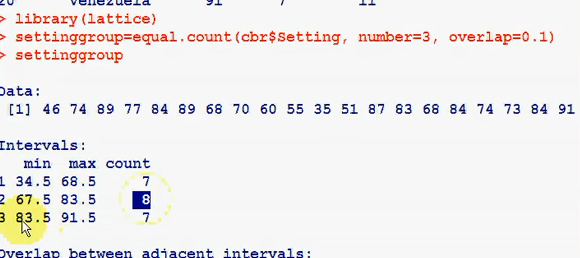
**R – Commands**

* Data pre-processing
  + Data cleaning
    - Identifying and correcting data types
    - Change variable types (recoding data)
    - Resolve inconsistencies
    - Identify and/or remove outliers
    - Deal with missing values
  + Data reduction
    - Combine variables to reduce data volume (too many variables)
    - Use sample data (too many rows)
  + Data transformation
    - Normalization and aggregation
* 4. Data visualization
  + Tables (summary, pivot)
  + One dimensional Graphs
    - Histograms, bar charts
  + Two dimensional Graphs
    - Scatter plots, mosaic charts, overlay plots
  + Higher dimensions
    - 3-D Scatter plots
    - Tree maps
    - Bubble plots
    - Principal components
* 5. Pattern Discovery
  + Clustering
  + Sequence analysis
* 6. Predictive Modeling
* 7. Model Comparison

--------------------------------------------------



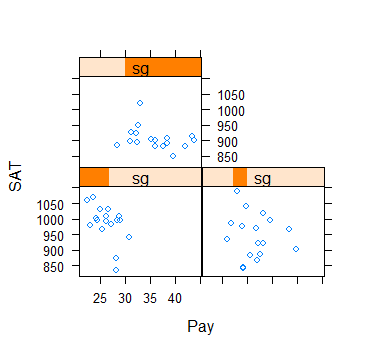


**/\* uploading and using lattice package \*/**

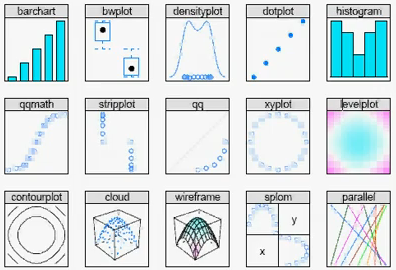
library (lattice) -- running the lattice packige

sg = equal.count (data1$Spending, number = 3, overlap = 0.1) -- splitting the Spending column on 3 sets

xyplot (SAT~Pay|sg, data = data1) -- plotting SAT on Pay scatter plot in regrads to three splittings



Different Commands



Running help on a particular function, e.g. log

> ?log

help.start()

> help(mean)

**> help.search ("mean") -- if you don’t know that exact name of the function**

quit() – quit R

fix (x) – changing/ amending values

Ctrl + L -- clean the screen

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

attach (data1)

density

This makes “density” as a variable

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> x = c (1,2,3,4,5)

> x

[1] 1 2 3 4 5

> print (x)

[1] 1 2 3 4 5

> x <- c (1,3,7) – creating a vector

> x – showing the vector on the screen

[1] 1 3 7

v <- c (1:10) creates a vector of 10 elements numbered 1 through 10

> x2 <- 1:5

> x2

[1] 1 2 3 4 5

> x3 <- seq (1,20,3) -- from 1 to 20, STEP 3

> x3

[1] 1 4 7 10 13 16 19

>

> x <- 1:5

> y = 6:10

> x

[1] 1 2 3 4 5

> y

[1] 6 7 8 9 10

> z = c(x,y)

> z

[1] 1 2 3 4 5 6 7 8 9 10

> z = c (y,x)

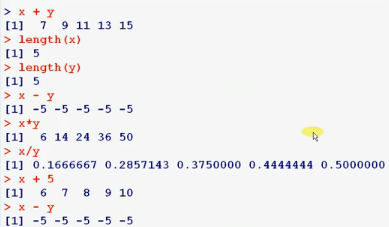
> z

[1] 6 7 8 9 10 1 2 3 4 5 -- notice the difference when we change x and y

>

> length (x)

[1] 5



> y

[1] 6 7 8 9 10

> y [2] -- shows the second element in the vector

[1] 7

> y [c(2,3)] – shows the second and third elements

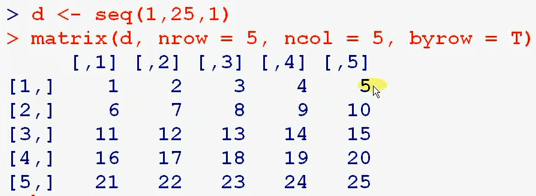
[1] 7 8

> y [-c(2,3)] – shows all elements except for the second and third

[1] 6 9 10

>

**/\* Creating a Matrix \*/**



> d <- seq (1,25,1)

> matrix (d, nrow = 5, ncol = 5, byrow = T)

[,1] [,2] [,3] [,4] [,5]

[1,] 1 2 3 4 5

[2,] 6 7 8 9 10

[3,] 11 12 13 14 15

[4,] 16 17 18 19 20

[5,] 21 22 23 24 25

> matrix (d, nrow = 5, ncol = 5, byrow = F)

[,1] [,2] [,3] [,4] [,5]

[1,] 1 6 11 16 21

[2,] 2 7 12 17 22

[3,] 3 8 13 18 23

[4,] 4 9 14 19 24

[5,] 5 10 15 20 25

>

> matrix (d, nrow = 5, ncol = 5, byrow = F)**->m**

> m

[,1] [,2] [,3] [,4] [,5]

[1,] 1 6 11 16 21

[2,] 2 7 12 17 22

[3,] 3 8 13 18 23

[4,] 4 9 14 19 24

[5,] 5 10 15 20 25

>

> m [1,]

[1] 1 6 11 16 21

> m [2,2]

[1] 7

> m [2,1:3]

[1] 2 7 12

> m [4,c(1,3,5)]

[1] 4 14 24

> m [4,-c(1,3,5)]

[1] 9 19

*as.data.frame()* function to create the data frame.

then edit that object via the *edit*() or *fix*() functions

**I. SUMMARY STATISTICS AND DATASET ANALYSIS**

> data1 = read.table ("C:\\RProjects\\SAT.txt", header = T, sep = ",")

> names (data1) -- getting headers

[1] "State" "Students" "Pay" "Spending" "SAT"

> data1$Spending -- shows data for a particular column

[1] 3648 7887 4231 3334 4826 4809 7914 6016 8210 5154 4860 5008 3200 5062 5051

[16] 4839 5009 4390 4012 5894 6164 6351 5257 5260 3322 4415 5184 4381 4564 5504

[31] 9159 4446 8500 4802 3685 5639 3742 5291 6534 6989 4327 3730 3707 4238 2993

[46] 5740 5360 5045 5046 5946 5255

> data1 [1:5,3] -- getting values from 1 to 5 from the column 3

[1] 27.3 43.4 30.8 23.0 39.8

> x <- c (1,3,7)

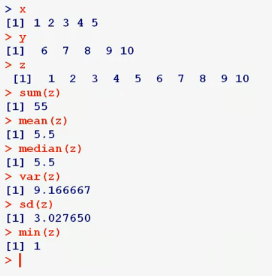
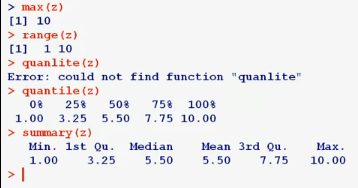
> x

[1] 1 3 7

> summary (x)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.000 2.000 3.000 3.667 5.000 7.000



**> str (SAT) -- gives the following results**

**'data.frame': 51 obs. of 5 variables:**

$ State : Factor w/ 51 levels "AK","AL","AR",..: 2 1 4 3 5 6 7 9 8 10 ...

$ Students: int 8 42 25 6 45 28 74 58 68 44 ...

$ Pay : num 27.3 43.4 30.8 23 39.8 31.8 43.8 35.2 39.6 30.6 ...

$ Spending: int 3648 7887 4231 3334 4826 4809 7914 6016 8210 5154 ...

$ SAT : int 984 914 942 981 903 969 901 903 850 884 ...

**> summary (SAT) -- gives the summary of the table**

State Students Pay Spending SAT

AK : 1 Min. : 4.00 Min. :22.40 Min. :2993 Min. : 834.0

AL : 1 1st Qu.:11.50 1st Qu.:27.65 1st Qu.:4354 1st Qu.: 893.0

AR : 1 Median :25.00 Median :30.80 Median :5045 Median : 933.0

AZ : 1 Mean :33.75 Mean :31.30 Mean :5175 Mean : 945.5

CA : 1 3rd Qu.:57.50 3rd Qu.:33.85 3rd Qu.:5690 3rd Qu.: 994.5

CO : 1 Max. :74.00 Max. :43.80 Max. :9159 Max. :1088.0

(Other):45

> density (SAT)

Call:

density.default(x = SAT)

Data: SAT (51 obs.); Bandwidth 'bw' = 26.55

x y

Min. : 754.3 Min. :3.712e-06

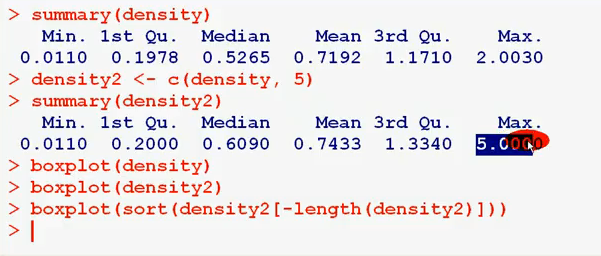
1st Qu.: 857.7 1st Qu.:3.692e-04

Median : 961.0 Median :2.173e-03

Mean : 961.0 Mean :2.417e-03

3rd Qu.:1064.3 3rd Qu.:4.312e-03

Max. :1167.7 Max. :5.595e-03



**II. DATA VISUALIZATION**

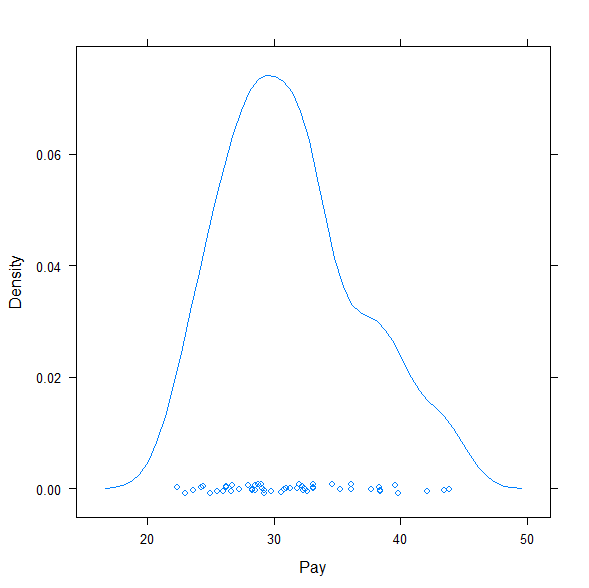
Use Lattice Package

library(lattice)

densityplot(purchase\_size)

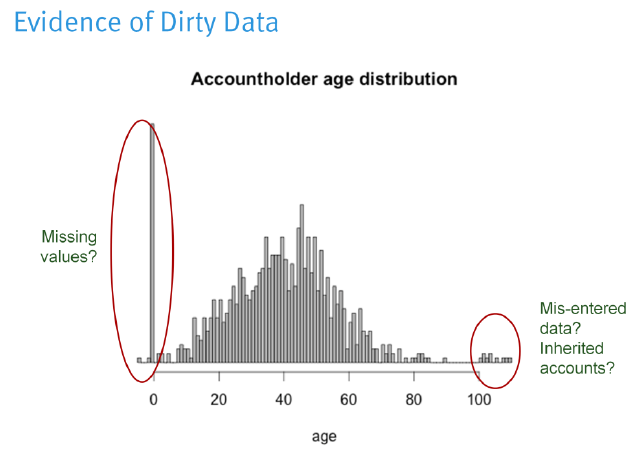
> library(lattice)

> densityplot(Pay)

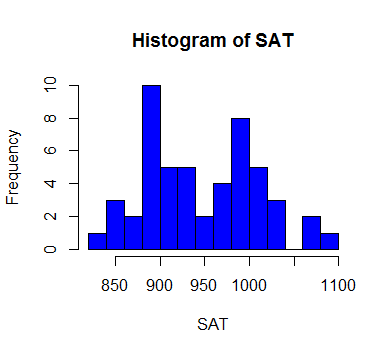


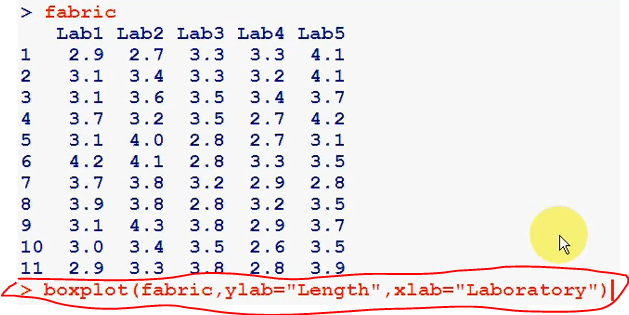
If this data is in a vector called *age*, then the plot is made by:

hist(age, breaks=100, main="Accountholder age distribution", xlab="age", col="gray")

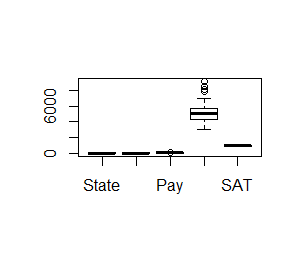


> hist (SAT, breaks = 40, col = "blue",xlab = "SAT")





> Boxplot (data1) -- to see boxplots for each variable in the data set



46 VT 62 31.0 5740 897

47 VA 58 32.4 5360 895

48 WA 44 33.1 5045 923

49 WV 15 26.0 5046 933

50 WI 11 33.1 5946 1019

51 WY 13 29.0 5255 977

> summary (j)

State Students Pay Spending SAT

AK : 1 Min. : 4.00 Min. :22.40 Min. :2993 Min. : 834.0

AL : 1 1st Qu.:11.50 1st Qu.:27.65 1st Qu.:4354 1st Qu.: 893.0

AR : 1 Median :25.00 Median :30.80 Median :5045 Median : 933.0

AZ : 1 Mean :33.75 Mean :31.30 Mean :5175 Mean : 945.5

CA : 1 3rd Qu.:57.50 3rd Qu.:33.85 3rd Qu.:5690 3rd Qu.: 994.5

CO : 1 Max. :74.00 Max. :43.80 Max. :9159 Max. :1088.0

(Other):45

> plot (j$SAT)

> plot (j$Spending)

> plot (j$Pay)

**/\* Creating a diagrams in R \*/**

> x

[1] 1 2 3 4 5

> t

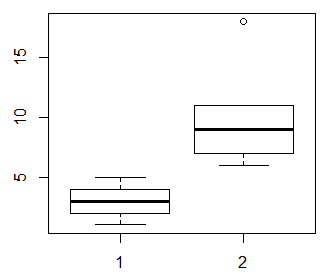
[1] 6 7 18 11 9

> plot (x,t)

>plot (x) – plotting a graph

> boxplot (x)

> boxplot (x,t)



> help (plot)

> hist (x) -- histogram

> plot (density (x)) -- density plot

> plot (x, t, type = "b")

> plot (x,t,type = "b", main = "My TITLE")

type

what type of plot should be drawn. Possible types are

* "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,
* "S" for other **s**teps, see ‘Details’ below,
* "n" for no plotting.
* main
* an overall title for the plot: see [title](http://127.0.0.1:14511/library/graphics/help/title).
* sub
* a sub title for the plot: see [title](http://127.0.0.1:14511/library/graphics/help/title).
* xlab
* a title for the x axis: see [title](http://127.0.0.1:14511/library/graphics/help/title).
* ylab
* a title for the y axis: see [title](http://127.0.0.1:14511/library/graphics/help/title).
* asp
* the *y/x* aspect ratio, see [plot.window](http://127.0.0.1:14511/library/graphics/help/plot.window).

**/\* Histogram \*/**

> data1 [1:50,3]->m – CREATING A VECTOR FROM THE TABLE

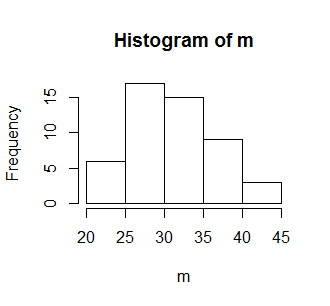
> m

[1] 27.3 43.4 30.8 23.0 39.8 31.8 43.8 35.2 39.6 30.6 29.2 32.5 25.5 34.6 32.0 28.0 29.8 29.1

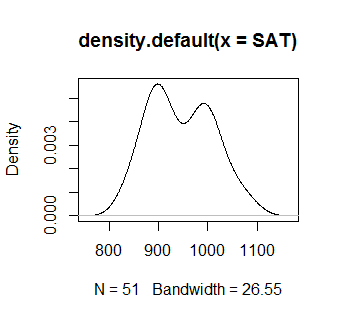
[19] 26.2 28.5 38.4 36.1 38.3 33.1 24.4 28.5 26.7 26.6 32.2 31.3 38.4 26.2 42.1 29.2 23.6 32.6

[37] 24.3 32.3 36.1 37.7 28.3 22.4 28.8 28.3 25.0 31.0 32.4 33.1 26.0 33.1

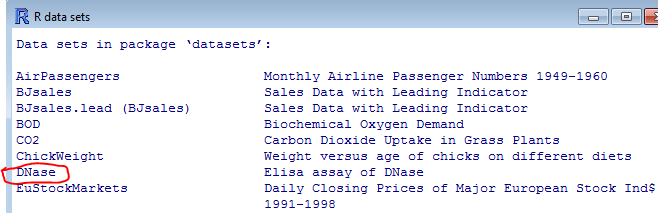
> hist (m)



> plot (density (SAT))

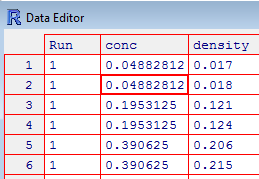


> data () -- some of internal datasets in R



> data (DNase)

> edit (DNase)



**/\* Saving data file \*/**

> datafile = DNase

> sink ("c:\\111\\out1.txt")

> datafile

> sink ()

**/\* write txt file \*/**

> write.table (datafile, file = "c:\\111\\out2.txt")

**/\* write csv file table \*/**

> write.csv (datafile, file = "c:\\111\\out3.csv")

**III. Getting Data Into R from External Dataset**





**R has the ability to read in data in many different formats**.

The *read.table()* function is the most used, although there are multiple helper functions such as *read.csv()*, *read.delim()* and *read.fwf()* for reading fixed-length fields.

Multiple import functions also exist, including reading in data from SPSS, SAS, Sysstat, and other statistical packages. The file name argument to read.table() can also be a URL: this is useful in reading a data file from the Internet. Consult the help subsystem ( *help(read.table)* for more options).

**> read.csv ("SAT.txt")**

State Students Pay Spending SAT

1 AL 8 27.3 3648 984

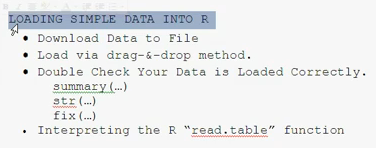
2 AK 42 43.4 7887 914

3 AZ 25 30.8 4231 942

4 AR 6 23.0 3334 981

5 CA 45 39.8 4826 903

Read.table (“SAT.txt”, sep = “,”, header = True)



**/\* Comma Separated File CSV\*/**

> MY <- read.table ("C:\\RProjects\\DataTest\\GiveMe.csv",header = T, sep = ",")

Drug and Drop the file into the R to get the path

> **read.table** ("C:\\RProjects\\DataTest\\SAT.csv", sep = ",", header = T)

Where:

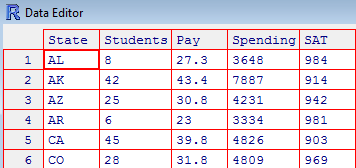
Sep = “,” – defining delimiter

Header = T – defining the first row as a header

> **SAT** <-read.table ("C:\\RProjects\\DataTest\\SAT.csv", sep = ",", header = T)

Defining SAT as a table name and upload data there

> fix (SAT) -- gives us an opportunity to change data



Another Way

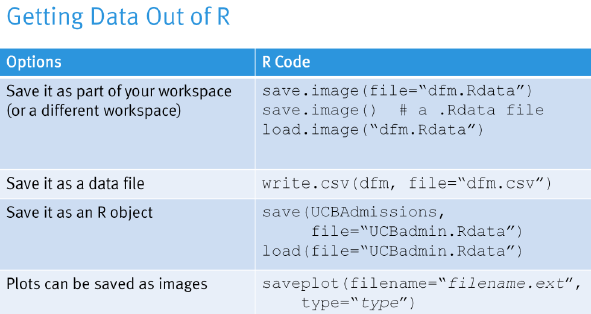
> T <- read.csv ("C:\\RProjects\\DataTest\\SAT.csv", header = T) -- we do not need to use sep = “,” here

**/\* If a file is a Space Separated File \*/**

> MY <- read.table ("C:\\RProjects\\DataTest\\GiveMe.csv",header = T, sep = "") -- we do not use space in between “”

**/\* TAB Separated File\*/**

> MY <- read.table ("C:\\RProjects\\DataTest\\GiveMe.csv",header = T, sep = "\t")



**> write.csv (SAT, file = "mysas1.csv")**