Birkbeck
UNIVERSITY OF LONDON

Big Data Analytics

Session 1b

Introduction to R

What is R?

Birkbeck
UNIVERSITY OF LONDON

- A suite of operators for calculations on arrays, in particular matrices,
- A large, coherent, integrated collection of intermediate tools for data analysis,
- Graphical facilities for data analysis and display either on-screen or on hardcopy, and
- A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.
- Free (as in beer *and* speech), open-source software

Installing and running R

- How to get R:
  - http://www.r-project.org/
  - Google: "R"
  - Windows, Linux, Mac OS X, source
  - In this lab:
    - user@ba1:~$> R [terminal only]
    - user@ba1:~$> R –g Tk & [application window]

- Files for this tutorial:
  - http://web.mit.edu/tkp/www/R/R_Tutorial_Data.txt
  - http://web.mit.edu/tkp/www/R/R_Tutorial_Inputs.txt

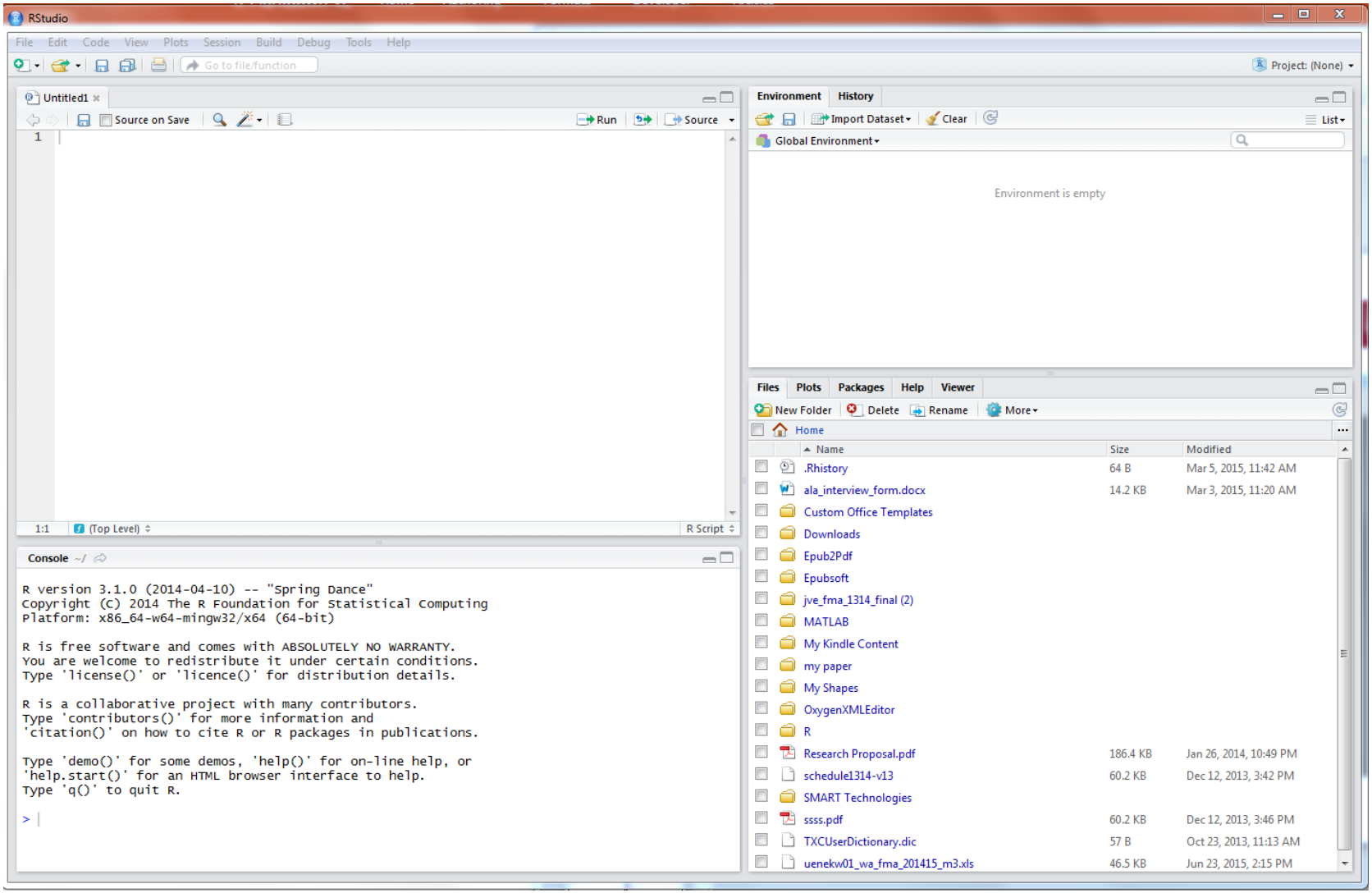
RStudio

Birkbeck
UNIVERSITY OF LONDON

- **RStudio** is an integrated development environment (IDE) for R.
  - Free and open source
  - Available for MS Windows, Mac OS X and Linux
  - RStudio Desktop and Server

- Important features
  - code completion
  - execute from source
  - searchable history
  - support for authoring Sweave documents (R Markdown)

RStudio


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Project: (None)
Untitled1
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Console ~/
R version 3.1.0 (2014-04-10) -- "Spring Dance"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment History
Import Dataset
Clear
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder
Delete
Rename
More
Home
Name
Size
Modified
.Rhistory 64 B Mar 5, 2015, 11:42 AM
ala_interview_form.docx 14.2 KB Mar 3, 2015, 11:20 AM
Custom Office Templates
Downloads
Epub2Pdf
Epubsoft
jve_fma_1314_final (2)
MATLAB
My Kindle Content
my paper
My Shapes
OxygenXMLEditor
R
Research Proposal.pdf 186.4 KB Jan 26, 2014, 10:49 PM
schedule1314-v13 60.2 KB Dec 12, 2013, 3:42 PM
SMART Technologies
ssss.pdf 60.2 KB Dec 12, 2013, 3:46 PM
TXCUserDictionary.dic 57 B Oct 23, 2013, 11:13 AM
uenekw01_wa_fma_201415_m3.xls 46.5 KB Jun 23, 2015, 2:15 PM

Outline

- R language
  - Basic commands
  - Vectors
  - Matrices
  - Reading data from files
  - Graphical and numerical summaries
  - Selecting subsets of data
  - Getting help
- RStudio tutorial

Basic commands

Math:

```
> 1 + 1
[1] 2

> 1 + 1 * 7
[1] 8

> (1 + 1) * 7
[1] 14
```

Variables:

```
> x <- 1
> x
[1] 1
> y = 2
> y
[1] 2
> 3 -> z
> z
[1] 3
> (x + y) * z
[1] 9
```

In RStudio

- In console
  - type in code
  - retrieve past code by 'up arrow'
- In source code section
  - new, save, search
  - run current line
  - select and run
  - re-run previous code region
  - comment and uncomment lines (#)

Vectors

```
> x <- c(0,1,2,3,4)
> x
[1] 0 1 2 3 4

> y <- seq(1,5)     #func seq(): create a sequence of numbers
> y                  # seq(0,1,length=10)
[1] 1 2 3 4 5

> y <- 1:6           #shorthand for seq(1,6)
> y
[1] 1 2 3 4 5 6

> z <- 1:50
> z
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
[16] 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
[31] 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
[46] 46 47 48 49 50
```

In RStudio:
Notice the value change in the global environment

Vector operations

Birkbeck UNIVERSITY OF LONDON

Math on vectors

```
> x <- c(0,1,2,3,4)
> y <- 1:5
> z <- 1:50
> x + y
[1] 1 3 5 7 9
> x * y
[1]  0  2  6 12 20
> x * z
 [1]   0   2   6  12  20   0   7  16  27  40   0
[12]  12  26  42  60   0  17  36  57  80   0  22
[23]  46  72 100   0  27  56  87 120   0  32  66
[34] 102 140   0  37  76 117 160   0  42  86 132
[45] 180   0  47  96 147 200
```

Length of a vector:

```
> length(x)
[1] 3
> length(y)
[1] 3
> x+y
[1]  2 10  5
```

Basic Commands

- Getting previous commands
  - Hitting up arrow
- Comments
  - #
- On quitting R (in plain R)
  - q()
  - savehistory()
  - loadhistory()

- List and remove objects

```
> ls()
[1] "x" "y"
> rm(x,y)
> ls()
character(0)
```

- Remove all objects

```
> rm(list=ls())
```

- Getting help on functions

```
> help(funcname)
> ?funcname
```

Exercises

- Create a vector x
  - starting from 5.3
  - ending at 8.00
  - length is 10
- Create another vector y
  - starting from 3.5
  - ending at or less than 7.9
  - each term is 0.4 more than the previous one

  y= 3.5, 3.9, 4.3, …
- Add x and y
  - did you get a warning message? did your RStudio crash?
- Increase all terms in x by 1

Matrix

- Creating a matrix, order by columns by default

```
> x=matrix(data=c(1,2,3,4), nrow=2, ncol=2)
> x
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

  - Omit nrow, ncol

```
> x=matrix(c(1,2,3,4),2,2)
```

  - byrow

```
> matrix(c(1,2,3,4),2,2,byrow=TRUE)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

- Matrix operations

```
> sqrt(x)
     [,1] [,2]
[1,] 1.00 1.73
[2,] 1.41 2.00
> x^2
     [,1] [,2]
[1,]    1    9
[2,]    4   16
```

- Dimension of Matrix

```
> dim(A)
[1] 2  2
```

Num.rows Num col.

Indexing Data

Given

```
> A=matrix(1:16,4,4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

Select one element

```
> A[2,3]
[1] 10
```

Select multiple rows and columns

```
> A[c(1,3),c(2,4)]
     [,1] [,2]
[1,]    5   13
[2,]    7   15
> A[1:3,2:4]
     [,1] [,2] [,3]
[1,]    5    9   13
[2,]    6   10   14
[3,]    7   11   15
> A[1:2,]
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
> A[,1:2]
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```

Indexing Data

Given

```
> A=matrix(1:16,4,4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
```

Negative index

```
> A[-c(1,3),]
     [,1] [,2] [,3] [,4]
[1,]    2    6   10   14
[2,]    4    8   12   16
> A[-c(1,3),-c(1,3,4)]
[1] 6 8
```

No index

```
> A[1,]
[1]  1  5  9 13
```

Exercises

- How to create this matrix?

```
> B
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]   11   13   15   17   19
[3,]   21   23   25   27   29
[4,]   31   33   35   37   39
```

- What could the commands be to have

```
     [,1] [,2]
[1,]   13   15
[2,]   33   35
```

  - try positive, negative indices

- How to create this matrix?

```
> C
     [,1] [,2] [,3]
[1,]    1    3   13
[2,]    1    5   21
[3,]    2    8   34
```

- What could the commands be to have
  - [1] 1
  - [1] 1 1
  - [1] 1 1 2

Outline

- R language
  - Basic commands
  - Vectors
  - Matrices
  - Reading data from files
  - Graphical and numerical summaries
  - Selecting subsets of data
  - Getting help
- RStudio tutorial -- R Markdown

Introduction to R Markdown

- R Markdown is a file format for making dynamic documents with R
  - Source : R Markdown file (.Rmd)
    - text + chunks of embedded R code
  - Target: PDF, HTML or MS Word
    - text + chunks of embedded R code (optional) + result
  - See an example
- You may 'compile notebook from R script' without any text.
- More info:
  - http://rmarkdown.rstudio.com/authoring_basics.html

Outline

- R language
  - Basic commands
  - Vectors
  - Matrices
  - Reading data from files
  - Graphical and numerical summaries
  - Selecting subsets of data
  - Getting help
- RStudio tutorial -- R Markdown

These will be talked about in the following sessions.

Random normal distribution

- `rnorm(n,mean=0,sd=1)` generates a vector of random normal variables
  - n: sample size
  - default mean=0 and sd=1
  - each time different

```
> x=rnorm(50)
> y=x+rnorm(50,mean=50,sd=.1)
> cor(x,y)
[1] 0.995
```

- `cor(), mean(), var(), sd()`

- `set.seed(`**m**`)` reproduces the exact same set of random numbers as long as the arbitrary integer argument **m** stays the same.

```
> set.seed(3)
> y=rnorm(100)
> mean(y)
[1] 0.0110
> var(y)
[1] 0.7329
> sqrt(var(y))
[1] 0.8561
> sd(y)
[1] 0.8561
```

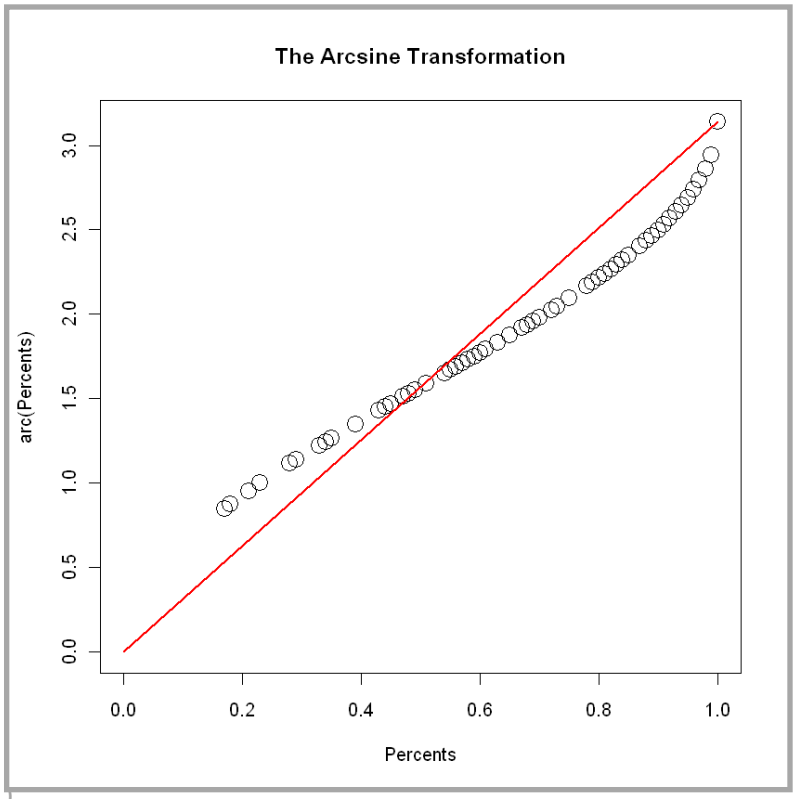
```
> set.seed(1303)
> rnorm(50)
[1] -1.1440  1.3421  2.1854  0.5364  0.0632  0.5022 -0.0004
. . .
```

Functions

```
> arc <- function(x) 2*asin(sqrt(x))
> arc(0.5)
[1] 1.570796
> x <- c(0,1,2,3,4)
> x <- x / 10
> arc(x)
[1] 0.0000000 0.6435011 0.9272952
[4] 1.1592795 1.3694384
```

```
> g <- function(x,y) y/(1+x^2)
> g(0.5,30)
[1] 24
```

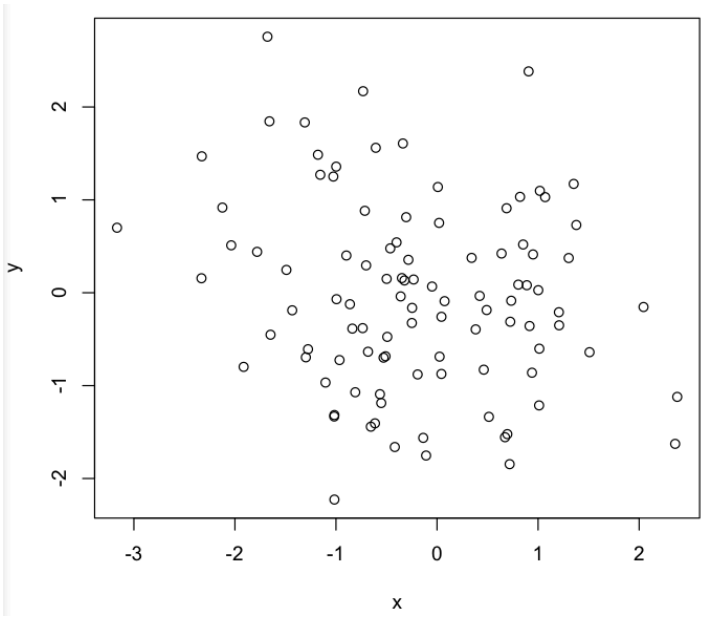
```
> plot(arc(Percents)~Percents,
+ pch=21,cex=2,xlim=c(0,1),ylim=c(0,pi),
+ main="The Arcsine Transformation")
> lines(c(0,1),c(0,pi),col="red",lwd=2)
```

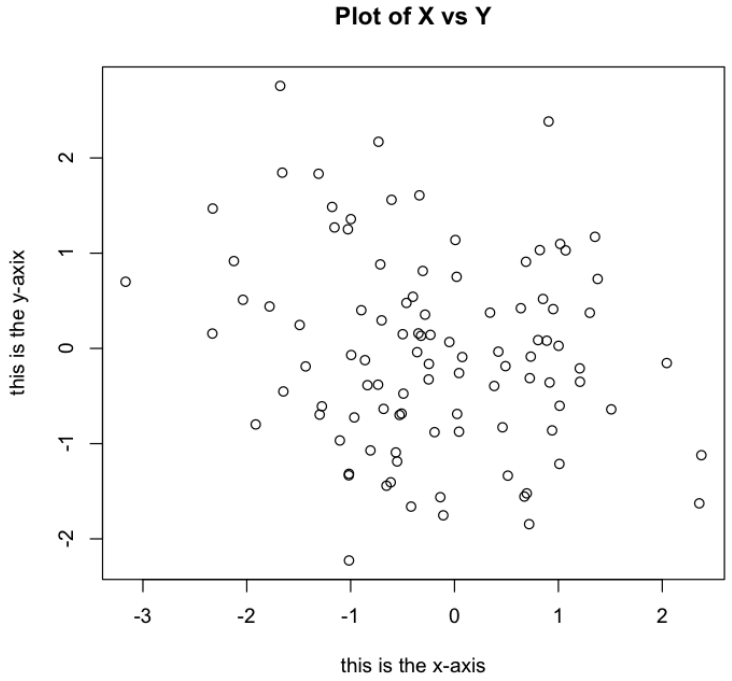


Basic Graphics

- `plot()`

```
> x=rnorm(100)
> y=rnorm(100)
> plot(x,y)
> plot(x,y,xlab="this is the x-axis",ylab="this is the y-axis",
      main="Plot of X vs Y")
```





Basic Graphics

- Save output to a file
  - `pdf()` or `jpeg()`

```
> pdf("Figure.pdf")
> plot(x,y,col="green")
> dev.off()
null device
          1
```

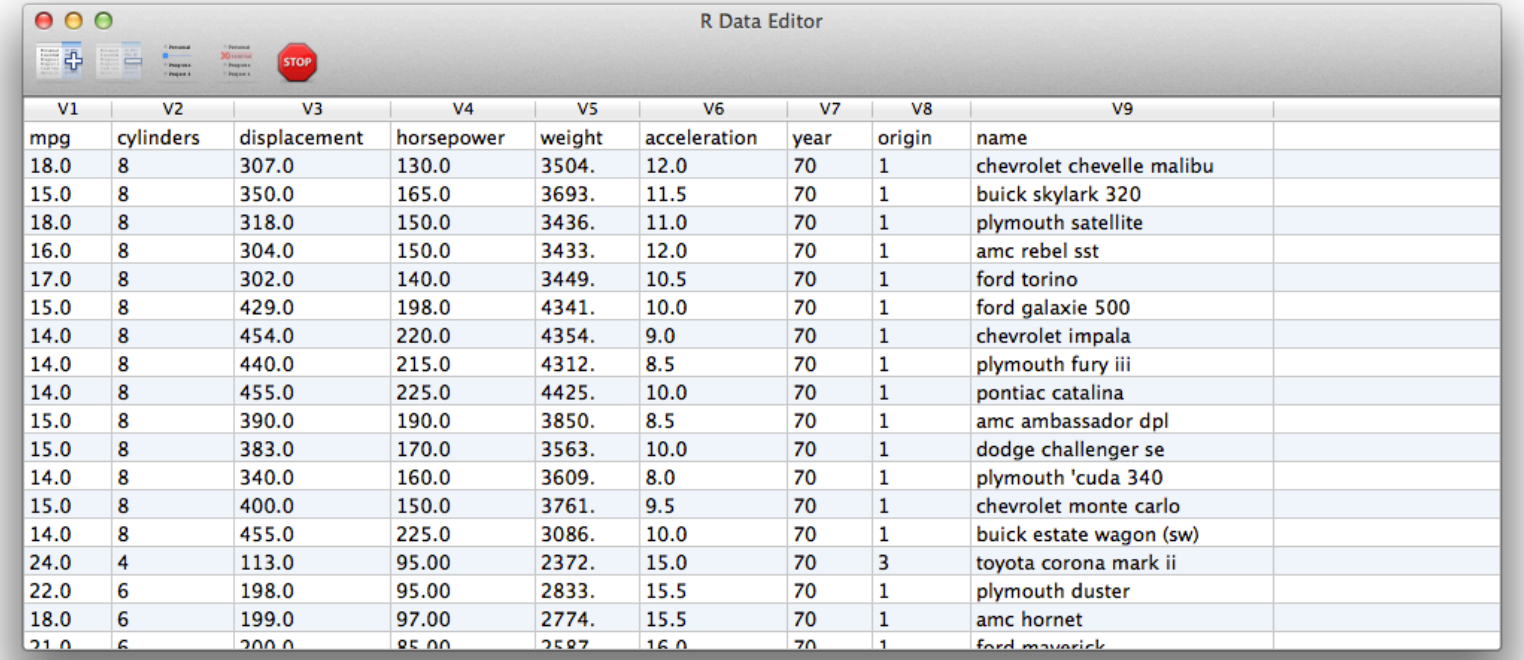
  - `dev.off()`
    - indicates to R that we are done creating the plot

  - Alternatively, we can simply copy the plot window and paste it into an appropriate file type, such as a Word document.

Reading dataset from files

- `read.table()` loads data file into R and stores it as an object in a format as a data frame.
- use `fix()` to view data in a spreadsheet like window. The window must be closed before further R commands can be entered.

```
> Auto = read.table("/User/…/data/Auto.data")
> fix(Auto)
```



V1	V2	V3	V4	V5	V6	V7	V8	V9
mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
18.0	8	307.0	130.0	3504.	12.0	70	1	chevrolet chevelle malibu
15.0	8	350.0	165.0	3693.	11.5	70	1	buick skylark 320
18.0	8	318.0	150.0	3436.	11.0	70	1	plymouth satellite
16.0	8	304.0	150.0	3433.	12.0	70	1	amc rebel sst
17.0	8	302.0	140.0	3449.	10.5	70	1	ford torino
15.0	8	429.0	198.0	4341.	10.0	70	1	ford galaxie 500
14.0	8	454.0	220.0	4354.	9.0	70	1	chevrolet impala
14.0	8	440.0	215.0	4312.	8.5	70	1	plymouth fury iii
14.0	8	455.0	225.0	4425.	10.0	70	1	pontiac catalina
15.0	8	390.0	190.0	3850.	8.5	70	1	amc ambassador dpl
15.0	8	383.0	170.0	3563.	10.0	70	1	dodge challenger se
14.0	8	340.0	160.0	3609.	8.0	70	1	plymouth 'cuda 340
15.0	8	400.0	150.0	3761.	9.5	70	1	chevrolet monte carlo
14.0	8	455.0	225.0	3086.	10.0	70	1	buick estate wagon (sw)
24.0	4	113.0	95.00	2372.	15.0	70	3	toyota corona mark ii
22.0	6	198.0	95.00	2833.	15.5	70	1	plymouth duster
18.0	6	199.0	97.00	2774.	15.5	70	1	amc hornet
21.0	6	200.0	85.00	2587.	16.0	70	1	ford maverick

Reading dataset from files

- **`header = T`** tells R that the first line of the file contains variable names

- **`na.string = "?"`** tells R that a particular character or a set of particular characters (in this case **?**) should be treated as a missing element. view data in a spreadsheet like window. The window must be closed before further R commands can be entered.

```
> Auto = read.table("/User/…/data/Auto.data",header=T,na.string="?")
> fix(Auto)
```

- Use `read.csv()` to load common format data (e.g., excel data). `csv` file stands for `common separated value` file.

```
> Auto=read.csv("Auto.csv",header=T,na.strings="?")
> fix(Auto)
> dim(Auto)
[1] 397  9
> Auto[1:4,]
```

Examining datasets

Birkbeck UNIVERSITY OF LONDON

- Remove rows with missing values

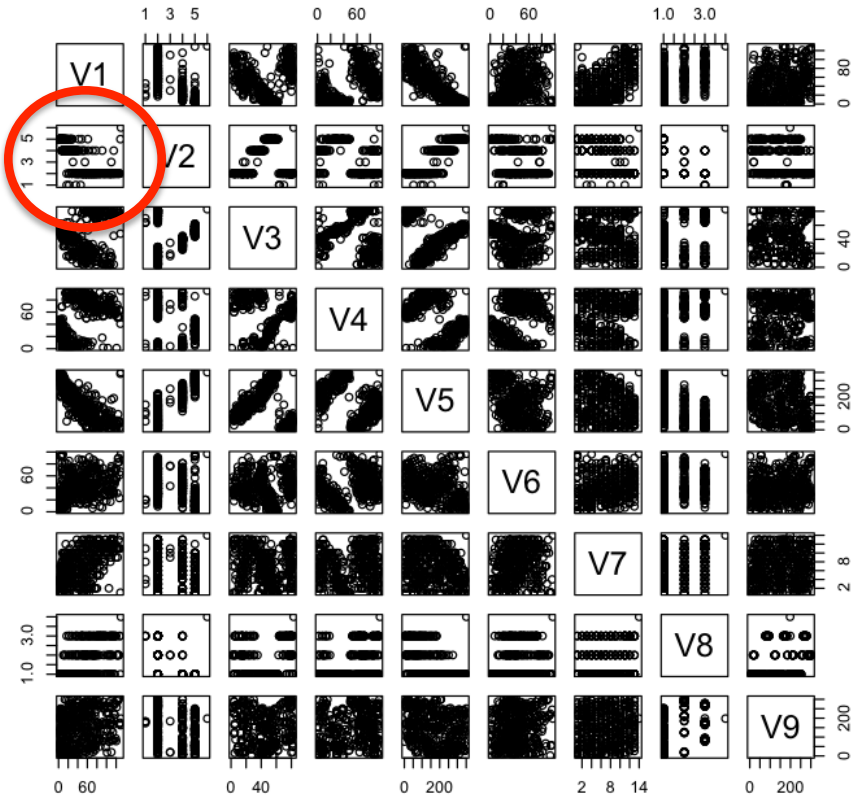
```
> Auto=na.omit(Auto)
> dim(Auto)
[1] 392   9
```

- Check variable names

```
> names(Auto)
[1] "mpg"          "cylinders"    "displacement" "horsepower"
[5] "weight"       "acceleration" "year"         "origin"
[9] "name"
```

Additional graphical commands

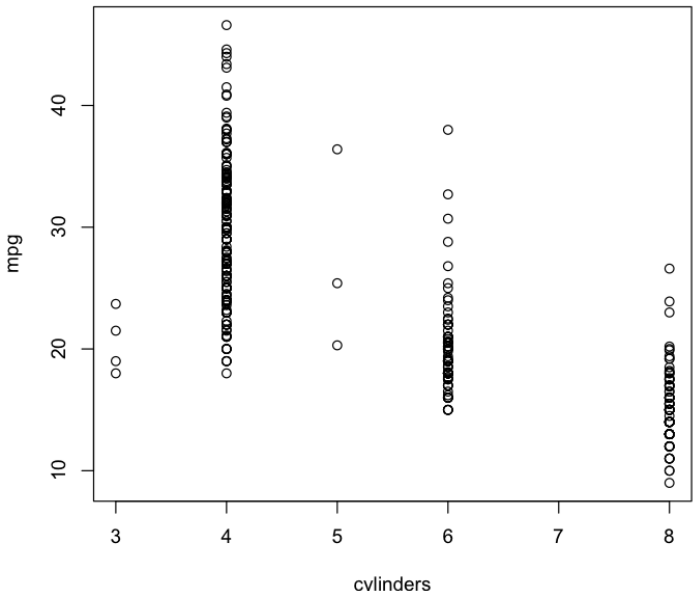
- `plot(Auto)` creates a scatter plot matrix for every pair of variables
- Alternatively, `pairs(Auto)` could create the same scatter plot matrix
- `plot(Auto$cylinder, Auto$mpg)`
- `attach(Auto)`

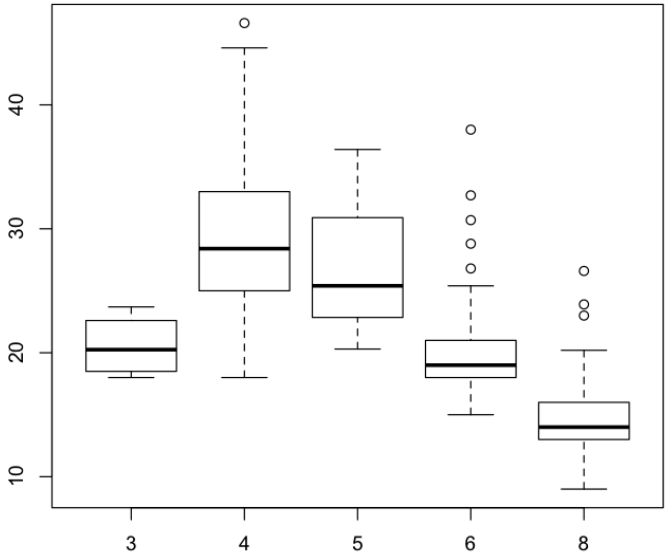



```
> plot(Auto$cylinders , Auto$mpg)
> attach(Auto)
> plot(cylinders , mpg)
```

Additional graphical commands

- `cylinders` variable is stored as a numeric (quantitative) vector, but can be treated as a categorical (qualitative) variable
- Use `as.factor()` to do that
- As a result, boxplot will automatically produced by the plot function






```
> plot(cylinders,mpg)
> cylinders = as.factor(cylinders)
> plot(cylinders,mpg)
```

Additional graphical commands

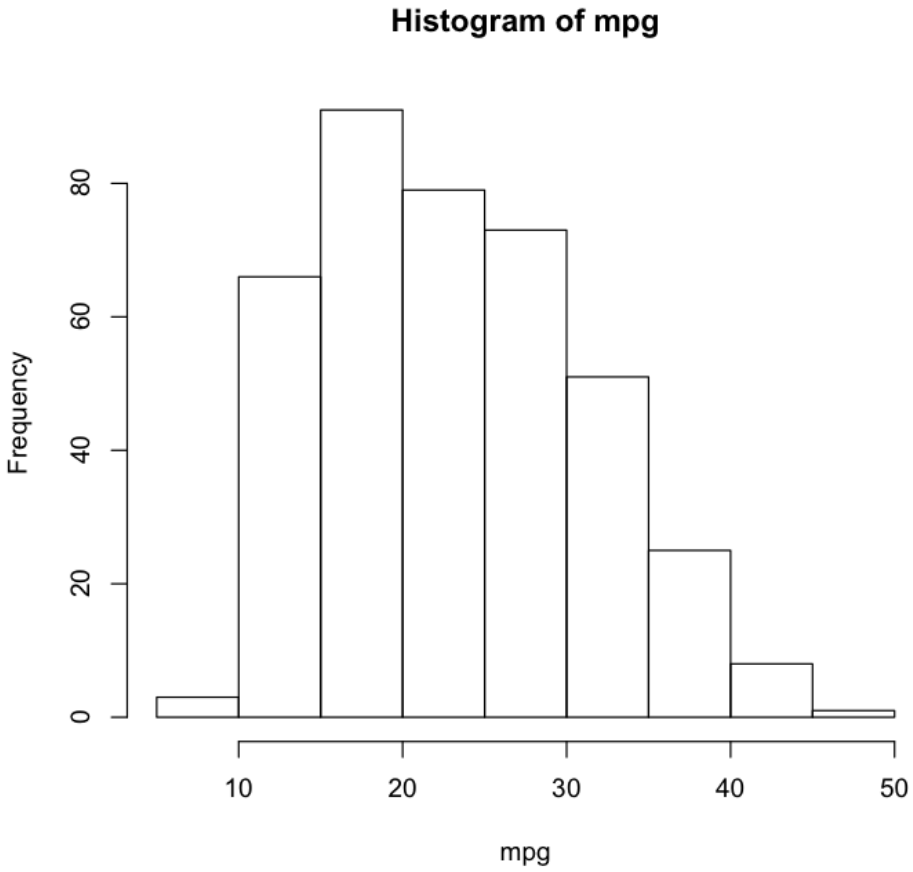
- More options

```
> plot(cylinders, mpg)
> plot(cylinders, mpg, col="red")
> plot(cylinders, mpg, col="red", varwidth=T)
> plot(cylinders, mpg, col="red", varwidth=T,horizontal=T)
> plot(cylinders, mpg, col="red", varwidth=T, xlab="cylinders",
    ylab="MPG")
```

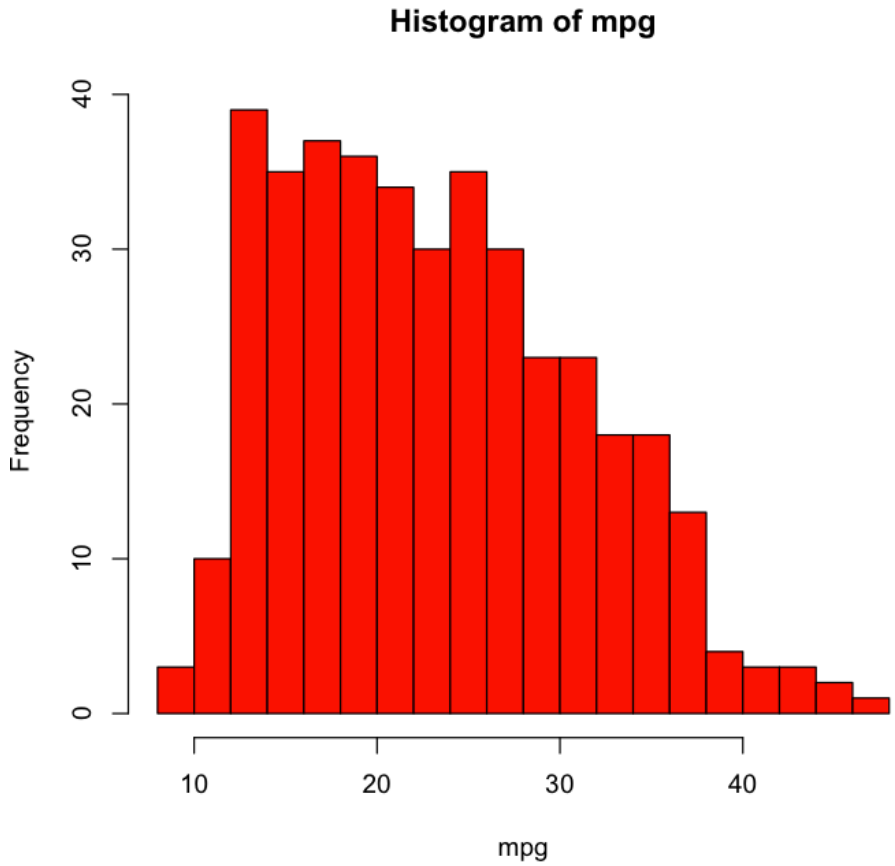
- Try yourselves!

Plot histograms

```
> hist(mpg)
```




```
> hist(mpg,col=2,breaks=15)
```



Select subsets of data

Birkbeck
UNIVERSITY OF LONDON

```
> Auto$cylinders
  [1] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 4 6 6 6 4 4 4 4 4 4 6 8 8 8 8 4 4 4 4 6 6 6 6 6 8 8 8 8 8 8 8 6
 [47] 4 6 6 4 4 4 4 4 4 4 4 4 4 4 4 4 4 8 8 8 8 8 8 8 8 8 3 8 8 8 8 4 4 4 4 4 4 4 4 4 8 8 8 8 8 8 8
 [93] 8 8 8 8 8 8 6 6 6 6 6 4 8 8 8 8 6 4 4 4 3 4 6 4 8 8 4 4 4 4 8 4 6 8 6 6 6 6 4 4 4 4 6 6 6 8 8
[139] 8 8 8 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 6 8 8 8 8 6 6 6 6 6 8 8 4 4 6 4 4 4 4 6 4 6 4 4 4 4 4 4 4
[185] 4 4 4 8 8 8 8 6 6 6 6 4 4 4 4 6 6 6 6 4 4 4 4 4 4 8 4 6 6 8 8 8 8 4 4 4 4 4 4 8 8 8 8 6 6 6 6 8
[231] 8 8 8 4 4 4 4 4 4 4 4 6 4 3 4 4 4 4 4 8 8 8 6 6 6 4 6 6 6 6 6 6 8 6 8 8 4 4 4 4 4 4 4 4 4 5 6
[277] 4 6 4 4 6 6 4 6 6 8 8 8 8 8 8 8 8 4 4 4 4 5 8 4 8 4 4 4 4 4 4 6 6 4 4 4 4 4 4 4 4 6 4 4 4 4 4
[323] 4 4 4 4 4 4 5 4 4 4 4 4 4 6 3 4 4 4 4 4 4 4 6 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 6 8 6 6 4
[369] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 4 6 4 4 4 4 4 4 4 4
> cylinders[mpg=="15"]
 [1] 8 8 8 8 8 8 8 8 8 8 6 6 6 8 6 8
> cylinders[mpg!="15"]
  [1] 8 8 8 8 8 8 8 8 8 4 6 6 6 4 4 4 4 4 4 6 8 8 8 8 4 4 4 4 6 6 6 6 6 8 8 8 8 8 8 8 6 4 6 6 4 4
 [47] 4 4 4 4 4 4 4 4 4 4 4 4 8 8 8 8 8 8 8 8 8 3 8 8 8 8 4 4 4 4 4 4 4 4 4 4 8 8 8 8 8 8 8 8 8 8 6 6 6
 [93] 6 6 4 8 8 8 8 8 6 4 4 4 3 4 6 4 8 4 4 4 4 4 6 8 6 6 6 4 4 4 4 6 6 6 8 8 8 8 8 4 4 4 4 4 4 4 4 4
[139] 4 4 4 6 6 8 8 8 8 6 6 6 6 8 8 4 4 6 4 4 4 4 6 4 6 4 4 4 4 4 4 4 4 4 4 4 8 8 8 8 6 6 6 6 4 4 4 4
[185] 6 6 6 6 4 4 4 4 4 4 8 4 6 6 8 8 8 8 4 4 4 4 4 4 8 8 8 6 6 6 6 8 8 8 8 4 4 4 4 4 4 4 4 6 4 3 4 4
[231] 4 4 4 8 8 8 6 6 6 4 6 6 6 6 6 6 8 6 8 8 4 4 4 4 4 4 4 4 5 6 4 6 4 4 6 6 4 6 6 8 8 8 8 8 8 8 8
[277] 8 4 4 4 4 4 5 8 4 8 4 4 4 4 4 4 6 6 4 4 4 4 4 4 4 4 6 4 4 4 4 4 4 4 4 4 4 5 4 4 4 4 4 4 6 3 4 4 4
[323] 4 4 4 6 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6 6 8 6 6 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
[369] 4 6 6 4 6 4 4 4 4 4 4 4 4
> cylinders[mpg<="15"&year=="71"]
[1] 8 8 8 8 8 8 8
```

Showing variable summary

```
> summary(Auto)
      mpg          cylinders      displacement     horsepower        weight      acceleration
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00
 1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.0   1st Qu.: 75.0   1st Qu.:2223   1st Qu.:13.80
 Median :23.00   Median :4.000   Median :146.0   Median : 93.5   Median :2800   Median :15.50
 Mean   :23.52   Mean   :5.458   Mean   :193.5   Mean   :104.5   Mean   :2970   Mean   :15.56
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd Qu.:3609   3rd Qu.:17.10
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80
                                                 NA's   :5
      year           origin                      name
 Min.   :70.00   Min.   :1.000   ford pinto      :  6
 1st Qu.:73.00   1st Qu.:1.000   amc matador     :  5
 Median :76.00   Median :1.000   ford maverick   :  5
 Mean   :75.99   Mean   :1.574   toyota corolla  :  5
 3rd Qu.:79.00   3rd Qu.:2.000   amc gremlin     :  4
 Max.   :82.00   Max.   :3.000   amc hornet      :  4
                                 (Other)         :368

> summary(mpg)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   9.00   17.50   23.00   23.52   29.00   46.60
```

Quantitative variables: show quartiles and mean

Qualitative variables: show the number of observations that fall in each category

Getting help

`> help(funcname)`	Info on certain function
`> ?funcname`	
`> help.search("topic")`	Info on certain topic
`> ??"topic"`	
`> library()`	List all libraries
`> data()`	List all datasets

```
> help(c)
> ?fix
>
> help.search("standard deviation")
> ??"correlation"
```