CIND 123: Data Analytics Basic Methods: Assignment-3

Assignment 3 (10%)

Total 100 Marks

[Qian (Jessie) Ma]

[CIND 123 Section D40 & student number: 501274167]

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 (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 $(2, 4, 6, \ldots, 20)$.

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

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

Question 1 [15 Pts]

a. [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.

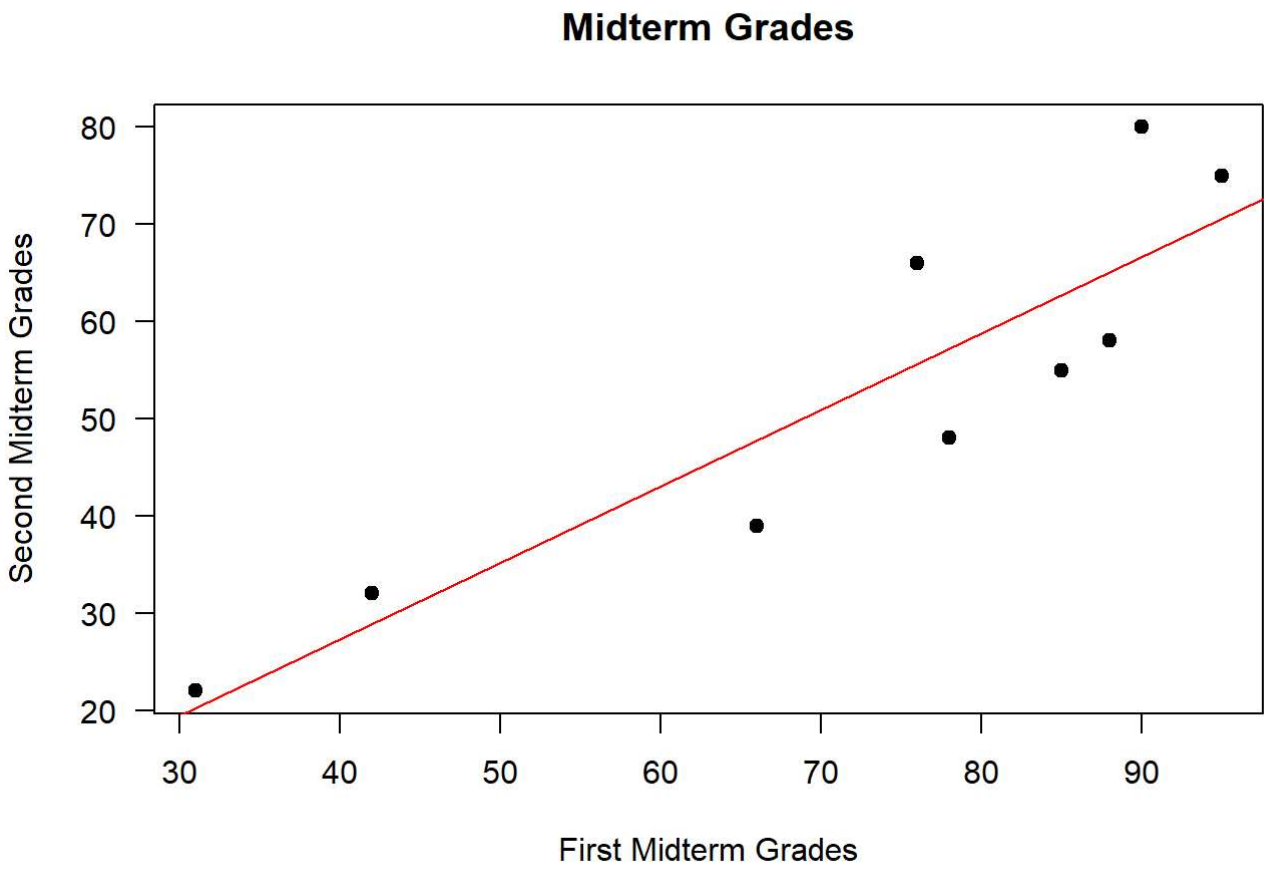
```
#INSERT YOUR ANSWER HERE.
#First set variables for vectors
first<-c(85,76,78,88,90,95,42,31,66)
second<-c(55,66,48,58,80,75,32,22,39)
linear_model<-lm(second~first)
summary(linear_model)
```

```
##
## 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
```

```
#Based on the output of the summary function, the least-squares line relating the second midterm to the first midterm is second = 0.7870 * first -4.1516. Since the p-value of 0.00102 = 0.102% is less than 5%, it is significant, therefore the assumption of a linear relationship appears to be reasonable in this case. Also, the Multiple R-squared value is 80.58% which is high meaning the linear regression model is fitting the data decently well.
```

b. [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.

```
#INSERT YOUR ANSWER HERE.
plot(first,second,main="Midterm Grades",xlab="First Midterm Grades",ylab="Second Midterm Grades",pch=19,las=1)
abline(linear_model,col="red")
```



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

```
#INSERT YOUR ANSWER HERE.
df1=data.frame(first=c(81,23))
prediction=predict(linear_model,df1)
prediction
```

```
##        1        2
## 59.59881 13.95039
```

```
#When the first midterm grade is 81, the second midterm grade is 59.59881; when the first midter
m grade is 23, the second midterm grade is 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)
```

a. [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.

```
#INSERT YOUR ANSWER HERE.
#Display first 8 rows
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 variable names
colnames(Crime)
```

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

```
#Display number of variables
ncol(Crime)
```

```
## [1] 44
```

```
#Display descriptive summaries of variables
desc_summ<-sapply(Crime, summary)
desc_summ
```

```
## $county
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    51.0   103.0   100.6   151.0   197.0
##
## $year
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      81      82      84      84      86      87
##
## $crmrte
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001812 0.018352 0.028441 0.031588 0.038406 0.163835
##
## $prbarr
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.05882 0.21790 0.27824 0.30737 0.35252 2.75000
##
## $prbconv
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.06838  0.34769  0.47437  0.68862  0.63560 37.00000
##
## $prbpris
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1489  0.3744  0.4286  0.4255  0.4832  0.6786
##
## $avgsen
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.220   7.160   8.495   8.955  10.197  25.830
##
## $polpc
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0004585 0.0011913 0.0014506 0.0019168 0.0018033 0.0355781
##
## $density
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1977  0.5329  0.9526  1.3861  1.5078  8.8277
##
## $taxpc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.30   23.43   27.79   30.24   33.27  119.76
##
## $region
##   other    west central
##     245     147     238
##
## $smsa
##  no yes
## 574  56
##
## $pctmin
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.284  10.005  24.852  25.713  38.223  64.348
##
```

```
## $wcon
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   65.62  201.66  236.46  245.67  269.69 2324.60
##
## $wtuc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   28.86  317.60  358.20  406.10  411.02 3041.96
##
## $wtrd
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.87  168.05  185.48  192.82  204.82 2242.75
##
## $wfir
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.516 235.705 264.423 272.059 302.440 509.466
##
## $wser
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    1.844  191.319  216.475  224.671  247.155 2177.068
##
## $wmfg
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   101.8   234.0   271.6   285.2   320.0   646.9
##
## $wfed
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   255.4   361.5   404.0   403.9   444.6   598.0
##
## $wsta
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   173.0   258.2   289.4   296.9   331.5   548.0
##
## $wloc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   163.6   226.8   253.1   258.0   289.3   388.1
##
## $mix
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.002457 0.075324 0.102089 0.139396 0.149009 4.000000
##
## $pctymle
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.06216 0.07859 0.08316 0.08897 0.08919 0.27436
##
## $lcrmrte
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -6.314  -3.998  -3.560  -3.609  -3.260  -1.809
##
## $lprbarr
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -2.833  -1.524  -1.279  -1.274  -1.043   1.012
##
```

```
## $lprbconv
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.6827 -1.0564 -0.7458 -0.6929 -0.4532  3.6109
##
## $lprbpris
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.9042 -0.9824 -0.8473 -0.8786 -0.7273 -0.3878
##
## $lavgsen
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.440   1.969   2.139   2.153   2.322   3.252
##
## $lpolpc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -7.688  -6.733  -6.536  -6.491  -6.318  -3.336
##
## $ldensity
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.62091 -0.62934 -0.04857 -0.01593  0.41066  2.17789
##
## $lwcon
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.184   5.307   5.466   5.463   5.597   7.751
##
## $lwtuc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.362   5.761   5.881   5.916   6.019   8.020
##
## $lwtrd
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.826   5.124   5.223   5.232   5.322   7.715
##
## $lwfir
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.257   5.463   5.578   5.579   5.712   6.233
##
## $lwser
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6118  5.2539  5.3775  5.3646  5.5100  7.6857
##
## $lwmfg
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.623   5.455   5.604   5.615   5.768   6.472
##
## $lwfed
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.543   5.890   6.001   5.989   6.097   6.394
##
## $lwsta
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.153   5.554   5.668   5.678   5.804   6.306
##
```

```
## $lwloc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.097   5.424   5.534   5.540   5.667   5.961
##
## $lpctymle
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -2.778  -2.543  -2.487  -2.443  -2.417  -1.293
##
## $lpctmin
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2497  2.3030  3.2127  2.9134  3.6434  4.1643
##
## $ltaxpc
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.660   3.154   3.325   3.356   3.505   4.786
##
## $lmix
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -6.009  -2.586  -2.282  -2.234  -1.904   1.386
```

```
#First 8 rows of the Crime dataset are listed, including the names of all the variables/columns,
the number of variables/columns (44), and the descriptive statistics of each variable/column.
```

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

```
#INSERT YOUR ANSWER HERE.
prbarr1=na.omit(Crime$prbarr)
print(mean(prbarr1))
```

```
## [1] 0.3073682
```

```
print(var(prbarr1))
```

```
## [1] 0.02931104
```

```
print(sd(prbarr1))
```

```
## [1] 0.1712047
```

```
#For the probability of arrest or prbarr: the mean is 0.3073682, the variance is 0.02931104 and
the standard deviation is 0.1712047, omitting missing values.
```

c. [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)

```
#INSERT YOUR ANSWER HERE.
#Build first linear regression model (x variables = lpolpc + smsa, y variable = prbarr)
linear_model1<-lm(prbarr~lpolpc+smsa,data=Crime)
summary(linear_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
```

```
#Build second linear regression model (x variables = polpc + smsa, y variable = prbarr)
linear_model2<-lm(prbarr~polpc+smsa,data=Crime)
summary(linear_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 results of the summaries show that the second model (using polpc and smsa to predict prbar
r) fits the data better than the first model (using lpolpc and smsa to predict prbarr). The p-va
lues of the predictors are statistically significant for both linear regression models. However,
the Multiple R-squared and Adjusted R-squared values for both models are low, indicating the mod
el is not fitting the data well in both cases (only 10.4% of the variation within prbarr is expl
ained by lpolpc and smsa and only 11.89% of the variation within prbarr is explained by polpc an
d smsa.) Residuals in both models also show a relatively larger variance between the actual and
predicted values. It's interesting to see that the p-values are low yet the R-squared values are
also low, indicating although the predictors have an effect on the dependent variable, the model
is not very good at making accurate predictions in general because there is a lot unexplained va
riance. Overall, the R-squared values for the second model are still higher than the first, the
polpc p-value (2.29e-14) is smaller than the lpolpc p-value (4.80e-12), the overall p-value of t
he model is smaller for the second than the first (2.2e-16 vs. 1.109e-15), and residual values a
re also smaller for the second than the first model, indicating the second model performs better
than the first.
```

d. [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

```
#INSERT YOUR ANSWER HERE.
intercept=0.28213
coeff_polpc=18.34603
coeff_smsa=-0.11163
polpc=0.0015
#First equation is for when smsa="no"
prbarr_no=intercept+coeff_polpc*polpc
prbarr_no
```

```
## [1] 0.309649
```

```
#Second equation is for when smsa="yes"
prbarr_yes=intercept+coeff_polpc*polpc+coeff_smsa
prbarr_yes
```

```
## [1] 0.198019
```

```
#First prediction is for when smsa="no"
df2=data.frame(polpc=c(0.0015),smsa=c("no"))
prediction1=predict(linear_model2,df2)
prediction1
```

```
##         1
## 0.3096441
```

```
#Second prediction is for when smsa="yes"
df3=data.frame(polpc=c(0.0015),smsa=c("yes"))
prediction2=predict(linear_model2,df3)
prediction2
```

```
##         1
## 0.1980168
```

```
#After comparing the results of the equations with those of the predict() for when smsa is yes a
nd no and polpc is 0.0015, they are the same.
```

e. [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.

```
#INSERT YOUR ANSWER HERE.
cor(Crime$prbpris,Crime$taxpc)
```

```
## [1] -0.1120631
```

```
cor(Crime$prbconv,Crime$prbarr)
```

```
## [1] 0.0355689
```

```
#From the resulting correlation outputs we can conclude that there is very little correlation be
tween each set of variables as the outputs are near 0. The closer the correlation is to 1 the mo
re highly correlated the variables are. The negative correlation between prbpris and taxpc means
that if one variable increases the other decreases and vice versa. The positive correlation betw
een prbconv and prbarr means that if one variable increases the other also increases and if one
decreases, the other also decreases. Overall, the variables will be affected only slightly due t
o the weak correlation between each set of variables.
```

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

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

```
#INSERT YOUR ANSWER HERE.
cor(Crime[c("prbconv","prbpris","avgsen","polpc")])
```

```
##             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
```

```
#From the correlation matrix we can conclude that polpc and prbconv are the most highly correlat
ed out of all 4 variables. They have a positive correlation which means if one of them increase
s, the other also increases, and if one of them decreases the other also decreases. The variable
s avgsen and prbpris have the weakest correlation with a negative correlation of -0.004299394, i
t is the closest number to 0 out of all the correlations.
```

Question 3 [15 Pts]

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

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

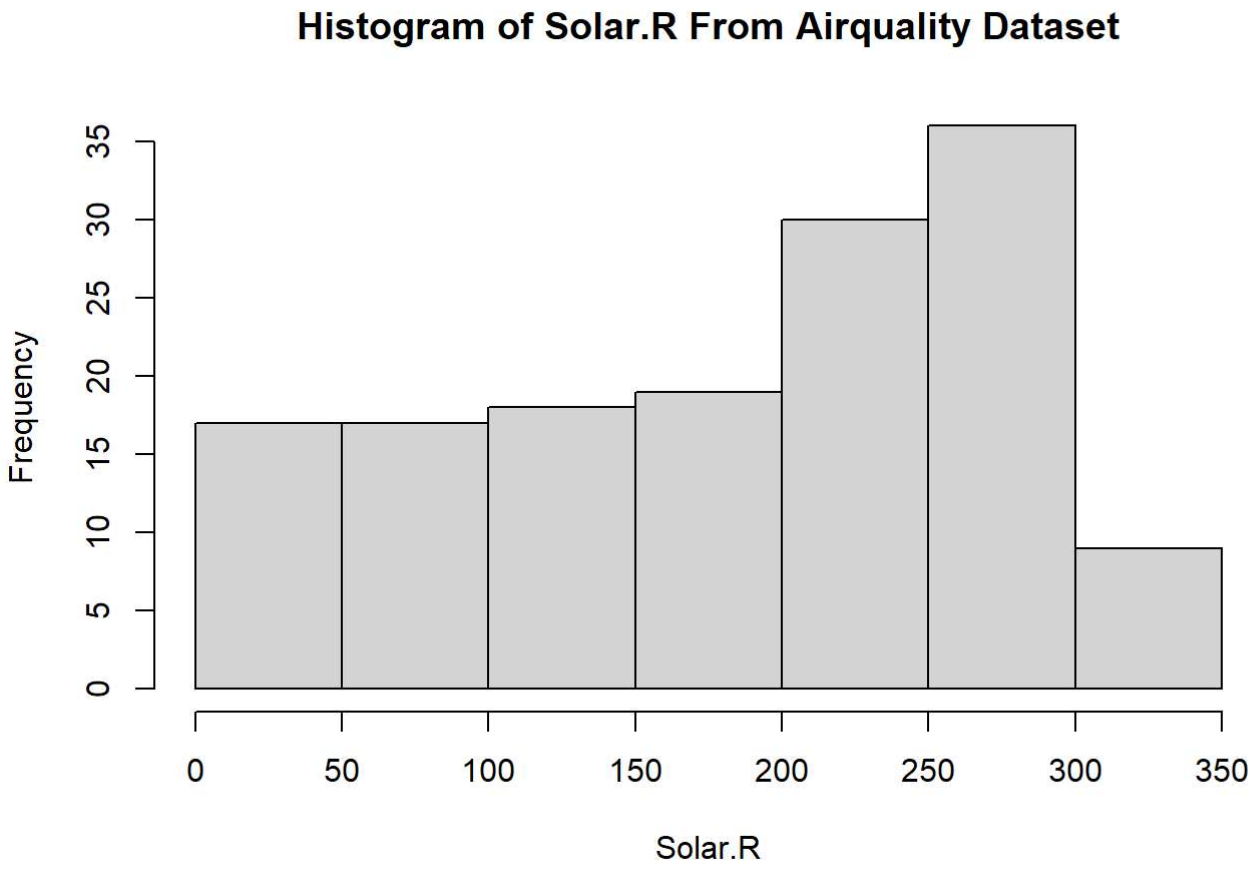
```
## Warning: package 'ISwR' was built under R version 4.3.2
```

```
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 ...
```

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

```
#INSERT YOUR ANSWER HERE.
hist(airquality$Solar.R,main="Histogram of Solar.R From Airquality Dataset",xlab="Solar.R")
```



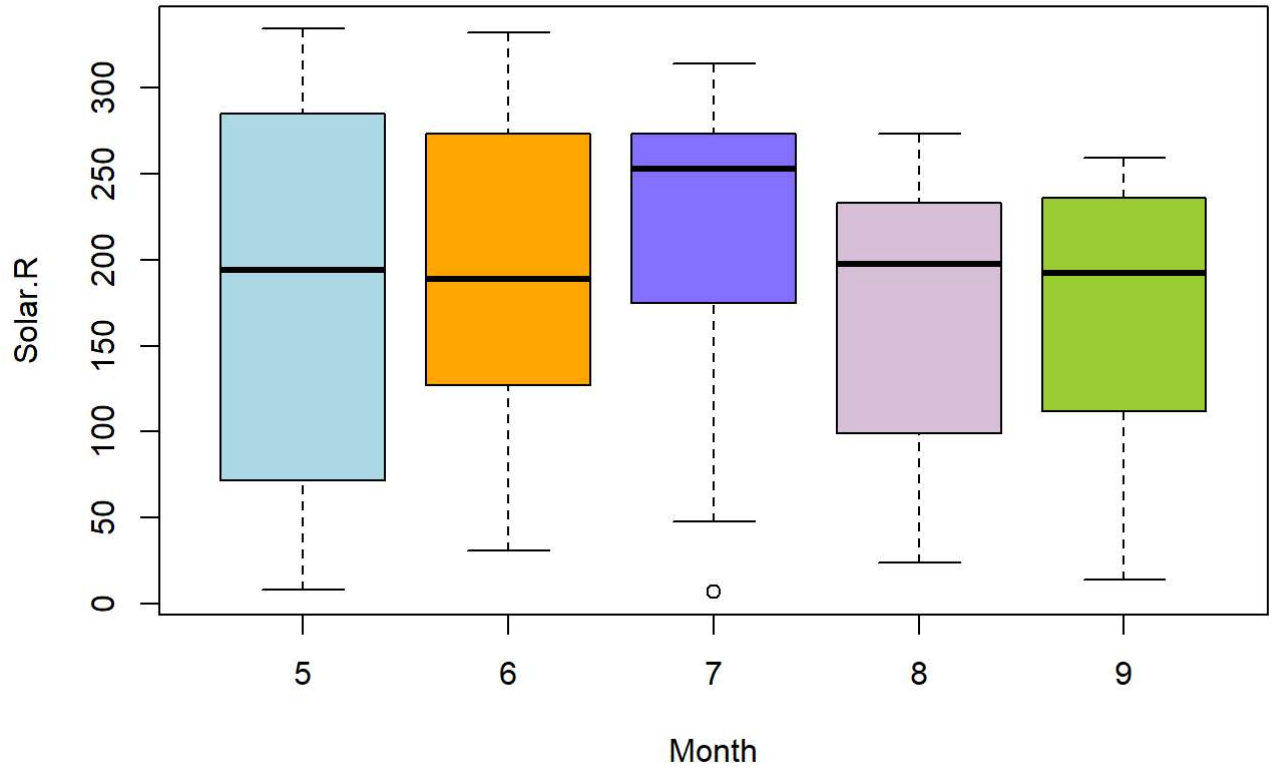

```
#The histogram shown indicates the Solar.R variable does not appear to be normally distributed b
ecause it is skewed to the left. This means the bulk of the observations are larger in value wit
h a few smaller observations. In a normal distribution, the left and right sides of the distribu
tion are symmetrical which is not the case here. The mean, median and mode are not equal in a le
ft skew distribution.
```

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

```
#INSERT YOUR ANSWER HERE.
#First familiarize with the data
head(airquality,3)
```

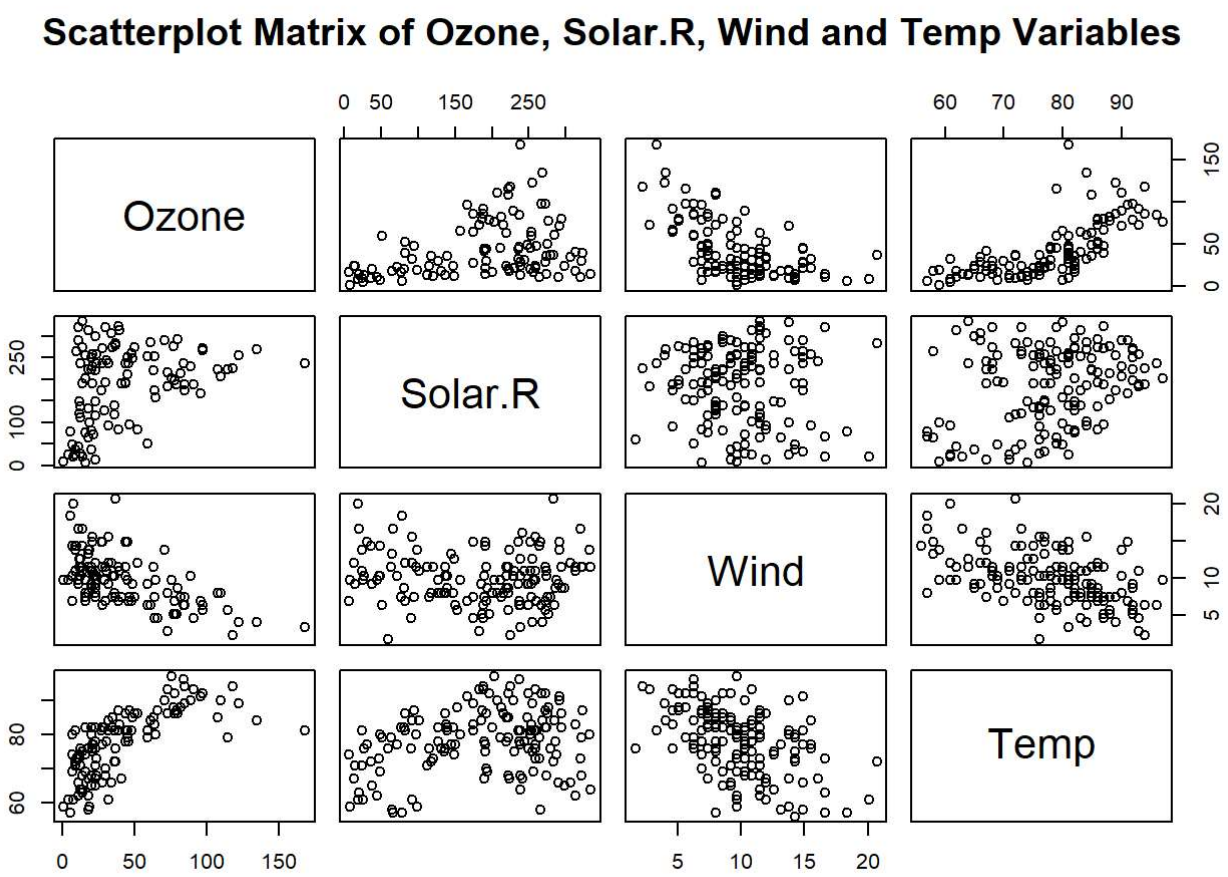
```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
```

```
boxplot_colors<-c("lightblue","orange","lightslateblue","thistle","yellowgreen")
boxplot(Solar.R~Month,data=airquality,col=boxplot_colors)
```



c. [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)

```
#INSERT YOUR ANSWER HERE.
pairs(airquality[c(1:4)],main="Scatterplot Matrix of Ozone, Solar.R, Wind and Temp Variables")
```



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.

a. [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?

```
#INSERT YOUR ANSWER HERE.
#This question can be interpreted in 2 ways, the first being only determining the probability of
getting an even number with one die roll and the second being determining the probability of get
ting an even number in one die roll, replicated for 10 rolls. I will show both.
#Getting an even number with one roll
outcomes<-6
favourable<-3
answer<-favourable/outcomes
answer
```

```
## [1] 0.5
```

```
#Getting an even number with one roll first, then replicated for 10 rolls for an overall probabi
lity as inferred from the first sentence in the question
num_rolls<-10
roll_die<-function() {
  return(sample(1:6,1,replace=TRUE))
}
simulations<-replicate(num_rolls,roll_die())
prob_even<-mean(simulations%%2==0)
prob_even
```

```
## [1] 0.3
```

b. [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() {
  num_rolls1<-10
  rolls<-sample(1:6,num_rolls1,replace=TRUE)
  num_even<-sum(rolls%%2==0)
  return(num_even>3)
}
one.trial()
```

```
## [1] TRUE
```

c. [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.

```
#INSERT YOUR ANSWER HERE.
N<-100000
desired.output<-sum(replicate(N,one.trial()))
#desired.output
resulting_prob<-desired.output/N
resulting_prob
```

```
## [1] 0.82795
```

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

```
#INSERT YOUR ANSWER HERE.
#Method 1: calculate using the binomial function
prob_onetrial<-sum(dbinom(4:10,10,0.5))
desired.output1<-sum(replicate(N,prob_onetrial))
resulting_prob1<-desired.output1/N
resulting_prob1
```

```
## [1] 0.828125
```

```
#Method 2: calculate using the binomial probability distribution formula
probs_vector<-c(choose(10,4)*(0.5^4)*(1-0.5)^6,
                choose(10,5)*(0.5^5)*(1-0.5)^5,
                choose(10,6)*(0.5^6)*(1-0.5)^4,
                choose(10,7)*(0.5^7)*(1-0.5)^3,
                choose(10,8)*(0.5^8)*(1-0.5)^2,
                choose(10,9)*(0.5^9)*(1-0.5^1),
                choose(10,10)*(0.5^10)*(1-0.5)^0)
total_probs<-sum(probs_vector)
desired.output2<-sum(replicate(N,total_probs))
resulting_prob2<-desired.output2/N
resulting_prob2
```

```
## [1] 0.828125
```

```
#The resulting probability from the 2 methods (using the function and formula) are very close to
the probability calculated from part c). The slight variance is due to the function in part c) b
eing called that would generate different outcomes in each trial.
```

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.

```
N.trials<-function() {
  N<-100000
  desired.output<-sum(replicate(N,one.trial()))
  resulting_prob<-desired.output/N
  resulting_prob
}
desired.output3<-sum(replicate(50,N.trials()))
resulting_prob3<-desired.output3/50
resulting_prob3
```

```
## [1] 0.8283424
```

```
#I noticed that the resulting probabilities from part c), d) and in this section are all very cl
ose, indicating the increase in trials from 100,000 to 50*100,000 did not change the probabilit
y.
```

** End of Assignment **