Sadikshya_Concrete_Strength

by Sadikshya Duwadi

Submission date: 31-Mar-2024 05:57PM (UTC+0545)
Submission ID: 2335818794
File name: Sadikshya_Duwadi_Sec-E.docx (4.17M)
Word count: 1843
Character count: 11508

School of Computing, Creative Technology and Engineering

Student ID	77356725
Student Name	Sadikshya Duwadi
Module Name & CRN	Applied Machine Learning-18910
Level	5
Assessment Name & Part No.	Case Study-3: Concrete Strength, PART-I
Project Title	Data Analysis on Concrete Strength
Date of Submission	31st March, 2024
Course	BSc. (Hons) Computing
Academic Year	2024

Table of Contents

PART I

Data Analysis on Concrete Strength
Sadikshya Duwadi, Bsc (hons) Computing, 2024

Introduction:

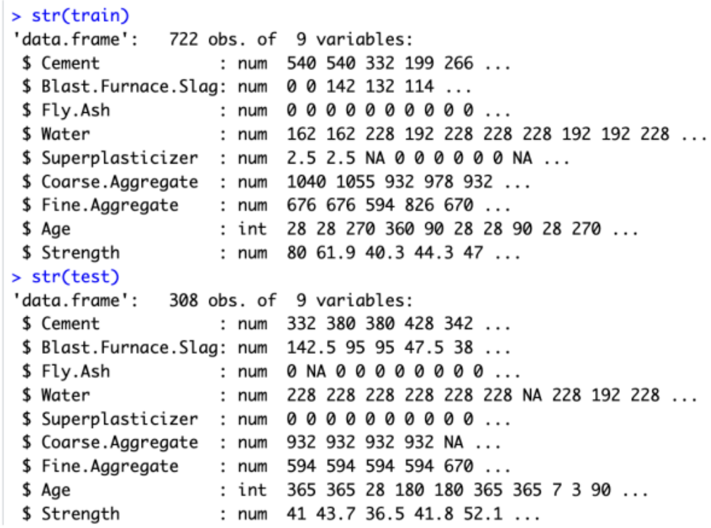
Concrete, one of the world's most produced material, is structural material made of a hard, chemically inert particle component known as aggregate (often sand and gravel) that is formed by the interaction of cement and water which cures overtime (Concrete | Definition, Composition, Uses, Types, & Facts | Britannica, 2024). When making a building, houses or any structure, it is very important to know the compressive strength of the concrete to make it more durable and last long. The strength of the concrete mainly depends on the amount and quality of the material (Cement, fly ash, Water, Coarse aggregate, etc.) mixed to make it. The use of data analysis or machine learning for knowing the compressive strength of the concrete have all contributed to the progress of concrete mix designs.

Literature review:

The study examined several areas of measuring and optimising concrete strength. It investigated how various local sources of fine aggregates impact concrete characteristics, highlighting the critical importance of aggregates in this respect (Rahman, 2020). Furthermore, the research emphasised the need of attaining the appropriate bond strength between layers of high-strength and lightweight concrete, and they proposed several strategies for improving interlayer bonding (Eisa, Aboul-Nour and Mohamad, 2024). A statistical study of concrete compressive strength measurement methods was also conducted, with the goal of determining the optimal sample sizes and proposing improved evaluation methodologies (Sujeet Kumar Mahato and Kumar, 2024). Predictive methods, such as the IABC-MLP algorithm, were praised for their accuracy in forecasting concrete strength by combining heuristic algorithms and neural networks (Li et al., 2024). Furthermore, the use of machine learning approaches to optimise concrete mix designs was noteworthy, with artificial neural networks and data mining being used to effectively estimate compressive strength (Ziolkowski and Maciej Niedostatkiewicz, 2019). The historical evolution of cement and concrete, as well as their reactivity to environmental conditions, demonstrated the continual need for study and improvement in concrete engineering (Gagg, 2014). Furthermore, research into aggregate grading and natural sand composition revealed their major influence on concrete strength, emphasising the need of knowing them for appropriate mix design (S. Hasdemir, A. Tuğrul and M. Yılmaz, 2016). Finally, the study of the influence of specimen size on compressive strength evaluation demonstrated the need of include specimen features in concrete testing methods (Banarjee, Alam and Ahmad, n.d.).

- **Exploratory Data Analysis**

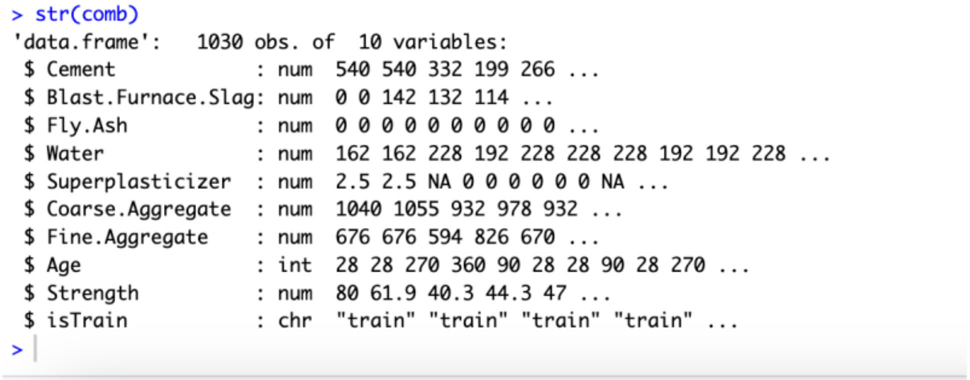
  EDA employs statistical and visualisation approaches to uncover hidden patterns and trends in the data which can help us to understand the data easily.
  The two datasets, train and test are combined to find the compressive strength of the concrete. The combined dataset has1030 number of rows and 10 number of columns. The ultimate variable, 'Strength', indicates the concrete's compressive strength, which we are trying to predict in this case study.



```
> str(train)
'data.frame':	722 obs. of  9 variables:
 $ Cement            : num  540 540 332 199 266 ...
 $ Blast.Furnace.Slag: num  0 0 142 132 114 ...
 $ Fly.Ash           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Water             : num  162 162 228 192 228 228 228 192 192 228 ...
 $ Superplasticizer  : num  2.5 2.5 NA 0 0 0 0 0 0 NA ...
 $ Coarse.Aggregate  : num  1040 1055 932 978 932 ...
 $ Fine.Aggregate    : num  676 676 594 826 670 ...
 $ Age               : int  28 28 270 360 90 28 28 90 28 270 ...
 $ Strength          : num  80 61.9 40.3 44.3 47 ...
> str(test)
'data.frame':	308 obs. of  9 variables:
 $ Cement            : num  332 380 380 428 342 ...
 $ Blast.Furnace.Slag: num  142.5 95 95 47.5 38 ...
 $ Fly.Ash           : num  0 NA 0 0 0 0 0 0 0 0 ...
 $ Water             : num  228 228 228 228 228 228 NA 228 192 228 ...
 $ Superplasticizer  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Coarse.Aggregate  : num  932 932 932 932 NA ...
 $ Fine.Aggregate    : num  594 594 594 594 670 ...
 $ Age               : int  365 365 28 180 180 365 365 7 3 90 ...
 $ Strength          : num  41 43.7 36.5 41.8 52.1 ...
```

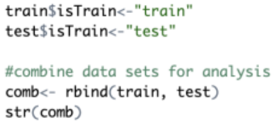
Figure 1: Structure of dataset Train and Test

The above figure 1 provides the structure of two data sets i.e. >str(train) for dataset Train and >str(test) for dataset Test. They both consist same structure with same row and column.



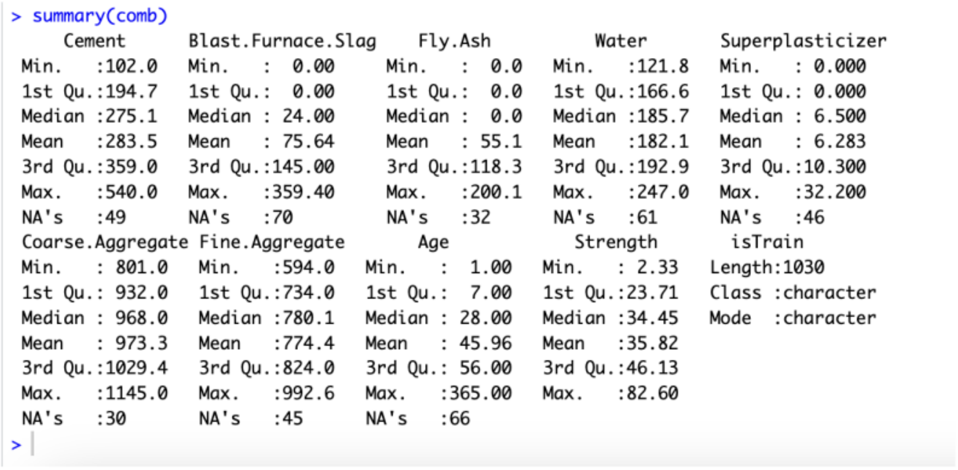
```
> str(comb)
'data.frame':	1030 obs. of  10 variables:
 $ Cement            : num  540 540 332 199 266 ...
 $ Blast.Furnace.Slag: num  0 0 142 132 114 ...
 $ Fly.Ash           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Water             : num  162 162 228 192 228 228 228 192 192 228 ...
 $ Superplasticizer  : num  2.5 2.5 NA 0 0 0 0 0 0 NA ...
 $ Coarse.Aggregate  : num  1040 1055 932 978 932 ...
 $ Fine.Aggregate    : num  676 676 594 826 670 ...
 $ Age               : int  28 28 270 360 90 28 28 90 28 270 ...
 $ Strength          : num  80 61.9 40.3 44.3 47 ...
 $ isTrain           : chr  "train" "train" "train" "train" ...
> 
```

Figure 2: Structure of combined dataset



```
train$isTrain<-"train"
test$isTrain<-"test"

#combine data sets for analysis
comb<- rbind(train, test)
str(comb)
```

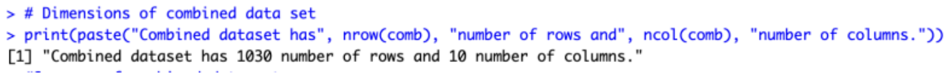
In fig 2, I have combined the above two datasets into one named as comb. Here, >str(comb) shows the structure of the combined datasets. I used rbind( ) function to combine them. I added an extra column 'isTrain' to distinguish them.



```
> summary(comb)
     Cement      Blast.Furnace.Slag    Fly.Ash          Water       Superplasticizer
 Min.   :102.0   Min.   :  0.00     Min.   :  0.0   Min.   :121.8   Min.   : 0.000  
 1st Qu.:194.7   1st Qu.:  0.00     1st Qu.:  0.0   1st Qu.:166.6   1st Qu.: 0.000  
 Median :275.1   Median : 24.00     Median :  0.0   Median :185.7   Median : 6.500  
 Mean   :283.5   Mean   : 75.64     Mean   : 55.1   Mean   :182.1   Mean   : 6.283  
 3rd Qu.:359.0   3rd Qu.:145.00     3rd Qu.:118.3   3rd Qu.:192.9   3rd Qu.:10.300  
 Max.   :540.0   Max.   :359.40     Max.   :200.1   Max.   :247.0   Max.   :32.200  
 NA's   :49      NA's   :70         NA's   :32      NA's   :61      NA's   :46      
 Coarse.Aggregate Fine.Aggregate      Age            Strength        isTrain         
 Min.   : 801.0   Min.   :594.0   Min.   :  1.00   Min.   : 2.33   Length:1030       
 1st Qu.: 932.0   1st Qu.:734.0   1st Qu.:  7.00   1st Qu.:23.71   Class :character  
 Median : 968.0   Median :780.1   Median : 28.00   Median :34.45   Mode  :character  
 Mean   : 973.3   Mean   :774.4   Mean   : 45.96   Mean   :35.82                     
 3rd Qu.:1029.4   3rd Qu.:824.0   3rd Qu.: 56.00   3rd Qu.:46.13                     
 Max.   :1145.0   Max.   :992.6   Max.   :365.00   Max.   :82.60                     
 NA's   :30       NA's   :45      NA's   :66                                         
> 
```

Figure 3: Summary of two combined datasets

Fig 3, provides the summary of two combined datasets. The summary consists of Minimum value, 1st Quartile, Median, Mean, 3rd Quartile, Maximum value and Missing value for each column.



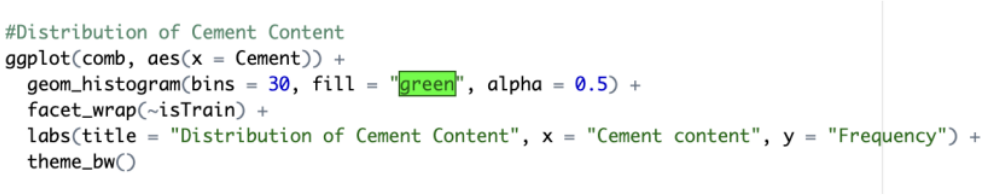
```
> # Dimensions of combined data set
> print(paste("Combined dataset has", nrow(comb), "number of rows and", ncol(comb), "number of columns."))
[1] "Combined dataset has 1030 number of rows and 10 number of columns."
```

Figure 4: Dimension of combined datasets

Above fig 4, prints the dimension of combined datasets. Combined dataset has 1030 number of rows and 10 number of columns.

- **Visualization**

1. Distribution of Cement Content



```
#Distribution of Cement Content
ggplot(comb, aes(x = Cement)) +
  geom_histogram(bins = 30, fill = "green", alpha = 0.5) +
  facet_wrap(~isTrain) +
  labs(title = "Distribution of Cement Content", x = "Cement content", y = "Frequency") +
  theme_bw()
```

Figure 1.1.1 Code snippet of Distribution of Cement Content

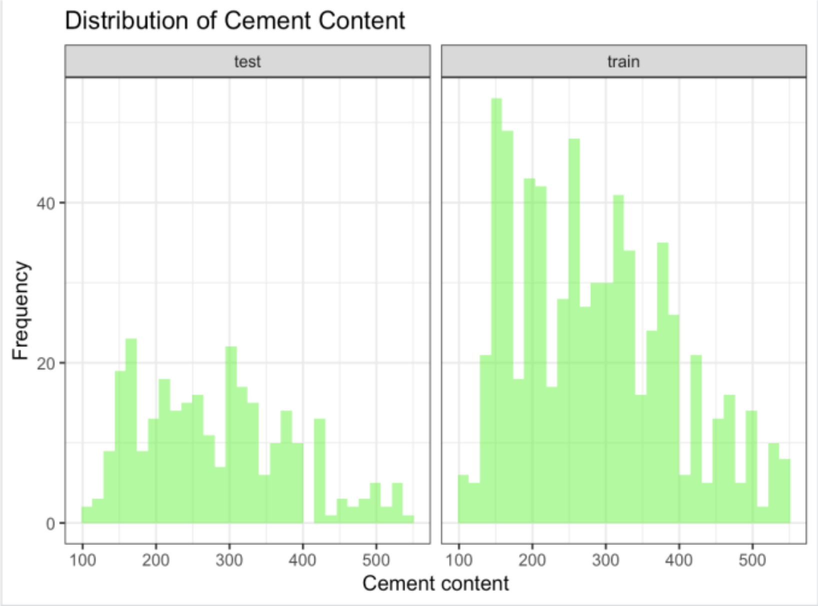
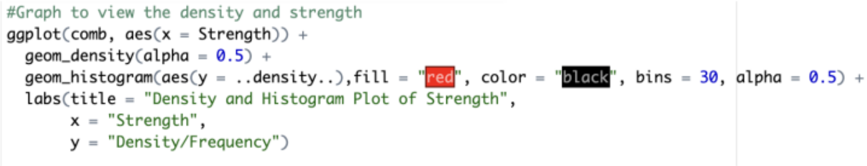


Figure 1.1.2 Histogram of distribution of cement content and their frequency

Figure 1.1.1 represents the ggplot code to give histogram of two combined datasets, where x-axis shows the cement content and x-axis shows the frequency in fig 1.1.2.

2. Histogram of Density and Strength



```
#Graph to view the density and strength
ggplot(comb, aes(x = Strength)) +
  geom_density(alpha = 0.5) +
  geom_histogram(aes(y = ..density..),fill = "red", color = "black", bins = 30, alpha = 0.5) +
  labs(title = "Density and Histogram Plot of Strength",
       x = "Strength",
       y = "Density/Frequency")
```

Figure 1.2.1 Code snippet to view density and strength

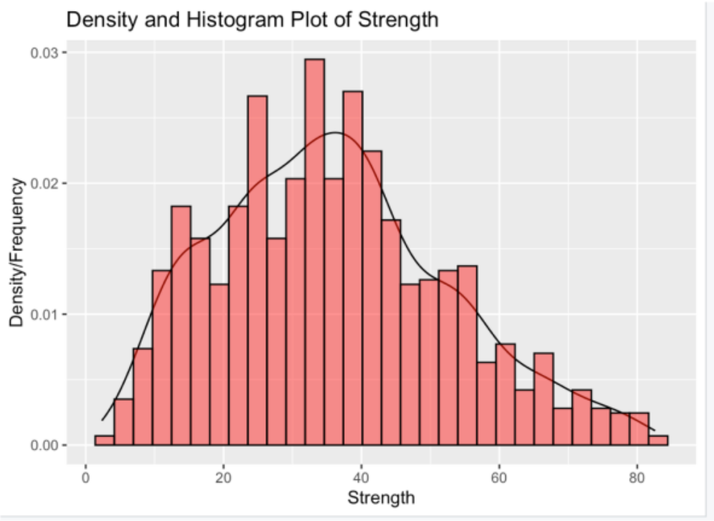
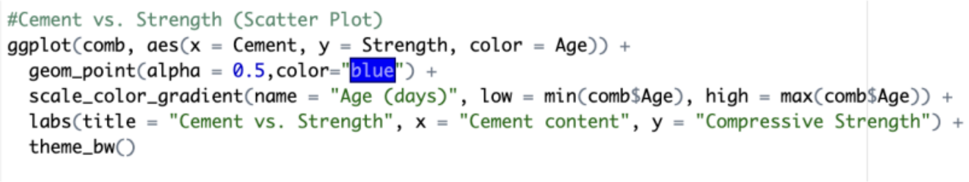


Figure 1.2.2 Histogram to view density and strength

Figure 1.2.2 shows the distribution of 'Strength' in the concrete dataset using both a density plot and a histogram. The smooth curve in the density plot represents a continuous picture of the data's probability distribution and histogram's bars provide a segmented representation of the data's frequency distribution.

3. Scatter plot for Cement vs. Strength



```
#Cement vs. Strength (Scatter Plot)
ggplot(comb, aes(x = Cement, y = Strength, color = Age)) +
  geom_point(alpha = 0.5,color="blue") +
  scale_color_gradient(name = "Age (days)", low = min(comb$Age), high = max(comb$Age)) +
  labs(title = "Cement vs. Strength", x = "Cement content", y = "Compressive Strength") +
  theme_bw()
```

Figure 1.3.1 Code snippet of Cement vs. Strength

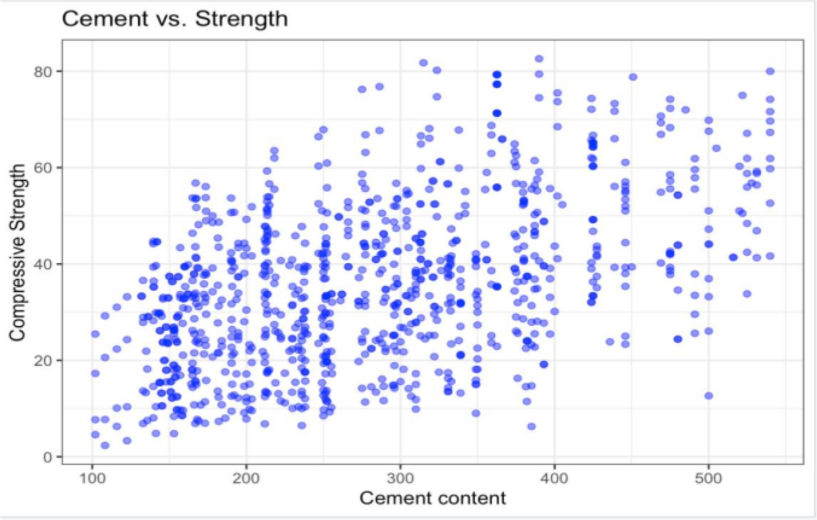
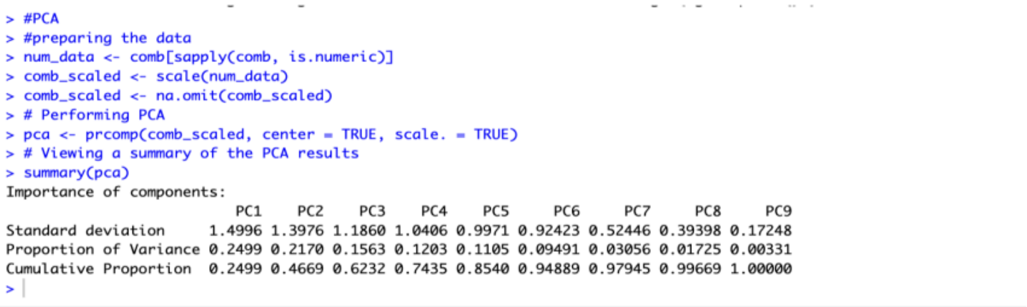


Figure 1.3.2 Scatterplot of Cement vs. Strength

Figure 1.3.1 is a scatterplot that examines the connection between 'Cement' and 'Strength' in the concrete dataset. The y-axis shows compressive strength and the x-axis shows cement concentration.

- **PCA**

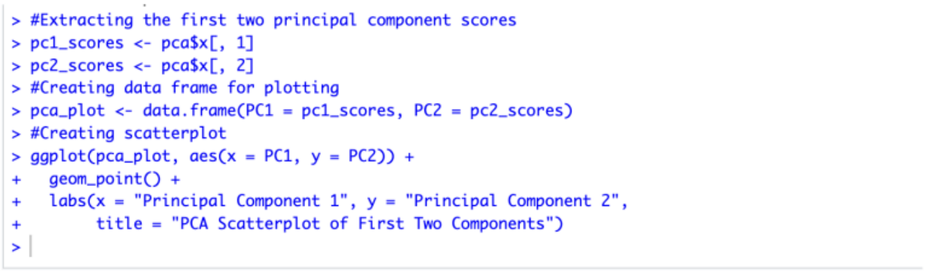


```
> #PCA
> #preparing the data
> num_data <- comb[sapply(comb, is.numeric)]
> comb_scaled <- scale(num_data)
> comb_scaled <- na.omit(comb_scaled)
> # Performing PCA
> pca <- prcomp(comb_scaled, center = TRUE, scale. = TRUE)
> # Viewing a summary of the PCA results
> summary(pca)
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7     PC8     PC9
Standard deviation     1.4996 1.3976 1.1860 1.0406 0.9971 0.92423 0.52446 0.39398 0.17248
Proportion of Variance 0.2499 0.2170 0.1563 0.1203 0.1105 0.09491 0.03056 0.01725 0.00331
Cumulative Proportion  0.2499 0.4669 0.6232 0.7435 0.8540 0.94889 0.97945 0.99669 1.00000
> 
```

Figure1.4 Performing PCA with output

In the above fig 1.4, PCA is performed on the scaled numerical basis of combined dataset to comprehend the data's underlying structure. The result provides: Standard Deviation (PC1 and PC2 have the highest standard deviations, implying that they describe the majority of the variance in the data.), Proportion of Variance (PC1 makes up 24.99% of the overall variance, while PC2 contributes an additional 21.70%.) and Cumulative Proportion (It shows as we include more components, the variance percentage increases.)

- **Extracting the first two PCA and plotting**



```
> #Extracting the first two principal component scores
> pc1_scores <- pca$x[, 1]
> pc2_scores <- pca$x[, 2]
> #Creating data frame for plotting
> pca_plot <- data.frame(PC1 = pc1_scores, PC2 = pc2_scores)
> #Creating scatterplot
> ggplot(pca_plot, aes(x = PC1, y = PC2)) +
+   geom_point() +
+   labs(x = "Principal Component 1", y = "Principal Component 2",
+        title = "PCA Scatterplot of First Two Components")
> |
```

Figure 1.5.1 Extracting the first two PCA and creating scatter plot

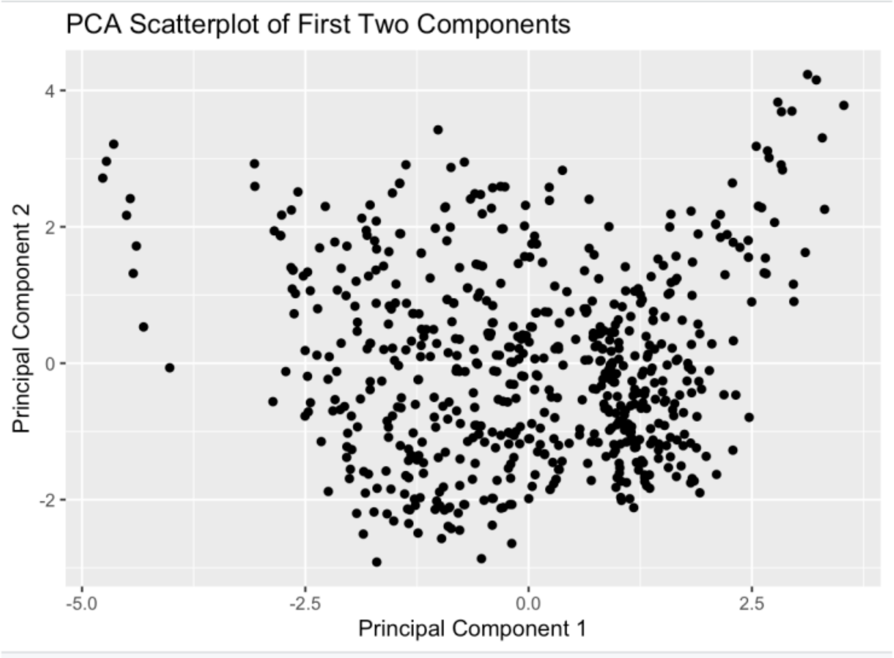
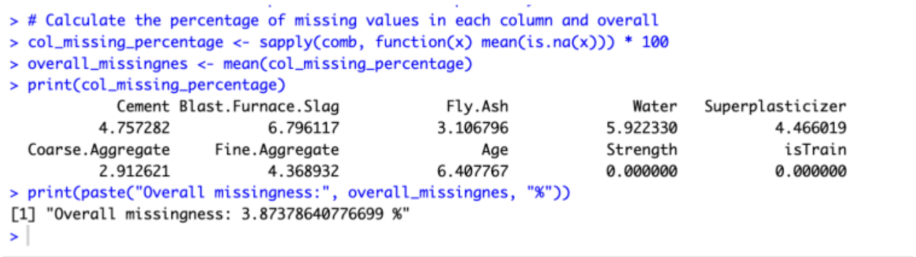


Figure 1.5.2 Scatter plot of first two PCA components

Figure 1.5.2 shows the scatter plot of first two PCA components i.e. PC1 and PC2 that we extracted in fig 1.5.1. Every data point is a physical sample represented into the dimension specified by these two main components.

- **Data Pre-processing**

1. Missing Values



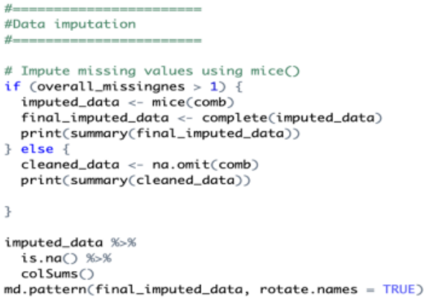
```
> # Calculate the percentage of missing values in each column and overall
> col_missing_percentage <- sapply(comb, function(x) mean(is.na(x))) * 100
> overall_missingnes <- mean(col_missing_percentage)
> print(col_missing_percentage)
          Cement Blast.Furnace.Slag            Fly.Ash              Water   Superplasticizer 
        4.757282           6.796117           3.106796           5.922330           4.466019 
 Coarse.Aggregate     Fine.Aggregate                Age           Strength            isTrain 
        2.912621           4.368932           6.407767           0.000000           0.000000 
> print(paste("Overall missingness:", overall_missingnes, "%"))
[1] "Overall missingness: 3.87378640776699 %"
> 
```

Figure 1.6.1 Calculation of missing values in each column and overall

Missing Data Percentage for each component: Cement'(4.76%), Blast Furnace Slag (6.8%), Fly Ash (3.11%), Water (5.92%), Superplasticizer (4.47%), Coarse Aggregate (2.91%), Fine Aggregate (4.37%), and Age'(6.41%).

Overall Missing Data Percentage: 3.873%.

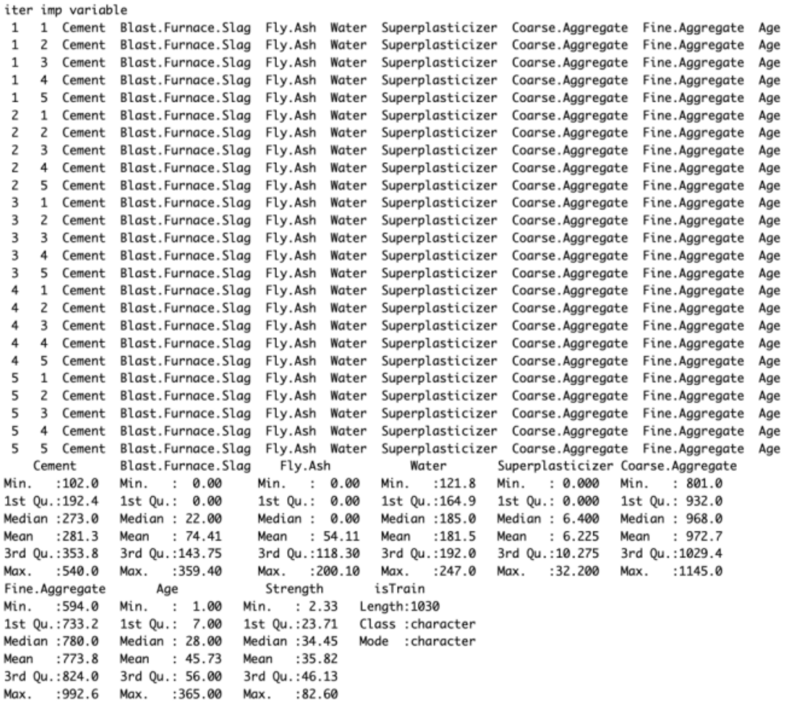
- **Imputation**



```
#========================
#Data imputation
#========================

# Impute missing values using mice()
if (overall_missingnes > 1) {
  imputed_data <- mice(comb)
  final_imputed_data <- complete(imputed_data)
  print(summary(final_imputed_data))
} else {
  cleaned_data <- na.omit(comb)
  print(summary(cleaned_data))

}

imputed_data %>%
  is.na() %>%
  colSums()
md.pattern(final_imputed_data, rotate.names = TRUE)
```

Figure 1.6.2 Code snippet of Data Imputation

In above fig 1.6.2, the mice function is used for cleaning data rather than omit function to generate numerous imputed datasets by filling in missing values based on known correlations between variables, if the average of missing values in column is greater than 1%, as the omit function may result in the loss of data. The md pattern for the final imputed data was also generated which can be found in fig 1.6.4.



```
iter imp variable
  1   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  1   2  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  1   3  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  1   4  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  1   5  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  2   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  2   2  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  2   3  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  2   4  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  2   5  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  3   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  3   2  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  3   3  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  3   4  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  3   5  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  4   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  4   2  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  4   3  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  4   4  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  4   5  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  5   1  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  5   2  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  5   3  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  5   4  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
  5   5  Cement  Blast.Furnace.Slag  Fly.Ash  Water  Superplasticizer  Coarse.Aggregate  Fine.Aggregate  Age
     Cement       Blast.Furnace.Slag    Fly.Ash           Water        Superplasticizer Coarse.Aggregate
 Min.   :102.0   Min.   :  0.00     Min.   :  0.00   Min.   :121.8   Min.   : 0.000   Min.   : 801.0  
 1st Qu.:192.4   1st Qu.:  0.00     1st Qu.:  0.00   1st Qu.:164.9   1st Qu.: 0.000   1st Qu.: 932.0  
 Median :273.0   Median : 22.00     Median :  0.00   Median :185.0   Median : 6.400   Median : 968.0  
 Mean   :281.3   Mean   : 74.41     Mean   : 54.11   Mean   :181.5   Mean   : 6.225   Mean   : 972.7  
 3rd Qu.:353.8   3rd Qu.:143.75     3rd Qu.:118.30   3rd Qu.:192.0   3rd Qu.:10.275   3rd Qu.:1029.4  
 Max.   :540.0   Max.   :359.40     Max.   :200.10   Max.   :247.0   Max.   :32.200   Max.   :1145.0  
 Fine.Aggregate        Age            Strength        isTrain         
 Min.   :594.0   Min.   :  1.00   Min.   : 2.33   Length:1030       
 1st Qu.:733.2   1st Qu.:  7.00   1st Qu.:23.71   Class :character  
 Median :780.0   Median : 28.00   Median :34.45   Mode  :character  
 Mean   :773.8   Mean   : 45.73   Mean   :35.82                     
 3rd Qu.:824.0   3rd Qu.: 56.00   3rd Qu.:46.13                     
 Max.   :992.6   Max.   :365.00   Max.   :82.60                     
```

Figure 1.6.3 Output of Data Imputation

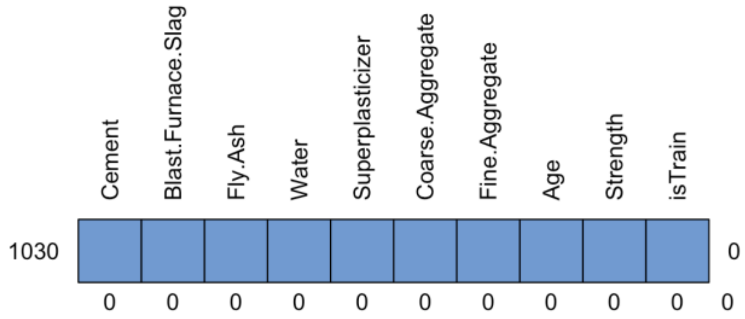
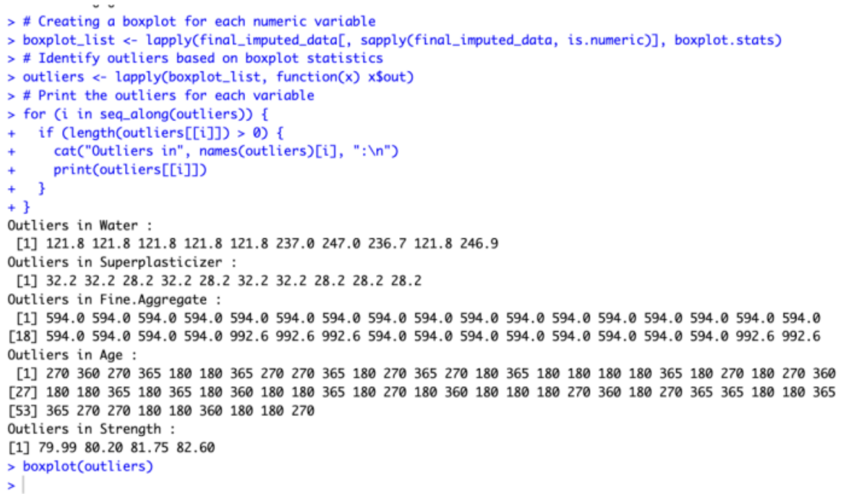


Figure 1.6.4 md pattern of Final Imputed Data

2. Outliers



```
> # Creating a boxplot for each numeric variable
> boxplot_list <- lapply(final_imputed_data[, sapply(final_imputed_data, is.numeric)], boxplot.stats)
> # Identify outliers based on boxplot statistics
> outliers <- lapply(boxplot_list, function(x) x$out)
> # Print the outliers for each variable
> for (i in seq_along(outliers)) {
+   if (length(outliers[[i]]) > 0) {
+     cat("Outliers in", names(outliers)[i], ":\n")
+     print(outliers[[i]])
+   }
+ }
Outliers in Water :
 [1] 121.8 121.8 121.8 121.8 121.8 237.0 247.0 236.7 121.8 246.9
Outliers in Superplasticizer :
 [1] 32.2 32.2 28.2 32.2 28.2 32.2 32.2 28.2 28.2 28.2
Outliers in Fine.Aggregate :
 [1] 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0
[18] 594.0 594.0 594.0 594.0 992.6 992.6 992.6 594.0 594.0 594.0 594.0 594.0 594.0 594.0 594.0 992.6 992.6
Outliers in Age :
 [1] 270 360 270 365 180 180 365 270 270 365 180 270 365 270 180 365 180 180 180 180 365 180 270 180 270 360
[27] 180 180 365 180 365 180 360 180 180 365 180 270 180 360 180 180 180 270 360 180 270 365 365 180 180 365
[53] 365 270 270 180 180 360 180 180 270
Outliers in Strength :
[1] 79.99 80.20 81.75 82.60
> boxplot(outliers)
> 
```

Figure 1.7.1 Output of Outlier detection

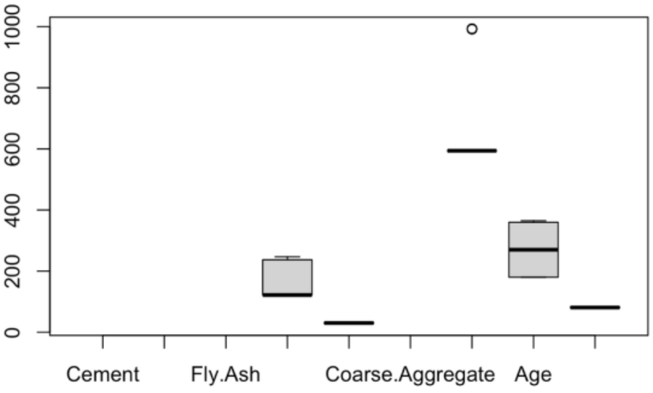
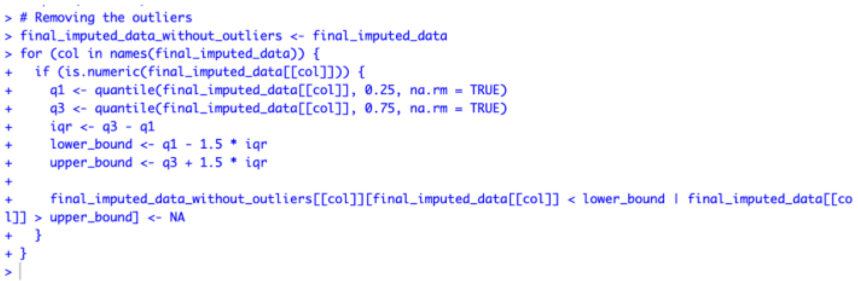


Figure 1.7.2 Boxplot of Outlier detection

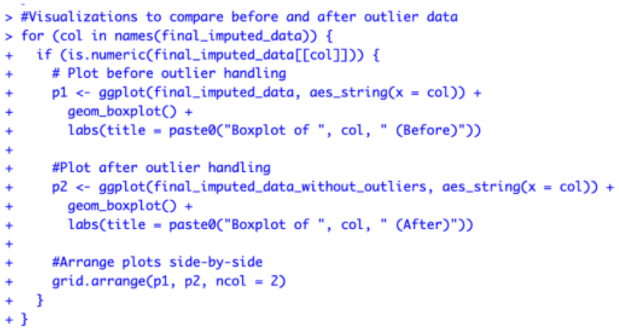
To detect outliers in fig 1.7.1, Boxplot statistic method is used. Outliers can be detected in several components as shown in fig 1.7.2.



```
> # Removing the outliers
> final_imputed_data_without_outliers <- final_imputed_data
> for (col in names(final_imputed_data)) {
+   if (is.numeric(final_imputed_data[[col]])) {
+     q1 <- quantile(final_imputed_data[[col]], 0.25, na.rm = TRUE)
+     q3 <- quantile(final_imputed_data[[col]], 0.75, na.rm = TRUE)
+     iqr <- q3 - q1
+     lower_bound <- q1 - 1.5 * iqr
+     upper_bound <- q3 + 1.5 * iqr
+ 
+     final_imputed_data_without_outliers[[col]][final_imputed_data[[col]] < lower_bound | final_imputed_data[[co
l]] > upper_bound] <- NA
+   }
+ }
> 
```

Figure 1.7.3 Removing outliers

Removing outliers can help clean up data and focus study on key patterns. But rather than removing the outliers completely, the above code flags the outliers, as they may represent valid events within datasets. It computes the interquartile range and establishes outlier bounds. It substitutes values that fall outside of certain boundaries with missing values (NA) to identify outliers and generates a new data frame with the original data and highlighted outliers.



```
> #Visualizations to compare before and after outlier data
> for (col in names(final_imputed_data)) {
+   if (is.numeric(final_imputed_data[[col]])) {
+     # Plot before outlier handling
+     p1 <- ggplot(final_imputed_data, aes_string(x = col)) +
+       geom_boxplot() +
+       labs(title = paste0("Boxplot of ", col, " (Before)"))
+ 
+     #Plot after outlier handling
+     p2 <- ggplot(final_imputed_data_without_outliers, aes_string(x = col)) +
+       geom_boxplot() +
+       labs(title = paste0("Boxplot of ", col, " (After)"))
+ 
+     #Arrange plots side-by-side
+     grid.arrange(p1, p2, ncol = 2)
+   }
+ }
```

Figure 1.7.4 Visualizations to compare before and after outlier handling

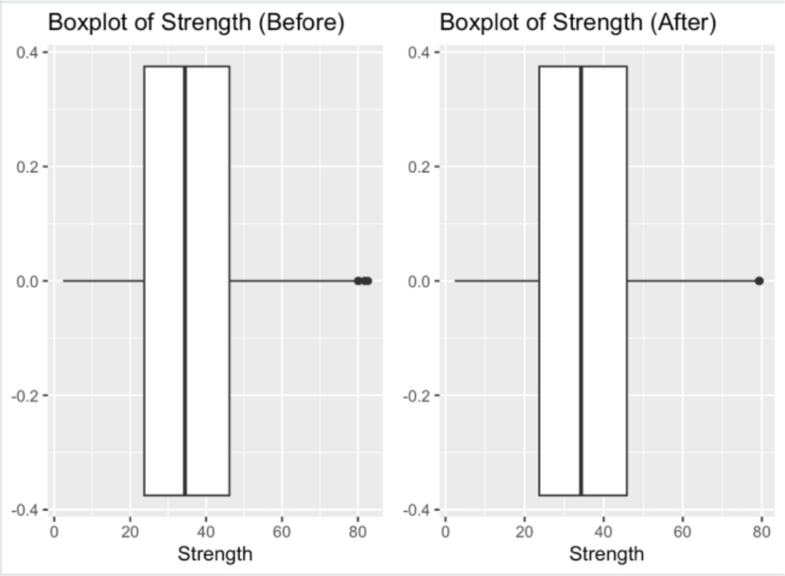
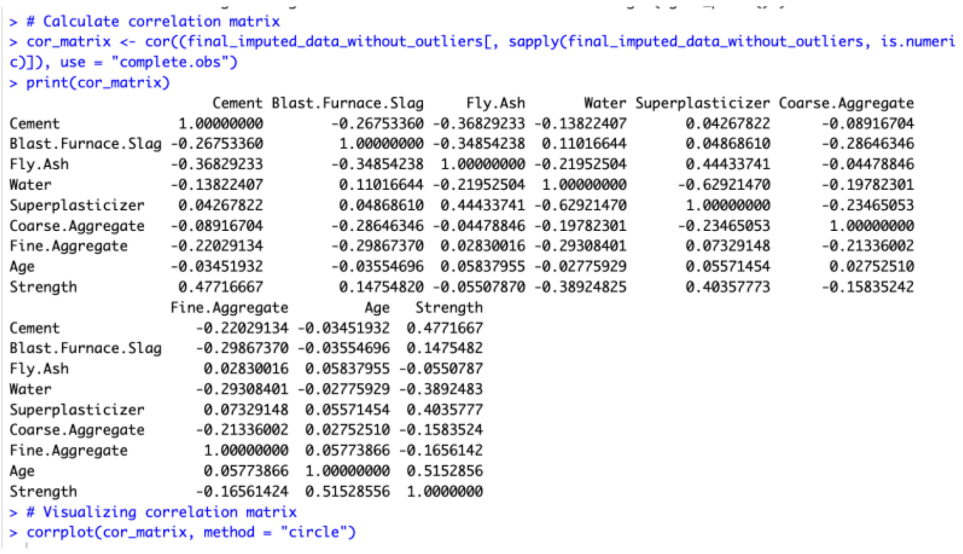


Figure 1.7.5 Box plot compare before and after outlier handling

Above fig 1.7.5, represents box plot to compare before and after outlier handling.

3. Multicollinearity



```
> # Calculate correlation matrix
> cor_matrix <- cor((final_imputed_data_without_outliers[, sapply(final_imputed_data_without_outliers, is.numeri
c)]), use = "complete.obs")
> print(cor_matrix)
                       Cement Blast.Furnace.Slag     Fly.Ash       Water Superplasticizer Coarse.Aggregate
Cement             1.00000000        -0.26753360 -0.36829233 -0.13822407       0.04267822      -0.08916704
Blast.Furnace.Slag -0.26753360        1.00000000 -0.34854238  0.11016644       0.04868610      -0.28646346
Fly.Ash            -0.36829233       -0.34854238  1.00000000 -0.21952504       0.44433741      -0.04478846
Water              -0.13822407        0.11016644 -0.21952504  1.00000000      -0.62921470      -0.19782301
Superplasticizer    0.04267822        0.04868610  0.44433741 -0.62921470       1.00000000      -0.23465053
Coarse.Aggregate   -0.08916704       -0.28646346 -0.04478846 -0.19782301      -0.23465053       1.00000000
Fine.Aggregate     -0.22029134       -0.29867370  0.02830016 -0.29308401       0.07329148      -0.21336002
Age                -0.03451932       -0.03554696  0.05837955 -0.02775929       0.05571454       0.02752510
Strength            0.47716667        0.14754820 -0.05507870 -0.38924825       0.40357773      -0.15835242
                   Fine.Aggregate         Age   Strength
Cement                -0.22029134 -0.03451932  0.4771667
Blast.Furnace.Slag    -0.29867370 -0.03554696  0.1475482
Fly.Ash                0.02830016  0.05837955 -0.0550787
Water                 -0.29308401 -0.02775929 -0.3892483
Superplasticizer       0.07329148  0.05571454  0.4035777
Coarse.Aggregate      -0.21336002  0.02752510 -0.1583524
Fine.Aggregate         1.00000000  0.05773866 -0.1656142
Age                    0.05773866  1.00000000  0.5152856
Strength              -0.16561424  0.51528556  1.0000000
> # Visualizing correlation matrix
> corrplot(cor_matrix, method = "circle")
```

Figure 1.8.1 Calculating Multicollinearity

The output in fig 1.8.1 represents a correlation matrix, a table displaying the correlation coefficients between pairs of variables in a dataset.

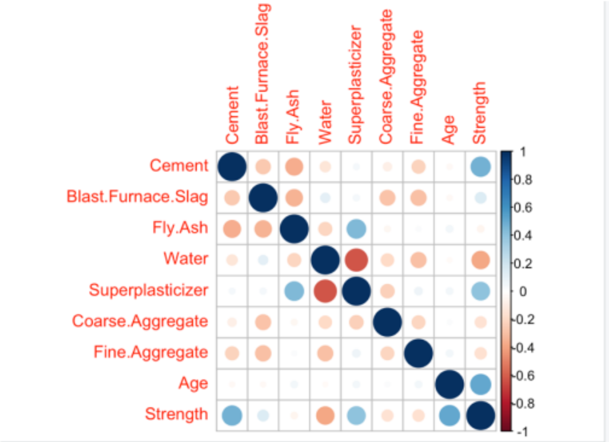
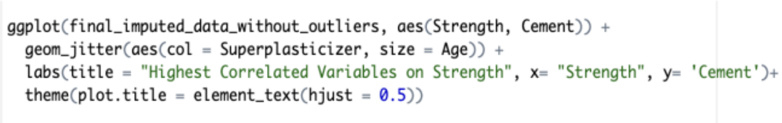


Figure 1.8.2 Visualizing correlation matrix

Figure 1.8.2 illustrates a correlation matrix with a circle packing visual representation. The blue circles represent strong positive correlation and the red circles represents strong negative correlation.

- **Scatterplot visualizing the relationship between Strength and Cement**



```
ggplot(final_imputed_data_without_outliers, aes(Strength, Cement)) +
  geom_jitter(aes(col = Superplasticizer, size = Age)) +
  labs(title = "Highest Correlated Variables on Strength", x= "Strength", y= 'Cement')+
  theme(plot.title = element_text(hjust = 0.5))
```

Figure 1.8.3 Code snippet of visualizing the relationship between Strength and Cement

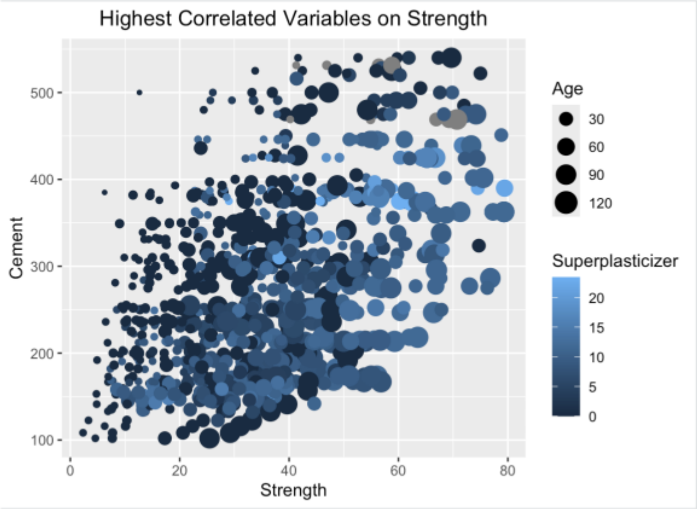
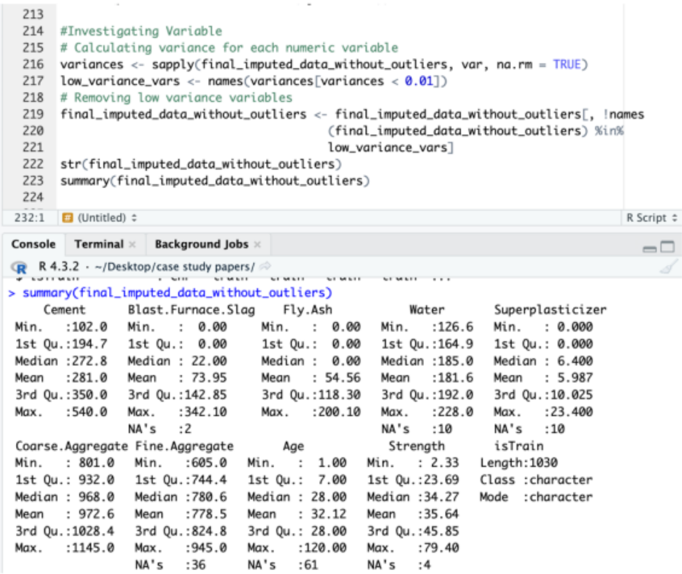


Figure 1.8.4 Scatterplot between Strength & Cement

Figure 1.8.4 shows the scatterplot between Strength & Cement, highlighted as highest correlated variables. The colour and size of the data points (black for age and blue for superplasticizer) may be used to visually investigate the effects of age and superplasticizer in this plot.

- **Investigating Low Variance Variable**



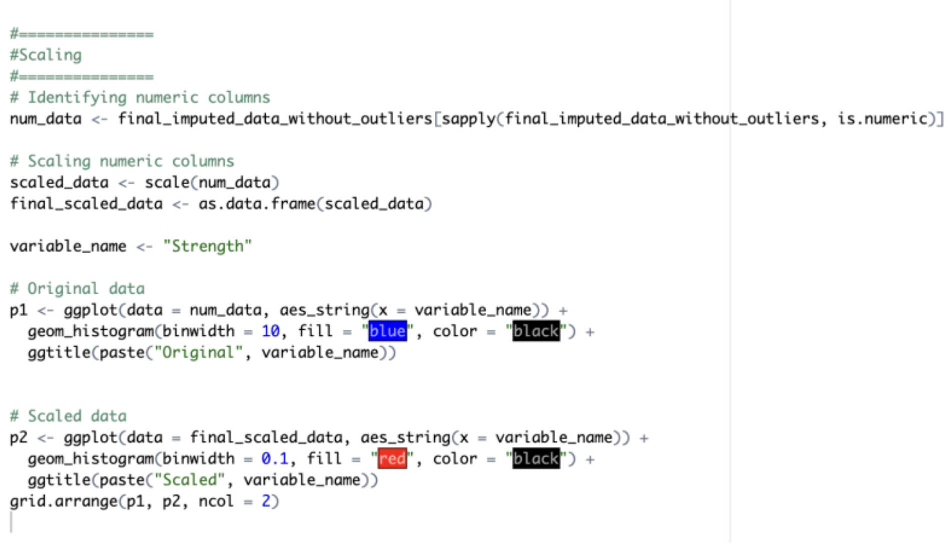
```
#Investigating Variable
# Calculating variance for each numeric variable
variances <- sapply(final_imputed_data_without_outliers, var, na.rm = TRUE)
low_variance_vars <- names(variances[variances < 0.01])
# Removing low variance variables
final_imputed_data_without_outliers <- final_imputed_data_without_outliers[, !names
                                     (final_imputed_data_without_outliers) %in%
                                     low_variance_vars]
str(final_imputed_data_without_outliers)
summary(final_imputed_data_without_outliers)
```

```
> summary(final_imputed_data_without_outliers)
    Cement       Blast.Furnace.Slag    Fly.Ash           Water        Superplasticizer
 Min.   :102.0   Min.   :  0.00     Min.   :  0.00   Min.   :126.6   Min.   : 0.000  
 1st Qu.:194.7   1st Qu.:  0.00     1st Qu.:  0.00   1st Qu.:164.9   1st Qu.: 0.000  
 Median :272.8   Median : 22.00     Median :  0.00   Median :185.0   Median : 6.400  
 Mean   :281.0   Mean   : 73.95     Mean   : 54.56   Mean   :181.6   Mean   : 5.987  
 3rd Qu.:350.0   3rd Qu.:142.85     3rd Qu.:118.30   3rd Qu.:192.0   3rd Qu.:10.025  
 Max.   :540.0   Max.   :342.10     Max.   :200.10   Max.   :228.0   Max.   :23.400  
                 NA's   :2                           NA's   :10      NA's   :10      
 Coarse.Aggregate Fine.Aggregate       Age           Strength        isTrain         
 Min.   : 801.0   Min.   :605.0   Min.   :  1.00   Min.   : 2.33   Length:1030       
 1st Qu.: 932.0   1st Qu.:744.4   1st Qu.:  7.00   1st Qu.:23.69   Class :character  
 Median : 968.0   Median :780.6   Median : 28.00   Median :34.27   Mode  :character  
 Mean   : 972.6   Mean   :778.5   Mean   : 32.12   Mean   :35.64                     
 3rd Qu.:1028.4   3rd Qu.:824.8   3rd Qu.: 28.00   3rd Qu.:45.85                     
 Max.   :1145.0   Max.   :945.0   Max.   :120.00   Max.   :79.40                     
                  NA's   :36      NA's   :61       NA's   :4                         
```

Figure 1.8.5 Removing variable having low variance

The code in fig 1.8.5 seeks to recognise and delete variables with low variance, which may not contribute much to modelling or analysis. Removing low variance variables can minimise noise, simplify models, and perhaps enhance performance.

4. Scaling



```
#===============
#Scaling
#===============
# Identifying numeric columns
num_data <- final_imputed_data_without_outliers[sapply(final_imputed_data_without_outliers, is.numeric)]

# Scaling numeric columns
scaled_data <- scale(num_data)
final_scaled_data <- as.data.frame(scaled_data)

variable_name <- "Strength"

# Original data
p1 <- ggplot(data = num_data, aes_string(x = variable_name)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black") +
  ggtitle(paste("Original", variable_name))


# Scaled data
p2 <- ggplot(data = final_scaled_data, aes_string(x = variable_name)) +
  geom_histogram(binwidth = 0.1, fill = "red", color = "black") +
  ggtitle(paste("Scaled", variable_name))
grid.arrange(p1, p2, ncol = 2)
```

Figure 1.9.1 Code Snippet of Impact of Scaling on the target variable 'Strength'

The above code in fig 1.9. compares the distribution of the "Strength" variable before and after scaling. Scaling may dramatically alter the shape and spread of a variable's distribution. Visualising this helps us grasp how scale impacts the data.

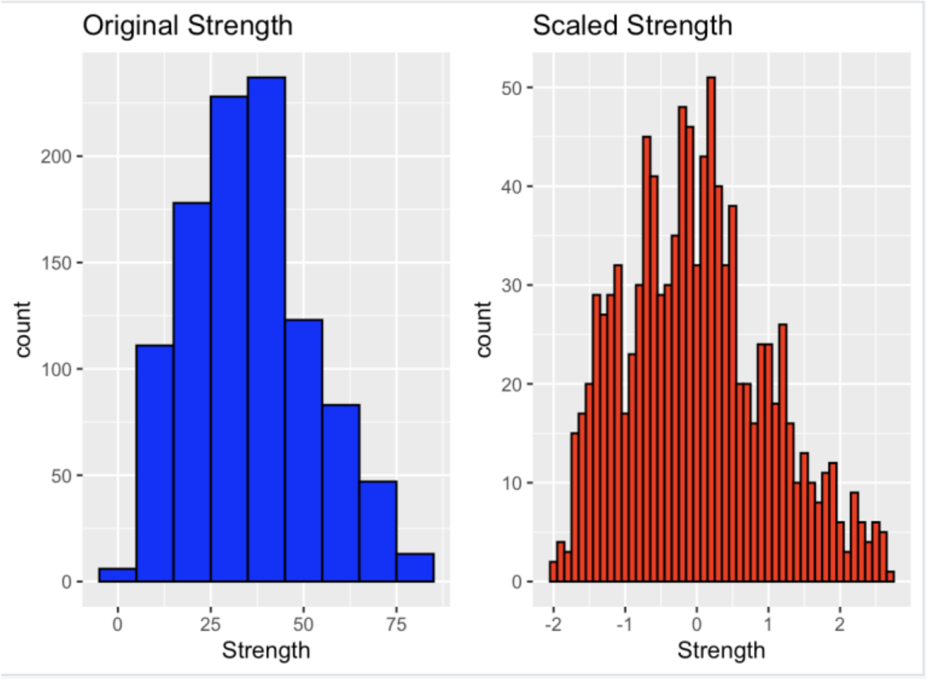


Figure 1.9.2 gg Plot of Original and Scaled value of Strength

Figure 1.9.2 shows the histogram *of* Original and Scaled value of Strength variable before and after scaling.

- **Bibliography**

1. Rahman, S. (2020). *Analysis on Compressive Strength of Concrete Using Different Sources of Fine Aggregates*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/338548502_Analysis_on_Compressive_Strength_of_Concrete_Using_Different_Sources_of_Fine_Aggregates [Accessed 26 Mar. 2024].

2. Eisa, A., Aboul-Nour, L.A. and Mohamad, A. (2024). *Experimental and theoretical investigation on the bond strength between high-strength and lightweight concrete*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/377993550_Experimental_and_theoretical_investigation_on_the_bond_strength_between_high-strength_and_lightweight_concrete [Accessed 26 Mar. 2024].

3. Sujeet Kumar Mahato and Kumar, A. (2024). *Statistical Analysis of Compressive Strength of Concrete*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/378149940_Statistical_Analysis_of_Compressive_Strength_of_Concrete [Accessed 26 Mar. 2024].

4. Li, P., Zhang, Y., Gu, J. and Duan, S. (2024). *Prediction of compressive strength of concrete based on IABC-MLP algorithm*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/377308680_Prediction_of_compressive_strength_of_concrete_based_on_IABC-MLP_algorithm [Accessed 26 Mar. 2024].

5. Ziolkowski, P. and Maciej Niedostatkiewicz (2019). Machine Learning Techniques in Concrete Mix Design. *Materials*, [online] 12(8), pp.1256–1256. doi:https://doi.org/10.3390/ma12081256.

6. Gagg, C.R. (2014). *Cement and Concrete as an engineering material: an historic appraisal and case study analysis*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/260439461_Cement_and_Concrete_as_an_engineering_material_an_historic_appraisal_and_case_study_analysis [Accessed 26 Mar. 2024].

7. Kumar, D.P. and Biable, A. (2020). *Experimental Investigation on the Compressive Strength of Concrete With Different Sizes of Coarse Aggregate*. [online] Social Science Research Network. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3713640.

8. Wang, S. and Baxter, L. (2007). Comprehensive study of biomass fly ash in concrete: Strength, microscopy, kinetics and durability. *Fuel Processing Technology*, [online] 88(11-12), pp.1165–1170. doi:https://doi.org/10.1016/j.fuproc.2007.06.016.

9. S. Hasdemir, A. Tuğrul and M. Yılmaz (2016). The effect of natural sand composition on concrete strength. *Construction and Building Materials*, [online] 112, pp.940–948. doi:https://doi.org/10.1016/j.conbuildmat.2016.02.188.

10. Banarjee, R., Alam, M. and Ahmad, Z. (n.d.). *Study of Compressive Strength of Various Grades of Concrete using Different Sizes of Cubes*. [online] Available at: https://www.ijert.org/research/study-of-compressive-strength-of-various-grades-of-concrete-using-different-sizes-of-cubes-IJERTV4IS070455.pdf.

11. Concrete | Definition, Composition, Uses, Types, & Facts | Britannica. (2024). In: *Encyclopædia Britannica*. [online] Available at: https://www.britannica.com/technology/concrete-building-material [Accessed 31 Mar. 2024].

Sadikshya_Concrete_Strength

ORIGINALITY REPORT

10%	6%	4%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Leeds Beckett University Student Paper	3%
2	Submitted to WWETB Student Paper	2%
3	Submitted to University of Witwatersrand Student Paper	1%
4	Submitted to University of York Student Paper	1%
5	umpir.ump.edu.my Internet Source	1%
6	assets.researchsquare.com Internet Source	1%
7	jurnal.poliupg.ac.id Internet Source	1%
8	Khadka, Chandra. "Penalized Regression Models for Concrete Strength Estimation", Southern Illinois University at Carbondale, 2021 Publication	<1%

9	dokumen.pub Internet Source	<1 %
10	www.scielo.br Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		