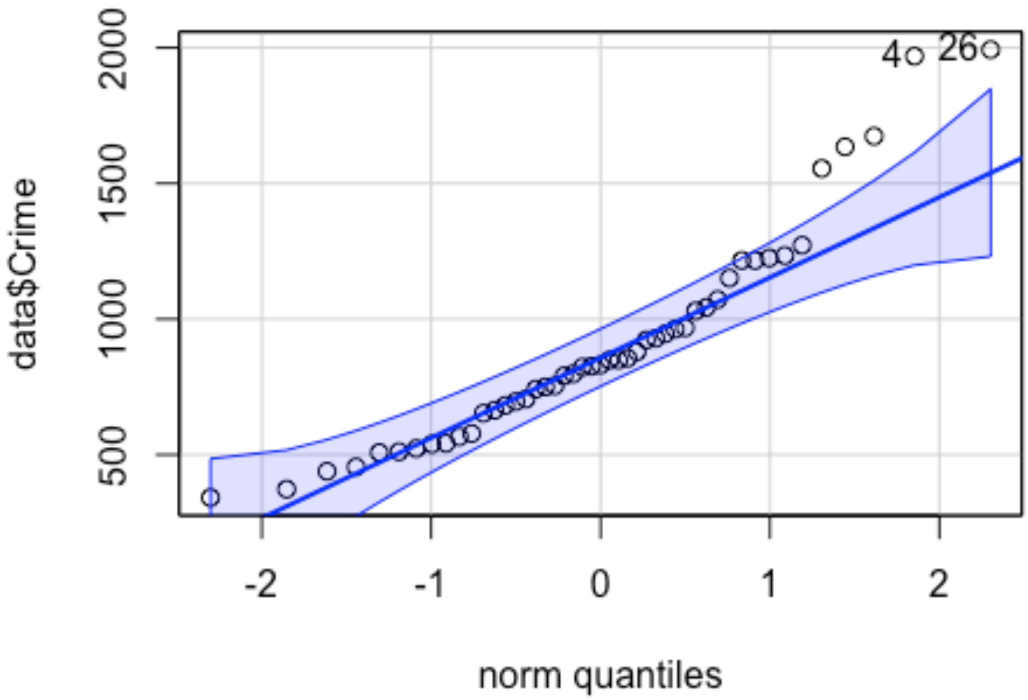
Question 9.1

Using the same crime data set `uscrime.txt` as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function `prcomp` for PCA. (**Note** that to first scale the data, you can include `scale. = TRUE` to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

```
##Import the data
setwd("/Users/[redacted]/GT/Course/ISYE6501/HW6")
data<- read.table("uscrime.txt", header = TRUE)


head(data)

##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Pr
ob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.0846
02
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.0295
99
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.0834
01
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.0158
01
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.0413
99
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.0342
01
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682

library(ggplot2)
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://g
oo.gl/ve3WBa

library(car)

## Loading required package: carData
```

```
##Use qqPlot to determine whether a box-cox transformation is needed
qqPlot(data$Crime)
```

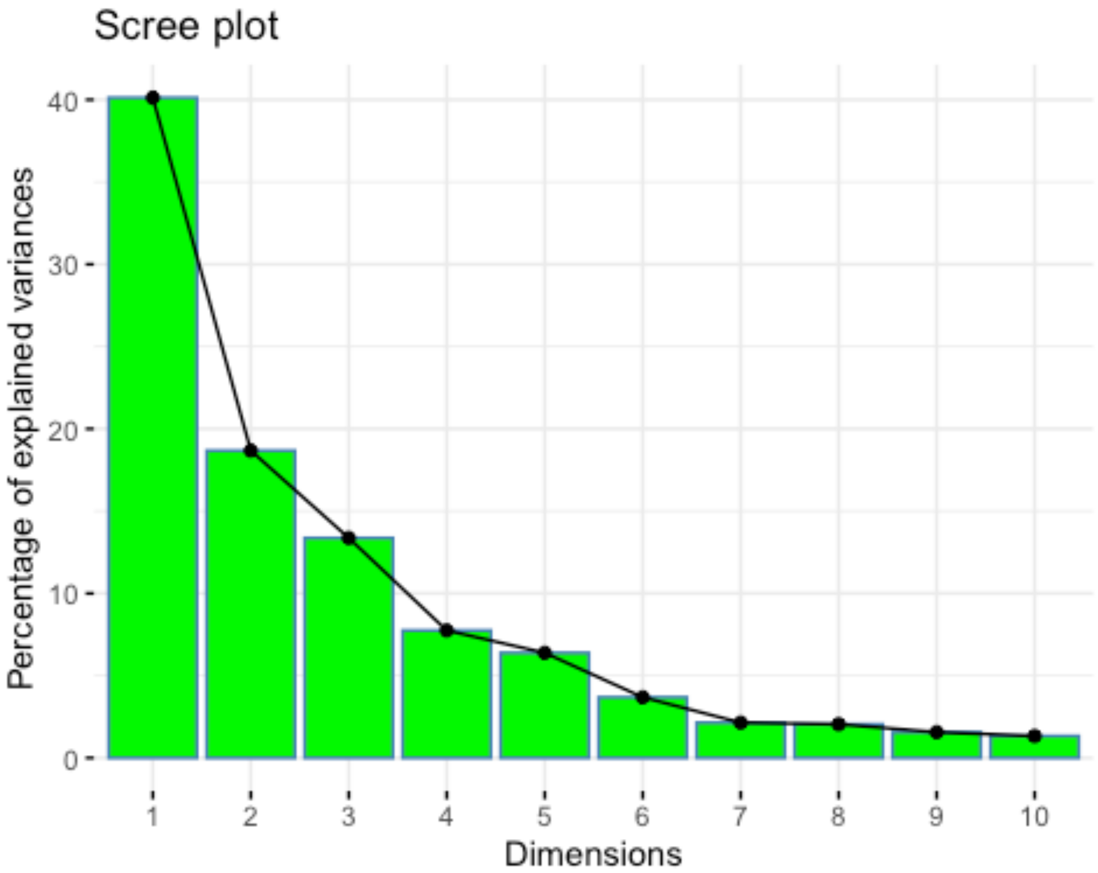



```
## [1] 26  4
```

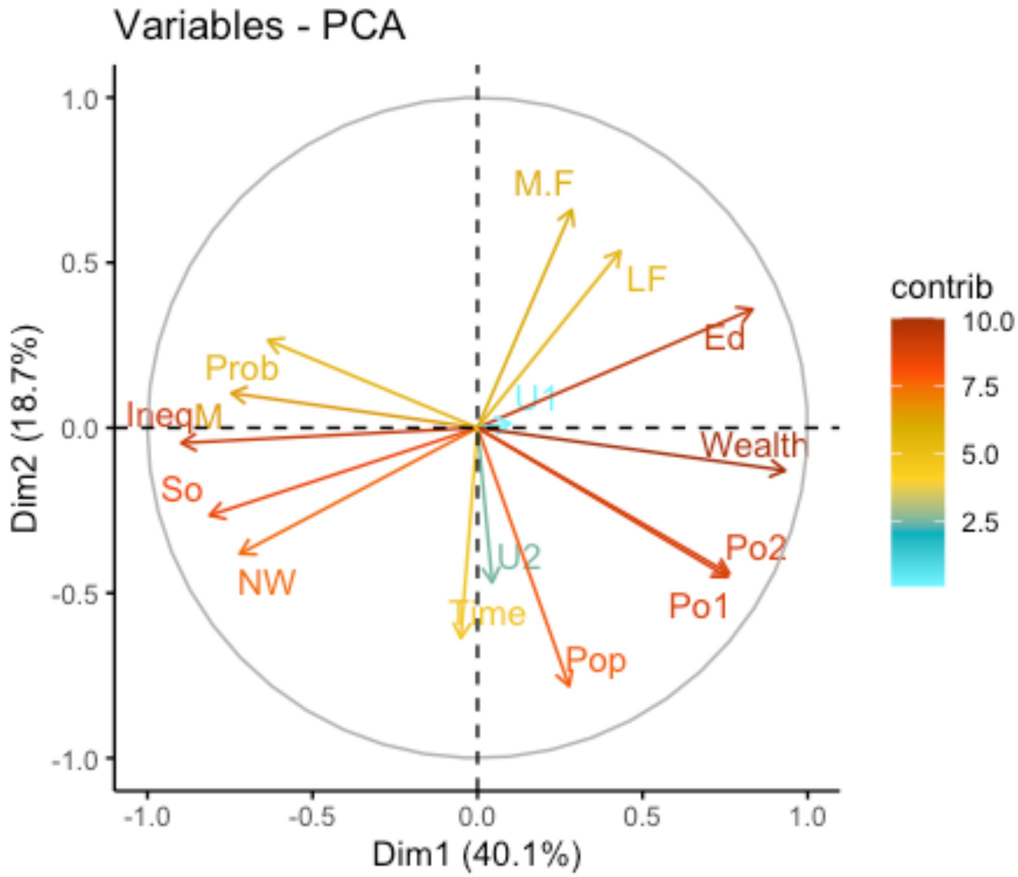
```
##Build up the principle components
PCA<- prcomp(data[,1:15], center = TRUE, scale=TRUE )
summary(PCA)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC
7
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.5672
9
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.0214
5
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.9214
2
##                            PC8     PC9    PC10    PC11    PC12    PC13    P
C14
## Standard deviation     0.55444 0.48493 0.44708 0.41915 0.35804 0.26333 0.2
418
## Proportion of Variance 0.02049 0.01568 0.01333 0.01171 0.00855 0.00462 0.0
```

```
039
## Cumulative Proportion  0.94191 0.95759 0.97091 0.98263 0.99117 0.99579 0.9
997
##                           PC15
## Standard deviation     0.06793
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000

##Calculate the eigenvalue value
Eig_value<- get_eigenvalue(PCA)
Eig_value

##         eigenvalue variance.percent cumulative.variance.percent
## Dim.1  6.018952657       40.1263510                    40.12635
## Dim.2  2.801847026       18.6789802                    58.80533
## Dim.3  2.004944334       13.3662956                    72.17163
## Dim.4  1.162207801        7.7480520                    79.91968
## Dim.5  0.958298972        6.3886598                    86.30834
## Dim.6  0.553193900        3.6879593                    89.99630
## Dim.7  0.321818687        2.1454579                    92.14176
## Dim.8  0.307401270        2.0493418                    94.19110
## Dim.9  0.235155292        1.5677019                    95.75880
## Dim.10 0.199880931        1.3325395                    97.09134
## Dim.11 0.175685403        1.1712360                    98.26258
## Dim.12 0.128190107        0.8546007                    99.11718
## Dim.13 0.069341691        0.4622779                    99.57945
## Dim.14 0.058467765        0.3897851                    99.96924
## Dim.15 0.004614165        0.0307611                   100.00000

##Plot the variation of each PCA components
fviz_eig(PCA, barfill = 'green')
```




```
fviz_pca_var(PCA, col.var = "contrib", gradient.cols=c("#70f6ff","#00AFBB","#ffd224","#d8ac00","#FC4E07","#a73203"), repel =TRUE, ggtheme= theme_classic() )
```




```
##We use the first 4 components to build up the linear regression model
Crime_Matrix<- cbind(PCA$x[,1:4], data[,16])
Regression_model<- lm(V5~., data=as.data.frame(Crime_Matrix))
summary(Regression_model)

##
## Call:
## lm(formula = V5 ~ ., data = as.data.frame(Crime_Matrix))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -557.76 -210.91  -29.08  197.26  810.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      49.07  18.443  < 2e-16 ***
## PC1            65.22      20.22   3.225  0.00244 **
## PC2           -70.08      29.63  -2.365  0.02273 *
## PC3            25.19      35.03   0.719  0.47602
## PC4            69.45      46.01   1.509  0.13872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 336.4 on 42 degrees of freedom
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.2433
## F-statistic: 4.698 on 4 and 42 DF,  p-value: 0.003178

##AIC and BIC inspection
AIC(Regression_model)

## [1] 687.0241

BIC(Regression_model)

## [1] 698.125

##Old model's AIC is 650, BIC is 681. The new model with PCA is better than t
he old model

##Use constructed model to predict crime based on the new data
##For convenient inputting, new data was put in a new txt file
Predict_data<-read.table('test.txt', header=TRUE)
Prediction<- data.frame(predict(PCA, Predict_data))
Predicted_Crime<-predict(Regression_model, Prediction)
Predicted_Crime

##        1
## 1112.678
```