A Brief introduction to R

An Open Source software environment for statistics and graphics

Albyn Jones[1]

[1]jones@reed.edu
www.people.reed.edu/∼jones/courses/141

What is R?

R is an open source version of the **S** language environment for statistical computing and graphics.

The **R** project homepage
Albyn's Math 141 Lab notes

You should be able to find **R** in the Applications folder.

Why use R?

- Price: **R** is free.

Why use R?

- Price: **R** is free.
- Quality: **R** is the platform of choice for the development of new statistical software.

Why use R?

- Price: **R** is free.
- Quality: **R** is the platform of choice for the development of new statistical software.
- Extensibility: **R** is a programing language, plus there are over 5100 packages on **CRAN**, almost 750 at BioConductor, and more, for example RStudio.

Data Types

R has several basic data types, plus more complicated objects:

- numeric, character, logical, factor
- matrix, array, data.frame, list

```
X <- c(1, 27, pi/2)
Blue <- "blue"
Big <- X > 3
```

Arithmetic

Arithmetic operations in **R** were designed to facilitate standard operations arising in statistical work. The basic operators are

```
+ - * \ ^
```

```
5+2
```

```
X <- seq(1,2,.1)
X
```

```
X+2
```

Try them!

Functions

R includes standard mathematical functions

exp(), *log*(), *log*10(), *sin*(), *cos*(), *etc.*

as well as statistical functions, for example:

Finally, **R** has many functions for matrix computations, optimization, plus packages devoted to biostatistics, econometrics, psychometrics, GIS and maps, analyzing networks etc.

Functions

R includes standard mathematical functions

exp(), *log*(), *log*10(), *sin*(), *cos*(), *etc.*

as well as statistical functions, for example:

mean(), *sd*(), *var*(), *median*(), *cor*(), *t.test*(), *etc.*

Finally, **R** has many functions for matrix computations, optimization, plus packages devoted to biostatistics, econometrics, psychometrics, GIS and maps, analyzing networks etc.

Data import/export

R can import and export data in most standard formats: plain text, csv files, etc.

We will see examples soon.

Reproducibility

It is good policy to create a document containing the commands used in your analysis!

On a Mac: select New Document from the files menu. Don't forget to save it periodically!

RStudio makes this easy.

Programming: Loops

A silly way to compute 10 factorial:

```
N <- 1    # initialize the variable N
for(i in 1:10){
    N <- N*i
    }
N

#  compare: prod(1:10), factorial(10), gamma(11)
```

Programming: Functions

Computers don't do exact arithmetic, there is usually some rounding error. Let's find the machine epsilon (the maximum relative rounding error for a single arithmetic operation):

```
Macheps <- function(){
    eps <- 1
    done <- FALSE
    while(!done){
        eps <- eps/2
        if( 1 == 1+eps) done <- TRUE
        }
    eps
    }
```

The value to be returned is the expression on the last line before the final close bracket.

Import Smoking and Birthweight data

```
Bwt <- read.csv(
    "http://people.reed.edu/~jones/141/Bwt.csv")

   #  What do we have here?

names(Bwt)
dim(Bwt)
head(Bwt)
summary(Bwt)
```

Look at the data!

```
attach(Bwt)
   # convenient, but dangerous
   # never do this inside a function!
plot(bwt ~ gestation)
plot(bwt ~ age)
   # etc.
plot(bwt ~ gestation, pch=19,
          col=ifelse(smoke==1,"red","blue"))
   # any interesting cases here?
```

Fit a regression model

```
Bwt.lm <- lm(bwt ~ smoke+gestation,data=Bwt)
summary(Bwt.lm)
#
plot(bwt ~ gestation, pch=19,
          col=ifelse(smoke==1,"red","blue"))

abline(-3.18, 0.45, col="blue",lwd=2)

abline(-3.18-8.37, 0.45, col="red",lwd=2)
```

Look at the residuals

```
plot(residuals(Bwt.lm) ~ fitted(Bwt.lm),
       col=ifelse(smoke==1,"red","blue"))
abline(h=c(-2,0,2)*16.25,lwd=2,lty=2,col="purple")

   # normal quantile plot
qqnorm(residuals(Bwt.lm),pch=19, col="limegreen")
qqline(residuals(Bwt.lm),col='gold',lwd=2)

   # fun with colors
qqnorm(residuals(Bwt.lm1),pch=19,
         col=rgb(1,.1,.1))
```

Logistic Regression

Logistic Regression is a regression model suitable for binary categories or more generally a binomial response variable. Since least squares is usually inappropriate for fitting with non-normally distributed error distributions, logistic regression models (and other **generalized linear models**) are fit by a method known as Maximum Likelihood (ML).

The Challenger O-Ring Data

```
ORings <- read.csv(
       "http://people.reed.edu/~jones/141/ORings.cs

ORings  # look at the data

with(ORings,plot(temp,Y,pch=19,col="red",
         xlim=c(30,85)))

OR.glm <- glm(cbind(Y, N - Y) ~ temp,
           family = binomial, data = ORings)
summary(OR.glm)
```

The Challenger O-Ring Data

```
   # replot as proportions
with(ORings,plot(temp,Y/N,pch=19,col="red",
                 xlim=c(30,85)))

TEMP <- 32:85

odds <- exp(6.89699-0.14212*TEMP)

Prob <- odds/(1+odds)

lines(TEMP,Prob,lwd=2,col="red")
```

Where's Waldo?

Is Waldo randomly distributed on the page as he travels through space and time?

```
Waldo <- read.table(
      "http://people.reed.edu/~jones/141/Waldo.dat"
                  header=TRUE)

attach(Waldo)  # never do this inside a function!

plot(WaldoX,WaldoY,pch=19,col="darkgreen")
```

Sample R Code for Waldo

Maps and Choropleths

See Math 141 Lab Notes

Homework

`http://bit.ly/qsr_sp14`

Please provide Feedback!