if your (statistically significant) results are actually relevant in practice

What is a statistical test?

- Calculation of a *sample statistic* assuming that the null hypothesis is true

- The calculated value of the statistic has a certain *probability*, given that the null hypothesis is true (*p-value*)

First choice: parametric VS non-parametric tests

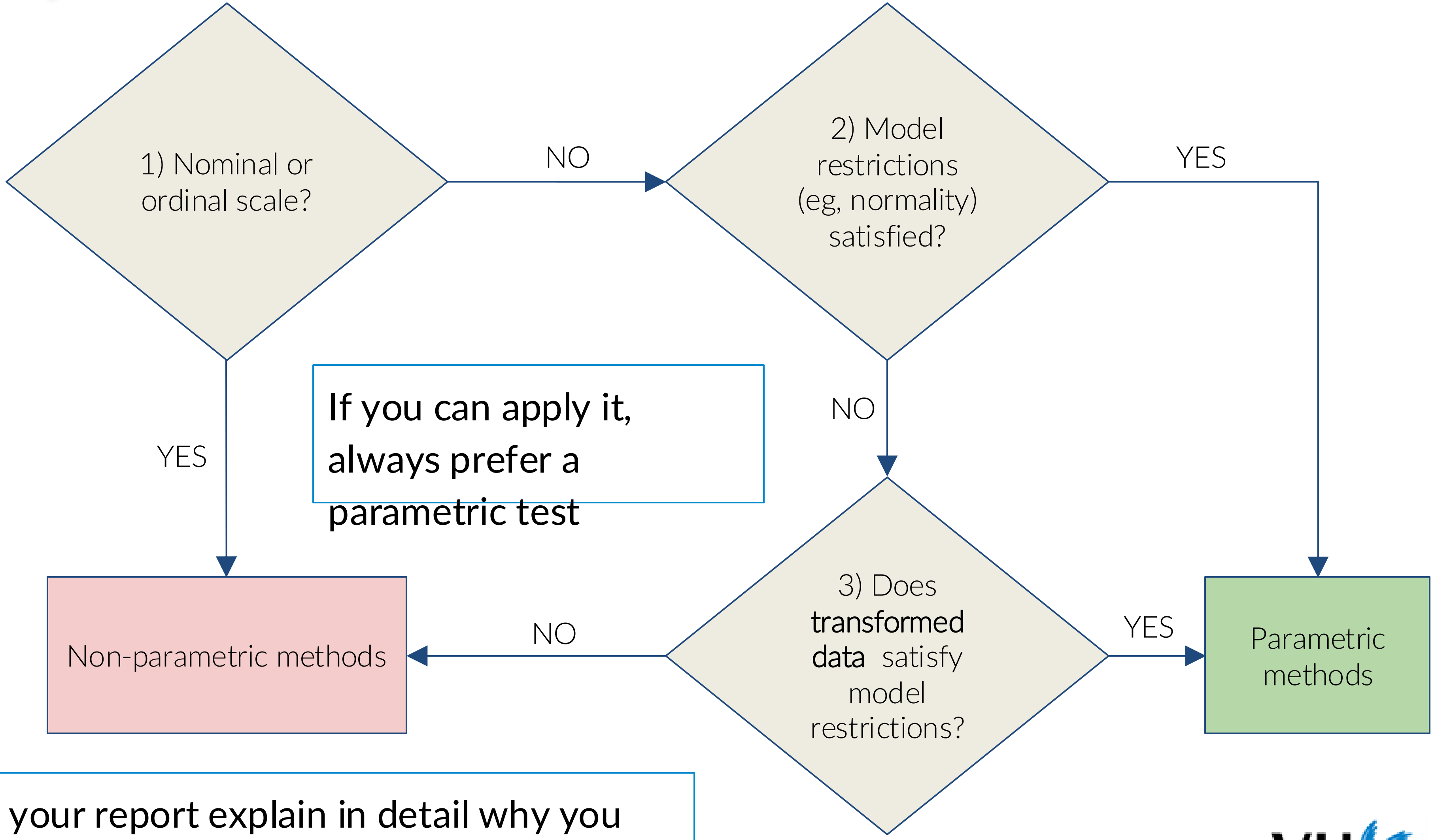
- **Parametric tests** assume specific characteristics about the data
  - typically, normal distribution
  - more powerful

    → lower chances of having Type II errors

- **Non-parametric tests** do not make any assumption about the data
  - more general
  - less powerful

    → larger samples are needed

How to choose?



1) Nominal or ordinal scale?

NO

2) Model restrictions (eg, normality) satisfied?

YES

YES

NO

Non-parametric methods

NO

3) Does **transformed data** satisfy model restrictions?

YES

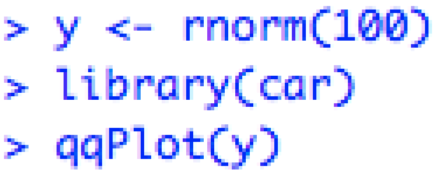
Parametric methods

If you can apply it, always prefer a parametric test

In your report explain in detail why you choose a specific test!

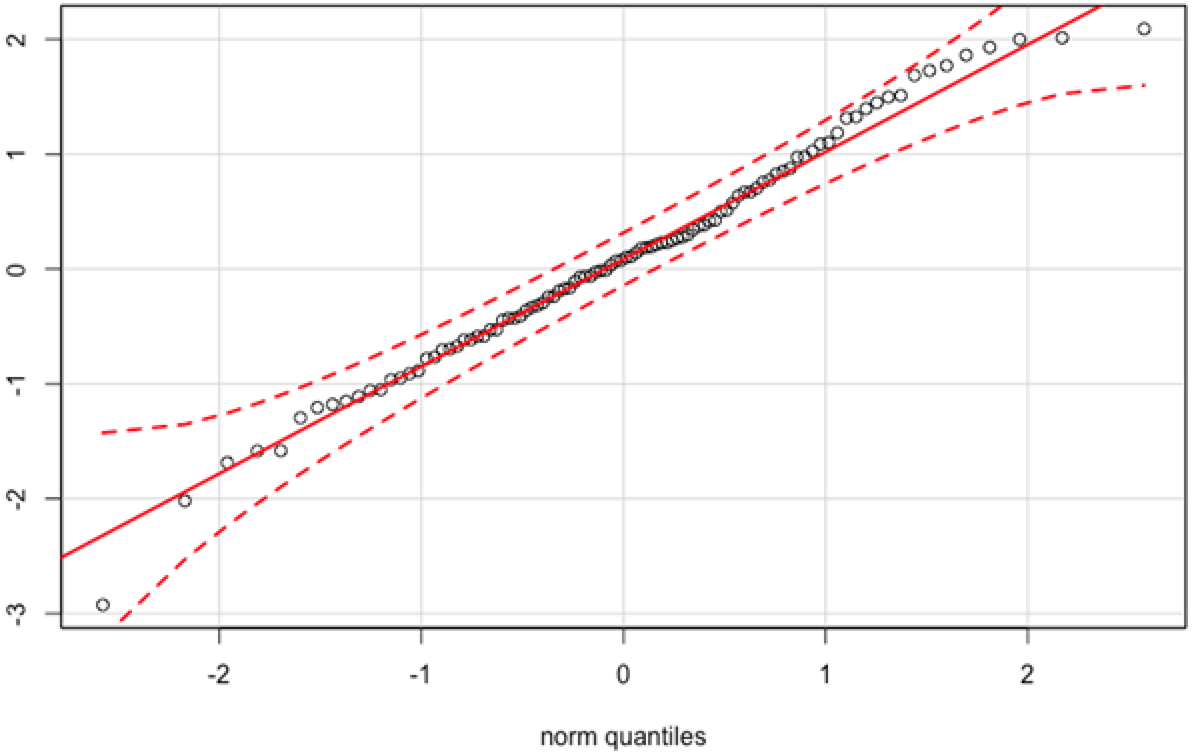
Check for normality

Graphical check (Q-Q plot)



```
> y <- rnorm(100)
> library(car)
> qqPlot(y)
```

qqplotr: R library for Q-Q plots



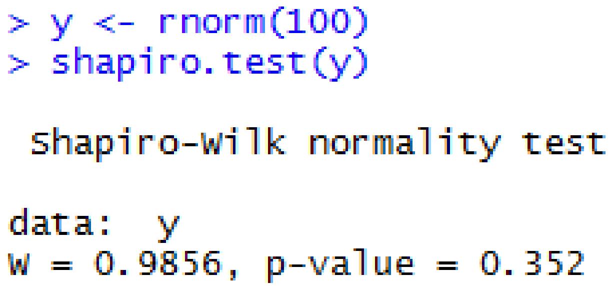
Normality tests

- Normality tests
  - H_0: sample is drawn from a normal distribution

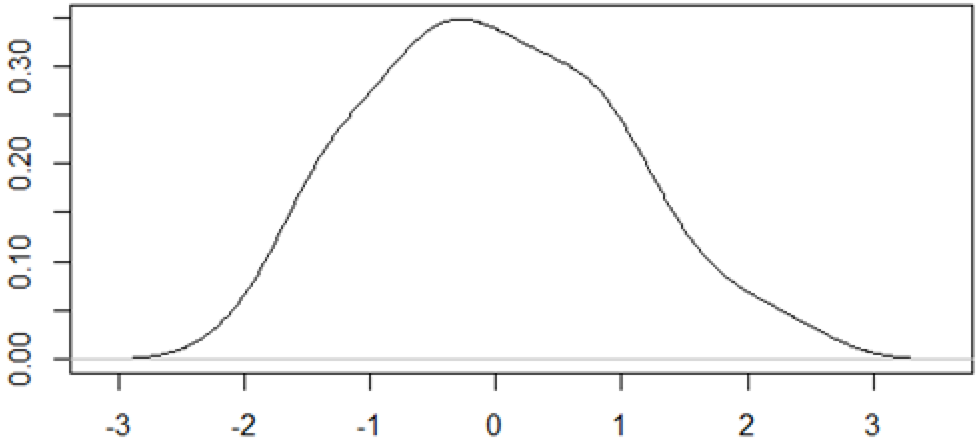
- Shapiro-Wilk test (AKA Shapiro-Wilk's W)

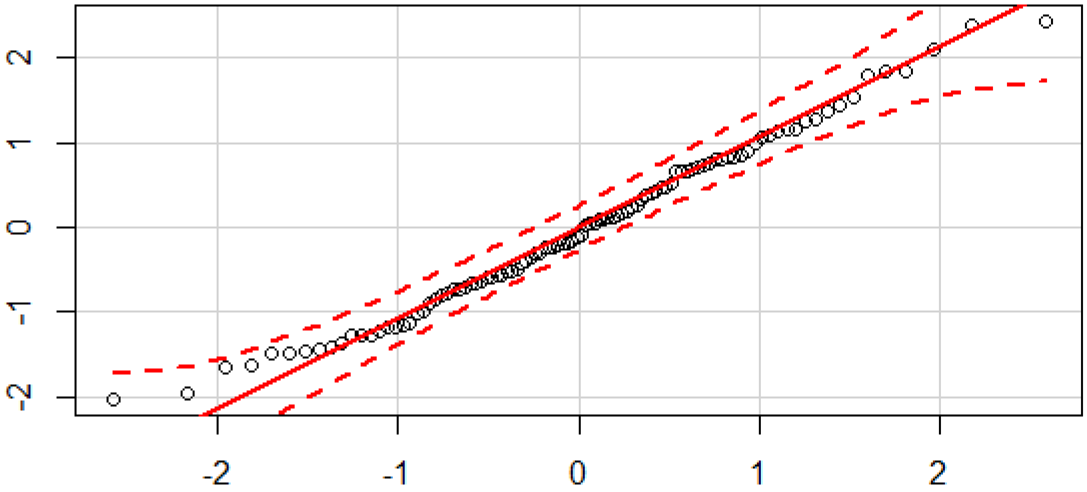
- If p-value <α for a given sample, we can conclude data is **NOT** normally distributed

Shapiro-Wilk test

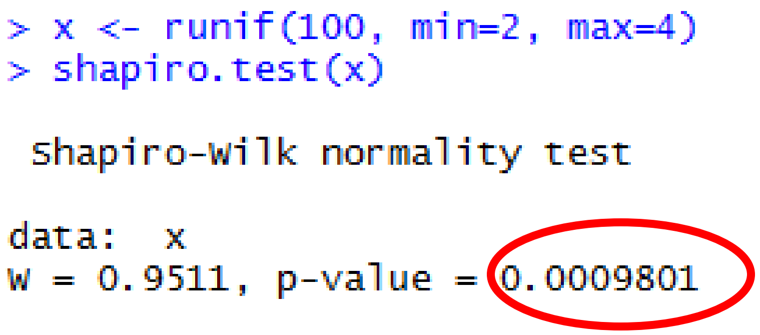


```
> y <- rnorm(100)
> shapiro.test(y)

 Shapiro-Wilk normality test

data:  y
W = 0.9856, p-value = 0.352
```

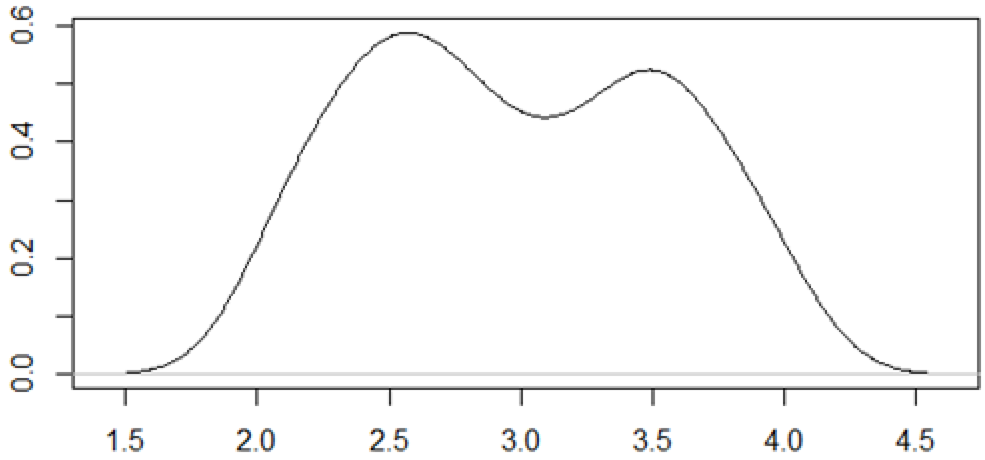


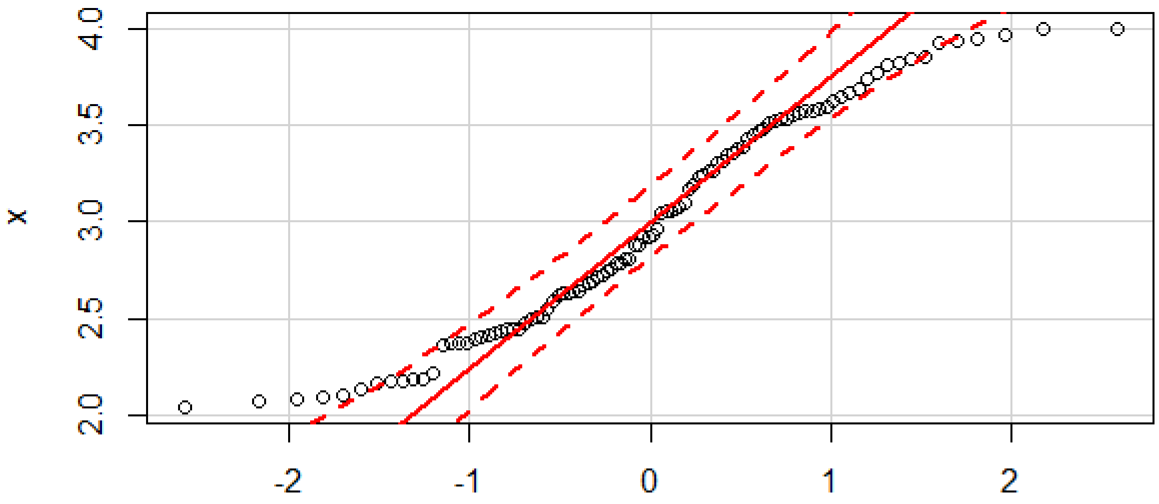


Shapiro-Wilk test



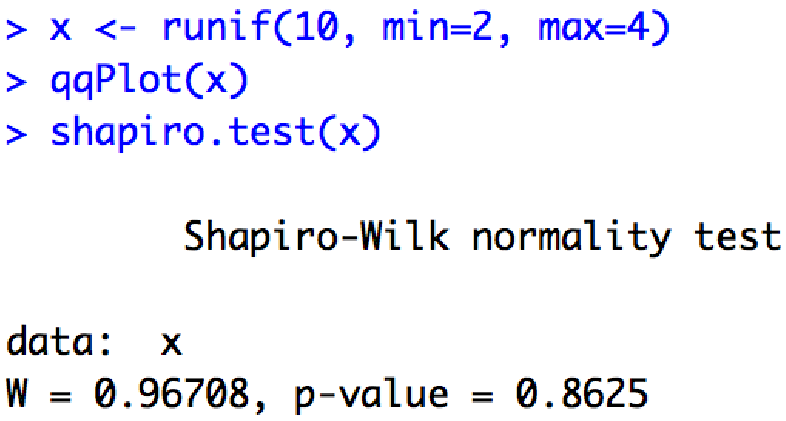
```
> x <- runif(100, min=2, max=4)
> shapiro.test(x)

 Shapiro-Wilk normality test

data:  x
W = 0.9511, p-value = 0.0009801
```



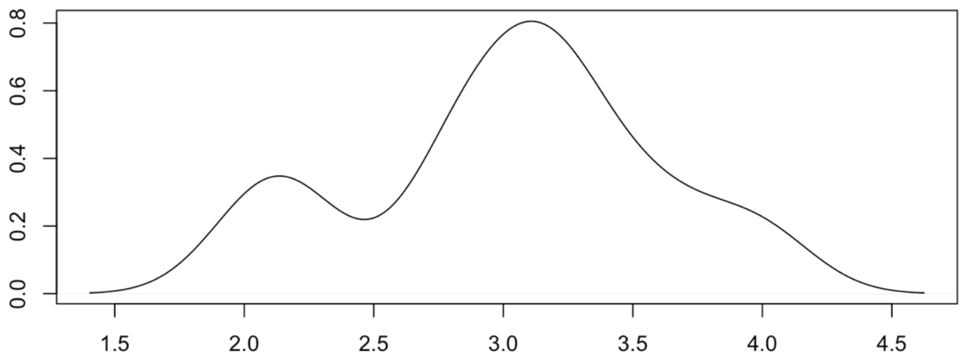


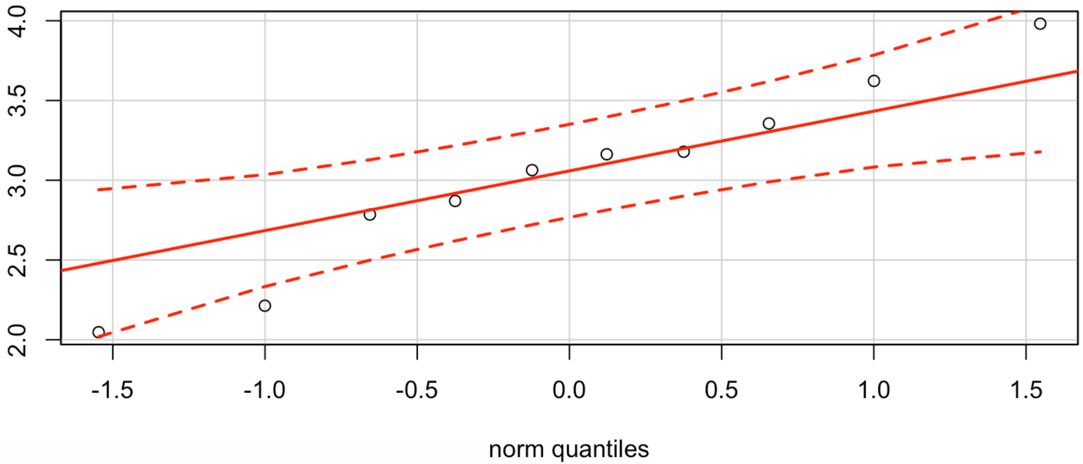
Shapiro-Wilk test

- Warning: Shapiro-Wilk is **not** robust for small samples!
  - Additional verification (e.g. via Q-Q plot) is always needed



```
> x <- runif(10, min=2, max=4)
> qqPlot(x)
> shapiro.test(x)

        Shapiro-Wilk normality test

data:  x
W = 0.96708, p-value = 0.8625
```





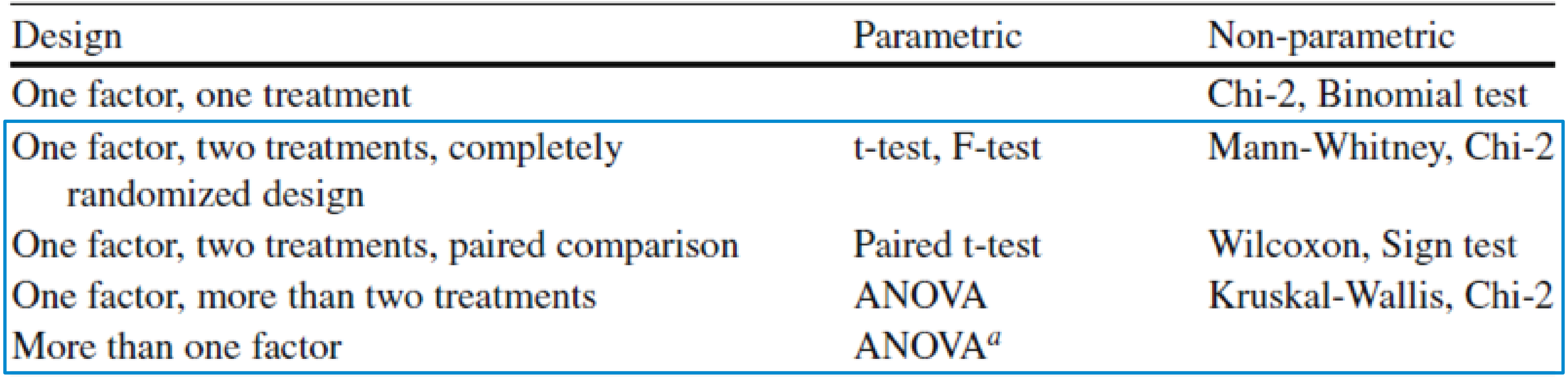
Inspiration for checking assumptions

Check the papers EASE_2020 and MobileSoft_2020 on Canvas

A nice online resource is also available here: https://www.datanovia.com/en/lessons/t-test-in-r/#assumptions-and-preliminary-tests-1

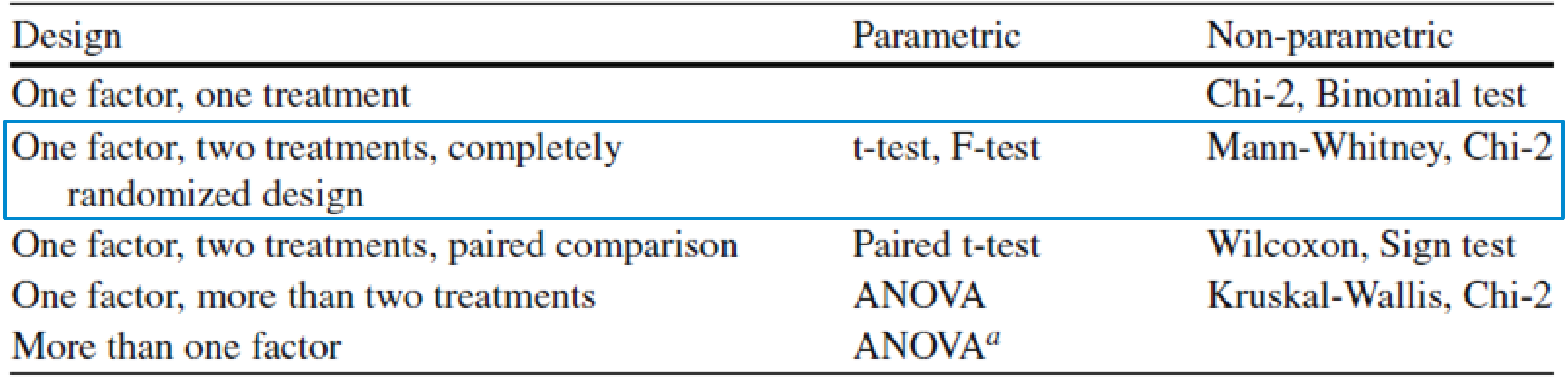
Main statistical tests

Statistical tests VS experiment design



Design	Parametric	Non-parametric
One factor, one treatment		Chi-2, Binomial test
One factor, two treatments, completely randomized design	t-test, F-test	Mann-Whitney, Chi-2
One factor, two treatments, paired comparison	Paired t-test	Wilcoxon, Sign test
One factor, more than two treatments	ANOVA	Kruskal-Wallis, Chi-2
More than one factor	ANOVA[a]	

One factor - 2 treatments - random design



Design	Parametric	Non-parametric
One factor, one treatment		Chi-2, Binomial test
One factor, two treatments, completely randomized design	t-test, F-test	Mann-Whitney, Chi-2
One factor, two treatments, paired comparison	Paired t-test	Wilcoxon, Sign test
One factor, more than two treatments	ANOVA	Kruskal-Wallis, Chi-2
More than one factor	ANOVA[a]	

t-Test

Parametric

Goal: compare independent samples

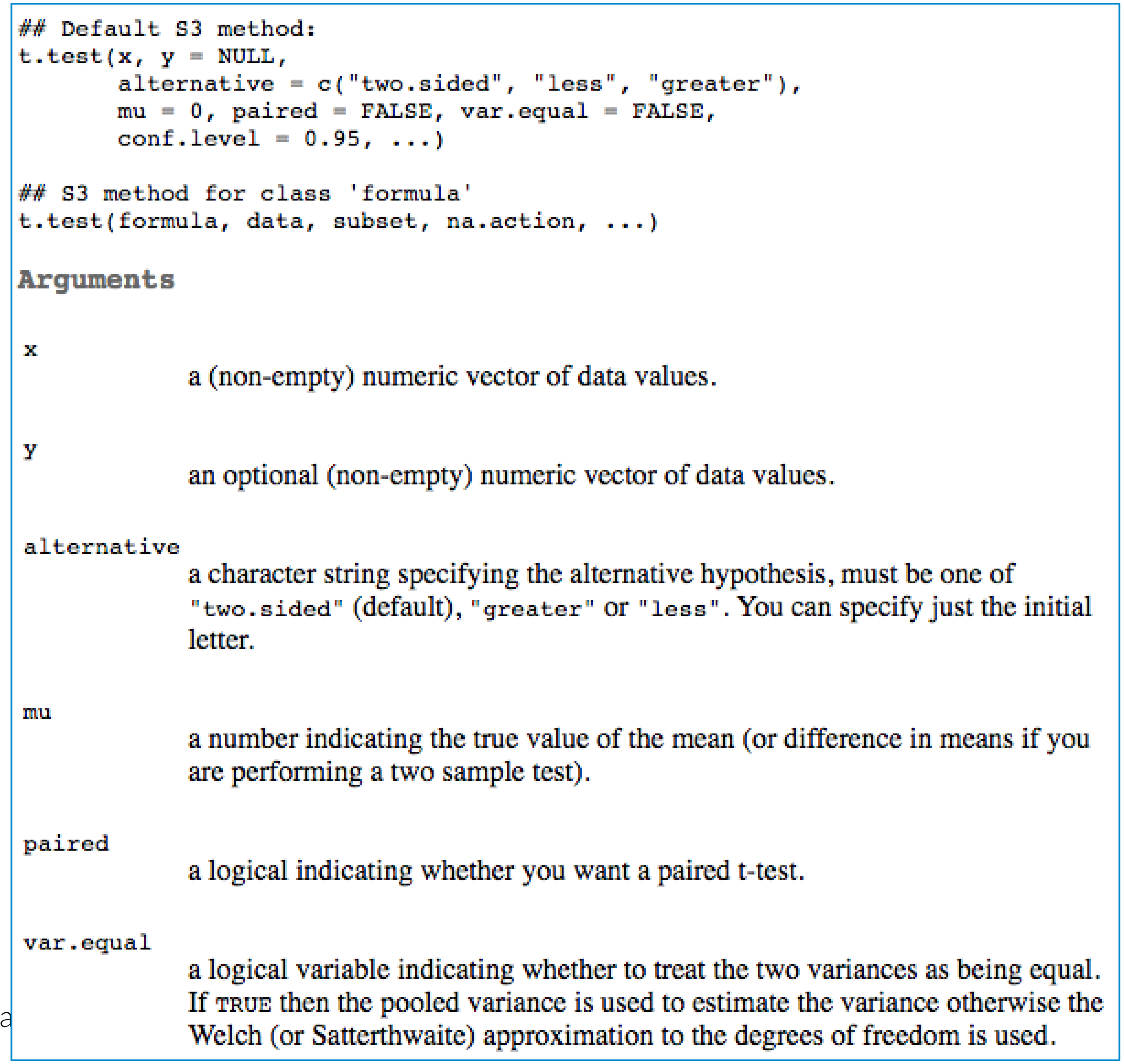
- ○ Values of the dependent variable obtained with different treatments
- ○ For each treatment you are measuring different subjects

Hypotheses:

- Two-tailed
  - ○ H_0: $\mu_2 = \mu_1$ H_a: $\mu_2 \neq \mu_1$
- One-tailed (alternative: greater)
  - ○ H_0: $\mu_2 = \mu_1$ H_a: $\mu_2 > \mu_1$
- One-tailed (alternative: less)
  - ○ H_0: $\mu_2 = \mu_1$ H_a: $\mu_2 < \mu_1$

- More powerful
- Cannot say anything in the opposite direction

t-Test in R



```
## Default S3 method:
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)

## S3 method for class 'formula'
t.test(formula, data, subset, na.action, ...)
```

Arguments

`x`

a (non-empty) numeric vector of data values.

`y`

an optional (non-empty) numeric vector of data values.

`alternative`

a character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`. You can specify just the initial letter.

`mu`

a number indicating the true value of the mean (or difference in means if you are performing a two sample test).

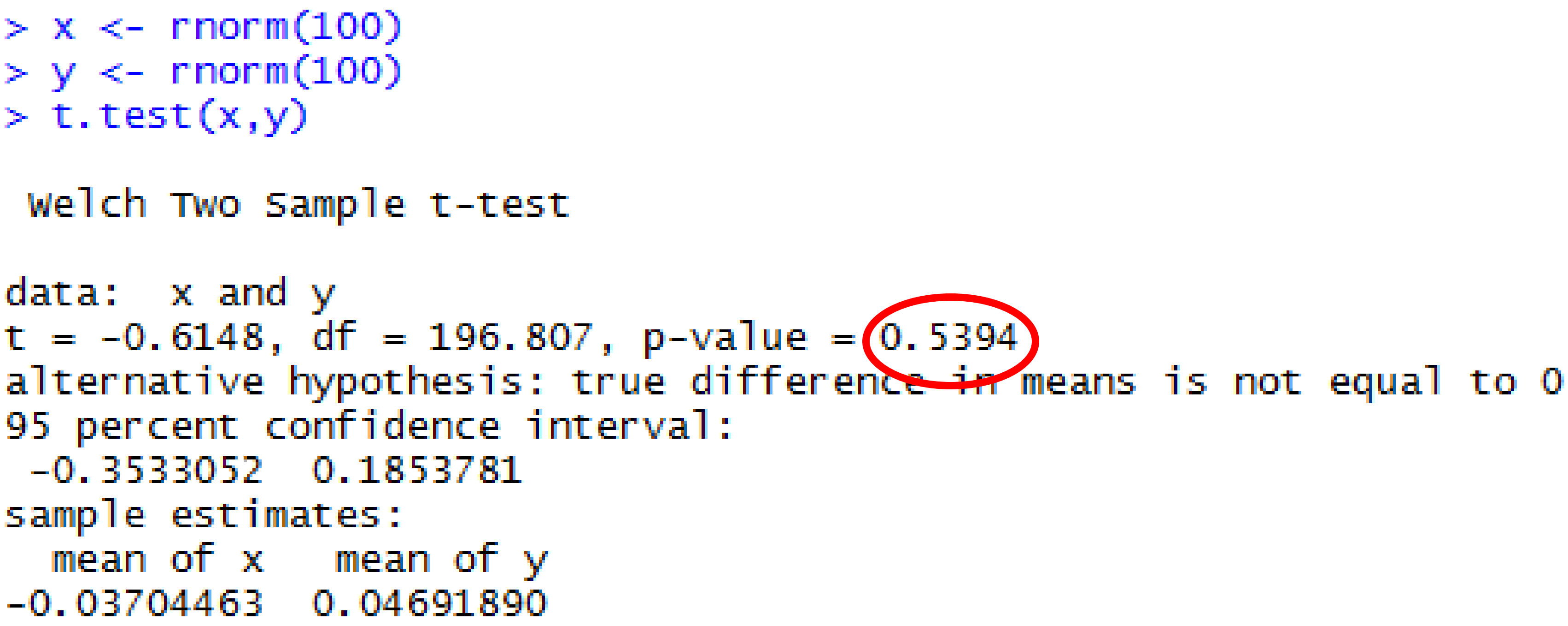
`paired`

a logical indicating whether you want a paired t-test.

`var.equal`

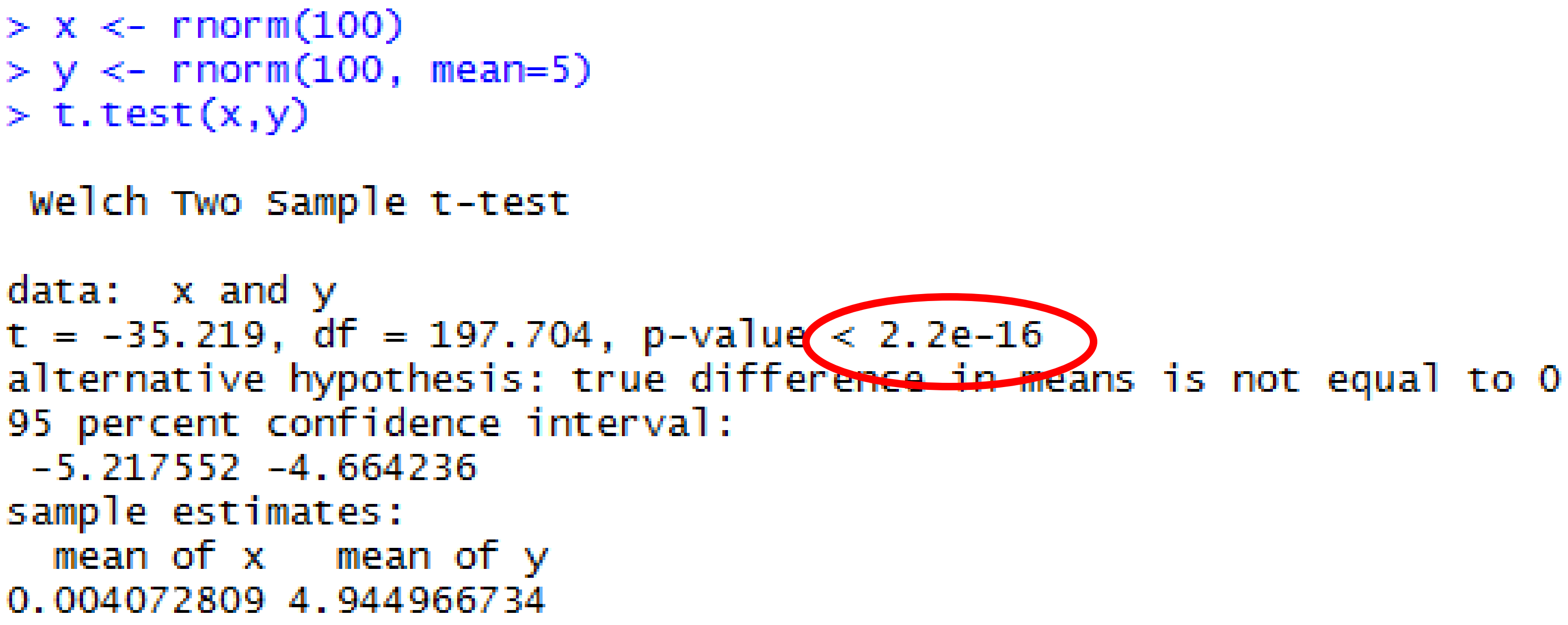
a logical variable indicating whether to treat the two variances as being equal. If `TRUE` then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

t-Test: example



```
> x <- rnorm(100)
> y <- rnorm(100)
> t.test(x,y)

 Welch Two Sample t-test

data:  x and y
t = -0.6148, df = 196.807, p-value = 0.5394
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3533052  0.1853781
sample estimates:
  mean of x   mean of y
-0.03704463  0.04691890
```

t-Test: example 2



```
> x <- rnorm(100)
> y <- rnorm(100, mean=5)
> t.test(x,y)

 Welch Two Sample t-test

data:  x and y
t = -35.219, df = 197.704, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5.217552 -4.664236
sample estimates:
  mean of x   mean of y
0.004072809 4.944966734
```

Mann-Whitney test

Non-parametric

Goal: compare independent samples

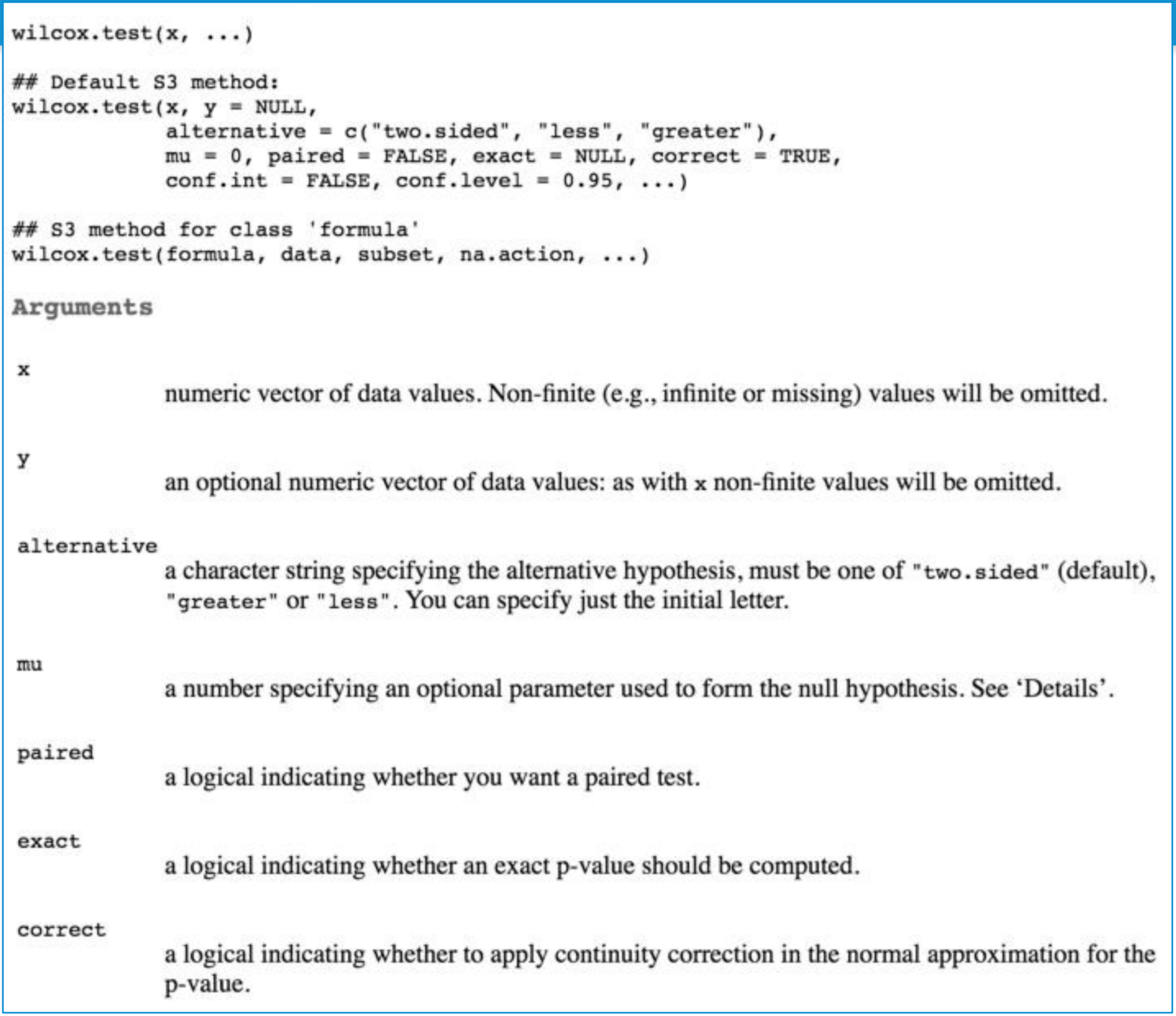
- It can be used instead of the t-test when data is not normal

Hypotheses:

- Two-tailed
  - H_0: $\mu_2 = \mu_1$ H_a: $\mu_2 \neq \mu_1$
- One-tailed (alternative: greater)
  - H_0: $\mu_2 = \mu_1$ H_a: $\mu_2 > \mu_1$
- One-tailed (alternative: less)
  - H_0: $\mu_2 = \mu_1$ H_a: $\mu_2 < \mu_1$

Same hypotheses as the t-test

Mann-Whitney test in R



```
wilcox.test(x, ...)

## Default S3 method:
wilcox.test(x, y = NULL,
            alternative = c("two.sided", "less", "greater"),
            mu = 0, paired = FALSE, exact = NULL, correct = TRUE,
            conf.int = FALSE, conf.level = 0.95, ...)

## S3 method for class 'formula'
wilcox.test(formula, data, subset, na.action, ...)
```

Arguments

`x`
: numeric vector of data values. Non-finite (e.g., infinite or missing) values will be omitted.

`y`
: an optional numeric vector of data values: as with `x` non-finite values will be omitted.

`alternative`
: a character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`. You can specify just the initial letter.

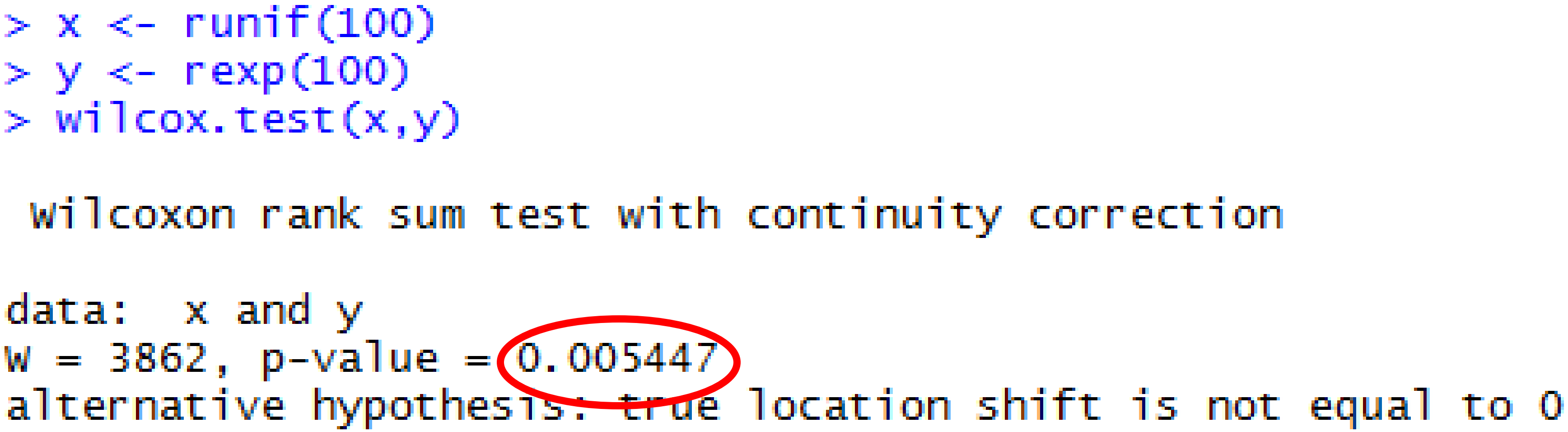
`mu`
: a number specifying an optional parameter used to form the null hypothesis. See ‘Details’.

`paired`
: a logical indicating whether you want a paired test.

`exact`
: a logical indicating whether an exact p-value should be computed.

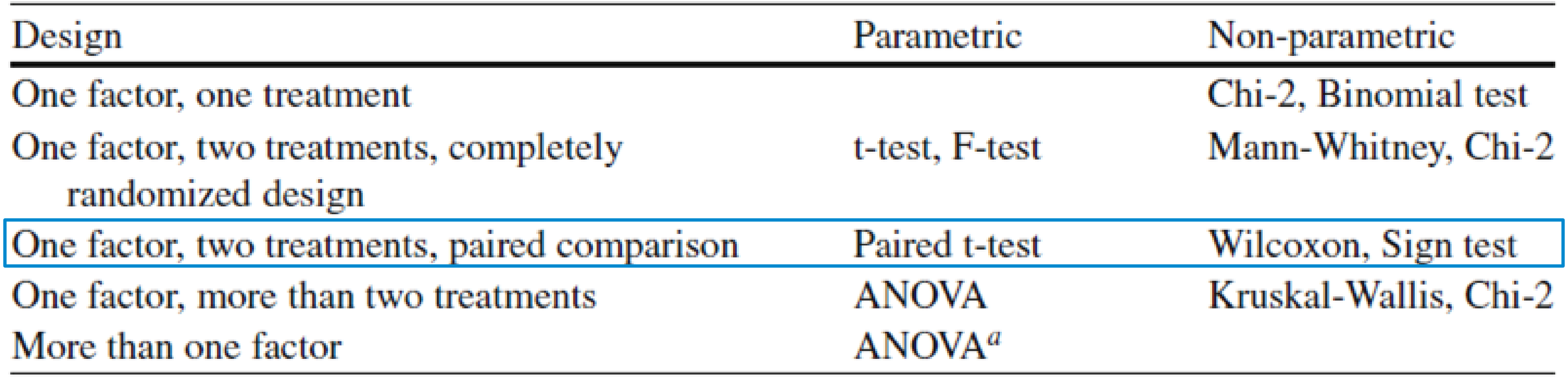
`correct`
: a logical indicating whether to apply continuity correction in the normal approximation for the p-value.

Mann-Whitney test: example



```
> x <- runif(100)
> y <- rexp(100)
> wilcox.test(x,y)

 Wilcoxon rank sum test with continuity correction

data:  x and y
W = 3862, p-value = 0.005447
alternative hypothesis: true location shift is not equal to 0
```

One factor - 2 treatments - paired design



Design	Parametric	Non-parametric
One factor, one treatment		Chi-2, Binomial test
One factor, two treatments, completely randomized design	t-test, F-test	Mann-Whitney, Chi-2
One factor, two treatments, paired comparison	Paired t-test	Wilcoxon, Sign test
One factor, more than two treatments	ANOVA	Kruskal-Wallis, Chi-2
More than one factor	ANOVA[a]	

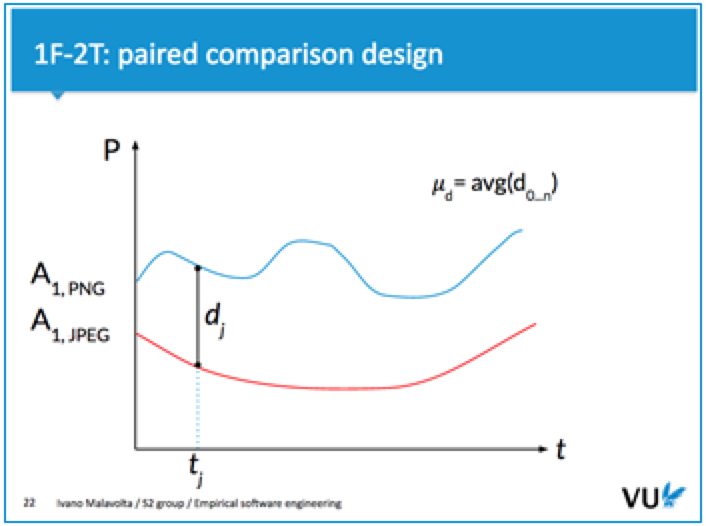
Paired t-Test

Goal: compare independent samples from repeated measures

- Each subject receives different treatments
- We focus on the differences exhibited by each subject with different treatments
- Samples must be equal in size

Hypotheses:

- Two-tailed
  - $H_0: \mu_d = 0$ $H_a: \mu_d \neq 0$
- One-tailed (alternative: greater)
  - $H_0: \mu_d = 0$ $H_a: \mu_d > 0$
- One-tailed (alternative: less)
  - $H_0: \mu_d = 0$ $H_a: \mu_d < 0$

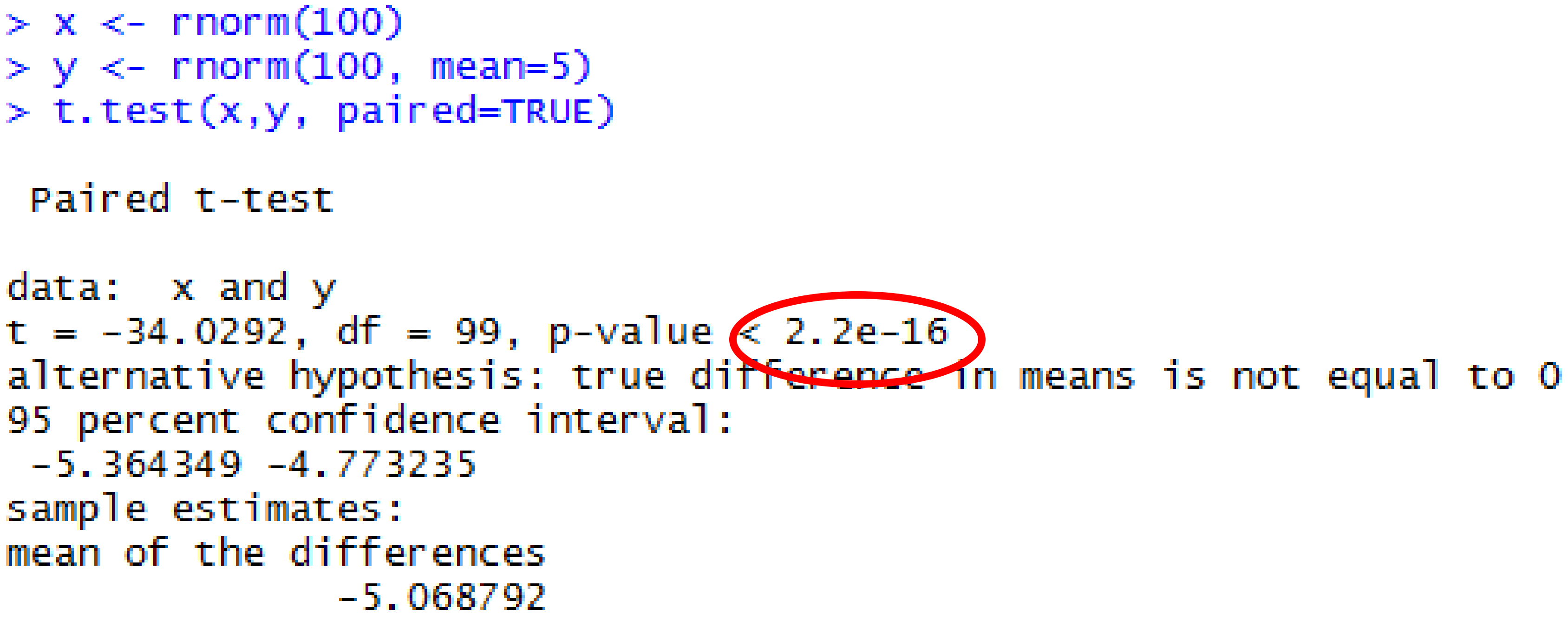


1F-2T: paired comparison design

22 Ivano Malavolta / S2 group / Empirical software engineering

Parametric

Paired t-Test: example



```
> x <- rnorm(100)
> y <- rnorm(100, mean=5)
> t.test(x,y, paired=TRUE)

 Paired t-test

data:  x and y
t = -34.0292, df = 99, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5.364349 -4.773235
sample estimates:
mean of the differences
              -5.068792
```

Wilcoxon signed-rank test

Goal: compare independent samples from repeated measures

- It can be used instead of the paired t-test in case of not normal data

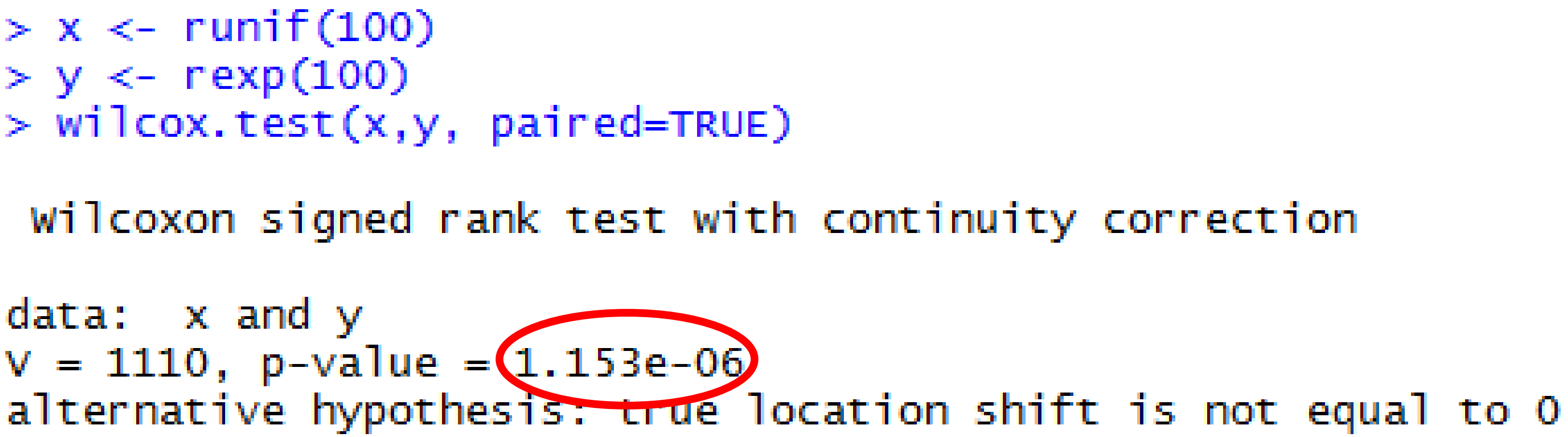
Hypotheses:

- Two-tailed
  - H_0: $\mu_d = 0$ H_a: $\mu_d \neq 0$
- One-tailed (alternative: greater)
  - H_0: $\mu_d = 0$ H_a: $\mu_d > 0$
- One-tailed (alternative: less)
  - H_0: $\mu_d = 0$ H_a: $\mu_d < 0$

Same hypotheses as the paired t-test

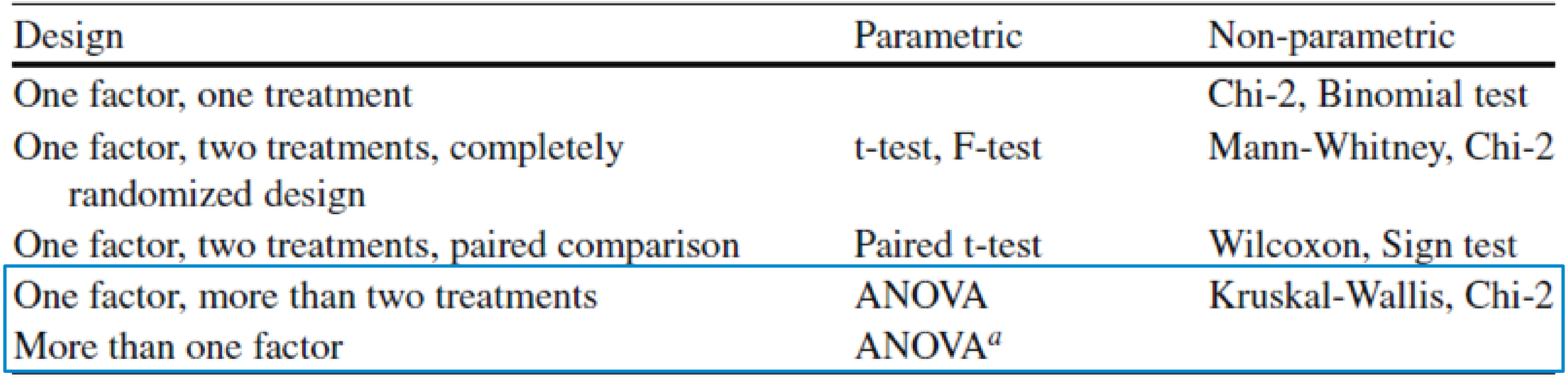
Non-parametric

Wilcoxon signed-rank test: example



```
> x <- runif(100)
> y <- rexp(100)
> wilcox.test(x,y, paired=TRUE)

 Wilcoxon signed rank test with continuity correction

data:  x and y
V = 1110, p-value = 1.153e-06
alternative hypothesis: true location shift is not equal to 0
```

>=1 factors - >2 treatments



Design	Parametric	Non-parametric
One factor, one treatment		Chi-2, Binomial test
One factor, two treatments, completely randomized design	t-test, F-test	Mann-Whitney, Chi-2
One factor, two treatments, paired comparison	Paired t-test	Wilcoxon, Sign test
One factor, more than two treatments	ANOVA	Kruskal-Wallis, Chi-2
More than one factor	ANOVA[a]	

ANOVA (ANalysis Of VAriance)

Goal: understand how much of the total variance is due to differences within factors, and how much is due to differences across factors

- Many types of ANOVA tests
- Works for many experiment designs

Hypotheses:

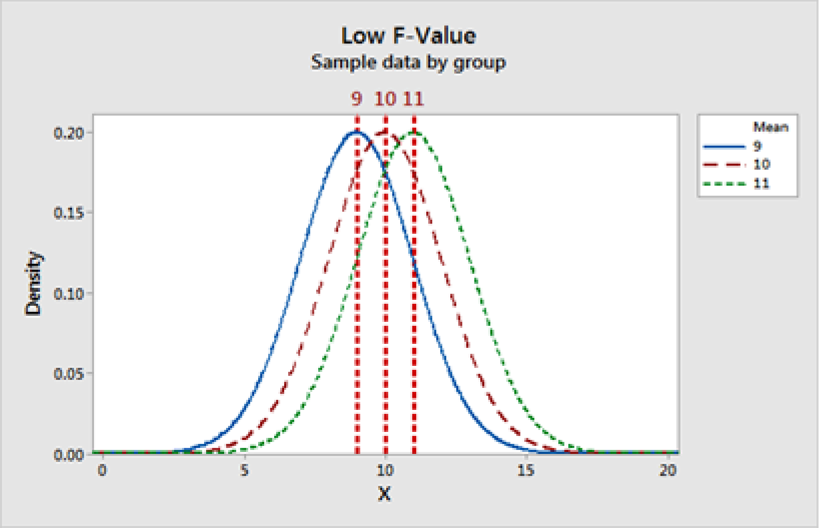
$H_0: \mu_1 = \mu_2 = \mu_3$

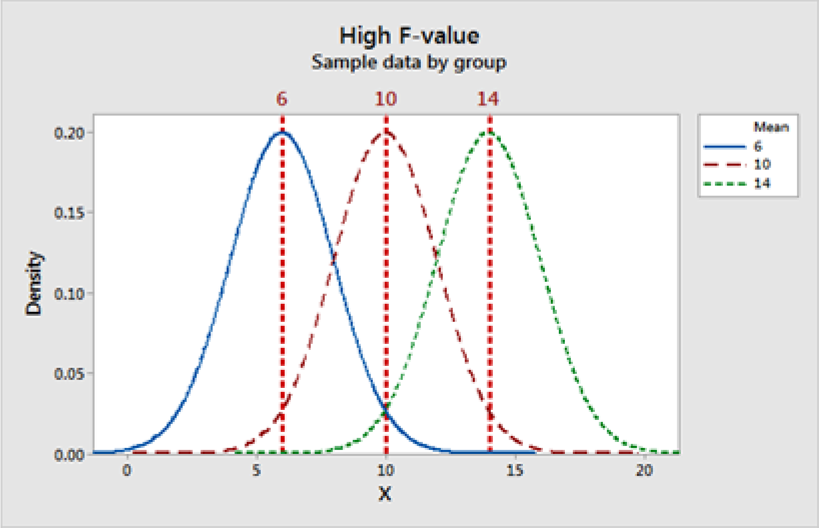
$H_a: \mu_1 \neq \mu_2 \lor \mu_1 \neq \mu_3 \lor \mu_2 \neq \mu_3$

Parametric

F-statistic

F = Variation among sample means / variation within the samples





when H_0 → F follows a known F-distribution

- the mean of the F-distribution tends to be 1

https://goo.gl/jHWo8A

Significance

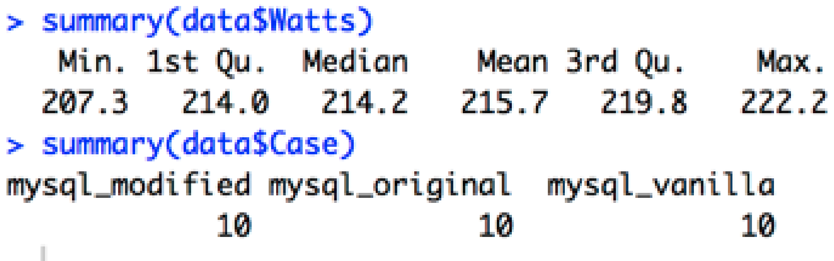
F tends to be larger if H_0 is false

→ the more F deviates from 1, the stronger the evidence for unequal population variances

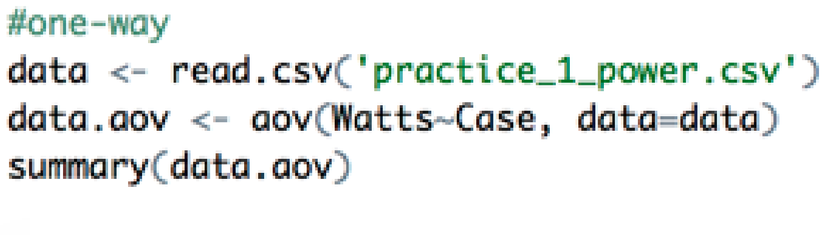
- Methods to determine significance level:
  - *textbook:* compare F against a table of critical values (according to DF and α). If $F > F_{critical}$, reject H_0
  - ***computer-based:*** compute the p-value of finding F greater than the observed value. If $p < \alpha$, reject H_0

Types of ANOVA

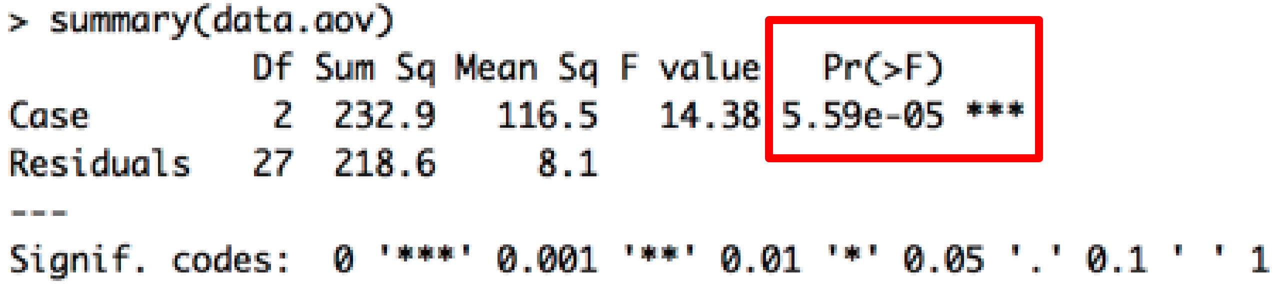
- *One-way* ANOVA
  - one factor, >2 treatments
  - if 2 treatments: equivalent to *t-test* (almost never used)



```
> summary(data$Watts)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  207.3   214.0   214.2   215.7   219.8   222.2 
> summary(data$Case)
mysql_modified mysql_original  mysql_vanilla 
            10             10             10 
```



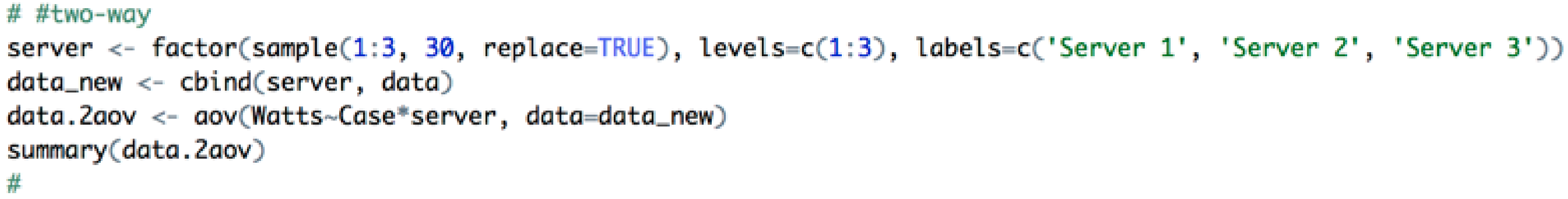
```
#one-way
data <- read.csv('practice_1_power.csv')
data.aov <- aov(Watts~Case, data=data)
summary(data.aov)
```



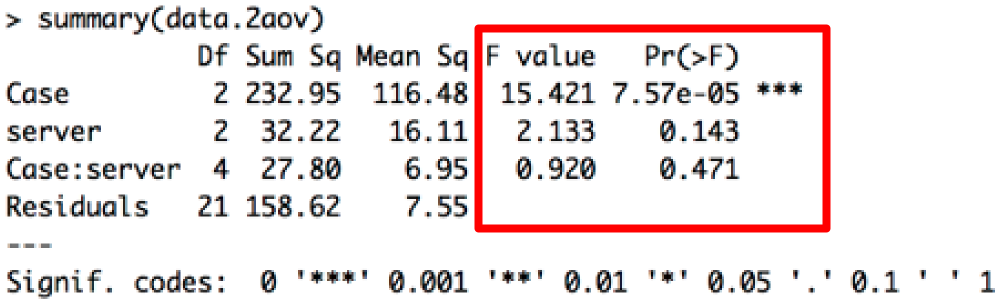
```
> summary(data.aov)
            Df Sum Sq Mean Sq F value   Pr(>F)    
Case         2  232.9   116.5   14.38 5.59e-05 ***
Residuals   27  218.6     8.1                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Types of ANOVA

- *Factorial* ANOVA
  - 2 (two-way) or more factors
  - any number of treatments
  - also computes interactions



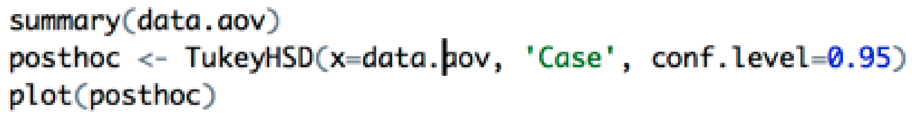
```
# #two-way
server <- factor(sample(1:3, 30, replace=TRUE), levels=c(1:3), labels=c('Server 1', 'Server 2', 'Server 3'))
data_new <- cbind(server, data)
data.2aov <- aov(Watts~Case*server, data=data_new)
summary(data.2aov)
#
```



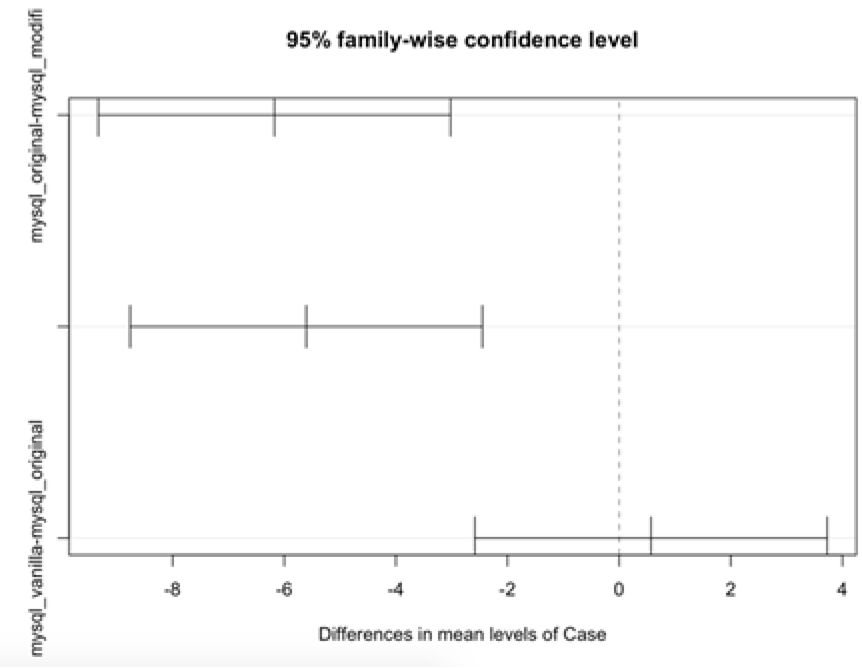
```
> summary(data.2aov)
            Df Sum Sq Mean Sq F value   Pr(>F)    
Case         2 232.95  116.48  15.421 7.57e-05 ***
server       2  32.22   16.11   2.133    0.143    
Case:server  4  27.80    6.95   0.920    0.471    
Residuals   21 158.62    7.55                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

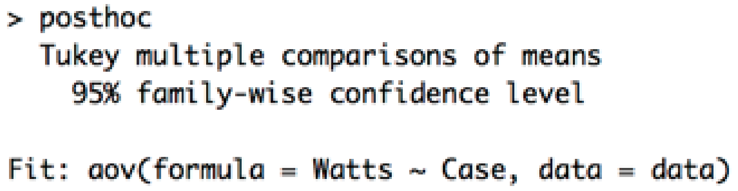
How to know which treatments really differ?

Tukey's test

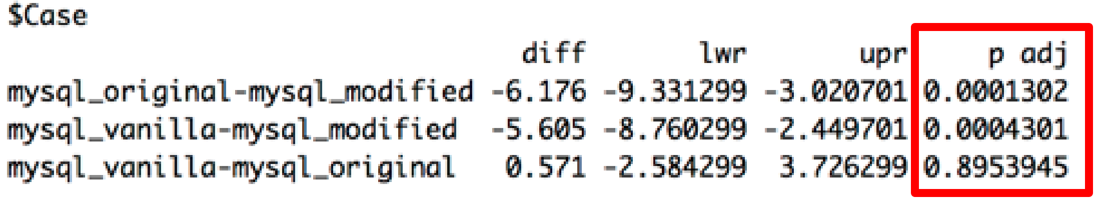


```
summary(data.aov)
posthoc <- TukeyHSD(x=data.aov, 'Case', conf.level=0.95)
plot(posthoc)
```





```
> posthoc
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Watts ~ Case, data = data)
```



```
$Case
                               diff       lwr       upr     p adj
mysql_original-mysql_modified -6.176 -9.331299 -3.020701 0.0001302
mysql_vanilla-mysql_modified  -5.605 -8.760299 -2.449701 0.0004301
mysql_vanilla-mysql_original   0.571 -2.584299  3.726299 0.8953945
```

ANOVA assumptions

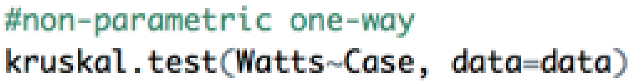
- The dependent variable should be **continuous**
- Samples must be **independent**
- **Normal distribution** of the dependent variable between the groups (approximately)
- Residuals (aka errors in the sample) should be normally distributed
  - qqPlot(residuals(myData.aov))
- **Homoscedasticity**
  - variance between groups should be the same
    - leveneTest(x ~ y, data=myData)

Assumptions violated

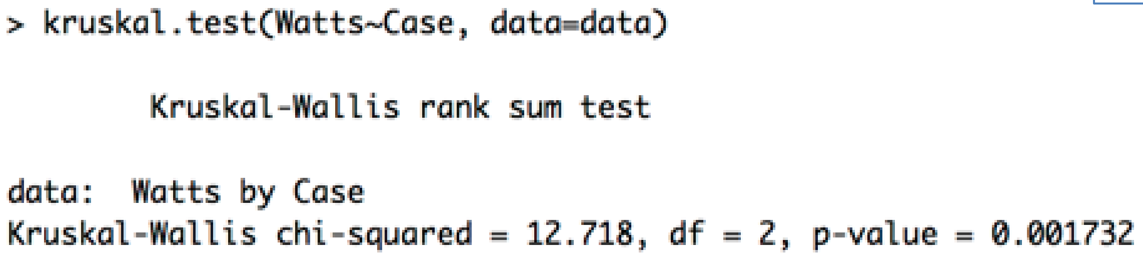
→ non-parametric alternative

ANOVA: non-parametric alternative

- Kruskal-Wallis: one-way non-parametric ANOVA
  - one factor, multiple treatments
  - no estimate of the treatment effect (due to ranking)



```
#non-parametric one-way
kruskal.test(Watts~Case, data=data)
```



```
> kruskal.test(Watts~Case, data=data)

        Kruskal-Wallis rank sum test

data:  Watts by Case
Kruskal-Wallis chi-squared = 12.718, df = 2, p-value = 0.001732
```

Use ARTool when you have >2 factors

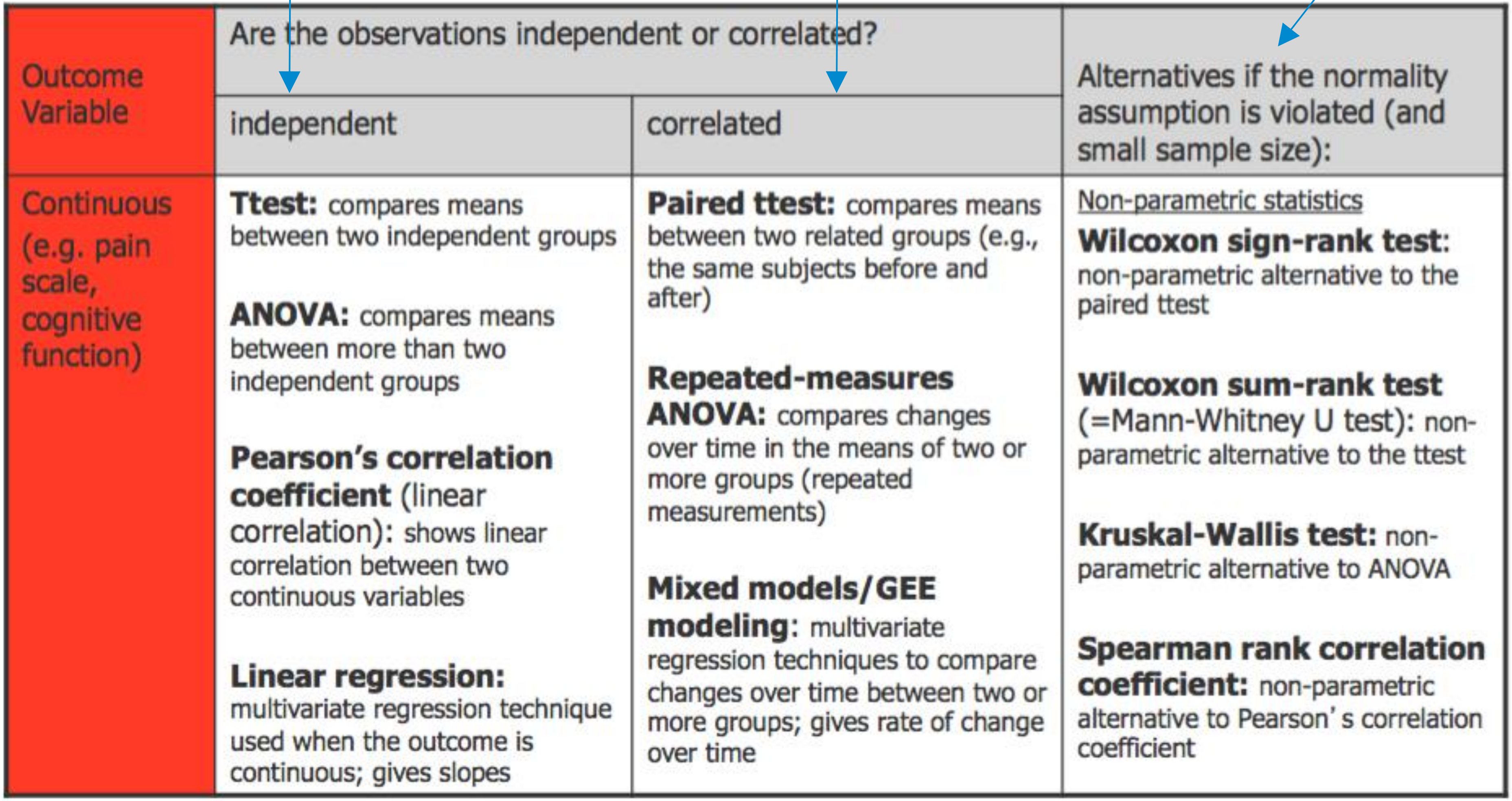
Non-parametric

Main statistical tests

You are measuring different subjects

You are measuring the same subject against different treatments

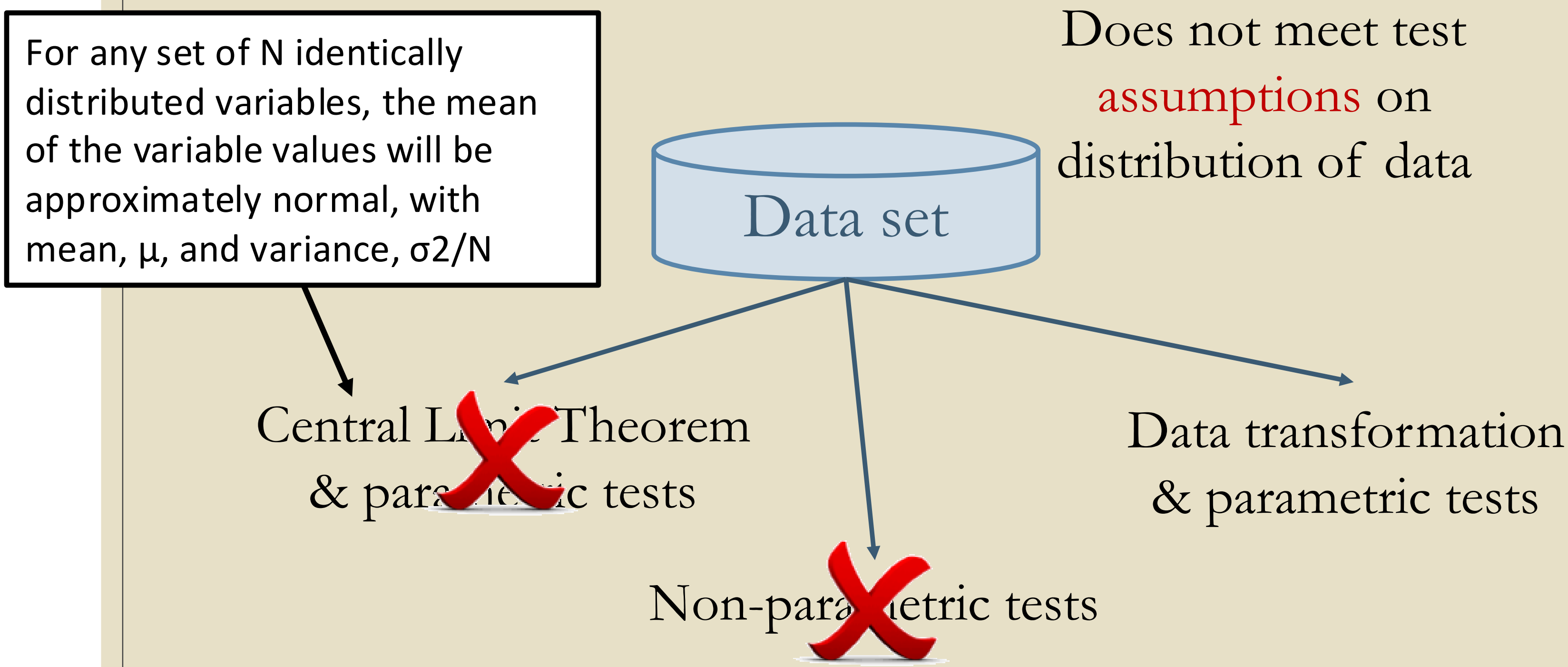
Use this in case the values of your dep. var are not normally distributed



Outcome Variable	Are the observations independent or correlated?		Alternatives if the normality assumption is violated (and small sample size):
	independent	correlated	
Continuous (e.g. pain scale, cognitive function)	**Ttest:** compares means between two independent groups **ANOVA:** compares means between more than two independent groups **Pearson's correlation coefficient** (linear correlation): shows linear correlation between two continuous variables **Linear regression:** multivariate regression technique used when the outcome is continuous; gives slopes	**Paired ttest:** compares means between two related groups (e.g., the same subjects before and after) **Repeated-measures ANOVA:** compares changes over time in the means of two or more groups (repeated measurements) **Mixed models/GEE modeling**: multivariate regression techniques to compare changes over time between two or more groups; gives rate of change over time	Non-parametric statistics **Wilcoxon sign-rank test**: non-parametric alternative to the paired ttest **Wilcoxon sum-rank test** (=Mann-Whitney U test): non-parametric alternative to the ttest **Kruskal-Wallis test:** non-parametric alternative to ANOVA **Spearman rank correlation coefficient:** non-parametric alternative to Pearson' s correlation coefficient

Data transformation

PARAMETRIC VS. NON-PARAMETRIC



For any set of N identically distributed variables, the mean of the variable values will be approximately normal, with mean, µ, and variance, σ2/N

Does not meet test assumptions on distribution of data

Data set

Central Limit Theorem & parametric tests

Non-parametric tests

Data transformation & parametric tests

These slides are available on Canvas!
File: Vegas TB ICSE17.pdf

https://dl.acm.org/doi/10.1145/2889160.2891054

DATA TRANSFORMATION

- Corrects several problems in data:
  - Non-normality
  - Unequal variances

- Will not change the relationships between variables
  - The relative differences between scores for a given variable stay the same

- Does change the differences between different variables
  - Changes units of measurement

(SOME) POSSIBLE TRANSFORMATIONS

Transformation	Calculation	Can correct for
Log	Log (X_i)	Positive skew Unequal variances
Square root	▯ X_i	Positive skew Unequal variances
Reciprocal	$1/X_i$	Positive skew Unequal variances
Reverse score	Substract X_i from highest score	Negative skew

https://dl.acm.org/doi/10.1145/2889160.2891054

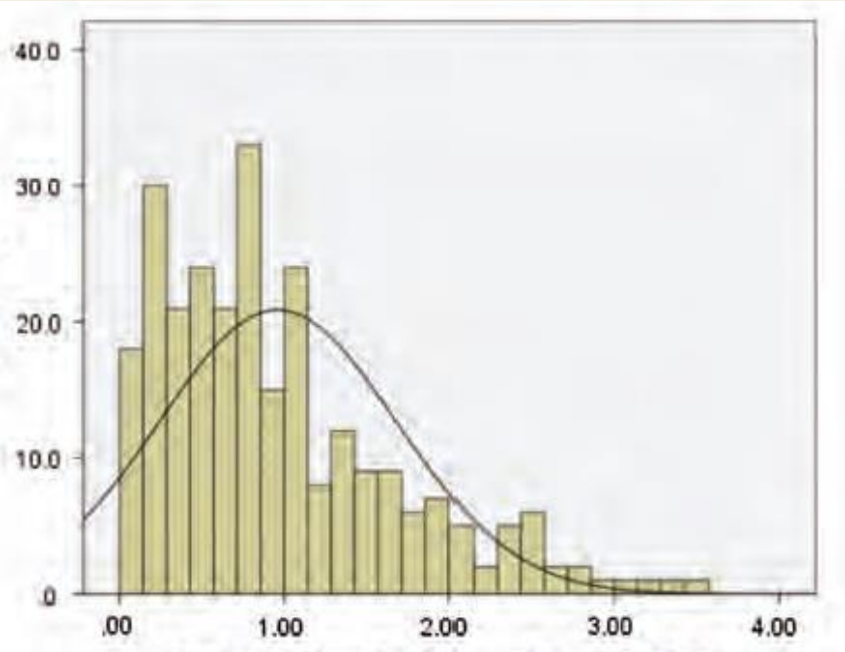
(SOME) POSSIBLE TRANSFORMATIONS

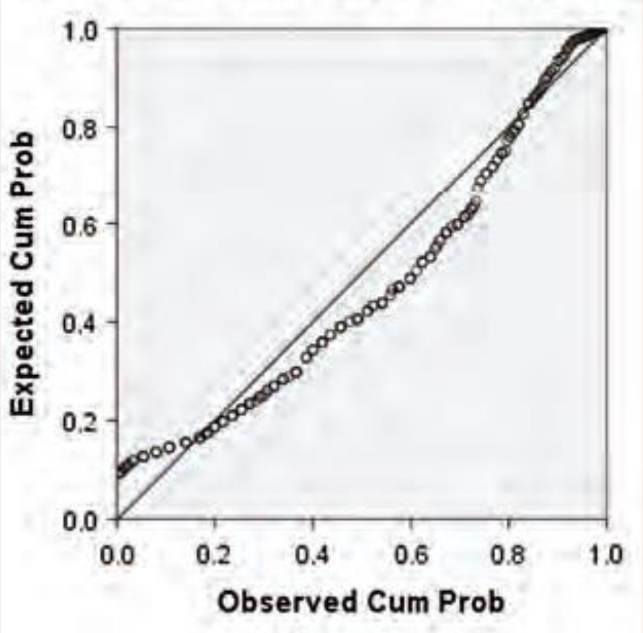
Transformation	Calculation	Can correct for
Log	Log (X_i)	Positive skew riances
Sq		
Recip		ew Unequal variances
Reverse score	Substract X_i from highest score	Negative skew

Depending on the problem of the data we should try a different one

https://dl.acm.org/doi/10.1145/2889160.2891054

AN EXAMPLE: ORIGINAL DATA

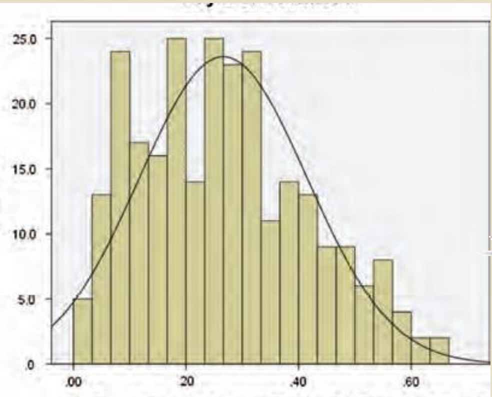




https://dl.acm.org/doi/10.1145/2889160.2891054

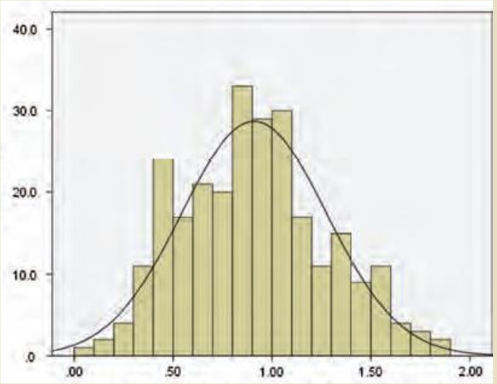
AN EXAMPLE: TRANSFORMED DATA

Log transformation



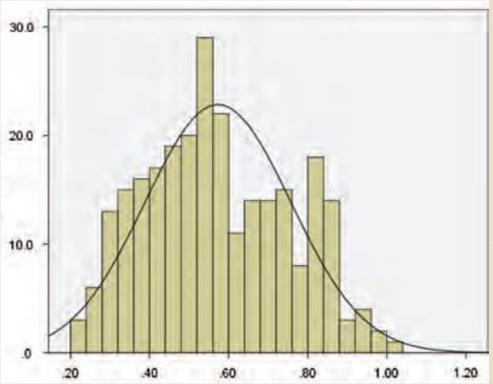
✗

Square root transformation



✓

Reciprocal transformation



✗

https://dl.acm.org/doi/10.1145/2889160.2891054

DISCUSSION

- Data transformation looks like a better option
- Do not forget to “un-transform” for data interpretation
- Un-transformation is necessary for:
  - Mean
  - Confidence interval for mean
- Un-transformation is not necessary for:
  - Significance
  - Power

https://dl.acm.org/doi/10.1145/2889160.2891054

TIPS

- When selecting statistical test check test assumptions on distribution of data

- If distribution assumptions are not met:
  - Do not use limit central theorem
  - Try data transformation first
  - If it does not work, use non-parametric tests

- If you transform, do not forget un-transform

https://dl.acm.org/doi/10.1145/2889160.2891054

Other tips

- You can use the bestNormalize package to discover the best transformation to apply
- Remember to apply the same transformation **to ALL the measures of a dependent variable**
  - multiple variables, in case you analyze interactions
- When you will visualize tables and plots you will need to show the non-transformed data
- If you do not manage to satisfy the assumptions of your statistical test (after transforming), then indeed you can go with a non-parametric one (this is always the safest way, even though it will negatively impact the power of your tests)

Correction of p-values

Example

Dependent variable = energy consumption of the app

Independent variables =

- A: Image encoding algorithm: {png, jpeg}
- B: Mobile device type: {high-end, low-end}
- C: Network conditions: {wifi, 3G}

You perform 3 tests:

- t.test(A, B)
- t.test(A, C)
- t.test(B, C)

P(at least one significant result) = 1 – P(no significant results)

= 1 – $(1 - 0.05)^3$

≈ 0.15

→ 15% chance of seeing relevant results, when there may be none

The problem

Multiple tests → higher probability of getting (statistically significant) results

→ you have to adjust your α (it was 0.05)

Three main correction techniques:

- Bonferroni
- Holm
- Benjamini- Hochberg

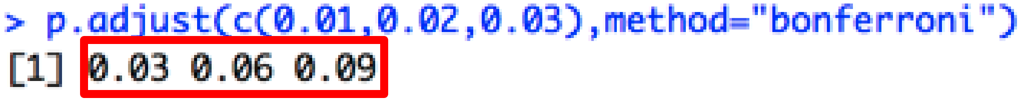
Bonferroni correction

Supposing we are doing N tests,

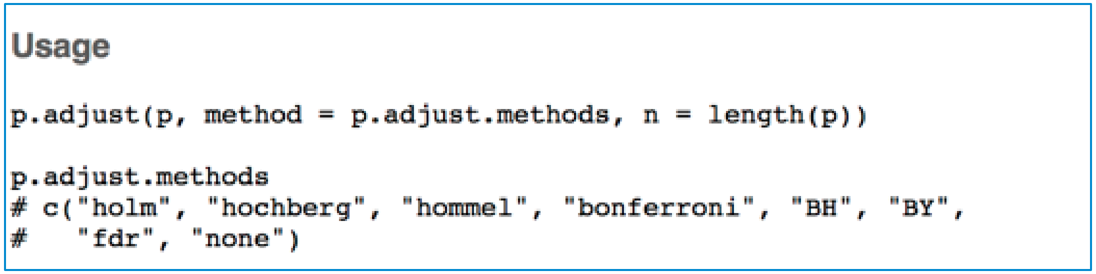
we can reject H_0 if the p-values of those tests are below α/N

We can reject the H_0 if a test provides a p-value < 0.05/3=0.016

→ 0.016 is our new significance threshold!



```
> p.adjust(c(0.01,0.02,0.03),method="bonferroni")
[1] 0.03 0.06 0.09
```



Usage

```
p.adjust(p, method = p.adjust.methods, n = length(p))

p.adjust.methods
# c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY",
#   "fdr", "none")
```

Holm's correction

Less stringent than Bonferroni's

Procedure:

- rank your p-values from the smallest to the largest
- multiply the first by N, the second by N-1, etc.
- a p-value is significant if, after multiplied, it is <0.05

P-values of the tests: {0.01, 0.02, 0.03}

Bonferroni:

- 0.01 * 3 = 0.03
- 0.02 * 3 = 0.06
- 0.03 * 3 = 0.09

Holm:

- 0.01 * 3 = 0.03
- 0.02 * 2 = 0.04
- 0.03 * 1 = 0.03

Benjamini- Hochberg 's correction

The least stringent correction (**highly suggested**)

Procedure:

- rank your p-values from the smallest to the largest
- assign ranks to each p-value according to its position

  - first=1, second=2, third=3, …

- compute the BH critical value for each p-value as (i/N)Q

  i = the i^{th} p-value

  N = the total number of p-values

  Q = the acceptable false discovery rate as percentage (e.g., 50%)

- identify **P** as the highest p-value that is smaller than the BH critical value
- [illegible] considered as significant

Benjamini- Hochberg ’s correction

P-values of the tests: {0.01, 0.02, 0.03, 0.04, 0.2, 0.4, 0.8, 0.9}

Original p-value	Rank	BH
0.01	1	(1/8)*0.5= **0.0625**
0.02	2	(2/8)*0.5= **0.125**
0.03	3	(3/8)*0.5= **0.1875**
0.04	4	(4/8)*0.5= **0.25**
0.2	5	(5/8)*0.5= **0.3125**
0.4	6	(6/8)*0.5= **0.375**
0.8	7	(7/8)*0.5= **0.4375**
0.9	8	(8/8)*0.5= **0.5**

Effect size

Effect Size

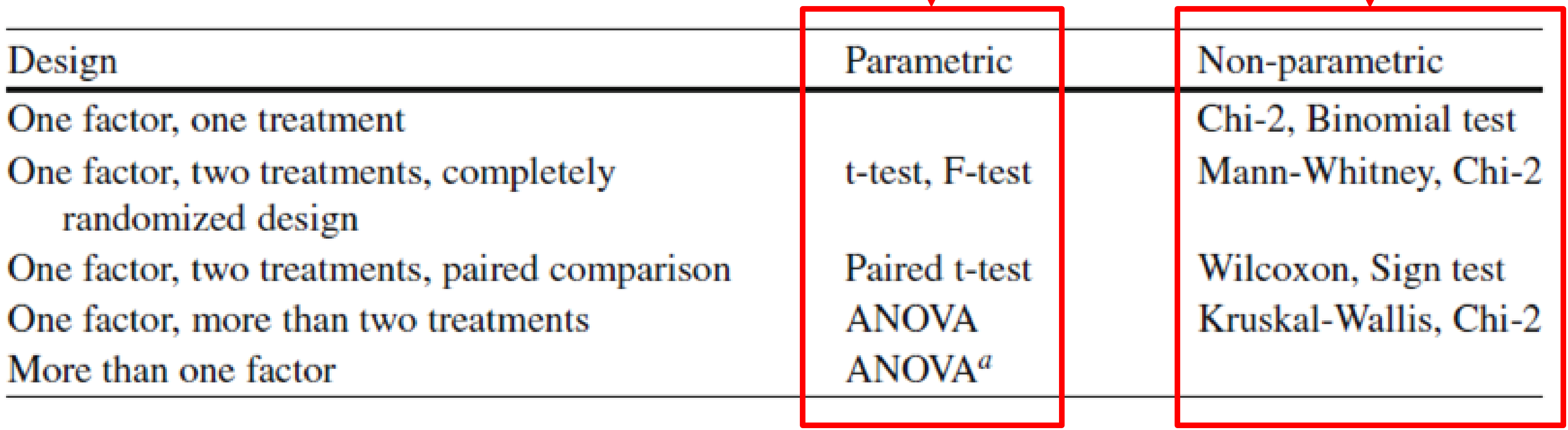
- $p < 0.05$

Effect Size: quantitative measure of the **strength** of a phenomenon

- Actual difference: 0.0001%

Effect size measures

- **Cohen's d**
  - parametric statistics
- **Cliff's delta**
  - non-parametric statistics



Design	Parametric	Non-parametric
One factor, one treatment		Chi-2, Binomial test
One factor, two treatments, completely randomized design	t-test, F-test	Mann-Whitney, Chi-2
One factor, two treatments, paired comparison	Paired t-test	Wilcoxon, Sign test
One factor, more than two treatments	ANOVA	Kruskal-Wallis, Chi-2
More than one factor	ANOVA[a]	

Cohen’s d

The magnitude of a main factor treatment effect on the dependent variable

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s}$$

Values:

0 = full overlap

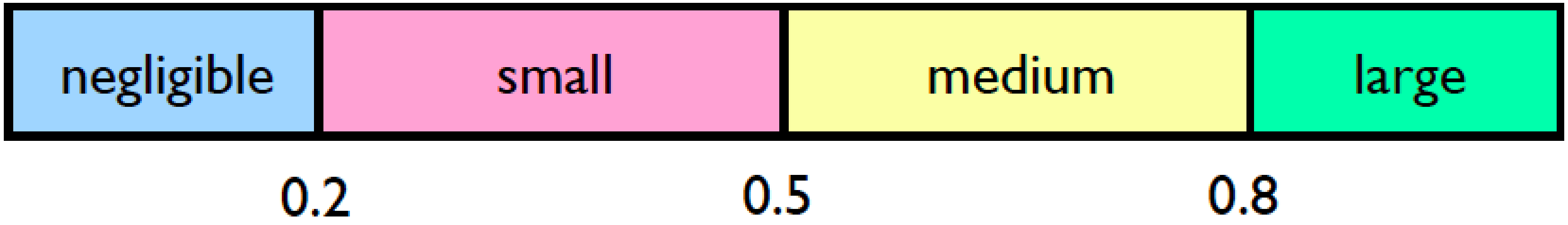
1 = 1-sigma distance between the means

…

3 = 3-sigma distance → ~no overlap

Where:

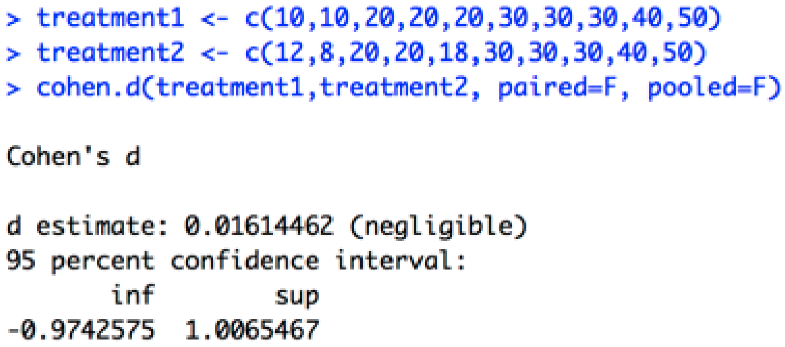
- x_1 , x_2 = the means of the two groups
- s = standard deviation



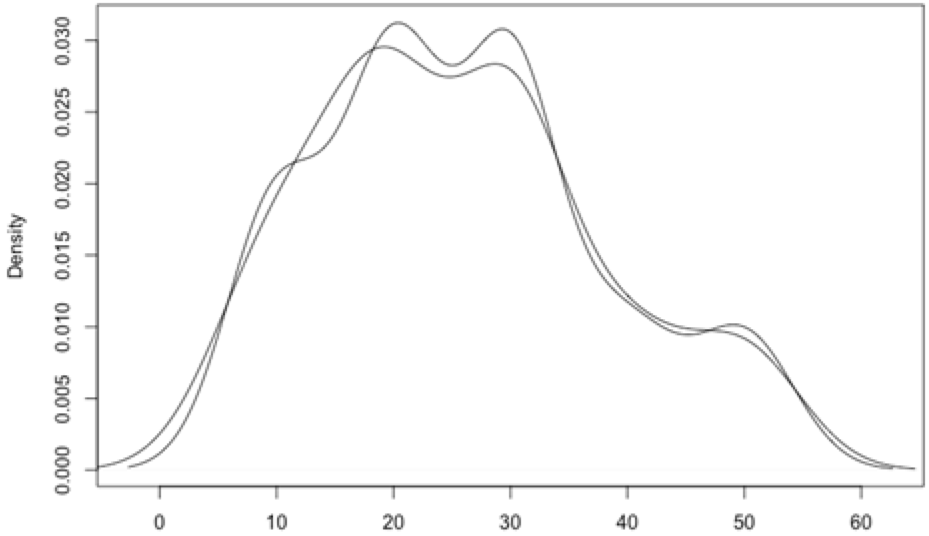
negligible | small | medium | large

0.2 | 0.5 | 0.8

Cohen’s d in R



```
> treatment1 <- c(10,10,20,20,20,30,30,30,40,50)
> treatment2 <- c(12,8,20,20,18,30,30,30,40,50)
> cohen.d(treatment1,treatment2, paired=F, pooled=F)

Cohen's d

d estimate: 0.01614462 (negligible)
95 percent confidence interval:
       inf        sup
-0.9742575  1.0065467
```



Cliff’s delta

Represents the degree of overlap between the two distributions of scores

$$d = \frac{\#(x_i > x_j) - \#(x_i < x_j)}{mn}$$

Values:

0 = full overlap

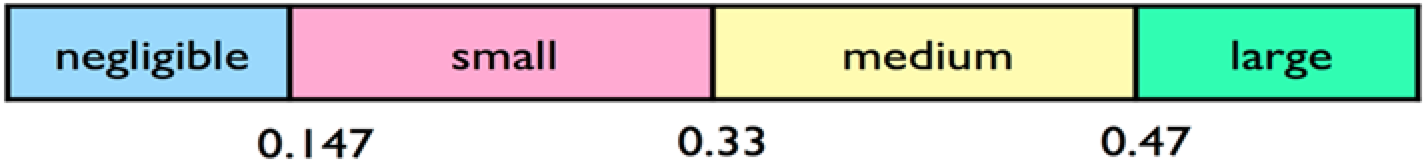
+1 = all the values of one group > all the values of the other one

…

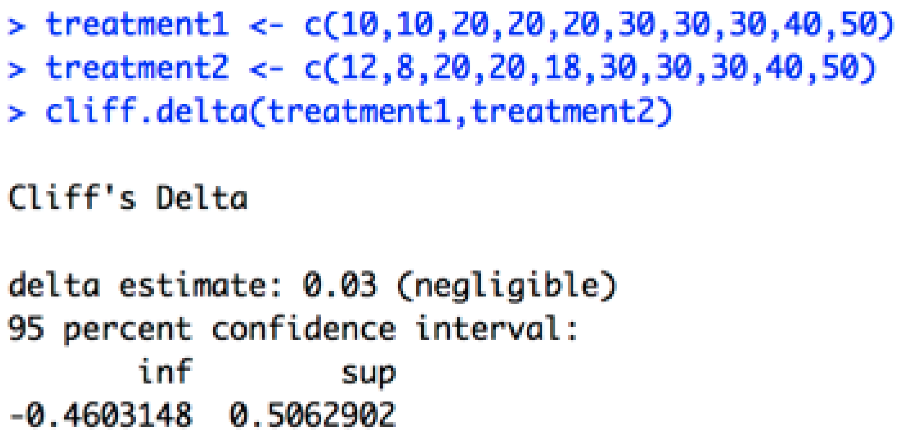
-1 = the inverse

Where:

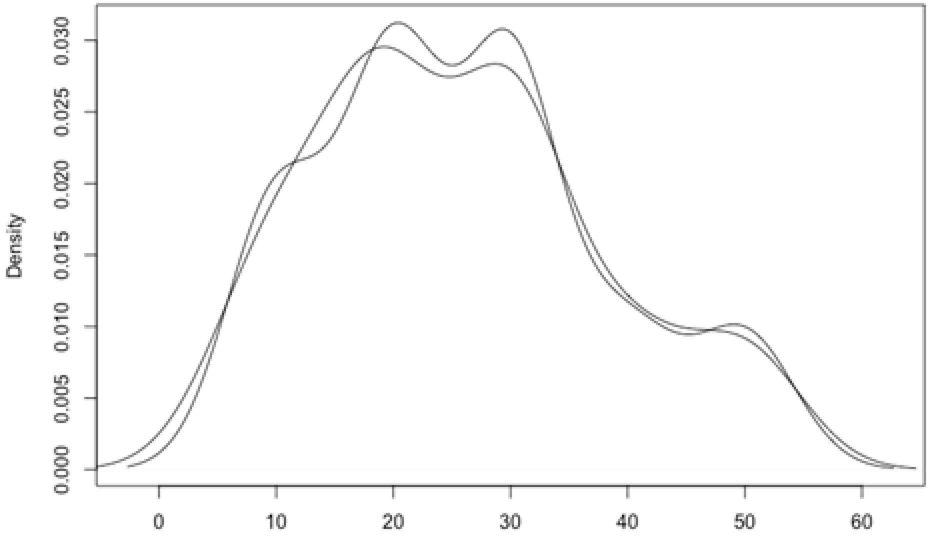
- x_i = the values of the first group
- x_j = the values of the second group
- m, n = the cardinalities of the two groups



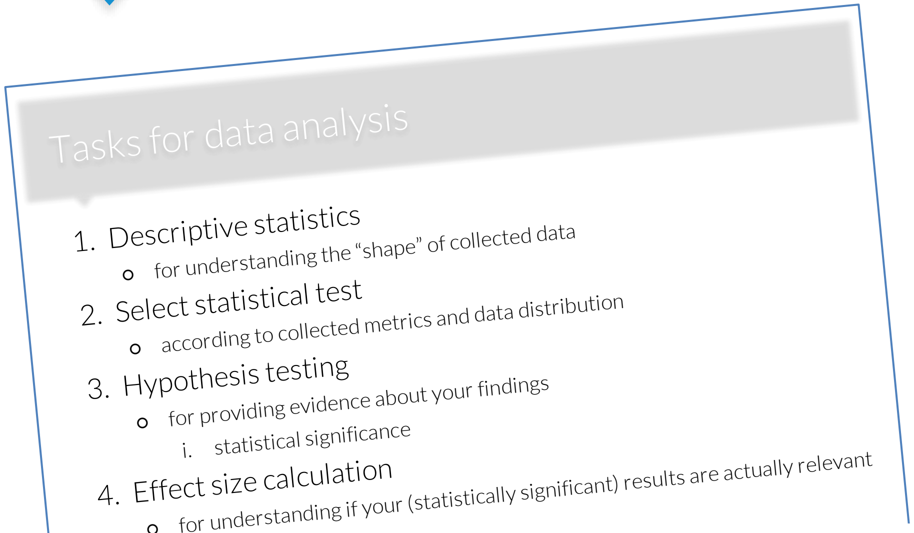
Cliff’s delta in R



```
> treatment1 <- c(10,10,20,20,20,30,30,30,40,50)
> treatment2 <- c(12,8,20,20,18,30,30,30,40,50)
> cliff.delta(treatment1,treatment2)

Cliff's Delta

delta estimate: 0.03 (negligible)
95 percent confidence interval:
       inf        sup 
-0.4603148  0.5062902 
```



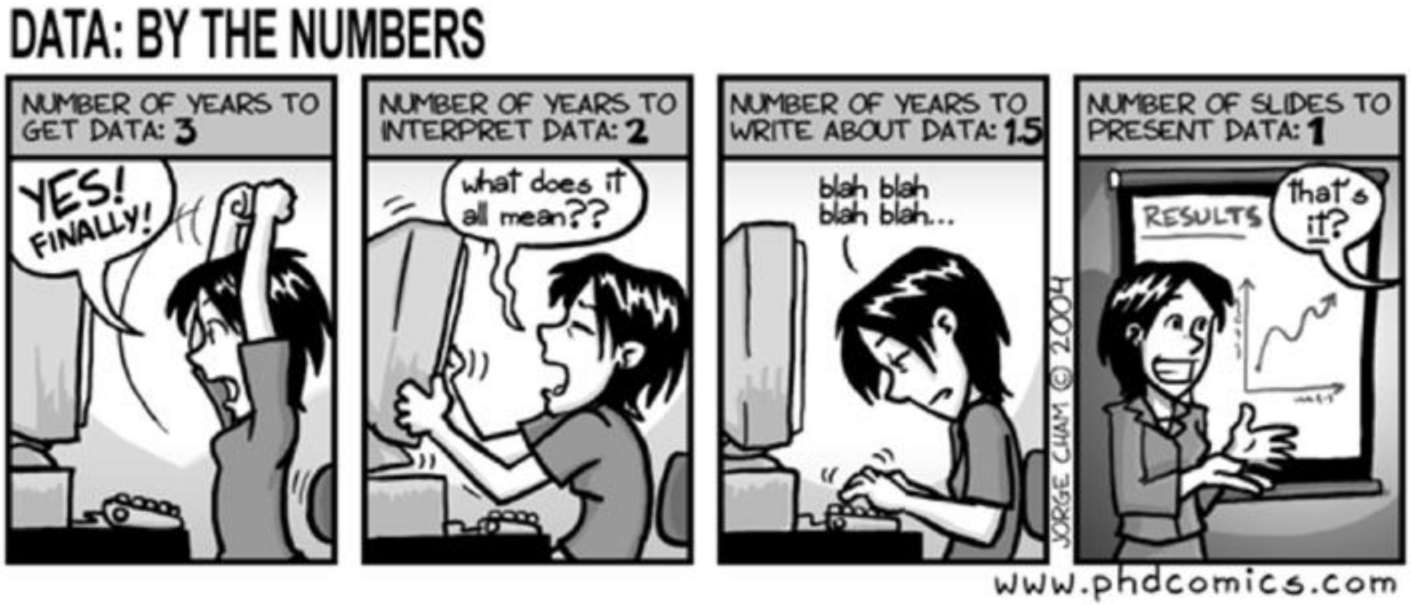
What this module means to you?



Tasks for data analysis

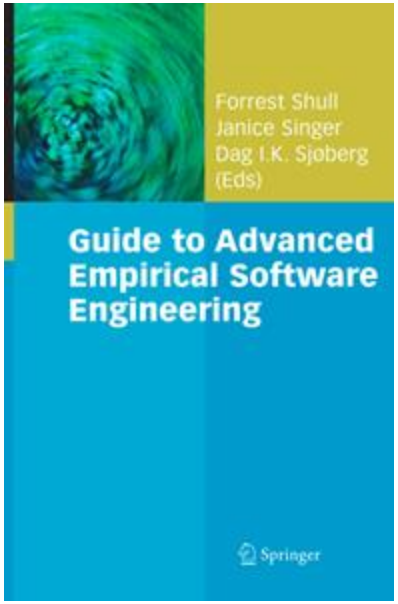
1. Descriptive statistics
   - for understanding the “shape” of collected data
2. Select statistical test
   - according to collected metrics and data distribution
3. Hypothesis testing
   - for providing evidence about your findings
     i. statistical significance
4. Effect size calculation
   - for understanding if your (statistically significant) results are actually relevant in practice
5. Power analysis
   - for knowing if your results a

4 Ivano Malavolta / S2 group / Statistical tests and ef



DATA: BY THE NUMBERS
NUMBER OF YEARS TO GET DATA: 3
YES! FINALLY!
NUMBER OF YEARS TO INTERPRET DATA: 2
what does it all mean??
NUMBER OF YEARS TO WRITE ABOUT DATA: 1.5
blah blah blah blah...
NUMBER OF SLIDES TO PRESENT DATA: 1
RESULTS
that's it?
JORGE CHAM © 2004
www.phdcomics.com

Readings



Chapter 6

Basics of Software Engineering Experimentation
Natalia Juristo and Ana M. Moreno
Foreword by Shari Lawrence Pfleeger
Springer Science+Business Media, LLC

Part 3

Slides of Sira Vegas's technical briefings at ICSE 2017 (on Canvas)

[1] Dybå, Tore, Vigdis By Kampenes, and Dag IK Sjøberg. "A systematic review of statistical power in software engineering experiments." Information and Software Technology 48.8 (2006): 745-755.

Acknowledgements

Some contents of lecture extracted from:

- Giuseppe Procaccianti's lectures at VU