CONCRETE COMPRESSIVE STRENGTH

Load the required Packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(readxl)
library(datarium)
library(qqplotr)
```

```
##
## Attaching package: 'qqplotr'
##
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```

```
library(rcompanion)
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 4.4.2
```

```
library(car)
```

```
## Loading required package: carData

## Warning: package 'carData' was built under R version 4.4.2

##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(FSA)
```

```
## Registered S3 methods overwritten by 'FSA':
##   method       from
##   confint.boot car
##   hist.boot    car
## ## FSA v0.9.5. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
##
## Attaching package: 'FSA'
##
## The following object is masked from 'package:car':
##
##     bootCase
```

```
library(multcomp)
```

```
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
##
## Attaching package: 'TH.data'
##
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(RVAideMemoire)
```

```
## *** Package RVAideMemoire v 0.9-83-7 ***
##
## Attaching package: 'RVAideMemoire'
##
## The following object is masked from 'package:FSA':
##
##     se
```

Load the data

```
df <- read_excel('data/concrete compressive strength.xlsx')
df <- as.data.frame(df)
# check the head of the dataset
head(df)
```

```
##   Cement (component 1)(kg in a m^3 mixture)
## 1                                     540.0
## 2                                     540.0
## 3                                     332.5
## 4                                     332.5
## 5                                     198.6
## 6                                     266.0
##   Blast Furnace Slag (component 2)(kg in a m^3 mixture)
## 1                                                   0.0
## 2                                                   0.0
## 3                                                 142.5
## 4                                                 142.5
## 5                                                 132.4
## 6                                                 114.0
##   Fly Ash (component 3)(kg in a m^3 mixture)
## 1                                          0
## 2                                          0
## 3                                          0
## 4                                          0
## 5                                          0
## 6                                          0
##   Water  (component 4)(kg in a m^3 mixture)
## 1                                       162
## 2                                       162
## 3                                       228
## 4                                       228
## 5                                       192
## 6                                       228
##   Superplasticizer (component 5)(kg in a m^3 mixture)
## 1                                                 2.5
## 2                                                 2.5
## 3                                                 0.0
## 4                                                 0.0
## 5                                                 0.0
## 6                                                 0.0
##   Coarse Aggregate  (component 6)(kg in a m^3 mixture)
## 1                                               1040.0
## 2                                               1055.0
## 3                                                932.0
## 4                                                932.0
## 5                                                978.4
## 6                                                932.0
##   Fine Aggregate (component 7)(kg in a m^3 mixture) Age (day) Concrete Category
## 1                                             676.0        28            Coarse
## 2                                             676.0        28            Coarse
## 3                                             594.0       270            Coarse
## 4                                             594.0       365            Coarse
## 5                                             825.5       360              Fine
## 6                                             670.0        90            Coarse
##   Contains Fly Ash Concrete compressive strength(MPa, megapascals)
## 1            FALSE                                        79.98611
## 2            FALSE                                        61.88737
## 3            FALSE                                        40.26954
## 4            FALSE                                        41.05278
```

```
## 5            FALSE                                          44.29608
## 6            FALSE                                          47.02985
```

TASK 1 - EDA AND SUMMARY STATISTICS

```
# check for missing values
any(is.na(df))
```

```
## [1] FALSE
```

```
# check for duplicates
sum(duplicated(df))
```

```
## [1] 25
```

No missing values, but there are 25 duplicate rows, these duplicates would be removed

```
# remove duplicate rows
df <- df %>% distinct()
```

```
str(df)
```

```
## 'data.frame':    1005 obs. of  11 variables:
##  $ Cement (component 1)(kg in a m^3 mixture)            : num  540 540 332 332 199 ...
##  $ Blast Furnace Slag (component 2)(kg in a m^3 mixture): num  0 0 142 142 132 ...
##  $ Fly Ash (component 3)(kg in a m^3 mixture)           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Water  (component 4)(kg in a m^3 mixture)            : num  162 162 228 228 192 228 228 228 228 2
##  $ Superplasticizer (component 5)(kg in a m^3 mixture)  : num  2.5 2.5 0 0 0 0 0 0 0 0 ...
##  $ Coarse Aggregate  (component 6)(kg in a m^3 mixture) : num  1040 1055 932 932 978 ...
##  $ Fine Aggregate (component 7)(kg in a m^3 mixture)    : num  676 676 594 594 826 ...
##  $ Age (day)                                            : num  28 28 270 365 360 90 365 28 28 28 ...
##  $ Concrete Category                                    : chr  "Coarse" "Coarse" "Coarse" "Coarse" .
##  $ Contains Fly Ash                                     : logi  FALSE FALSE FALSE FALSE FALSE FALSE
##  $ Concrete compressive strength(MPa, megapascals)      : num  80 61.9 40.3 41.1 44.3 ...
```

To make things easier, we are going to rename to columns

```
names(df)
```

```
##  [1] "Cement (component 1)(kg in a m^3 mixture)"
##  [2] "Blast Furnace Slag (component 2)(kg in a m^3 mixture)"
##  [3] "Fly Ash (component 3)(kg in a m^3 mixture)"
##  [4] "Water  (component 4)(kg in a m^3 mixture)"
##  [5] "Superplasticizer (component 5)(kg in a m^3 mixture)"
##  [6] "Coarse Aggregate  (component 6)(kg in a m^3 mixture)"
##  [7] "Fine Aggregate (component 7)(kg in a m^3 mixture)"
##  [8] "Age (day)"
##  [9] "Concrete Category"
## [10] "Contains Fly Ash"
## [11] "Concrete compressive strength(MPa, megapascals)"
```

```r
# Rename the columns to easier names
df <- df %>%
  rename(
    cement = `Cement (component 1)(kg in a m^3 mixture)`,
    blast_furnace_slag = `Blast Furnace Slag (component 2)(kg in a m^3 mixture)`,
    fly_ash = `Fly Ash (component 3)(kg in a m^3 mixture)`,
    water = `Water  (component 4)(kg in a m^3 mixture)`,
    superplasticizer = `Superplasticizer (component 5)(kg in a m^3 mixture)`,
    coarse_aggregate = `Coarse Aggregate  (component 6)(kg in a m^3 mixture)`,
    fine_aggregate = `Fine Aggregate (component 7)(kg in a m^3 mixture)`,
    age = `Age (day)`,
    concrete_category = `Concrete Category`,
    contains_fly_ash = `Contains Fly Ash`,
    strength = `Concrete compressive strength(MPa, megapascals)`
  )

names(df)
```

```
##  [1] "cement"             "blast_furnace_slag" "fly_ash"
##  [4] "water"              "superplasticizer"   "coarse_aggregate"
##  [7] "fine_aggregate"     "age"                "concrete_category"
## [10] "contains_fly_ash"   "strength"
```

```r
#check the structure of the data
str(df)
```

```
## 'data.frame':    1005 obs. of  11 variables:
##  $ cement            : num  540 540 332 332 199 ...
##  $ blast_furnace_slag: num  0 0 142 142 132 ...
##  $ fly_ash           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ water             : num  162 162 228 228 192 228 228 228 228 228 ...
##  $ superplasticizer  : num  2.5 2.5 0 0 0 0 0 0 0 0 ...
##  $ coarse_aggregate  : num  1040 1055 932 932 978 ...
##  $ fine_aggregate    : num  676 676 594 594 826 ...
##  $ age               : num  28 28 270 365 360 90 365 28 28 28 ...
##  $ concrete_category : chr  "Coarse" "Coarse" "Coarse" "Coarse" ...
##  $ contains_fly_ash  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ strength          : num  80 61.9 40.3 41.1 44.3 ...
```

We can see that we have 11 columns, 9 are numerical, 2 categorical

Summary Statistics

```r
#check summary of the data to get a quick view of the means, medians, ranges, etc.
summary(df)
```

```
##      cement      blast_furnace_slag    fly_ash           water
##  Min.   :102.0   Min.   :  0.00     Min.   :  0.00   Min.   :121.8
##  1st Qu.:190.7   1st Qu.:  0.00     1st Qu.:  0.00   1st Qu.:166.6
##  Median :265.0   Median : 20.00     Median :  0.00   Median :185.7
##  Mean   :278.6   Mean   : 72.04     Mean   : 55.54   Mean   :182.1
##  3rd Qu.:349.0   3rd Qu.:142.50     3rd Qu.:118.27   3rd Qu.:192.9
```

```
##  Max.   :540.0   Max.   :359.40     Max.   :200.10   Max.   :247.0
##  superplasticizer coarse_aggregate fine_aggregate       age
##  Min.   : 0.000   Min.   : 801.0   Min.   :594.0   Min.   :  1.00
##  1st Qu.: 0.000   1st Qu.: 932.0   1st Qu.:724.3   1st Qu.:  7.00
##  Median : 6.100   Median : 968.0   Median :780.0   Median : 28.00
##  Mean   : 6.032   Mean   : 974.4   Mean   :772.7   Mean   : 45.86
##  3rd Qu.:10.000   3rd Qu.:1031.0   3rd Qu.:822.2   3rd Qu.: 56.00
##  Max.   :32.200   Max.   :1145.0   Max.   :992.6   Max.   :365.00
##  concrete_category  contains_fly_ash    strength
##  Length:1005        Mode :logical    Min.   : 2.332
##  Class :character   FALSE:541        1st Qu.:23.524
##  Mode  :character   TRUE :464        Median :33.798
##                                      Mean   :35.250
##                                      3rd Qu.:44.868
##                                      Max.   :82.599
```

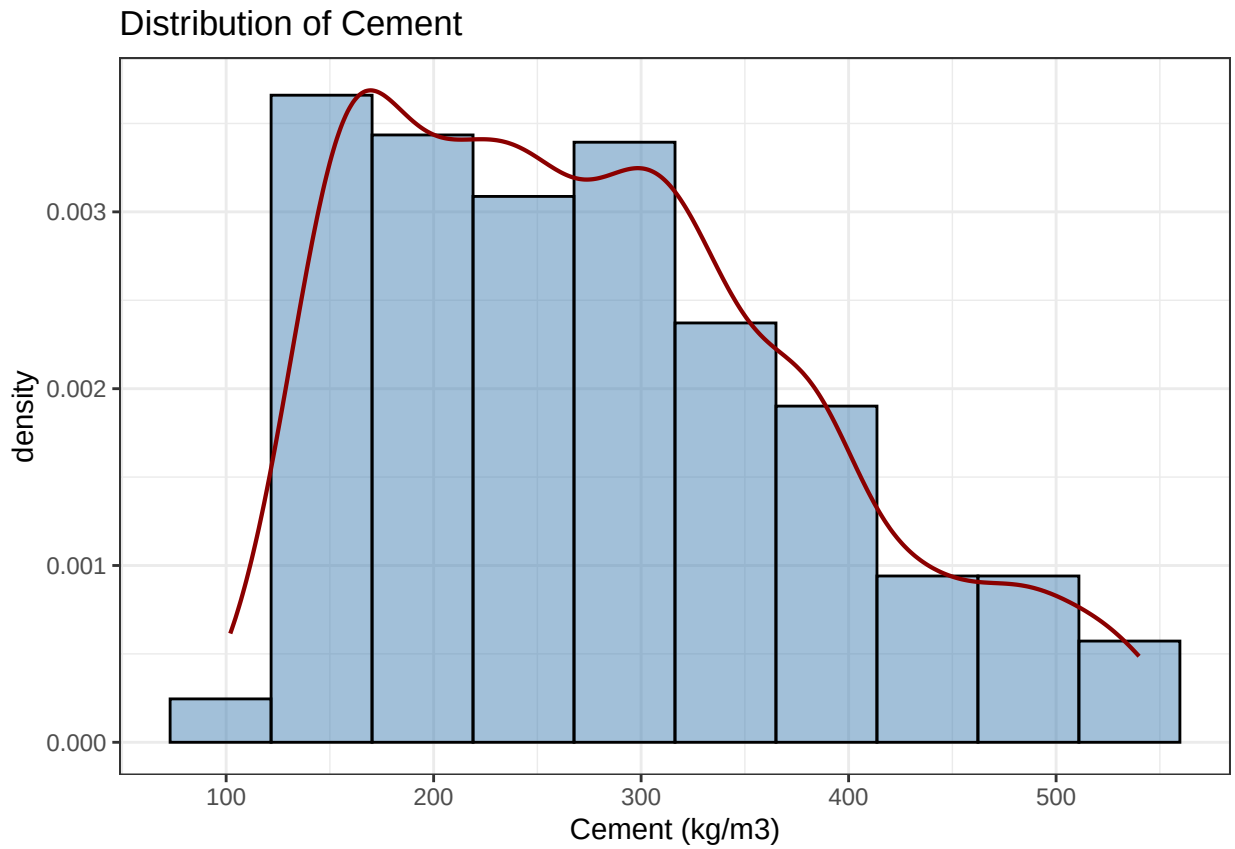
With just quick glance, we can see that some of the columns have a widely different means and medians, e.g Blast Furnace Slag, Fly ash, etc. This means that these columns are skewed! more EDA to verify.

Distribution of the columns and Tests for Normality

We set alpha = 0.05

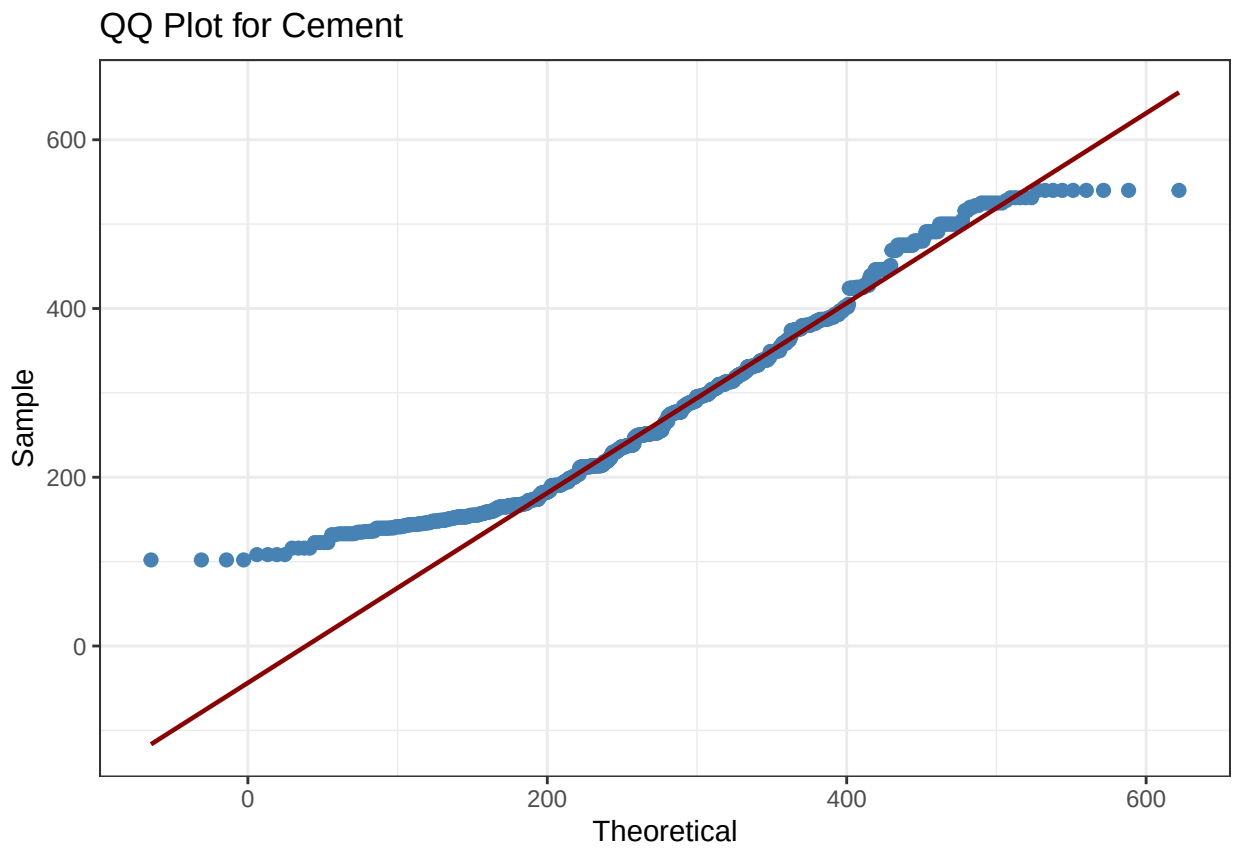
A. Histogram & Density plot for Numerical columns, and Tests for Normality

```
# Plot a histogram, and add a density plot to show the distribution
df %>%
  ggplot(aes(x=cement))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Cement", x = "Cement (kg/m3)")+
  theme_bw()
```



Cement: Slightly right-skewed distribution

```
# QQ plot
df %>%
  ggplot(aes(sample=cement))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Cement", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



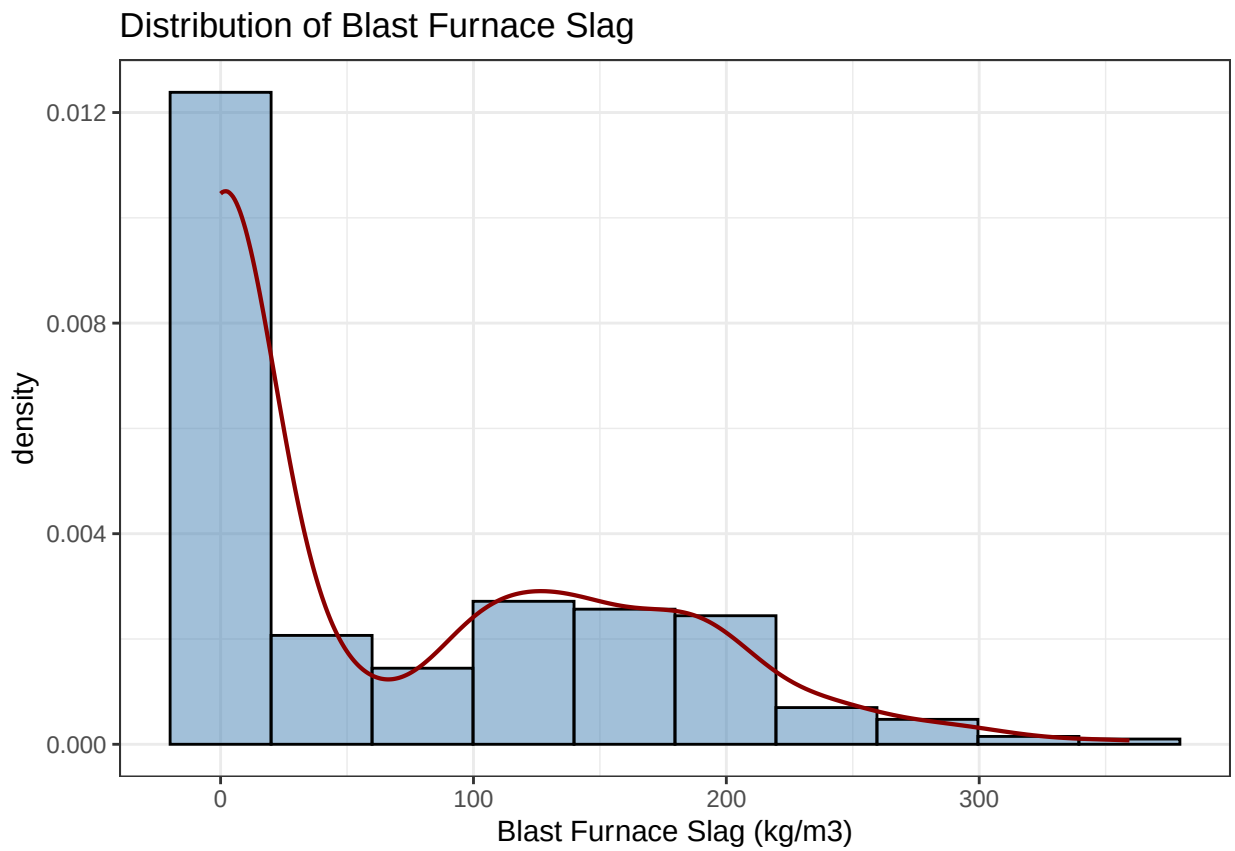
Cement: Deviations from the line suggest non-normality, particularly in the tails.

```
# Shapiro-Wilk Test
shapiro.test(df$cement)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$cement
## W = 0.95528, p-value < 2.2e-16
```

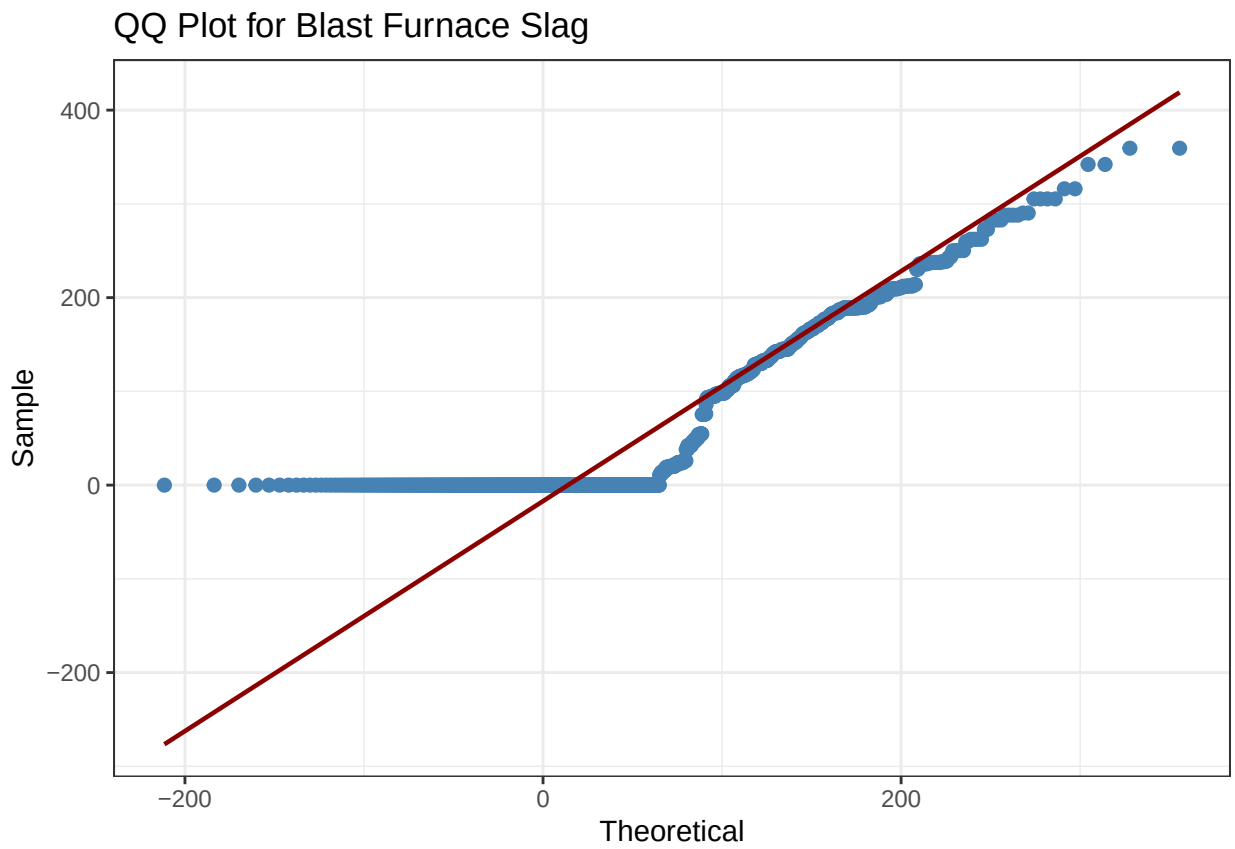
p-value < 0.05, Not normal

```
df %>%
  ggplot(aes(x=blast_furnace_slag))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Blast Furnace Slag", x = 'Blast Furnace Slag (kg/m3)')+
  theme_bw()
```



Blast Furnace Slag: Strong right skew, with many zero values.

```
# QQ plot
df %>%
  ggplot(aes(sample=blast_furnace_slag))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Blast Furnace Slag", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



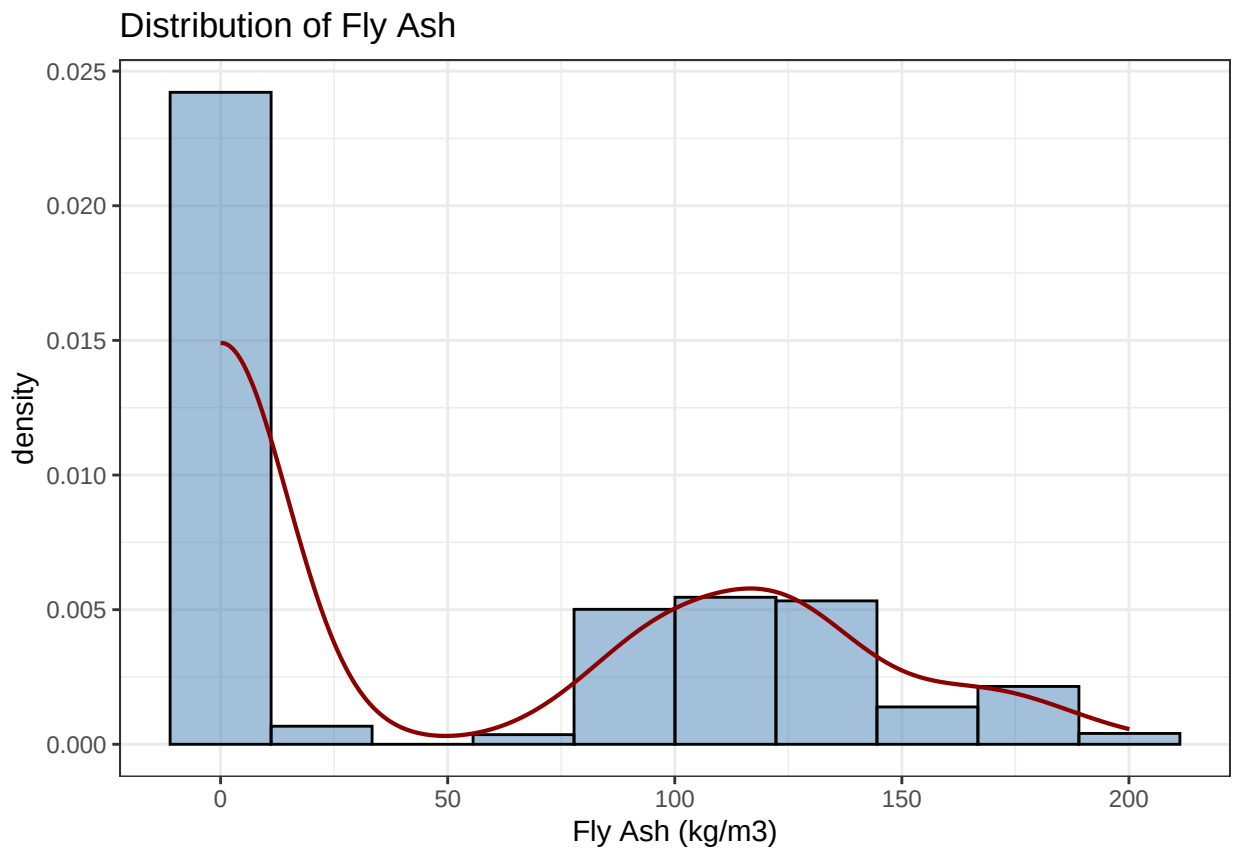
Blast Furnace Slag: Significant deviations from normality, confirming the right-skewed nature.

```
# Shapiro-Wilk Test
shapiro.test(df$blast_furnace_slag)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$blast_furnace_slag
## W = 0.80469, p-value < 2.2e-16
```

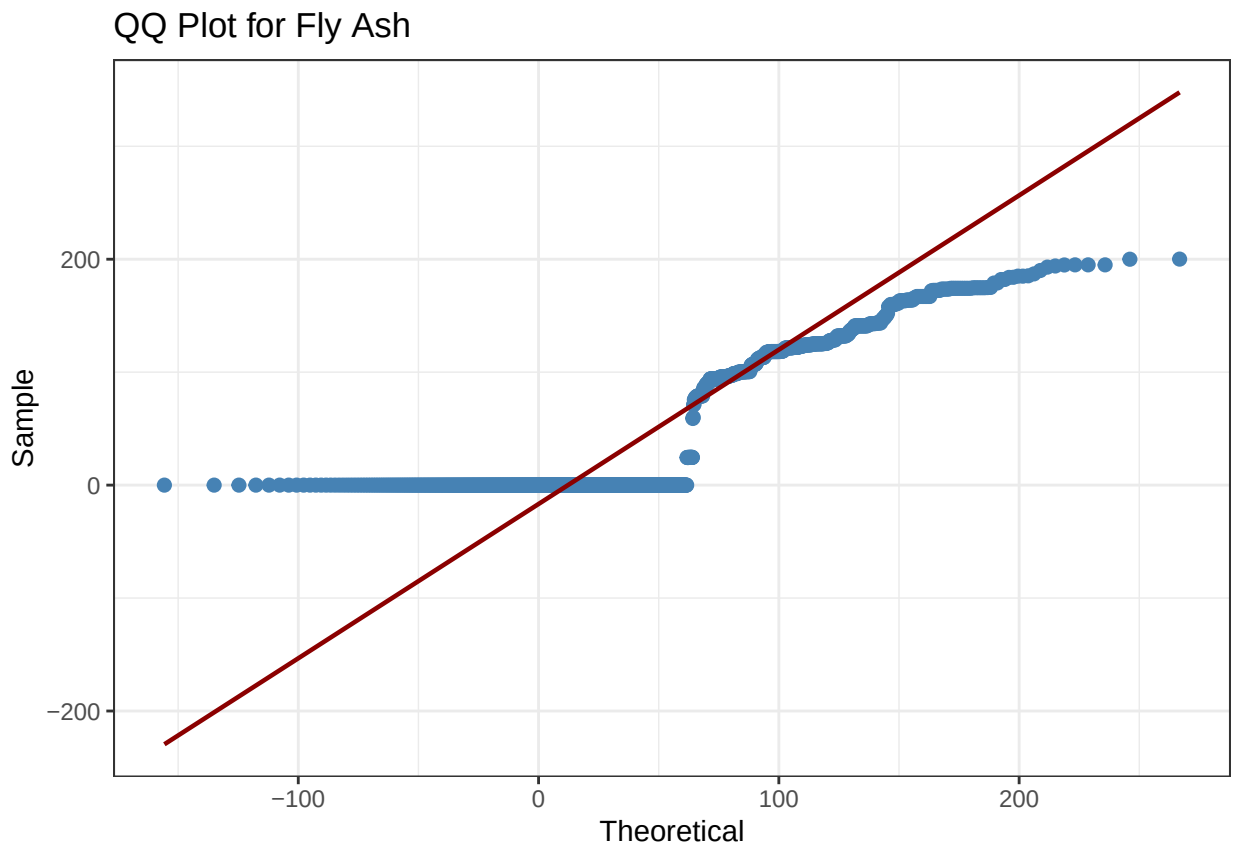
p-value < 0.05, Not normal

```
df %>%
  ggplot(aes(x=fly_ash))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Fly Ash", x = "Fly Ash (kg/m3)")+
  theme_bw()
```



Fly Ash: Similar to blast furnace slag, right-skewed with many zero values.

```
# QQ plot
df %>%
  ggplot(aes(sample=fly_ash))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Fly Ash", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



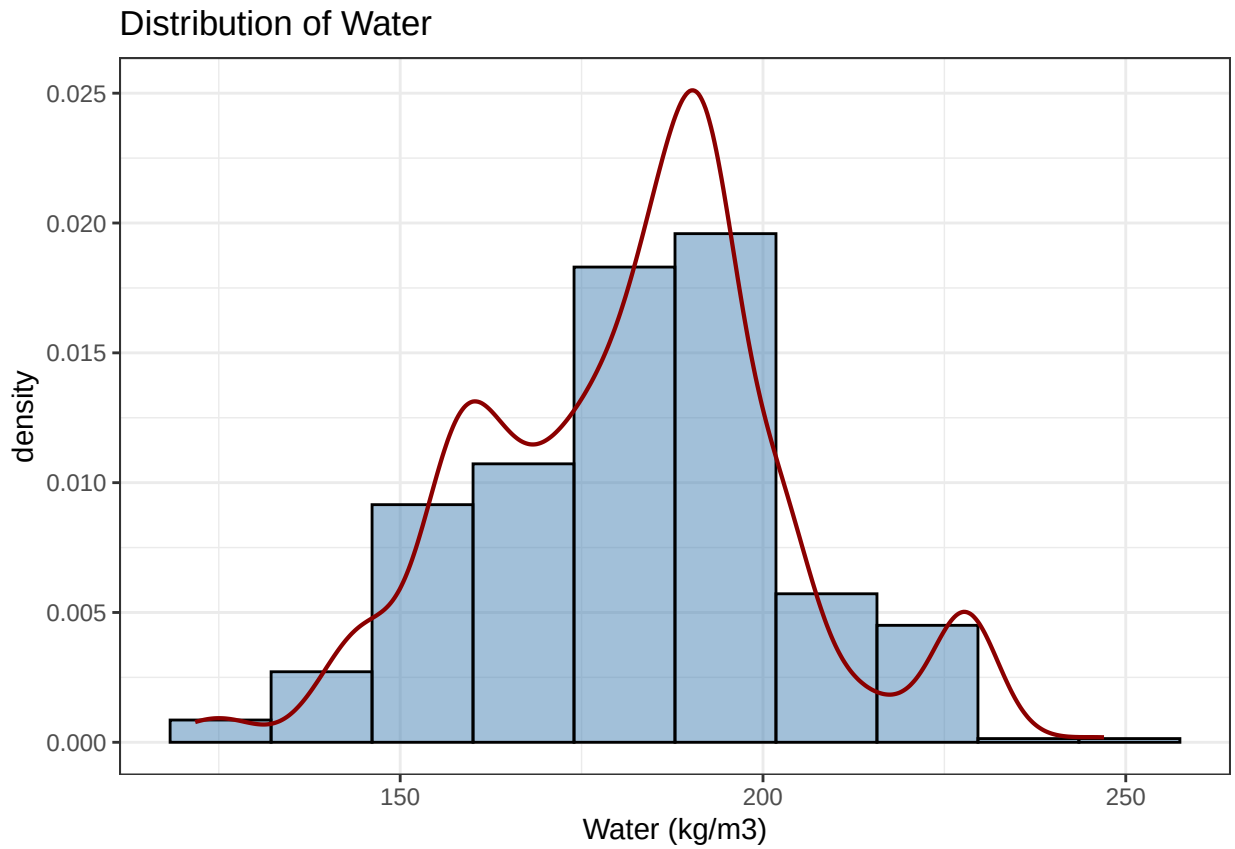
Fly Ash: Significant deviations from normality, confirming the right-skewed nature.

```
# Shapiro-Wilk Test
shapiro.test(df$fly_ash)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$fly_ash
## W = 0.76875, p-value < 2.2e-16
```

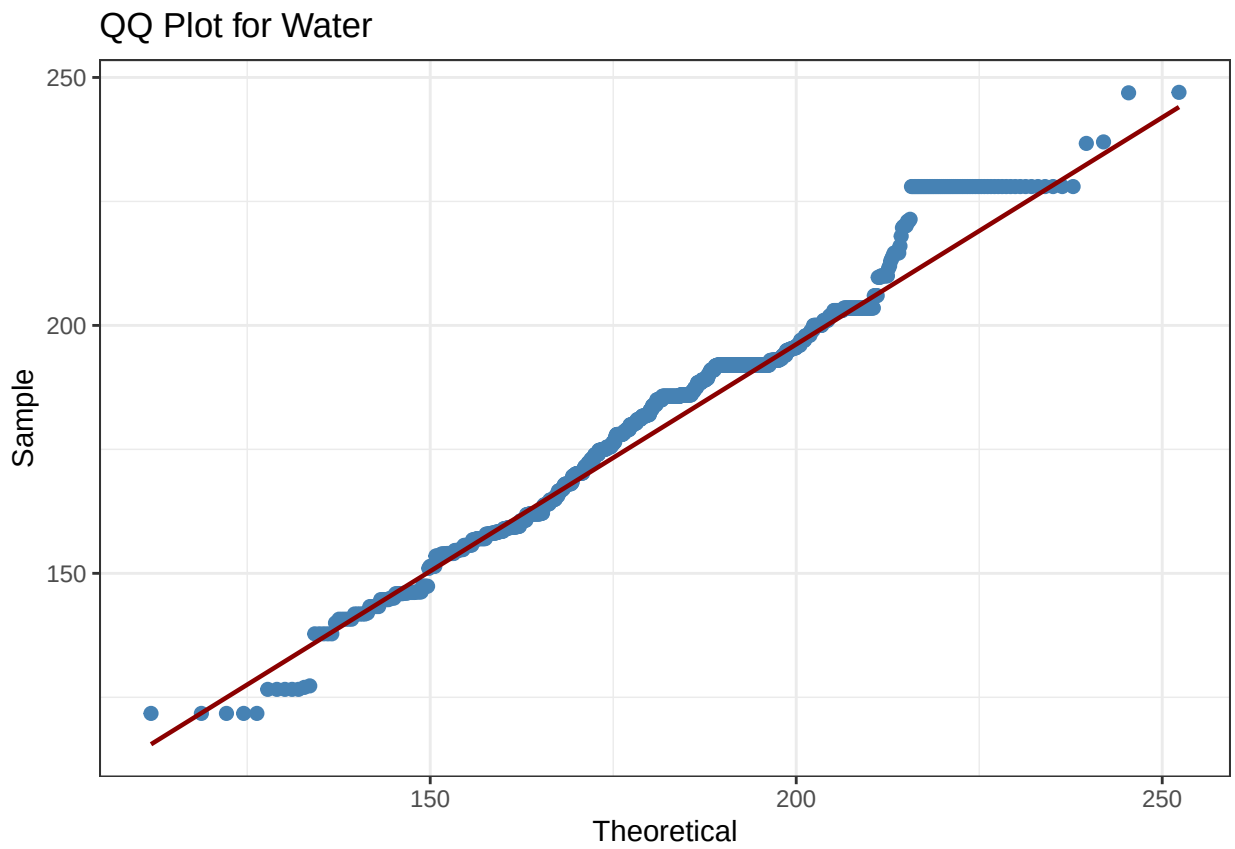
p-value < 0.05, Not normal

```
df %>%
  ggplot(aes(x=water))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Water", x = 'Water (kg/m3)')+
  theme_bw()
```



Water: More symmetrical distribution, with values mostly centered around the mean, suggesting less skew.

```
# QQ plot
df %>%
  ggplot(aes(sample=water))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Water", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```

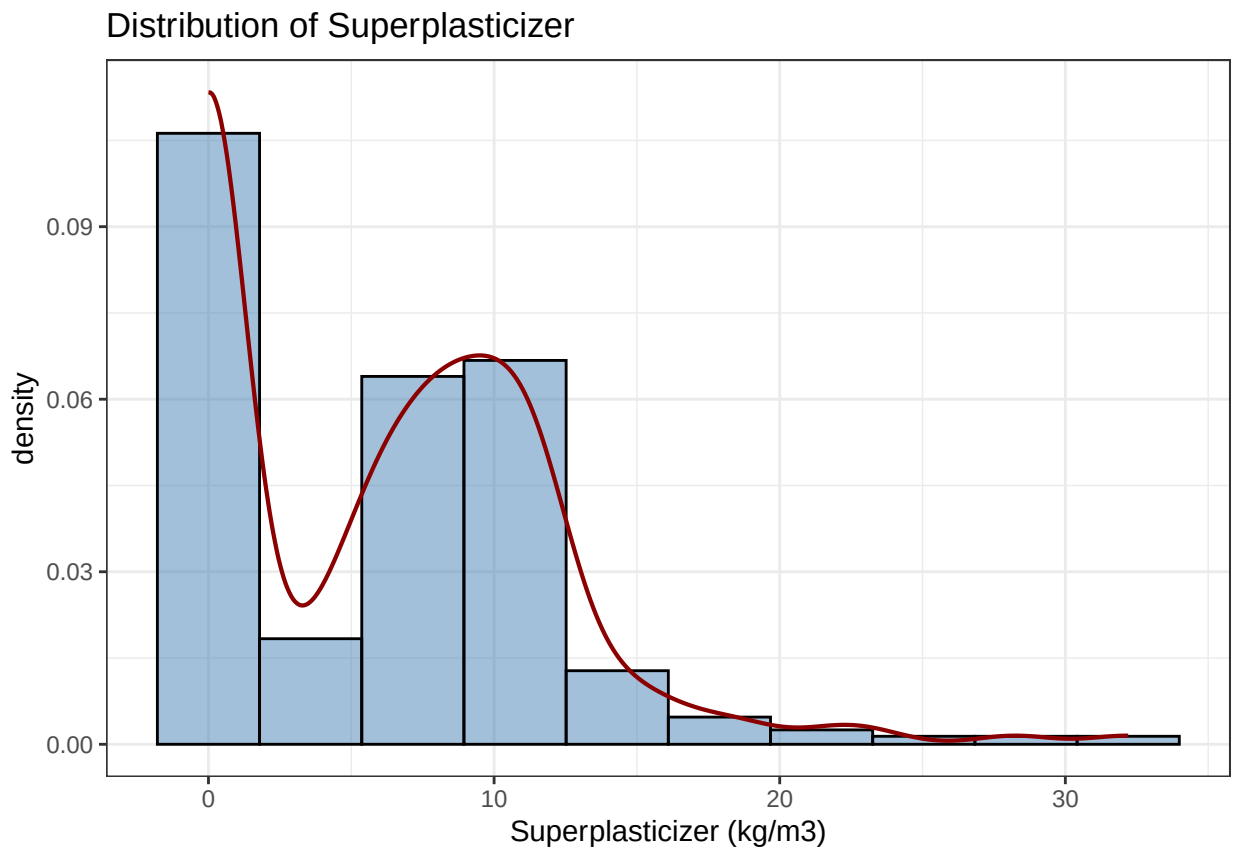


Water: Close to normal, but slight deviations in the tails.

```
# Shapiro-Wilk Test
shapiro.test(df$water)
```

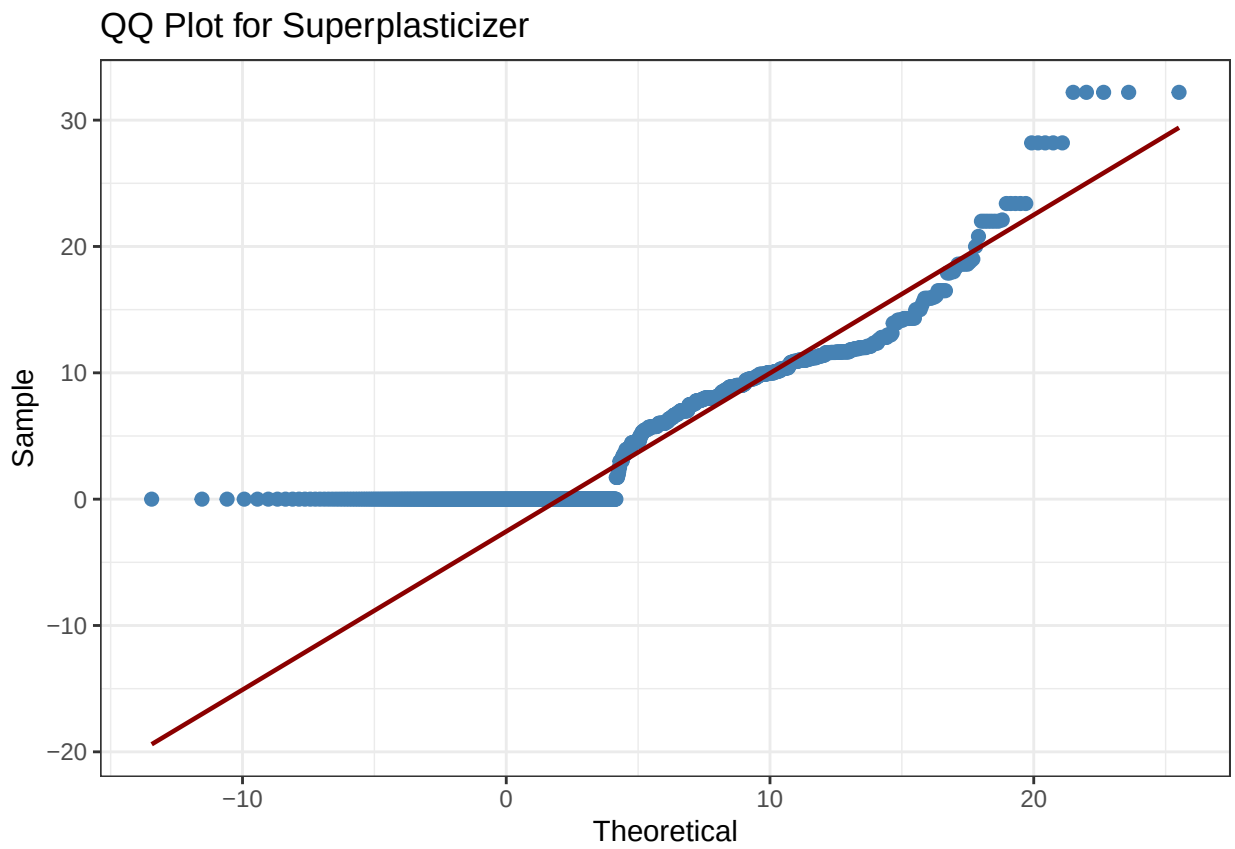
```
##
##  Shapiro-Wilk normality test
##
## data:  df$water
## W = 0.97972, p-value = 1.274e-10
```

```
df %>%
  ggplot(aes(x=superplasticizer))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Superplasticizer", x = 'Superplasticizer (kg/m3)')+
  theme_bw()
```



Superplasticizer: Highly skewed, with many zero values.

```
# QQ plot
df %>%
  ggplot(aes(sample=superplasticizer))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Superplasticizer", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```

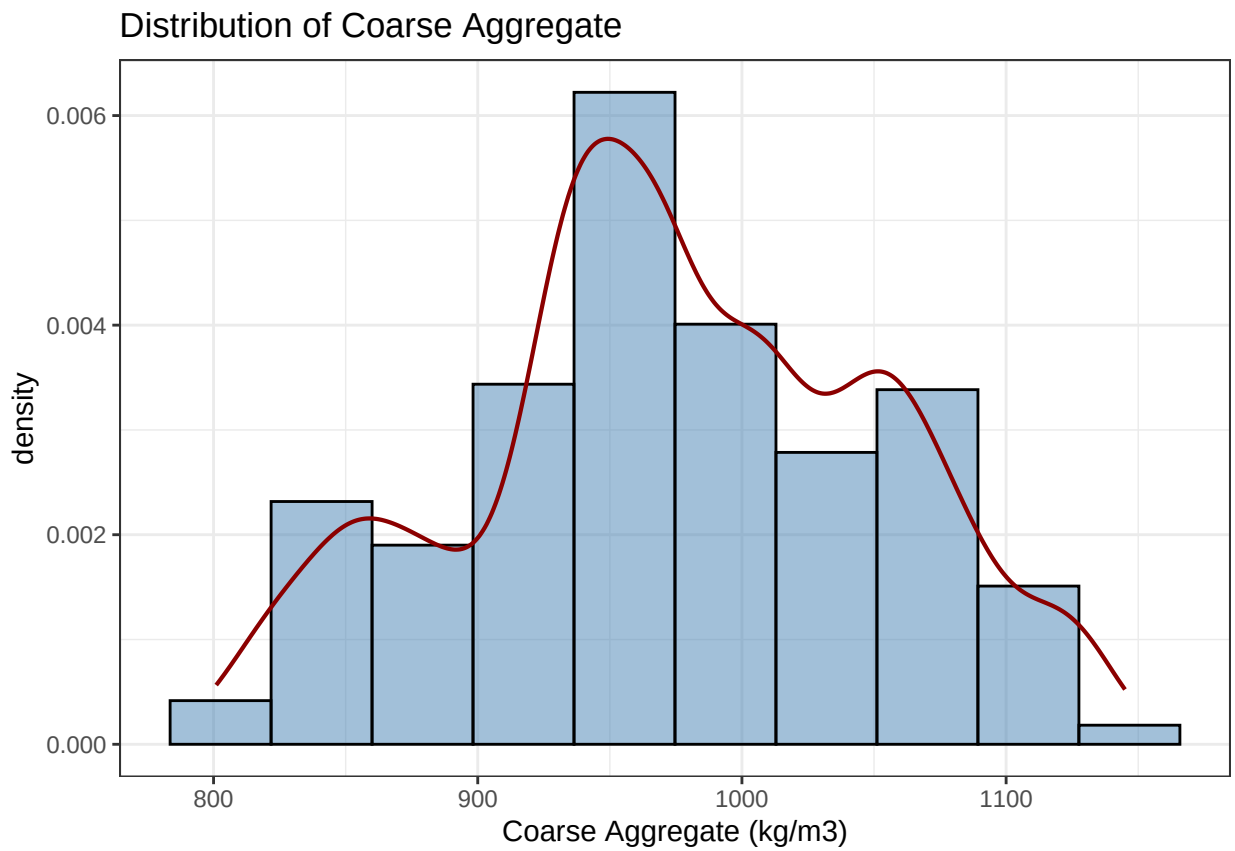


Superplasticizer: Deviates from the line, confirming non-normality.

```
# Shapiro-Wilk Test
shapiro.test(df$superplasticizer)
```

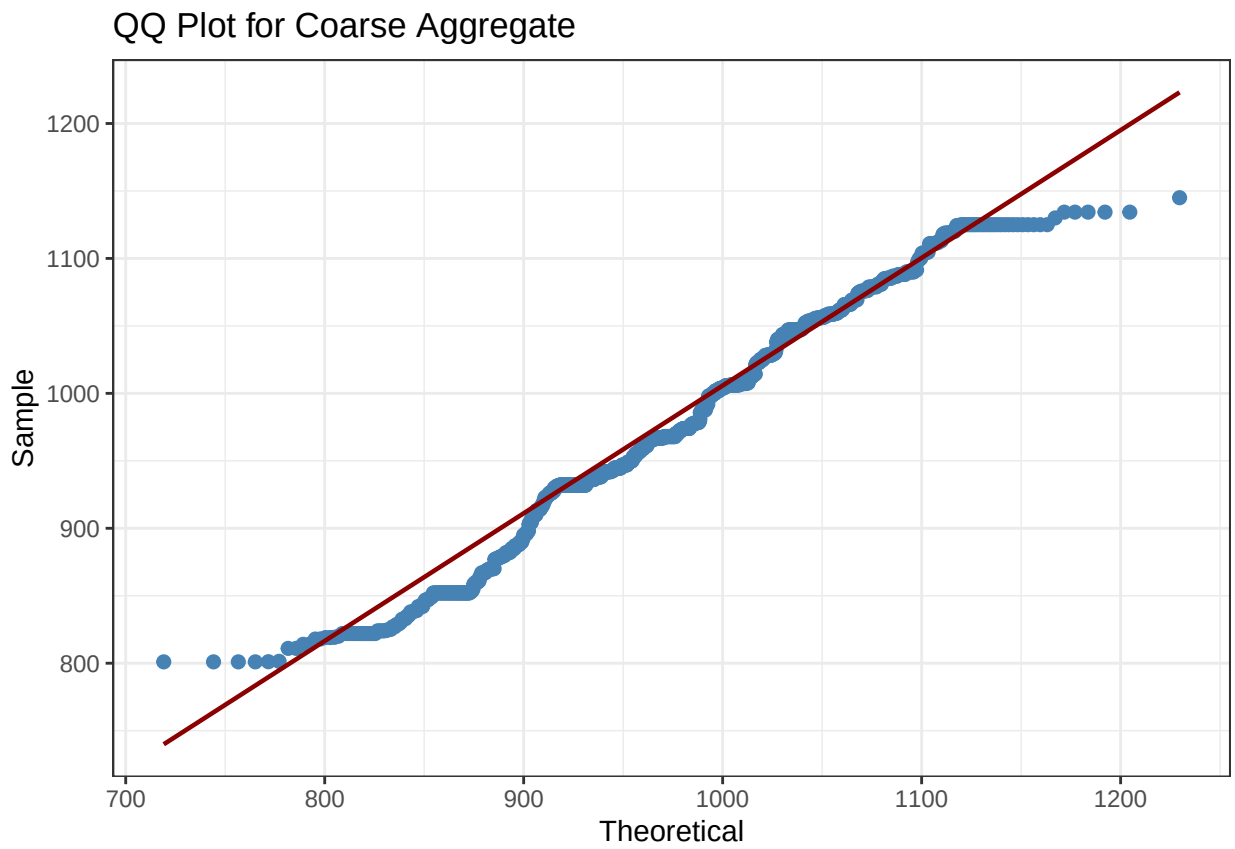
```
##
##  Shapiro-Wilk normality test
##
## data:  df$superplasticizer
## W = 0.85861, p-value < 2.2e-16
```

```
df %>%
  ggplot(aes(x=coarse_aggregate))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Coarse Aggregate", x = 'Coarse Aggregate (kg/m3)')+
  theme_bw()
```



Coarse Aggregate: Roughly symmetric

```
# QQ plot
df %>%
  ggplot(aes(sample=coarse_aggregate))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Coarse Aggregate", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```

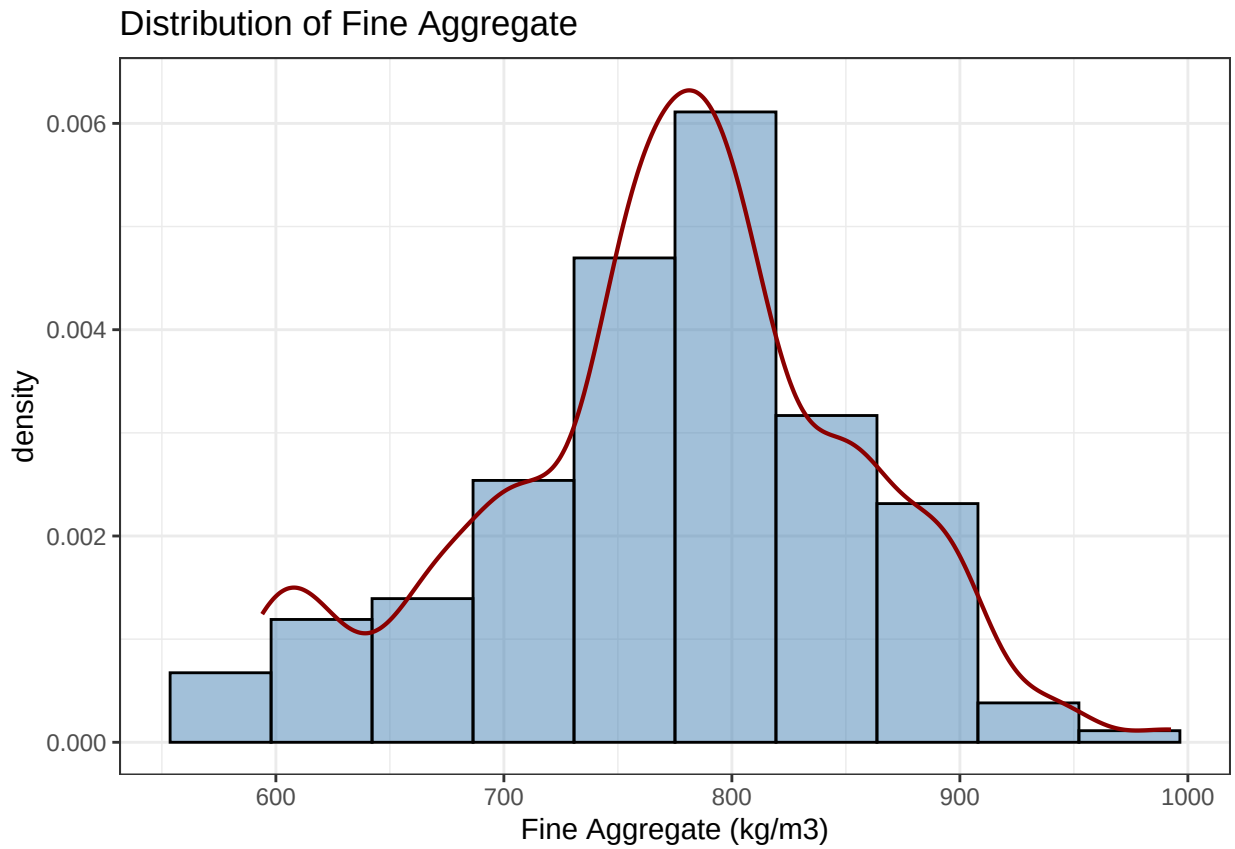


Coarse Aggregate: slight deviations, probably none normal.

```
# Shapiro-Wilk Test
shapiro.test(df$coarse_aggregate)
```

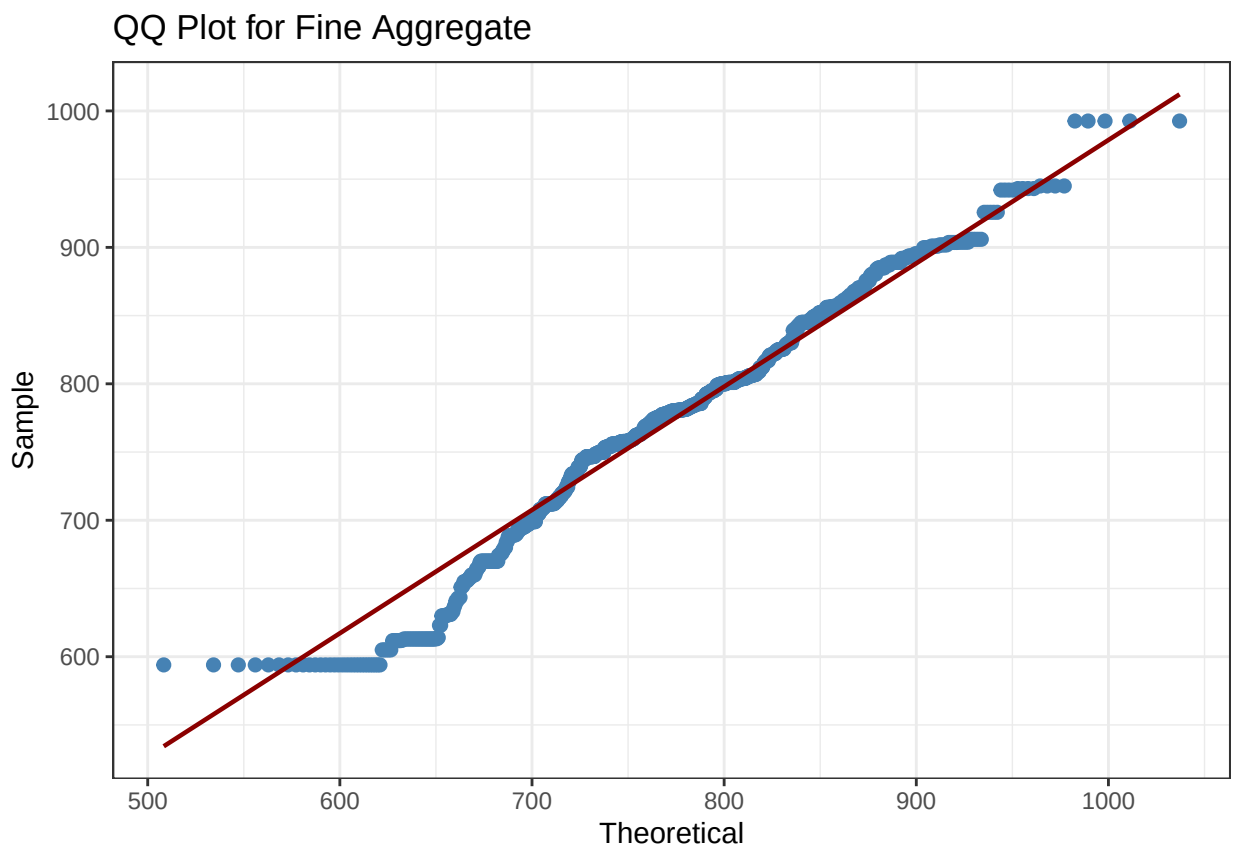
```
##
##  Shapiro-Wilk normality test
##
## data:  df$coarse_aggregate
## W = 0.98335, p-value = 2.679e-09
```

```
df %>%
  ggplot(aes(x=fine_aggregate))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Fine Aggregate", x = 'Fine Aggregate (kg/m3)')+
  theme_bw()
```



Fine Aggregate: Nearly normal, although slightly skewed to the left.

```
# QQ plot
df %>%
  ggplot(aes(sample=fine_aggregate))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Fine Aggregate", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```

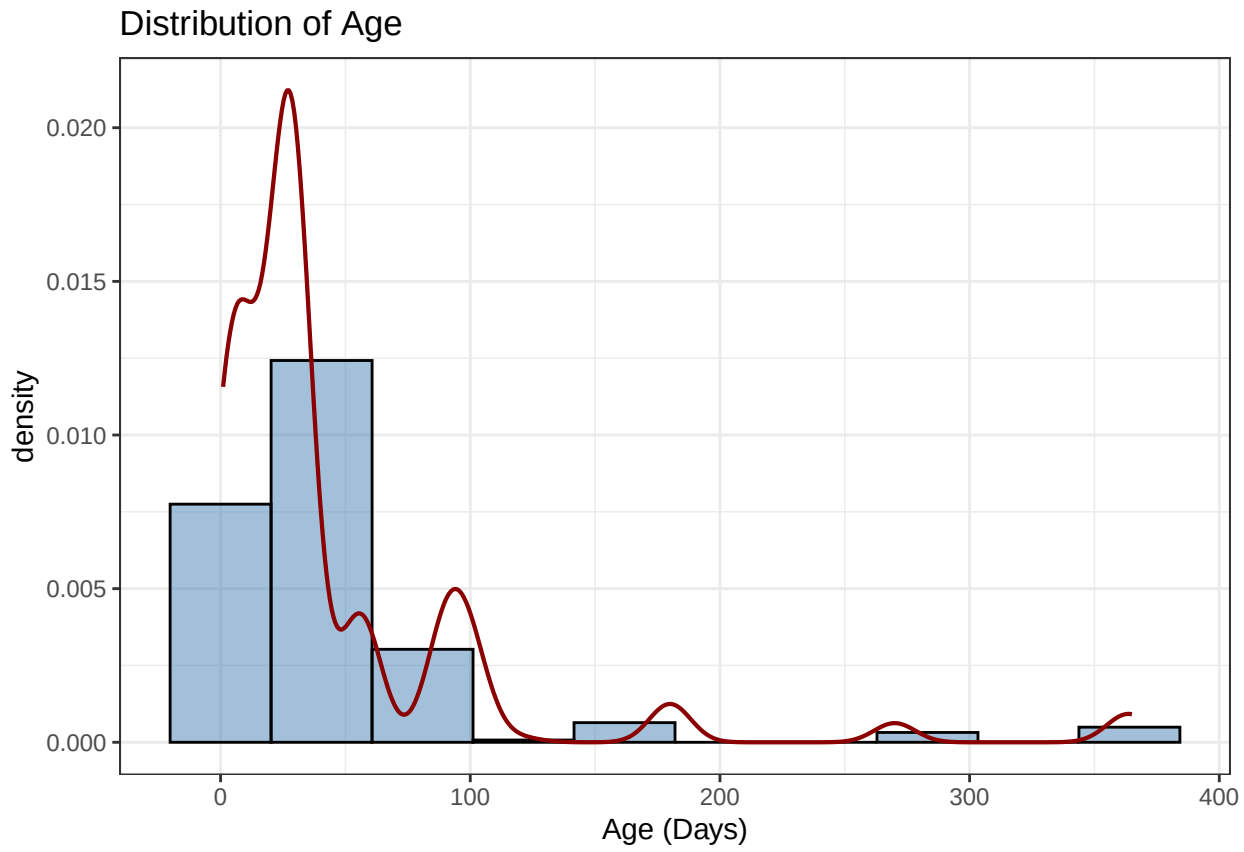


Fine Aggregate:slight deviations, probably none normal.

```
# Shapiro-Wilk Test
shapiro.test(df$fine_aggregate)
```

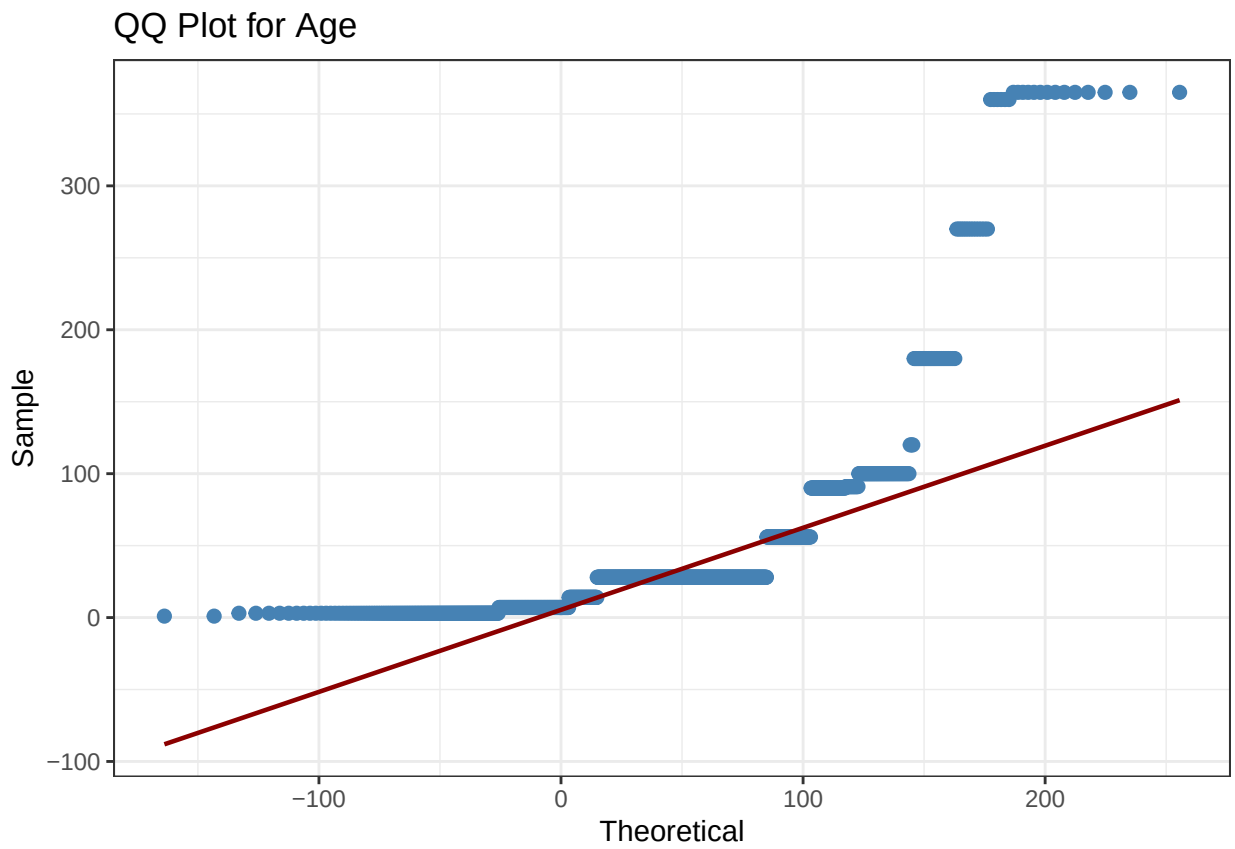
```
##
##  Shapiro-Wilk normality test
##
## data:  df$fine_aggregate
## W = 0.9809, p-value = 3.284e-10
```

```
df %>%
  ggplot(aes(x=age))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Age", x = 'Age (Days)')+
  theme_bw()
```



Age: Right-skewed, many samples at lower ages.

```
# QQ plot
df %>%
  ggplot(aes(sample=age))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Age", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```

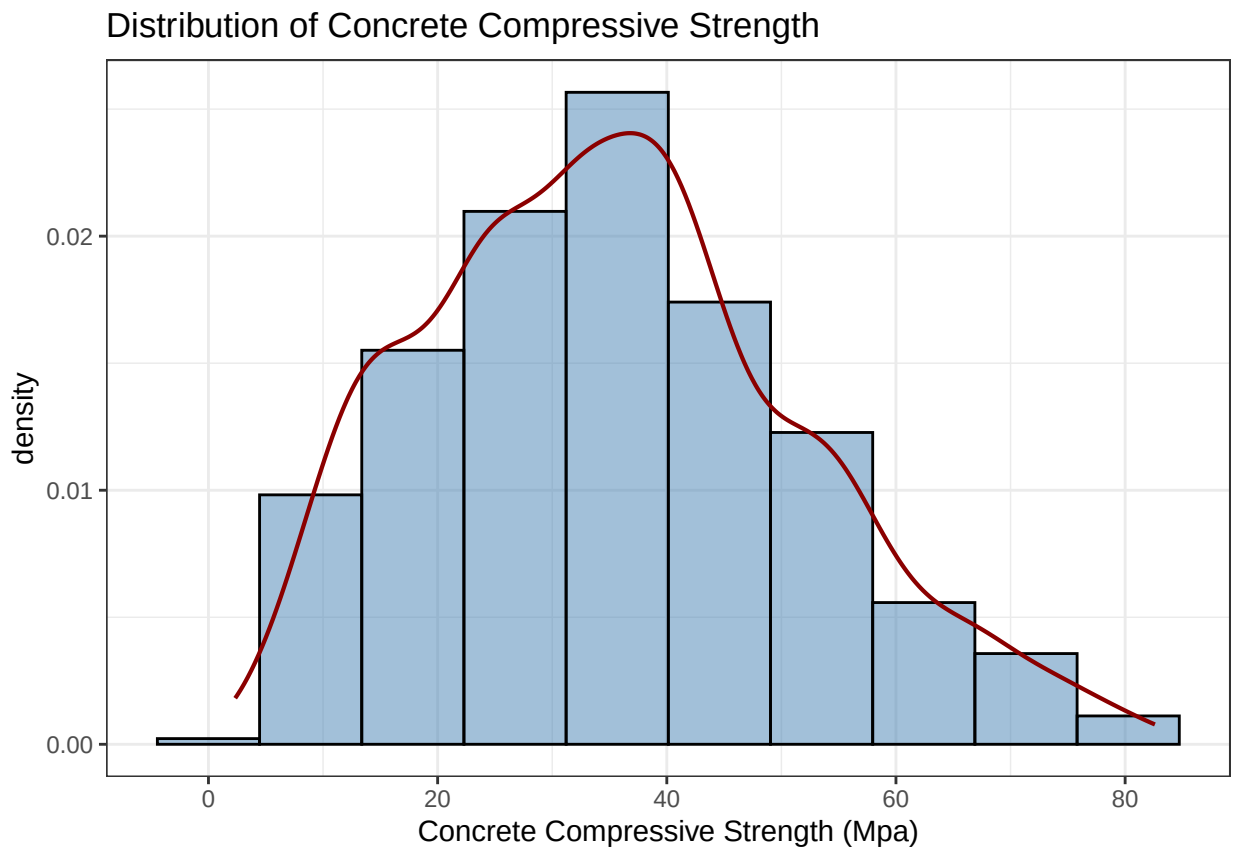


Age: Major deviations from the line, confirming non-normality.

```
# Shapiro-Wilk Test
shapiro.test(df$age)
```

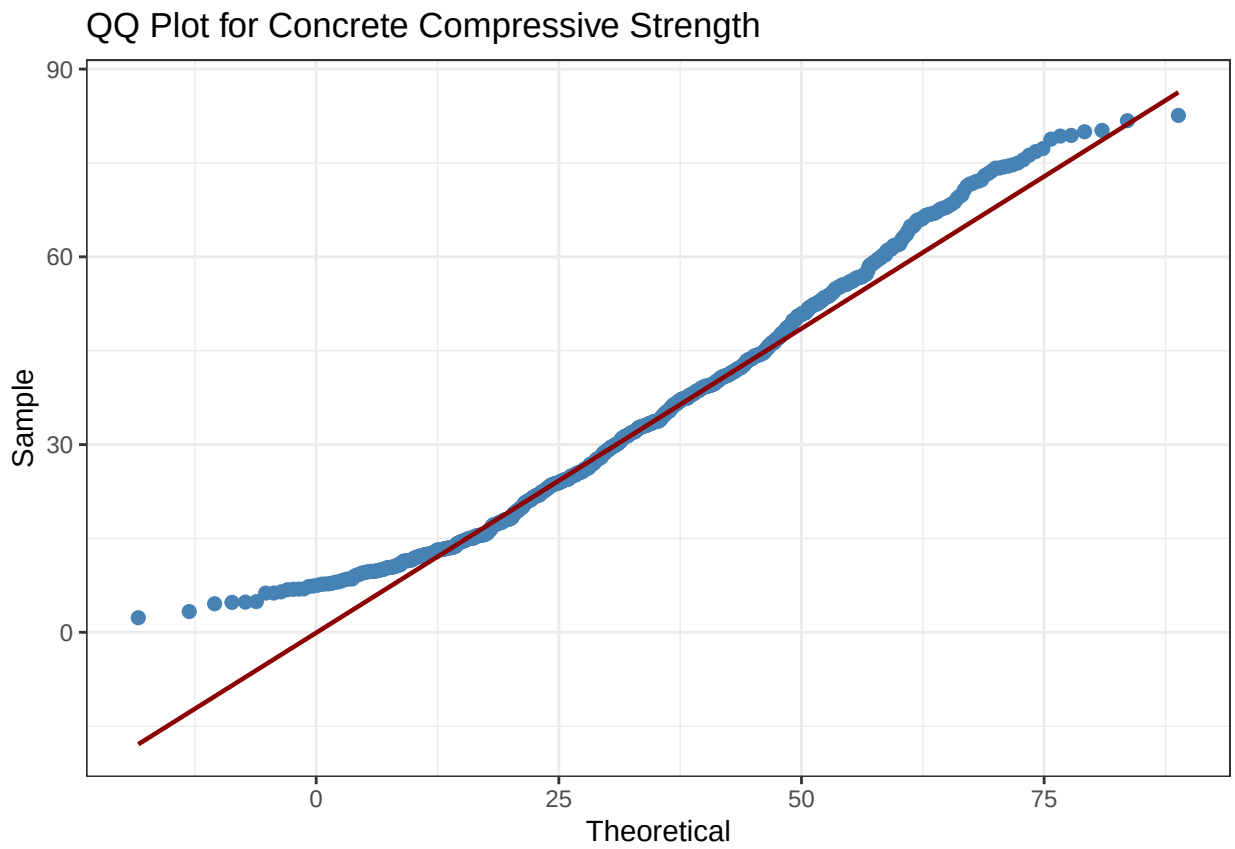
```
##
##  Shapiro-Wilk normality test
##
## data:  df$age
## W = 0.58849, p-value < 2.2e-16
```

```
df %>%
  ggplot(aes(x=strength))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, fill = 'steelblue', alpha = 0.5, color = 'bla
  geom_density(color = 'darkred', linewidth=0.75)+
  labs(title = "Distribution of Concrete Compressive Strength", x = 'Concrete Compressive Strength (Mpa
  theme_bw()
```



Compressive Strength: Somewhat symmetric but with slight right skew.

```
# QQ plot
df %>%
  ggplot(aes(sample=strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Concrete Compressive Strength", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



Compressive Strength: Minor deviations, could be normal?

```
# Shapiro-Wilk Test
shapiro.test(df$strength)
```

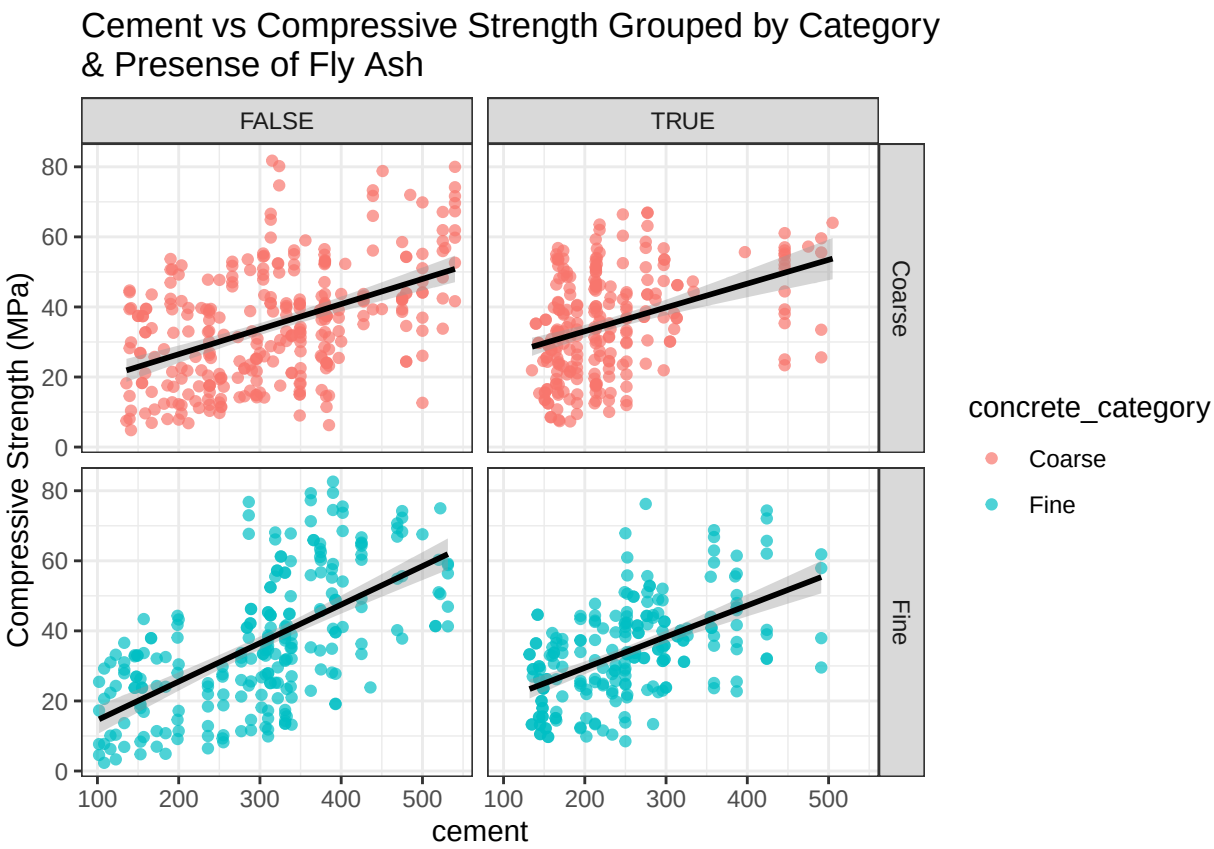
```
##
##  Shapiro-Wilk normality test
##
## data:  df$strength
## W = 0.98174, p-value = 6.651e-10
```

All p-values are less than 0.05, so we reject the null hypothesis of normality, confirming that all the continuous columns are non-normal.

PLOT SCATTER PLOTS VS COMPRESSIVE STRENGTHS FOR NUMERIC COLUMNS

```
df %>%
  ggplot(aes(x=cement, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "Cement vs Compressive Strength Grouped by Category \n& Presense of Fly Ash",
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

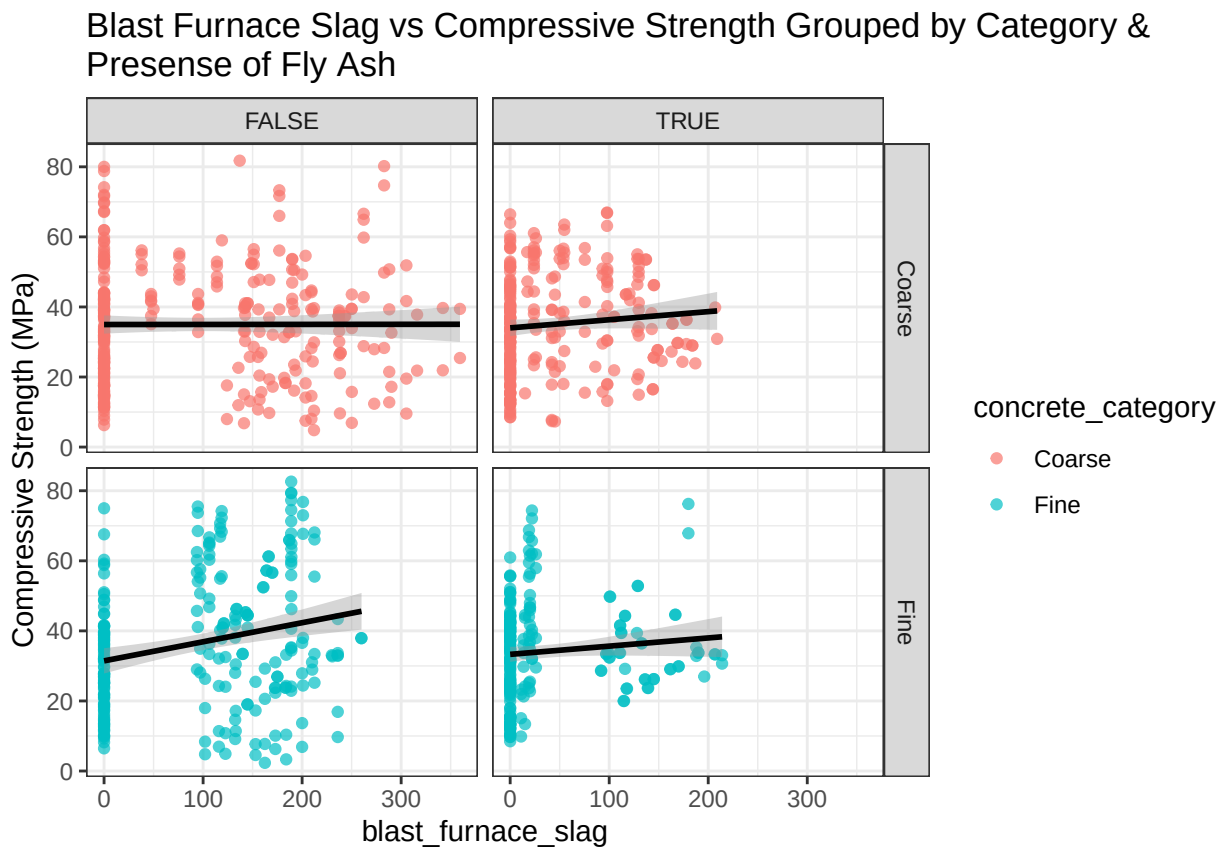
```
## `geom_smooth()` using formula = 'y ~ x'
```



Cement vs. Compressive Strength: Positive trend across categories, indicating that higher cement content generally increases compressive strength

```
df %>%
  ggplot(aes(x=blast_furnace_slag, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "Blast Furnace Slag vs Compressive Strength Grouped by Category & \nPresense of Fly Ash"
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

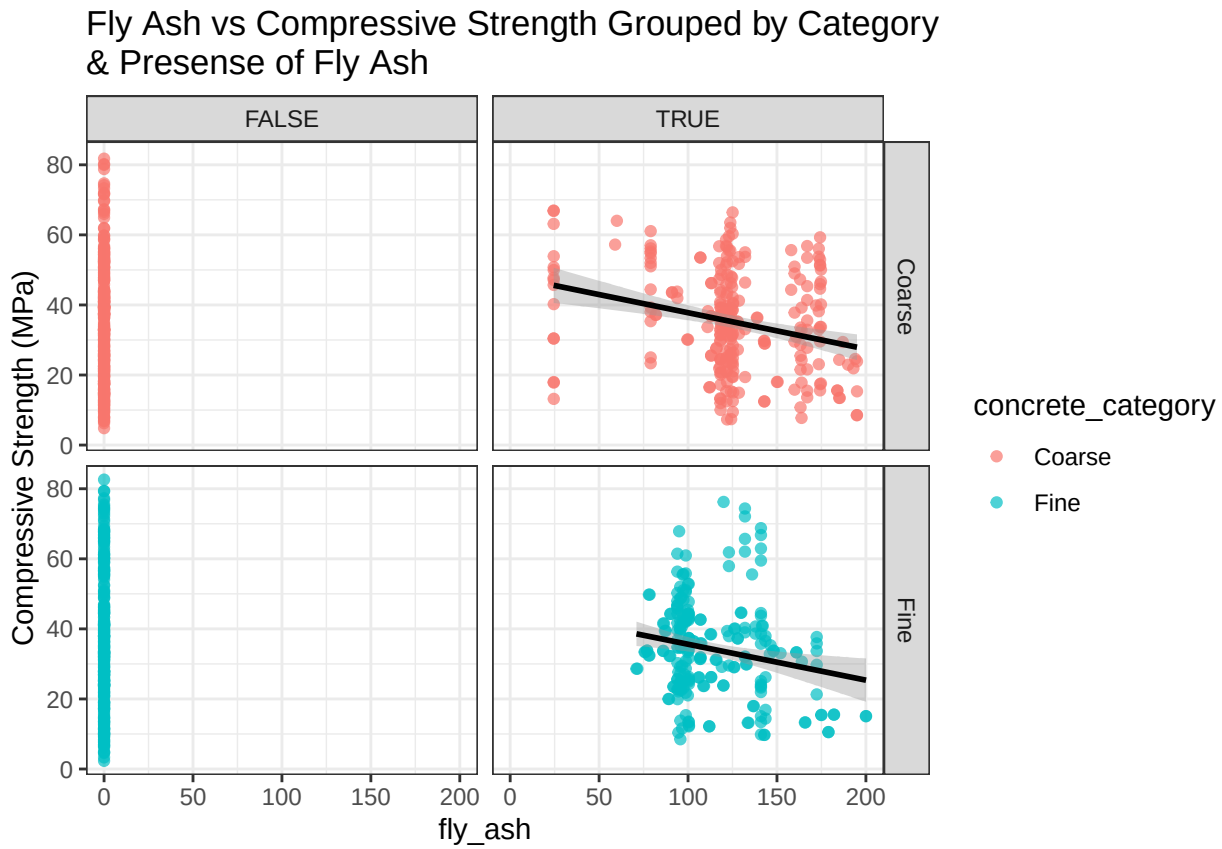
```
## `geom_smooth()` using formula = 'y ~ x'
```



Blast Furnace Slag: somewhat positive relationship with compressive strength.

```
df %>%
  ggplot(aes(x=fly_ash, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "Fly Ash vs Compressive Strength Grouped by Category \n& Presense of Fly Ash",
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

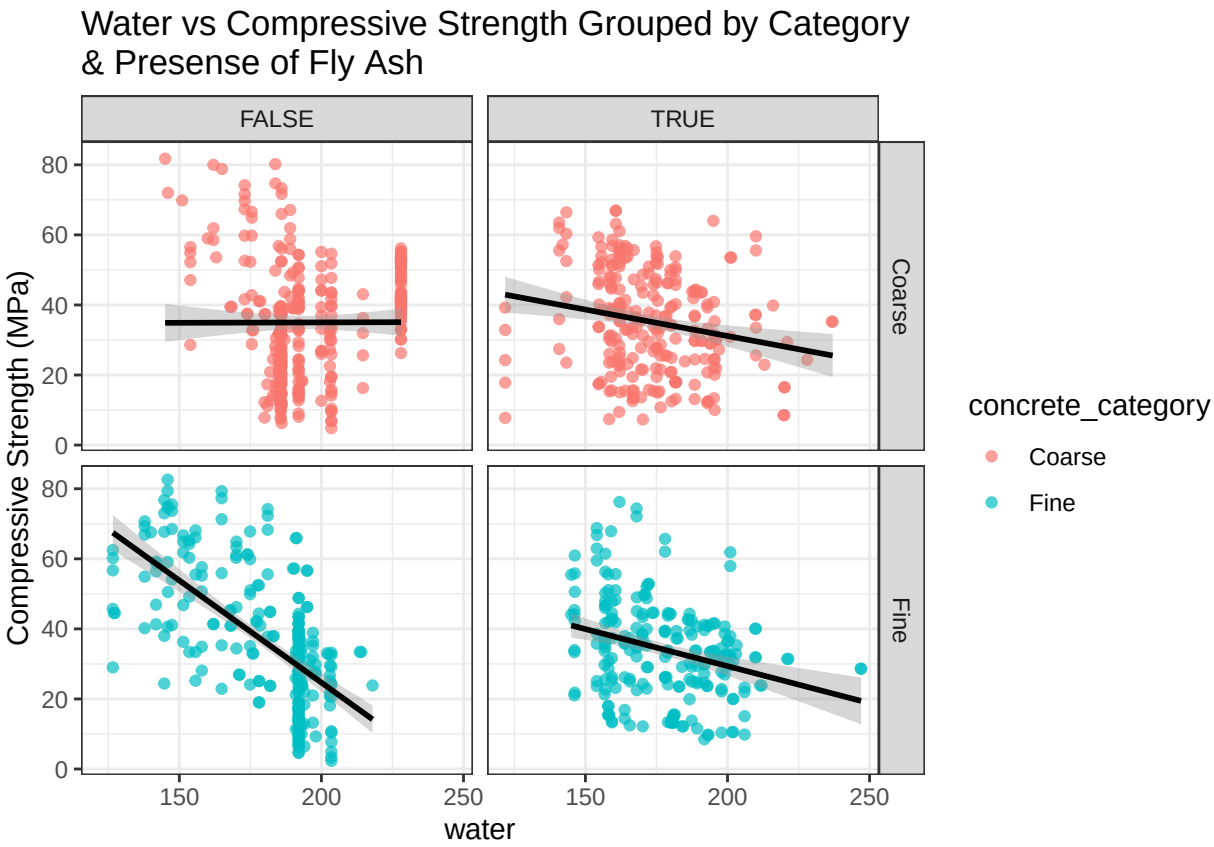
```
## `geom_smooth()` using formula = 'y ~ x'
```



Fly Ash: Multiple observations without fly ash, inverse relationship with concrete strength.

```
df %>%
  ggplot(aes(x=water, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "Water vs Compressive Strength Grouped by Category \n& Presense of Fly Ash",
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

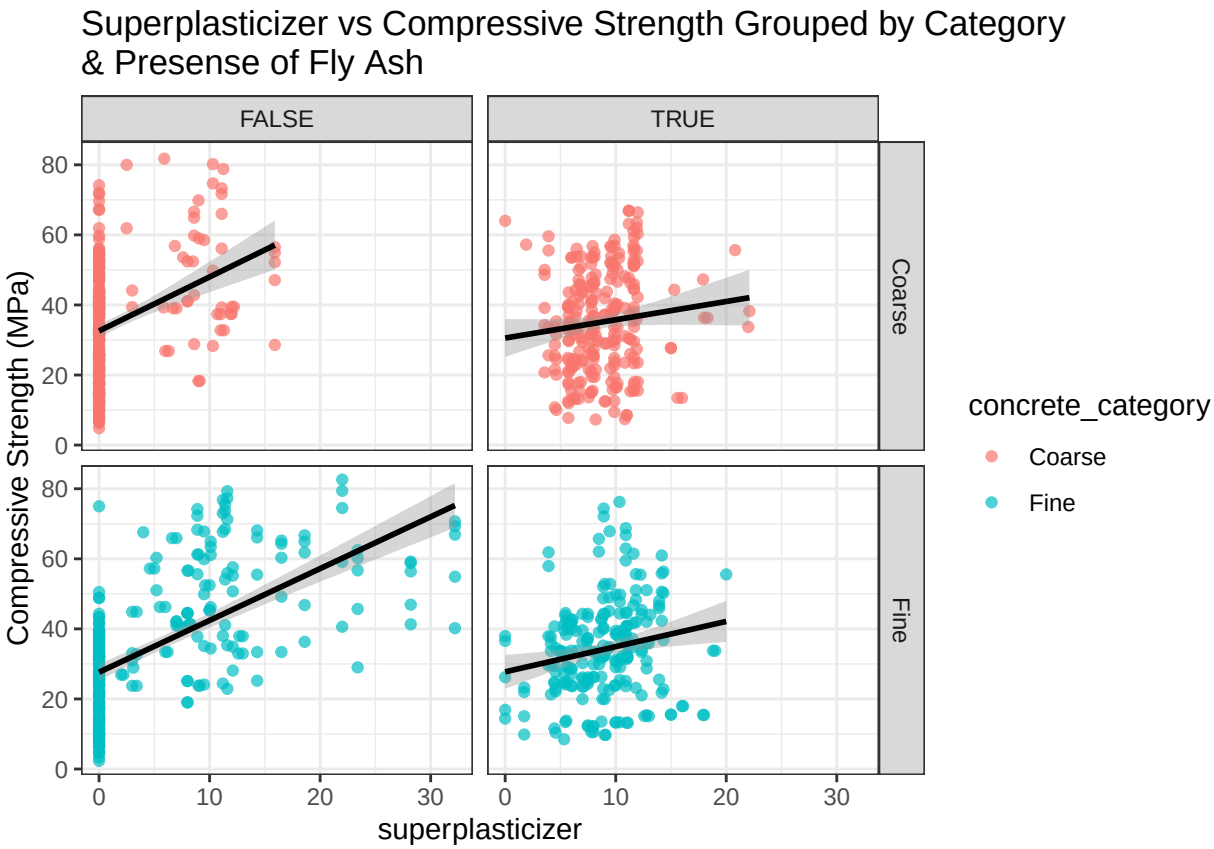
```
## `geom_smooth()` using formula = 'y ~ x'
```



Water: Inverse relationship, suggesting higher water content may reduce compressive strength.

```
df %>%
  ggplot(aes(x=superplasticizer, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "Superplasticizer vs Compressive Strength Grouped by Category \n& Presense of Fly Ash",
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

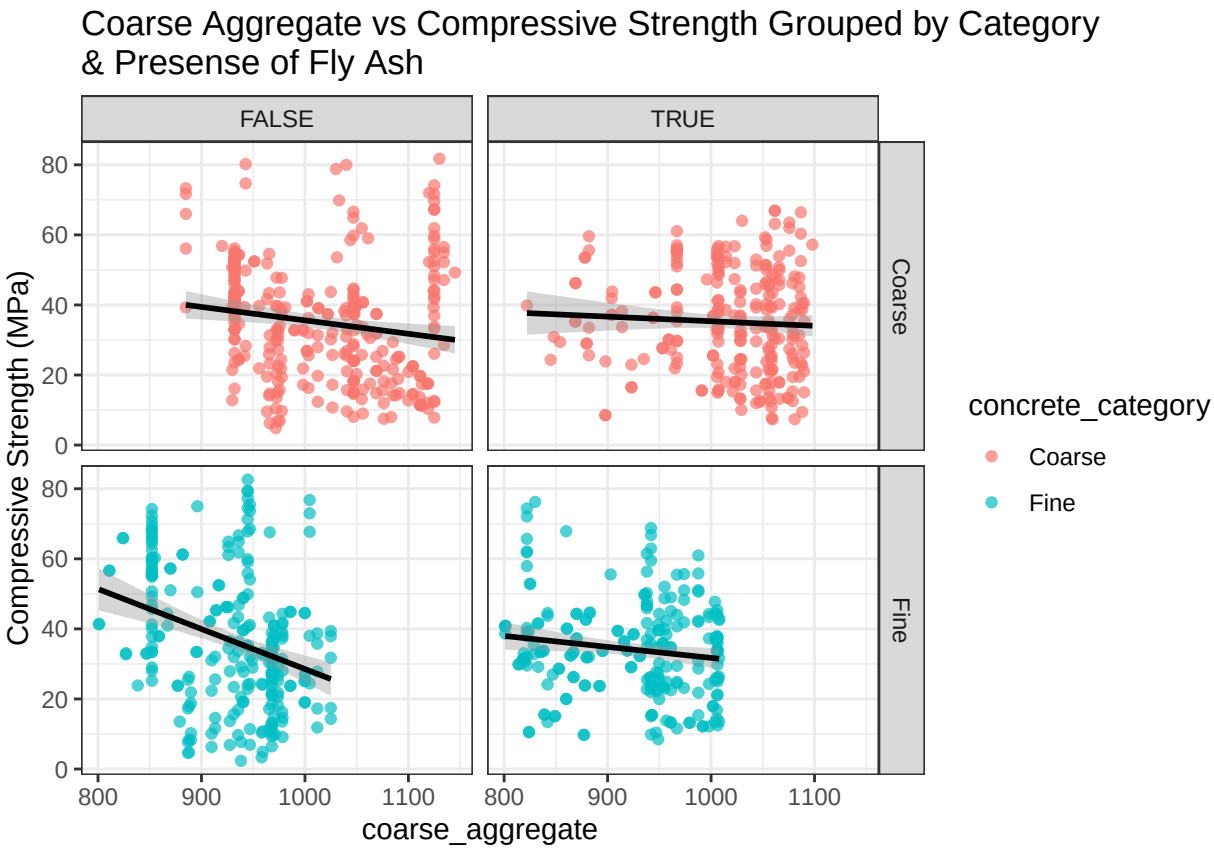
```
## `geom_smooth()` using formula = 'y ~ x'
```



Superplasticizer: Positive association with compressive strength in most cases.

```
df %>%
  ggplot(aes(x=coarse_aggregate, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "Coarse Aggregate vs Compressive Strength Grouped by Category \n& Presense of Fly Ash",
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

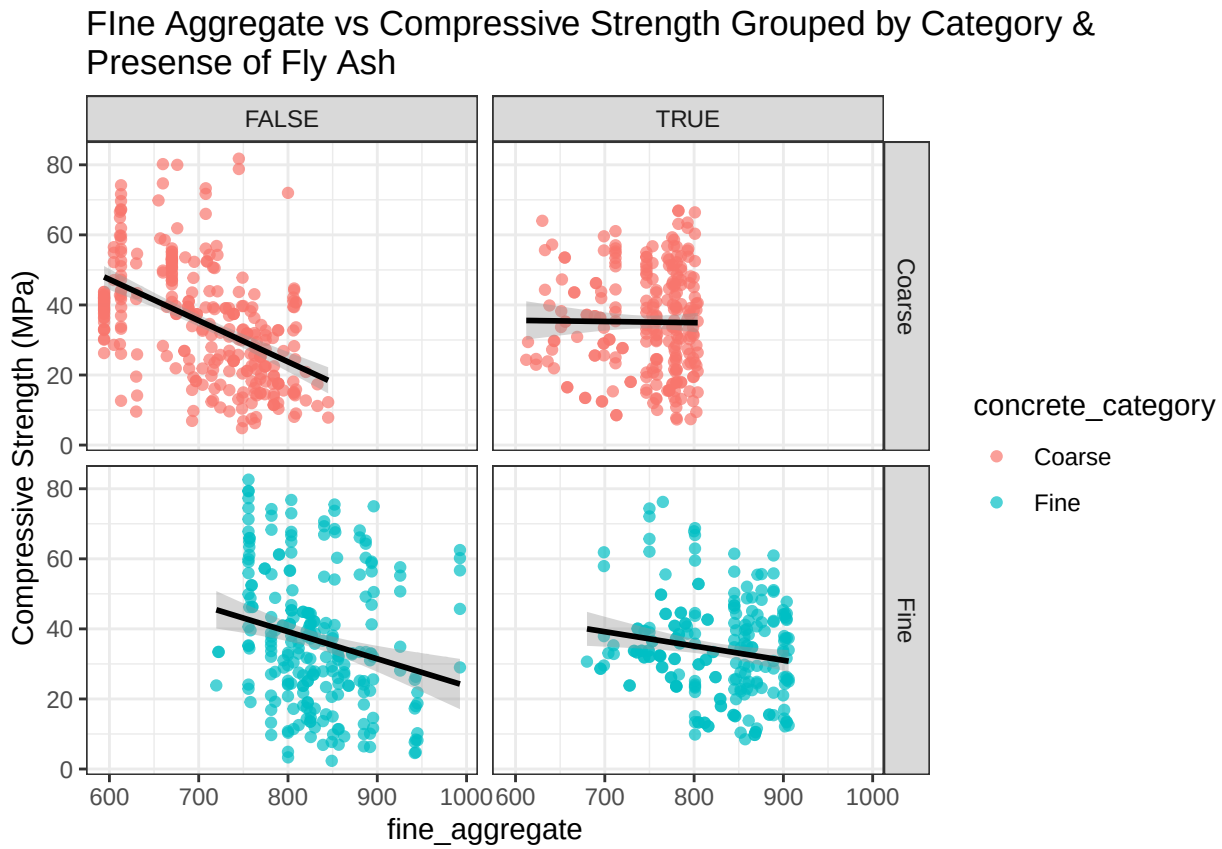
```
## `geom_smooth()` using formula = 'y ~ x'
```



Coarse Aggregate: Coarse aggregate shows slightly negative association

```
df %>%
  ggplot(aes(x=fine_aggregate, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "FIne Aggregate vs Compressive Strength Grouped by Category & \nPresense of Fly Ash",
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

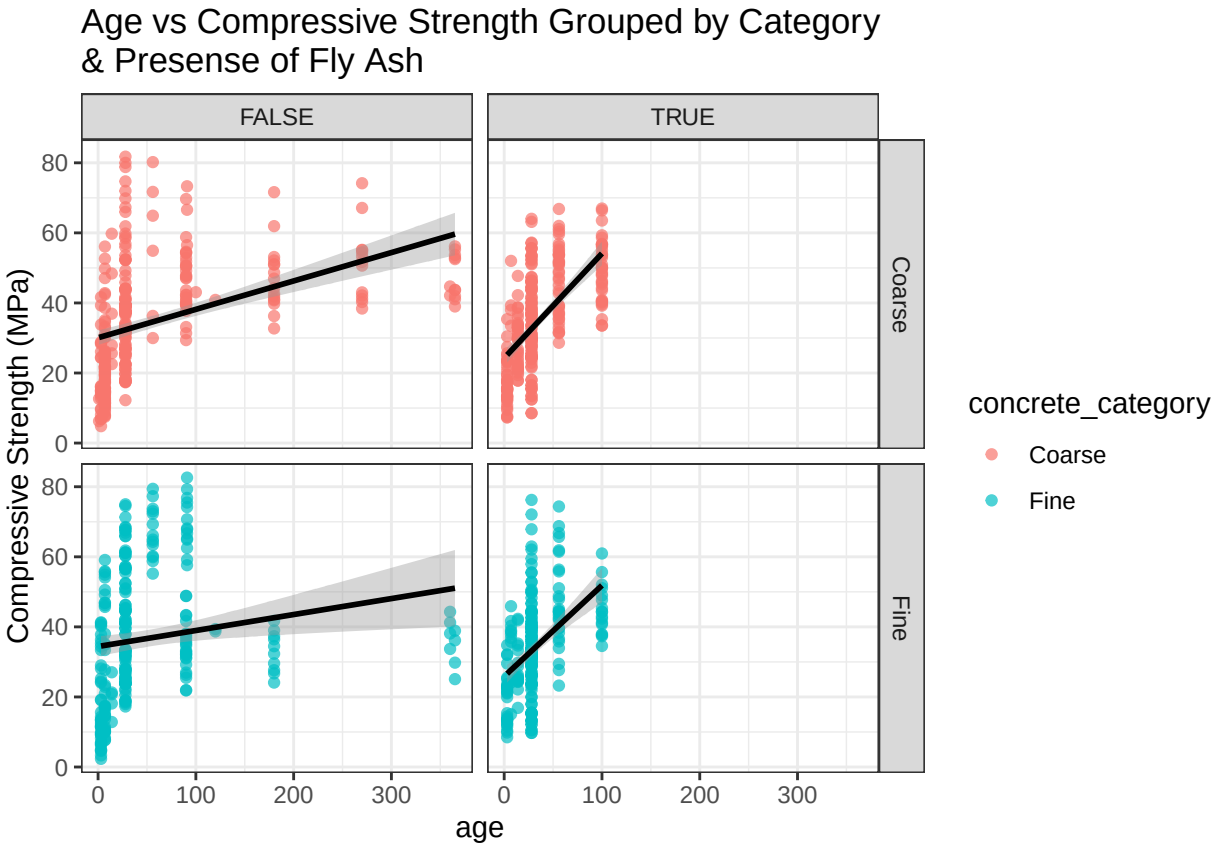
```
## `geom_smooth()` using formula = 'y ~ x'
```



Fine Aggregate: Coarse aggregate shows generally negative association

```
df %>%
  ggplot(aes(x=age, y=strength, color=concrete_category))+
  geom_point(alpha=0.7)+
  geom_smooth(color='black', method = "lm")+
  labs(title = "Age vs Compressive Strength Grouped by Category \n& Presense of Fly Ash",
       y="Compressive Strength (MPa)")+
  facet_grid(concrete_category ~ contains_fly_ash)+
  theme_bw()
```

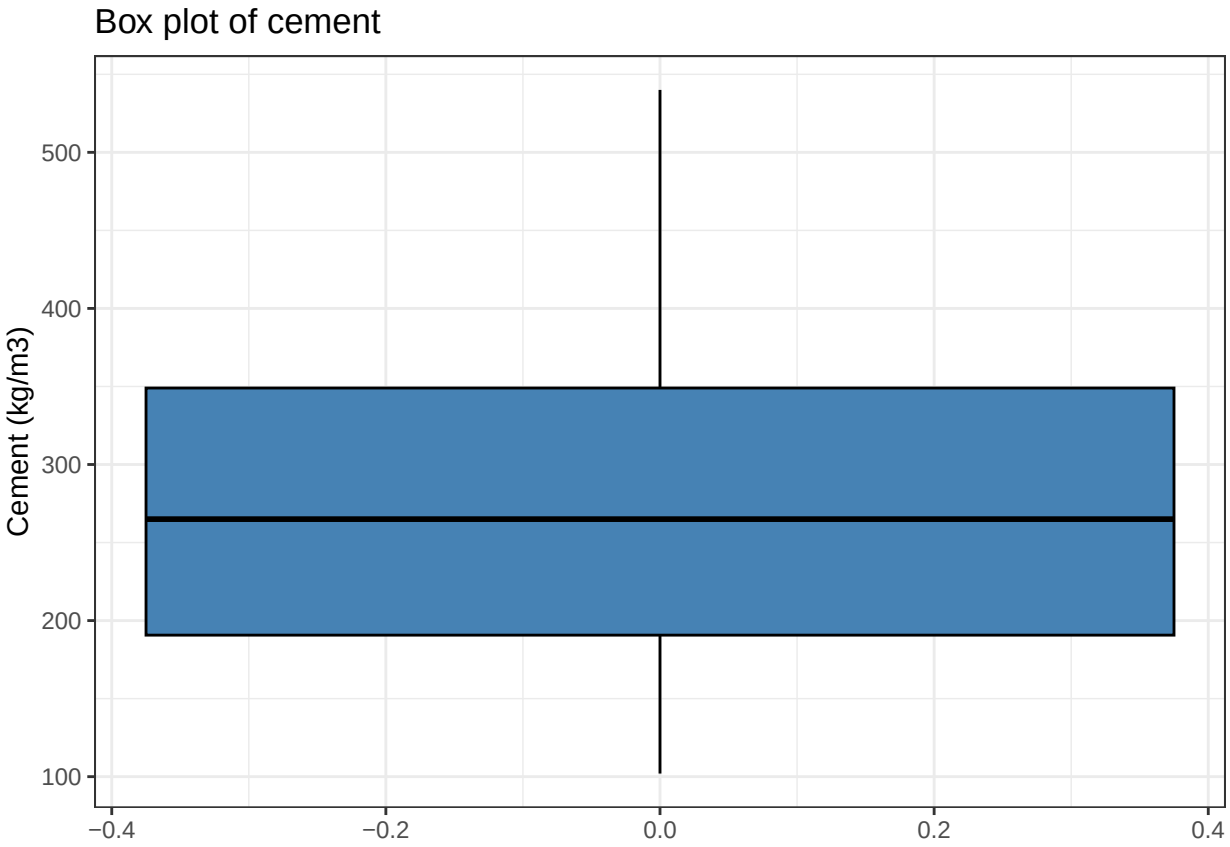
```
## `geom_smooth()` using formula = 'y ~ x'
```



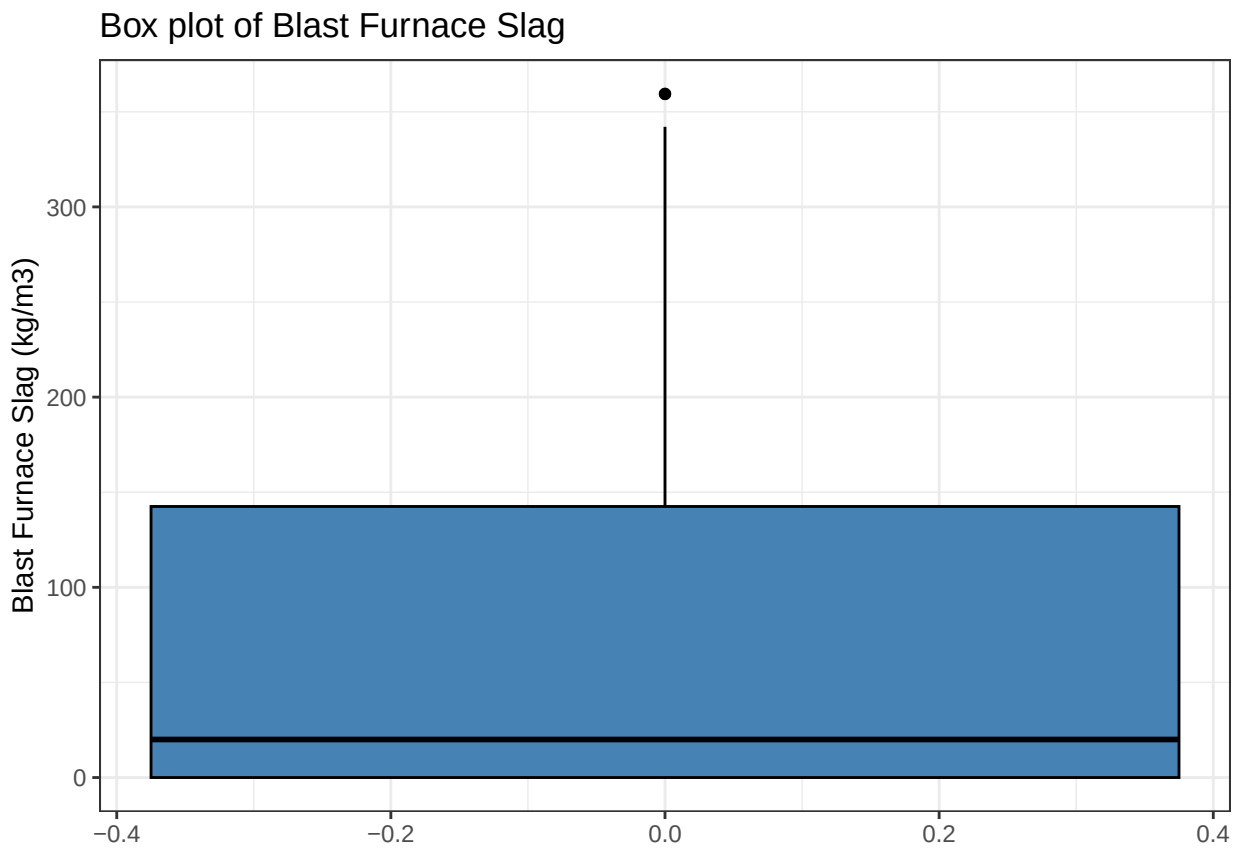
Age: Positive relationship with compressive strength, indicating concrete strengthens over time.

BOX PLOTS TO CHECK FOR OUTLIERS

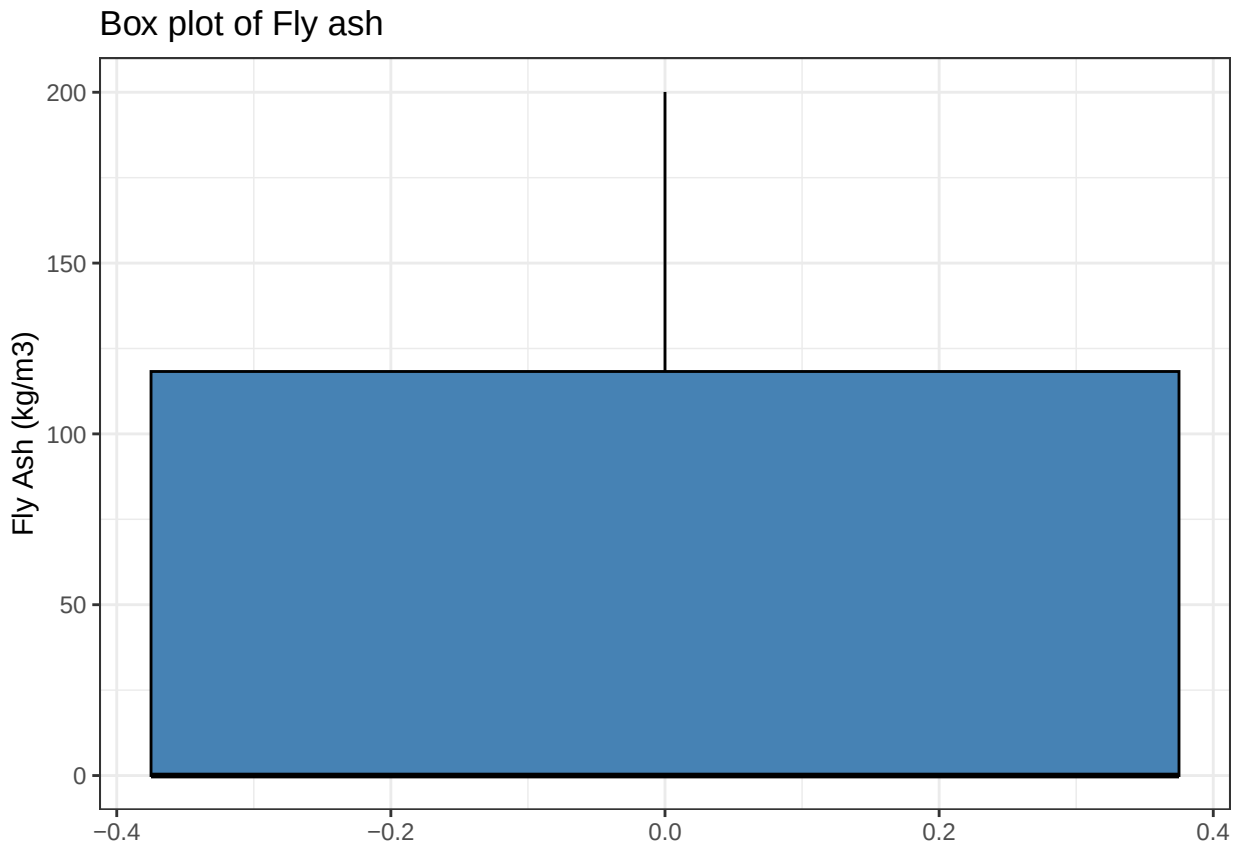
```
df %>%
  ggplot(aes(y=cement))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of cement', y = "Cement (kg/m3)")+
  theme_bw()
```



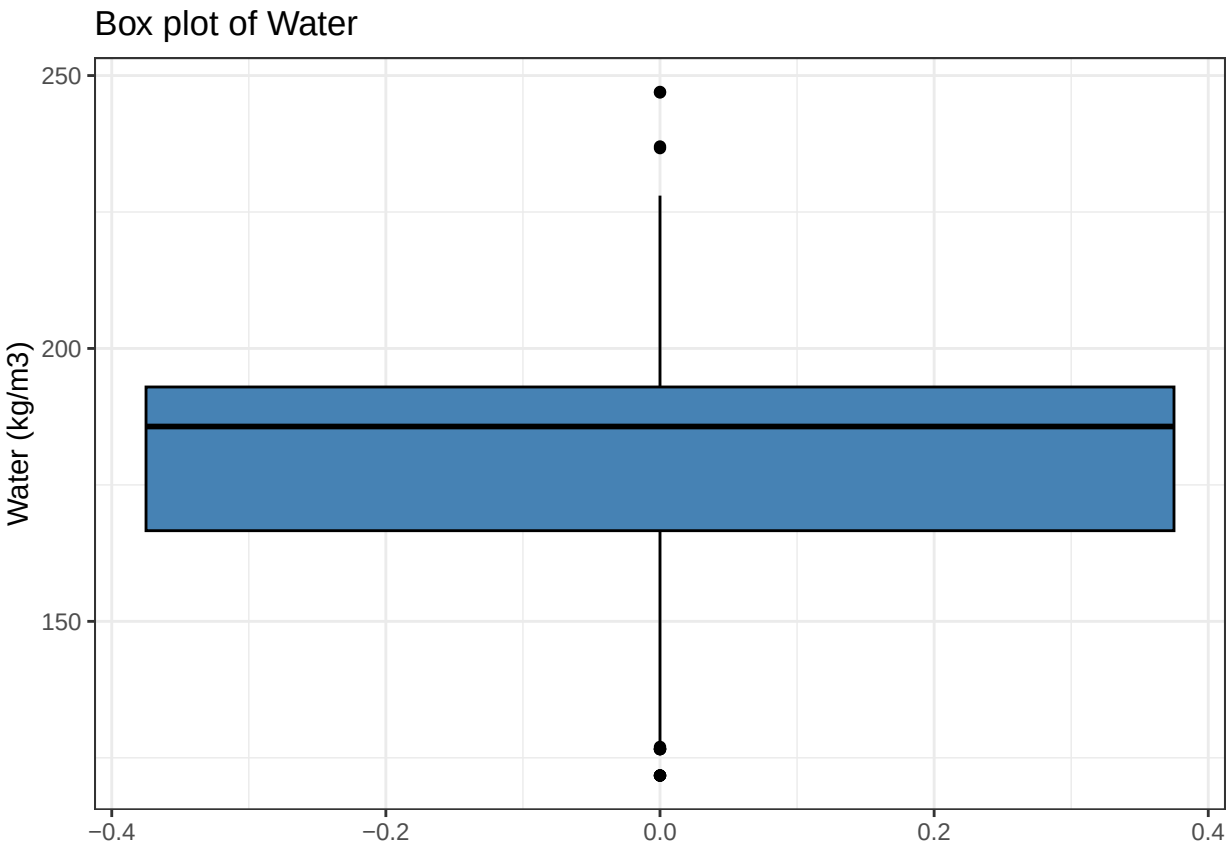

```
df %>%
  ggplot(aes(y=blast_furnace_slag))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of Blast Furnace Slag', y = "Blast Furnace Slag (kg/m3)")+
  theme_bw()
```



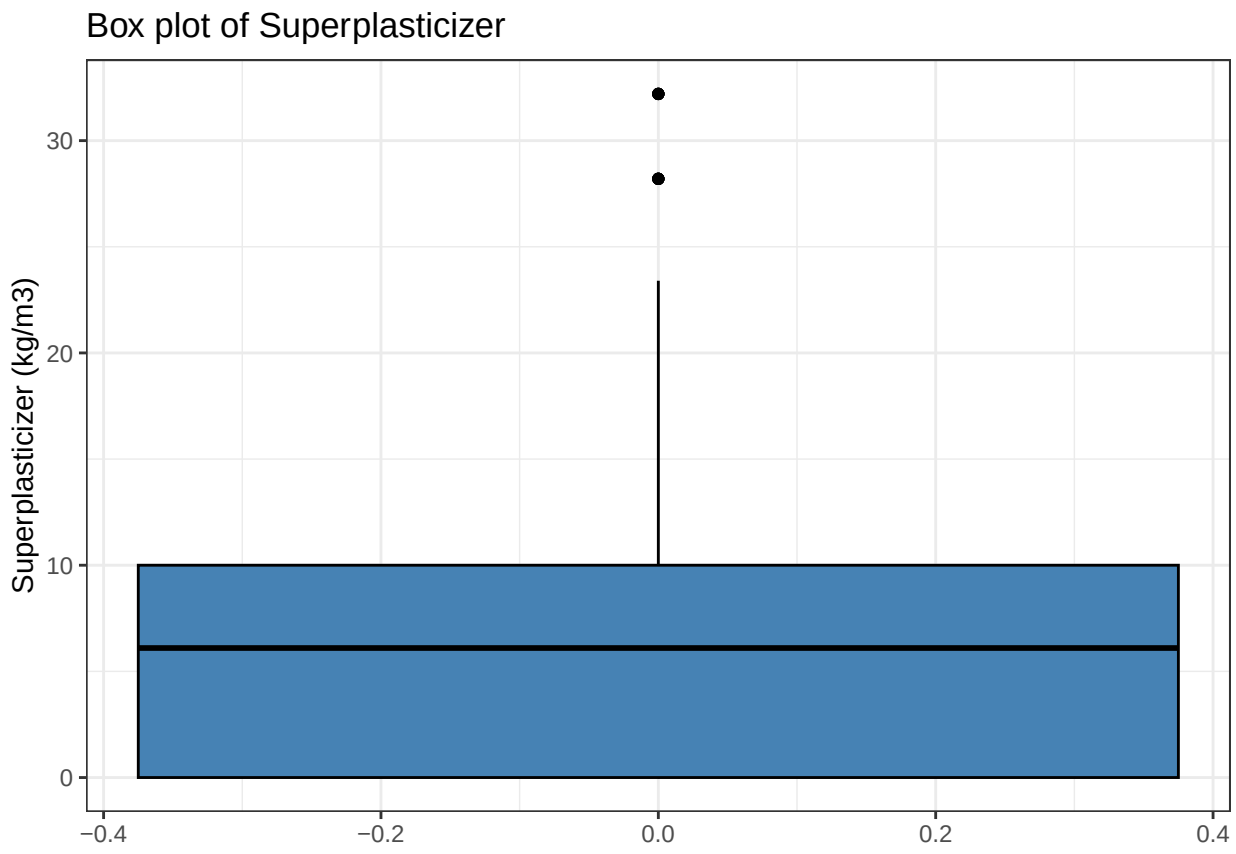

```
df %>%
  ggplot(aes(y=fly_ash))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of Fly ash', y = "Fly Ash (kg/m3)")+
  theme_bw()
```



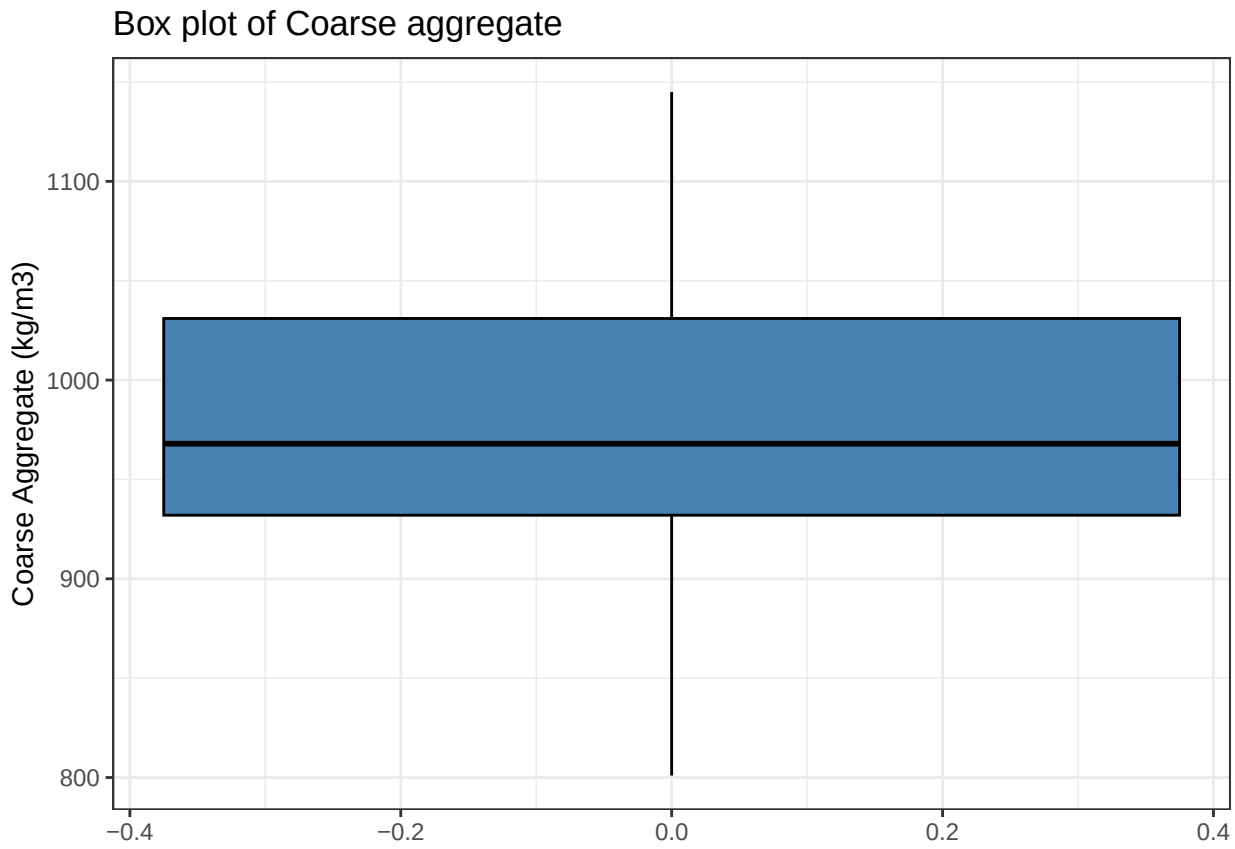

```
df %>%
  ggplot(aes(y=water))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of Water', y = "Water (kg/m3)")+
  theme_bw()
```



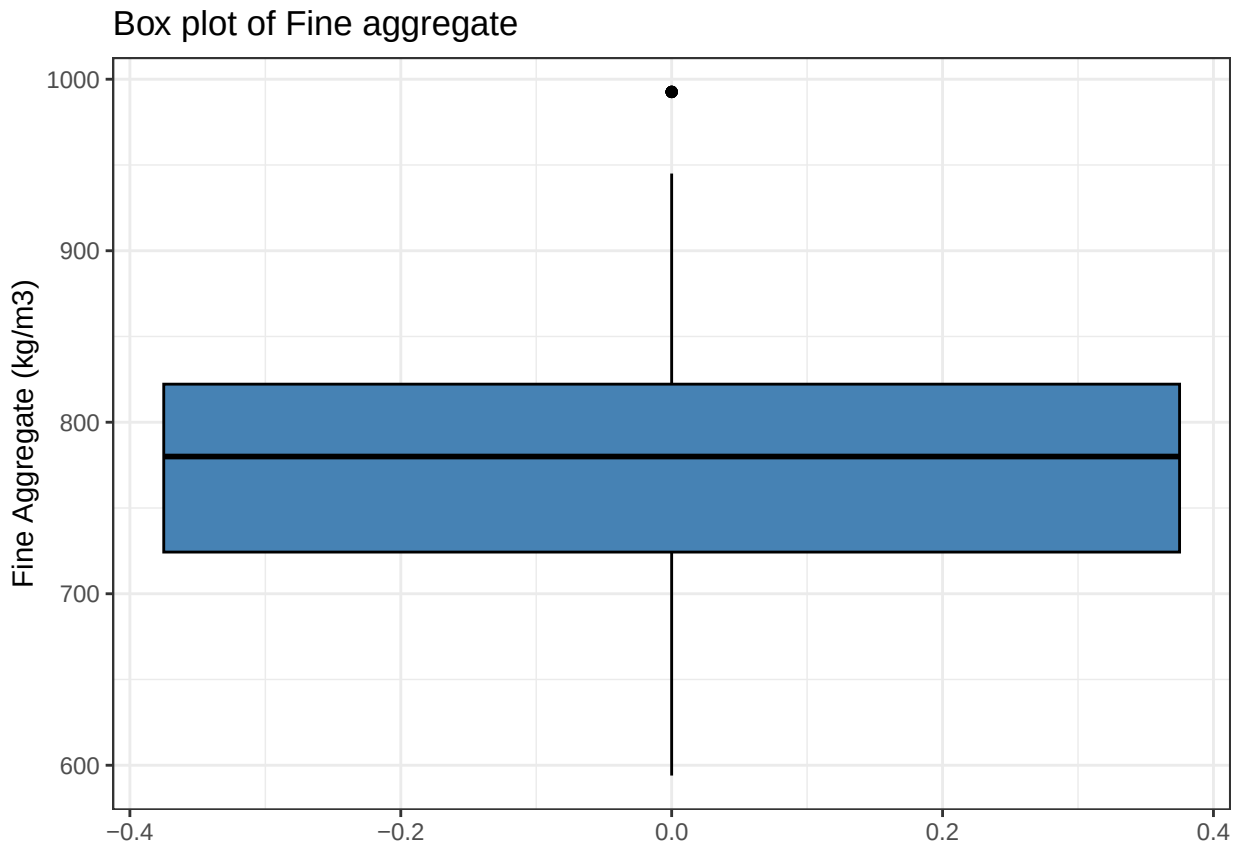

```
df %>%
  ggplot(aes(y=superplasticizer))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of Superplasticizer', y = "Superplasticizer (kg/m3)")+
  theme_bw()
```



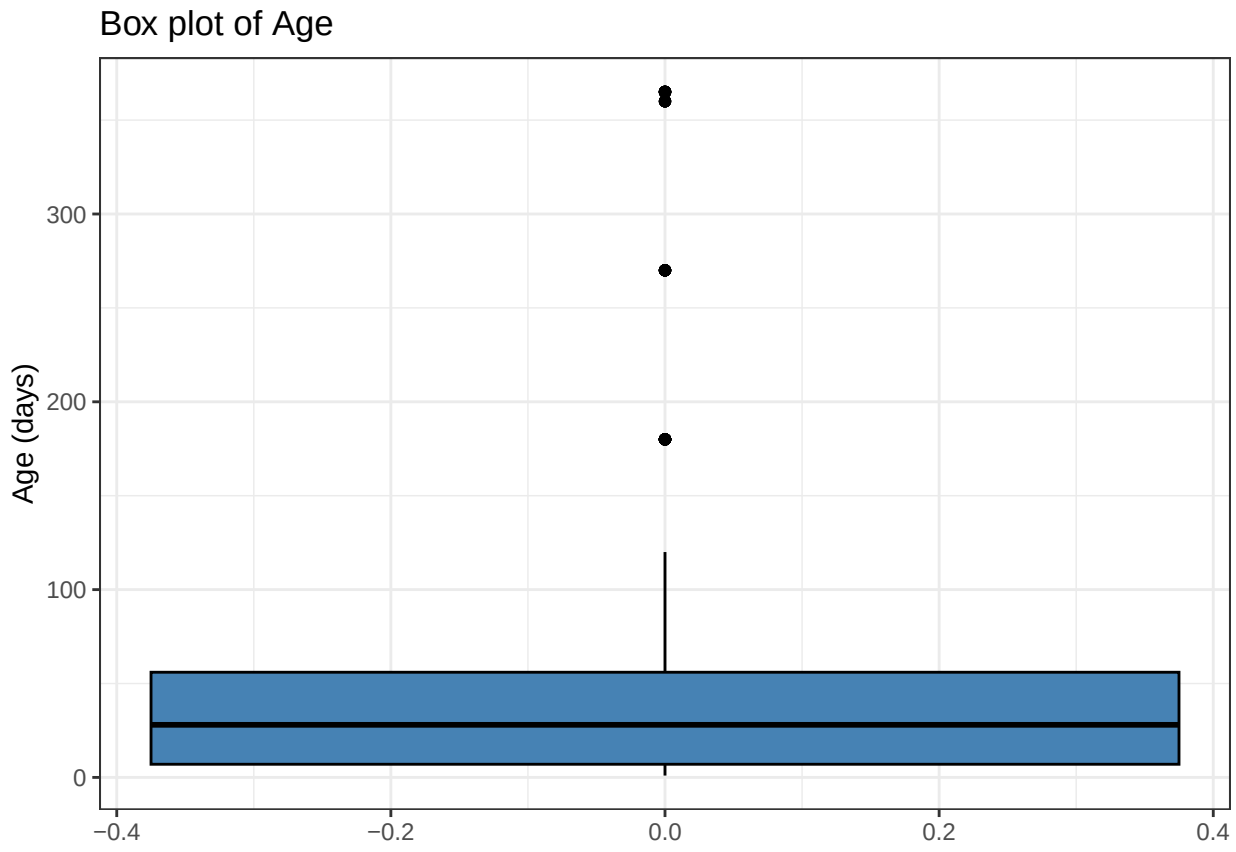

```
df %>%
  ggplot(aes(y=coarse_aggregate))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of Coarse aggregate', y = "Coarse Aggregate (kg/m3)")+
  theme_bw()
```



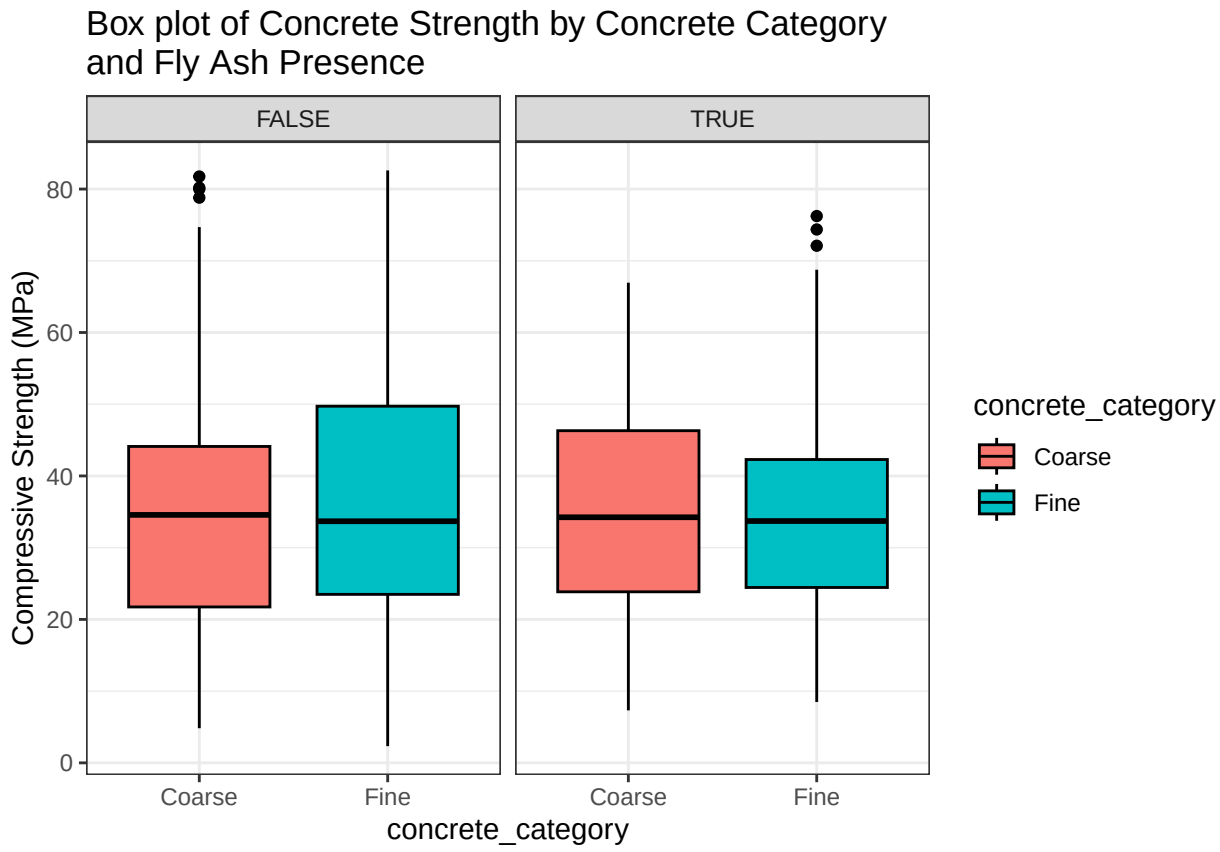

```
df %>%
  ggplot(aes(y=fine_aggregate))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of Fine aggregate', y = "Fine Aggregate (kg/m3)")+
  theme_bw()
```




```
df %>%
  ggplot(aes(y=age))+
  geom_boxplot(fill='steelblue', color='black')+
  labs(title = 'Box plot of Age', y = "Age (days)")+
  theme_bw()
```

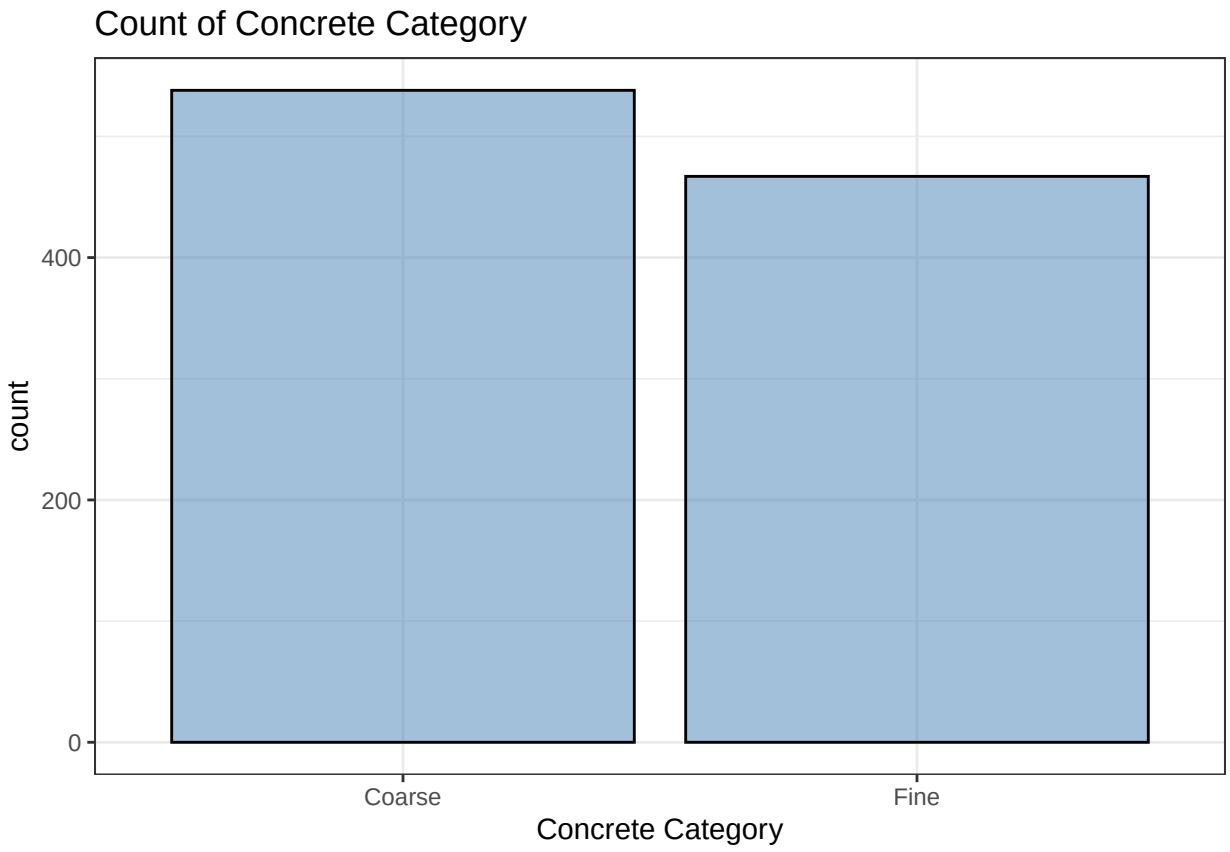



```
df %>%
  ggplot(aes(y=strength, x=concrete_category, fill=concrete_category))+
  geom_boxplot(color='black')+
  labs(title = 'Box plot of Concrete Strength by Concrete Category \nand Fly Ash Presence',
       y='Compressive Strength (MPa)')+
  facet_grid(~contains_fly_ash)+
  theme_bw()
```

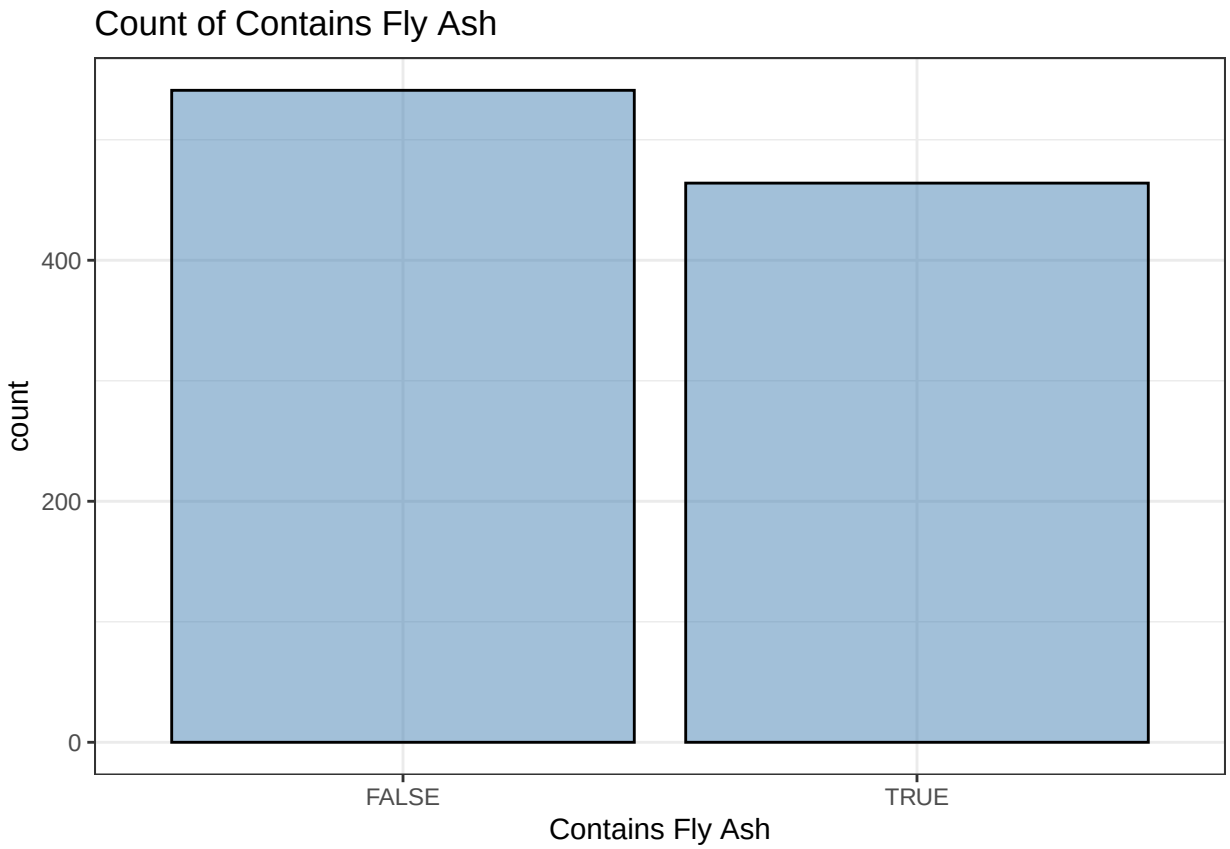


B - Categorical Variables

```
df %>%
  ggplot(aes(x=concrete_category))+
  geom_bar(fill = 'steelblue', alpha = 0.5, color = 'black')+
  labs(title = "Count of Concrete Category", x = 'Concrete Category')+
  theme_bw()
```

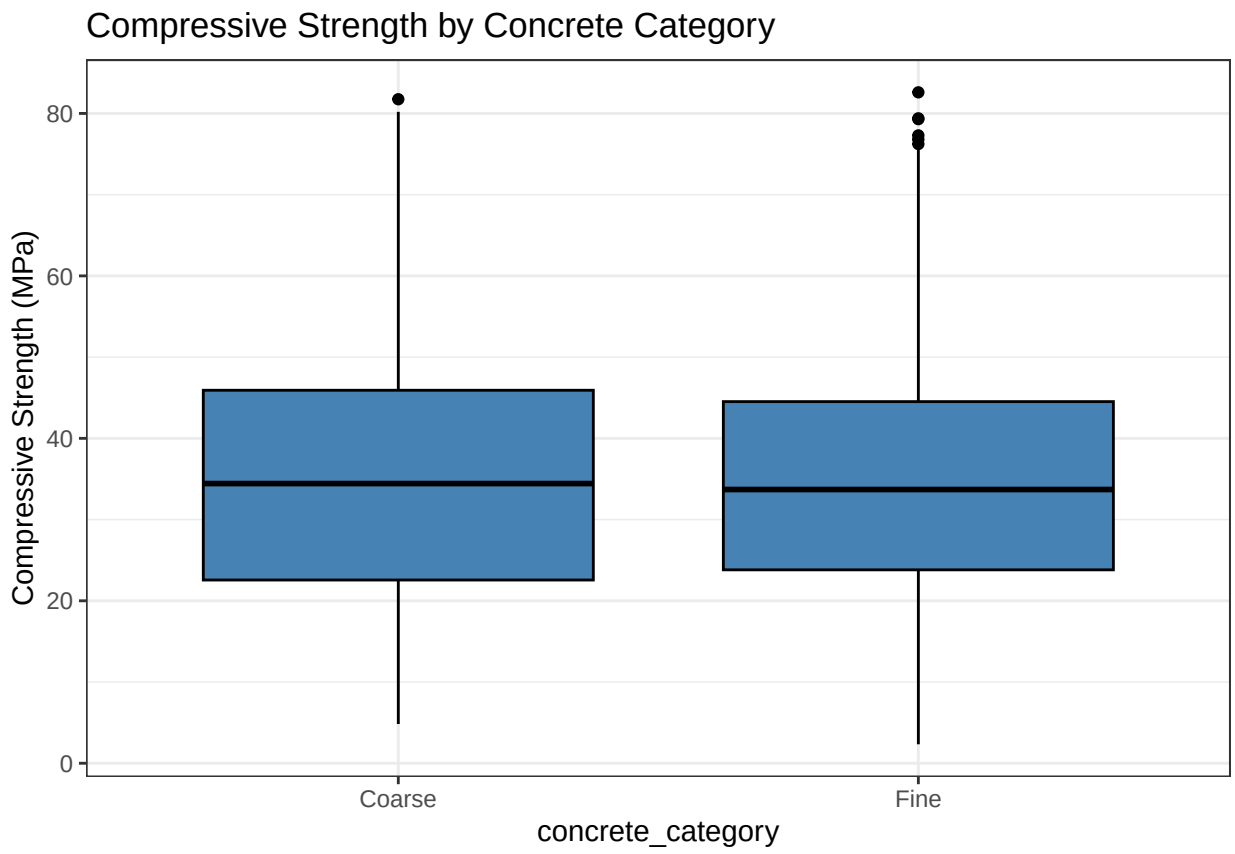



```
df %>%
  ggplot(aes(x=contains_fly_ash))+
  geom_bar(fill = 'steelblue', alpha = 0.5, color = 'black')+
  labs(title = "Count of Contains Fly Ash", x = 'Contains Fly Ash')+
  theme_bw()
```

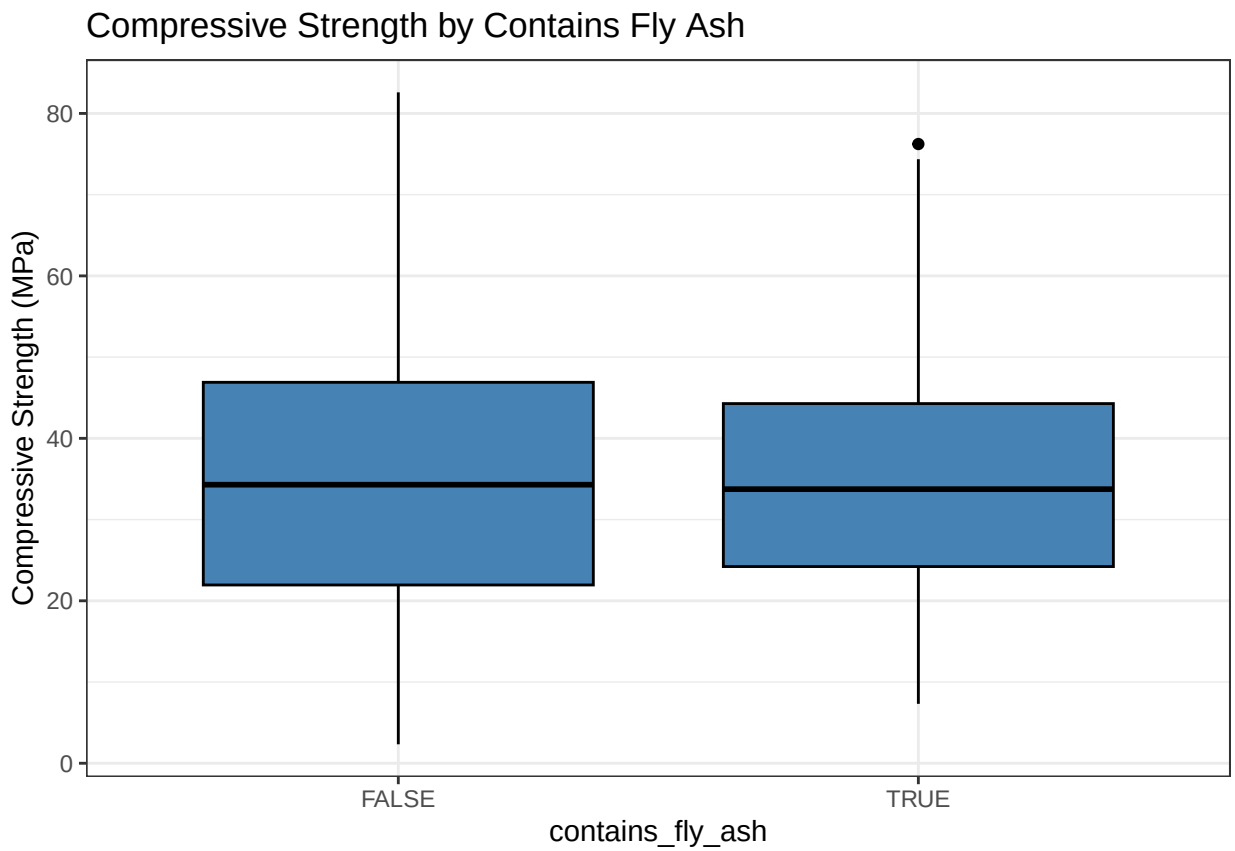


BOX PLOT

```
df %>%
  ggplot(aes(x=concrete_category, y=strength))+
  geom_boxplot(color='black', fill='steelblue')+
  labs(title='Compressive Strength by Concrete Category', y='Compressive Strength (MPa)')+
  theme_bw()
```




```
df %>%
  ggplot(aes(x=contains_fly_ash, y=strength))+
  geom_boxplot(color='black', fill='steelblue')+
  labs(title='Compressive Strength by Contains Fly Ash', y='Compressive Strength (MPa)')+
  theme_bw()
```



CORRELATION ANALYSIS

A. Correlation Matrix for Continuous Variables

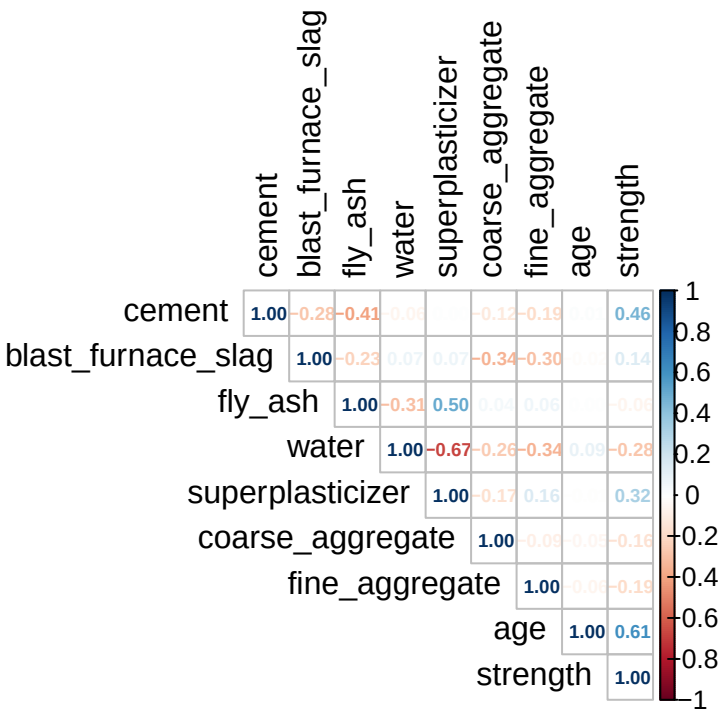
```
# Select all the continuous variables
continous_vars <- df %>%
  dplyr::select(cement, blast_furnace_slag, fly_ash, water, superplasticizer,
         coarse_aggregate, fine_aggregate, age, strength)

head(continous_vars)
```

```
##   cement blast_furnace_slag fly_ash water superplasticizer coarse_aggregate
## 1  540.0                0.0       0   162              2.5           1040.0
## 2  540.0                0.0       0   162              2.5           1055.0
## 3  332.5              142.5       0   228              0.0            932.0
## 4  332.5              142.5       0   228              0.0            932.0
## 5  198.6              132.4       0   192              0.0            978.4
## 6  266.0              114.0       0   228              0.0            932.0
##   fine_aggregate age strength
## 1          676.0  28 79.98611
## 2          676.0  28 61.88737
## 3          594.0 270 40.26954
## 4          594.0 365 41.05278
## 5          825.5 360 44.29608
## 6          670.0  90 47.02985
```

```r
# get the correlation matrix using spearman, and round to 3 decimal places
corr_matrix <- round(cor(continous_vars, method = 'spearman'), digits = 2)

corr_matrix
```

```
##                    cement blast_furnace_slag fly_ash water superplasticizer
## cement               1.00              -0.28   -0.41 -0.06             0.00
## blast_furnace_slag  -0.28               1.00   -0.23  0.07             0.07
## fly_ash             -0.41              -0.23    1.00 -0.31             0.50
## water               -0.06               0.07   -0.31  1.00            -0.67
## superplasticizer     0.00               0.07    0.50 -0.67             1.00
## coarse_aggregate    -0.12              -0.34    0.04 -0.26            -0.17
## fine_aggregate      -0.19              -0.30    0.06 -0.34             0.16
## age                  0.01              -0.02    0.00  0.09            -0.01
## strength             0.46               0.14   -0.06 -0.28             0.32
##                    coarse_aggregate fine_aggregate   age strength
## cement                        -0.12          -0.19  0.01     0.46
## blast_furnace_slag            -0.34          -0.30 -0.02     0.14
## fly_ash                        0.04           0.06  0.00    -0.06
## water                         -0.26          -0.34  0.09    -0.28
## superplasticizer              -0.17           0.16 -0.01     0.32
## coarse_aggregate               1.00          -0.09 -0.05    -0.16
## fine_aggregate                -0.09           1.00 -0.06    -0.19
## age                           -0.05          -0.06  1.00     0.61
## strength                      -0.16          -0.19  0.61     1.00
```

```r
# Visualize it using heatmap
corrplot(corr_matrix, method = "number", type = "upper", tl.col = "black",
         title = ("Spearman Correlation for Continuos Variables"),
         # mar to adjust the margins, cex to reduce the text size
         mar = c(1, 0, 3, 0), number.cex = 0.6)
```



Age and Strength (0.61): This moderate positive correlation suggests that as the concrete ages, its compressive strength generally increases.

Cement and Strength (0.46): This moderate positive correlation indicates that higher cement content is associated with greater compressive strength.

Water and Superplasticizer (-0.67): This strong negative correlation suggests that as water content increases, superplasticizer content decreases, and vice versa.

Fly Ash and Cement (-0.41): This moderate negative correlation indicates that mixes with higher fly ash content tend to use less cement.

B - Correlation between the 2 Categorical Variables (Contains Fly Ash & Concrete Category) Both are Nominal, so we use Cramer V

```
cramerV(df$concrete_category, df$contains_fly_ash)
```

```
## Cramer V
##  0.02157
```

Concrete Category and Contains Fly Ash (Cramer's V = 0.02157): This very low Cramer's V value implies virtually no association between concrete category (Fine vs. Coarse) and the presence of fly ash. This suggests that fly ash usage does not vary between Fine and Coarse concrete mixes.

C - Correlation Between Continuous Variables and Binary Categorical Variable Use Point Biseral correlation

```
# create new columns to change the datatypes of the categorical variables to numeric (0 and 1)
df_copy <- df %>%
```

```
  mutate(contains_fly_ash_numeric = as.numeric(contains_fly_ash),
         concrete_category = as.factor(concrete_category),  # convert it to factor first
         concrete_category_numeric = as.numeric(concrete_category) - 1) # minus 1 to make it 0 and 1

head(df_copy)
```

```
##   cement blast_furnace_slag fly_ash water superplasticizer coarse_aggregate
## 1  540.0               0.0       0   162              2.5           1040.0
## 2  540.0               0.0       0   162              2.5           1055.0
## 3  332.5             142.5       0   228              0.0            932.0
## 4  332.5             142.5       0   228              0.0            932.0
## 5  198.6             132.4       0   192              0.0            978.4
## 6  266.0             114.0       0   228              0.0            932.0
##   fine_aggregate age concrete_category contains_fly_ash strength
## 1          676.0  28            Coarse            FALSE 79.98611
## 2          676.0  28            Coarse            FALSE 61.88737
## 3          594.0 270            Coarse            FALSE 40.26954
## 4          594.0 365            Coarse            FALSE 41.05278
## 5          825.5 360              Fine            FALSE 44.29608
## 6          670.0  90            Coarse            FALSE 47.02985
##   contains_fly_ash_numeric concrete_category_numeric
## 1                        0                         0
## 2                        0                         0
## 3                        0                         0
## 4                        0                         0
## 5                        0                         1
## 6                        0                         0
```

```
# Perform point biserial correlation on cement and concrete_category
cor.test(df$cement, df_copy$concrete_category_numeric)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$cement and df_copy$concrete_category_numeric
## t = -0.059836, df = 1003, p-value = 0.9523
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.06372052  0.05995626
## sample estimates:
##          cor
## -0.001889355
```

Cement and Concrete Category (-0.001889355): This near-zero correlation indicates that cement content does not differ meaningfully between Fine and Coarse concrete categories. Both categories likely use a similar range of cement proportions.

```
# Perform point biserial correlation on cement and contains_fly_ash
cor.test(df$cement, df_copy$contains_fly_ash_numeric)
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  df$cement and df_copy$contains_fly_ash_numeric
## t = -11.205, df = 1003, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3873910 -0.2774262
## sample estimates:
##        cor
## -0.3335427
```

Cement and Contains Fly Ash (-0.3335427): This moderate negative correlation suggests that mixes containing fly ash tend to have lower cement content.

```
# Perform point biserial correlation on strength and contains_fly_ash
cor.test(df$strength, df_copy$contains_fly_ash_numeric)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$strength and df_copy$contains_fly_ash_numeric
## t = -1.0745, df = 1003, p-value = 0.2829
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09554674  0.02798882
## sample estimates:
##         cor
## -0.03390848
```

Strength and Contains Fly Ash (-0.03390848): This near-zero negative correlation suggests a minimal relationship between fly ash presence and compressive strength.

```
# Perform point biseral correlation on strength and concrete_category
cor.test(df$strength, df_copy$concrete_category_numeric)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$strength and df_copy$concrete_category_numeric
## t = 0.41107, df = 1003, p-value = 0.6811
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.04889938  0.07475709
## sample estimates:
##        cor
## 0.01297848
```

Strength and Concrete Category (0.01297848): This very low positive correlation indicates that compressive strength does not really vary by concrete category (Fine vs. Coarse).

REGRESSION

MULTIPLE LINEAR REGRESSION

Objective of the regression analysis: We want to examine the possible linear relation between compressive strength and more than one Independent Variable

MODEL 1 Lets start with cement, blast_furnace_slag, water, and superplasticizer

Y = strength X1 = cement X2 = blast_furnace_slag X3 = water X4 = superplasticizer

```
# MLR model with cement, blast_furnace_slag, water, and superplasticizer
model_1 <- lm(strength ~ cement + blast_furnace_slag + water + superplasticizer, continous_vars)

summary.lm(model_1)
```

```
##
## Call:
## lm(formula = strength ~ cement + blast_furnace_slag + water +
##     superplasticizer, data = continous_vars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.139  -9.693  -0.260   8.617  36.136
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        22.397972   4.936345   4.537 6.39e-06 ***
## cement              0.086402   0.003988  21.666  < 2e-16 ***
## blast_furnace_slag  0.053765   0.004907  10.957  < 2e-16 ***
## water              -0.102732   0.024744  -4.152 3.58e-05 ***
## superplasticizer    0.598461   0.088669   6.749 2.51e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.52 on 1000 degrees of freedom
## Multiple R-squared:  0.4109, Adjusted R-squared:  0.4086
## F-statistic: 174.4 on 4 and 1000 DF,  p-value: < 2.2e-16
```

All the coefficients and intercept are significant at 0.05 level

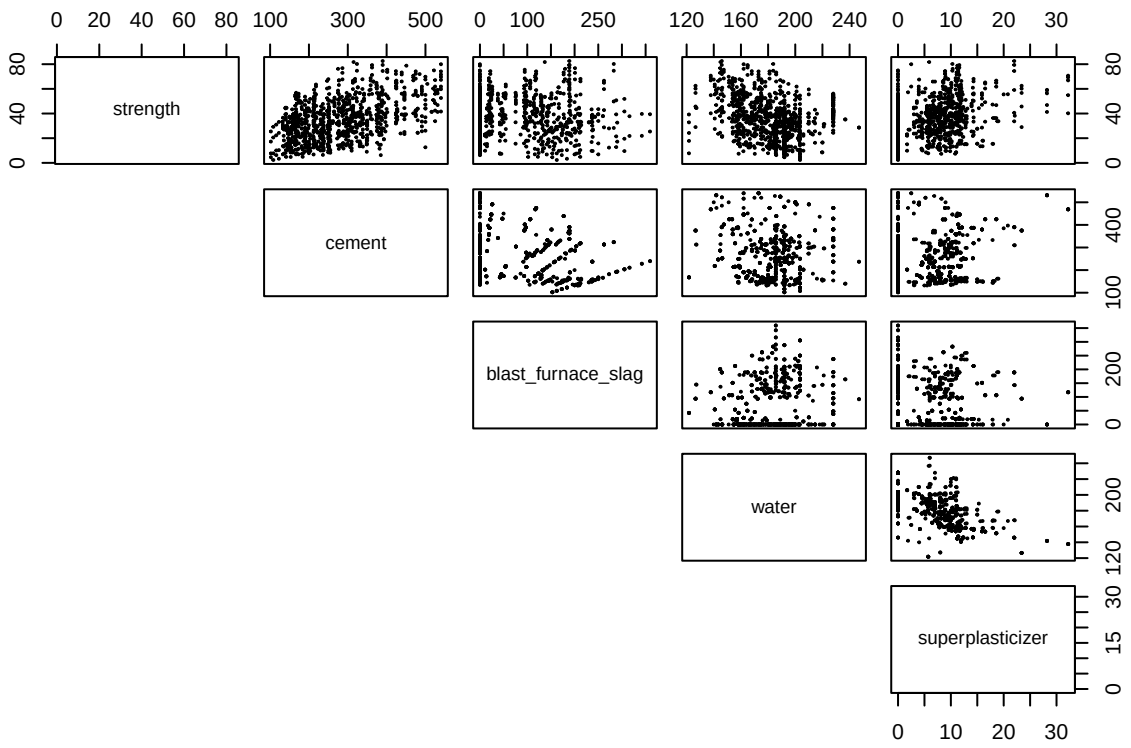
strength = 22.397972 + 0.086402 * cement + 0.053765 * blast_furnace_slag - 0.102732 * water + 0.598461 * superplasticizer

Adjusted R-squared = 0.4086, means that they can predict 40.9% of the total variability of compressive strength.

Check Assumptions

Assumption 1. Linearity between Strength and each variable

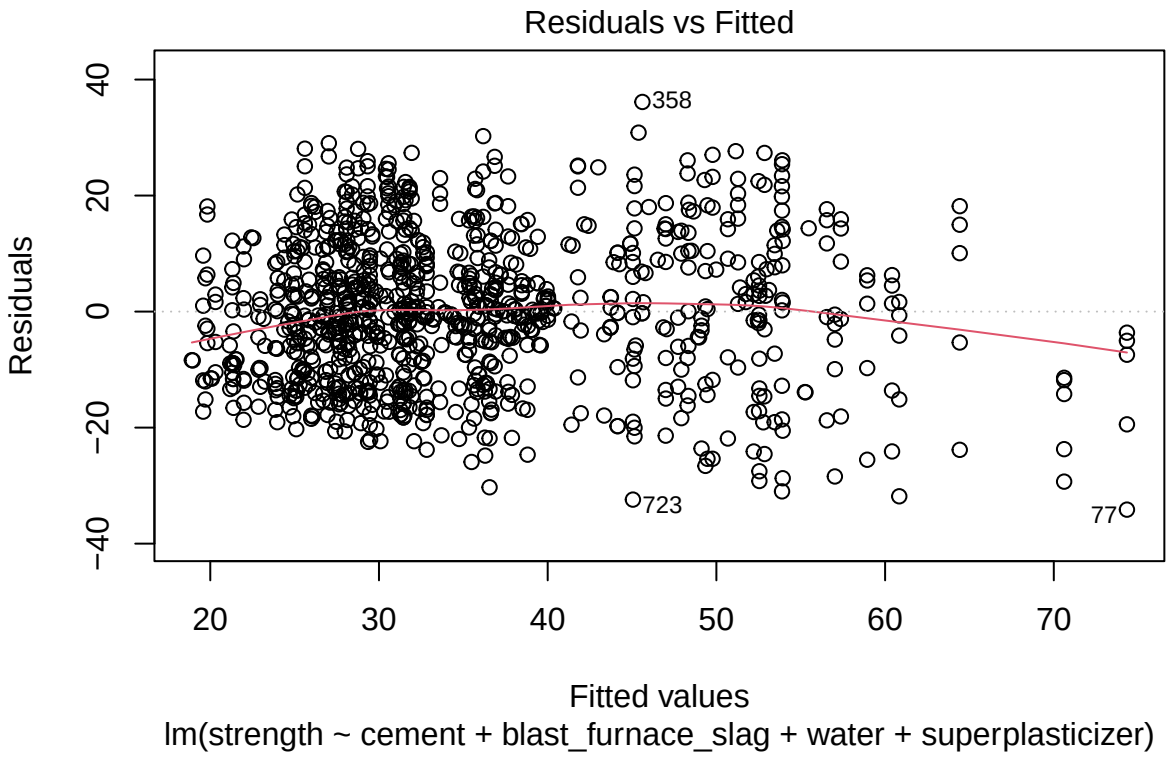
```
pairs(continous_vars[, c(9, 1, 2, 4, 5)], lower.panel = NULL, pch = 19, cex=0.2)
```



All the variables appear to have an approximately linear relationship with strength

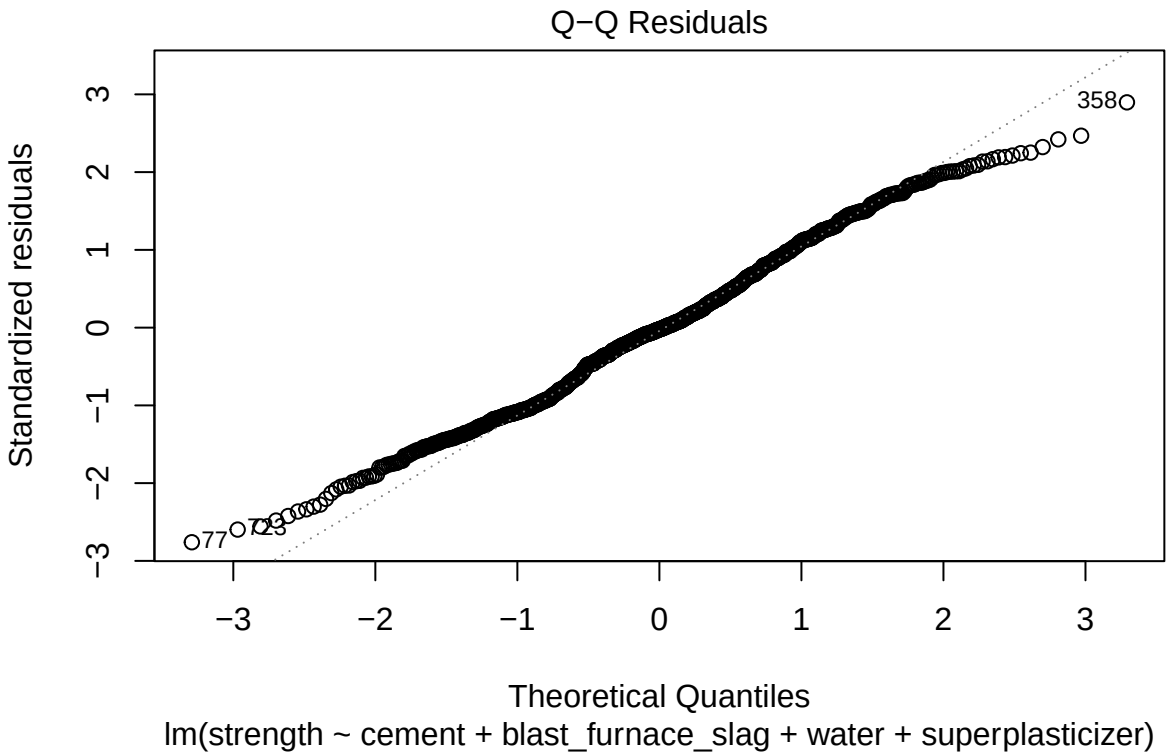
Assumption 2: Residuals' Independence

```
# plot residuals vs fitted values
plot(model_1, 1)
```



Assumption 3: Normality of Residuals

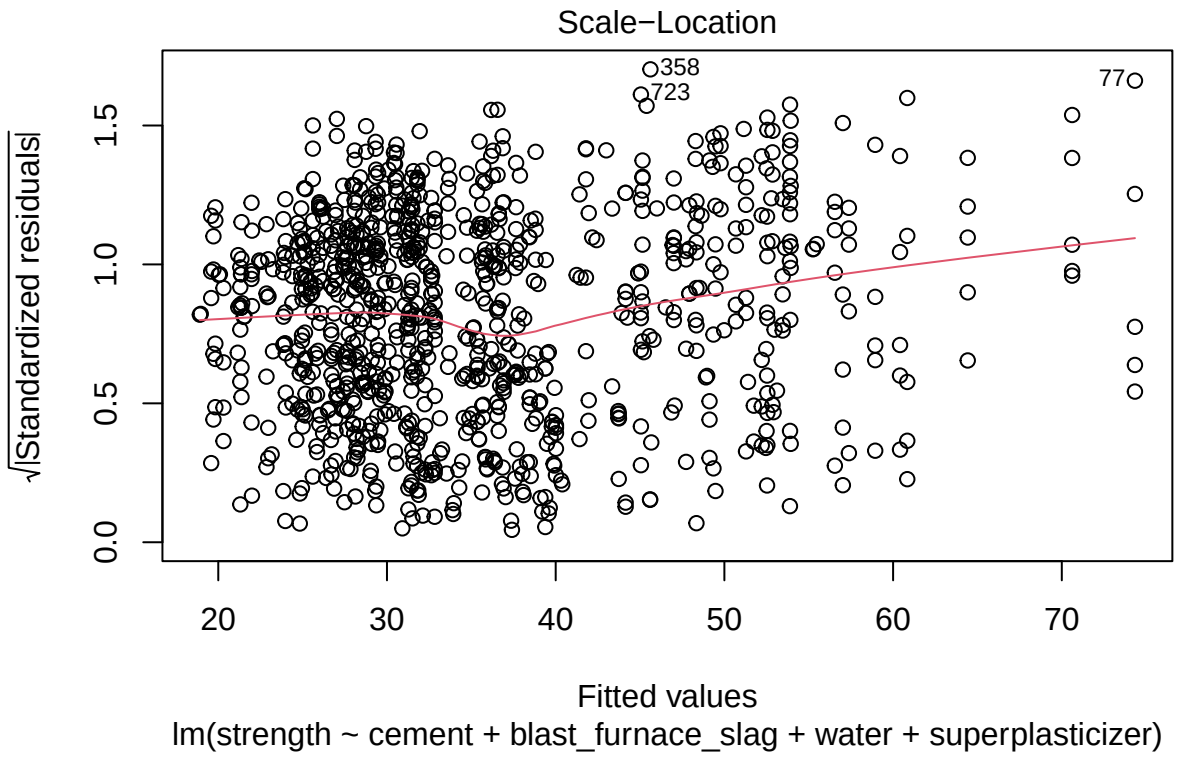
```
# plot qq plot of residuals
plot(model_1, 2)
```



The residuals are approximately normally distributed.

Assumption 4: Homescedacity (Equal Variance)

```
plot(model_1, 3)
```



Assumption 5: No Multicolinearity

```
vif(model_1)
```

```
##             cement blast_furnace_slag              water   superplasticizer
##           1.108397           1.144478           1.784889           1.763519
```

All VIFs less than 5: No multicolinearity.

All 5 assumptions were approved, so we can confirm the equation

strength = 22.397972 + 0.086402 * cement + 0.053765 * blast_furnace_slag - 0.102732 * water + 0.598461 * superplasticizer

MODEL 2: Cement, blast furnace slag, water, fly_ash

Y = strength X1 = cement X2 = blast_furnace_slag X3 = water X4 = fly_ash

```
# MLR model with cement, blast_furnace_slag, water, and fly ash
model_2 <- lm(strength ~ cement + blast_furnace_slag + water + fly_ash, continous_vars)

summary.lm(model_2)
```

```
##
## Call:
## lm(formula = strength ~ cement + blast_furnace_slag + water +
##     fly_ash, data = continous_vars)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.655  -8.861  -0.210   8.574  35.044
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        23.841431   4.352206   5.478 5.44e-08 ***
## cement              0.110835   0.004714  23.510  < 2e-16 ***
## blast_furnace_slag  0.081224   0.005452  14.898  < 2e-16 ***
## water              -0.159829   0.019433  -8.225 6.05e-16 ***
## fly_ash             0.067997   0.007952   8.551  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.36 on 1000 degrees of freedom
## Multiple R-squared:  0.426,  Adjusted R-squared:  0.4237
## F-statistic: 185.6 on 4 and 1000 DF,  p-value: < 2.2e-16
```

All the coefficients and intercept are significant at 0.05 level

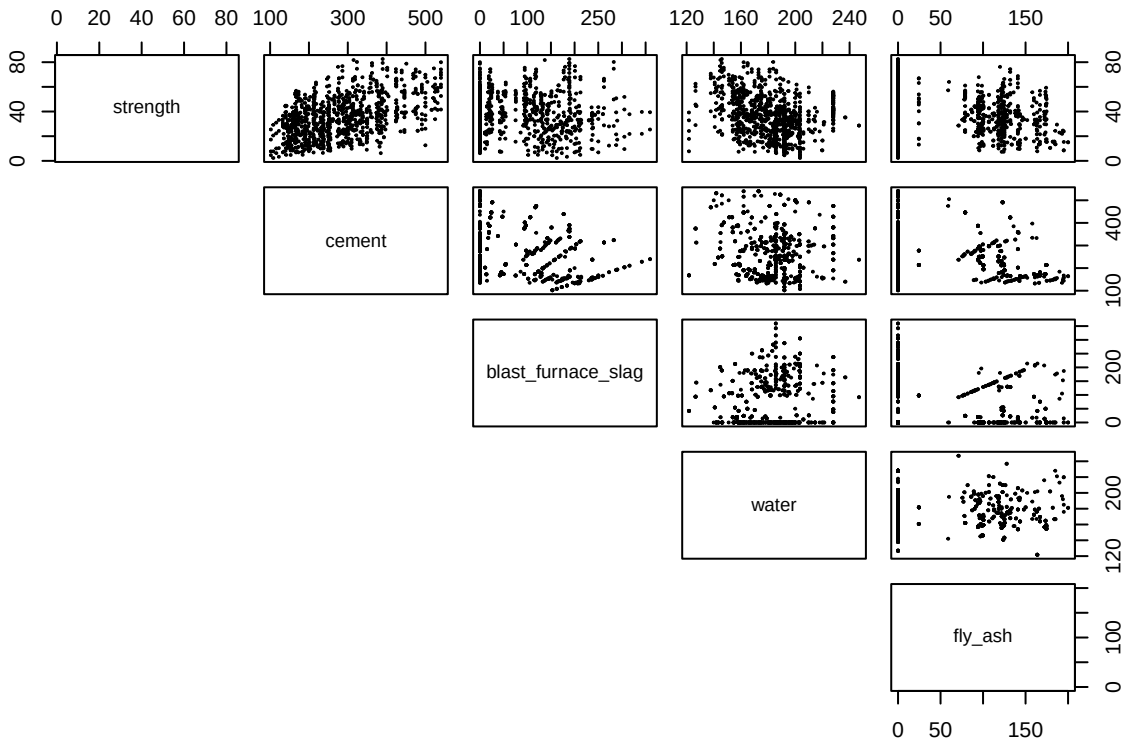
strength = 23.841431 + 0.110835 * cement + 0.081224 * blast_furnace_slag - 0.159829 * water + 0.067997 * fly_ash

Adjusted R-squared = 0.42, means that they can predict 42% of the total variability of compressive strength. -> Best model

Check Assumptions

Assumption 1. Linearity between Strength and each variable

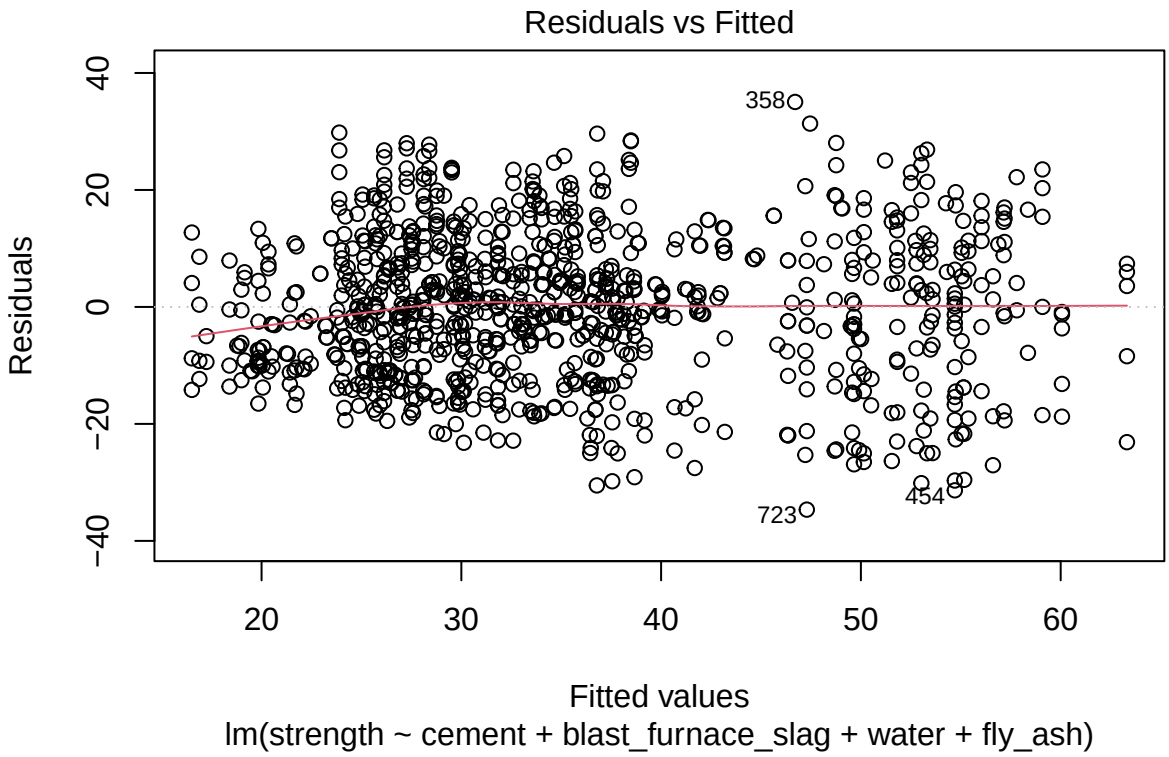
```
pairs(continous_vars[, c(9, 1, 2, 4, 3)], lower.panel = NULL, pch = 19, cex=0.2)
```



All the variables appear to have an approximately linear relationship with strength

Assumption 2: Residuals' Independence

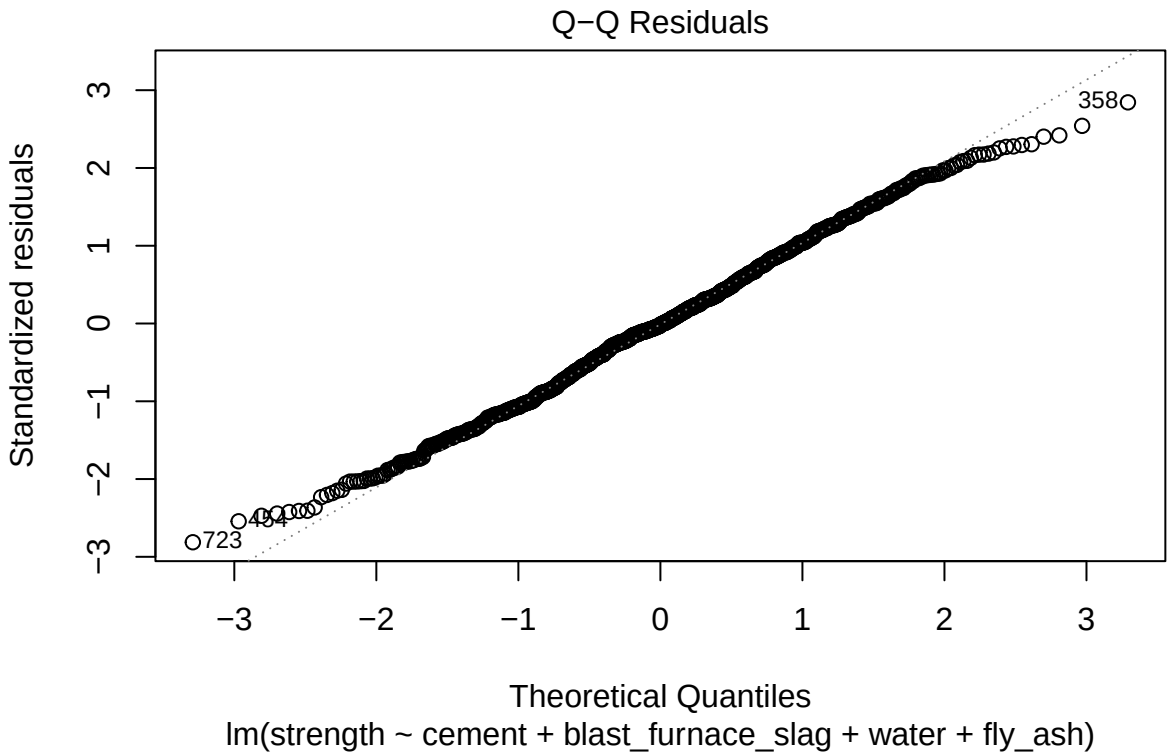
```
# plot residuals vs fitted
plot(model_2, 1)
```



The correlation is approximately 0

Assumption 3: Normality of Residuals

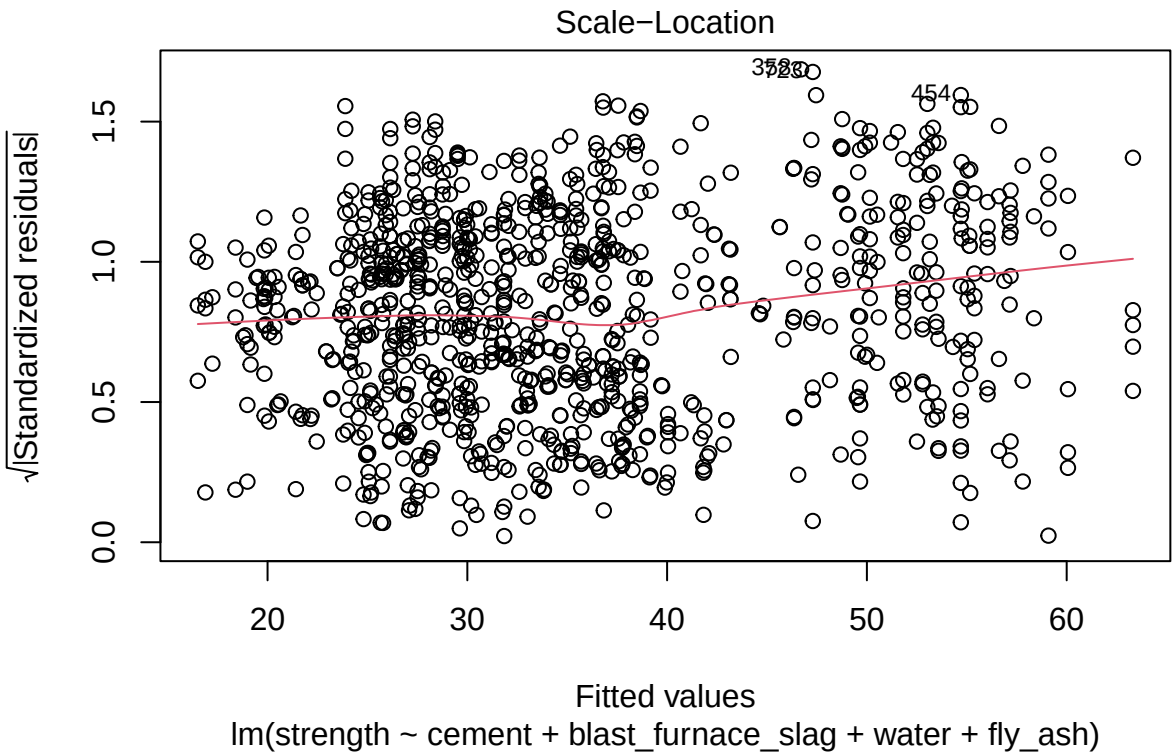
```
#plot qq plot of residuals
plot(model_2, 2)
```



The residuals are approximately normally distributed.

Assumption 4: Homescedacity (Equal Variance)

```
plot(model_2, 3)
```



The variance is constant.

Assumption 5: No Multicolinearity

```
vif(model_2)
```

```
##             cement blast_furnace_slag              water            fly_ash
##           1.589760           1.450024           1.129965           1.712742
```

All VIFs less than 5, so no multicolinearity.

ALL 5 ASSUMPTIONS WERE APPROVED, THEEFORE:

strength = 24.369098 + 0.112530 * cement + 0.084340 * blast_furnace_slag - 0.165824 * water + 0.068003 * fly_ash

MODEL 3

Y = strength X1 = cement X2 = water

```
# MLR model with cement, blast_furnace_slag, water, and fly ash
model_3 <- lm(strength ~ cement + water, continous_vars)

summary.lm(model_3)
```

```
##
## Call:
## lm(formula = strength ~ cement + water, data = continous_vars)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.134 -10.755   0.028   9.420  41.931
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 48.34263    3.94734  12.247   <2e-16 ***
## cement       0.07406    0.00414  17.889   <2e-16 ***
## water       -0.18524    0.02024  -9.151   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.67 on 1002 degrees of freedom
## Multiple R-squared:  0.2972, Adjusted R-squared:  0.2958
## F-statistic: 211.8 on 2 and 1002 DF,  p-value: < 2.2e-16
```

All the coefficients and intercept are significant at 0.05 level

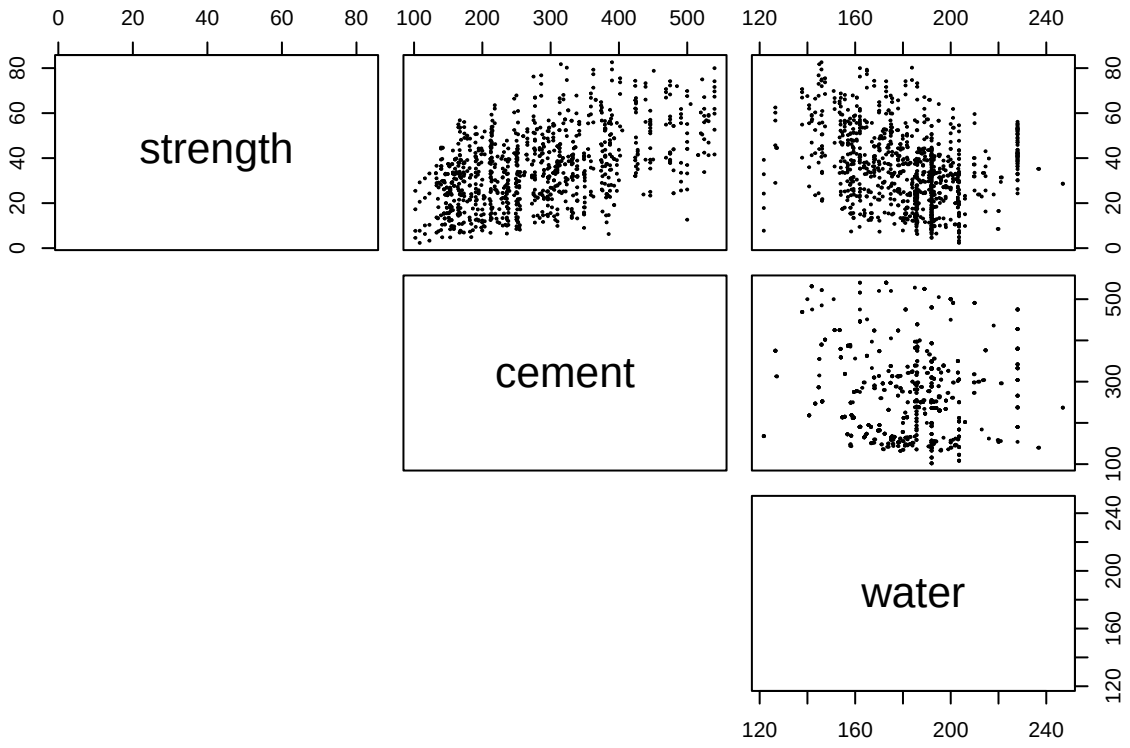
strength = 48.34263 + 0.07406 * cement - 0.18524 * water

Adjusted R-squared = 0.31, means that they can predict 31% of the total variability of compressive strength.

Check Assumptions

Assumption 1. Linearity between Strength and each variable

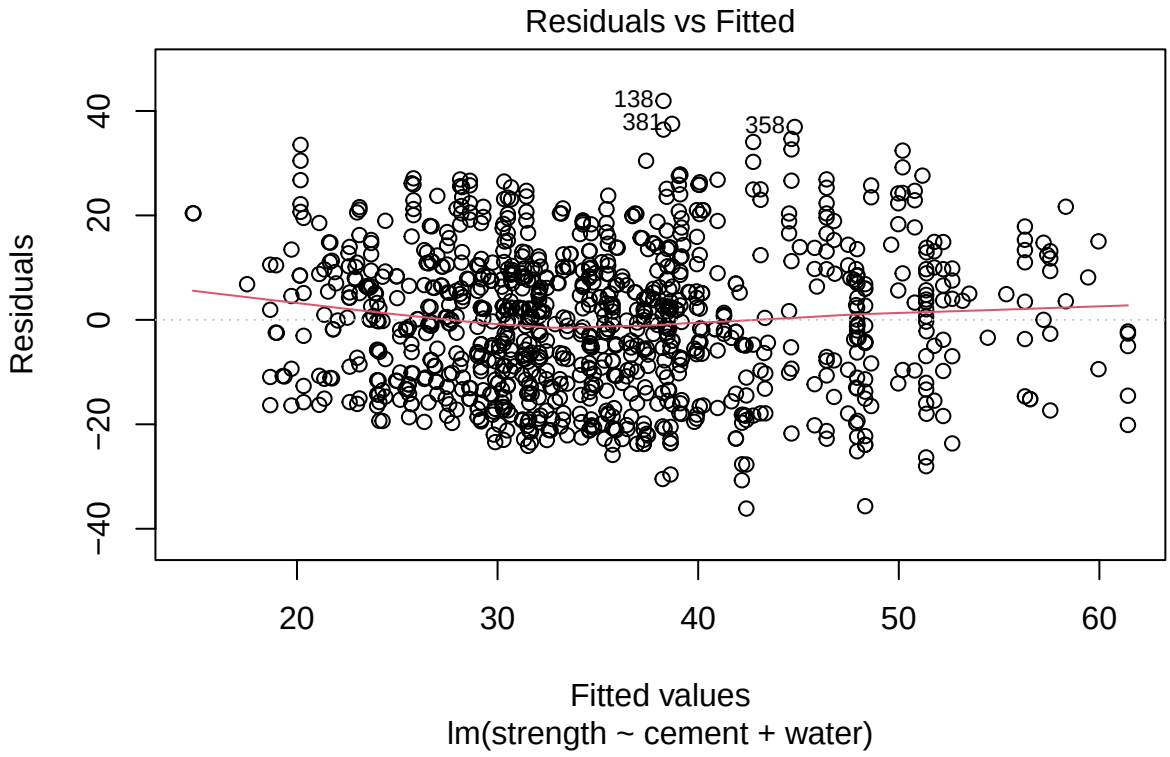
```
pairs(continous_vars[, c(9, 1, 4)], lower.panel = NULL, pch = 19, cex=0.2)
```



All the variables appear to have an approximately linear relationship with strength

Assumption 2: Residuals' Independence

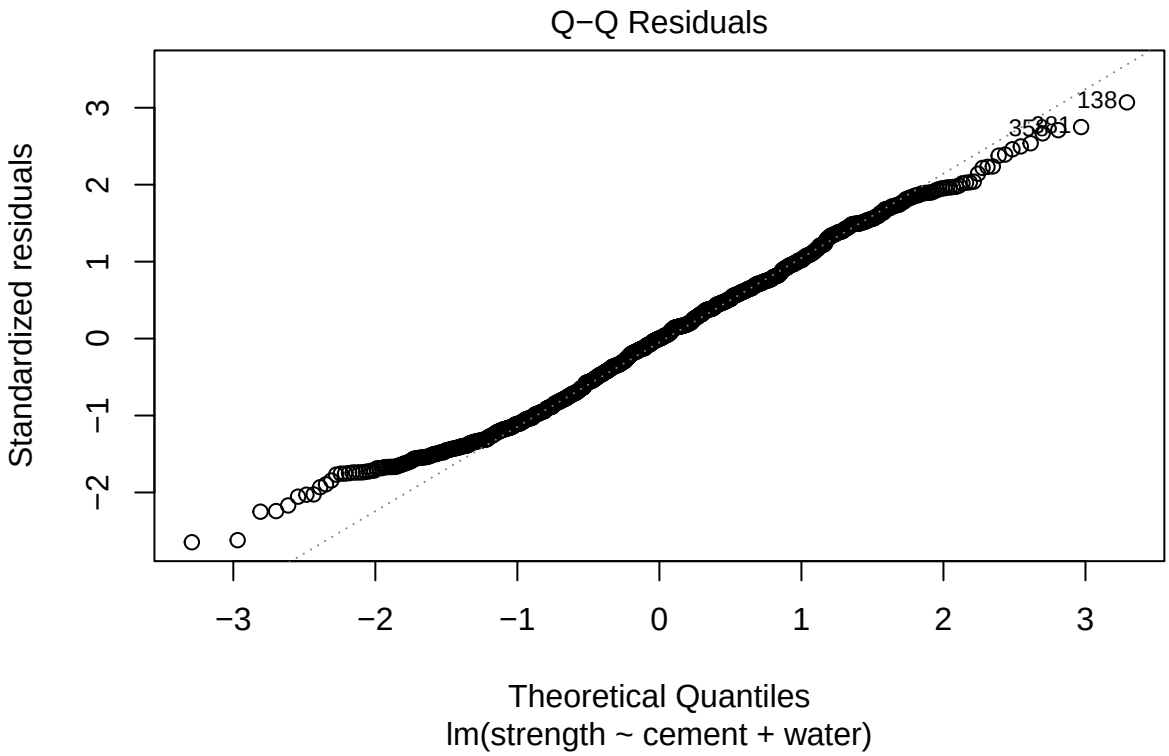
```
# plot residuals vs fitted
plot(model_3, 1)
```



The residuals are independent

Assumption 3: Normality of Residuals

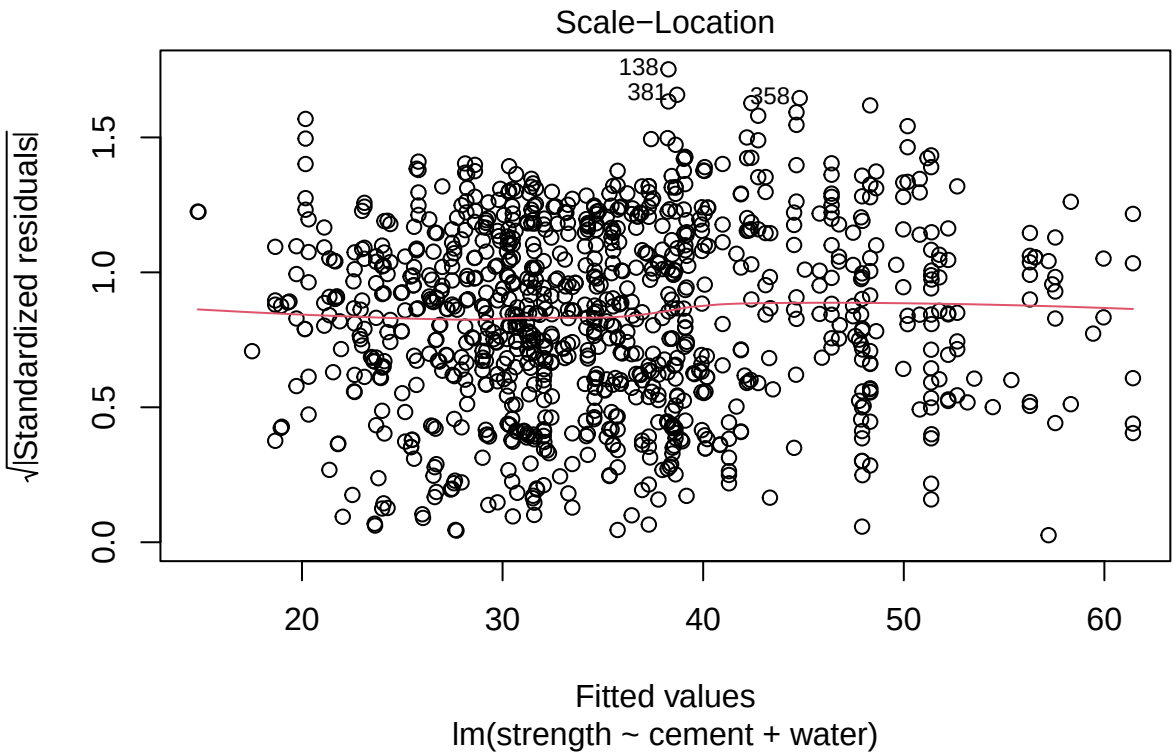
```
#plot qq plot of residuals
plot(model_3, 2)
```



The residuals are approximately normally distributed.

Assumption 4: Homescedacity (Equal Variance)

```
plot(model_3, 3)
```



The variance is constant

Assumption 5: No Multicolinearity

```
vif(model_3)
```

```
##   cement    water
## 1.003212 1.003212
```

VIF less than 5, so no multicolinearity

All 5 Assumptions were met, THEEFORE:

strength = 48.34263 + 0.07406 * cement - 0.18524 * water

HYPOTHESIS TESTING

1. Hypothesis Test 1:

alpha = 0.05

Null Hypothesis (H0): The mean compressive strength of concrete with fly ash is the same as the mean compressive strength of concrete without fly ash.

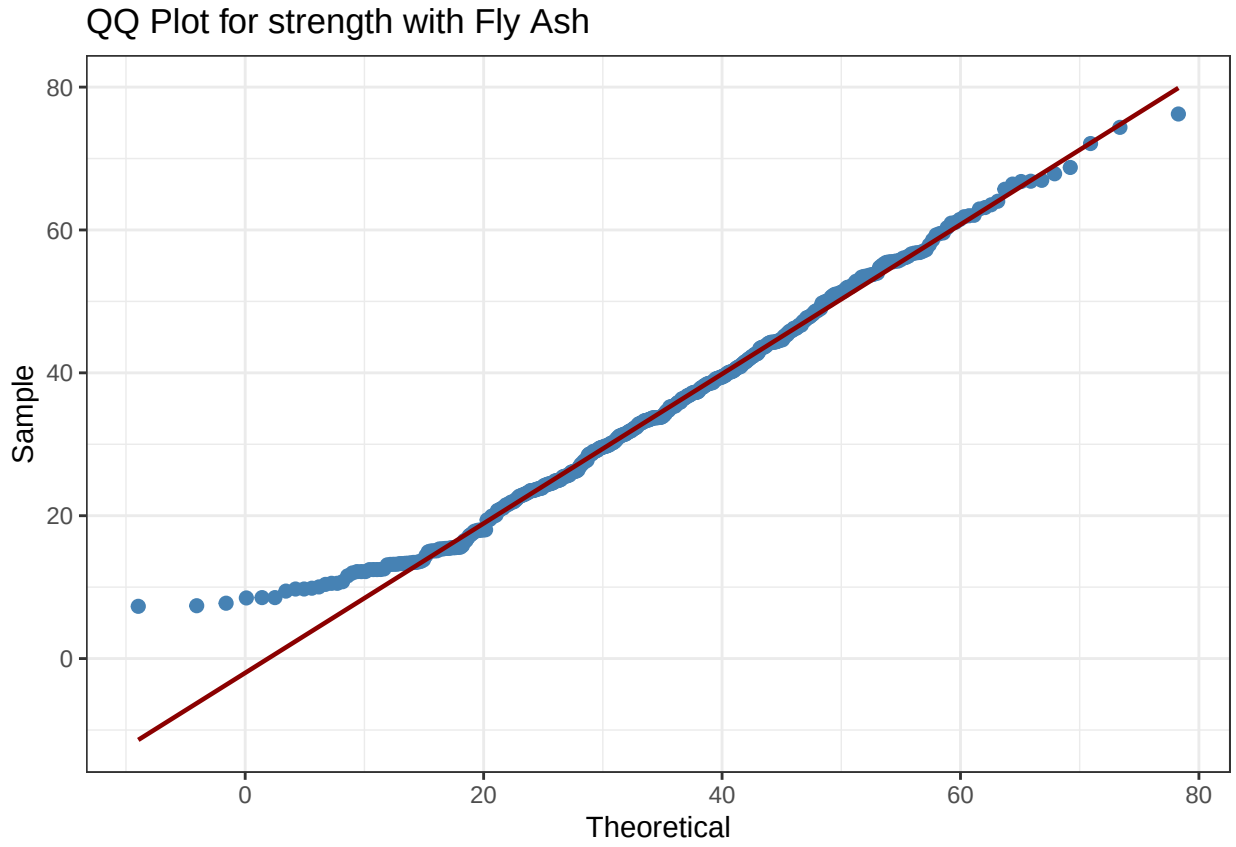
Alternative Hypothesis (H1): There is a significant difference in the mean compressive strength of concrete with and without fly ash.

This is a two tailed test.

Check assumptions:

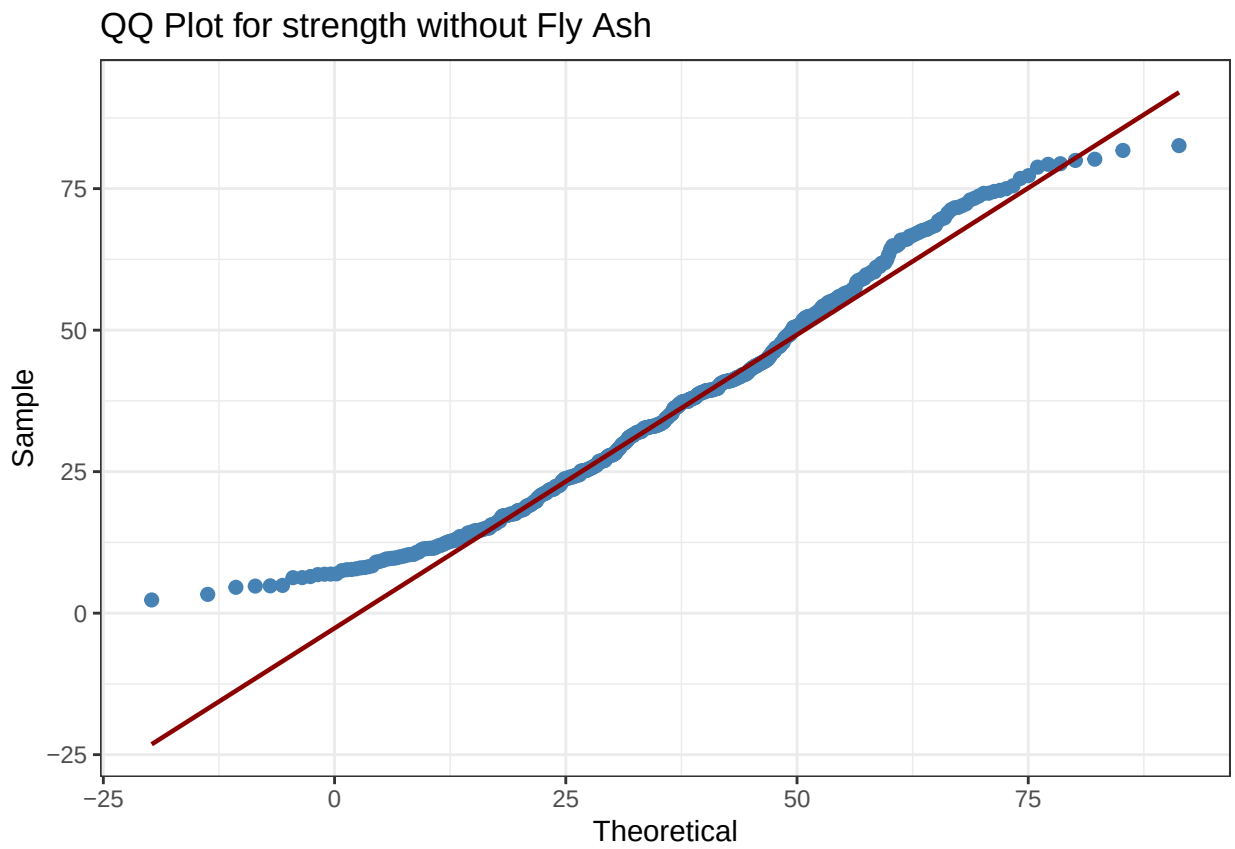
Assumption 1. check for normality

```
# Filter the samples that contain fly ash
has_fly_ash <- df %>%
  filter(contains_fly_ash == TRUE)

has_fly_ash %>%
  ggplot(aes(sample=strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for strength with Fly Ash", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



QQ plot seems like it is approximately normally distributed

```
# Filter the samples that don't contain fly ash
no_fly_ash <- df %>%
  filter(contains_fly_ash == FALSE)

no_fly_ash %>%
  ggplot(aes(sample=strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for strength without Fly Ash", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```

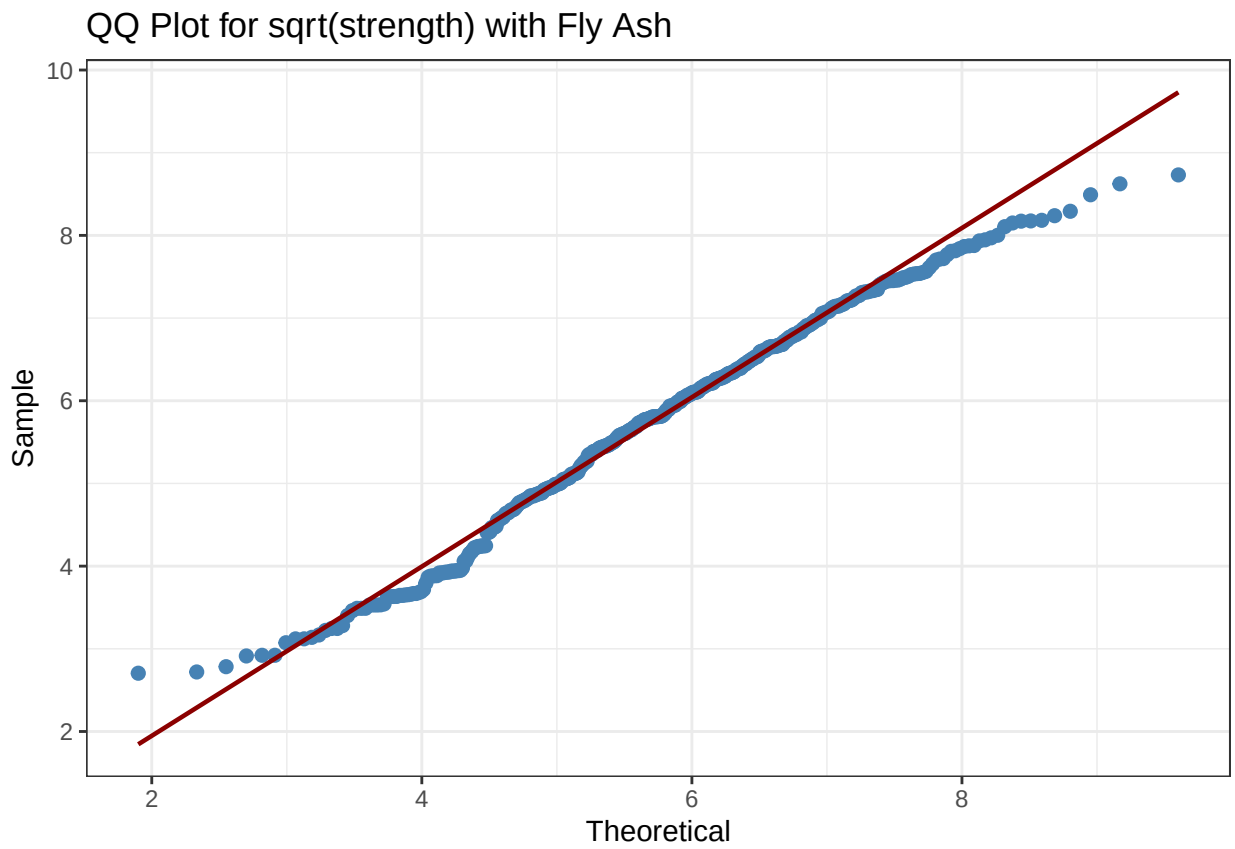


QQ plot seems like there are some deviations from normality

Since at least one of the distributions is not normal, we can transform them to get an approximately normal distribution

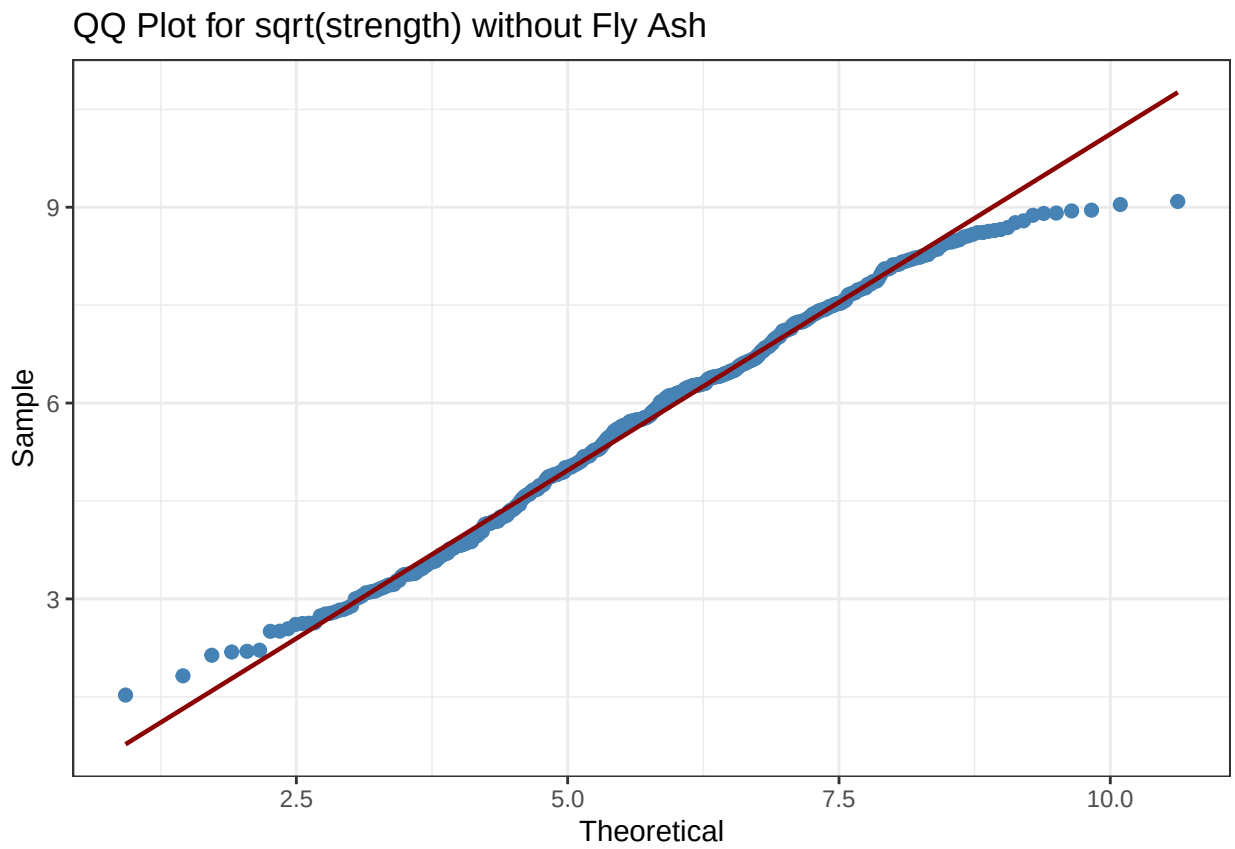
```
# Transform the distributions using sqrt

df <- df %>%
  mutate(sqrt_strength = sqrt(strength))

# Filter the samples that contain fly ash
has_fly_ash <- df %>%
  filter(contains_fly_ash == TRUE)

# Filter the samples that don't contain fly ash
no_fly_ash <- df %>%
  filter(contains_fly_ash == FALSE)
```

```
has_fly_ash %>%
  ggplot(aes(sample=sqrt_strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for sqrt(strength) with Fly Ash", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



The QQ plot seems approximately normally distributed

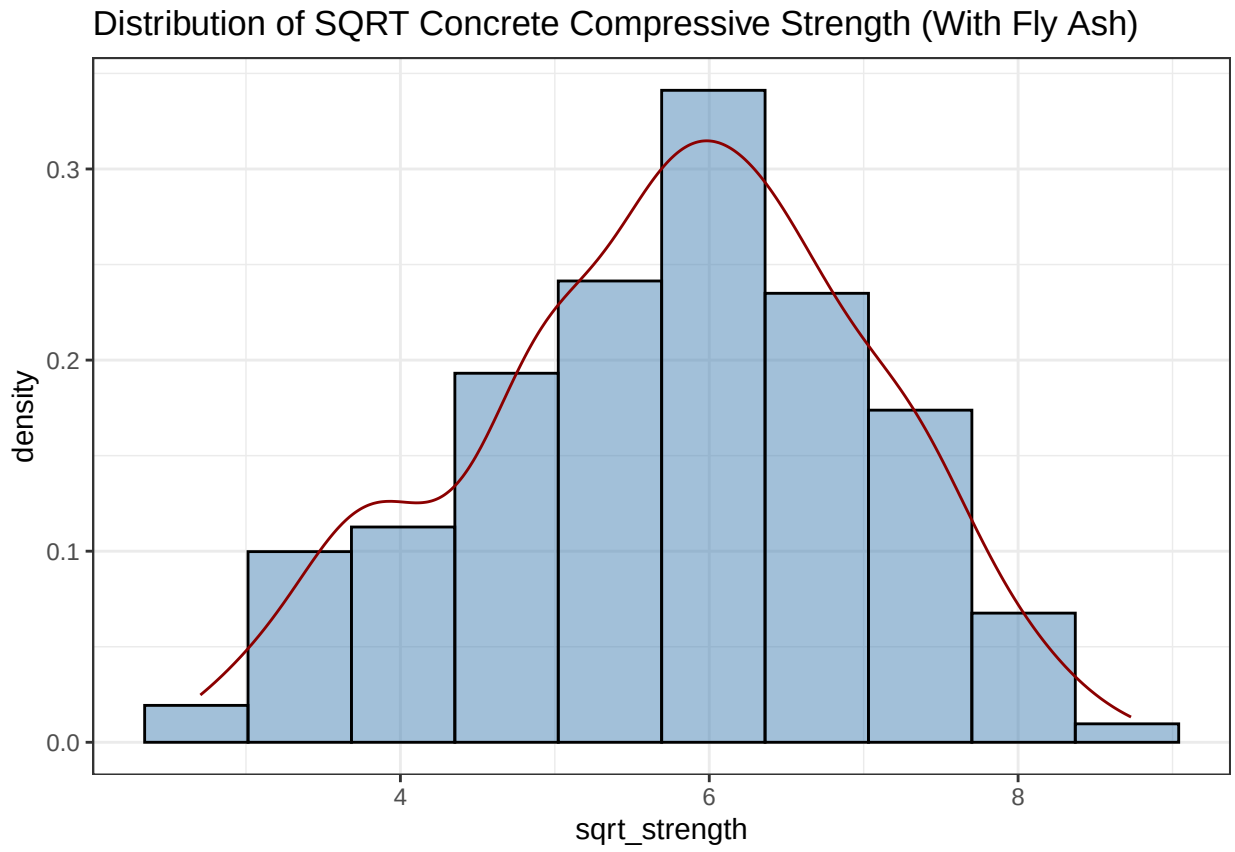
```
no_fly_ash %>%
  ggplot(aes(sample=sqrt_strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for sqrt(strength) without Fly Ash", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



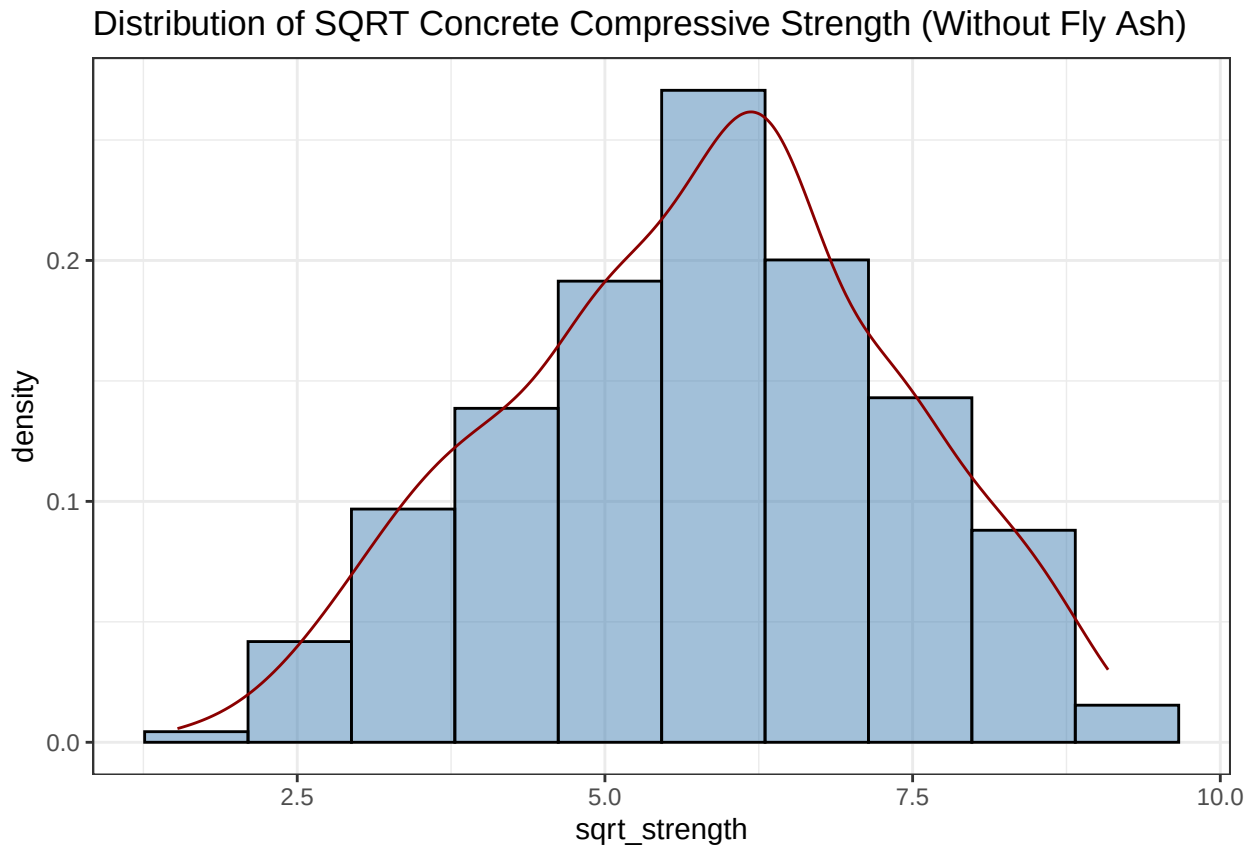
The QQ plot seems approximately normally distributed

Check their distributions

```
has_fly_ash %>%
  ggplot(aes(x=sqrt_strength))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, alpha = 0.5, color='black', fill='steelblue')+
  geom_density(color='darkred')+
  labs(title = "Distribution of SQRT Concrete Compressive Strength (With Fly Ash)")+
  theme_bw()
```



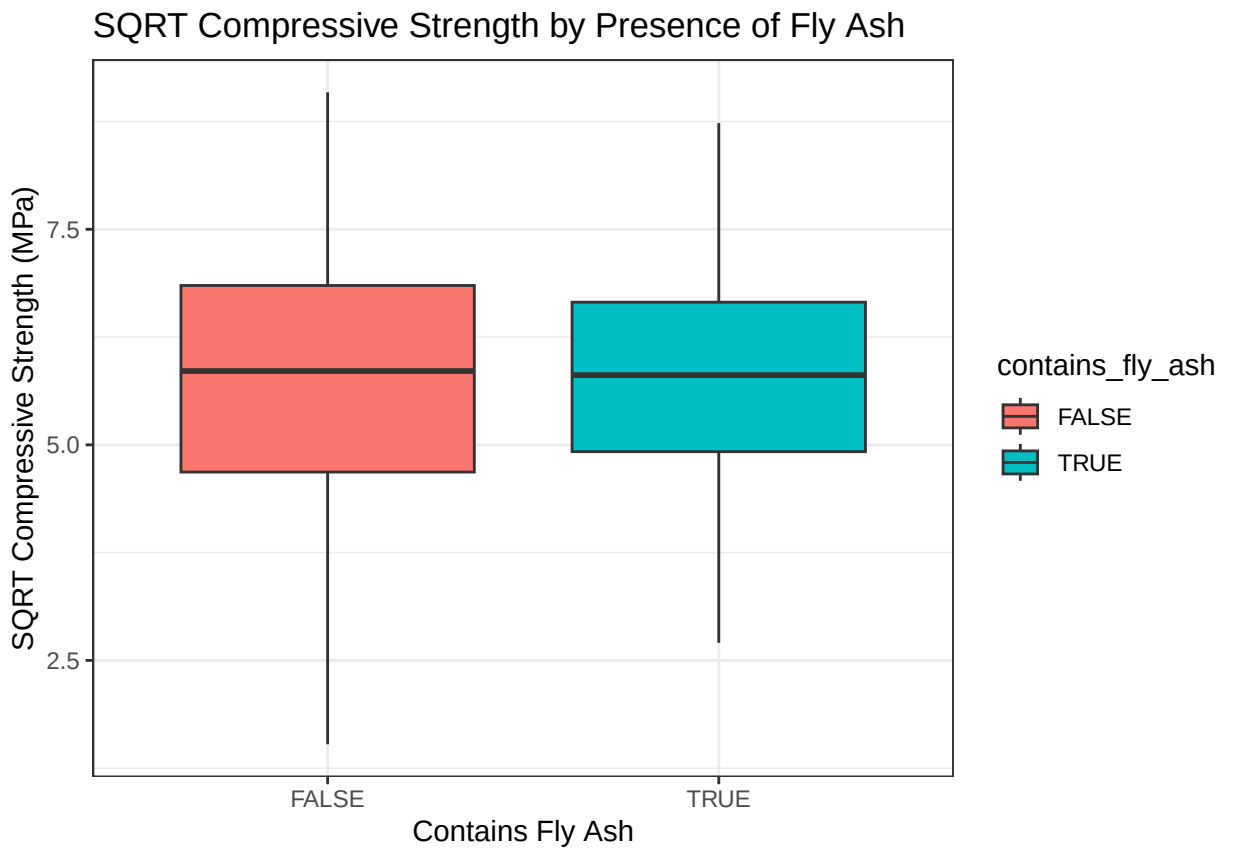

```
no_fly_ash %>%
  ggplot(aes(x=sqrt_strength))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, alpha = 0.5, color='black', fill='steelblue')+
  geom_density(color='darkred')+
  labs(title = "Distribution of SQRT Concrete Compressive Strength (Without Fly Ash)")+
  theme_bw()
```



Both Distributions appear approximately normal

Assumption 3: No significant outliers in each group

```
# Check for outliers with box plot
df %>%
  ggplot(aes(x=contains_fly_ash, y=sqrt_strength, fill=contains_fly_ash)) +
  geom_boxplot()+
  labs(title = "SQRT Compressive Strength by Presence of Fly Ash",
       x = "Contains Fly Ash",
       y = "SQRT Compressive Strength (MPa)") +
  theme_bw()
```



There are no outliers present

Assumption 3: Independence of observations – They are independent.

Assumption 4: Homogeneity of variances

```
bartlett.test(sqrt_strength ~ contains_fly_ash, data=df)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  sqrt_strength by contains_fly_ash
## Bartlett's K-squared = 22.649, df = 1, p-value = 1.945e-06
```

$p < 0.05$, therefore the variances are not equal

All assumptions met for the Independent Two-sample T test except for the equal variances. So we run WELCH T-Test instead

Run Welch T Test

```
t.test(sqrt_strength ~ contains_fly_ash, data=df, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  sqrt_strength by contains_fly_ash
```

```
## t = 0.25253, df = 999.28, p-value = 0.8007
## alternative hypothesis: true difference in means between group FALSE and group TRUE is not equal to 0
## 95 percent confidence interval:
##  -0.1519062  0.1967779
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            5.773822            5.751386
```

The p-value > 0.05, we fail to reject the null hypothesis

This indicates that the mean compressive strength of concrete with fly ash is the same as the mean compressive strength of concrete without fly ash.

The Welch Independent two sample T test results show no statistically significant difference in compressive strength between concrete mixes with and without fly ash ($p = 0.8007$).

2. Hypothesis Test 2:

alpha = 0.05

Null Hypothesis (H0): There is no difference in compressive strength between Fine and Coarse concrete categories.

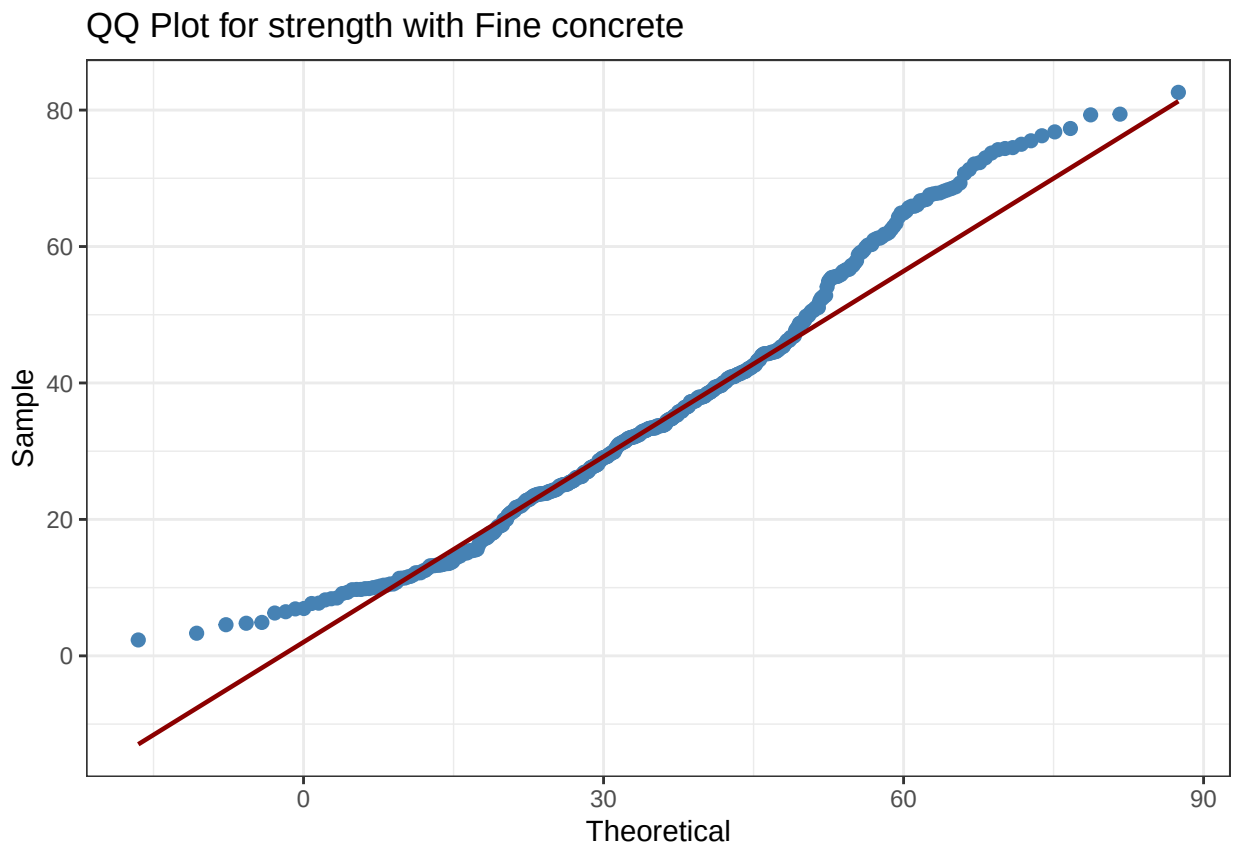
Alternative Hypothesis (H1): There is a significant difference in compressive strength between Fine and Coarse concrete categories.

This is a two tailed test.

Check assumptions:

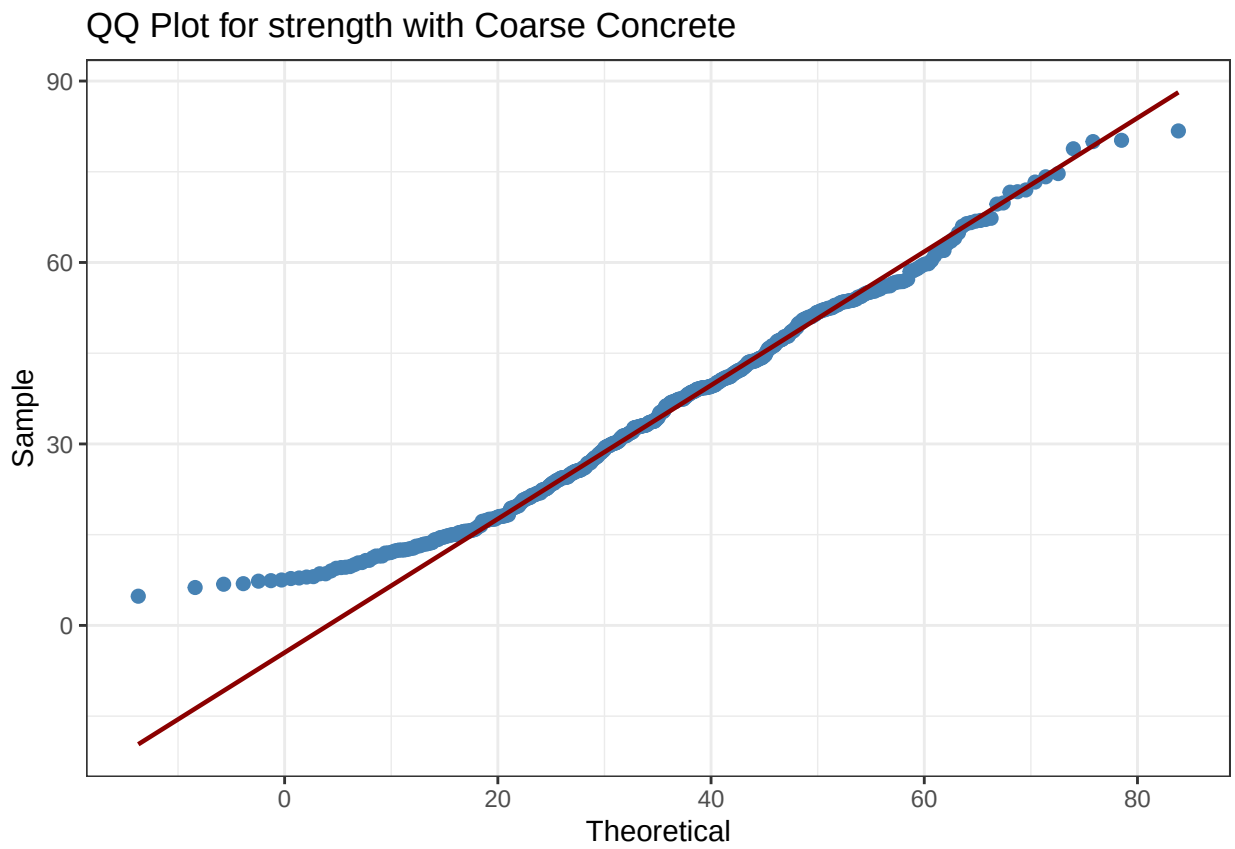
1. check for normality

```
# Filter the samples that contain fly ash
fine <- df %>%
  filter(concrete_category == "Fine")

fine %>%
  ggplot(aes(sample=strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for strength with Fine concrete", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



QQ plot seems like it is not normal

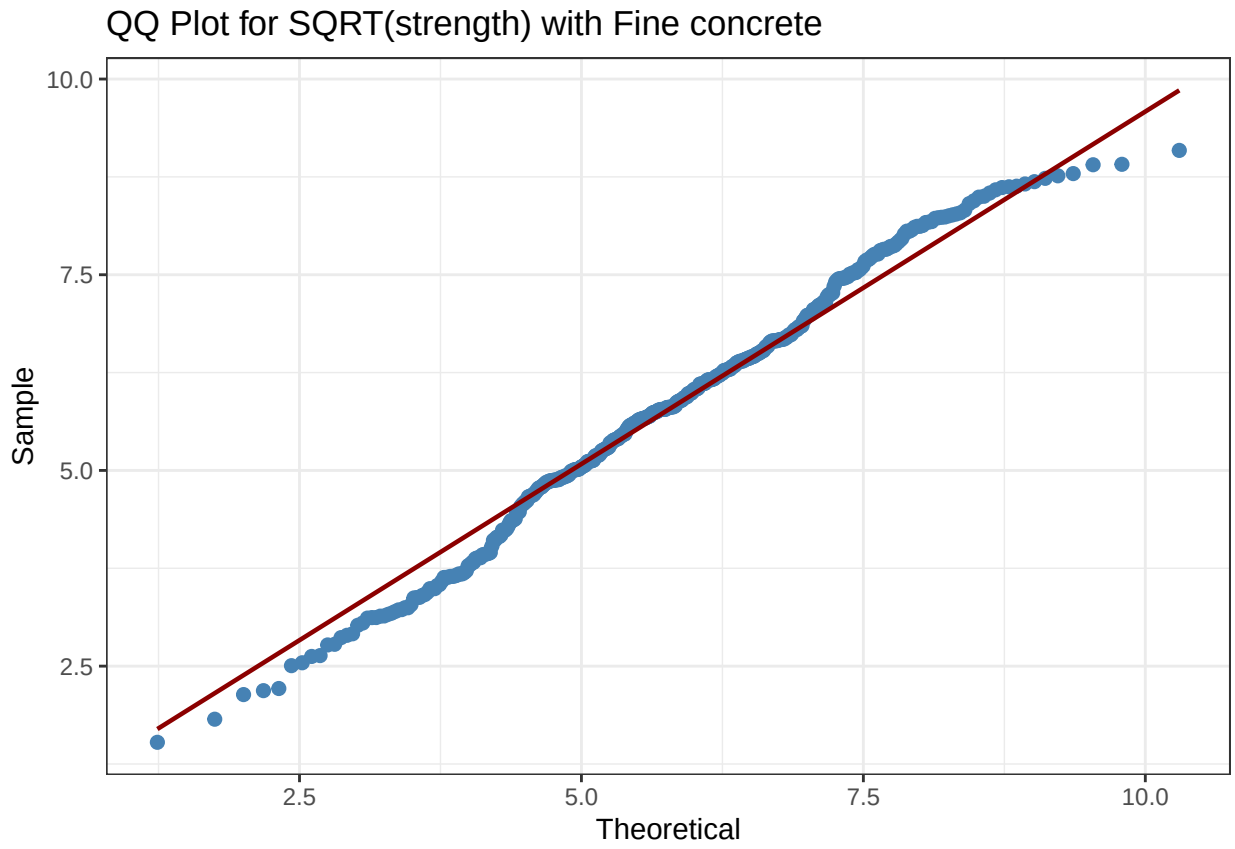
```
# Filter the samples that dont contain fly ash
coarse <- df %>%
  filter(concrete_category == "Coarse")

coarse %>%
  ggplot(aes(sample=strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for strength with Coarse Concrete", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



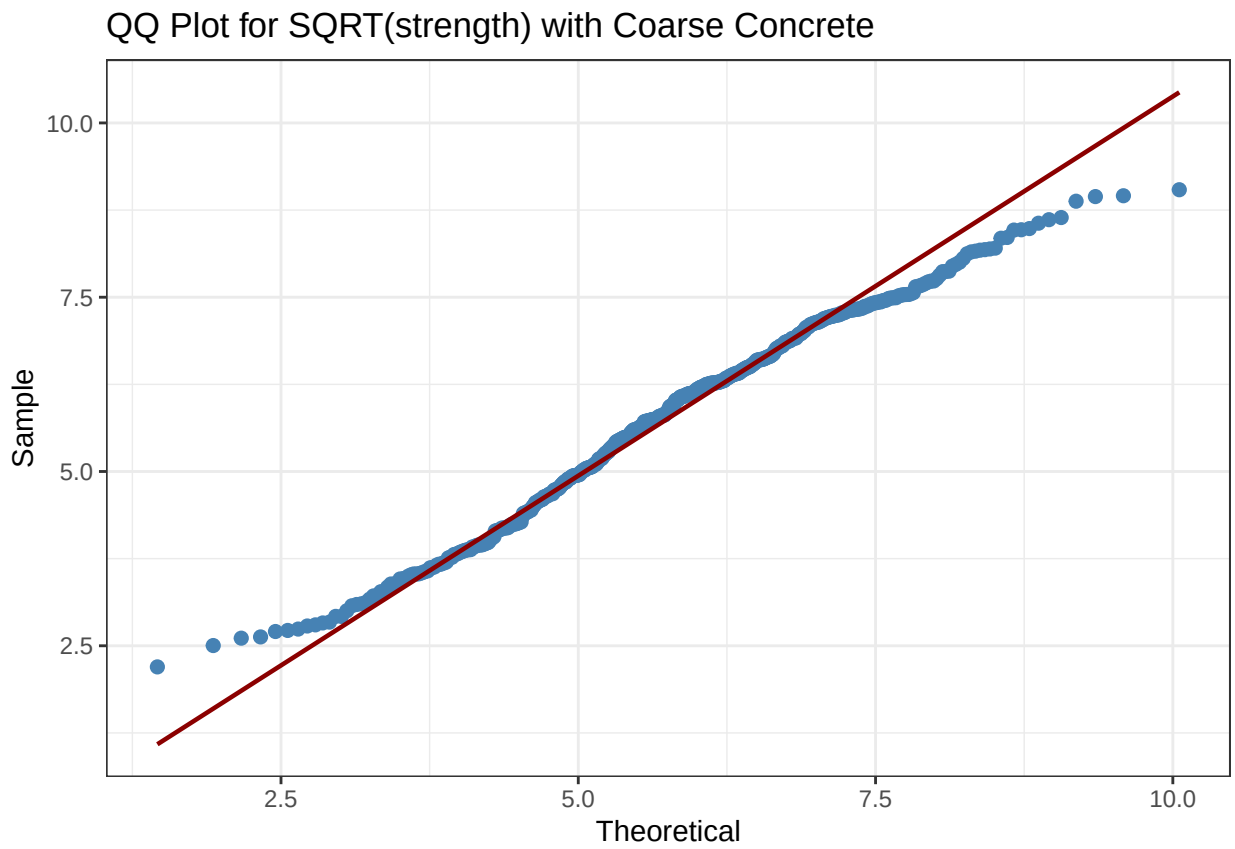
QQ plot seems not normal

Check for normality with transformed strength

```
# check for normality in the transformed variable
fine %>%
  ggplot(aes(sample=sqrt_strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for SQRT(strength) with Fine concrete", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



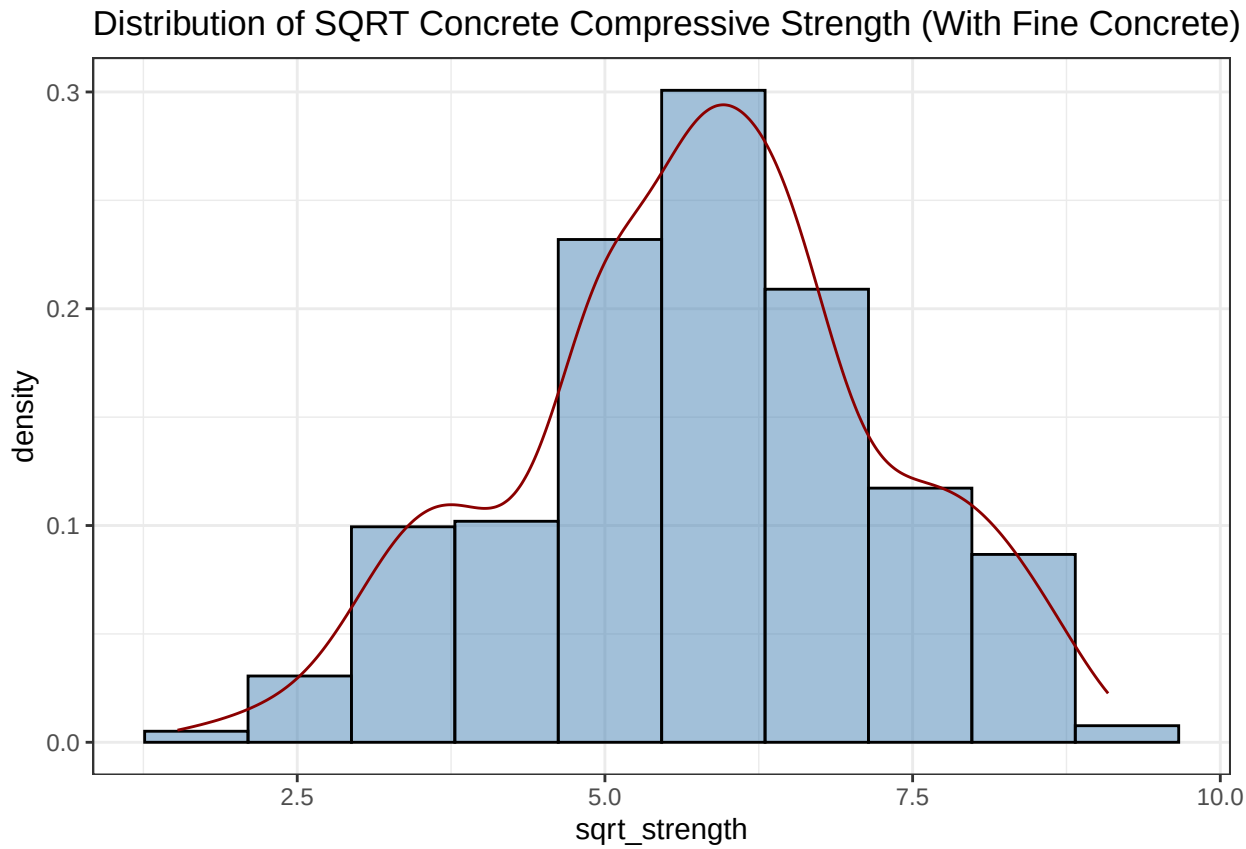

```
coarse %>%
  ggplot(aes(sample=sqrt_strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for SQRT(strength) with Coarse Concrete", x = 'Theoretical', y = 'Sample')+
  theme_bw()
```



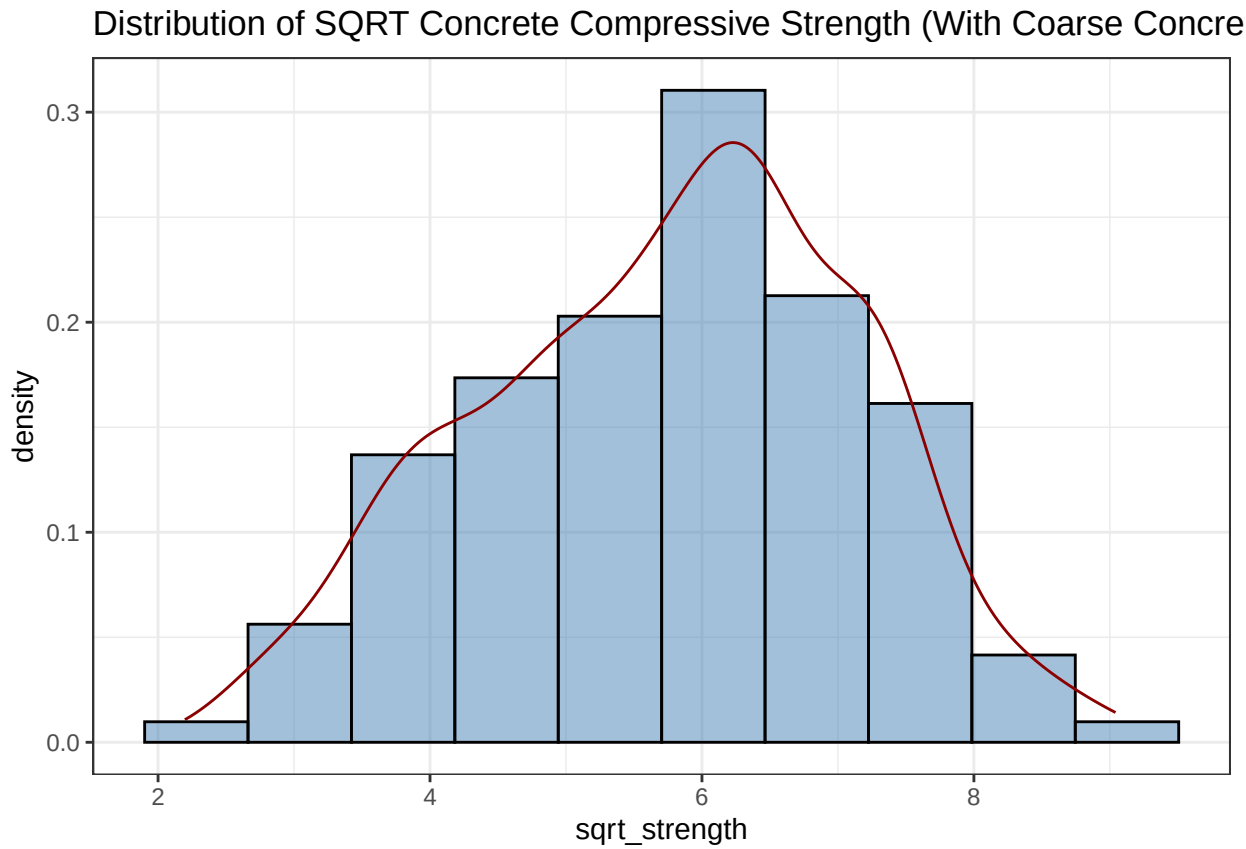
Both distributions are still NOT normally distributed

plot the distributions

```
fine %>%
  ggplot(aes(x=sqrt_strength))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, alpha = 0.5, color='black', fill='steelblue')+
  geom_density(color='darkred')+
  labs(title = "Distribution of SQRT Concrete Compressive Strength (With Fine Concrete)")+
  theme_bw()
```



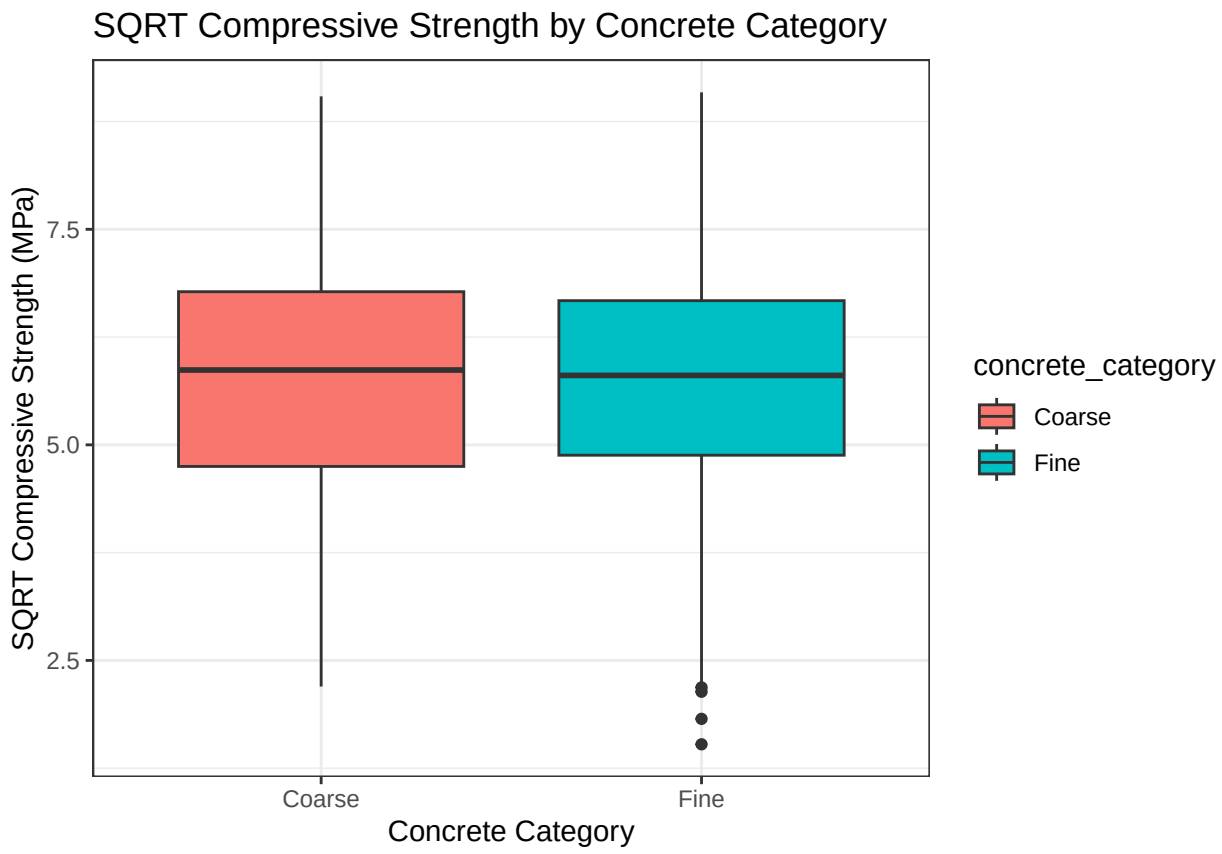

```
coarse %>%
  ggplot(aes(x=sqrt_strength))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, alpha = 0.5, color='black', fill='steelblue')+
  geom_density(color='darkred')+
  labs(title = "Distribution of SQRT Concrete Compressive Strength (With Coarse Concrete)")+
  theme_bw()
```



Both Distributions appear approximately normally distributed

Assumption 3: No significant outliers

```
# Check for outliers with box plot
df %>%
  ggplot(aes(x=concrete_category, y=sqrt_strength, fill=concrete_category)) +
  geom_boxplot()+
  labs(title = "SQRT Compressive Strength by Concrete Category",
       x = "Concrete Category",
       y = "SQRT Compressive Strength (MPa)") +
  theme_bw()
```



outliers ??????

Assumption 3: Independence of observations – They are independent.

Assumption 4: Homogeneity of variances

```
bartlett.test(sqrt_strength ~ concrete_category, data=df)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  sqrt_strength by concrete_category
## Bartlett's K-squared = 2.2238, df = 1, p-value = 0.1359
```

$p < 0.05$, therefore the variances are not equal

All assumptions met for the Independent Two-sample T test except for the equal variances. So we run WELCH T-Test instead

Run Welch T Test All assumptions met for Independent two sample T test

Run Welch T-Test

```
t.test(sqrt_strength ~ concrete_category, data=df, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  sqrt_strength by concrete_category
## t = -0.14543, df = 961.43, p-value = 0.8844
## alternative hypothesis: true difference in means between group Coarse and group Fine is not equal to
## 95 percent confidence interval:
##  -0.1911586  0.1647808
## sample estimates:
## mean in group Coarse   mean in group Fine
##             5.757335             5.770523
```

The p-value > 0.05, we fail to reject the null hypothesis

The Welch Independent two sample T test shows no statistically significant difference in compressive strength between Fine and Coarse concrete categories (p = 0.8844).

3. Hypothesis Test 3: Difference in compressive strength across age groups

Null Hypothesis (H0): The mean compressive strength is the same across different age groups (Early Age, Standard Strength, Mature Age).

Alternative Hypothesis (H1): The mean compressive strength differs across these age groups.

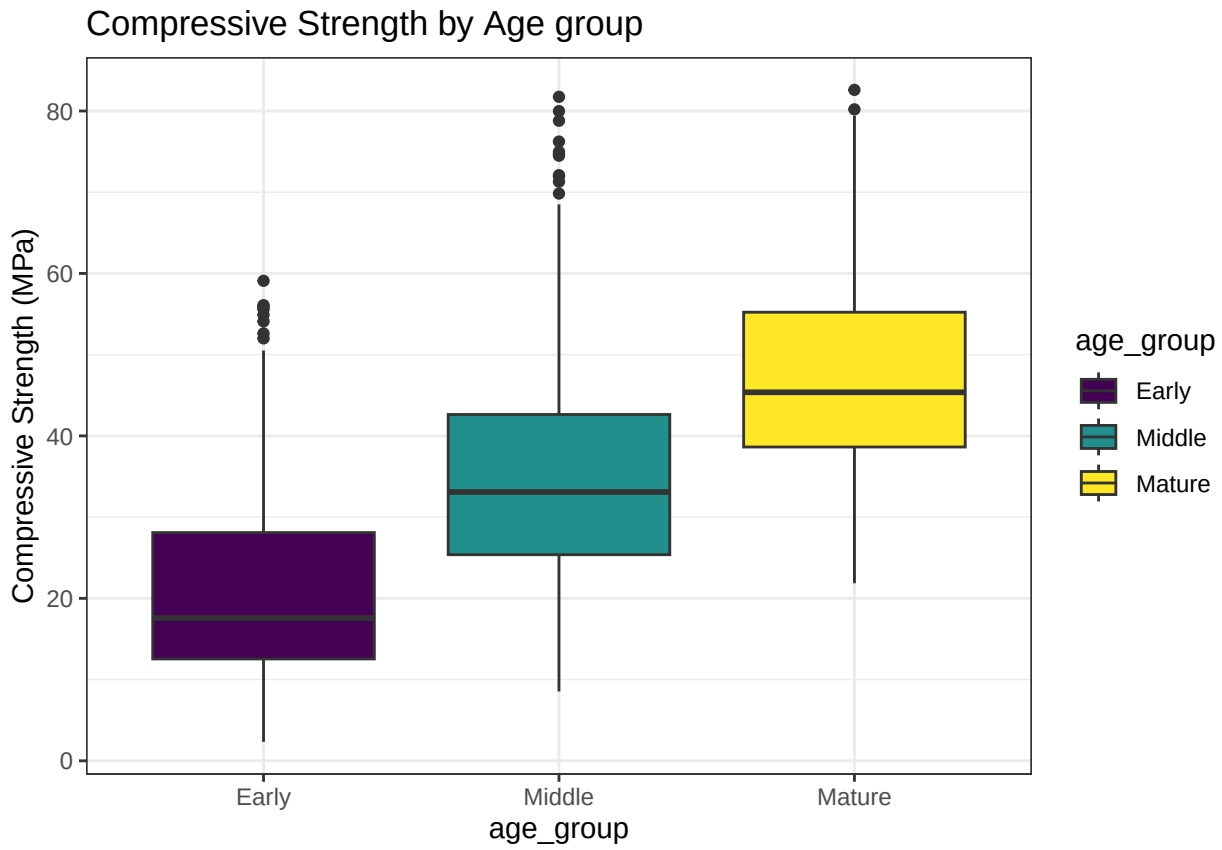
```r
# Create the age groups column
df <- df %>%
  mutate(age_group = case_when(
    age <= 7 ~ "Early",
    age >= 8 & age <= 28 ~ "Middle",
    age >= 29 ~ "Mature"
  ),
  age_group = factor(age_group, levels = c("Early", "Middle", "Mature"), ordered = TRUE))

head(df)
```

```
##   cement blast_furnace_slag fly_ash water superplasticizer coarse_aggregate
## 1  540.0                0.0       0   162              2.5           1040.0
## 2  540.0                0.0       0   162              2.5           1055.0
## 3  332.5              142.5       0   228              0.0            932.0
## 4  332.5              142.5       0   228              0.0            932.0
## 5  198.6              132.4       0   192              0.0            978.4
## 6  266.0              114.0       0   228              0.0            932.0
##   fine_aggregate age concrete_category contains_fly_ash strength sqrt_strength
## 1          676.0  28            Coarse            FALSE 79.98611      8.943495
## 2          676.0  28            Coarse            FALSE 61.88737      7.866852
## 3          594.0 270            Coarse            FALSE 40.26954      6.345828
## 4          594.0 365            Coarse            FALSE 41.05278      6.407244
## 5          825.5 360              Fine            FALSE 44.29608      6.655530
## 6          670.0  90            Coarse            FALSE 47.02985      6.857831
##   age_group
## 1    Middle
## 2    Middle
## 3    Mature
## 4    Mature
## 5    Mature
## 6    Mature
```

One way ANOVA ASSUMPTIONS 1. Dependent variable should be continuous: Compressive strength is continuous 2. Independent variables should be categorical with two or more categories: Age group is categorical with 3 ordered categories 3. Observations should be independent: Yes they are independent.

4. There should be no significant outliers

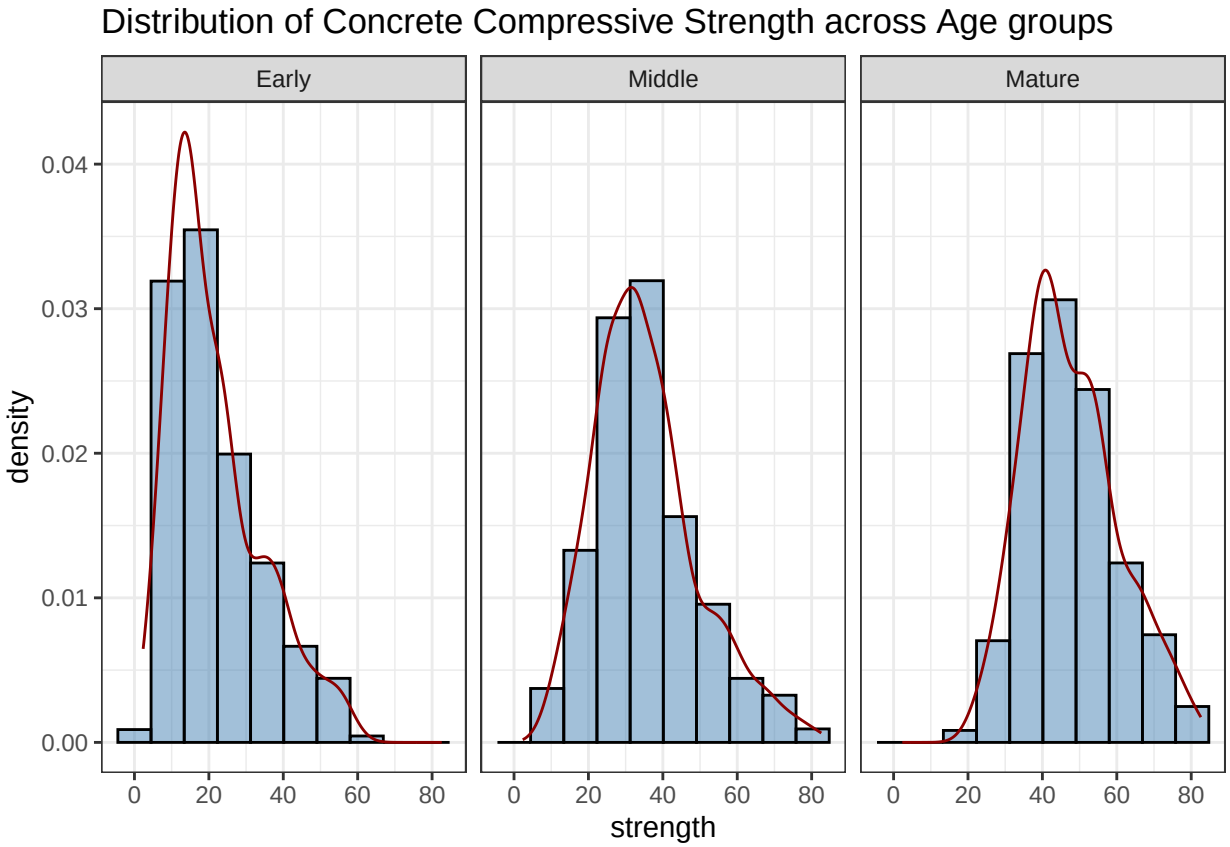
```
# Create a box plot to check for outliers
df %>%
  ggplot(aes(x=age_group, y=strength, fill=age_group))+
  geom_boxplot()+
  labs(title = "Compressive Strength by Age group", y = "Compressive Strength (MPa)") +
  theme_bw()
```



There are some outliers present and they seem significant and not due to errors.

5. Dependent variable should be approximately normally distributed for each category of the independent variable

```
df %>%
  ggplot(aes(x=strength))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, alpha = 0.5, color='black', fill='steelblue')+
  geom_density(color='darkred')+
  labs(title = "Distribution of Concrete Compressive Strength across Age groups")+
  facet_grid(~age_group)+
  theme_bw()
```



The distribution of early age and middle age seem right skewed, mature age seems normal, but we can confirm with shapiro test

```
byf.shapiro(strength ~ age_group, data=df)
```

```
##
##  Shapiro-Wilk normality tests
##
## data:  strength by age_group
##
##              W   p-value
## Early  0.9138 6.555e-11 ***
## Middle 0.9621 8.533e-10 ***
## Mature 0.9764 0.0001858 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All the p-values are less than 0.05, so none of them are normally distributed.

Let us check the normality of the transformed variable

```
# test for normality on sqrt strength
byf.shapiro(sqrt_strength ~ age_group, data=df)
```

```
##
##  Shapiro-Wilk normality tests
```

```
##
## data:  sqrt_strength by age_group
##
##             W   p-value
## Early  0.9723 7.696e-05 ***
## Middle 0.9930   0.02342 *
## Mature 0.9906   0.07749 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Two of the p-values are still less than 0.05 so they are not normal.

Since at least 2 of our assumptions were not met, we will use a non parametric test.

Assumptions of Kruskal Wallis test 1. Independence of Observations: Each observation is independent. 2. Ordinal or Continuous Outcome: Compressive strength is continuous. 3. Non-overlapping Groups: Each sample belongs to only one age group. 4. Similar Shape of Distributions: There appears to be some minor differences in their distributions. Therefore we will also run the dunns test.

Kruskal-Wallis test

```
# Run the Kruskal-Wallis test
kruskal.test(strength ~ age_group, data=df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  strength by age_group
## Kruskal-Wallis chi-squared = 356.28, df = 2, p-value < 2.2e-16
```

The p-value < 0.05, we reject the null hypothesis, indicating that compressive strength distributions differ significantly across the age groups.

3b. Post Hoc Test (Dunn Test) to see which specific age groups differ significantly from each other

```
# run the Dunns test
dunnTest(df$strength, df$age_group, method='bonferroni')
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Bonferroni method.

##        Comparison         Z      P.unadj        P.adj
## 1  Early - Mature -18.87193 1.940932e-79 5.822796e-79
## 2  Early - Middle -11.27465 1.750733e-29 5.252200e-29
## 3 Mature - Middle  10.19294 2.132247e-24 6.396740e-24
```

The p-values are all less than 0.05, therefore, the Dunn's test results with Bonferroni correction indicate significant differences in compressive strength across each of the three age groups.

4. Hypothesis Test 4: Difference in Compressive Strength Based on Water Content Levels

Null Hypothesis (H0): There is no difference in compressive strength among groups with different levels of water content.

Alternative Hypothesis (H1): There is a significant difference in compressive strength among groups with different levels of water content

```
# Divide water into 3 groups based on quantiles
one_third <- quantile(df$water, 0.33)
two_third <- quantile(df$water, 0.66)
print(c(one_third, two_third))
```

```
##    33%    66%
## 173.96 192.00
```

so we divide the levels as: water level < 172.34, 172.34 < water level < 192, and water level >= 192

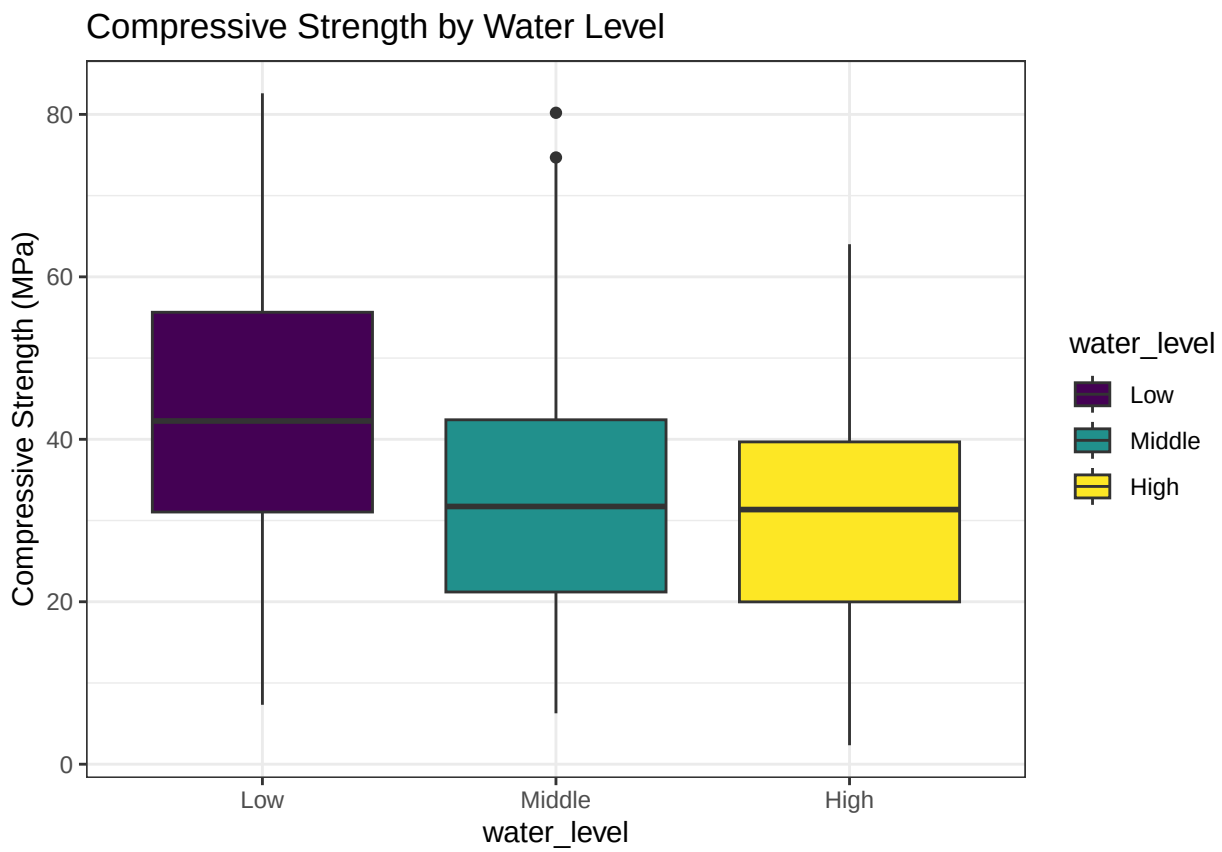
```
# create a new water_level column
df <- df %>%
  mutate(water_level = case_when(
    water < one_third ~ "Low",
    water >= one_third & water < two_third ~ "Middle",
    water >= two_third ~ "High"
  ),
  water_level = factor(water_level, levels = c("Low", "Middle", "High"), ordered = TRUE))

head(df)
```

```
##   cement blast_furnace_slag fly_ash water superplasticizer coarse_aggregate
## 1  540.0                0.0       0   162              2.5           1040.0
## 2  540.0                0.0       0   162              2.5           1055.0
## 3  332.5              142.5       0   228              0.0            932.0
## 4  332.5              142.5       0   228              0.0            932.0
## 5  198.6              132.4       0   192              0.0            978.4
## 6  266.0              114.0       0   228              0.0            932.0
##   fine_aggregate age concrete_category contains_fly_ash strength sqrt_strength
## 1          676.0  28            Coarse            FALSE 79.98611      8.943495
## 2          676.0  28            Coarse            FALSE 61.88737      7.866852
## 3          594.0 270            Coarse            FALSE 40.26954      6.345828
## 4          594.0 365            Coarse            FALSE 41.05278      6.407244
## 5          825.5 360              Fine            FALSE 44.29608      6.655530
## 6          670.0  90            Coarse            FALSE 47.02985      6.857831
##   age_group water_level
## 1    Middle         Low
## 2    Middle         Low
## 3    Mature        High
## 4    Mature        High
## 5    Mature        High
## 6    Mature        High
```

One way ANOVA ASSUMPTIONS 1. Dependent variable should be continuous: Compressive strength is continuous 2. Independent variables should be categorical with two or more categories: Water_level is categorical with 3 ordered categories 3. Observations should be independent: Yes they are independent.

4. There should be no significant outliers

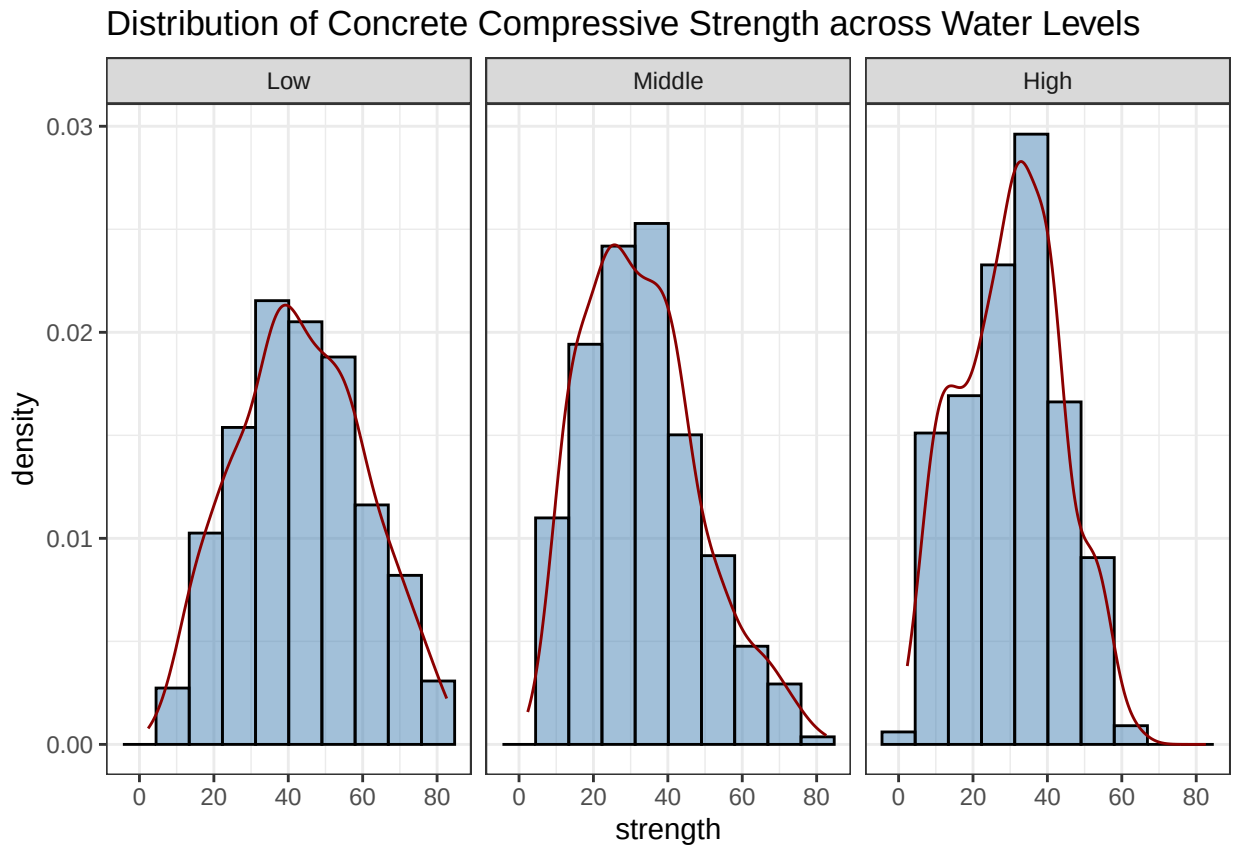
```r
# Create a box plot to check for outliers
df %>%
  ggplot(aes(x=water_level, y=strength, fill=water_level))+
  geom_boxplot()+
  labs(title = "Compressive Strength by Water Level", y = "Compressive Strength (MPa)") +
  theme_bw()
```



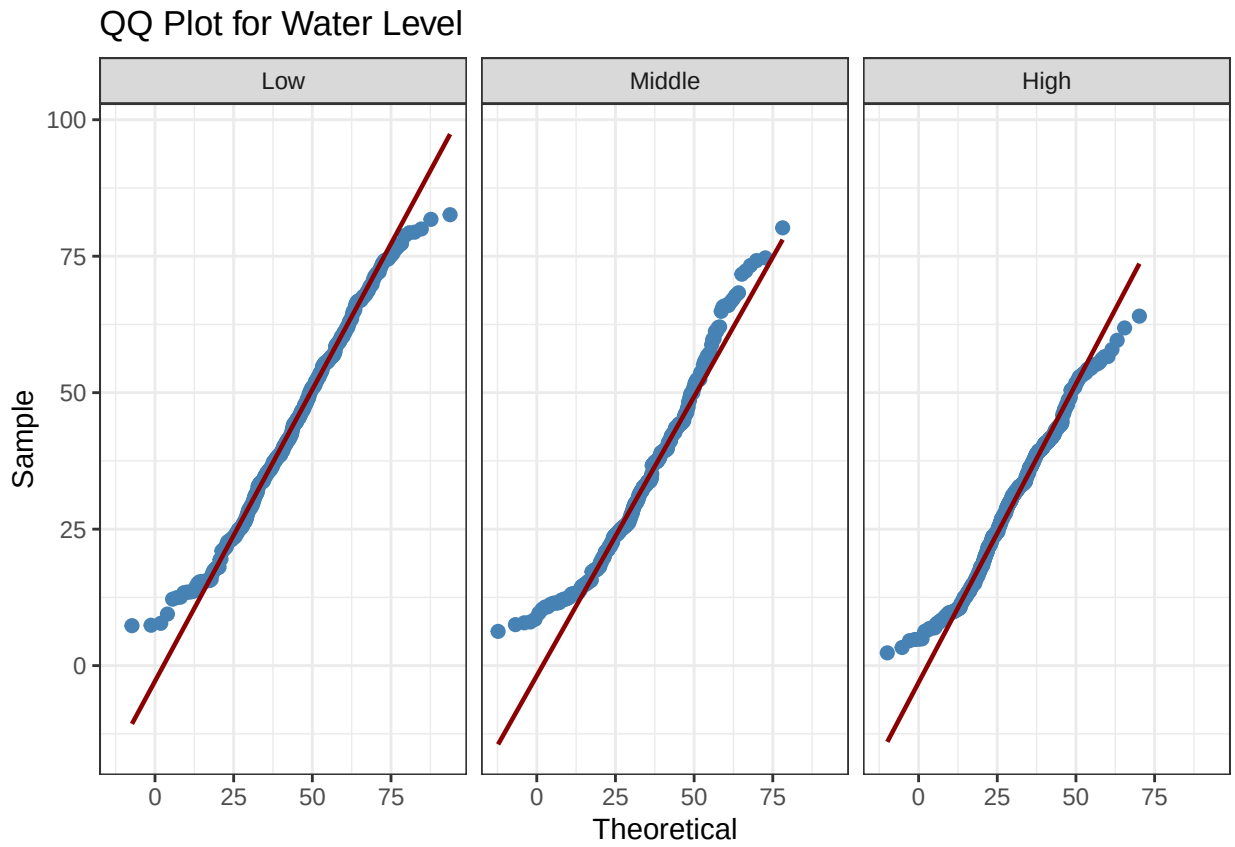
There are some outliers, but it doesn't seem to be influential so we can approve the 4th assumption.

5. Dependent variable should be approximately normally distributed for each category of the independent variable

```r
# check distributions for normality
df %>%
  ggplot(aes(x=strength))+
  geom_histogram(aes(y = after_stat(density)), bins = 10, alpha = 0.5, color='black', fill='steelblue')+
  geom_density(color='darkred')+
  labs(title = "Distribution of Concrete Compressive Strength across Water Levels")+
  facet_grid(~water_level)+
  theme_bw()
```




```
# plot qq plots to check for normality
df %>%
  ggplot(aes(sample=strength))+
  stat_qq_point(size = 2,color = "steelblue")+
  stat_qq_line(color="darkred")+
  labs(title = "QQ Plot for Water Level", x = 'Theoretical', y = 'Sample')+
  facet_grid(~water_level)+
  theme_bw()
```



The QQ plots show that all three levels are approximately normally distributed.

6. Variances of the dependent variable within each category should be homogeneous

```
# use bartlett.test
bartlett.test(strength ~ water_level, data=df)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  strength by water_level
## Bartlett's K-squared = 20.962, df = 2, p-value = 2.807e-05
```

the p-value is less than 0.05, indicating evidence that the variances are not equal.

We can run a WELCH ANOVA instead.

```
# Run the WELCH ANOVA test
oneway.test(strength ~ water_level, data=df, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  strength and water_level
## F = 64.089, num df = 2.00, denom df = 640.21, p-value < 2.2e-16
```

p-value < 0.05, We reject the null hypothesis, indicating a statistically significant difference in compressive strength across the different water content groups (Low, Medium, and High). We can conclude that at least one of the water levels is different from the others in terms of compressive strength.

In order to find out which levels are different, we run a post hoc test

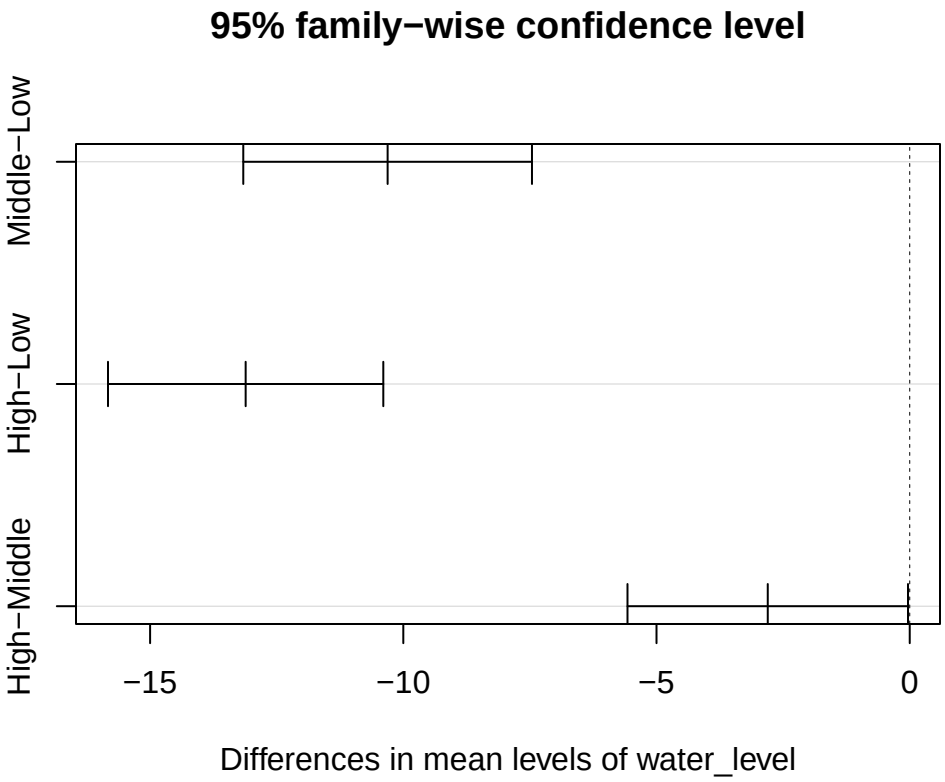
4b. Post Hoc Test (Tukey HSD Test) to see which specific water levels differ significantly from each other

```
# run anova using aov function
water_aov <- aov(strength ~ water_level, data=df)
# run Tukey HSD test
water_post_test <- TukeyHSD(water_aov)
water_post_test
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = strength ~ water_level, data = df)
##
## $water_level
##                    diff        lwr          upr     p adj
## Middle-Low  -10.309837 -13.160241  -7.45943295 0.0000000
## High-Low    -13.113355 -15.831509 -10.39520036 0.0000000
## High-Middle  -2.803518  -5.573048  -0.03398796 0.0464801
```

The TUKEY HSD test results show that all three adjusted p-values are less than 0.05, so we reject the null hypothesis, which means that all water levels are significantly different in terms of compressive strength.

```
par(mar = c(5, 5, 5, 5))
plot(water_post_test)
```


95% family−wise confidence level
Middle−Low
High−Low
High−Middle
−15
−10
−5
0
Differences in mean levels of water_level