Statistics and Machine Learning in R

Kevin Rue-Albrecht
Oxford Biomedical Data Science Training Programme

2020-05-28 (updated: 2020-05-27)

Learning objectives

- Learn to use the builtin statistical distributions
- Learn to use the builtin statistical tests
- Run tests and interpret results
- Visualise data and test results

R is built for statistics

- R includes a number of common statistical distributions:
  - The Normal Distribution
  - The Binomial Distribution
  - The Poisson Distribution
  - ...
- R 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
  - ...

Functions for Probability Distributions

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

- 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

The normal distribution

Notation

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

Parameters

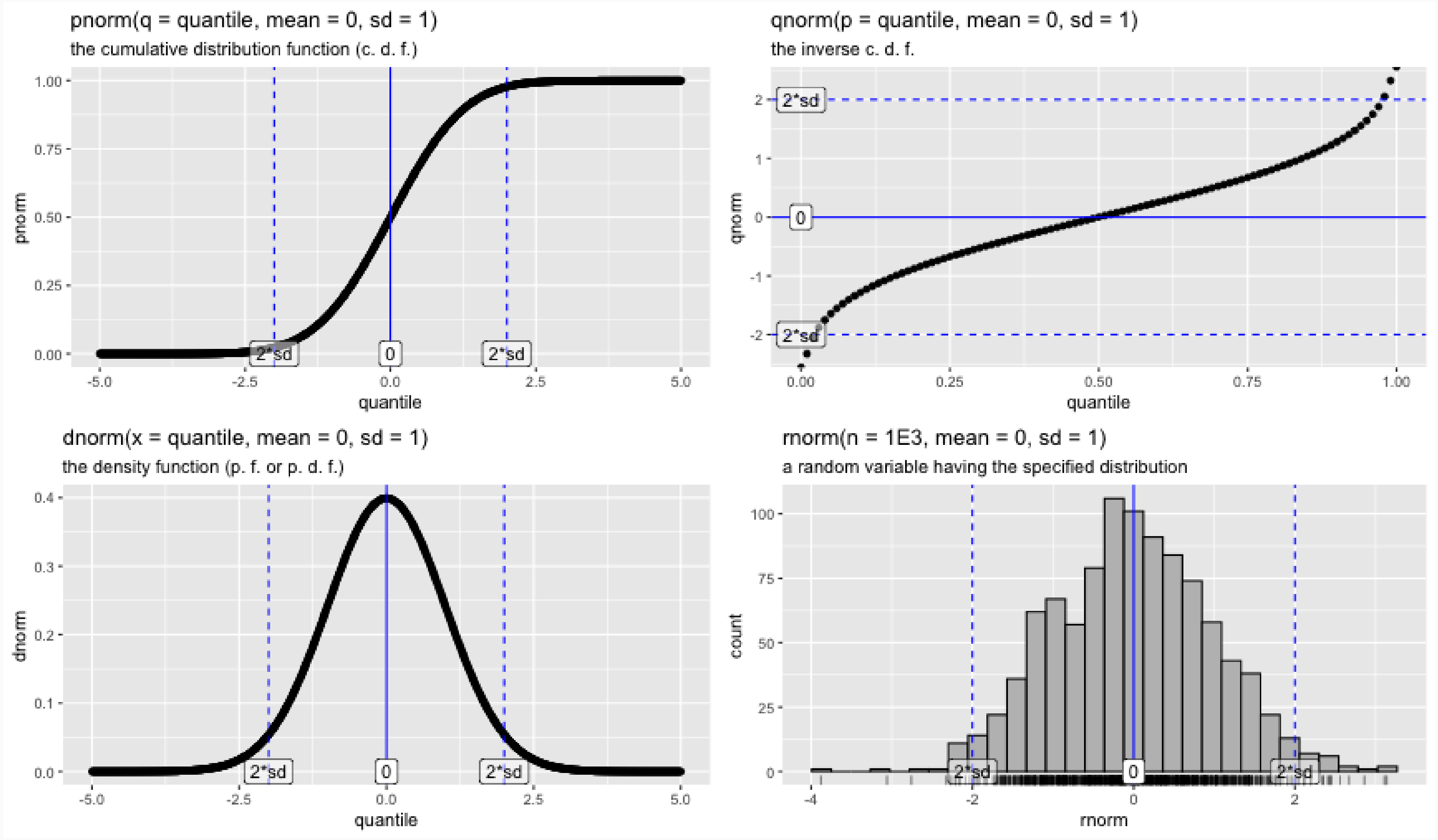
- $\mu \in \mathbb{R}$ = mean (location)
- $\sigma^2 > 0$ = variance (squared scale)

Properties

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

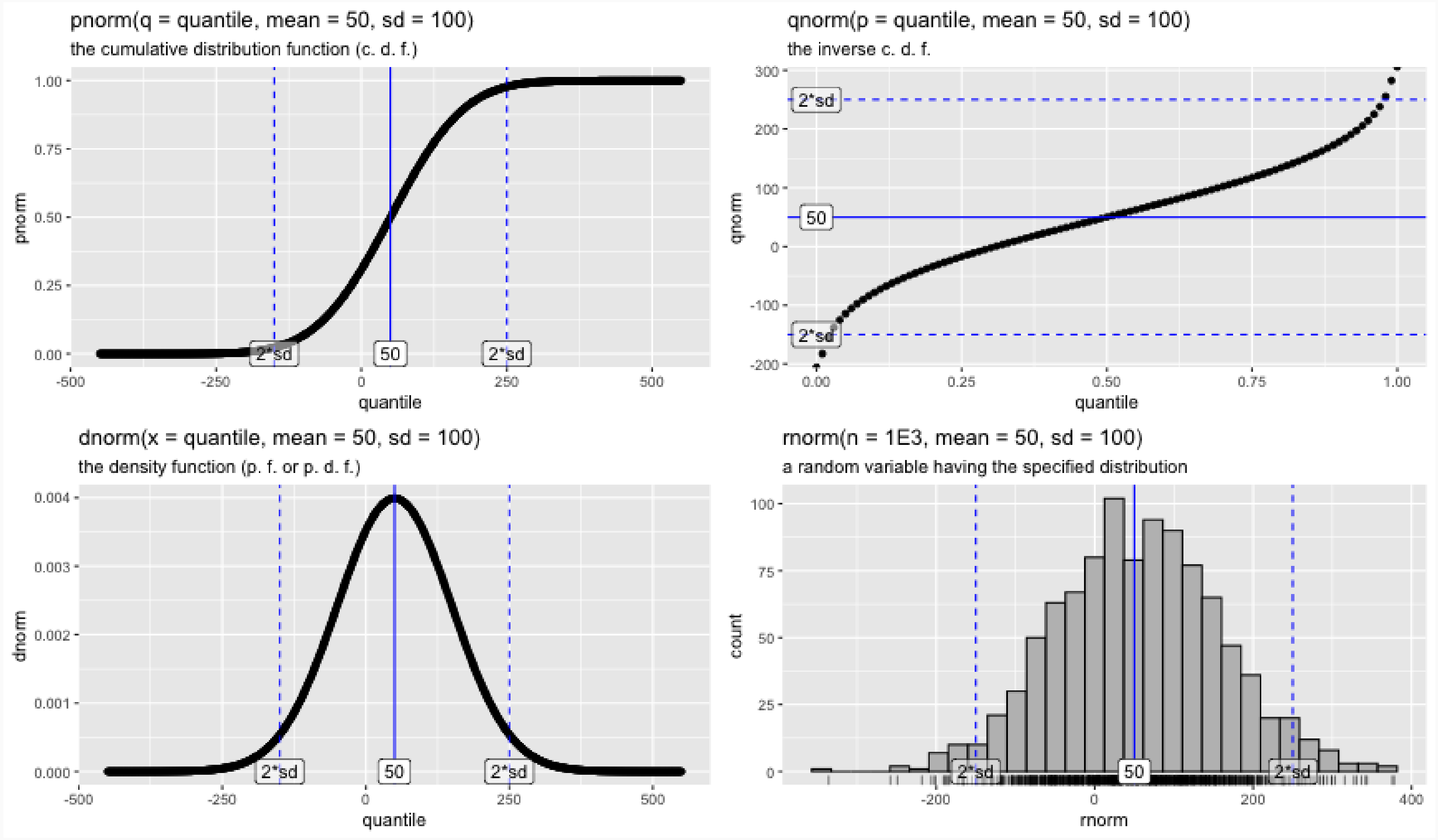
The standard normal distribution

Standard normal distribution with mean 0 and standard deviation 1.



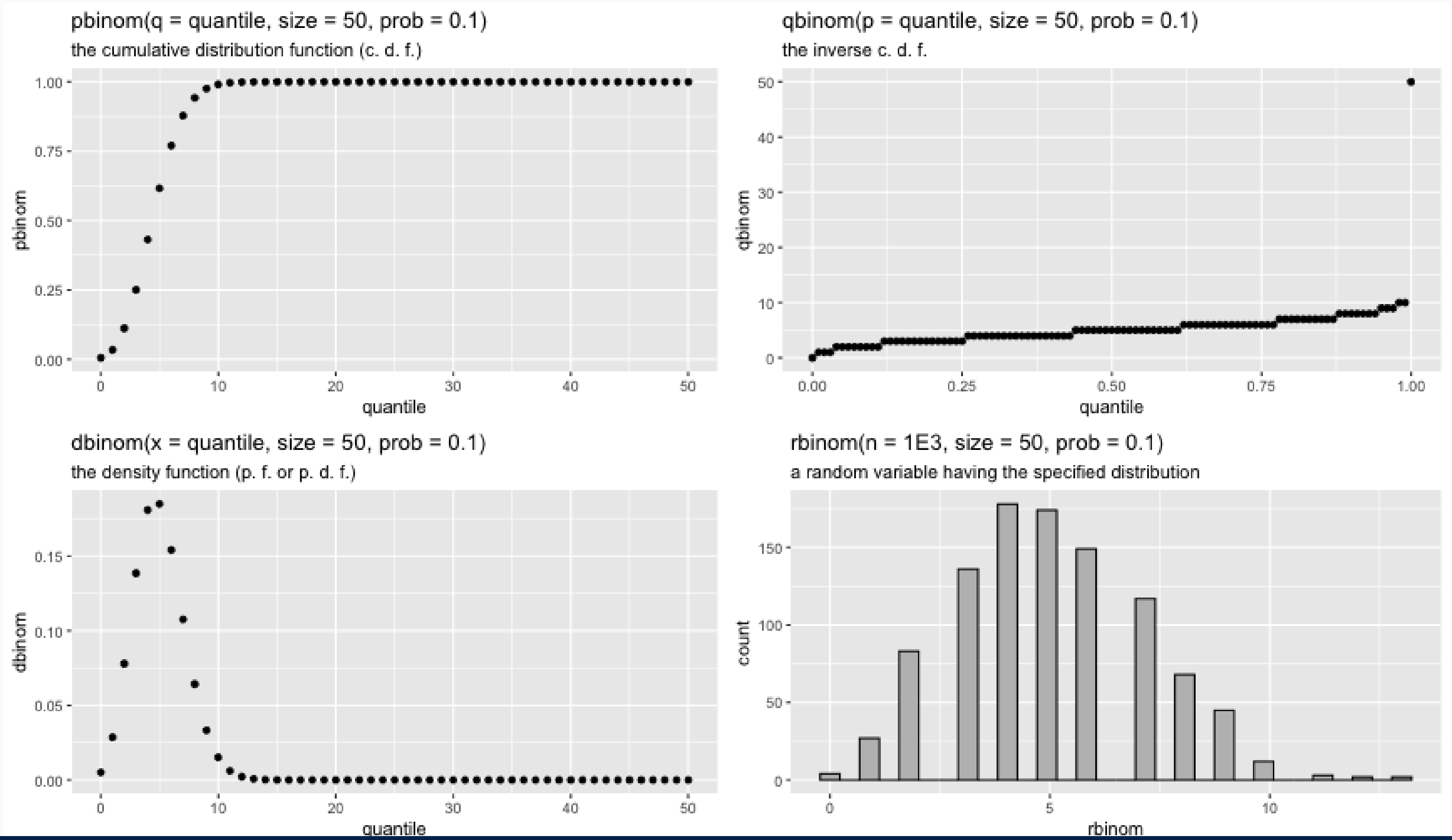
A parameterised normal distribution

Normal distribution parameterised with mean 50 and standard deviation 100.



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%.



Functions for Statistical Testing

`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`

The five steps of hypothesis testing

General principles of hypothesis testing

1. Decide on the effect that you are interested in, **design** a suitable experiment or study, pick a data summary function and test statistic.

2. Set up a **null hypothesis**, which is a simple, computationally tractable model of reality that lets you compute the null distribution, i.e., the possible outcomes of the test statistic and their probabilities under the assumption that the null hypothesis is true.

3. Decide on the **rejection region**, i.e., a subset of possible outcomes whose total probability is small.

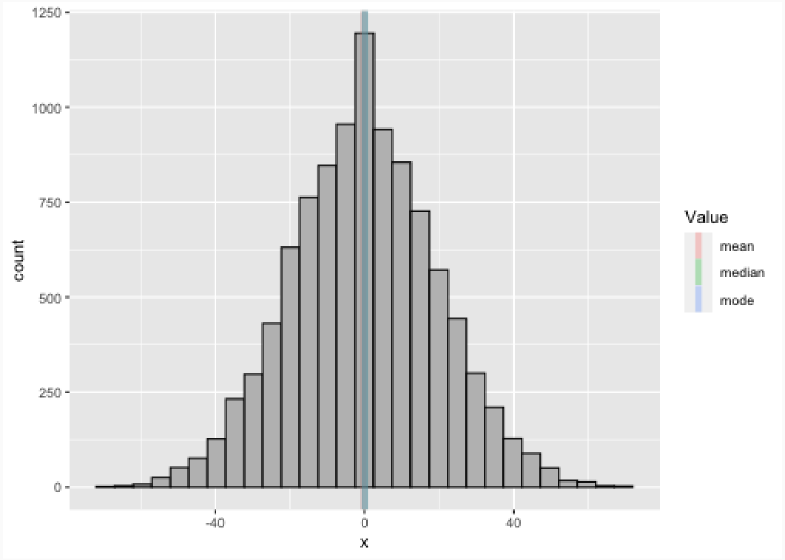
4. Do the experiment and collect the data; compute the **test statistic**.

5. Make a **decision**: reject the null hypothesis if the test statistic is in the rejection region.

Knowledge assumptions - Central tendency

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
- the mode[1]

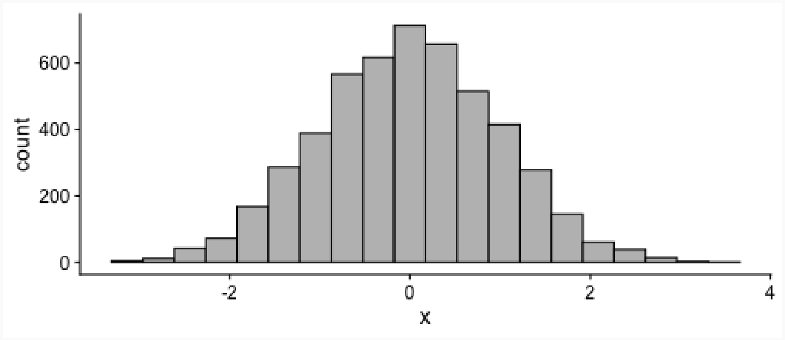
1. R does not have a standard in-built function to calculate mode. Instead, `mode()` allows users to get or set the type or *storage mode* of an object.

Knowledge assumptions - Normality

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

Normal distribution

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

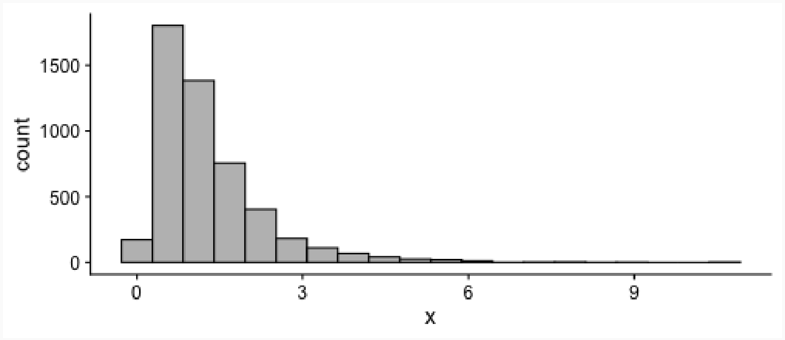


```r
shapiro.test(x)
```

```
##
##      Shapiro-Wilk normality test
##
## data:   x
## W = 0.99951, p-value = 0.2324
```

Log-normal distribution

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



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

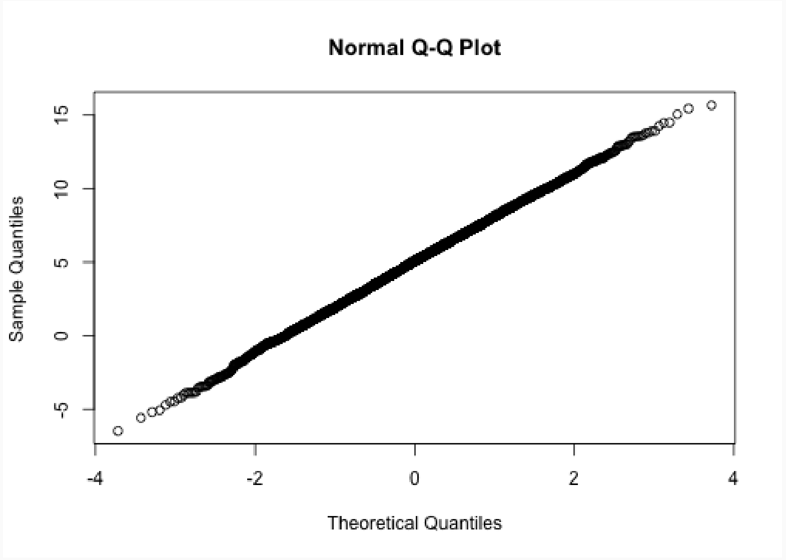
```
##
##      Shapiro-Wilk normality test
##
## data:   x
## W = 0.79049, p-value < 2.2e-16
```

Knowledge assumptions - Normality

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

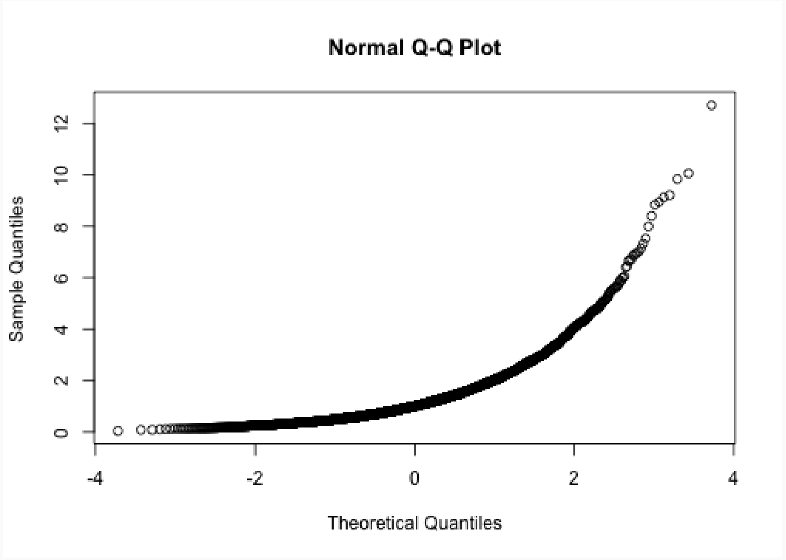
Normal distribution

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



Log-normal distribution

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



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

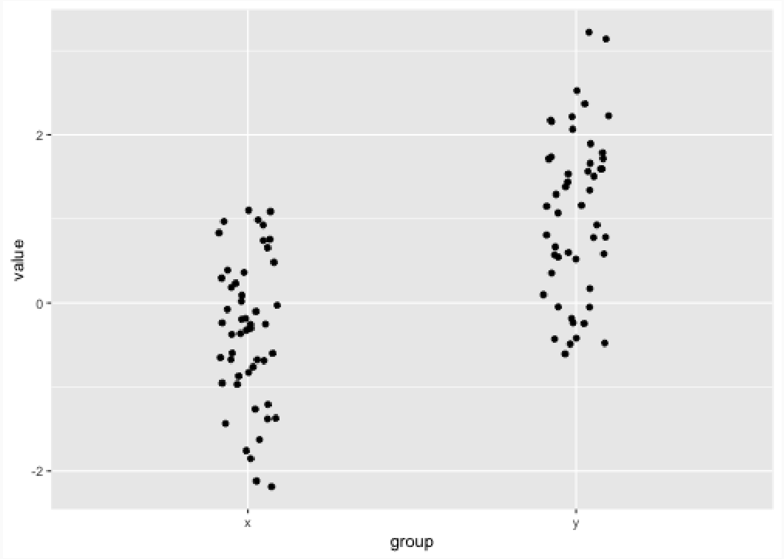
Non-parametric tests make fewer assumptions, as such:

- 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.

Parametric t-test

Two normal distributions

```r
set.seed(10)
x <- rnorm(n = 50, mean = 0, sd = 1)
y <- rnorm(n = 50, mean = 1, sd = 1)
```



Unpaired t-test

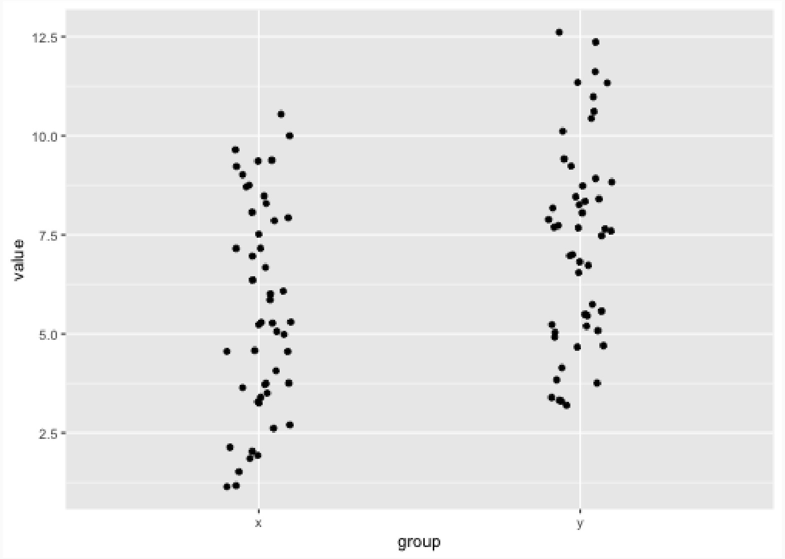
```r
t.test(value ~ group, test_data)
```

```
##
##     Welch Two Sample t-test
##
## data:  value by group
## t = -7.6342, df = 96.629, p-value = 1.623e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.775946 -1.043040
## sample estimates:
## mean in group x mean in group y
##      -0.3412954       1.0681976
```

Non-parametric wilcoxon test

Two uniform distributions

```
set.seed(10)
x <- runif(n = 50, min = 1, max = 11)
y <- runif(n = 50, 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 = 826, p-value = 0.003506
## alternative hypothesis: true location shift is not equal to 0
```

Directed hypothesis

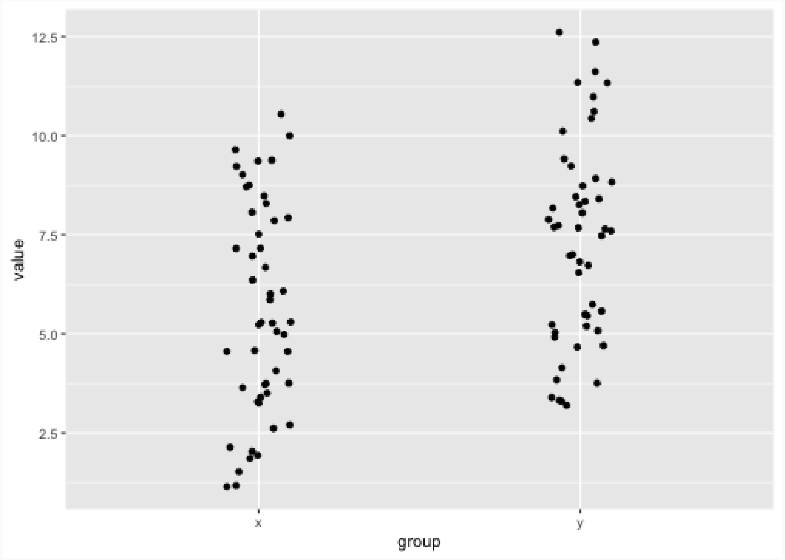
```
wilcox.test(value ~ group, test_data, alternative = "less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  value by group
## W = 826, p-value = 0.001753
## alternative hypothesis: true location shift is less than 0
```

Non-parametric wilcoxon test

Two uniform distributions

```
set.seed(10)
x <- runif(n = 50, min = 1, max = 11)
y <- runif(n = 50, min = 3, max = 13)
```



Parametric (unpaired) t-test

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

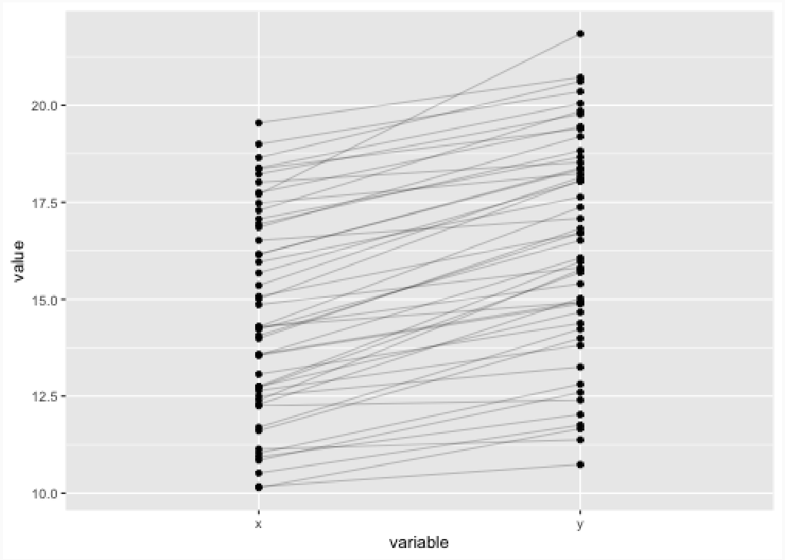
```
##
##     Welch Two Sample t-test
##
## data:  value by group
## t = -3.3315, df = 97.863, p-value = 0.00122
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.7676677 -0.7012921
## sample estimates:
## mean in group x mean in group y
##        5.586011        7.320490
```

Warning: Beware of interpreting inadequate tests!

Paired test

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

```r
set.seed(10)
n_sample <- 50
x <- runif(n = n_sample, min = 10, max = 20)
y <- x + 2 + rnorm(n = n_sample, mean = 0, sd = 1)
```



```r
t.test(value ~ variable, test_data, paired = TRUE)
```

```
##
##      Paired t-test
##
## data:  value by variable
## t = -14.238, df = 49, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal t
## 95 percent confidence interval:
##  -2.058353 -1.549172
## sample estimates:
## mean of the differences
##               -1.803762
```

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

Multiple-testing correction

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

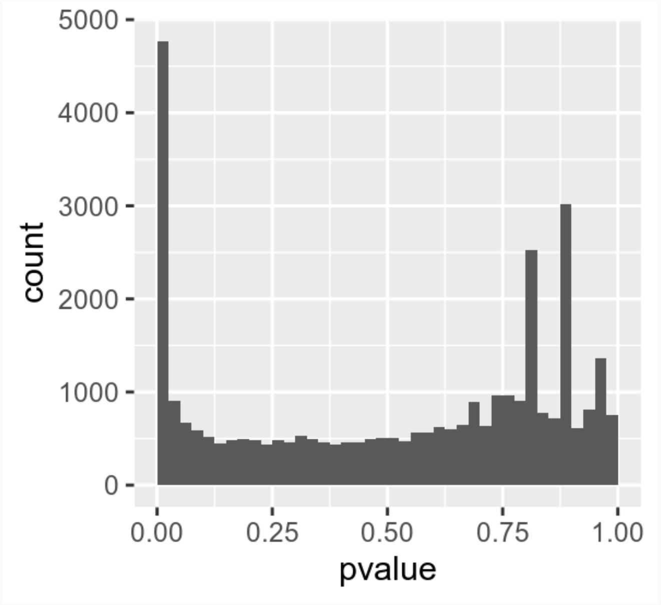
> Green jelly beans linked to acne! 95% confidence! Only 5% chance of coincidence!

https://xkcd.com/882/

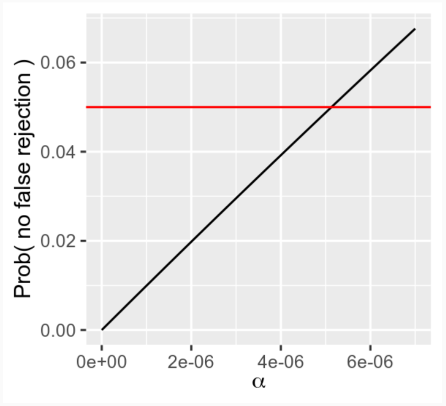
Multiple-testing correction

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

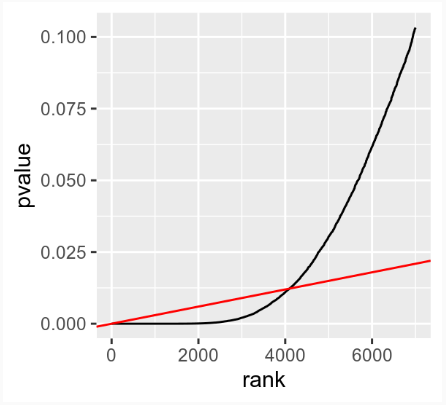
- True positive
- True negative
- False positive (type I error)
- False negative (type I error)



Bonferroni correction

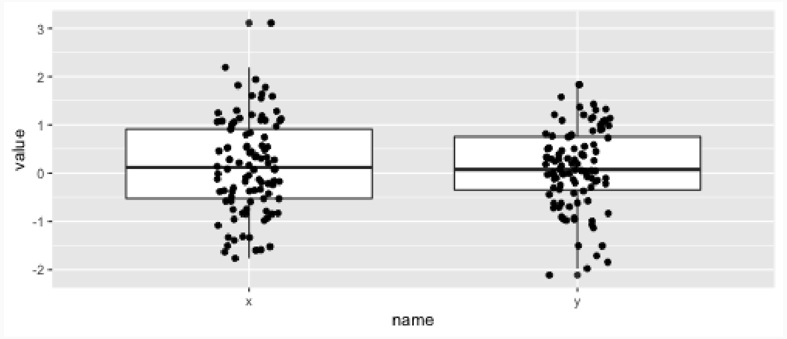


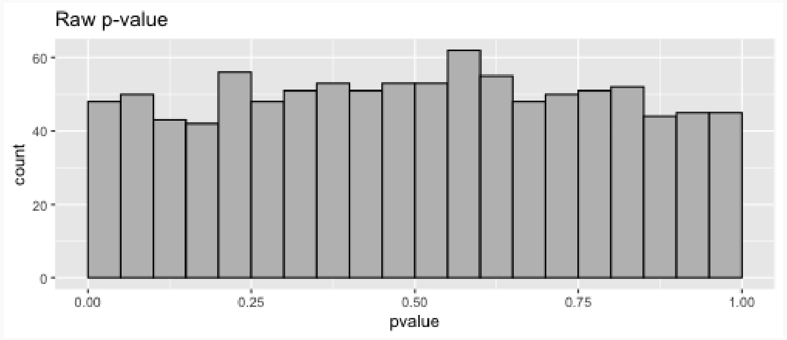
Benjamini-Hochberg procedure



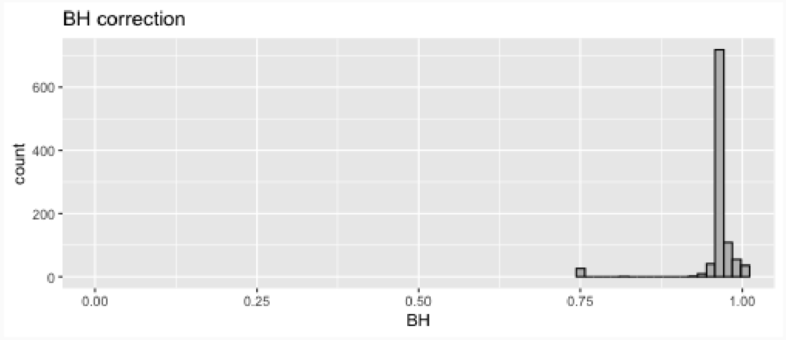
Multiple-testing correction

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

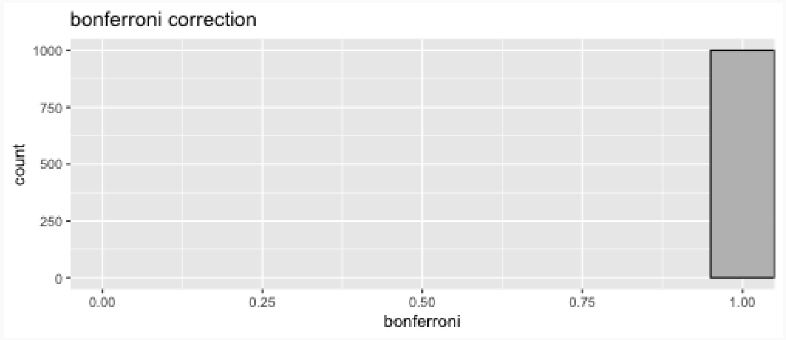




There are 48 raw p-values smaller than 0.05



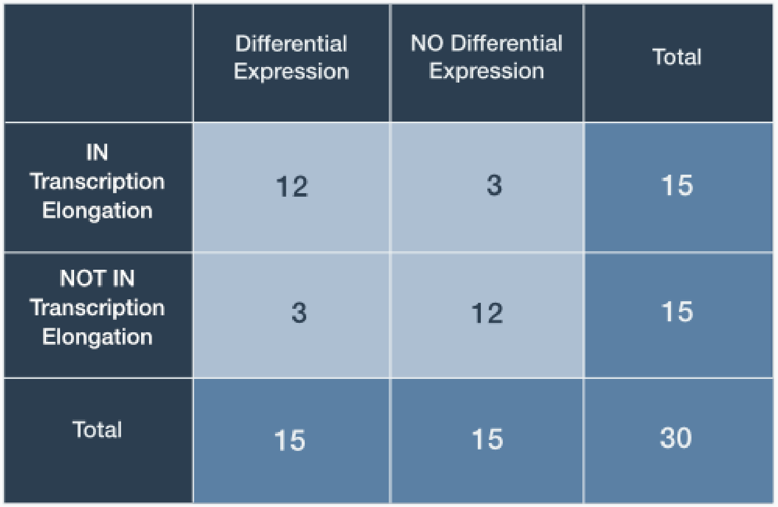
There are 0 BH-corrected p-values smaller than 0.05



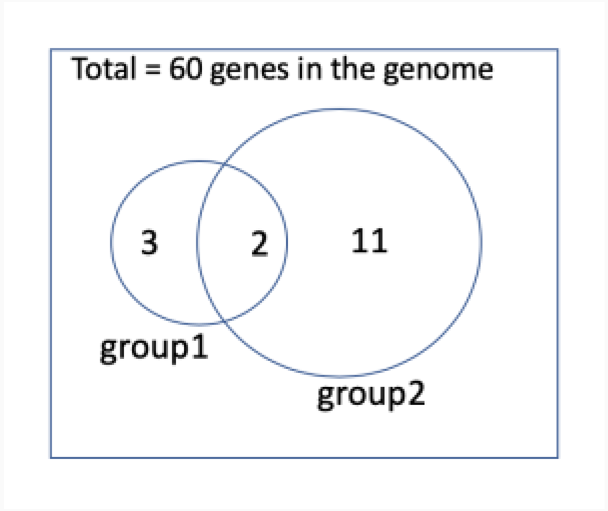
There are 0 bonferonni corrected p-values smaller than 0.05

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)



	Differential Expression	NO Differential Expression	Total
IN Transcription Elongation	12	3	15
NOT IN Transcription Elongation	3	12	15
Total	15	15	30



Further reading: Towards data science

Fisher's Exact Test

Men	Women	Row.total
1	9	10
11	3	14
12	12	24

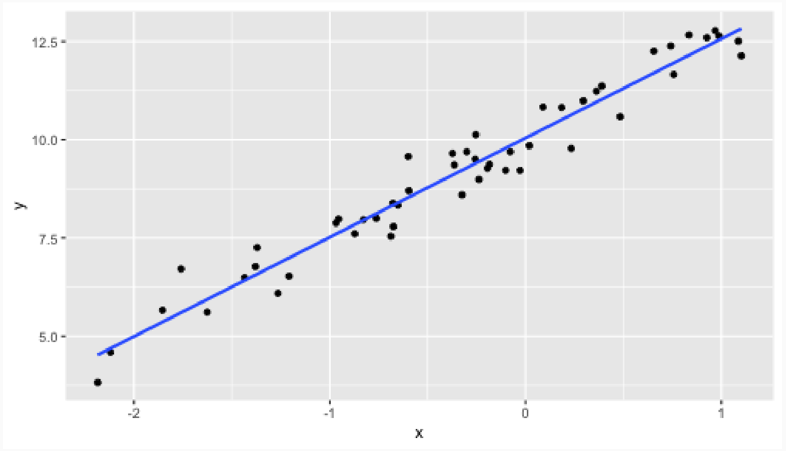
Knowing that 10 of these 24 teenagers are studying, and that 12 of the 24 are female, and assuming the null hypothesis that men and women are equally likely to study, what is the probability that these 10 teenagers who are studying would be so unevenly distributed between the women and the men?

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

$$p = \binom{10}{1}\binom{14}{11}\Big/\binom{24}{12} = \frac{10!\;14!\;12!\;12!}{1!\;9!\;11!\;3!\;24!} \approx 0.001346076$$

Linear models

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



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

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

```
##
## Call:
## lm(formula = y ~ x, data = test_data)
##
## Coefficients:
## (Intercept)            x
##      10.043        2.525
```

Linear models - Summary

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

```
##
## Call:
## lm(formula = y ~ x, data = test_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85187 -0.31598  0.06065  0.30831  1.11183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.04268    0.07502  133.87   <2e-16 ***
## x            2.52516    0.08126   31.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4929 on 48 degrees of freedom
## Multiple R-squared:  0.9526,    Adjusted R-squared:  0.9517
## F-statistic: 965.5 on 1 and 48 DF,  p-value: < 2.2e-16
```

Exercises: the normal distribution

- Generate a vector of 1000 normally distributed values with mean 10 and standard deviation 5.
- Print summary statistics about those values.
- Verify the mean and standard deviation. Inspect the deciles of those values.
- Visualise the distribution of those values. Draw vertical lines to indicate the mean and 1 standard deviation either side. Bonus point if the lines are colored.
    - Using base R.
    - Using `ggplot`.
- Verify that approximately 64% and 95% of the values are within 1 and 2 standard deviations of the mean, respectively.
- Generate a new vector with *a lot* more values (e.g., one million). Draw again a histogram. Does the distribution look better? worse?

Exercises: probabilities

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

- Plot the cumulative distribution function in the range $[-5, 5]$.
- Plot the density function in the range $[-5, 5]$.
- 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?

Exercises: statistical testing

- In the `iris` dataset, visualise the distribution of sepal length stratified by species.
- Print summary statistics for each column in the dataset.
  - How many species are there in the dataset? What are their names? How many observations do we have for each species?
- Is the sepal length normally distributed overall? Within each species?
- Is there a significant variation between the sepal length between the different species?
- Do `setosa` and `versicolor` species have significantly different sepal length?

Exercises: multiple testing

Use the `ALL` microarray gene expression dataset in the *ALL* package. The normalized expression data is stored in `exprs(ALL)`. Sample metadata is stored in `pData(ALL)`

Use the following code to select samples from B-cell lymphomas harboring the BCR/ABL translocation and from lymphomas with no observed cytogenetic abnormalities (NEG).

```r
bcell = grep("^B", as.character(ALL$BT))
moltyp = which(as.character(ALL$mol.biol) %in% c("NEG", "BCR/ABL"))
ALL_bcrneg = ALL[, intersect(bcell, moltyp)]
ALL_bcrneg$mol.biol = factor(ALL_bcrneg$mol.biol)
```

- Test each microarray probeset between patients who achieved remission and those who were refractory to treatment.
- Correct p-values for multiple testing. How many probesets remain significant?
- Plot the expression of the most significant probeset in the two groups of samples.
- Bonus point: Does the most significant probeset map to a gene? If so, which one?
- Visualise the distribution of unadjusted p-values for the two probesets with high and low variance. How would you use variance be used to reduce the burden of multiple testing correction?

Exercises: Over-representation analysis (ORA)

Use the following code to fetch the list of Gene Ontology Biological Processes, and associated probeset identifiers.

```r
library(hgu95av2.db)
go_table <- hgu95av2GO2ALLPROBES %>%
  as.data.frame() %>%
  as_tibble() %>%
  filter(Ontology == "BP") %>%
  dplyr::select(probe_id, go_id) %>%
  unique()
go_list <- split(x = go_table$probe_id, f = go_table$go_id)
```

- Identify GO categories over-represented in the set of DE probesets identified in the previous slide.
    - Save computational time: Focus on GO categories with more than 10 probesets.
- What is the top hit? Does it makes sense / match existing literature?

Exercises: linear regression models

- Estimate a simple linear regression model that explains the expression level for probeset "1636_g_at" by the molecular biology of the cancer factor, `mol.biol`. Save the model as `ALL_bcrneg_mod`.
- Print a summary of the coefficients for the linear model.
- Visualise the two variables in a plot. Indicate the intercept of the linear model, and the effect of the BCR/ABL mutation.

Exercises: linear regression models

- Regress the expression level for probeset "1636_g_at" by the molecular biology of the cancer factor, `mol.biol` the age of the patient, `age`, and whether the patient received a bone marrow transplant or not, `transplant`. Save the model as `mod`. Put differently, estimate the model:

$$1636\_g\_at_i = \beta_0 + \beta_1 mol.biol + \beta_2 age + \beta_3 transplant + u_i$$

- What is the effect of each explanatory variable on the gene expression?
- Visualise the relationship between significant explanatory variables and gene expression.
- Can you make a hypothesis about any interaction between explanatory variables that has an effect on gene expression? How would you test such a hypothesis?

Further reading

- 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

Machine learning using the caret package

The *caret* package (short for **C**lassification **A**nd **RE**gression **T**raining) is a set of functions that attempt to streamline the process for creating predictive models. It is the R

The package contains diverse functionality, including tools for:

- data splitting
- pre-processing
- feature selection
- model tuning using resampling
- variable importance estimation

Training models using caret

Partition dataset in training and test sets

```r
set.seed(998)
inTraining <- createDataPartition(iris$Species, p = .75, list = FALSE)
training <- iris[ inTraining,]
testing  <- iris[-inTraining,]
```

Set the training parameters

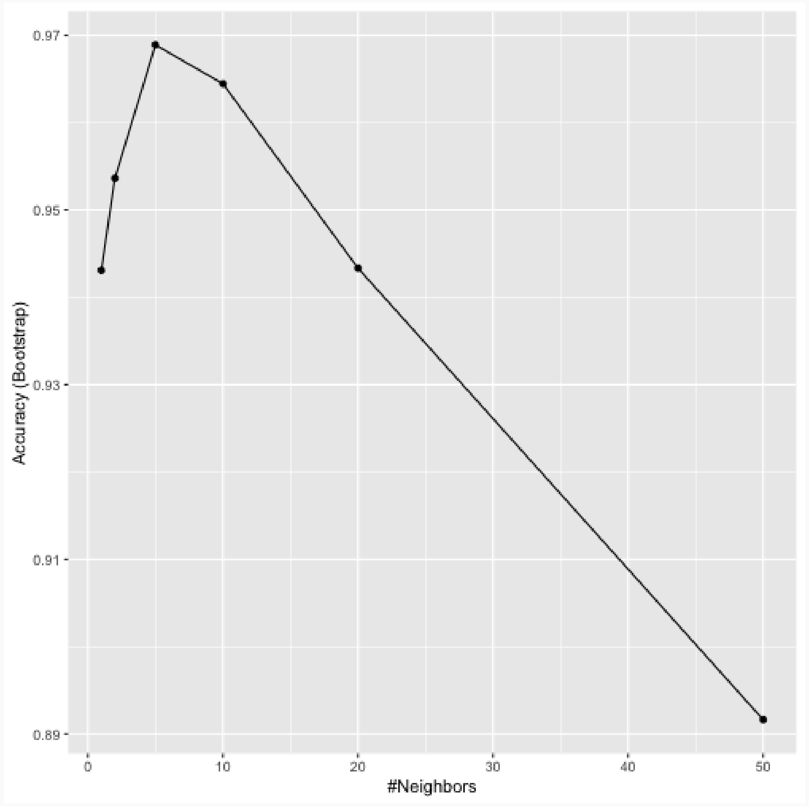
```r
fitControl <- trainControl(
  ## bootstrap
  method = "boot",
  ## repeated ten times
  number = 5)
```

Train the model

```r
knnFit <- train(
  Species ~ ., data = training,
  method = "knn",
  trControl = fitControl,
  tuneGrid = data.frame(
    k = c(1, 2, 5, 10, 20, 50)))
```

Visualising model performance in caret

```r
ggplot(knnFit)
```



Apply a model to make predictions on new data

Make predictions

```
knnPred <- predict(knnFit, newdata = testing)
```

Measure performance

```
confusionMatrix(data = knnPred, testing$Species)$table
```

```
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         12          0         0
##   versicolor      0         12         1
##   virginica       0          0        11
```

```
confusionMatrix(data = knnPred, testing$Species)$overall["Accuracy"]
```

```
##  Accuracy
## 0.9722222
```

Further reading

- The caret Package: https://topepo.github.io/caret/
- Cheatsheet for Scikit-learn (Python) & caret (R) packages, by Kunal Jain
- Machine Learning with Python scikit-learn Vs R Caret, by Fisseha Berhane
- Machine Learning with caret in R DataCamp
- 238 models available in caret: https://topepo.github.io/

Exercise: Machine learning

```r
library(ExperimentHub)
ehub <- ExperimentHub()
logcounts <- ehub[["EH3094"]] # logcounts
col_data <- ehub[["EH3095"]] # colData
```

We aim to predict the `label.main` covariate using gene expression.

1. Subset the dataset to a reasonable number of variable genes.
2. Set up training and testing data subsets.
3. Train a random forest classifier over a grid of parameters, and evaluate it on training and test datasets.
4. Train a k nearest neighbors classifier over a grid of parameters, and compare it to the random forest.
5. Experiment with more classifiers.