CIND 123 Winter 2018 - Assignment #4

Write your name here

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.

Use RStudio for this assignment. Edit the file `assignment-4.Rmd` and insert your R code where wherever you see the string "INSERT YOUR ANSWER HERE"

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

Sample Question and Solution

Use `seq()` to create the vector $(2, 4, 6, \ldots, 20)$.

```
#Insert your code here.
seq(2,20,by = 2)
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

In this assignment, questions 1 - 4 make use of data that is provided by the `mosaic` package. (install mosaic package and load KidsFeet using data(KidsFeet)).

```
#install.packages('mosaic')
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 3.4.4
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
## Warning: package 'ggformula' was built under R version 3.4.4
```

```
## Loading required package: ggplot2
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Warning: package 'mosaicData' was built under R version 3.4.4
```

```
## Loading required package: Matrix
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
data(KidsFeet)
```

Question 1 - 30%

This question makes use of package "plm", and load Crime dataset as following:

```
#install.packages("plm")
library(plm)
```

```
## Warning: package 'plm' was built under R version 3.4.4

## Loading required package: Formula

##
## Attaching package: 'plm'

## The following object is masked from 'package:mosaic':
##
##     r.squared

## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
data(Crime)
```

a) Display the first 10 rows of crime and make note of all the variables.

```
head(Crime, 10)
```

```
##    county year    crmrte   prbarr  prbconv  prbpris avgsen     polpc
## 1       1   81 0.0398849 0.289696 0.402062 0.472222   5.61 0.0017868
## 2       1   82 0.0383449 0.338111 0.433005 0.506993   5.59 0.0017666
## 3       1   83 0.0303048 0.330449 0.525703 0.479705   5.80 0.0018358
```

```
## 4       1   84 0.0347259 0.362525 0.604706 0.520104   6.89 0.0018859
## 5       1   85 0.0365730 0.325395 0.578723 0.497059   6.55 0.0019244
## 6       1   86 0.0347524 0.326062 0.512324 0.439863   6.90 0.0018952
## 7       1   87 0.0356036 0.298270 0.527596 0.436170   6.71 0.0018279
## 8       3   81 0.0163921 0.202899 0.869048 0.465753   8.45 0.0005939
## 9       3   82 0.0190651 0.162218 0.772152 0.377049   5.71 0.0007047
## 10      3   83 0.0151492 0.181586 1.028170 0.438356   8.69 0.0006587
##     density    taxpc  region smsa   pctmin     wcon      wtuc     wtrd
## 1  2.307159 25.69763 central   no 20.21870 206.4803  333.6209 182.3330
## 2  2.330254 24.87425 central   no 20.21870 212.7542  369.2964 189.5414
## 3  2.341801 26.45144 central   no 20.21870 219.7802 1394.8030 196.6395
## 4  2.346420 26.84235 central   no 20.21870 223.4238  398.8604 200.5629
## 5  2.364896 28.14034 central   no 20.21870 243.7562  358.7830 206.8827
## 6  2.385681 29.74098 central   no 20.21870 257.9139  369.5465 218.5165
## 7  2.422633 30.99368 central   no 20.21870 281.4259  408.7245 221.2701
## 8  0.976834 14.56088 central   no  7.91632 188.7683  292.6422 151.4234
## 9  0.992278 35.64073 central   no  7.91632 186.9658  345.7217 156.8826
## 10 1.003861 19.26188 central   no  7.91632 193.5983  604.9115 157.1295
##        wfir     wser   wmfg   wfed   wsta   wloc       mix   pctymle
## 1  272.4492 215.7335 229.12 409.37 236.24 231.47 0.0999179 0.0876968
## 2  300.8788 231.5767 240.33 419.70 253.88 236.79 0.1030491 0.0863767
## 3  309.9696 240.1568 269.70 438.85 250.36 248.58 0.0806787 0.0850909
## 4  350.0863 252.4477 281.74 459.17 261.93 264.38 0.0785035 0.0838333
## 5  383.0707 261.0861 298.88 490.43 281.44 288.58 0.0932486 0.0823065
## 6  409.8842 269.6129 322.65 478.67 286.91 306.70 0.0973228 0.0800806
## 7  453.1722 274.1775 334.54 477.58 292.09 311.91 0.0801688 0.0778710
## 8  202.4292 191.3742 210.75 381.72 247.38 213.17 0.0561224 0.0870046
## 9  225.0409 208.8190 217.77 386.42 374.07 219.18 0.0473118 0.0864722
## 10 248.1390 219.0847 236.64 382.65 268.90 223.06 0.0596206 0.0859426
```

```
str(Crime)
```

```
## 'data.frame':    630 obs. of  24 variables:
##  $ county : int  1 1 1 1 1 1 1 3 3 3 ...
##  $ year   : int  81 82 83 84 85 86 87 81 82 83 ...
##  $ crmrte : num  0.0399 0.0383 0.0303 0.0347 0.0366 ...
##  $ prbarr : num  0.29 0.338 0.33 0.363 0.325 ...
##  $ prbconv: num  0.402 0.433 0.526 0.605 0.579 ...
##  $ prbpris: num  0.472 0.507 0.48 0.52 0.497 ...
##  $ avgsen : num  5.61 5.59 5.8 6.89 6.55 6.9 6.71 8.45 5.71 8.69 ...
##  $ polpc  : num  0.00179 0.00177 0.00184 0.00189 0.00192 ...
##  $ density: num  2.31 2.33 2.34 2.35 2.36 ...
##  $ taxpc  : num  25.7 24.9 26.5 26.8 28.1 ...
##  $ region : Factor w/ 3 levels "other","west",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ smsa   : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ pctmin : num  20.2 20.2 20.2 20.2 20.2 ...
##  $ wcon   : num  206 213 220 223 244 ...
##  $ wtuc   : num  334 369 1395 399 359 ...
##  $ wtrd   : num  182 190 197 201 207 ...
##  $ wfir   : num  272 301 310 350 383 ...
##  $ wser   : num  216 232 240 252 261 ...
##  $ wmfg   : num  229 240 270 282 299 ...
##  $ wfed   : num  409 420 439 459 490 ...
##  $ wsta   : num  236 254 250 262 281 ...
##  $ wloc   : num  231 237 249 264 289 ...
```

```
##  $ mix    : num  0.0999 0.103 0.0807 0.0785 0.0932 ...
##  $ pctymle: num  0.0877 0.0864 0.0851 0.0838 0.0823 ...
```

b) Calculate the mean,variance and standard deviation of tax revenue per capita (taxpc) by omitting the missing values, if any.

```
# require(Hmisc)
# describe(na.omit(Crime$taxpc))
# describe(Crime$taxpc)

mean(na.omit(Crime$taxpc))
```

```
## [1] 30.23919
```

```
sd(na.omit(Crime$taxpc))
```

```
## [1] 11.4547
```

```
var(na.omit(Crime$taxpc))
```

```
## [1] 131.21
```

c) Use `density` and `smsa` to predict tax per capita and build a univariate linear regression model, display a summary of your model indicating Residuals, Coefficients..etc. What can you say about your model?

```
#Insert your code here
```

d)Based on the output of your model, write the equations based on the intercept and factors of `smsa` when `density` is set to 2.4, and compare the result with `predict()` function. Hint: Explore `predict()` function

```
#Insert your answer here
```

e)Find Pearson correlation between tax per capita and density. Please comment on the result with a sentence.

```
#Insert your code here
```

f)Write the correlation matrix of the variables: avgsen, polpc, density, taxpc. Hint: Explore the variables by ?Crime. Comment on the result with a sentence.

```
#Insert your code here
```

Question 2 -30%

a) First and second midterm grades of some students are given as c(85,76,78,88,90,95,42,31) and c(55,76,48,58,80,75,32,22). Set R variables `first` and `second` respectively.

b) Apply the lm() function to observe the relationship between the first and the second midterm grades. Hint: Second midterm is the response variable.

```
#Insert your code here
```

c) Find the second midterm grade of a student given that his/her first midterm grade is 72. Print the result by using print() function.

```
#Insert your code here
```

Question 3 - 40%

π appears in the formula for the standard normal distribution, the most important probability distribution in statistics. Why not give it a try to calculate π using statistics! In fact, you'll use a simulation technique

called the *Monte Carlo Method*.

Recall that the area of a circle of radius r is $A = \pi r^2$. Therefore the area of a circle of radius 1, aka a *unit circle*, is π. You'll compute an approximation to the area of this circle using the Monte Carlo Method.

a) The Monte Carlo Method uses random numbers to simulate some process. Here the process is throwing darts at a square. Assume the darts are uniformly distributed over the square. Imagine a unit circle enclosed by a square whose sides are of length 2. Set an R variable `area.square` to be the area of a square whose sides are of length 2.

```
#Insert your code here
```

b) The points of the square can be given x-y coordinates. Let both x and y range from -1 to +1 so that the square is centred on the origin of the coordinate system. Throw some darts at the square by generating random numeric vectors x and y, each of length `N = 10,000`. Set R variables `x` and `y` each to be uniformly distributed random numbers in the range -1 to +1. (hint: runif() generates random number for the uniform distribution)

```
#Insert your code here
```

c) Now count how many darts landed inside the unit circle. Recall that a point is inside the unit circle when $x^2 + y^2 < 1$. Save the result of successfull hits in a variable named hit. (hint: a for loop over the length of x and y is one option to reach hit)

```
#Insert your code here
```

d) The probability that a dart hits inside the circle is proportional to the ratio of the area of the circle to the area of the square. Use this fact to calculate an approximation to π and print the result

```
#Insert your code here
```

Wow you got the first estimate for `pi` π, congratulations you have completed the first run of the Monte Carlo simulation. If there is further intrest put all the above logic in a function, and call it 50 times store the results in a vector called pi then take the mean of pi vector.