CIND 123 Summer 2019 - Assignment #3

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 A3-S19-Q 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 .

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

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

##Question 1

Install the marketing dataset on your computer using the command install.packages("datarium"). Then load the datarium package into your session using the following command. Understand the dataset by using ??marketing command.

library(datarium)

## Warning: package 'datarium' was built under R version 3.5.3

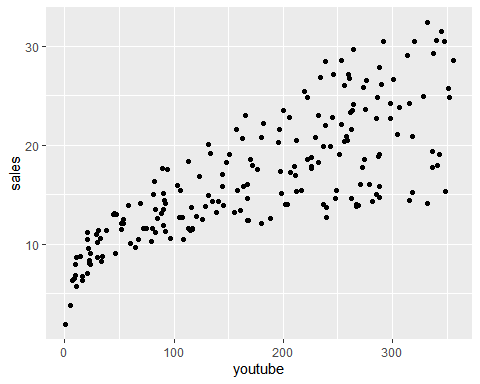
attach(marketing)

1. Plot the advertising budget of Youtube against Sales. Comment on their relationship. Hint: You may use the ggplot() function from ggplot2 package.

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.3

ggplot(marketing) + aes(youtube , sales) + geom\_point()



#there is a strong correlation between the cost of sales and amount of money spent on the advertisment.

1. Find the correlation between advertising budget of Youtube against Sales. Comment on the output. Does it match your intuition from part (a).

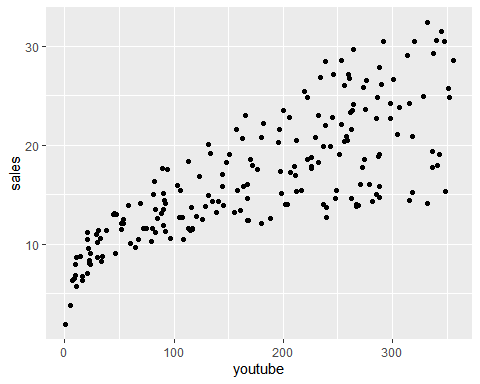
cor(marketing$youtube, marketing$sales)

## [1] 0.7822244

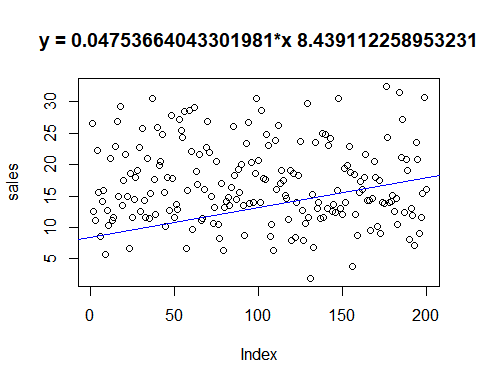
#The advertisment budget and sales have a strong correlation, it matched my intuition from everyday life and the scatter plot

1. Plot the Sales as a function of Youtube variable using a scatterplot, and graph the least-square line on the same plot.

library(ggplot2)  
ggplot(marketing) + aes(youtube , sales) + geom\_point()



reg <- lm(sales ~ youtube, data = marketing)  
coeff=coefficients(reg)  
eq = paste0("y = ", coeff[2],1, "\*x ", coeff[1],1)  
  
plot(sales, main=eq)  
abline(reg, col="blue")



1. Use the regression line to predict the Sales amount when Youtube budget is $69K.

linear\_model <- lm(sales ~ youtube, data = marketing)  
linear\_model

##   
## Call:  
## lm(formula = sales ~ youtube, data = marketing)  
##   
## Coefficients:  
## (Intercept) youtube   
## 8.43911 0.04754

marketing

## youtube facebook newspaper sales  
## 1 276.12 45.36 83.04 26.52  
## 2 53.40 47.16 54.12 12.48  
## 3 20.64 55.08 83.16 11.16  
## 4 181.80 49.56 70.20 22.20  
## 5 216.96 12.96 70.08 15.48  
## 6 10.44 58.68 90.00 8.64  
## 7 69.00 39.36 28.20 14.16  
## 8 144.24 23.52 13.92 15.84  
## 9 10.32 2.52 1.20 5.76  
## 10 239.76 3.12 25.44 12.72  
## 11 79.32 6.96 29.04 10.32  
## 12 257.64 28.80 4.80 20.88  
## 13 28.56 42.12 79.08 11.04  
## 14 117.00 9.12 8.64 11.64  
## 15 244.92 39.48 55.20 22.80  
## 16 234.48 57.24 63.48 26.88  
## 17 81.36 43.92 136.80 15.00  
## 18 337.68 47.52 66.96 29.28  
## 19 83.04 24.60 21.96 13.56  
## 20 176.76 28.68 22.92 17.52  
## 21 262.08 33.24 64.08 21.60  
## 22 284.88 6.12 28.20 15.00  
## 23 15.84 19.08 59.52 6.72  
## 24 273.96 20.28 31.44 18.60  
## 25 74.76 15.12 21.96 11.64  
## 26 315.48 4.20 23.40 14.40  
## 27 171.48 35.16 15.12 18.00  
## 28 288.12 20.04 27.48 19.08  
## 29 298.56 32.52 27.48 22.68  
## 30 84.72 19.20 48.96 12.60  
## 31 351.48 33.96 51.84 25.68  
## 32 135.48 20.88 46.32 14.28  
## 33 116.64 1.80 36.00 11.52  
## 34 318.72 24.00 0.36 20.88  
## 35 114.84 1.68 8.88 11.40  
## 36 348.84 4.92 10.20 15.36  
## 37 320.28 52.56 6.00 30.48  
## 38 89.64 59.28 54.84 17.64  
## 39 51.72 32.04 42.12 12.12  
## 40 273.60 45.24 38.40 25.80  
## 41 243.00 26.76 37.92 19.92  
## 42 212.40 40.08 46.44 20.52  
## 43 352.32 33.24 2.16 24.84  
## 44 248.28 10.08 31.68 15.48  
## 45 30.12 30.84 51.96 10.20  
## 46 210.12 27.00 37.80 17.88  
## 47 107.64 11.88 42.84 12.72  
## 48 287.88 49.80 22.20 27.84  
## 49 272.64 18.96 59.88 17.76  
## 50 80.28 14.04 44.16 11.64  
## 51 239.76 3.72 41.52 13.68  
## 52 120.48 11.52 4.32 12.84  
## 53 259.68 50.04 47.52 27.12  
## 54 219.12 55.44 70.44 25.44  
## 55 315.24 34.56 19.08 24.24  
## 56 238.68 59.28 72.00 28.44  
## 57 8.76 33.72 49.68 6.60  
## 58 163.44 23.04 19.92 15.84  
## 59 252.96 59.52 45.24 28.56  
## 60 252.84 35.40 11.16 22.08  
## 61 64.20 2.40 25.68 9.72  
## 62 313.56 51.24 65.64 29.04  
## 63 287.16 18.60 32.76 18.84  
## 64 123.24 35.52 10.08 16.80  
## 65 157.32 51.36 34.68 21.60  
## 66 82.80 11.16 1.08 11.16  
## 67 37.80 29.52 2.64 11.40  
## 68 167.16 17.40 12.24 16.08  
## 69 284.88 33.00 13.20 22.68  
## 70 260.16 52.68 32.64 26.76  
## 71 238.92 36.72 46.44 21.96  
## 72 131.76 17.16 38.04 14.88  
## 73 32.16 39.60 23.16 10.56  
## 74 155.28 6.84 37.56 13.20  
## 75 256.08 29.52 15.72 20.40  
## 76 20.28 52.44 107.28 10.44  
## 77 33.00 1.92 24.84 8.28  
## 78 144.60 34.20 17.04 17.04  
## 79 6.48 35.88 11.28 6.36  
## 80 139.20 9.24 27.72 13.20  
## 81 91.68 32.04 26.76 14.16  
## 82 287.76 4.92 44.28 14.76  
## 83 90.36 24.36 39.00 13.56  
## 84 82.08 53.40 42.72 16.32  
## 85 256.20 51.60 40.56 26.04  
## 86 231.84 22.08 78.84 18.24  
## 87 91.56 33.00 19.20 14.40  
## 88 132.84 48.72 75.84 19.20  
## 89 105.96 30.60 88.08 15.48  
## 90 131.76 57.36 61.68 20.04  
## 91 161.16 5.88 11.16 13.44  
## 92 34.32 1.80 39.60 8.76  
## 93 261.24 40.20 70.80 23.28  
## 94 301.08 43.80 86.76 26.64  
## 95 128.88 16.80 13.08 13.80  
## 96 195.96 37.92 63.48 20.28  
## 97 237.12 4.20 7.08 14.04  
## 98 221.88 25.20 26.40 18.60  
## 99 347.64 50.76 61.44 30.48  
## 100 162.24 50.04 55.08 20.64  
## 101 266.88 5.16 59.76 14.04  
## 102 355.68 43.56 121.08 28.56  
## 103 336.24 12.12 25.68 17.76  
## 104 225.48 20.64 21.48 17.64  
## 105 285.84 41.16 6.36 24.84  
## 106 165.48 55.68 70.80 23.04  
## 107 30.00 13.20 35.64 8.64  
## 108 108.48 0.36 27.84 10.44  
## 109 15.72 0.48 30.72 6.36  
## 110 306.48 32.28 6.60 23.76  
## 111 270.96 9.84 67.80 16.08  
## 112 290.04 45.60 27.84 26.16  
## 113 210.84 18.48 2.88 16.92  
## 114 251.52 24.72 12.84 19.08  
## 115 93.84 56.16 41.40 17.52  
## 116 90.12 42.00 63.24 15.12  
## 117 167.04 17.16 30.72 14.64  
## 118 91.68 0.96 17.76 11.28  
## 119 150.84 44.28 95.04 19.08  
## 120 23.28 19.20 26.76 7.92  
## 121 169.56 32.16 55.44 18.60  
## 122 22.56 26.04 60.48 8.40  
## 123 268.80 2.88 18.72 13.92  
## 124 147.72 41.52 14.88 18.24  
## 125 275.40 38.76 89.04 23.64  
## 126 104.64 14.16 31.08 12.72  
## 127 9.36 46.68 60.72 7.92  
## 128 96.24 0.00 11.04 10.56  
## 129 264.36 58.80 3.84 29.64  
## 130 71.52 14.40 51.72 11.64  
## 131 0.84 47.52 10.44 1.92  
## 132 318.24 3.48 51.60 15.24  
## 133 10.08 32.64 2.52 6.84  
## 134 263.76 40.20 54.12 23.52  
## 135 44.28 46.32 78.72 12.96  
## 136 57.96 56.40 10.20 13.92  
## 137 30.72 46.80 11.16 11.40  
## 138 328.44 34.68 71.64 24.96  
## 139 51.60 31.08 24.60 11.52  
## 140 221.88 52.68 2.04 24.84  
## 141 88.08 20.40 15.48 13.08  
## 142 232.44 42.48 90.72 23.04  
## 143 264.60 39.84 45.48 24.12  
## 144 125.52 6.84 41.28 12.48  
## 145 115.44 17.76 46.68 13.68  
## 146 168.36 2.28 10.80 12.36  
## 147 288.12 8.76 10.44 15.84  
## 148 291.84 58.80 53.16 30.48  
## 149 45.60 48.36 14.28 13.08  
## 150 53.64 30.96 24.72 12.12  
## 151 336.84 16.68 44.40 19.32  
## 152 145.20 10.08 58.44 13.92  
## 153 237.12 27.96 17.04 19.92  
## 154 205.56 47.64 45.24 22.80  
## 155 225.36 25.32 11.40 18.72  
## 156 4.92 13.92 6.84 3.84  
## 157 112.68 52.20 60.60 18.36  
## 158 179.76 1.56 29.16 12.12  
## 159 14.04 44.28 54.24 8.76  
## 160 158.04 22.08 41.52 15.48  
## 161 207.00 21.72 36.84 17.28  
## 162 102.84 42.96 59.16 15.96  
## 163 226.08 21.72 30.72 17.88  
## 164 196.20 44.16 8.88 21.60  
## 165 140.64 17.64 6.48 14.28  
## 166 281.40 4.08 101.76 14.28  
## 167 21.48 45.12 25.92 9.60  
## 168 248.16 6.24 23.28 14.64  
## 169 258.48 28.32 69.12 20.52  
## 170 341.16 12.72 7.68 18.00  
## 171 60.00 13.92 22.08 10.08  
## 172 197.40 25.08 56.88 17.40  
## 173 23.52 24.12 20.40 9.12  
## 174 202.08 8.52 15.36 14.04  
## 175 266.88 4.08 15.72 13.80  
## 176 332.28 58.68 50.16 32.40  
## 177 298.08 36.24 24.36 24.24  
## 178 204.24 9.36 42.24 14.04  
## 179 332.04 2.76 28.44 14.16  
## 180 198.72 12.00 21.12 15.12  
## 181 187.92 3.12 9.96 12.60  
## 182 262.20 6.48 32.88 14.64  
## 183 67.44 6.84 35.64 10.44  
## 184 345.12 51.60 86.16 31.44  
## 185 304.56 25.56 36.00 21.12  
## 186 246.00 54.12 23.52 27.12  
## 187 167.40 2.52 31.92 12.36  
## 188 229.32 34.44 21.84 20.76  
## 189 343.20 16.68 4.44 19.08  
## 190 22.44 14.52 28.08 8.04  
## 191 47.40 49.32 6.96 12.96  
## 192 90.60 12.96 7.20 11.88  
## 193 20.64 4.92 37.92 7.08  
## 194 200.16 50.40 4.32 23.52  
## 195 179.64 42.72 7.20 20.76  
## 196 45.84 4.44 16.56 9.12  
## 197 113.04 5.88 9.72 11.64  
## 198 212.40 11.16 7.68 15.36  
## 199 340.32 50.40 79.44 30.60  
## 200 278.52 10.32 10.44 16.08

x <- 69  
predicted\_amount = 0.04754 \* x + 8.43911

1. Use youtube and facebook variables to build a linear regression model to predict sales Display a summary of your model indicating Residuals, Coefficients, …, etc. What conclusion can you draw from this summary?

linear\_model2 <- lm(sales ~ youtube + facebook, data = marketing)  
linear\_model2

##   
## Call:  
## lm(formula = sales ~ youtube + facebook, data = marketing)  
##   
## Coefficients:  
## (Intercept) youtube facebook   
## 3.50532 0.04575 0.18799

a <- 69  
b <- 39.36  
prediction1 = 0.04575 \* a + 0.18799\*b + 3.50532  
summary(linear\_model2)

##   
## Call:  
## lm(formula = sales ~ youtube + facebook, data = marketing)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.5572 -1.0502 0.2906 1.4049 3.3994   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.50532 0.35339 9.919 <2e-16 \*\*\*  
## youtube 0.04575 0.00139 32.909 <2e-16 \*\*\*  
## facebook 0.18799 0.00804 23.382 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.018 on 197 degrees of freedom  
## Multiple R-squared: 0.8972, Adjusted R-squared: 0.8962   
## F-statistic: 859.6 on 2 and 197 DF, p-value: < 2.2e-16

#The summary gave us the coefficients and the intercept. Standard deviation.

1. Use the regression line to predict the Sales amount when youtube budget is $69K and facebook is $39.36K.

linear\_model2 <- lm(sales ~ youtube + facebook, data = marketing)  
linear\_model2

##   
## Call:  
## lm(formula = sales ~ youtube + facebook, data = marketing)  
##   
## Coefficients:  
## (Intercept) youtube facebook   
## 3.50532 0.04575 0.18799

a <- 69  
b <- 39.36  
prediction1 = 0.04575 \* a + 0.18799\*b + 3.50532  
prediction1

## [1] 14.06136

1. What is the difference between the output in (f) and the output in (d)

#f uses both, facebook and youtube data to predict the sales making it a lot more accurate. d only uses the youtube data and is not accurate.

1. Display the correlation matrix of the variables: youtube, facebook, newspaper and sales. What conclusion can you draw?

data("marketing")  
cor(marketing)

## youtube facebook newspaper sales  
## youtube 1.00000000 0.05480866 0.05664787 0.7822244  
## facebook 0.05480866 1.00000000 0.35410375 0.5762226  
## newspaper 0.05664787 0.35410375 1.00000000 0.2282990  
## sales 0.78222442 0.57622257 0.22829903 1.0000000

1. In your opinion, which statistical test should be used to discuss the relationship between youtube and sales? Hint: Review the differnce between Pearson and Spearman tests.

#I believe the Pearson test would be the best for the dataset. The pearson test is better for finding the correlation betwqeen two qauntitative attributes that follow a linear trend

## Question 2

This question makes use of package “ISwR”. Please load airquality dataset as following:

#install.packages("ISwR")  
library(ISwR)

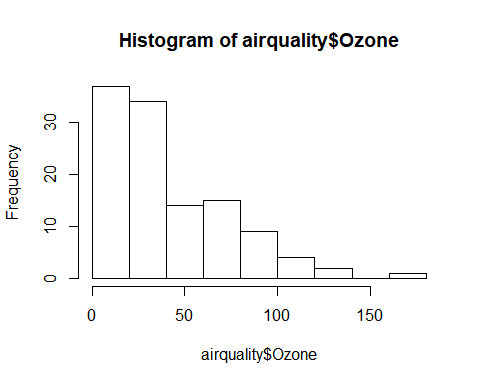
## Warning: package 'ISwR' was built under R version 3.5.3

data(airquality)  
str(airquality)

## 'data.frame': 153 obs. of 6 variables:  
## $ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...  
## $ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...  
## $ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...  
## $ Temp : int 67 72 74 62 56 66 65 59 61 69 ...  
## $ Month : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ Day : int 1 2 3 4 5 6 7 8 9 10 ...

1. Use a histogram to assess the normality of the Ozone variable, then explain why it does not appear normally distributed.

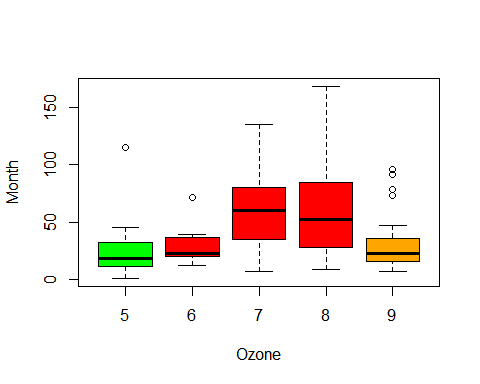
attach(airquality)  
hist(airquality$Ozone)



#The histogram is very skewed to the lft.

1. Create a boxplot that shows the distribution of Ozone in each month. Use different colors for each month.

boxplot(Ozone ~as.factor(Month),col=c("green",rep("red",3),"Orange"),xlab="Ozone", ylab="Month")



## Question 3

appears in the formula for the standard normal distribution, the most important probability distribution in statistics. Why not give it a try to calculate using statistics! In fact, you’ll use a simulation technique called the *Monte Carlo Method*.

Recall that the area of a circle of radius is . Therefore the area of a circle of radius 1, aka a *unit circle*, is . You’ll compute an approximation to the area of this circle using the Monte Carlo Method.

1. 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.

area.circle <- 2^2

1. 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)

simulations<- 100000   
perimeter <-1  
pintInCircle <- function(perimeter=1){  
 x<-runif(n=1, min = -perimeter, max = perimeter)  
 y <- runif(n=1, min= -perimeter, max = perimeter)  
 return(list(x=x,  
 y=y,  
 inCircle = x^2 + y^2 <= perimeter^2))  
  
   
}

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

simulation = function(long){  
 c = rep(0,long)  
 numberIn = 0  
 for(i in 1:long){  
 x = runif(2,-1,1)  
 if(sqrt(x[1]\*x[1] + x[2]\*x[2]) < 1){  
 numberIn = numberIn + 1  
 }  
 }  
 return(numberIn)  
   
   
}  
size = 10000  
res=simulation(size)

1. 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 and print the result

simulation.pi <- function(iterations = 1000000) {  
   
 x.pos <- runif(iterations, min=-1, max=1)  
 y.pos <- runif(iterations, min=-1, max=1)  
   
 draw.pos <- ifelse(x.pos^2 + y.pos^2 <= 1, TRUE, FALSE)  
 draws.in <- length(which(draw.pos == TRUE))  
   
 return(4\*(draws.in/iterations))  
   
   
}  
simulation.pi()

## [1] 3.142756

You got the first estimate for pi , congratulations you have completed the first run of the Monte Carlo simulation. If there is further interest 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.