R for Statistics and Graphics

Session 4

R for Inferential Statistics (Hypothesis Testing)

II. Correlation, t-test, ANOVA, Regression

Mehmet Tevfik DORAK, MD PhD

School of Life Sciences, Pharmacy & Chemistry

Kingston University London

Istanbul University, Capa Faculty of Medicine

19 April 2019

Kingston University London

Outline

Correlation

t-test and ANOVA

Regression analysis

Inferential Statistics with R

Correlation Analysis in Base R

`cor(x,y)` for two vectors

```
data(iris)
cor(iris$Sepal.Length, iris$Sepal.Width)
                                    # By default, Pearson correlation
cor(iris$Sepal.Length, iris$Sepal.Width, method = ("pearson"))
                                    # Specifically, Pearson correlation
cor(iris$Sepal.Length, iris$Sepal.Width, method = ("spearman"))
                                    # Specifically, Spearman correlation
cor(iris$Sepal.Length, iris$Sepal.Width, method = ("kendall"))
                                    # Specifically, Kendall correlation

# cor(x,y, method = ("xxxx"))
    # This function provides only the correlation coefficients (r, rho or tau)
      and not the accompanying P values
```

Kingston University London

Inferential Statistics with R

Correlation Analysis in Base R

`cor.test(x,y)` for two vectors

```
data(iris)
cor.test(iris$Sepal.Length, iris$Sepal.Width)
cor.test(iris$Sepal.Length, iris$Sepal.Width, method=("pearson"))
cor.test(iris$Sepal.Length, iris$Sepal.Width, method=("spearman"))
cor.test(iris$Sepal.Length, iris$Sepal.Width, method=("kendall"))

# cor.test(x,y, method = ("xxxx")
```

This function provides the correlation coefficients (r, rho or tau) and the accompanying *P* values

Inferential Statistics with R

Correlation Analysis with Additional R Packages

Compute correlation matrix in R

We have already mentioned the `cor()` function, at the intoductory part of this document dealing with the correlation test for a bivariate case. It be used to compute a correlation matrix. A simplified format of the function is :

```
cor(x, method = c("pearson", "kendall", "spearman"))
```

Here:

- `x` is numeric matrix or a data frame.
- `method` : indicates the correlation coefficient to be computed. The default is “pearson”" correlation coefficient which measures the linear dependence between two variables. As already explained “kendall” and “spearman” correlation methods are non-parametric rank-based correlation tests.

If your data contain missing values, the following R code can be used to handle missing values by case-wise deletion:

```
cor(x, method = "pearson", use = "complete.obs")
```

Inferential Statistics with R

Correlation Analysis in Base R

`cor(x)` for a matrix/dataframe

```
data(iris)
cor(iris[1:4], method = "spearman")      # OR: cor(iris[-5])

Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1667777    0.8818981   0.8342888
Sepal.Width    -0.1667777   1.0000000   -0.3096351  -0.2890317
Petal.Length    0.8818981  -0.3096351    1.0000000   0.9376668
Petal.Width     0.8342888  -0.2890317    0.9376668   1.0000000
```

Good but, how about statistical significance?

Kingston University London

Inferential Statistics with R

Correlation Analysis with Additional R Packages

`Hmisc::rcorr(as.matrix(x))` for a matrix/dataframe

```
install.packages("Hmisc")
library("Hmisc")
data(iris)
correlations  <- rcorr(as.matrix(iris[1:4]))
correlations
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length         1.00       -0.12         0.87        0.82
Sepal.Width         -0.12        1.00        -0.43       -0.37
Petal.Length         0.87       -0.43         1.00        0.96
Petal.Width          0.82       -0.37         0.96        1.00

P
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length              0.1519      0.0000       0.0000
Sepal.Width  0.1519                   0.0000       0.0000
Petal.Length 0.0000       0.0000                   0.0000
Petal.Width  0.0000       0.0000      0.0000
```

Kingston University London

Inferential Statistics with R

Correlation Analysis with Additional R Packages

If your data contain missing values, the following R code can be used to handle missing values by case-wise deletion:

```
cor(x, method = "pearson", use = "complete.obs")
```

Options to deal with missing values:

```
cor(x,y, use="pairwise")
                # to leave out the pairs with missing values
cor(x,y, use="complete")
                # to leave out the whole variable with any missing value
```

Kingston University London

Correlation tests, correlation matrix, and corresponding visualization methods in R

Igor Hut

Inferential Statistics with R

Correlation Analysis in Base R

R-bloggers
R news and tutorials contributed by (750) R bloggers

d your blog! | Learn R | R jobs | Contact us

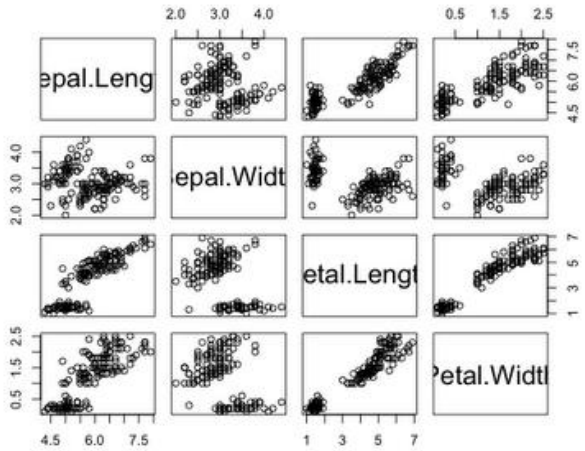
Five ways to visualize your pairwise comparisons

March 5, 2011
By Scott Chamberlain

(This article was first published on Recology, and kindly contributed to R-bloggers)

1) Using base graphics, function "pairs"

```
pairs(iris[1:4], pch = 21)
```



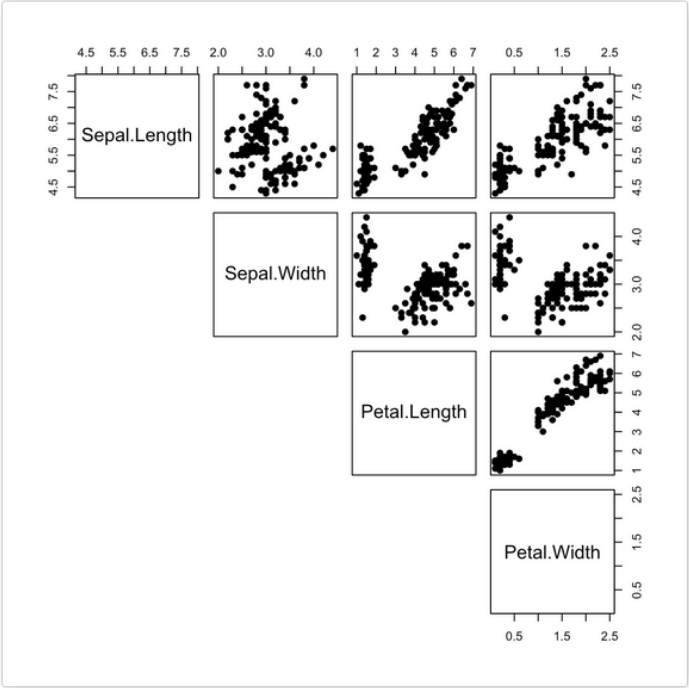
Kingston University London

Inferential Statistics with R

Correlation Analysis in Base R

Now, try this:

```
pairs(iris[1:4], pch = 21, lower.panel = NULL)
```



You may also want to try:

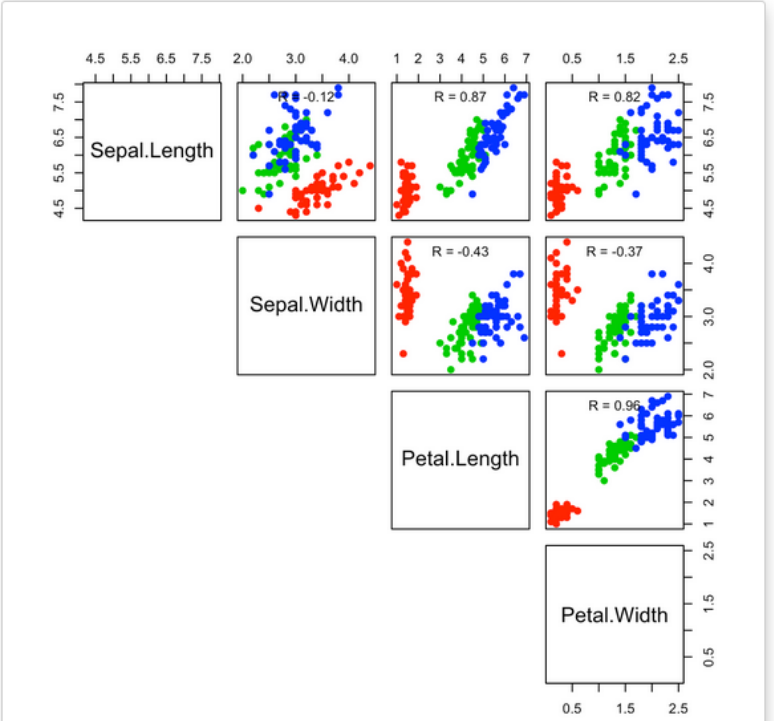
```
pairs(iris[1:4], pch = 21, upper.panel = NULL)
```

Kingston University London

Inferential Statistics with R

Correlation Analysis in Base R

You can also add colour:




```
data(iris)
my_cols <- c("red", "blue", "green")
pairs(iris[,1:4], pch = 19, cex = 0.5, col = my_cols[iris$Species], lower.panel=NULL)
```

SEE:

http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs

Kingston University London

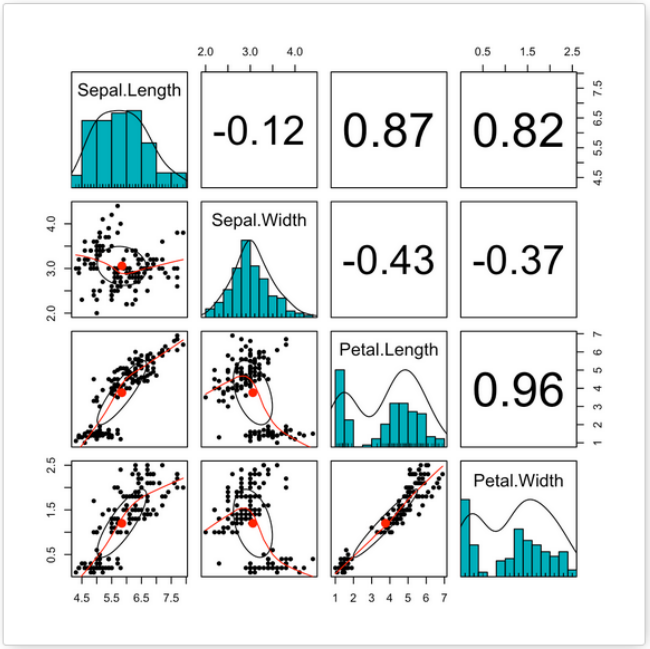
Inferential Statistics with R

Correlation Analysis with Additional R Packages

Use the R package psych

The function **pairs.panels** [in psych package] can be also used to create a scatter plot of matrices, with bivariate scatter plots below the diagonal, histograms on the diagonal, and the Pearson correlation above the diagonal.

```
library(psych)
pairs.panels(iris[,-5],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



psych

by William Revelle

Kingston University London

STHDA

Statistical tools for high-throughput data analysis

Inferential Statistics with R

Correlation Analysis with Additional R Packages

RGui (64-bit)

File History Resize Windows

R Console

```
> data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> install.packages("psych")
--- Please select a CRAN mirror for use in this session ---
also installing the dependency 'mnormt'

trying URL 'https://cran.ma.imperial.ac.uk/bin/windows/contrib/3.5/mnormt_1.5-$
Content type 'application/zip' length 373388 bytes (364 KB)
downloaded 364 KB

trying URL 'https://cran.ma.imperial.ac.uk/bin/windows/contrib/3.5/psych_1.8.4$
Content type 'application/zip' length 5741016 bytes (5.5 MB)
downloaded 5.5 MB

package 'mnormt' successfully unpacked and MD5 sums checked
package 'psych' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\ku63175\AppData\Local\Temp\RtmpaO96FD\downloaded_packages
> library(psych)
> pairs.panels(iris[1:4], method = "spearman")
>
```

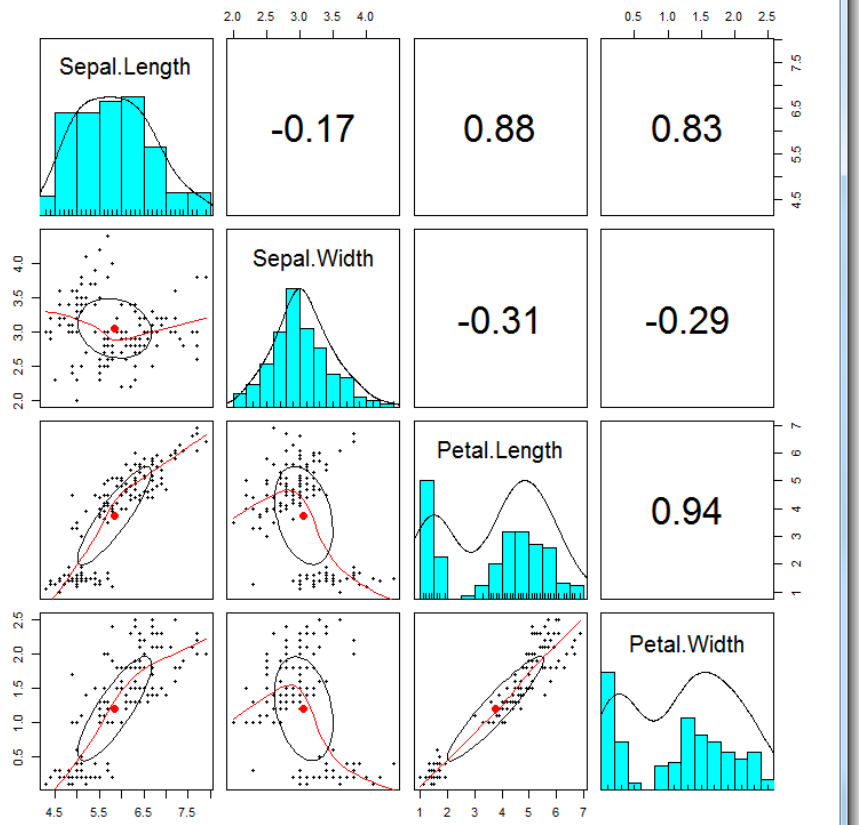
```
data(iris)
library(psych)
pairs.panels(iris[1:4],
      method = "spearman")
```

psych

by William Revelle

R Graphics: Device 2 (ACTIVE)

File History Resize



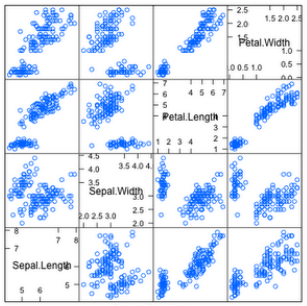
Kingston University London

Inferential Statistics with R

Correlation Analysis with Additional R Packages

2) Using lattice package, function "splom"

```
splom(~iris[1:4])
```



Scatter Plot Matrix

R-bloggers

R news and tutorials contributed by (750) R bloggers

Five ways to visualize your pairwise comparisons

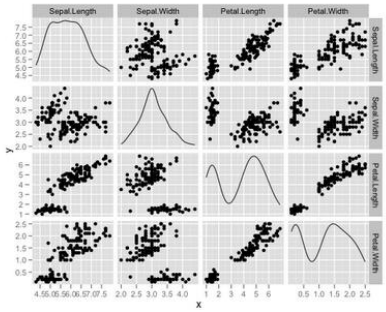
March 5, 2011

By Scott Chamberlain

(This article was first published on Recology, and kindly contributed to R-bloggers)

3) Using package ggplot2, function "plotmatrix"

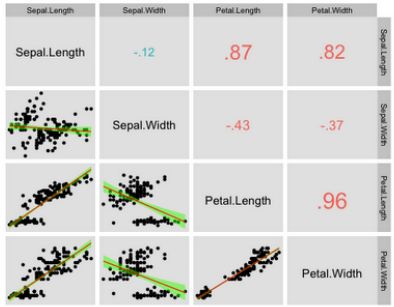
```
plotmatrix(iris[1:4])
```



4) a function called ggcorplot by Mike Lawrence at Dalhousie University

-get ggcorplot function at this link

```
ggcorplot(
  data = iris[1:4],
  var_text_size = 5,
  cor_text_limits = c(5,10))
```



5) panel.cor function using pairs, similar to ggcorplot, but using base graphics.
Not sure who wrote this function, but here is where I found it.

```
panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits=digits)[1]
    txt <- paste(prefix, txt, sep="")
    if(missing(cex.cor)) cex <- 0.8/strwidth(txt)

    test <- cor.test(x,y)
    # borrowed from printCoefmat
    Signif <- symnum(test$p.value, corr = FALSE, na =
FALSE,
                     cutpoints = c(0, 0.001, 0.01, 0.05,
0.1, 1),
                     symbols = c("***", "**", "*", ".", "
"))

    text(0.5, 0.5, txt, cex = cex * r)
    text(.8, .8, Signif, cex=cex, col=2)
}

pairs(iris[1:4], lower.panel=panel.smooth,
upper.panel=panel.cor)
```

Kingston University London

Inferential Statistics with R

Correlation Analysis with Additional R Packages

Package 'corrgram'

Correlograms

Correlograms help us visualize the data in correlation matrices. For details, see Corrgrams: Exploratory displays for correlation matrices.

In R, correlograms are implimented through the **corrgram(x, order = , panel=, lower.panel=, upper.panel=, text.panel=, diag.panel=)** function in the corrgram package.

Options

x is a data frame with one observation per row.

order=TRUE will cause the variables to be ordered using principal component analysis of the correlation matrix.

panel= refers to the off-diagonal panels. You can use **lower.panel=** and **upper.panel**= to choose different options below and above the main diagonal respectively. **text.panel=** and **diag.panel=** refer to the main diagnonal. Allowable parameters are given below.

off diagonal panels
panel.pie (the filled portion of the pie indicates the magnitude of the correlation)
panel.shade (the depth of the shading indicates the magnitude of the correlation)
panel.ellipse (confidence ellipse and smoothed line)
panel.pts (scatterplot)

main diagonal panels
panel.minmax (min and max values of the variable)
panel.txt (variable name).

Kingston University London

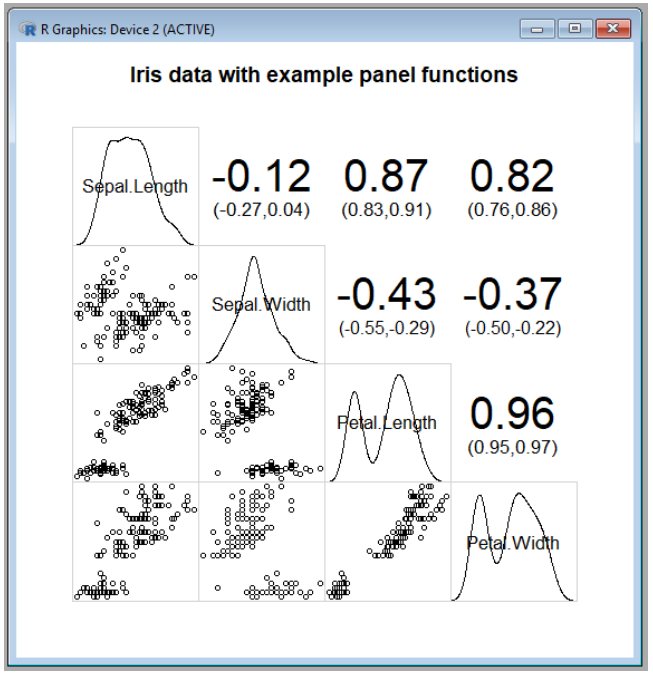
Quick-R powered by DataCamp

Inferential Statistics with R

Correlation Analysis with Additional R Packages

Package 'corrgram'

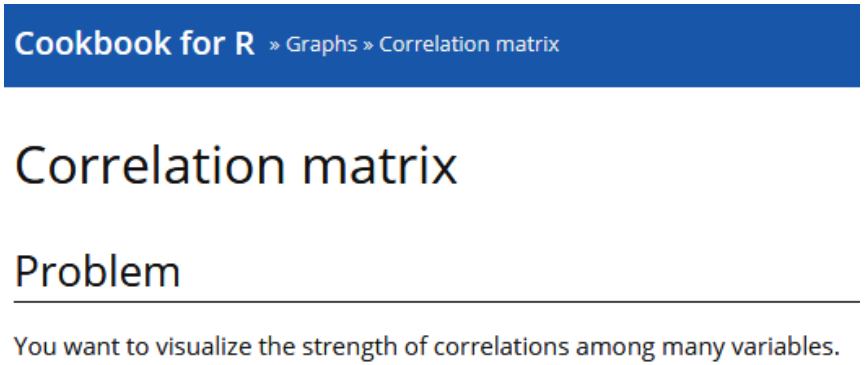
```
library("corrgram")
data(iris)
corrgram(iris,
+          main = "Iris data with example panel functions",
+          lower.panel = panel.pts, upper.panel = panel.conf,
+          diag.panel = panel.density)
```



Kingston University London

Inferential Statistics with R

Correlation




```
install.packages("ellipse")
library("ellipse")
data(iris)
corrtab <- cor(iris[,1:4])
corrtab
round(corrtab, 2)
plotcorr(corrtab, mar = c(0.1, 0.1, 0.1, 0.1))
```

Kingston University London

Inferential Statistics with R

Correlation

Correlation matrix with heatmap using the R package "*DescTools*"

 # DescTools:PlotCorr():

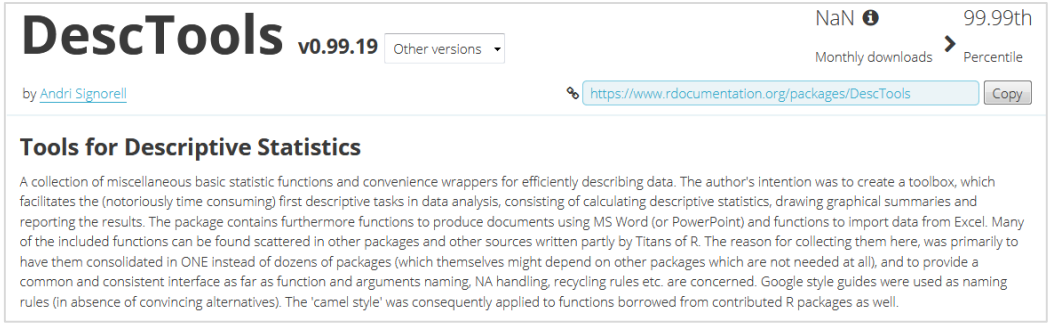
```
a <- cor(iris[-5])
PlotCorr(a)
```

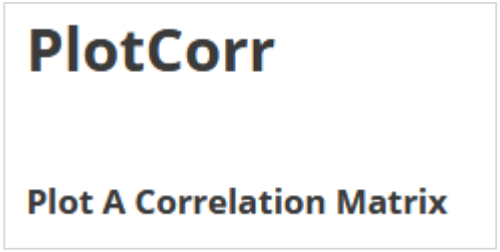
 # generates a correlation matrix as a heatmap (based on correlation coefficients, which can be changed to Spearman etc within `cor()`)

 # Another PlotCorr() example with options:

```
PlotCorr(cor(mtcars), col=Pal("RedWhiteBlue1", 100), border="grey",
args.colorlegend=list(labels=Format(seq(-1,1,.25), digits=2),
frame="grey"))
```

 # SEE: https://www.rdocumentation.org/packages/DescTools/versions/0.99.19/topics/PlotCorr for options of PlotCorr()





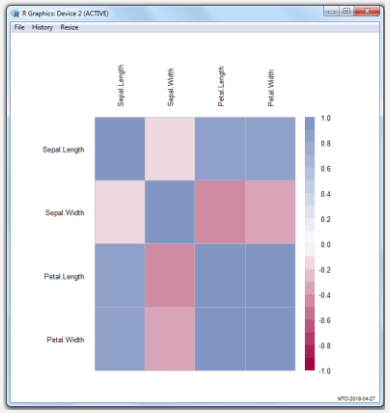
Kingston University London

Inferential Statistics with R

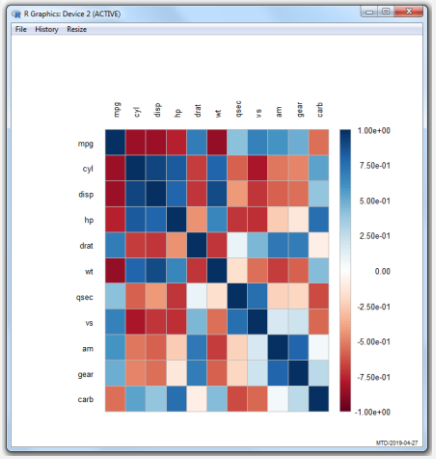
Correlation

Correlation matrix with heatmap using the R package "*DescTools*"

```
a <- cor(iris[-5])
PlotCorr(a)
```




```
PlotCorr(cor(mtcars), col=Pal("RedWhiteBlue1", 100), border="grey",
args.colorlegend=list(labels=Format(seq(-1,1,.25), digits=2), frame="grey"))
```



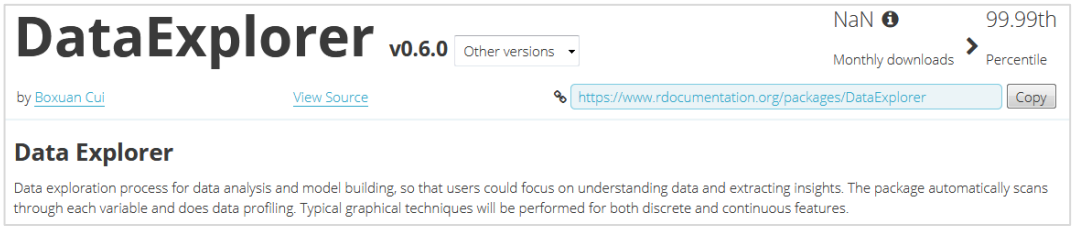
Kingston University London

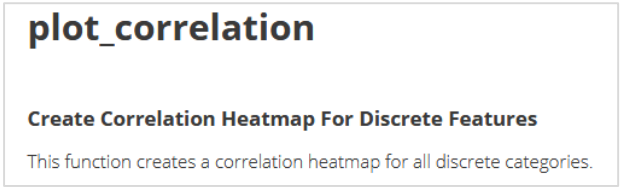
Inferential Statistics with R

Correlation

Correlation matrix with heatmap using the R package "*DataExplorer*"

```
install.packages("DataExplorer")
library("DataExplorer")
data(iris)

plot_correlation(iris)          # All pairwise correlations with heatmap
      # For options, see: https://www.rdocumentation.org/packages/DataExplorer/versions/0.6.0/topics/plot_correlation
```





Kingston University London

Inferential Statistics with R

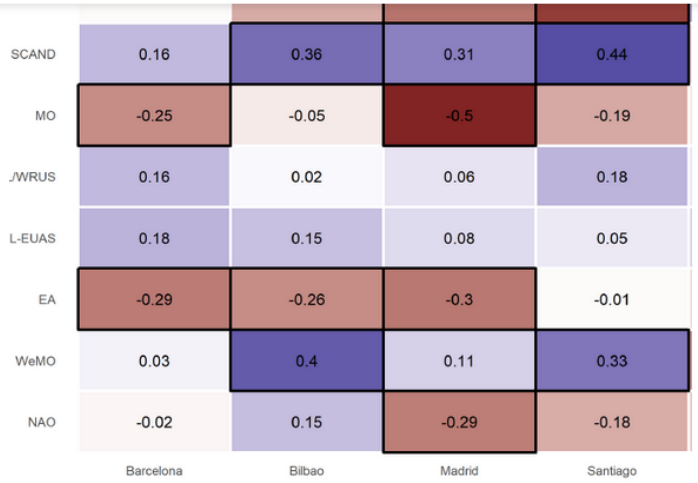
Correlation

Dominic Royé

home blog publications graphs me

Tidy correlation tests in R

2019-04-17 · 0 Comments · statistics, R, R:advanced



When we try to estimate the correlation coefficient between multiple variables, the task is more complicated in order to obtain a simple and tidy result. A simple solution is to use the `tidy()` function from the *{broom}* package. In this post we are going to estimate the correlation coefficients between the annual precipitation of several Spanish cities and climate teleconnections indices: download. The data of the teleconnections are preprocessed, but can be downloaded directly from crudata.uea.ac.uk. The daily precipitation data comes from ECA&D.

Kingston University London

Inferential Statistics with R

t-test

The t-test is used as a test of difference
(between two means)

The means may be from unrelated (unpaired) or related (paired) samples

The t-test is a parametric test, and requires fulfilling of several assumptions

There are alternatives for more than two samples, non-parametric ones and for the violation of variance assumption

Inferential Statistics with R

t-test

Assumptions of the t-test

✓ Data in each comparison group should be distributed normally

✓ For two-sample t-test, the two datasets should be independent

✓ Variances of the two groups being compared should be comparable (up to 3 times difference; in the case of SD, up to 10 times difference)

Inferential Statistics with R

t-test

Typical questions tested by the t-test
(for unpaired/independent samples)

Is there a difference between bone densities of males and females over 50 years?

Is there a difference between the height of trees grown in two different clay types?

Are two different types of diet resulting in a difference in milk production in cows?

Inferential Statistics with R

t-test in R

Variable layout

```
> head(iris, 10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
> tail(iris, 10)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
141          6.7         3.1          5.6         2.4 virginica
142          6.9         3.1          5.1         2.3 virginica
143          5.8         2.7          5.1         1.9 virginica
144          6.8         3.2          5.9         2.3 virginica
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

Sample layout

```
> head(iris, 10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
> tail(iris, 10)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
141          6.7         3.1          5.6         2.4 virginica
142          6.9         3.1          5.1         2.3 virginica
143          5.8         2.7          5.1         1.9 virginica
144          6.8         3.2          5.9         2.3 virginica
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

Inferential Statistics with R

t-test in R

Variable layout
(stacked/vertical)

```
> head(iris, 10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
> tail(iris, 10)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
141          6.7         3.1          5.6         2.4 virginica
142          6.9         3.1          5.1         2.3 virginica
143          5.8         2.7          5.1         1.9 virginica
144          6.8         3.2          5.9         2.3 virginica
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

Sample layout
(horizontal)

```
> head(iris, 10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
> tail(iris, 10)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
141          6.7         3.1          5.6         2.4 virginica
142          6.9         3.1          5.1         2.3 virginica
143          5.8         2.7          5.1         1.9 virginica
144          6.8         3.2          5.9         2.3 virginica
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

Create a subset of `iris` with two species and one variable:

```
iris2 <- subset(iris[, 4:5], iris$Species ==
  "virginica" | iris$Species == "setosa")
```

Create a subset of `iris` with one species and two variables:

```
iris3 <- subset(iris[, 3:4], iris$Species ==
  "setosa")
```

Kingston University London

Biomeasurement 3e
Dawn Hawkins

R HELP SHEET: *t*-test

Inferential Statistics with R

t-test in R

Variable layout

```
> head(iris, 10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
> tail(iris, 10)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
141          6.7         3.1          5.6         2.4 virginica
142          6.9         3.1          5.1         2.3 virginica
143          5.8         2.7          5.1         1.9 virginica
144          6.8         3.2          5.9         2.3 virginica
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

`t.test(iris2$Petal.Width ~ iris2$Species)`

{the grouping variable "Species" must be a factor}

Sample layout

```
> head(iris, 10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
> tail(iris, 10)
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
141          6.7         3.1          5.6         2.4 virginica
142          6.9         3.1          5.1         2.3 virginica
143          5.8         2.7          5.1         1.9 virginica
144          6.8         3.2          5.9         2.3 virginica
145          6.7         3.3          5.7         2.5 virginica
146          6.7         3.0          5.2         2.3 virginica
147          6.3         2.5          5.0         1.9 virginica
148          6.5         3.0          5.2         2.0 virginica
149          6.2         3.4          5.4         2.3 virginica
150          5.9         3.0          5.1         1.8 virginica
```

`t.test(iris3$Petal.Length, iris3$Petal.Width)`

Kingston University London

Biomeasurement 3e
Dawn Hawkins

R HELP SHEET: *t*-test

Inferential Statistics with R

t-test in R

RDocumentation

t.test

Student's T-Test

Performs one and two sample t-tests on vectors of data.

Usage

```
t.test(x, …)

# S3 method for default
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 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.
conf.level	confidence level of the interval.

Kingston University London

Inferential Statistics with R

t-test in R

```
> t.test(female,male,var.equal=TRUE)

        Two Sample t-test

data:  bmd by sex
t = -2.0607, df = 38, p-value = 0.04623
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.129252391 -0.001147609
sample estimates:
mean of x mean of y
  0.81475   0.87995
```

Statistic

Degrees of Freedom

P Value

95% CI of Effect

Difference between these values is EFFECT SIZE

Inferential Statistics with R

t-test in R

Phrase

From DescTools v0.99.19
by Andri Signorell

99.99th Percentile

Phrasing Results Of T-Test

Formulating the results of a comparison of means is quite common. This function assembles a descriptive text about the results of a t-test, describing group sizes, means, p-values and confidence intervals.

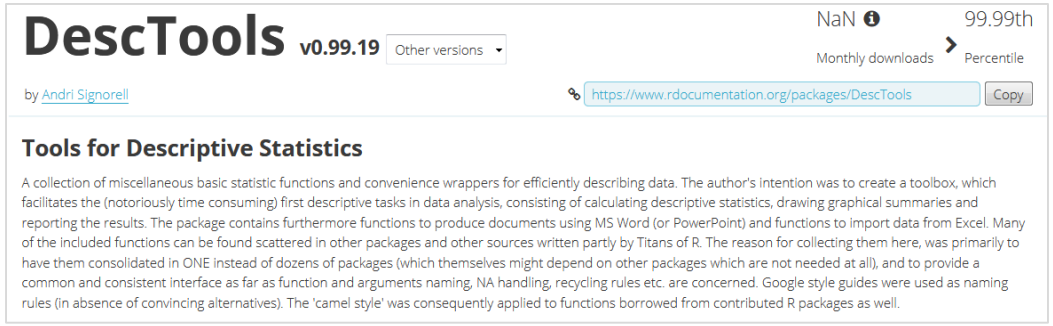
Keywords character

Usage

```
Phrase(x, g, glabels = NULL, xname = NULL, unit = NULL, lang = "engl")
```

Arguments

x	a (non-empty) numeric vector of data values.
g	a vector or factor object giving the group for the corresponding elements of x. The number of levels must equal 2.
glabels	the labels of the two groups, if left to NULL, the levels will be used.
xname	the name of the variable to be used in the text.
unit	an optional unit for be appended to the numeric results.



Kingston University London

Inferential Statistics with R

t-test in R

One-sample t-test:
You have a sample from a population. Given this sample, you want to know if the mean of the population could reasonably be a particular value *m*.

Apply the `t.test` function to the sample *x* with the argument `mu = m`:

```
> t.test(x, mu = m)
```

Demonstration:

```
x <- rnorm(50, mean = 100, sd = 15)
t.test(x, mu = 95)
```

```
    One Sample t-test
    data: x
    t = 3.2832, df = 49, p-value = 0.001897
    alternative hypothesis: true mean is not equal to 95
    95 percent confidence interval:
    97.16167 103.98297
    sample estimates:
    mean of x
    100.5723
```

Kingston University London

Inferential Statistics with R

t-test in R

Paired t-test:

Perform a *t* test by calling the `t.test` function:

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

By default, `t.test` assumes that your data are not paired.

If the observations are paired (i.e., if each *xi* is paired with one *yi*), then specify the argument paired = TRUE:

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

A t-test demonstration:

```
a <- rnorm(25, 30, 5)          # To generate a sample of size 25 from a normal distribution with mean 30 and standard deviation 5
b <- rnorm(25, 25, 7)          # To generate a sample of size 25 from a normal distribution with mean 25 and standard deviation 7
t.test(a, b, paired = TRUE)    # To run the paired t-test

# rnorm arguments: sample size (first); mean (second); SD (third)
```

Kingston University London

Inferential Statistics with R

t-test in R

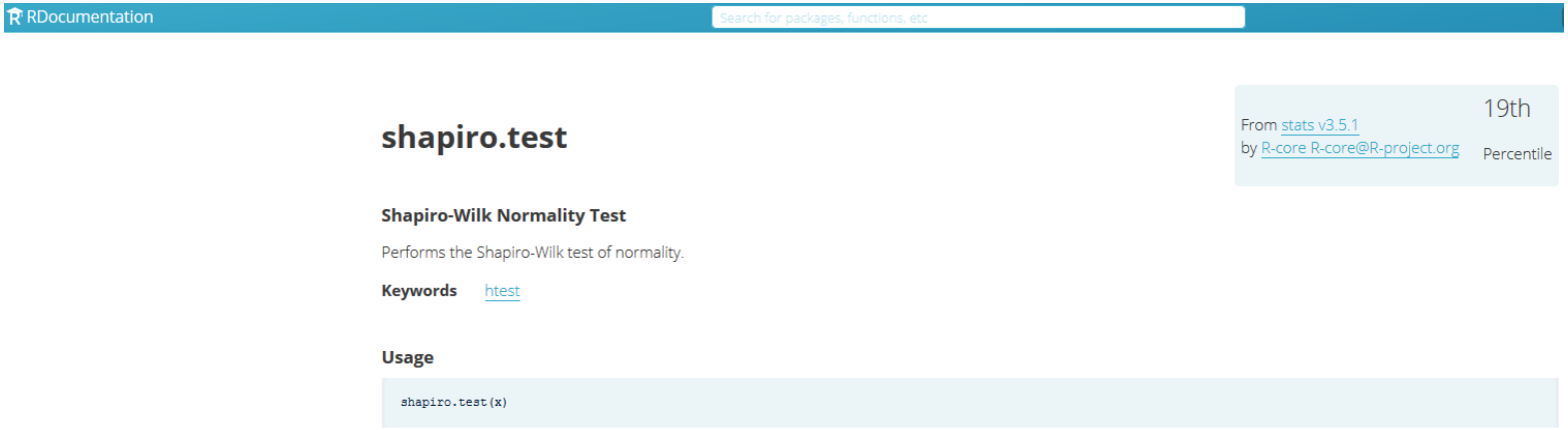
A t-test demonstration:

```
a <- rnorm(25, 30, 5)  # To generate a sample of size 25 from a normal distribution with mean 30 and standard deviation 5
b <- rnorm(25, 25, 7)  # To generate a sample of size 25 from a normal distribution with mean 25 and standard deviation 7
t.test(a,b)            # To run the t-test

# Change the sample size (first argument) in a and b to show the statistical power concept
# Change the SD (third argument) to show the effect of intra-group variability
# Change the difference between means (second argument) to show the effect of difference in the mean (effect size)
```

Inferential Statistics with R

t-test in R

How to check normality and equal variance assumptions




```
a <- rnorm(100, 30, 5)
```
To generate a sample of size 25 from a normal distribution with mean 30 and standard deviation 5

```
shapiro.test(a)
```
To run the Shapiro test on vector a

```
b <- runif(100, min = 20, max = 30)
```
To generate a sample size of 25 random numbers between 20 and 30

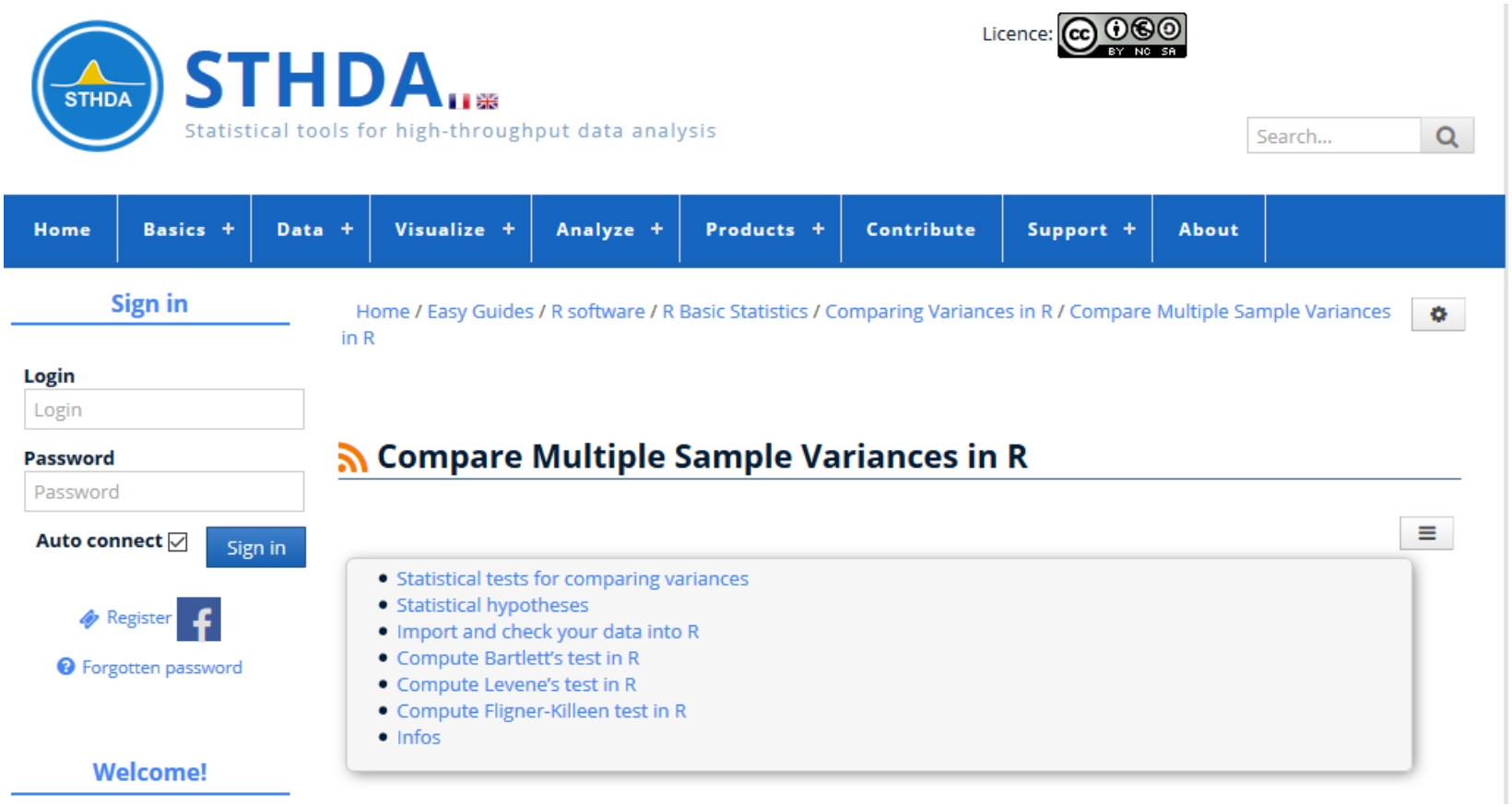
```
shapiro.test(b)
```
To run the Shapiro test on vector b

Kingston University London

Inferential Statistics with R

t-test in R

How to check normality and equal variance assumptions



See also: http://www.cookbook-r.com/Statistical_analysis/Homogeneity_of_variance

Kingston University London

Inferential Statistics with R

t-test in R

Cookbook for R » Statistical analysis » t-test

t-test

Problem

You want to test whether two samples are drawn from populations with different means, or test whether one sample is drawn from a population with a mean different from some theoretical mean.

Solution

Sample data

We will use the built-in `sleep` data set.

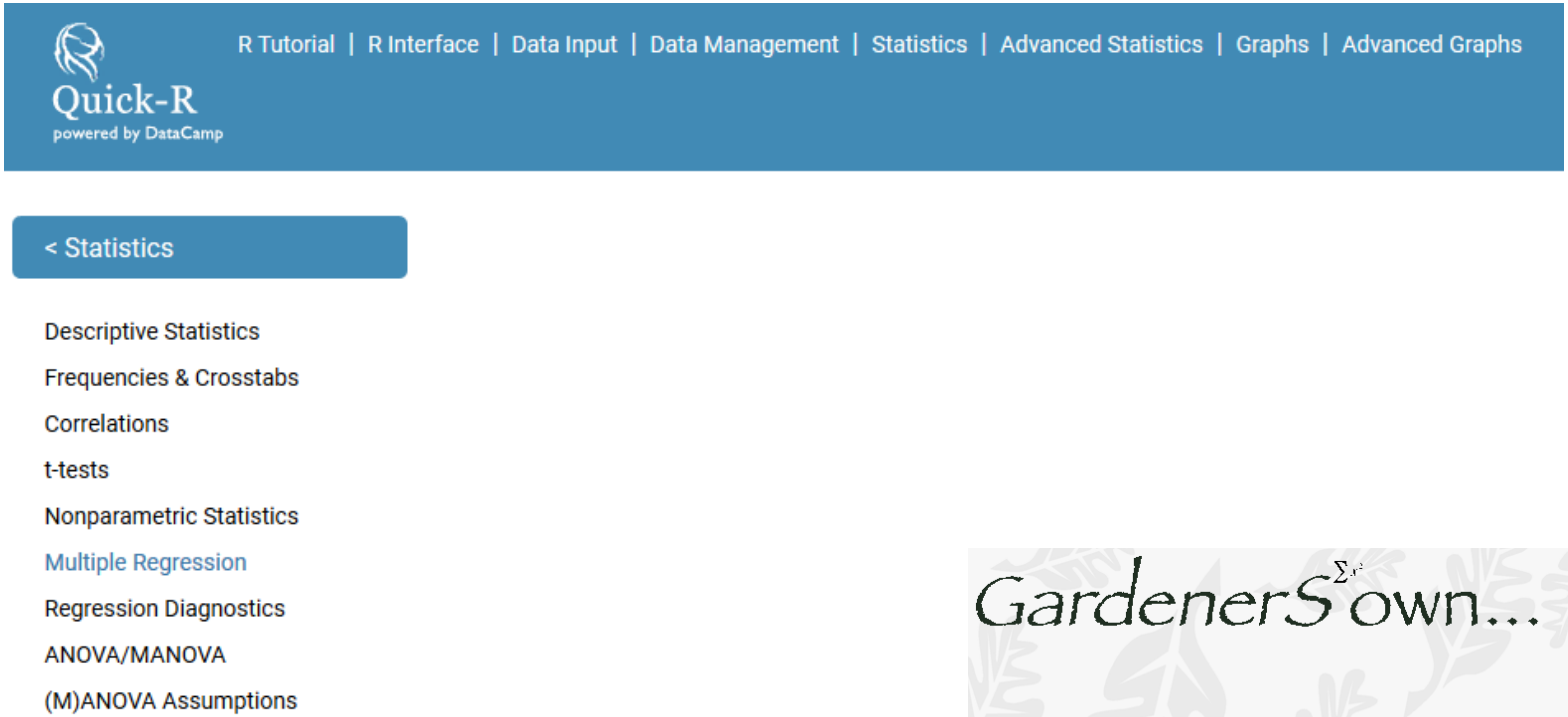
```
sleep
#>    extra group ID
#> 1    0.7     1  1
#> 2   -1.6     1  2
#> 3   -0.2     1  3
#> 4   -1.2     1  4
#> 5   -0.1     1  5
#> 6    3.4     1  6
#> 7    3.7     1  7
#> 8    0.8     1  8
#> 9    0.0     1  9
#> 10   2.0     1 10
#> 11   1.9     2  1
#> 12   0.8     2  2
#> 13   1.1     2  3
#> 14   0.1     2  4
#> 15  -0.1     2  5
#> 16   4.4     2  6
#> 17   5.5     2  7
#> 18   1.6     2  8
#> 19   4.6     2  9
#> 20   3.4     2 10
```

An example for self-study

Kingston University London

Inferential Statistics with R

t-test in R



Kingston University London

Inferential Statistics with R

ANOVA

The t-test is for the comparison of two means.

For more than two means, analysis of variance (ANOVA) is used

Inferential Statistics with R

ANOVA in R: One-way ANOVA

	A	B
1	site	nitrogen
2	1	2.92
3	1	2.88
4	1	3.25
5	1	2.64
6	1	3.28
7	2	3.06
8	2	2.6
9	2	2.55
10	2	2.42
11	2	2.35
12	3	3.41
13	3	3.23
14	3	3.93
15	3	3.74
16	3	3.18

`data(iris)`: Various measurements for three different species of iris

`data(airquality)`: Various meteorological measurements in different months of the year

```
#Conducting an Anova (replace stars with appropriate text e.g., anova, nitrogen, site, anova, anova)
******<-aov(***~as.factor(******))
summary(*****)
TukeyHSD(*****)
```

Kingston University London

Biomeasurement 3e
Dawn Hawkins

R HELP SHEET: Anova

Inferential Statistics with R

ANOVA in R

```
> anova<-aov(nitrogen~as.factor(site))
> summary(anova)
                Df    Sum Sq   Mean Sq  F value   Pr(>F)
as.factor(site)  2     2.043    1.0217    11.95  0.00139 **
Residuals       12     1.026    0.0855
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> TukeyHSD(anova)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = nitrogen ~ as.factor(site))

$`as.factor(site)`
      diff       lwr        upr     p adj
2-1 -0.398 -0.89135525 0.09535525 0.1207456
3-1  0.504  0.01064475 0.99735525 0.0451782
3-2  0.902  0.40864475 1.39535525 0.0010226
```

Degrees of Freedom

Statistic

P Value

Effect Sizes

95% CI of Effect Sizes

Inferential Statistics with R

ANOVA in R

One-way ANOVA:

```
> data(airquality)
> oneway.test(airquality$Wind ~ airquality$Month)

One-way analysis of means (not assuming equal variances)
data:  airquality$Wind and airquality$Month
F = 3.5408, num df = 4.000, denom df = 73.788, p-value = 0.01067

> oneway.test(airquality$Wind ~ airquality$Month, var.equal = TRUE)

One-way analysis of means
data:  airquality$Wind and airquality$Month
F = 3.529, num df = 4, denom df = 148, p-value = 0.00879

> result <- aov(airquality$Wind ~ as.factor(airquality$Month))
> summary(result)
                    Df Sum Sq Mean Sq F value Pr(>F)
airquality$Month     4  164.3   41.07   3.529 0.00879
Residuals          148 1722.3   11.64
(The function aov() always assumes equal variance)
```

Kingston University London

Inferential Statistics with R

ANOVA in R

- Section 1: importation and descriptive analysis
- Section 2: ANOVA
- Section 3: Model check
- Section 4: Mutiple comparisons
- Section 5: Two-way ANOVA
- Section 5: Practicals

Code

ANOVA with R: analysis of the *diet* dataset

D.-L. Couturier / R. Nicholls / M. Fernandes

Last modified: 10 May 2018

A full version of the dataset *diet* may be found online on the U. of Sheffield website https://www.sheffield.ac.uk/polopoly_fs/1.570199!/file/stcp-Rdataset-Diet.csv.

A slightly modified version is available in the data file is stored under data/diet.csv. The data set contains information on 76 people who undertook one of three diets (referred to as diet *A*, *B* and *C*). There is background information such as age, gender, and height. The aim of the study was to see which diet was best for losing weight.

Section 1: importation and descriptive analysis

Lets starts by

- importing the data set *diet* with the function `read.csv()`
- defining a new column *weight.loss*, corresponding to the difference between the initial and final weights (respectively the corresponding to the columns `initial.weight` and `final.weight` of the dataset)
- displaying *weight loss* per *diet type* (column `diet.type`) by means of a boxplot.

Hide

```
diet = read.csv("data/diet.csv",row.names=1)
diet$weight.loss = diet$initial.weight - diet$final.weight
boxplot(weight.loss~diet.type,data=diet,col="light gray",
        ylab = "Weight loss (kg)", xlab = "Diet type")
abline(h=0,col="blue")
```

A full ANOVA analysis example

Kingston University London

./ Bioinformatics Training Materials

Collaboration between Mark Dunning (CRUK Cambridge Institute) and Thomas Carroll (MRC Clincial Sciences)

View on GitHub

Inferential Statistics with R

ANOVA: Assumptions

RDocumentation

Search for packages, functions, etc

bartlett.test

From stats v3.5.1
by R-core R-core@R-project.org

19th Percentile

Bartlett Test Of Homogeneity Of Variances

Performs Bartlett's test of the null that the variances in each of the groups (samples) are the same.

Keywords htest

Usage

```
bartlett.test(x, …)

# S3 method for default
bartlett.test(x, g, …)

# S3 method for formula
bartlett.test(formula, data, subset, na.action, …)
```

Arguments

x	a numeric vector of data values, or a list of numeric data vectors representing the respective samples, or fitted linear model objects (inheriting from class `"lm"`).
g	a vector or factor object giving the group for the corresponding elements of `x`. Ignored if `x` is a list.
formula	a formula of the form `lhs ~ rhs` where `lhs` gives the data values and `rhs` the corresponding groups.
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")`.

Kingston University London

Inferential Statistics with R

ANOVA: Assumptions

INSTANT R

Home | The Book | Datasets | Twitter

Performing Bartlett's test in R

Posted on December 12, 2012 by Sarah Stowell | Comments off

Bartlett's test

Bartlett's test allows you to compare the variance of two or more samples to determine whether they are drawn from populations with equal variance. It is suitable for normally distributed data. The test has the null hypothesis that the variances are equal and the alterntive hypothesis that they are not equal.[1]

This test is useful for checking the assumptions of an analysis of variance.

You can perform Bartlett's test with the `bartlett.test` function. If your data is in stacked form (with the values for both samples stored in one variable), use the command:

```
> bartlett.test(values~groups, dataset)
```

where `values` is the name of the variable containing the data values and `groups` is the name of the variable that specifies which sample each value belongs too.

If your data is in unstacked form (with the samples stored in separate variables) nest the variable names inside the `list` function as shown below.

```
> bartlett.test(list(dataset$sample1, dataset$sample2,
   dataset$sample3))
```

If you are unsure whether your data is in stacked or unstacked form, see the article *Stacking a dataset in R* for examples of data in both forms.

Kingston University London

Inferential Statistics with R

ANOVA: Assumptions

Bartlett test for homogeneity of variances:

```
> data(airquality)
> oneway.test(airquality$Wind ~ airquality$Month)

    One-way analysis of means (not assuming equal variances)
    data:  airquality$Wind and airquality$Month
    F = 3.5408, num df = 4.000, denom df = 73.788,
    p-value = 0.0107

> bartlett.test(airquality$Wind ~ airquality$Month)

    Bartlett test of homogeneity of variances
    data:  airquality$Wind by airquality$Month
    Bartlett's K-squared = 1.6178, df = 4,
    p-value = 0.8056        # P = 0.81 → variances are homogeneous
```


```
> data(airquality)
> oneway.test(airquality$Wind ~ airquality$Month)

        One-way analysis of means (not assuming equal variances)

data:  airquality$Wind and airquality$Month
F = 3.5408, num df = 4.000, denom df = 73.788, p-value = 0.01067

> bartlett.test(airquality$Wind ~ airquality$Month)

        Bartlett test of homogeneity of variances

data:  airquality$Wind by airquality$Month
Bartlett's K-squared = 1.6178, df = 4, p-value = 0.8056
```

Inferential Statistics with R

ANOVA: Assumptions

RDocumentation

Search for packages, functions, etc

levene.test

From lawstat v3.2
by Vyacheslav Lyubchich

94th Percentile

Levene's Test Of Equality Of Variances

The function performs the following tests for equality of the k population variances: classical Levene's test, the robust Brown-Forsythe Levene-type test using the group medians and the robust Levene-type test using the group trimmed mean. More robust versions of the test using the correction factor or structural zero removal method are also available. Two options for calculating critical values, namely, approximated and bootstrapped, are available. Instead of the ANOVA statistic suggested by Levene, the Kruskal-Wallis ANOVA may also be applied using this function. By default, NAs from the data are omitted.

Keywords htest

Usage

```
levene.test(y, group, location=c("median", "mean", "trim.mean"), trim.alpha=0.25,
bootstrap = FALSE, num.bootstrap=1000, kruskal.test=FALSE,
correction.method=c("none","correction.factor","zero.removal","zero.correction"))
```

Arguments

y	a numeric vector of data values.
group	factor of the data.
location	the default option is "median" corresponding to the robust Brown-Forsythe Levene-type procedure; "mean" corresponds to the classical Levene's procedure, and "trim.mean" corresponds to the robust Levene-type procedure using the group trimmed means.
trim.alpha	the fraction (0 to 0.5) of observations to be trimmed from each end of 'x' before the mean is computed.
bootstrap	the default option is FALSE, i.e., no bootstrap; if the option is set to TRUE, the function performs the bootstrap method described in Lim and Loh (1996) for Levene's test.
num.bootstrap	number of bootstrap samples to be drawn when the bootstrap option is set to TRUE; the default value is 1000.
kruskal.test	use of Kruskal-Wallis statistic. The default option is FALSE, i.e., the usual ANOVA statistic is used in place of Kruskal-Wallis statistic.

Kingston University London

Inferential Statistics with R

ANOVA: Assumptions

6. Fisher's F-Test

Fisher's F test can be used to check if two samples have same variance.

```
var.test(x, y)  # Do x and y have the same variance?
```

Alternatively fligner.test() and bartlett.test() can be used for the same purpose.

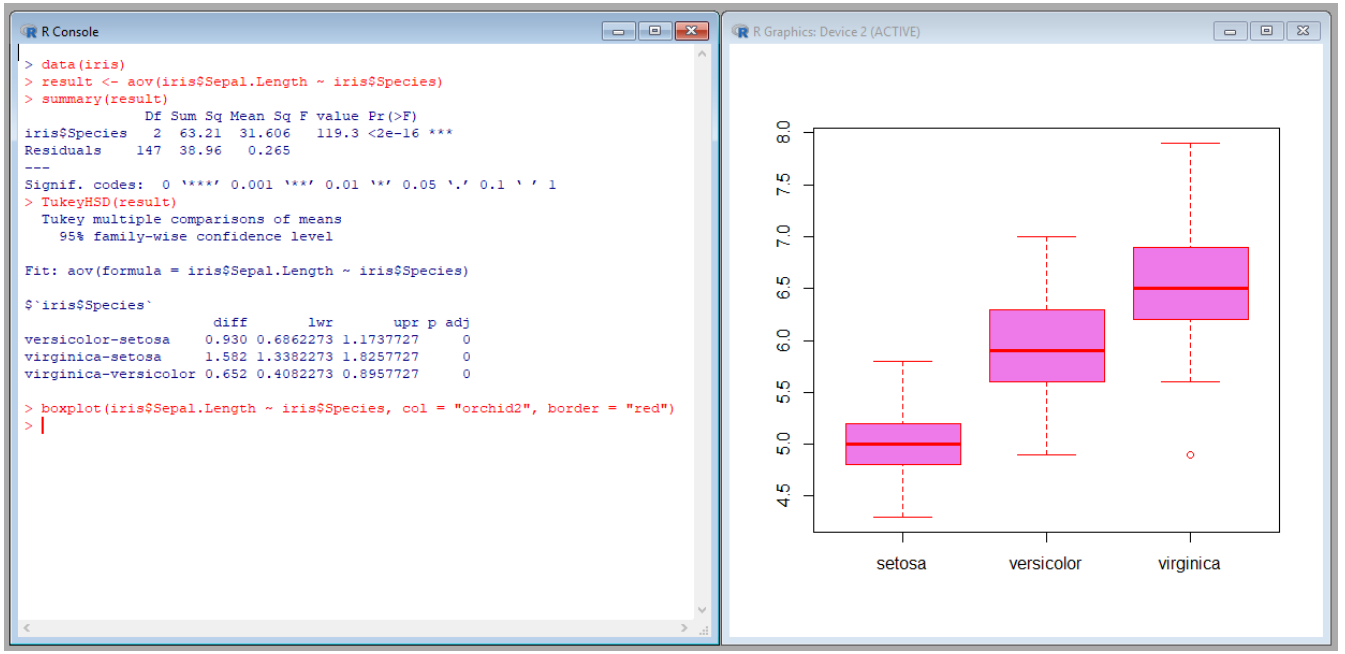
```
> data(iris)
> var.test(iris$Sepal.Length, iris$Petal.Length)

F test to compare two variances
data:  iris$Sepal.Length and iris$Petal.Length
F = 0.22004, num df = 149, denom df = 149, p-value < 2.2e-16
95 percent confidence interval: 0.1594015 0.3037352
ratio of variances: 0.2200361

> var.test(iris$Sepal.Length, iris$Petal.Width)

F test to compare two variances
data:  iris$Sepal.Length and iris$Petal.Width
F = 1.1802, num df = 149, denom df = 149, p-value = 0.313
95 percent confidence interval: 0.8549641 1.6291105
ratio of variances: 1.180183

Check also:
> ks.test(iris$Sepal.Length, iris$Petal.Width)
> boxplot(iris$Sepal.Length, iris$Petal.Width)
```

Inferential Statistics with R

ANOVA: Post-hoc tests

Tukey's HSD test as a post-hoc pairwise difference test:

```
data(iris)
result <- aov(iris$Sepal.Length ~ iris$Species)
summary(result)
TukeyHSD(result)
boxplot(iris$Sepal.Length ~ iris$Species, col = "orchid2",
border = "red")
```



Kingston University London

Inferential Statistics with R

ANOVA in R

Cookbook for R » Statistical analysis » ANOVA

ANOVA

Problem

You want to compare multiple groups using an ANOVA.

Solution

Suppose this is your data:

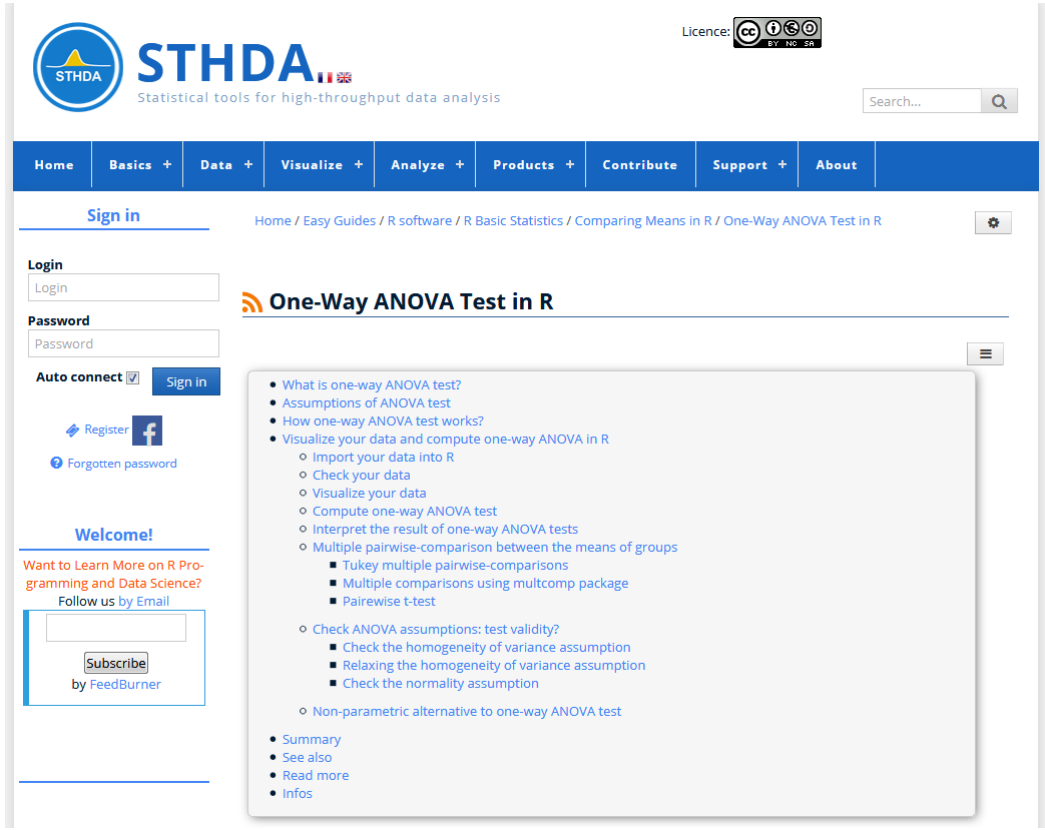
```
data <- read.table(header=TRUE, text='
 subject sex   age before after
       1   F   old    9.5   7.1
       2   M   old   10.3  11.0
       3   M   old    7.5   5.8
       4   F   old   12.4   8.8
       5   M   old   10.2   8.6
       6   M   old   11.0   8.0
       7   M young    9.1   3.0
       8   F young    7.9   5.2
       9   F   old    6.6   3.4
      10   M young    7.7   4.0
      11   M young    9.4   5.3
      12   M   old   11.6  11.3
      13   M young    9.9   4.6
      14   F young    8.6   6.4
      15   F young   14.3  13.5
      16   F   old    9.2   4.7
      17   M young    9.8   5.1
      18   F   old    9.9   7.3
      19   F young   13.0   9.5
      20   M young   10.2   5.4
      21   M young    9.0   3.7
      22   F young    7.9   6.2
      23   M   old   10.1  10.0
      24   M young    9.0   1.7
      25   M young    8.6   2.9
      26   M young    9.4   3.2
      27   M young    9.7   4.7
      28   M young    9.3   4.9
      29   F young   10.7   9.8
      30   M   old    9.3   9.4
')

# Make sure subject column is a factor, so that it's not treated as a continuous
# variable.
data$subject <- factor(data$subject)
```

An example for self-study

Kingston University London

Inferential Statistics with R

ANOVA in R



An example for self-study

Kingston University London

Inferential Statistics with R

ANOVA in R

```
data(InsectSprays)
```

ANOVA in R

1-Way ANOVA

We're going to use a data set called InsectSprays. 6 different insect sprays (1 Independent Variable with 6 levels) were tested to see if there was a difference in the number of insects found in the field after each spraying (Dependent Variable).

```
> attach(InsectSprays)
> data(InsectSprays)
> str(InsectSprays)
'data.frame':   72 obs. of  2 variables:
 $ count: num  10 7 20 14 14 12 10 23 17 20 ...
 $ spray: Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

One-way ANOVA

taylor.curley@gatech.edu

Last Updated: 11/19/2017

- Overview
- Dataset
- Computing the One-Way ANOVA
- Pairwise-Comparisons
  - Tukey's HSD
  - Multiple Comparisons - General Linear Hypothesis Test
  - Scheffe's Test
  - Pairwise T-Tests
- ANOVA Assumptions
  - Homoscedasicity
  - Normality of Residuals

An example for self-study

Kingston University London

Inferential Statistics with R

ANOVA in R

R Tutorial Series

By John M Quick

The *R Tutorial Series* provides a collection of user-friendly tutorials to people who want to learn how to use *R* for statistical analysis.

My *Statistical Analysis with R* book is available from Packt Publishing and Amazon.

A full tutorial on ANOVA

ANOVA

One-Way Omnibus ANOVA

Two-Way Omnibus ANOVA

One-Way ANOVA with Pairwise Comparisons

Two-Way ANOVA with Pairwise Comparisons

Two-Way ANOVA with Interactions

One-Way Repeated Measures ANOVA

Two-Way Repeated Measures ANOVA

Two-Way ANOVA with Unequal Sample Sizes

ANOVA Pairwise Comparison Methods

Reshape Package for ANOVA Data

Kingston University London

Inferential Statistics with R

ANOVA in R
Non-parametric version

11.23 Performing Robust ANOVA (Kruskal–Wallis Test)

Problem

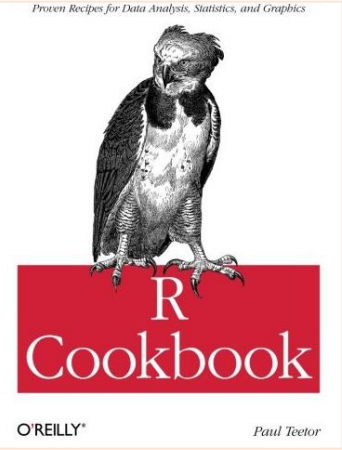
Your data is divided into groups. The groups are not normally distributed, but their distributions have similar shapes. You want to perform a test similar to ANOVA—you want to know if the group medians are significantly different.

Solution

Create a factor that defines the groups of your data. Use the `kruskal.test` function, which implements the Kruskal–Wallis test. Unlike the ANOVA test, this test does not depend upon the normality of the data:

```
> kruskal.test(x ~ f)
```

Here, `x` is a vector of data and `f` is a grouping factor. The output includes a *p*-value. Conventionally, $p < 0.05$ indicates that there is a significant difference between the medians of two or more groups whereas $p > 0.05$ provides no such evidence.



Kingston University London

Inferential Statistics with R

t-test: Statistical Power

RDocumentation

Search for packages, functions, etc

power.t.test

From stats v3.5.1
by R-core R-core@R-project.org

19th Percentile

Power Calculations For One And Two Sample T Tests

Compute the power of the one- or two- sample t test, or determine parameters to obtain a target power.

Keywords htest

Usage

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
             power = NULL,
             type = c("two.sample", "one.sample", "paired"),
             alternative = c("two.sided", "one.sided"),
             strict = FALSE, tol = .Machine$double.eps^0.25)
```

Arguments

n	number of observations (per group)
delta	true difference in means
sd	standard deviation
sig.level	significance level (Type I error probability)
power	power of test (1 minus Type II error probability)
type	string specifying the type of t test. Can be abbreviated.
alternative	one- or two-sided test. Can be abbreviated.
strict	use strict interpretation in two-sided case
tol	numerical tolerance used in root finding, the default providing (at least) four significant digits.

Kingston University London

Inferential Statistics with R

ANOVA: Statistical Power

RDocumentation

Search for packages, functions, etc

power.anova.test

From stats v3.5.1
by R-core R-core@R-project.org

19th Percentile

Power Calculations For Balanced One-Way Analysis Of Variance Tests

Compute power of test or determine parameters to obtain target power.

Keywords htest

Usage

```
power.anova.test(groups = NULL, n = NULL,
                 between.var = NULL, within.var = NULL,
                 sig.level = 0.05, power = NULL)
```

Arguments

groups	Number of groups
n	Number of observations (per group)
between.var	Between group variance
within.var	Within group variance
sig.level	Significance level (Type I error probability)
power	Power of test (1 minus Type II error probability)

Details

Exactly one of the parameters `groups`, `n`, `between.var`, `power`, `within.var`, and `sig.level` must be passed as NULL, and that parameter is determined from the others. Notice that `sig.level` has non-NULL default so NULL must be explicitly passed if you want it computed.

Kingston University London

Inferential Statistics with R

Regression Analysis in R

Quick-R
powered by DataCamp

R Tutorial | R Interface | Data Input | Data Management | Statistics | Advanced Statistics | Graphs | Advanced Graphs

< Statistics

- Descriptive Statistics
- Frequencies & Crosstabs
- Correlations
- t-tests
- Nonparametric Statistics
- Multiple Regression
- Regression Diagnostics
- ANOVA/MANOVA
- (M)ANOVA Assumptions

Multiple (Linear) Regression

R provides comprehensive support for multiple linear regression. The topics below are provided in order of increasing complexity.

Fitting the Model

```
# Multiple Linear Regression Example
fit <- lm(y ~ x1 + x2 + x3, data=mydata)
summary(fit) # show results
```

R

```
1  myModel <- lm(q4 ~ q1 + q2 + q3, data = mydata100)
2  summary(myModel)
3  plot(myModel)
```

Kingston University London

Inferential Statistics with R

Regression Analysis in R

Typical regression analysis in R:

Univariable:

```
data(iris)
attach(iris)
modelname <- lm(Sepal.Length ~ Sepal.Width)
summary(modelname)
plot(modelname)
```

Multivariable:

```
modelname <- lm(Sepal.Length ~ Sepal.Width +
           Petal.Length + Petal.Width)
summary(modelname)
plot(modelname)

# You can also explore interactions:
modelname <- lm(Sepal.Length ~ Sepal.Width *
           Petal.Length * Petal.Width)
```

Inferential Statistics with R

Regression Analysis in R

3. Identifying the key elements of the output
Following the instructions above will produce the following output in the **R Console** window: the **key elements** are annotated in blue.

```
> #Importing data from tab delimited file (replace stars with an appropriate object name e.g.,reeds)
> species<-read.table(file.choose(),header=TRUE)
> attach(species)
> names(species)
[1] "reserve"  "area"     "richness"
>
> #Conducting a regression (replace stars with appropriate text e.g.,nitrogen,site)
> regression<-lm(richness~area)
> summary(regression)

Call:
lm(formula = richness ~ area)

Residuals:
     1      2      3      4      5      6
0.4143 -2.8844  3.0011  0.1401  2.7983 -3.4693

Coefficients:
             Estimate   Std.Error  t value  Pr(>|t|)
(Intercept) 8.3102269  1.8215891    4.562   0.0103 *
area        0.0027429  0.0007245    3.786   0.0193 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.057 on 4 degrees of freedom
Multiple R-squared: 0.7818,    Adjusted R-squared: 0.7273
F-statistic: 14.33 on 1 and 4 DF,  p-value: 0.01934

> confint(regression)
                  2.5 %        97.5 %
(Intercept)  3.2526847891 13.367769102
area         0.0007312762  0.004754461
```

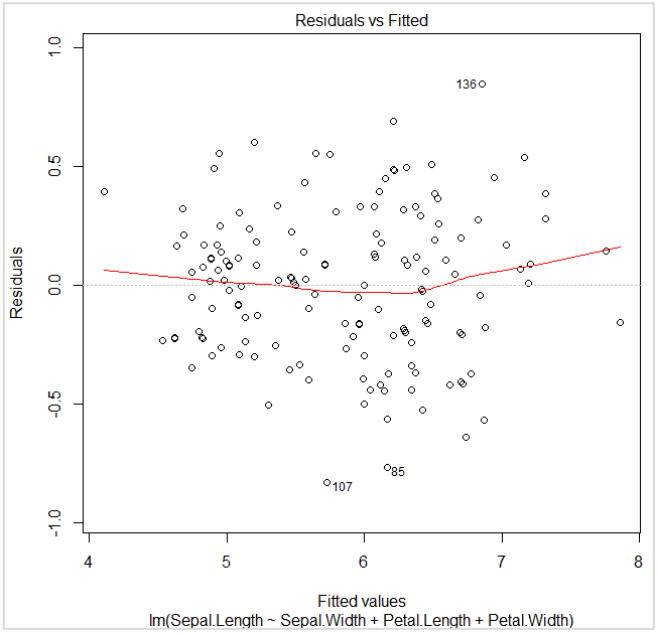
Intercept
Effect Size (slope)
Coefficient of determination
Statistic
P Value
Degrees of Freedom
95% CI of Effect Size (slope)

In summary the key information from the test is
$y = 0.003\ x + 8.310$; $F_{1,4} = 14.332$, $P = 0.019$, $R^2 = 0.782$

Inferential Statistics with R

Regression Analysis in R

```
> data(iris)
> # Modelname will be "fit"
> fit <- lm(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width, data = iris)
> summary(fit)

Call:
lm(formula = Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width,
    data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-0.82816 -0.21989  0.01875  0.19709  0.84570

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.85600    0.25078   7.401 9.85e-12 ***
Sepal.Width   0.65084    0.06665   9.765  < 2e-16 ***
Petal.Length  0.70913    0.05672  12.502  < 2e-16 ***
Petal.Width  -0.55648    0.12755  -4.363 2.41e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3145 on 146 degrees of freedom
Multiple R-squared:  0.8586,    Adjusted R-squared:  0.8557
F-statistic: 295.5 on 3 and 146 DF,  p-value: < 2.2e-16

> plot(fit)
```

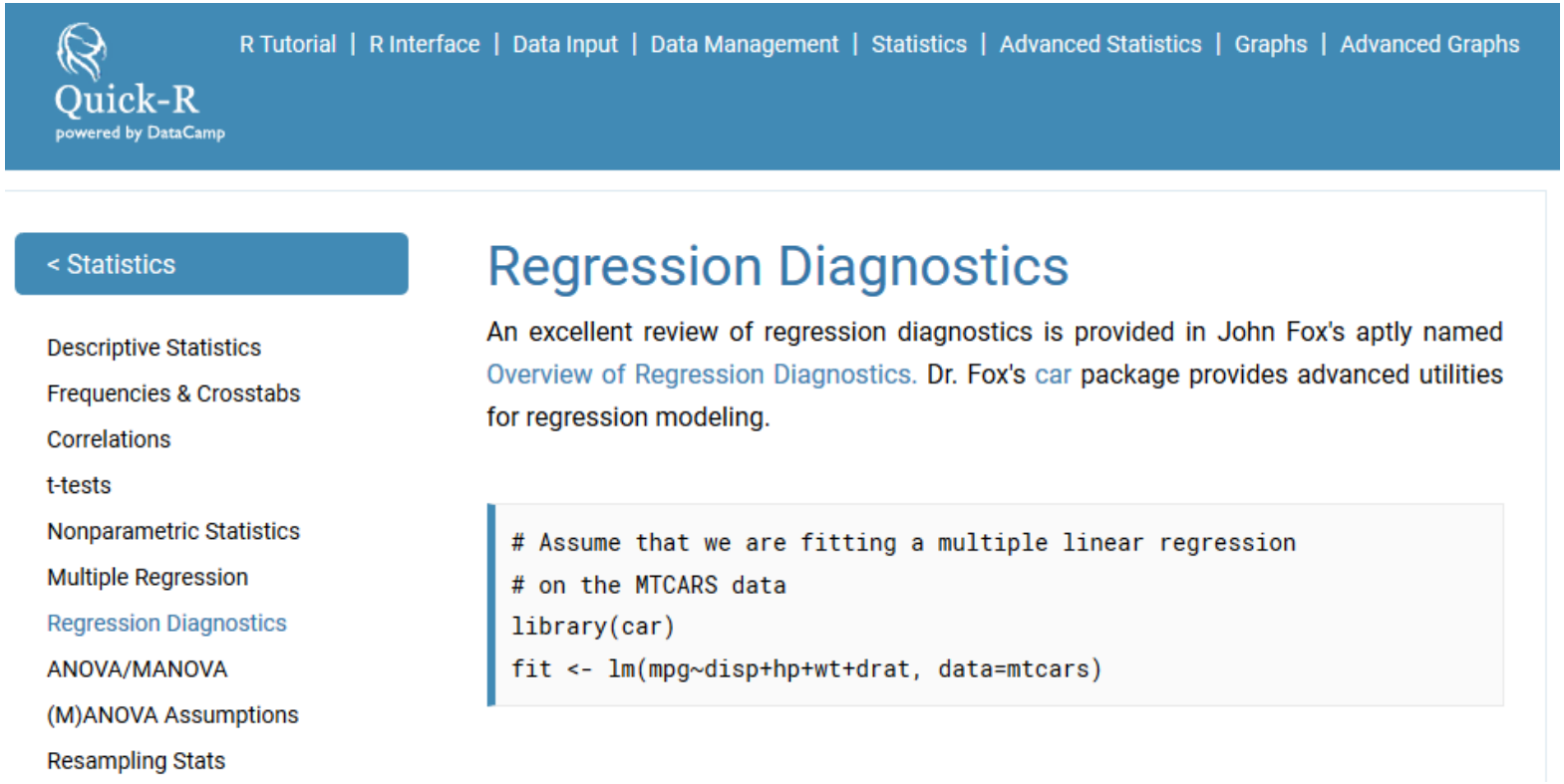



```r
myModel <- lm(q4 ~ q1 + q2 + q3, data = mydata100)
summary(myModel)
plot(myModel)
```

Kingston University London

Inferential Statistics with R

Regression Analysis in R



https://www.statmethods.net/stats/rdiagnostics.html

Kingston University London

Inferential Statistics with R

Regression Analysis in R

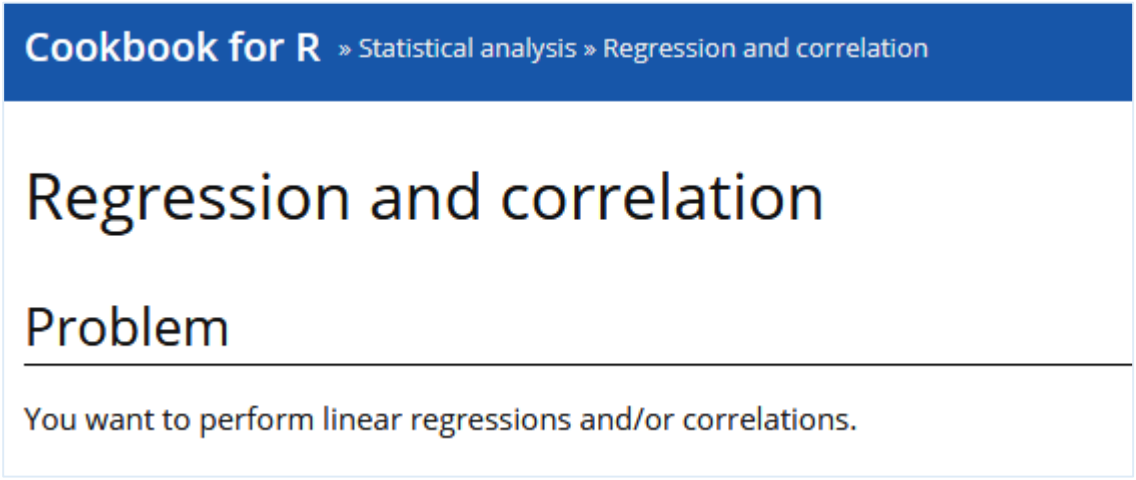
COEFFICIENT OF DETERMINATION (r^2)

The coefficient of determination expresses the proportion of the variance in one variable that is accounted for, or "explained," by the variance in the other variable. It is found by squaring the value of r, and its symbol is therefore r2. So if a study finds a correlation (r) of 0.40 between salt intake and blood pressure, it could be concluded that 0.40 x 0.40 = 0.16, or 16% of the variance in blood pressure in this study is accounted for by variance in salt intake. A correlation between two variables does not demonstrate a causal relationship between the two variables, no matter how strong it is. Correlation is merely a measure of the variable's statistical association, not of their causal relationship -so the correlation between salt intake and blood pressure does not necessarily mean that the changes in salt intake causes the changes in blood pressure. Inferring a causal relationship between two variables on the basis of a correlation is a common and fundamental error. Furthermore, the fact that a correlation is present between two variables in a sample does not necessarily mean that the correlation actually exists in the population. When a correlation has been found between two variables in a sample, the researcher will normally wish to test the null hypothesis that there is no correlation between the two variables (i.e., that r = 0) in the population. This is done with a special form of t-test.

Inferential Statistics with R

Regression Analysis in R

Example



Kingston University London

Inferential Statistics with R

Regression Analysis in R

8.6.3. Basic logistic regression analysis using R ②

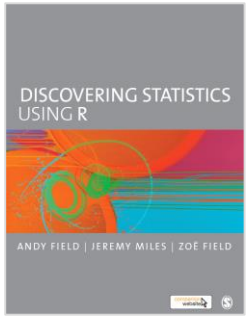
To do logistic regression, we use the **glm()** function. The *glm()* function is very similar to the *lm()* function that we saw in Chapter 7. While *lm* stands for 'linear model', *glm* stands for 'generalized linear model' – that is, the basic linear model that has been generalized to other sorts of situations. The general form of this function is:

```
newModel<-glm(outcome ~ predictor(s), data = dataFrame, family = name of a
distribution, na.action = an action)
```

in which:

- *newModel* is an object created that contains information about the model. We can get summary statistics for this model by executing *summary(newModel)*.
- *outcome* is the variable that you're trying to predict, also known as the dependent variable. In this example it will be the variable Cured.
- *predictor(s)* lists the variable or variables from which you're trying to predict the outcome variable. In this example it will be the variables Cured and Duration.
- *dataFrame* is the name of the dataframe from which your outcome and predictor variables come.
- *family* is the name of a distribution (e.g., Gaussian, binomial, poisson, gamma).
- *na.action* is an optional command. If you have complete data (as we have here) you can ignore it, but if you have missing values (i.e., NAs in the dataframe) then it can be useful – see R's Souls' Tip 7.1).

Kingston University London



Inferential Statistics with R

Regression Analysis in R

Logistic Regression

Logistic regression is useful when you are predicting a binary outcome from a set of continuous predictor variables. It is frequently preferred over discriminant function analysis because of its less restrictive assumptions.

```
# Logistic Regression
# where F is a binary factor and
# x1-x3 are continuous predictors
fit <- glm(F~x1+x2+x3,data=mydata,family=binomial())
summary(fit) # display results
confint(fit) # 95% CI for the coefficients
exp(coef(fit)) # exponentiated coefficients
exp(confint(fit)) # 95% CI for exponentiated coefficients
predict(fit, type="response") # predicted values
residuals(fit, type="deviance") # residuals
```

Inferential Statistics with R

Regression Analysis in R

```
> data(infert)
> head(infert, 2)
  education age parity induced case spontaneous stratum pooled.stratum
1    0-5yrs  26      6       1    1           2       1               3
2    0-5yrs  42      1       1    1           0       2               1
> dim(infert)
[1] 248   8
> fit <- glm(case ~ age + parity + induced + spontaneous, data = infert, family = binomial())
> summary(fit)

Call:
glm(formula = case ~ age + parity + induced + spontaneous, family = binomial(),
    data = infert)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6281  -0.8055  -0.5299   0.8669   2.6141

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.85239    1.00428  -2.840  0.00451 **
age          0.05318    0.03014   1.764  0.07767 .
parity      -0.70883    0.18091  -3.918 8.92e-05 ***
induced      1.18966    0.28987   4.104 4.06e-05 ***
spontaneous  1.92534    0.29863   6.447 1.14e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

"age" is not statistically significant ($P > 0.05$) as a predictor

Predictive model:

case = -2.85 + (0.053 * age) + (-0.709 * parity) + (1.190 * induced) + (1.925 * spontaneous)

Kingston University London

Inferential Statistics with R

Regression Analysis in R

```
> confint(fit, level = 0.95)
Waiting for profiling to be done...
                  2.5 %     97.5 %
(Intercept) -4.87438890 -0.9222185
age         -0.00542269  0.1132530
parity      -1.08149001 -0.3692163
induced      0.63613264  1.7774032
spontaneous  1.36636751  2.5412467
> exp(coef(fit))
(Intercept)         age      parity     induced spontaneous
 0.05770622  1.05462050  0.49221973  3.28595133  6.85746772
> exp(confint(fit))
Waiting for profiling to be done...
                  2.5 %     97.5 %
(Intercept) 0.007639761  0.3976359
age         0.994591986  1.1199153
parity      0.339089903  0.6912759
induced     1.889160664  5.9144775
spontaneous 3.921081497 12.6954881
```

exp(beta coefficients) = odds ratio

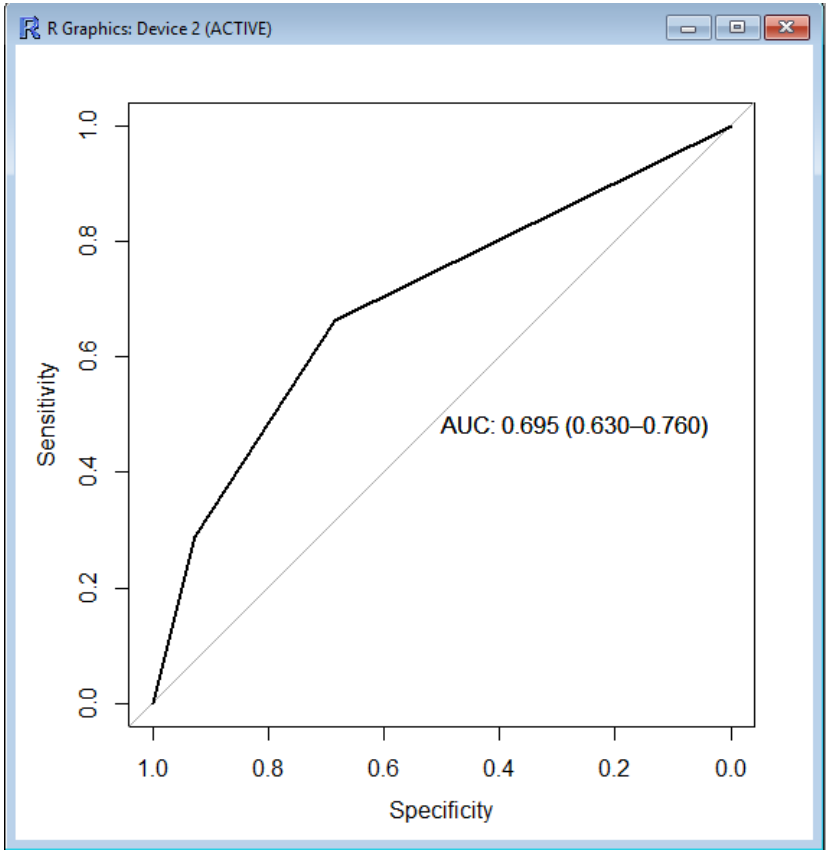
Kingston University London

Inferential Statistics with R

Regression Analysis in R

The variable "spontaneous" is the strongest predictor of the outcome (coeff=1.925 SE=0.298 Z=6.447 P=1.14e-10 OR=6.3). Let's check the AUC value for its predictability power.

```
> library(pROC)
> plot.roc(infert$case, infert$spontaneous, print.auc = TRUE, ci = TRUE)
```



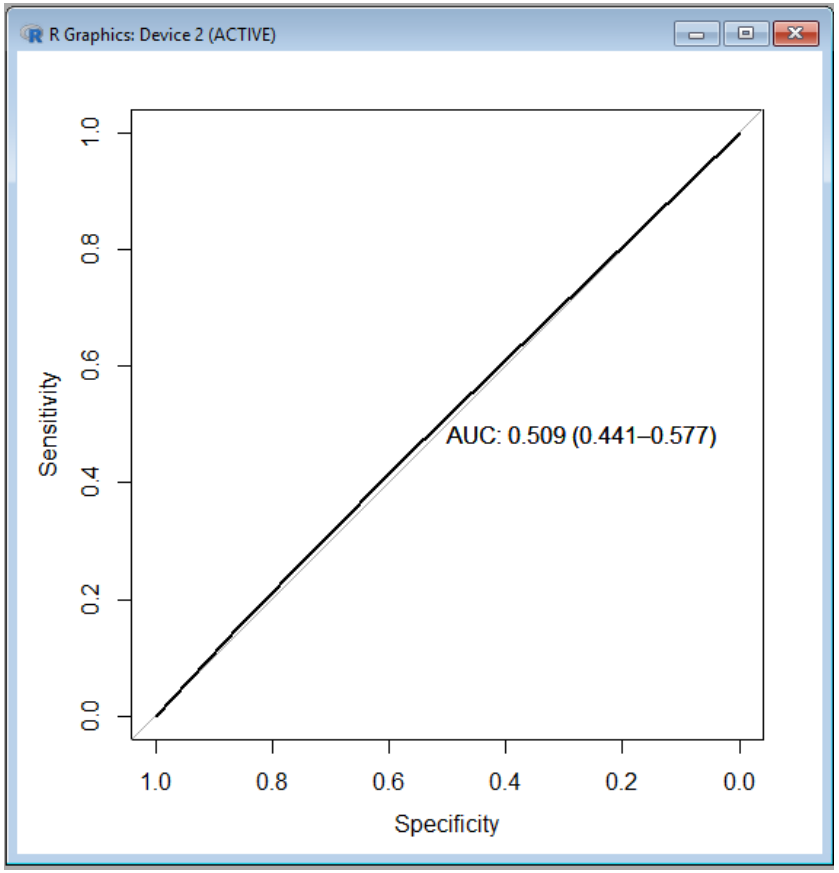
Kingston University London

Inferential Statistics with R

Regression Analysis in R

The variable "induced" is the second strongest predictor of the outcome (coeff=1.189 SE=0.289 Z=4.104 P=4.06e-05 OR=3.3). Let's check the AUC value for its predictability power.

```
plot.roc(infert$case, infert$induced, print.auc = TRUE, ci = TRUE)
```



Kingston University London

Inferential Statistics with R

Regression Analysis in R

The C-statistics for the whole model (including all predictors) can be calculated using the DescTools::Cstat function

```
library(DescTools); Cstat(fit)
[1] 0.7725039

# A reasonable model as C-stat > 0.70
# Models with C-stat > 0.80 are called
strong
```

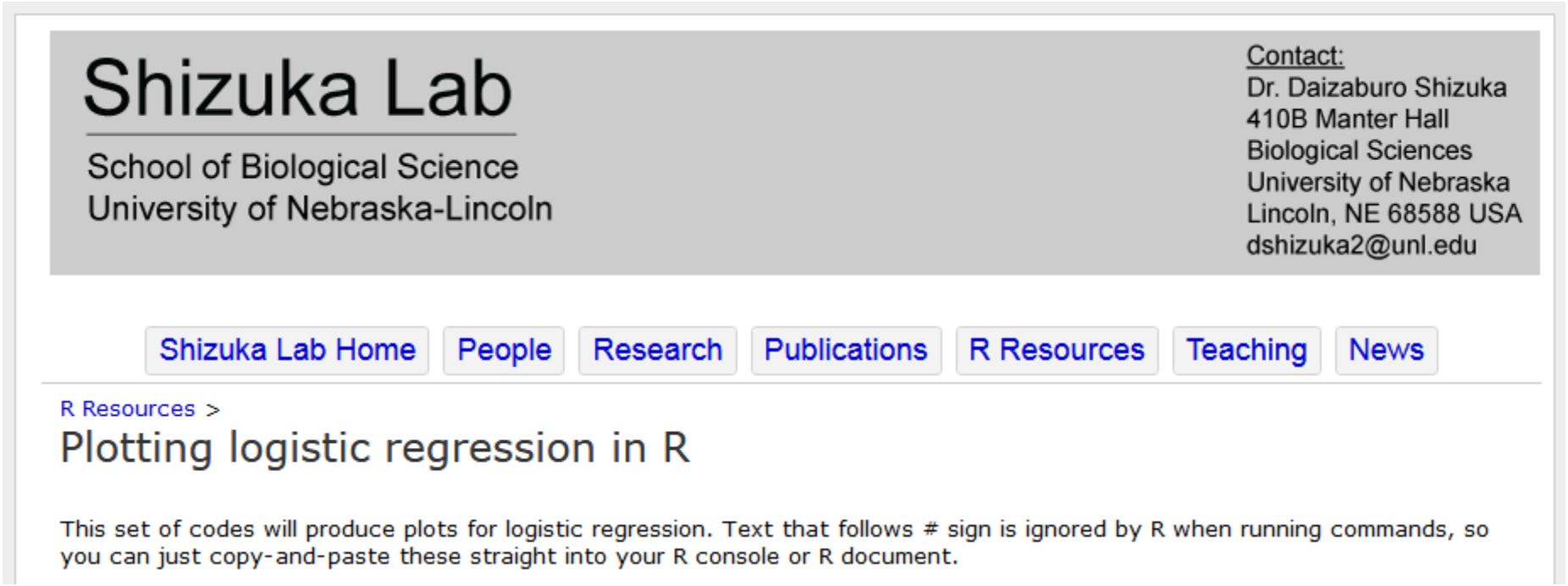
Inferential Statistics with R

Regression Analysis in R

Now, run Script: `s4.R`

Inferential Statistics with R

Regression Analysis in R



Shizuka Lab

School of Biological Science
University of Nebraska-Lincoln

Contact:
Dr. Daizaburo Shizuka
410B Manter Hall
Biological Sciences
University of Nebraska
Lincoln, NE 68588 USA
dshizuka2@unl.edu

Shizuka Lab Home | People | Research | Publications | R Resources | Teaching | News

R Resources >

Plotting logistic regression in R

This set of codes will produce plots for logistic regression. Text that follows # sign is ignored by R when running commands, so you can just copy-and-paste these straight into your R console or R document.

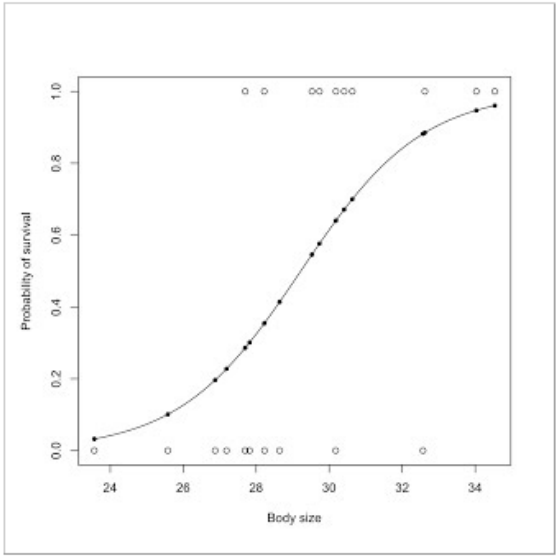
Kingston University London

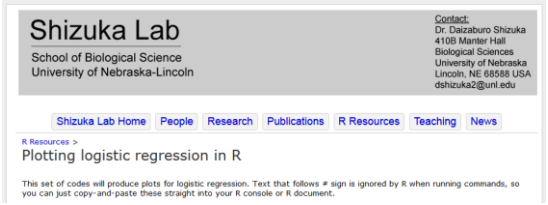
Inferential Statistics with R

Regression Analysis in R

Plotting method 1: manual plotting with actual logistic regression

```
quartz(title="bodysize vs. survival") # creates a quartz window with title

plot(bodysize,survive,xlab="Body size",ylab="Probability of survival") # plot with body size on x-axis
and survival (0 or 1) on y-axis
g=glm(survive~bodysize,family=binomial,dat) # run a logistic regression model (in this case, generalized
linear model with logit link). see ?glm

curve(predict(g,data.frame(bodysize=x),type="resp"),add=TRUE) # draws a curve based on prediction from
logistic regression model

points(bodysize,fitted(g),pch=20) # optional: you could skip this draws an invisible set of points of
body size survival based on a 'fit' to glm model. pch= changes type of dots.
```




Shizuka Lab

School of Biological Science
University of Nebraska-Lincoln

Contact:
Dr. Daizaburo Shizuka
410B Manter Hall
Biological Sciences
University of Nebraska
Lincoln, NE 68588 USA
dshizuka2@unl.edu

Shizuka Lab Home | People | Research | Publications | R Resources | Teaching | News

R Resources >

Plotting logistic regression in R

This set of codes will produce plots for logistic regression. Text that follows # sign is ignored by R when running commands, so you can just copy-and-paste these straight into your R console or R document.

Kingston University London

Inferential Statistics with R

Regression Analysis in R

stackoverflow Search...

Home

PUBLIC

Stack Overflow

Tags

Users

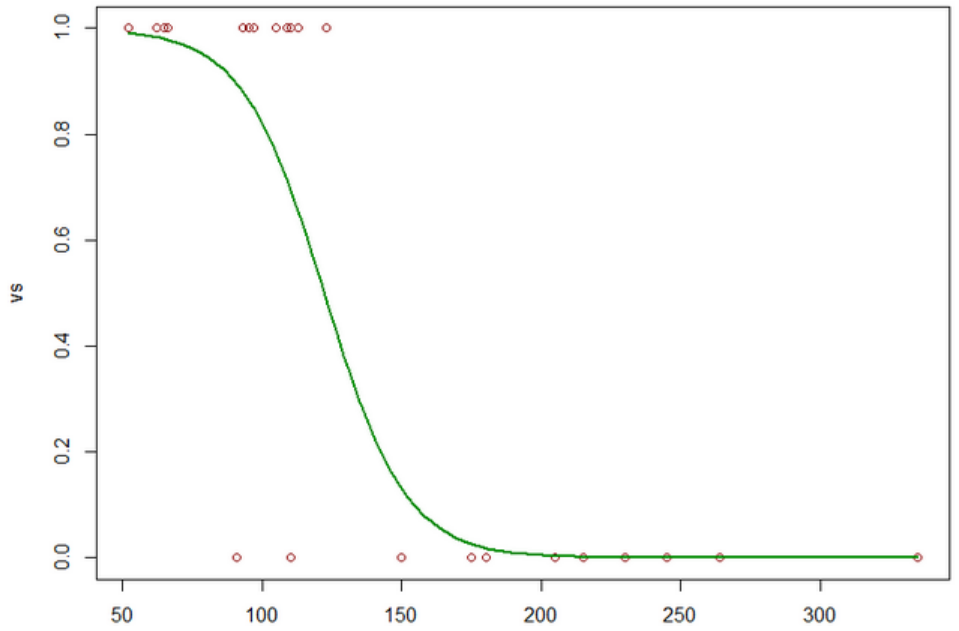
Jobs

Teams
Q&A for work

Learn More

7

```
fit = glm(vs ~ hp, data=mtcars, family=binomial)
newdat <- data.frame(hp=seq(min(mtcars$hp), max(mtcars$hp),len=100))
newdat$vs = predict(fit, newdata=newdat, type="response")
plot(vs~hp, data=mtcars, col="red4")
lines(vs ~ hp, newdat, col="green4", lwd=2)
```



Kingston University London

Inferential Statistics with R

Regression Analysis in R

For more on logistic regression
and on the use of " `ggplot2` "
for plot generation:

Cookbook for R » Statistical analysis » Logistic regression

Logistic regression

Problem

You want to perform a logistic regression.

Solution

A logistic regression is typically used when there is one dichotomous outcome variable (such as winning or losing), and a continuous predictor variable which is related to the probability or odds of the outcome variable. It can also be used with categorical predictors, and with multiple predictors.

Next

Beyond Basic Statistics