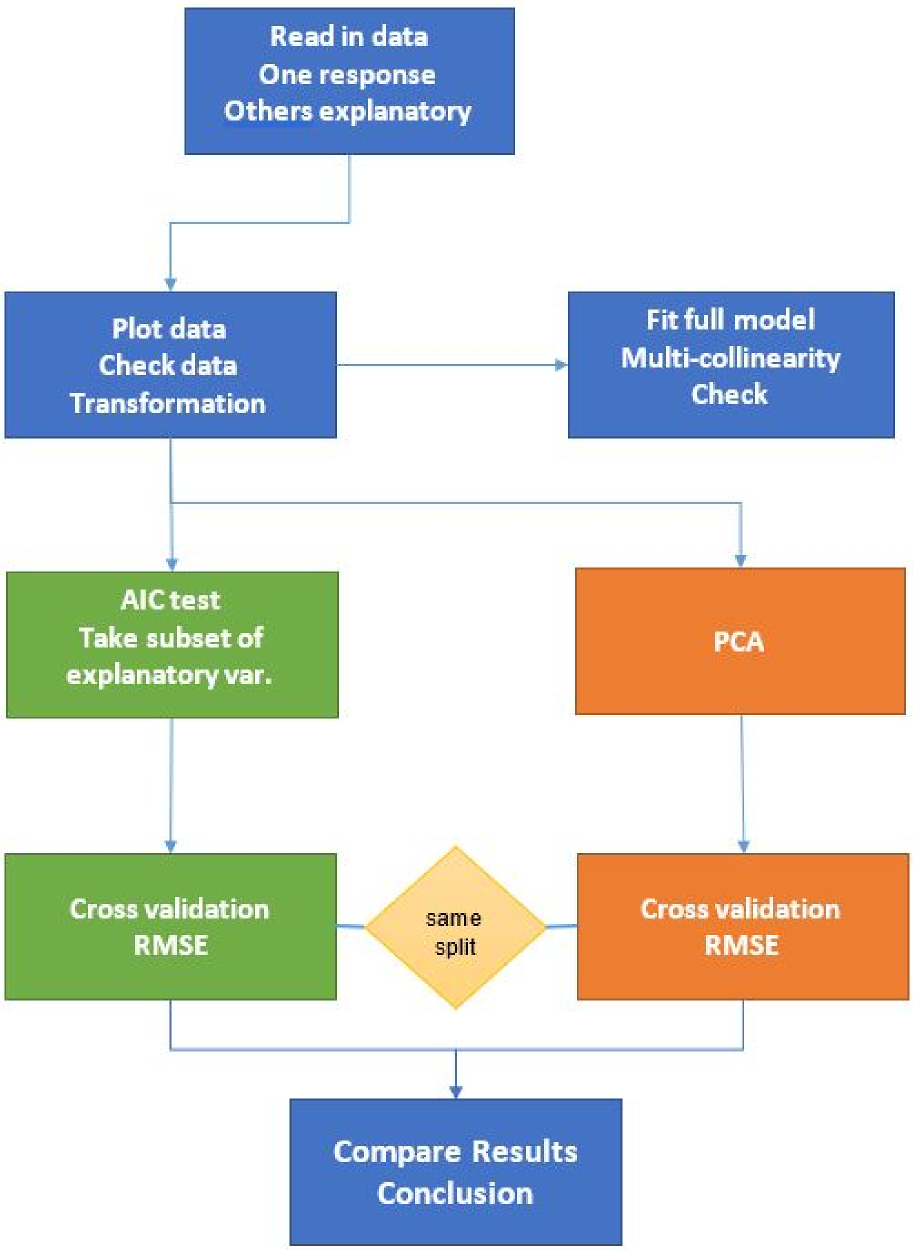
STAT 350 Final Project

Dawu Liu (dla189), Siyul Sam Byun (siyulb), Wing Yuk Cheung (wingyukc)

Please have a look at the flow chart of the project:



In this project, we will look at the ingredients of concrete and analyze what determines the compressive strength of concrete using the Concrete Compressive Strength Data Set. The data can be downloaded from the website: https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength Our goal is to fit Multiple Linear Regression models into the data, and get findings on the relationship between the response variable and the explanatory variables

All the variables are numeric and non-negative:

One responsive variable: Concrete compressive strength (Strength) measured in MPa.

Eight explanatory variables: seven of them are cement, blast furnace slag (Slag), fly ash, water, superplasticzier (SP), coarse aggregate (CA), fine aggregate (FA). Those seven variables are ingredients of the concrete, measured in KG per cubic meter. And the eighth explanatory variable is age which is measured in days.

First of all, fit everything in a linear model and see how things look like.

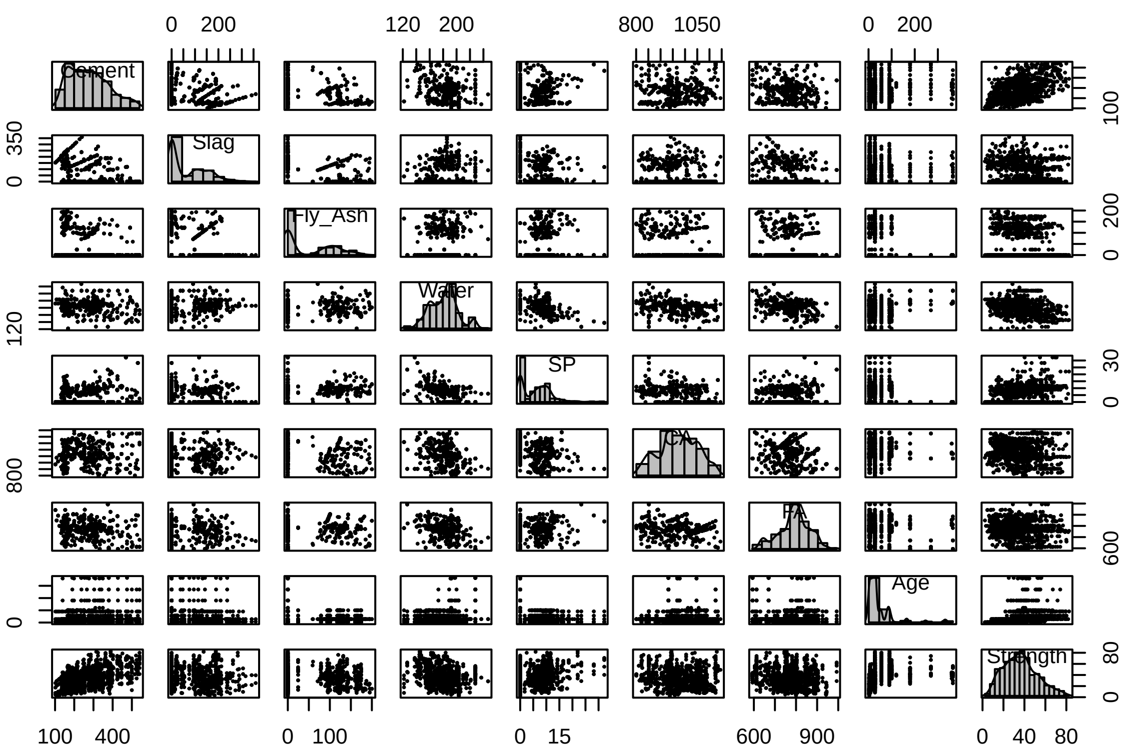
```
summary(lm(Strength ~ ., data=concrete))$adj.r.squared  # Adjusted R squared
```

```
## [1] 0.6124517
```

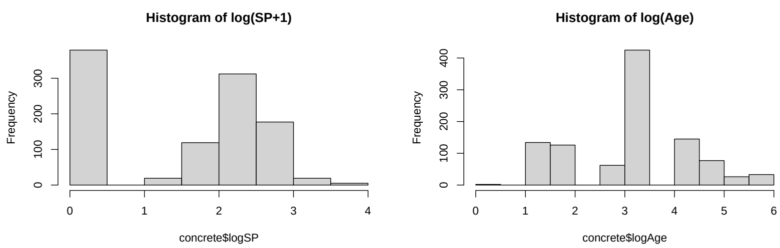
```
sum(lm(Strength ~ ., data=concrete)$residuals^2)/(n-9) # MSE
```

```
## [1] 108.1569
```

Adjusted R squared and MSE are not looking too impressive. Let us see if we can spot any problems



In the scatter plot and histograms, some of the explanatory variables Slag, Fly_Ash, SP, and Age are heavily right skewed. If look closely, we can see that this is due to those zero values and small values close to zero. If we don't look at those value, the distribution of Slag and Flu_Ash appear to not have much skewness, but SP and Age still show heavy right skewness. After trying different transformations, we ended up taking log on variable Age. For SP, because it has zero values, to deal with this issue, an constant 1 was added to the variable SP then a log transformation was performed, i.e. log(SP+1). We tried different transformations on the responsible y as well, but non of them provided much help. For easier interpretation of the model, we will just leave y as it is. Now the histogram of log(Age) and log(SP+1) are significantly improved. The rest of the variables remain unchanged.



Therefore, our old full model was:

$y = \beta_0 + \beta_{Cement} * x_{Cement} + \beta_{Slag} * x_{Slag} + ... + \beta_{SP} * x_{SP} + \beta_{Age} * x_{Age} + \epsilon$

Now, we change x_{SP} and x_{Age} to $log(x_{SP} + 1)$ and $log(x_{Age})$.

Define: $x^*_{Age} = log(x_{Age})$ and $x^*_{SP} = log(x_{SP} + 1)$. New full model is:

$y = \beta_0 + \beta_{Cement} * x_{Cement} + \beta_{Slag} * x_{Slag} + ... + \beta^*_{SP} * x^*_{SP} + \beta^*_{Age} * x^*_{Age} + \epsilon$

Due to the scaling differences, the explanatory variables are standardized before any further analysis.

Our new explanatory variable values are now: $z_{ij} = (x_{ij} - \bar{x_j})/sd(x_j)$.

Therefore, our standardized full model is:

$y = \beta_0 + \beta_{Cement} * z_{Cement} + \beta_{Slag} * z_{Slag} + ... + \beta_{SP} * z_{SP} + \beta_{Age} * z_{Age} + \epsilon$

Full model analysis

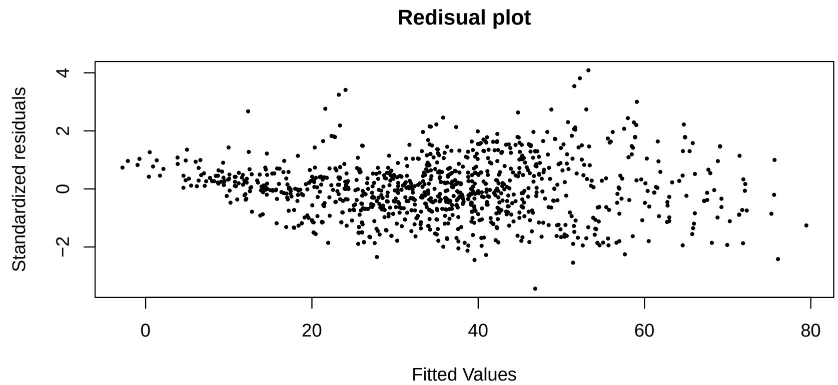
```
FULL.model = lm(Strength ~. , data = concrete)
summary(FULL.model)$adj.r.squared  # Adjusted R squared after transformation
```

```
## [1] 0.8245296
```

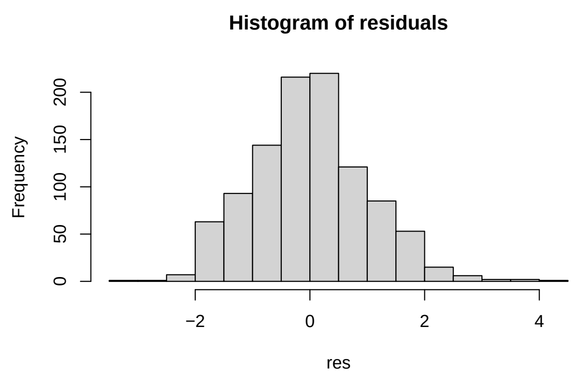
```
sum(FULL.model$residuals^2)/(n-9) # MSE after transformation
```

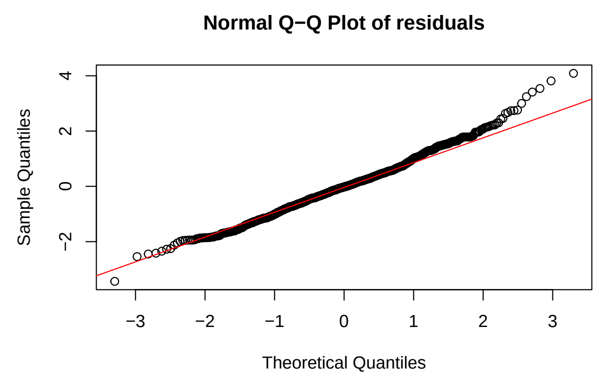
```
## [1] 48.97023
```

Adjusted R squared increased from 0.612 to 0.825, and MSE decreased from 108.16 to 48.97. That's significant improvement.



The left side of the residuals looks a little bit narrow, the variances of residuals become slightly larger as the fitted value gets larger, but no major issue. Also, the residuals are not showing patterns.



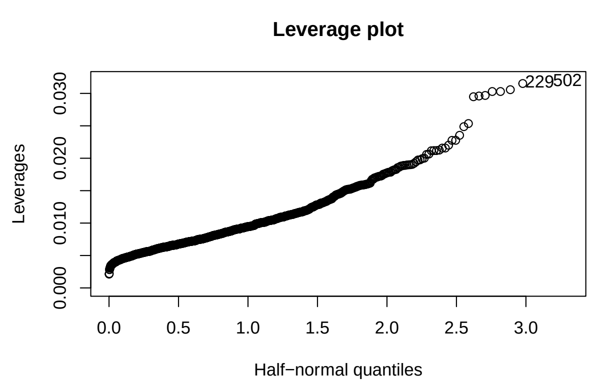


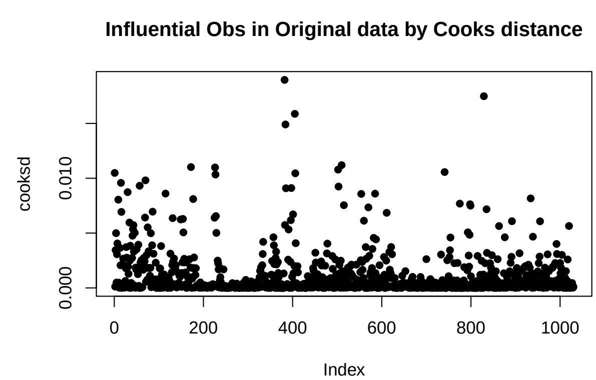
The residual histogram looks approximately normal. QQ-plot shows some right skewness.

VIF values:

```
##   Cement     Slag  Fly_Ash    Water    logSP       CA       FA   logAge
## 7.589391 7.391087 6.943933 6.142170 3.281782 4.731465 6.883141 1.052909
```

Leverages and Cooks distance:





The leverage plot shows there might be potential outliers, but they are not extremely off from the others, so we do not remove those observations. Even with the cooks' distance, we do not notice any significant outliers. Therefore, we will keep all the observations in the full model.

What if we don't really need all the explanatory variables to explain the response variable? In the following sections, we will be doing AIC test followed by its k-fold cross validation and PCA followed by its k-fold cross validation (both cross validations are using the same split), and try to extract some information while make things more cost efficient.

Akaike Information Criterion

AIC is used to compare different possible models and determine which one is the best fit for the data. It can also help us to identify the most/least variables.

Forward search.

```
summ.fit.forward$outmat # forward
```

```
##          Cement Slag Fly_Ash Water logSP CA  FA  logAge
## 1  ( 1 ) " "    " "  " "     " "   " "   " " " " "*"
## 2  ( 1 ) "*"    " "  " "     " "   " "   " " " " "*"
## 3  ( 1 ) "*"    " "  " "     " "   "*"   " " " " "*"
## 4  ( 1 ) "*"    "*"  " "     " "   "*"   " " " " "*"
## 5  ( 1 ) "*"    "*"  " "     "*"   "*"   " " " " "*"
## 6  ( 1 ) "*"    "*"  "*"     "*"   "*"   " " " " "*"
## 7  ( 1 ) "*"    "*"  "*"     "*"   "*"   "*" " " "*"
## 8  ( 1 ) "*"    "*"  "*"     "*"   "*"   "*" "*" "*"
```

```
summ.fit.forward$adjr2
```

```
## [1] 0.3042319 0.5528057 0.6942405 0.7789862 0.8118589 0.8190956 0.8202666
## [8] 0.8245296
```

FA is identified as the least important variable, followed by CA and Fly__Ash. For the adjusted R squared, only keeping the most important variable logAge gives us 0.304, removing the three least important variables FA, CA, and Fly__Ash will still maintain 0.819 for the adjusted R squared.

Backward search

```
summ.fit.backward$outmat # backward
```

```
##          Cement Slag Fly_Ash Water logSP CA  FA  logAge
## 1  ( 1 ) " "    " "  " "     " "   " "   " " " " "*"
## 2  ( 1 ) "*"    " "  " "     " "   " "   " " " " "*"
## 3  ( 1 ) "*"    "*"  " "     " "   " "   " " " " "*"
## 4  ( 1 ) "*"    "*"  "*"     " "   " "   " " " " "*"
## 5  ( 1 ) "*"    "*"  "*"     " "   " "   " " "*" "*"
## 6  ( 1 ) "*"    "*"  "*"     " "   " "   "*" "*" "*"
## 7  ( 1 ) "*"    "*"  "*"     " "   "*"   "*" "*" "*"
## 8  ( 1 ) "*"    "*"  "*"     "*"   "*"   "*" "*" "*"
```

```r
summm.fit.backward$adjr2
```

```
## [1] 0.3042319 0.5528057 0.6386487 0.7233351 0.7635009 0.8089210 0.8228881
## [8] 0.8245296
```

Water is identified as the least important variable, followed by logSP and CA. Removing the two least important variables Water, and logSP will maintain 0.809 for the adjusted R squared.

Stepwise search.

```r
summm.fit.all$outmat #stepwise
```

```
##          Cement Slag Fly_Ash Water logSP CA  FA  logAge
## 1  ( 1 ) " "    " "  " "     " "   " "   " " " " "*"
## 2  ( 1 ) "*"    " "  " "     " "   " "   " " " " "*"
## 3  ( 1 ) "*"    " "  " "     " "   "*"   " " " " "*"
## 4  ( 1 ) "*"    "*"  " "     "*"   " "   " " " " "*"
## 5  ( 1 ) "*"    "*"  "*"     "*"   " "   " " " " "*"
## 6  ( 1 ) "*"    "*"  "*"     "*"   "*"   " " " " "*"
## 7  ( 1 ) "*"    "*"  "*"     " "   "*"   "*" "*" "*"
## 8  ( 1 ) "*"    "*"  "*"     "*"   "*"   "*" "*" "*"
```

```r
summm.fit.all$adjr2
```

```
## [1] 0.3042319 0.5528057 0.6942405 0.7799138 0.8130718 0.8190956 0.8228881
## [8] 0.8245296
```

FA and CA are least important, followed by logSP and Fly_Ash, notice that water and logSP leave the model then re-enters, and leave. This can sometimes happen since effects are conditional on other variables being in or out of the model. Removing the three least important variables FA, CA, and logSP will maintain 0.813 for the adjusted R squared.

FA and CA are the 'winners' of this competition, with two and three votes respectively in the 'top 3' spot for least important variables. Now, let's extract the AIC values from any suggested model by above, also compare it with a full model with all variables in it. Here are the AIC values:

Full model: 4016.909
Removing CA and FA: 4046.338
Removing CA, FA, and Fly_Ash: 4085.745
Removing Water and logSP: 4102.698
Removing CA, FA, and logSP:4079.083

The full model still has the lowest AIC value. After some testing we found out that removing any variable will increase the AIC value. In conclusion, we will remove CA and FA in our model. Only removing CA and FA will have the lowest AIC value and highest adjusted r squared compare to other adjusted model, and adjusted R squared is 0.8191 compare to 0.8238 of the full model. Also the MSE is 50.49 compare to 49.17 of the full model. That's some minor sacrifice for removing 25% (two out of eight) of the explanatory variables.

Finally, let's do a k-fold cross validation (k=4) and find the $SSE's$ and $RMSE_{cv}$. In k-fold cross validation, the data is evenly split into k different groups, where the model is built using k-1 groups of the data, and tested on the remaining group of the data. The idea is to test the model's ability to predict new data. Then we calculate the sum of squares of errors (SSE) of prediction in each test fold and the Root Mean Square Error (RMSE), the lower the better. And here is the result:

```
## [1] "The SSE for AIC adjusted linear model on test data 1 is 39321.4168590015"
## [1] "The SSE for AIC adjusted linear model on test data 2 is 39032.4746670695"
## [1] "The SSE for AIC adjusted linear model on test data 3 is 38126.1525227549"
## [1] "The SSE for AIC adjusted linear model on test data 4 is 38463.7723990549"

## [1] "The RMSE for AIC adjusted linear model is 196.814517025473"
```

This result will be used later for comparing to other models.

Principal Component Analysis

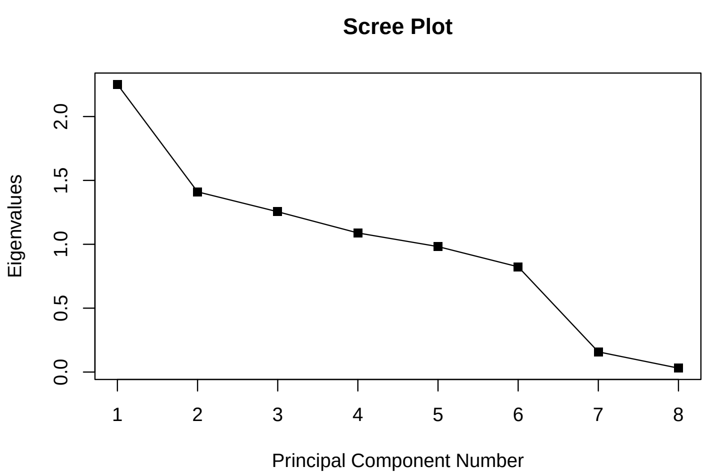
Principal Component Analysis (PCA) can give us insights and discover underlying patterns of the data. It helps fix the multicollinearity issues and can also reduce the dimension of the data (i.e. using less variales to explain the data without losing much information).

Briefly recall on the mechanics of PCA: we transform the original data set and express them in new orthogonal eigenvectors basis. To do this, we perform spectral decomposition on the symmetric covariance/correlation matrix of the explanatory variables and derive the eigenvalues and eigenvectors. The proportion of each individual eigenvalue to the sum of eigenvalues is the proportion of total variance explained by the that PC, and the corresponding eigenvector is the PC (basis). PCs are ordered by eigenvalues in descending order. Our new explanatory variables are just a linear transformation of the original data set, while the responsible variable remains unchanged.

Eigenvalue table:

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 2.25161793        28.145224                    28.14522
## Dim.2 1.41051620        17.631452                    45.77668
## Dim.3 1.25469276        15.683660                    61.46034
## Dim.4 1.08879057        13.609882                    75.07022
## Dim.5 0.98199774        12.274972                    87.34519
## Dim.6 0.82373679        10.296710                    97.64190
## Dim.7 0.15795833         1.974479                    99.61638
## Dim.8 0.03068968         0.383621                   100.00000
```

We obtain 87.35% of the variance by keeping five PC, and 97.64% of the variance by keeping six PCs.

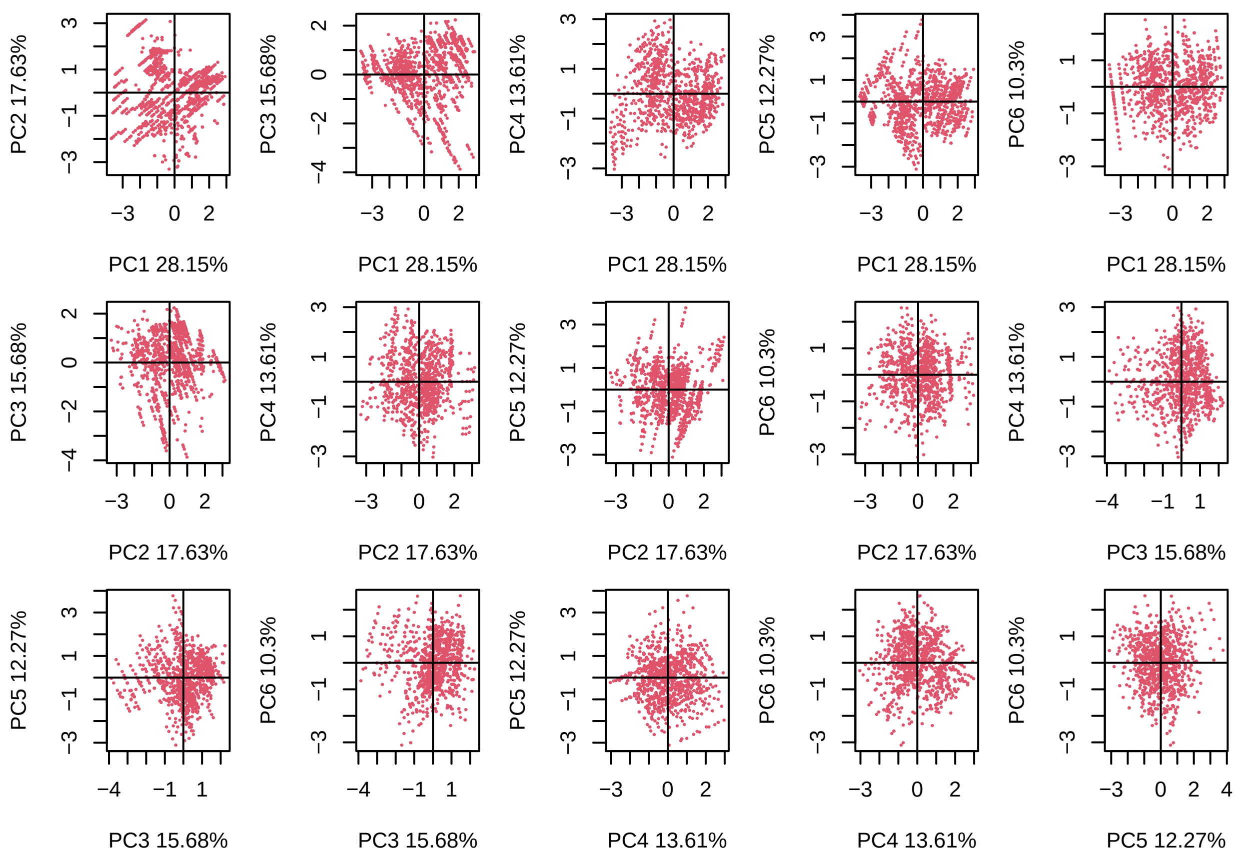


The ‘bend’ occurs at PC7, in the scree plot, indicating keep the first six PCs. The above two checks suggest us to keep five or six PCS. We will keep six PCs in our analysis. Now, let’s have a look at the first six PCs (the eigenvectors).

```
##  Cement    Slag Fly_Ash   Water   logSP      CA      FA  logAge
## -0.1510 -0.2066  0.4696 -0.5172  0.5314 -0.0194  0.3858 -0.1221
##  Cement    Slag Fly_Ash   Water   logSP      CA      FA  logAge
##  0.4265 -0.6543 -0.1119 -0.1655 -0.2354  0.5337  0.0563 -0.0815
##  Cement    Slag Fly_Ash   Water   logSP      CA      FA  logAge
## -0.6453 -0.1231  0.4218  0.2428 -0.1909  0.4880 -0.1639  0.1735
##  Cement    Slag Fly_Ash   Water   logSP      CA      FA  logAge
## -0.3296  0.2658 -0.3041 -0.1488 -0.2595  0.2242  0.3342 -0.6905
##  Cement    Slag Fly_Ash   Water   logSP      CA      FA  logAge
##  0.1894  0.3241  0.1004 -0.2403  0.3073  0.3564 -0.6951 -0.2943
##  Cement    Slag Fly_Ash   Water   logSP      CA      FA  logAge
## -0.0943  0.3318 -0.3792 -0.4433  0.0395  0.3593  0.1675  0.6181
```

We can see that most of the PC's are showing contrast between different ingredients, indicating some of the mix ratio of the concrete. PC3 is strongly negatively associated with cement, and PC4 is strongly negatively associated with log(Age)

Now let's have a look at the plot of each pair of the transformed data.



Each pair of transformed data appear to be centered around the origin. Appeared to close to approximately bivariate normal.

Therefore, the PCA adjusted model is defined by fitting a linear model on the response variable and the transformed data, which is:

$y = \beta_0 + \beta_{PC1} * PC1 + \beta_{PC2} * PC2 + ... + \beta_{PC5} * PC5 + \beta_{PC6} * PC6 + \epsilon$

```r
round(PCA.summary$coefficients,4)[,1] # coefficients (beta's)
```

```
## (Intercept)         PC1         PC2         PC3         PC4         PC5
##     35.8178      0.3523     -0.4489     -6.3166     -9.4310      3.3268
##         PC6
##      8.7934
```

Most of the PC might indicates some of the ratio we need to pay attention to when mixing concrete. PC3 has negative coefficient, suggesting having more cement might help with the compression strength. PC4 is negatively associated with log(Age) and it has a negative coefficient with the largest magnitude, indicating concrete get harder as it ages. PC6 has a relatively high and positive coefficient, and it is positively associated with log(Age), which also supports the point that concrete get harder as it ages.

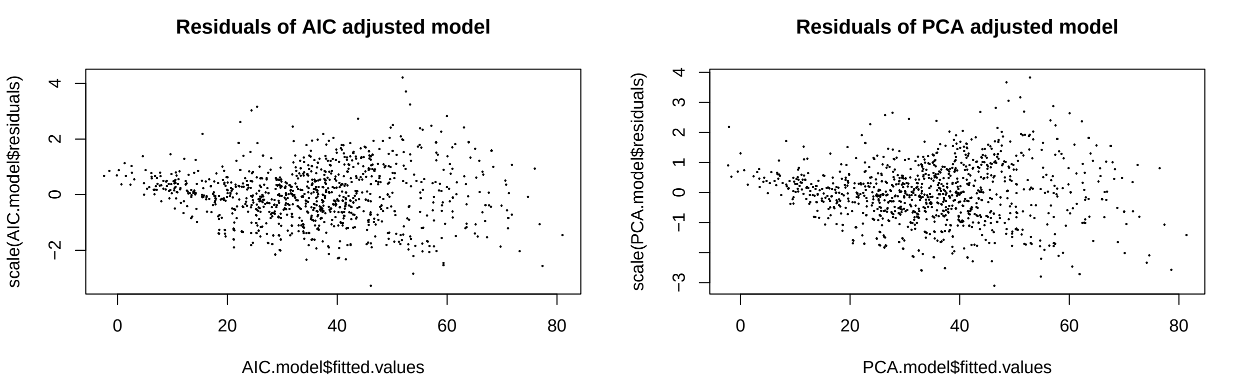
Now, let us do the same test train split we did in AIC on the PCA data set for prediction models. Then do a k-fold cross validation (k=4) and find the $SSE's$ and $RMSE_{cv}$

```
## [1] "The SSE for PCA adjusted linear model on test data 1 is 45673.4546075537"
## [1] "The SSE for PCA adjusted linear model on test data 2 is 47358.7352026733"
## [1] "The SSE for PCA adjusted linear model on test data 3 is 45037.1971504774"
## [1] "The SSE for PCA adjusted linear model on test data 4 is 47277.9902845657"
```

```
## [1] 215.2599
```

```
## [1] "The RMSE for AIC adjusted linear model is 215.259945905683"
```

Final conclusion



Residuals for both AIC and PCA model are very similar, there doesn't appear to be any pattern. The variance of residuals seems to slightly increase as the predicted value gets larger, but no major issues here.

Model	Adj.R.Squared	MSE	RMSE
Full Model(8 Explanatories)	0.8245	48.97023	198.5763
AIC Model(6 Explanatories)	0.8191	50.48674	196.8145
PC Model(6 PCs)	0.7944	57.38534	215.2599

All three models showing close results, with PCA model being the worst. Full model has the highest adjusted r squared and the lowest MSE, but RMSE is slightly higher than AIC. Indicating it might be slightly over fitting. We ran the test with random seed multiple times, all yielded the same conclusion where full model has higher RMSE than AIC adjusted model. AIC model indicates that adding variable CA and FA doesn't give much information, and would possibly over fit the model. Overall, AIC model is considered the best out of the three, it takes the most important variables, removes overfitting issue, while being cost efficient.

```
summary(AIC.model)
```

```
##
## Call:
## lm(formula = Strength ~ ., data = AIC.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.2118  -4.3855  -0.0211   4.1856  29.8596
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.8178     0.2214 161.782  < 2e-16 ***
## Cement       10.8868     0.3013  36.137  < 2e-16 ***
## Slag          6.6727     0.2991  22.306  < 2e-16 ***
## Fly_Ash       2.5074     0.3871   6.478 1.44e-10 ***
## Water        -4.3457     0.2957 -14.694  < 2e-16 ***
## logSP         2.2313     0.3766   5.924 4.28e-09 ***
## logAge       10.2071     0.2258  45.212  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.105 on 1023 degrees of freedom
## Multiple R-squared:  0.8202, Adjusted R-squared:  0.8191
## F-statistic: 777.5 on 6 and 1023 DF,  p-value: < 2.2e-16
```

In the AIC adjusted model, all parameters have a p-value that is close to zero, showing significance. Cement and Age have the two largest positive coefficients, indicating relatively more cement can help with the compressive strength and concrete gets harder as it ages at a logarithmic growth rate, this comes to a similar conclusion that the PCA adjusted model suggested. Water has a negative coefficient, adding more water to the concrete increases workability, but adding too much might decrease the strength and durability.

It was a enjoyable linear regression experiment on analyzing compressive strength of concrete. We were able to build three different models using materials we learned within and outside of this class. By applying data transformation and validation techniques, we found the best model among the three we built that is easy to interpret, cost efficient, and relatively accurate. The compressive strength of concrete can have significant effect and ensure the quality of the product. It was fun to learn this property of concrete from a statistical view, and also we did some research online to extend our domain knowledge as well.

Code used:

```
library(factoextra)
library(faraway)
library(dplyr)
library(readxl)
library(leaps)
library(MASS)
library(MESS)

#Read in Data and transformation#
concrete = read_excel("C:/Users/Dawu/Desktop/STAT350/Final project/Concrete_Data.xls")
name = c('Cement', 'Slag', 'Fly_Ash', 'Water', 'SP',
         'CA', 'FA', 'Age', 'Strength')
colnames(concrete) = name
n = length(concrete$Strength) # sample size

summary(lm(Strength ~ ., data=concrete))$adj.r.squared  # Adjusted R squared
sum(lm(Strength ~ ., data=concrete)$residuals^2)/(n-9) # MSE

pairs(concrete, diag.panel = panel.hist, cex=0.2)
concrete$SP=log(concrete$SP+1)
concrete$Age=log(concrete$Age)
names(concrete)[names(concrete) == 'SP'] = "logSP"
names(concrete)[names(concrete) == 'Age'] = "logAge"

par(mfrow=c(1,2))
hist(concrete$logSP, main = "Histogram of log(SP)")
hist(concrete$logAge, main = "Histogram of log(Age)")
concrete[,1:8] = scale(concrete[,1:8]) # standardize explanatory r.v.

# Full model analysis
FULL.model = lm(Strength ~. , data = concrete)
summary(FULL.model)$adj.r.squared  # Adjusted R squared after transformation
sum(FULL.model$residuals^2)/(n-9) # MSE after transformation

# Residuals plot
res = FULL.model$residuals
res = scale(res)
pred = FULL.model$fitted.values
plot(x=pred, y=res, xlab="Fitted Values", ylab="Standardized residuals",
     main = 'Redisual plot', pch=16, cex=.5)

# Histogram of residuals
par(mfrow=c(1,2))
hist(res, main="Histogram of residuals")
# Q-Q plot
qqnorm(res, main="Normal Q-Q Plot of residuals")
qqline(res, col="red")

# VIF
vif(FULL.model)

# Leverages
```

```r
par(mfrow=c(1,2))
hatv = hatvalues(FULL.model)
predictors = row.names(concrete)
halfnorm(hatv, labs=predictors, ylab="Leverages", main="Leverage plot")
# Cooks distance
cooksd = cooks.distance(FULL.model)
plot(cooksd, pch = 16, main="Influential Obs in Original data by Cooks distance")

# AIC
# forward
fit.forward <- regsubsets(Strength ~ .,data = concrete, nbest=1,nvmax=8,method="forward")
summ.fit.forward <- summary(fit.forward)
# backward
fit.backward <- regsubsets(Strength~.,data = concrete, nbest=1,nvmax=8,method="backward")
summ.fit.backward <- summary(fit.backward)
# stepwise
fit.all <- regsubsets(Strength ~ .,data = concrete, nbest=1,nvmax=8,method="exhaustive")
summ.fit.all <- summary(fit.all)

summ.fit.forward$outmat # forward
summ.fit.forward$adjr2

summ.fit.backward$outmat # backward
summ.fit.backward$adjr2

summ.fit.all$outmat #stepwise
summ.fit.all$adjr2

extractAIC(lm(Strength ~ . , data = concrete)) # full
extractAIC(lm(Strength ~ . -CA -FA , data = concrete)) # forward suggestion
extractAIC(lm(Strength ~ . -CA -FA - Fly_Ash, data = concrete)) # forward suggestion
extractAIC(lm(Strength ~ . -Water -logSP, data = concrete)) # backward suggestion
extractAIC(lm(Strength ~ . -CA -FA - logSP, data = concrete)) # stepwise suggestion
sum(lm(Strength ~ . -CA -FA , data = concrete)$residuals^2)/(n-7) # MSE for model without CA FA

#AIC data set
AIC.data = concrete[c('Cement','Slag','Fly_Ash', 'Water', 'logSP', 'logAge', 'Strength')]

# K-fold split index, will be used later for PCA as well
n <- length(AIC.data$Strength)
set.seed(350)
index <- sample(rep(1:4, each = ceiling(n /4))[1:n])

# Prints out SSE and RMSE
AIC.RMSE = 0
for(i in 1:4){
  AIC.train = AIC.data[index==i,]
  AIC.test = AIC.data[index!=i,]
  AIC.model = lm(Strength ~ . , data = AIC.data)
  AIC.prediction = predict(AIC.model, AIC.test)
  AIC.true = AIC.test$Strength
  AIC.SSE = sum((AIC.true - AIC.prediction)^2)
  print(paste0("The SSE for AIC adjusted linear model on test data ", i, " is ", AIC.SSE))
```

```
  AIC.RMSE = AIC.RMSE + AIC.SSE
}
AIC.RMSE = sqrt(AIC.RMSE/4)
print(paste0("The RMSE for AIC adjusted linear model is ", AIC.RMSE))


X = concrete[,1:8]
# PCA
PCA = prcomp(X, center = TRUE, scale. = FALSE)
X_eigen_table = get_eigenvalue(PCA)
X_eigen_table

plot(X_eigen_table[,1], type = "o", pch = 15, main = "Scree Plot",
     xlab = "Principal Component Number", ylab = "Eigenvalues")

# coefficients for PCs
for (i in 1:6) {print(round(PCA$rotation[,i],4))}

# scatter plot for PCs
par(mfrow=c(3, 5),mar= c(4,4,1,1), oma=c(0,0,0,0))
for (i in 1:5){
  for (j in (i+1):6){
    plot(PCA$x[,i], PCA$x[,j],
         xlab = paste0("PC",i," ",round(X_eigen_table[i,2],2),"%"),
         ylab = paste0("PC",j," ",round(X_eigen_table[j,2],2),"%"),
         pch = 20, col = 2, cex = 0.2)
    abline(v = 0, h = 0) } }

# assign the PCs and data set for PCA
for(i in 1:6){ assign(paste0("PC", i), as.numeric(PCA$x[,i]))}
Strength = concrete$Strength
# PCA data set
PCA.data = as.data.frame(cbind(PC1, PC2, PC3, PC4, PC5, PC6, Strength))
PCA.model = lm(Strength ~ ., data = PCA.data)
PCA.summary = summary(PCA.model)

round(PCA.summary$coefficients,4)[,1] # coefficients (beta's)

# Prints out SSE and RMSE
PCA.RMSE = 0
for(i in 1:4){
  train = PCA.data[index==i,]
  test = PCA.data[index!=i,]
  pcmodel = lm(Strength ~. , data = train)
  pcprediction = predict(pcmodel, test)
  pctrue = test$Strength
  pcSSE = sum((pctrue - pcprediction)^2)
  print(paste0("The SSE for PCA adjusted linear model on test data ", i," is ", pcSSE))
  PCA.RMSE = PCA.RMSE + pcSSE
}
PCA.RMSE = sqrt(PCA.RMSE/4)
print(paste0("The RMSE for AIC adjusted linear model is ", PCA.RMSE))
```

```r
# Final conclusion

# full model RMSE
fullRMSE = 0
for(i in 1:4){
  train = concrete[index==i,]
  test = concrete[index!=i,]
  fullmodel = lm(Strength ~. , data = train)
  fullprediction = predict(fullmodel, test)
  fulltrue = test$Strength
  fullSSE = sum((fulltrue - fullprediction)^2)
  fullRMSE = fullRMSE + fullSSE
}
fullRMSE = sqrt(fullRMSE/4)


FULL.model = lm(Strength ~. , data = concrete)
AIC.model = lm(Strength ~. , data = AIC.data)
PCA.model = lm(Strength ~. , data = PCA.data)

plot(AIC.model$fitted.values, scale(AIC.model$residuals), main = 'Residuals of AIC adjusted model', pch=
plot(PCA.model$fitted.values, scale(PCA.model$residuals), main = 'Residuals of PCA adjusted model', pch=

FULL.Adjr = round(summary(FULL.model)$adj.r.squared, 4)
AIC.Adjr = round(summary(AIC.model)$adj.r.squared, 4)
PCA.Adjr = round(summary(PCA.model)$adj.r.squared, 4)

FULL.MSE = sum(FULL.model$residuals^2)/(n-9)
AIC.MSE = sum(AIC.model$residuals^2)/(n-7)
PCA.MSE = sum(PCA.model$residuals^2)/(n-7)

Model = c("Full Model(8 Explanatories)", "AIC Model(6 Explanatories)", "PC Model(6 PCs)")
`Adj R Squared` = c(FULL.Adjr, AIC.Adjr, PCA.Adjr)
MSE = c(FULL.MSE, AIC.MSE, PCA.MSE)
RMSE = c(fullRMSE, AIC.RMSE, PCA.RMSE)

result =  data.frame(Model, `Adj R Squared`, MSE, RMSE)
knitr::kable(result)

summary(AIC.model)
```