PROG8430_Assignment 4

Geedhu Kizhakepura Velayudhan (8899510)

2023-07-30

Basic Set Up

```r
# Clear plots
if(!is.null(dev.list())) dev.off()
```

```
## null device
##           1
```

```r
# Clear console
cat("\014")
```

```r
# Clean workspace
rm(list=ls())

### Calling libraries required
#if(!require(lattice)){install.packages("lattice")}
library("lattice")
#if(!require(corrgram)){install.packages("corrgram")}
library("corrgram")
```

```
##
## Attaching package: 'corrgram'
```

```
## The following object is masked from 'package:lattice':
##
##     panel.fill
```

```r
#if(!require(pastecs)){install.packages("pastecs")}
library("pastecs")
#if(!require(cowplot)){install.packages("cowplot")}
library("cowplot")
```

PART 1 : Preliminary and Exploratory

```
# Read the data PROG8430_Assign04_23W.txt
data = read.csv("PROG8430_Assign04_23W.txt")

# Display first few rows in the dataset
head(data)
```

```
##     DL  VN PG CS   ML DM HZ       CR  WT
## 1  8.1 324  5 13  313  C  N  Sup Del 216
## 2  8.4 135  2 13  830  I  N  Sup Del 160
## 3  8.6 391  3 12  304  C  N  Sup Del  25
## 4 11.3 245  6  7 1258  C  N  Sup Del  67
## 5  5.4 321  1  2  221  C  N Def Post  14
## 6  9.4 397  2  8 1002  I  N  Sup Del  47
```

```
# Appending initials to all variables in the data frame

new_names <- paste0(names(data), "_kv")

# Assigning new names to the column names of the dataframe
colnames(data) <- new_names

# To check if column names are changed or not
head(data)
```

Question 1: Rename all variables with your initials appended

```
##   DL_kv VN_kv PG_kv CS_kv ML_kv DM_kv HZ_kv    CR_kv WT_kv
## 1   8.1   324     5    13   313     C     N  Sup Del   216
## 2   8.4   135     2    13   830     I     N  Sup Del   160
## 3   8.6   391     3    12   304     C     N  Sup Del    25
## 4  11.3   245     6     7  1258     C     N  Sup Del    67
## 5   5.4   321     1     2   221     C     N Def Post    14
## 6   9.4   397     2     8  1002     I     N  Sup Del    47
```

Question 2. Examine the data using the exploratory techniques

```
SD_Data <- stat.desc(data)
print(format(SD_Data,digits=2))
```

Statistical Decsription of Data

```
##                DL_kv   VN_kv    PG_kv   CS_kv   ML_kv DM_kv HZ_kv CR_kv   WT_kv
## nbr.val      4.9e+02 4.9e+02  487.000  487.00 4.9e+02    NA    NA    NA 4.9e+02
## nbr.null     0.0e+00 0.0e+00    0.000    2.00 0.0e+00    NA    NA    NA 0.0e+00
## nbr.na       0.0e+00 0.0e+00    0.000    0.00 0.0e+00    NA    NA    NA 0.0e+00
## min          1.8e+00 8.5e+01   -2.000    0.00 3.5e+01    NA    NA    NA 1.0e-01
## max          1.4e+01 5.0e+02    9.000   24.00 2.0e+03    NA    NA    NA 5.0e+02
## range        1.3e+01 4.1e+02   11.000   24.00 1.9e+03    NA    NA    NA 5.0e+02
## sum          4.1e+03 1.6e+05 1437.000 4494.00 3.7e+05    NA    NA    NA 5.2e+04
## median       8.5e+00 3.2e+02    3.000    8.00 7.0e+02    NA    NA    NA 8.7e+01
## mean         8.5e+00 3.2e+02    2.951    9.23 7.5e+02    NA    NA    NA 1.1e+02
## SE.mean      7.9e-02 3.3e+00    0.069    0.23 1.9e+01    NA    NA    NA 4.2e+00
## CI.mean.0.95 1.5e-01 6.5e+00    0.136    0.46 3.7e+01    NA    NA    NA 8.2e+00
## var          3.0e+00 5.4e+03    2.339   26.65 1.7e+05    NA    NA    NA 8.6e+03
## std.dev      1.7e+00 7.3e+01    1.529    5.16 4.1e+02    NA    NA    NA 9.3e+01
## coef.var     2.0e-01 2.3e-01    0.518    0.56 5.4e-01    NA    NA    NA 8.6e-01
```

From statistical description, we can see that there exists two null values in CS_kv. Let's analyze.

```r
### Check number of null values
nbr.null <- sum(is.na(data))
cat("The number of null values in the variable is:", nbr.null)
```

```
## The number of null values in the variable is: 0
```

During analysis, it became clear that there are no null values. The field CS_kv represents the number of orders a customer has made in the past, and it is acceptable for this value to be 0 as some customers may not have made any orders previously.

```r


# Checking each datatype of dataframe
str(data)
```

Before doing exploratory analysis, let's convert character variables into factor variable, which enables analyzing categorical data more efficiently.

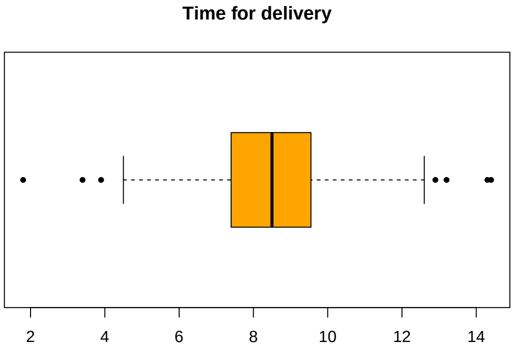
```
## 'data.frame':    487 obs. of  9 variables:
##  $ DL_kv: num  8.1 8.4 8.6 11.3 5.4 9.4 8.2 9.4 9.3 9.7 ...
##  $ VN_kv: int  324 135 391 245 321 397 390 252 355 159 ...
##  $ PG_kv: int  5 2 3 6 1 2 6 2 4 1 ...
##  $ CS_kv: int  13 13 12 7 2 8 13 8 2 12 ...
##  $ ML_kv: int  313 830 304 1258 221 1002 655 1367 675 888 ...
##  $ DM_kv: chr  "C" "I" "C" "C" ...
##  $ HZ_kv: chr  "N" "N" "N" "N" ...
##  $ CR_kv: chr  "Sup Del" "Sup Del" "Sup Del" "Sup Del" ...
##  $ WT_kv: num  216 160 25 67 14 47 7 6 30 177 ...
```

```r
# From this we came to know that, out of 9 variables, there are 3 char variables namely DM_kv, HZ_kv,CR_
# Converting categorical variables to factor variables using factor()
data$DM_kv <- factor(data$DM_kv)
data$HZ_kv <- factor(data$HZ_kv)
data$CR_kv <- factor(data$CR_kv)

# Check if categorical values are converted to factor variables or not?
str(data)
```

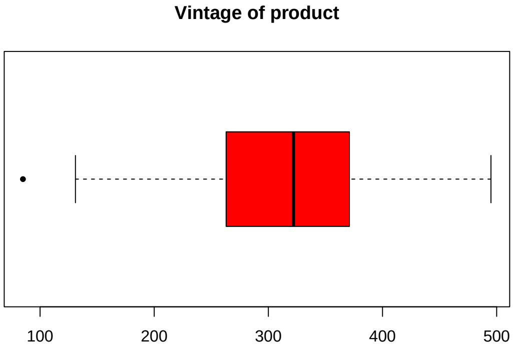
```
## ’data.frame’:    487 obs. of  9 variables:
##  $ DL_kv: num  8.1 8.4 8.6 11.3 5.4 9.4 8.2 9.4 9.3 9.7 ...
##  $ VN_kv: int  324 135 391 245 321 397 390 252 355 159 ...
##  $ PG_kv: int  5 2 3 6 1 2 6 2 4 1 ...
##  $ CS_kv: int  13 13 12 7 2 8 13 8 2 12 ...
##  $ ML_kv: int  313 830 304 1258 221 1002 655 1367 675 888 ...
##  $ DM_kv: Factor w/ 2 levels "C","I": 1 2 1 1 1 2 1 2 1 1 ...
##  $ HZ_kv: Factor w/ 2 levels "H","N": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CR_kv: Factor w/ 2 levels "Def Post","Sup Del": 2 2 2 2 1 2 2 2 2 2 ...
##  $ WT_kv: num  216 160 25 67 14 47 7 6 30 177 ...
```

```
boxplot(data$DL_kv, horizontal=TRUE, pch=20,
        main = "Time for delivery",col="Orange")
```

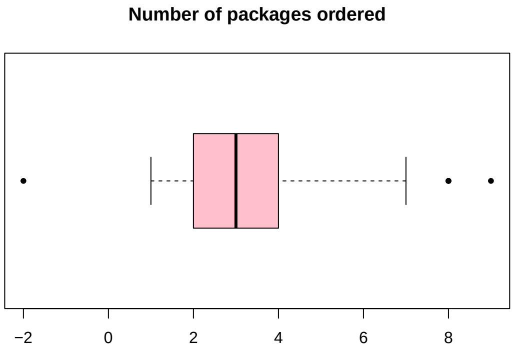


Analyzing Numerical data

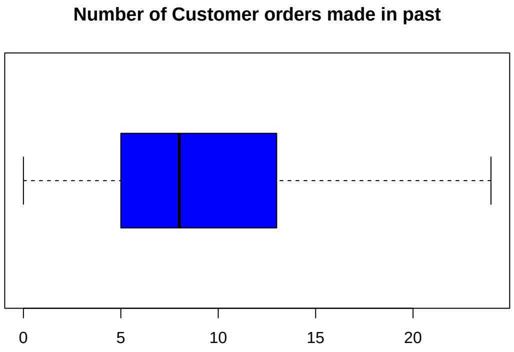
```
boxplot(data$VN_kv, horizontal=TRUE, pch=20,
        main = "Vintage of product",col="Red")
```



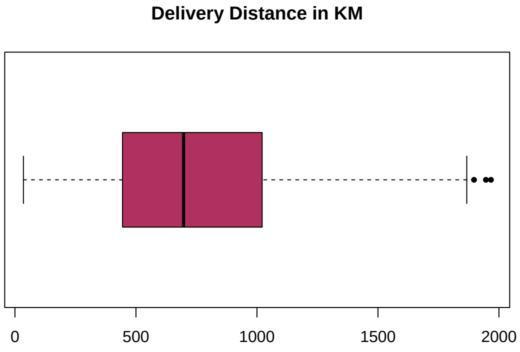
```
boxplot(data$PG_kv, horizontal=TRUE, pch=20,
        main = "Number of packages ordered",col="Pink")
```



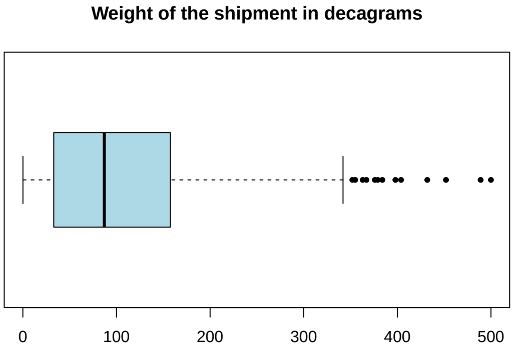
```
boxplot(data$CS_kv, horizontal=TRUE, pch=20,
        main = "Number of Customer orders made in past",col="Blue")
```



```
boxplot(data$ML_kv, horizontal=TRUE, pch=20,
        main = "Delivery Distance in KM",col="Maroon")
```



```
boxplot(data$WT_kv, horizontal=TRUE, pch=20,
        main = "Weight of the shipment in decagrams",col="Light Blue")
```

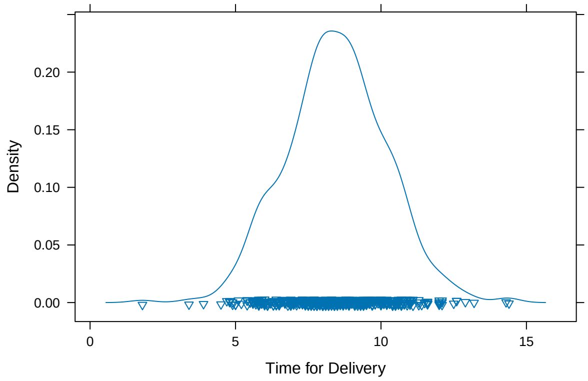


From the boxplots, we could see the presence of outliers in the following attributes:

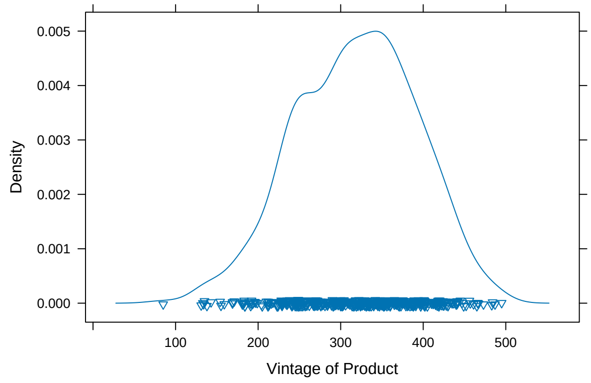
- Time for Delivery
- Vintage of Product
- Number of packages ordered
- Delivery Distance
- Weight of shipment

```
library ("lattice")
densityplot(~ data$DL_kv, pch=6, xlab = "Time for Delivery")
```

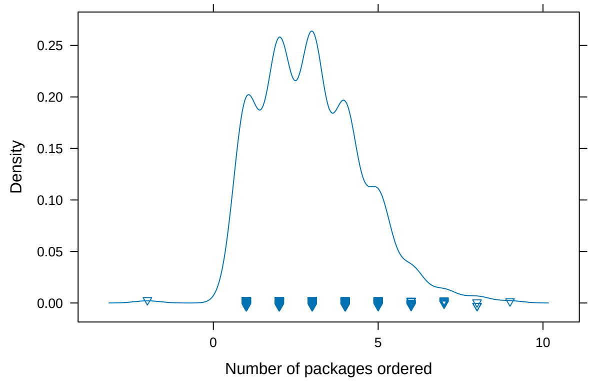
For more detailed analysis, lets plot "Density Plot" for those that has outliers in boxplot



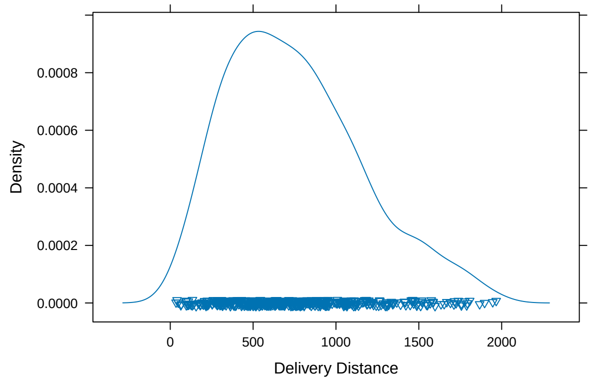
```
densityplot(~ data$VN_kv, pch=6, xlab = "Vintage of Product")
```



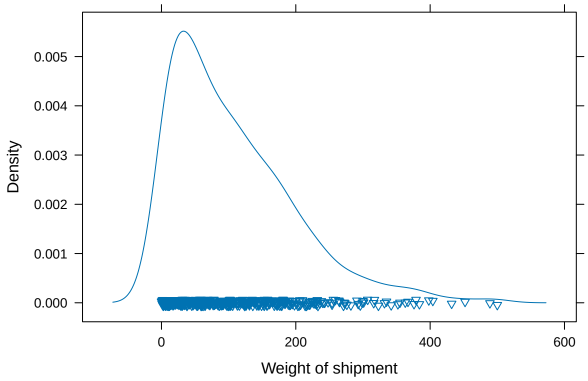
```
densityplot(~ data$PG_kv, pch=6, xlab = "Number of packages ordered")
```



```
densityplot(~ data$ML_kv, pch=6, xlab = "Delivery Distance")
```

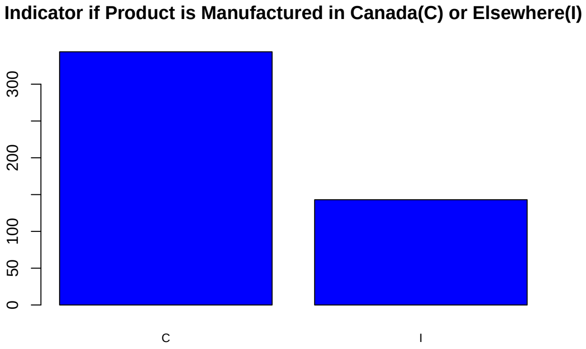


```
densityplot(~ data$WT_kv, pch=6, xlab = "Weight of shipment")
```



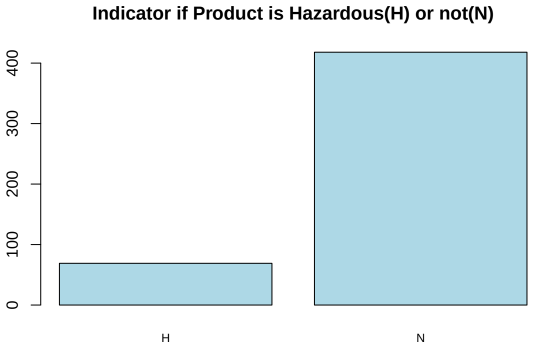
From density plots, it is evident that there are no numerical attributes has much extreme observations that that needs to be removed. Hence not removing any outliers.

```
barplot(table(data$DM_kv), cex.names=.75,col='blue',
        main = "Indicator if Product is Manufactured in Canada(C) or Elsewhere(I)")
```

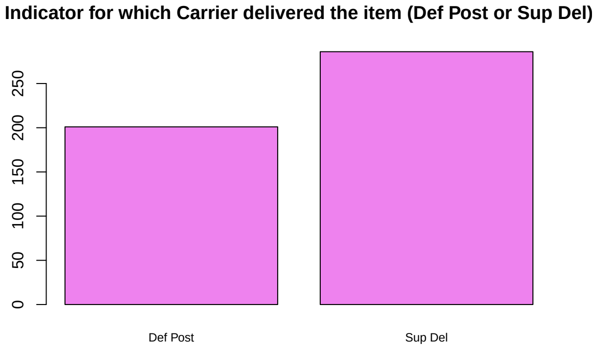


Analyzing Categorical data

```
barplot(table(data$HZ_kv), cex.names=.75,col='Light blue',
        main = "Indicator if Product is Hazardous(H) or not(N)")
```



```
barplot(table(data$CR_kv), cex.names=.75,col='violet',
        main = "Indicator for which Carrier delivered the item (Def Post or Sup Del)")
```



From categorical data, it is clear that most of the products are manufactured in Canada that are not Hazardous and are delivered by “Sup Del” Carrier.

```
# Let's have numerical data in separate data frame

data_num <- data[,-c(6,7,8)]
# Checking if the dataframe contains only numerical data or not
cat("\nNumerical Dataframe: \n")
```

Analyzing Correlations

```
##
## Numerical Dataframe:
```

```
print(head(data_num))
```

```
##   DL_kv VN_kv PG_kv CS_kv ML_kv WT_kv
## 1   8.1   324     5    13   313   216
## 2   8.4   135     2    13   830   160
## 3   8.6   391     3    12   304    25
## 4  11.3   245     6     7  1258    67
## 5   5.4   321     1     2   221    14
## 6   9.4   397     2     8  1002    47
```

```
#Checking the correlation
data_cr <- cor(data_num)

cat("\nCorrelation matrix of Numerical dataset: \n")
```

```
##
## Correlation matrix of Numerical dataset:
```

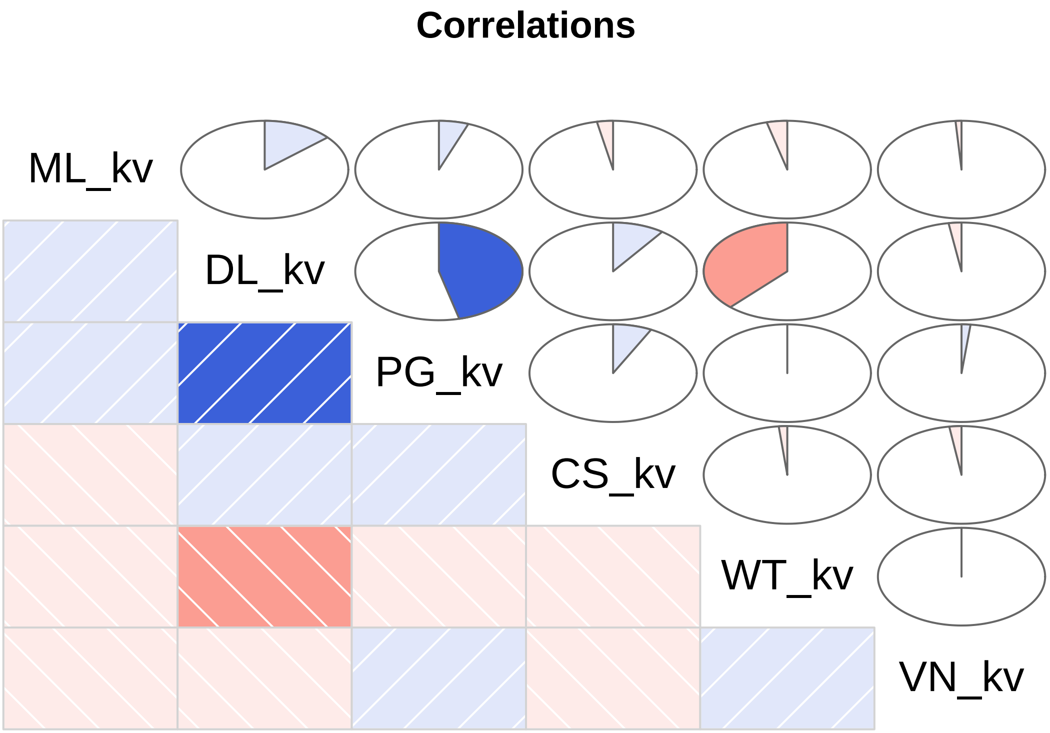
```
print(round(data_cr,2))
```

```
##       DL_kv VN_kv PG_kv CS_kv ML_kv WT_kv
## DL_kv  1.00 -0.02  0.46  0.10  0.14 -0.38
## VN_kv -0.02  1.00  0.02 -0.02 -0.01  0.00
## PG_kv  0.46  0.02  1.00  0.08  0.06 -0.01
## CS_kv  0.10 -0.02  0.08  1.00 -0.03 -0.02
## ML_kv  0.14 -0.01  0.06 -0.03  1.00 -0.04
## WT_kv -0.38  0.00 -0.01 -0.02 -0.04  1.00
```

```
cat("\n Graphical representation of Correlation Matrix: \n")
```

```
##
##  Graphical representation of Correlation Matrix:
```

```
corrgram(data_num, order=TRUE, lower.panel=panel.shade,
         upper.panel=panel.pie, text.panel=panel.txt,
         main="Correlations")
```



Based on the correlation matrix, it is evident that there are no strong correlations between any of the variables. Therefore, there is no need to remove any variables.

Question 3. Using an appropriate technique from class, determine if there is any evidence if one Carrier has faster delivery times than the other. Make sure you explain the approach

you took and your conclusions. Answer: To identify which carrier has faster delivery times than the other we need to consider two attributes one is “Time for Delivery (DL)” and “Indicator for which Carrier delivered the item (Def post or Sup Del)”

Technique used to determine which carrier has faster delivery is through t-test.

Step 1: Formulate Hypothesis: - Null hypothesis (H0): There is no significant difference in the average delivery times between the two carriers. - Alternative hypothesis (Ha): There is a significant difference in the average delivery times between the two carriers.

Step 2: Let’s create a separate dataframe that contains delivery time and Carrier type.

```
delivery_data <- data[,-c(2,3,4,5,6,7,9)]

cat("\n Delivery Data : \n")
```

```
##
##  Delivery Data :
```

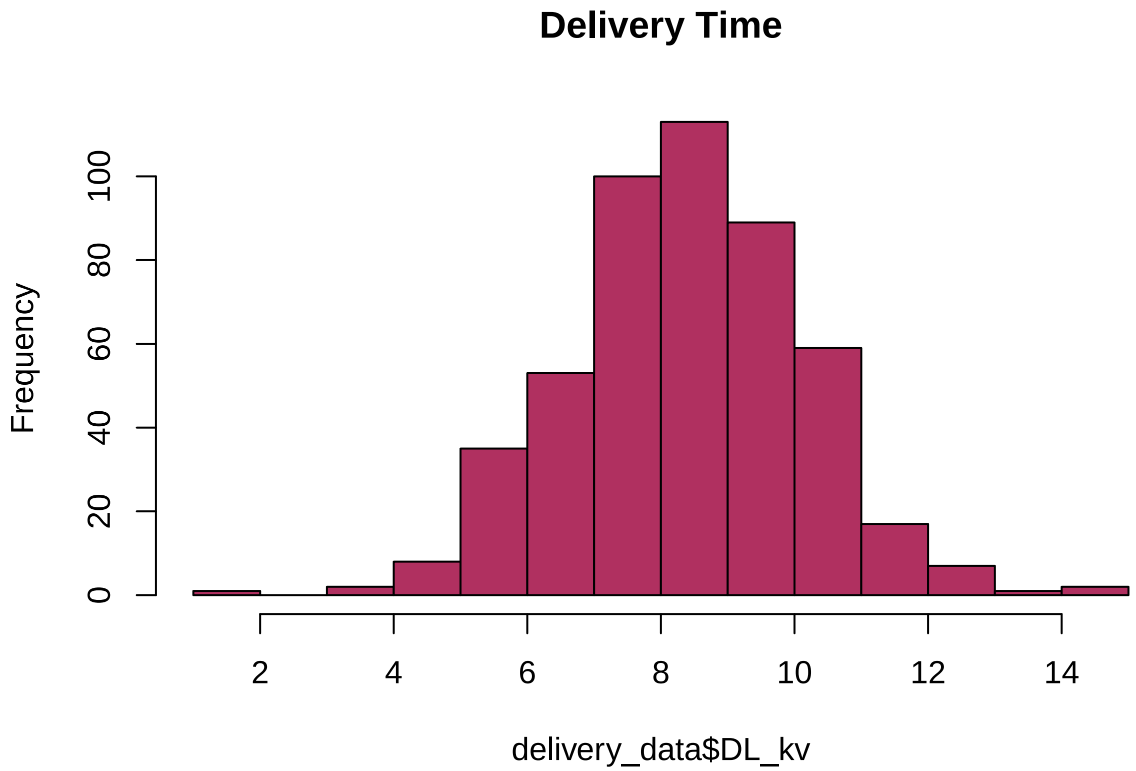
```
print(head(delivery_data))
```

```
##   DL_kv    CR_kv
## 1   8.1  Sup Del
## 2   8.4  Sup Del
## 3   8.6  Sup Del
## 4  11.3  Sup Del
## 5   5.4 Def Post
## 6   9.4  Sup Del
```

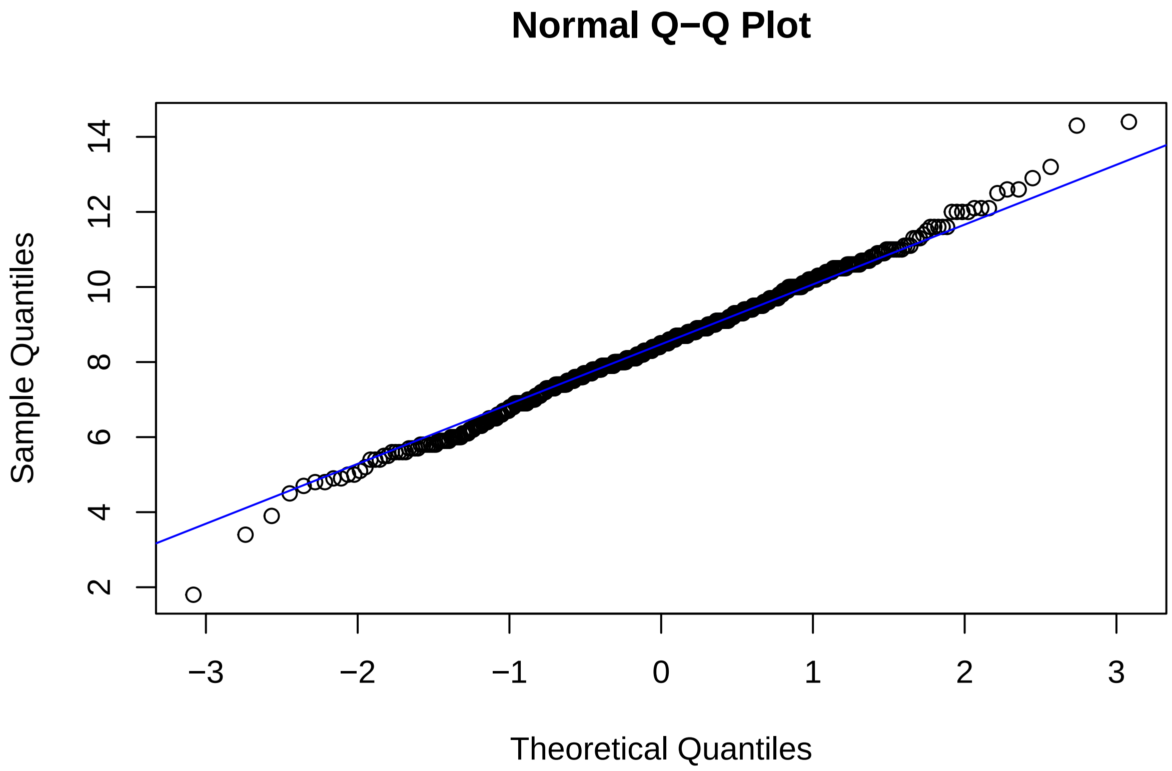
Step 3: Before proceeding with the t-test, it is essential to verify if data is approximately normally distributed and variance is stabilized(homoscedasticity).

Let’s check if data transformation is required or not for numerical data using quantile-quantile plot and by plotting histogram.

```
# Plotting Histogram
hist(delivery_data$DL_kv,col="maroon",main = "Delivery Time")
```



```
# Plotting quantile-quantile plot
qqnorm(delivery_data$DL_kv)
qqline(delivery_data$DL_kv,col="blue")
```



It is evident from the plots that the numerical data for "Delivery Time" is distributed almost normally, thus no transformation is required.

Step 4: Separate the dataframe into two groups based on Carrier type.

```
# Extract delivery times for each carrier
delivery_times_D <- as.numeric(data[data[, 8] == "Def Post", 1])
delivery_times_S <- as.numeric(data[data[, 8] == "Sup Del", 1])

cat("\n Delivery time of Carrier type:Def Post\n")
```

```
##
##  Delivery time of Carrier type:Def Post
```

```
print(head(delivery_times_D))
```

```
## [1] 5.4 8.9 7.4 8.2 6.0 5.7
```

```
cat("\n Delivery time of Carrier type:Sup Del\n")
```

```
##
##  Delivery time of Carrier type:Sup Del
```

```r
print(head(delivery_times_S))
```

```
## [1]  8.1  8.4  8.6 11.3  9.4  8.2
```

Step 5: Perform t-test

```r
ttest_result <- t.test(delivery_times_D, delivery_times_S)

cat("\nT-Test result is: \n")
```

```
##
## T-Test result is:
```

```r
print(ttest_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  delivery_times_D and delivery_times_S
## t = -6.9666, df = 440.2, p-value = 1.19e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.3513810 -0.7566731
## sample estimates:
## mean of x mean of y
##  7.845274  8.899301
```

Following are the evidences to support that one Carrier has faster delivery times than the other:

1. p-value is 1.19e-11 which is less than 0.05, hence we reject null Hypothesis and we accept Alternative hypothesis (Ha) that is "There is a significant difference in the average delivery times between the two carriers".
2. The 95% confidence interval for the true difference in means between carriers D and S is (-1.3513810, -0.7566731). This interval does not contain zero, which supports the conclusion that there is a significant difference in the average delivery times.
3. Sample Estimates: The sample mean delivery time for carrier D is 7.845274. The sample mean delivery time for carrier S is 8.899301.

   As mean of carrier S(Sup Del) is greater than carrier D(Def Post), hence "Sup Del" carrier has longer delivery time when compared "Def Post" carrier.Hence "Def Post" carrier is fastest among the two.

```r


# Total number of records present in the dataset
no_of_records = length(data$DL_kv)
cat("Total number of records in the dataset is ",no_of_records)
```

Question 4: As demonstrated in class, split the dataframe into a training and a test file. This should be a 80/20 split. For the set.seed(), use the last four digits of your student number. The training set will be used to build the following models and the test set will be used to validate them

```
## Total number of records in the dataset is  487
```

```
# set.seed() ensures that the same sequence of random numbers is generated every time you run the
# code with the same seed.

set.seed(9510)

# Sample function is used to draw random samples from a given set of elements.Using this
# function we are randomly selecting 80% of dataset size(487) as train_index.
train_index=sample(no_of_records,no_of_records*0.8)

# Assign 80% of data as training data
train_set <- data[train_index, ]

# Assign 20% of data as test data
test_set <-data[-train_index,]
```

Doing Wilcox test to confirm if both columns have no evidence of statistically significant difference, then it can be assumed that they're from same distribution set and thus acceptable.

Note: Wilcox test is performed only on numerical data

```
wilcox.test(train_set$DL_kv,test_set$DL_kv)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train_set$DL_kv and test_set$DL_kv
## W = 19885, p-value = 0.5085
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(train_set$VN_kv,test_set$VN_kv)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train_set$VN_kv and test_set$VN_kv
## W = 20131, p-value = 0.3906
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(train_set$PG_kv,test_set$PG_kv)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train_set$PG_kv and test_set$PG_kv
## W = 20209, p-value = 0.3469
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(train_set$CS_kv,test_set$CS_kv)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train_set$CS_kv and test_set$CS_kv
## W = 17529, p-value = 0.2177
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(train_set$ML_kv,test_set$ML_kv)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train_set$ML_kv and test_set$ML_kv
## W = 19319, p-value = 0.8362
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(train_set$WT_kv,test_set$WT_kv)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train_set$WT_kv and test_set$WT_kv
## W = 17718, p-value = 0.2809
## alternative hypothesis: true location shift is not equal to 0
```

Since all the wilcox test performed on all the attributes has p-value greater than the significance level 0.05, we do not have enough evidence to reject the null hypothesis. Therefore, we can conclude that there is no significant difference in the distribution of all the attributes between the train_set and test_set data.

PART 2 : Simple Linear Regression

Question 1: Correlations: Create both numeric and graphical correlations (as demonstrated in class) and comment on noteworthy correlations you observe. Are these surprising? Do they make sense? Before using the training data for training the model, it is vital to understand the correlation between the dependent and independent variable of the training data.

```
# Numerical data are extracted and stored in a new data frame
data_numerical <- train_set[,-c(6,7,8)]
# Checking if the dataframe contains only numerical data or not

cat("\nNumerical Dataframe is: \n")
```

```
##
## Numerical Dataframe is:
```

```
print(head(data_numerical))
```

```
##     DL_kv VN_kv PG_kv CS_kv ML_kv WT_kv
## 31    5.5   314     3     3   336   207
## 66    7.5   422     3     3  1179    33
## 232   6.3   213     3     5   420   215
## 258  10.5   457     6     5  1298    82
## 118   8.4   449     4     6   500   113
## 52    5.9   388     2     3  1897    86
```

```
#### Calculating correlation using Pearson menthod
train_cor_pearson <- cor(data_numerical)
print("Correlation value using Pearson correlation Method")
```

Checking Correlation Numerically

```
## [1] "Correlation value using Pearson correlation Method"
```

```
print(round(train_cor_pearson,2))
```

```
##       DL_kv VN_kv PG_kv CS_kv ML_kv WT_kv
## DL_kv  1.00 -0.04  0.45  0.16  0.17 -0.39
## VN_kv -0.04  1.00  0.01  0.00  0.02  0.00
## PG_kv  0.45  0.01  1.00  0.12  0.04  0.01
## CS_kv  0.16  0.00  0.12  1.00 -0.01 -0.03
## ML_kv  0.17  0.02  0.04 -0.01  1.00 -0.06
## WT_kv -0.39  0.00  0.01 -0.03 -0.06  1.00
```

```
#### Calculating correlation using Spearman menthod
train_cor_spearman <- cor(data_numerical,method="spearman")
print("Correlation value using Spearman correlation Method")
```

```
## [1] "Correlation value using Spearman correlation Method"
```

```
round(train_cor_spearman,2)
```

```
##       DL_kv VN_kv PG_kv CS_kv ML_kv WT_kv
## DL_kv  1.00 -0.04  0.45  0.15  0.17 -0.33
## VN_kv -0.04  1.00  0.01  0.00  0.01 -0.03
## PG_kv  0.45  0.01  1.00  0.10  0.01  0.01
## CS_kv  0.15  0.00  0.10  1.00 -0.02 -0.05
## ML_kv  0.17  0.01  0.01 -0.02  1.00 -0.08
## WT_kv -0.33 -0.03  0.01 -0.05 -0.08  1.00
```

The understandings drawn from the output are as follows:

1. Both the results has almost provided the same output.
2. Among all other correlations, the one that has highest correlation is between PG_kv("How many packages of product have been ordered") and DL_kv("Time for delivery (in days, rounded to nearest 10th)") having value 0.45 (in both Spearman and Pearson) indicating "Positive Weak Linear Relationship".This indicates that as value of DL_kv increases PG_kv will also increase and vice-versa.
3. The second highest correlation is between DL_kv ("Time for delivery (in days, rounded to nearest 10th)") and WT_kv("Weight of the shipment (in decagrams)") having vale -0.39 in Pearson and -0.33 in Spearman also indicating "Negative Weak Linear Relationship". This indicates that as value of DL_kv increases WT_kv will decrease and vice-versa.
4. The rest all attributes correlation has correlation value that comes under "Almost no linear relationship".

```r
corrgram(data_numerical, order=TRUE, lower.panel=panel.shade,
         upper.panel=panel.pie, text.panel=panel.txt,
         main="Correlations")
```

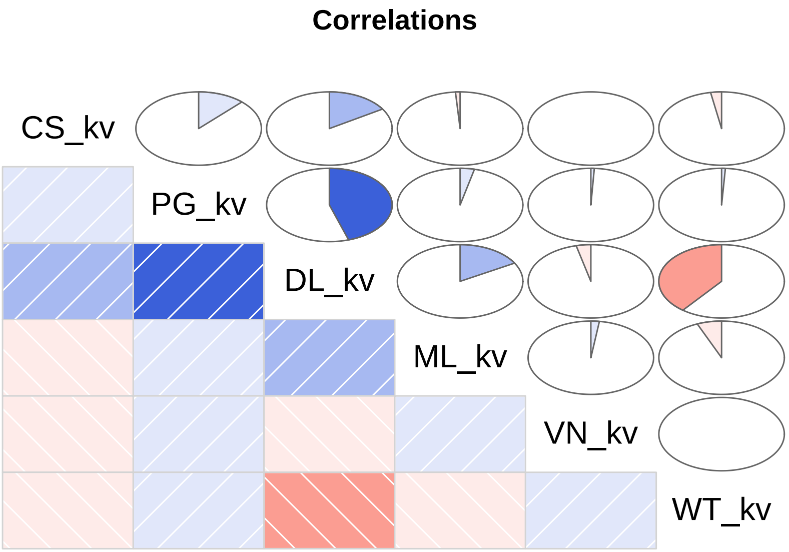


Figure 1: Checking Correlation matrix Graphically

Checking Correlation matrix Graphically Analyzing data through graphical representation is much easier than using numerical values. However, to obtain exact values, numerical calculations are necessary. In graphical representation, blue shades indicate a positive relationship, while red shades indicate a negative relationship.

This draws the same level of attention as numerical correlation calculations. Among the positive correlations, PG_kv and DL_kv have the strongest correlation, while among the negative correlations, DL_kv and WT_kv have the strongest correlation.

The points that are suprising: It makes sense and is not surprising that as the "number of packages of the ordered product" increases, "the time for delivery" similarly increases and vice-versa.

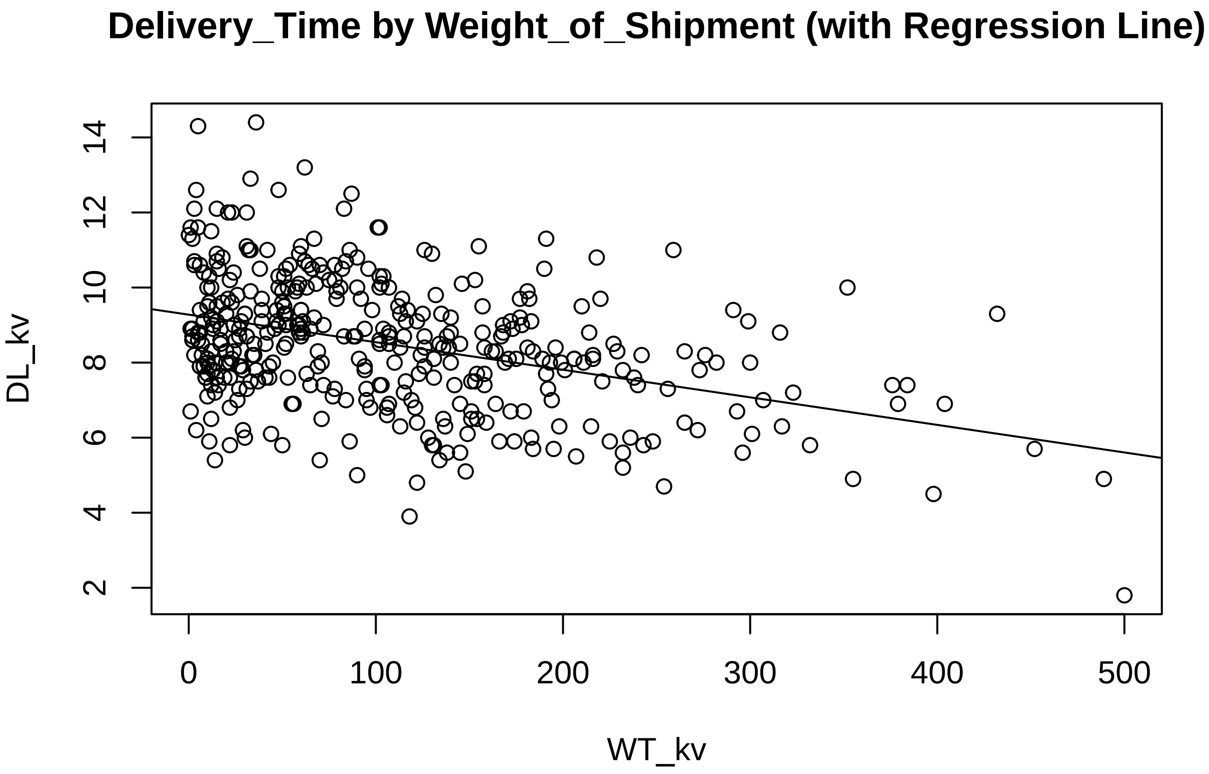
It is shocking because it contradicts the previous statement that "time for delivery" decreases as "weight of the shipment" grows. The rationale is that when the quantity of items grows, weight likewise does, and in the ideal scenario, both WT_kv and PG_kv should have either a positive or negative connection with the delivery time.

```r
# Creating a linear regression model with "time for delivery" as the dependent variable and "weight of
lr_model <- lm(DL_kv ~ WT_kv, data=train_set)
lr_model
```

Question 2: Create a simple linear regression model using time for delivery as the dependent variable and weight of the shipment as the independent. Create a scatter plot of the two variables and overlay the regression line.

```
##
## Call:
## lm(formula = DL_kv ~ WT_kv, data = train_set)
##
## Coefficients:
## (Intercept)        WT_kv
##    9.277346    -0.007343
```

```
# Scatter plot between the two variable of lr_model
plot(DL_kv ~ WT_kv, data=train_set,
     main="Delivery_Time by Weight_of_Shipment (with Regression Line)")
abline(lr_model)
```



The linear regression model created with “time for delivery” (DL_kv) as the dependent variable and “weight of the shipment” (WT_kv) as the independent has intercept of the regression line (B0): 9.277346 and regression coefficient that is slope of the regression line (B1): -0.007343.

Regression equation of the model is : DL_kv = 9.277346 + (-0.007343) * WT_kv

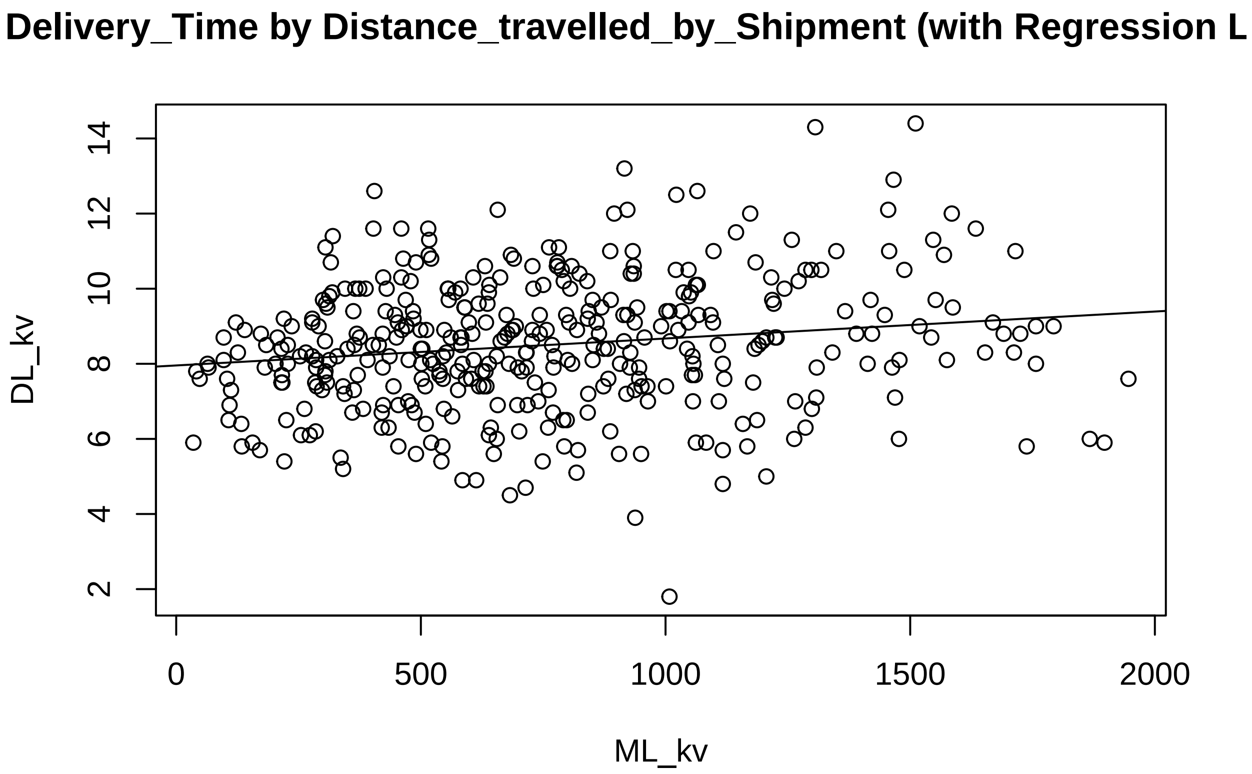
```
# Creating a linear regression model with "time for delivery" as the dependent variable and "distance t
```

```
lr_model_2 <- lm(DL_kv ~ ML_kv, data=train_set)
lr_model_2
```

Question 3. Create a simple linear regression model using time for delivery as the dependent variable and distance the shipment needs to travel as the independent. Create a scatter plot of the two variables and overlay the regression line.

```
##
## Call:
## lm(formula = DL_kv ~ ML_kv, data = train_set)
##
## Coefficients:
## (Intercept)        ML_kv
##   7.9560037    0.0007175
```

```
# Scatter plot between the two variable of lr_model
plot(DL_kv ~ ML_kv, data=train_set,
     main="Delivery_Time by Distance_travelled_by_Shipment (with Regression Line)")
abline(lr_model_2)
```



The linear regression model created with “time for delivery” (DL_kv) as the dependent variable and “distance the shipment needs to travel”(ML_kv) as the independent has intercept of the regression line (B0): 7.9560037 and regression coefficient that is slope of the regression line (B1): 0.0007175.

Regression equation of the model is : DL_kv = 7.9560037 + (0.0007175) * ML_kv

Question 4:As demonstrated in class, compare the models. Which model is superior? Why? Model 1: Linear regression model with “time for delivery” as the dependent variable and “weight of the shipment” as the independent.(lr_model)

```
### Model's RMSE value with train data
pred <- predict(lr_model, newdata=train_set)
RMSE_trn_1 <- sqrt(mean((train_set$DL_kv - pred)^2))
cat("RMSE value of model 1 with train data is ",round(RMSE_trn_1,3))
```

```
## RMSE value of model 1 with train data is  1.616
```

```
### Model's RMSE value with test data
pred <- predict(lr_model, newdata=test_set)
RMSE_tst_1 <- sqrt(mean((test_set$DL_kv - pred)^2))
cat("\nRMSE value of model 1 with test data is ",round(RMSE_tst_1,3))
```

```
##
## RMSE value of model 1 with test data is  1.536
```

```
### Summary of Model 1
cat("\n Summary of Model 1")
```

```
##
##  Summary of Model 1
```

```
print(summary(lr_model))
```

```
##
## Call:
## lm(formula = DL_kv ~ WT_kv, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5109 -1.2039  0.0632  1.0450  5.3870
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.2773456  0.1236198  75.047  < 2e-16 ***
## WT_kv       -0.0073427  0.0008702  -8.438  6.5e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.621 on 387 degrees of freedom
## Multiple R-squared:  0.1554, Adjusted R-squared:  0.1532
## F-statistic: 71.21 on 1 and 387 DF,  p-value: 6.5e-16
```

Model 2: Linear regression model with “time for delivery” as the dependent variable and “distance the shipment needs to travel” as the independent.(lr_model_2)

```r
### Model's RMSE value with train data
pred <- predict(lr_model_2, newdata=train_set)
RMSE_trn_2 <- sqrt(mean((train_set$DL_kv - pred)^2))
cat("RMSE value of model 2 with train data is ",round(RMSE_trn_2,3))
```

```
## RMSE value of model 2 with train data is  1.734
```

```r
### Model's RMSE value with test data
pred <- predict(lr_model_2, newdata=test_set)
RMSE_tst_2 <- sqrt(mean((test_set$DL_kv - pred)^2))
cat("\nRMSE value of model 2 with test data is ",round(RMSE_tst_2,3))
```

```
##
## RMSE value of model 2 with test data is  1.642
```

```r
### Summary of Model 2
cat("\n Summary of Model 2")
```

```
##
##  Summary of Model 2
```

```r
print(summary(lr_model_2))
```

```
##
## Call:
## lm(formula = DL_kv ~ ML_kv, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8792 -1.0122 -0.0535  1.1214  5.4070
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.9560037  0.1848352  43.044  < 2e-16 ***
## ML_kv       0.0007175  0.0002151   3.336 0.000933 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.738 on 387 degrees of freedom
## Multiple R-squared:  0.02795,    Adjusted R-squared:  0.02544
## F-statistic: 11.13 on 1 and 387 DF,  p-value: 0.0009329
```

Comparing the two Linear Regression Models

```
**Characteristics**          **Model 1**     **Model 2**
1. RMSE of Train Set             1.616           1.734

2. RMSE of Test Set              1.536           1.642

3. F-Statistics value          71.21         11.13

4. P -value of F-Statistics      6.5e-16       0.0009329

5. p-value of T-Statistics of  < 2e-16         < 2e-16
   Intercept

6. p-value of T-Statistics of  6.5e-16         0.000933
   Independent variable

7. Median value                  0.0632        -0.0535

8. R squared                   0.1532        0.02544
```

Following are the conclusions made:

1. Both models are fitting reasonably well, as indicated by the lower RMSE values for both the test and train sets. Additionally, while the values for the test and train sets are nearly identical for each model, the test set has a slightly lower value than the train set, which is considered a positive outcome.
2. The F-statistics value for Model 1 is higher than that of Model 2.
3. Despite both models having a p-value of F-statistics less than 0.05, Model 1 has a lower p-value.
4. The p-value for the T-Statistics of the Intercept is identical for both models and is less than 2e-16.
5. The p-value of the T-Statistics for the independent variable is less than 0.05 in both models, but Model 1 has a lower p-value.
6. The median value of both models is close to 0, which is a positive sign.
7. The Adjusted R-squared value should fall between 0 and 1. A value closer to 1 is better, and in this case, model1 has a higher value when compared to model2.

When comparing Model 1 and Model 2, it is evident that Model 1 is superior due to its lower RMSE value, higher F-statistics value, and lower p-value.

PART 3 : Model Development – Multivariate

Question 1: As demonstrated in class, create two models, one using all the variables and the other using backward selection. This should be built using the train set created in Step 2. For each model interpret and comment on the main measures we discussed in class (including RMSE for train and test). MLR_Model1: Multiple Linear Regression Model created using all features of the Data set

```
# MLR_Model 1 created using all features of the Data set
MLR_Model1 = lm(DL_kv ~ ., data=train_set, na.action=na.omit)

#Description about the Model
cat("Description about the MLR_Model 1: \n")
```

```
## Description about the MLR_Model 1:
```

```
print(summary(MLR_Model1))
```

```
##
## Call:
## lm(formula = DL_kv ~ ., data = train_set, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2179 -0.7731  0.0014  0.7759  3.9465
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.2545980  0.3814224  19.020  < 2e-16 ***
## VN_kv        -0.0008197  0.0008606  -0.952 0.341453
## PG_kv         0.5257475  0.0413005  12.730  < 2e-16 ***
## CS_kv         0.0240431  0.0123971   1.939 0.053191 .
## ML_kv         0.0004636  0.0001553   2.985 0.003020 **
## DM_kvI        0.4844956  0.1392516   3.479 0.000561 ***
## HZ_kvN       -0.8092129  0.1764513  -4.586 6.14e-06 ***
## CR_kvSup Del  1.0443097  0.1300452   8.030 1.23e-14 ***
## WT_kv        -0.0066608  0.0006750  -9.868  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.246 on 380 degrees of freedom
## Multiple R-squared:  0.5096, Adjusted R-squared:  0.4993
## F-statistic: 49.36 on 8 and 380 DF,  p-value: < 2.2e-16
```

```
#RMSE Evaluation of Model with Train Data
pred <- predict(MLR_Model1, newdata=train_set)
RMSE_trn_MLR_Model1 <- sqrt(mean((train_set$DL_kv - pred)^2))
cat("RMSE value of MLR_Model1 evaluated using train_set: ",round(RMSE_trn_MLR_Model1,2))
```

```
## RMSE value of MLR_Model1 evaluated using train_set:  1.23
```

```
#RMSE Evaluation of Model with Test Data
pred <- predict(MLR_Model1, newdata=test_set)
RMSE_test_MLR_Model1 <- sqrt(mean((test_set$DL_kv - pred)^2))
cat("\nRMSE value of MLR_Model1 evaluated using test_set: ",round(RMSE_test_MLR_Model1,2))
```

```
##
## RMSE value of MLR_Model1 evaluated using test_set:  1.27
```

MLR__Model2: Mulitple Linear Regression Model created using features selected through Backward Selection

```
# MLR_Model 2 created using features selected through Backward Selection

MLR_Model2 = step(MLR_Model1, direction="backward", details=TRUE)
```

```
## Start:  AIC=180.11
## DL_kv ~ VN_kv + PG_kv + CS_kv + ML_kv + DM_kv + HZ_kv + CR_kv +
##     WT_kv
##
##         Df Sum of Sq    RSS    AIC
## - VN_kv  1     1.409 591.52 179.03
## <none>               590.11 180.11
## - CS_kv  1     5.841 595.95 181.94
## - ML_kv  1    13.836 603.94 187.12
## - DM_kv  1    18.799 608.91 190.30
## - HZ_kv  1    32.660 622.77 199.06
## - CR_kv  1   100.142 690.25 239.08
## - WT_kv  1   151.210 741.32 266.85
## - PG_kv  1   251.646 841.75 316.27
##
## Step:  AIC=179.03
## DL_kv ~ PG_kv + CS_kv + ML_kv + DM_kv + HZ_kv + CR_kv + WT_kv
##
##         Df Sum of Sq    RSS    AIC
## <none>               591.52 179.03
## - CS_kv  1     5.863 597.38 180.87
## - ML_kv  1    13.602 605.12 185.88
## - DM_kv  1    18.188 609.70 188.81
## - HZ_kv  1    33.035 624.55 198.17
## - CR_kv  1   101.571 693.09 238.68
## - WT_kv  1   151.141 742.66 265.55
## - PG_kv  1   251.573 843.09 314.89
```

```
cat("\nDuring the process of creating the model, the \"backward selection\" technique was utilized for f
```

```
##
## During the process of creating the model, the "backward selection" technique was utilized for feature
```

```
#Description about the Model2
cat("\nDescription about the MLR_Model 2: \n")
```

```
##
## Description about the MLR_Model 2:
```

```
print(summary(MLR_Model2))
```

```
##
## Call:
## lm(formula = DL_kv ~ PG_kv + CS_kv + ML_kv + DM_kv + HZ_kv +
##     CR_kv + WT_kv, data = train_set, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2710 -0.7813 -0.0134  0.7536  3.8904
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.9975406  0.2695009  25.965  < 2e-16 ***
## PG_kv         0.5256704  0.0412954  12.730  < 2e-16 ***
## CS_kv         0.0240876  0.0123955   1.943 0.052721 .
## ML_kv         0.0004595  0.0001552   2.960 0.003270 **
## DM_kvI        0.4754521  0.1389106   3.423 0.000687 ***
## HZ_kvN       -0.8135634  0.1763707  -4.613 5.43e-06 ***
## CR_kvSup Del  1.0504446  0.1298698   8.088 8.15e-15 ***
## WT_kv        -0.0066593  0.0006749  -9.867  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.246 on 381 degrees of freedom
## Multiple R-squared:  0.5084, Adjusted R-squared:  0.4994
## F-statistic: 56.29 on 7 and 381 DF,  p-value: < 2.2e-16
```

```
#RMSE Evaluation of Model2 with Train Data
pred <- predict(MLR_Model2, newdata=train_set)
RMSE_trn_MLR_Model2 <- sqrt(mean((train_set$DL_kv - pred)^2))
cat("RMSE value of MLR_Model2 evaluated using train_set: ",round(RMSE_trn_MLR_Model2,2))
```

```
## RMSE value of MLR_Model2 evaluated using train_set:  1.23
```

```
cat('\n')
```

```
#RMSE Evaluation of Model2 with Test Data
pred <- predict(MLR_Model2, newdata=test_set)
RMSE_test_MLR_Model2 <- sqrt(mean((test_set$DL_kv - pred)^2))
cat("\nRMSE value of MLR_Model2 evaluated using test_set: ",round(RMSE_test_MLR_Model2,2))
```

```
##
## RMSE value of MLR_Model2 evaluated using test_set:  1.26
```

** Interpreting on the two Multiple Linear Regression Models**

```
**Characteristics**                **Model 1**                             **Model 2 **
1. RMSE of Train Set                   1.23                                    1.23

2. RMSE of Test Set                    1.27                                    1.26

3. F-Statistics value                 49.36                                    56.29

4. P -value of F-Statistics            < 2.2e-16                               < 2.2e-16

5. p-value of T-Statistics of      All variables has p-value <0.05         All variables has
   Independent variable            except for VN_kv which is               p-value <0.05.
                                   greater than 0.05.

6. p-value of T-Statistics of        < 2e-16                                   < 2e-16
   Intercept

7. Median value                         0.0014                                     -0.0134

8. Adjusted R-squared                 0.4993                                    0.4994
```

Following are the conclusions made:

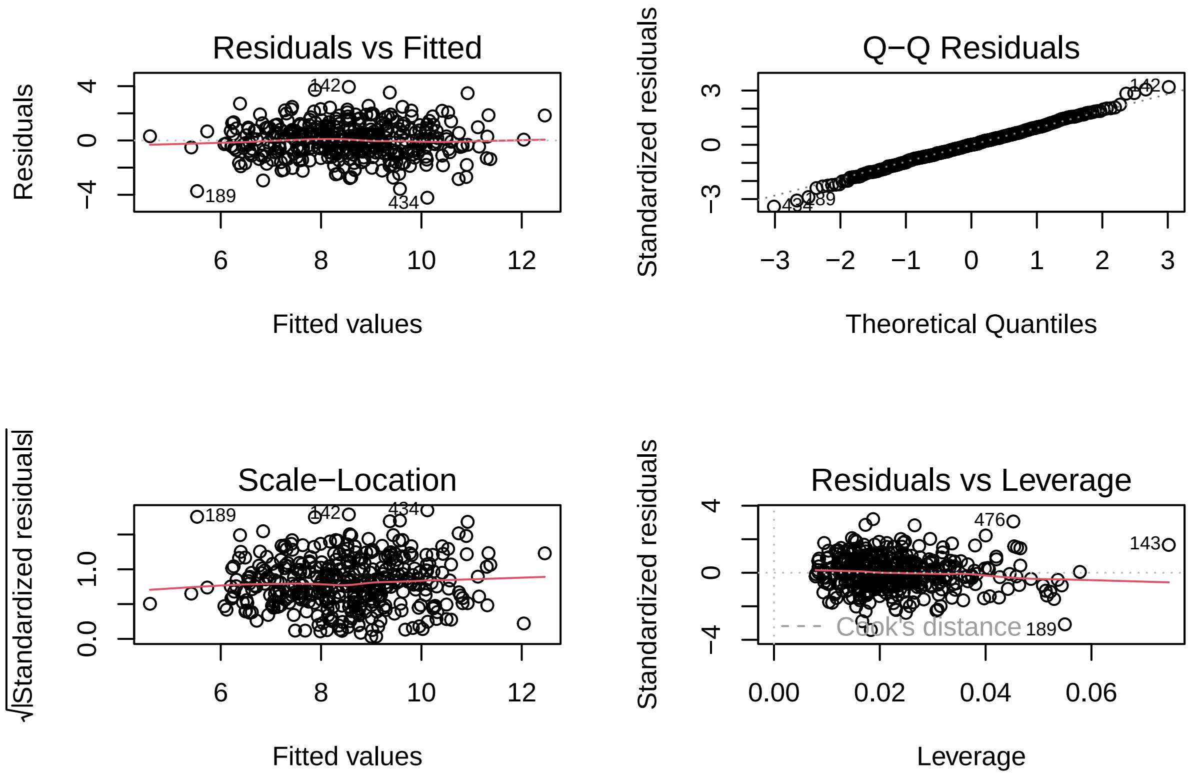
1. Both models are fitting reasonably well, as indicated by the lower RMSE values for both the test and train sets. Additionally, while the values for the test and train sets are nearly identical for each model, the train set has a slightly lower value than the test set, which is acceptable.
2. The F-statistics value for Model 2 is higher than that of Model 1, indicating a significant relationship between the regression coefficients in Model 2 compared to Model 1.
3. The p-value for the F-statistics in both models is less than 0.05, indicating that the null hypothesis is rejected, and at least one coefficient is not equal to zero.
4. The p-value for the T-Statistics of the Intercept in both models is less than 2e-16, indicating that it is below the significance level of 0.05 and therefore rejects the null hypothesis.
5. The p-value for the T-Statistics of the independent variable in Model 1 is 0.05, except for the VN_kv feature. However, all variables in Model 2 have a p-value less than 0.05.
6. The median value of both models is close to zero, which is a positive sign.
7. The Adjusted R-squared value should fall between 0 and 1. A value closer to 1 is better, and in this case, model2 has a slightly higher value by a difference of 0.0001 compared to model1.

When comparing Model 1 and Model 2, it is evident that Model 2 is superior due to higher F-statistics value and p-value of T-Statistics of all independent variables less than 0.05.

PART 4 : Model Evaluation – Verifying Assumptions - Multivariate

Question 1:For both models created in Step 4, evaluate the main assumptions of regression (for example, Error terms mean of zero, constant variance and normally distributed, etc.) MLR_Model1: Multiple Linear Regression Model created using all features of the Data set

```
par(mfrow = c(2, 2))
plot(MLR_Model1)
```



Following are the inferences made:

1. Residual vs Fitted graph:
    - The range of the y-axis is currently between -4 to 4, which indicates the existence of outliers. Ideally, it should be between -2 to 2. However, despite the presence of outliers, the deviation of the line from the fitted dotted line (representing zero residual) has not been altered significantly. As a result, Model1 meets the assumptions of linearity and homoscedasticity quite well.
2. Q-Q Residuals Plot:
    - With the exception of a few countable observations, all other observations align closely along the 45-degree line on the QQ-plot. Therefore, it is reasonable to assume that normality is maintained in this case.
3. Scale-Location plot:

- The line is almost horizontal satisfying assumptions of equal variance(homoscedasticity).

4. Residual vs Leverage:

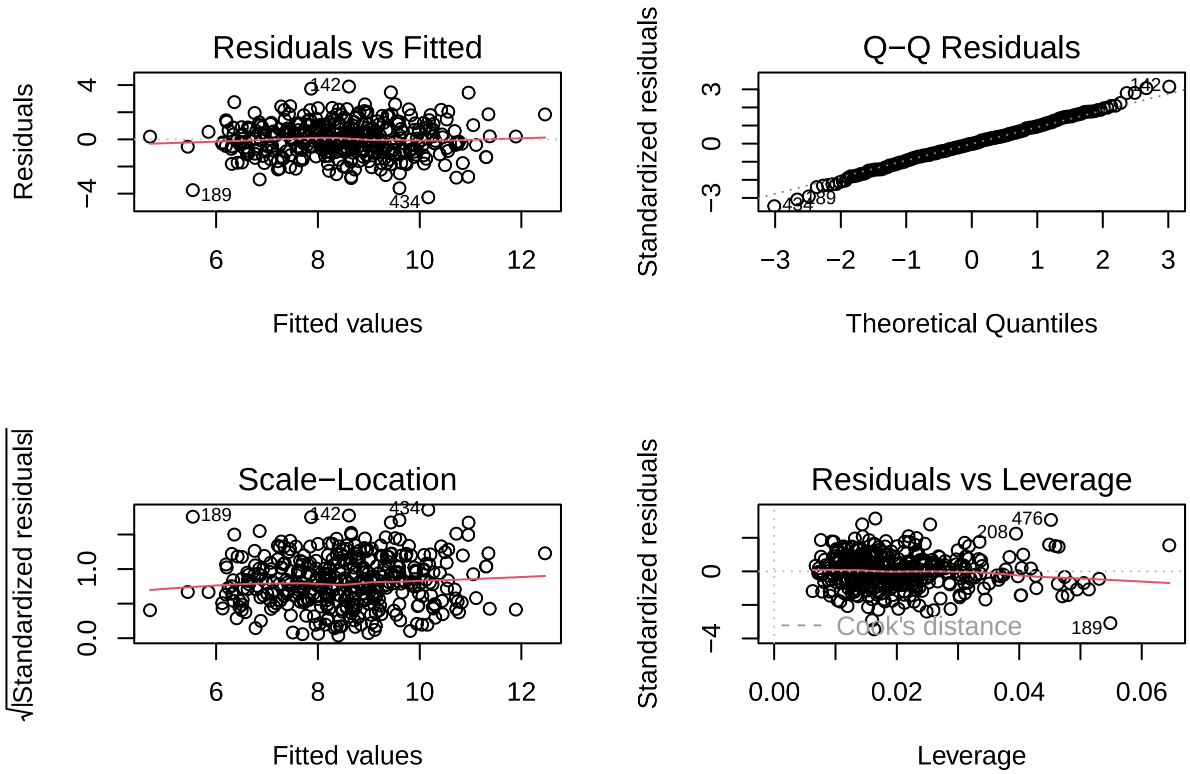
- For model1, potential outliers are observations with standardized residuals greater than 3 in absolute value. In this case, observations 189 and 476 are identified as possible outliers.

- There are no points outside the cook's distance, hence there are no leverage points in the plot.

Overall, there are no observations that influences our model,hence no need to exclude any.

MLR_Model2: Multiple Linear Regression Model created using features selected through Backward Selection

```
par(mfrow = c(2, 2))
plot(MLR_Model2)
```



Following are the inferences made:

1. Residual vs Fitted graph:
   - The range of the y-axis is currently between -4 to 4, which indicates the existence of outliers. Ideally, it should be between -2 to 2. However, despite the presence of outliers, the deviation of the line from the fitted dotted line (representing zero residual) has not been altered significantly. As a result, Model2 meets the assumptions of linearity and homoscedasticity quite well.
2. Q-Q Residuals Plot:
   - With the exception of a few countable observations, all other observations align closely along the 45-degree line on the QQ-plot. Therefore, it is reasonable to assume that normality is maintained in this case.
3. Scale-Location plot:

- The line is almost horizontal satisfying assumptions of equal variance(homoscedasticity).

4. Residual vs Leverage:

- For model2, potential outliers are observations with standardized residuals greater than 3 in absolute value. In this case, observations 189, 208 and 476 are identified as possible outliers.
- There are no points outside the cook's distance, hence there are no leverage points in the plot.

Overall, there are no observations that influences our model,hence no need to exclude any.

PART 5 :Final Recommendation - Multivariate 1

Question 1: Which of the two models from step 4 should be used and why. After comparing Model 1 and Model 2, it is clear that **Model 2** is the better option. Although it is challenging to make conclusions based on Regression diagnostics alone, when using Statistics to evaluate the Regression model, it becomes apparent that Model 2 had a higher F-statistics value and R-squared value, and all independent variables had p-values of T-Statistics that were less than 0.05.