MSG500/MVE190
Linear Models - Lab 1

Rebecka Jörnsten
Mathematical Statistics
University of Gothenburg/Chalmers University of Technology

October 24, 2011

1 Getting Started

Directories and Histories: You can run R in different directories for different projects and save the results separately. A .RData file will be created at the end of each session (if you save it). You can also opt to save .Rhistory, i.e. all the commands you issued in the session.
HELP: You can always get information about a command by typing `help(command)` at the prompt. If you don't know the command name, try `help.search("phrase")`.
EDITORS: For the projects, and the labs, it is best to use an editor so you don't have to retype the commands. You can always cut-and-paste into the R command window, though this is not always possible in the windows environment on the lab computers. An alternative is to write a series of commands in your editor of choice, and save the file as "file.r". You can execute the commands in the file by writing `source("file.r")`. R has its own editor which you can access through the file menu. Personally, I prefer to use emacs or Winedt, but this is up to you.
PLOTS AND GRAPHICS: In R you open the graphics window with `x11()`. Under windows you don't have to open the graphics window, it will open automatically when you plot something. If you want to display multiple plots in the graphics window, use commands `par(mfrow=c(m,n))` or `split.screen(c(m,n))` to divide into m by n small graphics displays. Read the help files on these commands. You can save figures in any number of formats - just left-click on the figure to make it "active". The menus are now figure related and under "File" you can choose to save your figure.

2 Data Manipulation

Let's start by revisiting the data from lecture: the Television data. The data is available at the course homepage.
There are several variables: life span, people per doctor, people per TV, female life span and male life span.
Load this data set into R by calling the function `read.table`.

```
> TVdat <- read.table("TV.dat", sep = "\t")
> names(TVdat)

[1] "life"  "ppTV"  "ppDr"  "flife" "mlife"
```

You can access the variables in the data set (data frame) using the $ sign. Let's say you want to draw a histogram of the life span of the different countries:

```
> hist(TVdat$life)

> hist(TVdat$l)
```

You can access elements of a variable using indexing: here I draw a histogram of only the first 10 observations:

```
> hist(TVdat$life[1:10])
```

or without the first 10 observations

```
> hist(TVdat$life[-c(1:10)])
```

and here without the 2nd and 4th observation

```
> hist(TVdat$life[-c(2, 4)])
```

Play around with `hist()` (histograms), `mean()`, `median()`, `quantile()` (summary statistics) and subset of observations.

3 Regression

Let's try our hand at linear modeling. Following a similar approach to the demo in class, perform a simple regression analysis and diagnostic of female life span as a function of male life span. Do you need to transform the data? Are there any outliers? What's the R^2?

Create a second data set where you add a the country "Utopia" to the data set. You can choose the values for the variables for Utopia yourself. Rerun the regression analysis on life span or people per TV and see if you can detect Utopia as an outlier. How about "Dystopia"?

This is how to add a country to the data set:

```
> newvals <- c(a, b, c, d, e)
```

Create the new values first, you pick a-e to be the life span, people per Dr, etc for Utopia.

```
> TVdatExtra <- rbind(TVdat, newvals)
> row.names(TVdatExtra) <- c(row.names(TVdat), "Utopia")
```