CIND 123: Data Analytics Basic Methods: Assignment-3

Assignment 3 (10%)

Total 100 Marks

[Stephanie Boissonneault]

## Instructions

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. Complete the assignment by inserting your R code 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. You can embed an R code chunk like this:

Submit **both** the rmd and generated output files. Failing to submit both files will be subject to mark deduction.

## Sample Question and Solution

Use seq() to create the vector .

#INSERT YOUR ANSWER HERE.  
seq(2,20,by = 2)

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

## Question 1 [15 Pts]

1. [5 Pts] First and second midterm grades of some students are given as c(85,76,78,88,90,95,42,31,66) and c(55,66,48,58,80,75,32,22,39). Set R variables first and second respectively. Then find the least-squares line relating the second midterm to the first midterm.

* Does the assumption of a linear relationship appear to be reasonable in this case? Give reasons to your answer as a comment.

first <- c(85,76,78,88,90,95,42,31,66)  
second <- c(55,66,48,58,80,75,32,22,39)  
  
lsmodel <- lm(second ~ first)  
summary(lsmodel)

##   
## Call:  
## lm(formula = second ~ first)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.238 -7.747 1.753 4.383 13.318   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -4.1516 10.9987 -0.377 0.71702   
## first 0.7870 0.1461 5.389 0.00102 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.175 on 7 degrees of freedom  
## Multiple R-squared: 0.8058, Adjusted R-squared: 0.778   
## F-statistic: 29.04 on 1 and 7 DF, p-value: 0.001021

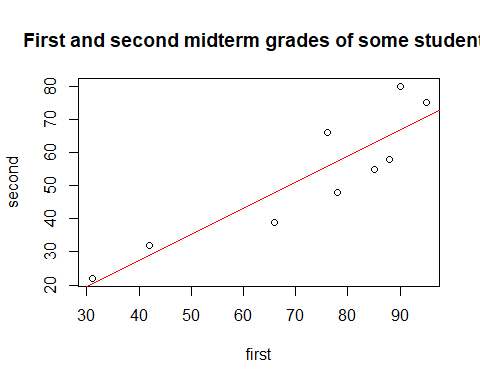
#The assumption of a linear relationship appears to be reasonable because the summary of the line of best fit has a p value = 0.001 indicating that the overall model is statistically significant. In addition, the adjusted R squared value is 0.778 (this is fairly high and close to 1, indicating that the independent variables of the model can well predict the variation of the predicted variables). Here the residuals range from -9 to 13 which is a fairly symmetrical distribution. This model demonstrates that the line of best fit has a y intercept of -4.15. For every increase in one mark form the first midterm grade, the second midterm grade increases by about 0.79 (this is the slope).

1. [5 Pts] Plot the second midterm as a function of the first midterm using a scatterplot and graph the least-square line in red color on the same plot.

print(plot(first, second, main = "First and second midterm grades of some students"))

## NULL

print(abline(lsmodel, col = "red"))



## NULL

1. [5 Pts] Use the regression line to predict the second midterm grades when the first midterm grades are 81 and 23.

first.81 <- data.frame(first=c(81))  
predict(lsmodel, first.81)

## 1   
## 59.59881

first.23 <- data.frame(first=c(23))  
predict(lsmodel, first.23)

## 1   
## 13.95039

## Question 2 [45 Pts]

This question makes use of package “plm”. Please load Crime dataset as follows:

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

## Warning: package 'plm' was built under R version 4.3.2

data(Crime)

1. [5 Pts] Display the first 8 rows of ‘crime’ data and display the names of all the variables, the number of variables, then display a descriptive summary of each variable.

#Display first 8 rows  
print(head(Crime, 8))

## county year crmrte prbarr prbconv prbpris avgsen polpc density  
## 1 1 81 0.0398849 0.289696 0.402062 0.472222 5.61 0.0017868 2.307159  
## 2 1 82 0.0383449 0.338111 0.433005 0.506993 5.59 0.0017666 2.330254  
## 3 1 83 0.0303048 0.330449 0.525703 0.479705 5.80 0.0018358 2.341801  
## 4 1 84 0.0347259 0.362525 0.604706 0.520104 6.89 0.0018859 2.346420  
## 5 1 85 0.0365730 0.325395 0.578723 0.497059 6.55 0.0019244 2.364896  
## 6 1 86 0.0347524 0.326062 0.512324 0.439863 6.90 0.0018952 2.385681  
## 7 1 87 0.0356036 0.298270 0.527596 0.436170 6.71 0.0018279 2.422633  
## 8 3 81 0.0163921 0.202899 0.869048 0.465753 8.45 0.0005939 0.976834  
## taxpc region smsa pctmin wcon wtuc wtrd wfir wser  
## 1 25.69763 central no 20.21870 206.4803 333.6209 182.3330 272.4492 215.7335  
## 2 24.87425 central no 20.21870 212.7542 369.2964 189.5414 300.8788 231.5767  
## 3 26.45144 central no 20.21870 219.7802 1394.8030 196.6395 309.9696 240.1568  
## 4 26.84235 central no 20.21870 223.4238 398.8604 200.5629 350.0863 252.4477  
## 5 28.14034 central no 20.21870 243.7562 358.7830 206.8827 383.0707 261.0861  
## 6 29.74098 central no 20.21870 257.9139 369.5465 218.5165 409.8842 269.6129  
## 7 30.99368 central no 20.21870 281.4259 408.7245 221.2701 453.1722 274.1775  
## 8 14.56088 central no 7.91632 188.7683 292.6422 151.4234 202.4292 191.3742  
## wmfg wfed wsta wloc mix pctymle lcrmrte lprbarr  
## 1 229.12 409.37 236.24 231.47 0.0999179 0.0876968 -3.221757 -1.238923  
## 2 240.33 419.70 253.88 236.79 0.1030491 0.0863767 -3.261134 -1.084381  
## 3 269.70 438.85 250.36 248.58 0.0806787 0.0850909 -3.496449 -1.107303  
## 4 281.74 459.17 261.93 264.38 0.0785035 0.0838333 -3.360270 -1.014662  
## 5 298.88 490.43 281.44 288.58 0.0932486 0.0823065 -3.308445 -1.122715  
## 6 322.65 478.67 286.91 306.70 0.0973228 0.0800806 -3.359507 -1.120668  
## 7 334.54 477.58 292.09 311.91 0.0801688 0.0778710 -3.335309 -1.209756  
## 8 210.75 381.72 247.38 213.17 0.0561224 0.0870046 -4.110956 -1.595047  
## lprbconv lprbpris lavgsen lpolpc ldensity lwcon lwtuc  
## 1 -0.9111490 -0.7503061 1.724551 -6.327340 0.8360171 5.330205 5.810005  
## 2 -0.8370060 -0.6792581 1.720979 -6.338704 0.8459773 5.360137 5.911600  
## 3 -0.6430188 -0.7345839 1.757858 -6.300291 0.8509204 5.392628 7.240509  
## 4 -0.5030129 -0.6537265 1.930071 -6.273361 0.8528909 5.409070 5.988612  
## 5 -0.5469313 -0.6990466 1.879465 -6.253162 0.8607340 5.496169 5.882718  
## 6 -0.6687981 -0.8212920 1.931521 -6.268420 0.8694848 5.552626 5.912277  
## 7 -0.6394244 -0.8297232 1.903599 -6.304609 0.8848549 5.639869 6.013041  
## 8 -0.1403569 -0.7640998 2.134166 -7.428766 -0.0234386 5.240520 5.678950  
## lwtrd lwfir lwser lwmfg lwfed lwsta lwloc lpctymle  
## 1 5.205835 5.607452 5.374044 5.434246 6.014619 5.464848 5.444450 -2.433870  
## 2 5.244607 5.706707 5.444911 5.482013 6.039540 5.536862 5.467174 -2.449038  
## 3 5.281372 5.736475 5.481292 5.597310 6.084157 5.522900 5.515765 -2.464036  
## 4 5.301128 5.858180 5.531204 5.640985 6.129421 5.568077 5.577387 -2.478925  
## 5 5.332152 5.948220 5.564850 5.700042 6.195282 5.639919 5.664972 -2.497306  
## 6 5.386862 6.015875 5.596987 5.776568 6.171011 5.659169 5.725870 -2.524721  
## 7 5.399384 6.116272 5.613776 5.812757 6.168732 5.677062 5.742715 -2.552702  
## 8 5.020080 5.310390 5.254230 5.350673 5.944687 5.510926 5.362090 -2.441794  
## lpctmin ltaxpc lmix  
## 1 3.006608 3.246399 -2.303407  
## 2 3.006608 3.213833 -2.272549  
## 3 3.006608 3.275311 -2.517281  
## 4 3.006608 3.289981 -2.544612  
## 5 3.006608 3.337204 -2.372487  
## 6 3.006608 3.392526 -2.329722  
## 7 3.006608 3.433783 -2.523621  
## 8 2.068926 2.678338 -2.880219

#Display all variable names  
print(ls(Crime))

## [1] "avgsen" "county" "crmrte" "density" "lavgsen" "lcrmrte"   
## [7] "ldensity" "lmix" "lpctmin" "lpctymle" "lpolpc" "lprbarr"   
## [13] "lprbconv" "lprbpris" "ltaxpc" "lwcon" "lwfed" "lwfir"   
## [19] "lwloc" "lwmfg" "lwser" "lwsta" "lwtrd" "lwtuc"   
## [25] "mix" "pctmin" "pctymle" "polpc" "prbarr" "prbconv"   
## [31] "prbpris" "region" "smsa" "taxpc" "wcon" "wfed"   
## [37] "wfir" "wloc" "wmfg" "wser" "wsta" "wtrd"   
## [43] "wtuc" "year"

#Display number of variables  
print(length(ls(Crime)))

## [1] 44

#Display each variable's descriptive summary   
print(summary(Crime))

## county year crmrte prbarr   
## Min. : 1.0 Min. :81 Min. :0.001812 Min. :0.05882   
## 1st Qu.: 51.0 1st Qu.:82 1st Qu.:0.018352 1st Qu.:0.21790   
## Median :103.0 Median :84 Median :0.028441 Median :0.27824   
## Mean :100.6 Mean :84 Mean :0.031588 Mean :0.30737   
## 3rd Qu.:151.0 3rd Qu.:86 3rd Qu.:0.038406 3rd Qu.:0.35252   
## Max. :197.0 Max. :87 Max. :0.163835 Max. :2.75000   
## prbconv prbpris avgsen polpc   
## Min. : 0.06838 Min. :0.1489 Min. : 4.220 Min. :0.0004585   
## 1st Qu.: 0.34769 1st Qu.:0.3744 1st Qu.: 7.160 1st Qu.:0.0011913   
## Median : 0.47437 Median :0.4286 Median : 8.495 Median :0.0014506   
## Mean : 0.68862 Mean :0.4255 Mean : 8.955 Mean :0.0019168   
## 3rd Qu.: 0.63560 3rd Qu.:0.4832 3rd Qu.:10.197 3rd Qu.:0.0018033   
## Max. :37.00000 Max. :0.6786 Max. :25.830 Max. :0.0355781   
## density taxpc region smsa pctmin   
## Min. :0.1977 Min. : 14.30 other :245 no :574 Min. : 1.284   
## 1st Qu.:0.5329 1st Qu.: 23.43 west :147 yes: 56 1st Qu.:10.005   
## Median :0.9526 Median : 27.79 central:238 Median :24.852   
## Mean :1.3861 Mean : 30.24 Mean :25.713   
## 3rd Qu.:1.5078 3rd Qu.: 33.27 3rd Qu.:38.223   
## Max. :8.8277 Max. :119.76 Max. :64.348   
## wcon wtuc wtrd wfir   
## Min. : 65.62 Min. : 28.86 Min. : 16.87 Min. : 3.516   
## 1st Qu.: 201.66 1st Qu.: 317.60 1st Qu.: 168.05 1st Qu.:235.705   
## Median : 236.46 Median : 358.20 Median : 185.48 Median :264.423   
## Mean : 245.67 Mean : 406.10 Mean : 192.82 Mean :272.059   
## 3rd Qu.: 269.69 3rd Qu.: 411.02 3rd Qu.: 204.82 3rd Qu.:302.440   
## Max. :2324.60 Max. :3041.96 Max. :2242.75 Max. :509.466   
## wser wmfg wfed wsta   
## Min. : 1.844 Min. :101.8 Min. :255.4 Min. :173.0   
## 1st Qu.: 191.319 1st Qu.:234.0 1st Qu.:361.5 1st Qu.:258.2   
## Median : 216.475 Median :271.6 Median :404.0 Median :289.4   
## Mean : 224.671 Mean :285.2 Mean :403.9 Mean :296.9   
## 3rd Qu.: 247.155 3rd Qu.:320.0 3rd Qu.:444.6 3rd Qu.:331.5   
## Max. :2177.068 Max. :646.9 Max. :598.0 Max. :548.0   
## wloc mix pctymle lcrmrte   
## Min. :163.6 Min. :0.002457 Min. :0.06216 Min. :-6.314   
## 1st Qu.:226.8 1st Qu.:0.075324 1st Qu.:0.07859 1st Qu.:-3.998   
## Median :253.1 Median :0.102089 Median :0.08316 Median :-3.560   
## Mean :258.0 Mean :0.139396 Mean :0.08897 Mean :-3.609   
## 3rd Qu.:289.3 3rd Qu.:0.149009 3rd Qu.:0.08919 3rd Qu.:-3.260   
## Max. :388.1 Max. :4.000000 Max. :0.27436 Max. :-1.809   
## lprbarr lprbconv lprbpris lavgsen   
## Min. :-2.833 Min. :-2.6827 Min. :-1.9042 Min. :1.440   
## 1st Qu.:-1.524 1st Qu.:-1.0564 1st Qu.:-0.9824 1st Qu.:1.969   
## Median :-1.279 Median :-0.7458 Median :-0.8473 Median :2.139   
## Mean :-1.274 Mean :-0.6929 Mean :-0.8786 Mean :2.153   
## 3rd Qu.:-1.043 3rd Qu.:-0.4532 3rd Qu.:-0.7273 3rd Qu.:2.322   
## Max. : 1.012 Max. : 3.6109 Max. :-0.3878 Max. :3.252   
## lpolpc ldensity lwcon lwtuc   
## Min. :-7.688 Min. :-1.62091 Min. :4.184 Min. :3.362   
## 1st Qu.:-6.733 1st Qu.:-0.62934 1st Qu.:5.307 1st Qu.:5.761   
## Median :-6.536 Median :-0.04857 Median :5.466 Median :5.881   
## Mean :-6.491 Mean :-0.01593 Mean :5.463 Mean :5.916   
## 3rd Qu.:-6.318 3rd Qu.: 0.41066 3rd Qu.:5.597 3rd Qu.:6.019   
## Max. :-3.336 Max. : 2.17789 Max. :7.751 Max. :8.020   
## lwtrd lwfir lwser lwmfg   
## Min. :2.826 Min. :1.257 Min. :0.6118 Min. :4.623   
## 1st Qu.:5.124 1st Qu.:5.463 1st Qu.:5.2539 1st Qu.:5.455   
## Median :5.223 Median :5.578 Median :5.3775 Median :5.604   
## Mean :5.232 Mean :5.579 Mean :5.3646 Mean :5.615   
## 3rd Qu.:5.322 3rd Qu.:5.712 3rd Qu.:5.5100 3rd Qu.:5.768   
## Max. :7.715 Max. :6.233 Max. :7.6857 Max. :6.472   
## lwfed lwsta lwloc lpctymle   
## Min. :5.543 Min. :5.153 Min. :5.097 Min. :-2.778   
## 1st Qu.:5.890 1st Qu.:5.554 1st Qu.:5.424 1st Qu.:-2.543   
## Median :6.001 Median :5.668 Median :5.534 Median :-2.487   
## Mean :5.989 Mean :5.678 Mean :5.540 Mean :-2.443   
## 3rd Qu.:6.097 3rd Qu.:5.804 3rd Qu.:5.667 3rd Qu.:-2.417   
## Max. :6.394 Max. :6.306 Max. :5.961 Max. :-1.293   
## lpctmin ltaxpc lmix   
## Min. :0.2497 Min. :2.660 Min. :-6.009   
## 1st Qu.:2.3030 1st Qu.:3.154 1st Qu.:-2.586   
## Median :3.2127 Median :3.325 Median :-2.282   
## Mean :2.9134 Mean :3.356 Mean :-2.234   
## 3rd Qu.:3.6434 3rd Qu.:3.505 3rd Qu.:-1.904   
## Max. :4.1643 Max. :4.786 Max. : 1.386

1. [5 Pts] Calculate the mean,variance and standard deviation of probability of arrest (prbarr) by omitting the missing values, if any.

#mean  
prbarr\_avg <- mean(Crime$prbarr, na.rm = TRUE)  
prbarr\_avg

## [1] 0.3073682

#variance  
prbarr\_var <- var(Crime$prbarr, na.rm = TRUE)  
prbarr\_var

## [1] 0.02931104

#standard deviation  
prbarr\_sd <- sqrt(prbarr\_var)  
prbarr\_sd

## [1] 0.1712047

1. [5 Pts] Use lpolpc (log-police per capita) and smsa variables to build a linear regression model to predict probability of arrest (prbarr). And, compare with another linear regression model that uses polpc (police per capita) and smsa.

* [5 Pts] How can you draw a conclusion from the results? (Note: Full marks requires comment on the predictors)

#Multiple linear regression  
model1 <- lm(prbarr ~ lpolpc + smsa, data = Crime)  
summary(model1)

##   
## Call:  
## lm(formula = prbarr ~ lpolpc + smsa, data = Crime)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.46050 -0.07973 -0.01784 0.05390 2.24094   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.88964 0.08152 10.913 < 2e-16 \*\*\*  
## lpolpc 0.08784 0.01246 7.048 4.80e-12 \*\*\*  
## smsayes -0.13638 0.02305 -5.918 5.38e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1623 on 627 degrees of freedom  
## Multiple R-squared: 0.104, Adjusted R-squared: 0.1012   
## F-statistic: 36.4 on 2 and 627 DF, p-value: 1.109e-15

model2 <- lm(prbarr ~ polpc + smsa, data = Crime)  
summary(model2)

##   
## Call:  
## lm(formula = prbarr ~ polpc + smsa, data = Crime)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.72651 -0.07840 -0.01759 0.04955 2.22692   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.28213 0.00807 34.958 < 2e-16 \*\*\*  
## polpc 18.34603 2.34684 7.817 2.29e-14 \*\*\*  
## smsayes -0.11163 0.02254 -4.953 9.40e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.161 on 627 degrees of freedom  
## Multiple R-squared: 0.1189, Adjusted R-squared: 0.1161   
## F-statistic: 42.31 on 2 and 627 DF, p-value: < 2.2e-16

#The p value of the F statistic in model1 and model2 are both highly significant. Model1 residuals have a smaller range than that of Model2 (-0.4 to 2.2 compared to -0.7 to 2.2) . The R square of both models is simular (0.1) indicating that about 10% of the variation in the output of prbarr are explained by the input variables.   
  
#Looking at the Coefficients from Model1, we can conclude that there is a highly significant (Pr(>|t|\*\*\*) positive relationship between lpolpc and prbarr and negative relationship between smsayes and prbarr. For an increase of one unit in log of police per capita (lpolpc), there is an estimated increase of 0.08 units of probability of arrest (prbarr). For each increase of one unit of individuals responding yes to residing in a standard metropolitan statistical area (smsayes), there is an estimated decrease of probability of arrest (prbarr) of 0.14.   
  
#Looking at the Coefficients from Model2, we can conclude that there is a highly significant (Pr(>|t|\*\*\*) positive correlation between polpc and prbarr, and negative relationship between smsayes and prbarr. As police per capita (polpc) increases by 1 unit, it is estimated that the probability of arrest (prbarr) increases by 18.35, and as the numper of individuls responding yes to residing in a standard metropolitan statistical area (smsayes) increases by one unit, the probability of arrest (prbarr) decreases by an estimated 0.11

1. [5 Pts] Based on the output of your model, write the equations using the intercept and factors of smsa when polpc is set to 0.0015. and compare the result with predict() function.  
   Hint: Explore predict() function

#Model 2 equation when smsa is no  
y.smsano = 0.28213 + 18.34603\*(0.0015)  
y.smsano

## [1] 0.309649

#Model2 equation when smsa is yes  
y.smsayes = 0.28213 + 18.34603\*(0.0015) - 0.11163  
y.smsayes

## [1] 0.198019

#Temporary dataframe to set polpc to 0.0015  
polpc.smsa.df <- data.frame(polpc = c(0.0015), smsa = c('no','yes'))  
predict(model2, polpc.smsa.df)

## 1 2   
## 0.3096441 0.1980168

#The results obtained from using the predict function are exactly the same up to 5 decimal points.

1. [5 Pts] Find Pearson correlation between probability of prison sentence prbpris and tax per capita taxpc; and also Pearson correlation between probability of conviction prbconv and probability of arrest prbarr.

* [5 Pts] What conclusions can you draw? Write your reasons as comments.

cor.test(Crime$prbpris, Crime$taxpc)

##   
## Pearson's product-moment correlation  
##   
## data: Crime$prbpris and Crime$taxpc  
## t = -2.8261, df = 628, p-value = 0.004862  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.18852675 -0.03424894  
## sample estimates:  
## cor   
## -0.1120631

#We can conclude that there is a very low negative linear correlation between prison sentence (prbpris) and tax per capita (taxpc) because the pearson correlation coefficient is -0.11. The p value for this result is also small (p-value = 0.0048) is < p-value 0.05 indicating that this result is statistically significant.

cor.test(Crime$prbconv, Crime$prbarr)

##   
## Pearson's product-moment correlation  
##   
## data: Crime$prbconv and Crime$prbarr  
## t = 0.89192, df = 628, p-value = 0.3728  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.04266359 0.11336788  
## sample estimates:  
## cor   
## 0.0355689

#We can conclude that there is a very low positive correlation linear relationship between the probability of conviction (prbconv) and probability of arrest (prbarr) because the pearson correlation coefficient is 0.3 (this is almost zero which is equivalent to no relationship). The p-value is = 0.3, which is higher than p-value of 0.05 indicating that this result is not statistically significant, therefore we cannot conclude that there is a correlation.

1. [5 Pts] Display the correlation matrix of the variables: prbconv, prbpris, avgsen, polpc.

* [5 Pts] Write what conclusion you can draw, as comments.

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

## Warning: package 'corrplot' was built under R version 4.3.2

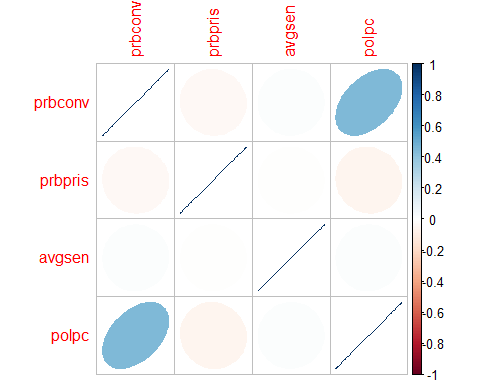
## corrplot 0.92 loaded

table\_cor<- cor(Crime[,5:8])  
table\_cor

## prbconv prbpris avgsen polpc  
## prbconv 1.00000000 -0.037340175 0.015304708 0.44963500  
## prbpris -0.03734017 1.000000000 -0.004299394 -0.05745238  
## avgsen 0.01530471 -0.004299394 1.000000000 0.01712970  
## polpc 0.44963500 -0.057452385 0.017129699 1.00000000

#We can conclude that there is a moderate positive correlation coefficient between prbconv and polpc (0.45).   
  
#We can also conclude that there is very little negative linear relationship between prbconv and prbpris (-0.037), prbpris and avgsen (-0.004), and polpc and prbpris(-0.057) because these values are between -0.1 and 0 (where zero represent no linear relationship).   
  
#We can also conclude that there is a very low positive correlation between prbconv and avgsen (0.015) and avgsen and polpc (0.017) because their pearson correlation coefficients lie between 0.1 and 0.

#Further visualizing pearson correlations  
corrplot(table\_cor, method = "ellipse")



## Question 3 [15 Pts]

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

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

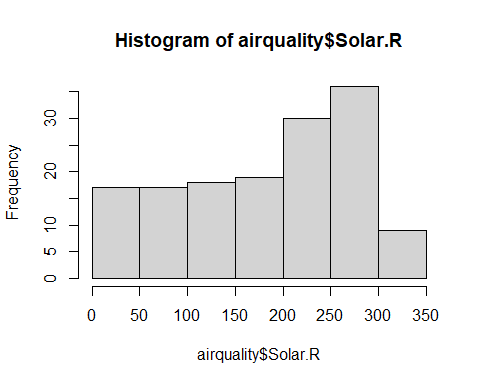
## Warning: package 'ISwR' was built under R version 4.3.2

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. [5 Pts] Plot a histogram to assess the normality of the Solar.R variable, then explain why it does not appear normally distributed.

print(hist(airquality$Solar.R))



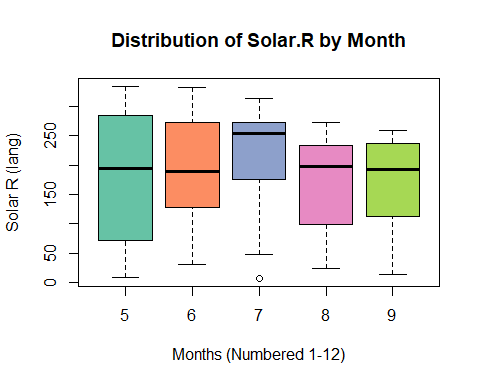
## $breaks  
## [1] 0 50 100 150 200 250 300 350  
##   
## $counts  
## [1] 17 17 18 19 30 36 9  
##   
## $density  
## [1] 0.002328767 0.002328767 0.002465753 0.002602740 0.004109589 0.004931507  
## [7] 0.001232877  
##   
## $mids  
## [1] 25 75 125 175 225 275 325  
##   
## $xname  
## [1] "airquality$Solar.R"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

#This variable does not appear normally distributed because the histogram is not a bell shaped curve (the mean, median, and mode do not appear to be centered with a symmetrical distribution on either side), rather, the data appears to be left-skewed.

1. [5 Pts] Create a boxplot that shows the distribution of Solar.R in each month. Use different colors for each month.

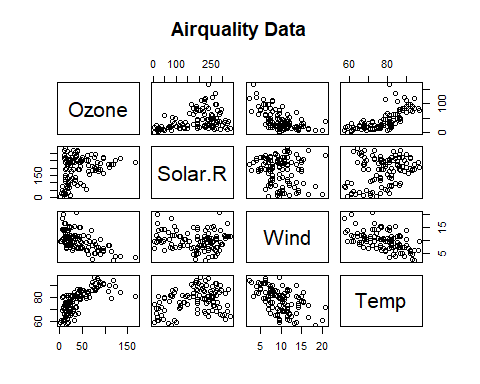
#Colour vector preparation  
library (RColorBrewer)  
  
boxplot(Solar.R ~ Month, data = airquality, main = "Distribution of Solar.R by Month", xlab = "Months (Numbered 1-12)", ylab = "Solar R (lang)", col= brewer.pal(12, name = "Set2"))

## Warning in brewer.pal(12, name = "Set2"): n too large, allowed maximum for palette Set2 is 8  
## Returning the palette you asked for with that many colors



1. [5 Pts] Create a matrix of scatterplots of all the numeric variables in the airquality dataset (i.e. Ozone, Solar.R, Wind and Temp.) (Hint: investigate pairs() function)

#Matrix only comparing: Ozone, Solar.R, Wind and Temp  
print(pairs(~ Ozone + Solar.R + Wind + Temp, data = airquality, main = "Airquality Data", na.action = na.omit))



## NULL

## Question 4 [25 Pts]

Many times in data analysis, we need a method that relies on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems. In fact, this is a mathematical technique, which is used to estimate the possible outcomes of an uncertain event and is called the *Monte Carlo Method*.

Consider that We roll a die 10 times and we want to know the probability of getting more than 3 times of even numbers. This is a problem for the Binomial distribution, but suppose we don’t know anything about Binomial distribution. We can easily solve this problem with a Monte Carlo Simulation.

1. [5 Pts] The Monte Carlo Method uses random numbers to simulate some process. Here the process is rolling a die 10 times. Assume the die is fair. What is the probability of success or getting an even number in rolling the die once?

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

## Warning: package 'gtools' was built under R version 4.3.2

one.dice <- c(1, 2, 3, 4, 5, 6)  
perm <- permutations(length(one.dice), 1, one.dice,  
repeats.allowed =TRUE)  
perm

## [,1]  
## [1,] 1  
## [2,] 2  
## [3,] 3  
## [4,] 4  
## [5,] 5  
## [6,] 6

#Probability of success of rolling an even number when rolling the die once is 3/6 or 0.5

1. [10 Pts] Define a function named one.trial, that simulates a single round of rolling a die 10 times and returns true if the number of even numbers is > 3.

#INSERT YOUR ANSWER HERE.  
one.trial <- function(){  
   
 die <- c()  
 num\_even <- 0  
   
 #Roll the die and store rolls in a vector called "die"  
 for (roll in 1:10){  
 die <- append(die, sample(1:6, size = 1, replace = TRUE))  
 }  
 #print(die)  
   
 #Count number of even numbers in the vector "die"   
 for (i in die) {  
 if (i == 2) {  
 num\_even <- num\_even + 1   
 } else if (i == 4){  
 num\_even <- num\_even + 1   
 } else if (i == 6){  
 num\_even <- num\_even + 1  
 } else {  
 num\_even <- num\_even + 0  
 }  
 }  
 #print(num\_even)  
   
 #Determine whether count of even numbers is over 3  
 if (num\_even > 3) {  
 return(TRUE)  
 } else{  
 return(FALSE)  
 }  
   
 }

one.trial()

## [1] TRUE

1. [5 pts] Repeat the function one.trial for N = 100,000 times and sum up the outcomes and store the result in a variable named desired.output. Compute the probability of getting more than 3 times of even numbers by using relative frequency.

#Returns the number one.trial() that is equal to TRUE (>3 even numbers)  
set.seed(10)  
desired.output <- sum(replicate(n = 100000, expr = one.trial()))  
desired.output

## [1] 82924

my.probability <- desired.output/100000  
my.probability

## [1] 0.82924

1. [5 pts] Use the Binomial formula you learned before to calculate such probability and Compare it with the probability value obtained in part (c).

set.seed(10)  
pbinom(q = 3, size = 10, prob = 1/2)

## [1] 0.171875

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 at least, and store the results in a vector called Prob then take the mean of Prob vector to be more accurate.

\*\* End of Assignment \*\*