#####

#

# Week4 Synchronous Session

# Quantiles, if, skewness, samples, replicate

#

v<-rnorm(100,50,10) # create random norm dist, 100 obs, mean 50, sd 10

hist(v)

#

v<-rnorm(10000,50,10)

hist(v)

v<-rnorm(100,50,10)

#

q<-quantile(v,c(.1,.9)) # how much of your data lies below a certain value

q # 10% of the numbers in v are below X,,, 90% of the numbers in v are below xx

#

q<-quantile(v,c(.1,.5,.9)) # store quantile cutoff points in vector q

q

#

hist(v)

#

q[1]

q[2]

q[3]

labels(q)

#

q[1]

#

if(q[1]<35){

print("lower")

}

#

if(q[1]<45){

print("lower")

}

#

#

if(q[1]<35){

print("lower")

} else {

print("higher")

}

#

# upcoming homework skewness

#

#

# Intuitively, the skewness is a measure of symmetry. As a rule, negative skewness indicates that the

# mean of the data values is less than the median, and the data distribution is left-skewed. Positive

# skewness would indicate that the mean of the data values is larger than the median, and the data

# distribution is right-skewed.

#

library("moments")

skewness(v)

v1<-c(1, 10, 11, 12,14,13,16,10)

skewness(v1)

v1<-c(1:19)

skewness(v1)

#

####### sampling

# create a jar of cookies, 100 peanut butter, 100 chocolate chip

#

pb<-"peanut butter"

cc<-"choc chip"

#

v.pb<-replicate(100,pb) #create a pb vector with 100 peanut butter

v.cc<-replicate(100,cc) # create a cc vector with 100 choc chip

#

v.pb[1] # explore pb vector

v.pb[50]

v.pb[100]

#

# combine pb & cc vectors

#

jar<-c(v.pb, v.cc) # combine cookie vectors

length(jar) # validate # of cookeis in jar

# # have 200 objects in jar, half pb, half cc

#

length(jar[jar=='choc chip']) # count # of cc

length(jar[jar=='peanut butter']) # count # of pb

#

newjar<-jar[jar=='choc chip']

newjar

newjar<-jar[jar==cc]

newjar

#

## sample jar

# how to take samples from Data http://www.dummies.com/programming/r/how-to-take-samples-from-data-in-r/

# sample jar -> pb or cc equally likely to occur,,, ie coin toss

#

sample(jar,1) # You tell sample() to return 1 value, in the range 1:200 from (jar).

#

sample(jar,7) # You tell sample() to return 7 values, each in the range 1:200 from (jar).

sample(jar,7,replace=TRUE) # You tell sample() to return 7 values, each in the range 1:200 from (Jar) and put it back.

sample(jar,7,replace=FALSE)

sample(jar,1000,replace=TRUE) # You tell sample() to return 1000 values, each in the range 1:200 from (Jar) and put it back.

#

mySample<-sample(jar,7,replace=TRUE)

mySample

length(mySample[mySample==pb]) # how many pb's in the sample

length(mySample[mySample==pb])/length(mySample) # what is the % of pb in the sample

#

myTest<-function(jar,num){ # input a vector( jar) input size of sample( num )

mySample<-sample(jar,num,replace=TRUE)

len<-length(mySample[mySample==cc]) # choc chip

pct<-len/num\*100

return(pct)

}

myTest(jar,10)

myTest(jar,20)

myTest(jar,100)

myTest(jar,1000)

#

replicate(100,myTest(jar,10)) # get 100 samples of 10

#

replicate(100,myTest(jar,10000)) # get 100 samples of 10000

#

v<-replicate(100,myTest(jar,10000)) # get 100 samples of 10000

mean(v)

#

#

# grep example

#

string = c("apple", "pear", "apple", "orange", "apple") # initialize vector string

grep("apple", string) # positional occurance of apple as well as how many occurances of apple

length(grep("apple", string)) # count number of occurance of "apple"

length(grep("apple", string))/length(string) # % of apple in the string

#

#

#

#

# homework hints

#

sample(jar,7,replace=TRUE) # how many choc Chip ie 4

sample(jar,7,replace=TRUE) # how many choc Chip ie 7

sample(jar,7,replace=TRUE) # how many choc Chip ie 3

sample(jar,7,replace=TRUE) # how many choc Chip ie 5

sample(jar,7,replace=TRUE) # how many choc Chip ie 3

#

# take the mean of 4,7,3,5,3

mean(4,7,3,5,3) #4

#

# replicate the sample 5 times, calculated 1 mean

# repicate all of the above 10 times, (lines 134 through 141) I expect to get 10 means

#