

Statistics



Kevin Rue-Albrecht

University of Oxford (for the African Institute of Biomedical Science and Technology)

2022-11-14 (updated: 2022-11-13)

Learning goals

- Identify statistical distributions and tests available in R.
- Learn to use statistical distributions and tests in R.
- Describe how to interpret the result of statistical tests.
- Understand how to visualise data in the context of statistical testing.



Learning objectives

- Query built-in statistical distributions.
- Perform statistical tests.
- Interpret test results.
- Apply multiple testing correction.
- Visualise data and test results.

Prerequisites

- A computer with **Microsoft Windows**.
- A working installation of **R**.
- A working installation of **RStudio Desktop**.

- Launch RStudio Desktop on your Windows computer.
- Make a copy of the template notebook for this lesson in your **git** repository.
- Create a new RStudio project called `week2`.
- Make a symbolic link to your copy of the notebook in the RStudio project for this week.
- Download the input data files from Microsoft Teams.
- Open the notebook and follow along, editing and running the code as needed.

-  includes a number of common statistical distributions:
 - The Normal Distribution
 - The Binomial Distribution
 - The Poisson Distribution
 - ...
-  implements a range of statistical tests:
 - Student's t-Test
 - Pearson's Chi-squared Test for Count Data
 - Wilcoxon Rank Sum and Signed Rank Tests
 - ...

Distribution	Probability	Quantile	Density	Random
Beta	<code>pbeta</code>	<code>qbeta</code>	<code>dbeta</code>	<code>rbeta</code>
Binomial	<code>pbinom</code>	<code>qbinom</code>	<code>dbinom</code>	<code>rbinom</code>
Cauchy	<code>pcauchy</code>	<code>qcauchy</code>	<code>dcauchy</code>	<code>rcauchy</code>
Chi-Square	<code>pchisq</code>	<code>qchisq</code>	<code>dchisq</code>	<code>rchisq</code>
Exponential	<code>pexp</code>	<code>qexp</code>	<code>dexp</code>	<code>rexp</code>
F	<code>pf</code>	<code>qf</code>	<code>df</code>	<code>rf</code>
Gamma	<code>pgamma</code>	<code>qgamma</code>	<code>dgamma</code>	<code>rgamma</code>
Geometric	<code>pgeom</code>	<code>qgeom</code>	<code>dgeom</code>	<code>rgeom</code>
Hypergeometric	<code>phyper</code>	<code>qhyper</code>	<code>dhyper</code>	<code>rhyper</code>
Logistic	<code>plogis</code>	<code>qlogis</code>	<code>dlogis</code>	<code>rlogis</code>
Log Normal	<code>plnorm</code>	<code>qlnorm</code>	<code>dlnorm</code>	<code>rlnorm</code>
Negative Binomial	<code>pnbinom</code>	<code>qnbinom</code>	<code>dnbinom</code>	<code>rnbinom</code>
Normal	<code>pnorm</code>	<code>qnorm</code>	<code>dnorm</code>	<code>rnorm</code>
Poisson	<code>ppois</code>	<code>qpois</code>	<code>dpois</code>	<code>rpois</code>
Student t	<code>pt</code>	<code>qt</code>	<code>dt</code>	<code>rt</code>
Studentized Range	<code>ptukey</code>	<code>qtukey</code>	<code>dtukey</code>	<code>rtukey</code>
Uniform	<code>punif</code>	<code>qunif</code>	<code>dunif</code>	<code>runif</code>
Weibull	<code>pweibull</code>	<code>qweibull</code>	<code>dweibull</code>	<code>rweibull</code>
Wilcoxon Rank Sum Statistic	<code>pwilcox</code>	<code>qwilcox</code>	<code>dwilcox</code>	<code>rwilcox</code>

- Each distribution has a root name, e.g. `norm`
- Every distribution has four functions.
- The root name is prefixed by one of the letters:
 - `p` for "probability", the cumulative distribution function (c. d. f.)
 - `q` for "quantile", the inverse c. d. f.
 - `d` for "density", the density function (p. f. or p. d. f.)
 - `r` for "random", a random variable having the specified distribution

Notation

$$\mathcal{N}(\mu, \sigma^2)$$

Parameters

- $\mu \in \mathbb{R}$ = mean (location)
- $\sigma^2 > 0$ = variance (squared scale)

Properties

- Median: μ
- Mode: μ
- Variance: σ^2
- Probability density function (PDF): $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

Mean and standard deviation

The base R functions `mean()` and `sd()` compute the mean and standard deviation of a distribution

To demonstrate, let us first generate a vector of random, normally distributed, values.

```
set.seed(1)
x <- rnorm(n = 100, mean = 2, sd = 5)
```

We can then use that vector to demonstrate the functions.

```
mean(x)
```

```
## [1] 2.544437
```

```
sd(x)
```

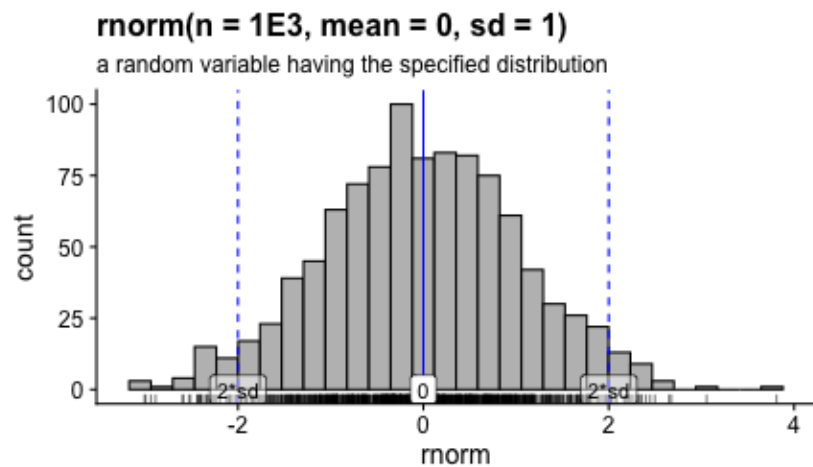
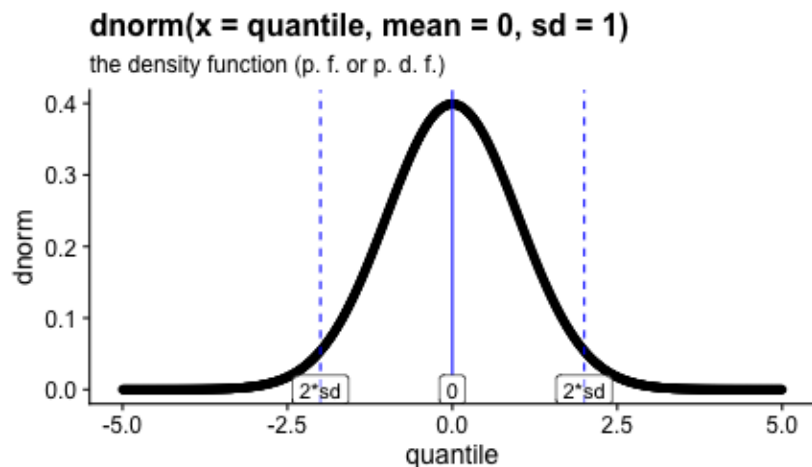
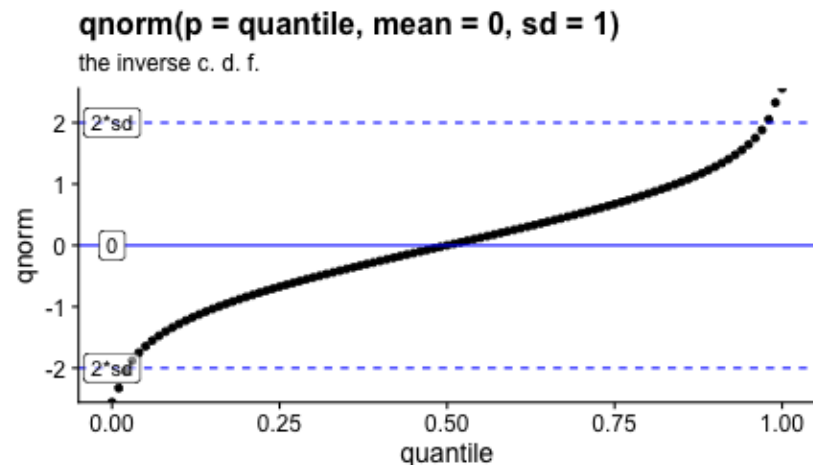
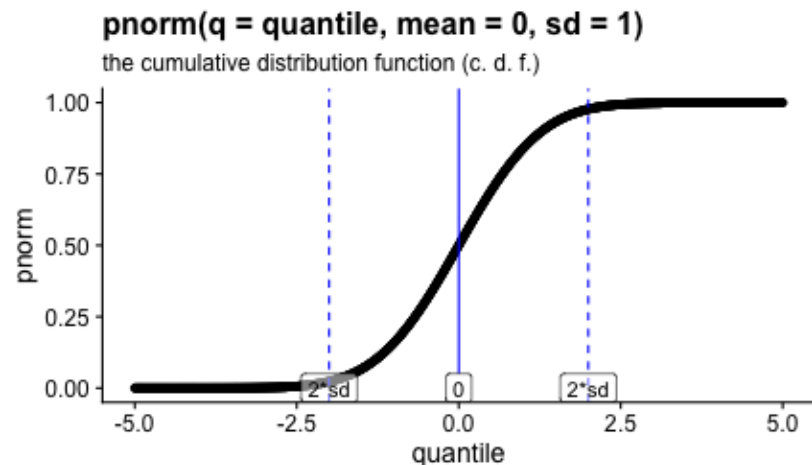
```
## [1] 4.490997
```

What are optional arguments for those functions?

Why do you think the mean and standard deviation are not exactly those that we would expect?

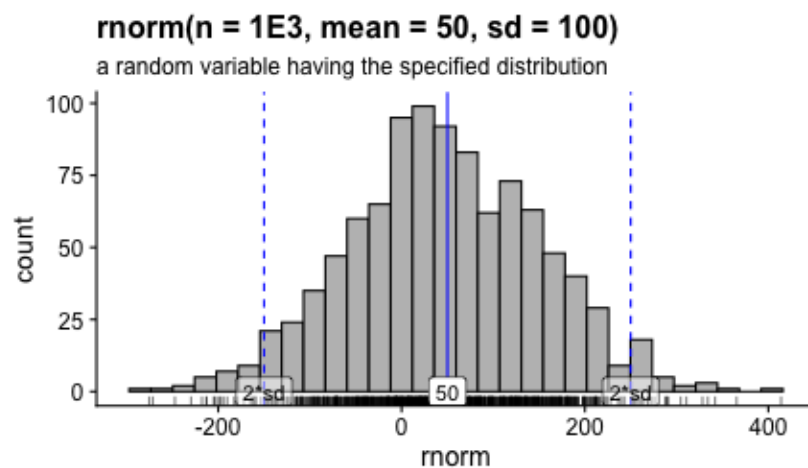
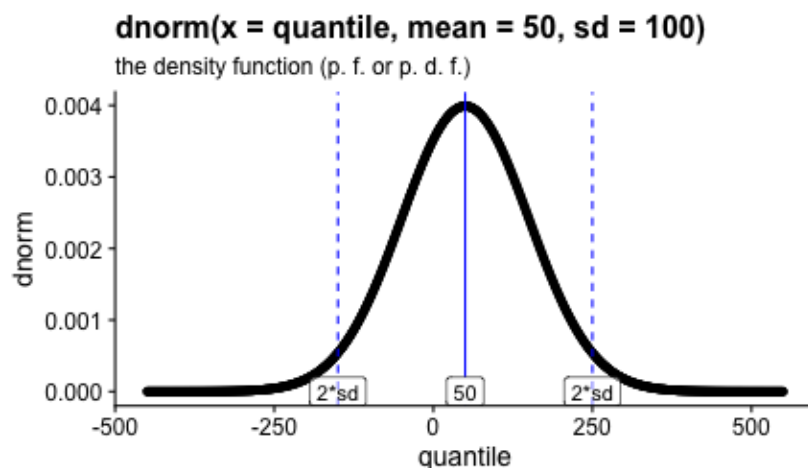
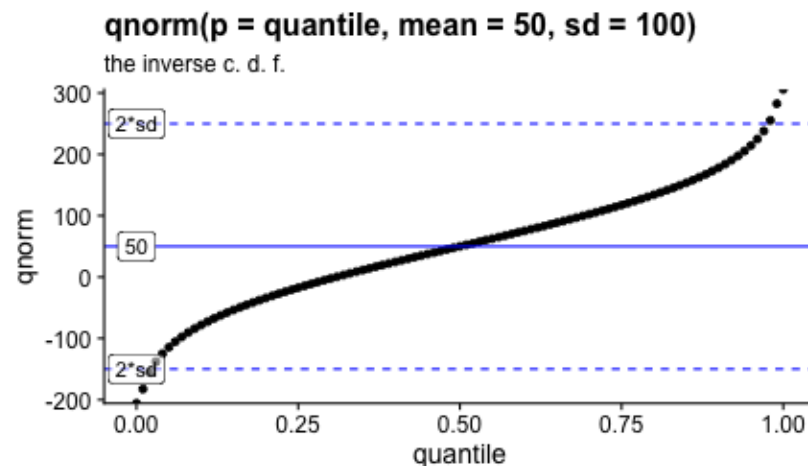
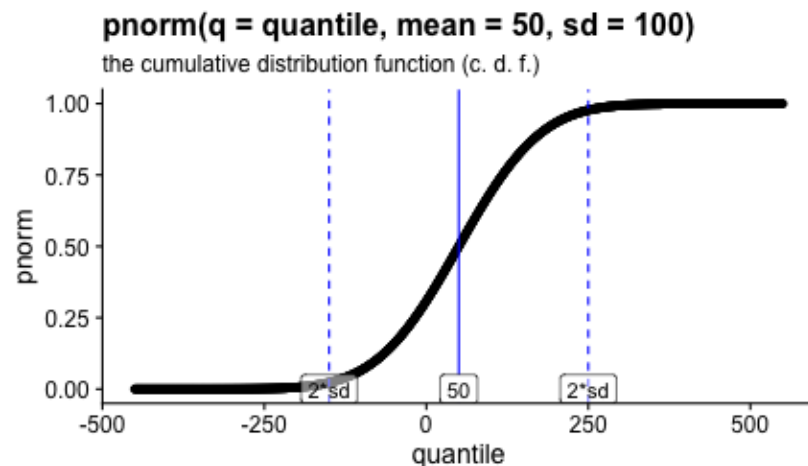
The standard normal distribution

Standard normal distribution with mean 0 and standard deviation 1.



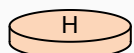
A parameterised normal distribution

Normal distribution parameterised with mean 50 and standard deviation 100.

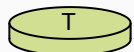


The binomial distribution

$P(H) = 50\%$



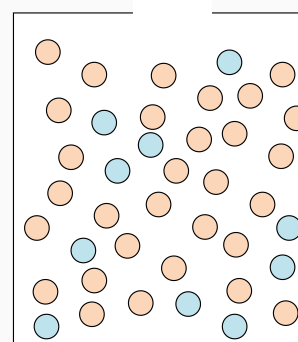
$P(T) = 50\%$



$P(4H \text{ in } 6 \text{ coins tossed}) = ?$

- Two mutually exclusive outcomes
 - $P(H) = 1 - P(T)$
- Size of experiment
 - e.g., coins tossed, balls drawn.
- Model number of occurrences of a specific outcome.

● 30 ● 10

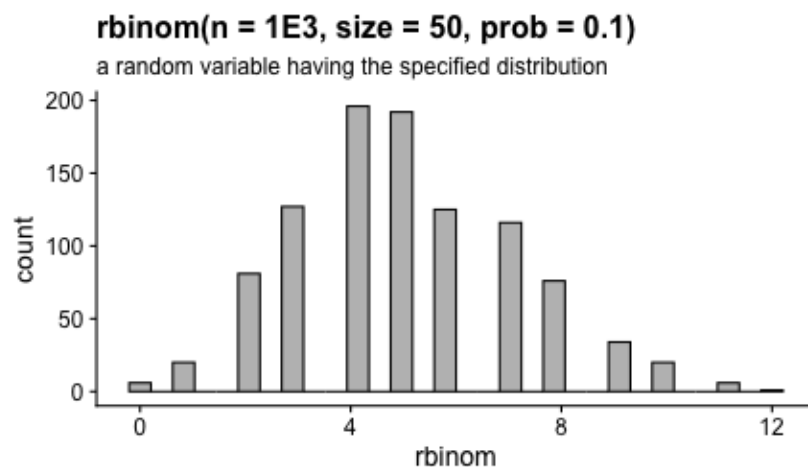
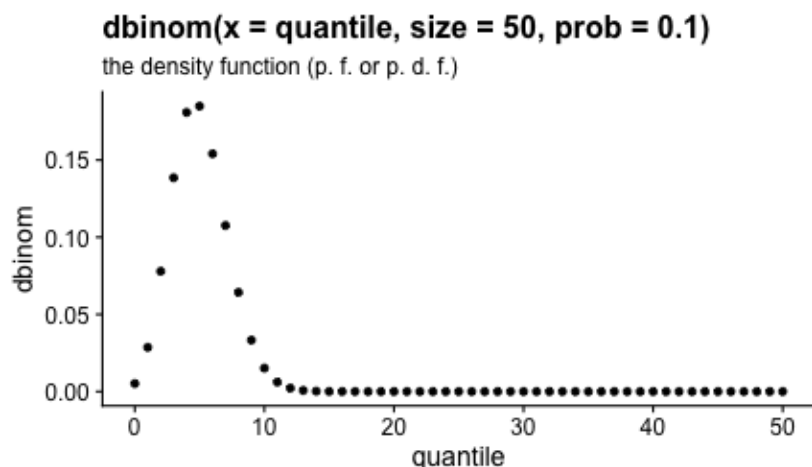
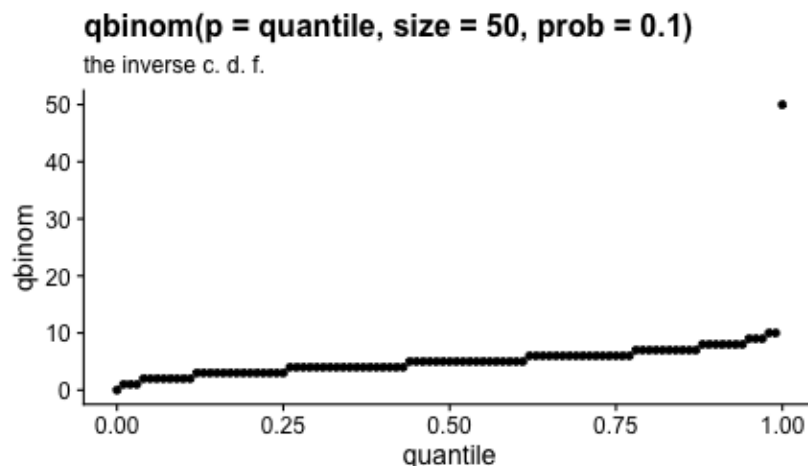
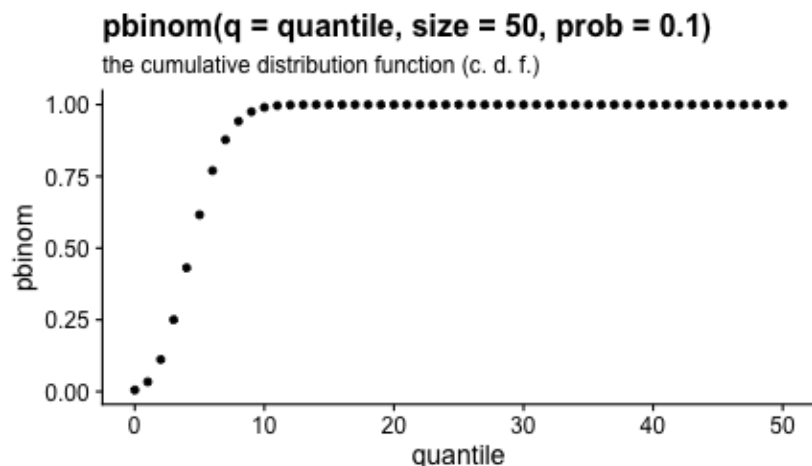


● ● ● ● ●

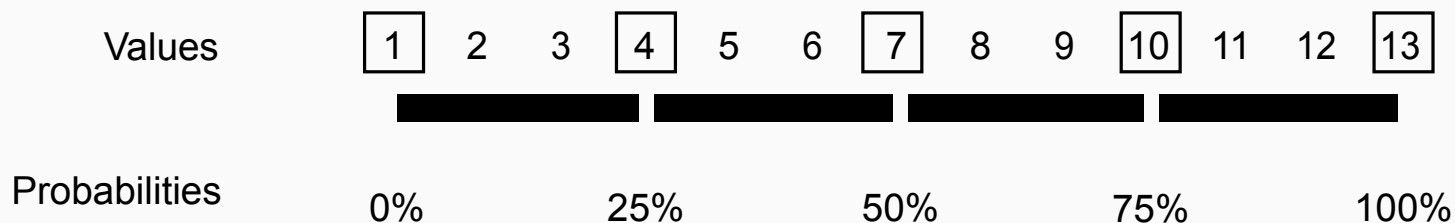
$P(2 \text{ in } 6 \text{ balls}) = ?$

A parameterised binomial distribution

Binomial distribution parameterised with size 50 and probability 0.1. This distribution models an experiment where a coin is tossed 50 times, and the probability of observing head is 10%.



Quantiles are the values at a selected set of evenly-spaced locations in the distribution.

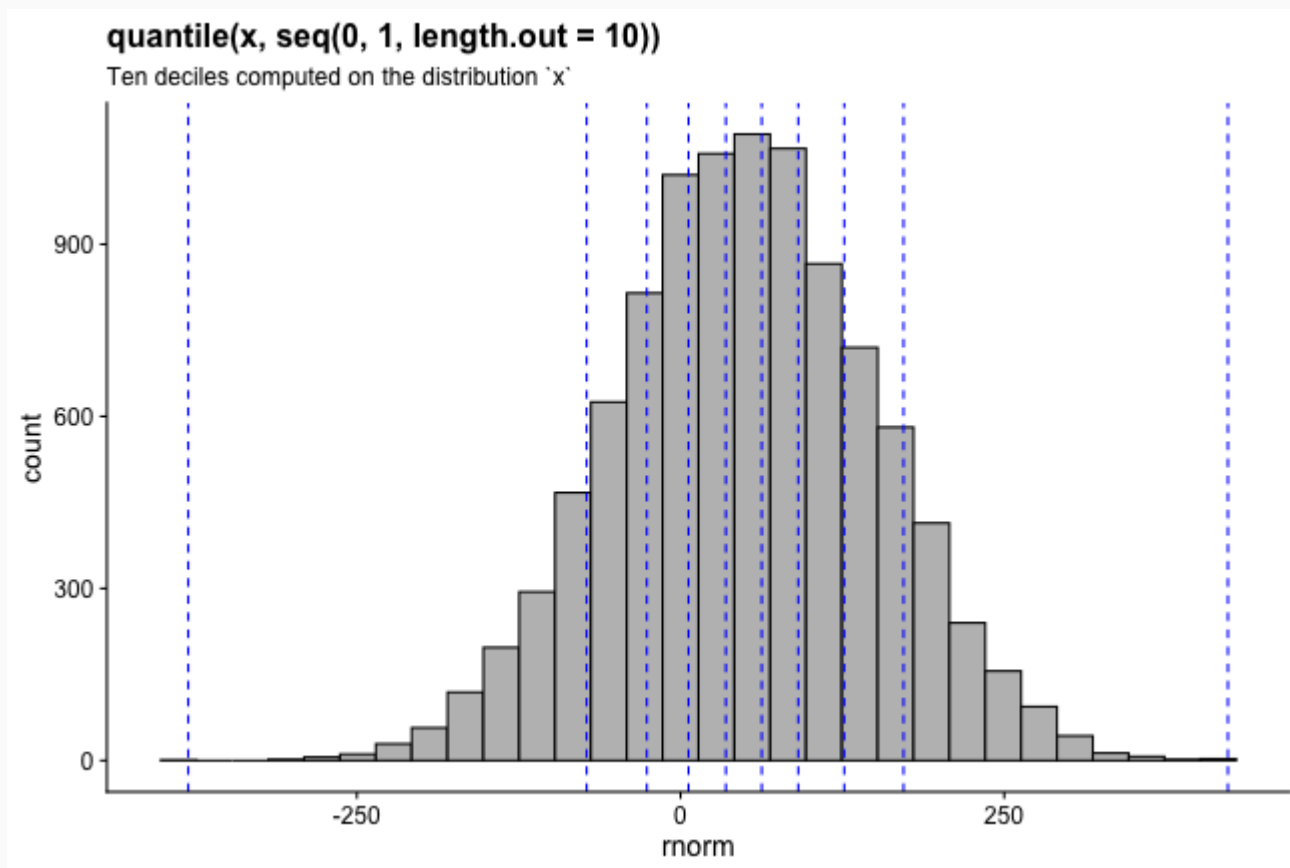


Quantiles are not always observed values.



Example - Quantiles of a normal distribution

For instance, the minimum value, the value that separates the lowest 10% values in the distribution, 20%, and so on, until the maximum value.



Generate and summarise a distribution

- Generate a vector of 1,000 normally distributed values with mean 10 and standard deviation 5.
- Inspect the output of the `summary()` function for that vector.
- Compute the mean and standard deviation for those values.
- Compute the deciles (i.e. 10 evenly spaced quantiles) for those values.
- Visualise the distribution of those values as a histogram.
- Visualise as vertical lines on the histogram: the mean (red solid), median (red dashed), one standard deviation below and above the mean (blue solid), and one median absolute deviation below and above the median (blue dashed).
- Generate a new vector with *a lot* more values (e.g., one million). Draw again a histogram. How does the distribution compare with more data points?

Query distributions and probabilities

For the standard normal distribution $\mathcal{N}(\mu = 0, \sigma^2 = 1)$:

- Plot the cumulative distribution function in the range $[-5, 5]$ in 0.1 increment.
- Plot the inverse cumulative distribution function for quantiles in 0.01 increment.
- Plot the density function in the range $[-5, 5]$ in 0.1 increment.
- What is the probability of observing a value greater than 2?
- What is the probability of observing a value between -2 and 2?
- What is the probability of observing a value more extreme than -2 or 2?

Empirical Cumulative Distribution Function

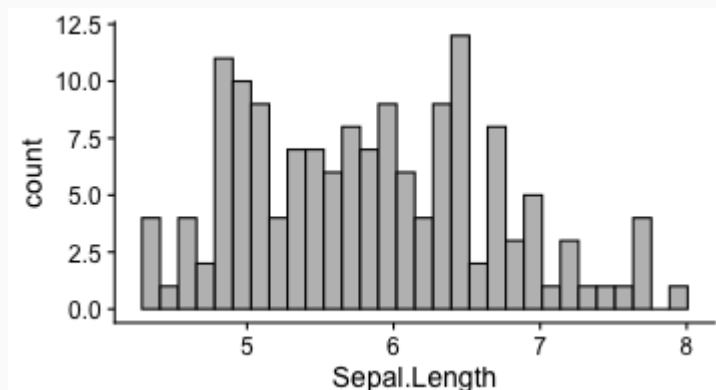
The `ecdf()` function computes an empirical cumulative distribution function.

It produces an object that can be plotted, printed, and used for further computations.

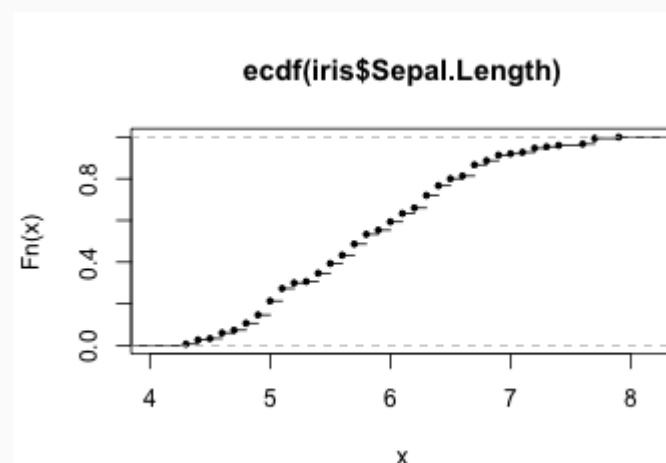
```
ecdf_iris_sepal_length <- ecdf(iris$Sepal.Length)
ecdf_iris_sepal_length
```

```
## Empirical CDF
## Call: ecdf(iris$Sepal.Length)
## x[1:35] = 4.3, 4.4, 4.5, ..., 7.7, 7.9
```

```
ggplot(iris, aes(Sepal.Length)) +
  geom_histogram(color = "black", fill = "grey") +
  cowplot::theme_cowplot()
```



```
plot(ecdf_iris_sepal_length, cex = 0.5)
```



Knots are the unique values observed in the empirical distribution. For ease of inspection, the function `knots()` automatically sorts knots in increasing order.

```
knots(ecdf_iris_sepal_length)
```

```
## [1] 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6.0 6.1  
## [20] 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7.0 7.1 7.2 7.3 7.4 7.6 7.7 7.9
```

In other words, the `knots()` function is equivalent to identifying the unique values in the original vector and sorting them in increasing order.

```
sort(unique(iris$Sepal.Length))
```

```
## [1] 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6.0 6.1  
## [20] 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7.0 7.1 7.2 7.3 7.4 7.6 7.7 7.9
```

The `quantile()` function can be applied to `ecdf` objects.

```
quantile(ecdf_iris_sepal_length, c(0, 0.25, 0.5, 0.75, 1))
```

```
##    0%   25%   50%   75%  100%  
##  4.3   5.1   5.8   6.4   7.9
```

In this case, the same function `quantile()` can be applied to the original vector of data for the same result.

```
quantile(iris$Sepal.Length, c(0, 0.25, 0.5, 0.75, 1))
```

```
##    0%   25%   50%   75%  100%  
##  4.3   5.1   5.8   6.4   7.9
```

Is this choice convenient or confusing to you? Why?

In the *stats* package, functions that implement statistical tests have a name that ends in `.test`.

`ansari.test`, `bartlett.test`, `binom.test`, `Box.test`, `chisq.test`, `cor.test`, `fisher.test`,
`fligner.test`, `friedman.test`, `kruskal.test`, `ks.test`, `mantelhaen.test`, `mauchly.test`,
`mcnemar.test`, `mood.test`, `oneway.test`, `pairwise.prop.test`, `pairwise.t.test`,
`pairwise.wilcox.test`, `poisson.test`, `power.anova.test`, `power.prop.test`, `power.t.test`, `PP.test`,
`prop.test`, `prop.trend.test`, `quade.test`, `shapiro.test`, `t.test`, `var.test`, `wilcox.test`

Each of those functions comes with a help page with programmatic usage, statistical advice, and external references to published work.

```
?pairwise.t.test  
help(pairwise.t.test)
```

The five steps of hypothesis testing

General principles of hypothesis testing

Design	Experimental design Data summary function (e.g., mean) Test statistic
Null hypothesis	How is the test statistic distributed in the hypothesis is not true?
Rejection region	How extreme should the test statistics be to reject the null hypothesis?
Collect data	Do the experiment; compute the test statistic.
Decide	Is the test statistics in the rejection region?

Parametric tests and Non-parametric equivalents

When parametric assumptions are not met, non-parametric tests equivalent should be used.

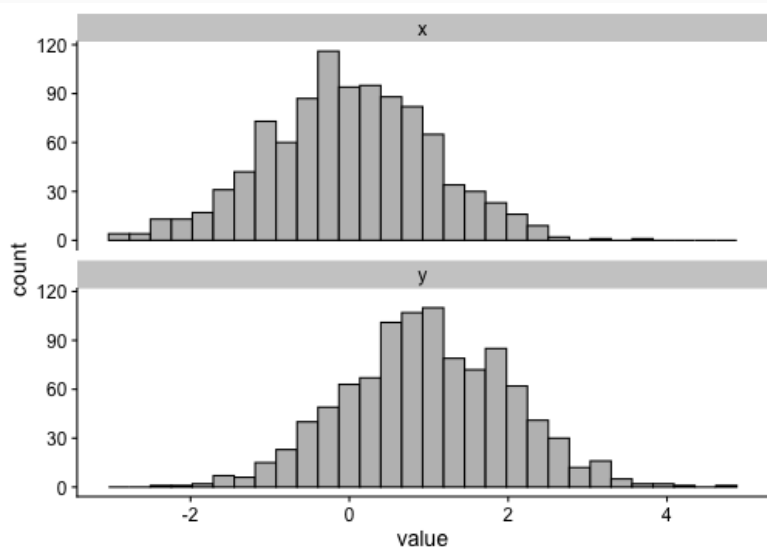
Parametric test	Non-parametric equivalent
Paired t-test	Wilcoxon Rank sum test
Unpaired t-test	Mann-Whitney U test
Pearson correlation	Spearman correlation
One-way Analysis of Variance	Kruskal-Wallis test

Non-parametric tests make fewer assumptions, as such:

- they have wider applicability.
- they may be applied in situations where less is known about the data.
- they are more robust.
- ..., however, fewer assumption gives non-parametric tests *less* power than their parametric equivalent.

Two normal distributions

```
set.seed(1)
x <- rnorm(n = 1000, mean = 0, sd = 1)
y <- rnorm(n = 1000, mean = 1, sd = 1)
```



Unpaired t-test

```
t.test(value ~ group, test_data)
```

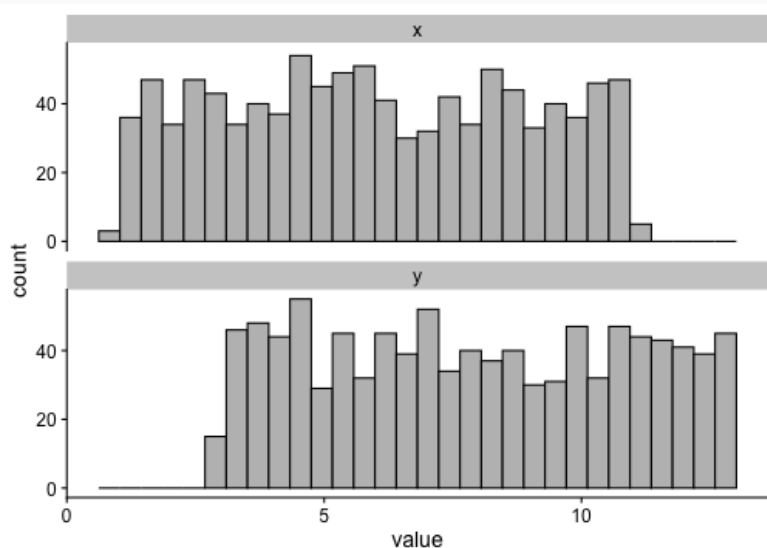
```
##
##      Welch Two Sample t-test
##
## data:  value by group
## t = -21.454, df = 1998, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group x and group y is not
## 95 percent confidence interval:
##  -1.0863763 -0.9043961
## sample estimates:
## mean in group x mean in group y
##      -0.01164814      0.98373809
```

Compare with

```
t.test(x, y)
t.test(y, x)
```

Two uniform distributions

```
set.seed(1)
x <- runif(n = 1000, min = 1, max = 11)
y <- runif(n = 1000, min = 3, max = 13)
```



Mann-Whitney U test

```
wilcox.test(value ~ group, test_data)
```

```
##
##      Wilcoxon rank sum test with continuity correction
##
## data:  value by group
## W = 329285, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Directed hypothesis

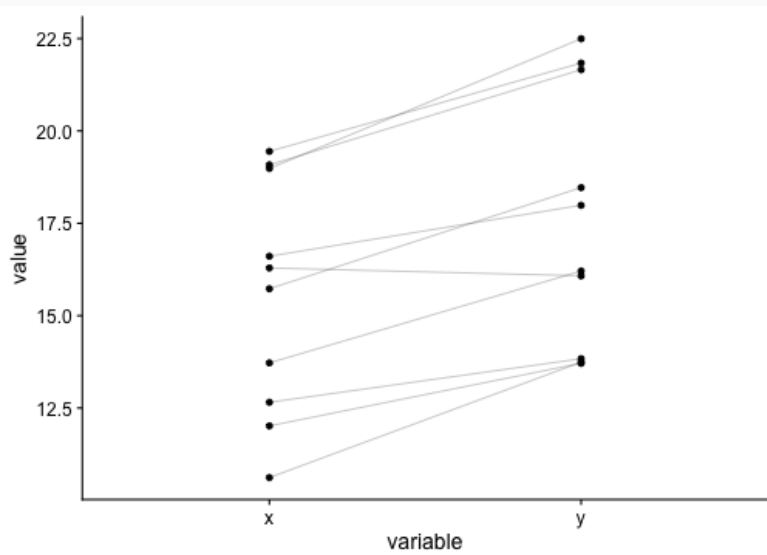
```
wilcox.test(value ~ group, test_data, alternative = "less")
```

```
##
##      Wilcoxon rank sum test with continuity correction
##
## data:  value by group
## W = 329285, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
```


Paired test

For each sample, the two measurements are related to one another; e.g. patients measured before and after a treatment.

```
set.seed(1)
n_sample <- 10
x <- runif(n = n_sample, min = 10, max = 20)
y <- x + 2 + rnorm(n = n_sample, mean = 0, sd = 1)
```



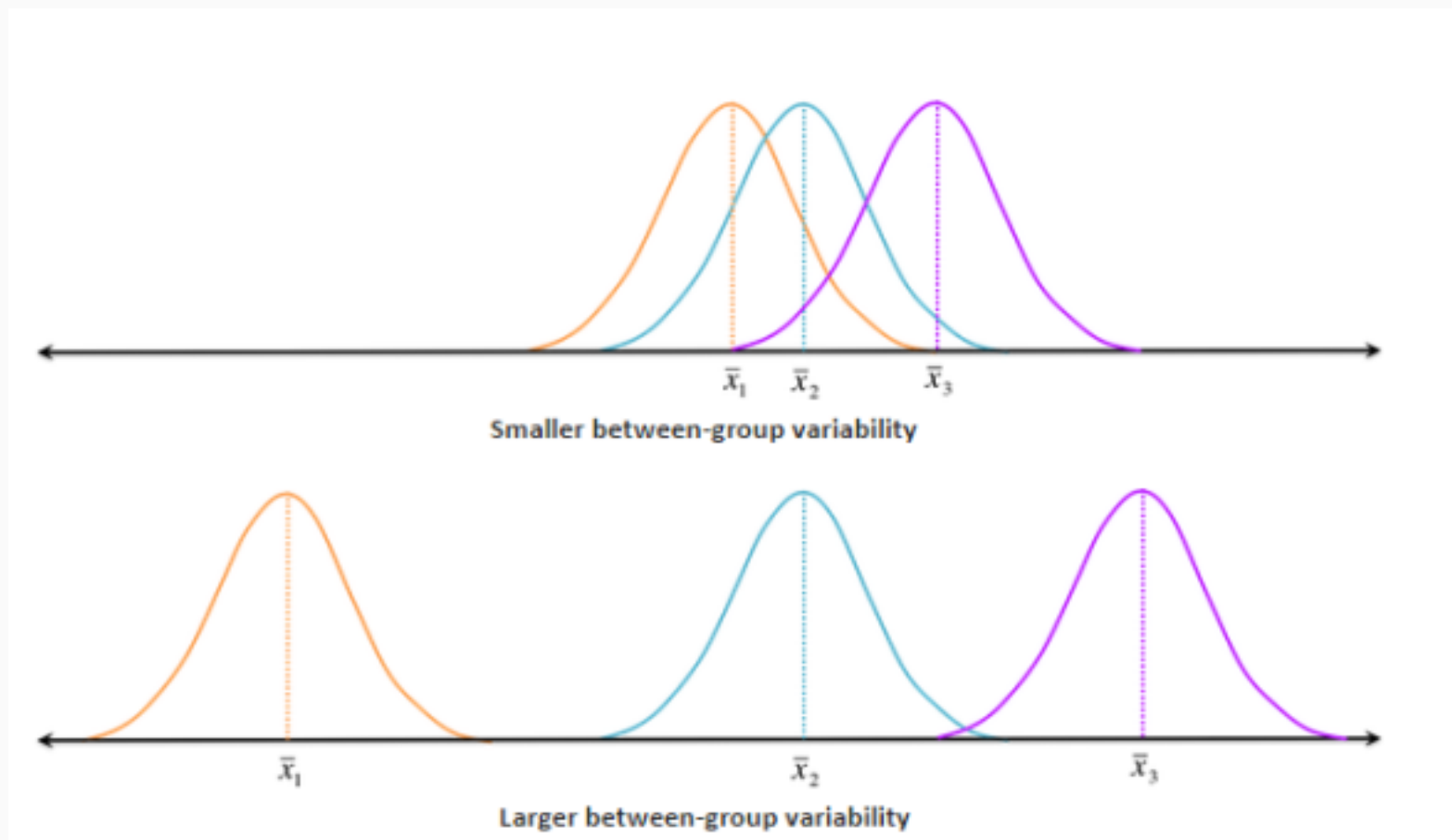
```
t.test(x, y, paired = TRUE)
```

```
##
##   Paired t-test
##
## data:  x and y
## t = -6.0238, df = 9, p-value = 0.0001967
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -2.870241 -1.303017
## sample estimates:
## mean difference
##      -2.086629
```

Note: What is actually tested is whether the mean of the differences between the paired (x) and (y) measurements is different from 0.

Analysis of Variance (ANOVA)

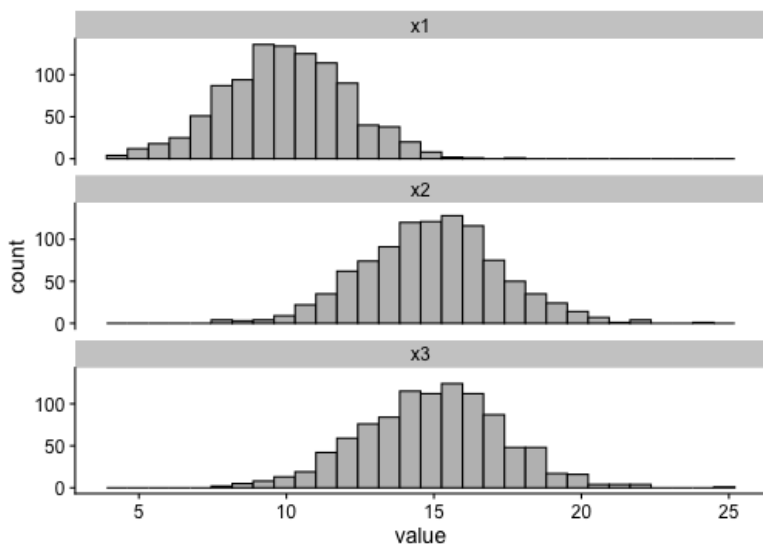
Is the variance *between* groups larger than the variance *within* groups?



Liguori Gabriel Romero AND Moreira (2018)

Analysis of Variance (ANOVA)

```
set.seed(1)
n_sample <- 1000
x1 <- rnorm(n = n_sample, mean = 10, sd = 2)
x2 <- x1 + 5 + rnorm(n = n_sample, mean = 0, sd = 1)
x3 <- x2 + 0 + rnorm(n = n_sample, mean = 0, sd = 0.5)
test_data <- bind_rows(
  tibble(group = "x1", value = x1),
  tibble(group = "x2", value = x2),
  tibble(group = "x3", value = x3)
)
```



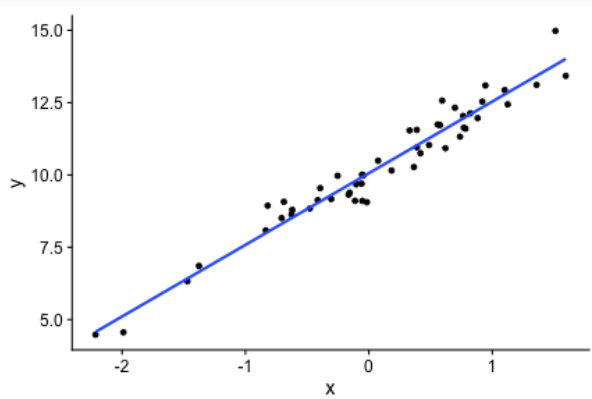
```
out <- aov(value ~ group, test_data)
out
```

```
## Call:
## aov(formula = value ~ group, data = test_data)
##
## Terms:
##              group Residuals
## Sum of Squares 16583.9 15450.7
## Deg. of Freedom      2    2997
##
## Residual standard error: 2.270548
## Estimated effects may be unbalanced
```

```
summary(out)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## group          2  16584    8292    1608 <2e-16 ***
## Residuals    2997  15451         5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Describe a continuous response variable as a function of one or more predictor variables.



- What is the slope?
- What is the intercept?

Null hypotheses:

- The slope is equal to 0.
- The intercept is equal to 0.

```
lm(y ~ x, test_data)
```

```
##  
## Call:  
## lm(formula = y ~ x, data = test_data)  
##  
## Coefficients:  
## (Intercept)          x  
##      10.061       2.477
```

Linear models - summary

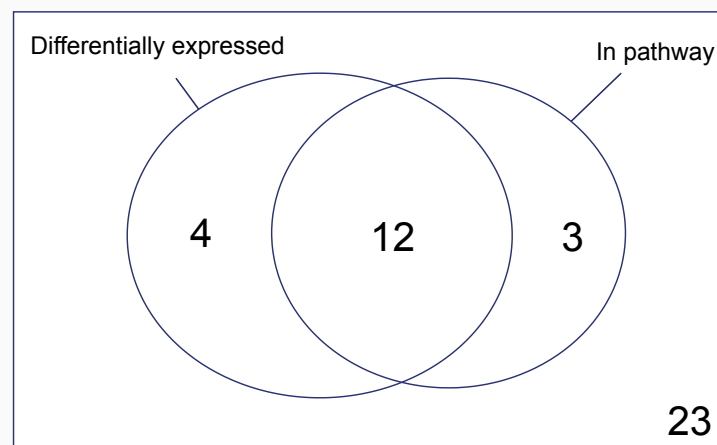
```
lm(y ~ x, test_data) %>% summary()

##
## Call:
## lm(formula = y ~ x, data = test_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96380 -0.33449 -0.00112  0.24384  1.17429
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.06095     0.06968  144.40  <2e-16 ***
## x           2.47723     0.08403   29.48  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4891 on 48 degrees of freedom
## Multiple R-squared:  0.9477,    Adjusted R-squared:  0.9466
## F-statistic:   869 on 1 and 48 DF,  p-value: < 2.2e-16
```

Fisher's Exact Test

- Test of independence between two categorical variables
- Alternative to the Chi-square test when the sample is not large enough.
 - Rule of thumb: when any of the *expected* values in the contingency table is less than 5.
 - e.g., Gene set over-representation analysis (ORA)

	Differentially expressed	Not Differentially expressed	Total
In pathway	12	3	15
Not in pathway	4	23	27
Total	16	26	42



Total = 42 genes in the genome

Further reading: [Towards data science](#)

Fisher's Exact Test

	DE	Not DE	Total
In pathway	a	b	$a + b$
Not in pathway	c	d	$c + d$
Total	$a + c$	$b + d$	$a + b + c + d (= n)$

What is the probability of observing a given distribution?

$$p = \frac{\binom{a+b}{a} \binom{c+d}{c}}{\binom{n}{a+c}} = \frac{\binom{a+b}{b} \binom{c+d}{d}}{\binom{n}{b+d}} = \frac{(a+b)! (c+d)! (a+c)! (b+d)!}{a! b! c! d! n!}$$

Fisher's Exact Test

	in_pathway	not_pathway
DE	12	3
Not DE	4	23

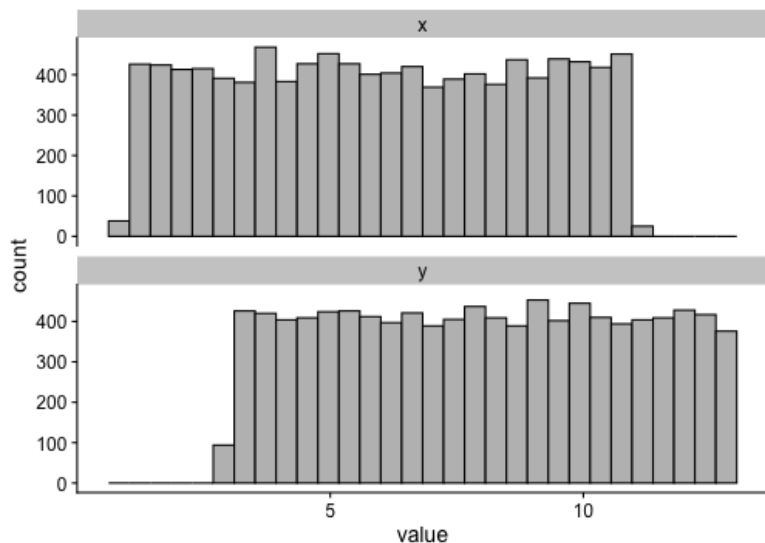
```
fisher.test(x_table)
```

```
##  
##      Fisher's Exact Test for Count Data  
##  
## data:  x_table  
## p-value = 4.983e-05  
## alternative hypothesis: true odds ratio is not equal to 1  
## 95 percent confidence interval:  
##    3.592731 170.706615  
## sample estimates:  
## odds ratio  
##    20.56889
```


Beware of interpreting inadequate tests!

Two uniform distributions

```
set.seed(1)
n_size <- 10E3
x <- runif(n = n_size, min = 1, max = 11)
y <- runif(n = n_size, min = 3, max = 13)
```

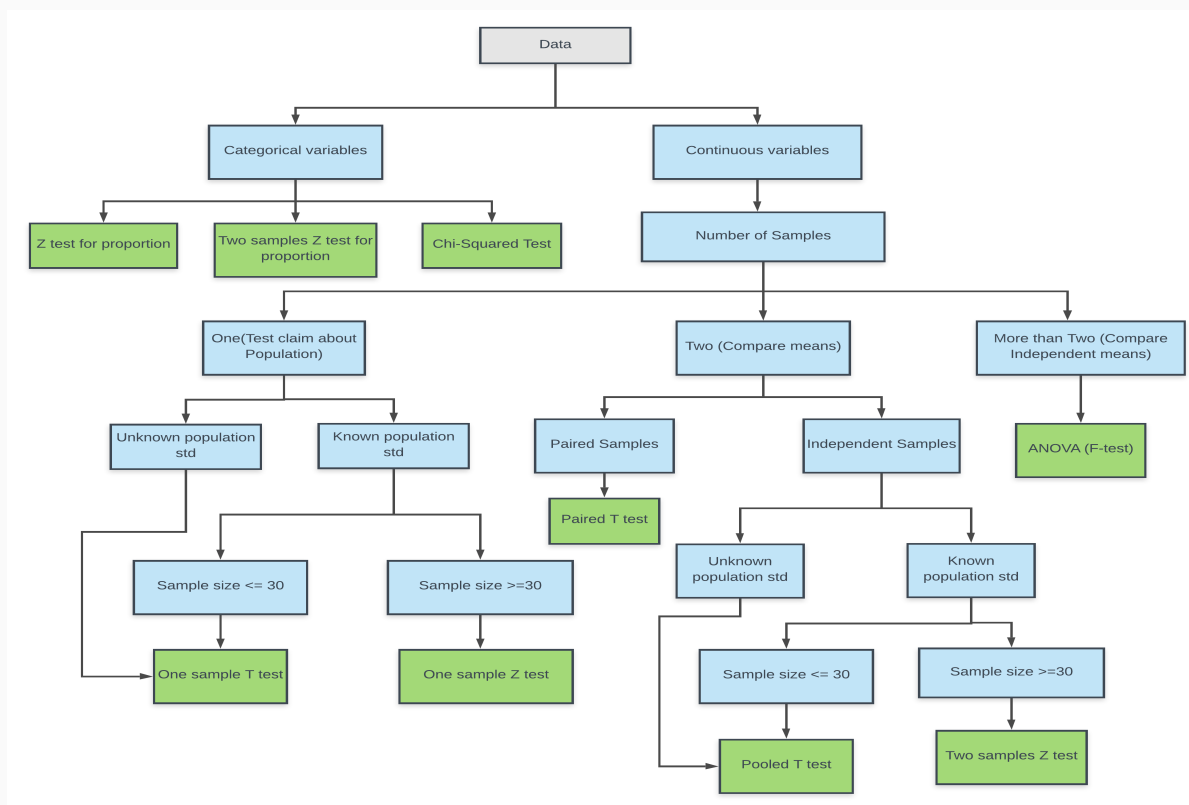


Parametric (unpaired) t-test

```
t.test(value ~ group, test_data)
```

```
##
##      Welch Two Sample t-test
##
## data:  value by group
## t = -48.793, df = 19997, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group x and group y is not
## 95 percent confidence interval:
##  -2.080581 -1.919876
## sample estimates:
## mean in group x mean in group y
##      6.001680      8.001909
```

Choosing a test

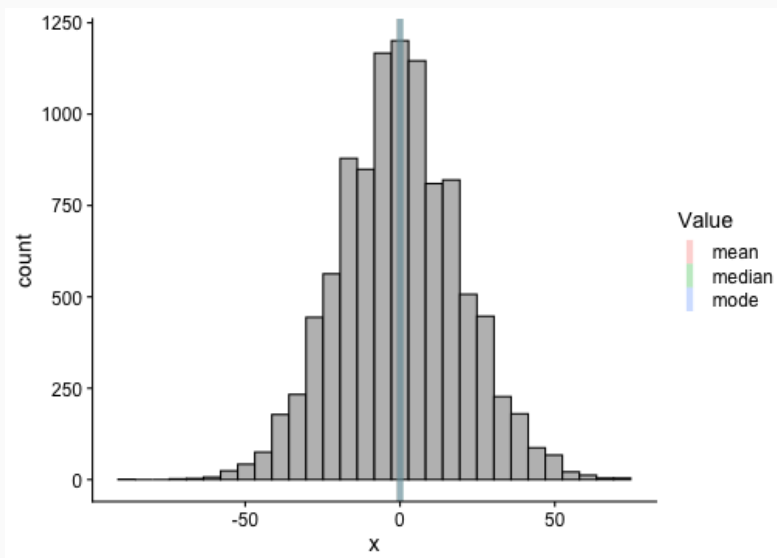


Liguori Gabriel Romero AND Moreira (2018)

See also <https://stats.idre.ucla.edu/other/mult-pkg/whatstat/>

Tests make assumptions that must be met to for the results to be interpreted properly and with validity.

For instance, Student's t-Test expects values to be located around a central or typical value.



Measures of central tendency include:

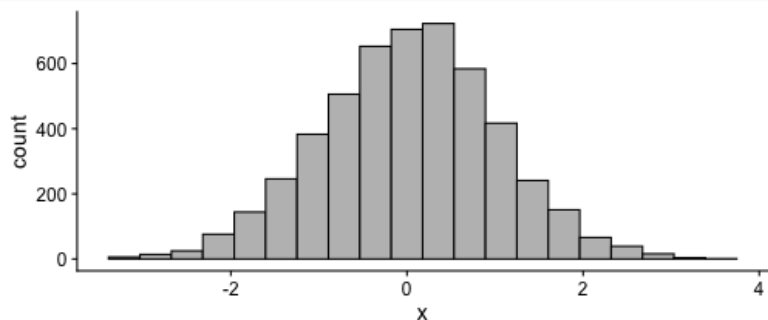
- the arithmetic mean
- the median
- ...

Knowledge assumptions - Normality

In addition, Student's t-Test also expects values to be normally distributed.

Normal distribution

```
x <- rnorm(n = 5000, mean = 0, sd = 1)
```

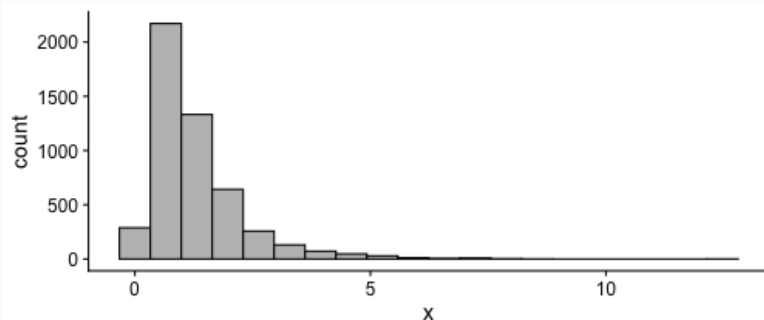


```
shapiro.test(x)
```

```
##  
##      Shapiro-Wilk normality test  
##  
## data:  x  
## W = 0.99947, p-value = 0.1682
```

Log-normal distribution

```
x <- 2^rnorm(n = 5000, mean = 0, sd = 1)
```



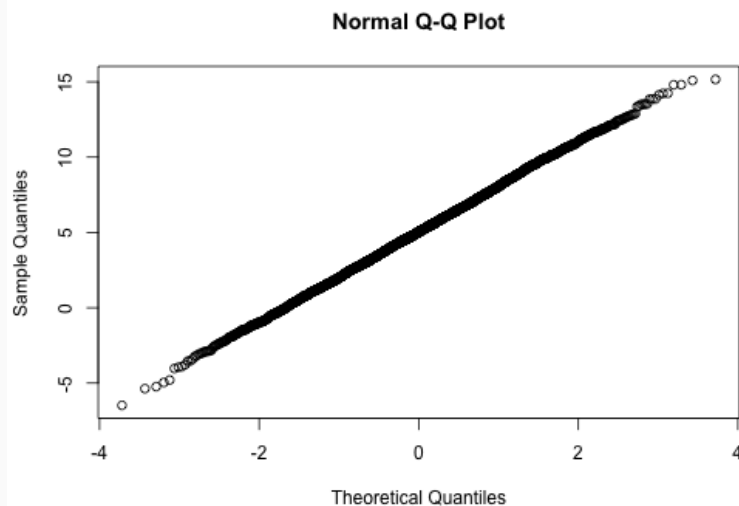
```
shapiro.test(x)
```

```
##  
##      Shapiro-Wilk normality test  
##  
## data:  x  
## W = 0.79333, p-value < 2.2e-16
```

The Quantile-Quantile Plots (QQ plot) contrasts the quantiles of the observed distribution to those of a theoretical distribution.

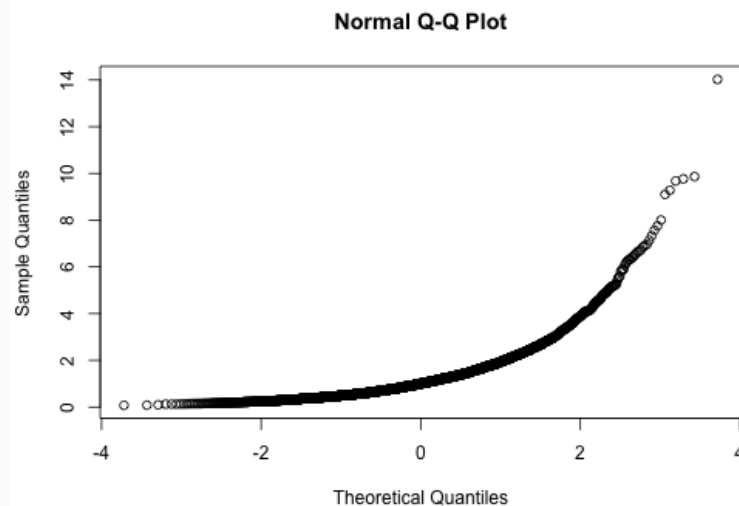
Normal distribution

```
x <- rnorm(n = 5000, mean = 5, sd = 3)
qqnorm(x)
```



Log-normal distribution

```
x <- 2^rnorm(n = 5000, mean = 0, sd = 1)
qqnorm(x)
```



Hypothesis

"Jelly beans cause acne."

Results

- No link between jelly beans and acne.
- No link between *brown* jelly beans and acne.
- No link between *pink* jelly beans and acne.
- ...
- Link between *green* jelly beans and acne.

News

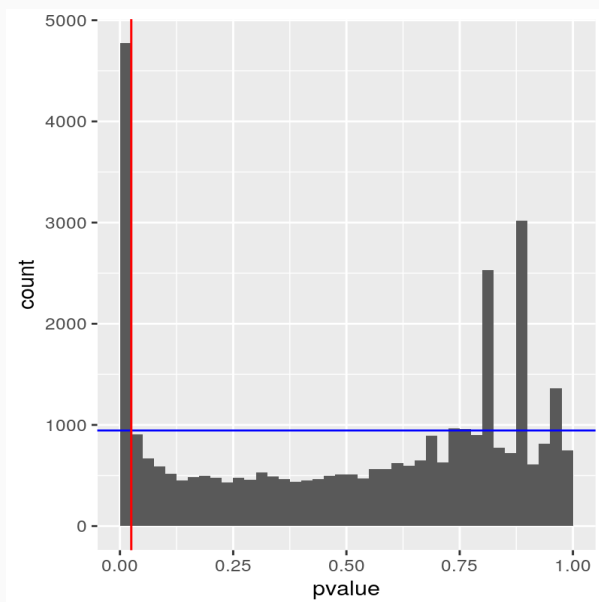
Green jelly beans linked to acne! 95% confidence! Only 5% chance of coincidence!

<https://xkcd.com/882/>

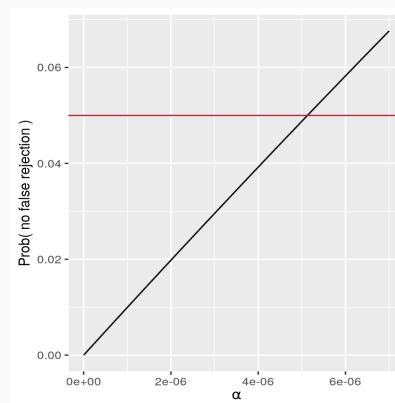
Multiple-testing correction

Distribution of p -values in an RNA-seq differential expression experiment

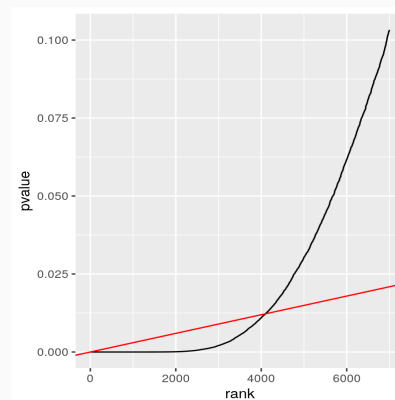
- True positive
- True negative
- False positive (type I error)
- False negative (type I error)



Bonferroni correction

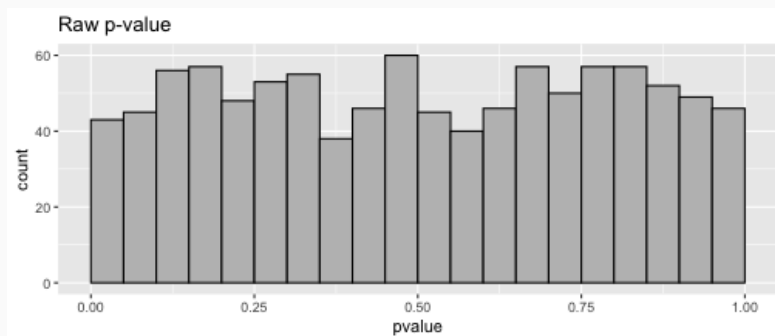
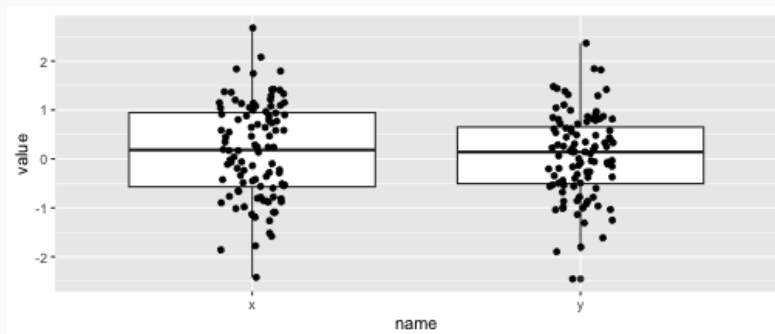


Benjamini-Hochberg procedure

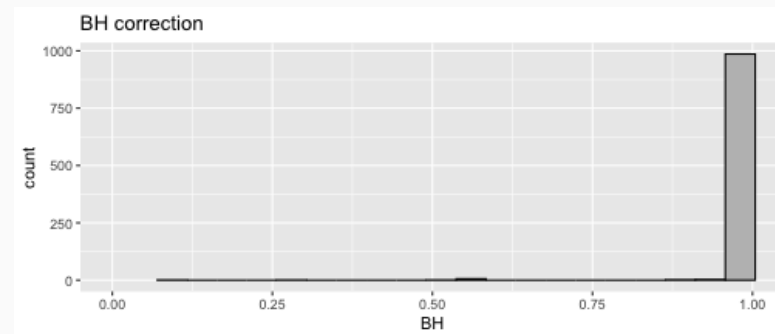


Multiple-testing correction

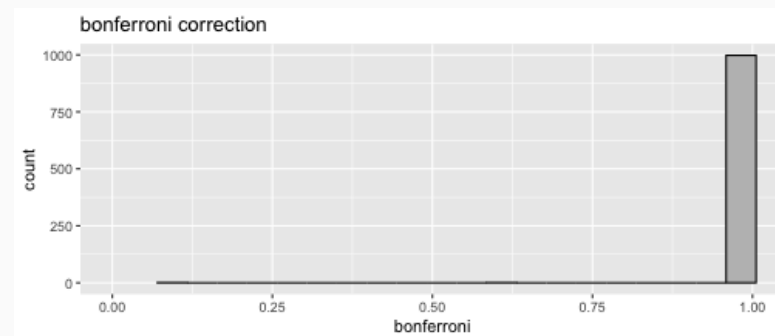
Let us carry 1000 tests between two normal distributions of mean 0 and standard deviation 1.



There are 43 out of 1000 raw p-values smaller than 0.05



There are 0 BH-corrected p-values smaller than 0.05



There are 0 bonferonni corrected p-values smaller than 0.05

Multiple-testing correction in

The `p.adjust()` function can run a number of methods for multiple testing correction.

The `p.adjust.methods` object list the available methods.

The function takes a vector of p-values and the name of a method.

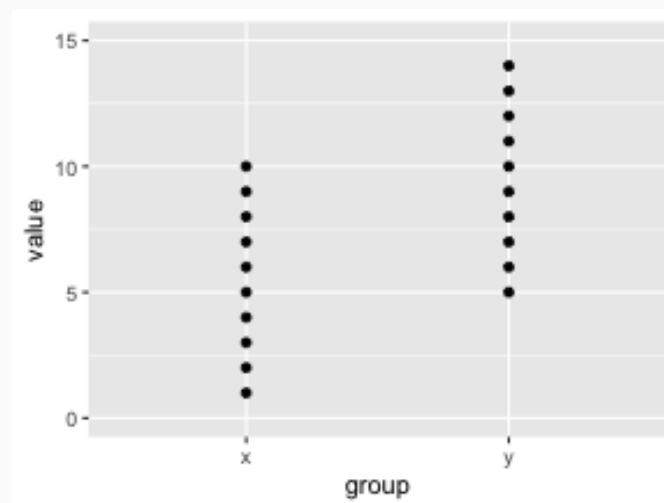
```
p.adjust(p_values, method = "bonferroni")
```

Statistical tests

Wilcoxon test

- Run a Wilcoxon test to compare the vectors `1:10` and `5:14`.

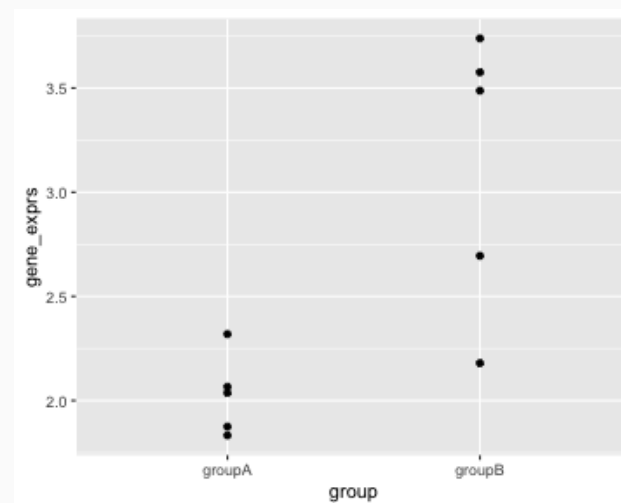
What is the p-value?



T-test

- Import the data in the file `gene_exprs.csv`.
- Run a t-test to compare the gene expression values between the two groups.

What is the p-value?



Testing & Multiple testing correction

Given an Excel file that contains a matrix of log-normalised counts (`logcounts`) and experimental metadata (`cell_info`), test each gene (i.e., row) in the matrix for differential expression between the two experimental groups. Start by importing the `logcounts` table and converting it to a matrix.

Approach

1. Write the code to test a single gene and access the p-value.
2. Write a function that generalises the code to test any one gene and return the p-value.
3. Use the function `vapply` to test every row in the matrix and collect a vector of p-values.

Bonus points

- Visualise a histogram of the p-values.
- Correct p-values for multiple testing. How many genes remain before and after multiple testing?
- Use `gene_info` to get the gene name for the gene identifier with the smallest p-value.

Illustration

Gene expression matrix

	sample1	sample2	...
gene1			
gene2			
...			

Sample metadata

sample	group
sample1	A
sample2	A
...	...

Over-representation analysis (ORA)

Given the list of genes (Ensembl gene identifiers) that you identified as differentially expressed in the previous exercise, and a list of gene sets (`go_db`), test each gene set for over-representation of differentially expressed genes. Start by importing the Gene Ontology table and converting it to a list.

Approach

1. Write the code to test a single gene set and access the p-value.
2. Write a function that generalises the code to test any one gene set and return the p-value.
3. Use the function `vapply` to test every gene set in the list and collect a vector of p-values.

Bonus points

- Visualise a bar plot of the p-values.
- Correct p-values for multiple testing. How many gene sets remain before and after multiple testing?
- Use `go_info` to annotate each GO gene set with its corrected p-value, and arrange the table by increasing p-value.

- [UCLouvain Bioinformatics Summer School 2019](#)
 - [Introduction to Statistics and Machine Learning](#) by Oliver M. Crook
 - [Practical: stats/ML](#)
- [CSAMA](#) by the European Molecular Biology Laboratory (EMBL).
- [Statistic with R and dplyr and ggplot](#) by Greg Martin
- [Susan Holmes - Introduction to Statistics for Biology and Biostatistics](#)
- Susan Holmes & Wolfgang Huber - [Modern Statistics for Modern Biology: Testing](#)
- [Bioconductor Case Studies](#)
- [Introduction to Econometrics with R](#)

Liguori Gabriel Romero AND Moreira, L. F. P. (2018). "Operating with Data - Statistics for the Cardiovascular Surgeon: Part III. Comparing Groups". En. In: *Brazilian Journal of Cardiovascular Surgery* 33, pp. V - X. ISSN: 0102-7638. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-76382018000600005&nrm=iso.