Introduction to R

A Language and Environment for Statistical Computing, Graphics

&

Bioinformatics

Why R?

- FREE to download
  - for teaching, research, development
- Powerful language and environment
  - (vectors/matrices like Matlab)
- Language and Applications Developed and Supported by Leading Researchers
- Exchange platform for packages/functions/data

… and R is the base for

- BIOCONDUCTOR
  - A collection of Bioinformatics tools such
    - Gene Ontologies
    - Genomics object classes: e.g. microarrray expression data sets
    - Analysis and visualization methods
  - more about *Bioconductor* later.

The Comprehensive R Archive Network - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search

Address http://cran.r-project.org/

Go Links

Google R cran Search Web PageRank 259 blocked AutoFill Options R cran

Downloading R …

R-2.0.1 for Windows

This directory contains a binary distribution of R-2.0.1 to run on Windows 95, 98, ME, NT4.0, 2000 and XP on Intel/clone chips.

Patches to this release are incorporated in the r-patched release.

A build of the development version (which will eventually become the next major release of R) is available in the r-devel release.

In this directory:

README.rw2001	Installation and other instructions.
CHANGES	New features of this Windows version.
NEWS	New features of all versions.
rw2001.exe	Setup program (about 23 megabytes). Please download this from a mirror near you. This corresponds to the file named **SetupR.exe** in pre-1.6.0 releases.
mini	Collection of diskette-sized files. Again, please download from a mirror.
old	The previous release.
md5sum.txt	md5sum output for the setup program. A Windows GUI version of md5sum is available at http://www.md5summer.org/; a Windows command line version is available at http://www.etree.org/md5com.html.

Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information, including upgrade advice.

Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.htm.

CRAN
Mirrors
What's new?
Search

About R
R Homepage

Software
R Sources
R Binaries
Packages
Other

Documentation
Manual
FAQs
Contributed
Newsletter

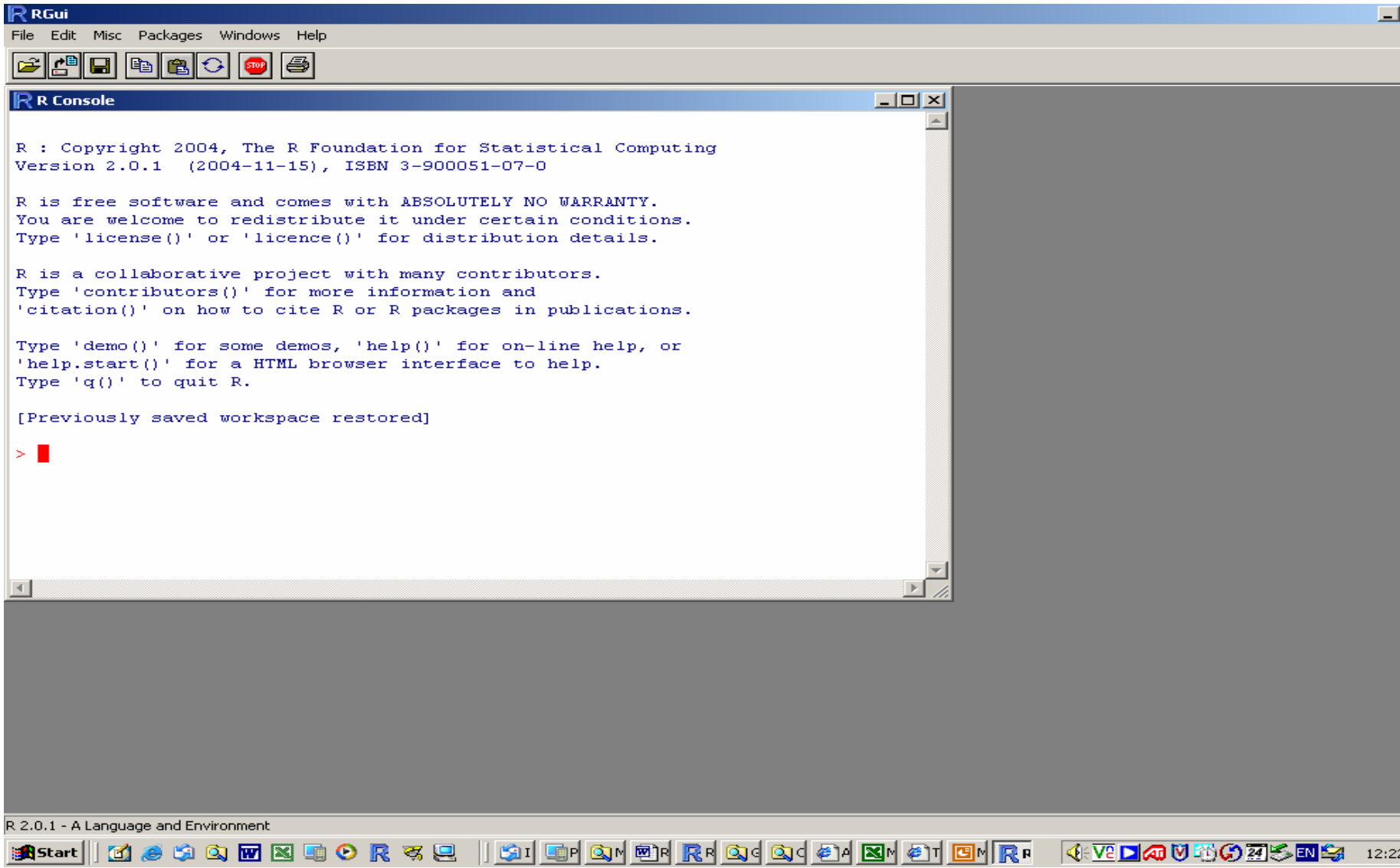
http://cran.r-project.org/bin/windows/base/rw2001.exe

Internet

Start 11:59 AM

The R Icon on my desktop

Opening R – the Console



So far …

- Depending on connection quality, you are "up and running" in 10 to 20 minutes

- Disadvantages:
  - No guarantees for programs etc (FDA!)
  - Too many packages/variations to choose from
    - No-one decides for you what is best algorithm
  - Some help files a little cryptic

Language basics - Data

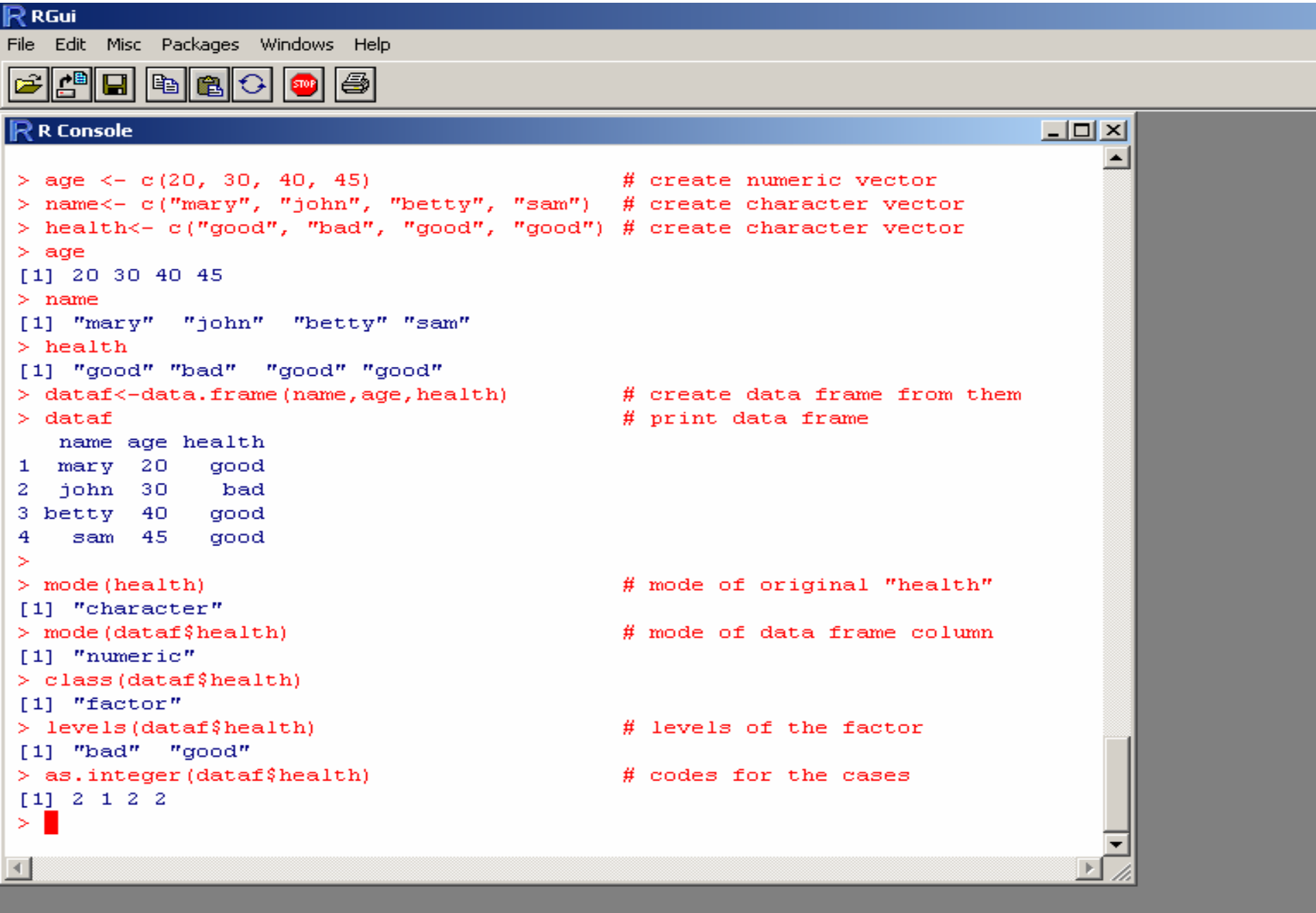
- "Data frame"
  - Standard statistical organization of data
  - Rows (records) are Cases (observations),
  - Columns are Variables ("features")
- Vectors, Matrices, Arrays
- Modes
  - Numeric – elements are numbers
  - Character – elements are strings
  - Logical – elements are (TRUE/FALSE)

Data Frame

	V1	V2	V3	V4	…
1	5.2	**“green”**	1	“yes”	…
2	3.4	**“blue”**	2	“yes”	…
3	5.1	**“brown”**	3	“no”	…
4	4.9	**“green”**	2	“yes”	…
5	2.6	**“blue”**	1	“no”	…
…	…	**…**	…	…	…

Object Orientation

- Everything is an Object
- Objects belong to Classes
- The Class determines how the Object is:
  - printed
  - plotted
  - analyzed
- E.g. Class “factor”
  - Treated like a factor or categorical variable by statistical functions


RGui
File Edit Misc Packages Windows Help
R Console
> age <- c(20, 30, 40, 45) # create numeric vector
> name<- c("mary", "john", "betty", "sam") # create character vector
> health<- c("good", "bad", "good", "good") # create character vector
> age
[1] 20 30 40 45
> name
[1] "mary" "john" "betty" "sam"
> health
[1] "good" "bad" "good" "good"
> dataf<-data.frame(name,age,health) # create data frame from them
> dataf # print data frame
name age health
1 mary 20 good
2 john 30 bad
3 betty 40 good
4 sam 45 good
>
> mode(health) # mode of original "health"
[1] "character"
> mode(dataf$health) # mode of data frame column
[1] "numeric"
> class(dataf$health)
[1] "factor"
> levels(dataf$health) # levels of the factor
[1] "bad" "good"
> as.integer(dataf$health) # codes for the cases
[1] 2 1 2 2
>

Extracting/Subscripting

Vector x: = c(5,6,7,8,9)

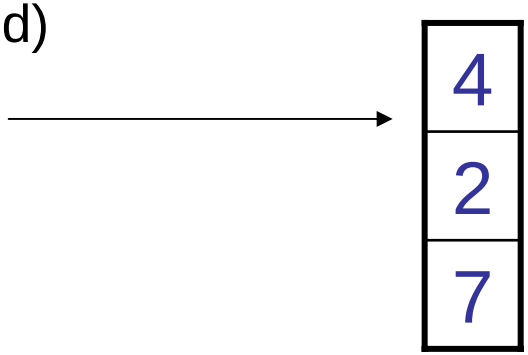
- Extracting
  - x[i] - the i-th element of x x[3] = 7
  - x[c(2,4,3)] - the vector consisting of the 2nd, 4th and 3rd elements of x = c(6,8,7)
  - x[c(TRUE,FALSE, FALSE,TRUE,TRUE)] – the vector consisting of 1st,4th,5th elements of x = c(5,8,9)
- Elementwise operations:
  - x>7 =c(FALSE,FALSE,FALSE,TRUE,TRUE)
  - x[x>7] =c(8,9)
  - x*x =c(25,36,49,64,81)

Extracting/Subscripting

Matrix m

- m[2,1] = 1
- m[2,] = c(1,2) (dimension dropped)
- m[,2] = c(4,2,7) (dimension dropped)
- m[,2,drop=FALSE] 3x1 matrix

3	4
1	2
6	7



Data Frame dataf

- dataf [[2]] 2nd variable (col)
- dataf$age the “age” variable
- dataf[[“age”]] “

Data Input

- From “txt” file with “white space” between fields

group	age	vital.capacity
1	39	4.62
1	40	5.29
1	41	5.52
1	41	3.71
1	45	4.02
1	49	5.09
1	52	2.7
1	47	4.31
1	61	2.7
1	65	3.03
1	58	2.73
1	59	3.67
2	29	5.21
2	29	5.17
2	33	4.88
2	32	4.5

…….


```
dataf2 <- read.table(“c:/temp/example.txt”,header=TRUE)
```

From Excel

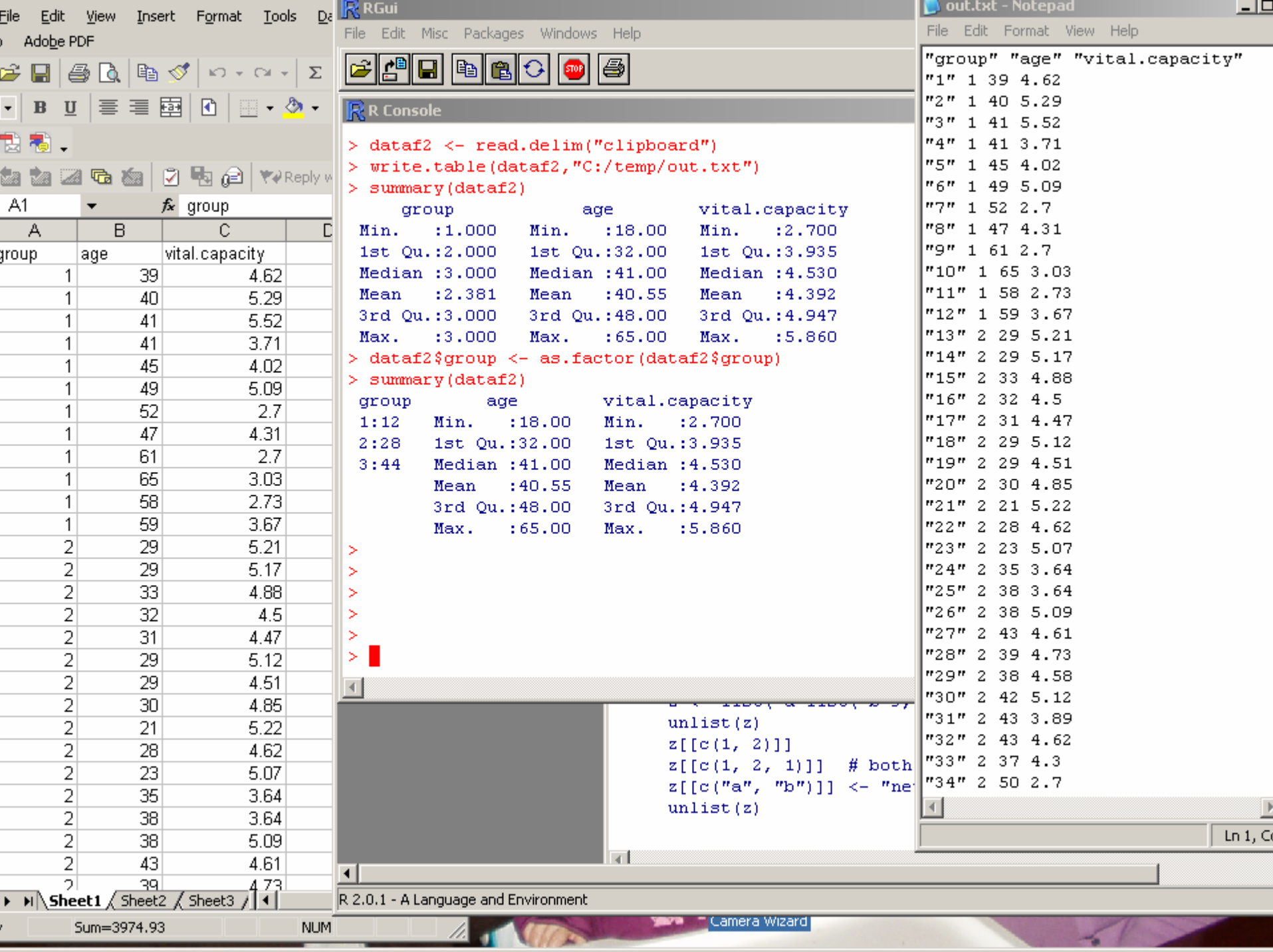
- Select area with data, including headers
- Copy to clipboard (CTRL+C)

In R:

> dataf2 <- read.delim(“clipboard”)

Back to txt

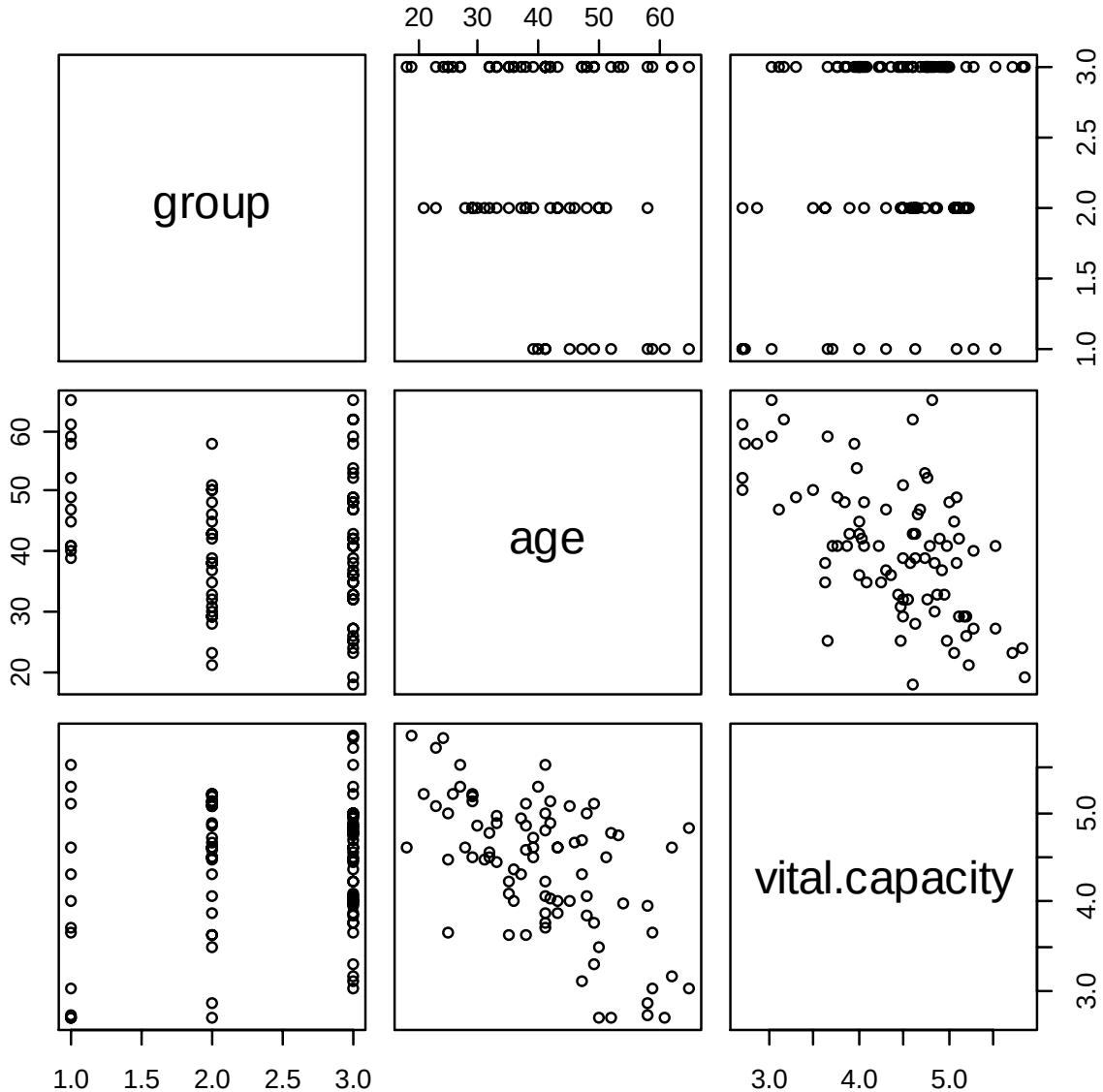
> write.table(dataf2,”c:/temp/out.txt”)


File Edit View Insert Format Tools
Adobe PDF
A1
group
A B C
group age vital.capacity
1 39 4.62
1 40 5.29
1 41 5.52
1 41 3.71
1 45 4.02
1 49 5.09
1 52 2.7
1 47 4.31
1 61 2.7
1 65 3.03
1 58 2.73
1 59 3.67
2 29 5.21
2 29 5.17
2 33 4.88
2 32 4.5
2 31 4.47
2 29 5.12
2 29 4.51
2 30 4.85
2 21 5.22
2 28 4.62
2 23 5.07
2 35 3.64
2 38 3.64
2 38 5.09
2 43 4.61
Sheet1 Sheet2 Sheet3
Sum=3974.93
NUM
RGui
File Edit Misc Packages Windows Help
R Console
> dataf2 <- read.delim("clipboard")
> write.table(dataf2,"C:/temp/out.txt")
> summary(dataf2)
group age vital.capacity
Min. :1.000 Min. :18.00 Min. :2.700
1st Qu.:2.000 1st Qu.:32.00 1st Qu.:3.935
Median :3.000 Median :41.00 Median :4.530
Mean :2.381 Mean :40.55 Mean :4.392
3rd Qu.:3.000 3rd Qu.:48.00 3rd Qu.:4.947
Max. :3.000 Max. :65.00 Max. :5.860
> dataf2$group <- as.factor(dataf2$group)
> summary(dataf2)
group age vital.capacity
1:12 Min. :18.00 Min. :2.700
2:28 1st Qu.:32.00 1st Qu.:3.935
3:44 Median :41.00 Median :4.530
Mean :40.55 Mean :4.392
3rd Qu.:48.00 3rd Qu.:4.947
Max. :65.00 Max. :5.860
>
>
>
>
>
>
unlist(z)
z[[c(1, 2)]]
z[[c(1, 2, 1)]] # both
z[[c("a", "b")]] <- "ne
unlist(z)
R 2.0.1 - A Language and Environment
out.txt - Notepad
File Edit Format View Help
"group" "age" "vital.capacity"
"1" 1 39 4.62
"2" 1 40 5.29
"3" 1 41 5.52
"4" 1 41 3.71
"5" 1 45 4.02
"6" 1 49 5.09
"7" 1 52 2.7
"8" 1 47 4.31
"9" 1 61 2.7
"10" 1 65 3.03
"11" 1 58 2.73
"12" 1 59 3.67
"13" 2 29 5.21
"14" 2 29 5.17
"15" 2 33 4.88
"16" 2 32 4.5
"17" 2 31 4.47
"18" 2 29 5.12
"19" 2 29 4.51
"20" 2 30 4.85
"21" 2 21 5.22
"22" 2 28 4.62
"23" 2 23 5.07
"24" 2 35 3.64
"25" 2 38 3.64
"26" 2 38 5.09
"27" 2 43 4.61
"28" 2 39 4.73
"29" 2 38 4.58
"30" 2 42 5.12
"31" 2 43 3.89
"32" 2 43 4.62
"33" 2 37 4.3
"34" 2 50 2.7
Ln 1, C
Camera Wizard

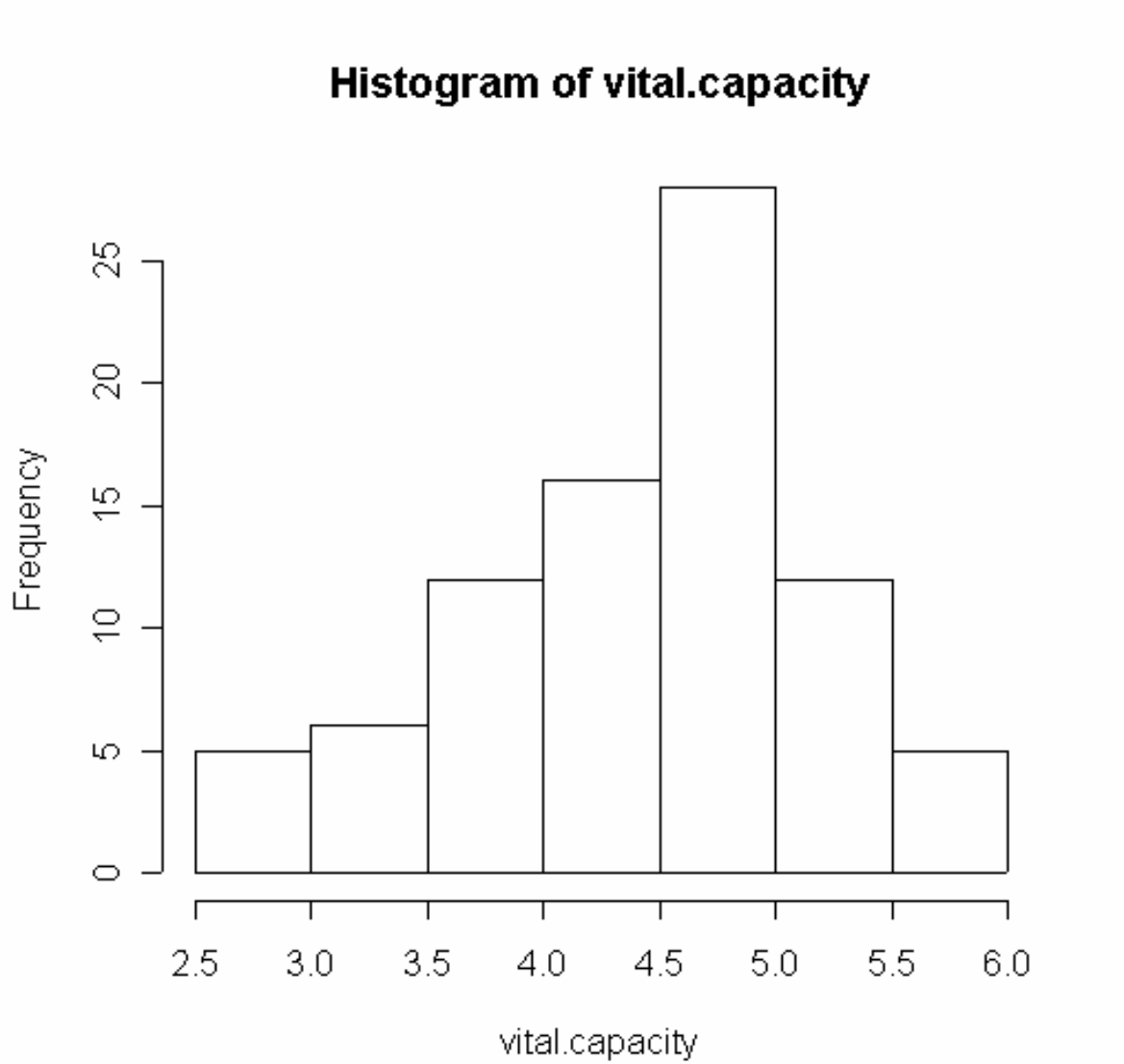
Simple Statistics

- Numerical Summary
  - summary(object) summary(dataf2)
- Graphical description
  - plot(object) plot(dataf2)
- Histogram
  - hist(data) hist(dataf2$vital.capacity)
  - attach(dataf2)
  - hist(vital.capacity)

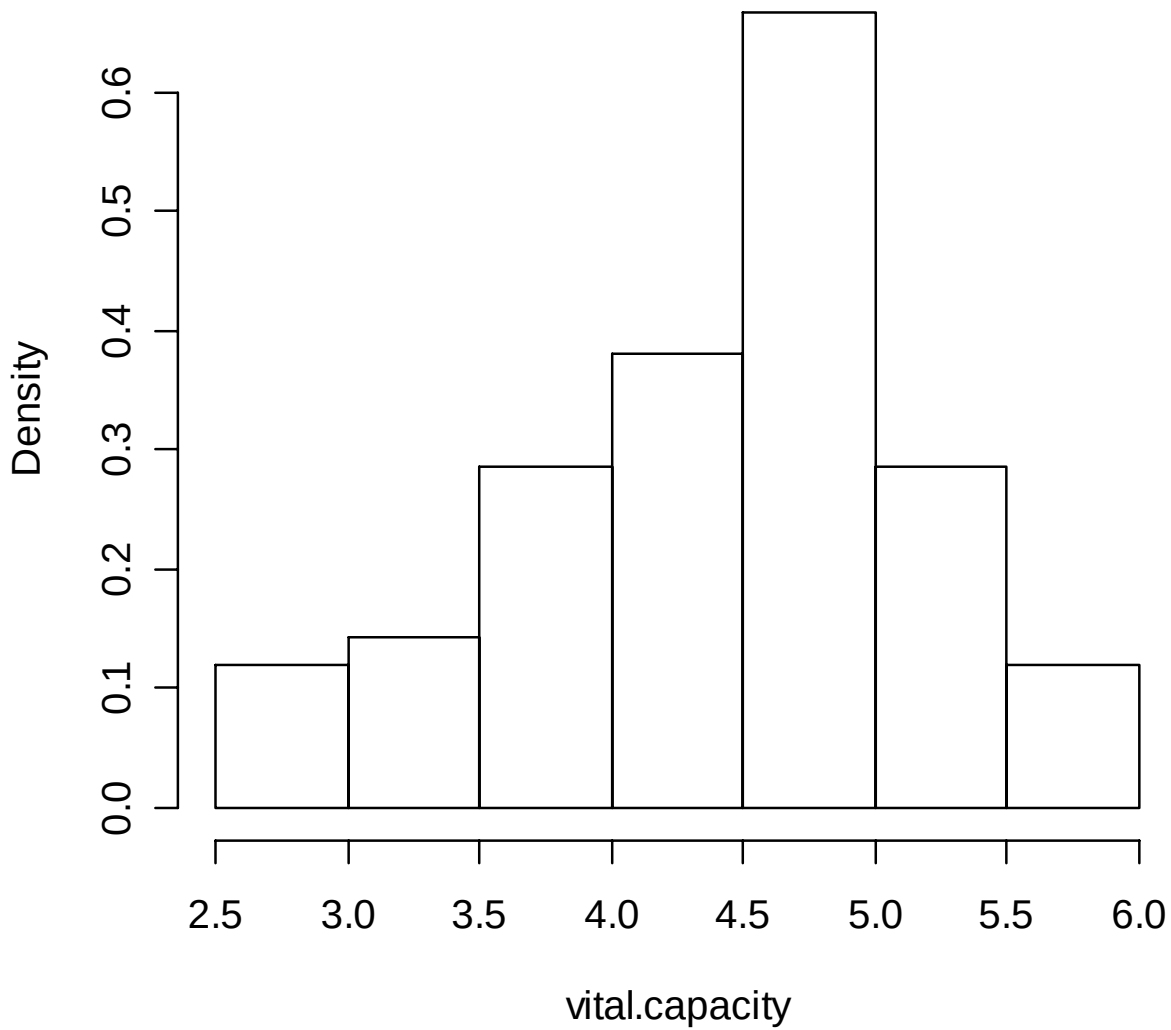
```
> plot(dataf2)
```



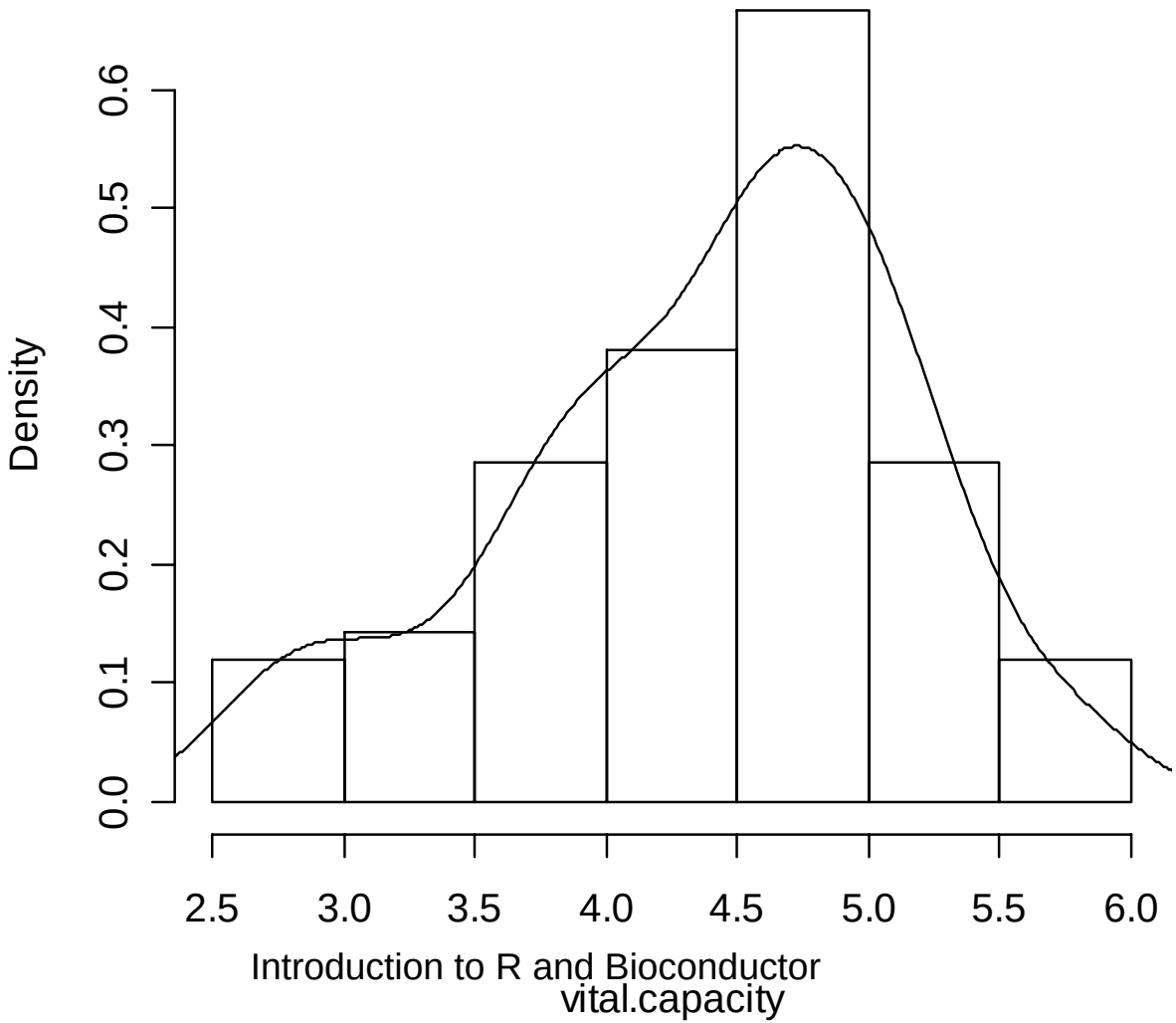
> hist(vital.capacity)



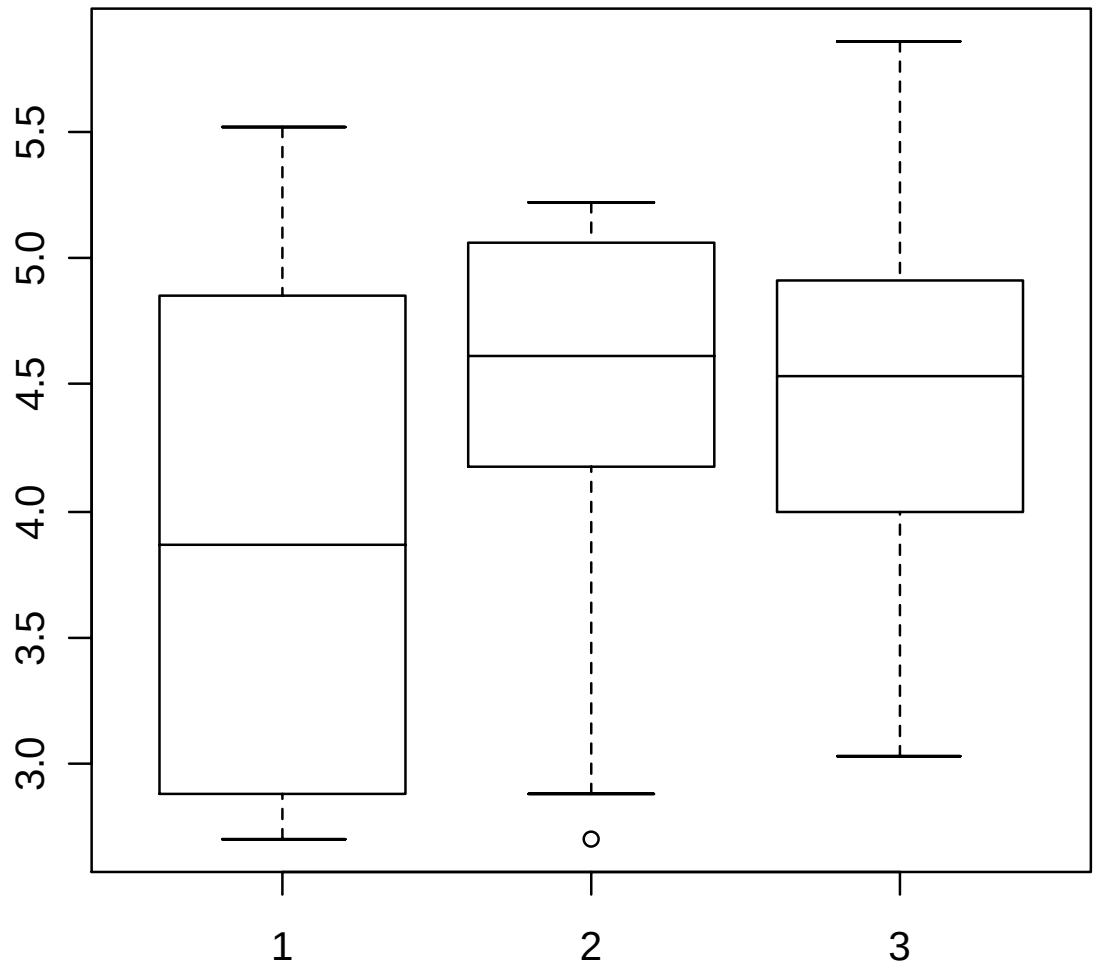
```
> hist( vital.capacity, freq=FALSE )
```



- hist(vital.capacity,freq=FALSE)
- lines(density(vital.capacity))



> boxplot (vital.capacity ~ group)



Formula for Modelling

- Dependent y
- Independent x1, x2, x3,
  - Numeric or factors
- Model
  - y ~ x1 + x2 + x3
- Linear model analysis(regression, anova)
  - lm (y ~ x1 + x2 + x3)

Regression using “lm”

```
> summary(lm(vital.capacity ~ age))

Call:
lm(formula = vital.capacity ~ age)

Residuals:
     Min       1Q   Median       3Q      Max
-1.35136 -0.37332  0.02796  0.40735  1.42776

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.033316   0.247487  24.378  < 2e-16 ***
age         -0.040478   0.005881  -6.883 1.08e-09 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.6068 on 82 degrees of freedom
Multiple R-Squared: 0.3662,     Adjusted R-squared: 0.3584
F-statistic: 47.37 on 1 and 82 DF,  p-value: 1.082e-09
```

Anova – using “lm”

```
> summary(lm(vital.capacity ~ group))

Call:
lm(formula = vital.capacity ~ group)

Residuals:
     Min       1Q   Median       3Q      Max
-1.77179 -0.45205  0.09808  0.51295  1.57083

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.9492     0.2149  18.376   <2e-16 ***
group2        0.5226     0.2569   2.035   0.0452 *
group3        0.5129     0.2425   2.115   0.0375 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.7445 on 81 degrees of freedom
Multiple R-Squared: 0.05767,    Adjusted R-squared: 0.0344
F-statistic: 2.478 on 2 and 81 DF,  p-value: 0.09021
```

Anova summary of “lm”

- `anova(lm(vital.capacity ~ group))`

```
Analysis of Variance Table

Response: vital.capacity
          Df Sum Sq Mean Sq F value  Pr(>F)
group      2  2.747   1.374  2.4785 0.09021 .
Residuals 81 44.894   0.554
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*'
  0.05 `.' 0.1 ` ' 1
```

Analysis of Covariance

```
> summary(lm(vital.capacity ~ age+group ))

Call:
lm(formula = vital.capacity ~ age + group)

Residuals:
     Min       1Q   Median       3Q      Max
-1.38054 -0.38050  0.01321  0.37909  1.37047

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.927982   0.360863  16.427  < 2e-16 ***
age         -0.039775   0.006322  -6.291 1.57e-08 ***
group2       0.046737   0.224536   0.208    0.836
group3       0.116935   0.209236   0.559    0.578
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.6127 on 80 degrees of freedom
Multiple R-Squared: 0.3696,     Adjusted R-squared: 0.3459
F-statistic: 15.63 on 3 and 80 DF,  p-value: 4.323e-08
```

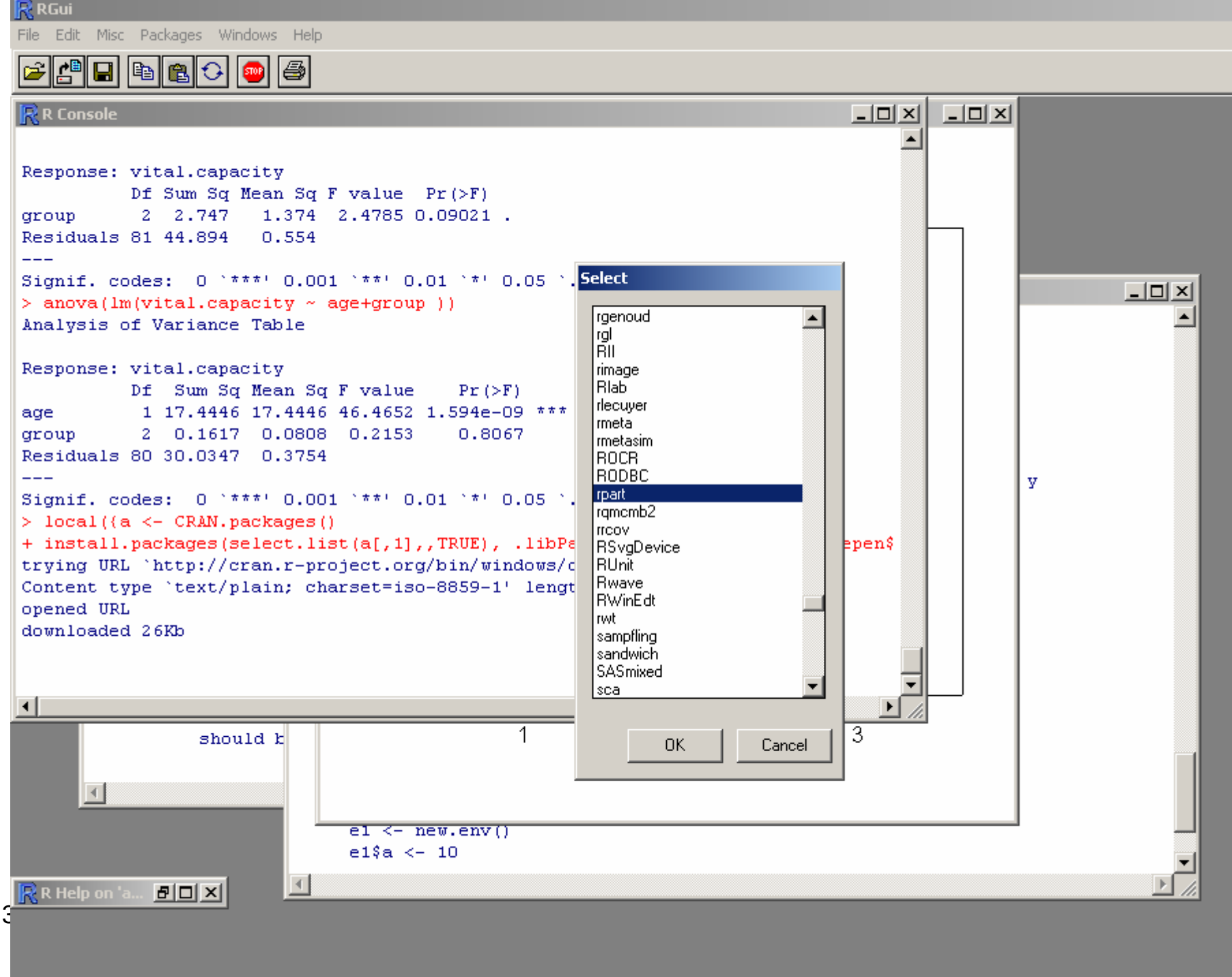
Analysis of Covariance – with ANOVA summary

```
> anova(lm(vital.capacity ~ age+group ))
Analysis of Variance Table

Response: vital.capacity
          Df  Sum Sq Mean Sq F value    Pr(>F)
age        1 17.4446 17.4446 46.4652 1.594e-09 ***
group      2  0.1617  0.0808  0.2153    0.8067
Residuals 80 30.0347  0.3754
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.'
  0.1 ` ' 1
```

Installing, Loading, Using Packages

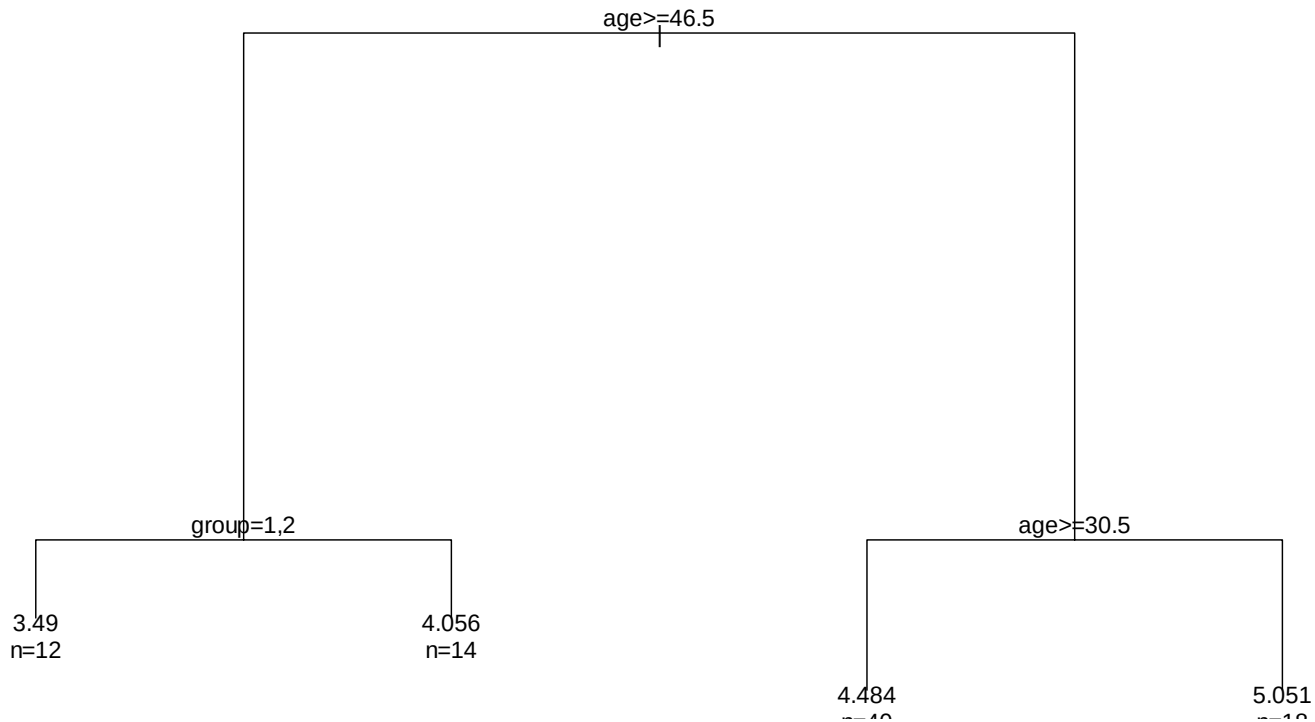
- Menu – Packages
  - install package(s) from CRAN
- Select package you want from list
    - (e.g. rpart)
  - Wait for downloading and installation
- Load package:
  - library(rpart)
- Help: > ?rpart or > help(rpart)


RGui
File Edit Misc Packages Windows Help
R Console
Response: vital.capacity
Df Sum Sq Mean Sq F value Pr(>F)
group 2 2.747 1.374 2.4785 0.09021 .
Residuals 81 44.894 0.554

Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.
> anova(lm(vital.capacity ~ age+group))
Analysis of Variance Table
Response: vital.capacity
Df Sum Sq Mean Sq F value Pr(>F)
age 1 17.4446 17.4446 46.4652 1.594e-09 ***
group 2 0.1617 0.0808 0.2153 0.8067
Residuals 80 30.0347 0.3754

Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.
> local({a <- CRAN.packages()
+ install.packages(select.list(a[,1],,TRUE), .libPa
trying URL `http://cran.r-project.org/bin/windows/c
Content type `text/plain; charset=iso-8859-1' lengt
opened URL
downloaded 26Kb
Select
rgenoud
rgl
RII
rimage
Rlab
rlecuyer
rmeta
rmetasim
ROCR
RODBC
rpart
rqmcmb2
rrcov
RSvgDevice
RUnit
Rwave
RWinEdt
rwt
sampfling
sandwich
SASmixed
sca
OK
Cancel
should b
1
3
e1 <- new.env()
e1$a <- 10
R Help on 'a...

```
➢ cart <- rpart(vital.capacity ~ age+group)
> plot(cart)
> text(cart, use.n=TRUE,pretty=0)
```



Summary of Rpart

```
Call:
rpart(formula = vital.capacity ~ age + group)
  n= 84

          CP nsplit rel error    xerror      xstd
1 0.28172635      0 1.0000000 1.0274703 0.1481476
2 0.08371287      1 0.7182736 0.9337297 0.1052320
3 0.04351568      2 0.6345608 0.8225681 0.1027480
4 0.01000000      3 0.5910451 0.8110266 0.1019493

Node number 1: 84 observations,    complexity param=0.2817264
  mean=4.392024, MSE=0.5671542
  left son=2 (26 obs) right son=3 (58 obs)
  Primary splits:
      age   < 46.5 to the right, improve=0.28172640, (0 missing)
      group splits as  LRR,      improve=0.05763348, (0 missing)
  Surrogate splits:
      group splits as  LRR, agree=0.714, adj=0.077, (0 split)

Node number 2: 26 observations,    complexity param=0.04351568
  mean=3.795, MSE=0.6265558
  left son=4 (12 obs) right son=5 (14 obs)
  Primary splits:
      group splits as  LLR,      improve=0.1272604, (0 missing)
      age   < 56   to the right, improve=0.1119463, (0 missing)
  Surrogate splits:
      age < 49.5 to the right, agree=0.577, adj=0.083, (0 split)

Node number 3: 58 observations,    complexity param=0.08371287
  mean=4.659655, MSE=0.3091171
  left son=6 (40 obs) right son=7 (18 obs)
  Primary splits:
      age   < 30.5 to the right, improve=0.2224445000, (0 missing)
      group splits as  LRL,      improve=0.0006047698, (0 missing)

Node number 4: 12 observations
  mean=3.49, MSE=0.63865
```

Defining functions - simple

- Harmonic mean function:
  - harm <- function(x) 1/(mean(1/x))

```
> harm(c(1,2,3))
[1] 1.636364

> 18/11
[1] 1.636364
> harm(c(1,2,3,0))
[1] 0

> 1/0
[1] Inf
```

Defining functions - cont

- harm0 <- function(x) {
  if (min(x) <= 0) {
  print(“values must be > 0”)
  return(as.numeric(NA))
  }
  else return(1/mean(1/x))
  }