Final Project

Sean Dudley

2024-05-01

Part 1)

Regression)

The Data set chosen for the regression section is on Concrete Compressive Strength as recorded along with the variables of Cement (kg in a cubic meter of mixture), Blast furnace slag (kg in a cubic meter of mixture), Fly Ash (kg in a cubic meter of mixture), Water (kg in a cubic meter of mixture), Superplasticizer (kg in a cubic meter of mixture), Coarse aggregate (kg in a cubic meter of mixture), Fine aggregate (kg in a cubic meter of mixture), Age (Days 1-365), with Concrete Compressive Strength as the final column (megapascal as MPa). In this data set there are 1030 observations, no missing data, and 8 variables that can be used as predictors with the one continuous dependent variable.

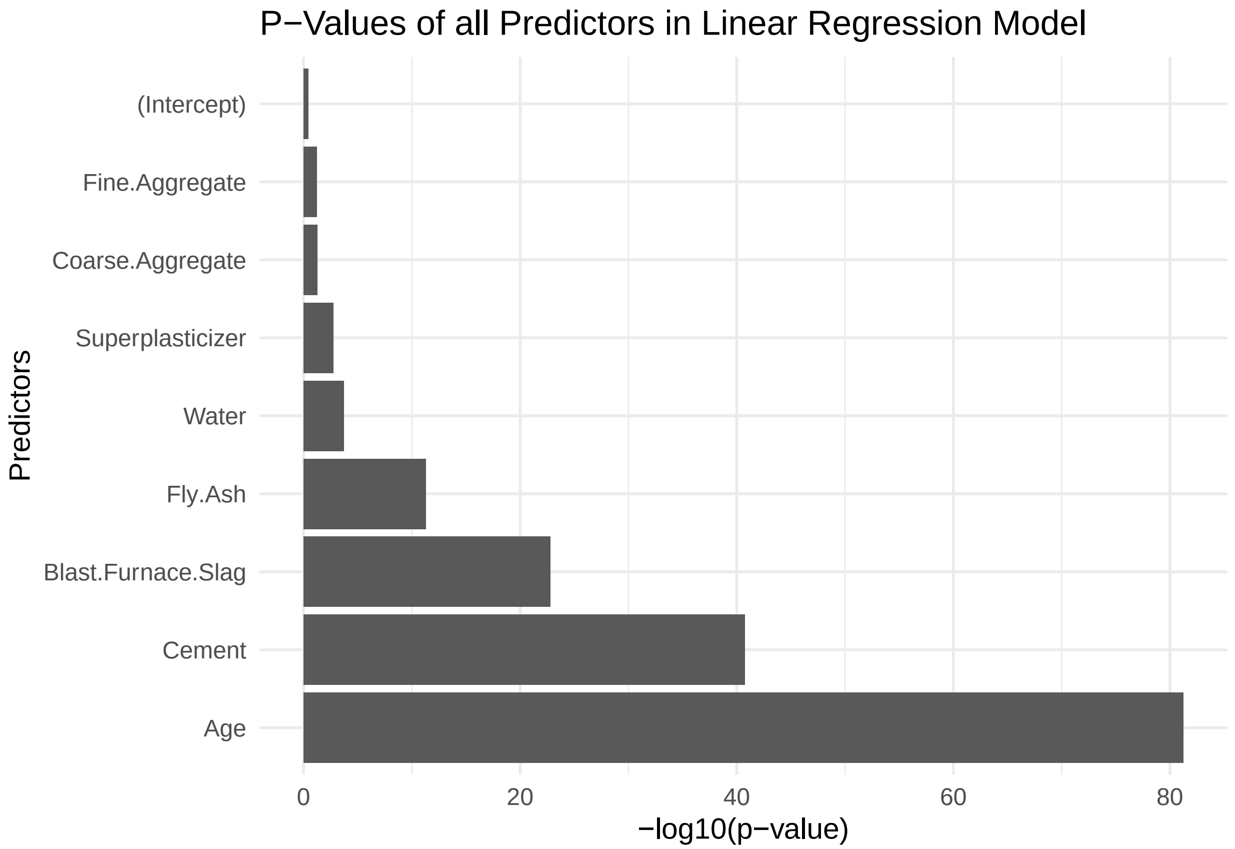
Linear regression is the method as the regression model to apply to the data set due to its' continuous variables. first by using all the variables to fit the linear model as it will give a p-value to better understand each of the variables

```
concrete <- read.csv("Concrete_Data2.csv")
concretelm <- lm(Concrete.compressive.strength ~ Cement + Blast.Furnace.Slag + Fly.Ash + Water + Superpl
summary(concretelm)
```

```
##
## Call:
## lm(formula = Concrete.compressive.strength ~ Cement + Blast.Furnace.Slag +
##     Fly.Ash + Water + Superplasticizer + Coarse.Aggregate + Fine.Aggregate +
##     Age, data = concrete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.654  -6.302   0.703   6.569  34.450
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -23.331214  26.585504  -0.878 0.380372
## Cement               0.119804   0.008489  14.113  < 2e-16 ***
## Blast.Furnace.Slag   0.103866   0.010136  10.247  < 2e-16 ***
## Fly.Ash              0.087934   0.012583   6.988 5.02e-12 ***
## Water               -0.149918   0.040177  -3.731 0.000201 ***
## Superplasticizer     0.292225   0.093424   3.128 0.001810 **
## Coarse.Aggregate     0.018086   0.009392   1.926 0.054425 .
## Fine.Aggregate       0.020190   0.010702   1.887 0.059491 .
## Age                  0.114222   0.005427  21.046  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 10.4 on 1021 degrees of freedom
## Multiple R-squared:  0.6155, Adjusted R-squared:  0.6125
## F-statistic: 204.3 on 8 and 1021 DF,  p-value: < 2.2e-16
```

To investigate the p-values as a method of selection of predictors will be done graphically a bar graph with -log10(p-value) as the y-axis will give a better understanding of the significance of each of the predictor variables.

```r
library(ggplot2)
concretelm <- lm(Concrete.compressive.strength ~ Cement + Blast.Furnace.Slag + Fly.Ash + Water + Superpl
summarymodel <- summary(concretelm)
coefficients <- as.data.frame(summarymodel$coefficients)
coefficients$Predictor <- rownames(coefficients)
colnames(coefficients) <- c("Estimate", "Std_Error", "t_value", "p_value", "Predictor")
ggplot(coefficients, aes(x = reorder(Predictor, p_value), y = -log10(p_value))) + geom_col() +labs(titl
```



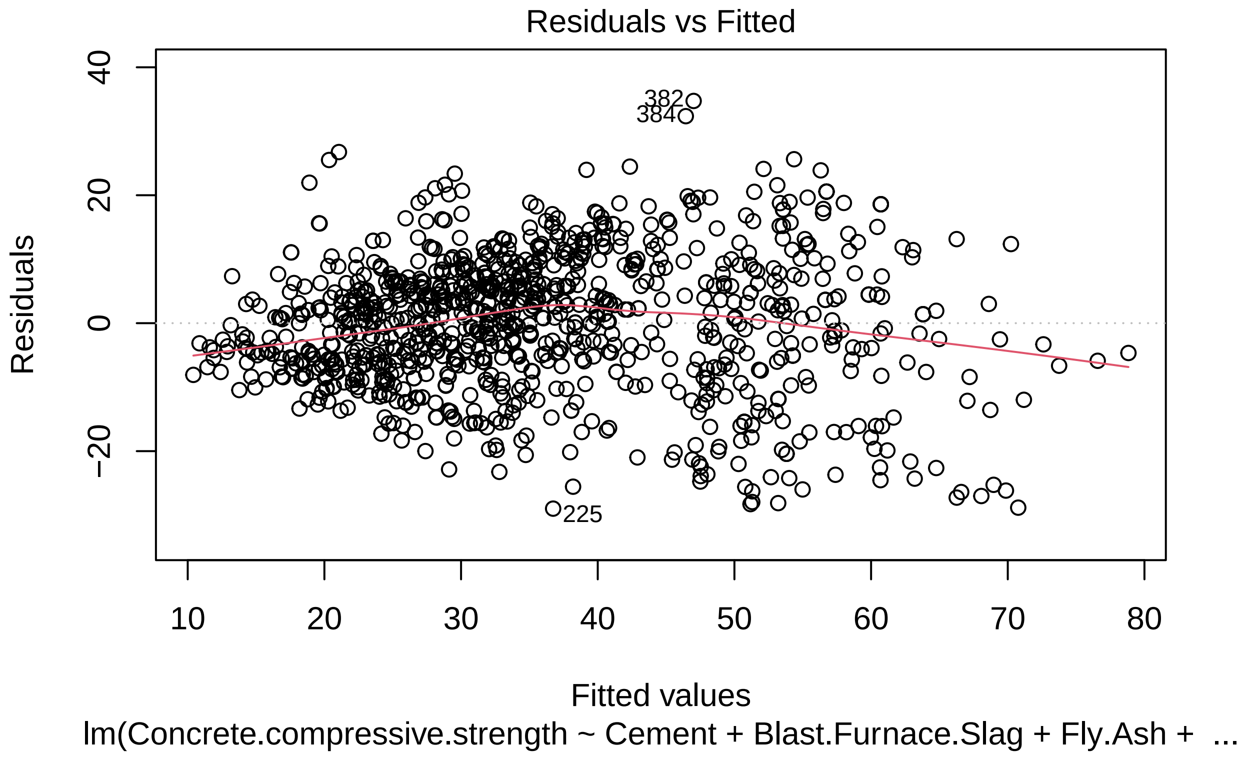
With this graph in conjunction with the summary of the data with the p-values the conclusion can be drawn that Fine and Coarse aggregate are not needed. Due to the P-values of the t-test (shown in graph and summary) for both the Coarse and Fine aggregate we can remove them from the predictor variables as their p-values are above 0.05 which indicates they are less significant. Thus our resulting model is

```
concretelm <- lm(Concrete.compressive.strength ~ Cement + Blast.Furnace.Slag + Fly.Ash + Water + Superpl
summary(concretelm)
```

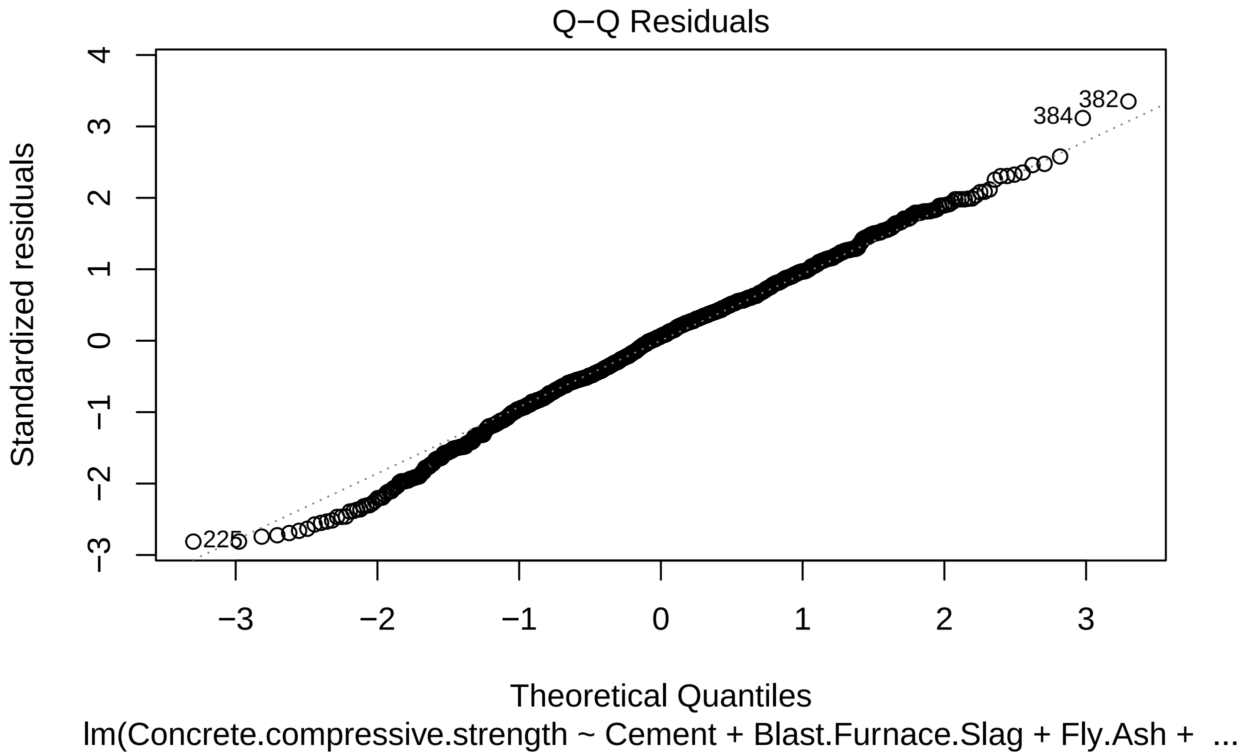
```
##
## Call:
## lm(formula = Concrete.compressive.strength ~ Cement + Blast.Furnace.Slag +
##     Fly.Ash + Water + Superplasticizer + Age, data = concrete)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.987  -6.469   0.653   6.547  34.732
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        28.992982   4.213202   6.881 1.03e-11 ***
## Cement              0.105413   0.004246  24.825  < 2e-16 ***
## Blast.Furnace.Slag  0.086472   0.004974  17.385  < 2e-16 ***
## Fly.Ash             0.068660   0.007735   8.877  < 2e-16 ***
## Water              -0.218088   0.021129 -10.322  < 2e-16 ***
## Superplasticizer    0.240311   0.084567   2.842  0.00458 **
## Age                 0.113492   0.005407  20.988  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.41 on 1023 degrees of freedom
## Multiple R-squared:  0.614,  Adjusted R-squared:  0.6118
## F-statistic: 271.2 on 6 and 1023 DF,  p-value: < 2.2e-16
```

This is the best version of the linear regression model. Gives us an incredibly low p-value which means the model is very significant. It's fit shows a very obvious relationship between the variables and concrete compressive strength. The model explains 61.4%(R-squared) of the variance in the concrete compressive strength. Total F statistic and P-value (<2.2e-16) show the model is very significant. The estimates show the impact on compressive strength each predictor has in terms of a kg added in a cubic meter of concrete. The linear regression is reinforced as the best model graphically through a Q-Q plot and a Residual vs Fitted plot

```
plot(concretelm, which = 1)
```



```r
plot(concretelm, which = 2)
```



As seen in the graph of the Q-Q plot forming a generally striaght line and the Residuals vs Fitted plot not really forming any sort of general shape besides being gathered mostly near 0 we have certainty that it is the best model. This leads to the conclusion that it is statistically significant.

Part 2)

Classification)

The classification data for this section is based on Maternal Health Risks. This data was collected at various medical facilities in Bangladesh. There are 6 independent variables collected from mothers and 1 dependent variable of associated risk. The variables are Age(Years), Systolic Blood Pressure(mmHg normal upper values of BP), Diastolic BP(mmHg lower value BP), Blood glucose molar concentration as BS(mmol/L), Body temp (Fahrenheit), Heart rate(bpm), and finally Risk level (high mid low as assessed). This is obviously a lot of variables and must be sized down since the method applied in this case is kNN(k-nearest neighbor)

To determine the most important variables Principal Component analysis will be applied as we have many variables we do not want to over fit the kNN model. k-nearest neighbor is the operation due to the 3 different possible classifications and multiple variables due to predict it. PCA also gives a good overall depiction of the data to start and better understand.

```r
library(caret)
```

```
## Loading required package: lattice
```

```
library(factoextra)
```

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

```
library(GGally)
```

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

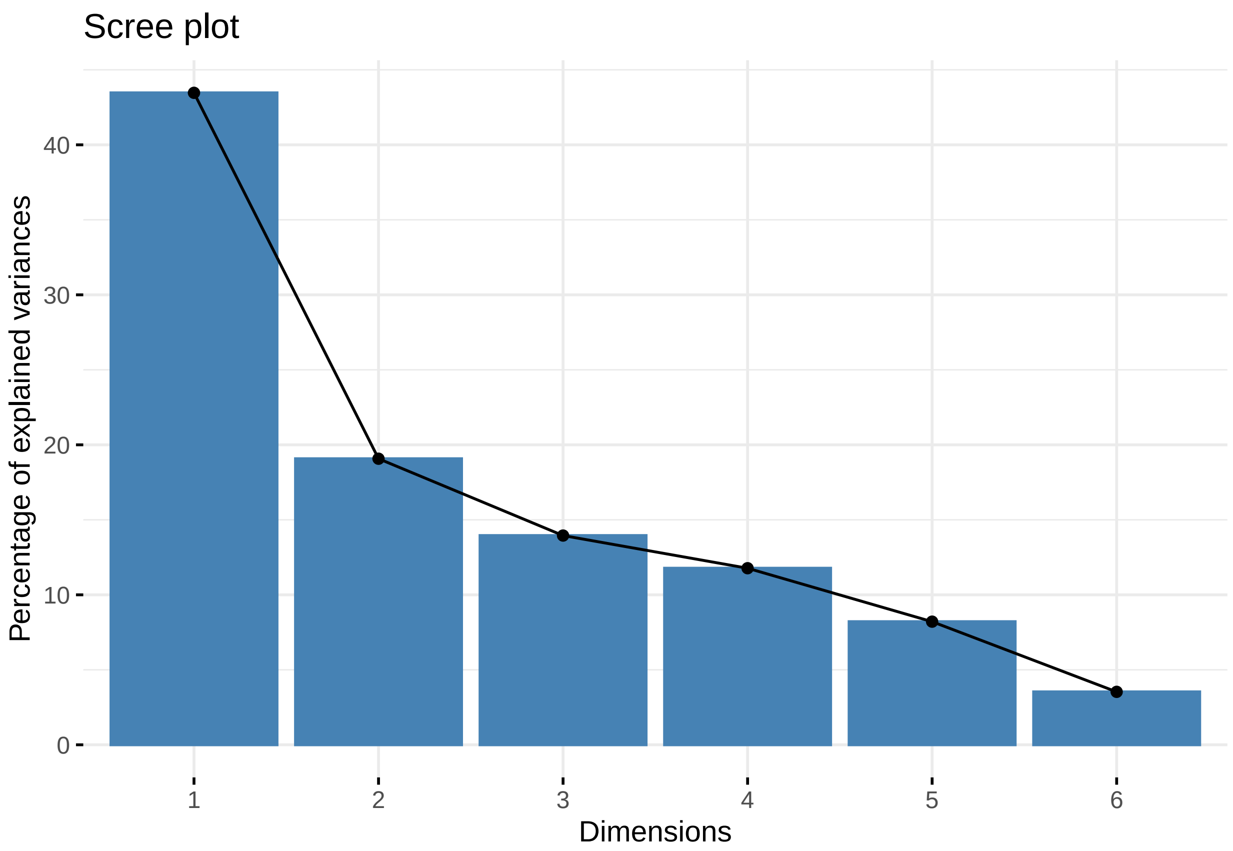
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggplot2)
library(class)
data <- read.csv("Maternal Health Risk Data Set.csv")
data$RiskLevel <- as.factor(data$RiskLevel)
features <-data[-7]
target <-  data$RiskLevel
pca <- prcomp(features, scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5    PC6
## Standard deviation     1.6149 1.0698 0.9149 0.8404 0.70181 0.4602
## Proportion of Variance 0.4346 0.1907 0.1395 0.1177 0.08209 0.0353
## Cumulative Proportion  0.4346 0.6254 0.7649 0.8826 0.96470 1.0000
```

Through this we can see that a large majority of the variance is held in the first variable labeled PC1(Age) and the majority of the variance 88.26% held in the first 4 leading to believe they would be the best variables to work with in this case, at the least the first 3 must be used. All this is shown very well through the scree plot

```
fviz_eig(pca)
```



Now that there is a better understanding of which variables/predictors are the best descriptors of the data. The 4 variables that are optimal to use are Age, Systolic BP, Diastolic BP and BS. Next to perform k-Nearest Neighbor classifictaion there is to find the optimal k value this is done using Cross Validation as advised in the project outline.

```
features <-data[1:4]
control <- trainControl(method = "cv", number = 10)
grid <- expand.grid(k = seq(1, 20, by = 1))
kmodel <- train(x = features, y = target, method = "knn", tuneGrid = grid, trControl = control)
kmodel$bestTune
```

```
##   k
## 1 1
```

```
kmodel$results
```

```
##     k  Accuracy     Kappa AccuracySD    KappaSD
## 1   1 0.8421418 0.7610769 0.03229440 0.04837225
## 2   2 0.7395076 0.6051214 0.04644706 0.06973449
## 3   3 0.6901652 0.5291083 0.05430746 0.08349656
## 4   4 0.7030749 0.5487266 0.05343621 0.08213354
## 5   5 0.7020268 0.5453143 0.04807736 0.07503094
## 6   6 0.6922615 0.5298358 0.04436093 0.06940688
## 7   7 0.6705267 0.4965085 0.04770874 0.07366771
## 8   8 0.6586259 0.4791172 0.05847365 0.08805422
## 9   9 0.6654211 0.4893728 0.05746231 0.08644688
## 10 10 0.6674690 0.4923810 0.05096339 0.07622186
## 11 11 0.6832335 0.5157508 0.05000321 0.07423348
## 12 12 0.6615092 0.4820608 0.04747867 0.07134668
## 13 13 0.6486956 0.4623952 0.05726762 0.08626026
## 14 14 0.6596447 0.4798507 0.06032987 0.09019485
## 15 15 0.6606160 0.4813423 0.04934018 0.07381609
## 16 16 0.6497931 0.4634533 0.04767530 0.07064419
## 17 17 0.6596263 0.4782211 0.04427298 0.06616582
## 18 18 0.6408431 0.4501114 0.05041483 0.07533535
## 19 19 0.6447352 0.4560080 0.05419389 0.08201191
## 20 20 0.6437449 0.4548584 0.05626934 0.08404447
```

Derived from this is that the best hyper parameter k value for the set of the 4 features is 1 though this is very model specific. This is determined in the Cross Validation and as seen in the results a best value for k would be 5 as it provides the best for fitting . Now is the application of the kNN function as used in the notes.

```
k <- 5
knnmodel<- knn3(RiskLevel ~ Age + SystolicBP + DiastolicBP + BS, k = k, data = data)
knnmodel
```

```
## 5-nearest neighbor model
## Training set outcome distribution:
##
## high  low  mid
##  272  406  336
```

This gives the kNN model as printed out with the outcome distribution of 272 of high risk, 336 of mid risk, and 406 of low risk. kNN categorizes this through the nearest neighbor of 5. The way to confirm this is applying predict to a known outcome and the model to see if it will give the correct outcome in this case a data frame will be created with known outcome of high level risk.

```
newbobs <- data.frame(Age = 30, SystolicBP = 140, DiastolicBP = 85, BS = 7.00)
pred <- predict(knnmodel, newdata = newbobs)
pred
```

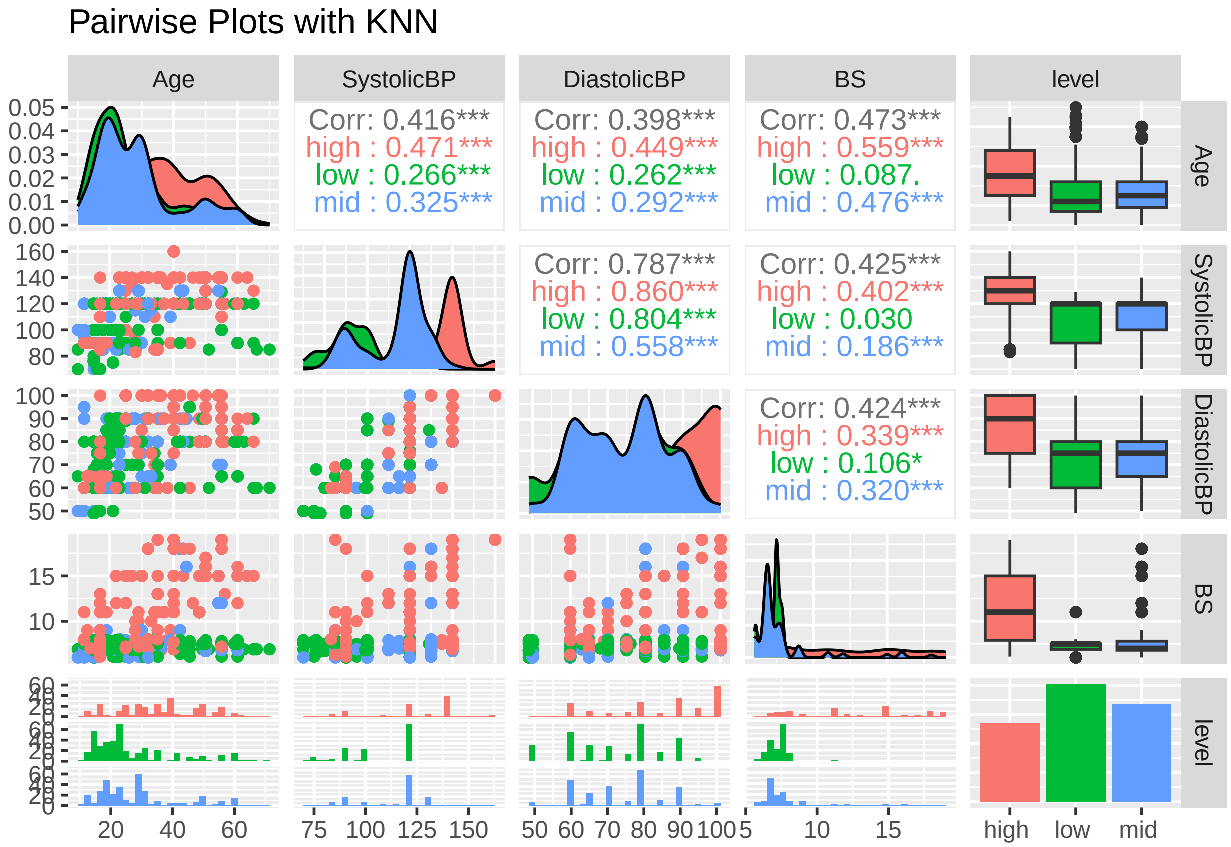
```
##      high low mid
## [1,]    1   0   0
```

As seen this give a 1 for high meaning it can only be given as a high level risk of maternal health issues. Which as before the applied method of kNN gives the correct outcome of the data as it matches the real value given of high level risk.

An assortment of plots to better get an understanding of data is to generate some pairwise plots that allow us to see the data.

```
data2 <- data.frame(features, level = target)
ggpairs(data2, aes(color = level), title = "Pairwise Plots with KNN")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Although some of the graphs do not work we get a better idea of how each variable and result is shown throughout the data and kNN model as it is processed.