**Group Members**

1. Akshay Surve
2. Amay Saxena
3. Aditya Godse

**Supply Chain and Logistic Analytics Group Assignment**

PGP-BABI (Apr 2019-20) - Group 1

**Contents**

[**1.** **Introduction** 2](#_Toc37509857)

[**1.1 Problem at Hand** 2](#_Toc37509858)

[**1.2 Data at Glance** 2](#_Toc37509859)

[**2.** **Exploratory Data Analysis** 2](#_Toc37509860)

[**2.1 Checking for Outliers** 6](#_Toc37509861)

[**2.2 Outlier Treatment** 8](#_Toc37509862)

[**3.** **Splitting Data Into Train And Test** 9](#_Toc37509863)

[**3.1 Initial Insights** 9](#_Toc37509864)

[**3.2 Down sampling major class** 11](#_Toc37509865)

[**4.** **Building a Model** 12](#_Toc37509866)

[**4.1 Predicting Values for Train Dataset** 14](#_Toc37509867)

[**4.2 Predicting Values for Test Dataset** 14](#_Toc37509868)

[**Business Outlook** 15](#_Toc37509869)

# **Introduction**

## **1.1 Problem at Hand**

  
Data is given by a company. It contains around 7000 transactions. Column headers are self-explanatory. Your delivery in-charge has been very professional in choosing the appropriate shipping mode. Your endeavor is to find out if you can build a model to predict correct shipping mode.

## **1.2 Data at Glance**

Loading dataset,

setwd("C:/Users/DELL/Desktop/Akshay/Group Assignments/Group Asssignment SCLA")

getwd()

library(readxl)

datafile <- read.csv("C:/Users/DELL/Desktop/Akshay/Group Assignments/Group Asssignment SCLA/Inventory CSV.csv")

Number of Observations and attributes in Data set,

dim(datafile)

[1] 7853 8

Data has 8 attributes and 7853 number of observations.

Checking if dataset contains any Missing values,

sum(is.na(datafile))

[1] 0

# **Exploratory Data Analysis**

Structure of dataset,

str(datafile)

'data.frame': 7853 obs. of 8 variables:

$ Order.Date : Factor w/ 1378 levels "1/1/2008","1/1/2009",..: 61 61 61 61 61 61 61 61 61 65 ...

$ Order.ID : int 24544 24544 24544 20422 55937 20422 17186 55937 17186 11686 ...

$ Order.Quantity : int 31 39 15 30 10 5 11 24 49 38 ...

$ Product.Container : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 4 3 2 6 5 5 5 1 4 5 ...

$ Product.Name : Factor w/ 1257 levels "\"While you Were Out\" Message Book, One Form per Page",..: 316 482 561 814 662 188 1184 295 405 1092 ...

$ Product.Sub.Category: Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 10 15 4 9 5 8 11 16 9 17 ...

$ Sales : num 6567 1780 578 611 517 ...

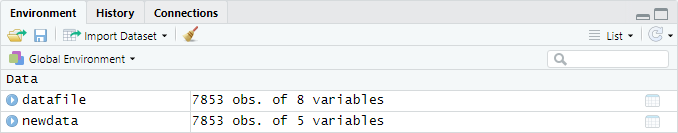
$ Ship.Mode : Factor w/ 3 levels "Delivery Truck",..: 2 3 1 3 3 3 3 1 3 3 ...

For now, Variables like Order Date, Order ID and Product Name are not relevant in data analysis and model building and can be removed.

newdata <- subset(datafile, select = (-c(1,2,5)))

dim(newdata)

[1] 7853 5



**Target variable** considered in this case is **Ship mode** and the rest are independent variables. Also, **Product Container**, **Product Sub-Category** and **Ship mode** are of categorical nature. Hence, declaring these variables as factor variables.

newdata$Product.Container <- as.factor(newdata$Product.Container)

newdata$Product.Sub.Category <- as.factor(newdata$Product.Sub.Category)

newdata$Ship.Mode <- as.factor(newdata$Ship.Mode)

Rechecking structure of dataset to see nature of attributes,

str(newdata)

'data.frame': 7853 obs. of 5 variables:

$ Order.Quantity : int 31 39 15 30 10 5 11 24 49 38 ...

$ Product.Container : Factor w/ 7 levels "Jumbo Box","Jumbo Drum",..: 4 3 2 6 5 5 5 1 4 5

$ Product.Sub.Category: Factor w/ 17 levels "Appliances","Binders and Binder Accessories",..: 10 15 4 9 5 8 11 16 9 17 ...

$ Sales : num 6567 1780 578 611 517 ...

$ Ship.Mode : Factor w/ 3 levels "Delivery Truck",..: 2 3 1 3 3 3 3 1 3 3 ...

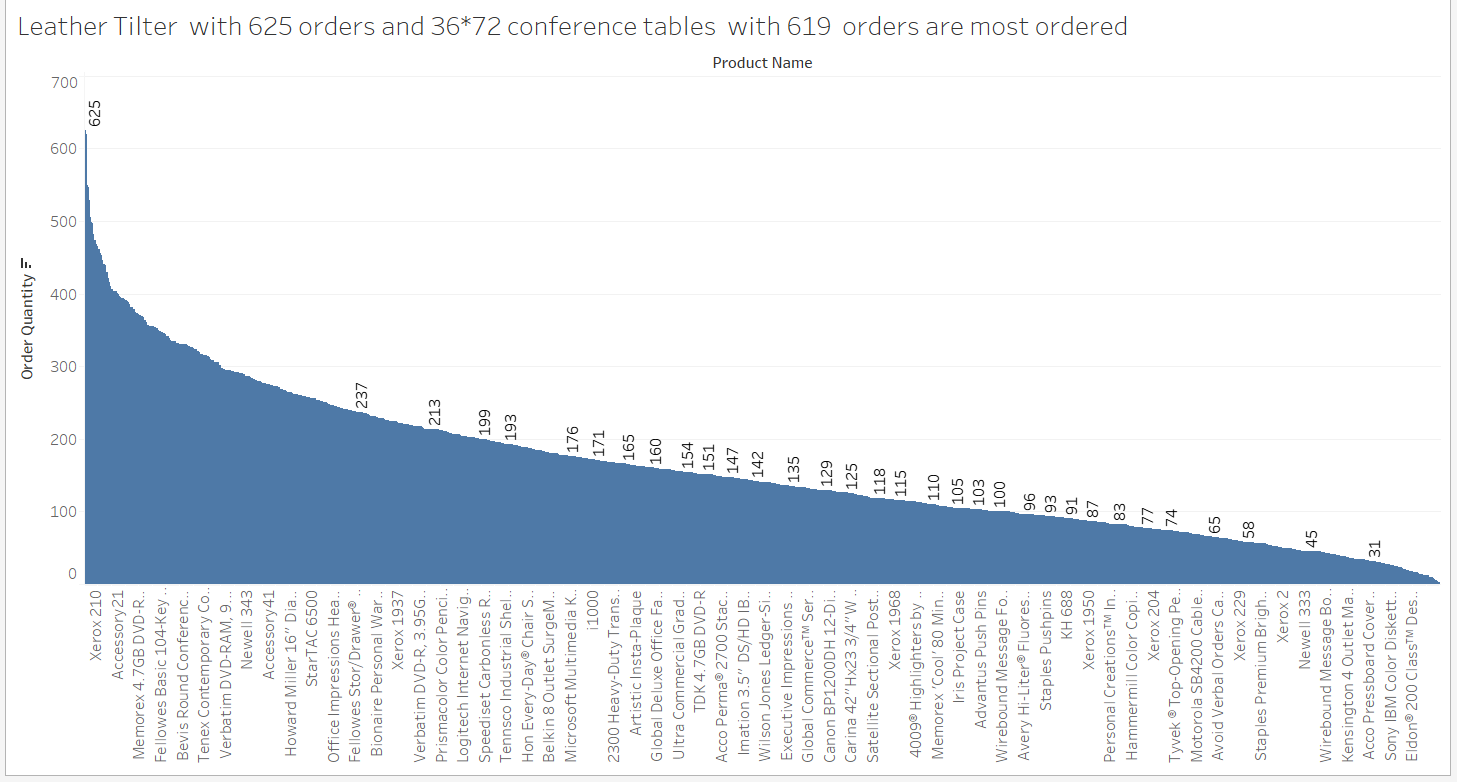


Fig1. Productname vs. Orders

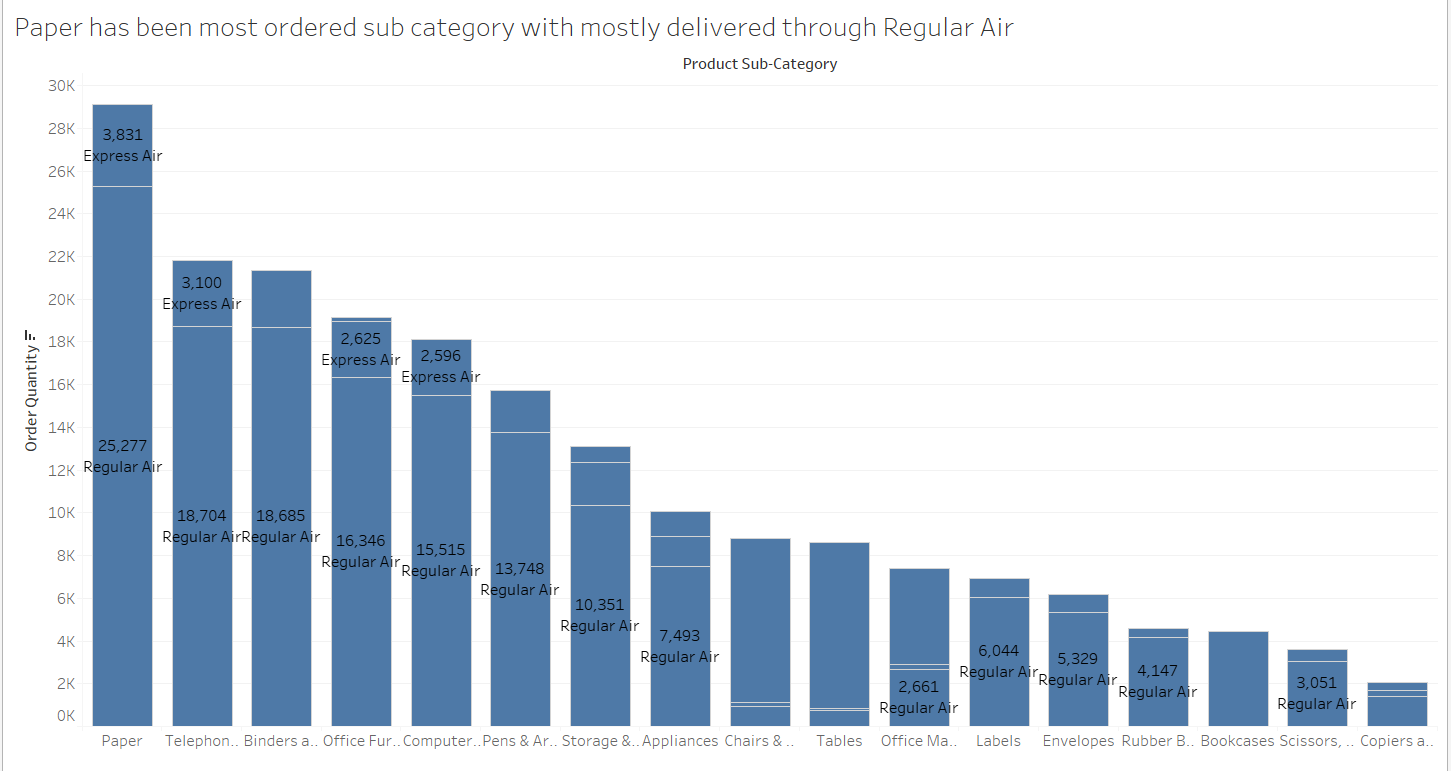


Fig2. Sub category vs. Orders

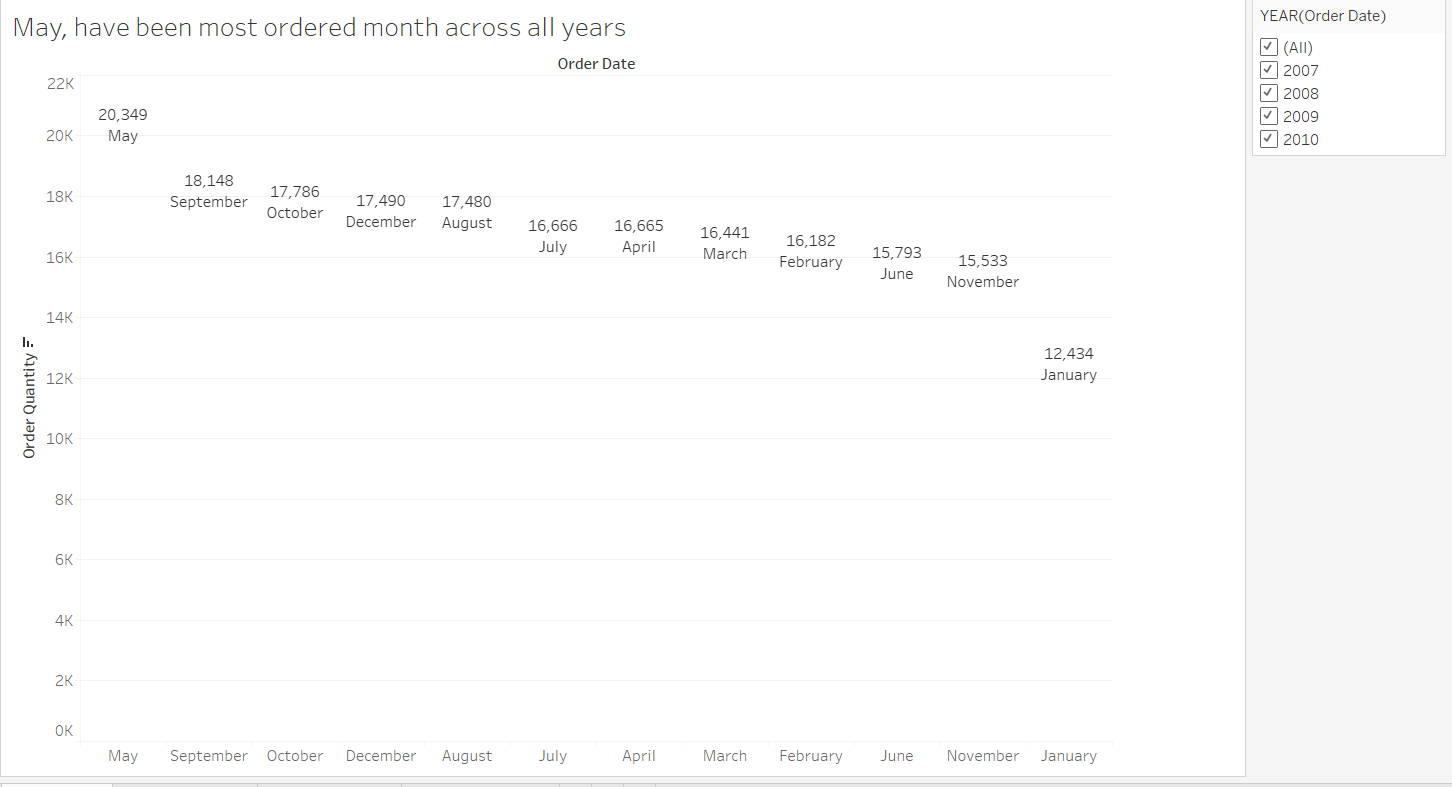


Fig3. Order date vs. Orders

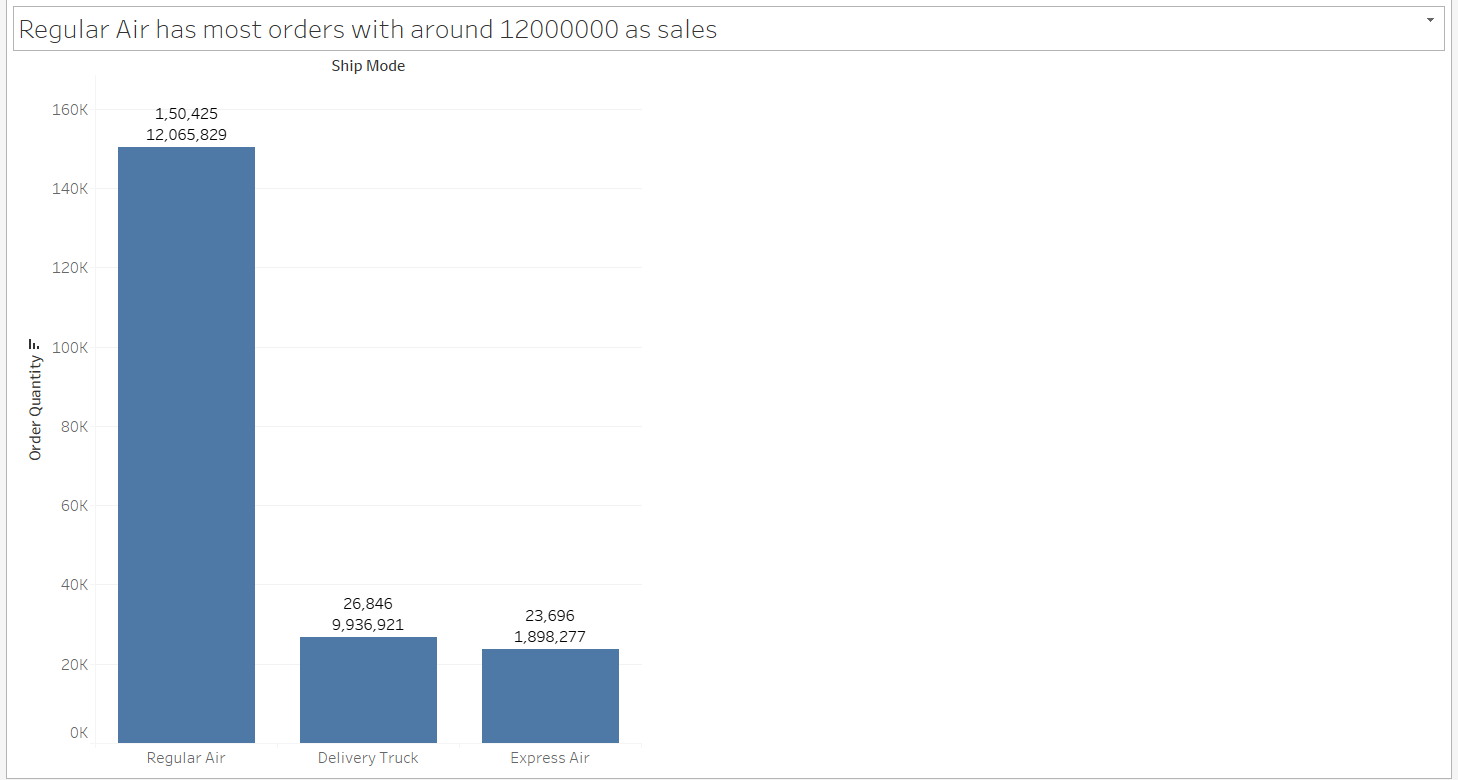


Fig4. Ship mode vs. Orders

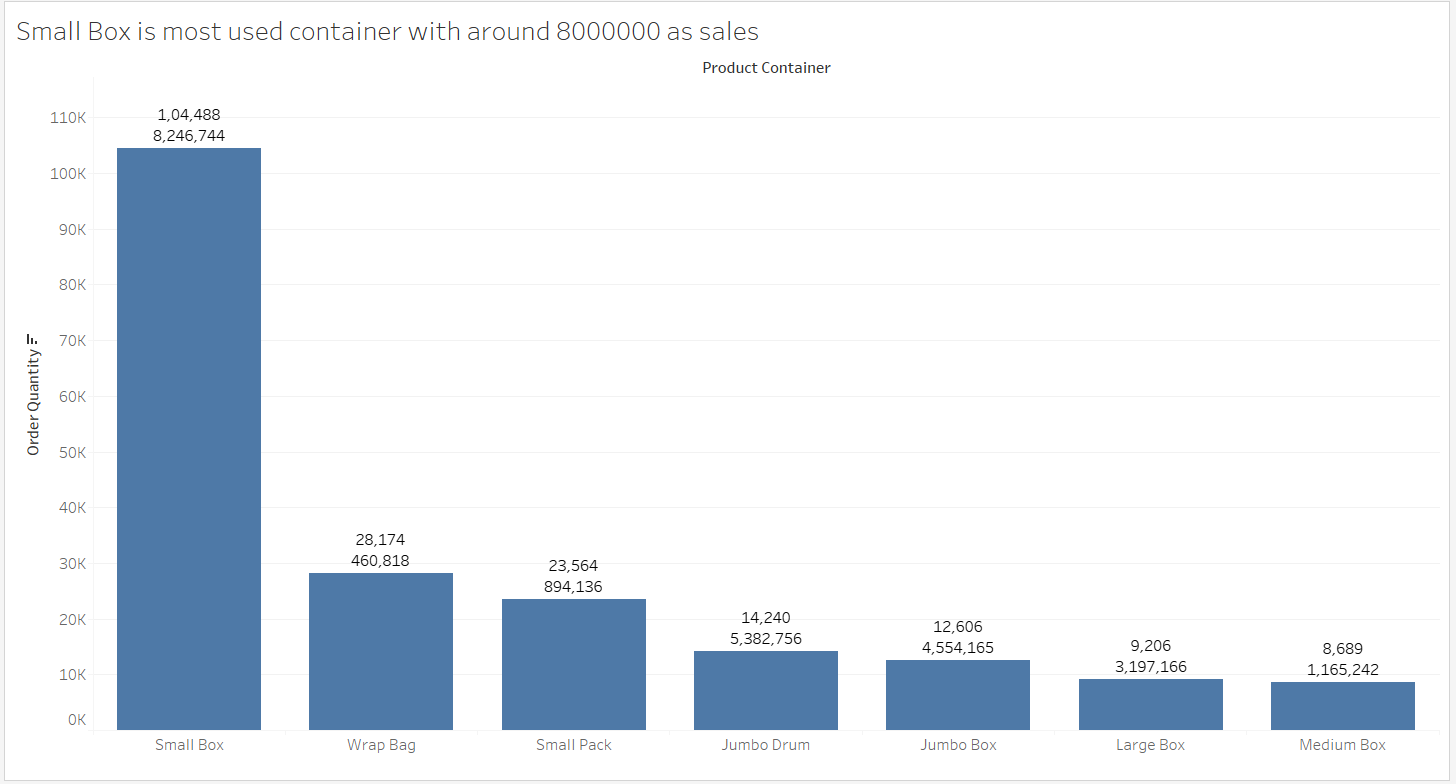


Fig5. Container vs. Orders

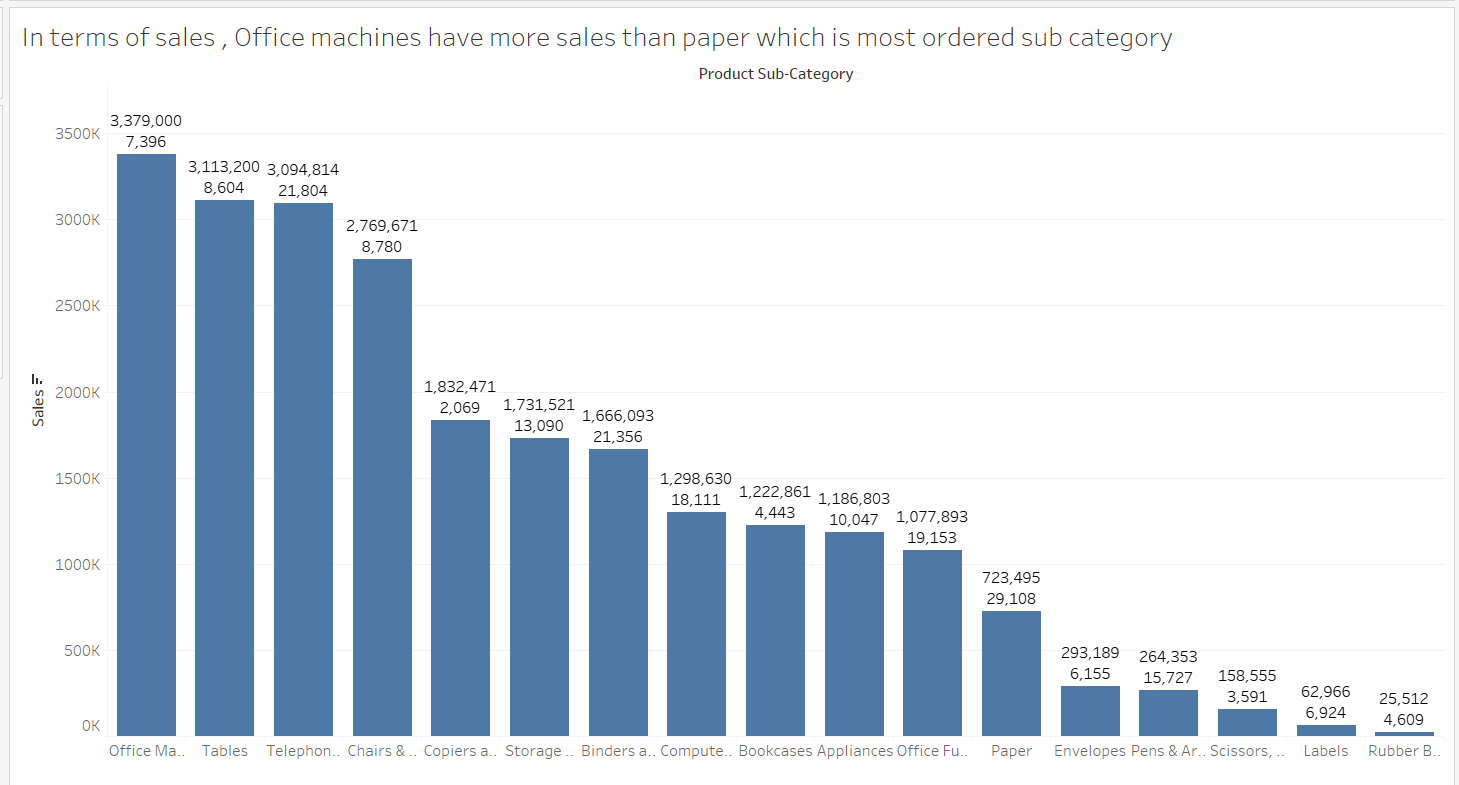


Fig6. Sub category vs. Sales

## **2.1 Checking for Outliers**

Data summary in this case will give slight idea about attributes which may contain such higher values.

summary(newdata)

Order.Quantity Product.Container Product.Sub.Category

Min. : 1.00 Jumbo Box : 490 Paper :1153

1st Qu.:13.00 Jumbo Drum: 573 Binders and Binder Accessories: 857

Median :26.00 Large Box : 378 Telephones and Communication : 831

Mean :25.59 Medium Box: 344 Office Furnishings : 726

3rd Qu.:38.00 Small Box :4081 Computer Peripherals : 715

Max. :50.00 Small Pack: 894 Pens & Art Supplies : 589

Wrap Bag :1093 (Other) :2982

Sales Ship.Mode

Min. : 4 Delivery Truck:1063

1st Qu.: 244 Express Air : 921

Median : 747 Regular Air :5869

Mean : 3044

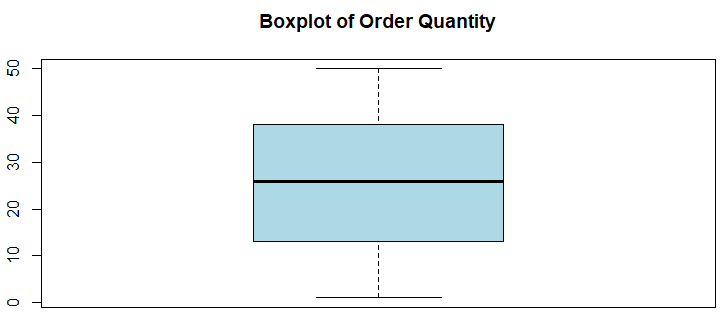
3rd Qu.: 2959

Max. :114362

Taking Boxplot to confirm presence of outliers,

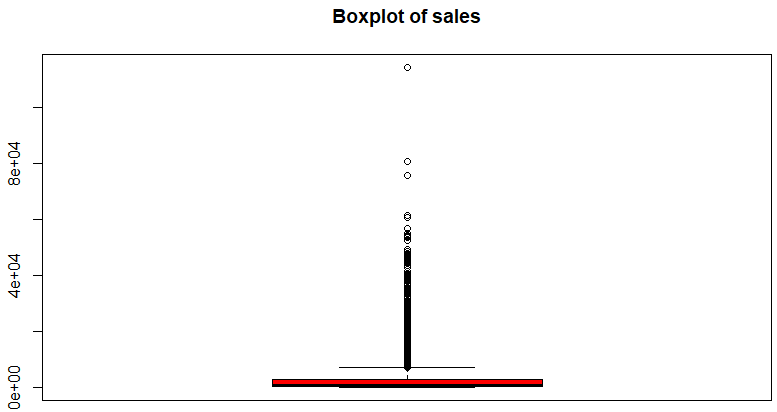
Taking the boxplot of attributes which have numeric data only, so considering this only **Order Quantity** and **Sales** have been considered.

boxplot(newdata$Order.Quantity, col = "light blue", main = "Boxplot of Order Quantity")



Boxplot of order quantity indicates there are no outliers in attribute order quantity.

boxplot(newdata$Sales, col = "red", main = "Boxplot of sales")



Boxplot of sales indicates complete different scenario here. There are many outliers. In order to avoid results being hampered by this, corrective action is required.

Above all the attributes, it seems the attribute Sales has extreme values towards both smaller and higher end. This needs to be handled.

## **2.2 Outlier Treatment**

We have used a common definition of an outlier that will use anything less the first quartile - 1.5 \* the interquartile range or above the third quartile + 1.5 \* the interquartile range is considered an outlier. The interquartile range (IQR) is the range between the first quartile and the third quartile (the middle 50% of the data).

So any value above or below IQR will be considered outlier and these extreme values will automatically get replaced by these middle 50% values.

capOutlier <- function(x){

qnt <- quantile(x, probs=c(.25, .75), na.rm = T)

caps <- quantile(x, probs=c(.05, .95), na.rm = T)

H <- 1.5 \* IQR(x, na.rm = T)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

return(x)

}

newdata$Sales = capOutlier(newdata$Sales)

summary(newdata$Sales)

Min. 1st Qu. Median Mean 3rd Qu. Max.

4 244 747 2752 2959 13386

Sales

Min. : 4

1st Qu.: 244

Median : 747

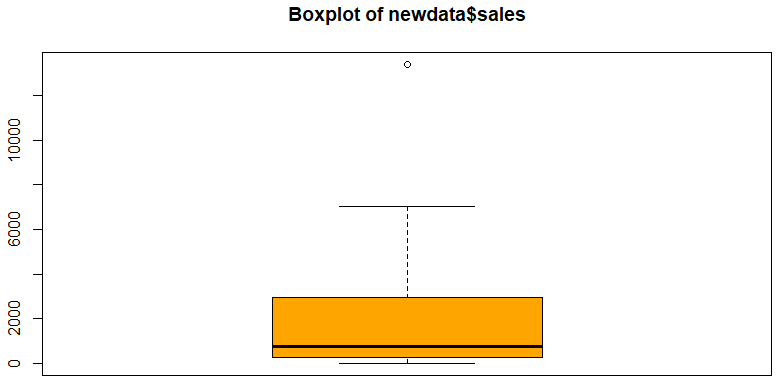
Mean : 3044

3rd Qu.: 2959

Max. : 114362

From above it can be seen that, like in previous case where data was distributed with extreme values has been replaced in later case, where outliers have been handled with proper values using IQR.

boxplot(newdata$Sales, col = "orange", main = "Boxplot of newdata$sales")



From the boxplot of Sales, it can be seen that outliers have been well handled.

# **Splitting Data Into Train And Test**

## **3.1 Initial Insights**

The imported data after dropping of redundant variables has been split into training and test sets using sample. Split on Ship.Mode variable with split ratio=0.7 from “caret” package. The models have been trained on training set and will be validated on test set.

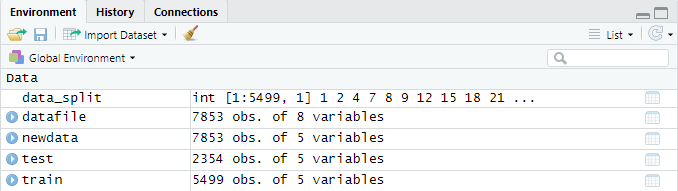
library(caret)

set.seed(123)

data\_split<-createDataPartition(newdata$Ship.Mode, p=0.7,list = FALSE,times = 1)

train<-newdata[data\_split,]

test<-newdata[-data\_split,]



dim(train)

[1] 5499 5

dim(test)

[1] 2354 5

Now check if there is class imbalance in the data after split,

table(train$Ship.Mode)

Delivery Truck Express Air Regular Air

745 645 4109

Percentage distribution of data after split,

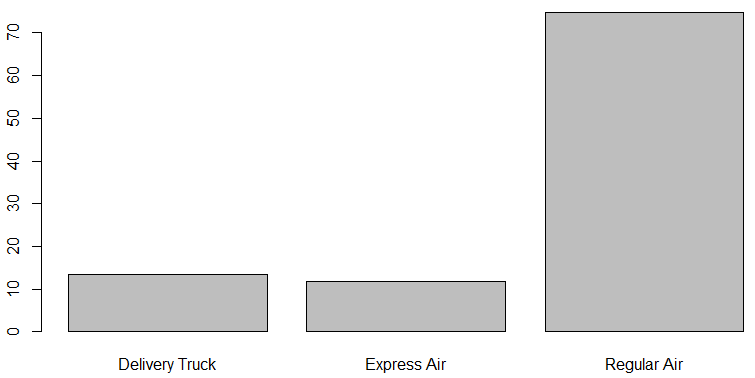
distribution = round(prop.table(table(train$Ship.Mode))\*100,2)

distribution

Delivery Truck Express Air Regular Air

13.55 11.73 74.72

bp = barplot(distribution)



From the graph it can be clearly seen that, there is a class imbalance in the dataset, data on Regular Air is more than 75% against Express Air which is at around 10% and Delivery Truck is at around 15%. With this data imbalance we are not going to get accurate model.

## **3.2 Down sampling major class**

Now, to address this data imbalance problem, down sampling needs to be done for majority class i.e. Regular Air in this case. Therefore, we are going to down sample the majority class so that it will be of the same size as smallest class.

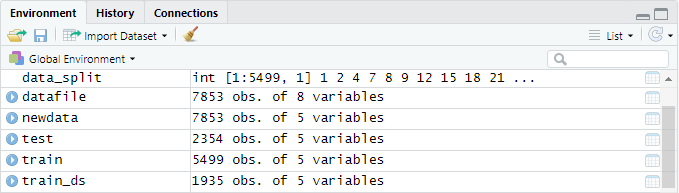
library(caret)

set.seed(123)

'%ni%' <- Negate('%in%')

train\_ds <- downSample(x=train[,colnames(train) %ni% "Ship.Mode"],

y = train$Ship.Mode, yname = "Ship.Mode")



Data has been downsized quite a bit.

table(train\_ds$Ship.Mode)

Delivery Truck Express Air Regular Air

645 645 645

Now that we have equal proportion of response classes for shipping modes, we can go ahead and build model using this down sampled data set.

# **Building a Model**

Dependent variable in our problem is Ship mode and has a nature of Multinomial category with 3 class values namely Delivery Truck, Express Air, Regular Air. Because of this, nature of our problem is of multinomial logistic regression and hence, Implementing Neural Net algorithm.

ml\_model <- nnet::multinom(train\_ds$Ship.Mode ~., data = train\_ds)

# weights: 78 (50 variable)

initial value 2125.814779

iter 10 value 990.328089

iter 20 value 911.148115

iter 30 value 895.315876

iter 40 value 884.930203

iter 50 value 884.758240

final value 884.757923

converged

summary(ml\_model)

Call:

nnet::multinom(formula = train\_ds$Ship.Mode ~ ., data = train\_ds)

Coefficients:

(Intercept) Order.Quantity Product.ContainerJumbo Drum

Express Air -36.04543 -0.3048416 6.545478

Regular Air -35.85529 -0.3043654 5.715302

Product.ContainerLarge Box Product.ContainerMedium Box

Express Air 81.43823 94.84623

Regular Air 80.80665 94.11141

Product.ContainerSmall Box Product.ContainerSmall Pack

Express Air 69.09652 71.23035

Regular Air 68.33600 70.72351

Product.ContainerWrap Bag

Express Air 68.39363

Regular Air 67.83354

Product.Sub.CategoryBinders and Binder Accessories

Express Air -3.182933

Regular Air -2.554824

Product.Sub.CategoryBookcases

Express Air -7.515255

Regular Air -9.110727

Product.Sub.CategoryChairs & Chairmats

Express Air -13.91165

Regular Air -13.06266

Product.Sub.CategoryComputer Peripherals

Express Air -7.180511

Regular Air -6.782357

Product.Sub.CategoryCopiers and Fax

Express Air -17.60234

Regular Air -18.02570

Product.Sub.CategoryEnvelopes Product.Sub.CategoryLabels

Express Air 6.544618 5.463510

Regular Air 7.399951 6.212225

Product.Sub.CategoryOffice Furnishings

Express Air -1.2303356

Regular Air -0.9866816

Product.Sub.CategoryOffice Machines Product.Sub.CategoryPaper

Express Air -5.871621 -2.989090

Regular Air -5.500257 -2.530437

Product.Sub.CategoryPens & Art Supplies

Express Air 1.289791

Regular Air 1.840266

Product.Sub.CategoryRubber Bands

Express Air 13.18910

Regular Air 14.07124

Product.Sub.CategoryScissors, Rulers and Trimmers

Express Air 11.93648

Regular Air 12.71241

Product.Sub.CategoryStorage & Organization

Express Air -3.874152

Regular Air -3.288240

Product.Sub.CategoryTables

Express Air -7.107164

Regular Air -7.149728

Product.Sub.CategoryTelephones and Communication Sales

Express Air -3.931078 0.001482672

Regular Air -3.677337 0.001497483

Std. Errors:

(Intercept) Order.Quantity Product.ContainerJumbo Drum

Express Air 0.04322529 0.002054372 1.129050e-10

Regular Air 0.04322529 0.002054372 1.007997e-10

Product.ContainerLarge Box Product.ContainerMedium Box

Express Air 0.004685188 0.00439628

Regular Air 0.004685188 0.00439628

Product.ContainerSmall Box Product.ContainerSmall Pack

Express Air 0.03626539 0.01891452

Regular Air 0.03626539 0.01891452

Product.ContainerWrap Bag

Express Air 0.02760159

Regular Air 0.02760159

Product.Sub.CategoryBinders and Binder Accessories

Express Air 0.01

Regular Air 0.01

Product.Sub.CategoryBookcases

Express Air 9.712289e-16

Regular Air 2.775215e-16

Product.Sub.CategoryChairs & Chairmats

Express Air 0.001059331

Regular Air 0.001059331

Product.Sub.CategoryComputer Peripherals

Express Air 0.006598382

Regular Air 0.006598382

Product.Sub.CategoryCopiers and Fax

Express Air 0.0005127107

Regular Air 0.0005127107

Product.Sub.CategoryEnvelopes Product.Sub.CategoryLabels

Express Air 0.002256308 0.003065255

Regular Air 0.002256308 0.003065255

Product.Sub.CategoryOffice Furnishings

Express Air 0.008231206

Regular Air 0.008231206

Product.Sub.CategoryOffice Machines Product.Sub.CategoryPaper

Express Air 0.001257666 0.0108044

Regular Air 0.001257666 0.0108044

Product.Sub.CategoryPens & Art Supplies

Express Air 0.01478882

Regular Air 0.01478882

Product.Sub.CategoryRubber Bands

Express Air 0.004216091

Regular Air 0.004216091

Product.Sub.CategoryScissors, Rulers and Trimmers

Express Air 0.002910778

Regular Air 0.002910778

Product.Sub.CategoryStorage & Organization

Express Air 0.00280559

Regular Air 0.00280559

Product.Sub.CategoryTables

Express Air 0.0004000746

Regular Air 0.0004000746

Product.Sub.CategoryTelephones and Communication Sales

Express Air 0.005590007 8.606657e-06

Regular Air 0.005590007 8.594971e-06

Residual Deviance: 1769.516

AIC: 1869.516

## **4.1 Predicting Values for Train Dataset**

train\_accuracy <- predict(ml\_model, newdata = train\_ds, "class")

Building classification table on train dataset, (Confusion Matrix)

train\_table = table(train\_ds$Ship.Mode, train\_accuracy)

train\_table

train\_accuracy

Delivery Truck Express Air Regular Air

Delivery Truck 645 0 0

Express Air 0 339 306

Regular Air 0 277 368

Calculating accuracy on Train data,

round((sum(diag(train\_table))/sum(train\_table))\*100,2)

[1] 69.87

The accuracy for train data set come out to be 69.87%.

## **4.2 Predicting Values for Test Dataset**

test\_accuracy <- predict(ml\_model, newdata = test, "class")

Building classification table on test dataset, (Confusion Matrix)

test\_table <- table(test$Ship.Mode, test\_accuracy)

test\_table

test\_accuracy

Delivery Truck Express Air Regular Air

Delivery Truck 318 0 0

Express Air 0 135 141

Regular Air 0 872 888

Calculating accuracy on Test data,

round((sum(diag(test\_table))/sum(test\_table))\*100,2)

[1] 56.97

The accuracy for test data set come out to be 56.97 % which is less compared to train accuracy 69.82 %.

Precision = (1\*318+0.1340\*276+0.8629\*1760)/2354 = 0.7959

Recall = (1\*318+0.4891\*276+0.5045\*1760)/2354 = 0.5696

|  |  |
| --- | --- |
|  | **Neural Network** |
| **Accuracy=(TP+TN)/total** | **0.5697** |
| **Misclassification Rate or error rate** | **0.4303** |
| **Sensitivity / recall / True Positive Rate** | **0.5696** |
| **Precision** | **0.7959** |

# **Business Outlook**

So, As Delivery in-charge, I would like to go by Regular Air and Delivery Truck for transportation as using Express Air can be a loss as we can see that accuracy has come down due to Express Air predictions. Also, we can see in most sub-categories which have higher order numbers, they have either Regular Air or Express Air in dominance as delivery option, so Express Air can be replaced by Regular Air, as that can be seen from Confusion matrix and that will also increase accuracy of model and profit margin for company in end.