9 Additional Graphs Homework

James Young

2/23/2019

# General Instructions

There are 5 exercises, each is worth 10 points. You are required to solve at least one exercise in R, and at least one in SAS. You are required to provide five solutions, each solution will be worth 10 points.

For this exercise, you may use whatever graphics library you desire.

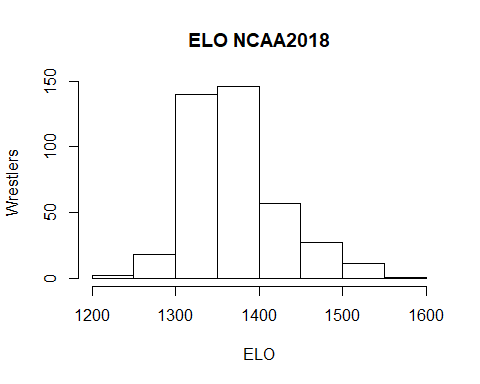
#### Experimental

Again, you will be allowed to provide one solution using Python. Elaborate on the similarities and differences between Python function definitions and R or IML or Macro language.

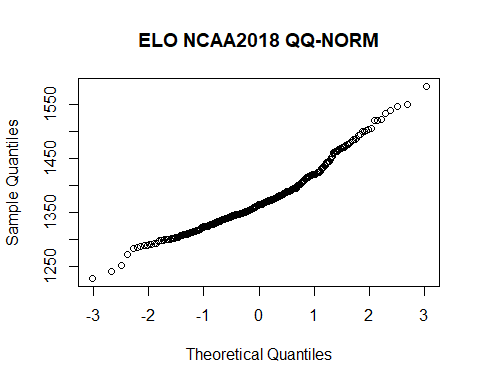
# Exercise 1 Please Grade.

Load the ncaa2018.csv data set and create histograms, QQ-norm and box-whisker plots for ELO. Add a title to each plot, identifying the data.

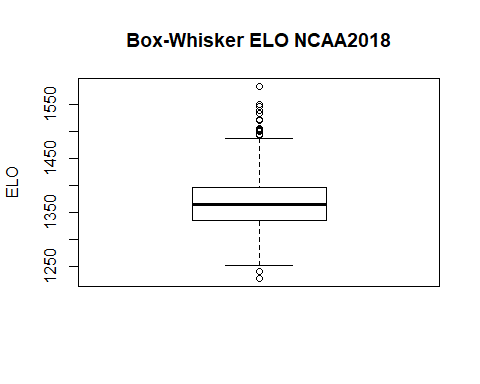
ncaa2018 <- read.csv("C://data/ncaa2018.csv")  
  
hist(ncaa2018$ELO, main = "ELO NCAA2018",  
 xlab = "ELO", ylab = "Wrestlers")



qqnorm(ncaa2018$ELO, main="ELO NCAA2018 QQ-NORM")



boxplot(ncaa2018$ELO, main="Box-Whisker ELO NCAA2018", ylab="ELO")

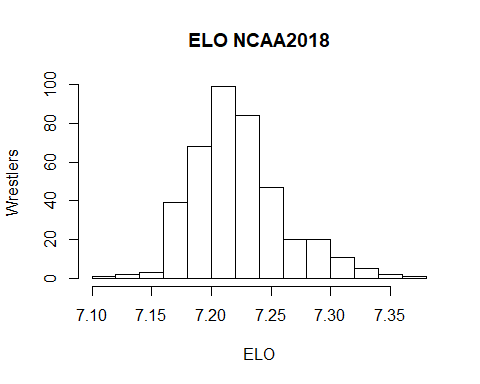


### Part b

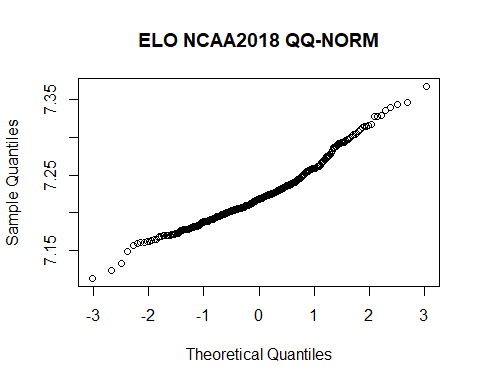
A common recommendation to address issues of non-normality is to transform data to correct for skewness. One common transformation is the log transform.

Transform ELO to log(ELO) and produce histograms, box-whisker and qqnorm plots of the transformed values. Are the transformed values more or less skewed than the original? (Note - the log transform is used to correct skewness, it is less useful for correcting kurtosis).

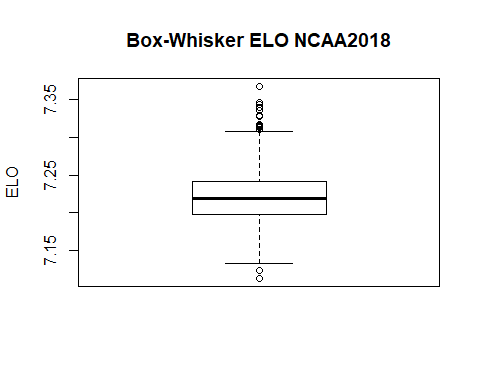
ncaa2018$ELOlog <- log(ncaa2018$ELO)  
  
hist(ncaa2018$ELOlog, main = "ELO NCAA2018",  
 xlab = "ELO", ylab = "Wrestlers")



qqnorm(ncaa2018$ELOlog, main="ELO NCAA2018 QQ-NORM")



boxplot(ncaa2018$ELOlog, main="Box-Whisker ELO NCAA2018", ylab="ELO")

 The transformed values appear less skewed than the original values.

# Exercise 2 Please Grade.

Review Exercise 4, Homework 6, where you calculated skewness and kurtosis. The reference for this exercise, <https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm>, gives four example statistical distributions. We will reproduce the histograms, and add qqnorm and box-whisker plots.

## Part a

Use the code below from lecture to draw 1000 samples from the normal distribution.

#Setting seed for reproducible results  
set.seed(2)  
norm.sample <- rnorm(1000, mean=0, sd=1)

summarydata = function(data = data){  
 data.mean = mean(data, na.rm = TRUE)  
   
 data = data[!is.na(data)]  
   
 data.len = length(data)  
   
 data.sd.nu = 0  
 data.sk.nu = 0  
 data.kur.nu = 0  
 for(i in 1:data.len){  
 data.sd.nu = ((data[[i]]-data.mean)^2)+data.sd.nu  
 data.sk.nu = ((data[[i]]-data.mean)^3)+data.sk.nu  
 data.kur.nu = ((data[[i]]-data.mean)^4)+data.kur.nu  
 }  
   
 data.sd = sqrt(data.sd.nu/(data.len))  
 data.sk = (data.sk.nu/data.len)/(data.sd^3)  
 data.kur = (data.kur.nu/data.len)/(data.sd^4)  
   
 return(list(Skewness=data.sk,  
 Kurtosis=data.kur))  
}

Look up the corresponding r\* functions in R for the Cauchy distribution (use location=0, scale=1), and the Weibull distribution (use shape = 1.5). For the double exponential, use you can use the \*laplace functions from the rmutil library, or you can use rexp(1000) - rexp(1000)

Draw 1000 samples from each of these distributions. Calculate skewness and kurtosis for each sample. You may use your own function, or use the moments library.

set.seed(57)  
cauc.sample <- rcauchy(1000, location = 0, scale = 1)  
weib.sample <- rweibull(1000, shape = 1.5, scale = 1)  
dexp.sample <- rexp(1000) - rexp(1000)

#Determine skewness and kurtosis for other distributions  
  
ns = summarydata(norm.sample)  
cs = summarydata(cauc.sample)  
ws = summarydata(weib.sample)  
ds = summarydata(dexp.sample)  
#Normal distribution  
print(ns)

## $Skewness  
## [1] 0.03622822  
##   
## $Kurtosis  
## [1] 2.784325

#Cauchy distribution  
print(cs)

## $Skewness  
## [1] -15.72096  
##   
## $Kurtosis  
## [1] 400.339

#Weibull distribution  
print(ws)

## $Skewness  
## [1] 0.9760254  
##   
## $Kurtosis  
## [1] 3.954969

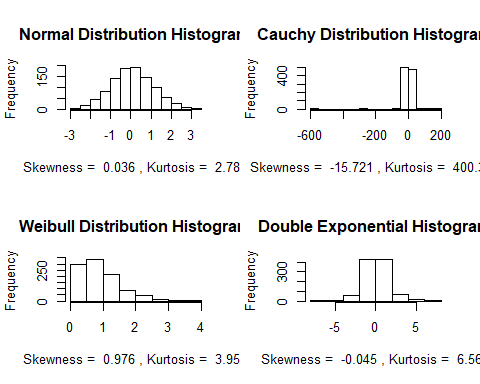
#double exponential distribution  
print(ds)

## $Skewness  
## [1] -0.04460131  
##   
## $Kurtosis  
## [1] 6.562675

## Part b

Plot the histograms for each distribution. Use par(mfrow=c(2,2)) in your code chunk to combine the four histogram in a single plot. Add titles to the histograms indicating the distribution. Set the x-axis label to show the calculated skewness and kurtosis, i.e. skewness = ####, kurtosis = ####

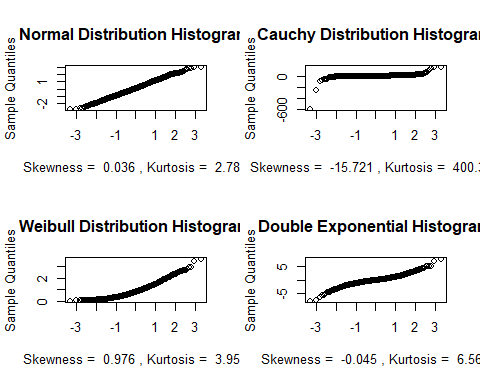
par(mfrow=c(2,2))  
hist(norm.sample,  
 main="Normal Distribution Histogram",   
 xlab = paste("Skewness = ", round(ns$Skewness,3), ", Kurtosis = ", round(ns$Kurtosis,3)))  
hist(cauc.sample,  
 main="Cauchy Distribution Histogram",   
 xlab = paste("Skewness = ", round(cs$Skewness,3), ", Kurtosis = ", round(cs$Kurtosis,3)))  
hist(weib.sample,  
 main="Weibull Distribution Histogram",   
 xlab = paste("Skewness = ", round(ws$Skewness,3), ", Kurtosis = ", round(ws$Kurtosis,3)))  
hist(dexp.sample,  
 main="Double Exponential Histogram",   
 xlab = paste("Skewness = ", round(ds$Skewness,3), ", Kurtosis = ", round(ds$Kurtosis,3)))



## Part c

Repeat Part b, but with QQ-norm plots.

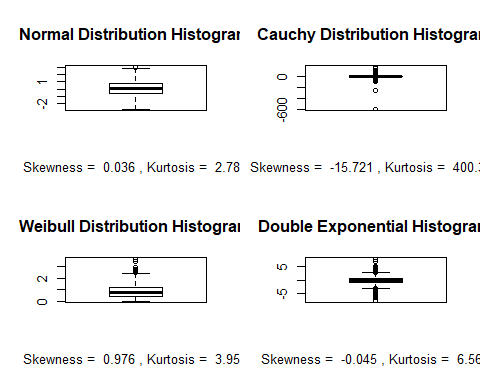
par(mfrow=c(2,2))  
qqnorm(norm.sample,  
 main="Normal Distribution Histogram",   
 xlab = paste("Skewness = ", round(ns$Skewness,3), ", Kurtosis = ", round(ns$Kurtosis,3)))  
qqnorm(cauc.sample,  
 main="Cauchy Distribution Histogram",   
 xlab = paste("Skewness = ", round(cs$Skewness,3), ", Kurtosis = ", round(cs$Kurtosis,3)))  
qqnorm(weib.sample,  
 main="Weibull Distribution Histogram",  
 xlab = paste("Skewness = ", round(ws$Skewness,3), ", Kurtosis = ", round(ws$Kurtosis,3)))  
qqnorm(dexp.sample,  
 main="Double Exponential Histogram",   
 xlab = paste("Skewness = ", round(ds$Skewness,3), ", Kurtosis = ", round(ds$Kurtosis,3)))



## Part d

Repeat Part b, but with box-whisker plots.

par(mfrow=c(2,2))  
boxplot(norm.sample,  
 main="Normal Distribution Histogram",   
 xlab = paste("Skewness = ", round(ns$Skewness,3), ", Kurtosis = ", round(ns$Kurtosis,3)))  
boxplot(cauc.sample,  
 main="Cauchy Distribution Histogram",   
 xlab = paste("Skewness = ", round(cs$Skewness,3), ", Kurtosis = ", round(cs$Kurtosis,3)))  
boxplot(weib.sample,  
 main="Weibull Distribution Histogram",  
 xlab = paste("Skewness = ", round(ws$Skewness,3), ", Kurtosis = ", round(ws$Kurtosis,3)))  
boxplot(dexp.sample,  
 main="Double Exponential Histogram",   
 xlab = paste("Skewness = ", round(ds$Skewness,3), ", Kurtosis = ", round(ds$Kurtosis,3)))

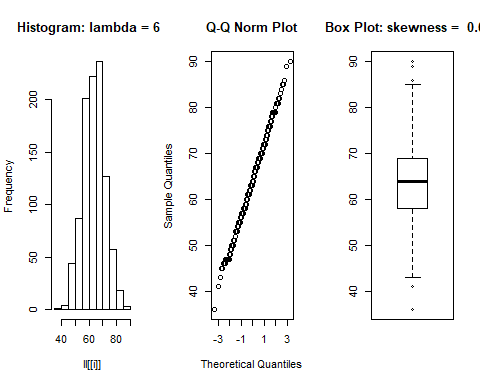
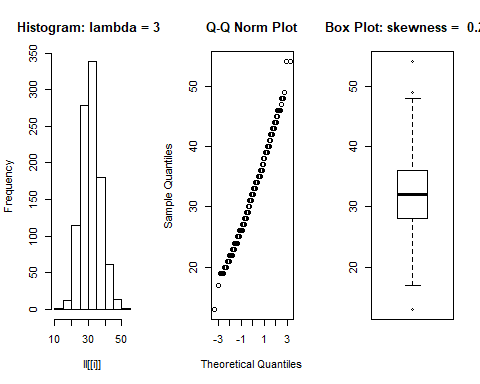
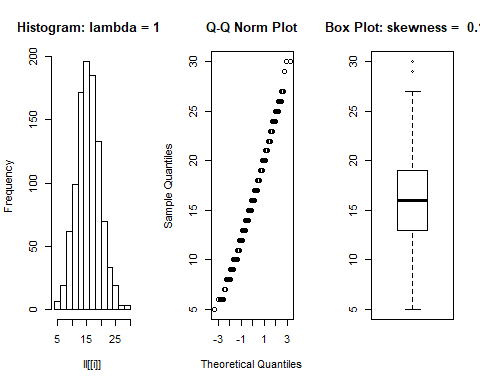
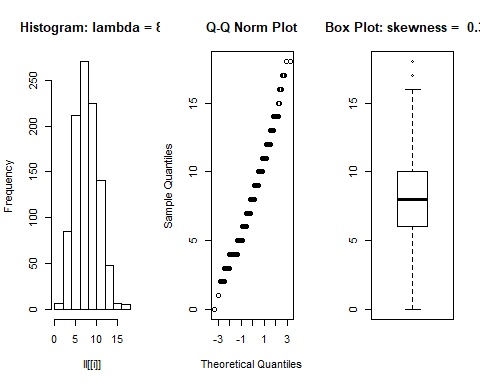
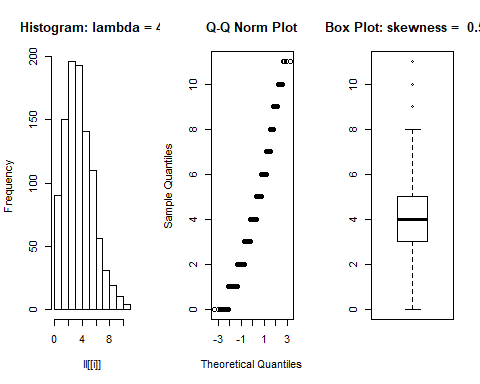
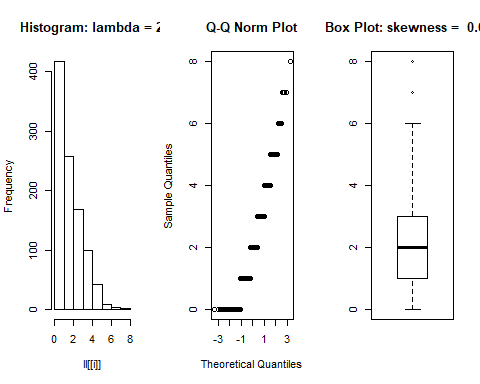


## Exercise 3 Please Grade.

We will create a series of graphs illustrating how the Poisson distribution approaches the normal distribution with large . We will iterate over a sequence of lambda, from 2 to 64, doubling lambda each time. For each ‘lambda’ draw 1000 samples from the Poisson distribution.

Calculate the skewness of each set of samples, and produce histograms, QQ-norm and box-whisker plots. You can use par(mfrow=c(1,3)) to display all three for one lambda in one line. Add lambda=## to the title of the histogram, and skewness=## to the title of the box-whisker plot.

#set seed for reproducibility  
set.seed(2)  
#create string for mu, doubling every time till 64  
lambda = c(2,4,8,16,32,64)  
ll = list()  
templs = list()  
#creating list of poisson distributions for each mu  
for(i in 1:length(lambda)){  
 varlam = lambda[i]  
 temp = rpois(1000, varlam)  
 templs = list(temp)  
 ll = c(ll, templs)  
}  
  
#using above function to calculate skewness for each distribution  
skewness = list()  
for(i in 1:length(ll)){  
 temp = summarydata(ll[[i]])[1]  
 skewness = c(skewness, temp)  
}  
  
#printing out a histogram, qqplot, and boxplot for each lambda  
#printing lambda and skewness on respective plots  
for(i in 1:length(ll)){  
 par(mfrow=c(1,3))  
 hist(ll[[i]],  
 main=paste("Histogram: lambda =", lambda[i]))  
 qqnorm(ll[[i]],  
 main="Q-Q Norm Plot")  
 boxplot(ll[[i]],  
 range=1.5,  
 main=paste("Box Plot: skewness = ", round(skewness[[i]],3)))  
}



## Part b.

Remember that lambda represents the mean of a discrete (counting) variable. At what size mean is Poisson data no longer skewed, relative to normally distributed data? You might run this 2 or 3 times, with different seeds; this number varies in my experience.

#find lambda that creates a skewness under 0.5  
set.seed(1)  
summarydata(rpois(1000, 6))[1]

## $Skewness  
## [1] 0.4909558

set.seed(6690)  
summarydata(rpois(1000, 6))[1]

## $Skewness  
## [1] 0.3489786

set.seed(126)  
summarydata(rpois(1000, 6))[1]

## $Skewness  
## [1] 0.4508549

At what is skewness of the Poisson distribution small enough to be considered normal?

Based on the results achieved, which should be reproducible since I set a seed, a lambda around 6 creates a skew with absolute value less than 0.5 in a Poisson distribution. Many internet discussion say 0.5 is considered moderate skew but is still normal, so I looked for a mean that would produce a skew under that threshold to be safe.

# Exercise 4 Please Grade.

## Part a

Write a function that accepts a vector vec, a vector of integers, a main axis label and an x axis label. This function should 1. iterate over each element in the vector of integers 2. produce a histogram for vec setting the number of bins in the histogram to 3. label main and x-axis with the specified parameters. 4. label the y-axis to read Frequency, bins = and the number of bins.

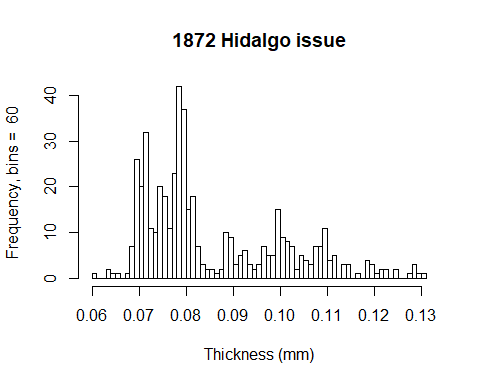
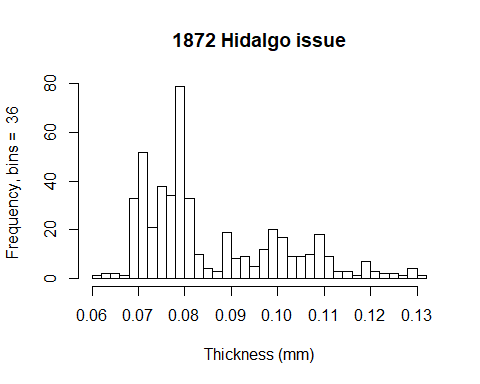
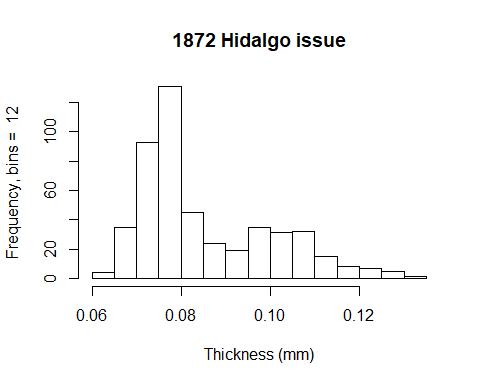
Hint: You can simplify this function by using the parameter ... - see ?plot or ?hist

plot.histograms = function(vec, vectorint, mainlab, xlab){  
 for(i in 1:length(vectorint)){  
 hist(vec,   
 breaks=vectorint[i],   
 main=mainlab,  
 xlab=xlab,  
 ylab=paste("Frequency, bins = ", vectorint[i]))  
 }  
}

## Part b

Test your function with the hidalgo data set (see below), using bin numbers 12, 36, and 60. You should be able to call your function with something like

hidalgo.dat = read.table("C://data/hidalgo.dat", comment.char="")  
  
plot.histograms(hidalgo.dat[,1],c(12,36,60), main="1872 Hidalgo issue",xlab= "Thickness (mm)")

 to plot three different histograms of the hidalgo data set.

## Data

The hidalgo data set is in the file hidalgo.dat These data consist of paper thickness measurements of stamps from the 1872 Hidalgo issue of Mexico. This data set is commonly used to illustrate methods of determining the number of components in a mixture (in this case, different batches of paper). See <https://www.jstor.org/stable/2290118>,  
<https://books.google.com/books?id=1CuznRORa3EC&lpg=PA95&pg=PA94#v=onepage&q&f=false> and <https://books.google.com/books?id=c2_fAox0DQoC&pg=PA180&lpg=PA180&f=false> .

Some analysis suggest there are three different mixtures of paper used to produce the 1872 Hidalgo issue; other analysis suggest seven. Why do you think there might be disagreement about the number of mixtures?

Based on binning number, different numbers of local maximums are observable. Higher number of bins allows more resolution and may lead one to conclude more kinds of paper were used while lower bin numbers may bring about the conclusion of three kinds of paper.

Done.