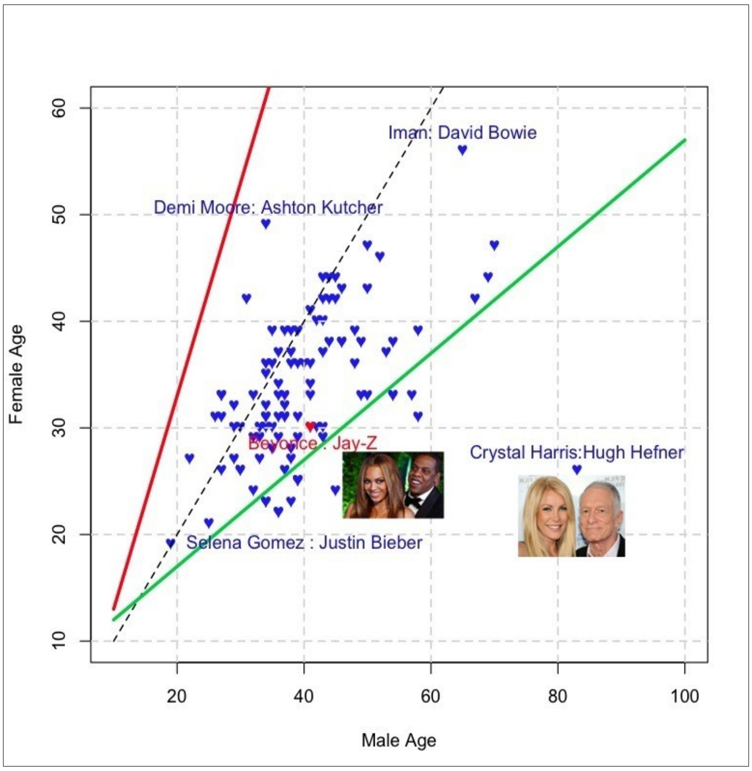
R USER-GUIDE

Tool for analysis and graphical representation of data

by C. Buja, C. Caporal, G. Tomat

Introduction

Have you ever heard of the *half your age plus seven rule*?
Well, if your answer is negative, first let us explain to you what it is about.
It is common belief - confront www.urbandictionary.com if you don't believe us! - that the age of your partner shouldn't be less than half your own, plus seven. No matter if you think love is timeless or ageless, this will be our starting point for the user guide; Our aim is not to give a proof or a normative analysis of the rule - obviously - but is to provide you with a clear, simple and easy way to remember R tutorial on **tools for analysis and graphical representation of data**.
Hence, to begin, we need a data set . Do celebrities respect the half your age plus seven decency criteria when choosing their beloved? To answer this question we considered zimbio.com's list of 100 hottest couples of 2011 - the data actually encloses 101 famous couples, due to some modifications -.
Here is what we found out:



Importing Data

The first step in order to build something like this is to import a data set. Once you have an excel type file, filled with all the data you need, save it in *.csv* format in order to let *R* import it.If you want to learn by doing, here there is the link where you can find the data used, copy, paste it and save it as .csv. https://docs.google.com/a/stud.unive.it/spreadsheet/pub?key=0AlqXhcl-KOFfdDlQc0htT29JaWFZU3M2UmpNNTJrdlE&single=true&gid=0&output=html

Open the *R* Program and type the command:

```
mydata <- read.table(file, header = FALSE, sep = "", …....)
```

where:

- *mydata* stands for the name you want to assign to your data set;

- `file` is the specific path of the *.csv* we previously saved; an example: "C:/UserMario/Unive Files/COMPTOOLS/celebrityDATA.csv";
- *header=FALSE* tells *R* that the first row in the data set is not the title of the columns;
- *sep* = *", "* for example indicates that the data inputted are separated by the comma.

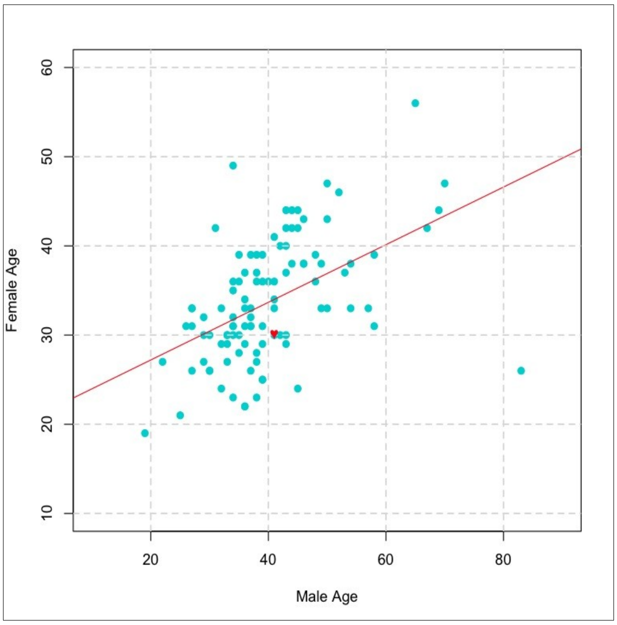
Once the data set is loaded successfully, you can give headers to the columns; it will make your life easier once you start working on the data. This is how you give headers:

```
	names(mydata) <- c("ColumnName1","name2","NameThree",...)
```

The structure command `str(mydata)` will give you details of the data set you have created, and it is also a useful tool to double-check very large data sets you don't want to load entirely on your console. If you want to know more about importing data you can visit http://virgo.unive.it/paolop/ct2012/user-guide_paolop.pdf.

Plotting Data

Start by telling *R* that the the only data set you will be working with will be `mydata`. This will save you from having to type things like `mydata$ColumnName1` when you want to refer to a specific column. To use a specific data set as default you need to input `attach(mydata)` in the console.
Now you can easily ask for statistical summaries of the variable just by typing `summary(ColumnName1), summary(name2), or summary(NameThree)`.
In our specific case we wanted a scatter plot of the V.I.P. couples' age. The command for plotting two variables is `plot(x, y, ...)`; We will explain this function using the original script we used in our worksheet as an example.



```
> plot(agem, agef, xlim=c(10,90),
ylim=c(10,60), xlab="Male Age",
	ylab="Female Age", pch=19,
col="cyan3")
```

Don't be scared. It is far less complicated than what it seems:

- agem and agef refer to the variables we want to represent in the scatter plot, the first on the x and the second on the y axis.
- xlim and ylim refers to the variable intervals you want to plot. If you want to reverse an axis, use xlim or ylim of the form c(hi, lo).

- xlab and ylab are the labels for the x and y axis that will appear an the scatter plot.
- pch is the input that specifies what symbol will be used to plot the data. There is a little trick to view all the symbols available on R. You just need to create a dummy data frame, for example

`df <- data.frame(x=1:50, y=1:50)`, and then plot it using 50 pch in sequence like this plot(df,pch=1:50)
Then all you need to do is change the interval of `pch` from 0-50 to 50-100, 100-150, and so on to view all the symbols available.

- `col` stands for the color you want to use. Here is link to a complete table of all colors available in R. Have fun! http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
- NOTE: if you are interested in different kinds of plot such as *stair steps* or *lines and points* there are specific commands to do so: You must add `type = “...“` to the formula, filled with one of the following letters

"p" for **p**oints,
" l " for **l**ines,
"b" for **b**oth,
"c" for the lines part alone of "b",
"o" for both ‘**o**verplotted’,
"h" for ‘**h**istogram’ like (or ‘high-density’) vertical lines,
"s" for stair **s**teps

```
> grid(col="lightgrey",lty=2,lwd=1.5)
```

The light gray grid you can see on the background can be simply obtained by the `grid()` command adding inside the brackets the desired color, pattern, and width; it is a low level command so there is no need to specify add=T while typing the command.

```
> abline(coef=coef(best.fit),col="red")
```

The `abline()` command is used to plot the theoretical line where the points of a linear model should lie. In the case above our model is `best.fit`, previously described as:

```
> best.fit<-lm(agef ~ agem)
```

The command `lm` is used to fit linear models, and it can be used to carry out regression, single analysis of variance and analysis of covariance. When you want to do that, don't forget to use the ~ (”*tilde*”), which is necessary to run the model. By plotting the model and it's corresponding `abline` we can notice if the variables are coherent or not with the reference line (abline). In this case you can see that there's a sufficient positive linear relationship between the two variables.

Going back to the initial plot you can see that there are 3 lines of different colors. The red and green ones refer respectively to the upper and lower bounds of socially acceptable region, also called the “*safe area*”, inside which, those couples are the Dating Rule followers. For each 1/2x +7, there is a corresponding 2(x-7) boundary to the initial dating equation. You can plot them by typing first the function, giving it a name and then type
`curve(expression,add = TRUE, type = "l", …).`

The black dashed line, you can observed in between the two colored lines, has a 1 to 1 relation and it has been plotted to better understand the path of the celebrities couples age relation. By making an easy and quick data analysis, you can see that the highest concentration is met in the bottom half "*safe area*", where men prefer to be accompanied by younger women.
Further more we can underline a specific point by, for example, using the function `points()` to draw a specific symbol in a specific coordinate, such as:

```
> points(41,30,pch="♥",col="red")
```

The one considered in the top plot has been highlighted because it represents the Hottest couple over the all 101 of the list.

You can also notice that there are few couples which lie outside the boundaries; these are called outliers and it is very interesting discover who those famous socially unacceptable couples are. In the first picture it is possible to notice the further outlier, composed by Playboy magazine's father Hugh Hefner and the model Crystal Harris. But are you able to find out the remaining ones?
What you are going to need to solve this problem, is a specific command called `identify()` ,in which you have to specify the variables of male age and female age. To show the name of the couples when identifying , you would have to type `labels=seq(namef,namem)`. Once you have finished identifying, move the mouse arrow to the upper left Stop bar in "pc" and click with a "two fingers tap" on the graph in Mac (alternatively you can simply close the graph window, R will return an error, just ignore it), and then you can continue computing with R program.

```
> identify(agem,agef,labels=seq(namef,namem),col=4)
```

Errors to be avoided

R is really a powerful program which makes your analysis work much simpler from the computational to the graphical part, but of course it has its weaknesses. Sometimes, it won't show you the error message when it should.
You will have to pay attention to the type of the command: if they are high or low level, if so, don't forget to type also add=TRUE if you wish to obtain the result on the same plot.
A frequent error students often do is to forget to attach the data set to the search path of the program, allowing *R* to take the objects needed by simply giving their names. This is simply done by `attach(mydata)`. It is convenient to remove the objects from the search path at the end of the work by detach the same, `detach(mydata)`.

Suggestions

This simple analysis with couples age could be easily reproduced taking into account age and worthiness of the two partners, or could be also interesting to see how non famous couples would behave to this Dating Rule. If you are interested in these kind of social data analysis, here are some helpful links. Enjoy!
http://www.celebritynetworth.com/