Programming with R Test #2

Instructions

R markdown is a plain-text file format for integrating text and R code, and creating transparent, reproducible and interactive reports. An R markdown file (.Rmd) contains metadata, markdown and R code "chunks", and can be "knit" into numerous output types. Answer the test questions by adding R code to the fenced code areas below each item. Once completed, you will "knit" and submit the resulting .html file, as well the .Rmd file. The .html will include your R code *and* the output. The .html file will be graded and returned with comments as a .pdf document.

Before proceeding, look to the top of the .Rmd for the (YAML) metadata block, where the *title* and *output* are given. Please change *title* from 'Programming with R Test #2' to your name, with the format 'lastName_firstName.'

If you encounter issues knitting the .html, please send an email via Canvas to your TA.

Each code chunk is delineated by six (6) backticks; three (3) at the start and three (3) at the end. After the opening ticks, arguments are passed to the code chunk and in curly brackets. **Please do not add or remove backticks, or modify the arguments or values inside the curly brackets**.

Depending on the problem, grading will be based on: 1) the correct result, 2) coding efficiency and 3) graphical presentation features (labeling, colors, size, legibility, etc). I will be looking for well-rendered displays. Do not print and display the contents of vectors or data frames unless requested by the problem. You should be able to display each solution in fewer than ten lines of code.

Submit both the .Rmd and .html files for grading.

Test Items (50 points total)

(1) R has probability functions available for use (see Davies, Chapter 16, and Kabacoff, Section 5.2.3). Using one distribution to approximate another is not uncommon.

(1)(a) (6 points) The normal distribution may be used to approximate the binomial distribution if np $>$ 5 and np(1-p) $>$ 5. Find the following binomial probabilities using *dbinom()* and *pbinom()* with a probability, p = 0.5, and n = 100. Then, estimate the same probabilities using the normal approximation **with continuity correction** and *pnorm()*.

(i) The probability of exactly 50 successes.

(ii) The probability of fewer than 40 successes.

(iii) The probability of 60 or more successes.

(1)(b) (4 points) With n = 100 and p = 0.02, use the binomial probabilities from *dbinom()* to calculate the expected value and variance for this binomial distribution using the general formula for mean and variance of a discrete distribution (To do this, you will need to use integer values from 0 to 100 as binomial outcomes along with the corresponding binomial probability). Calculate the same using the formulae np and np(1-p).

```
n <- 100
p <- 0.02

n * p           # expected value
```

```
## [1] 2
```

```r
n * p * (1 - p) # variance
```

```
## [1] 1.96
```

---

(2) A recurring problem in statistics is the identification of outliers. This problem involves plotting data to display outliers, and then classiying them.

(2)(a) (5 points) Generate a random sample, "x", of 100 values using *set.seed(123)* and *rexp(n = 100, rate = 1)*. Do not change this number. If you must draw another sample, start the process with *set.seed(123)* to maintain comparability with the answer sheet. Present "x" in side-by-side box- and QQ-plots, using *boxplot()* and *qqnorm()*, *qqline()*. Use *boxplot.stats()* and/or logical statements to identify the extreme outliers, if any.

(2)(b) (5 points) Transform the random sample, "x", generated in (a), to form a different variable, designated "y", using the Box-Cox Transformation: y = 3*(xˆ(1/3)) - 1). Display the values for "y" as in (a) and identify outliers similarly.

---

(3) Performing hypothesis tests using random samples is fundamental to statistical inference. The first part of this problem involves comparing two different diets. Using "ChickWeight" data available in the base R, "datasets" package, execute the following code to prepare a data frame for analysis.

```r
# load "ChickWeight" dataset
data(ChickWeight)

# Create T | F vector indicating observations with Time == 21 and Diet == "1" OR "3"
index <- ChickWeight$Time == 21 & (ChickWeight$Diet == "1" | ChickWeight$Diet == "3")

# Create data frame, "result," with the weight and Diet of those observations with "TRUE" "index"" valu
result <- subset(ChickWeight[index, ], select = c(weight, Diet))

# Encode "Diet" as a factor
result$Diet <- factor(result$Diet)
str(result)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   26 obs. of  2 variables
##  $ weight: num  205 215 202 157 223 157 305 98 124 175 ...
##  $ Diet  : Factor w/ 2 levels "1","3": 1 1 1 1 1 1 1 1 1 1 ...
```

The data frame, "result", will have chick weights for two diets, identified as diet "1" and "3". Use the data frame, "result," to complete the following item.

(3)(a) (4 points) Use the "weight" data for the two diets to test the null hypothesis of equal population weights for the two diets. Test at the 95% confidence level with a two-sided t-test. This can be done using *t.test()* in R. Assume equal variances. Display the results.

Working with paired data is another common statistical activity. The "ChickWeight" data will be used to illustrate how the weight gain from week 20 to 21 may be analyzed. Use the following code to prepare pre- and post-data from Diet == "3" for analysis.

```r
# load "ChickWeight" dataset
data(ChickWeight)

# Create T | F vector indicating observations with Diet == "3"
index <- ChickWeight$Diet == "3"

# Create vector of "weight" for observations where Diet == "3" and Time == 20
pre <- subset(ChickWeight[index, ], Time == 20, select = weight)$weight

# Create vector of "weight" for observations where Diet == "3" and Time == 21
post <- subset(ChickWeight[index, ], Time == 21, select = weight)$weight
```

(3)(b) (6 points) Conduct a paired t-test and construct a two-sided, 95% confidence interval for the average weight gain from week 20 to week 21. **Do not use *t.test()*.** Write the code for determination of the confidence interval endpoints. Present the resulting interval.

---

(4) Statistical inference depends on using a sampling distribution for a statistic in order to make confidence statements about unknown population parameters. The Central Limit Theorem is used to justify use of the normal distribution as a sampling distribution for statistical inference. Using Nile River flow data from 1871 to 1970, this problem demonstrates sampling distribution convergence to normality. Use the code below to prepare the data.

```r
data(Nile)
```

(4)(a) (3 points) Using Nile River flow data and the "moments" package, calculate skewness and kurtosis. Present side-by-side displays using *qqnorm()*, *qqline()* and *boxplot()*; i.e *par(mfrow = c(1, 2))*. Add features to these displays as you choose.

```r
library(moments)
```

(4)(b) (3 points) Using *set.seed(124)* and the Nile data, generate 1000 random samples of size n = 16, with replacement. For each sample drawn, calculate and store the sample mean. This will require a for-loop and use of the *sample()* function. Label the resulting 1000 mean values as "sample1". **Repeat these steps using *set.seed(127)* - a different "seed" - and samples of size n = 64.** Label these 1000 mean values as "sample2". Compute and present the mean value, sample standard deviation and sample variance for "sample1" and "sample2".

(4)(c) (4 points) Using "sample1" and "sample2", present separate histograms with the normal density curve superimposed (use *par(mfrow = c(2, 1))*). To prepare comparable histograms it will be necessary to use "freq = FALSE" and to maintain the same x-axis with "xlim = c(750, 1050)", and the same y-axis with "ylim = c(0, 0.025)." **To superimpose separate density functions, you will need to use the mean and standard deviation for each "sample" - each histogram - separately.**

---

(5) This problem deals with 2 x 2 contingency table analysis. This is an example of categorical data analysis (see Kabacoff, pp. 145-151). The method shown in this problem can be used to screen data for potential predictors that may be used in building a model.

The "Seatbelts" dataset contains monthly road casualties in Great Britain, 1969 to 1984. Use the code below to organize the data and generate two factor variables: "killed" and "month". These variables will be used for contingency table analysis.

```
data(Seatbelts)
Seatbelts <- as.data.frame(Seatbelts)

Seatbelts$Month <- seq(from = 1, to = nrow(Seatbelts))
Seatbelts <- subset(Seatbelts, select = c(DriversKilled, Month))
summary(Seatbelts)
```

```
##  DriversKilled       Month
##  Min.   : 60.0   Min.   :  1.00
##  1st Qu.:104.8   1st Qu.: 48.75
##  Median :118.5   Median : 96.50
##  Mean   :122.8   Mean   : 96.50
##  3rd Qu.:138.0   3rd Qu.:144.25
##  Max.   :198.0   Max.   :192.00
```

```
killed <- factor(Seatbelts$DriversKilled > 118.5, labels = c("below", "above"))

month <- factor(Seatbelts$Month > 96.5, labels = c("below", "above"))
```

(5)(a) (3 points) Using "Seatbelts," generate a scatterplot of the variables DriversKilled versus Month. This is a time series, and Seatbelts$Month should be on the horizontal axis. Show vertical and horizontal lines to indicate the median of each variable. Label as desired.

(5)(b) (2 points) A chi-square test of independence will be used (see Black, Section 16.2) to test the null hypothesis that the factor variables, "killed" and "month", are independent. Use *table()* to generate a 2 x 2 contingency table showing the fatality count classified by "killed" and "month". Use the **uncorrected** *chisq.test()* to test the null hypothesis that "killed" and "month" are independent at the 95% confidence level. Present these results.

(5)(c) (5 points) Write a function that computes the uncorrected Pearson Chi-squared statistic based on the a 2 x 2 contingency table with margins added (check Davies, Section 11.1.1, pp. 216-219, and Kabacoff, Section 20.1.3, pp. 473-474). Add margins to the contingency table from (b) using the function *addmargins()*. Submit this augmented table to the function you have written. Compare the result with (b). Your function should duplicate and output the X-squared value (chi-squared) and *p*-value. Present both results.

The statements shown below calculate the expected value for each cell in an augmented contingency table with margins added. Using these statements, the Pearson Chi-square statistic may be calculated. Other approaches are acceptable.

e11 <- x[3, 1] * x[1, 3] / x[3, 3], e12 <- x[3, 2] * x[1, 3] / x[3, 3], e21 <- x[3, 1] * x[2, 3] / x[3, 3], e22 <- x[3, 2] * x[2, 3] / x[3, 3]

```
# Write function for computing uncorrected Pearson Chi-squared statistic and associated p-value
```