tute2.Rmd

Andrej Panjkov

## BMSC203 2016 tute 2 in R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

Note I am not using echo=FALSE, to reveal the R code being executed.

### Part 2 Question 2

Part 2: Descriptive Statistics  
  
You have been given a set of data describing the weights (in kg) of randomly selected male members of a local gym.  
  
74.6 93.5 87.1 79.4 80.6 86.3 92.4 78.3 85.0 86.3  
  
2) From the same data set, calculate the following descriptive statistics using Excel (you will need to input the data into Excel to complete this question).  
  
a) The mean weight of male gym members  
  
b) The median weight of male gym members  
  
c) The modal weight of male gym members  
  
d) The standard deviation of the sample weights  
  
e) The size of the sample (n)  
  
f) The lowest weight in the sample  
  
g) The highest weight in the sample

malegymweights = c(74.6, 93.5, 87.1, 79.4, 80.6, 86.3, 92.4, 78.3, 85.0, 86.3)  
mean(malegymweights)

## [1] 84.35

median(malegymweights)

## [1] 85.65

sd(malegymweights)

## [1] 6.095034

For the modal weight: R does not have a simple builtin function for the mode - it depends on the data type. We do it this way:

First, generate a count table and give it the name mgwcounts:

mgwcounts <- table(malegymweights)

Now find the maximum count, and find the weight with that count:

names(mgwcounts)[mgwcounts == max(mgwcounts)]

## [1] "86.3"

Alternatively, we can use the function **summary** to quickly list some these descriptive statistics:

summary(malegymweights)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 74.60 79.70 85.65 84.35 86.90 93.50

### Part 4 Question 1

1) The first tab in the Excel workbook, called <Patient Data>, contains data about the number of patients seen during night shifts at a local hospital emergency room. Using these data, construct a frequency table. An appropriate number of interval classes is likely to be somewhere between 5 and 10. Start by sorting the data (ask for help if you don't know how to sort data in Excel). Then determine the interval classes that you will use.

First we will import the data from excel into R. There are a number of ways to do this - cut-paste-edit; export a csv file and import into R with read.csv; or install a third-party package for directly importing from .xls files. We'll used read.csv.

Begin by opening excel and exporting the spreadsheet as a .csv file (comma-separated-values.) (Use the file/save as menu.)

Now read the csv file into a *dataframe*, display it, and show the descriptive statistics and the frequency table:

patientdata <- read.csv("Patient Data.csv",header=TRUE)  
patientdata

## Patients.seen  
## 1 12  
## 2 22  
## 3 61  
## 4 33  
## 5 54  
## 6 19  
## 7 10  
## 8 50  
## 9 37  
## 10 51  
## 11 63  
## 12 12  
## 13 65  
## 14 34  
## 15 31  
## 16 31  
## 17 49  
## 18 22  
## 19 15  
## 20 26  
## 21 53  
## 22 59  
## 23 40  
## 24 56  
## 25 33  
## 26 37  
## 27 43  
## 28 21  
## 29 69  
## 30 36  
## 31 23  
## 32 44  
## 33 60  
## 34 36  
## 35 64  
## 36 41  
## 37 40  
## 38 10  
## 39 44  
## 40 14  
## 41 54  
## 42 59  
## 43 48  
## 44 57  
## 45 13  
## 46 61  
## 47 38  
## 48 24  
## 49 66  
## 50 45  
## 51 33  
## 52 67  
## 53 66  
## 54 28  
## 55 55  
## 56 26  
## 57 13  
## 58 34  
## 59 65  
## 60 42  
## 61 20  
## 62 51  
## 63 60  
## 64 19  
## 65 22  
## 66 10  
## 67 48  
## 68 22  
## 69 43  
## 70 47

length(patientdata$Patients.seen)

## [1] 70

summary(patientdata)

## Patients.seen   
## Min. :10.00   
## 1st Qu.:23.25   
## Median :40.00   
## Mean :39.37   
## 3rd Qu.:54.00   
## Max. :69.00

table(patientdata)

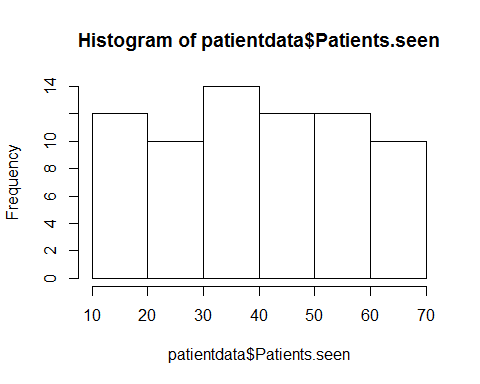
## patientdata  
## 10 12 13 14 15 19 20 21 22 23 24 26 28 31 33 34 36 37 38 40 41 42 43 44 45   
## 3 2 2 1 1 2 1 1 4 1 1 2 1 2 3 2 2 2 1 2 1 1 2 2 1   
## 47 48 49 50 51 53 54 55 56 57 59 60 61 63 64 65 66 67 69   
## 1 2 1 1 2 1 2 1 1 1 2 2 2 1 1 2 2 1 1

To generate the frequency table, use the hist function from the graphics package. We'll let R choose the breakpoints. We'll plot it too.

library(graphics)  
hist(patientdata$Patients.seen,plot=FALSE)

## $breaks  
## [1] 10 20 30 40 50 60 70  
##   
## $counts  
## [1] 12 10 14 12 12 10  
##   
## $density  
## [1] 0.01714286 0.01428571 0.02000000 0.01714286 0.01714286 0.01428571  
##   
## $mids  
## [1] 15 25 35 45 55 65  
##   
## $xname  
## [1] "patientdata$Patients.seen"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

hist(patientdata$Patients.seen)

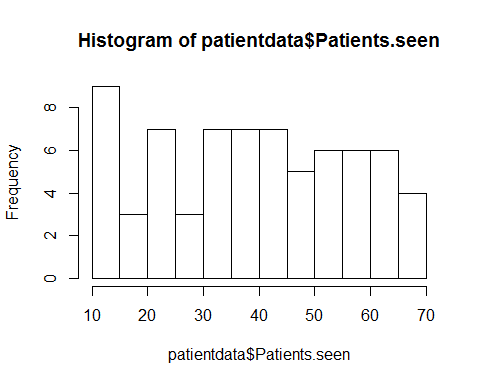


If we want different breakpoints:

hist(patientdata$Patients.seen,breaks=seq(10,70,5), plot=FALSE)

## $breaks  
## [1] 10 15 20 25 30 35 40 45 50 55 60 65 70  
##   
## $counts  
## [1] 9 3 7 3 7 7 7 5 6 6 6 4  
##   
## $density  
## [1] 0.025714286 0.008571429 0.020000000 0.008571429 0.020000000  
## [6] 0.020000000 0.020000000 0.014285714 0.017142857 0.017142857  
## [11] 0.017142857 0.011428571  
##   
## $mids  
## [1] 12.5 17.5 22.5 27.5 32.5 37.5 42.5 47.5 52.5 57.5 62.5 67.5  
##   
## $xname  
## [1] "patientdata$Patients.seen"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

hist(patientdata$Patients.seen,breaks=seq(10,70,5), plot=TRUE)



### Part 4 Question 2

2) Using the same data, construct a stem and leaf plot to visualise the distribution of the data.

A stem-and-leaf plot is more of a tool for a quick hand analysis, but R can do one for us:

stem(patientdata$Patients.seen,scale =2)

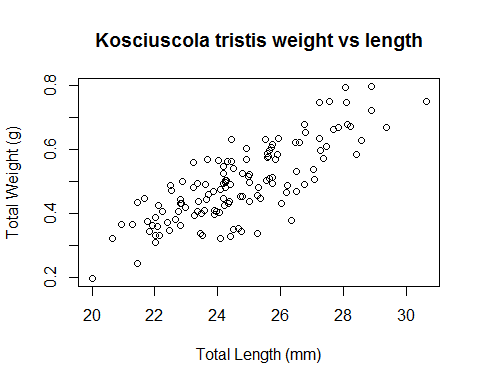
##   
## The decimal point is 1 digit(s) to the right of the |  
##   
## 1 | 00022334  
## 1 | 599  
## 2 | 01222234  
## 2 | 668  
## 3 | 1133344  
## 3 | 66778  
## 4 | 00123344  
## 4 | 57889  
## 5 | 011344  
## 5 | 56799  
## 6 | 001134  
## 6 | 556679

### Part 4 Question 3

3) The second tab in the Excel workbook, called <Grasshopper Data>, contains data about the total length and total weight of female chameleon grasshoppers (Kosciuscola tristis) sampled at Mount Kosciuszko. Using these data, prepare a scatterplot to visualise the data. Make sure that the finished graph is formatted correctly (see Lecture 4 if you are unsure what is meant by correct formatting).

A scatterplot of the grasshopper data.

ghdata=read.csv("Grasshopper Data.csv", header=TRUE)  
plot(ghdata$Total.Length..mm.,ghdata$Total.Weight..g.,  
 main="Kosciuscola tristis weight vs length",  
 xlab="Total Length (mm)",  
 ylab="Total Weight (g)")

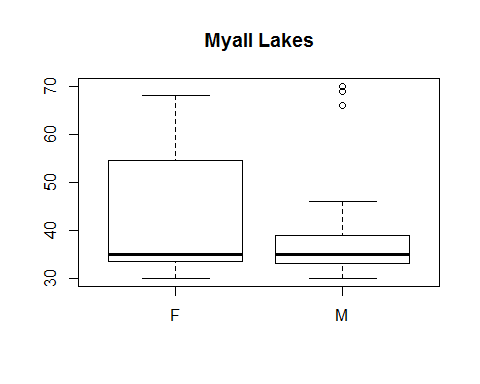


### Part 4 Question 4

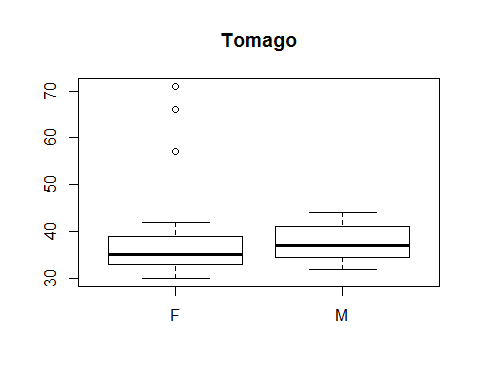
4) The third tab in the Excel workbook, called <Jacky Lizard Data>, contains data about the snout-vent length (SVL) and sex of jacky lizards (Amphibolurus muricatus) sampled at two locations, Myall Lakes and Tomago. Open the statistical software SPSS (refer to the textbook or to the file <SPSS V20 Instructions> on LEO if necessary). Use the jacky lizard data to produce a graph showing the comparison of mean ± standard deviation for lengths of male and female jacky lizards (Amphibolurus muricatus) at Myall Lakes and Tomago.

To do the boxplots, we have to reorganise the data. But first we can do boxplots for each site, grouped by sex.

jackyData <- read.csv("Jacky Lizard Data.csv",header=FALSE,skip=2)  
colnames(jackyData)<-c("MyallLakes","sexML","Tomago","sexT")  
boxplot(MyallLakes ~ sexML, jackyData)  
title(main="Myall Lakes")



boxplot(Tomago ~ sexT,jackyData)  
title(main="Tomago")

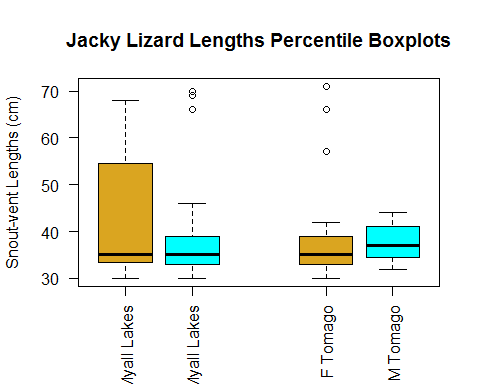


The jackydata spreadsheet is organised by rows, but the Myall Lakes observations are independent of the Tomago observations. We can either split the data into two data frames, one for each site, or four, (one for each site/sex), or reorganise the data into a single dataframe with factors site/sex. I choose the last.

jdM <- data.frame(lengths=jackyData$MyallLakes,sex=jackyData$sexML,site="Myall Lakes")  
jdT <- data.frame(lengths=jackyData$Tomago,sex=jackyData$sexT,site="Tomago")  
jd=rbind(jdM,jdT) # stacks the rows of the two dataframes

Now the boxplot (I polished this in Rstudio's console before putting the code in the R Markdown file):

boxplot(lengths ~ sex + site, jd,  
 names=c("F Myall Lakes", "M Myall Lakes", "F Tomago", "M Tomago"),  
 las=2,  
 at=c(1,2,4,5),   
 ylab="Snout-vent Lengths (cm)",  
 col=c("goldenrod","cyan","goldenrod","cyan"))   
# Note the formula specifies partitioning by both factors.  
# las rotates labels so they are all visible  
# at controls positioning, so we can group sites more clearly - notice 3 is missing.  
title("Jacky Lizard Lengths Percentile Boxplots") # I assume cm.

 This is a nice format, but the question asks for boxplots positioned by mean and sized by standard deviation. A web search gives some recipes, using the lower level function bxp. First we need to build the summary data. tapply applies the function at the end to the first data item, factored by the list of factors.

jdmeans = tapply(jd$lengths,list(jd$sex,jd$site),mean)  
jdstdevs = tapply(jd$lengths,list(jd$sex,jd$site),sd)  
jdmaxs = tapply(jd$lengths,list(jd$sex,jd$site),max)  
jdmins = tapply(jd$lengths,list(jd$sex,jd$site),min)   
print(jdmeans)

## Myall Lakes Tomago  
## F 42.93333 38.56000  
## M 39.12000 37.73333

print(jdstdevs)

## Myall Lakes Tomago  
## F 14.18483 10.512215  
## M 11.67019 3.899939

print(jdmaxs)

## Myall Lakes Tomago  
## F 68 71  
## M 70 44

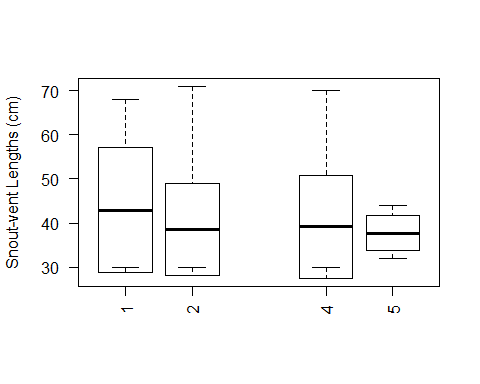
print(jdmins)

## Myall Lakes Tomago  
## F 30 30  
## M 30 32

# mean plus or minus stdev  
jdmeanPlusSD = jdmeans + jdstdevs  
jdmeanMinusSD = jdmeans - jdstdevs  
# assemble the columns into a list for supply to bxp  
# probably a better way to do this  
Fsummaries.df = rbind(jdmins["F",],jdmeanMinusSD["F",],jdmeans["F",],jdmeanPlusSD["F",],jdmaxs["F",])  
Msummaries.df = rbind(jdmins["M",],jdmeanMinusSD["M",],jdmeans["M",],jdmeanPlusSD["M",],jdmaxs["M",])  
Fsummaries.df

## Myall Lakes Tomago  
## [1,] 30.00000 30.00000  
## [2,] 28.74850 28.04778  
## [3,] 42.93333 38.56000  
## [4,] 57.11817 49.07222  
## [5,] 68.00000 71.00000

bxp(list(stats=data.matrix(cbind(Fsummaries.df, Msummaries.df)),n=rep(1,4)),  
 names=c("F Myall Lakes", "F Tomago", "M Myall Lakes", "M Tomago"),  
 las=2,  
 at=c(1,2,4,5),   
 ylab="Snout-vent Lengths (cm)",  
 col=c("goldenrod","cyan","goldenrod","cyan"))



Those are the boxplots showing min, mu-sigma,mu,mu+sigma, max. Note that in some cases mu-sigma is less than min. with some more work, I'm sure we can make this code cleaner, and get the colours and the boxplot labels working, but for now, we leave it here.