A Project Report on

BIG MARKET SALES PREDICTION

Submitted in partial fulfilment for the award of the degree of

B.Tech (Computer Science and Engineering)

By

Raghav Jindal - 18BCE2080

Nimish Batra - 18BCE2087

Prepared For

CSE3020 - Data Visualization

Under the guidance of

Prof. Pravat Kumar Jena

Associate Professor

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

2021- 2022

TABLE OF CONTENTS

1. ABSTRACT:

Nowadays shopping malls and Big Marts keep track of their sales data of each and every individual item for predicting future demand of the customer and update the inventory management as well.
These data stores basically contain a large number of customer data and individual item attributes in a data warehouse. Further, anomalies and frequent patterns are detected by mining the data store from the data warehouse. The resultant data can be used for predicting future sales volume with the help of different machine learning techniques for the retailers like Big Mart. In this paper, we propose a predictive model using Xgboost technique for predicting the sales of a company like Big Mart and found that the model produces better performance as compared to existing models. A comparative analysis of the model with others in terms performance metrics is also explained in detail.

2. INTRODUCTION:

2.1 BACKGROUND:

- Day by day competition among different shopping malls as well as big marts is getting more serious and aggressive only due to the rapid growth of the global malls and on-line shopping. Every mall or mart is trying to provide personalized and short-time offers for attracting more customers depending upon the day, such that the volume of sales for each item can be predicted for inventory management of the organization, logistics and transport service, etc.
- Present machine learning algorithms are very sophisticated and provide techniques to predict or forecast the future demand of sales for an organization, which also helps in overcoming the cheap availability of computing and storage systems.
- In this project, we are addressing the problem of big mart sales prediction or forecasting of an item on customer's future demand in different big mart stores across various locations and products based on the previous record.
- Different machine learning algorithms like linear regression analysis, random forest, etc are used for prediction or forecasting of sales volume.

- As good sales are the life of every organization so the forecasting of sales plays an important role in any shopping complex. Always a better

prediction is helpful, to develop as well as to enhance the strategies of business about the marketplace which is also helpful to improve the knowledge of the marketplace.

- A standard sales prediction study can help in deeply analyzing the situations or the conditions previously occurred and then, the inference can be applied about customer acquisition, funds inadequacy and strengths before setting a budget and marketing plans for the upcoming year.

2.2 OBJECTIVE:

- The aim is to build a predictive model and find out the sales of each product at a particular store and hence predict the sales of supermarkets according to the sample supermarket dataset. The idea is to find out the properties of a product, and store which impacts the sales of a product. Using this model, we will try to understand the properties of products and stores which play a key role in increasing sales.
- We will also use various visual plots between the data to better understand it.
- Algorithm we will use to build our models are:
  - ➢ Linear Regression
  - ➢ Ridge Regression
  - ➢ Decision Tree Regression

2.3 MOTIVATION:

Seeing the tremendous growth of sales in big markets, it has become very important for the owner of the big markets to visualize the data in order to gain various insights of the data. Understanding the data will help the owner determine the best product in his/her market and discard those products that are less preferred to be sold. As the products are ordered in bulk in big markets, visualization of data will allow the owner to order those products that are sold the most so that all the products ordered by the owner are sold to the customers. Analysing the data is also important as it will help the owner to predict the sales of their products. This has motivated us to take up the project so that we can help the big market owners to analyse their data and grow their business.

2.4 CONTRIBUTION OF THE PROJECT:

Both of us contributed equally towards analyzing the visualizations of different attributes using scatter plots, bar plots, box plots, cow plot, correlation plots, etc.
Then we studied about the three regression models - linear, ridge and decision tree and using these models we worked on predicting the overall sales for the supermarket.

2.5 ORGANIZATION OF THE PROJECT:

- Business User: Super markets like Big Mart etc.
- Project Manager: Ensures that key milestones and objectives are met on time and at the expected quality.
- Business Intelligence Analyst: Provides business domain expertise based on a deep understanding of the data, key performance indicators (KPIs), key metrics, and business intelligence from a reporting perspective. Business Intelligence Analysts generally create dashboards and reports and know the data feeds and sources.
- Database Administrator (DBA): Provisions and configures the database environment to support the analytics needs of the working team. These responsibilities may include providing access to key databases or tables and ensuring the appropriate security levels are in place related to the data repositories.
- Data Scientist: Provides subject matter expertise for analytical techniques, data modeling, and applying valid analytical techniques to given business problems. Ensures overall analytics objectives are met. Designs and executes analytical methods and approaches with the data available to the project.

3. PROJECT RESOURCE REQUIREMENTS:

3.1 SOFTWARE REQUIREMENTS:

- R x64 3.6.1 - It provides the R library functions which can be used to work, analyze and visualize various datasets.
- RStudio - RStudio is an integrated development environment for R, a programming language for statistical computing and graphics. ● Google Collab - Collab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members.

4. LITERATURE SURVEY:

4.1 BACKGROUND:

We started with making some hypotheses about the data without looking at it. Then we moved on to data exploration where we found out some nuances in the data which required remediation. Next, we performed data cleaning and feature engineering, where we imputed missing values and solved other irregularities, made new features and also made the data model-friendly by one-hot-coding. Finally, we made a linear regression, decision tree and ridge regression model and got a glimpse of how to tune them for better results.

4.2 LITERATURE REVIEW:

Research Paper	Authors	Published Year	Description
Business data mining machine learning perspective. Information & management 39(3), 211–225	Bose, I., Mahapatra, R.K.	2001	This paper mainly talks about the various statistical and computational methods using machine learning techniques. It also elaborates upon the automated process of knowledge acquisition in the field of Machine Learning. Various machine learning (ML) techniques with their applications in different sectors are also presented in the paper
A few useful things to know about machine learning. Commun. acm 55(10), 78–87	Domingos, P.M.	2012	Machine learning is the process where a machine will learn from data in the form of statistically or computationally method and process knowledge acquisition from experiences. This paper focuses on the above fact and beautifully explains

			the concept of machine learning.
Applications of machine learning and rule induction. Communications of the ACM 38(11), 54– 64	Langley, P., Simon, H.A.	1995	This paper mainly points out that the most widely used data mining technique in the field of business is the Rule Induction (RI) technique as compared to other data mining techniques.
A two-level statistical model for big mart sales prediction. IEEE	Punam, K., Pamula, R., Jain, P.K	2018	This paper presents forth a two-level statistical model for analysing the big market sales which provides us a guide to start off with our project.

A comparative study of linear and nonlinear models for aggregate retail sales forecasting. International Journal of production economics 86(3), 217–231	Chu, C.W., Zhang, G.P.	2003	Linear and non-linear comparative analysis models for sales forecasting is proposed for the retailing sector. This paper provides forth a comparison between the linear and non-linear models helping us to determine when to use each of them.
A seasonal discrete grey forecasting model for fashion retailing. Knowledge-Based Systems 57, 119– 126	Xia, M., Wong, W.K	2014	This paper proposes the differences between classical methods (based on mathematical and statistical models) and modern heuristic methods and also named exponential smoothing,

			regression, auto regressive integrated moving average (ARIMA), generalized auto regressive conditionally heteroscedastic (GARCH) methods.
Forecasting methods and applications.	Makridakis, S., Wheelwright, S.C., Hyndman, R.J.	2008	This paper mainly focuses on the challenges that are faced by linear models to deal with the asymmetric behaviour in most real-world sales data. Some of the challenging factors include lack of historical data, consumer-oriented markets face uncertain demands, and short life cycles of prediction methods.

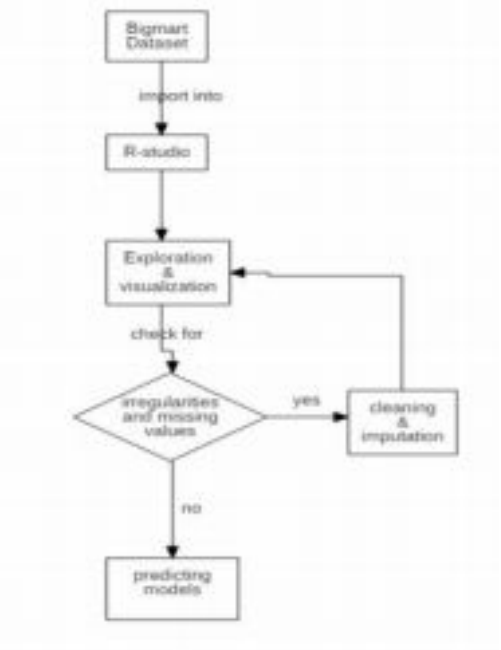
Ridge Regression in Practice	Donald W. Marquardt, Ronald D. Snee	2012	A review of the theory of ridge regression and its relation to generalized inverse regression is presented along with the results of a simulation experiment and three examples of the use of ridge regression in practice.
Prediction of retail sales of footwear using feedforward and recurrent neural networks. Neural Computing and Applications 16(4- 5), 491–502	Das, P., Chaudhury, S.	2007	This paper focuses on using neural networks for predicting weekly retail sales, which decrease the uncertainty present in the short-term planning of sales. This helps in understanding how we can also use neural networks in the field of prediction and even in big mart sales analysis.

4.3 SUMMARY:

Sales forecasting as well as analysis of sale forecasting has been conducted by many authors. Not only in the field of big markets but also in various other fields consisting of huge amounts of data various kinds of forecasting has already been carried out. Researchers have used various data visualization and analysis techniques in order to interpret various results from the collected data which has helped owners of different markets to increase their sales. In all the papers reviewed by us, we have found out various techniques that have been used by researchers to analyze and predict the data. Depending on the dataset we are working upon, we figured out some of the best techniques suitable for our dataset which we have presented in our report. We also found out that the data visualization was not properly carried out in any of the papers including analysis of data. Thus, we have presented in our report both the data visualization and analysis techniques for the dataset we have chosen.

5. PROPOSED METHODOLOGY:

5.1 PROPOSED ARCHITECTURE:



5.2 TABLE & CONSTRAINTS:

Attribute	Datatype	Constraint
Item_Identifier	chr	pk
Item_Weight	num	
Item_Fat_Content	chr	Item_Fat_Content in (Low Fat, Regular, LF)
Item_Visibility	num	Not null
Item_Type	chr	Not null
Item_MRP	num	Not null
Outlet_Identifier	chr	Foreign Key (Outlet)
Outlet_Establishment_Year	int	Not null

Outlet_Size	chr	
Outlet_Location_Type	chr	Outlet_Location_Type in (Tier 1, Tier 2, Tier 3)
Outlet_Type	chr	Outlet_Type in (Supercart, Grocery Store)
Item_Outlet_Sales	num	Not null

6. IMPLEMENTATION DETAILS & USER MANUALS:

6.1 INTRODUCTION:

Sales forecasting in order to carry out the big markets' sales prediction, we had divided out implementation into five stages which helped us to understand our dataset in various ways and use appropriate machine learning models to predict the data. This has also helped us in carrying out proper analysis of the data and exploring various techniques to predict the sales in big markets. The stages followed are:
Hypothesis Generation: This is a very pivotal step in the process of analyzing data. This involves understanding the problem and making some hypothesis about what could potentially have a good impact on the outcome. This is done BEFORE looking at the data, and we end up creating a laundry list of the different analysis which we can potentially perform if data is available. Making a proper hypothesis from the problem allows us to perform our analysis in a much efficient way as we now know what we should be looking for in the data provided.
Data Exploration: In the step of Data Exploration, we have explored the dataset available with us. We have performed some basic data exploration steps including finding the number of columns, dimensions of each column, and figuring out some irregularities in the dataset. This step helped us to understand the data and also helped us to figure out what further analysis can be done on the dataset. In this particular step, we have also visualized the data using various data visualization techniques which has helped us to infer a lot of information from the dataset including the growth of sales, regularity of the dataset and so on.
Data Cleaning: This step typically involves imputing missing values and treating outliers. Though outlier removal is very important in regression techniques, advanced tree-based algorithms are impervious to outliers. Here

we modify some of the data values so that our predictive model would be able to learn in a better way from the training dataset provided to it. Feature Engineering: This step is mainly to overcome the nuances found in the data exploration phase. This is the final step of making our data ready for analysis. Here some new variables are also created based on the existing ones in order to have a better analysis of the dataset. This phase includes combining the outlet_type, modifying the item_visibility, creating a broad category of type of item, determining the years of operation of the store, modifying categories of item_fat_content, numerical and one hot encoding of categorical variables, and exporting the final dataset to be analysed. Model Building: Now the

dataset is ready for it to be analysed by the predictive models. In this particular step, we have implemented three predictive models in order to determine the Item_Outlet_Sales. The three models are linear regression model, ridge regression model and decision tree regression model.

6.2 IMPLEMENTATION DETAILS:

The first step will be declaring variables that will do the calculations of data. The variables should be declared for Item visibility, Item type, Outlet size, Outlet location type, Outlet type, and Item outlet sales. The data is categorized, and the first step will be to the correction of irregularities through data pre-processing. The variation of data is a real tough task as there are around 1562 unique items in a single store.
The second step is to combine the outlet type through various parameters such as item visibility, years of operation, etc. Then create a broad category for item type using many item identifiers. Then the algorithm of ML will study the variations. A generic function that makes the model and performs cross validation should be made.
The next step will be the model making of the application, which will comprise the linear regression model, ridge regression model, decision tree model to decide the results, etc. The data fed to the application will go through sorting and arrangements which will be efficiently performed by Machine Learning.

Missing Value Treatment:

There are different methods to treat missing values based on the problem and the data. Some of the common techniques are as follows:
1. Deletion of rows: In the train dataset, observations having missing values in any variable are deleted.
2. The downside of this method is the loss of information and drop in prediction power of the model.
3. Mean/Median/Mode Imputation: In case of continuous variable, missing values can be replaced with mean or median of all known values of that variable. For categorical variables, we can use mode of the given values to replace the missing values.

Treatment1: We have missing values in Item_Weight and Item_Outlet_Sales. Missing data in Item_Outlet_Sales can be ignored since they belong to the test dataset. We'll now impute Item_Weight with mean weight based on the Item_Identifier variable.

```
sm = sum(is.na(combi$Item_Weight))
nonNullval = nrow(combi) - sm
missing_index = which(is.na(combi$Item_Weight))
for(i in missing_index){
  item = combi$Item_Identifier[i]
  combi$Item_Weight[i] = sum(combi$Item_Weight, na.rm = T)/nonNullval
}
```

Treatment2: Replacing 0's in Item_Visibility variable Similarly, zeroes in Item_Visibility variable can be replaced with Item_Identifier wise mean values of Item_Visibility. It can be visualized in the plot below. Before: Since item visibility cannot be zero so we replace it with the mean of the Item_Identifier.

```
ggplot(combi) + geom_histogram(aes(Item_Visibility), bins=100)
```

R CODE:

```
install.packages("caret")
install.packages("corrplot")
install.packages("xgboost")
install.packages("cowplot")
install.packages("magritrr")
install.packages("ggplot2")

library(data.table) # used for reading and manipulation of data
library(dplyr) # used for data manipulation and joining
library(ggplot2) # used for plotting
library(caret) # used for modeling
library(corrplot) # used for making correlation plot
library(xgboost) # used for building XGBoost model
library(cowplot) # used for combining multiple plots

train<-read.csv("C:/Users/Varun's PC/Desktop/DV_Dataset/train_v9rqX0R.csv")
test<-read.csv("C:/Users/Varun's PC/Desktop/DV_Dataset/test_AbJTz2l.csv")

dim(train)
dim(test)

names(train)
names(test)
```

```
str(train)
str(test)

combi <- rbind(train[1, 11], test)
combi <- rbind(test, train[, names(test)])
combi

# Code for combining training and testing dataset this is helpful if we want to make
any change in dataset then we don't have to change in test and training dataset, it
saves a lot of time.

dim(combi)

ggplot(train)+geom_histogram(aes(Item_Outlet_Sales),binwidth=100,fill="darkgr
een")+xlab("Item_Outlet_Sales")

p1 = ggplot(combi) + geom_histogram(aes(Item_Weight), binwidth = 0.5, fill =
"blue")
p2 = ggplot(combi) + geom_histogram(aes(Item_Visibility), binwidth = 0.005, fill
= "blue")
 p3 = ggplot(combi) + geom_histogram(aes(Item_MRP), binwidth = 1, fill = "blue")
16 26/10/2020 plot_grid(p1, p2, p3, nrow = 1)

ggplot(combi %>% group_by(Item_Fat_Content) %>% summarise(Count=n())) +
geom_bar(aes(Item_Fat_Content,Count), stat="identity", fill="coral")

combi$Item_Fat_Content[combi$Item_Fat_Content == "LF"] = "Low Fat"
combi$Item_Fat_Content[combi$Item_Fat_Content == "low fat"] = "Low Fat"
combi$Item_Fat_Content[combi$Item_Fat_Content == "reg"] = "Regular"
ggplot(combi %>% group_by(Item_Fat_Content) %>% summarise(Count=n())) +
geom_bar(aes(Item_Fat_Content, Count), stat="identity", fill="coral")

p4=ggplot(combi %>% group_by(Item_Type) %>% summarise(Count=n()))
+geom_bar(aes(Item_Type, Count), stat="identity",
fill="coral")+xlab("")+geom_label(aes(Item_Type,Count,label=Count),vjust=0.5)
+theme(axis.text.x=element_text(angle=45,hjust=1))+ggtitle("Item_Type") p4

p5=ggplot(combi %>% group_by(Outlet_Identifier) %>%
summarise(Count=n())) + geom_bar(aes(Outlet_Identifier, Count), stat="identity",
fill="coral")+geom_label(aes(Outlet_Identifier,Count,label=Count),vjust=0.5)+the
me(axis.text.x=element_text(angle=45,hjust=1))
p5

p6=ggplot(combi %>% group_by(Outlet_Establishment_Year) %>%
summarise(Count=n()))+ geom_bar(aes(Outlet_Establishment_Year,Count),
stat="identity",
```

```
fill="coral")+geom_label(aes(Outlet_Establishment_Year,Count,label=Count),vjust
=0.5)+theme(axis.text.x=element_text(angle=45,hjust=1))
p6

p7=ggplot(combi %>% group_by(Outlet_Type) %>% summarise(Count=n())) +
geom_bar(aes(Outlet_Type,Count), stat="identity",
fill="coral")+geom_label(aes(Outlet_Type,Count,label=Count),vjust=0.5)+theme(
axis.text.x=element_text(angle=45,hjust=1))
p7

p8 = ggplot(train) + geom_point(aes(Item_Weight, Item_Outlet_Sales), colour =
"violet", alpha = 0.3) +theme(axis.title = element_text(size = 8.5)) p8

sm = sum(is.na(combi$Item_Weight))
nonNullVal = nrow(combi) - sm
missing_index = which(is.na(combi$Item_Weight))
for(i in missing_index){
 item = combi$Item_Identifier[i]
 combi$Item_Weight[i] = sum(combi$Item_Weight, na.rm = T)/nonNullVal}
 ggplot(combi) + geom_histogram(aes(Item_Visibility), bins=100)

zero_index = which(combi$Item_Visibility == 0)
for(i in zero_index){ item = combi$Item_Identifier[i]
combi$Item_Visibility[i] = mean(combi$Item_Visibility[combi$Item_Identifier ==
item], na.rm = T) }

corMatrix<-cor(combi[1:nrow(train),][sapply(combi[1:nrow(train),],is.numeric)])
corMatrix

corrplot::corrplot(corMatrix, method="number", type="upper")
corrplot::corrplot(corMatrix, method="number", type="upper", order="hclust")
corrplot::corrplot(corMatrix, method="number", type="upper")
corrplot::corrplot(corMatrix, method="number", type="upper", order="hclust")


ggplot(train[1:nrow(train),],aes(Item_Visibility,
Item_Outlet_Sales))+geom_point(size=2.5,
aes(colour=factor(Outlet_Identifier)))+theme(axis.text.x=element_text(angle =
70,vjust = 0.5,color = "black")) + xlab("Item Visibility")+ylab("Item Outlet
Sales")+ggtitle("Item Sales Vs Item Visibility")


ggplot(train[1:nrow(train),],aes(x=Item_Type,y=Item_Outlet_Sales,fill=Outlet_Typ
e))+geom_boxplot()+theme(axis.title.x = element_text(angle=70,vjust=0.5,color
= "black"))+xlab("Item Type")+ylab("Sales")+ggtitle("Sales vs Item type")
```

```
ggplot(train,aes(Item_Type,Item_Outlet_Sales/Item_MRP,fill=Outlet_Type))+geom_
bar(stat = "identity",position = position_dodge())+theme(axis.title.x =
element_text(angle=70,vjust = 0.5,color = "black"))+xlab("Item
Type")+ylab("Item Outlet Sales")+ggtitle("Item Sales vs Item type")
```

Python Code (Regression Models and Predictions):

```
# -*- coding: utf-8 -*-
"""DV.ipynb

Automatically generated by Collaboratory.

Original file is located at
https://colab.research.google.com/drive/13jvl6dGVIGDRRkeC6j1I99lv-M9nCD7E
"""

import pandas as pd
import numpy as np

train=pd.read_csv("train_v9rqX0R.csv", na_values={"Item_Visibility":[0]})
test=pd.read_csv("test_AbJTz2l.csv", na_values={"Item_Visibility":[0]})

train['source’] ='train'
test['source’] ='test'
data=pd.concat([train,test],ignore_index=True)

#the one thing we have to focus is item_outlet_Sales
discpt=data.describe()

#Lets find out how many zero'es values are
nan_descript=data.apply(lambda x: sum(x.isnull()))

#Now lets find out the unique values in each of the catogorical columns
uniq=data.apply(lambda x: len(x.unique()))

#let do grouping in each catogorical columns
col=["Item_Fat_Content","Item_Type","Outlet_Location_Type","Outlet_Size"]
for i in col:
 print("The frequency distribution of each catogorical columns is--" + i+"\n")
print(data[i].value_counts())

#Replacing the minimum nan values in the Item_Weight with its mean value
data.fillna({"Item_Weight":data["Item_Weight"].mean()},inplace=True)

#checking the current status of nan values in the dataframe
```

```
nan_descript=data.apply(lambda x: sum(x.isnull()))

#Now we have 0 nan values in Item_Weight
data["Outlet_Size"].fillna(method="ffill",inplace=True)
nan_descript=data.apply(lambda x: sum(x.isnull()))

#Now working on the item_visibility
visibilty_avg=data.pivot_table(values="Item_Visibility", index="Item_Identifier")
itm_visi=data.groupby('Item_Type')
data_frames=[]
for item,item_df in itm_visi:
 data_frames.append(itm_visi.get_group(item))
for i in data_frames:
 i["Item_Visibility"].fillna(value=i["Item_Visibility"].mean(),inplace=True)
i["Item_Outlet_Sales"].fillna(value=i["Item_Outlet_Sales"].mean(),inplace=True)
new_data=pd.concat(data_frames)
nan_descript=new_data.apply(lambda x: sum(x.isnull()))
new_data

#Now we have successfully cleaned our complete dataset.
new_data["Item_Fat_Content"].replace({'LF':'Low Fat','reg':'Regular','low fat':'Low
Fat'},inplace=True)
new_data["Item_Fat_Content"].value_counts()

#Implementing one-hot-Coding method for getting the categorical variables
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data=new_data
data['Outlet'] = le.fit_transform(data['Outlet_Identifier'])
var_mod =
['Item_Fat_Content','Outlet_Location_Type','Outlet_Size','Item_Type','Outlet_Type']
le = LabelEncoder()
for i in var_mod:
 data[i] = le.fit_transform(data[i])

data = pd.get_dummies(data,
columns=['Item_Fat_Content','Outlet_Location_Type','Outlet_Size','Outlet_Type',
'Item_Type'])

#Exporting the datas
train = data.loc[data['source’] =="train"]
test = data.loc[data['source’] =="test"]

#Drop unnecessary columns:
test.drop(['Item_Outlet_Sales','source'],axis=1,inplace=True)

#Here we are dropping the "Item_Outlet_Sales because this only we want to be
```

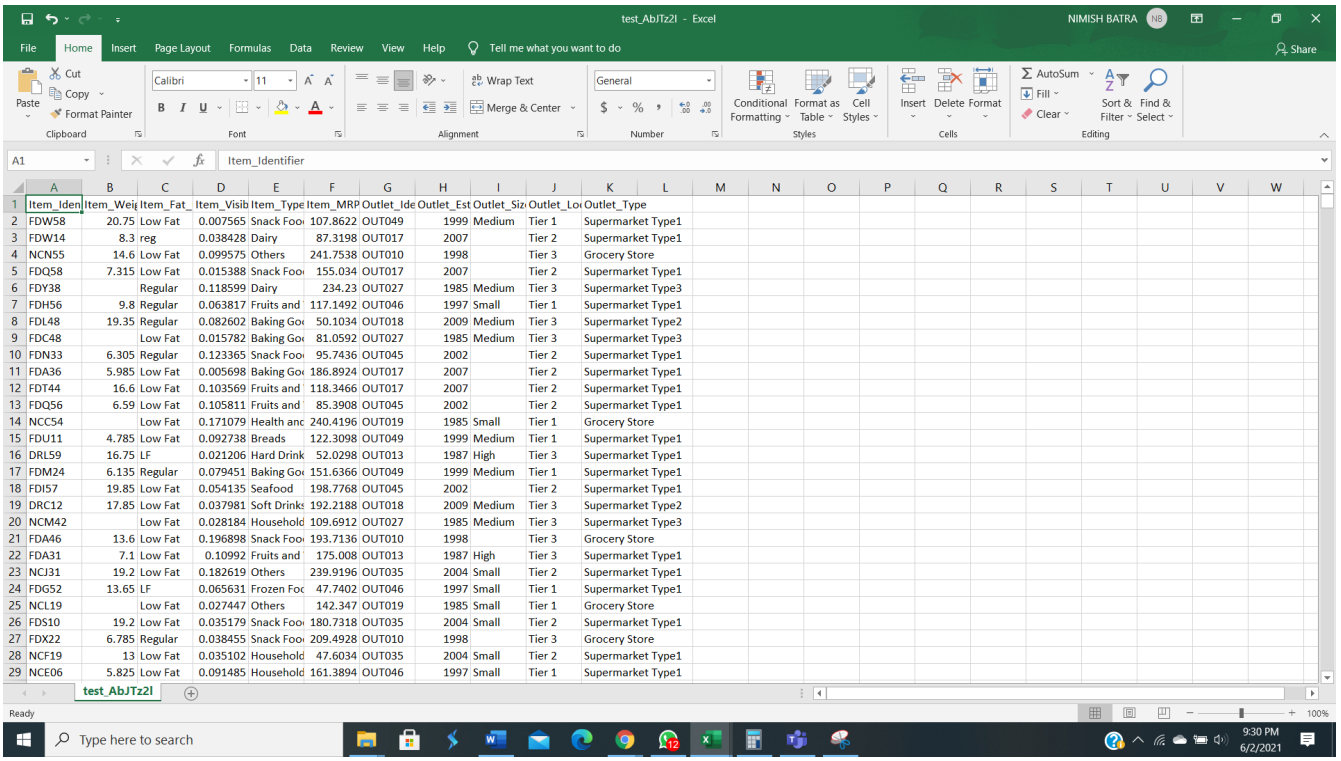
```
predicted from the model that we are going to built
train.drop(['source'],axis=1,inplace=True)

#Export files as modified versions:
train.to_csv("train_modified.csv",index=False)
test.to_csv("test_modified.csv",index=False)

#Let's start building the baseline model as it is non -predicting model and also
commonly known as informed guess
#Mean based:
mean_sales = train['Item_Outlet_Sales'].mean()

#Define a dataframe with IDs for submission:
base1 = test[['Item_Identifier','Outlet_Identifier']]
base1['Item_Outlet_Sales'] = mean_sales

#Export submission file
base1.to_csv("alg0.csv", index=False)

#Define target and ID columns:
target = 'Item_Outlet_Sales'
IDcol = ['Item_Identifier','Outlet_Identifier']

import numpy as np
import sklearn
from sklearn.model_selection import cross_validate
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
def modelfit(alg, dtrain, dtest, predictors, target, IDcol, filename):
#Fit the algorithm on the data
 alg.fit(dtrain[predictors], dtrain[target])

 #Predict training set:
 dtrain_predictions = alg.predict(dtrain[predictors])

 #Perform cross-validation:
     cv_score = sklearn.model_selection.cross_val_score(alg, dtrain[predictors],
     dtrain[target], cv=20, scoring='neg_mean_squared_error')
 cv_score = np.sqrt(np.abs(cv_score))

 #Print model report:
 print ("\nModel Report")
 print ("RMSE : %.4g" %
np.sqrt(metrics.mean_squared_error(dtrain[target].values, dtrain_predictions)))
print ("CV Score : Mean - %.4g | Std - %.4g | Min - %.4g | Max - %.4g" %
(np.mean(cv_score),np.std(cv_score),np.min(cv_score),np.max(cv_score)))
```

```
#Predict on testing data:
dtest[target] = alg.predict(dtest[predictors])

#Export submission file:
IDcol.append(target)

submission = pd.DataFrame({ x: dtest[x] for x in IDcol})
submission.to_csv(filename, index=False)

#Linear Regression model
print("Creating the models and processing")
from sklearn.linear_model import LinearRegression, Ridge
predictors = [x for x in train.columns if x not in [target]+IDcol]
# print predictors
alg1 = LinearRegression(normalize=True)
modelfit(alg1, train, test, predictors, target, IDcol, 'alg1.csv')
coef1 = pd.Series(alg1.coef_, predictors).sort_values()
coef1.plot(kind='bar', title='Model Coefficients')

#Ridge Regression Model
predictors = [x for x in train.columns if x not in [target]+IDcol]
alg2 = Ridge(alpha=0.05,normalize=True)
modelfit(alg2, train, test, predictors, target, IDcol, 'alg2.csv')
coef2 = pd.Series(alg2.coef_, predictors).sort_values()
coef2.plot(kind='bar', title='Model Coefficients')
print("Model has been successfully created and trained. The predicted result is in
alg2.csv")

#Decision Tree Model
from sklearn.tree import DecisionTreeRegressor
predictors = [x for x in train.columns if x not in [target]+IDcol]
alg3 = DecisionTreeRegressor(max_depth=15, min_samples_leaf=100)
modelfit(alg3, train, test, predictors, target, IDcol, 'alg3.csv')
coef3 = pd.Series(alg3.feature_importances_,
predictors).sort_values(ascending=False)
coef3.plot(kind='bar', title='Feature Importances')
print ("Model has been successfully created and trained. The predicted result is in
alg3.csv")
```

7. EXPERIMENTAL RESULTS AND ANALYSIS:

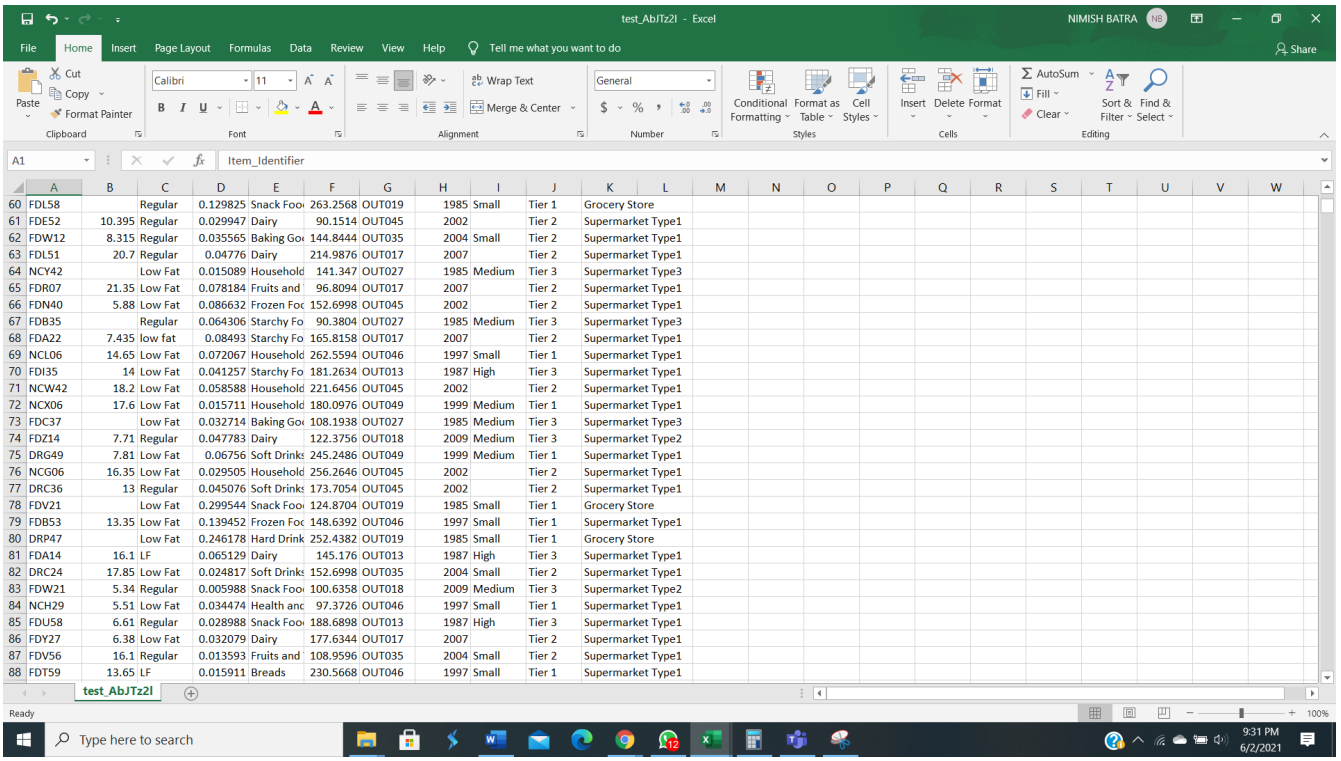
7.1 INTRODUCTION:

- The dataset has 8523 rows (instances) of 12 variables (attributes).
- It is taken from kaggle.com.
- Data attributes: Item_Identifier, Item_Weight, Item_Type, Item_MRP,Item_Visibility, etc
- Algorithms we used to build our models are Linear Regression, Ridge Regression and Decision Tree.

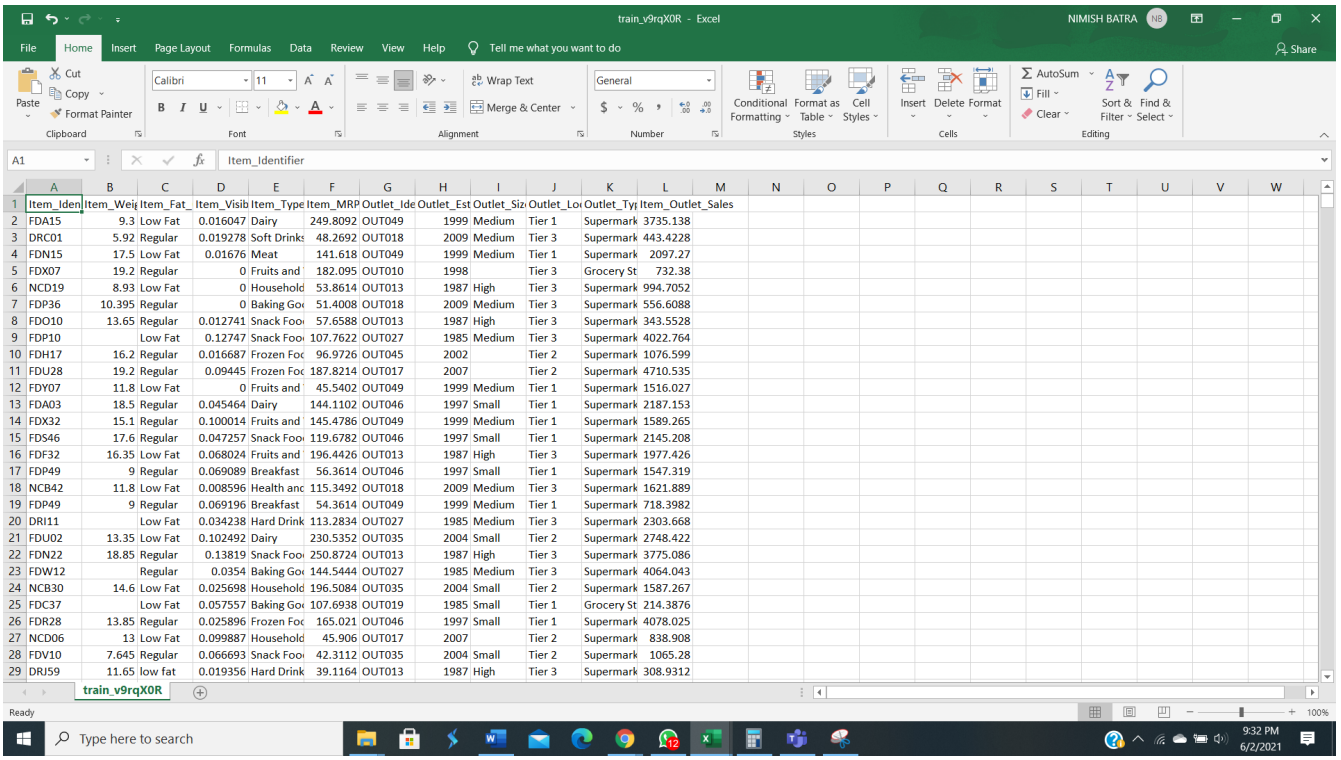


test_AbJTz2I - Excel

Item_Iden	Item_Wei	Item_Fat_	Item_Visib	Item_Type	Item_MRP	Outlet_Ide	Outlet_Est	Outlet_Siz	Outlet_Lo	Outlet_Type
FDW58	20.75	Low Fat	0.007565	Snack Foo	107.8622	OUT049	1999	Medium	Tier 1	Supermarket Type1
FDW14	8.3	reg	0.038428	Dairy	87.3198	OUT017	2007		Tier 2	Supermarket Type1
NCN55	14.6	Low Fat	0.099575	Others	241.7538	OUT010	1998		Tier 3	Grocery Store
FDQ58	7.315	Low Fat	0.015388	Snack Foo	155.034	OUT017	2007		Tier 2	Supermarket Type1
FDY38		Regular	0.118599	Dairy	234.23	OUT027	1985	Medium	Tier 3	Supermarket Type3
FDH56	9.8	Regular	0.063817	Fruits and	117.1492	OUT046	1997	Small	Tier 1	Supermarket Type1
FDL48	19.35	Regular	0.082602	Baking Go	50.1034	OUT018	2009	Medium	Tier 3	Supermarket Type2
FDC48		Low Fat	0.015782	Baking Go	81.0592	OUT027	1985	Medium	Tier 3	Supermarket Type3
FDN33	6.305	Regular	0.123365	Snack Foo	95.7436	OUT045	2002		Tier 2	Supermarket Type1
FDA36	5.985	Low Fat	0.005698	Baking Go	186.8924	OUT017	2007		Tier 2	Supermarket Type1
FDT44	16.6	Low Fat	0.103569	Fruits and	118.3466	OUT017	2007		Tier 2	Supermarket Type1
FDQ56	6.59	Low Fat	0.105811	Fruits and	85.3908	OUT045	2002		Tier 2	Supermarket Type1
NCC54		Low Fat	0.171079	Health and	240.4196	OUT019	1985	Small	Tier 1	Grocery Store
FDU11	4.785	Low Fat	0.092738	Breads	122.3098	OUT049	1999	Medium	Tier 1	Supermarket Type1
DRL59	16.75	LF	0.021206	Hard Drink	52.0298	OUT013	1987	High	Tier 3	Supermarket Type1
FDM24	6.135	Regular	0.079451	Baking Go	151.6366	OUT049	1999	Medium	Tier 1	Supermarket Type1
FDI57	19.85	Low Fat	0.054135	Seafood	198.7768	OUT045	2002		Tier 2	Supermarket Type1
DRC12	17.85	Low Fat	0.037981	Soft Drinks	192.2188	OUT018	2009	Medium	Tier 3	Supermarket Type2
NCM42		Low Fat	0.028184	Household	109.6912	OUT027	1985	Medium	Tier 3	Supermarket Type3
FDA46	13.6	Low Fat	0.196898	Snack Foo	193.7136	OUT010	1998		Tier 3	Grocery Store
FDA31	7.1	Low Fat	0.10992	Fruits and	175.008	OUT013	1987	High	Tier 3	Supermarket Type1
NCJ31	19.2	Low Fat	0.182619	Others	239.9196	OUT035	2004	Small	Tier 2	Supermarket Type1
FDG52	13.65	LF	0.065631	Frozen Foo	47.7402	OUT046	1997	Small	Tier 1	Supermarket Type1
NCL19		Low Fat	0.027447	Others	142.347	OUT019	1985	Small	Tier 1	Grocery Store
FDS10	19.2	Low Fat	0.035179	Snack Foo	180.7318	OUT035	2004	Small	Tier 2	Supermarket Type1
FDX22	6.785	Regular	0.038455	Snack Foo	209.4928	OUT010	1998		Tier 3	Grocery Store
NCF19	13	Low Fat	0.035102	Household	47.6034	OUT035	2004	Small	Tier 2	Supermarket Type1
NCE06	5.825	Low Fat	0.091485	Household	161.3894	OUT046	1997	Small	Tier 1	Supermarket Type1



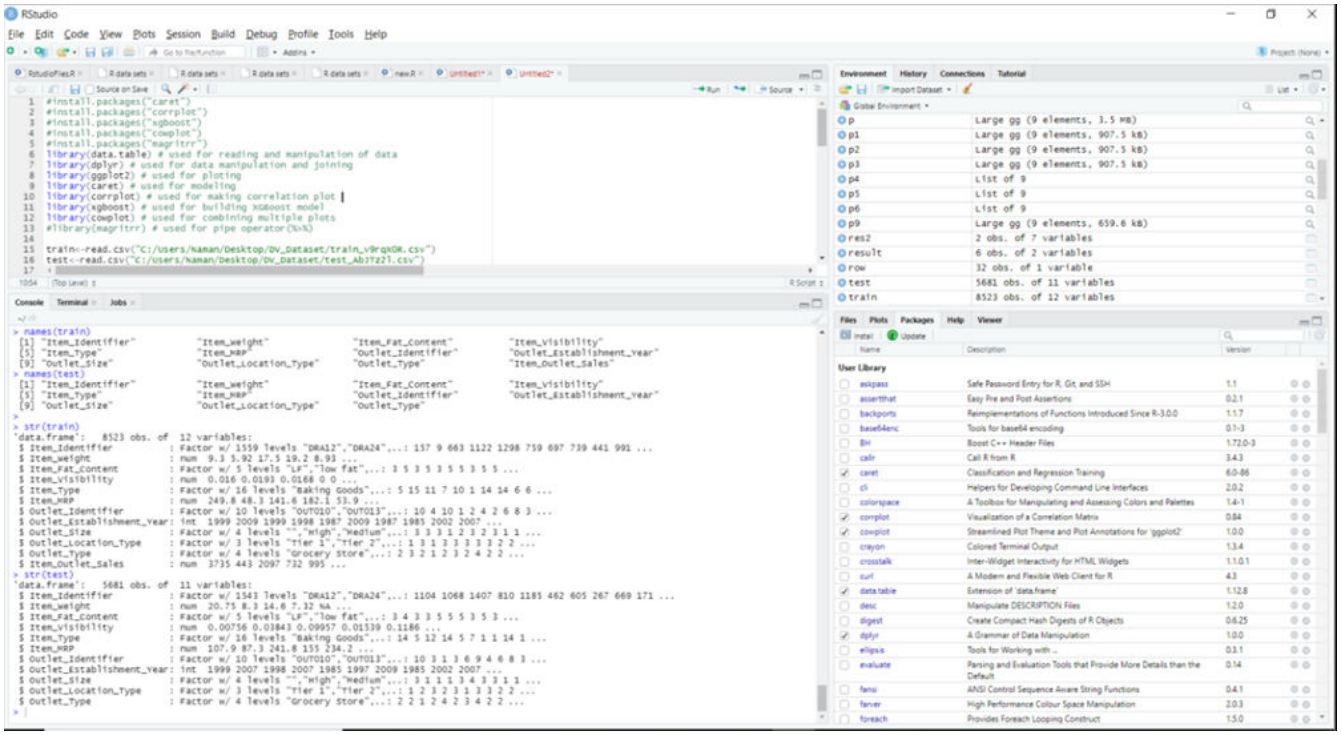
	A	B	C	D	E	F	G	H	I	J	K	L
60	FDL58		Regular	0.129825	Snack Foo	263.2568	OUT019	1985	Small	Tier 1	Grocery Store	
61	FDE52	10.395	Regular	0.029947	Dairy	90.1514	OUT045	2002		Tier 2	Supermarket Type1	
62	FDW12	8.315	Regular	0.035565	Baking Goo	144.8444	OUT035	2004	Small	Tier 2	Supermarket Type1	
63	FDL51	20.7	Regular	0.04776	Dairy	214.9876	OUT017	2007		Tier 2	Supermarket Type1	
64	NCY42		Low Fat	0.015089	Household	141.347	OUT027	1985	Medium	Tier 3	Supermarket Type3	
65	FDR07	21.35	Low Fat	0.078184	Fruits and	96.8094	OUT017	2007		Tier 2	Supermarket Type1	
66	FDN40	5.88	Low Fat	0.086632	Frozen Foo	152.6998	OUT045	2002		Tier 2	Supermarket Type1	
67	FDB35		Regular	0.064306	Starchy Fo	90.3804	OUT027	1985	Medium	Tier 3	Supermarket Type3	
68	FDA22	7.435	low fat	0.08493	Starchy Fo	165.8158	OUT017	2007		Tier 2	Supermarket Type1	
69	NCL06	14.65	Low Fat	0.072067	Household	262.5594	OUT046	1997	Small	Tier 1	Supermarket Type1	
70	FDI35	14	Low Fat	0.041257	Starchy Fo	181.2634	OUT013	1987	High	Tier 3	Supermarket Type1	
71	NCW42	18.2	Low Fat	0.058588	Household	221.6456	OUT045	2002		Tier 2	Supermarket Type1	
72	NCX06	17.6	Low Fat	0.015711	Household	180.0976	OUT049	1999	Medium	Tier 1	Supermarket Type1	
73	FDC37		Low Fat	0.032714	Baking Goo	108.1938	OUT027	1985	Medium	Tier 3	Supermarket Type3	
74	FDZ14	7.71	Regular	0.047783	Dairy	122.3756	OUT018	2009	Medium	Tier 3	Supermarket Type2	
75	DRG49	7.81	Low Fat	0.06756	Soft Drinks	245.2486	OUT049	1999	Medium	Tier 1	Supermarket Type1	
76	NCG06	16.35	Low Fat	0.029505	Household	256.2646	OUT045	2002		Tier 2	Supermarket Type1	
77	DRC36	13	Regular	0.045076	Soft Drinks	173.7054	OUT045	2002		Tier 2	Supermarket Type1	
78	FDV21		Low Fat	0.299544	Snack Foo	124.8704	OUT019	1985	Small	Tier 1	Grocery Store	
79	FDB53	13.35	Low Fat	0.139452	Frozen Foo	148.6392	OUT046	1997	Small	Tier 1	Supermarket Type1	
80	DRP47		Low Fat	0.246178	Hard Drink	252.4382	OUT019	1985	Small	Tier 1	Grocery Store	
81	FDA14	16.1	LF	0.065129	Dairy	145.176	OUT013	1987	High	Tier 3	Supermarket Type1	
82	DRC24	17.85	Low Fat	0.024817	Soft Drinks	152.6998	OUT035	2004	Small	Tier 2	Supermarket Type1	
83	FDW21	5.34	Regular	0.005988	Snack Foo	100.6358	OUT018	2009	Medium	Tier 3	Supermarket Type2	
84	NCH29	5.51	Low Fat	0.034474	Health and	97.3726	OUT046	1997	Small	Tier 1	Supermarket Type1	
85	FDU58	6.61	Regular	0.028988	Snack Foo	188.6898	OUT013	1987	High	Tier 3	Supermarket Type1	
86	FDY27	6.38	Low Fat	0.032079	Dairy	177.6344	OUT017	2007		Tier 2	Supermarket Type1	
87	FDV56	16.1	Regular	0.013593	Fruits and	108.9596	OUT035	2004	Small	Tier 2	Supermarket Type1	
88	FDT59	13.65	LF	0.015911	Breads	230.5668	OUT046	1997	Small	Tier 1	Supermarket Type1	



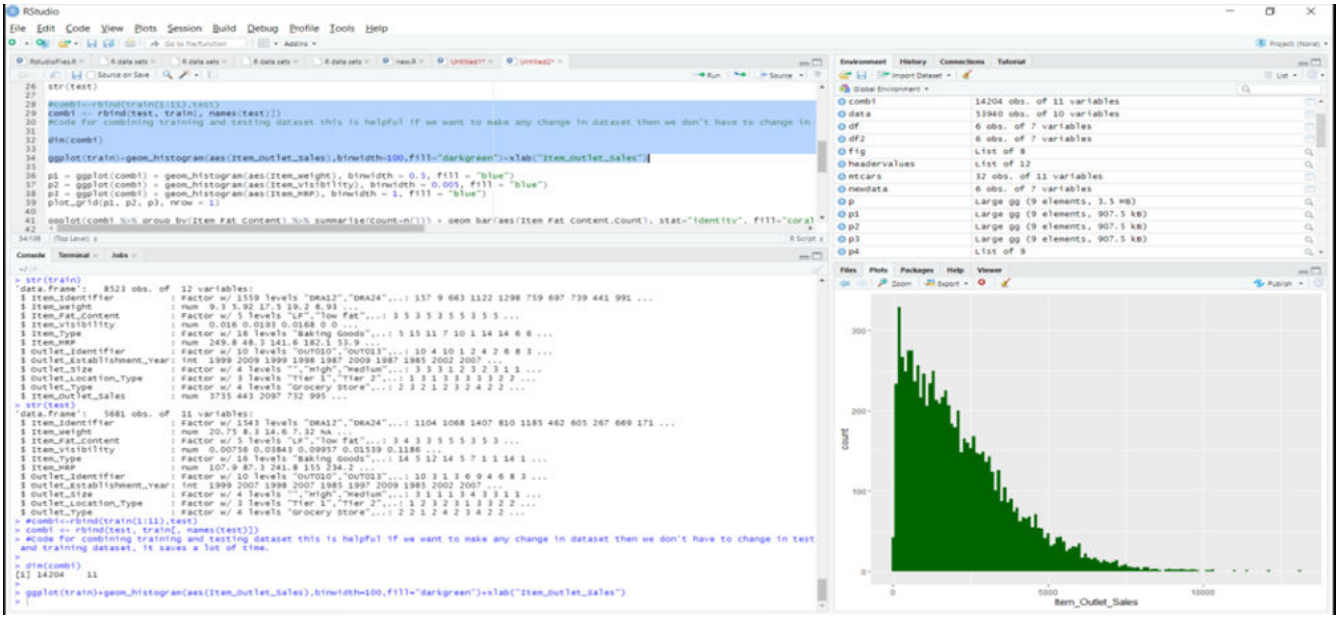
	A	B	C	D	E	F	G	H	I	J	K	L
1	Item_Iden	Item_Wei	Item_Fat_	Item_Visib	Item_Type	Item_MRP	Outlet_Ide	Outlet_Est	Outlet_Siz	Outlet_Lo	Outlet_Ty	Item_Outlet_Sales
2	FDA15	9.3	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermark	3735.138
3	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermark	443.4228
4	FDN15	17.5	Low Fat	0.01676	Meat	141.618	OUT049	1999	Medium	Tier 1	Supermark	2097.27
5	FDX07	19.2	Regular	0	Fruits and	182.095	OUT010	1998		Tier 3	Grocery St	732.38
6	NCD19	8.93	Low Fat	0	Household	53.8614	OUT013	1987	High	Tier 3	Supermark	994.7052
7	FDP36	10.395	Regular	0	Baking Goo	51.4008	OUT018	2009	Medium	Tier 3	Supermark	556.6088
8	FDO10	13.65	Regular	0.012741	Snack Foo	57.6588	OUT013	1987	High	Tier 3	Supermark	343.5528
9	FDP10		Low Fat	0.12747	Snack Foo	107.7622	OUT027	1985	Medium	Tier 3	Supermark	4022.764
10	FDH17	16.2	Regular	0.016687	Frozen Foo	96.9726	OUT045	2002		Tier 2	Supermark	1076.599
11	FDU28	19.2	Regular	0.09445	Frozen Foo	187.8214	OUT017	2007		Tier 2	Supermark	4710.535
12	FDY07	11.8	Low Fat	0	Fruits and	45.5402	OUT049	1999	Medium	Tier 1	Supermark	1516.027
13	FDA03	18.5	Regular	0.045464	Dairy	144.1102	OUT046	1997	Small	Tier 1	Supermark	2187.153
14	FDX32	15.1	Regular	0.100014	Fruits and	145.4786	OUT049	1999	Medium	Tier 1	Supermark	1589.265
15	FDS46	17.6	Regular	0.047257	Snack Foo	119.6782	OUT046	1997	Small	Tier 1	Supermark	2145.208
16	FDF32	16.35	Low Fat	0.068024	Fruits and	196.4426	OUT013	1987	High	Tier 3	Supermark	1977.426
17	FDP49	9	Regular	0.069089	Breakfast	56.3614	OUT046	1997	Small	Tier 1	Supermark	1547.319
18	NCB42	11.8	Low Fat	0.008596	Health and	115.3492	OUT018	2009	Medium	Tier 3	Supermark	1621.889
19	FDP49	9	Regular	0.069196	Breakfast	54.3614	OUT049	1999	Medium	Tier 1	Supermark	718.3982
20	DRI11		Low Fat	0.034238	Hard Drink	113.2834	OUT027	1985	Medium	Tier 3	Supermark	2303.668
21	FDU02	13.35	Low Fat	0.102492	Dairy	230.5352	OUT035	2004	Small	Tier 2	Supermark	2748.422
22	FDN22	18.85	Regular	0.13819	Snack Foo	250.8724	OUT013	1987	High	Tier 3	Supermark	3775.086
23	FDW12		Regular	0.0354	Baking Goo	144.5444	OUT027	1985	Medium	Tier 3	Supermark	4064.043
24	NCB30	14.6	Low Fat	0.025698	Household	196.5084	OUT035	2004	Small	Tier 2	Supermark	1587.267
25	FDC37		Low Fat	0.057557	Baking Goo	107.6938	OUT019	1985	Small	Tier 1	Grocery St	214.3876
26	FDR28	13.85	Regular	0.025896	Frozen Foo	165.021	OUT046	1997	Small	Tier 1	Supermark	4078.025
27	NCD06	13	Low Fat	0.099887	Household	45.906	OUT017	2007		Tier 2	Supermark	838.908
28	FDV10	7.645	Regular	0.066693	Snack Foo	42.3112	OUT035	2004	Small	Tier 2	Supermark	1065.28
29	DRJ59	11.65	low fat	0.019356	Hard Drink	39.1164	OUT013	1987	High	Tier 3	Supermark	308.9312

7.2 RESULTS:

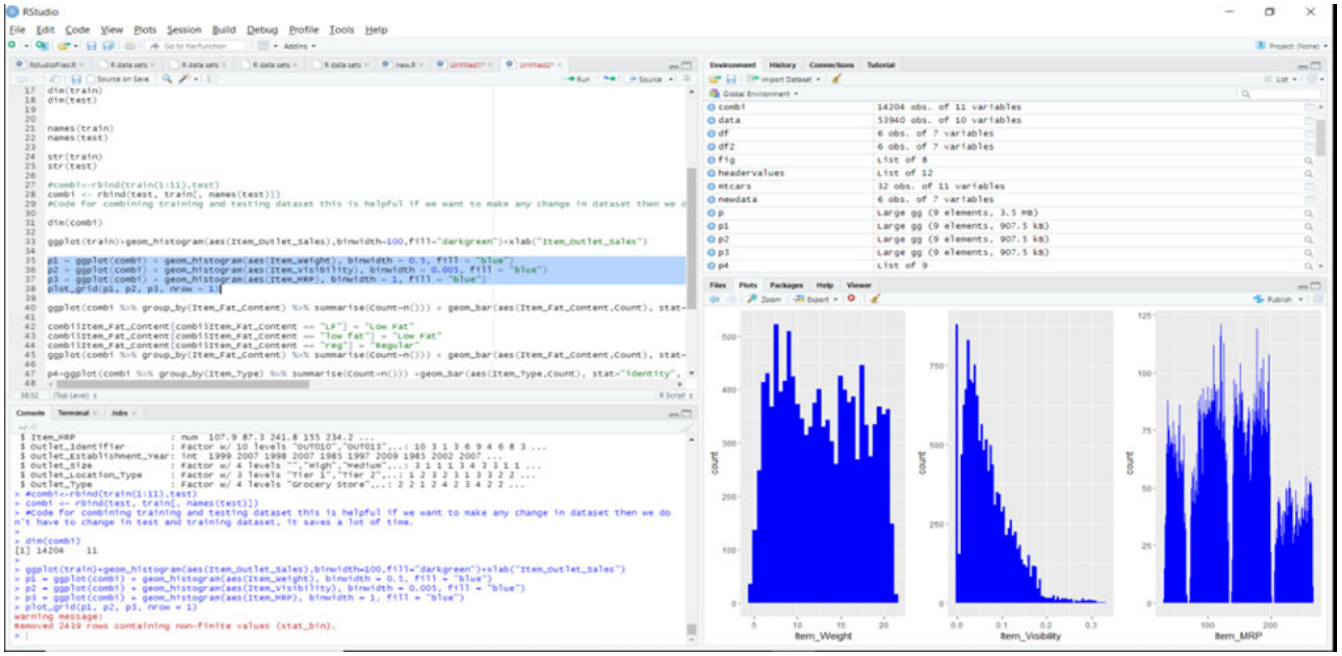
- Only very few outlets which are medium in size have item Visibility low and high outlet sales
- Outlet 27 and 35 belong to the outlet which have low item visibility but have more outlet sales.
- Lesser number of observations in the data for the outlets established in the year 1998 as compared to the other years.
- Supermarket Type 1 seems to be the most popular category of Outlet_Type.
- Item_Visibility is right-skewed and should be transformed to curb its skewness



Histogram(ggplot2)- item_outlet_sales



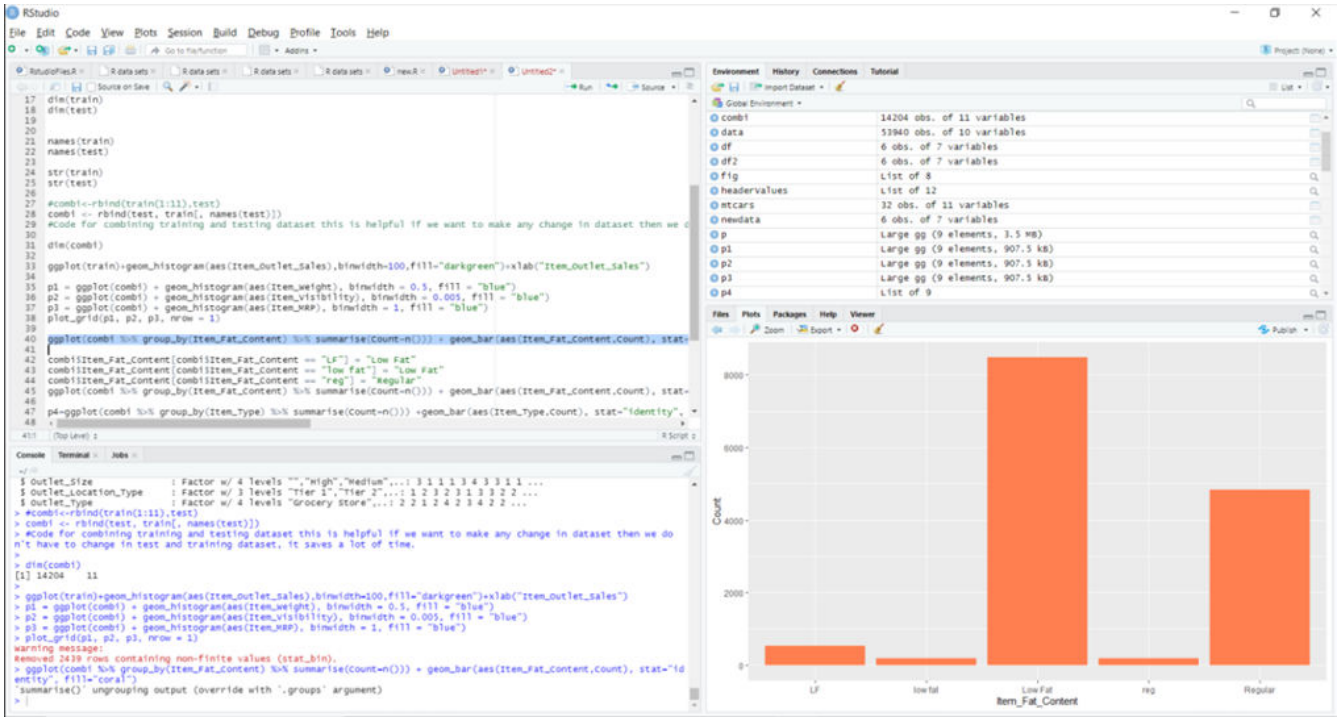
Histogram(cowplot)- item_weight, item_visibility, item_mrp



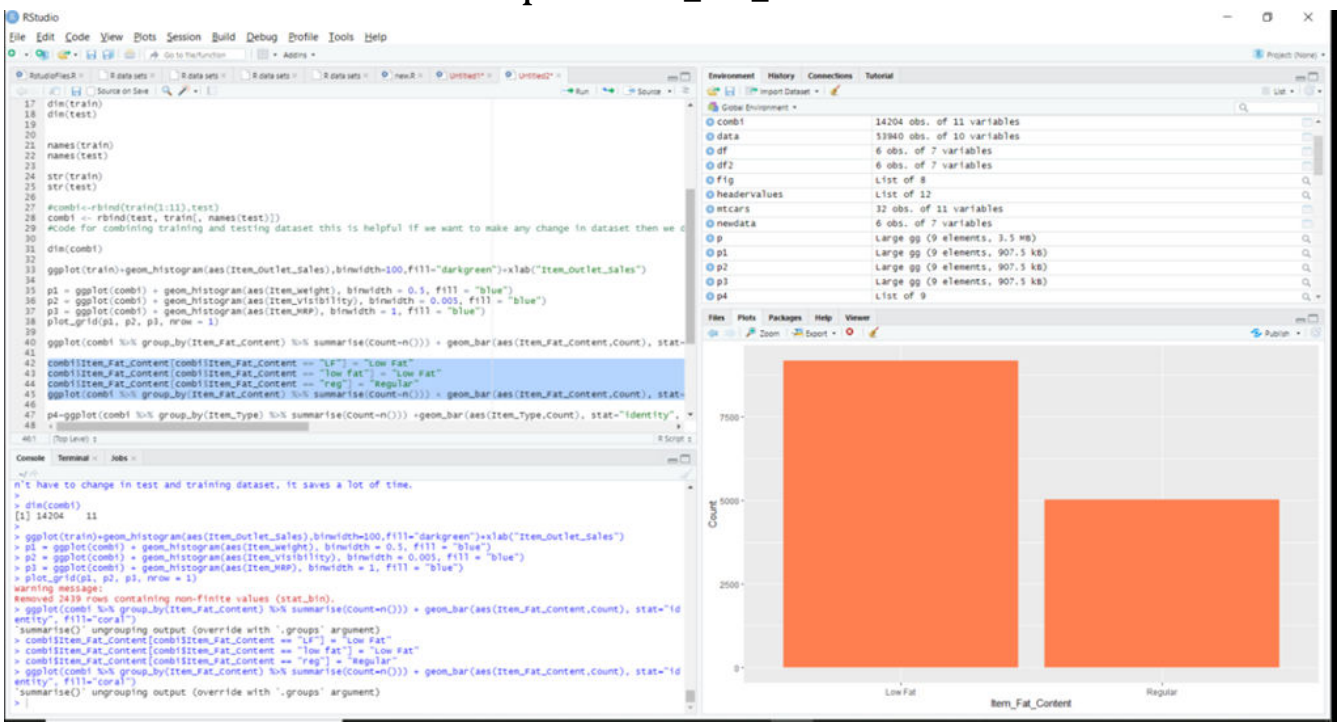
OBSERVATION:

- No clear-cut pattern in: Item_Weight, thus we get to know that it is randomly distributed.
- Item_Visibility is right-skewed-thus must be transformed to curb skewness.
- 4 different distributions for Item_MRP, thus all the items are classified under 4 price ranges.

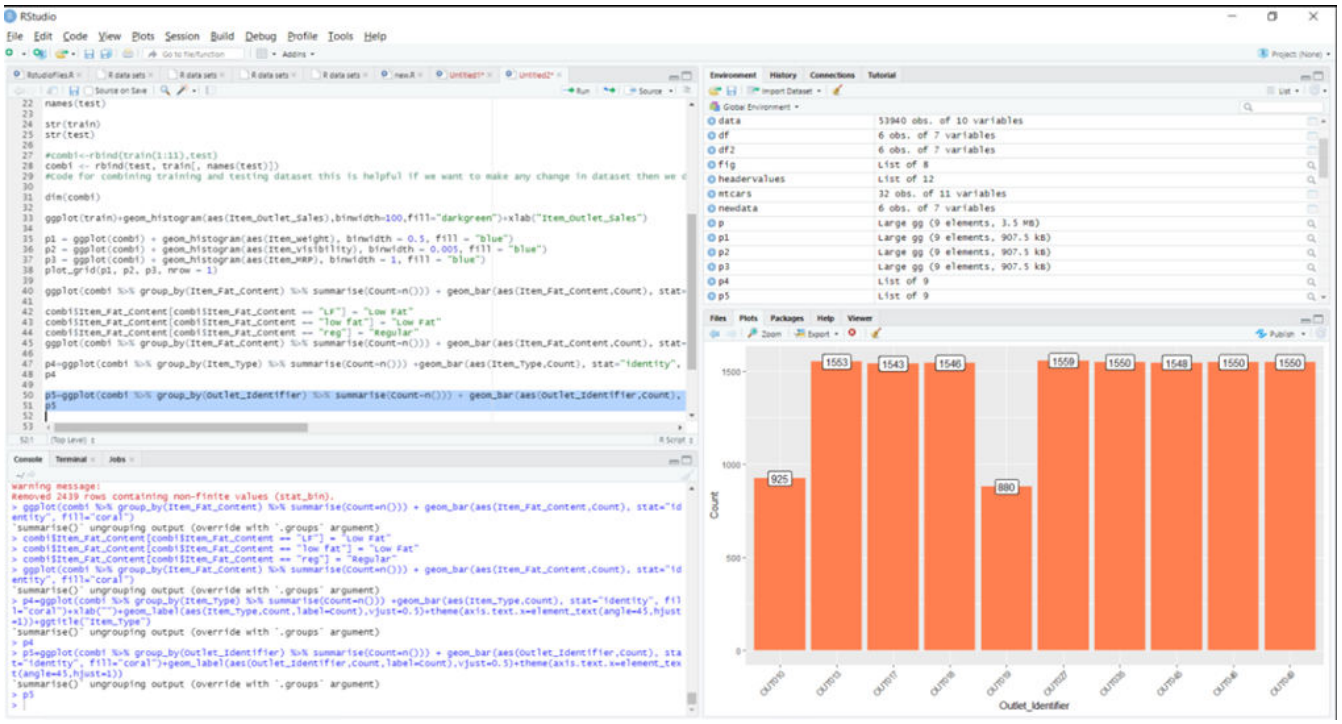
Bar Graph (dplyr)- item_fat_content



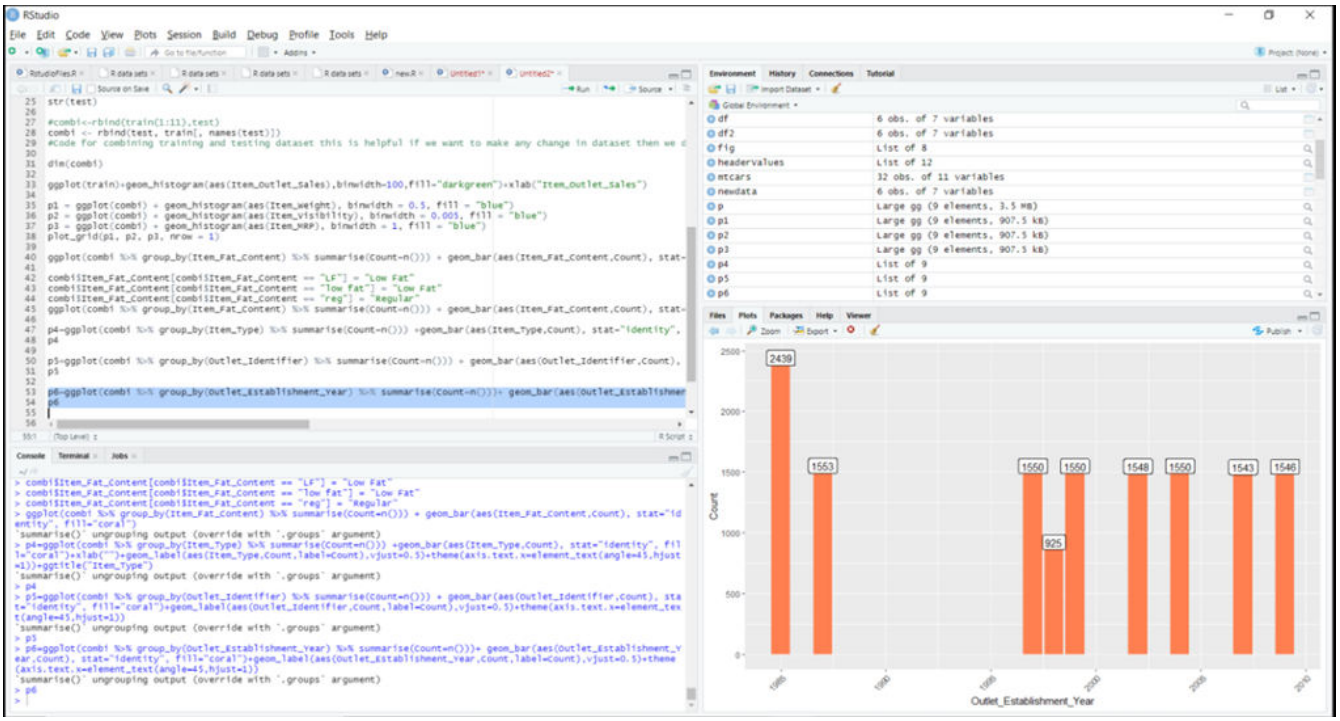
Bar Graph- item_fat_content



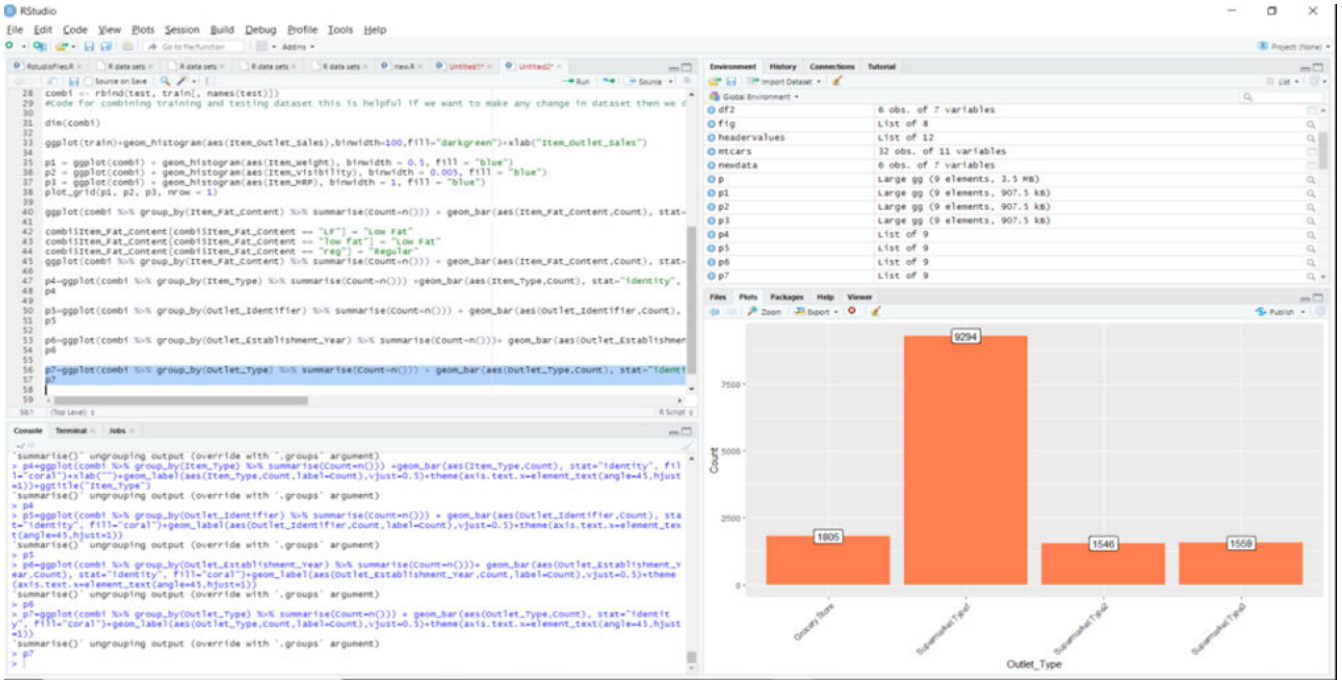
Bar Graph- item_type Bar Graph- outlet_identifier



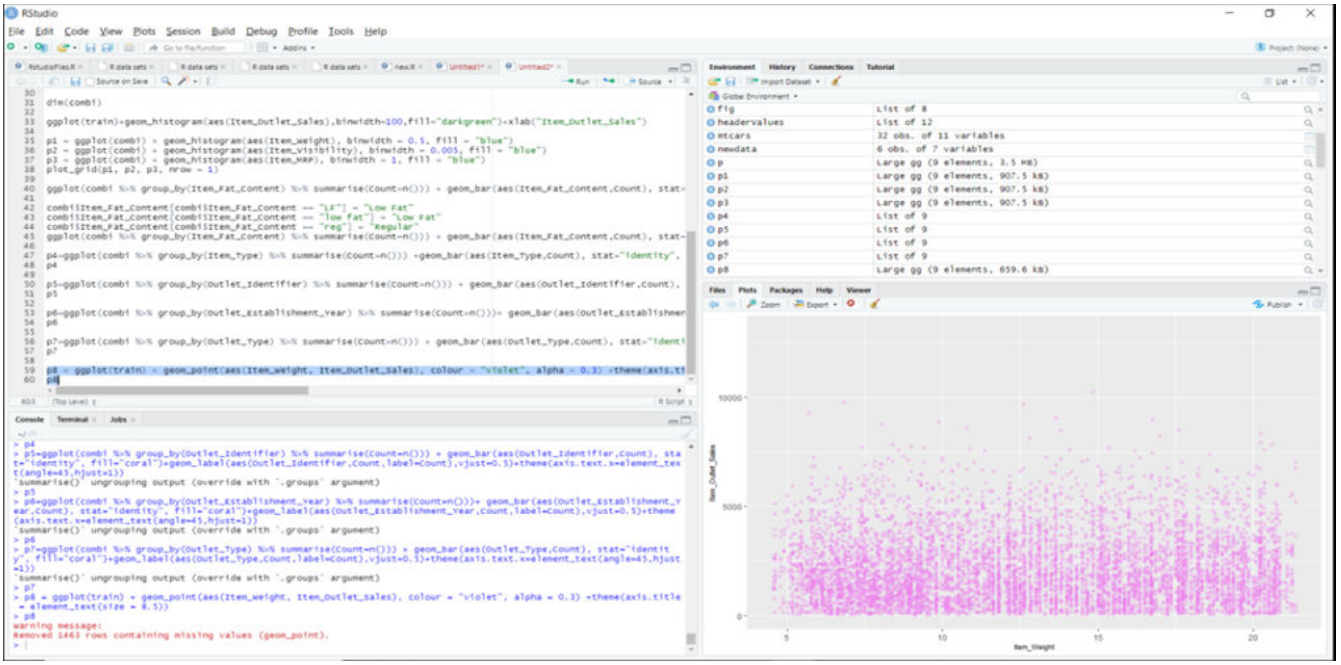
Bar Graph- outlet_establishment_year



Bar Graph- outlet_type

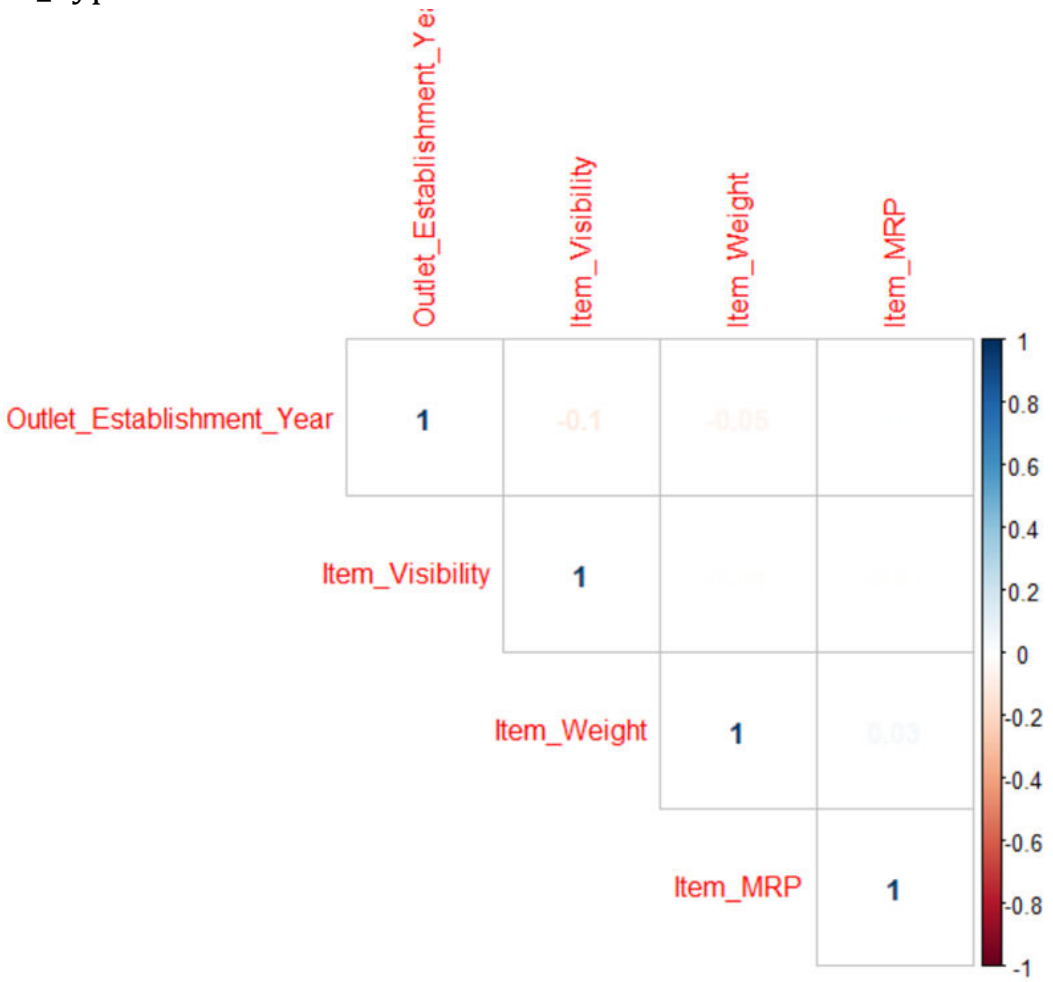


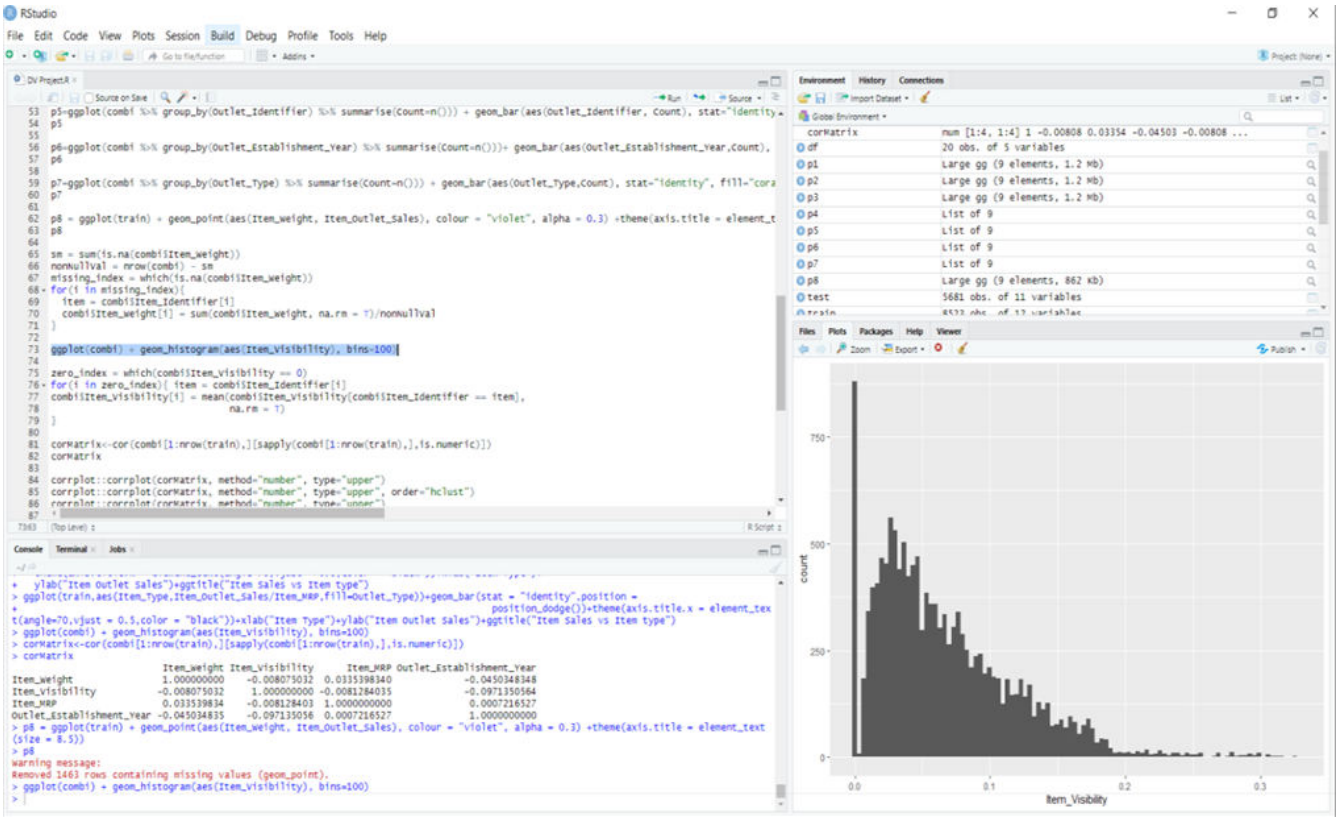
Scatter Plot- item_weight

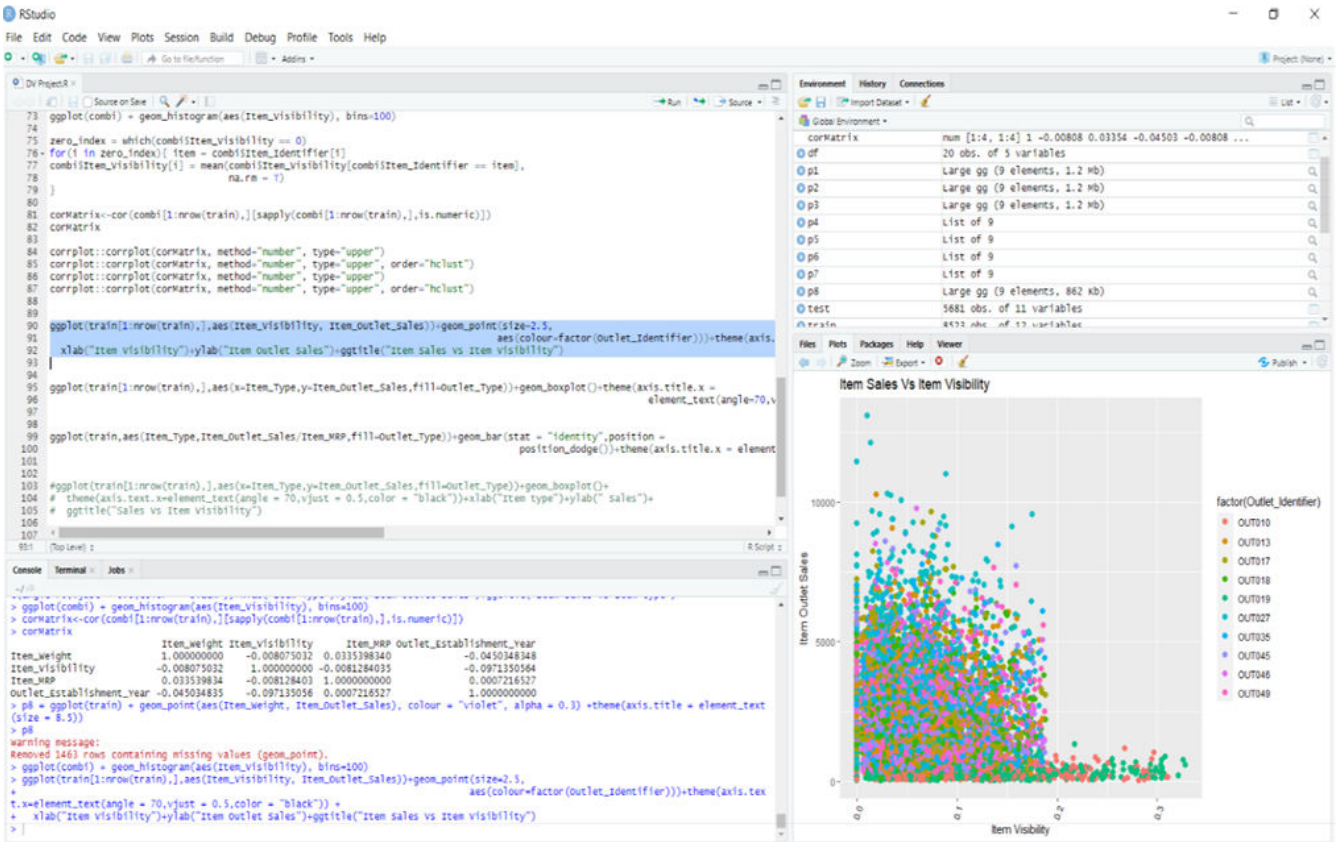


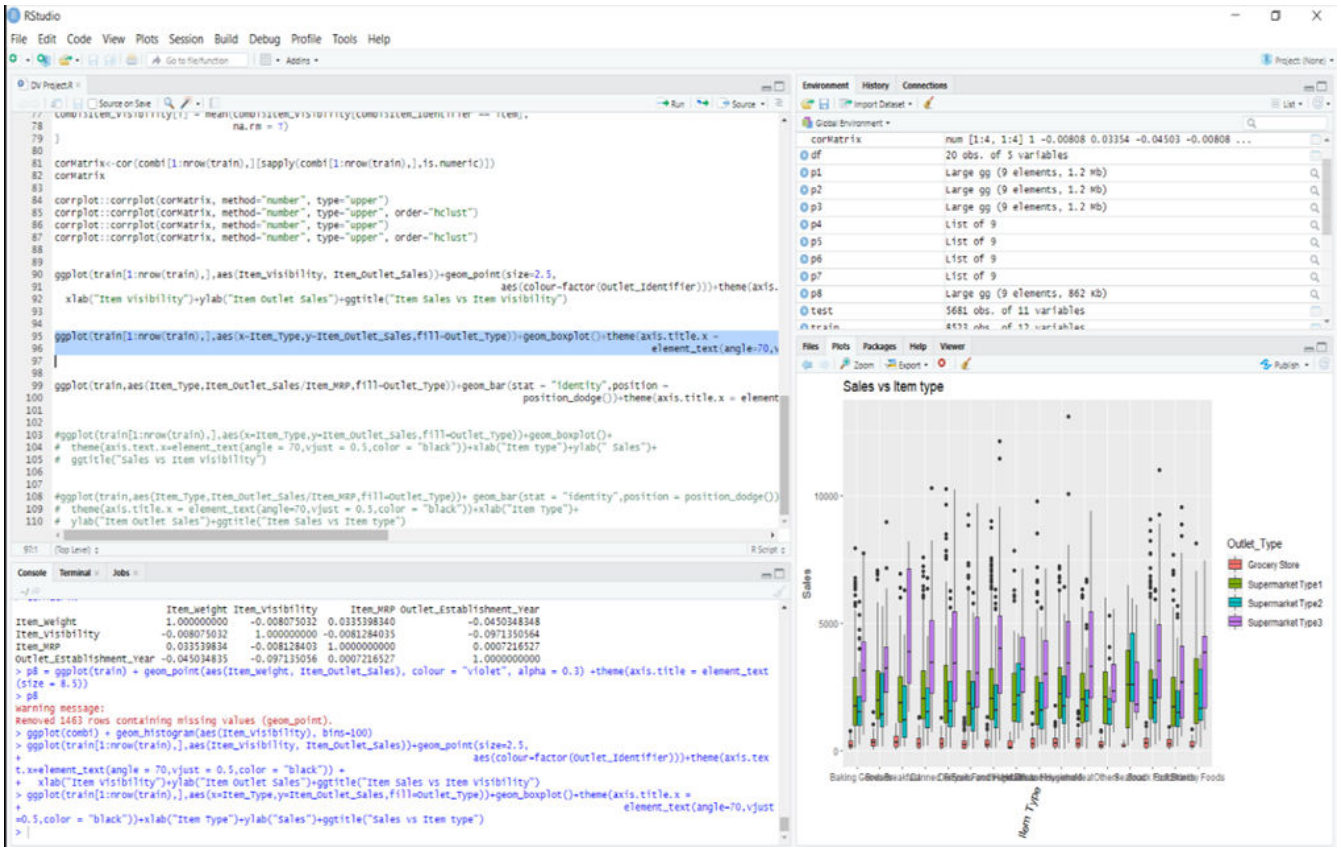
OBSERVATION:

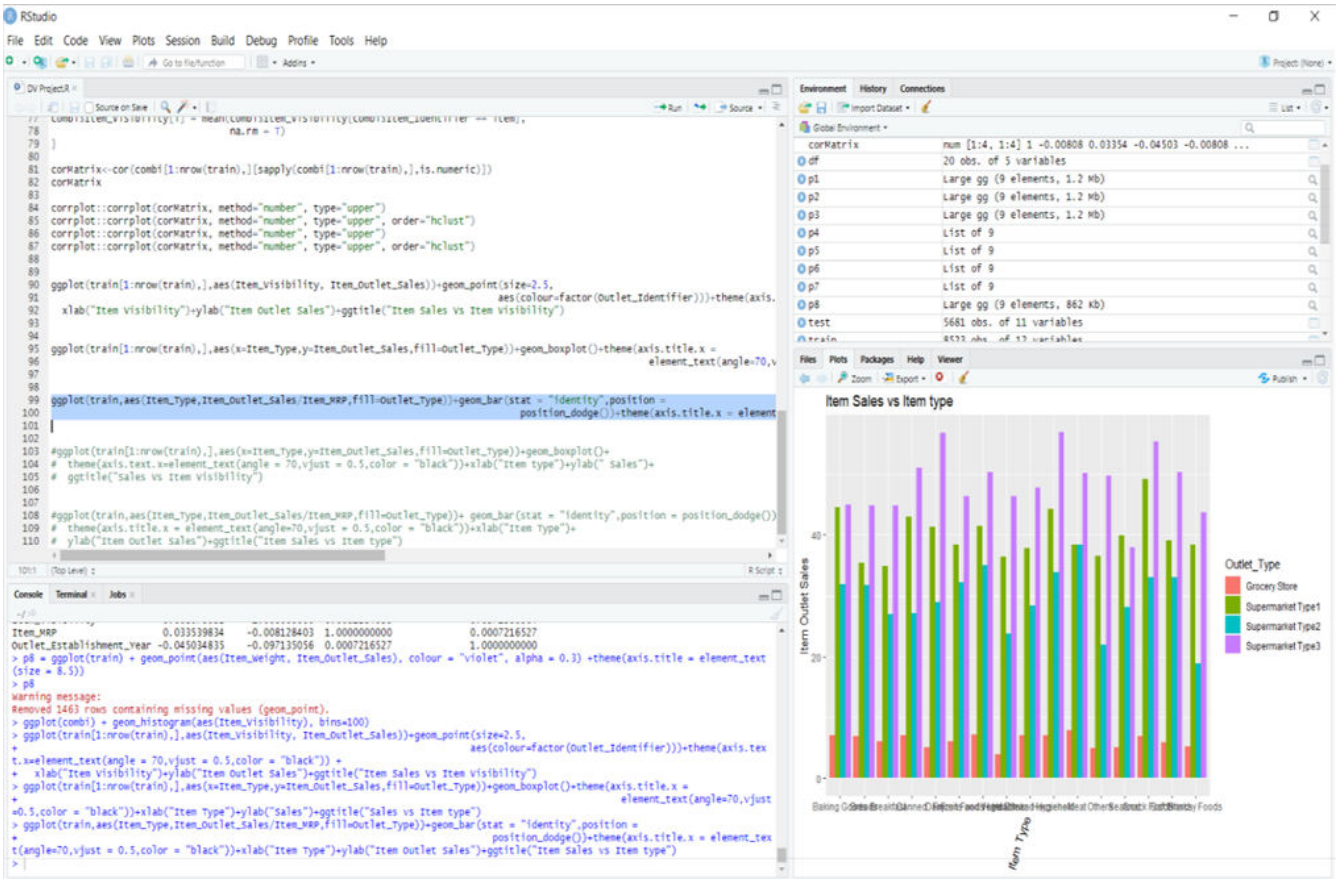
- Lesser number of observations in the data for the outlets established in the year 1998 as compared to the other years.
- Supermarket Type 1 seems to be the most popular category of Outlet_Type.



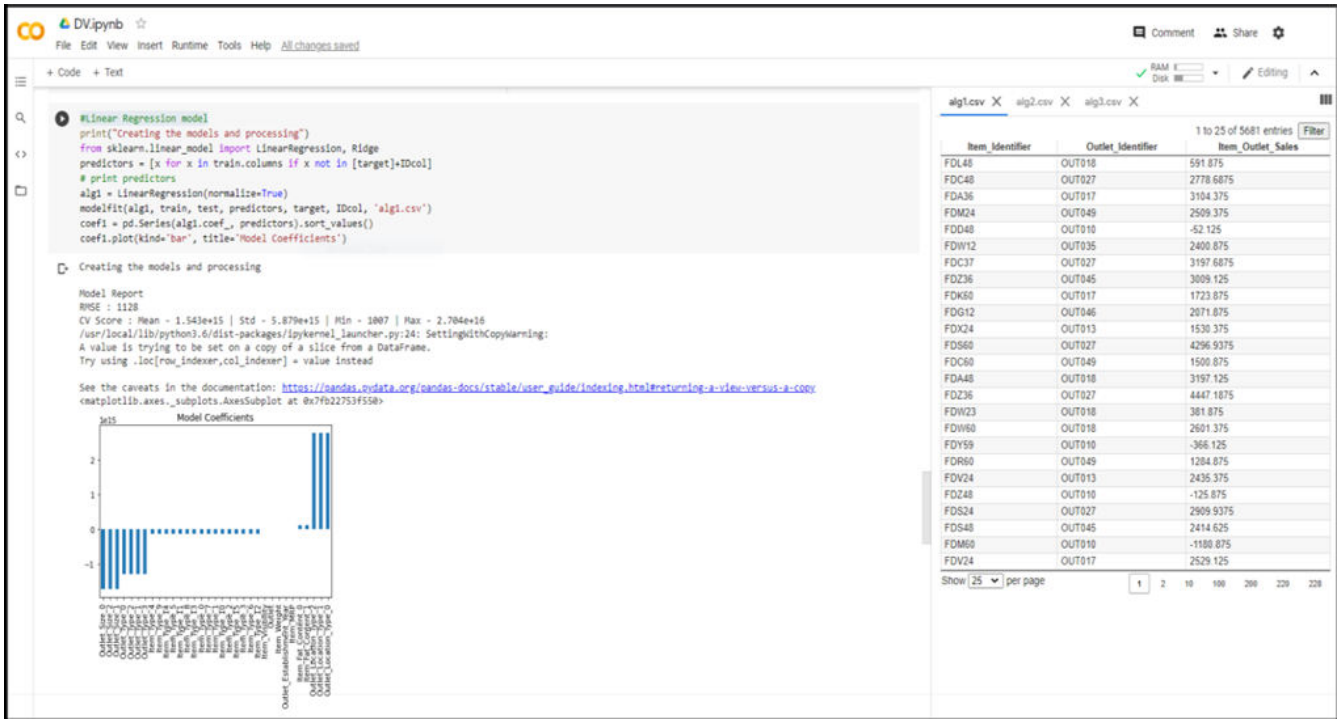

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
DV Project.R
p5=ggplot(combi %>% group_by(Outlet_Identifier) %>% summarise(Count=n())) + geom_bar(aes(Outlet_Identifier, Count), stat="identity
p5
p6=ggplot(combi %>% group_by(Outlet_Establishment_Year) %>% summarise(Count=n()))+ geom_bar(aes(Outlet_Establishment_Year,Count),
p6
p7=ggplot(combi %>% group_by(Outlet_Type) %>% summarise(Count=n())) + geom_bar(aes(Outlet_Type,Count), stat="identity", fill="cora
p7
p8 = ggplot(train) + geom_point(aes(Item_Weight, Item_Outlet_Sales), colour = "violet", alpha = 0.3) +theme(axis.title = element_t
p8
sm = sum(is.na(combi$Item_Weight))
nonNullVal = nrow(combi) - sm
missing_index = which(is.na(combi$Item_Weight))
for(i in missing_index){
item = combi$Item_Identifier[i]
combi$Item_Weight[i] = sum(combi$Item_Weight, na.rm = T)/nonNullVal
}
ggplot(combi) + geom_histogram(aes(Item_Visibility), bins=100)
zero_index = which(combi$Item_Visibility == 0)
for(i in zero_index){ item = combi$Item_Identifier[i]
combi$Item_Visibility[i] = mean(combi$Item_Visibility[combi$Item_Identifier == item],
na.rm = T)
}
corMatrix<-cor(combi[1:nrow(train),][sapply(combi[1:nrow(train),],is.numeric)])
corMatrix
corrplot::corrplot(corMatrix, method="number", type="upper")
corrplot::corrplot(corMatrix, method="number", type="upper", order="hclust")
Console Terminal Jobs
+ ylab("Item Outlet Sales")+ggtitle("Item Sales vs Item Type")
> ggplot(train,aes(Item_Type,Item_Outlet_Sales/Item_MRP,fill=Outlet_Type))+geom_bar(stat = "identity",position =
+ position_dodge())+theme(axis.title.x = element_tex
t(angle=70,vjust = 0.5,color = "black"))+xlab("Item Type")+ylab("Item Outlet Sales")+ggtitle("Item Sales vs Item type")
> ggplot(combi) + geom_histogram(aes(Item_visibility), bins=100)
> corMatrix<-cor(combi[1:nrow(train),][sapply(combi[1:nrow(train),],is.numeric)])
> corMatrix
Item_Weight Item_Visibility Item_MRP Outlet_Establishment_Year
Item_Weight 1.000000000 -0.008075032 0.0335398340 -0.0450348348
Item_Visibility -0.008075032 1.000000000 -0.0081284035 -0.0971350564
Item_MRP 0.033539834 -0.008128403 1.0000000000 0.0007216527
Outlet_Establishment_Year -0.045034835 -0.097135056 0.0007216527 1.0000000000
> p8 = ggplot(train) + geom_point(aes(Item_Weight, Item_Outlet_Sales), colour = "violet", alpha = 0.3) +theme(axis.title = element_text
(size = 8.5))
> p8
warning message:
Removed 1463 rows containing missing values (geom_point).
> ggplot(combi) + geom_histogram(aes(Item_visibility), bins=100)
Environment History Connections
corMatrix num [1:4, 1:4] 1 -0.00808 0.03354 -0.04503 -0.00808 ...
df 20 obs. of 5 variables
p1 Large gg (9 elements, 1.2 Mb)
p2 Large gg (9 elements, 1.2 Mb)
p3 Large gg (9 elements, 1.2 Mb)
p4 List of 9
p5 List of 9
p6 List of 9
p7 List of 9
p8 Large gg (9 elements, 862 Kb)
test 5681 obs. of 11 variables
Files Plots Packages Help Viewer
count
Item_Visibility


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
DV Project.R
ggplot(combi) + geom_histogram(aes(Item_visibility), bins=100)
zero_index = which(combi$Item_Visibility == 0)
for(i in zero_index){ item = combi$Item_Identifier[i]
combi$Item_Visibility[i] = mean(combi$Item_Visibility[combi$Item_Identifier == item],
na.rm = T)
}
corMatrix<-cor(combi[1:nrow(train),][sapply(combi[1:nrow(train),],is.numeric)])
corMatrix
corrplot::corrplot(corMatrix, method="number", type="upper")
corrplot::corrplot(corMatrix, method="number", type="upper", order="hclust")
corrplot::corrplot(corMatrix, method="number", type="upper")
corrplot::corrplot(corMatrix, method="number", type="upper", order="hclust")
ggplot(train[1:nrow(train),],aes(Item_visibility, Item_Outlet_Sales))+geom_point(size=2.5,
aes(colour=factor(Outlet_Identifier)))+theme(axis.
xlab("Item visibility")+ylab("Item Outlet Sales")+ggtitle("Item Sales vs Item visibility")
ggplot(train[1:nrow(train),],aes(x=Item_Type,y=Item_Outlet_Sales,fill=Outlet_Type))+geom_boxplot()+theme(axis.title.x =
element_text(angle=70,v
ggplot(train,aes(Item_Type,Item_Outlet_Sales/Item_MRP,fill=Outlet_Type))+geom_bar(stat = "identity",position =
position_dodge())+theme(axis.title.x = element
#ggplot(train[1:nrow(train),],aes(x=Item_Type,y=Item_Outlet_Sales,fill=Outlet_Type))+geom_boxplot()+
theme(axis.text.x=element_text(angle = 70,vjust = 0.5,color = "black"))+xlab("Item type")+ylab(" sales")+
ggtitle("Sales Vs Item visibility")
Console Terminal Jobs
> ggplot(combi) + geom_histogram(aes(Item_visibility), bins=100)
> corMatrix<-cor(combi[1:nrow(train),][sapply(combi[1:nrow(train),],is.numeric)])
> corMatrix
Item_Weight Item_Visibility Item_MRP Outlet_Establishment_Year
Item_Weight 1.000000000 -0.008075032 0.0335398340 -0.0450348348
Item_Visibility -0.008075032 1.000000000 -0.0081284035 -0.0971350564
Item_MRP 0.033539834 -0.008128403 1.0000000000 0.0007216527
Outlet_Establishment_Year -0.045034835 -0.097135056 0.0007216527 1.0000000000
> p8 = ggplot(train) + geom_point(aes(Item_Weight, Item_Outlet_Sales), colour = "violet", alpha = 0.3) +theme(axis.title = element_text
(size = 8.5))
> p8
warning message:
Removed 1463 rows containing missing values (geom_point).
> ggplot(combi) + geom_histogram(aes(Item_visibility), bins=100)
> ggplot(train[1:nrow(train),],aes(Item_visibility, Item_Outlet_Sales))+geom_point(size=2.5,
+ aes(colour=factor(Outlet_Identifier)))+theme(axis.tex
t.x=element_text(angle = 70,vjust = 0.5,color = "black")) +
+ xlab("Item visibility")+ylab("Item Outlet Sales")+ggtitle("Item Sales vs Item visibility")
Environment History Connections
corMatrix num [1:4, 1:4] 1 -0.00808 0.03354 -0.04503 -0.00808 ...
df 20 obs. of 5 variables
p1 Large gg (9 elements, 1.2 Mb)
p2 Large gg (9 elements, 1.2 Mb)
p3 Large gg (9 elements, 1.2 Mb)
p4 List of 9
p5 List of 9
p6 List of 9
p7 List of 9
p8 Large gg (9 elements, 862 Kb)
test 5681 obs. of 11 variables
Files Plots Packages Help Viewer
Item Sales Vs Item Visibility
Item Outlet Sales
Item Visibility
factor(Outlet_Identifier)
OUT010
OUT013
OUT017
OUT018
OUT019
OUT027
OUT035
OUT045
OUT046
OUT049


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Sales vs Item type
Item_Type
Sales
Outlet_Type
Grocery Store
Supermarket Type1
Supermarket Type2
Supermarket Type3


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Item Sales vs Item type
Item_Type
Item Outlet Sales
Outlet_Type
Grocery Store
Supermarket Type1
Supermarket Type2
Supermarket Type3

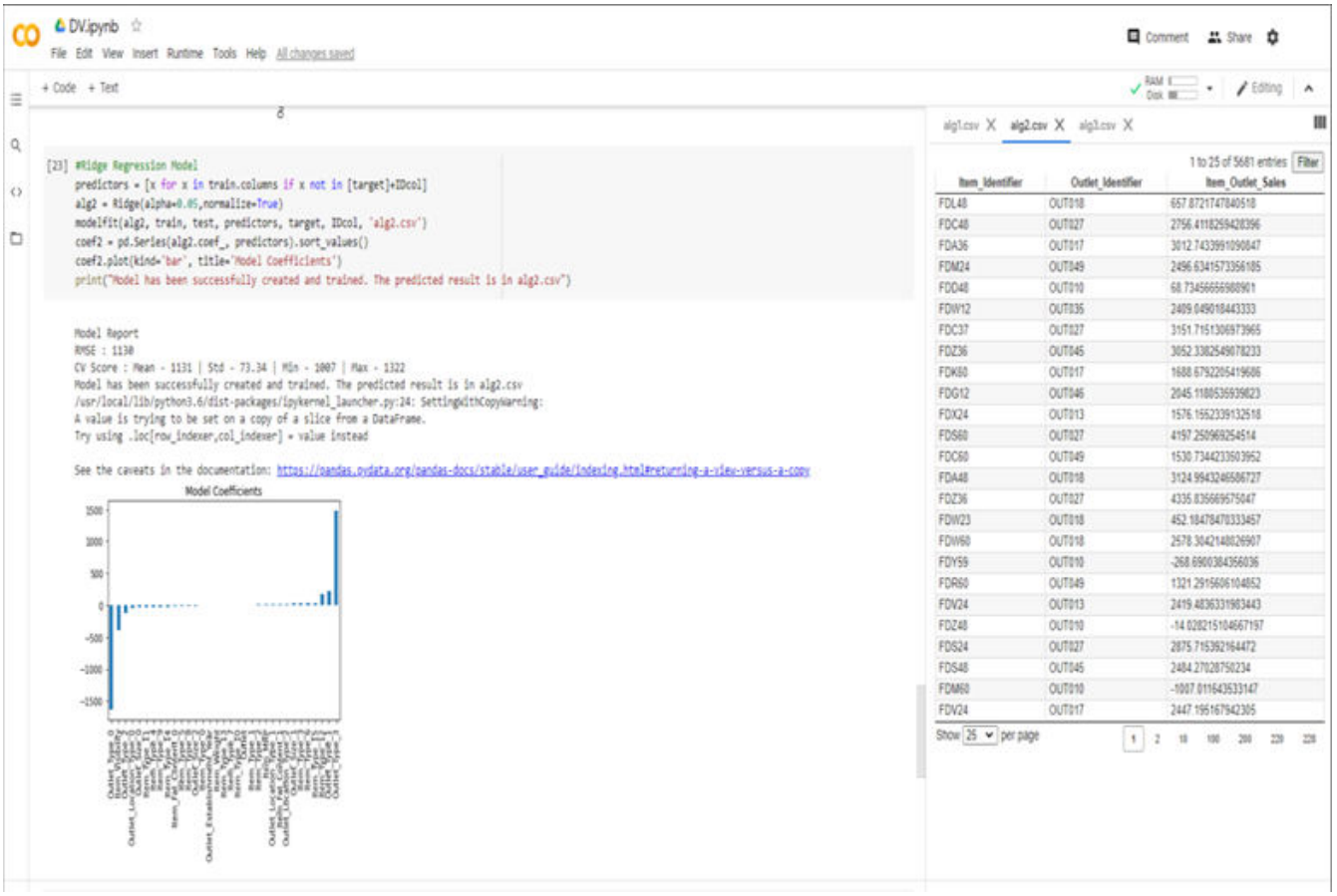
Linear Regression Model:



	A	B	C
1	Item_Identifier	Outlet_Identifier	Item_Outlet_Sales
2	FDL48	OUT018	591.875
3	FDC48	OUT027	2778.6875
4	FDA36	OUT017	3104.375
5	FDM24	OUT049	2509.375
6	FDD48	OUT010	-52.125
7	FDW12	OUT035	2400.875
8	FDC37	OUT027	3197.6875
9	FDZ36	OUT045	3009.125
10	FDK60	OUT017	1723.875
11	FDG12	OUT046	2071.875
12	FDX24	OUT013	1530.375
13	FDS60	OUT027	4296.9375
14	FDC60	OUT049	1500.875
15	FDA48	OUT018	3197.125
16	FDZ36	OUT027	4447.1875
17	FDW23	OUT018	381.875
18	FDW60	OUT018	2601.375
19	FDY59	OUT010	-366.125
20	FDR60	OUT049	1284.875
21	FDV24	OUT013	2435.375
22	FDZ48	OUT010	-125.875
23	FDS24	OUT027	2909.9375
24	FDS48	OUT045	2414.625
25	FDM60	OUT010	-1180.875
26	FDV24	OUT017	2529.125
27	FDS12	OUT027	3470.1875
28	FDP12	OUT027	2118.1875
29	FDV24	OUT046	2464.875

alg1 (1)

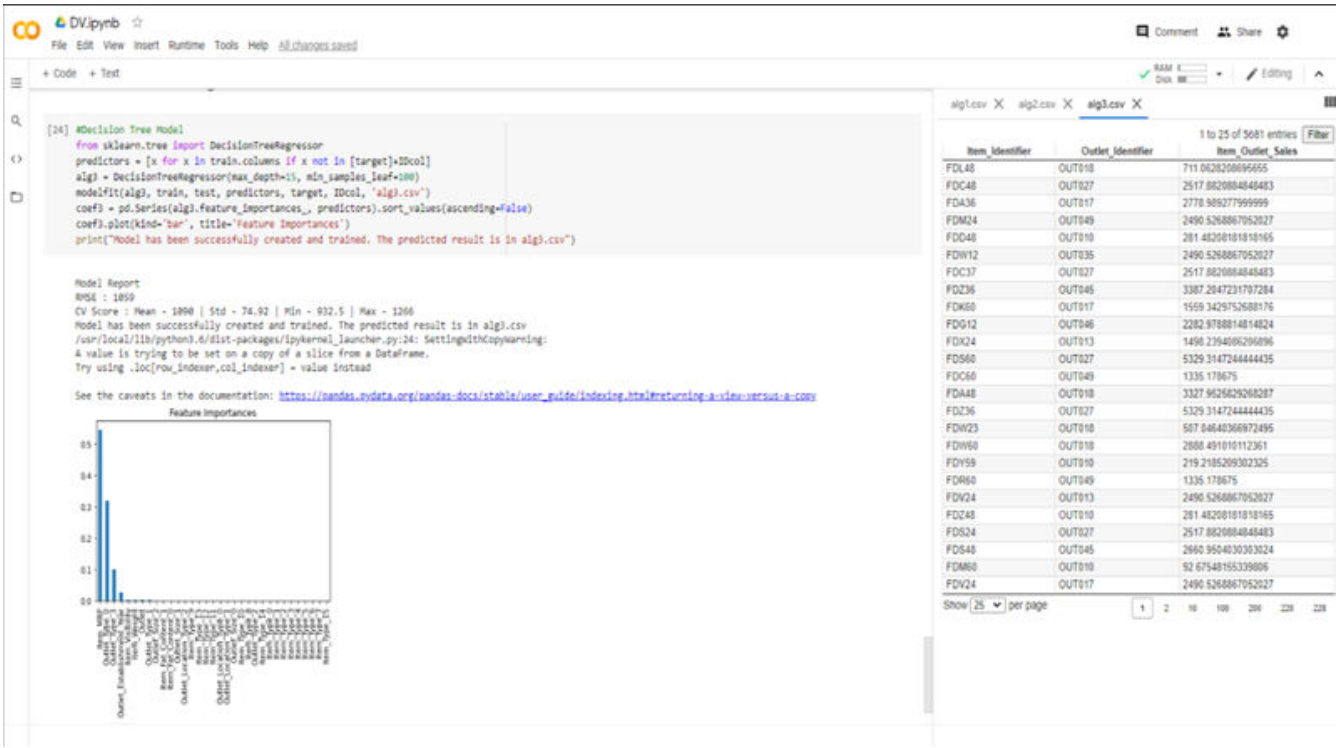
Ridge Regression Model:



	A	B	C
1	Item_Identifier	Outlet_Identifier	Item_Outlet_Sales
2	FDL48	OUT018	657.8721748
3	FDC48	OUT027	2756.411826
4	FDA36	OUT017	3012.743399
5	FDM24	OUT049	2496.634157
6	FDD48	OUT010	68.73456657
7	FDW12	OUT035	2409.049018
8	FDC37	OUT027	3151.715131
9	FDZ36	OUT045	3052.338255
10	FDK60	OUT017	1688.679221
11	FDG12	OUT046	2045.118054
12	FDX24	OUT013	1576.155234
13	FDS60	OUT027	4197.250969
14	FDC60	OUT049	1530.734423
15	FDA48	OUT018	3124.994325
16	FDZ36	OUT027	4335.83567
17	FDW23	OUT018	452.1847847
18	FDW60	OUT018	2578.304215
19	FDY59	OUT010	-268.6900384
20	FDR60	OUT049	1321.291561
21	FDV24	OUT013	2419.483633
22	FDZ48	OUT010	-14.0282151
23	FDS24	OUT027	2875.715392
24	FDS48	OUT045	2484.270288
25	FDM60	OUT010	-1007.011644
26	FDV24	OUT017	2447.195168
27	FDS12	OUT027	3384.625822
28	FDP12	OUT027	2125.929266
29	FDV24	OUT046	2397.080702

alg2

Decision Tree Model:



	A	B	C
1	Item_Identifier	Outlet_Identifier	Item_Outlet_Sales
2	FDL48	OUT018	711.0628209
3	FDC48	OUT027	2517.882088
4	FDA36	OUT017	2778.989278
5	FDM24	OUT049	2490.526887
6	FDD48	OUT010	281.4820818
7	FDW12	OUT035	2490.526887
8	FDC37	OUT027	2517.882088
9	FDZ36	OUT045	3387.204723
10	FDK60	OUT017	1559.342975
11	FDG12	OUT046	2282.978881
12	FDX24	OUT013	1498.239409
13	FDS60	OUT027	5329.314724
14	FDC60	OUT049	1335.178675
15	FDA48	OUT018	3327.952583
16	FDZ36	OUT027	5329.314724
17	FDW23	OUT018	507.0464037
18	FDW60	OUT018	2888.49101
19	FDY59	OUT010	219.2185209
20	FDR60	OUT049	1335.178675
21	FDV24	OUT013	2490.526887
22	FDZ48	OUT010	281.4820818
23	FDS24	OUT027	2517.882088
24	FDS48	OUT045	2660.950403
25	FDM60	OUT010	92.67548155
26	FDV24	OUT017	2490.526887
27	FDS12	OUT027	3242.062072
28	FDP12	OUT027	1378.263185
29	FDV24	OUT046	2490.526887

alg3

8. CONCLUSION AND FUTURE WORK:

8.1 CONCLUSION:

Nowadays shopping malls and Big Marts keep track of their sales data of each and every individual item for predicting future demand of the customer and update the inventory management as well. These data stores basically contain a large number of customer data and individual item attributes in a data warehouse. Further, anomalies and frequent patterns are detected by mining the data store from the data warehouse. The resultant data can be used for predicting future sales volume with the help of different machine learning techniques for the retailers like Big Mart. Some of the inferences that can be drawn from our research are -

- Item_MRP is the most important variable in predicting the target variable. New features created by us, like price_per_unit_wt, Outlet_Years, Item_MRP_Clusters, are also among the top most important variables.
- Item_outlet_Sales has a strong positive correlation with Item_MRP and a somewhat weaker negative with item_Visibility.
- In most of the cases the supermarket type3 has the most amount of sales irrespective of item_type.
- There seems to be no clear-cut pattern in Item_Weight
- We can clearly see 4 different distributions for Item_MRP. It is an interesting insight.

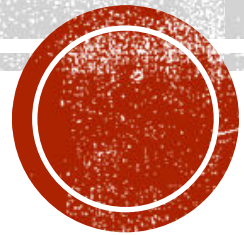
8.2 FUTURE WORK:

In present era of digitally connected world every shopping mall desires to know the customer demands beforehand to avoid the shortfall of sale items in all seasons. Day to day the companies or the malls are predicting more accurately the demand of product sales or user demands. Extensive research in this area at enterprise level is happening for accurate sales prediction. As the profit made by a company is directly proportional to the accurate predictions of sales, the Big marts are desiring more accurate prediction algorithms so that the company will not suffer any losses. The attributes that we considered till now in our prediction models can be increased further to make the results more effective and legit. Also, we can build a recommendation system customized to the respective supermarket which will help them to give them suggestions on how they can increase the overall sale.

THANKYOU!!! ????

CSE3020-DATA VISUALIZATION REVIEW-II

BIG MARKET SALES PREDICTION



Made By:
-Raghav Jindal (18BCE2080)
-Nimish Batra(18BCE2087)

SLOT:
B1+TB1

OBJECTIVE

This project aims to build a predictive model and find out the sales of each product at a particular store and hence predict the sales of supermarket according to the sample supermarket dataset.

The idea is to find out the properties of a product, and store which impacts the sales of a product. Using this model, we will try to understand the properties of products and stores which play a key role in increasing sales

We've also used various visual plots between the data to better understand it.

Algorithms used to build our models are:

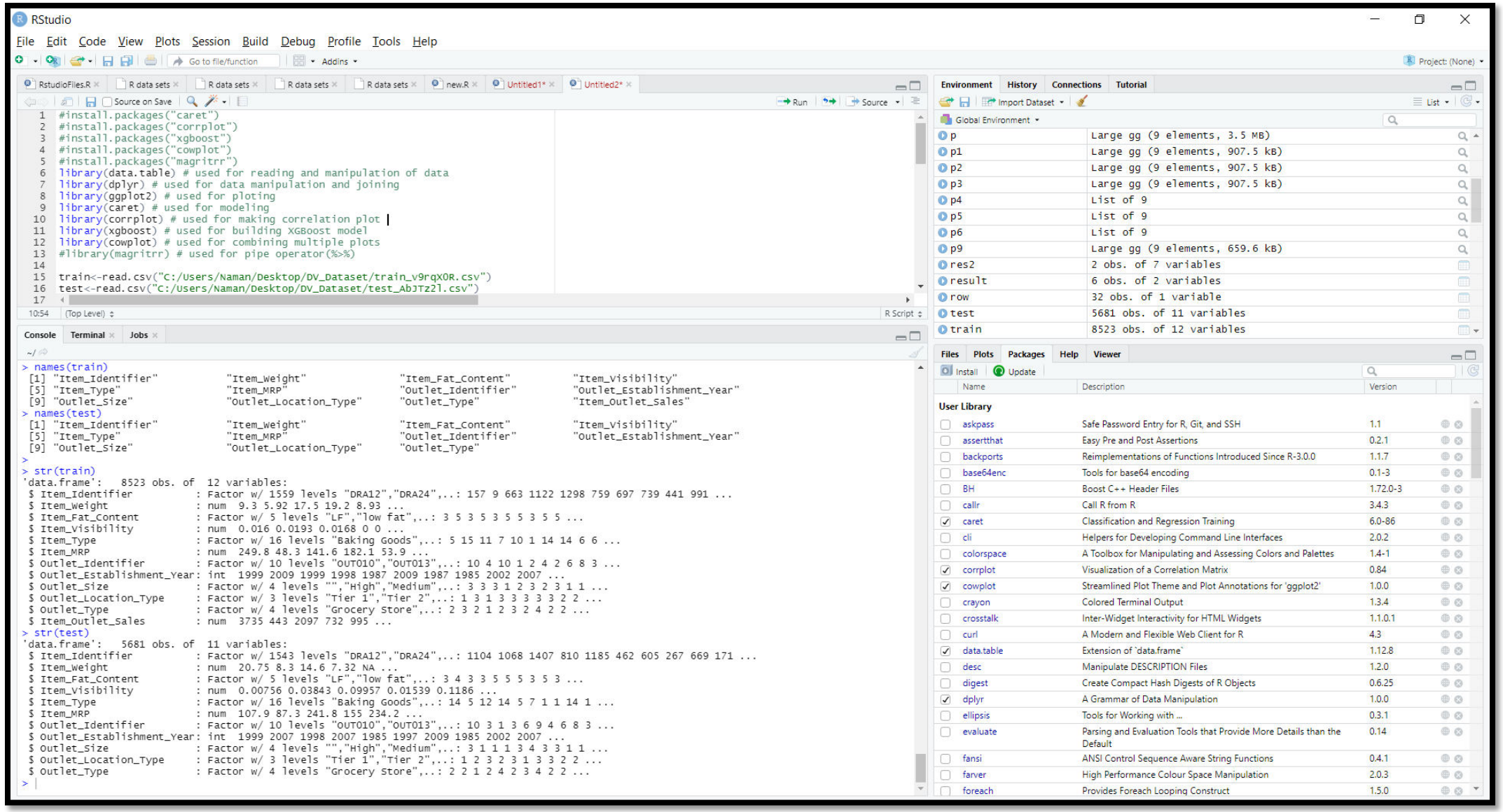
- Linear Regression
- Ridge Regression
- Random Forest Regression

DATASET USED

- Big_Mart_Sales
- Taken from Kaggle.com
- The training set contains 8523 rows (instances) of 12 variables (attributes).
- Testing set contains 5681 rows (instances) of 11 variables (attributes).

```
> dim(train)
[1] 8523   12
> dim(test)
[1] 5681   11
>
```

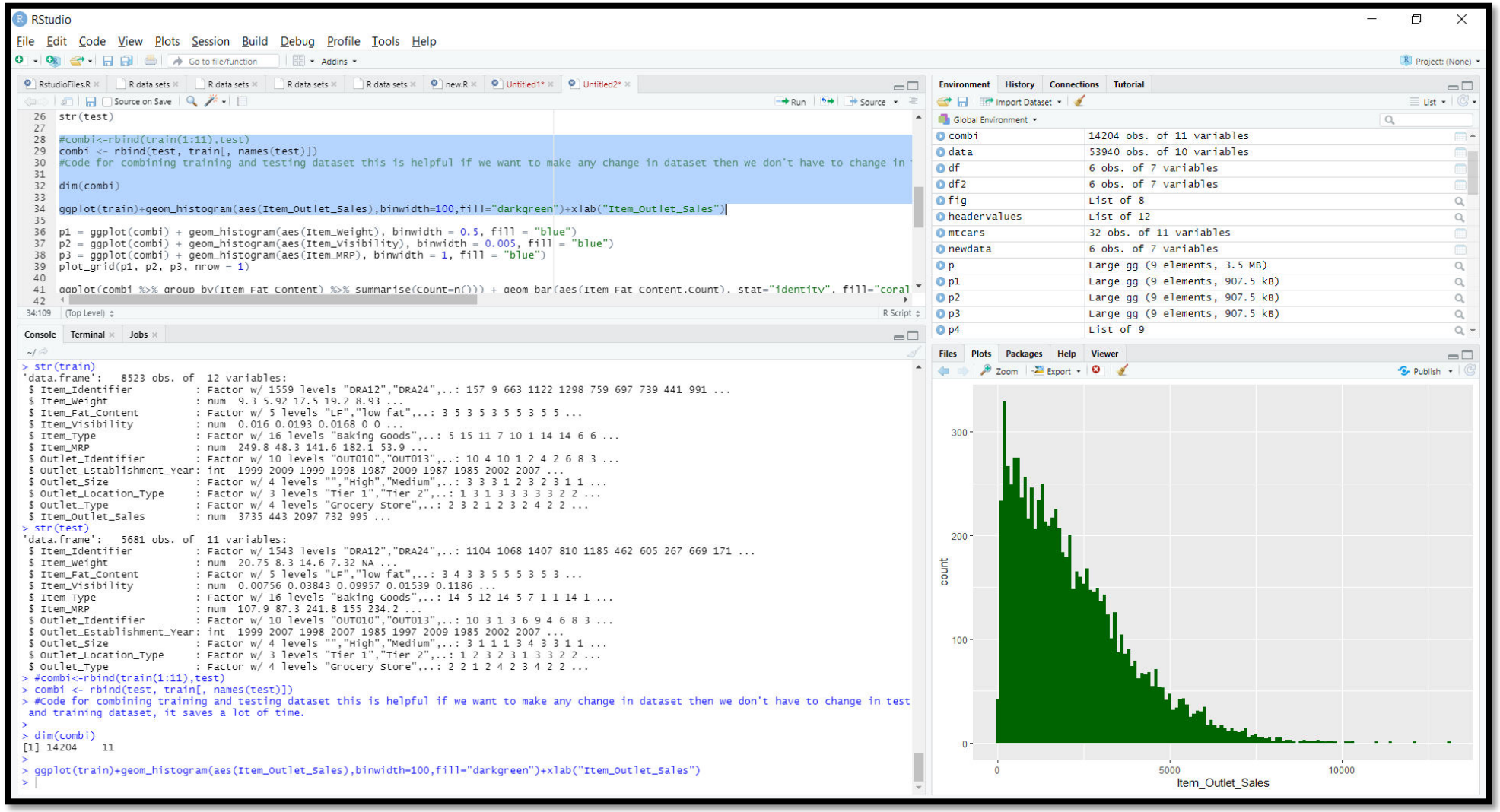
- Data attributes : Item_Identifier, Item_Weight, Item_Type, Item_MRP, etc.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
RstudioFiles.R
R data sets
R data sets
R data sets
R data sets
new.R
Untitled1*
Untitled2*
Source on Save
Run
Source
#install.packages("caret")
#install.packages("corrplot")
#install.packages("xgboost")
#install.packages("cowplot")
#install.packages("magritrr")
library(data.table) # used for reading and manipulation of data
library(dplyr) # used for data manipulation and joining
library(ggplot2) # used for ploting
library(caret) # used for modeling
library(corrplot) # used for making correlation plot
library(xgboost) # used for building XGBoost model
library(cowplot) # used for combining multiple plots
#library(magritrr) # used for pipe operator(%>%)
train<-read.csv("C:/Users/Naman/Desktop/DV_Dataset/train_v9rqX0R.csv")
test<-read.csv("C:/Users/Naman/Desktop/DV_Dataset/test_AbJTz2l.csv")
10:54
(Top Level)
R Script
Console
Terminal
Jobs
~/
> names(train)
[1] "Item_Identifier" "Item_Weight" "Item_Fat_Content" "Item_Visibility"
[5] "Item_Type" "Item_MRP" "Outlet_Identifier" "Outlet_Establishment_Year"
[9] "Outlet_Size" "Outlet_Location_Type" "Outlet_Type" "Item_Outlet_Sales"
> names(test)
[1] "Item_Identifier" "Item_Weight" "Item_Fat_Content" "Item_Visibility"
[5] "Item_Type" "Item_MRP" "Outlet_Identifier" "Outlet_Establishment_Year"
[9] "Outlet_Size" "Outlet_Location_Type" "Outlet_Type"
> str(train)
'data.frame': 8523 obs. of 12 variables:
$ Item_Identifier : Factor w/ 1559 levels "DRA12","DRA24",..: 157 9 663 1122 1298 759 697 739 441 991 ...
$ Item_Weight : num 9.3 5.92 17.5 19.2 8.93 ...
$ Item_Fat_Content : Factor w/ 5 levels "LF","low fat",..: 3 5 3 5 3 5 5 3 5 5 ...
$ Item_Visibility : num 0.016 0.0193 0.0168 0 0 ...
$ Item_Type : Factor w/ 16 levels "Baking Goods",..: 5 15 11 7 10 1 14 14 6 6 ...
$ Item_MRP : num 249.8 48.3 141.6 182.1 53.9 ...
$ Outlet_Identifier : Factor w/ 10 levels "OUT010","OUT013",..: 10 4 10 1 2 4 2 6 8 3 ...
$ Outlet_Establishment_Year: int 1999 2009 1999 1998 1987 2009 1987 1985 2002 2007 ...
$ Outlet_Size : Factor w/ 4 levels "","High","Medium",..: 3 3 3 1 2 3 2 3 1 1 ...
$ Outlet_Location_Type : Factor w/ 3 levels "Tier 1","Tier 2",..: 1 3 1 3 3 3 3 3 2 2 ...
$ Outlet_Type : Factor w/ 4 levels "Grocery Store",..: 2 3 2 1 2 3 2 4 2 2 ...
$ Item_Outlet_Sales : num 3735 443 2097 732 995 ...
> str(test)
'data.frame': 5681 obs. of 11 variables:
$ Item_Identifier : Factor w/ 1543 levels "DRA12","DRA24",..: 1104 1068 1407 810 1185 462 605 267 669 171 ...
$ Item_Weight : num 20.75 8.3 14.6 7.32 NA ...
$ Item_Fat_Content : Factor w/ 5 levels "LF","low fat",..: 3 4 3 3 5 5 5 3 5 3 ...
$ Item_Visibility : num 0.00756 0.03843 0.09957 0.01539 0.1186 ...
$ Item_Type : Factor w/ 16 levels "Baking Goods",..: 14 5 12 14 5 7 1 1 14 1 ...
$ Item_MRP : num 107.9 87.3 241.8 155 234.2 ...
$ Outlet_Identifier : Factor w/ 10 levels "OUT010","OUT013",..: 10 3 1 3 6 9 4 6 8 3 ...
$ Outlet_Establishment_Year: int 1999 2007 1998 2007 1985 1997 2009 1985 2002 2007 ...
$ Outlet_Size : Factor w/ 4 levels "","High","Medium",..: 3 1 1 1 3 4 3 3 1 1 ...
$ Outlet_Location_Type : Factor w/ 3 levels "Tier 1","Tier 2",..: 1 2 3 2 3 1 3 3 2 2 ...
$ Outlet_Type : Factor w/ 4 levels "Grocery Store",..: 2 2 1 2 4 2 3 4 2 2 ...
Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
p Large gg (9 elements, 3.5 MB)
p1 Large gg (9 elements, 907.5 kB)
p2 Large gg (9 elements, 907.5 kB)
p3 Large gg (9 elements, 907.5 kB)
p4 List of 9
p5 List of 9
p6 List of 9
p9 Large gg (9 elements, 659.6 kB)
res2 2 obs. of 7 variables
result 6 obs. of 2 variables
row 32 obs. of 1 variable
test 5681 obs. of 11 variables
train 8523 obs. of 12 variables
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
User Library
askpass Safe Password Entry for R, Git, and SSH 1.1
assertthat Easy Pre and Post Assertions 0.2.1
backports Reimplementations of Functions Introduced Since R-3.0.0 1.1.7
base64enc Tools for base64 encoding 0.1-3
BH Boost C++ Header Files 1.72.0-3
callr Call R from R 3.4.3
caret Classification and Regression Training 6.0-86
cli Helpers for Developing Command Line Interfaces 2.0.2
colorspace A Toolbox for Manipulating and Assessing Colors and Palettes 1.4-1
corrplot Visualization of a Correlation Matrix 0.84
cowplot Streamlined Plot Theme and Plot Annotations for 'ggplot2' 1.0.0
crayon Colored Terminal Output 1.3.4
crosstalk Inter-Widget Interactivity for HTML Widgets 1.1.0.1
curl A Modern and Flexible Web Client for R 4.3
data.table Extension of `data.frame` 1.12.8
desc Manipulate DESCRIPTION Files 1.2.0
digest Create Compact Hash Digests of R Objects 0.6.25
dplyr A Grammar of Data Manipulation 1.0.0
ellipsis Tools for Working with ... 0.3.1
evaluate Parsing and Evaluation Tools that Provide More Details than the Default 0.14
fansi ANSI Control Sequence Aware String Functions 0.4.1
farver High Performance Colour Space Manipulation 2.0.3
foreach Provides Foreach Looping Construct 1.5.0

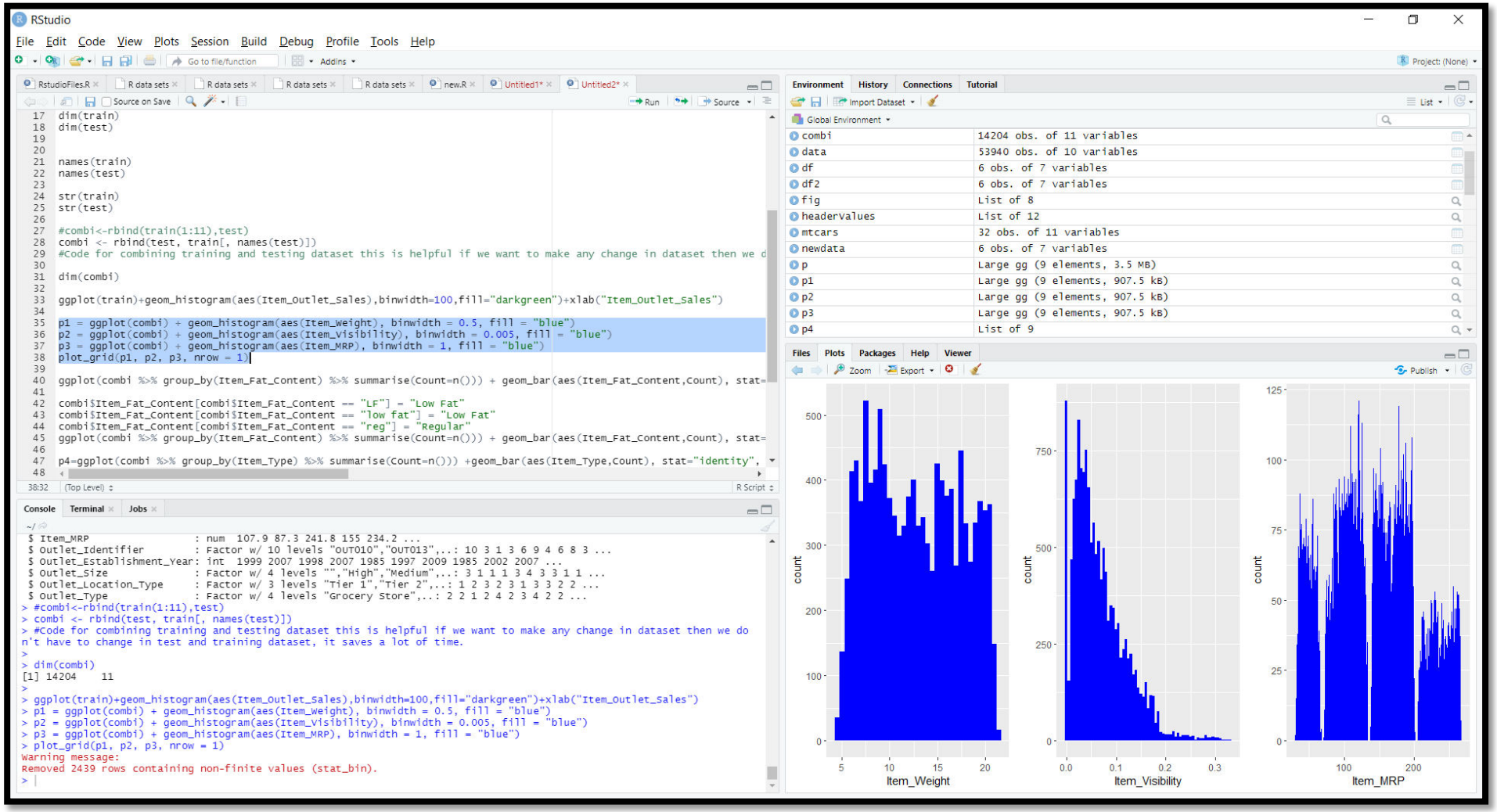
PACKAGES/LIBRARIES USED

```
#install.packages("caret")
#install.packages("corrplot")
#install.packages("xgboost")
#install.packages("cowplot")
#install.packages("magritrr")
library(data.table) # used for reading and manipulation of data
library(dplyr) # used for data manipulation and joining
library(ggplot2) # used for ploting
library(caret) # used for modeling
#library(corrplot) # used for making correlation plot
#library(xgboost) # used for building XGBoost model
#library(cowplot) # used for combining multiple plots
```

HISTOGRAM(GGPLOT2)-ITEM_OUTLET_SALES



HISTOGRAM(COWPLOT)-ITEM_WEIGHT, ITEM_VISIBILITY, ITEM_MRP



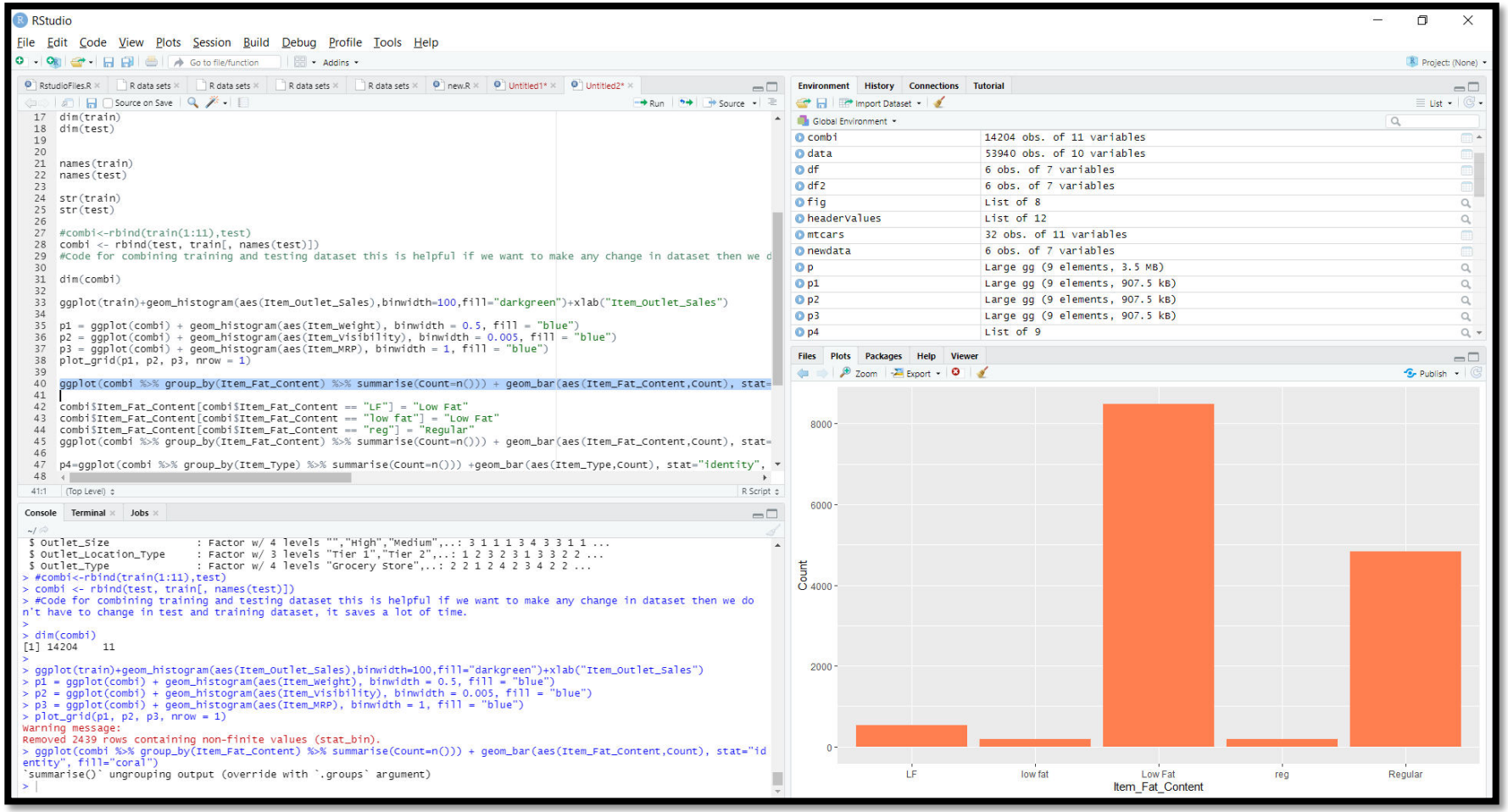
- **OBSERVATIONS:**

➢ No clear cut pattern in: Item_Weight, thus we get to know that it is randomly distributed.

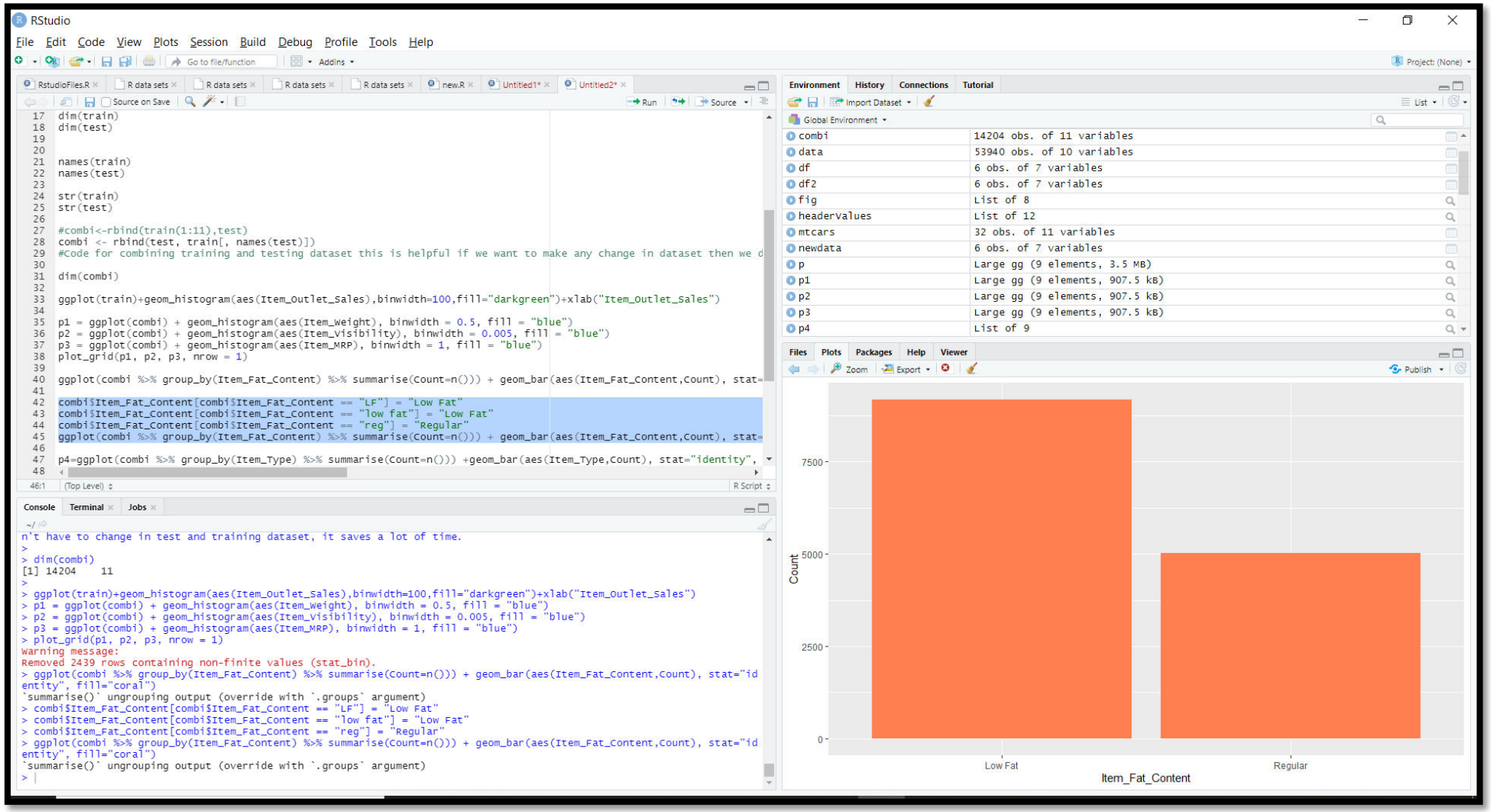
➢ Item_Visibility is right-skewed-thus must be transformed to curb skewness.

➢ 4 different distributions for Item_MRP, thus all the items are classified under 4 price ranges.

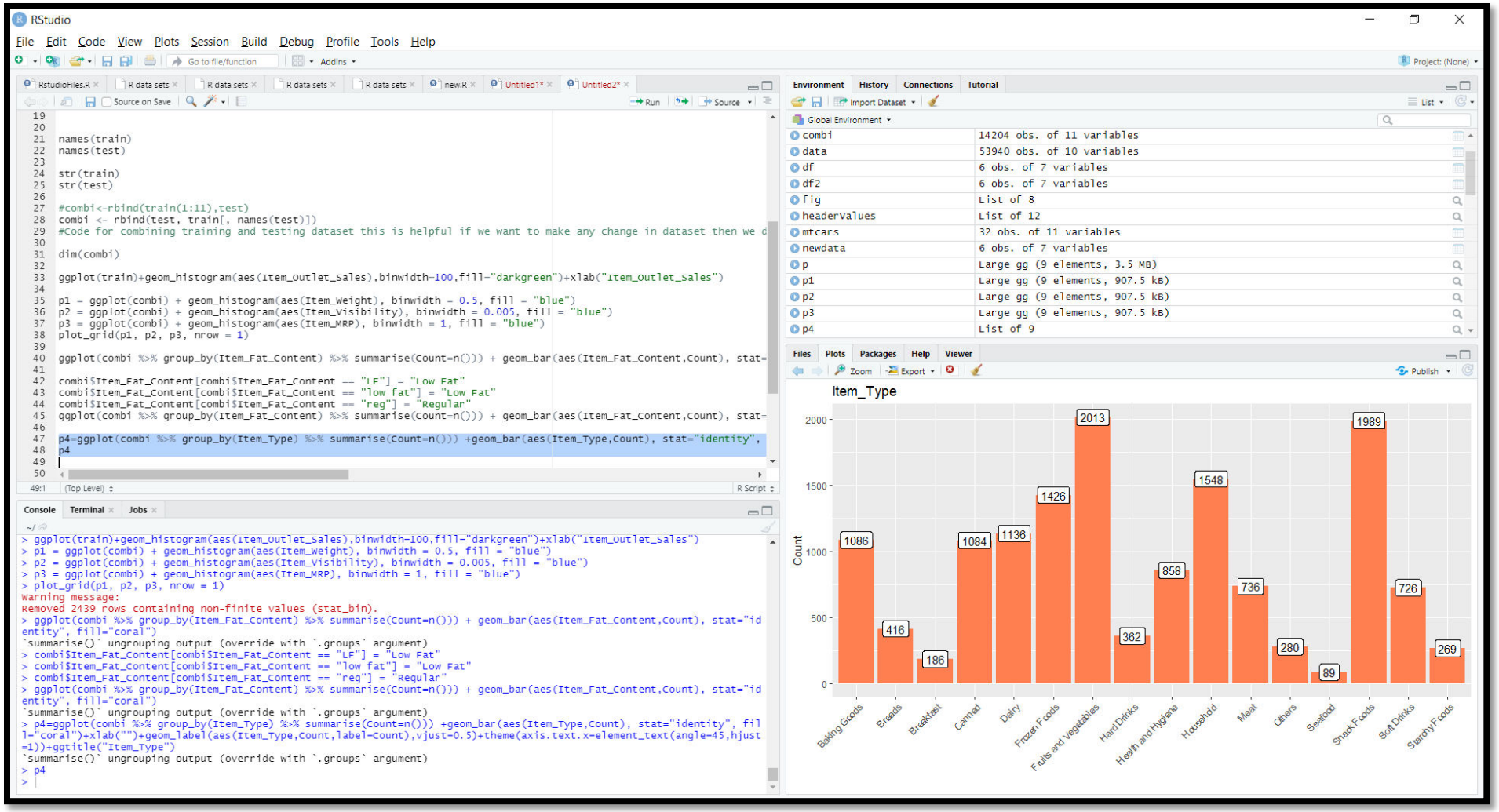
BAR GRAPH(DPLYR)-ITEM_FAT_CONTENT



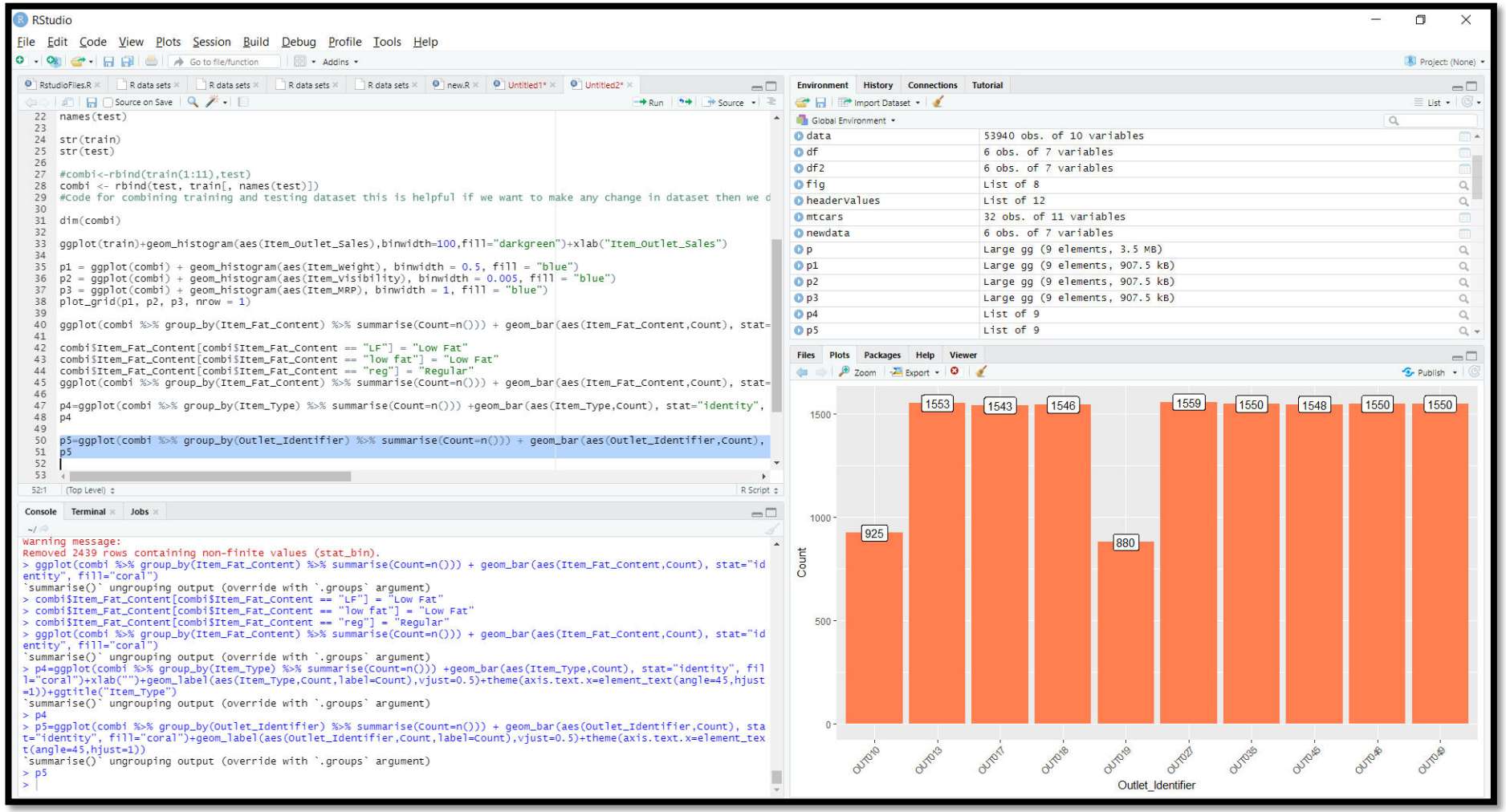
BAR GRAPH-ITEM_FAT_CONTENT



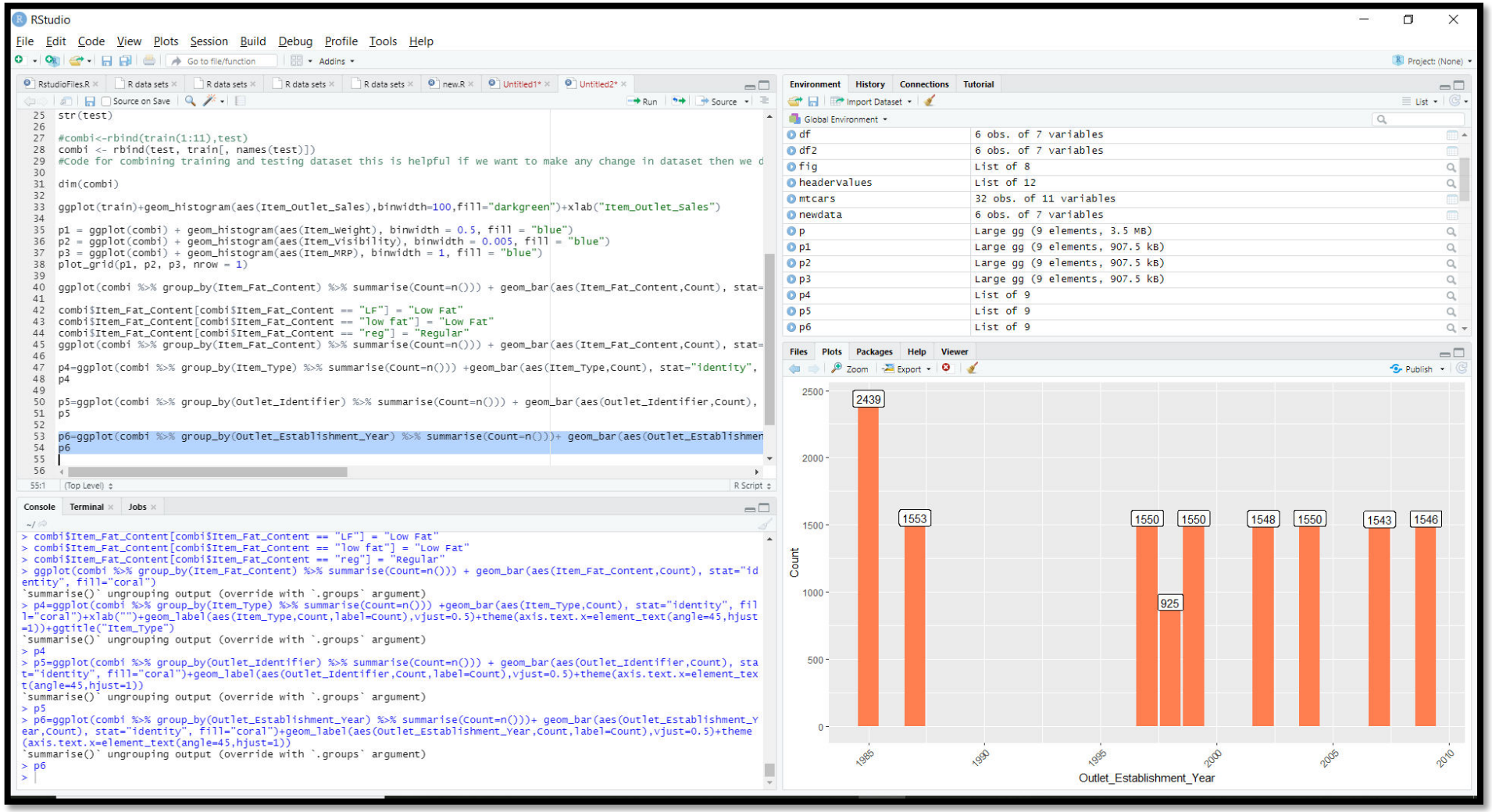
BAR GRAPH-ITEM_TYPE



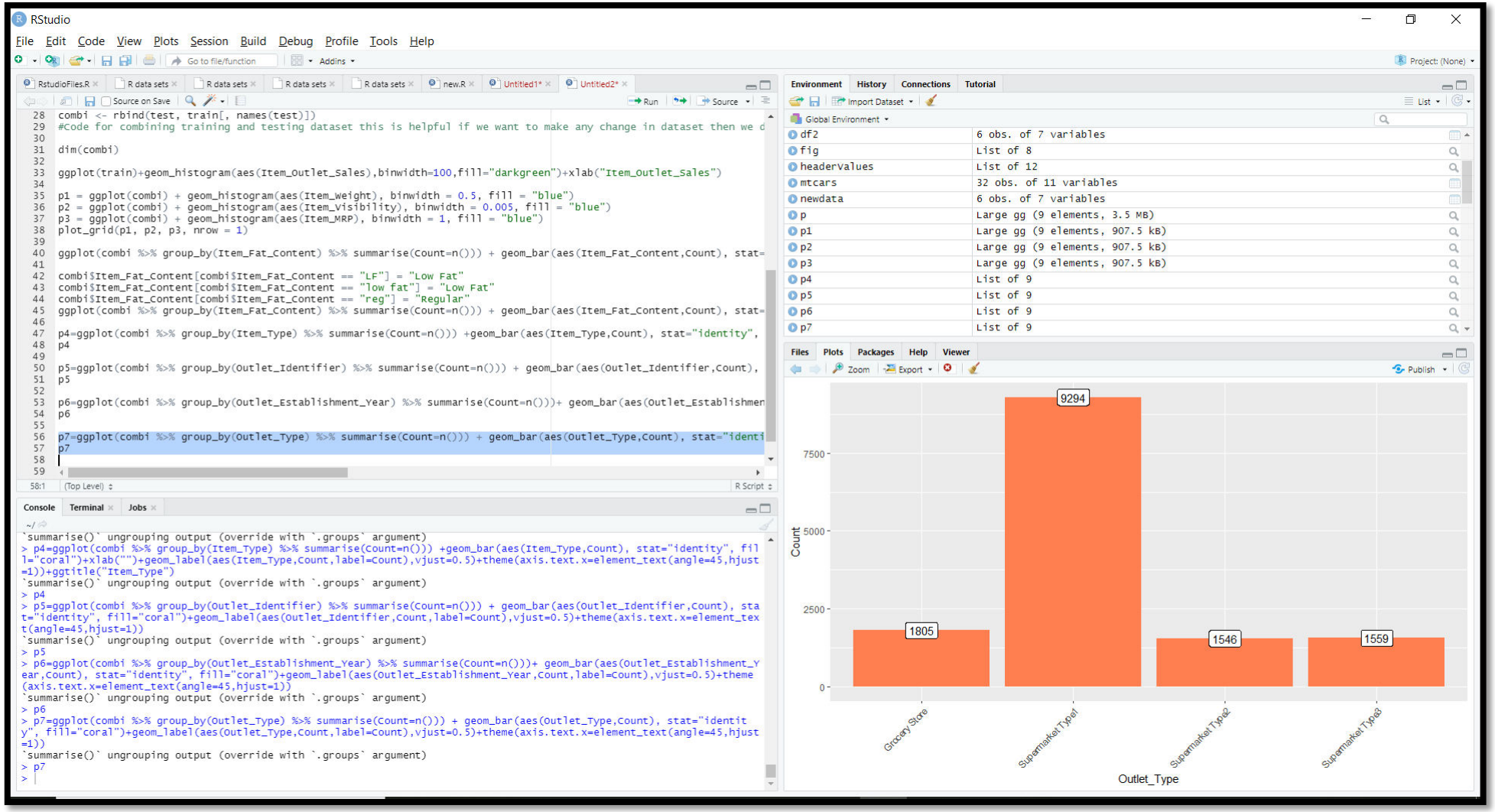
BAR GRAPH-OUTLET_IDENTIFIER



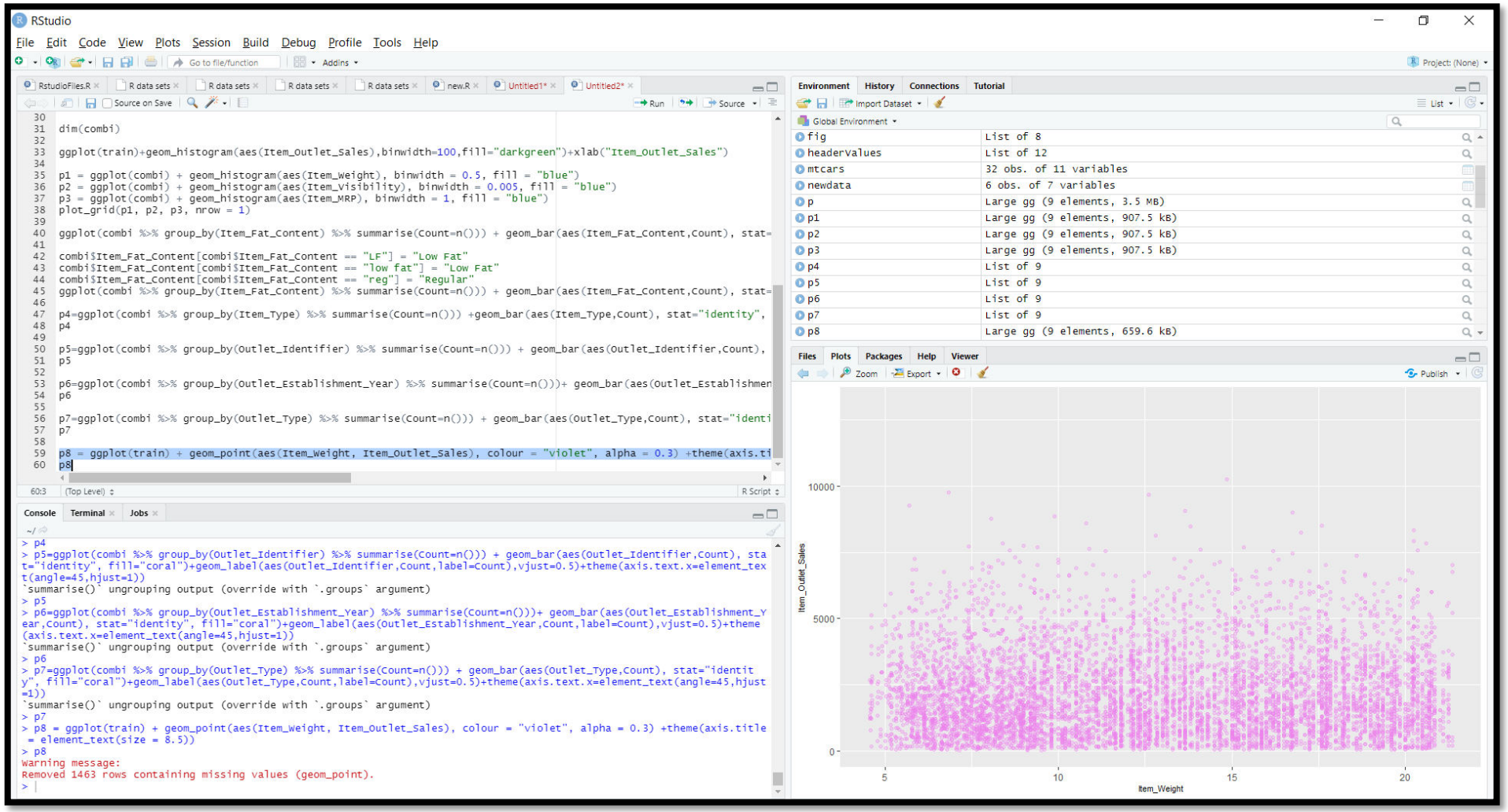
BAR GRAPH-OUTLET_ESTABLISHMENT_YEAR



BAR GRAPH-OUTLET_TYPE



SCATTER PLOT-ITEM_WEIGHT



- **OBSERVATIONS:**

➢Lesser number of observations in the data for the outlets established in the year 1998 as compared to the other years.

➢Supermarket Type 1 seems to be the most popular category of Outlet_Type.

- **MISSING VALUE HANDLING-**

If there are any missing values in the dataset's attributes, thus to make it work still, we have searched many ways to treat them based on the problem and the data. Some common techniques are as follows:

1. **Deletion Of Rows-**In train dataset, observations having missing values in any variable are deleted. The downside of this method is the loss of information and drop in prediction power of model.
2. **Mean/Median/Mode Imputation-**In case of continuous variable, missing values can be replaced with mean or median of all known values of that variable. For categorical variables, we can use mode of the given values to replace the missing values.
3. **Building Prediction Model-**We can even make a predictive model to impute missing data in a variable.

Here we will treat the variable having missing data as the target variable and the other variables as predictors.

We will divide our data into 2 datasets—one without any missing value for that variable and the other with missing values for that variable.

The former set would be used as training set to build the predictive model and it would then be applied to the latter set to predict the missing values.

- **TREATMENT-I:**

There are missing values in Item_Weight and Item_Outlet_Sales, Item_Outlet_Sales can be ignored as they belong to the testing dataset. For Item_Weight, we'll impute the missing ones with the mean weight on the Item_Identifier variable.

```
sum(is.na(combi$Item_Weight))
```

```
[1] 2439
```

```
missing_index=which(is.na(combi$Item_Weight))
```

```
> for(i in missing_index){
   + item=combi$Item_Identifier[i]
   + combi$Item_Weight[i]=mean(combi)
   + }
```

THANK YOU