Assign04

Shivam

2023-03-24

```
knitr::opts_chunk$set(echo = TRUE)
```

```
#if(!is.null(dev.list())) dev.off()
#cat("\014")
#rm(list=ls())
options(scipen=9)
```

-Loading the required packages for performing all the tasks.

```
if(!require(pastecs)){install.packages("pastecs")}
```

```
## Loading required package: pastecs
```

```
library("pastecs")

if(!require(lattice)){install.packages("lattice")}
```

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

```
library("lattice")

if(!require(corrgram)){install.packages("corrgram")}
```

```
## Loading required package: corrgram

##
## Attaching package: 'corrgram'

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

```
library("corrgram")
```

- Clearing the console and the whole workplace,to start with neat and clean workplace .

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

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

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

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

```r
#Setting the work directory
setwd("C:/Users/holys/OneDrive/Desktop/Data Analytics,Mathamatics,Algor/Assign04")
```

---

1 Preliminary and Exploratory

```r
dataset_SJ <- read.table("PROG8430_Assign04_23W.txt", header = TRUE, sep = ",")

dataset_SJ[0:15,]     #Printing first 5 data points from the database
```

```
##       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
## 7    8.2 390  6 13  655  C  N  Sup Del   7
## 8    9.4 252  2  8 1367  I  N  Sup Del   6
## 9    9.3 355  4  2  675  C  N  Sup Del  30
## 10   9.7 159  1 12  888  C  N  Sup Del 177
## 11   8.9 246  1  7  548  C  H Def Post  13
## 12   7.4 377  3 17  287  I  N Def Post  65
## 13   6.3 248  1  1 1286  C  N  Sup Del 137
## 14   8.2 368  3  7 1055  I  N Def Post 216
## 15   6.0 337  2  5  655  I  N Def Post 128
```

```r
                      #to make sure it looks correct

str(dataset_SJ)
```

```
## 'data.frame':    487 obs. of  9 variables:
##  $ DL: num  8.1 8.4 8.6 11.3 5.4 9.4 8.2 9.4 9.3 9.7 ...
##  $ VN: int  324 135 391 245 321 397 390 252 355 159 ...
##  $ PG: int  5 2 3 6 1 2 6 2 4 1 ...
##  $ CS: int  13 13 12 7 2 8 13 8 2 12 ...
##  $ ML: int  313 830 304 1258 221 1002 655 1367 675 888 ...
##  $ DM: chr  "C" "I" "C" "C" ...
##  $ HZ: chr  "N" "N" "N" "N" ...
##  $ CR: chr  "Sup Del" "Sup Del" "Sup Del" "Sup Del" ...
##  $ WT: num  216 160 25 67 14 47 7 6 30 177 ...
```

1. Rename all variables with your initials appended (just as was done in
   assignment 1,2 and 3)

```r
```

```
colnames(dataset_SJ) <- paste(colnames(dataset_SJ), "SJ", sep = "_")  # renaming all the
                                                                      # variables and adding
                                                                      # my initials SJ

  head(dataset_SJ)
```

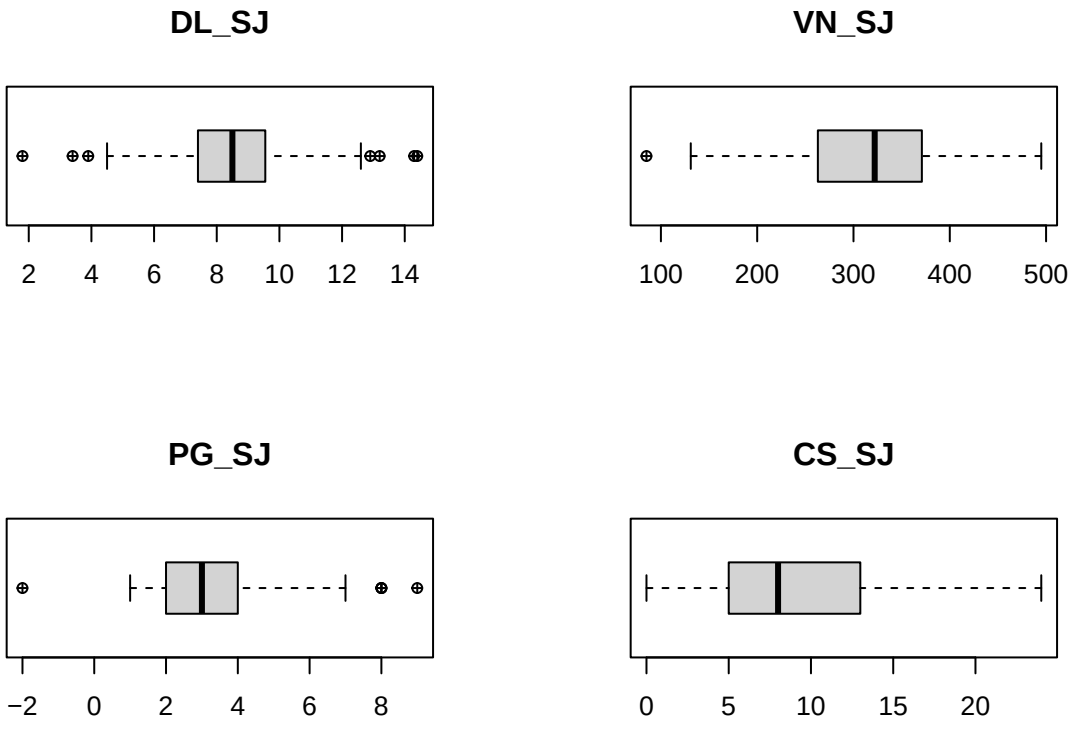
```
##   DL_SJ VN_SJ PG_SJ CS_SJ ML_SJ DM_SJ HZ_SJ    CR_SJ WT_SJ
## 1   8.1   324     5    13   313     C     N  Sup Del   216
## 2   8.4   135     2    13   830     I     N  Sup Del   160
## 3   8.6   391     3    12   304     C     N  Sup Del    25
## 4  11.3   245     6     7  1258     C     N  Sup 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
```

```r
# Changed all the character variables to factors as to use them
#to plot and examine with the help of code given in the announcement

dataset_SJ <- as.data.frame(unclass(dataset_SJ), stringsAsFactors = TRUE)
```

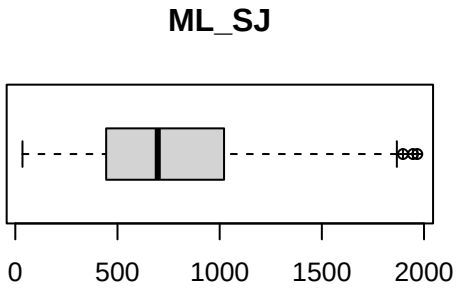
2. Examine the data using the exploratory techniques we have learned in class. Does the data look reasonable? Are there any outliers? If so, deal with them appropriately.

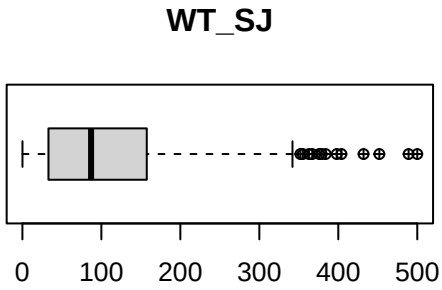
- Studying the data set, and going through each fields, which is giving information of the a mail order company, which tracks the time it takes for a customer to receive his/her order.The DL field represents the time taken for delivery in days, VN tells about how long the product has been in warehouse, PG tells about how many packages are made of the product which has been ordered, CS gives information about the customers and shows how many orders he/she had made in past, ML tells about the distance in km one have to travel to deliver the order, there are some fields which gives
  information regarding the individual products like whether the product is manufactured in Canada or elsewhere, indicating if the product is Hazardous(H) or not(N).CR indicates which type of Carrier delivered the item (Def Post, or Sup Del)and WT tells the weight of the shipment.
- lets plot box-plot to see the distribution of data points, and see if it has any outliers.

```
 par(mfrow=c(2,2))                                    #defining format of the boxplot

for (i in 1:ncol(dataset_SJ)) {                       # Generating box plot for each
                                                      # variable in th data set using
                                                      # for loop
  if (is.numeric(dataset_SJ[,i])) {
      boxplot(dataset_SJ[i], main= names(dataset_SJ)[i],
              horizontal=TRUE, pch=10)
  }
}
```



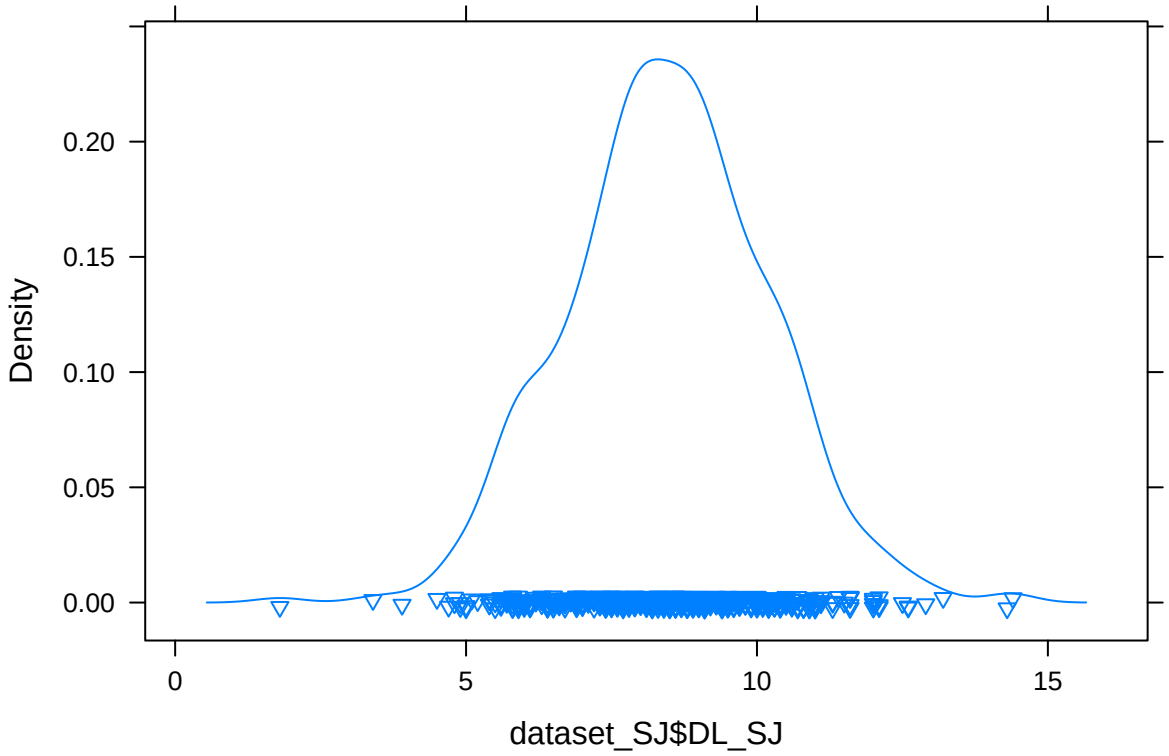

```
par(mfrow=c(1,1))
```



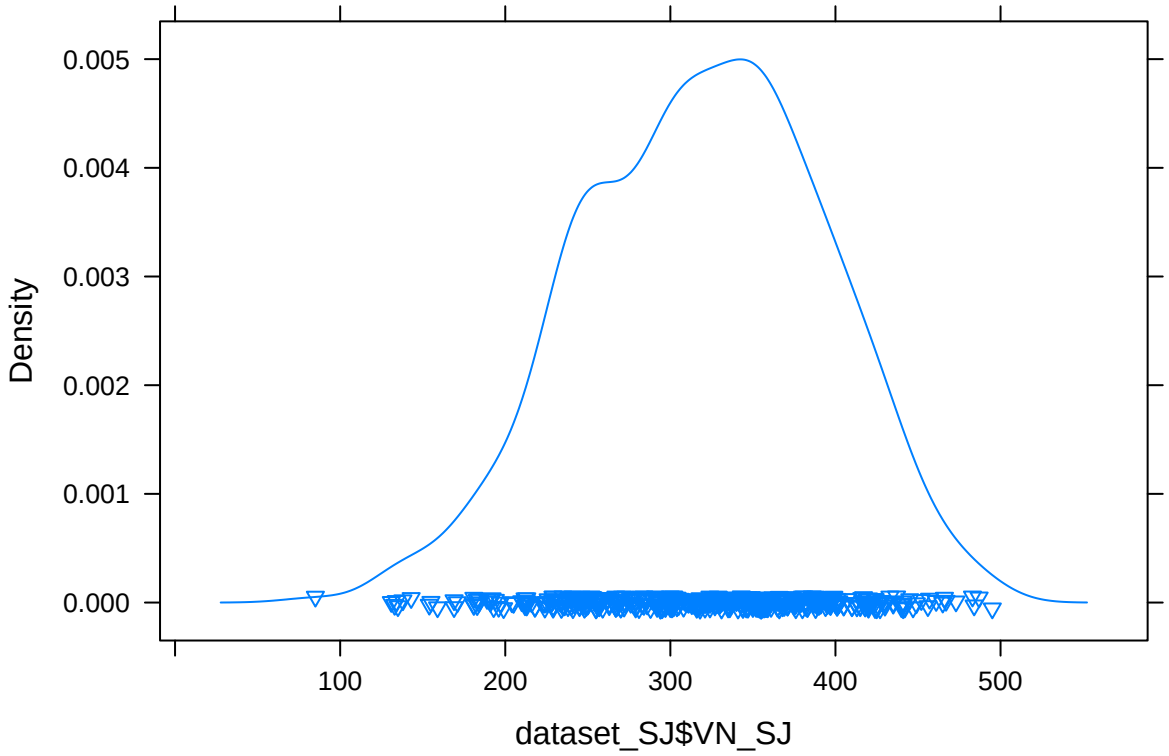


- I observed how the data is distributed in different fields and do any of the fields have outliers or un-important data. The data is distributed properly and some of the fields we have some extreme data points but still they are in the range and are reasonable enough to be considered in a data set except one data point which is negative in the field PG that is packages , which is
impossible to have value of -2, it clearly signifies that something is fishy going
there!lets make density plots for more clarity and to get details.

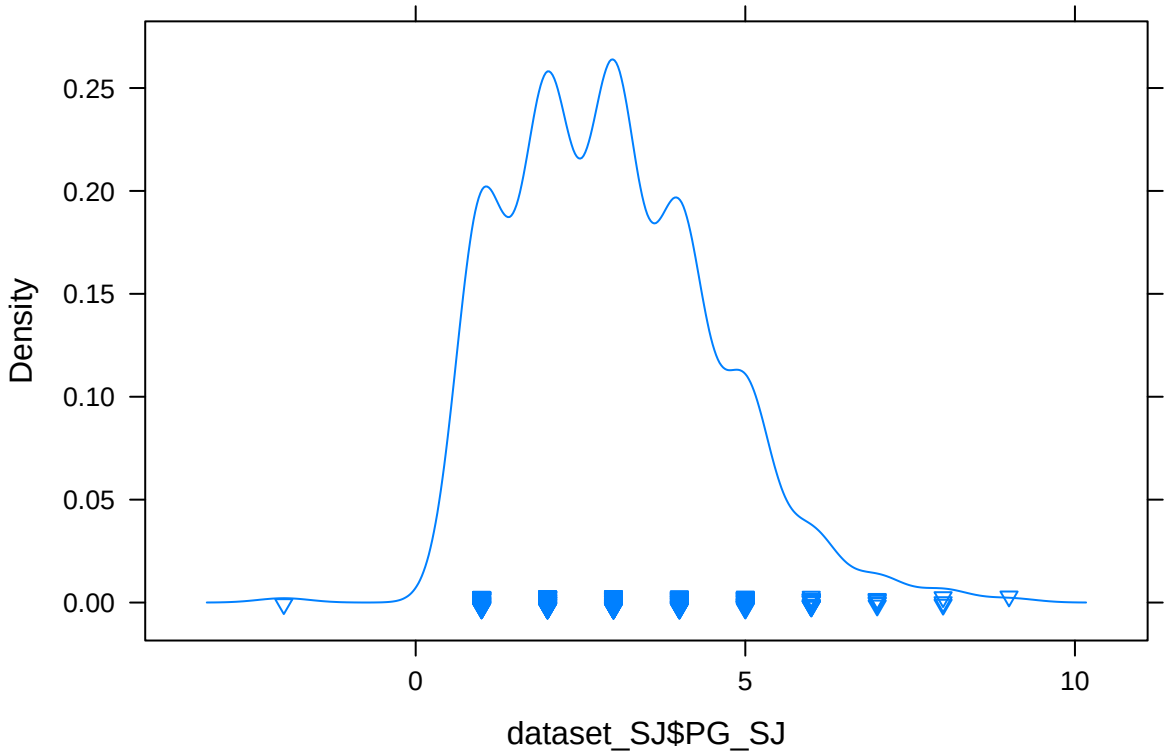
```
# plotting density plots for graphically
#analyzing data in detailed way of standardized variables
densityplot( ~ dataset_SJ$DL_SJ, pch=6)
```



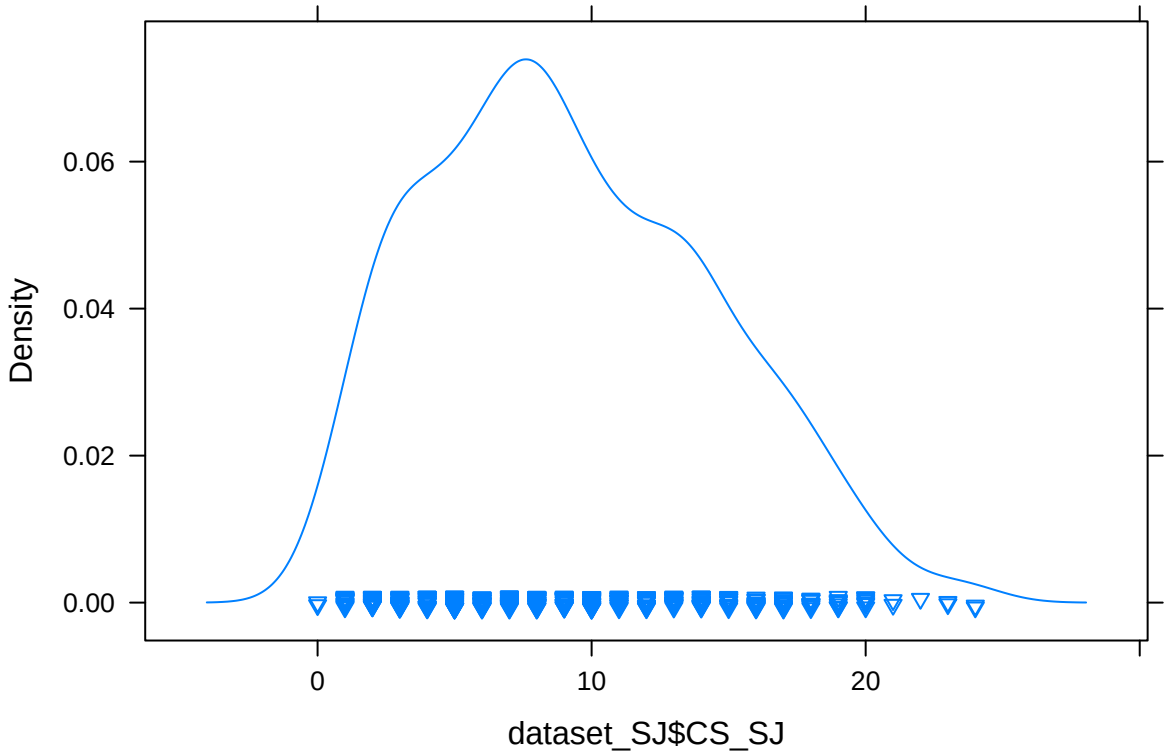

```
densityplot( ~ dataset_SJ$VN_SJ, pch=6)
```



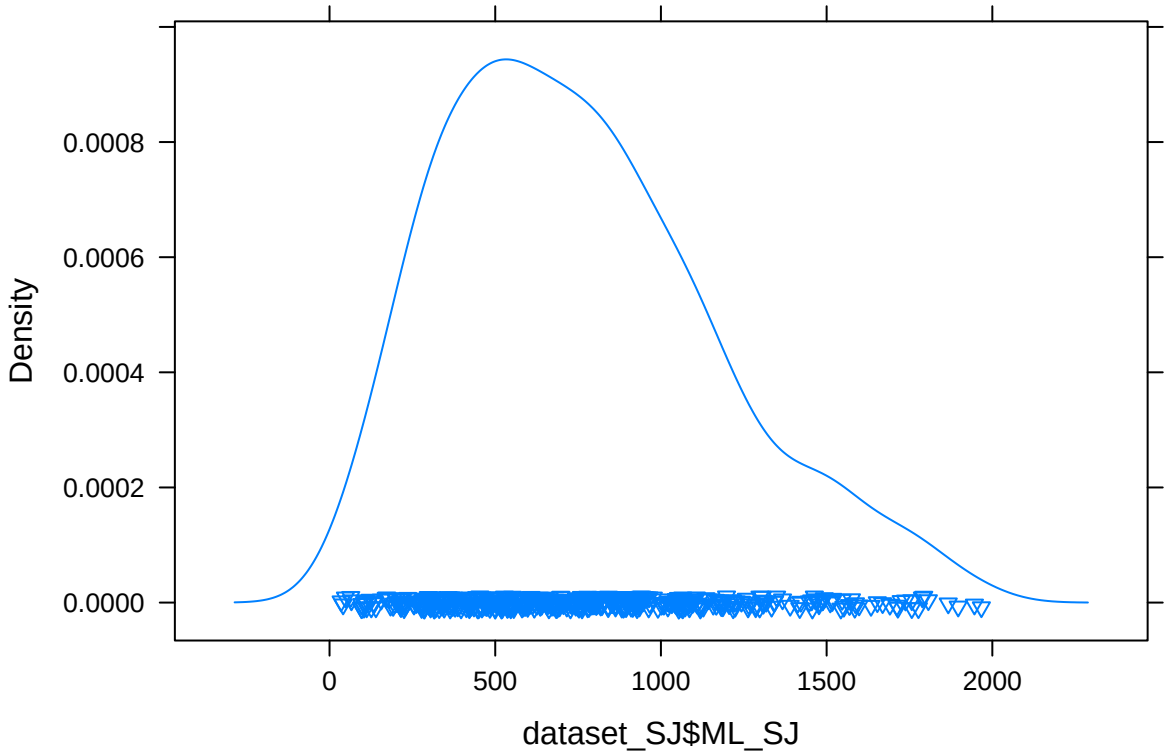

```
densityplot( ~ dataset_SJ$PG_SJ, pch=6)
```



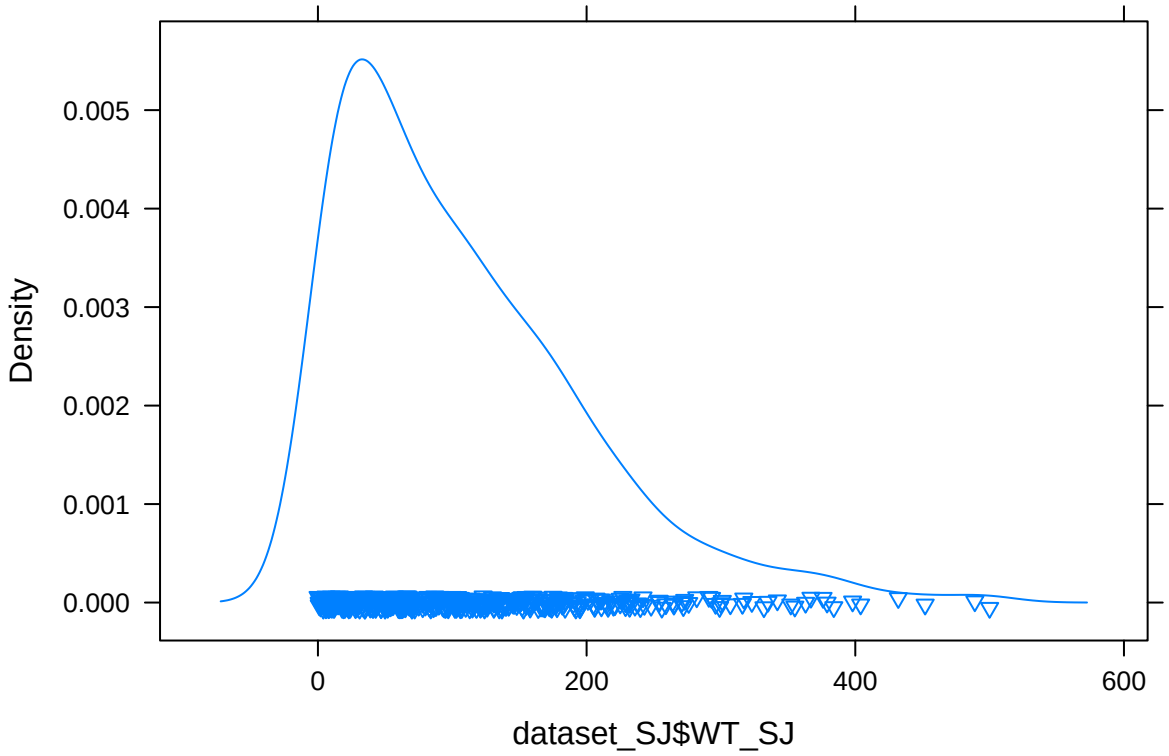

```
densityplot( ~ dataset_SJ$CS_SJ, pch=6)
```




```
densityplot( ~ dataset_SJ$ML_SJ, pch=6)
```

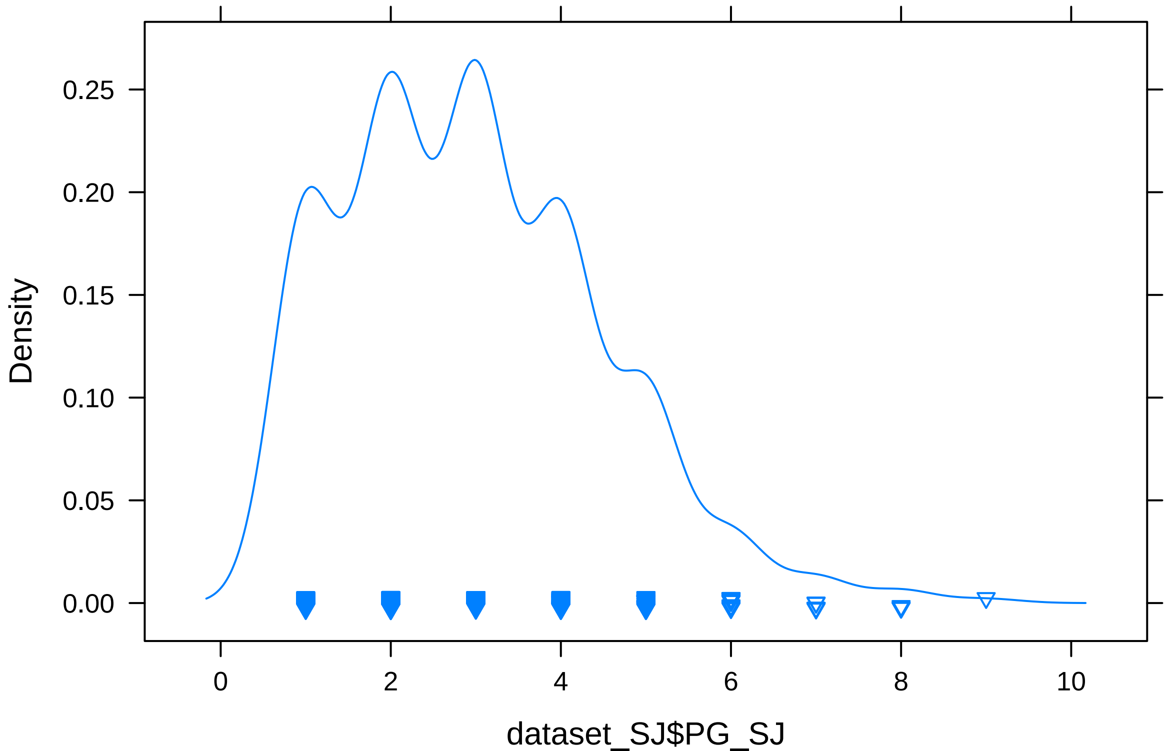



```
densityplot( ~ dataset_SJ$WT_SJ, pch=6)
```



-I can clearly see that the PG field have a negative observation which does not make any sense as PG refers to the number of packages of product that have been ordered, and that being negative is impossible unless it is a mistake or wrong interpretation. No product can have number of packages equal to -2. So, its a meaning less data point which should be removed.

```
# Removing the outlier


outlier_In_PG_SJ <- which(dataset_SJ$PG_SJ < 0) ##Fond row number with PG
                                                 #less then to 0
dataset_SJ <- dataset_SJ[-c(outlier_In_PG_SJ),]   #deleted the row
densityplot( ~ dataset_SJ$PG_SJ, pch=6)
```



~~~~~~ 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.

```
- As DL shows the delivery time and Carriers shows the type of carriers, if
I want to know of any of the Carrier has faster delivery time then others,
for that lets check the mean value of both the different kinds of carriers
and see if any of the one has greater value then the other.As their are two
variable,we will conduct t-test.For that we have to perform t-test to get the
mean values.
```

```
shapiro.test(dataset_SJ$DL_SJ)        # To check the normality First before
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dataset_SJ$DL_SJ
## W = 0.9964, p-value = 0.3443
```

```
                                      #performing t test to match the assumptions
```

- As we can see the p value is approx 0.33 which is more then 0.05, so i can conclude that the data set is normally distributed.
  - Also both the variables, delivery time and carriers are independent,because the type of carrier is s dependent on the type of product., similarly the delivery time is dependent on the distance
~~~~~~

where the order have to be delivered or to the weight of the ordered product and not dependent to the package type.

-Hence, as I understood that the variables are independent and the distributed normally, I will perform t-test for the two variables DL and CR.

```
#Performing t test of Carriers with their Delivery time
t.test(dataset_SJ$DL_SJ ~ dataset_SJ$CR_SJ)
```

```
##
##  Welch Two Sample t-test
##
## data:  dataset_SJ$DL_SJ by dataset_SJ$CR_SJ
## t = -6.9608, df = 440.63, p-value = 0.00000000001234
## alternative hypothesis: true difference in means between group Def Post and group Sup Del is not equa
## 95 percent confidence interval:
##  -1.3525268 -0.7569259
## sample estimates:
## mean in group Def Post  mean in group Sup Del
##               7.845274               8.900000
```

-As clearly observed from the t-test, there is significant variance between both the means of different groups of the field CR, its approx 7.84 for Def Post and its 8.9 for Sup Del. I can say that that Def Post Carrier delivers approx 1.05 times faster then Sup Del Carrier , so now I can conclude with evidence that one of the group that is Def Post has faster delivery times.

~~~~~~ 4. As demonstrated in class, split the data frame 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.

- Splitting the data set into two parts as learned in the classes, one train set and the other test set. I kept the sampling rate as 0.8, so splitting the data into 80/20 ratio. Also set the seed as 9647 according to the last four digits of my student number.

```
## [1] 8.4638
```

```
## [1] 8.423711
```

```
## [1] 8.622449
```

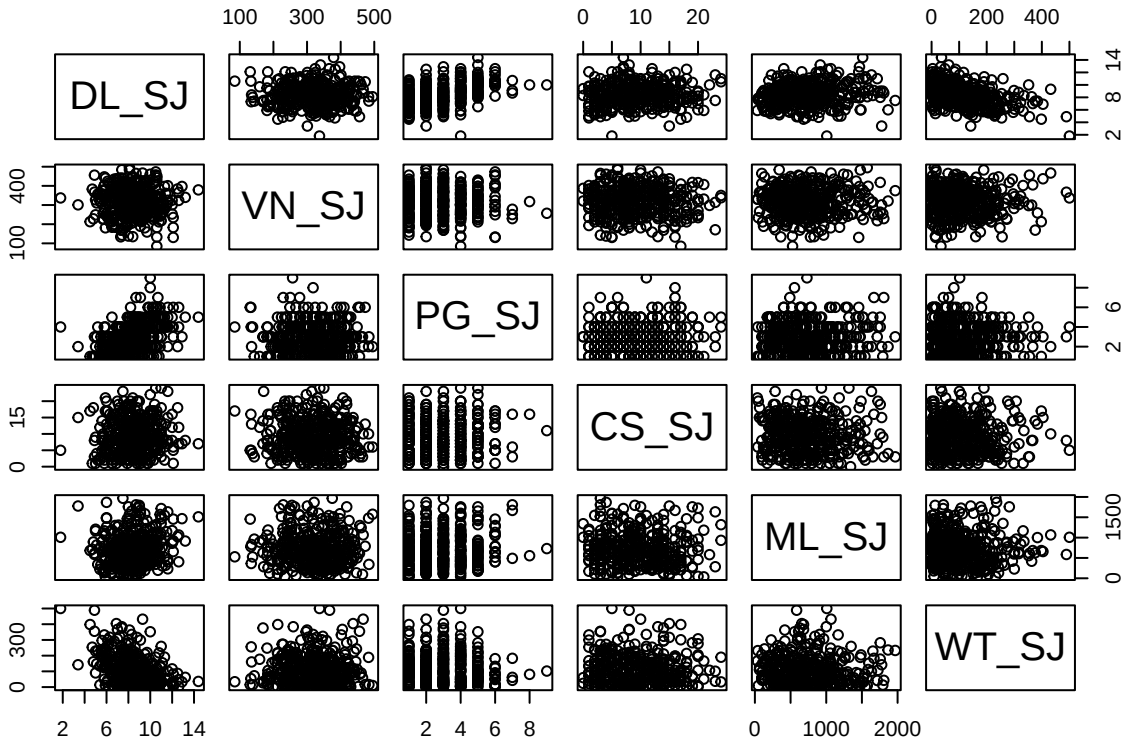
---

2 Simple Linear Regression

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?

-As the three fields, DM, HZ and CR are factor variables, they represent the information of the products like where it is manufactured, whether it is hazardous or not and in what type of carrier the order is transported into respectively. It does no have any numerical data, so such data cannot give results with plotting functions, so I will omit it and perform pairs() function and cor() function to see the relationships of different fields.
~~~~~~

```
train_cor_SJ <- train_SJ[-c(6:8)]    #Removing factor variables for performing
                                     #correlation functions

pairs(train_cor_SJ)
```



- From the result above, I can clearly sat that all the fields have different relationships with each other, all data points show appropriate observations and there's nothing not normal or something to get concerned for in the data. Some of the variables like DL and PG moderate positive relationship between them, also DL has moderate negative relationship. We can also see the variables having very weak relations with the variable DL, for instance we can see from the matrix that VN that is vintage or the time the product has been in the warehouse, has no relation with the delivery time.

Lets get the numeric coefficients of correlation by between different variables to understand the relationship in a better way.

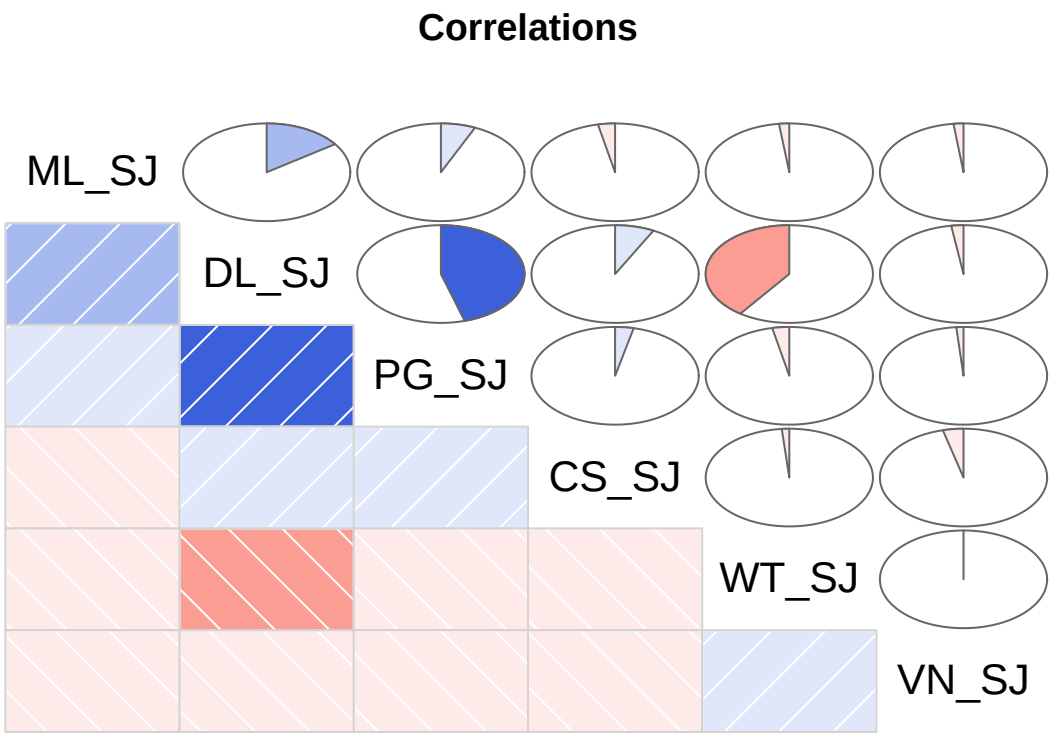
- Performing cor() function to get numerical correlation coefficients:

```
Corr_SJ <- cor(train_cor_SJ)
round(Corr_SJ,2)
```

```
##       DL_SJ VN_SJ PG_SJ CS_SJ ML_SJ WT_SJ
## DL_SJ  1.00 -0.02  0.45  0.08  0.15 -0.40
## VN_SJ -0.02  1.00 -0.01 -0.04 -0.02  0.00
## PG_SJ  0.45 -0.01  1.00  0.04  0.07 -0.03
## CS_SJ  0.08 -0.04  0.04  1.00 -0.03 -0.01
## ML_SJ  0.15 -0.02  0.07 -0.03  1.00 -0.02
## WT_SJ -0.40  0.00 -0.03 -0.01 -0.02  1.00
```

This shows as the weight increases the delivery time also get affected by -0.4 and the delivery time has a moderate positive relation with the number of packages with the correlation coefficient of 0.45. The distance the order need to travel to get delivered also has a weak positive relationship with the delivery time with the coefficient of 0.15. Similarly, I can obtain the same results by using the corrgram function from corrgram package. * -Graphical representation by using corrogram function

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



- As it is observed, the blue colour signifies the positive relationship and the red colour shows the negative correlation between variables. The brightness or the shade of the colour shows the strength of the relationship, darker the colour, stronger the relation and lighter the colour of the shade weaker the relation. As observed, DL and Pg has a moderate positive relationship which I can infer from the dark blue colour on the intersection of the two variables. The percentage of relationship can can be observed from the pie chart besides the variables labeled . As, there is negative moderate relationship of coefficient 0.4 between the weight of shipment and delivery time we can see a little bit darker tone of red to signify the relationship as well as the strength of it.

---

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.

-Lets make a simple linear model named delivery_weight_model_SJ between the two variables signifying

delivery time and weight of the shipment that is DL and WT respectively. Here, DL is the Dependent variable and weight is independent.

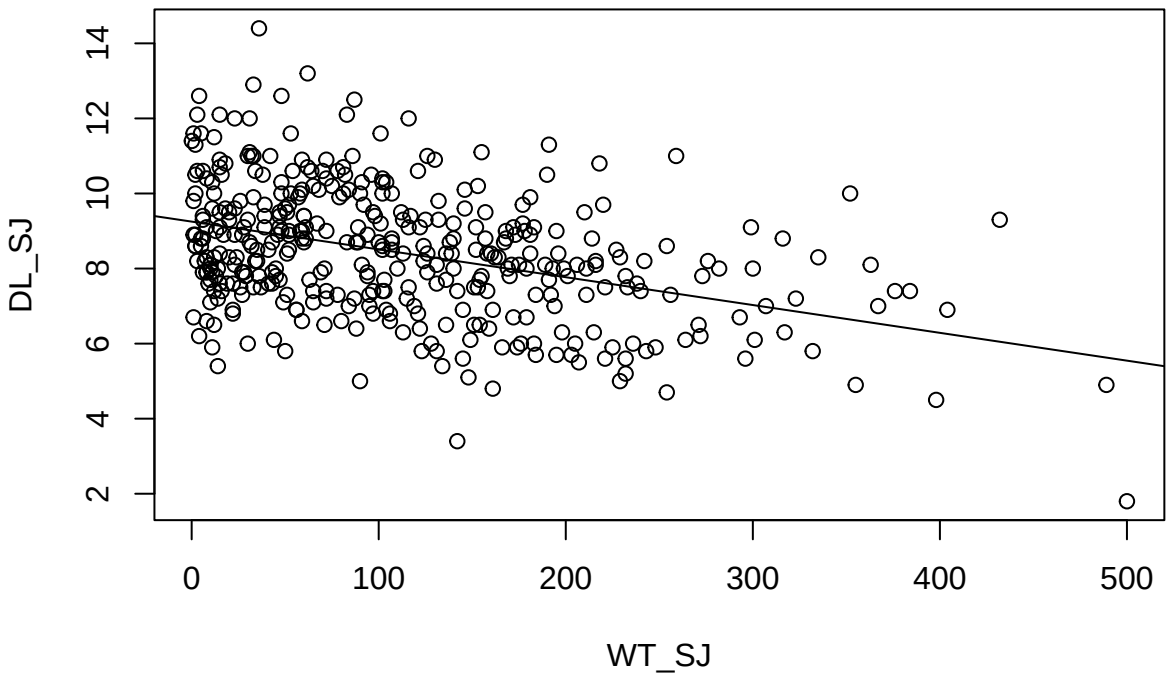
```
delivery_weight_model_SJ <- lm(DL_SJ ~ WT_SJ, data=train_SJ)
delivery_weight_model_SJ
```

```
##
## Call:
## lm(formula = DL_SJ ~ WT_SJ, data = train_SJ)
##
## Coefficients:
## (Intercept)        WT_SJ
##     9.25105     -0.00741
```

```
summary(delivery_weight_model_SJ)     #Observing the results and the model.
```

```
##
## Call:
## lm(formula = DL_SJ ~ WT_SJ, data = train_SJ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7988 -1.2275  0.0492  1.0688  5.4157
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.2510476  0.1259728  73.437   <2e-16 ***
## WT_SJ       -0.0074104  0.0008628  -8.589   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.599 on 386 degrees of freedom
## Multiple R-squared:  0.1604, Adjusted R-squared:  0.1583
## F-statistic: 73.77 on 1 and 386 DF,  p-value: < 2.2e-16
```

```
# Plotting the scatter plot to get the regression line
plot(DL_SJ ~ WT_SJ, data= train_SJ,
     main="Delivery time by Weight of shipment (with Regression Line)")
abline(delivery_weight_model_SJ)
```



- From the plot, the regression line, I can infer that all the data points are scattered in the specified range, and the line is inclined downwards due to some of the data points having high weight and low delivery time. Overall, it is a good dataset having a proper regression line.

---

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.

-Creating a simple linear model named delivery_distance_model_SJ between the two variables signifying delivery time and Distance the order needs to be delivered (in km)that is DL and ML respectively. Here, DL is the Dependent variable and ML is independent.

```
delivery_distance_model_SJ <- lm(DL_SJ ~ ML_SJ, data=train_SJ)
delivery_distance_model_SJ
```

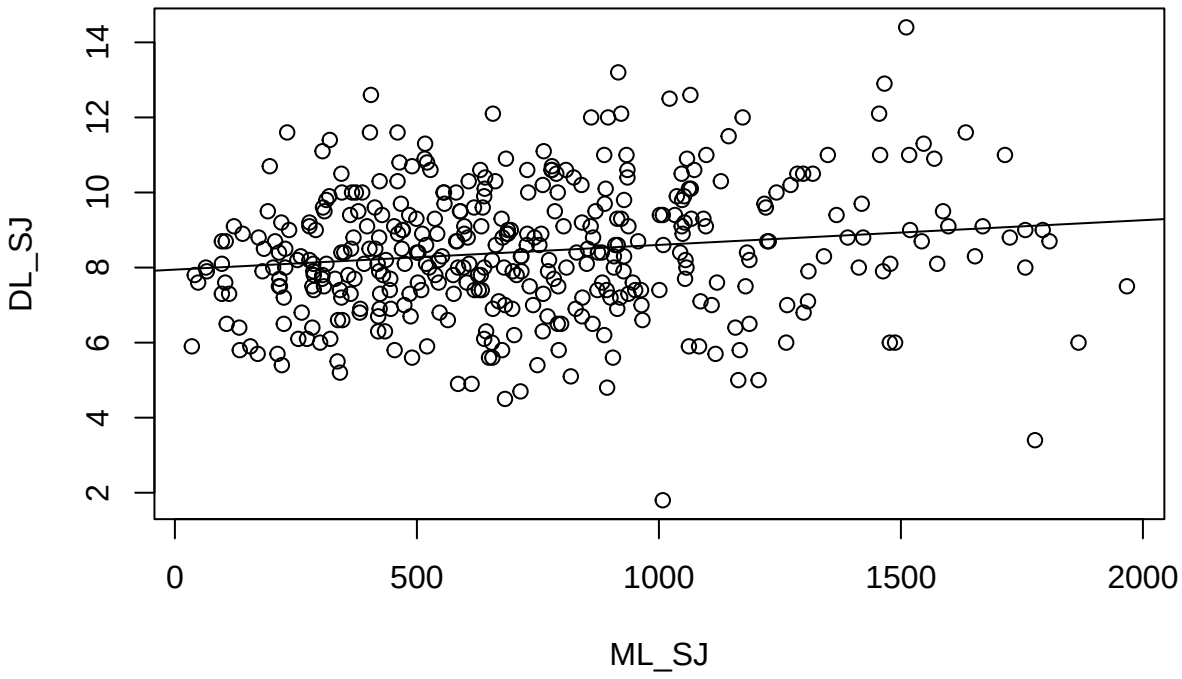
```
##
## Call:
## lm(formula = DL_SJ ~ ML_SJ, data = train_SJ)
##
## Coefficients:
## (Intercept)        ML_SJ
##   7.9443753    0.0006588
```

```r
summary(delivery_distance_model_SJ)  #Observing the results and the model
```

```
##
## Call:
## lm(formula = DL_SJ ~ ML_SJ, data = train_SJ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8085 -1.0872 -0.0691  1.1656  5.4601
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.9443753  0.1809196  43.911  < 2e-16 ***
## ML_SJ       0.0006588  0.0002176   3.028  0.00263 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.725 on 386 degrees of freedom
## Multiple R-squared:  0.0232, Adjusted R-squared:  0.02067
## F-statistic: 9.167 on 1 and 386 DF,  p-value: 0.00263
```

```r
# Plotting the scatter plot to get the regression line
plot(DL_SJ ~ ML_SJ, data= train_SJ,
     main="Delivery time by distance the order needs to be delivered (with Regression Line)")
abline(delivery_distance_model_SJ)
```

livery time by distance the order needs to be delivered (with Regressio



- From the plot, the regression line, I can infer that all the data points are scattered in the specified range and distributed near the mean, that is, the orders are delivered in near about 8 hours for the orders having range of distance from 0 km to 2000 km,the line of regression is consistent throughout
the dataset. Overall,it is a good dataset having a proper regression line.

---

4. As demonstrated in class, compare the models (F-Stat, R^2, RMSE for train and test, etc.) Which model is superior? Why?

```
summary(delivery_weight_model_SJ)       # for obtaining the F- stat, residuals,
```

```
##
## Call:
## lm(formula = DL_SJ ~ WT_SJ, data = train_SJ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7988 -1.2275  0.0492  1.0688  5.4157
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.2510476  0.1259728  73.437   <2e-16 ***
## WT_SJ       -0.0074104  0.0008628  -8.589   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.599 on 386 degrees of freedom
## Multiple R-squared:  0.1604, Adjusted R-squared:  0.1583
## F-statistic: 73.77 on 1 and 386 DF,  p-value: < 2.2e-16
```

```
                                        # Adjusted R-squared lest get the summary
                                        #  for the weight model

# Made a prediction model and calculated the RMSE for train set
pred_weight_SJ <- predict(delivery_weight_model_SJ, newdata=train_SJ)
RMSE_weight_trn_SJ <- sqrt(mean((train_SJ$DL_SJ - pred_weight_SJ)^2))
round(RMSE_weight_trn_SJ,3)
```

```
## [1] 1.595
```

```
# Made a prediction model and calculated the RMSE for test set
pred_weight_test_SJ <- predict(delivery_weight_model_SJ, newdata=test_SJ)
RMSE_weight_tst_SJ <- sqrt(mean((test_SJ$DL_SJ - pred_weight_test_SJ)^2))
round(RMSE_weight_tst_SJ,3)
```

```
## [1] 1.629
```

- Observing the summary of the model, I can infer that the residuals are satisfying that is the median is near to zero and between 1Q and 3Q same which are also similar similar just having different sign. The model also passes the f-test, as the f-stat value is less then 0.05, Adjusted R^2 value is 0.1583. While, comparing the RMSE for both the train and test set,that is 1.595 and 1.629 respectively, they are similar which tells that the model is good,and not under-fitting or over-fitting. It also passes the t-test with p-value less then 0.05,the coefficients are also consistent.

```
summary(delivery_distance_model_SJ)    # for obtaining the F- stat, residuals,
```

```
##
## Call:
## lm(formula = DL_SJ ~ ML_SJ, data = train_SJ)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8085 -1.0872 -0.0691  1.1656  5.4601
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.9443753  0.1809196  43.911  < 2e-16 ***
## ML_SJ       0.0006588  0.0002176   3.028  0.00263 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.725 on 386 degrees of freedom
## Multiple R-squared:  0.0232, Adjusted R-squared:  0.02067
## F-statistic: 9.167 on 1 and 386 DF,  p-value: 0.00263
```

```
                                    # Adjusted R-squared lest get the summary
                                    #  for the distance model

# Made a prediction model and calculated the RMSE for train set
pred_distance_SJ <- predict(delivery_distance_model_SJ, newdata=train_SJ)
RMSE_distance_trn_SJ <- sqrt(mean((train_SJ$DL_SJ - pred_distance_SJ)^2))
round(RMSE_distance_trn_SJ,3)
```

```
## [1] 1.72
```

```
# Made a prediction model and calculated the RMSE for test set
pred_distance_test_SJ <- predict(delivery_distance_model_SJ, newdata=test_SJ)
RMSE_distance_tst_SJ <- sqrt(mean((test_SJ$DL_SJ - pred_distance_test_SJ)^2))
round(RMSE_weight_tst_SJ,3)
```

```
## [1] 1.629
```

- Observing the summary of the model, I can infer that the residuals are satisfying that is the median is near to zero and between 1Q and 3Q same which are also similar similar just having different sign. The model also passes the f-test , as the f-stat value is less then 0.05, Adjusted R^2 value is 0.02067. While, comparing the RMSE for both the train and test set,that is 1.72 and 1.692 respectively, they are similar which tells that the model is good,and not under-fitting or over-fitting. It also passes the t-test with p-value less then 0.05,the coefficients are also consistent.
- As all the aspects like Residuals, F-stat, RMSE are giving similar and satisfactory results in both the models and only the adjusted R^2 value differs, now as thought in the lectures, I learned that higher the value of adjusted R^2
  better the model.

So, here I would suggest that the delivery_weight_model_SJ is better which has DL and WT as variables as compared to the model having distance, i.e;delivery_distance_model_SJ.

3 Model Development – Multivariate

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).(Your commentary should be yours, not simply copied from my example).

- Creating the model, using all the variables, that is full model named full_model_SJ, after that to get the stats and information about the residuals, t test, f test, adjusted R^2 values, we will peerform the summmary function.
We will buit predicted models for both the train and test sets and will calculate the RMSE values,and check how the model is.

```
########################################
## Creating Baseline/Full Model        ##
########################################

full_model_SJ = lm( DL_SJ ~ . ,
            data=train_SJ, na.action=na.omit)
summary(full_model_SJ)
```

```
##
## Call:
## lm(formula = DL_SJ ~ ., data = train_SJ, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0182 -0.7399 -0.0194  0.8059  4.0091
##
## Coefficients:
##                Estimate Std. Error t value          Pr(>|t|)
## (Intercept)   7.1321845  0.4111176  17.348           < 2e-16 ***
## VN_SJ        -0.0002326  0.0008735  -0.266          0.790184
## PG_SJ         0.5350046  0.0444819  12.027           < 2e-16 ***
## CS_SJ         0.0131028  0.0125090   1.047          0.295547
## ML_SJ         0.0004091  0.0001618   2.529          0.011851 *
## DM_SJI        0.4396536  0.1430752   3.073          0.002273 **
## HZ_SJN       -0.6903554  0.1906334  -3.621          0.000333 ***
## CR_SJSup Del  0.9864613  0.1323629   7.453 0.000000000000626 ***
## WT_SJ        -0.0065745  0.0006902  -9.526           < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.273 on 379 degrees of freedom
## Multiple R-squared:  0.4776, Adjusted R-squared:  0.4665
## F-statistic: 43.31 on 8 and 379 DF,  p-value: < 2.2e-16
```

```
# Made a prediction model and calculated the RMSE for train set
pred_full_SJ <- predict(full_model_SJ, newdata=train_SJ)
RMSE_trn_full_SJ <- sqrt(mean((train_SJ$DL_SJ - pred_full_SJ)^2))
round(RMSE_trn_full_SJ,2)
```

```
## [1] 1.26
```

```
# Made a prediction model and calculated the RMSE for test set
pred_full_SJ <- predict(full_model_SJ, newdata=test_SJ)
RMSE_test_full_SJ <- sqrt(mean((test_SJ$DL_SJ - pred_full_SJ)^2))
round(RMSE_test_full_SJ,2)
```

```
## [1] 1.14
```

- After creating the full model, lets create another model while performing backward selection.

```
########################################
## Creating Backward Selection Model   ##
########################################

back_model_SJ = step(full_model_SJ, direction="backward", details=TRUE)
```

```
## Start:  AIC=196.22
## DL_SJ ~ VN_SJ + PG_SJ + CS_SJ + ML_SJ + DM_SJ + HZ_SJ + CR_SJ +
##     WT_SJ
##
##         Df Sum of Sq    RSS    AIC
## - VN_SJ  1     0.115 614.32 194.29
## - CS_SJ  1     1.778 615.99 195.34
## <none>               614.21 196.22
## - ML_SJ  1    10.363 624.57 200.71
## - DM_SJ  1    15.303 629.51 203.77
## - HZ_SJ  1    21.253 635.46 207.42
## - CR_SJ  1    90.013 704.22 247.28
## - WT_SJ  1   147.054 761.26 277.50
## - PG_SJ  1   234.437 848.64 319.66
##
## Step:  AIC=194.29
## DL_SJ ~ PG_SJ + CS_SJ + ML_SJ + DM_SJ + HZ_SJ + CR_SJ + WT_SJ
##
##         Df Sum of Sq    RSS    AIC
## - CS_SJ  1     1.826 616.15 193.44
## <none>               614.32 194.29
## - ML_SJ  1    10.394 624.72 198.80
## - DM_SJ  1    15.190 629.51 201.77
## - HZ_SJ  1    21.266 635.59 205.50
## - CR_SJ  1    90.556 704.88 245.64
## - WT_SJ  1   147.053 761.38 275.56
## - PG_SJ  1   234.654 848.98 317.81
##
## Step:  AIC=193.44
## DL_SJ ~ PG_SJ + ML_SJ + DM_SJ + HZ_SJ + CR_SJ + WT_SJ
##
##         Df Sum of Sq    RSS    AIC
## <none>               616.15 193.44
## - ML_SJ  1    10.101 626.25 197.75
## - DM_SJ  1    16.322 632.47 201.59
```

```
## - HZ_SJ  1    21.641 637.79 204.84
## - CR_SJ  1    91.008 707.16 244.90
## - WT_SJ  1   147.374 763.52 274.65
## - PG_SJ  1   236.541 852.69 317.51
```

```
summary(back_model_SJ)
```

```
##
## Call:
## lm(formula = DL_SJ ~ PG_SJ + ML_SJ + DM_SJ + HZ_SJ + CR_SJ +
##     WT_SJ, data = train_SJ, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9306 -0.7606 -0.0177  0.7938  4.1121
##
## Coefficients:
##                Estimate Std. Error t value          Pr(>|t|)
## (Intercept)   7.1799103  0.2702107  26.572           < 2e-16 ***
## PG_SJ         0.5369585  0.0443984  12.094           < 2e-16 ***
## ML_SJ         0.0004036  0.0001615   2.499           0.01287 *
## DM_SJI        0.4498633  0.1416047   3.177           0.00161 **
## HZ_SJN       -0.6963359  0.1903534  -3.658           0.00029 ***
## CR_SJSup Del  0.9905362  0.1320419   7.502 0.000000000000448 ***
## WT_SJ        -0.0065814  0.0006894  -9.546           < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.272 on 381 degrees of freedom
## Multiple R-squared:  0.4759, Adjusted R-squared:  0.4677
## F-statistic: 57.67 on 6 and 381 DF,  p-value: < 2.2e-16
```

```
# Made a prediction model and calculated the RMSE for train set
pred_back_SJ <- predict(back_model_SJ, newdata=train_SJ)
RMSE_trn_back_SJ <- sqrt(mean((train_SJ$DL_SJ - pred_back_SJ)^2))
round(RMSE_trn_back_SJ,2)
```

```
## [1] 1.26
```

```
# Made a prediction model and calculated the RMSE for train set
pred_back_SJ <- predict(back_model_SJ, newdata=test_SJ)
RMSE_test_back_SJ <- sqrt(mean((test_SJ$DL_SJ - pred_back_SJ)^2))
round(RMSE_test_back_SJ,2)
```

```
## [1] 1.15
```

- We will take one by one all the variable and check the AIC value, the lower the AIC better the model, so in our case we took 6 variables PG_SJ, ML_SJ, DM_SJ, HZ_SJ, CR_SJ, WT_SJ to get the lowest AIC value of 193.44.

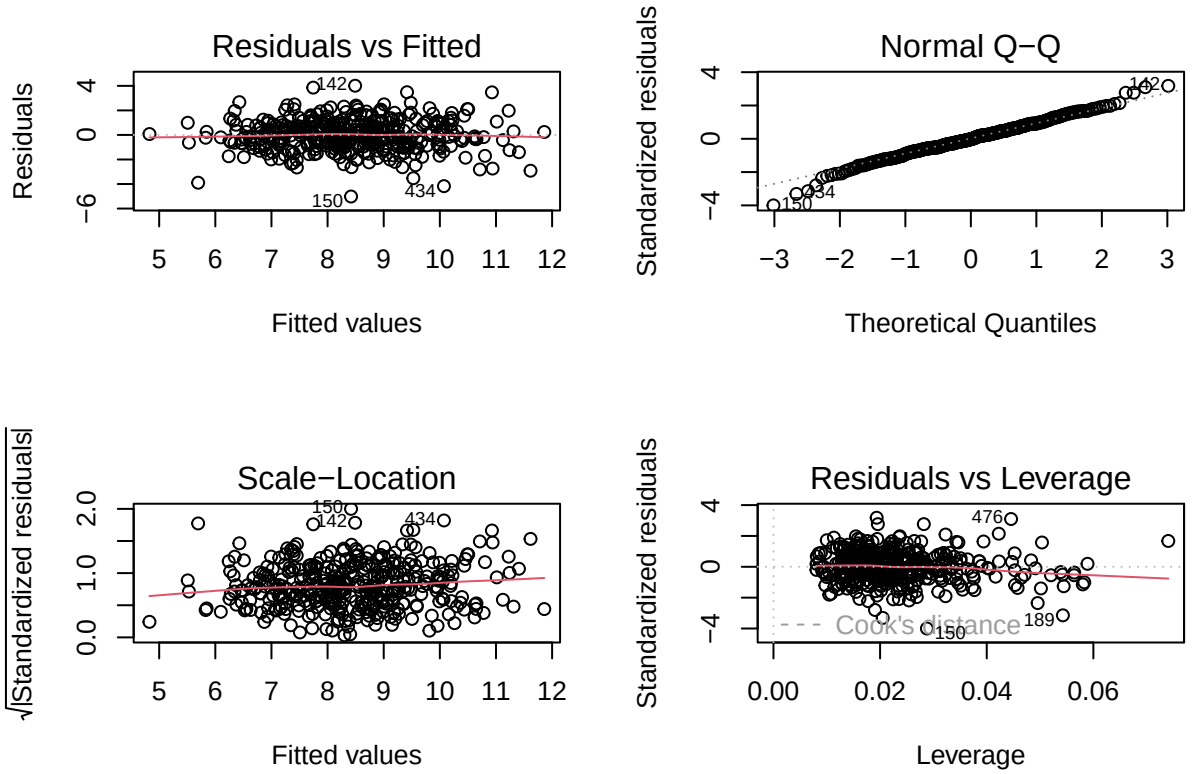
---

4 Model Evaluation – Verifying Assumptions - Multivariate

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.) - We have to check the assumptions for both the models we made above. The assumptions include : Independence of predictors, Linearity, Distribution of Error Terms, Homoscedasticity of Error Terms, Non-autocorrelation,

Lets plot the graphs to check that the model fulfill the assumptions.

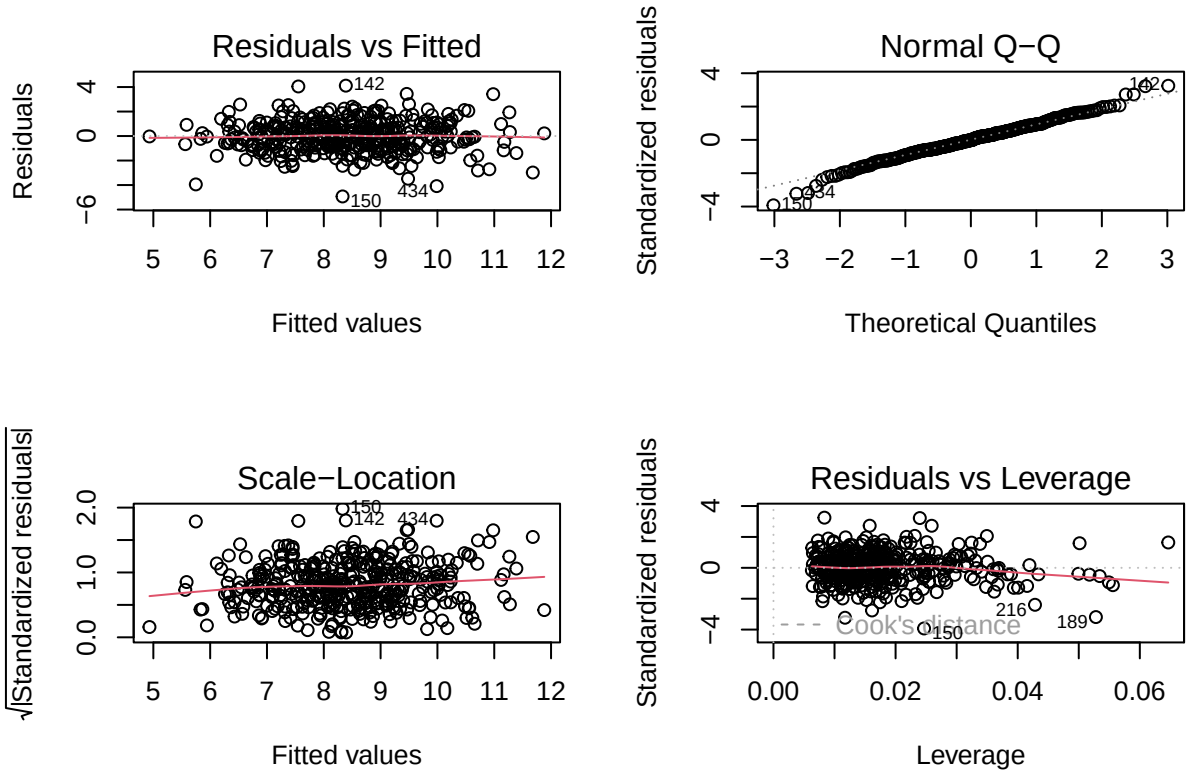
```
par(mfrow = c(2, 2))
plot(full_model_SJ)
```




```
par(mfrow = c(1, 1))
```

- It is observed that the residuals are scattered around the mean and near zero, that is it fits properly, also the QQ plot shows that the distribution is normal as it aligns the center line, I also observed that that is no pattern seen in the data points , also there is no autocorrelation observed. The variance of the errors is also constant throughout the model.No High leverage or high influence is observed. The model meets sall the assumptions of regression.

```
# plotting graphs for backward model
par(mfrow = c(2, 2))
plot(back_model_SJ)
```




```
par(mfrow = c(1, 1))
```

-Similarly, for the backward model we plotted the graphs and observed that it matches all the assumptions of regression.

---

5. Final Recommendation - Multivariate 1

Based on your preceding analysis, recommend which of the two models from step 4 should be used and why.

- Comparing both the models, I can infer that the backward model is better as it passes the t test for all variables. Unlike full model, the variables VN and CS fails the t-test. Comparing the other parameters like Adjusted R^2 , f-test, residuals, and RMSE values,both the models give reasonable results satisfying the conditions,but backward model overpowers the full model, as it as it passes the t-test for all variables and give a little bit better Adjusted R^2 value, also the RMSE's for both train and test set are very similar and near to each other.
