Question 8.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

One of my past business research projects is to study the relationship between the economic environment and our insurance products. Features are listed below:

(1) Consumer Purchase index
(2) Minimum wages
(3) Education level
(4) Types of vacancy
(5) Ages

Question 8.2

Using crime data from http://www.statsci.org/data/general/uscrime.txt (file `uscrime.txt`, description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is `lm` or `glm`) to predict the observed crime rate in a city with the following data:

M = 14.0
So = 0
Ed = 10.0
Po1 = 12.0
Po2 = 15.5
LF = 0.640
M.F = 94.0
Pop = 150
NW = 1.1
U1 = 0.120
U2 = 3.6
Wealth = 3200
Ineq = 20.1
Prob = 0.04
Time = 39.0

Show your model (factors used and their coefficients), the software output, and the quality of fit.

Note that because there are only 47 data points and 15 predictors, you'll probably notice some overfitting. We'll see ways of dealing with this sort of problem later in the course.

In Conclusion, this question applying generalized linear Model. As we can see, we import dataset first, then set random seed to make our code more reproduceable.

Then after we check the summary and head of data, we need to plot histogram to check the data's distribution. Then we calculate the sum of square of total of the dataset. We choose the rules of thumb, which is R square to evaluate the model. Hence, sum of square of total is crucial part.

After this, we set up our glm model. The key takeaway here is we set up family parameter to gaussian, because according to the histogram distribution of dataset, the distribution is roughly a normal distribution. Gaussian parameter can handle the normal distribution problem.

Referring to the results of glm_model, we can see that probabilities of coefficients were calculated. The important part here is to choose those with p value < 0.05. Recalled that null hypothesis, those coefficients with p value < 0.05 is statistically significant. So we keep those variables with p value < 0.05, combined with data listed in the question stem we can set up a new model. Check the glm_model_revised.

The coefficients in glm_model_revised is the variable with p value < 0.05. And the AIC of the revised model is lower than glm_model, which indicates revised model is likely to be a better model. But still we want to diminish the random effect on model. So I applied cross-validation with k=9.

The key takeaway here is understanding glm_cv$delta. The first component here is the raw crossvalidation estimate of prediction error. The second one is adjusted cross-validation estimate of prediction error. Those components are crucial for calculating the r square. R square measured the fitness of the model.

Based on the r square formula, r square = 1 – ssr/sst. Hence, we apply this formula. The r square for glm model with cross-validation is 0.459, for glm_model_revised is 0.654. The adjusted r square for glm model with cross-validation is 0.484, for glm_model_revised is 0.662. We can notice that r square for glm model with 15 factors is 0.459, the reason is we included all the predictors in the model, many of them are statistically insignificant. When we selected independent variables with p value < 0.05, we rebuild the model, the r square for revised model increased, indicates that the fitness of the model increased.

The adjusted r square is larger than ordinary r square. This is rare case. Because the adjusted r square is adjusted according to the independent variables, which the adjusted r square is often smaller. I speculate the reason that adjusted r is bigger in this case is that the models are over fitted here.

The predicted value of two models are listed. The first one is 155 by using all the factors. And the second one is 1304, which is more plausible because the value is closer to the observed data.

- **The detailed codes are listed in following pages.**

week5_hw.R

yangz

2022-09-29

```
# import the data
df <- read.delim("D:/GEORGIA INSTITUTE OF TECHNOLOGY/ISYE_6501/week5/hw5-SP22-1/data 8.2/uscrime.txt")
# set seed
set.seed(9876)
# check the head
head(df)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob    Time Crime
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602 26.2011   791
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599 25.2999  1635
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401 24.3006   578
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801 29.9012  1969
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399 21.2998  1234
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201 20.9995   682
```

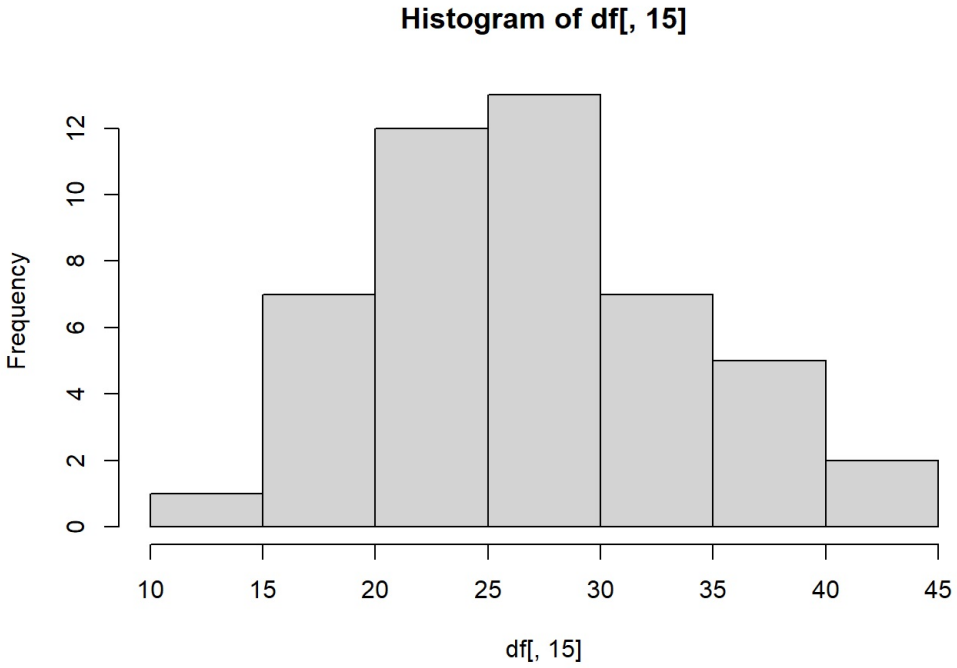
```
# check the summary
summary(df)
```

```
##        M               So               Ed             Po1             Po2               LF              M.F
Pop               NW
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50   Min.   : 4.100   Min.   :0.4800   Min.   : 9
3.40   Min.   :  3.00   Min.   : 0.20
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25   1st Qu.: 5.850   1st Qu.:0.5305   1st Qu.: 9
6.45   1st Qu.: 10.00   1st Qu.: 2.40
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80   Median : 7.300   Median :0.5600   Median : 9
7.70   Median : 25.00   Median : 7.60
##  Mean   :13.86   Mean   :0.3404   Mean   :10.56   Mean   : 8.50   Mean   : 8.023   Mean   :0.5612   Mean   : 9
8.30   Mean   : 36.62   Mean   :10.11
##  3rd Qu.:14.60   3rd Qu.:1.0000   3rd Qu.:11.45   3rd Qu.:10.45   3rd Qu.: 9.700   3rd Qu.:0.5930   3rd Qu.: 9
9.20   3rd Qu.: 41.50   3rd Qu.:13.25
##  Max.   :17.70   Max.   :1.0000   Max.   :12.20   Max.   :16.60   Max.   :15.700   Max.   :0.6410   Max.   :10
7.10   Max.   :168.00   Max.   :42.30
##        U1               U2            Wealth          Ineq            Prob              Time           Crime
##  Min.   :0.07000   Min.   :2.000   Min.   :2880   Min.   :12.60   Min.   :0.00690   Min.   :12.20   Min.   : 3
42.0
##  1st Qu.:0.08050   1st Qu.:2.750   1st Qu.:4595   1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 6
58.5
##  Median :0.09200   Median :3.400   Median :5370   Median :17.60   Median :0.04210   Median :25.80   Median : 8
31.0
##  Mean   :0.09547   Mean   :3.398   Mean   :5254   Mean   :19.40   Mean   :0.04709   Mean   :26.60   Mean   : 9
05.1
##  3rd Qu.:0.10400   3rd Qu.:3.850   3rd Qu.:5915   3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:10
57.5
##  Max.   :0.14200   Max.   :5.800   Max.   :6890   Max.   :27.60   Max.   :0.11980   Max.   :44.00   Max.   :19
93.0
```

```
#
dim(df)
```

```
## [1] 47 16
```

```
#
hist(df[,15])
```




```
# calculate the sum of squares total
SST <- sum((df$Crime - mean(df$Crime))^2)
# build up general linear model
glm_model <- glm(Crime ~ . , data=df, family="gaussian")
# check the factors used and coefficients
summary(glm_model)
```

```
##
## Call:
## glm(formula = Crime ~ ., family = "gaussian", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -395.74   -98.09    -6.69   112.99   512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 43707.93)
##
##     Null deviance: 6880928  on 46  degrees of freedom
## Residual deviance: 1354946  on 31  degrees of freedom
## AIC: 650.03
##
## Number of Fisher Scoring iterations: 2
```

```
# assign values to variable we need to predict
M <- 14.0
So <- 0
Ed <- 10.0
Po1 <- 12.0
Po2 <- 15.5
LF <- 0.640
M.F <- 94.0
Pop <- 150
NW <- 1.1
U1 <- 0.120
U2 <- 3.6
Wealth <- 3200
Ineq <- 20.1
Prob <- 0.04
Time <- 39.0
# fit the variable in glm
glm_model_revised <- glm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob , data=df, family="gaussian")
# check the factors
summary(glm_model_revised)
```

```
##
## Call:
## glm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, family = "gaussian",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -470.68   -78.41   -19.68   133.12   556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 40276.42)
##
##     Null deviance: 6880928  on 46  degrees of freedom
## Residual deviance: 1611057  on 40  degrees of freedom
## AIC: 640.17
##
## Number of Fisher Scoring iterations: 2
```

```
# used cross validation with k=9
glm_cv <- cv.glm(df,glm_model,K=9)
glmr_cv <- cv.glm(df,glm_model_revised,K=9)
# calculate the cross-validated prediction error
glm_cv$delta
```

```
## [1] 79190.22 75568.42
```

```
# calculate R square
1 - glm_cv$delta[1]*nrow(df)/SST
```

```
## [1] 0.4590932
```

```
1 - glmr_cv$delta[1]*nrow(df)/SST
```

```
## [1] 0.6544089
```

```
# calculate adjusted R square
1 - glm_cv$delta[2]*nrow(df)/SST
```

```
## [1] 0.4838319
```

```
1 - glmr_cv$delta[2]*nrow(df)/SST
```

```
## [1] 0.6617965
```

```
# predict using glm
test_data <-data.frame(M = 14.0,So = 0, Ed = 10.0, Po1 = 12.0, Po2 = 15.5,LF = 0.640, M.F = 94.0, Pop = 150, NW =
1.1, U1 = 0.120, U2 = 3.6, Wealth = 3200, Ineq = 20.1, Prob = 0.040,Time = 39.0)
pred_model <- predict(glm_model, test_data)
pred_model
```

```
##        1
## 155.4349
```

```
pred_revised_model <- predict(glm_model_revised, test_data)
pred_revised_model
```

```
##        1
## 1304.245
```