Basic statistical tests in R

Einar Andreas Rødland
Anja Bråthen Kristoffersen

UNIVERSITY OF OSLO

Outline

- Example: Tea testing
  - Hypotese testing, type I and type II error
- More tests and how to find and read help files for different tests in R

Famous hypotheses test example:

The Design of Experiments (1935), Sir Ronald A. Fisher

- A tea party in Cambridge in the 1920ties
- A lady claims that she can taste whether milk is poured in cup before or after the tea
- All professors agree: impossible
- Fisher: this is statistically interesting!

  He organized a test

The lady tasting tea

- Test with 8 trials, 2 cups in each trial
  - In each trial: guess which cup had the milk poured in first
- Binomial experiment
  - Independent trials
  - Two possible outcomes, she guesses right cup (success), wrong cup (failure)
  - Constant probability of success in each trial
- X = number of correct guesses in 8 trials, each with probability of success *p*
  - X is Binomially (8,p) distributed $P(X = x) = \binom{8}{x} p^x (1-p)^{(8-x)}$

The lady tasting tea cont.

- The null (conservative) hypothesis
  - The one we initially believe in
- The alternative hypothesis
  - The new claim we wish to test

- She has no special ability to taste the difference

$$p = 0.5$$

- She has a special ability to taste the difference

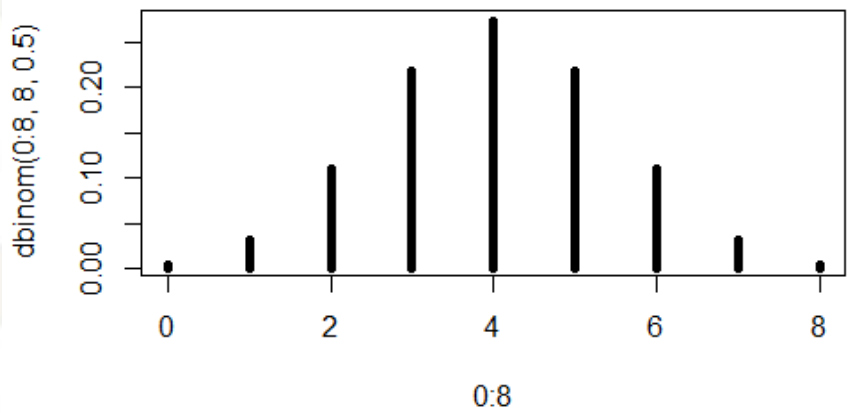
$$p > 0.5$$

How many right to be convinced?

How many correct guesses would we expect by chance?

- If we assume p=0.5 (random guesses):
  - How likely is she to get 6 or more right?
  - How about 7 or 8?

- We need a rule that says how many she would need to convince us.




```
plot(0:8,dbinom(0:8,8,0.5),type='h',lwd=5)
```

How many right to be convinced?

- Rule: We reject H_0 if the observed data have a small probability under H_0 (given H_0 is true).
- Compute the P-value.
  - The probability to obtain the observed value or something more extreme, given that is true
  - **NB!** The p-value is NOT the probability that is true

Small p-value: reject the null hypothesis

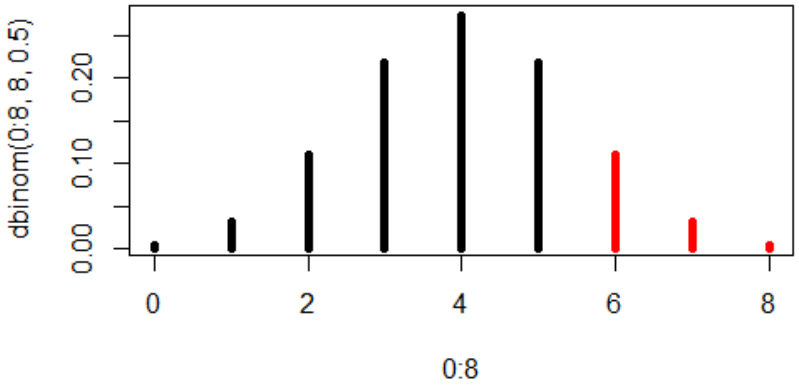
Large p-value: keep the null hypothesis

The lady tasting tea, cont.

- Say: she identified 6 cups correctly
- P-value:

$$P(X \geq 6|H_0\ true)$$
$$= P(X = 6|p = 0.5) + P(X = 7|p = 0.5) + P(X = 8|p = 0.5)$$
$$= dbinom(6, 8, 0.5) + dbinom(7, 8, 0.5) + dbinom(8, 8, 0.5)$$
$$= sum(dbinom(6{:}8, 8, 0.5)) = 1 - pbinom(5, 8, 0.5)$$
$$= 0.1445$$

- Is this enough to be convinced?
- Need a limit.
  - We must know about the types of errors we can make.




```
plot(0:8,dbinom(0:8,8,0.5),type='h',lwd=5,col=c(rep(1,6),rep(2,3)))
```

Two types of error

	H_0 true	H_0 false
Accept H_0	OK / True positive	Type II error / False negative
Reject H_0	Type I error / False positive	OK / True negative

- Type I error most serious
  - Wrongly reject the null hypothesis
  - Example:
    - person is not guilty
    - person is guilty
    - To say a person is guilty when he is not is far more serious than to say he is not guilty when he is.

When to reject

- Decide on the hypothesis' level of significance
  - Choose a level of significance α
  - This guarantees P(type I error) $\leq \alpha$
  - Example
    - Level of significance at 0.05 gives 5 % probability to reject a true
- Reject H_0 if P-value is less than α
  - Finding is **statistically significant**

Important parameters in hypothesis testing

- Null hypothesis
- Alternative hypothesis
- Level of significance

Must be decided upon before we know the results of the experiment

The lady tasting tea, cont.

- Having chosen 5 % level of significance:
  - Say: she identified 6 cups correctly
  - Is this evidence enough?
- P-value
  - The probability to obtain the observed value or something more extreme, given that H_0 is true

$$P(X \geq 6 | H_0\ true) = sum(dbinom(6{:}8, 8, 0.5))$$
$$= 0.1445$$

The lady tasting tea, cont.

- We obtained a P-value of 0.1443
- The rejection rule says
  - Reject H_0 if P-value is less than the level of significance α
  - Since $\alpha = 0.05$ we do NOT H_0 reject

Small p-value: reject the null hypothesis

Large p-value: keep the null hypothesis

The lady tasting tea, cont.

- In the tea party in Cambridge:
  - The lady got every trial correct!
  - This gives P=0.004.
- Comment:
  - Why does it taste different?
    - Pouring hot tea into cold milk makes the milk curdle, but not so pouring cold milk into hot tea*

*http://binomial.csuhayward.edu/applets/appletNullHyp.html Curdle = å skille seg

Area of rejection

- Reject H_0 if p-value $\leq \alpha$
- Reject H_0 if observed value (x) $\geq$ critical value (x_c)
- P(type I error) = P(reject H_0 | H_0 true)
  $= P(X \geq x_c \mid p = 0.5)$
  - $x_c = 7 \rightarrow$ sum(dbinom(7:8, 8, 0.5)) = 0.035 $\leq$ 0.05
  - $x_c = 6 \rightarrow$ sum(dbinom(6:8, 8, 0.5)) = 0.145 > 0.05

  Area of rejection: $\{x: x \geq x_c\} \rightarrow \{x: x \geq 7\}$

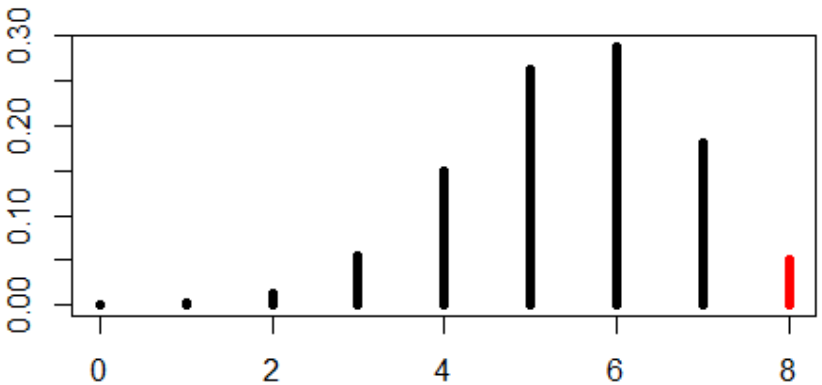
Confidence interval

- How good was the lady at tasting the tea?
  - She got 8 out of 8 trials right!
  - Better than random (p=0.5) at 5% sign. level
  - How low can the true p still be?
- H0: Success likelihood p in each tasting
  - For which values of p can this be rejected?

Confidence interval

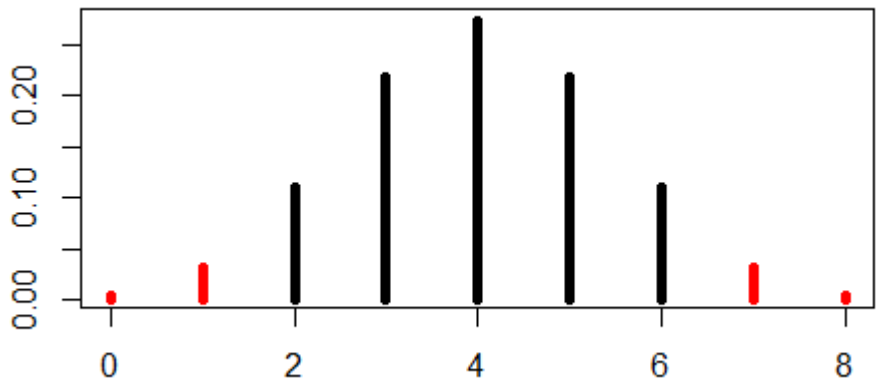
- With 8 successes in 8 trials (α=5%):
  - All hypotheses with p<0.688 can be rejected
  - 95% confidence interval for *p* is 0.688—1

```
binom.test(8,8,alternative="greater")
Exact binomial test data: 8 and 8
number of successes = 8, number of trials = 8, p-value = 0.003906
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
 0.687656 1.000000
sample estimates:
probability of success 1
plot(0:8,dbinom(0:8,8,0.687656),type='h',lwd=5,col=c(rep(1,8),2))
```



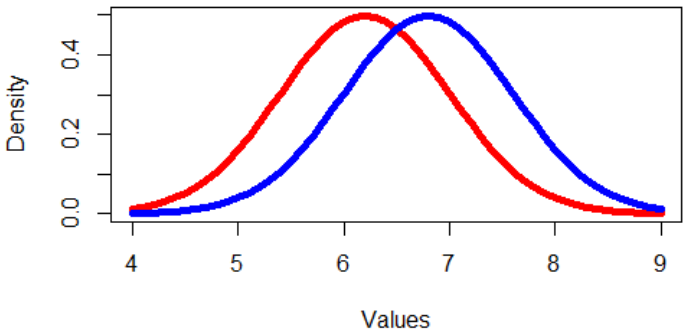
Two-sided test

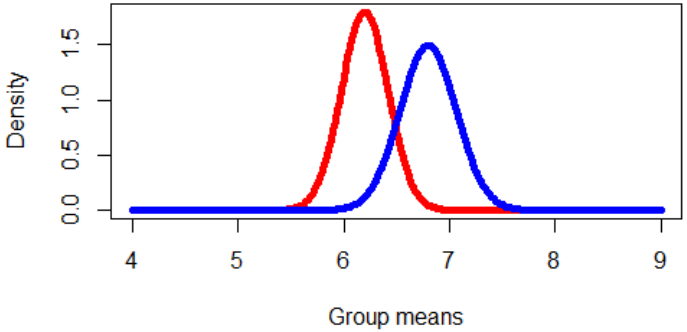
- The tea tasting test was **one-sided**:
  - Alternative hypothesis was $p>0.5$
- The **two-sided** test is more common:
  - Alternative hypothesis: $p\neq0.5$



T-test

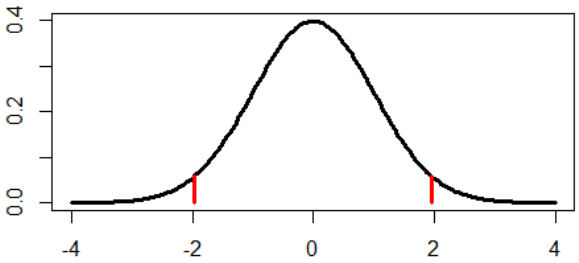
- Compare two groups
  - Assumes normal distribution
- Compare group means:
  - H_0: means are the same
  - H_1: means differ
- Distribution has unknown width (standard deviation)





Sample size: N_{Red}=13, N_{Blue}=9

T-test



- T-statistic for group difference:

$$T = \frac{mean(A) - mean(B)}{estimated\ standard\ error}$$

  – If we assume same standard deviation:

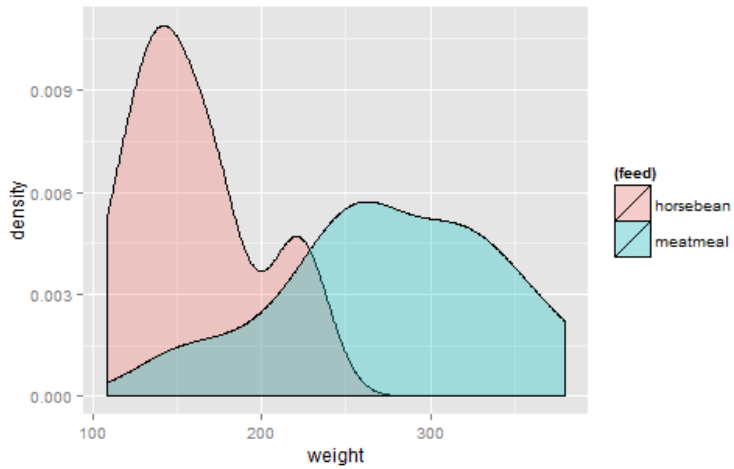
$$T = \frac{mean(A) - mean(B)}{std(A\&B)\sqrt{\frac{1}{N_A}+\frac{1}{N_B}}}$$

  – If we allow different standard deviation:

$$T = \frac{mean(A) - mean(B)}{\sqrt{\frac{var(A)}{N_A}+\frac{var(B)}{N_B}}}$$

T-test

- Statistically significant if $|T|>t_\alpha$:
  - |estimated difference|=|mean(A)-mean(B)|> t_α ·std.err.
- Confidence interval:
  - |estimated difference – true difference|> t_α ·std.err.




```
library(ggplot2)
dt=subset(chickwts,feed %in% c('horsebean','meatmeal'))
ggplot(dt,aes(x=weight,fill=feed))+geom_density(alpha=0.3)
t.test(weight~feed,data=dt)
Welch Two Sample t-test
data: weight by feed
t = -5.0594, df = 16.524, p-value = 0.0001054
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  -165.48454   -67.93364
sample estimates:
mean in group horsebean   mean in group meatmeal
               160.2000                276.9091
```

Type II error / False negative

P(type I error) ≤ α P(type II error) ≤ β

- Want both errors as small as possible
  - especially type I.
- β is not explicitly given, depends on H_1
  - There is one β for each *p>0.5* under H_1

	H_0 true	H_1 true
Accept H_0	OK	Type II error
Accept H_1	Type I error	OK

Example: type II error

- P(type II error) = P(not reject H_0 | H_1 true)
  - $p = 0.7$:

  P(not reject H_0 | $p = 0.7$) = 1 - P(reject H_0 | $p = 0.7$) =

  $1 - P(X \geq 7 \mid p = 0.7) = 1 - (1 - P(X < 7 \mid p = 0.7) =$

  $P(X \leq 6 \mid p = 0.7)$ = sum(dbinom(1:6, 8, 0.7)) = 0.745

$p = 0.7$: a 74.5% chance of accepting H_0

Power of the test

- The probability that a false H_0 is rejected

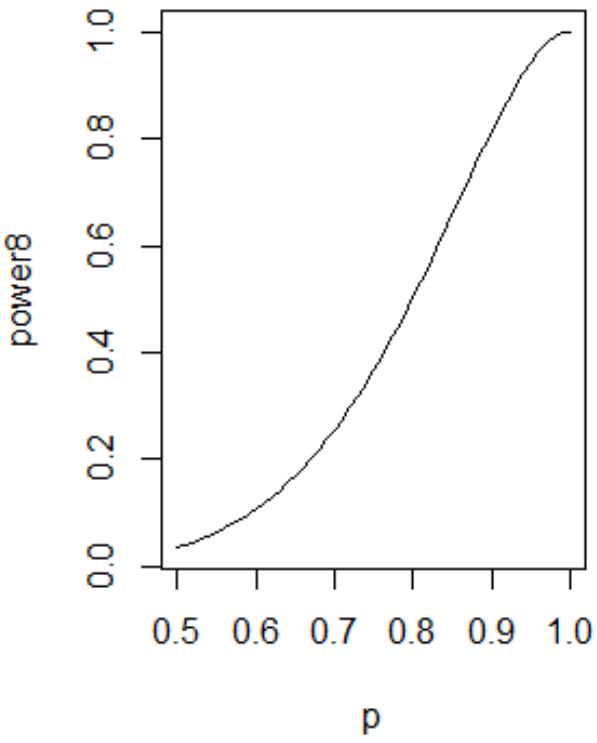
$$P(\text{reject } H_0 \mid H_1 \text{ true}) = 1 - P(\text{accept } H_0 \mid H_1 \text{ true}) = 1 - \beta$$

- A test with large power has:
  - larger probability to draw the right conclusion
  - larger probability to reject a false null hypothesis

  then a test with low power.
- α and β is connected:
  - Decreasing α will give an increased β which again will decrease the power of the test

Example: power function

```
p <- seq(0.5, 1, 0.01)
antall <- length(p)
beta8 <- rep(NA, antall)
for(i in 1:antall){
  beta8[i] <- sum(dbinom(0:6, 8, p[i]))
}
power8 <- 1 - beta8
plot(p, power8, type = "l")
```



http://www.stanford.edu/~stephsus/basic-stats-tests.pdf

Summary of Basic Statistical Tests in R

Type of data	What you want to know…	If data are…	then, in R, do…
1 numerical vector	normal distribution?		`shapiro.test()`, `ks.test()`
	equal probabilities?	counts	`chisq.test()`
	location of mean?	normal	`t.test()`
		non-normal	`wilcox.test()`
2 independent vectors	same distribution?		`ks.test()`, `w jitter`
	same means?	normal	`t.test()`
		non-normal	`wilcox.test()`
	same variances?	normal	`var.test()`
2 paired vectors	same means?	normal	`t.test(paired = T)`
		non-normal	`wilcox.test(paired = T)`
	functional relation?	normal	`lm()`[1]
	correlated?	normal	`cor.test()`
		non-normal	`cor.test(method='spearman')`
1 numerical vector + 1 factor	different group means?	normal, same variances	`lm()`[1], `anova()`[2], `aov()`
		different variances	`kruskal.test()`
2 numerical vectors + 1 factor	different means? interactions?	normal	`lm()`
2 vectors of counts	different proportions?		`chisq.test()`, `fischer.test()`

[1]In linear regression, watch out for outliers and nonlinear covariates.
[2]In anova with factor levels > 2, multiple comparisons inflate chances of a significant result; use Bonferroni correction or Tukey's HSD.

(adapted from Lab Syntax lecture on Baayen ch. 4 by Joan Bresnan, February 2011)

help(binom.test)

binom.test {stats} R Documentation

Exact Binomial Test

Description

Performs an exact test of a simple null hypothesis about the probability of success in a Bernoulli experiment.

Usage

```
binom.test(x, n, p = 0.5,
           alternative = c("two.sided", "less", "greater"),
           conf.level = 0.95)
```

Arguments

`x`	number of successes, or a vector of length 2 giving the numbers of successes and failures, respectively.
`n`	number of trials; ignored if `x` has length 2.
`p`	hypothesized probability of success.
`alternative`	indicates the alternative hypothesis and must be one of `"two.sided"`, `"greater"` or `"less"`. You can specify just the initial letter.
`conf.level`	confidence level for the returned confidence interval.

binom.test()

- > binom.test(6,8,p=0.5) # two-sided test by default

```
Exact binomial test
data: 6 and 8 number of successes = 6, number of trials = 8,
  p-value = 0.2891
alternative hypothesis:
  true probability of success is not equal to 0.5
95 percent confidence interval:
  0.3491442  0.9681460
sample estimates:
probability of success 0.75
```

chisq.test()

```
> M <- as.table(rbind(c(762, 327, 468), c(484,239,477)))
> dimnames(M) <- list(gender=c("M","F"), party=c("Democrat","Independent", "Republican"))
> M
      party
gender Democrat Independent Republican
     M      762         327        468
     F      484         239        477
> (Xsq <- chisq.test(M))  # Prints test summary

        Pearson's Chi-squared test

data:  M
X-squared = 30.0701, df = 2, p-value = 2.954e-07

> names(Xsq)
[1] "statistic" "parameter" "p.value"   "method"    "data.name" "observed"  "expected"  "residuals"
[9] "stdres"
> Xsq$observed   # observed counts (same as M)
      party
gender Democrat Independent Republican
     M      762         327        468
     F      484         239        477
> Xsq$expected   # expected counts under the null
      party
gender Democrat Independent Republican
     M 703.6714    319.6453   533.6834
     F 542.3286    246.3547   411.3166
> |
```

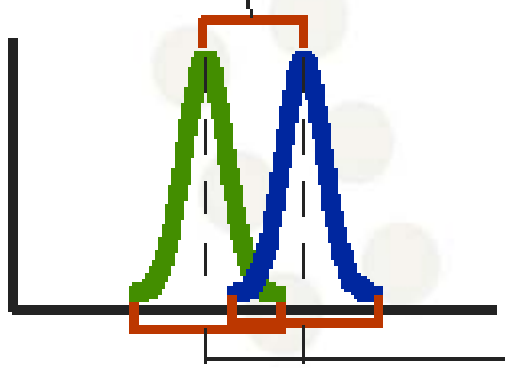
Reject the null hypotheses and assume that there are differences between the groups

t.test()

$$\frac{\text{signal}}{\text{noise}} = \frac{\text{difference between group means}}{\text{variability of groups}}$$

$$= \frac{\bar{X}_T - \bar{X}_C}{SE(\bar{X}_T - \bar{X}_C)}$$

$$= \text{t-value}$$



t.test()

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> y1 <- c(20, 40, 20, 30, 40, 50)
> y2 <- c(30, 45, 25, 40, 50, 60)
> t.test(y1,y2)

        Welch Two Sample t-test

data:  y1 and y2
t = -1.1532, df = 9.959, p-value = 0.2758
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -24.443863   7.777196
sample estimates:
mean of x mean of y
 33.33333  41.66667
```

t.test(, paired = TRUE)

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> y1 <- c(20, 40, 20, 30, 40, 50)
> y2 <- c(30, 45, 25, 40, 50, 60)
> t.test(y1,y2,paired=TRUE)

        Paired t-test

data:  y1 and y2
t = -7.9057, df = 5, p-value = 0.0005211
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -11.042965  -5.623702
sample estimates:
mean of the differences
              -8.333333
```

wilcox.test()

- Same mean? But not normally distributed data.

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> y1 <- c(20, 40, 20, 30, 40, 50)
> y2 <- c(30, 45, 25, 40, 50, 60)
> wilcox.test(y1,y2)

        Wilcoxon rank sum test with continuity correction

data:  y1 and y2
W = 11, p-value = 0.292
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(y1, y2) : cannot compute exact p-value with ties
```

wilcox.test(, paired = TRUE)

RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> y1 <- c(20, 40, 20, 30, 40, 50)
> y2 <- c(30, 45, 25, 40, 50, 60)
> wilcox.test(y1,y2, paired = TRUE)

        Wilcoxon signed rank test with continuity correction

data:  y1 and y2
V = 0, p-value = 0.03054
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(y1, y2, paired = TRUE) :
  cannot compute exact p-value with ties
```

help(cor.test)

Test for Association/Correlation Between Paired Samples

Description

Test for association between paired samples, using one of Pearson's product moment correlation coefficient, Kendall's *tau* or Spearman's *rho*.

Usage

```
cor.test(x, ...)

## Default S3 method:
cor.test(x, y,
         alternative = c("two.sided", "less", "greater"),
         method = c("pearson", "kendall", "spearman"),
         exact = NULL, conf.level = 0.95, continuity = FALSE, ...)

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

Arguments

`x, y`	numeric vectors of data values. `x` and `y` must have the same length.
`alternative`	indicates the alternative hypothesis and must be one of `"two.sided"`, `"greater"` or `"less"`. You can specify just the initial letter. `"greater"` corresponds to positive association, `"less"` to negative association.
`method`	a character string indicating which correlation coefficient is to be used for the test. One of `"pearson"`, `"kendall"`, or `"spearman"`, can be abbreviated.

If `method` is `"kendall"` or `"spearman"`, Kendall's *tau* or Spearman's *rho* statistic is used to estimate a rank-based measure of association. These tests may be used if the data do not necessarily come from a bivariate normal distribution.

For Kendall's test, by default (if `exact` is NULL), an exact p-value is computed if there are less than 50 paired samples containing finite values and there are no ties. Otherwise, the test statistic is the estimate scaled to zero mean and unit variance, and is approximately normally distributed.

For Spearman's test, p-values are computed using algorithm AS 89 for $n < 1290$ and `exact = TRUE`, otherwise via the asymptotic t approximation. Note that these are 'exact' for $n < 10$, and use an Edgeworth series approximation for larger sample sizes (the cutoff has been changed from the original paper).

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the test statistic.
`parameter`	the degrees of freedom of the test statistic in the case that it follows a t distribution.
`p.value`	the p-value of the test.
`estimate`	the estimated measure of association, with name `"cor"`, `"tau"`, or `"rho"` corresponding to the method employed.
`null.value`	the value of the association measure under the null hypothesis, always `0`.
`alternative`	a character string describing the alternative hypothesis.
`method`	a character string indicating how the association was measured.
`data.name`	a character string giving the names of the data.
`conf.int`	a confidence interval for the measure of association. Currently only given for Pearson's product moment correlation coefficient in case of at least 4 complete pairs of observations.

References

D. J. Best & D. E. Roberts (1975), Algorithm AS 89: The Upper Tail Probabilities of Spearman's *rho*. *Applied Statistics*, **24**, 377–379.

Myles Hollander & Douglas A. Wolfe (1973), *Nonparametric Statistical Methods*. New York: John Wiley & Sons. Pages 185–194 (Kendall and Spearman tests).

Arguments

`x, y`	numeric vectors of data values. `x` and `y` must have the same length.
`alternative`	indicates the alternative hypothesis and must be one of `"two.sided"`, `"greater"` or `"less"`. You can specify just the initial letter. `"greater"` corresponds to positive association, `"less"` to negative association.
`method`	a character string indicating which correlation coefficient is to be used for the test. One of `"pearson"`, `"kendall"`, or `"spearman"`, can be abbreviated.
`exact`	a logical indicating whether an exact p-value should be computed. Used for Kendall's *tau* and Spearman's *rho*. See 'Details' for the meaning of `NULL` (the default).
`conf.level`	confidence level for the returned confidence interval. Currently only used for the Pearson product moment correlation coefficient if there are at least 4 complete pairs of observations.
`continuity`	logical: if true, a continuity correction is used for Kendall's *tau* and Spearman's *rho* when not computed exactly.
`formula`	a formula of the form `~ u + v`, where each of `u` and `v` are numeric variables giving the data values for one sample. The samples must be of the same length.
`data`	an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
`subset`	an optional vector specifying a subset of observations to be used.
`na.action`	a function which indicates what should happen when the data contain `NA`s. Defaults to `getOption("na.action")`.
`...`	further arguments to be passed to or from methods.

Details

The three methods each estimate the association between paired samples and compute a test of the value being zero. They use different measures of association, all in the range *[-1, 1]* with *0* indicating no association. These are sometimes referred to as tests of no *correlation*, but that term is often confined to the default method.

If `method` is `"pearson"`, the test statistic is based on Pearson's product moment correlation coefficient `cor(x, y)` and follows a t distribution with `length(x)-2` degrees of freedom if the samples follow independent normal distributions. If there are at least 4 complete pairs of observation, an asymptotic confidence interval is given based on Fisher's Z transform.

See Also

`Kendall` in package **Kendall**.

`pKendall` and `pSpearman` in package **SuppDists**, `spearman.test` in package **pspearman**, which supply different (and often more accurate) approximations.

Examples

```
## Hollander & Wolfe (1973), p. 187f.
## Assessment of tuna quality.  We compare the Hunter L measure of
##  lightness to the averages of consumer panel scores (recoded as
##  integer values from 1 to 6 and averaged over 80 such values) in
##  9 lots of canned tuna.

x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
y <- c( 2.6,  3.1,  2.5,  5.0,  3.6,  4.0,  5.2,  2.8,  3.8)

##  The alternative hypothesis of interest is that the
##  Hunter L value is positively associated with the panel score.

cor.test(x, y, method = "kendall", alternative = "greater")
## => p=0.05972

cor.test(x, y, method = "kendall", alternative = "greater",
         exact = FALSE) # using large sample approximation
## => p=0.04765

## Compare this to
cor.test(x, y, method = "spearm", alternative = "g")
cor.test(x, y,                    alternative = "g")

## Formula interface.
require(graphics)
pairs(USJudgeRatings)
cor.test(~ CONT + INTG, data = USJudgeRatings)
```

```
> x <- c(44.4, 45.9, 41.9, 53.3, 44.7, 44.1, 50.7, 45.2, 60.1)
> y <- c( 2.6,  3.1,  2.5,  5.0,  3.6,  4.0,  5.2,  2.8,  3.8)
>
> cor.test(x, y, method = "spearm", alternative = "g")

        Spearman's rank correlation rho

data:  x and y
S = 48, p-value = 0.0484
alternative hypothesis: true rho is greater than 0
sample estimates:
rho
0.6

> cor.test(x, y,                    alternative = "g")

        Pearson's product-moment correlation

data:  x and y
t = 1.8411, df = 7, p-value = 0.05409
alternative hypothesis: true correlation is greater than 0
95 percent confidence interval:
 -0.02223023  1.00000000
sample estimates:
      cor
0.5711816
```

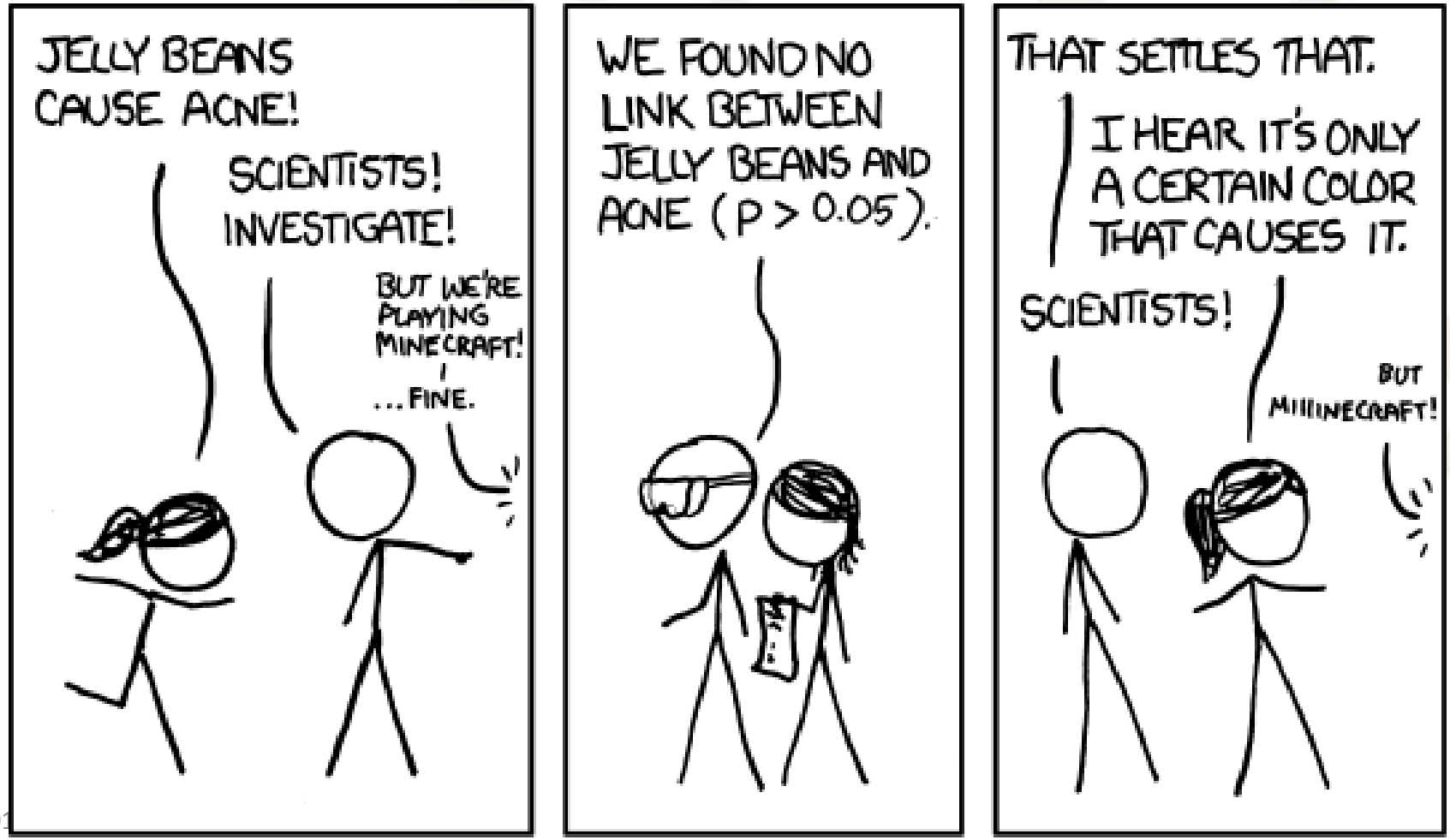
shapiro.test()

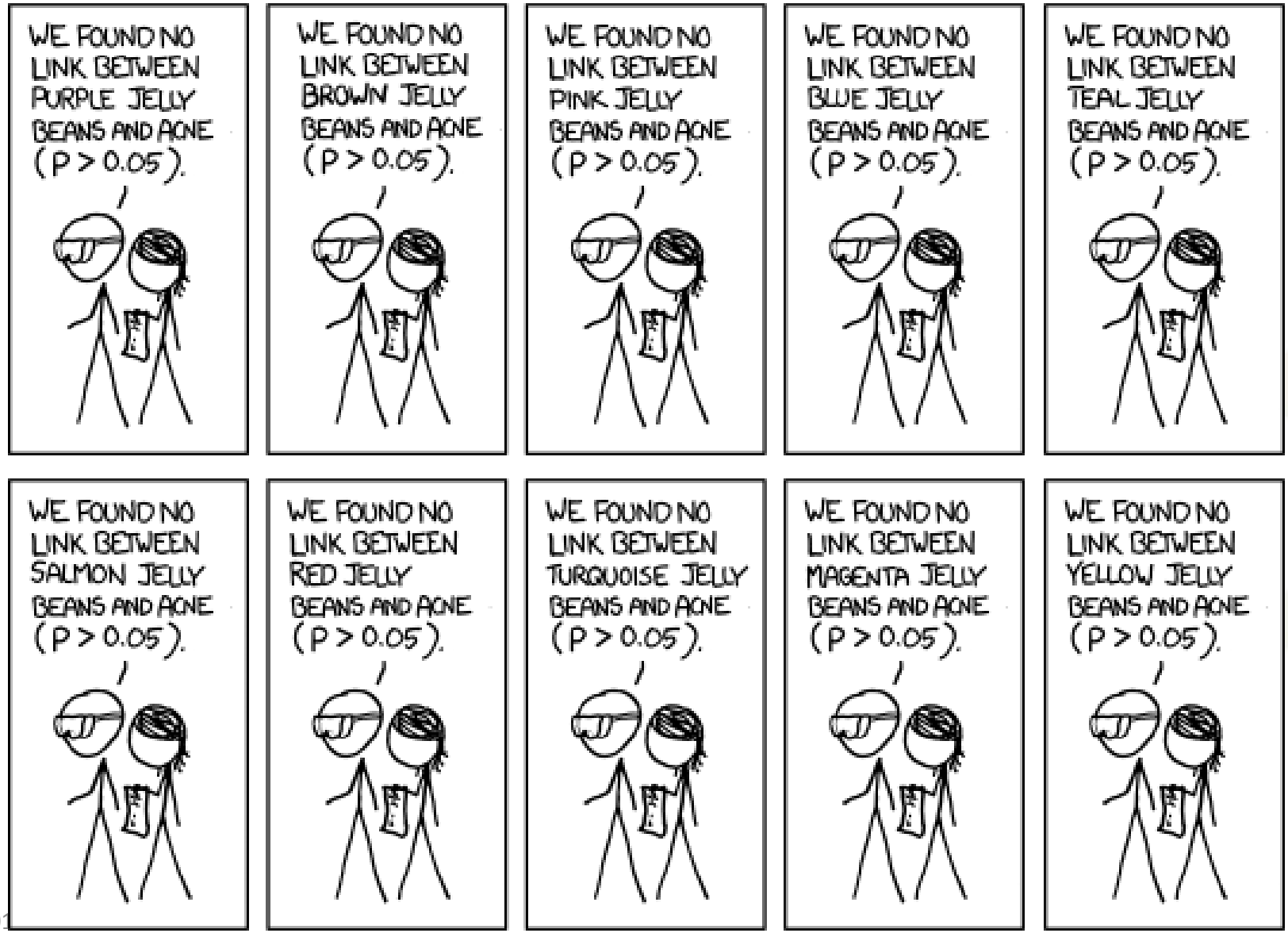
```
> shapiro.test(rnorm(100, mean = 5, sd = 3))

        Shapiro-Wilk normality test

data:  rnorm(100, mean = 5, sd = 3)
W = 0.984, p-value = 0.2676

> shapiro.test(runif(100, min = 2, max = 4))

        Shapiro-Wilk normality test

data:  runif(100, min = 2, max = 4)
W = 0.9349, p-value = 9.553e-05

>
>
```

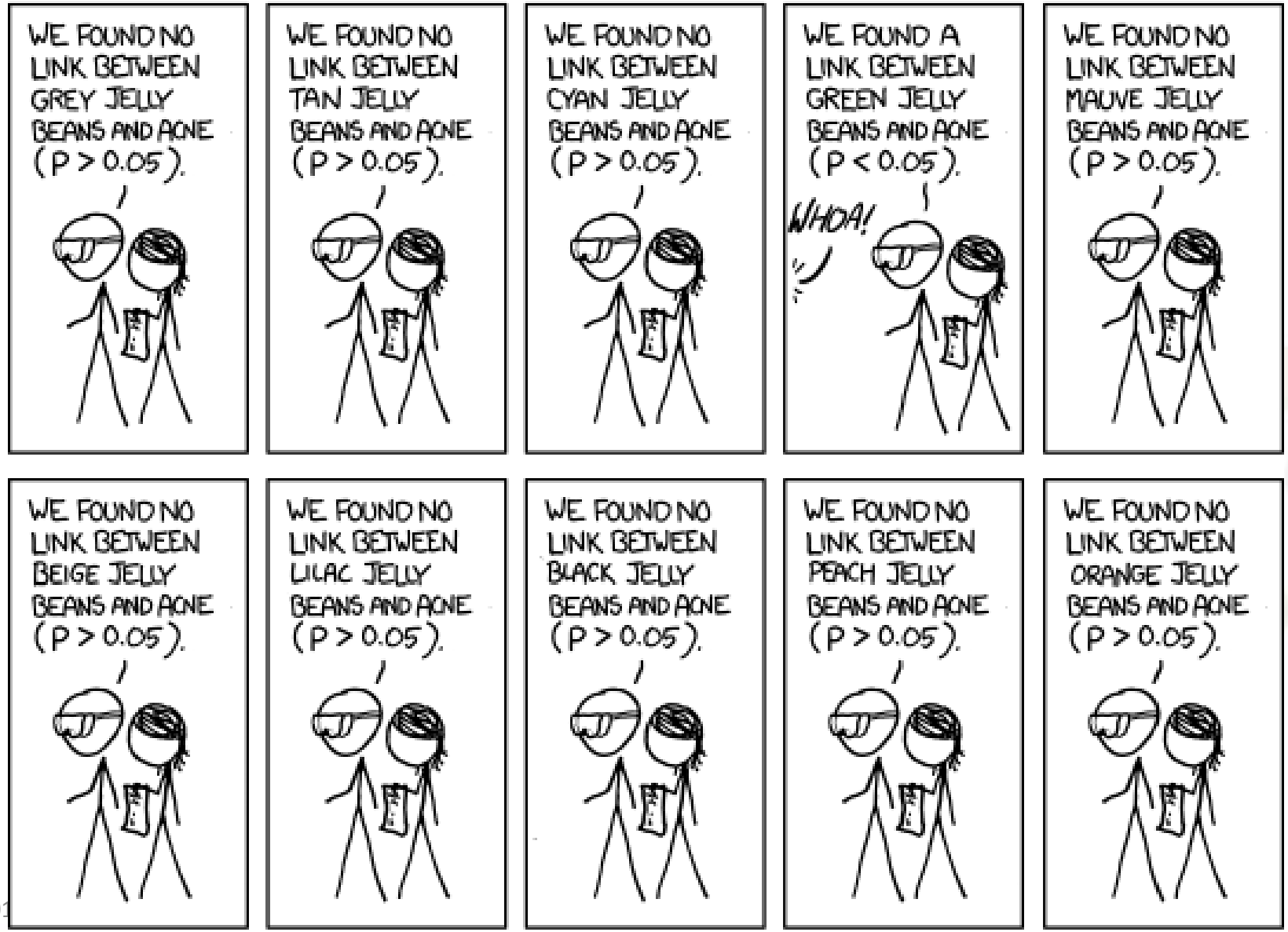
Multiple hypothesis testing

- Tests are designed such that it has an expected proportion of incorrectly rejected null hypotheses, most often this level is 5%.
- When many tests are done the probability of rejecting a null hypotheses falsely increase, hence we can correct the probabilities according to how many tests that are done.

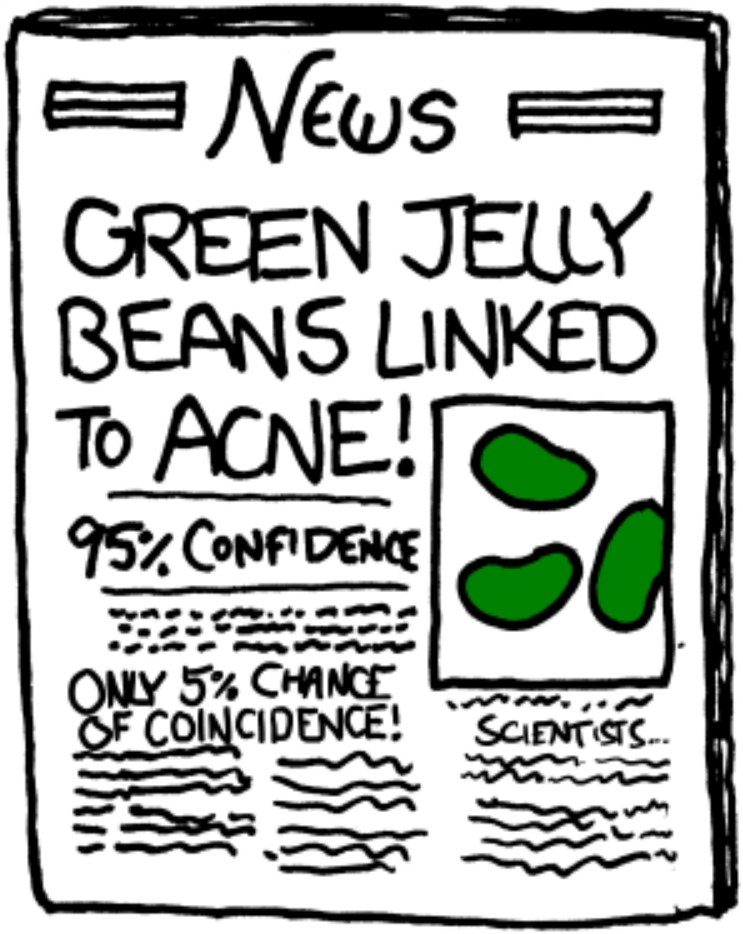
Multiple hypothesis testing







Multiple hypothesis testing



Example: 10000 genes

- Q: is gene g, g = 1, …, 10 000, differentially expressed?
- Gives 10 000 null hypothesis: $H_0^1, H_0^2, \ldots, H_0^{10000}$
  - H_0^1: gene 1 not differentially expressed
  - …
- Assume: no genes differentially expressed
  - H_0^g true for all g
- For *M*=10000 tests at significance level α ≤ 0.01
  - Expected number of false rejections of H_0^g:
    up to α × *M* = 0.01 × 10000 = 100

Need to control the risk of false positive Type I error

- Corrected p-value:
  - The original p-values do not tell the full story.
  - Instead of using the original p-values for decision making, we should use corrected ones.

Different correction methods

- Bonferroni (1935)
  - Just multiply all the p-values by the number of tests
  - Often conservative
    - need very small p-value to reject H_0
    - may result in low power
- Methods that control the family-wise error rate (FWER).
- Methods that control the false discovery rate (FDR).

Family-Wise Error Rate (FWER)

- Control type I errors at a level α
  - Bonferroni
  - Sidak
  - Bonferroni-Holm
  - Westfall & Young
- Use one of these if to limit risk of getting one or more false positive findings

False Discovery Rate (FDR)

- Calculate the expected proportion of type I error among the rejected hypotheses
- Technique that applies to a set of p-values
  - Benjamini & Hochberg
  - Different newer variants of Benjamini & Hochberg
- Use one of these if you are more afraid of missing out on interesting stuff, and accept some portion of false positive findings.

help(p.adjust)

p.adjust {stats} R Docu

Adjust P-values for Multiple Comparisons

Description

Given a set of p-values, returns p-values adjusted using one of several methods.

Usage

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

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

Arguments

`p`	numeric vector of p-values (possibly with `NA`s). Any other `R` is coerced by `as.numeric`.
`method`	correction method
`n`	number of comparisons, must be at least `length(p)`; only set this (to non-default) when you know what you are doing!

False discovery rate (fdr)

```
> x <- rnorm(50, mean=c(rep(0,25),rep(3,25)))
> round(x, 3)
 [1]  1.431 -0.302 -0.607 -2.164  0.381  0.227 -1.140  1.658 -0.720 -1.298
[11]  1.587 -0.245  1.722  1.135 -0.806 -1.838 -1.332 -0.830 -2.034  0.800
[21] -0.301  0.692  0.391  0.050  0.892  2.067  3.490  2.833  3.567  2.861
[31]  1.138  3.056 -0.044  3.471  2.617  3.064  2.270  3.828  3.004  2.929
[41]  3.877  2.243  2.726  2.783  3.683  3.883  2.945  2.156  5.059  3.996
> p <- 2*pnorm( sort(-abs(x)))
> round(p, 3)
 [1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.002 0.002 0.003
[13] 0.003 0.003 0.004 0.005 0.005 0.006 0.009 0.023 0.025 0.030 0.031 0.039
[25] 0.042 0.066 0.085 0.097 0.113 0.153 0.183 0.194 0.254 0.255 0.256 0.372
[37] 0.406 0.420 0.424 0.471 0.489 0.544 0.696 0.703 0.763 0.763 0.806 0.821
[49] 0.960 0.965
> round(p.adjust(p,"fdr"), 3)
 [1] 0.000 0.001 0.001 0.001 0.001 0.002 0.003 0.003 0.003 0.010 0.010 0.011
[13] 0.012 0.012 0.014 0.014 0.016 0.018 0.023 0.058 0.059 0.068 0.068 0.081
[25] 0.084 0.127 0.158 0.174 0.194 0.254 0.295 0.304 0.366 0.366 0.366 0.517
[37] 0.543 0.543 0.543 0.589 0.596 0.647 0.799 0.799 0.830 0.830 0.855 0.855
[49] 0.965 0.965
```