Assignment_5

Ajay Kanubhai Patel

2022-12-09

#Loading the package

```
#Load packages to convert file in PDF.

if(!require(tinytex)){install.packages("tinytex")}

## Loading required package: tinytex
```

This section is for the basic set up. It will clear all the plots, the console and the workspace. It also sets the overall format for numbers.

```
if(!is.null(dev.list())) dev.off()

## null device
##           1

cat("\014")
```

```r
rm(list=ls())
options(scipen=9)
```

#To read Excel file in R data frame.

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

## Loading required package: readxl

library("readxl")

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(vcd)){install.packages("vcd")}

## Loading required package: vcd

## Loading required package: grid

library("vcd")

if(!require(HSAUR)){install.packages("HSAUR")}

## Loading required package: HSAUR

## Loading required package: tools

library("HSAUR")

if(!require(rmarkdown)){install.packages("rmarkdown")}

## Loading required package: rmarkdown

library("rmarkdown")


if(!require(ggplot2)){install.packages("ggplot2")}

## Loading required package: ggplot2

library("ggplot2")

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

```
## Loading required package: klaR

## Loading required package: MASS

library("klaR")

if(!require(partykit)){install.packages("partykit")}

## Loading required package: partykit

## Loading required package: libcoin

## Loading required package: mvtnorm

library("partykit")
```

To get working directory

To read PROG8430_Assign04_22F.txt file located at

“D:/Final Assignment/DATA/Assignment5”

```
getwd()

## [1] "D:/Final Assignment/DATA/Assignment5"

Assignment05_AP <- read.table(file = "D:/Final
Assignment/DATA/Assignment5/PROG8430_Assign05_22F.txt", header = TRUE, sep =
",")

head(Assignment05_AP)

##    Del Vin Pkg Cst  Mil Dom Haz              Car
## 1  9.5   6   6  13 1447   C   H M-Press Delivery
## 2 11.9  18   7   7 1874   I   N         Fed Post
## 3 14.6   7   7   8 1865   I   N         Fed Post
## 4 17.5  11   5  16 3111   I   H M-Press Delivery
## 5 10.7  12   4  10 1319   C   H         Fed Post
## 6 10.5  12   3   5 1415   C   N M-Press Delivery

str(Assignment05_AP)

## 'data.frame':    6332 obs. of  8 variables:
##  $ Del: num  9.5 11.9 14.6 17.5 10.7 10.5 10.7 11.9 8.9 7.4 ...
##  $ Vin: int  6 18 7 11 12 12 21 12 13 16 ...
##  $ Pkg: int  6 7 7 5 4 3 1 4 6 5 ...
##  $ Cst: int  13 7 8 16 10 5 10 12 8 10 ...
##  $ Mil: int  1447 1874 1865 3111 1319 1415 1599 2361 1394 1121 ...
##  $ Dom: chr  "C" "I" "I" "I" ...
```

```
##  $ Haz: chr  "H" "N" "N" "H" ...
##  $ Car: chr  "M-Press Delivery" "Fed Post" "Fed Post" "M-Press Delivery"
...
```

1.Preliminary Data Preparation

1(1) Rename all variables with your initials appended.

My name is Ajay Patel so I have appended all column name #with _AP

```
colnames(Assignment05_AP) <- paste(colnames(Assignment05_AP), "AP", sep =
"_")


head(Assignment05_AP)

##   Del_AP Vin_AP Pkg_AP Cst_AP Mil_AP Dom_AP Haz_AP          Car_AP
## 1    9.5      6      6     13   1447      C      H M-Press Delivery
## 2   11.9     18      7      7   1874      I      N         Fed Post
## 3   14.6      7      7      8   1865      I      N         Fed Post
## 4   17.5     11      5     16   3111      I      H M-Press Delivery
## 5   10.7     12      4     10   1319      C      H         Fed Post
## 6   10.5     12      3      5   1415      C      N M-Press Delivery
```

Q1 (2). As demonstrated in class and conducted in previous assignments, make quick exploratory graphs of all variables. Remember to adjust categorical variables to factor variables

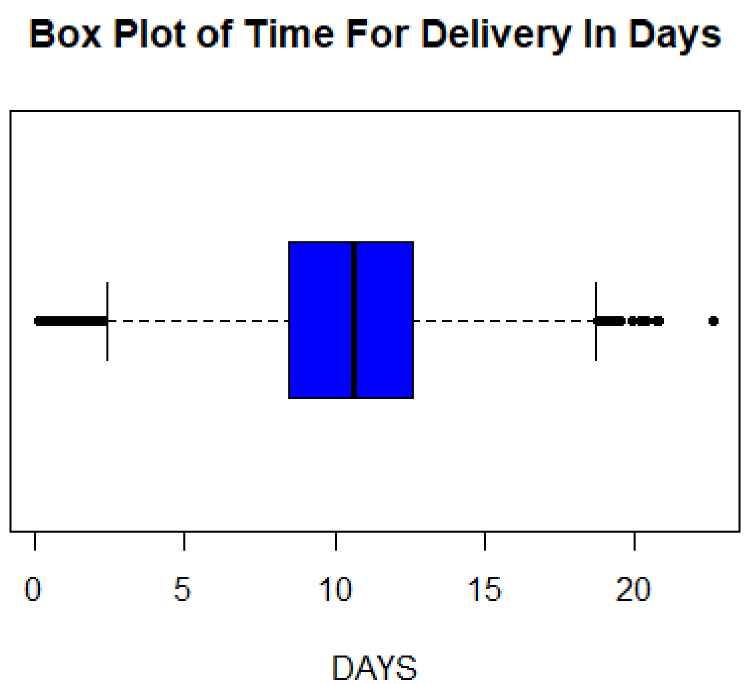
First, I am going to convert Character variables into the factor variables. There are three character variables namely Dom_AP, Haz_AP and Car_AP

```
#To find character variables and convert into the factor variables.

Assignment05_AP <- as.data.frame(unclass(Assignment05_AP), stringsAsFactors =
TRUE)
#let's check whether the character variables converted into factor variables
#or not

str(Assignment05_AP)

## 'data.frame':    6332 obs. of  8 variables:
##  $ Del_AP: num  9.5 11.9 14.6 17.5 10.7 10.5 10.7 11.9 8.9 7.4 ...
##  $ Vin_AP: int  6 18 7 11 12 12 21 12 13 16 ...
##  $ Pkg_AP: int  6 7 7 5 4 3 1 4 6 5 ...
##  $ Cst_AP: int  13 7 8 16 10 5 10 12 8 10 ...
##  $ Mil_AP: int  1447 1874 1865 3111 1319 1415 1599 2361 1394 1121 ...
##  $ Dom_AP: Factor w/ 2 levels "C","I": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Haz_AP: Factor w/ 2 levels "H","N": 1 2 2 1 1 2 1 2 2 1 ...
##  $ Car_AP: Factor w/ 2 levels "Fed Post","M-Press Delivery": 2 1 1 2 1 2 2
2 1 2 ...
```

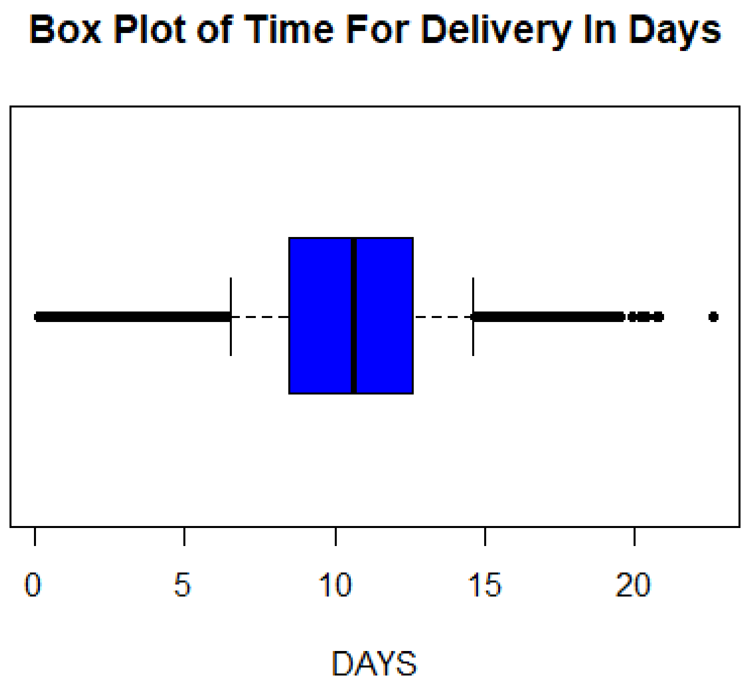
exploratory graphs of all variables

First, we gonna check for outlier with the help of boxplot and density plot.

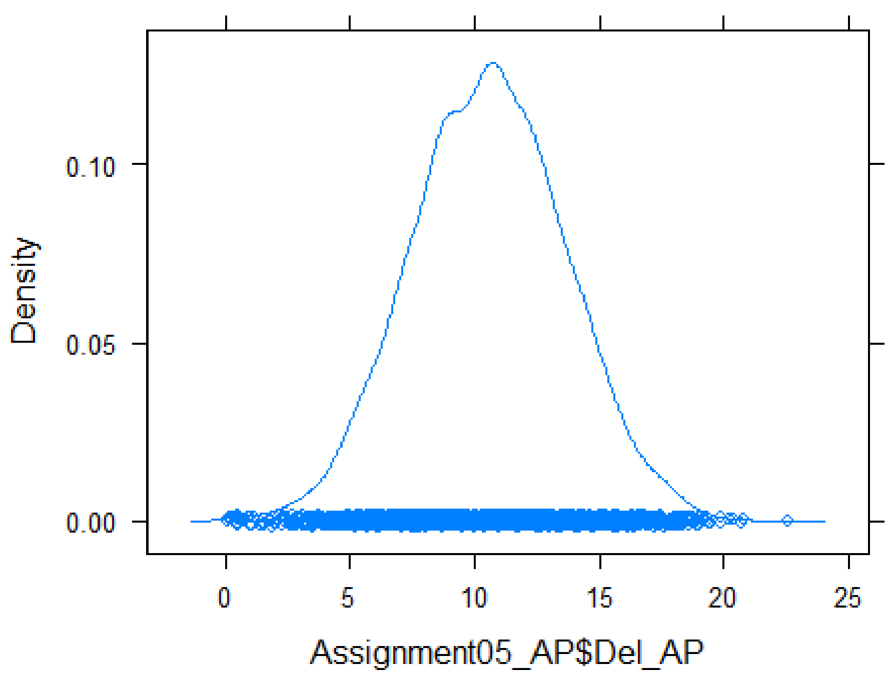
```
#FOR TIME FOR DELIVERY
boxplot(Assignment05_AP$Del_AP,
       main="Box Plot of Time For Delivery In Days",
        xlab="DAYS",
        col="blue", horizontal=TRUE, pch=20)
```



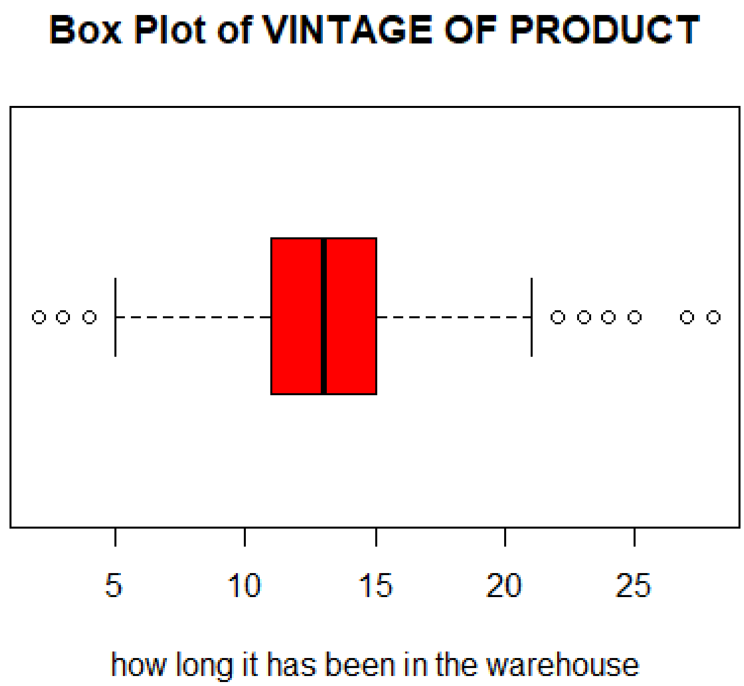

```
# let's shrink the graph and observe

boxplot(Assignment05_AP$Del_AP,
       main="Box Plot of Time For Delivery In Days",
        xlab="DAYS",
        col="blue", horizontal=TRUE, pch=20, range = 0.5)
```



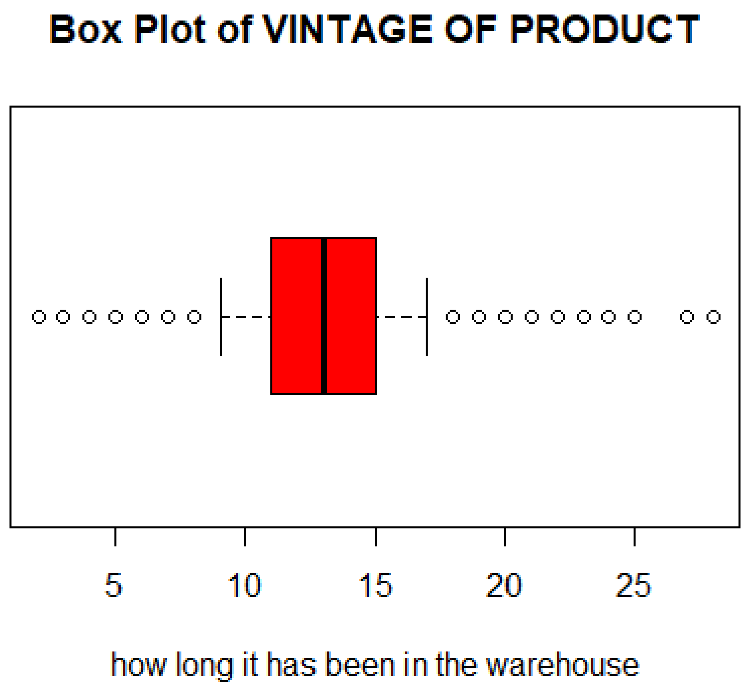

```
densityplot( ~ Assignment05_AP$Del_AP, pch=1)
```



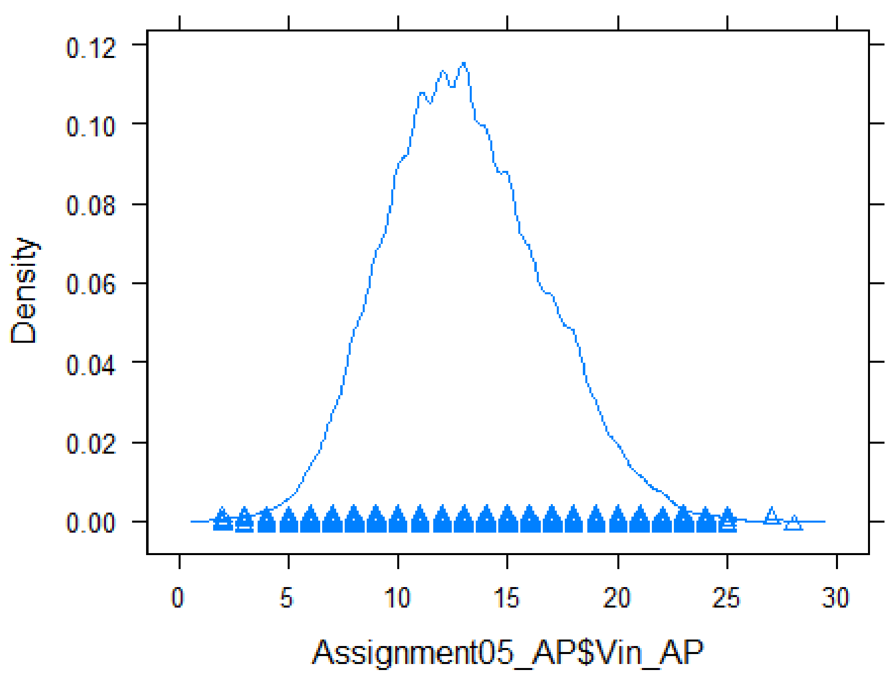
```r
# FOR VINTAGE OF PRODUCT

boxplot(Assignment05_AP$Vin_AP,
       main="Box Plot of VINTAGE OF PRODUCT",
        xlab="how long it has been in the warehouse",
        col="red", horizontal=TRUE, pch=21)
```



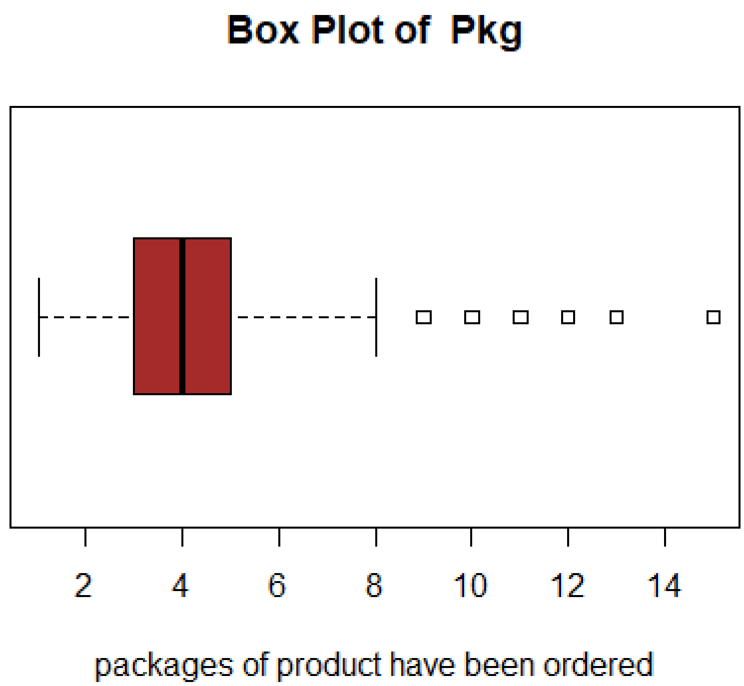

```r
# let's shrink the graph and observe

boxplot(Assignment05_AP$Vin_AP,
       main="Box Plot of VINTAGE OF PRODUCT",
        xlab="how long it has been in the warehouse",
        col="red", horizontal=TRUE, pch=21, range = 0.5)
```



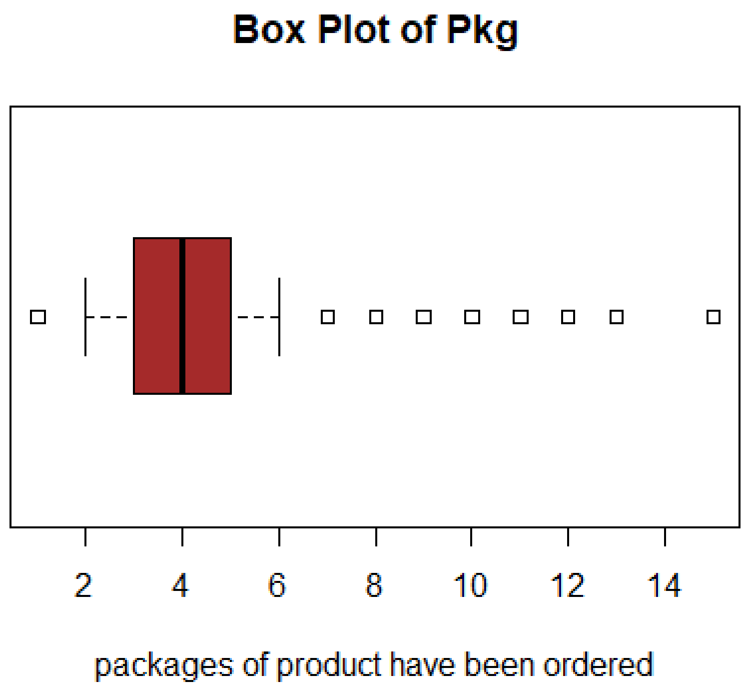

```
densityplot( ~ Assignment05_AP$Vin_AP, pch=2)
```



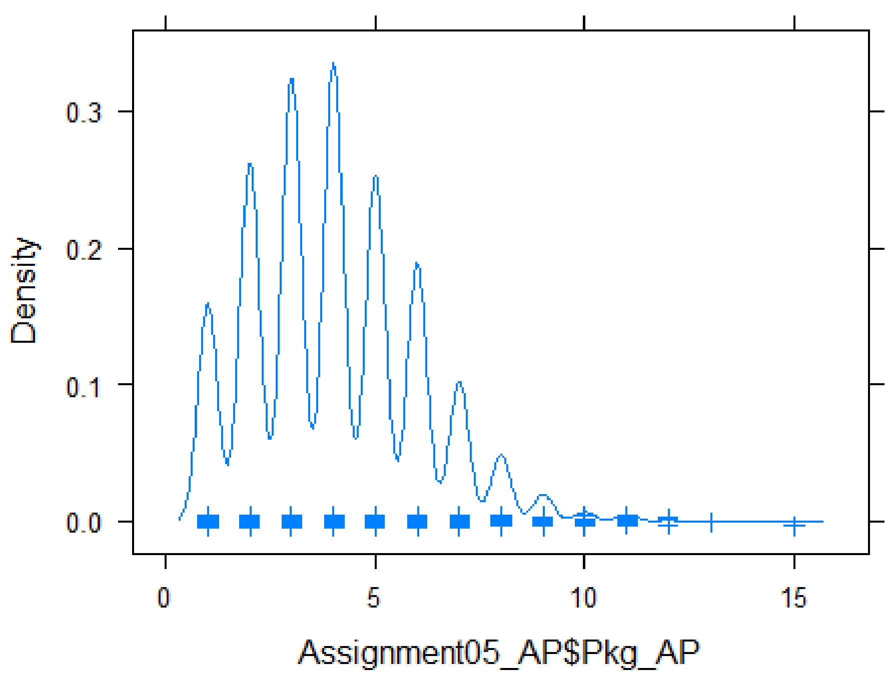
```
# FOR Pkg

boxplot(Assignment05_AP$Pkg_AP,
       main="Box Plot of  Pkg",
        xlab="packages of product have been ordered",
        col="brown",horizontal=TRUE, pch=22)
```



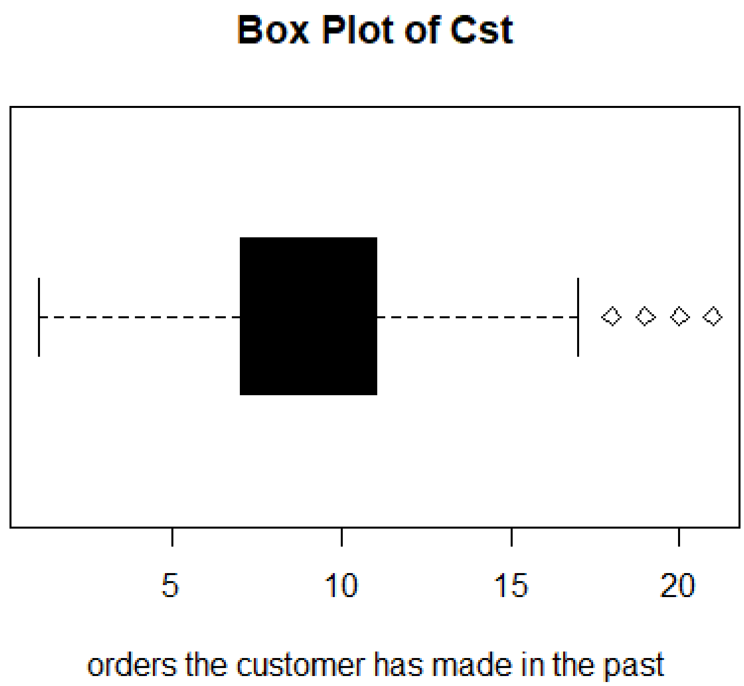

```
# let's shrink the graph and observe

boxplot(Assignment05_AP$Pkg_AP,
       main="Box Plot of Pkg",
        xlab="packages of product have been ordered",
        col="brown",horizontal=TRUE, pch=22, range = 0.5)
```



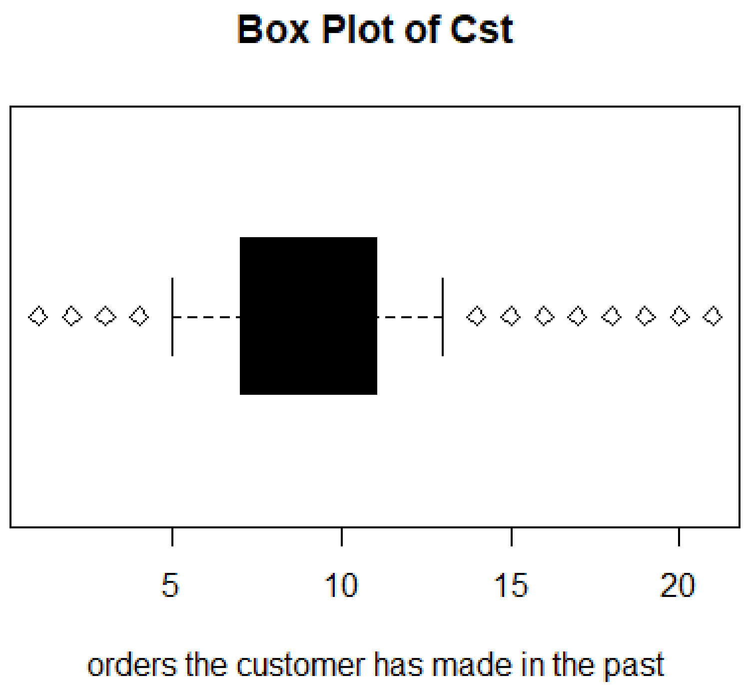

```
densityplot( ~ Assignment05_AP$Pkg_AP, pch=3)
```



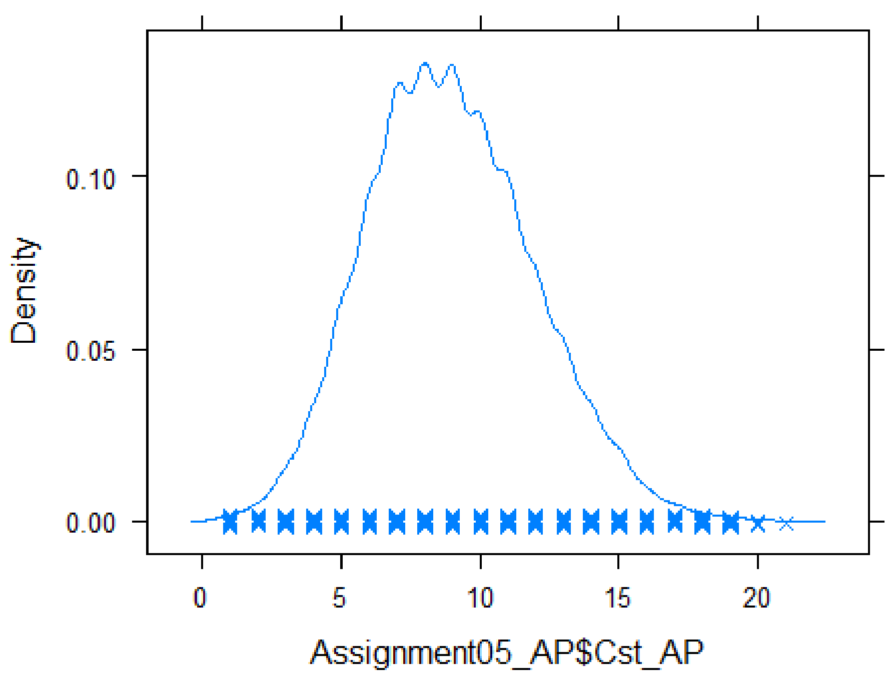
```r
# FOR Cst

boxplot(Assignment05_AP$Cst_AP,
       main="Box Plot of Cst",
        xlab="orders the customer has made in the past",
        col=1,horizontal=TRUE, pch=23)
```



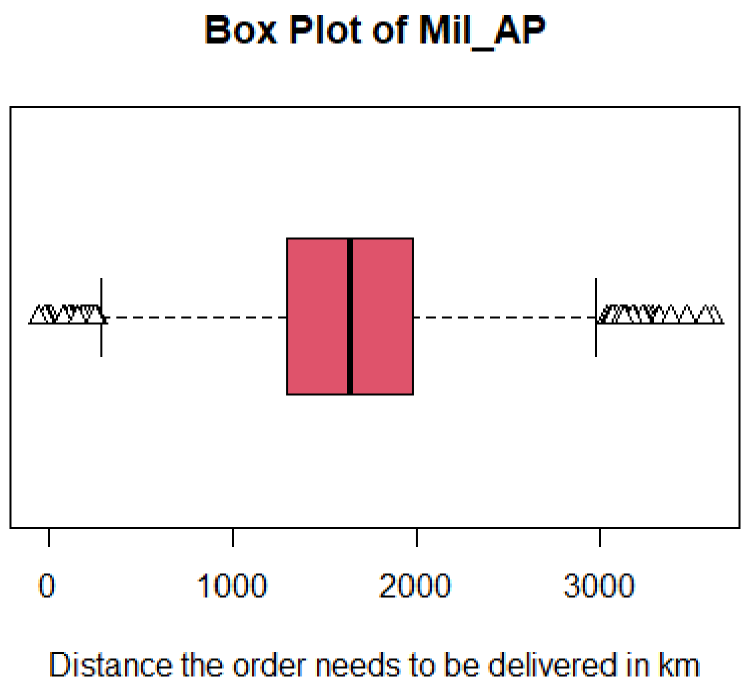

```r
# let's shrink the graph and observe

boxplot(Assignment05_AP$Cst_AP,
       main="Box Plot of Cst",
        xlab="orders the customer has made in the past",
        col=1,horizontal=TRUE, pch=23, range = 0.5)
```



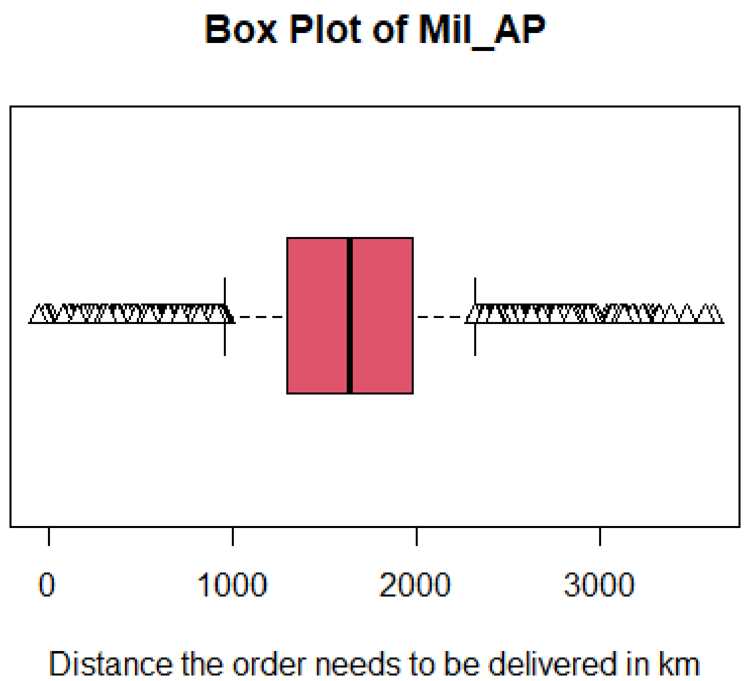

```
densityplot( ~ Assignment05_AP$Cst_AP, pch=4)
```



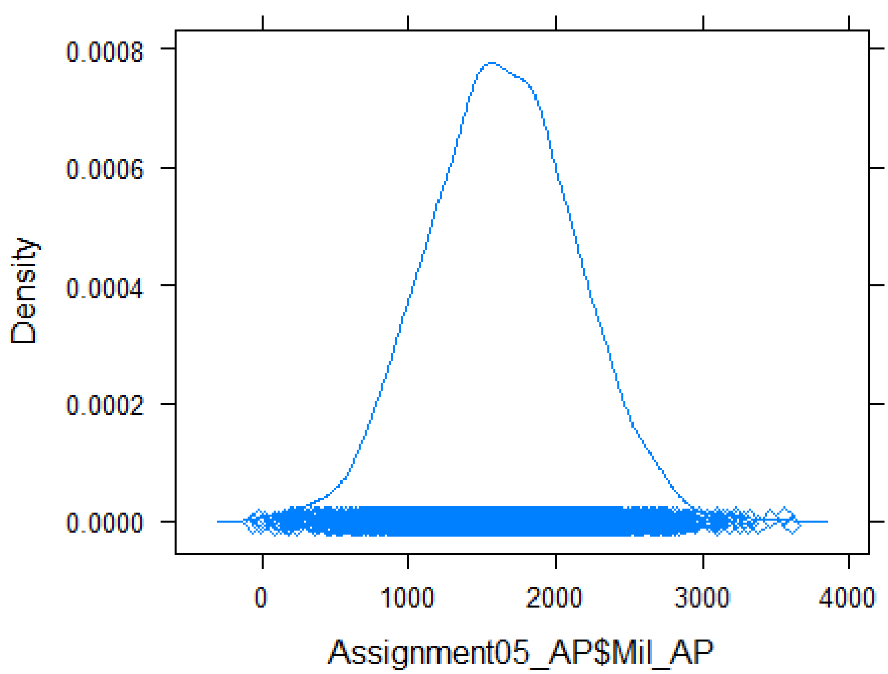
```
# FOR Mil_AP

boxplot(Assignment05_AP$Mil_AP,
       main="Box Plot of Mil_AP",
        xlab="Distance the order needs to be delivered in km",
        col=2,horizontal=TRUE, pch=24)
```




```
# let's shrink the graph and observe

boxplot(Assignment05_AP$Mil_AP,
       main="Box Plot of Mil_AP",
        xlab="Distance the order needs to be delivered in km",
        col=2,horizontal=TRUE, pch=24, range = 0.5)
```



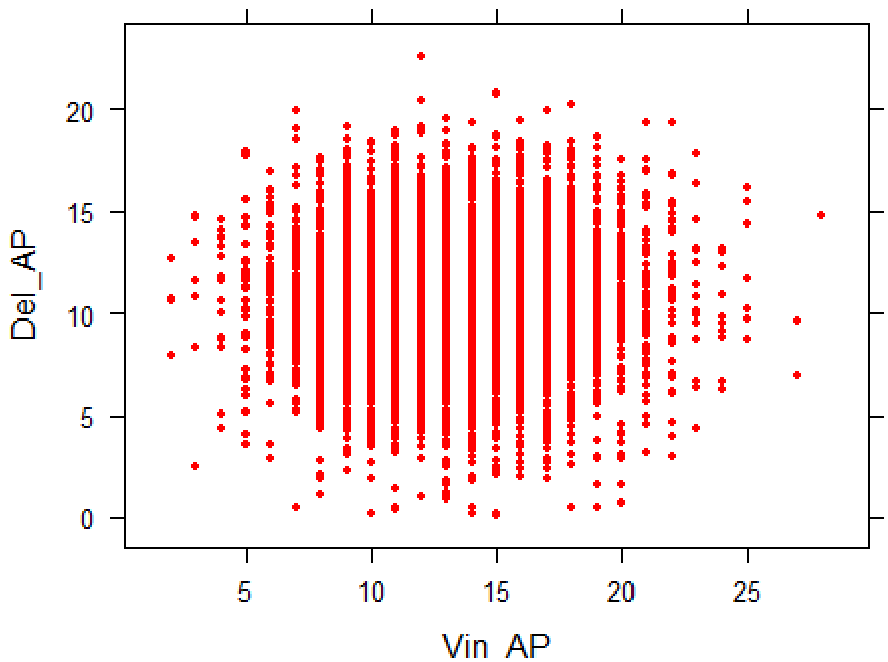

```
densityplot( ~ Assignment05_AP$Mil_AP, pch=5)
```



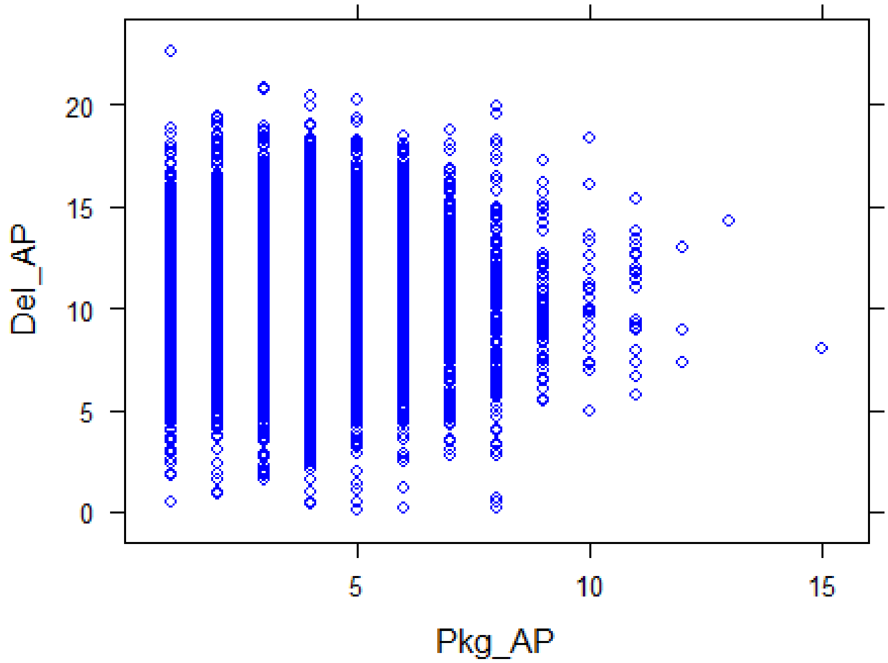
from box plot and density plot all the numeric varibles seem fine so I keep them as they are.

let’s check correlation between two variables. There are five numeric variables so we have to check for 5(5-1)/2 = 10 pairs.

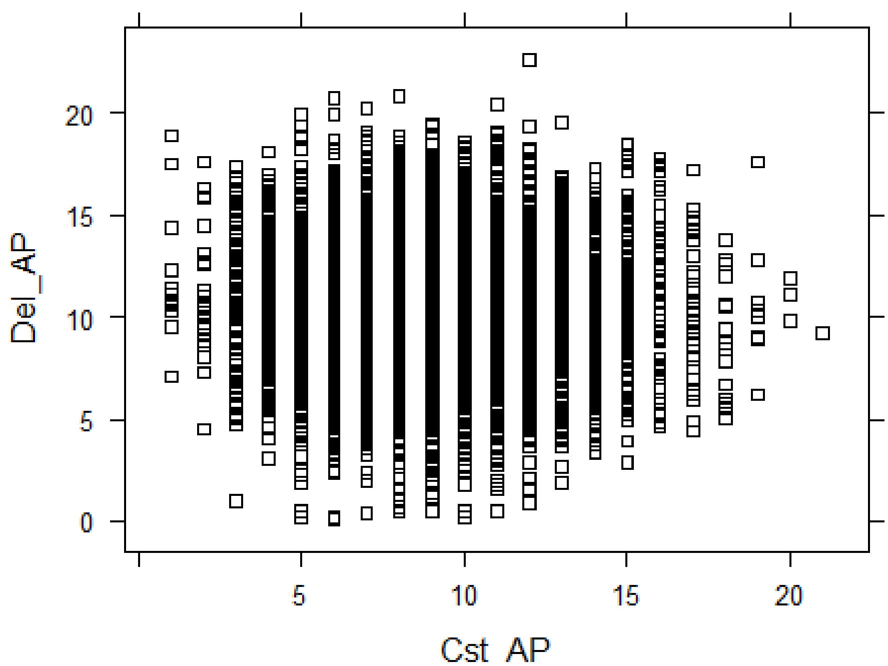
```
xyplot(Del_AP~Vin_AP, data = Assignment05_AP, col="red", pch=20)
```



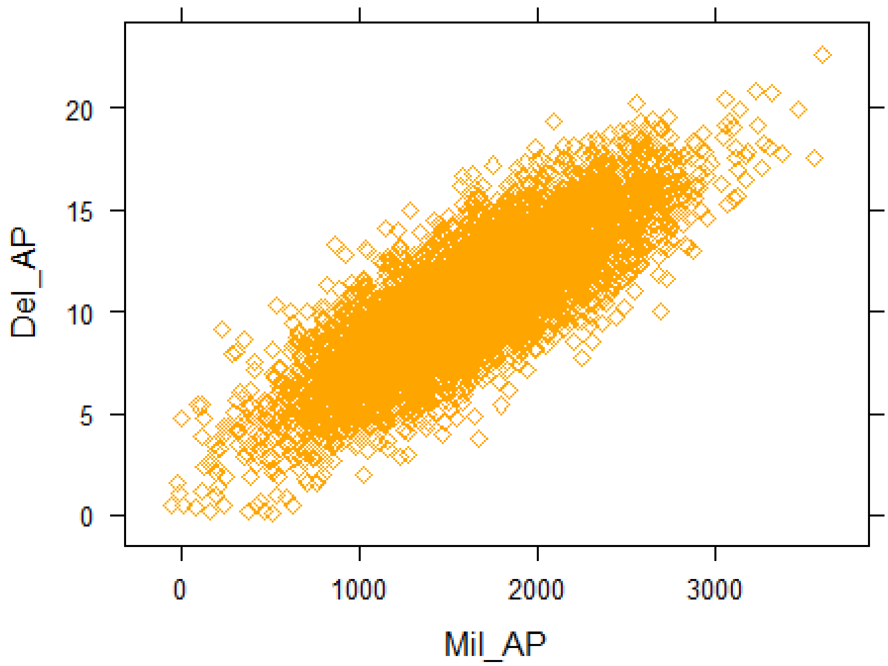

```
xyplot(Del_AP~Pkg_AP, data = Assignment05_AP, col="blue", pch=21)
```



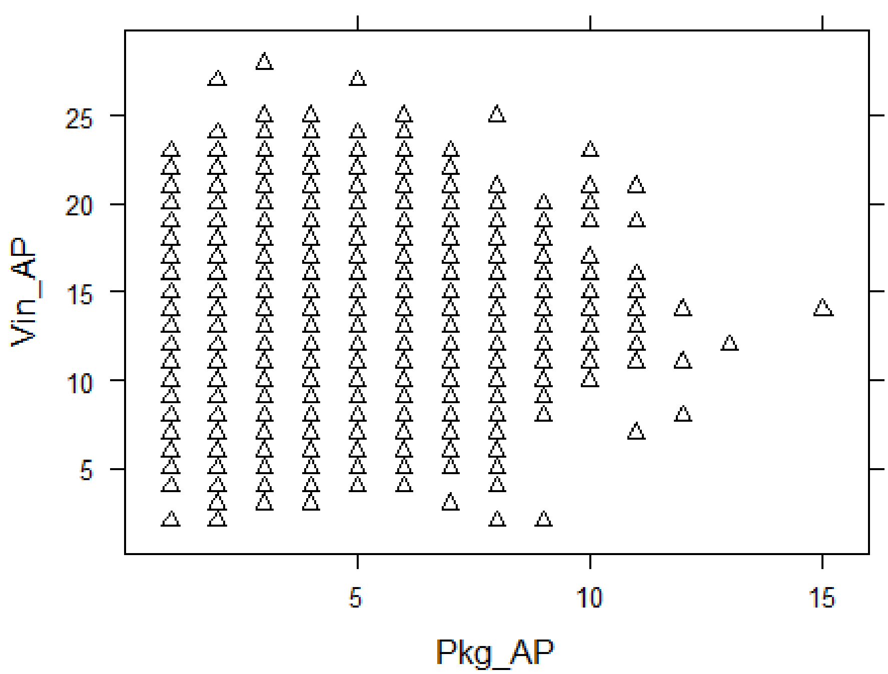

```
xyplot(Del_AP~Cst_AP, data = Assignment05_AP, col="black", pch=22)
```



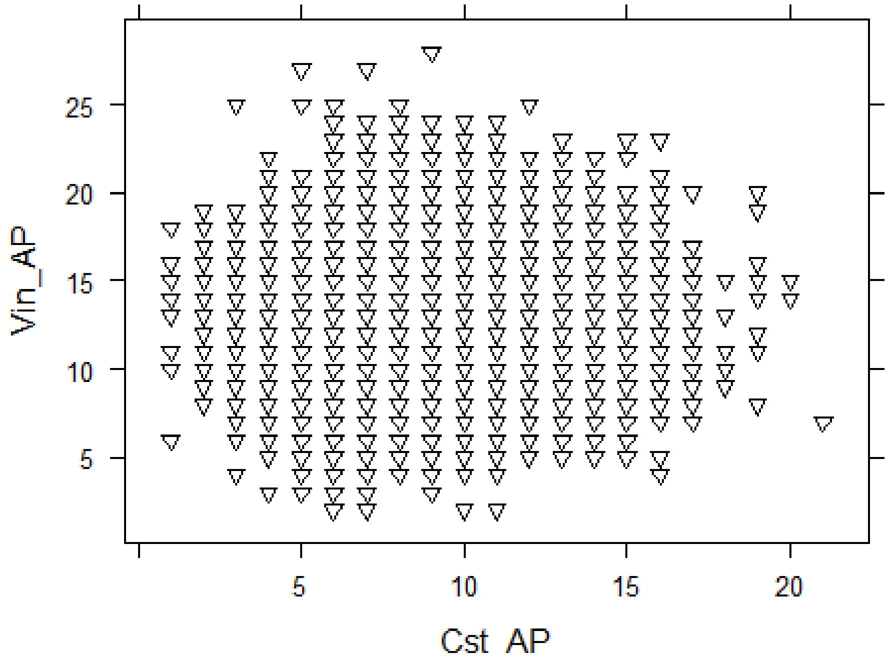

```
xyplot(Del_AP~Mil_AP, data = Assignment05_AP, col="orange", pch=23)
```



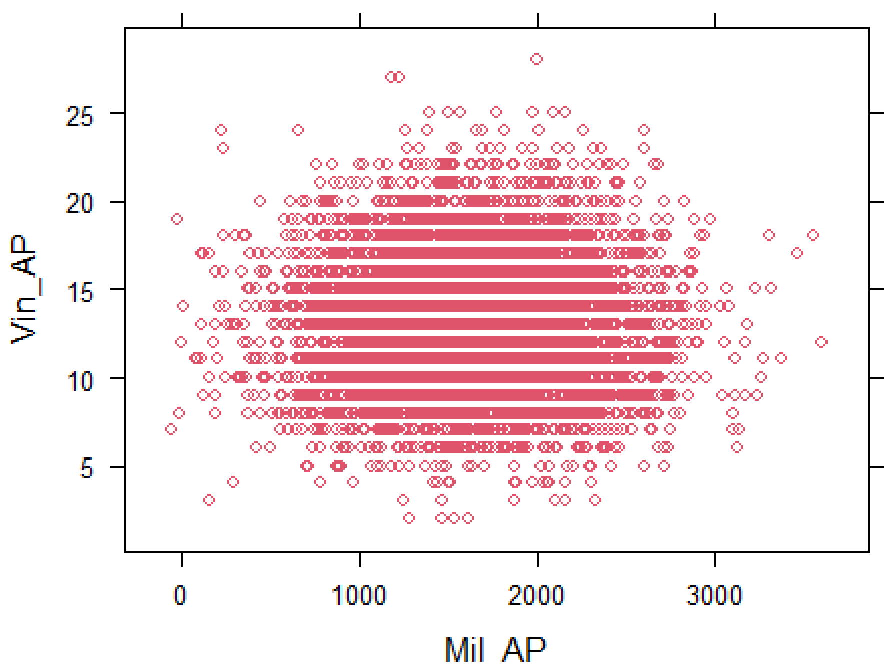

```
xyplot(Vin_AP~Pkg_AP, data = Assignment05_AP, col="black", pch=24)
```



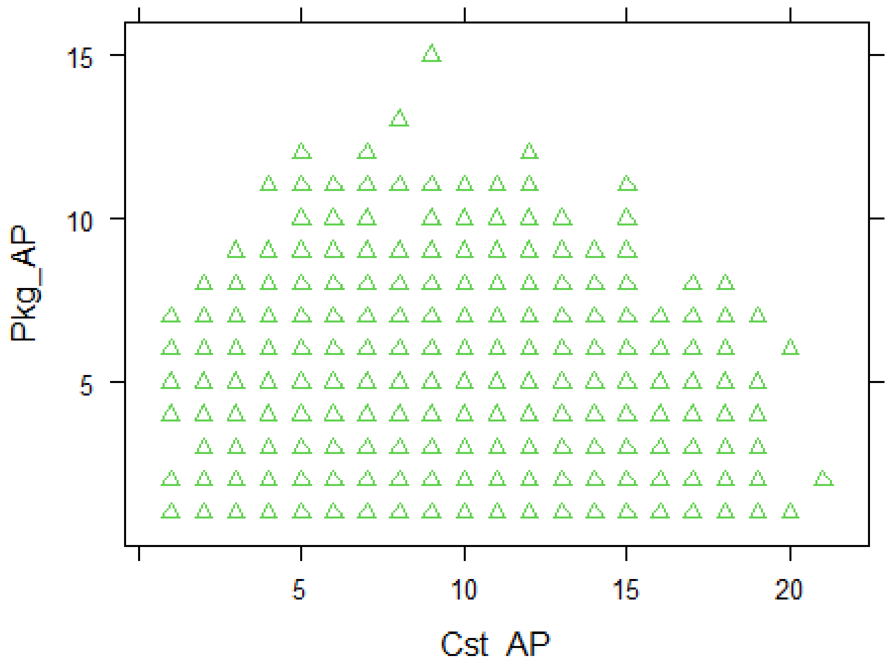

```
xyplot(Vin_AP~Cst_AP, data = Assignment05_AP, col=1, pch=25)
```



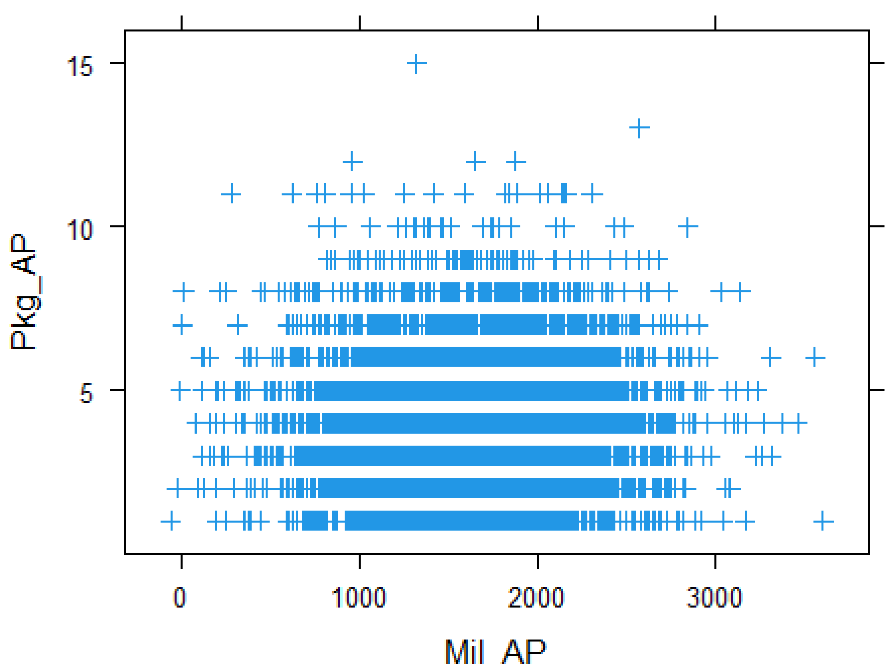

```r
xyplot(Vin_AP~Mil_AP, data = Assignment05_AP, col=2, pch=1)
```



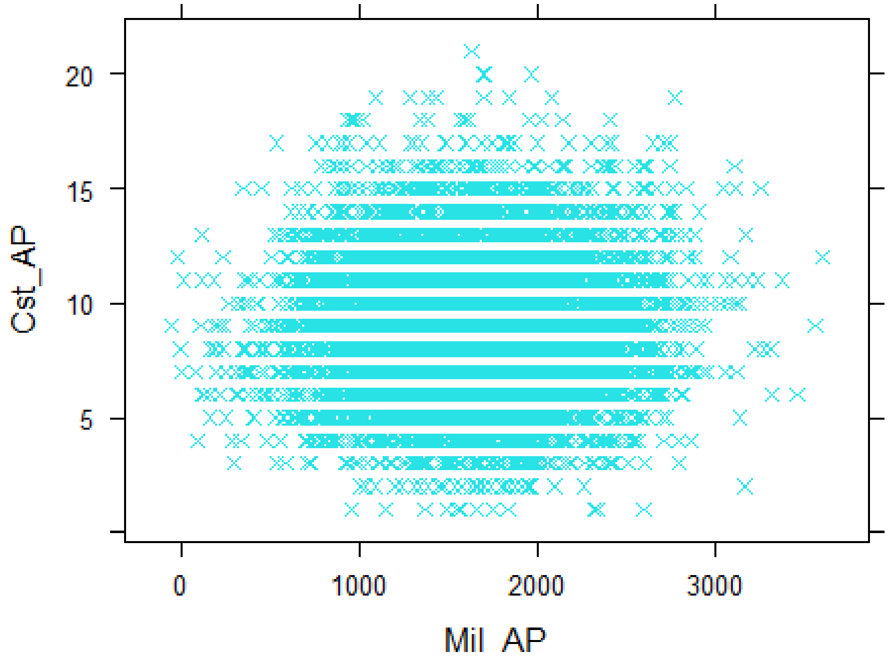

```r
xyplot(Pkg_AP~Cst_AP, data = Assignment05_AP, col=3, pch=2)
```




```
xyplot(Pkg_AP~Mil_AP, data = Assignment05_AP, col=4, pch=3)
```




```
xyplot(Cst_AP~Mil_AP, data = Assignment05_AP, col=5, pch=4)
```

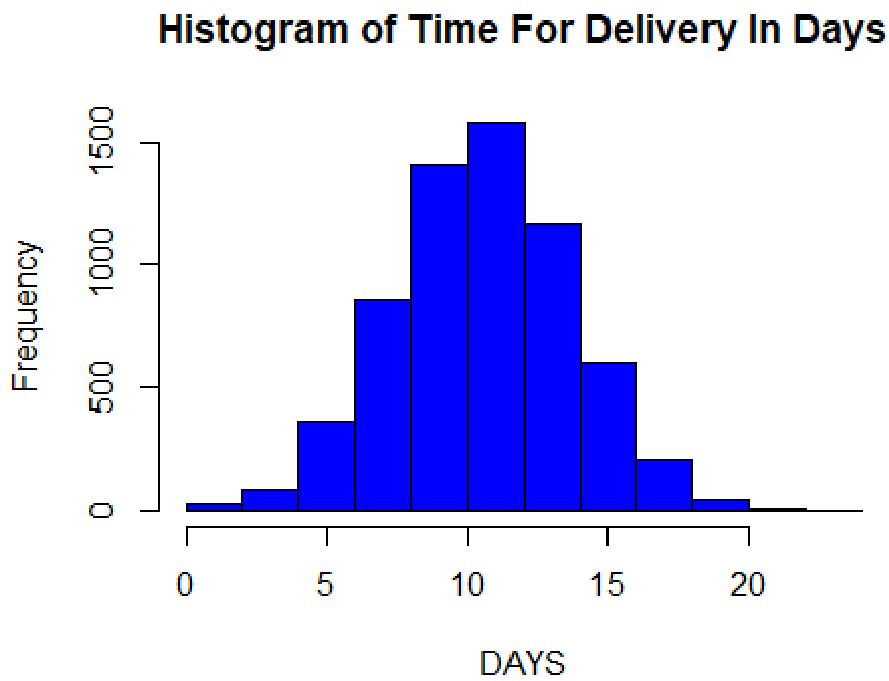


Conclusion from xyplot: There is no strong correlation between two variable except Del_AP and Mil_AP.

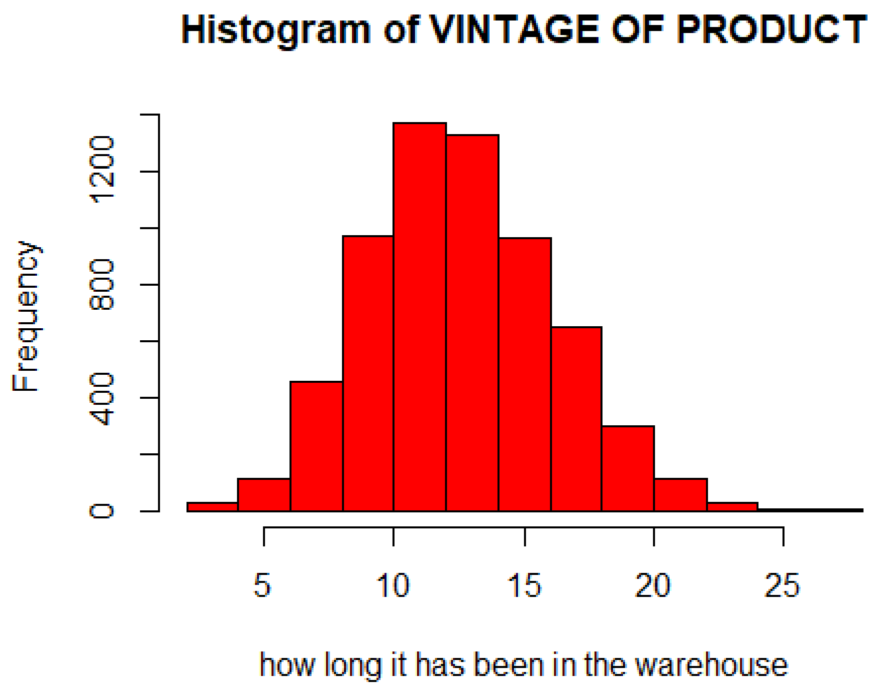
NOTE: Here, rest of the variables are binary so we cannot use barplot for check correlation of one with another variable.

Let’s check skewness of the given numeric variables.

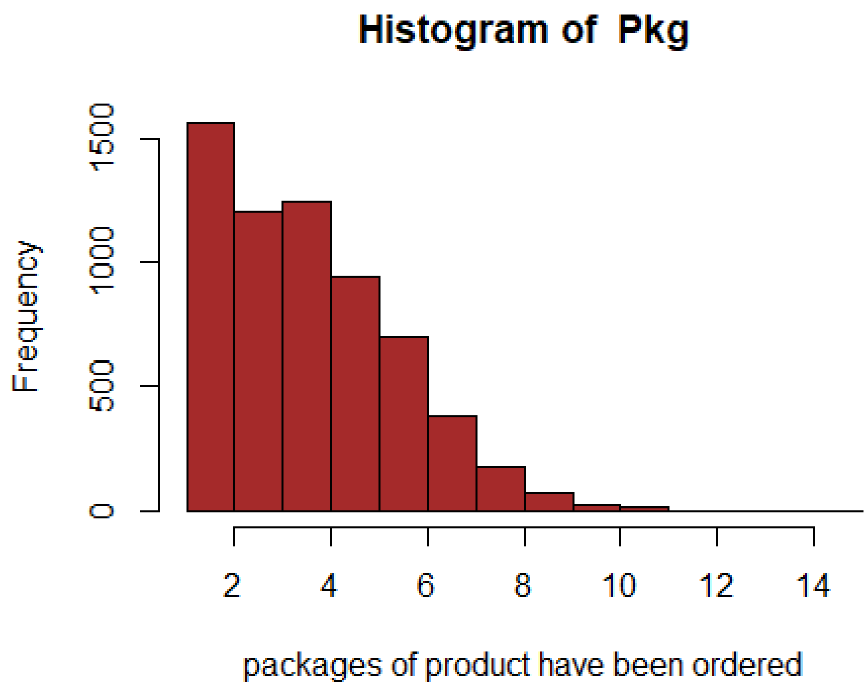
```
#For Del_AP
hist(Assignment05_AP$Del_AP,
       main="Histogram of Time For Delivery In Days",
        xlab="DAYS",
        col="blue", pch=20)
```



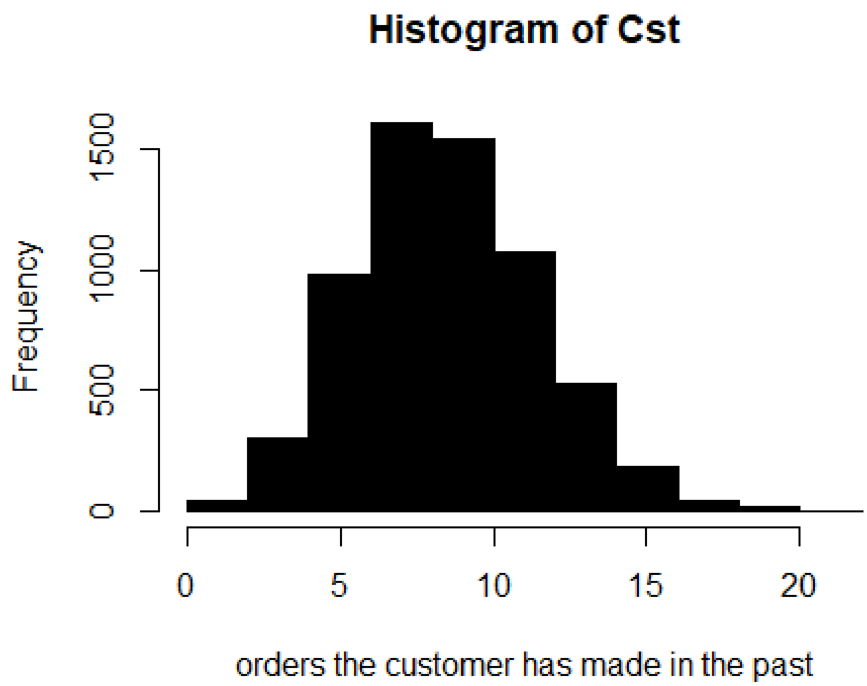

```
# FOR VINTAGE OF PRODUCT

hist(Assignment05_AP$Vin_AP,
       main="Histogram of VINTAGE OF PRODUCT",
        xlab="how long it has been in the warehouse",
        col="red", pch=21)
```



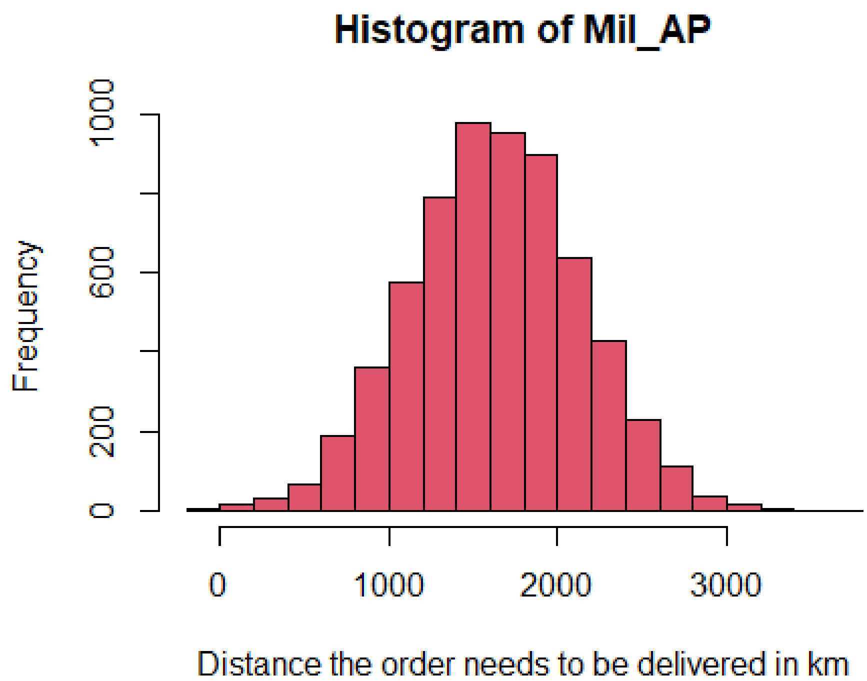

```
# FOR Pkg

hist(Assignment05_AP$Pkg_AP,
       main="Histogram of  Pkg",
        xlab="packages of product have been ordered",
        col="brown", pch=22)
```




```
# FOR Cst

hist(Assignment05_AP$Cst_AP,
       main="Histogram of Cst",
        xlab="orders the customer has made in the past",
        col=1, pch=23)
```



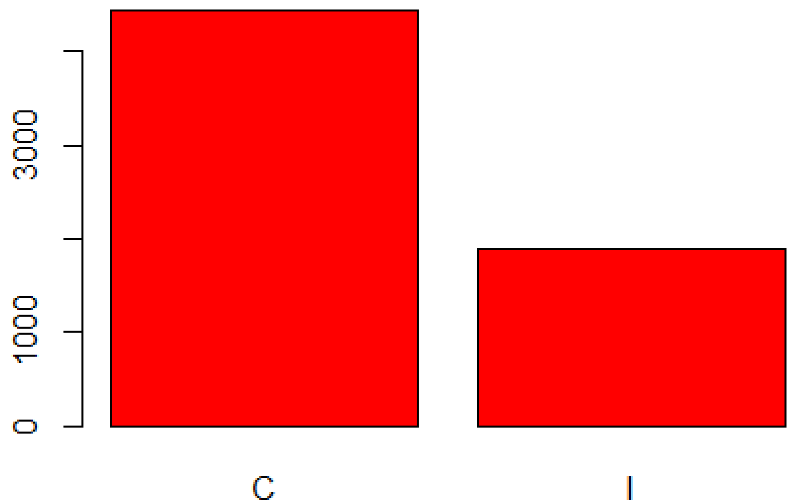

```
# FOR Mil_AP

hist(Assignment05_AP$Mil_AP,
       main="Histogram of Mil_AP",
        xlab="Distance the order needs to be delivered in km",
        col=2, pch=24)
```



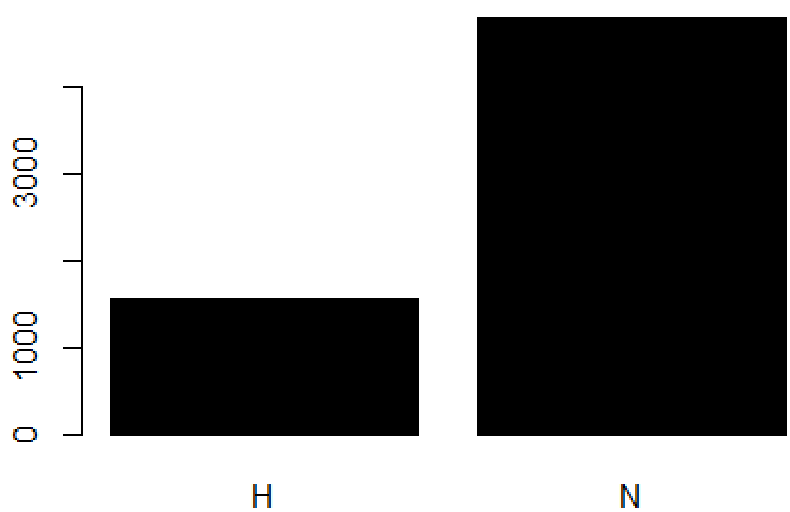
Conclusion from Histogram: Pkg_AP variable is right skewed. others seem fine.

let’s create barplot for factor variables.

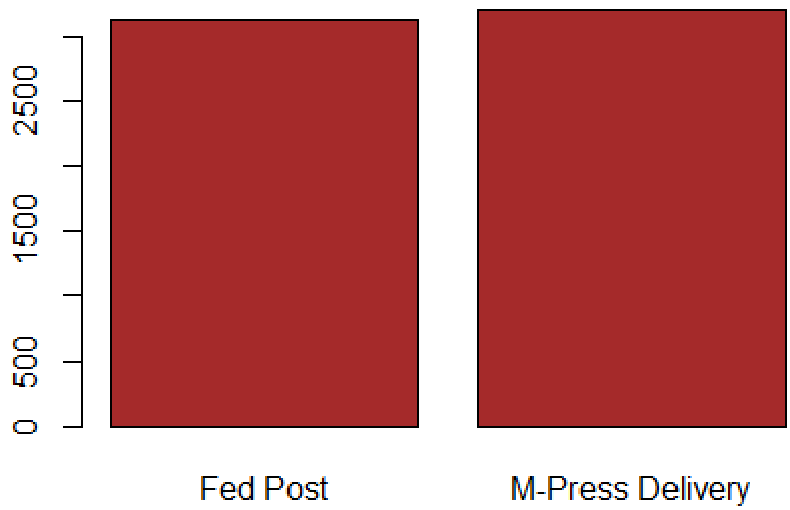
```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:pastecs':
##
##     first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

table(Assignment05_AP$Dom_AP) %>% barplot(col = "red")
```




```r
table(Assignment05_AP$Haz_AP) %>% barplot(col = "black")
```




```r
table(Assignment05_AP$Car_AP) %>% barplot(col = "brown")
```



Conclusion From Barplot:

1. There are more products which are manufactured in Canada.
2. Majority of products are non-hazardous.
3. number of delivery done by Fed Post and M-Press is almost same.

Q1 (3) Create a new variable in the dataset called OT_[Intials] which will have a value of 1 if Del ≤ 10 and 0 otherwise. If you have forgotten how to do this, the code to accomplish it is included in the appendix.

```
OT_AP <- as.factor(ifelse(Assignment05_AP$Del_AP <= 10, 1,0))
head(OT_AP)

## [1] 1 0 0 0 0 0
## Levels: 0 1

summary(OT_AP) # To see how many '0' and '1'.

##    0    1
## 3596 2736
```

2. Exploratory Analysis

Q2 (1) Correlations: Create numeric correlations (as demonstrated) and comment on what you see. Are there co-linear variables?

High correlation between two variables means they have similar trends and are likely to carry similar information.

Pearson,Spearman,and Kendall methods can be used to measure the degree of association between two variables.

We can only check for numerical and we have 5 column with numeric data so

n(n-1)/2 (5*4/2 = 10) combinations should be checked.

```
#We used Spearman because it is non-parametric

cor(Assignment05_AP$Del_AP, Assignment05_AP$Vin_AP)

## [1] 0.02634195

cor.test(Assignment05_AP$Del_AP, Assignment05_AP$Vin_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Del_AP,
Assignment05_AP$Vin_AP, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Del_AP and Assignment05_AP$Vin_AP
## S = 41214624060, p-value = 0.03891
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.02595306

cor(Assignment05_AP$Del_AP, Assignment05_AP$Pkg_AP)

## [1] -0.01607883

cor.test(Assignment05_AP$Del_AP, Assignment05_AP$Pkg_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Del_AP,
Assignment05_AP$Pkg_AP, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Del_AP and Assignment05_AP$Pkg_AP
## S = 43009848289, p-value = 0.1899
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.01647442

cor(Assignment05_AP$Del_AP, Assignment05_AP$Cst_AP)

## [1] -0.02047519
```

```
cor.test(Assignment05_AP$Del_AP, Assignment05_AP$Cst_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Del_AP,
Assignment05_AP$Cst_AP, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Del_AP and Assignment05_AP$Cst_AP
## S = 43222167401, p-value = 0.08725
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.02149227

cor(Assignment05_AP$Del_AP, Assignment05_AP$Mil_AP)

## [1] 0.8168552

cor.test(Assignment05_AP$Del_AP, Assignment05_AP$Mil_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Del_AP,
Assignment05_AP$Mil_AP, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Del_AP and Assignment05_AP$Mil_AP
## S = 8160311078, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.8071431

cor(Assignment05_AP$Vin_AP, Assignment05_AP$Pkg_AP)

## [1] 0.001513925

cor.test(Assignment05_AP$Vin_AP, Assignment05_AP$Pkg_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Vin_AP,
Assignment05_AP$Pkg_AP, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Vin_AP and Assignment05_AP$Pkg_AP
## S = 42396387766, p-value = 0.8751
## alternative hypothesis: true rho is not equal to 0
```

```
## sample estimates:
##          rho
## -0.001976183

cor(Assignment05_AP$Vin_AP, Assignment05_AP$Cst_AP)

## [1] 0.003905538

cor.test(Assignment05_AP$Vin_AP, Assignment05_AP$Cst_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Vin_AP,
Assignment05_AP$Cst_AP, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Vin_AP and Assignment05_AP$Cst_AP
## S = 42146550562, p-value = 0.7546
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## 0.003928352

cor(Assignment05_AP$Vin_AP, Assignment05_AP$Mil_AP)

## [1] 0.01577437

cor.test(Assignment05_AP$Vin_AP, Assignment05_AP$Mil_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Vin_AP,
Assignment05_AP$Mil_AP, :
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Vin_AP and Assignment05_AP$Mil_AP
## S = 41635785494, p-value = 0.203
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.01599953

cor(Assignment05_AP$Pkg_AP, Assignment05_AP$Cst_AP)

## [1] -0.0003002829

cor.test(Assignment05_AP$Pkg_AP, Assignment05_AP$Cst_AP, method="spearman")

## Warning in cor.test.default(Assignment05_AP$Pkg_AP,
Assignment05_AP$Cst_AP, :
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Pkg_AP and Assignment05_AP$Cst_AP
## S = 42376081573, p-value = 0.9052
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.001496276
```

```
cor(Assignment05_AP$Pkg_AP, Assignment05_AP$Mil_AP)
```

```
## [1] -0.007966585
```

```
cor.test(Assignment05_AP$Pkg_AP, Assignment05_AP$Mil_AP, method="spearman")
```

```
## Warning in cor.test.default(Assignment05_AP$Pkg_AP,
Assignment05_AP$Mil_AP, :
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Pkg_AP and Assignment05_AP$Mil_AP
## S = 42573255040, p-value = 0.6243
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.00615618
```

```
cor(Assignment05_AP$Cst_AP, Assignment05_AP$Mil_AP)
```

```
## [1] 0.01967505
```

```
cor.test(Assignment05_AP$Cst_AP, Assignment05_AP$Mil_AP, method="spearman")
```

```
## Warning in cor.test.default(Assignment05_AP$Cst_AP,
Assignment05_AP$Mil_AP, :
## Cannot compute exact p-value with ties
```

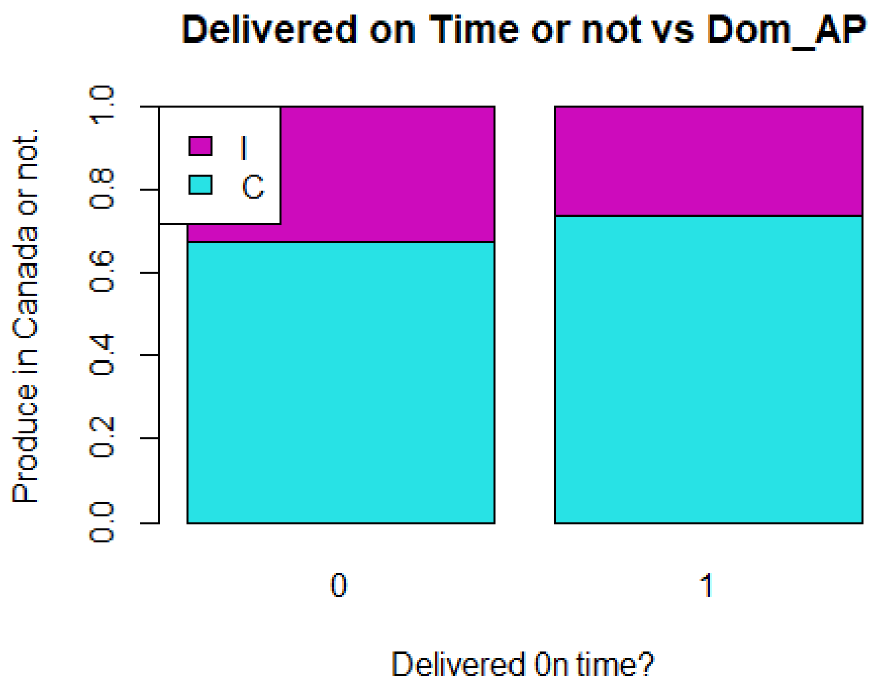
```
##
##  Spearman's rank correlation rho
##
## data:  Assignment05_AP$Cst_AP and Assignment05_AP$Mil_AP
## S = 41810635983, p-value = 0.3451
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.0118672
```

Conclusion: There is strong positive linear relationship between Del_Ap and Mil_AP.

Q2(2) Identify the most significant predictor of an on time delivery and provide statistical evidence (in addition to the correlation coefficient) that suggest they are associated with an on time delivery (Think of the contingency tables bar plots we did in class).

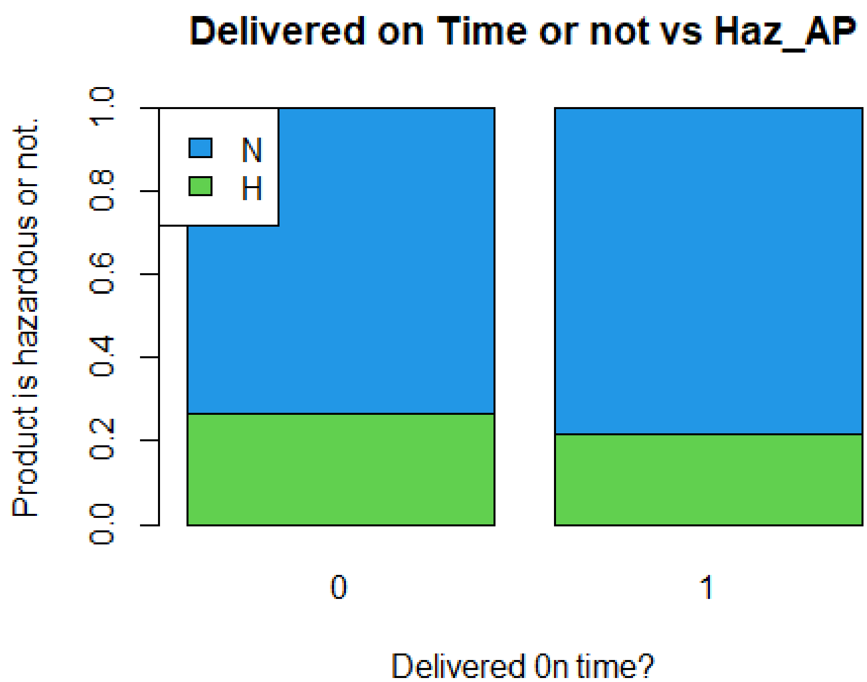
both of the factors are categorical (rather than numeric variables).

```
str(Assignment05_AP)

## 'data.frame':    6332 obs. of  8 variables:
##  $ Del_AP: num  9.5 11.9 14.6 17.5 10.7 10.5 10.7 11.9 8.9 7.4 ...
##  $ Vin_AP: int  6 18 7 11 12 12 21 12 13 16 ...
##  $ Pkg_AP: int  6 7 7 5 4 3 1 4 6 5 ...
##  $ Cst_AP: int  13 7 8 16 10 5 10 12 8 10 ...
##  $ Mil_AP: int  1447 1874 1865 3111 1319 1415 1599 2361 1394 1121 ...
##  $ Dom_AP: Factor w/ 2 levels "C","I": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Haz_AP: Factor w/ 2 levels "H","N": 1 2 2 1 1 2 1 2 2 1 ...
##  $ Car_AP: Factor w/ 2 levels "Fed Post","M-Press Delivery": 2 1 1 2 1 2 2
2 1 2 ...

#Contingency table for OT_AP and Dom_AP.

ODTbl_Rct_AP <- table(Assignment05_AP$Dom_AP,OT_AP, dnn=list("Time on
Delivery","Vintage Of Product"))

ODTbl_Rct_AP

##                  Vintage Of Product
## Time on Delivery    0    1
##                 C 2420 2018
##                 I 1176  718

prop.table(ODTbl_Rct_AP,2)

##                  Vintage Of Product
## Time on Delivery         0         1
##                 C 0.6729700 0.7375731
##                 I 0.3270300 0.2624269

#Vertical Bar Chart
barplot(prop.table(ODTbl_Rct_AP,2), xlab='Delivered 0n time?',ylab='Produce
in Canada or not.',main="Delivered on Time or not vs Dom_AP",
col=c(5,6)
,legend=rownames(ODTbl_Rct_AP), args.legend = list(x = "topleft"))
```




```
ODchisqRct_AP <- chisq.test(Assignment05_AP$Dom_AP,OT_AP, correct=FALSE)
ODchisqRct_AP

##
##  Pearson's Chi-squared test
##
## data:  Assignment05_AP$Dom_AP and OT_AP
## X-squared = 30.933, df = 1, p-value = 0.00000002671

#Contingency table for OT_AP and Haz_AP.

OHTbl_Rct_AP <- table(Assignment05_AP$Haz_AP,OT_AP, dnn=list("Delivery on
time?","Hazardous or not"))

OHTbl_Rct_AP

##                   Hazardous or not
## Delivery on time?    0    1
##                  H  959  594
##                  N 2637 2142

prop.table(OHTbl_Rct_AP,2)

##                   Hazardous or not
## Delivery on time?         0         1
##                  H 0.2666852 0.2171053
##                  N 0.7333148 0.7828947
```

```r
#Vertical Bar Chart
barplot(prop.table(OHTbl_Rct_AP,2), xlab='Delivered 0n time?',ylab='Product is hazardous or not.',main="Delivered on Time or not vs Haz_AP",
col=c(3,4)
,legend=rownames(OHTbl_Rct_AP), args.legend = list(x = "topleft"))
```

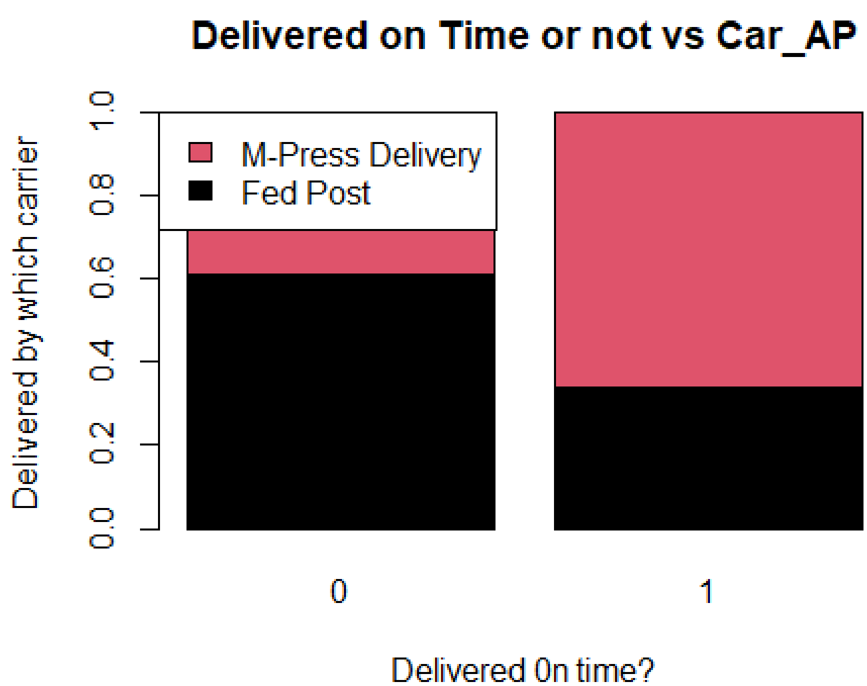



```r
OHchisqRct_AP <- chisq.test(Assignment05_AP$Haz_AP,OT_AP, correct=FALSE)
OHchisqRct_AP

##
##  Pearson's Chi-squared test
##
## data:  Assignment05_AP$Haz_AP and OT_AP
## X-squared = 20.634, df = 1, p-value = 0.00000556

#Contingency table for OT_AP and Car_AP.

OCTbl_Rct_AP <- table(Assignment05_AP$Car_AP,OT_AP, dnn=list("Delivered 0n time?","Whcih carrier delivered"))

OCTbl_Rct_AP

##                     Whcih carrier delivered
## Delivered 0n time?     0    1
##   Fed Post          2193  933
##   M-Press Delivery 1403 1803
```

```
prop.table(OCTbl_Rct_AP,2)
```

```
##                    Whcih carrier delivered
## Delivered 0n time?         0         1
##   Fed Post         0.6098443 0.3410088
##   M-Press Delivery 0.3901557 0.6589912
```

```
#Vertical Bar Chart
barplot(prop.table(OCTbl_Rct_AP,2), xlab='Delivered 0n time?',ylab='Delivered
by which carrier',main="Delivered on Time or not vs Car_AP",
col=c(1,2)
,legend=rownames(OCTbl_Rct_AP), args.legend = list(x = "topleft"))
```




```
OCchisqRct_AP <- chisq.test(Assignment05_AP$Car_AP,OT_AP, correct=FALSE)
OCchisqRct_AP
```

```
##
##  Pearson's Chi-squared test
##
## data:  Assignment05_AP$Car_AP and OT_AP
## X-squared = 449.26, df = 1, p-value < 2.2e-16
```

Conclusion: 1.More product are delivered on time when produced in Canada. while, Product produced outside of Canada mostly delivered not on time.

```
          Product manufactured in Canada and delivered on Time: 2018
          Product not manufactured in Canada and delivered on Time:718
          Product manufactured outside of Canada and not delivered on Time:
```

```
    1176
    Product manufactured in Canada and not  delivered on Time:2420
    From Chi-Squared,p value is below 0.05 so we van reject Null
    Hypothesis and say there is correlation between OT_AP and Dom_AP.

  2. When product falls under Hazardous category then they are not
     delivered on time compared to product falls under non-hazardous
     category.

     Hazardous and delivered on time:594
     Hazardous and not delivered on time:959
     non- Hazardous and delivered on time:2142
     non-Hazardous and not delivered on time:2637

    From Chi-Squared,p value is below 0.05 so we van reject Null
    Hypothesis and say there is correlation between OT_AP and Haz_AP.

  3.Around 60% products are not delivered on time by Fed Post so we
    can say M-Press delivery carrier is best when delivery on time is
    the priority.

    Fed-Post and delivered on Time:933
    Fed-Post and not delivered on Time:2193
    M-press and delivered on Time:1803
    M-press and not delivered on Time:1403

    From Chi-Squared,p value is below 0.05 so we van reject Null
    Hypothesis and say there is correlation between OT_AP and Car_AP.
```

Q3 Model Development As demonstrated in class, create two logistic regression models. 1. A full model using all of the variables. 2. An additional model using backward selection. For each model, interpret and comment on the main measures we discussed in class: (1) AIC (2) Deviance (3) Residual symmetry (4) z-values (5) Parameter Co-Efficients Based on your preceding analysis, recommend which model should be selected and explain why.

Here, A full model using all of the variables I am calling it Model-1

```r #Here, we need to drop Del_AP as we are building model taking OT_AP as a #dependent variable which is created from Del_AP.

Assignment05_AP <- Assignment05_AP[-c(1)] str(Assignment05_AP) ```

`## 'data.frame':    6332 obs. of  7 variables: ##  $ Vin_AP: int  6 18 7 11 12 12 21 12 13 16 ... ##  $ Pkg_AP: int  6 7 7 5 4 3 1 4 6 5 ... ##  $ Cst_AP: int  13 7 8 16 10 5 10 12 8 10 ... ##  $ Mil_AP: int  1447 1874 1865 3111 1319 1415 1599 2361 1394 1121 ... ##  $ Dom_AP: Factor w/ 2 levels "C","I": 1 2 2 2 1 1 1 1 2 2 ... ##  $ Haz_AP: Factor w/ 2 levels "H","N": 1 2 2 1 1 2 1 2 2 1 ... ##  $ Car_AP: Factor w/ 2 levels "Fed Post","M-Press Delivery": 2 1 1 2 1 2 2 2 1 2 ...`

```r #model with all the variables. Fullglm.fit_AP <- glm(OT_AP ~ ., data=Assignment05_AP,

family = “binomial”)

summary(Fullglm.fit_AP) ```

```
Call: ## glm(formula = OT_AP ~ ., family = "binomial", data =
Assignment05_AP) ## ## Deviance Residuals: ##   Min   1Q   Median
3Q   Max ## -3.0579 -0.4641 -0.0798   0.4310   3.3751 ## ##
Coefficients: ##   Estimate Std. Error z value
Pr(>|z|) ## (Intercept)   7.1141805 0.2991113 23.784   < 2e-
16 *** ## Vin_AP   0.0190391 0.0111076   1.714   0.0865
. ## Pkg_AP   0.0231763 0.0201096   1.153   0.2491 ##
Cst_AP   0.0558559 0.0132619   4.212 0.00002533765 *** ##
Mil_AP   -0.0061375 0.0001591 -38.586   < 2e-16 *** ##
Dom_API   -0.7614954 0.0880636 -8.647   < 2e-16 *** ##
Haz_APN   0.5528416 0.0924725   5.978 0.00000000225 *** ##
Car_APM-Press Delivery 2.4106869 0.0921436 26.162   < 2e-16 *** ## ---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ##
(Dispersion parameter for binomial family taken to be 1) ## ##   Null
deviance: 8660.9 on 6331 degrees of freedom ## Residual deviance: 4105.9
on 6324 degrees of freedom ## AIC: 4121.9 ## ## Number of Fisher Scoring
iterations: 6
```

```r #for my knowledge I am experimenting by removing and adding variables #glm.fit_AP1 <- glm(OT_AP ~ Vin_AP + Pkg_AP + Cst_AP + Dom_AP + Haz_AP + Car_AP,data=Assignment05_AP, family = “binomial”)

#summary(glm.fit_AP1)

#glm.fit_AP2 <- glm(OT_AP ~ Vin_AP + Pkg_AP + Mil_AP + Dom_AP + Haz_AP + Car_AP,data=Assignment05_AP, family = “binomial”)

#summary(glm.fit_AP2)

#glm.fit_AP3 <- glm(OT_AP ~ Pkg_AP + Dom_AP + Haz_AP + Car_AP,data=Assignment05_AP, family = “binomial”)

#summary(glm.fit_AP3) ``` Conclusion:

Here, number of iteration is 6 which is good.

(1) AIC - Which indicates Measure of fitness and lower is the better.

(2) Deviance - Null deviance indicates errors when we just make assumption and Residual deviance tell us about summarization of errors in particualr model. Here, Residual deviance is smaller than Null deviance and difference between them is 4555 which is high so our model is good.

(3) Residual symmetry - From 1Q, Median, and 3Q, Residuals are symmetrical.

(4) z-values - From p valueof z-test, all variables are statistically significant but Pkg_AP which has p-value 0.2491

(5) Parameter Co-Efficients - generally it is compared with correlation value and Mil_AP is in positive linear relation but this model gives negative co-efficient for Mil_AP which is not good sign. Moreover, Dom_ap * 1 (if I then -0.7614, if C then 0) Haz_AP * 1 (if N then 0.5528, if H then 0) Car_AP * 1 (if M-Press then 2.41. if Fed Post then 0)

```r #Using Backward Selection (Call it Model-2)

Backstep.fit_AP <- step(Fullglm.fit_AP, direction = “backward”) ```

```
Start: AIC=4121.95 ## OT_AP ~ Vin_AP + Pkg_AP + Cst_AP + Mil_AP + Dom_AP
+ Haz_AP + ##   Car_AP ## ##   Df Deviance   AIC ## - Pkg_AP 1
4107.3 4121.3 ## <none>   4105.9 4121.9 ## - Vin_AP 1   4108.9 4122.9
- Cst_AP 1   4123.8 4137.8 ## - Haz_AP 1   4142.3 4156.3 ## - Dom_AP 1
4183.1 4197.1 ## - Car_AP 1   4999.7 5013.7 ## - Mil_AP 1   8144.7 8158.7
Step: AIC=4121.27 ## OT_AP ~ Vin_AP + Cst_AP + Mil_AP + Dom_AP +
Haz_AP + Car_AP ## ##   Df Deviance   AIC ## <none>   4107.3
4121.3 ## - Vin_AP 1   4110.2 4122.2 ## - Cst_AP 1   4124.9 4136.9 ## -
Haz_AP 1   4143.6 4155.6 ## - Dom_AP 1   4184.3 4196.3 ## - Car_AP 1
5001.0 5013.0 ## - Mil_AP 1   8145.5 8157.5
```

```
r summary(Backstep.fit_AP)
```

```
## ## Call: ## glm(formula = OT_AP ~ Vin_AP + Cst_AP + Mil_AP + Dom_AP +
Haz_AP + ##     Car_AP, family = "binomial", data = Assignment05_AP) ## ##
Deviance Residuals: ##     Min       1Q   Median       3Q      Max ## -3.0412
-0.4666  -0.0804   0.4314   3.3941 ## ## Coefficients: ##
Estimate Std. Error z value      Pr(>|z|) ## (Intercept)
7.2027549  0.2896373  24.868       < 2e-16 *** ## Vin_AP
0.0189735  0.0111066   1.708        0.0876 . ## Cst_AP
0.0555673  0.0132600   4.191 0.00002782316 *** ## Mil_AP                 -
0.0061328  0.0001588 -38.608       < 2e-16 *** ## Dom_API                -
0.7605864  0.0880323  -8.640       < 2e-16 *** ## Haz_APN
0.5526388  0.0924502   5.978 0.00000000226 *** ## Car_APM-Press Delivery
2.4098859  0.0921049  26.165       < 2e-16 *** ## --- ## Signif. codes:  0
'***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ## ## (Dispersion parameter for
binomial family taken to be 1) ## ##     Null deviance: 8660.9  on 6331
degrees of freedom ## Residual deviance: 4107.3  on 6325  degrees of freedom
## AIC: 4121.3 ## ## Number of Fisher Scoring iterations: 6
```

Conclusion:

Regarding Model: when model is constructed with all variable than AIC value is 4121.95. in step-2, if we eliminate Pkg_AP then we will get lower AIC value which is 4121.27. Further, there is no change in AIC when we eliminate variables which are left in step-2 so Process is stopped.

Here, number of iteration is 6 which is good.

Model-2 summary

(1) AIC - Which indicates Measure of fitness and lower is the better. (right now cannot say anything withoud comparing value of AIC with other model which is built on same dataset.)

(2) Deviance - Null deviance indicates errors when we just make assumption and Residual deviance tell us about summarization of errors in particualr model. Here, Residual deviance is smaller than Null deviance and difference between them is 4553.6 which is high so our model is good.(Still we can see how better this model is by comparing another model built on same dataset).

(3) Residual symmetry - From 1Q, Median, and 3Q, Residuals are symmetrical.

(4) z-values - From p value of z-test, all variables are statistically significant.

(5) Parameter Co-Efficients - generally it is compared with correlation value and Mil_AP is

in positive linear relation but this model gives negative co-efficient for Mil_AP which is not good sign. Moreover, Dom_ap * 1 (if I then -0.7605, if C then 0) Haz_AP * 1 (if N then 0.5526, if H then 0) Car_AP * 1 (if M-Press then 2.41. if Fed Post then 0)

Comparing both model and conclusion:

From Number of Fisher Scoring iterations both model are good.

AIC: From AIC value, backward model(Model-2) is slightly better (As lower the AIC the better the model is). However, there is really small difference. (Model-1: 4121.9 and Model-2: 4121.3)

deviances: Model-1 has slightly high difference than model-2 for deviance so Model-1 is slightly better in terms of deviances.

Residucals: in both models, residuals are symmetrical.

Z-values: From p value of Z test, Model-2 is better than Model-1 as in Model-2 all the variables have p value for z test in 0.05 so all variables are significant.

Parameter Co-Efficients - in terms of co-efficients both models has Mil_AP negative but in positive correlation with Del_AP. however, model-2 smaller in size than Model-1.

Overall, both models seem fine but I will choose model-2 from above conclusion of various factors.

PART-B

Logistic Regression – Backward 1. As above, use the step option in the glm function to fit the model (using backward selection). 2. Summarize the results in a Confusion Matrix. 3. As demonstrated in class, calculate the time (in seconds) it took to fit the model and include this in your summary.

#NOTE: I am calling this model as Model-A in further discussion and comparisons.

```r start_time_AP <- Sys.time()

Fullglm.fit_AP1 <- glm(OT_AP ~ ., data=Assignment05_AP, family = “binomial”,na.action=na.omit)

Backstep.fit_AP1 <- step(Fullglm.fit_AP, direction = “backward”) ```

```
Start: AIC=4121.95 ## OT_AP ~ Vin_AP + Pkg_AP + Cst_AP + Mil_AP + Dom_AP
+ Haz_AP + ##   Car_AP ## ##   Df Deviance   AIC ## - Pkg_AP 1
4107.3 4121.3 ## <none>   4105.9 4121.9 ## - Vin_AP 1   4108.9 4122.9
- Cst_AP 1   4123.8 4137.8 ## - Haz_AP 1   4142.3 4156.3 ## - Dom_AP 1
4183.1 4197.1 ## - Car_AP 1   4999.7 5013.7 ## - Mil_AP 1   8144.7 8158.7
Step: AIC=4121.27 ## OT_AP ~ Vin_AP + Cst_AP + Mil_AP + Dom_AP +
Haz_AP + Car_AP ## ##   Df Deviance   AIC ## <none>   4107.3
4121.3 ## - Vin_AP 1   4110.2 4122.2 ## - Cst_AP 1   4124.9 4136.9 ## -
Haz_AP 1   4143.6 4155.6 ## - Dom_AP 1   4184.3 4196.3 ## - Car_AP 1
5001.0 5013.0 ## - Mil_AP 1   8145.5 8157.5
```

```r end_time_AP <- Sys.time()

Backglm_Time_AP <- end_time_AP - start_time_AP

Backglm_Time_AP ```

`## Time difference of 0.4478469 secs`

`r summary(Backstep.fit_AP1)`

```
## ## Call: ## glm(formula = OT_AP ~ Vin_AP + Cst_AP + Mil_AP + Dom_AP +
Haz_AP + ##     Car_AP, family = "binomial", data = Assignment05_AP) ## ##
Deviance Residuals: ##     Min       1Q   Median       3Q      Max ## -3.0412
-0.4666  -0.0804   0.4314   3.3941 ## ## Coefficients: ##
Estimate Std. Error z value      Pr(>|z|) ## (Intercept)
7.2027549  0.2896373  24.868       < 2e-16 *** ## Vin_AP
0.0189735  0.0111066   1.708        0.0876 . ## Cst_AP
0.0555673  0.0132600   4.191 0.00002782316 *** ## Mil_AP              -
0.0061328  0.0001588 -38.608       < 2e-16 *** ## Dom_API             -
0.7605864  0.0880323  -8.640       < 2e-16 *** ## Haz_APN
0.5526388  0.0924502   5.978 0.00000000226 *** ## Car_APM-Press Delivery
2.4098859  0.0921049  26.165       < 2e-16 *** ## --- ## Signif. codes:  0
'***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ## ## (Dispersion parameter for
binomial family taken to be 1) ## ##     Null deviance: 8660.9  on 6331
degrees of freedom ## Residual deviance: 4107.3  on 6325  degrees of freedom
## AIC: 4121.3 ## ## Number of Fisher Scoring iterations: 6
```

Summarize the results in a Confusion Matrix.

```
r resp_glm_AP <- predict(Backstep.fit_AP1, type="response") Class_glm_AP <-
ifelse(resp_glm_AP > 0.5,"1","0") CF_GLM_AP <- table(OT_AP, Class_glm_AP,
dnn=list("Act OT_AP","Predicted") ) CF_GLM_AP
```

```
##          Predicted ## Act OT_AP    0    1 ##         0 3164  432 ##
1  480 2256
```

```r BackTP_AP <- CF_GLM_AP[2,2] BackTN_AP <- CF_GLM_AP[1,1] BackFP_AP <- CF_GLM_AP[1,2] BackFN_AP <- CF_GLM_AP[2,1]

BackAccuracy_AP <- (BackTP_AP + BackTN_AP) / 6332 BackAccuracy_AP ```

`## [1] 0.8559697` here TP = 2256, TN = 3164, FP = 432, FN = 480

1.Accuracy of Backward = TP + TN / Total = (3164 + 2268)/ 6332 = 0.8559

2.Miss Classification Rate = FP + FN / Total = (432 + 480) / 6332 = 0.1440

3.Sensitivity = TP / (TP + FN) = 2256 / (2256 + 480) = 0.8245

4.Specificity = TN / (TN + FP) = 3164 / (3164 + 432) = 0.8798

5.Precision = TP / (TP + FP) = 2256 / (2256 + 432) = 0.8392

6.Prevalance = Actual ‘1’ / Total = (2256 + 480) / 6332 = 0.4320

As demonstrated in class, calculate the time (in seconds) it took to fit the model and include this in your summary.

```
r print(paste("the time (in seconds) it took to fit the Logistic Regression –
Backward:", Backglm_Time_AP))
```

```
[1] "the time (in seconds) it took to fit the Logistic Regression –
Backward: 0.44784688949585"
```
```

Naïve-Bayes Classification 1. Use all the variables in the dataset to fit a Naïve-Bayesian classification model. 2. Summarize the results in a Confusion Matrix. 3. As demonstrated in class, calculate the time (in seconds) it took to fit the model and include this in your summary.

#NOTE: I am calling this model as Model-B in further discussion and comparisons.

```
NBstart_time_AP <- Sys.time()

NB.fit_AP <- NaiveBayes(OT_AP ~ . ,data = Assignment05_AP, na.action=na.omit)

NBend_time_AP <- Sys.time()

NB_Time_AP <- NBend_time_AP - NBstart_time_AP

NB_Time_AP

## Time difference of 0.02713203 secs

pred_bay_AP <- predict(NB.fit_AP,Assignment05_AP)

#Creates Confusion Matrix

CF_NB_AP <- table(Actual=OT_AP, Predicted=pred_bay_AP$class)

#Confusion matrix of Naïve-Bayesian classification.

CF_NB_AP

##         Predicted
## Actual     0    1
##        0 3156  440
##        1  505 2231

NB_TP_AP <- CF_NB_AP[2,2]
NB_TN_AP <- CF_NB_AP[1,1]
NB_FP_AP <- CF_NB_AP[1,2]
NB_FN_AP <- CF_NB_AP[2,1]

NBAccuracy_AP <- (NB_TP_AP + NB_TN_AP) / 6332
NBAccuracy_AP

## [1] 0.8507581
```

TP = 2231, TN = 3156, FP = 440, FN = 505

Accuracy of Naïve-Bayesian = TP + TN / Total = 0.8507

Miss Classification Rate = FP + FN / Total = 0.1492

Sensitivity = TP / (TP + FN) = 0.8352

Specificity = TN / (TN + FP) = 0.8776

Precision = TP / (TP + FP) = 0.8352

Prevalance = Actual ‘1’ / Total = 0.4320

```
print(paste("the time (in seconds) it took to fit the Naïve-Bayesian:",
NB_Time_AP))

## [1] "the time (in seconds) it took to fit the Naïve-Bayesian:
0.0271320343017578"
```

Q3. Linear Discriminant Analysis 1. Use all the variables in the dataset to fit an LDA classification model. 2. Summarize the results in a Confusion Matrix. 3. As demonstrated in class, calculate the time (in seconds) it took to fit the model and include this in your summary.

#NOTE: I am calling this model as Model-C in further discussion and comparisons.

```
LDAstart_time_AP <- Sys.time()

LDA.fit_AP <- lda(OT_AP ~ ., data = Assignment05_AP, na.action=na.omit)

LDAend_time_AP <- Sys.time()

LDA_Time_AP <- LDAend_time_AP - LDAstart_time_AP

LDA_Time_AP

## Time difference of 0.014961 secs

#Predicting LDA Model

LDApred_AP <- predict(LDA.fit_AP, data=Assignment05_AP)

#Confusion matrix for LDA Model.

CF_LDA_AP <- table(Actual=OT_AP, Predicted=LDApred_AP$class)

CF_LDA_AP

##       Predicted
## Actual    0    1
##      0 3157  439
##      1  470 2266

LDA_TP_AP <- CF_LDA_AP[2,2]
LDA_TN_AP <- CF_LDA_AP[1,1]
LDA_FP_AP <- CF_LDA_AP[1,2]
```

```
LDA_FN_AP <- CF_LDA_AP[2,1]

LDA_Accuracy_AP <- (LDA_TP_AP + LDA_TN_AP) / 6332
LDA_Accuracy_AP

## [1] 0.8564435
```

TP = 2266, TN = 3157, FP = 439, FN = 470

Accuracy Of LDA = TP + TN / Total = 0.8564

Miss Classification Rate = FP + FN / Total = 0.1435

Sensitivity = TP / (TP + FN) = 0.8282

Specificity = TN / (TN + FP) = 0.8779

Precision = TP / (TP + FP) = 0.8377

Prevalence = Actual '1' / Total = 0.4320

```
print(paste("the time (in seconds) it took to fit LDA classification:",
LDA_Time_AP ))

## [1] "the time (in seconds) it took to fit LDA classification:
0.0149610042572021"
```

Q4 Compare All Three Classifiers For all questions below please provide evidence.

1. Which classifier is most accurate? (provide evidence)

```
BackAccuracy_AP

## [1] 0.8559697

NBAccuracy_AP

## [1] 0.8507581

LDA_Accuracy_AP

## [1] 0.8564435
```

Accuracy of Backward = TP + TN / Total = (3164 + 2268)/ 6332 = 0.8559 Accuracy Of LDA = TP + TN / Total = 0.8564 Accuracy of Naïve-Bayesian = TP + TN / Total = 0.8507

Conclusion: Accuracy of Linear Discriminant Analysis has the highest.

2. Which classifier is most suitable when processing speed is most important?

```
Backglm_Time_AP

## Time difference of 0.4478469 secs
```

```
NB_Time_AP

## Time difference of 0.02713203 secs

LDA_Time_AP

## Time difference of 0.014961 secs
```

LDA classification model should be considered when processing speed is most important.

NOTE: Here, my model time can be changed when I convert it in pdf and re run the code.

3. Which classifier minimizes false positives?

```
BackFP_AP

## [1] 432

NB_FP_AP

## [1] 440

LDA_FP_AP

## [1] 439

#Note: Here, I have built model on same dataset so I am not considering
#division with total.
```

Logistic Regression – Backward has the fewest false positives among three models which is 432.

4. Which classifier is best overall?

```
#Accuracy
BackAccuracy_AP

## [1] 0.8559697

NBAccuracy_AP

## [1] 0.8507581

LDA_Accuracy_AP

## [1] 0.8564435

#Fewest False Positive
BackFP_AP

## [1] 432

NB_FP_AP

## [1] 440

LDA_FP_AP
```

```
## [1] 439
#Fewest False Negatives
BackFN_AP
## [1] 480
NB_FN_AP
## [1] 505
LDA_FN_AP
## [1] 470
#Less time taken by
Backglm_Time_AP
## Time difference of 0.4478469 secs
NB_Time_AP
## Time difference of 0.02713203 secs
LDA_Time_AP
## Time difference of 0.014961 secs
```

LDA classification model has the highest Accuracy:0.8564 Logistic Regression – Backward has Fewest False Positives : 432 LDA classification model has fewest False Negatives: 470 LDA classification model takes less time than other

Conclusion: From Above factors we can say that LDA classification model superior to others. However if only consider Fewest False Positives than Logistic Regression – Backward is better.

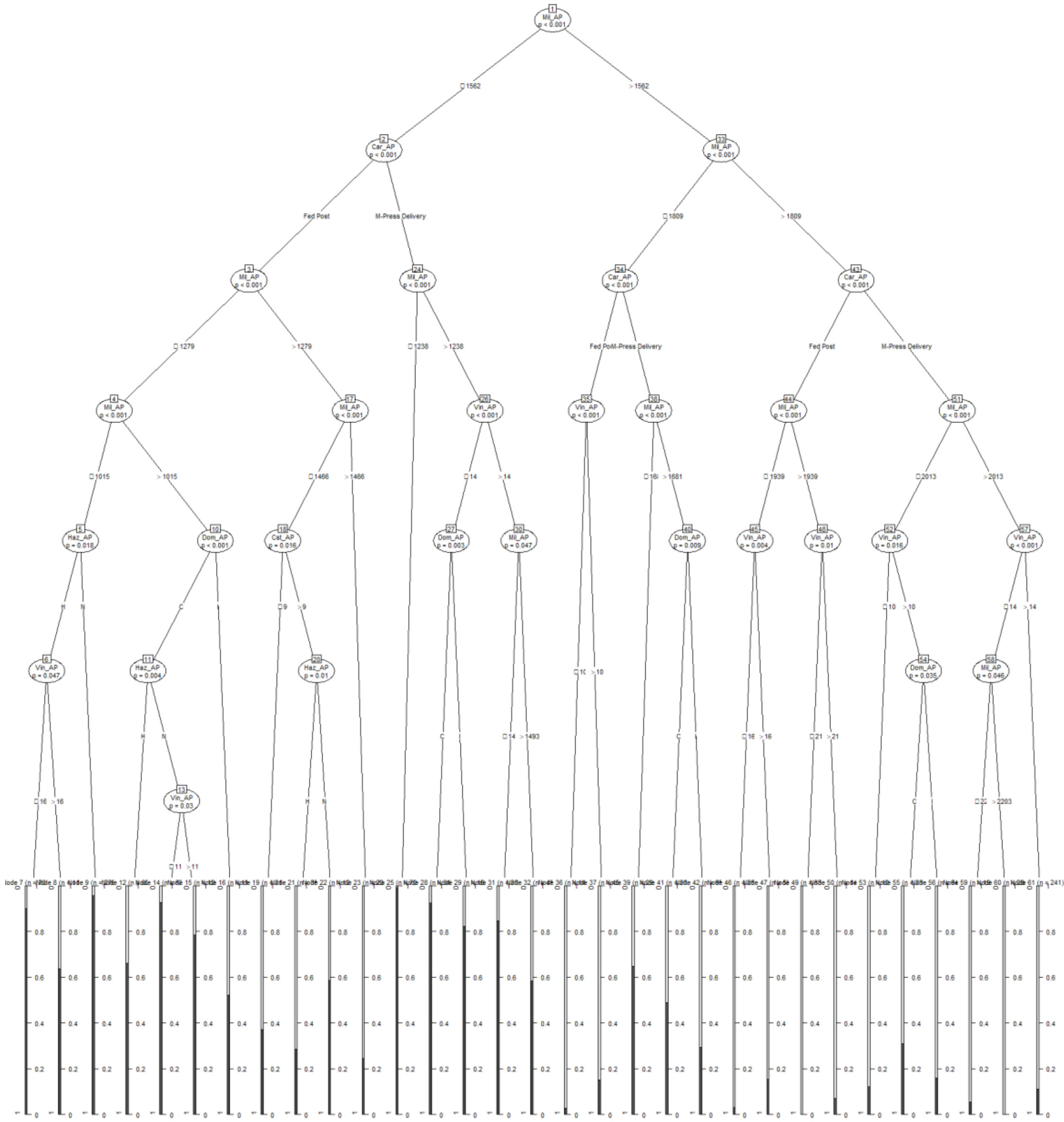
NOTE: Here, my model time can be changed when I convert it in pdf and re run the code.Therefore, time taken by model can be changes as there is minor difference between Naïve-Bayesian and LDA classification model.

#Bonus

Decision Tree

1. Use all the variables in the dataset to fit a Decision Tree classification model.
2. Summarize the results in a Confusion Matrix.
3. As demonstrated in class, calculate the time (in seconds) it took to fit the model and include this in your summary.

```
tree_start_time_AP <- Sys.time()
tree_AP <- ctree(OT_AP ~., data = Assignment05_AP)
plot(tree_AP, gp=gpar (fontsize= 8))
```



```
tree_end_time_AP <- Sys.time()
tree_Time_AP <- tree_end_time_AP - tree_start_time_AP

tree_Time_AP

## Time difference of 1.838158 secs

PredTree_AP <- predict(tree_AP, Assignment05_AP)

CF_tree_AP <- table(Actual=OT_AP, Predicted=PredTree_AP)

CF_tree_AP
```

```
##       Predicted
## Actual    0    1
##      0 3193  403
##      1  504 2232
```

Tree explanation:

1.The value with highest value contains Mil_AP with <=1562, Car_AP with M-Press, Mil_AP <= 1258.

2.The branch which has no positive value contains Mil_AP>2203, Vin_AP <=14, Mil_AP > 2013, Car_AP with M-press delivery, Mil_AP > 1809, Mil_AP >1562

3. Node with less than 0.5 positive value
    a. Cst_AP <+9, Mil_AP <=1466, Mil_AP > 1279, Car_AP with Fed Post, Mil_AP <= 1562

    b. Haz_AP with H, Cst_AP > 9, Mil_AP <+ 1466, Mil_AP > 1279, Car_AP with Fed Post, Mil_AP <= 1562.

    c. Mil_AP > 1466, Mil_AP > 1279, Car_AP with Fed Post, Mil_AP <= m1562.

4. Node with value 0.9 contains Vin_AP <= 16, Haz_AP with H, Mil_AP <= 1015, Mil_AP <= 1279, Car_AP with Fed Post, Mil_AP <= 1562.

TP =2232, TN = 3193, FP = 403, FN = 504

Accuracy = TP + TN / Total = 0.8567

Miss Classification Rate = FP + FN / Total = 0.1432

Sensitivity = TP / (TP + FN) = 0.8157

Specificity = TN / (TN + FP) = 0.8879

Precision = TP / (TP + FP) = 0.8470

Prevalence = Actual '1' / Total = 0.4320

```
print(paste("the time (in seconds) it took to fit Decision Tree:",
tree_Time_AP ))

## [1] "the time (in seconds) it took to fit Decision Tree: 1.83815813064575"
```

NOTE: Here, my model time can be changed when I convert it in pdf and rerun the code.

References:

David Marsh.(2022).[PROG8430-L10-22F].eConestoga.

David Marsh.(2022).[PROG8430-L11-22F].eConestoga.

David Marsh.(2022).[PROG8430-L12-22F].eConestoga.

David Marsh.(2022).[R Documents].eConestoga.

Exploratory Data Analysis with R Peng
https://bookdown.org/rdpeng/exdata/exploratory-graphs.html