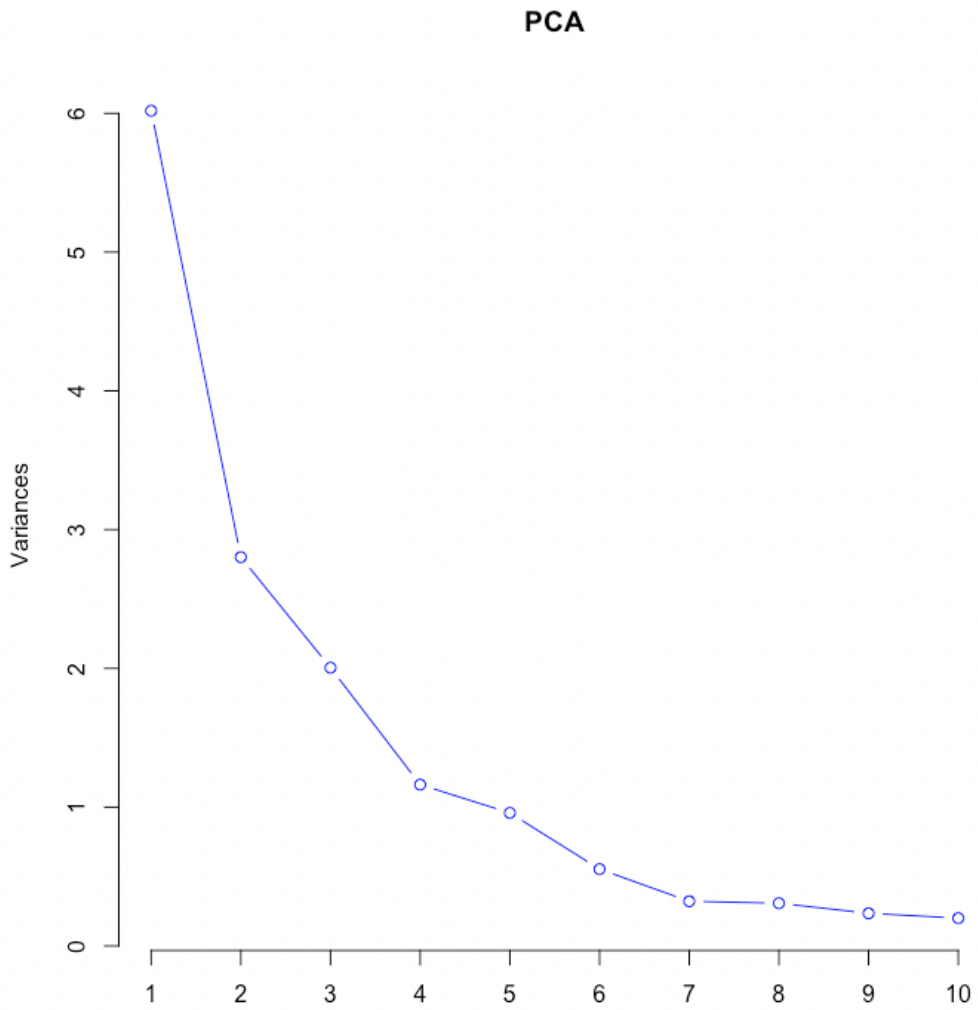
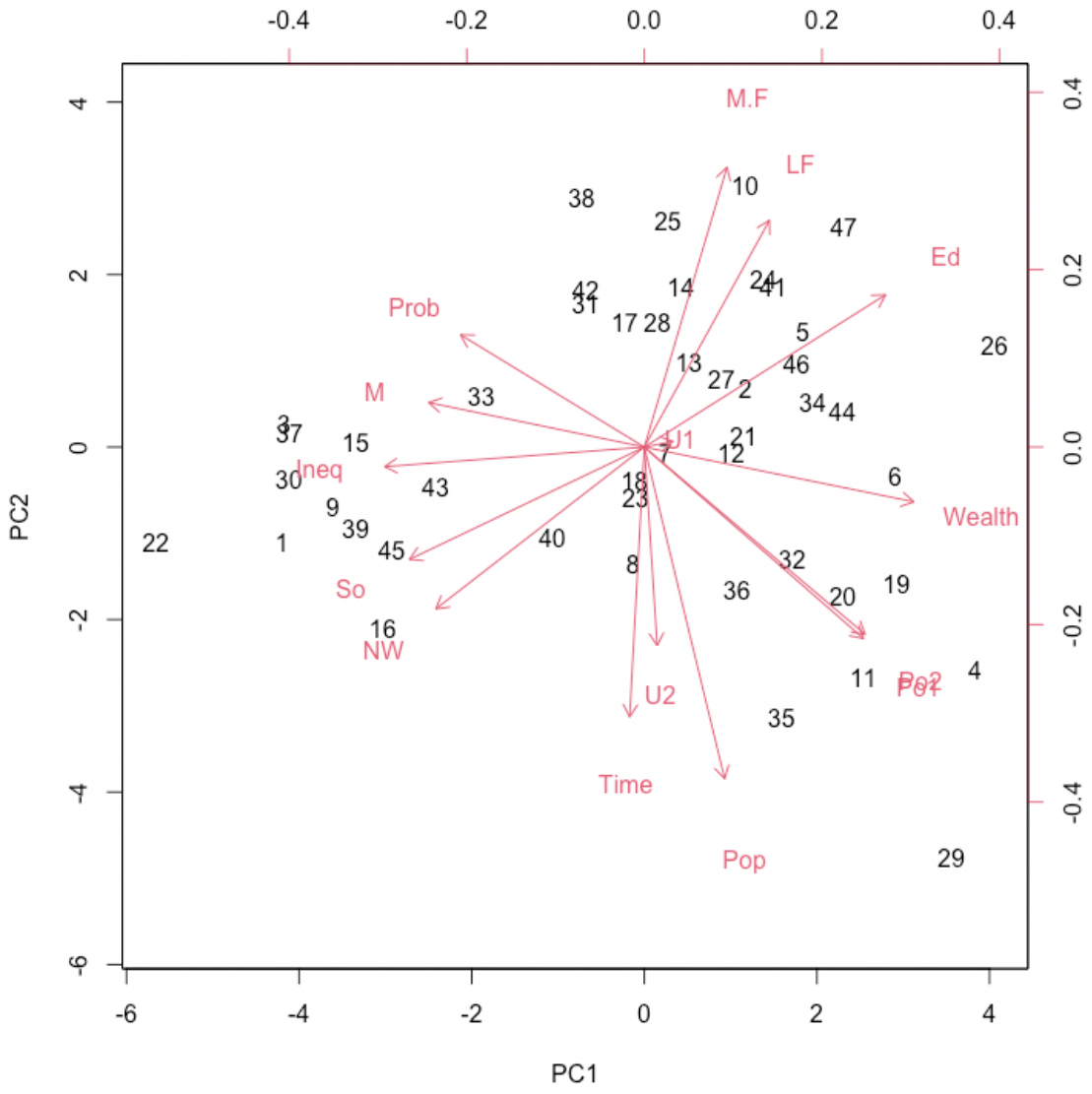
ISYE 6501 Homework 6

- ***Questions 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)!)***

I began by running the prcomp() function as mentioned in the above question. As suggested in the office hours, I ran a screeplot to see how many principal components I should use. The graph is as follows:



We will come back to this graph later to choose the principal components, but for now we will do some more exploratory graphing and then a test of the complete model to make sure it functions correctly. I then graphed a biplot() which shows the similar patterns among dependent variables, as follows:



As seen, variables like Po1 and Po2, are extremely correlated, which is to be expected. Because we want to be able to explain the data to those that are not familiar with PCA, we want to be able to move back to the original variables and unscale the data. To check our model, I first practiced using all 15 principal components. As mentioned in the office hours, if you run your linear model on all 15 principal components, then you should get the exact same answer as if you ran a linear model on the original data.

So we know what we are comparing the 15 principal components against, my general linear model against the original data “original_lm = lm(Crime~., data = crime_data)” gave the following output:

```
Coefficients:
              Estimate St
(Intercept) -5.984e+03  1
M            8.783e+01  4
So          -3.803e+00  1
Ed           1.883e+02  6
Po1          1.928e+02  1
Po2         -1.094e+02  1
LF          -6.638e+02  1
M.F          1.741e+01  2
Pop         -7.330e-01  1
NW           4.204e+00  6
U1          -5.827e+03  4
U2           1.678e+02  8
Wealth       9.617e-02  1
Ineq         7.067e+01  2
```

We will now try and match the data the prcomp() 15 principal components to the above. To do this, let’s discuss the following function from the lecture and piazza:

$$\hat{Y} = \left(\hat{\beta}_0 - \sum_{j=1}^{k} \hat{\beta}_j \frac{\bar{x}_j}{S_j}\right) + \sum_{j=1}^{k} \left(\frac{\hat{\beta}_j}{S_j}\right) x_j$$

The following explains the above function and will be used to transform the data back to the original variables and descaled:

- β_0 is the intercept of the PCA Model which is represented by the first column of $coefficients
- $\hat{\beta}_j$ is the transformation using the $rotation, using the function A=VB where B is the beta coefficients of the principal components

- $(\bar{x}_j/S_j)$ is the mean of the unscaled data and the standard deviation of the unscaled data which helps center and scale (and also reverse)
- x_j are the actual values of the 15 predictors in the crime data set

The full code will be at the end of the homework, but for the sake of above defintions, here is a screenshot of the implementation:

```
B0 = lmPCA$coefficients[1]
BSubj = PCA$rotation[,1:15]%*%lmPCA$coefficients[2:16]
unscaled_mean = sapply(crime_data[,1:15], mean)
unscaled_SD = sapply(crime_data[,1:15], sd)

originalalpha = BSubj/unscaled_SD
originalbeta0 = B0 - sum(BSubj*(unscaled_mean/unscaled_SD))
```

Now that we have implemented the formula, we can compare to the coefficients from the previous page and see if they are the same. We do this by printing the "originalalpha" and "originalbeta0" from above:

```
> originalalpha
                [,1]
M       8.783017e+01
So     -3.803450e+00
Ed      1.883243e+02
Po1     1.928043e+02
Po2    -1.094219e+02
LF     -6.638261e+02
M.F     1.740686e+01
Pop    -7.330081e-01
NW      4.204461e+00
U1     -5.827103e+03
U2      1.677997e+02
Wealth  9.616624e-02
Ineq    7.067210e+01
Prob   -4.855266e+03
Time   -3.479018e+00
```

```
> originalbeta0
(Intercept)
  -5984.288
```

As you can see, by implemting the formula above, we have been able to get our 15 principal components to match a general linear model perfectly.

Now that we have shown we know how to reverse the variables and unscale, we can rerun all of the code by this time choosing how many principal components we want to use. If we review our screeplot again, we note that the drop off seems to dimish are 5, so we will run our model with 5 principal components. Utilizing the same fnction above, we get the following coefficients and intercept after unscaling our data with the 5 principal components:

```
> originalalpha_actual
                [,1]
M         -16.9307630
So         21.3436771
Ed         12.8297238
Po1        21.3521593
Po2        23.0883154
LF       -346.5657125
M.F        -8.2930969
Pop         1.0462155
NW          1.5009941
U1      -1509.9345216
U2          1.6883674
Wealth      0.0400119
Ineq       -6.9020218
Prob      144.9492678
Time       -0.9330765
```

```
> originalbeta0_actual
(Intercept)
   1666.485
```

We now rerun the sample data frame from last week's homework through our new model to see if we can get a new crime rate. When we do so, we get the following:

```
> #testing model on numbers from 8.2
> tester = 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 = 0.04, Time = 39.0)
>
> pred_tester = data.frame(predict(PCA, tester))
> tester_model = predict(lmPCA_actual, pred_tester)
> tester_model
       1
1388.926
```

This seems to be in line with our data based on what we see from the original crime data set.

Code for homework follows on the next page

```
library(MASS)
library(ggplot2)
#install.packages("GGally")
library(GGally)

set.seed(12)

#read data into table
crime_data = read.table("uscrime.txt", stringsAsFactors = FALSE, header = TRUE)

#run PCA using the PRCOMP() function
PCA = prcomp(crime_data[,1:15], scale = TRUE)
summary(PCA)

PCA$rotation
#screeplot to get visual
screeplot(PCA, type = 'lines', col= "blue")

PCA$x

#plots variables in 2-dimensional view. those close have similar data patterns
biplot(PCA, scale = 0)

#practicing reverse of data
PC = PCA$x[,1:15]
crime_data_PC = cbind(PC,crime_data[,16])
lmPCA = lm(V16~., data = as.data.frame(crime_data_PC))
summary(lmPCA)

#general linear model to make sure mth is right
original_lm = lm(Crime~., data = crime_data)
summary(original_lm)

#named vector of coefficients from our model
lmPCA$coefficients

#reverse and unscale per lectures and piazza
B0 = lmPCA$coefficients[1]
BSubj = PCA$rotation[,1:15]%*%lmPCA$coefficients[2:16]
unscaled_mean = sapply(crime_data[,1:15], mean)
unscaled_SD = sapply(crime_data[,1:15], sd)

originalalpha = BSubj/unscaled_SD
originalbeta0 = B0 - sum(BSubj*(unscaled_mean/unscaled_SD))
```

```
originalalpha
originalbeta0


#repeat all of the above now that it works. Only using 5 PCs
PCactual = PCA$x[,1:5]
crime_data_PC_actual = cbind(PCactual,crime_data[,16])
lmPCA_actual = lm(V6~., data = as.data.frame(crime_data_PC_actual))
summary(lmPCA_actual)

B0actual = lmPCA_actual$coefficients[1]
BSubjactual = PCA$rotation[,1:4]%*%lmPCA_actual$coefficients[2:5]
unscaled_mean = sapply(crime_data[,1:15], mean)
unscaled_SD = sapply(crime_data[,1:15], sd)

originalalpha_actual = BSubjactual/unscaled_SD
originalbeta0_actual = B0actual - sum(BSubjactual*(unscaled_mean/unscaled_SD))
originalalpha_actual
originalbeta0_actual

#testing model on numbers from 8.2
tester = 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 = 0.04, Time = 39.0)

pred_tester = data.frame(predict(PCA, tester))
tester_model = predict(lmPCA_actual, pred_tester)
tester_model
```

Research Credit

- https://piazza.com/class/l502h1j6yrw410/post/1621
- https://stackoverflow.com/questions/12861734/calculating-standard-deviation-of-each-row
- https://stats.stackexchange.com/questions/74622/converting-standardized-betas-back-to-original-variables
- http://www.billconnelly.net/?p=697
- https://www.datacamp.com/tutorial/pca-analysis-r