Statistics with R – Beginner Level

Section 2

Descriptive Statistics

Lesson 8 - Using Base R to Generate Statistical Indicators

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to compute the main statistical indicators
### for a numeric variable in base R
##########

####### we will compute these indicators for the variable income

### mean

mean(demo$income)

### or

m <- mean(demo$income)

print(m)

### standard deviation and variance

sd(demo$income)
```

```
var(demo$income)

### minimum, maximum and range

min(demo$income)

max(demo$income)

range(demo$income)

max(demo$income) - min(demo$income)

### median

median(demo$income)

### quartiles

quantile(demo$income)
```

Lesson 9 - Descriptive Statistics with the Psych Package

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to compute the main statistical indicators for a
numeric variable
### with the psych package
##########

###### we will compute this indicators for the following
variables
###### age, income and car price

### create a matrix with the variables of interest

demo2 <- cbind(demo$age, demo$income, demo$carpr)

### give suggestive names to the matrix columns

colnames(demo2) <- c("age", "income", "price")
```

```
View(demo2)

### load the psych package

require(psych)

### use the describe function to generate the statistics
table

describe(demo2)

### the trimmed mean is computed with a default trim of 0.1

### mad - median absolute deviation (the median of the
absolute deviations from the data median)

######## more options for the describe function

describe(demo2, na.rm = TRUE, trim = 0.1, check = TRUE)

### na.rm - if TRUE it omits the missing values (if FALSE
it deletes the cases)

### trim -  sets the trimming fraction

### check - if TRUE it checks for non-numeric data
```

Lesson 10 - Descriptive Statistics with the Pastecs Package

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to compute the main statistical indicators for a
numeric variable
### with the pastecs package
##########

###### we will compute this indicators for the following
variables
###### age, income and car price
```

```
### create a matrix with the variables of interest

demo2 <- cbind(demo$age, demo$income, demo$carpr)

### give suggestive names to the matrix columns

colnames(demo2) <- c("age", "income", "carpr")

View(demo2)

### load the pastecs package

require(pastecs)

### before computing the indicators we set some options (in base R)

options(scipen=100)  ## force R to use the standard notation, NOT the exponential notation

options(digits=2)    ## make R show only the first two decimals

### run the stat.desc funtion from pastecs

### if we want ALL the statistics we run

stat.desc(demo2)

### if we want to omit the basic statistics we run

stat.desc(demo2, basic = FALSE)

### if we want the basic statistics only we can execute

stat.desc(demo2, desc = FALSE)
```

Lesson 11 - Determining the Skewness and Kurtosis

```
demo <- read.csv("demographics.csv")

View(demo)
```

```
##########
### how to compute skewness and kurtosis
### with the e1071 package
##########

### we will use the variable income for our examples

### load the package

require(e1071)

### compute the skewness

skewness(demo$income)

### compute the kurtosis

kurtosis(demo$income)
```

Lesson 12 - Computing Quantiles

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to compute quantiles
##########

### we will use the variable income for our example

### compute the following percentiles
### 17%, 55% and 97%

### use the quantile function in the stats package
### (this package loads automatically when you start R)

quantile(demo$income, probs = c(0.17, 0.55, 0.97))

### to get the quartiles

quantile(demo$income, probs = c(0.25, 0.50, 0.75))
```

Lesson 13 - Determining the Mode

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to determine the mode of a numeric variable
### with the package modeest
##########

### we will find out the mode for the variable income

### load the package

require(modeest)

mlv(demo$income, method="mfv")   ### "mfv" stands for "most
frequent value"

### for the discrete variables, the best way to compute the
mode is to tabulate the frequencies
### as we will learn in a future lecture of this course
```

Lesson 14 - Getting the Statistical Indicators by Group with DoBy

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to compute the main statistical indicators by
groups or subsets
### with the package doBy
##########

### we will get the main statistical indicators for the
variable income
### by gender (separately for male and female subjects)
### load the package

require(doBy)

### define the function that generates the statistics
### N.B. descStat is the command (in the doBy package) that
computes the statistical indicators

func <- function(x) {descStat(x, na.rm=TRUE)}

### use the command summaryBy

summaryBy(income~gender, data=demo, FUN=func)

### get the main statistical indicators for the variables
income and age
### by gender

summaryBy(income+age~gender, data=demo, FUN=func)
```

Lesson 15 - Getting the Statistical Indicators by Group with DescribeBy

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to compute the main statistical indicators by
groups or subsets
### with the package psych (command describeBy)
##########

### we will get the main statistical indicators for the
variable income
### by education level

### load the package

require(psych)

### use the describeBy command
```

```
describeBy(demo$income, demo$educ)
```

Lesson 16 - Getting the Statistical Indicators by Group with Stats

```
demo <- read.csv("demographics.csv")

View(demo)

##########
### how to compute the main statistical indicators by
groups or subsets
### with the package stats (command aggregate)
##########

### we will get some statistical indicators for the
variable age
### by marital status (married/unmarried)

### compute the mean

aggregate(demo$age, by=list(demo$marital), FUN=mean)

### compute the standard deviation

aggregate(demo$age, by=list(demo$marital), FUN=sd)

### compute the median

aggregate(demo$age, by=list(demo$marital), FUN=median)

### compute the variance

aggregate(demo$age, by=list(demo$marital), FUN=var)

### etc.

### very useful when we want to combine the factor
categories
```

```
aggregate(demo$age, by=list(demo$marital, demo$gender),
FUN=mean)
```