QUIZ 3 (questions are in italics)

This quiz deals with distributions and samples taken from them.

Part I: Language

a) Write code to take a sample of size 100 from a

- Binomial distribution with parameters n = 20, pi = 0.5 . (This is the one where the language and symbols get confusing; so, read the help pages carefully, and feel free to tinker/experiment to figure out what you want.)

- Normal distribution with parameters mu = 10, sigma = 2 .

 rbinom(100, 20, 0.5) #0.5 for using rbinom, 0.5 for correct parameters specified

 rnorm(100, 10, 2) #0.5 for using rnorm, 0.5 for correct parameters specified

2pts total

Part II: N(mu,sigma)

b) Write code to make a 3x3 panel of figures, with each panel showing the histogram of a sample of size 500 taken from N(mu,sigma). Let the columns (or rows) have values mu = -3, 0, 3, and the rows (or columns) have sigma = 1, 2, 3. Limit the x-axis to the interval (-15, 15). Hint: The example in the prelab that shows how to overlay two histograms also shows how to limit the x-axis.

 n = 500

 par(mfrow=c(3,3)) #0.5pt for mfrow

 for(mu in c(-3,0,3)){

 for(sigma in c(1,2,3)){ #1pt for correct looping over mu and sigma values

 x = rnorm(n, mu, sigma)

 hist(x, xlim=c(-15,15)) #0.5pt for xlim

 }}

#-0.5 if the rnorm argument in the loops are incorrect

#Only give a maximum of 1pt if no for loop is used to get the 9 plots.

2pts total

c) Take a sample of size 1000 from N(2,3), transform the data to z=(x-mu)/sigma, and make a histogram of the resulting transformed data.

 n = 1000

 x = rnorm(n, 2, 3)

 z = (x - 2)/3 #1pt for correct standardization

 hist(z) # Note that the center and width are about 0 and 1, respectively.i

 # Later, we'll learn about a similar transformation where the mu

 # and sigma are replaced by quantities we compute from the sample itself.

#-0.5 if the rnorm argument is incorrect

1pt total

Part III: Percentile

d) Write code to

- take a sample of size 1000 from an exponential distribution with parameter 1,

- find and report the 0.1, 0.5, and 0.9 quantiles of that sample,

- find and report the 0.1, 0.5, and 0.9 quantiles of the distribution.

IMPORTANT: Start your code with set.seed(123)

 set.seed(123)

 x = rexp(1000,1) #1pt for rexp with correct parameters

 quantile(x, c(0.1, 0.5, 0.9)) # 0.09862473 0.73116671 2.27992175

#1pt for correct sample quantiles

 qexp(c(0.1, 0.5, 0.9), 1) # 0.1053605 0.6931472 2.3025851 (i.e., pretty close).

#1pt for correct qexp

3pts total

Part IV: sample size

e) Write code to make 10 (comparative) boxplots, each for samples of size 100, 200, 300, ..., 1000, taken from N(2,3).

 x = matrix(nrow=1000, ncol=10) # Allocate space for

#0.5pt for creating spaces to save results

 for(i in 1:10){

 n = i\*100 #0.5pt for correctly looping over the different sample sizes

 x[1:n, i] = rnorm(n, 2, 3) # the 1:n is important.

#0.5pt for correctly saving results from each run, particularly with 1:n

 }

 boxplot(x) #0.5pt for creating correct comparative boxplot

 boxplot(x, range=0) # FYI: Since one student asked about outliers.

 # range=0 treats them as part of the data.

#only give 1 pt if creating boxplots within the loops but with correct sequence of sample sizes

2pts total

Morals:

- Make sure you're clear what it means to "take a sample from a pop/dist."

- The mathematical notion of standardization, applies mto the data/sample as well.

- Quantiles/Percentiles are the basis of comparing samples and distributions.

- The width/spread of data does NOT increase with sample size.