Assignment 3 - Predicting Sales for Superstore

Jinping Bai, Joshua Dalphy, Choongil Kim and Gouri Kulkarni

Thursday, November 12th 2020

Contents

1 Business Introduction

1.1 Objective

The objective of this project is to develop a model or models which could be used to predict the amount of sales that a global superstore could generate, given their past transaction data.

2 Data Description

The dataset used for assignment 3 was obtained from https://www.kaggle.com/jr2ngb/superstore-data and contains four years (2011-2015) worth of retail data belonging to a global superstore. The dataset has 51,291 observations and 24 features, which are described in the following table:

Variable Name	Description
Row ID	Unique numbered identifier for each row
Order ID	Unique numbered identifier for each order
Order Date	Date the order was placed - dd/mm/yyyy
Ship Date	Date the order was shipped - dd/mm/yyyy
Ship Mode	Mode of shipment - First class, same day, standard class and second class
Customer ID	Unique numbered identifier for each customer
Customer Name	The name of the customer - <first name, last name>
Segment	The business segment - Consumer, home office and corporate
City	The city name where the order is to be shipped
State	The state name where the order is to be shipped
Country	The country name where the order is to be shipped
Postal Code	The postal code belonging to the individual
Market	The market where the order took place - Africa, APAC, Canada, EMEA, EU, LATAM and US
Region	The region where the sale took place - Africa, Canada, Caribbean, Central, Central Asia, East, EMEA, North, North Asia, Oceania, South, Southeast Asia and West
Product ID	Unique numbered identifier for each product
Category	The product's category - Furniture, office supplies and technology
Sub-Category	The product's subcategory
Product Name	The name of the product
Sales	The sales value in dollars
Quantity	The quantity of the order
Discount	The value of the discount if applicable in dollars
Profit	The value of profit associated to each order in dollars
Shipping Cost	The cost of shipping in dollars
Order Priority	The order's priority - Critical, high, medium and low

3 Data Exploration

3.1 General Data Exploration

In this section of the report, we will explore the dataset in order to help us better understand both the categorical features and the behavior of the data as a whole.

To begin the data exploration, we first start by inspecting the first few rows of the dataset.

```
##   Row.ID Order.Date  Ship.Date      Ship.Mode     Segment        City
## 1  42433 2011-01-01 2011-01-06 Standard Class    Consumer Constantine
## 2  22253 2011-01-01 2011-01-08 Standard Class    Consumer Wagga Wagga
## 3  48883 2011-01-01 2011-01-05   Second Class    Consumer    Budapest
## 4  11731 2011-01-01 2011-01-05   Second Class Home Office   Stockholm
##             State   Country Market  Region        Category Sub.Category   Sales
## 1     Constantine   Algeria Africa  Africa Office Supplies      Storage 408.300
## 2 New South Wales Australia   APAC Oceania Office Supplies     Supplies 120.366
## 3        Budapest   Hungary   EMEA    EMEA Office Supplies      Storage  66.120
## 4       Stockholm    Sweden     EU   North Office Supplies        Paper  44.865
##   Quantity Discount  Profit Shipping.Cost Order.Priority
## 1        2      0.0 106.140         35.46         Medium
## 2        3      0.1  36.036          9.72         Medium
## 3        4      0.0  29.640          8.17           High
## 4        3      0.5 -26.055          4.82           High
```

Then, the internal structure of the dataset is outputted.

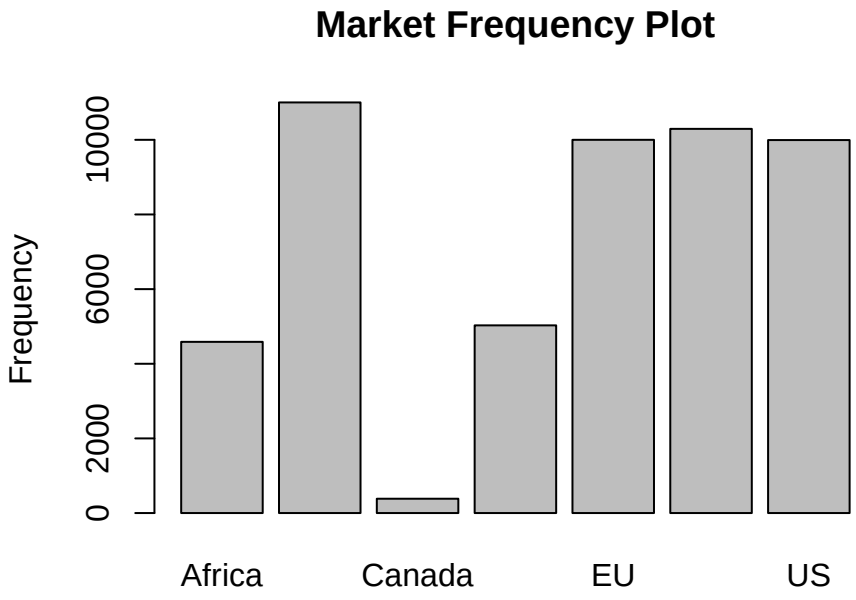
```
## 'data.frame':    51290 obs. of  18 variables:
##  $ Row.ID        : int  42433 22253 48883 11731 22255 22254 21613 34662 44508 23688 ...
##  $ Order.Date    : Date, format: "2011-01-01" "2011-01-01" ...
##  $ Ship.Date     : Date, format: "2011-01-06" "2011-01-08" ...
##  $ Ship.Mode     : chr  "Standard Class" "Standard Class" "Second Class" "Second Class" ...
##  $ Segment       : chr  "Consumer" "Consumer" "Consumer" "Home Office" ...
##  $ City          : chr  "Constantine" "Wagga Wagga" "Budapest" "Stockholm" ...
##  $ State         : chr  "Constantine" "New South Wales" "Budapest" "Stockholm" ...
##  $ Country       : chr  "Algeria" "Australia" "Hungary" "Sweden" ...
##  $ Market        : chr  "Africa" "APAC" "EMEA" "EU" ...
##  $ Region        : chr  "Africa" "Oceania" "EMEA" "North" ...
##  $ Category      : chr  "Office Supplies" "Office Supplies" "Office Supplies" "Office Supplies" ...
##  $ Sub.Category  : chr  "Storage" "Supplies" "Storage" "Paper" ...
##  $ Sales         : num  408.3 120.4 66.1 44.9 113.7 ...
##  $ Quantity      : int  2 3 4 3 5 2 2 2 1 3 ...
##  $ Discount      : num  0 0.1 0 0.5 0.1 0.1 0 0.15 0 0 ...
##  $ Profit        : num  106.1 36 29.6 -26.1 37.8 ...
##  $ Shipping.Cost : num  35.46 9.72 8.17 4.82 4.7 ...
##  $ Order.Priority: chr  "Medium" "Medium" "High" "High" ...
```

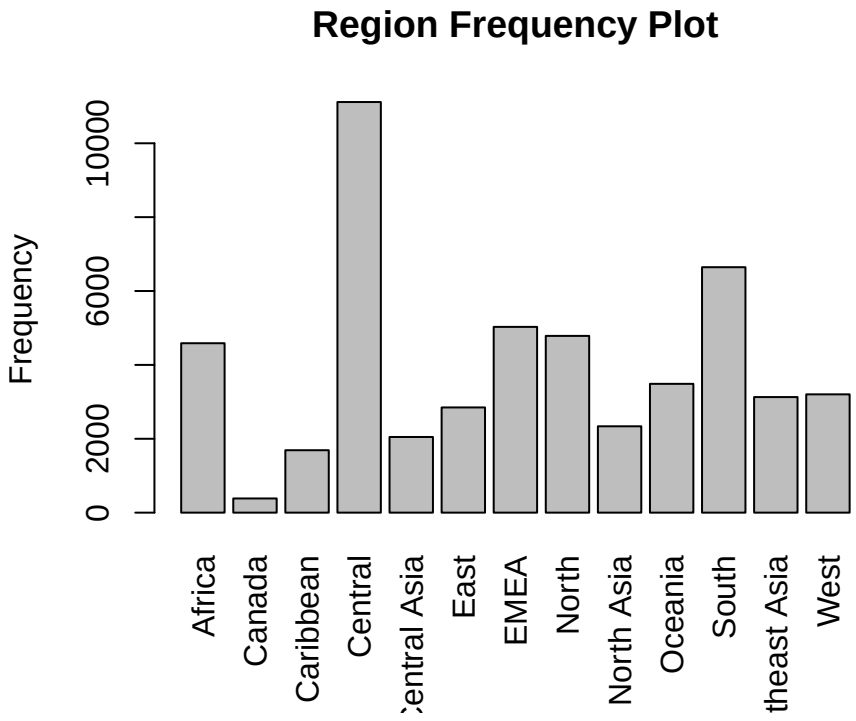
Then the summary function is used to understand the features at a high level and to retrieve the basic descriptive statistics.

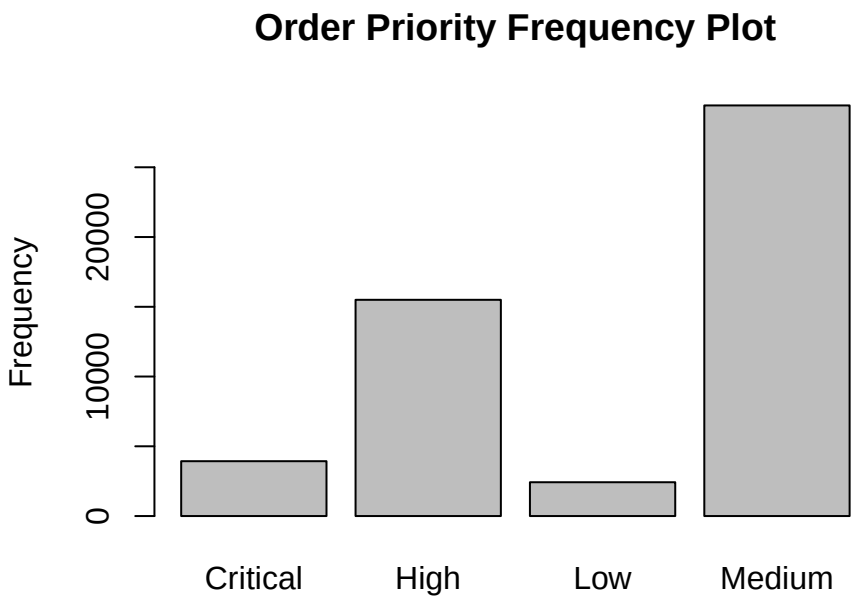
```
##      Row.ID        Order.Date           Ship.Date           Ship.Mode
##  Min.   :    1   Min.   :2011-01-01   Min.   :2011-01-03   Length:51290
##  1st Qu.:12823   1st Qu.:2012-06-08   1st Qu.:2012-06-11   Class :character
##  Median :25646   Median :2013-07-02   Median :2013-07-04   Mode  :character
##  Mean   :25646   Mean   :2013-05-02   Mean   :2013-05-10
##  3rd Qu.:38468   3rd Qu.:2014-05-11   3rd Qu.:2014-06-01
##  Max.   :51290   Max.   :2014-12-12   Max.   :2015-01-07
##                  NA's   :31223        NA's   :31456
##    Segment              City              State             Country
##  Length:51290       Length:51290       Length:51290       Length:51290
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

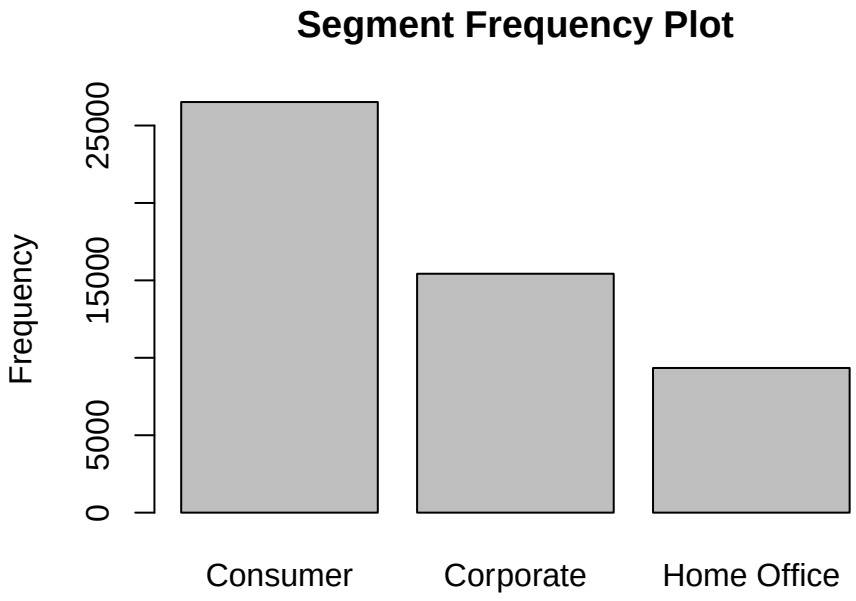
```
##
##
##
##
##     Market             Region            Category         Sub.Category
##  Length:51290       Length:51290       Length:51290       Length:51290
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      Sales              Quantity         Discount          Profit
##  Min.   :    0.444   Min.   : 1.000   Min.   :0.0000   Min.   :-6599.98
##  1st Qu.:   30.759   1st Qu.: 2.000   1st Qu.:0.0000   1st Qu.:    0.00
##  Median :   85.053   Median : 3.000   Median :0.0000   Median :    9.24
##  Mean   :  246.491   Mean   : 3.477   Mean   :0.1429   Mean   :   28.61
##  3rd Qu.:  251.053   3rd Qu.: 5.000   3rd Qu.:0.2000   3rd Qu.:   36.81
##  Max.   :22638.480   Max.   :14.000   Max.   :0.8500   Max.   : 8399.98
##
##  Shipping.Cost    Order.Priority
##  Min.   :  0.00   Length:51290
##  1st Qu.:  2.61   Class :character
##  Median :  7.79   Mode  :character
##  Mean   : 26.38
##  3rd Qu.: 24.45
##  Max.   :933.57
##
```

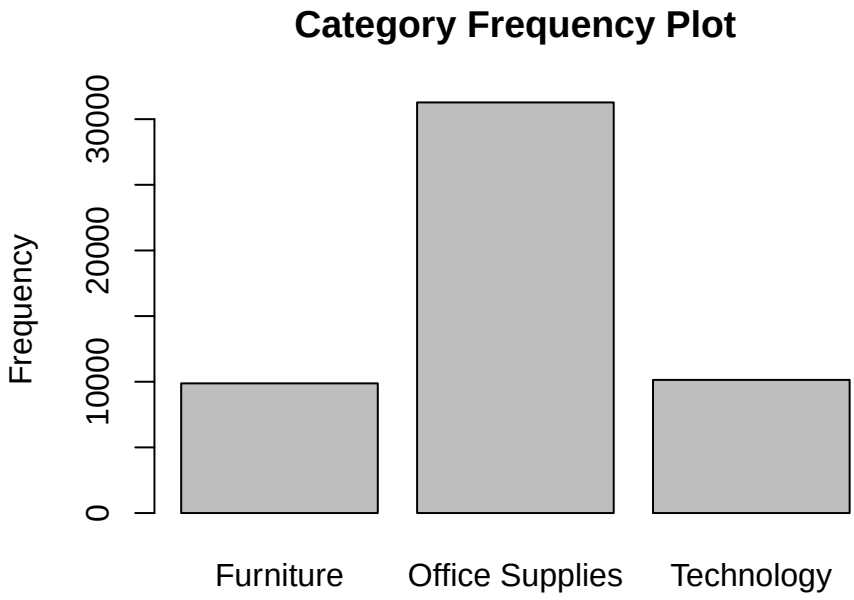
During the exploration of key categorical variables, plotting the frequency distribution can help us better understand the features.

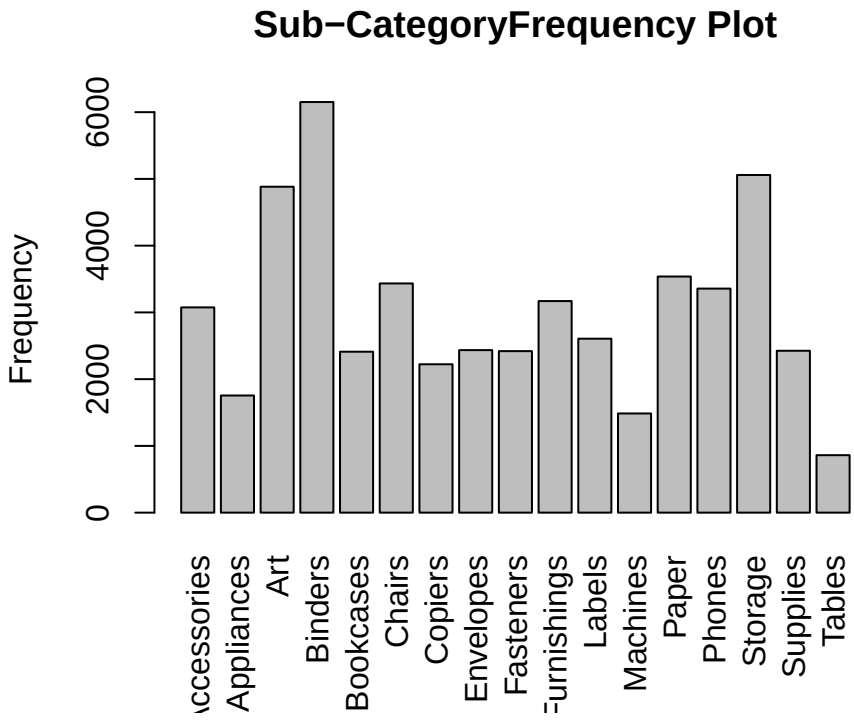

Market Frequency Plot
Frequency
0
2000
6000
10000
Africa
Canada
EU
US


Region Frequency Plot
Frequency
0
2000
6000
10000
Africa
Canada
Caribbean
Central
Central Asia
East
EMEA
North
North Asia
Oceania
South
theast Asia
West


Order Priority Frequency Plot
Frequency
0
10000
20000
Critical
High
Low
Medium


Segment Frequency Plot
Frequency
0
5000
15000
25000
Consumer
Corporate
Home Office

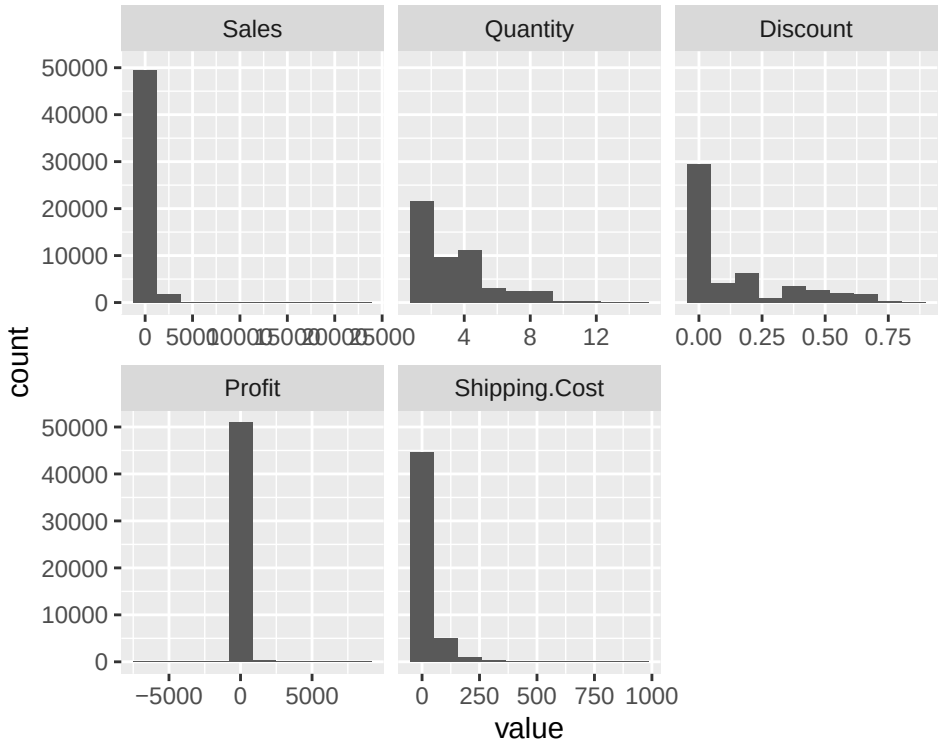

Category Frequency Plot
Frequency
0
10000
20000
30000
Furniture
Office Supplies
Technology


Sub−CategoryFrequency Plot
Frequency
0
2000
4000
6000
ccessories
Appliances
Art
Binders
Bookcases
Chairs
Copiers
Envelopes
Fasteners
urnishings
Labels
Machines
Paper
Phones
Storage
Supplies
Tables

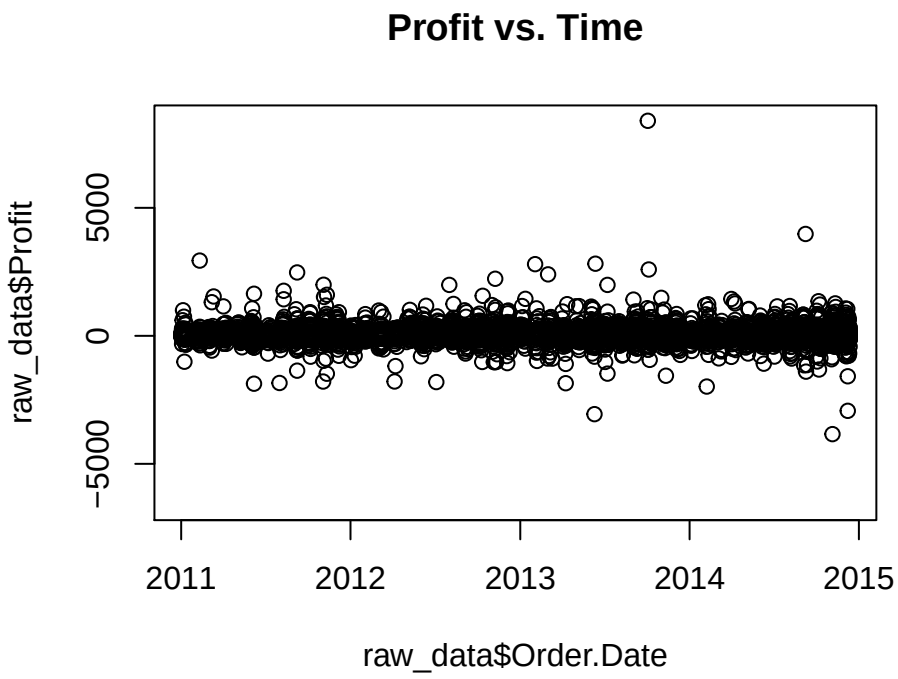
3.2 Numeric Data Exploration

In this section of the report, we will explore the numeric features to further our understanding of the chosen data.

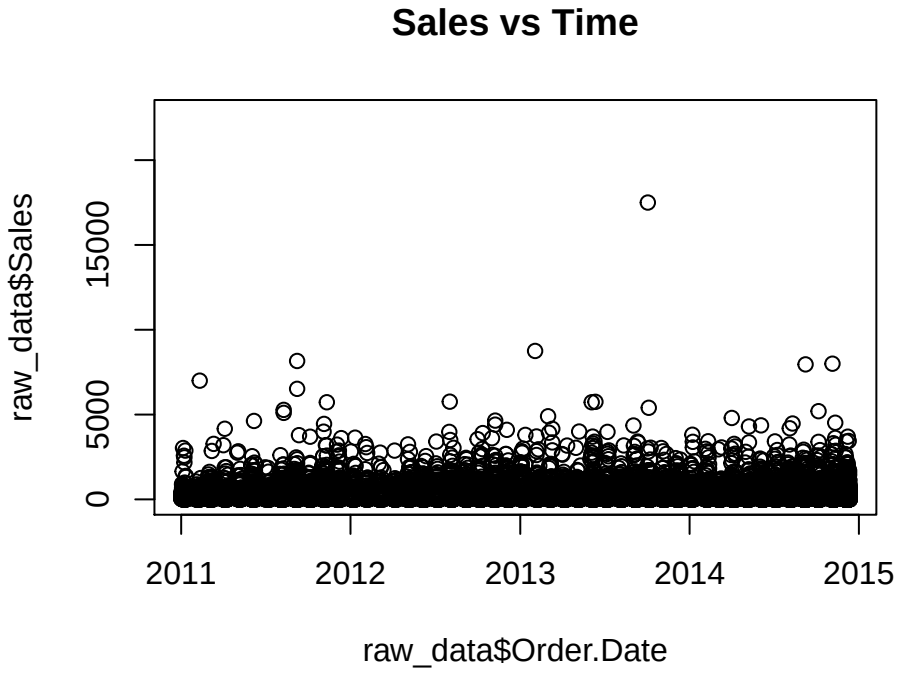
To begin the numeric data exploration, the distribution of each numeric feature was plotted and shown in the figure below



Then, we plotted the profits as a function of time in order to better understand how this feature varied throughout the years.

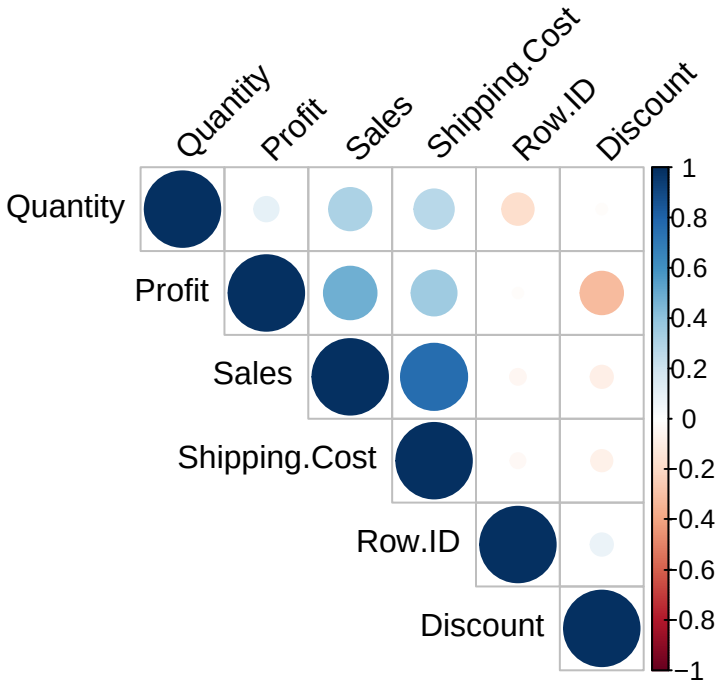


Similarly, the sales feature was plotted as a function of time.



Observing both figures, it can be seen that there are multiple outliers present in the dataset that will need to be addressed. This process will be discussed later in the report. Lastly, a correlation plot was produced to visualize any dependencies that could exist between the numeric features.

```
## corrplot 0.84 loaded
```



From the correlation plot, we can observe that certain variables are positively correlated. Sales and shipping cost as well as profit and sales seem to be the most heavily correlated variables, which intuitively makes sense.

4 Modelling

4.1 Unsupervised Modelling

4.1.1 Market Basket Analysis and Apriori

We will use the Market Basket Analysis and The Apriori Algorithm to see what kind of products that cutomers usually purchase together.

```
## [1] 51290    24
```

We will use the Association Rule Mining method to find frequent patterns in the transaction of 2011 to 2015. By knowing what items that customers frequently buy together, it will generate a set of rules. Store owner will use those rules for many marketing strategies:

- Change the store layout according to trends
- Customer behavior analysis
- Catalogs design
- Cross marketing on online stores
- Customer emails with add-on sales
- Consumer items-buys trending

First, let check how many unique Order.ID in the whole dataset. Order.ID was assigned to each transaction.

```
## [1] 25035
```

There are a total 25035 unique Order.ID. That means there might be more than 1 times of product in each transaction. We will find out the pattern what kind of products that customers usually purchase together.So that the store can rearrange the outlays of products, either put them together to maximize the sales of relevant products or put them far apart so that customer have chance to see other products.

Then, we will check how many kinds of products in total have been sold in the 4 year range.

```
## [1] 3788
```

There are 3788 kinds of products names. Remove "," in the column of products description for creating a transaction data.

```
## [1] "Tenex Lockers, Blue"
## [2] "Acme Trimmer, High Speed"
## [3] "Tenex Box, Single Width"
## [4] "Enermax Note Cards, Premium"
## [5] "Eldon Light Bulb, Duo Pack"
## [6] "Eaton Computer Printout Paper, 8.5 x 11"
```

4.1.2 Apply Market Basket Analysis Method

We use the Apriori "transaction" function to create a sparse matrix representing the all transactions and product name has been sold. Other attribute will not appear to the sparse matrix.

```
## Warning in asMethod(object): removing duplicated items in transactions
```

Let's have a general look and the transactions information.

```
## transactions as itemMatrix in sparse format with
##  25035 rows (elements/itemsets/transactions) and
##  3788 columns (items) and a density of 0.000540405
##
## most frequent items:
##                         Staples      Cardinal Index Tab  Clear
##                            222                             92
##  Eldon File Cart  Single Width Rogers File Cart  Single Width
##                              90                             84
##         Ibico Index Tab  Clear                         (Other)
##                              83                          50677
##
## element (itemset/transaction) length distribution:
## sizes
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 12264  6218  3214  1627   806   443   236    97    64    39    15     6     5
##    14
##     1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

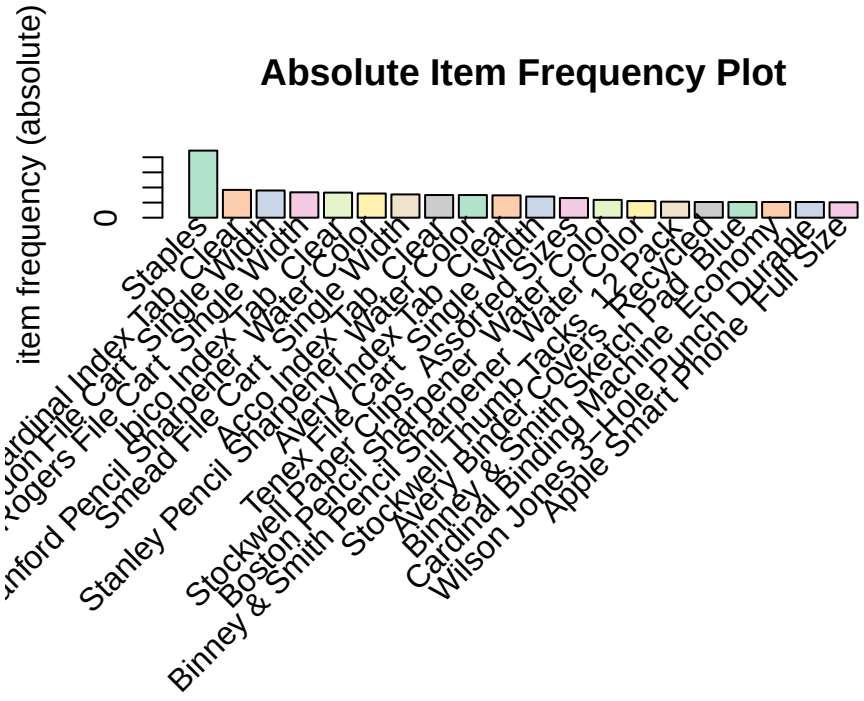
```
##   1.000   1.000   2.000   2.047   3.000  14.000
##
## includes extended item information - examples:
##                                                labels
## 1 "While you Were Out" Message Book  One Form per Page
## 2              #10- 4 1/8" x 9 1/2" Recycled Envelopes
## 3         #10- 4 1/8" x 9 1/2" Security-Tint Envelopes
##
## includes extended transaction information - examples:
##   transactionID
## 1  AE-2011-9160
## 2  AE-2013-1130
## 3  AE-2013-1530
```

There are 25035 transactions (row) and 3788 items (columns). The rows of transactions reflect the total number of Order.ID; the columns of items reflect the total Products.Name in the raw dataset. Density is 0.000540405. Density tells us the percentage of non-zero cells in a sparse matrix. We can calculate how many items were purchased by using the density.

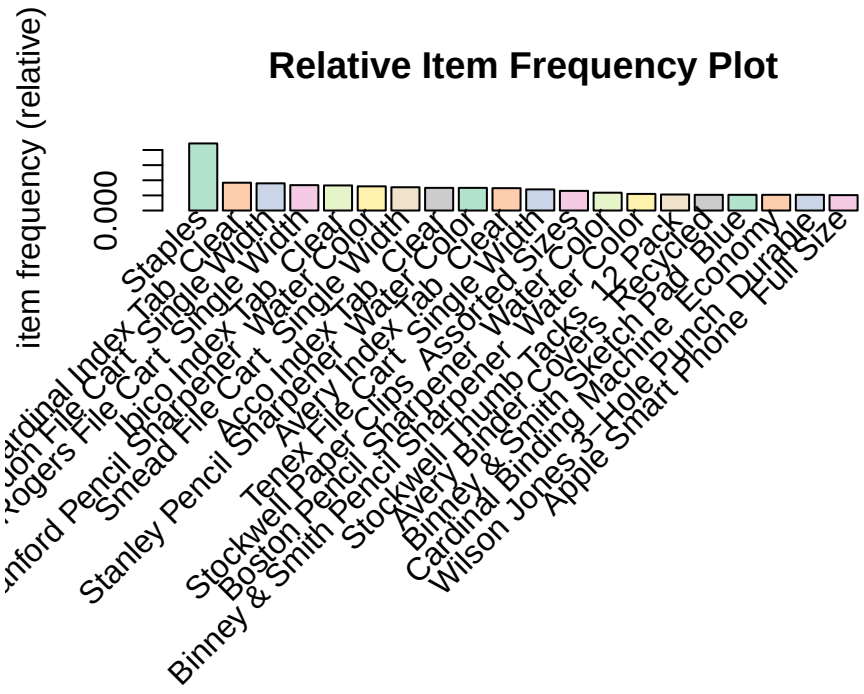
```
## [1] 51248
```

4.1.3 Visualization of the MBA.

Using "itemFrequencyPlot" to visualize the distribution of frequency of items that purchase. It can be evaluate base on absolute numbers of relative proportion.



Absolute will plot numeric frequencies of each item independently.



Relative plot will show how many times these items have appeared as compared to others.

4.1.4 Generating Rules

Using Apriori algorithm to generate association rules.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   7e-05      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[3788 item(s), 25035 transaction(s)] done [0.03s].
## sorting and recoding items ... [3690 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.06s].
## writing ... [1625 rule(s)] done [0.02s].
## creating S4 object  ... done [0.00s].
```

There are total 1625 rules generated. Let's select the top 10 rules to have a look.

```
##      lhs                                              rhs
## [1]  {Ativa V4110MDD Micro-Cut Shredder}         => {Staples}                                7.988
## [2]  {Bevis Conference Table  Fully Assembled,
##       Hon Conference Table  with Bottom Storage} => {Sharp Personal Copier  Laser}           7.988
## [3]  {Hon Conference Table  with Bottom Storage,
##       Sharp Personal Copier  Laser}              => {Bevis Conference Table  Fully Assembled} 7.988
## [4]  {Bevis Conference Table  Fully Assembled,
##       Sharp Personal Copier  Laser}              => {Hon Conference Table  with Bottom Storage} 7.988
## [5]  {Bevis Conference Table  Fully Assembled,
##       Hon Conference Table  with Bottom Storage} => {Jiffy Mailers  Set of 50}               7.988
## [6]  {Hon Conference Table  with Bottom Storage,
##       Jiffy Mailers  Set of 50}                  => {Bevis Conference Table  Fully Assembled} 7.988
## [7]  {Bevis Conference Table  Fully Assembled,
##       Jiffy Mailers  Set of 50}                  => {Hon Conference Table  with Bottom Storage} 7.988
## [8]  {Bevis Conference Table  Fully Assembled,
##       Hon Conference Table  with Bottom Storage} => {Motorola Audio Dock  with Caller ID}      7.988
## [9]  {Hon Conference Table  with Bottom Storage,
##       Motorola Audio Dock  with Caller ID}       => {Bevis Conference Table  Fully Assembled} 7.988
## [10] {Bevis Conference Table  Fully Assembled,
##       Motorola Audio Dock  with Caller ID}       => {Hon Conference Table  with Bottom Storage} 7.988
```

Explanation of the rules:

For example the top one rule explains that 79.88% transaction show “Ativa V4110MDD Micro-Cut Shredder” is bought with purchase of “Staples”; 100% of customers who purchase “Ativa V4110MDD Micro-cut Shredder” also bought “Staples”.

Removing redundant rules

```
## [1] 1622
```

There are 1622 which are subset rules. There are only 3 main rules.

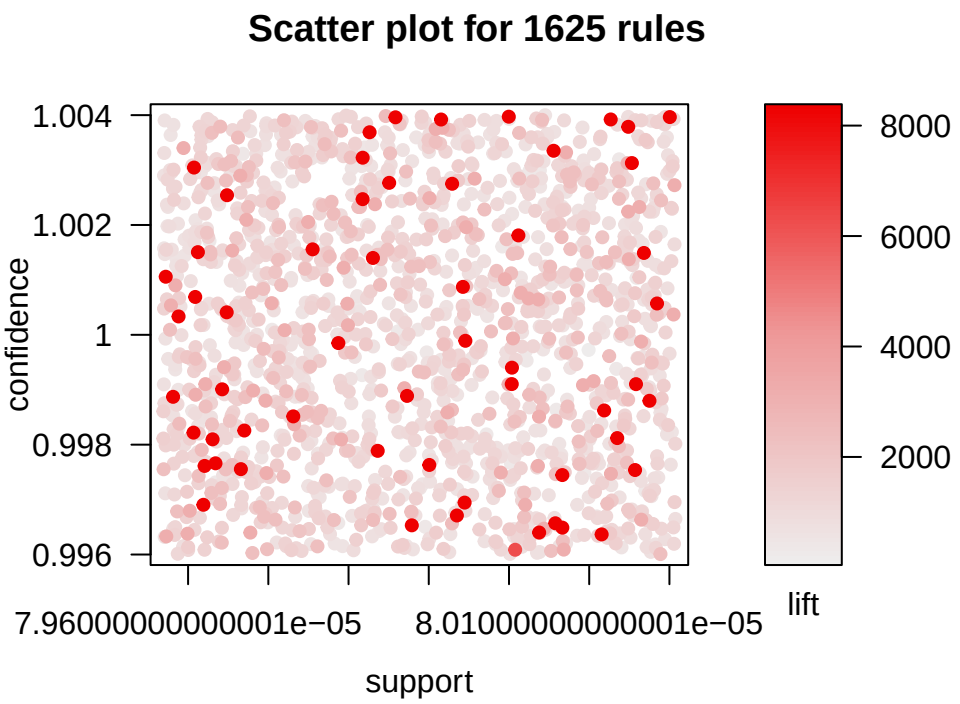
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   5e-05      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[3788 item(s), 25035 transaction(s)] done [0.03s].
## sorting and recoding items ... [3690 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.06s].
## writing ... [1 rule(s)] done [0.04s].
## creating S4 object  ... done [0.00s].
```

Check rules by giving a item, for example “staples.rules”

```
##     lhs                                       rhs       support      confidence
## [1] {Ativa V4110MDD Micro-Cut Shredder} => {Staples} 7.988816e-05 1
##     coverage     lift     count
## [1] 7.988816e-05 112.7703 2
```

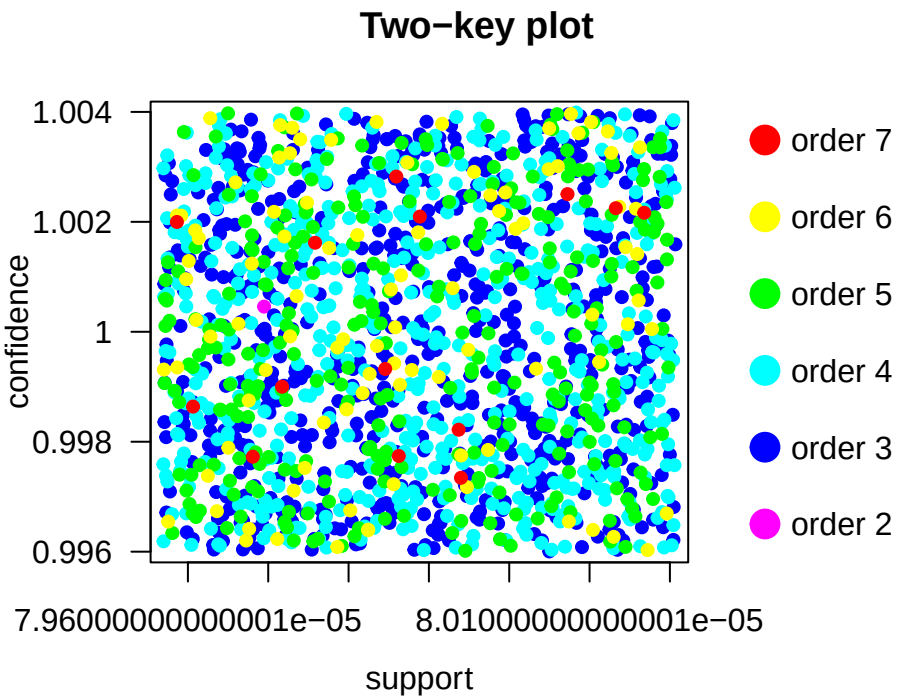
4.1.5 Visualization of Association Rules

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```

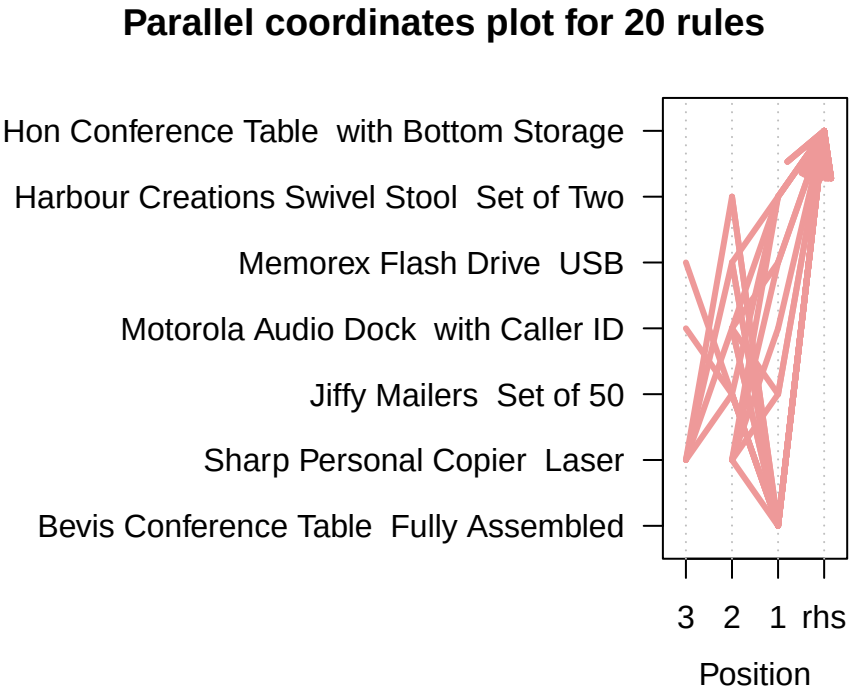


The above plot shows that rules with higher lift have a little less support. But in general it was evenly spread.

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Above two-key plot shows support and confidence respectively. The order shows how many items in the relevant rule. Use Parallel Coordinates Plot to check individual rule representation. The plot will explain which products along with which items cause what kind of sales.



The above plot shows that if a customer bought “Memorex Flash Drive USB” and “Motorola Audio Dock

with Caller ID", the customer is likely to buy "Jiffy Mailers Set of 50".

4.2 Supervised Modelling

4.2.1 Linear Regression Model

This practice is to see if we can predict the sales based on some certain cities, customer information, Month, and year. To predict the sales value, Linear Regression is chosen. The data exploration and data pre-processing occur are done at the same time.

To narrow down our target, the target becomes the United States. So we extracted the united states. In addition, for this trial, we disregarded some actual numeric values, such as profit, quantity, shipping.cost, and so on.

we are not considering the day, but the month and year, so we extracted the month and year from the date data.

```
##
## Attaching package: 'lubridate'

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

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

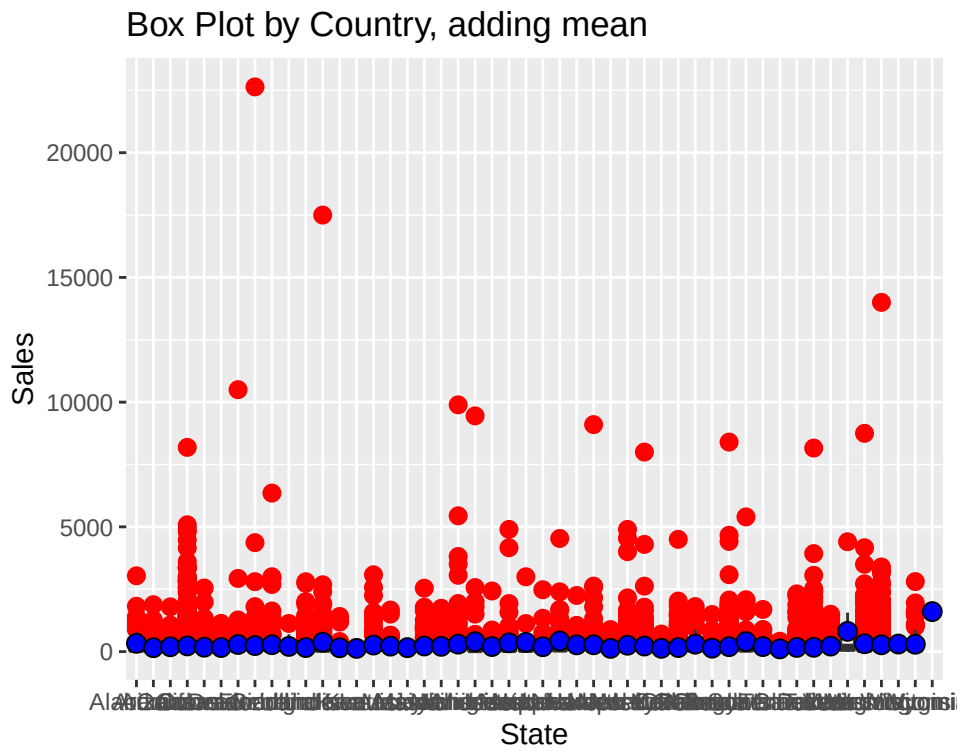
```
##        Ship.Mode Customer.ID     Segment          City      State       Country
## 1    First Class    LC-17050    Consumer Mission Viejo California United States
## 2 Standard Class    VF-21715 Home Office      Elmhurst   Illinois United States
## 3   Second Class    DB-13060    Consumer       Seattle Washington United States
## 4 Standard Class    GW-14605    Consumer       Houston      Texas United States
## 5 Standard Class    SC-20380    Consumer       El Paso      Texas United States
## 6 Standard Class    GW-14605    Consumer       Houston      Texas United States
##   Market  Region        Category Sub.Category   Sales Order.Priority Month Year
## 1     US    West       Furniture    Bookcases 290.666           High     2 2011
## 2     US Central       Furniture       Chairs 634.116           High     3 2011
## 3     US    West       Furniture       Chairs 457.568         Medium     3 2011
## 4     US Central       Furniture       Tables 376.509         Medium     3 2011
## 5     US Central       Furniture       Chairs 362.250         Medium     3 2011
## 6     US Central Office Supplies      Storage 137.352         Medium     3 2011
```

```
## 'data.frame':    8933 obs. of  14 variables:
##  $ Ship.Mode     : Factor w/ 4 levels "First Class",..: 1 4 3 4 4 4 4 4 4 4 ...
##  $ Customer.ID   : chr  "LC-17050" "VF-21715" "DB-13060" "GW-14605" ...
##  $ Segment       : Factor w/ 3 levels "Consumer","Corporate",..: 1 3 1 1 1 1 1 1 1 3 ...
##  $ City          : chr  "Mission Viejo" "Elmhurst" "Seattle" "Houston" ...
##  $ State         : chr  "California" "Illinois" "Washington" "Texas" ...
##  $ Country       : Factor w/ 1 level "United States": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Market        : Factor w/ 1 level "US": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Region        : Factor w/ 4 levels "Central","East",..: 4 1 4 1 1 1 1 1 1 1 ...
##  $ Category      : chr  "Furniture" "Furniture" "Furniture" "Furniture" ...
```

```
##  $ Sub.Category  : chr  "Bookcases" "Chairs" "Chairs" "Tables" ...
##  $ Sales         : num  291 634 458 377 362 ...
##  $ Order.Priority: Factor w/ 4 levels "Critical","High",..: 2 2 4 4 4 4 4 4 4 2 ...
##  $ Month         : num  2 3 3 3 3 3 3 3 3 3 ...
##  $ Year          : num  2011 2011 2011 2011 2011 ...
```

Check the outliers of the numeric data

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



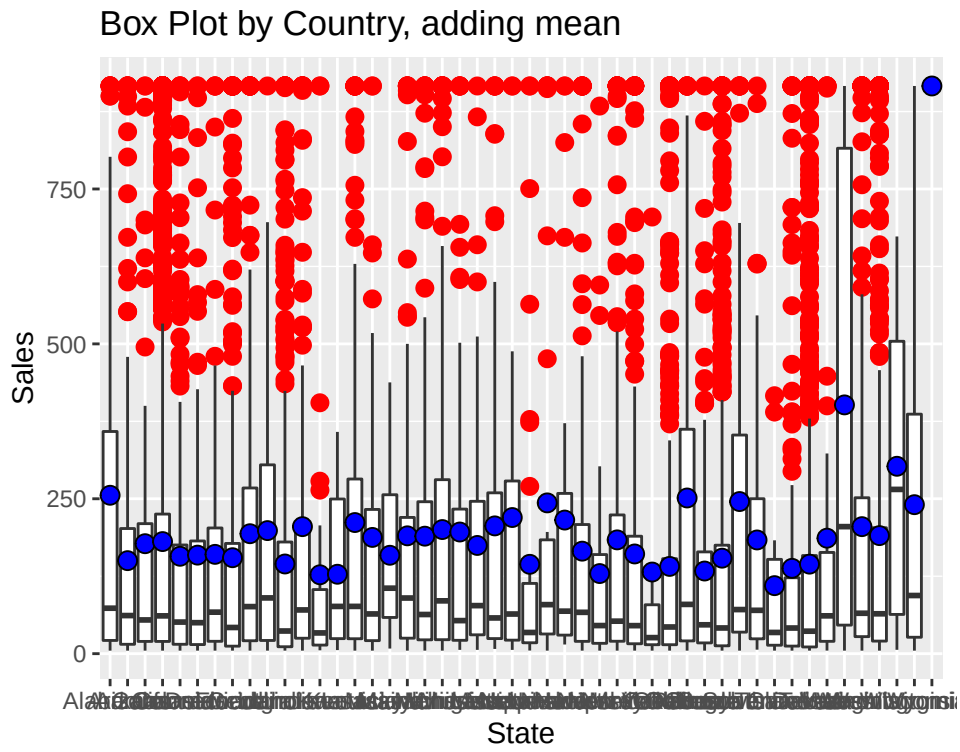

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     0.444    16.900    53.720   225.373   208.160 22638.480
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   9.000   7.784  11.000  12.000
```

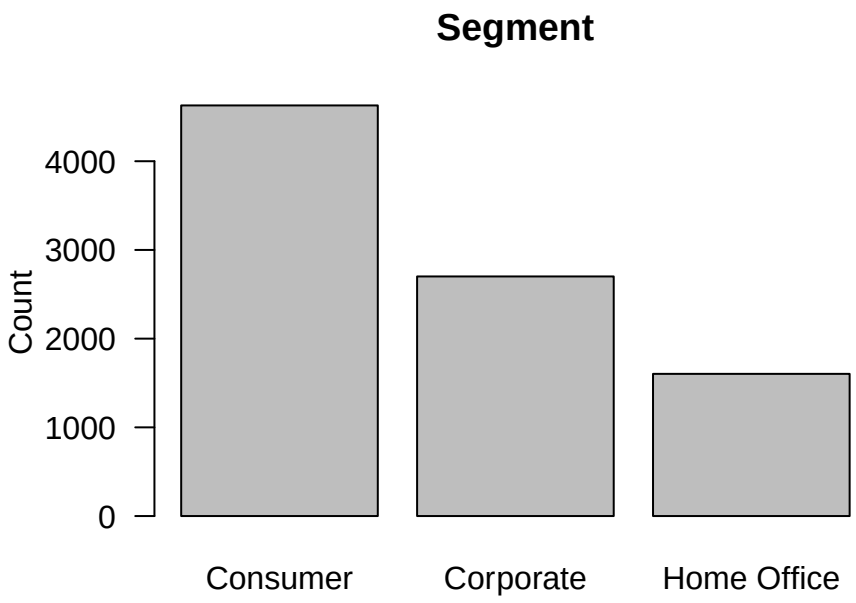
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2011    2012    2013    2013    2014    2014
```

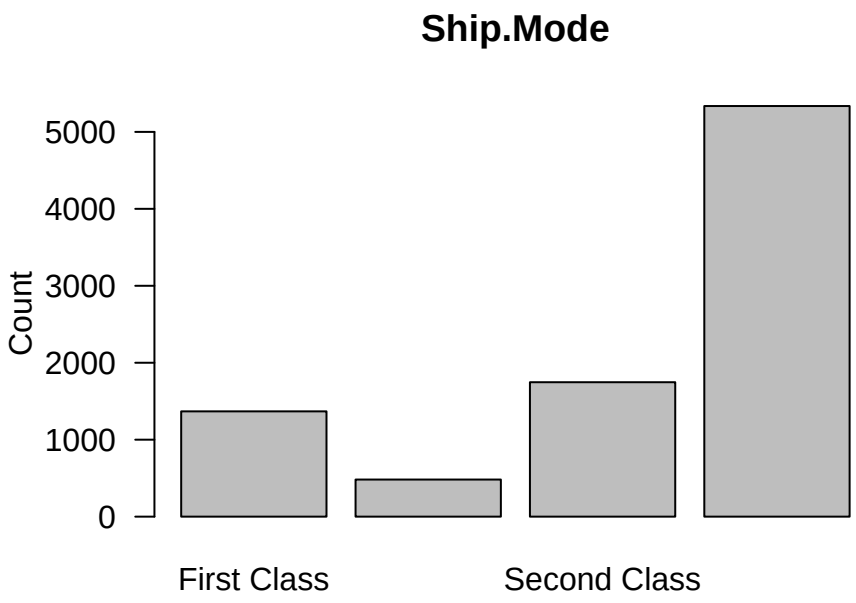
Outlier treated

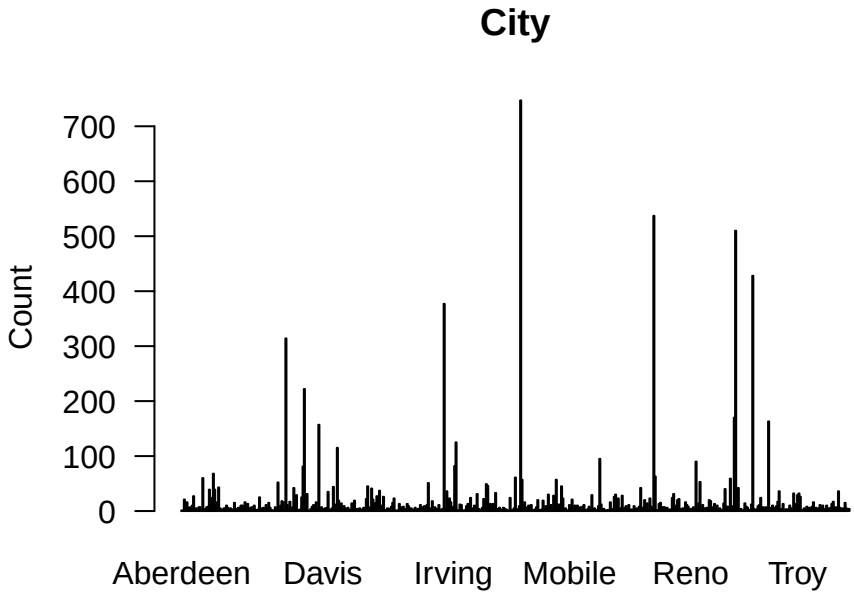
```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

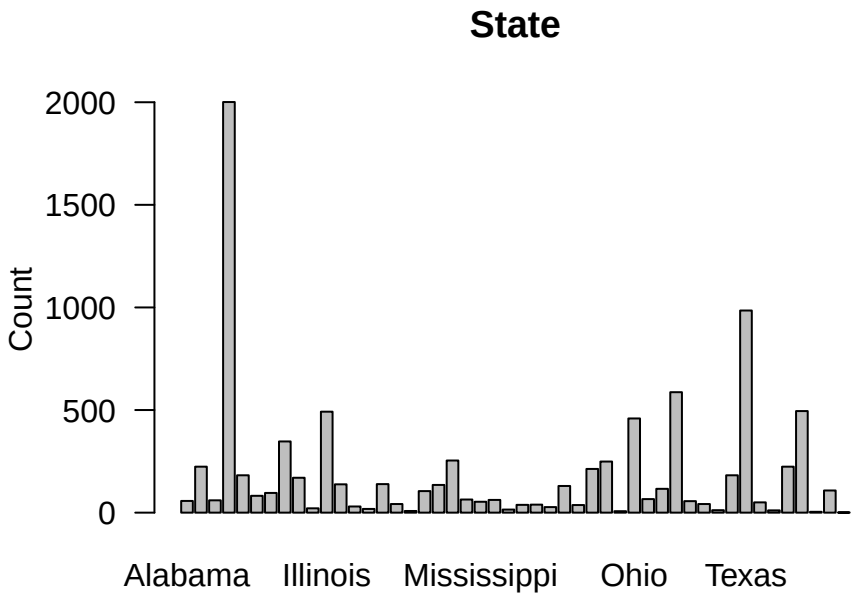


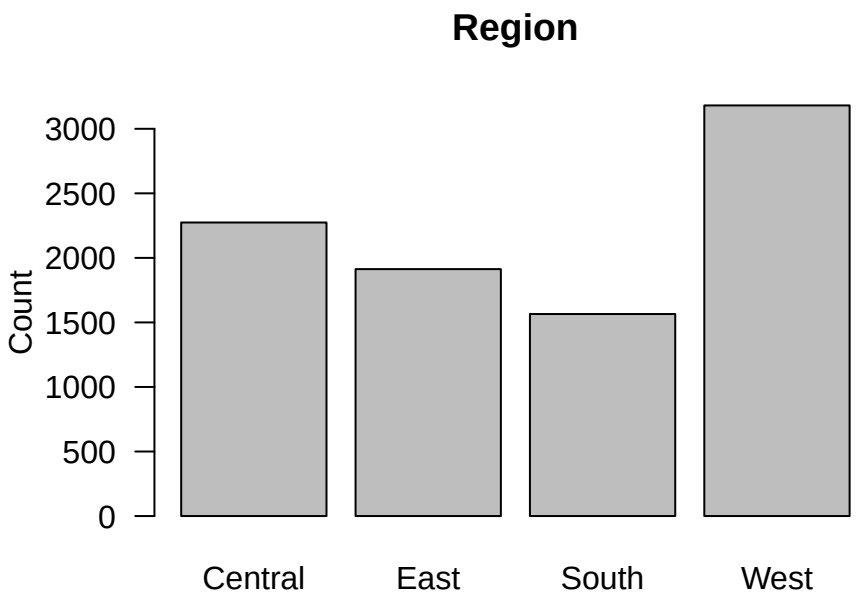

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.853  16.900  53.720 171.713 208.160 916.251

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   9.000   7.824  11.000  12.000

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2011    2012    2013    2013    2014    2014
```


Segment
Count
4000
3000
2000
1000
0
Consumer
Corporate
Home Office

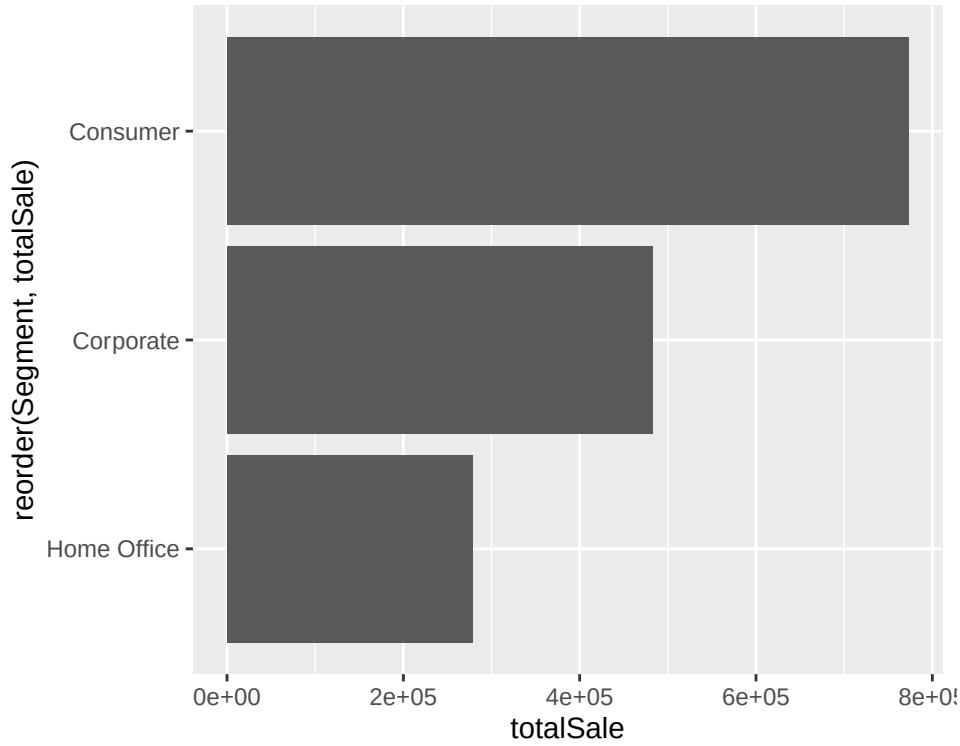

Ship.Mode
Count
5000
4000
3000
2000
1000
0
First Class
Second Class


City
Count
700
600
500
400
300
200
100
0
Aberdeen
Davis
Irving
Mobile
Reno
Troy


State
Count
2000
1500
1000
500
0
Alabama
Illinois
Mississippi
Ohio
Texas

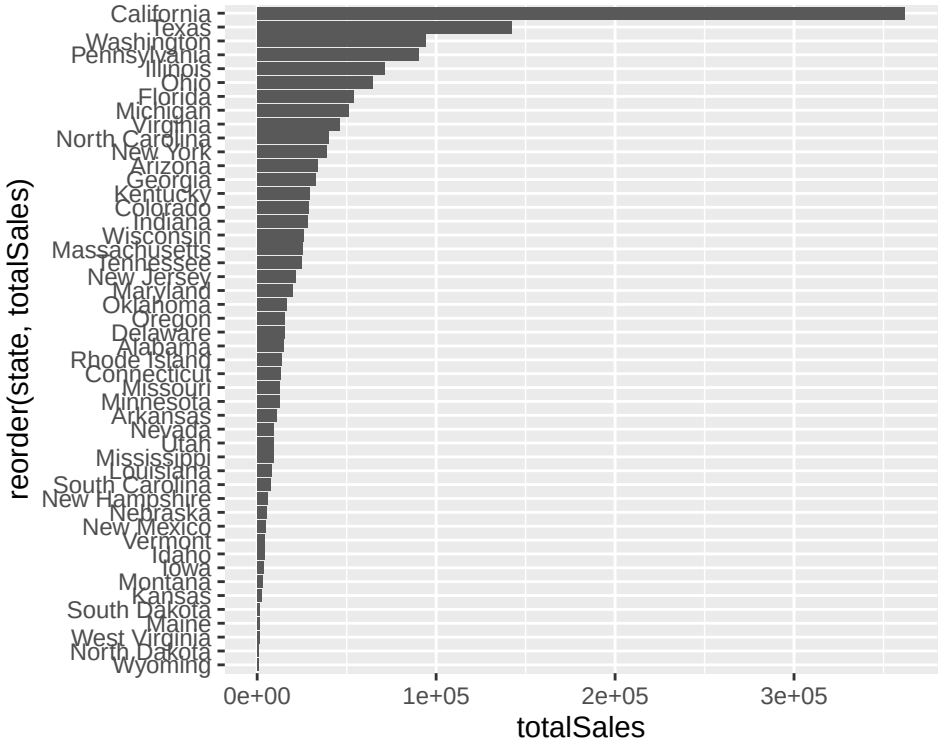


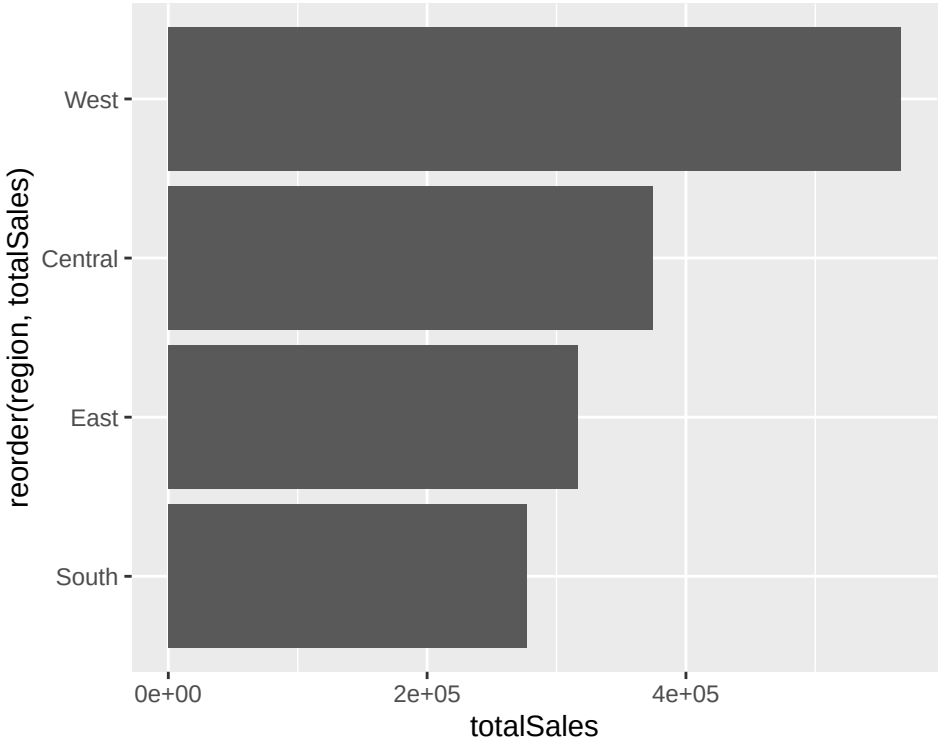

```
## Warning: package 'treemap' was built under R version 4.0.3
```

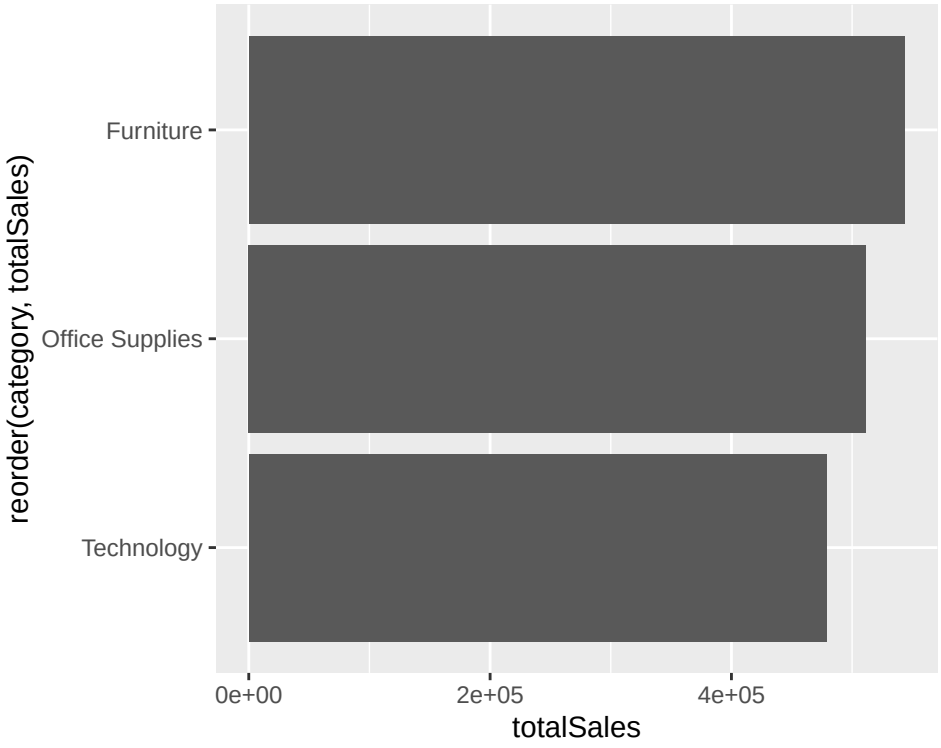


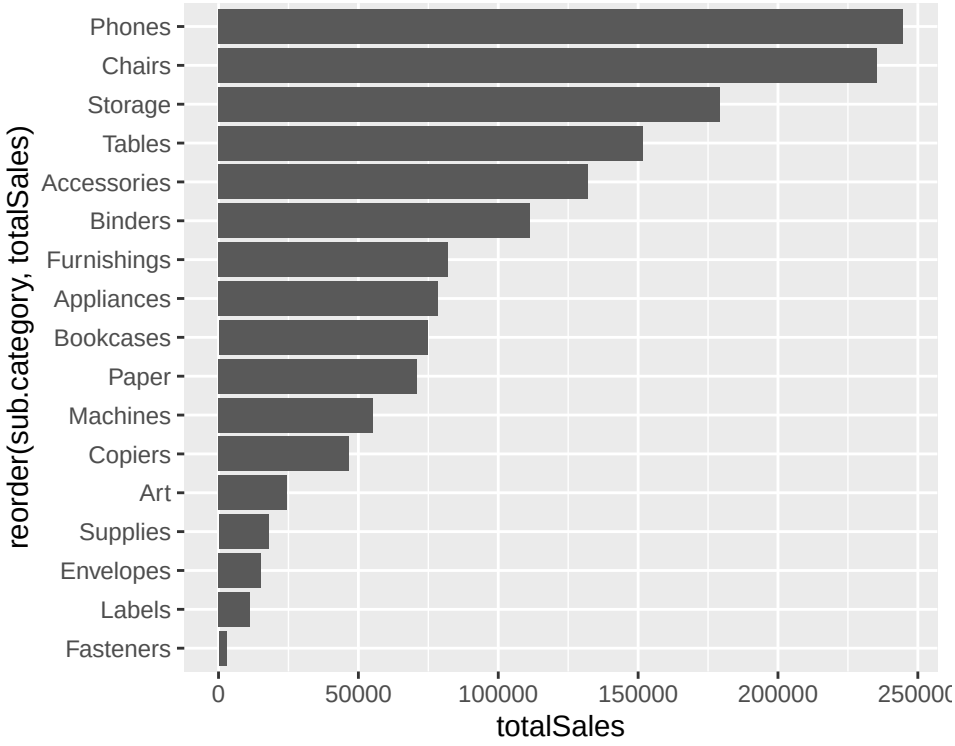

```
## The following objects are masked from ssds (pos = 3):
```

```
##
##     Category, City, Country, Customer.ID, Market, Month,
##     Order.Priority, Region, Sales, Segment, Ship.Mode, State,
##     Sub.Category, Year
```



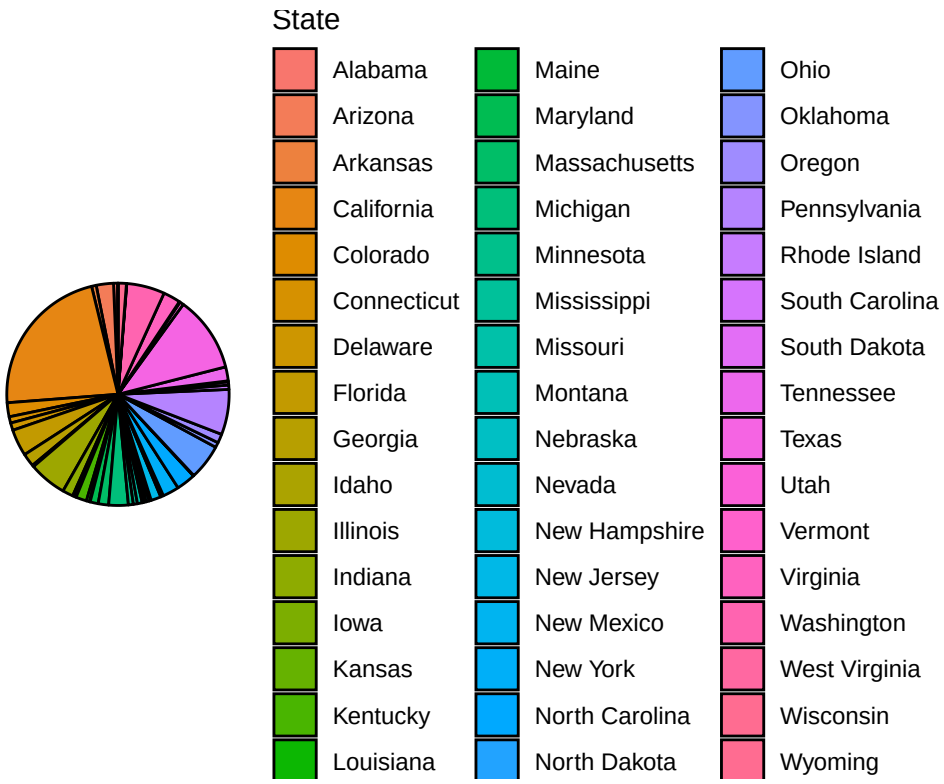







```
##
##     California          Texas   Pennsylvania     Washington       Illinois
##           2001            985            587            495            492
##           Ohio        Florida       Michigan North Carolina        Arizona
```

```
##          459          347          254          249          224
##     Virginia     New York     Colorado    Tennessee      Georgia
##          224          213          182          182          170
##     Kentucky      Indiana Massachusetts   New Jersey       Oregon
##          139          138          135          130          116
##    Wisconsin     Maryland     Delaware  Connecticut     Oklahoma
##          108          105           96           82           66
##    Minnesota     Missouri     Arkansas      Alabama Rhode Island
##           64           62           60           57           56
##  Mississippi         Utah    Louisiana South Carolina       Nevada
##           53           50           42           42           39
##     Nebraska   New Mexico         Iowa New Hampshire        Idaho
##           38           37           30           27           21
##       Kansas      Montana South Dakota      Vermont        Maine
##           18           15           12           11            8
## North Dakota West Virginia      Wyoming
##            7            4            1
```




```
##
##    Los Angeles   Philadelphia  San Francisco        Seattle
##            747            537            510            428
##        Houston        Chicago       Columbus      San Diego
##            377            314            222            170
##    Springfield         Dallas   Jacksonville        Detroit
##            163            157            125            115
##         Newark       Richmond        Jackson       Columbia
##             95             90             82             81
##         Aurora        Phoenix    Long Beach      Arlington
##             68             63             61             60
```

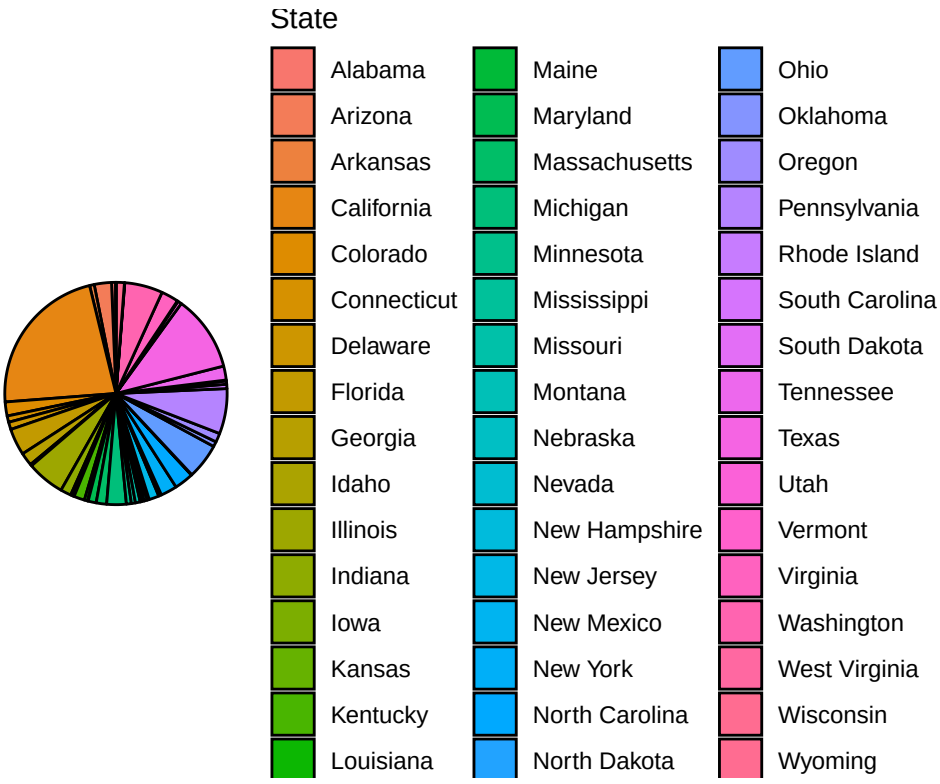
```
##       San Antonio        Louisville             Miami         Rochester
##                59                57                57                53
##         Charlotte         Henderson          Lakewood         Lancaster
##                52                51                49                46
##         Fairfield         Milwaukee            Denver         Baltimore
##                45                45                44                43
##         Cleveland          Pasadena          San Jose      Fayetteville
##                42                42                42                41
##             Salem           Atlanta            Austin          Franklin
##                40                39                39                37
##        Huntsville             Tampa        Wilmington           Decatur
##                36                36                36                35
##          Lawrence            Toledo            Tucson           Concord
##                33                32                32                31
##         Lafayette        Providence           Memphis         Oceanside
##                31                31                30                30
##           Clinton         Nashville              Troy              Mesa
##                29                29                29                28
##             Omaha          Anaheim        Fort Worth            Fresno
##                28                27                27                26
##           Oakland             Tulsa        Burlington  Colorado Springs
##                26                26                25                25
##            Auburn         Knoxville       Little Rock          Portland
##                24                24                24                24
##            Smyrna          Glendale      Indianapolis       Minneapolis
##                24                23                23                23
##     Oklahoma City            Peoria           Everett           Lakeland
##                23                23                22                22
##           Raleigh             Akron          Florence            Monroe
##                22                21                21                21
##            Marion          Paterson            Quincy         Roseville
##                20                20                20                20
##             Dover           El Paso           Mcallen        Chesapeake
##                19                19                19                18
##           Hialeah           Roswell        Cincinnati       Tallahassee
##                18                18                17                17
##       Westminster        Alexandria       Bakersfield         Brentwood
##                17                16                16                16
##           Chester          Cranston         Inglewood            Lowell
##                16                16                16                16
##   North Las Vegas          Redlands    Virginia Beach       Bloomington
##                16                16                16                15
##        Carrollton   Fort Lauderdale           Gilbert         Lakeville
##                15                15                15                15
##             Plano           Yonkers       Albuquerque            Dublin
##                15                15                14                14
##           Edmonds     Pembroke Pines         Riverside  Saint Petersburg
##                14                14                14                14
##         Santa Ana           Bristol     Grand Prairie        Greensboro
##                14                13                13                13
##              Kent            Laredo         Las Vegas        Plainfield
##                13                13                13                13
##        Sacramento             Tempe           Trenton          Westland
##                13                13                13                13
```

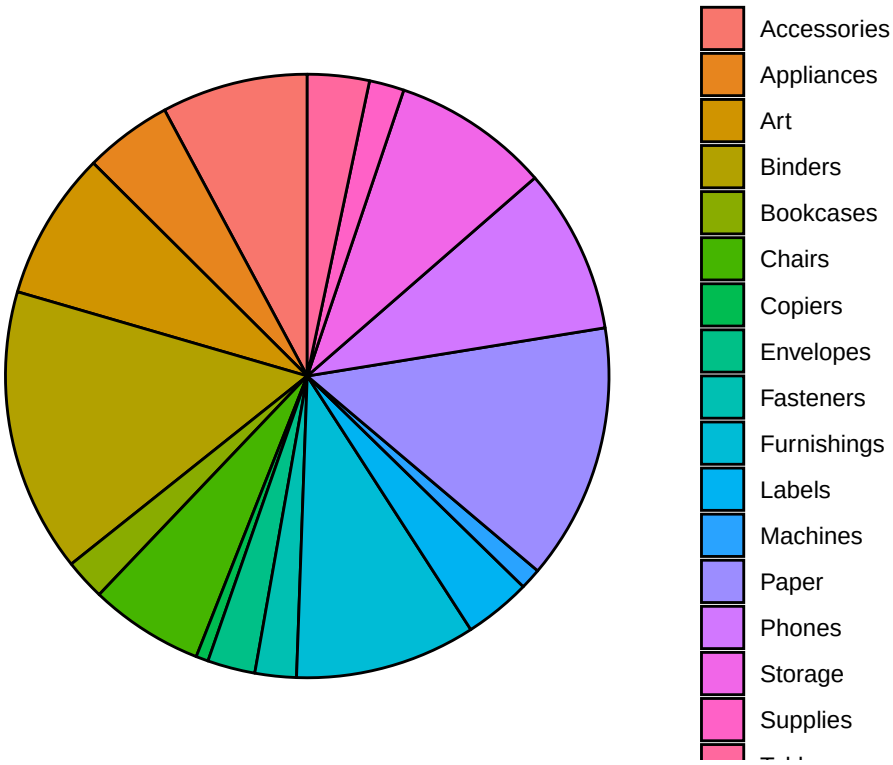
```
##        Des Moines      Johnson City           Midland      Newport News
##                12                12                12                12
##        Scottsdale          Carlsbad             Chico       Costa Mesa
##                12                11                11                11
##           Hampton         Hempstead         Hollywood         Jonesboro
##                11                11                11                11
##        Manchester           Meriden            Mobile      Murfreesboro
##                11                11                11                11
##           Orlando           Redmond          Rockford           Skokie
##                11                11                11                11
##         Waterbury           Amarillo       Belleville     Bowling Green
##                11                10                10                10
##     Boynton Beach           Buffalo       Chattanooga          Freeport
##                10                10                10                10
##          La Porte           Madison        Montgomery     Moreno Valley
##                10                10                10                10
##             Provo     Saint Charles          Suffolk          Thornton
##                10                10                10                10
##         Watertown        Waynesboro           Durham        Greenville
##                10                10                 9                 9
##       Hattiesburg           Kenosha           Lorain            Medina
##                 9                 9                 9                 9
##        Middletown      New Rochelle             Orem            Oxnard
##                 9                 9                 9                 9
##    San Bernardino       Sioux Falls         Superior            Athens
##                 9                 9                 9                 8
##    Corpus Christi           Fremont       Georgetown       Great Falls
##                 8                 8                 8                 8
##             Irving           Lansing      Mount Vernon        Naperville
##                 8                 8                 8                 8
##             Parker       Perth Amboy           Pueblo     Redondo Beach
##                 8                 8                 8                 8
##            Salinas    Salt Lake City     Santa Barbara             Utica
##                 8                 8                 8                 8
##           Vineland         Allentown            Apopka      Apple Valley
##                 8                 7                 7                 7
##         Asheville       Brownsville          Chandler       Clarksville
##                 7                 7                 7                 7
##          Danville             Fargo      Fort Collins      Grand Rapids
##                 7                 7                 7                 7
##      Harrisonburg     Highland Park          Marietta        Morristown
##                 7                 7                 7                 7
##    Pleasant Grove         Pocatello           Pomona            Revere
##                 7                 7                 7                 7
##        Round Rock         Southaven           Spokane         Wheeling
##                 7                 7                 7                 7
##           Wichita          Avondale          Bellevue       Bolingbrook
##                 7                 6                 6                 6
##      Bossier City             Bryan         Cambridge              Cary
##                 6                 6                 6                 6
##  Dearborn Heights       Des Plaines         Eau Claire           Eugene
##                 6                 6                 6                 6
##        Hackensack  Huntington Beach       Lake Forest       League City
##                 6                 6                 6                 6
```

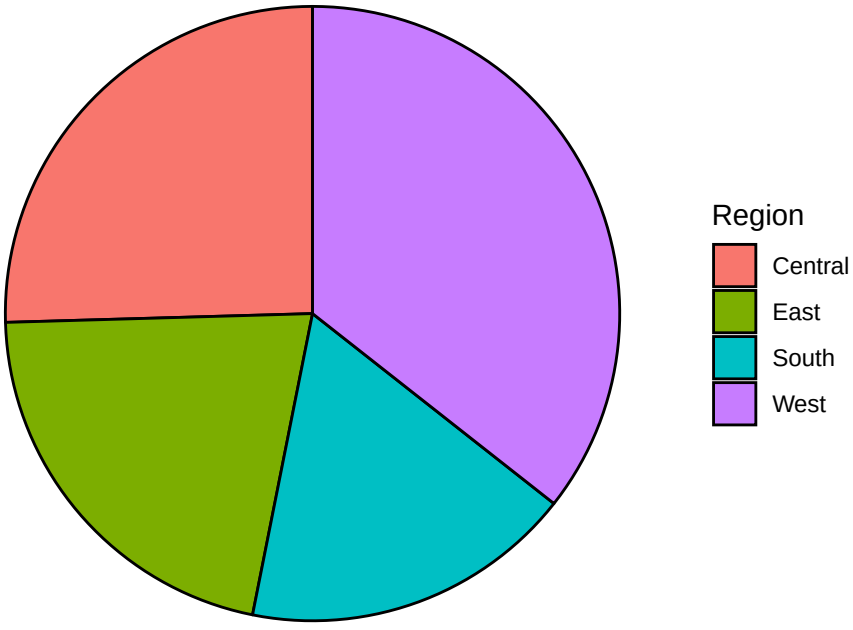
```
##        Leominster          Lubbock            Macon           Mentor
##                 6                6                6                6
##           Milford           Orange            Pasco          Passaic
##                 6                6                6                6
##       Port Arthur             Rome          Saginaw      Santa Clara
##                 6                6                6                6
##        South Bend          Stockton        Sunnyvale          Vallejo
##                 6                6                6                6
##              Waco        Westfield           Wilson        Ann Arbor
##                 6                6                6                5
##            Bangor          Beaumont          Bedford       Bethlehem
##                 5                5                5                5
##      Broken Arrow       Broomfield        Camarillo     Carol Stream
##                 5                5                5                5
##          Dearborn           Deltona       East Orange        Encinitas
##                 5                5                5                5
##       Garden City          Gresham         Harlingen    Hendersonville
##                 5                5                5                5
##         La Crosse            Logan          Mesquite     Mission Viejo
##                 5                5                5                5
##       New Bedford           Odessa           Olathe          Olympia
##                 5                5                5                5
##        Palm Coast          Reading         Rockville      Saint Louis
##                 5                5                5                5
##     Thousand Oaks            Tyler        Vancouver          Warwick
##                 5                5                5                5
##       West Jordan        Woodstock             York            Allen
##                 5                5                5                4
##           Andover           Arvada            Boise          Coppell
##                 4                4                4                4
##     Daytona Beach         Elmhurst         Evanston          Garland
##                 4                4                4                4
##          Gastonia        Green Bay        Greenwood       Haltom City
##                 4                4                4                4
##          Hamilton         Hesperia        Hillsboro       Hot Springs
##                 4                4                4                4
##     Laguna Niguel           Lawton         Lewiston          Loveland
##                 4                4                4                4
##           Medford          Meridian       Morgan Hill          Muskogee
##                 4                4                4                4
##        New Albany       Noblesville          Norwich            Pharr
##                 4                4                4                4
##  Rancho Cucamonga             Reno        San Angelo        San Marcos
##                 4                4                4                4
##         Sheboygan         Temecula         Torrance          Visalia
##                 4                4                4                4
##            Wausau       Woonsocket             Yuma          Bayonne
##                 4                4                4                3
##        Bellingham          Beverly        Boca Raton          Burbank
##                 3                3                3                3
##          Caldwell           Canton       Chula Vista   College Station
##                 3                3                3                3
##     Cuyahoga Falls      Delray Beach           Draper          Dubuque
##                 3                3                3                3
```

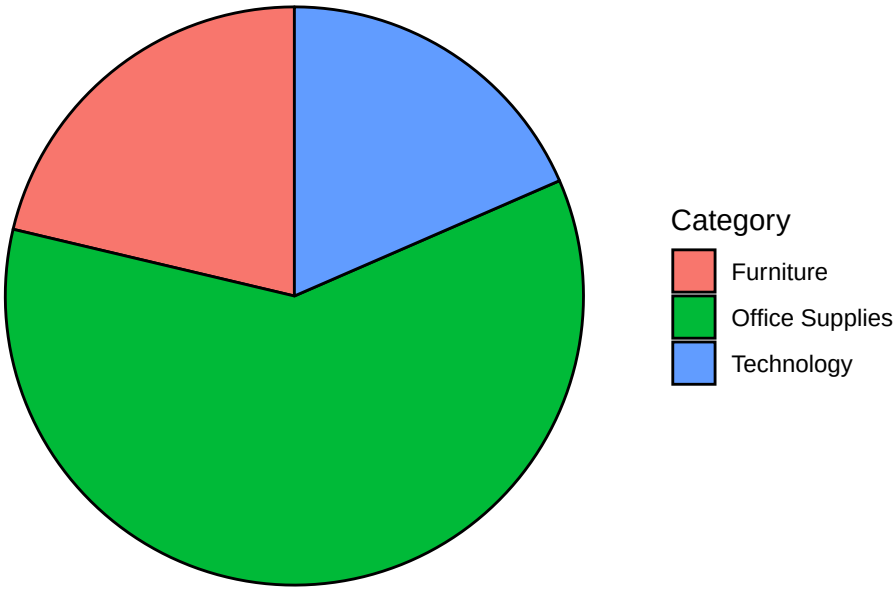
```
##        East Point          Edinburg         Escondido        Farmington
##                 3                 3                 3                 3
##          Gulfport           Helena           Holland         Homestead
##                 3                 3                 3                 3
##      Lake Charles       Las Cruces           Lebanon      Lincoln Park
##                 3                 3                 3                 3
##          Longmont         Longview            Malden           Manteca
##                 3                 3                 3                 3
##             Mason          Modesto     Niagara Falls          Oak Park
##                 3                 3                 3                 3
##            Oswego       Park Ridge          Pearland     Pompano Beach
##                 3                 3                 3                 3
##  Port Saint Lucie           Renton       Saint Cloud       San Gabriel
##                 3                 3                 3                 3
##      Sierra Vista           Sparks  Sterling Heights            Taylor
##                 3                 3                 3                 3
##        Texas City       The Colony       Thomasville         Urbandale
##                 3                 3                 3                 3
##   West Palm Beach         Woodland           Altoona          Appleton
##                 3                 3                 2                 2
##           Bozeman        Bridgeton     Buffalo Grove     Bullhead City
##                 2                 2                 2                 2
##        Cedar Hill   Charlottesville           Clifton            Clovis
##                 2                 2                 2                 2
##         Coachella     Cottage Grove            Edmond          El Cajon
##                 2                 2                 2                 2
##           Elkhart         Englewood         Frankfort            Frisco
##                 2                 2                 2                 2
##      Gaithersburg         Grapevine           Greeley        Grove City
##                 2                 2                 2                 2
##           Hickory      Independence         Jamestown            Kenner
##                 2                 2                 2                 2
##           La Mesa            Laurel              Lehi              Lodi
##                 2                 2                 2                 2
##         Mansfield       Marlborough        Marysville         Mishawaka
##                 2                 2                 2                 2
##          Moorhead    Mount Pleasant            Murray            Nashua
##                 2                 2                 2                 2
##     New Brunswick        New Castle            Norman  North Charleston
##                 2                 2                 2                 2
##         Owensboro        Pine Bluff        Rapid City        Richardson
##                 2                 2                 2                 2
##        Rio Rancho         Royal Oak        Saint Paul      San Clemente
##                 2                 2                 2                 2
##           Sanford          Santa Fe           Shelton       Summerville
##                 2                 2                 2                 2
##         Texarkana        Tuscaloosa        Twin Falls     Warner Robins
##                 2                 2                 2                 2
##          Aberdeen           Abilene           Antioch Arlington Heights
##                 1                 1                 1                 1
##     Atlantic City           Baytown          Billings      Cedar Rapids
##                 1                 1                 1                 1
##         Champaign       Chapel Hill          Cheyenne    Citrus Heights
##                 1                 1                 1                 1
```

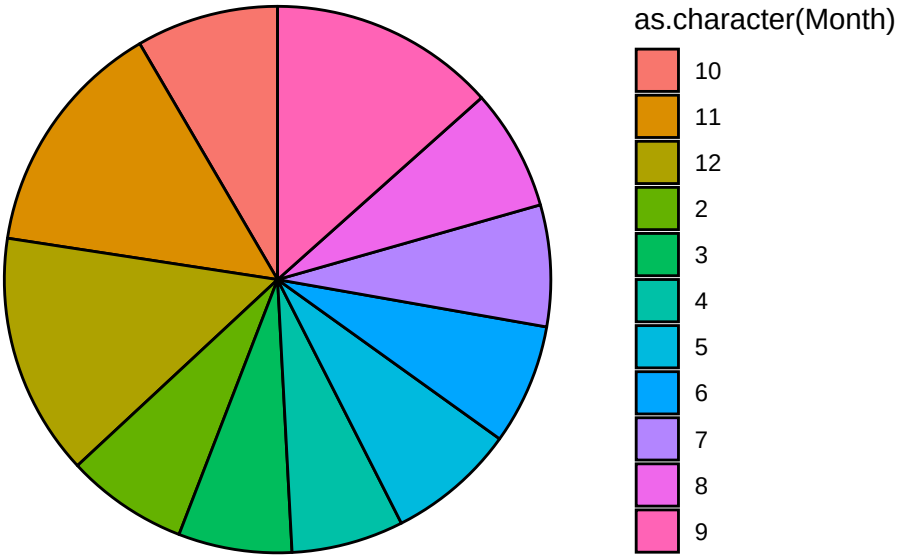
```
##     Commerce City            Conroe           Conway           Danbury
##                 1                 1                1                 1
##             Davis        Deer Park           Elyria          Glenview
##                 1                 1                1                 1
##         Goldsboro      Grand Island       Hagerstown          Holyoke
##                 1                 1                1                 1
##         Iowa City    Jefferson City          Jupiter           Keller
##                 1                 1                1                 1
##         Kissimmee         La Quinta     Lake Elsinore           Layton
##                 1                 1                1                 1
##            Linden       Lindenhurst        Littleton         Manhattan
##                 1                 1                1                 1
##         Melbourne           Missoula     Missouri City        Montebello
##                 1                 1                1                 1
##          Murrieta           Norfolk           Normal          Ontario
##                 1                 1                1                 1
##       Orland Park      Ormond Beach         Palatine         Pensacola
##                 1                 1                1                 1
##       Pico Rivera       Port Orange         Portage          Redding
##                 1                 1                1                 1
##      Redwood City         Rock Hill           Rogers        Romeoville
##                 1                 1                1                 1
##      Saint Peters   San Luis Obispo        San Mateo       Santa Maria
##                 1                 1                1                 1
##        Springdale       Tinley Park        Vacaville         Waterloo
##                 1                 1                1                 1
##          Waukesha          Whittier          Yucaipa
##                 1                 1                1
```

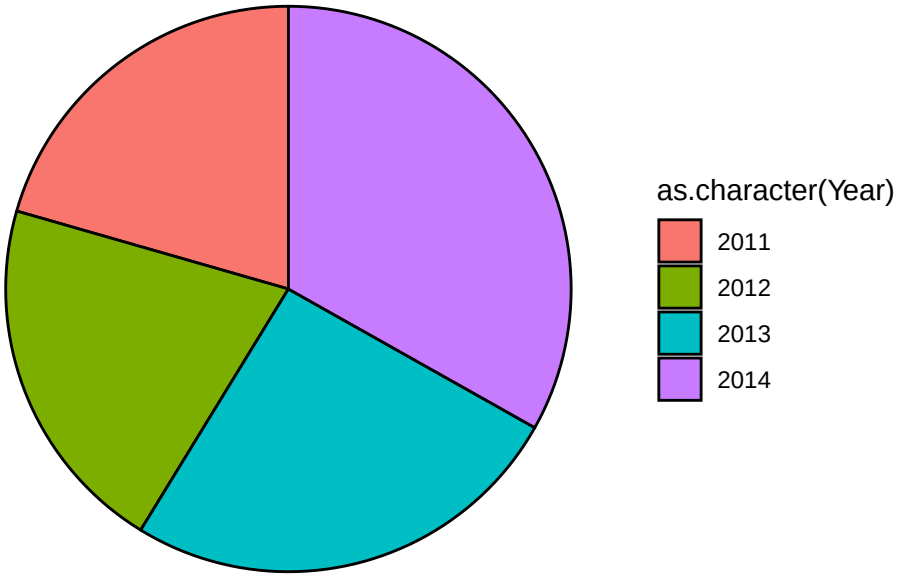



Accessories
Appliances
Art
Binders
Bookcases
Chairs
Copiers
Envelopes
Fasteners
Furnishings
Labels
Machines
Paper
Phones
Storage
Supplies


Region
Central
East
South
West


Category
Furniture
Office Supplies
Technology


as.character(Month)
10
11
12
2
3
4
5
6
7
8
9



Now we sum data according to the features, user, city, state, month, and year.

```
## `summarise()` regrouping output by 'Customer.ID', 'City', 'State', 'Region', 'Month' (override with

## `summarise()` regrouping output by 'City', 'State', 'Month' (override with `.groups` argument)

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

##
## Attaching package: 'modeltools'

## The following object is masked from 'package:arules':
##
##     info

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

##
## Attaching package: 'strucchange'

## The following object is masked from 'package:stringr':
##
##     boundary

## Loaded ROSE 0.0-3

##              Importance
## Customer.ID 205824366.9
## City        150496137.4
## State        27955021.5
## Month         2162043.0
## Year          1544316.5
## Region         926589.9
```

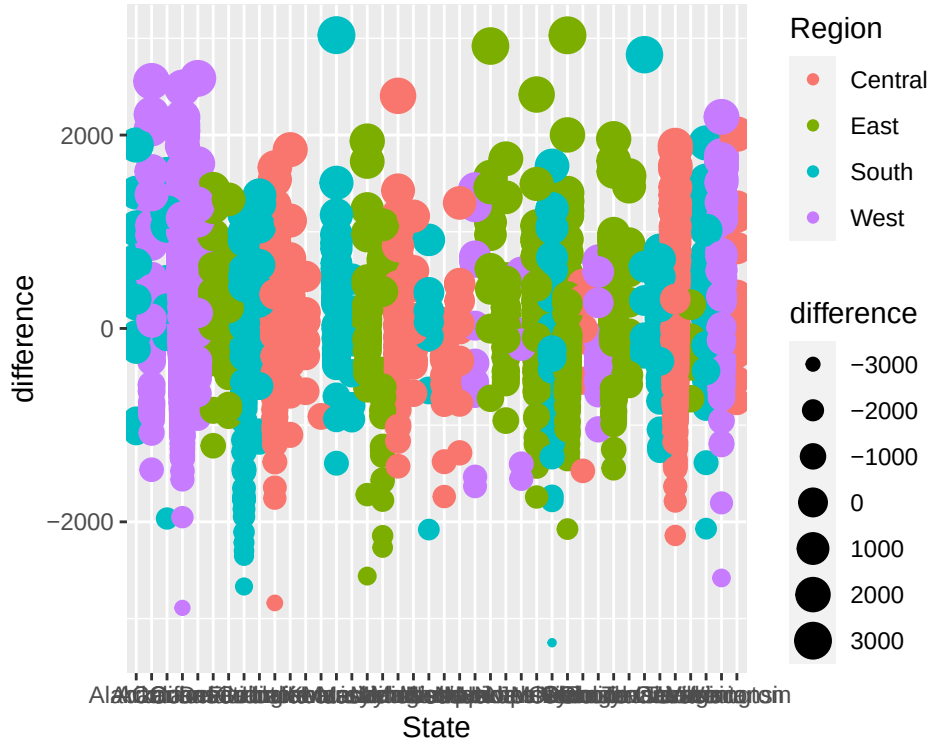
The most important features for prediction are Customer.Id to Year

```
## Loading required package: lattice

## Warning: The `i` argument of ``[`()` can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

## Warning in predict.lm(fit2, newdata = newTestDataSet %>% select(-
## summedSaleByCustomer), : prediction from a rank-deficient fit may be misleading

##   Customer.ID          City         State Region Month Year
## 1    AB-10060       Concord New Hampshire   East     8 2013
## 2    AB-10060       Seattle    Washington   West    11 2014
## 3    AD-10180  Philadelphia  Pennsylvania   East    11 2011
## 4    AD-10180 San Francisco    California   West     5 2014
## 5    AD-10180       Yonkers      New York   East    11 2014
## 6    AG-10390     Arlington      Virginia  South     4 2011
##   summedSaleByCustomer predicted.value2
## 1               27.930       -941.68885
## 2             1474.779        -73.65179
## 3                5.880        360.53912
## 4              163.960        194.24959
## 5              163.960       -767.64102
## 6              129.330       -332.21092
```

To draw a result on the US map, we need to join with the longitude and latitude data.

Evaluation

```
##   Customer.ID          City         State Region Month Year
## 1    AB-10060       Concord New Hampshire   East     8 2013
## 2    AB-10060       Seattle    Washington   West    11 2014
## 3    AD-10180  Philadelphia  Pennsylvania   East    11 2011
## 4    AD-10180 San Francisco    California   West     5 2014
## 5    AD-10180       Yonkers      New York   East    11 2014
## 6    AG-10390     Arlington      Virginia  South     4 2011
##   summedSaleByCustomer predicted.value2 difference
## 1               27.930       -941.68885  969.61885
## 2             1474.779        -73.65179 1548.43079
## 3                5.880        360.53912 -354.65912
## 4              163.960        194.24959  -30.28959
## 5              163.960       -767.64102  931.60102
## 6              129.330       -332.21092  461.54092
```



As the result shows, it is hard to predict the Regional Sales result based on its customer's location, City, State, and Regions. We may need more information to predict the Sales according to the month and year.

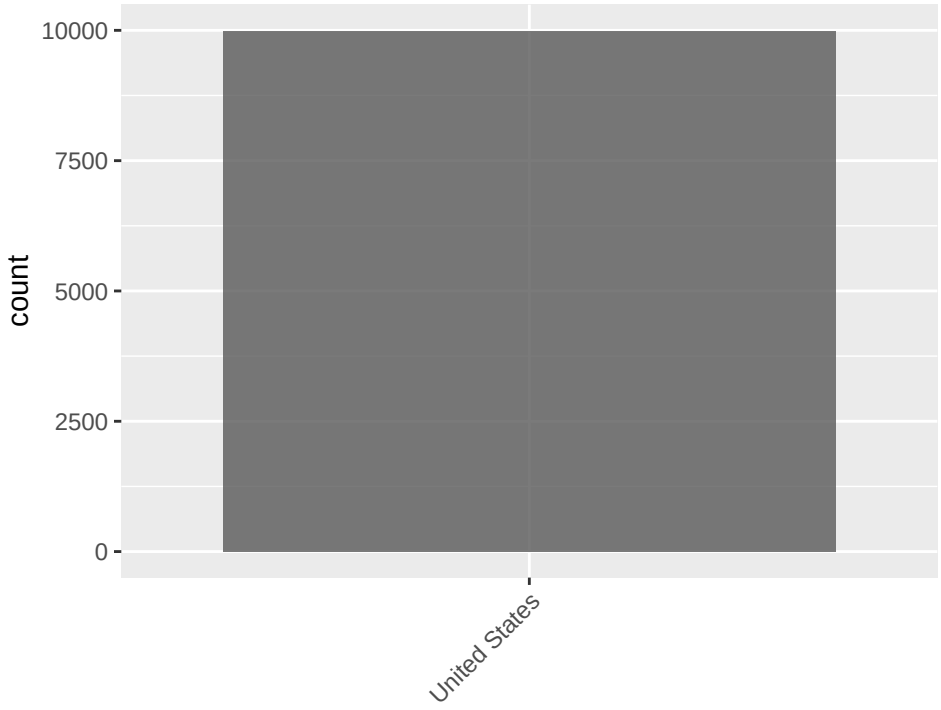
4.2.2 Time Series Analysis

Number of rows and columns

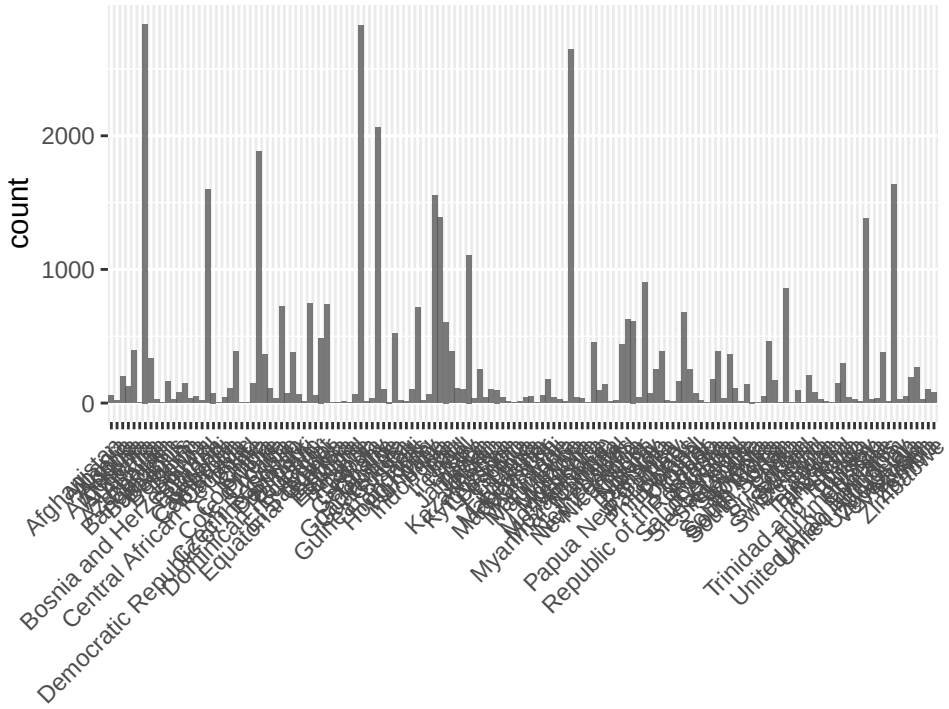
```
## [1] 19
```

```
## [1] 51290
```

Plot of US sales



Sales of countries other than the US



The main customer base is the US

We will remove the countries under a certain level of sales.

The range of sales is :

```
## [1]     0.444 22638.480
```

Remove the rows which have Sales below 5000

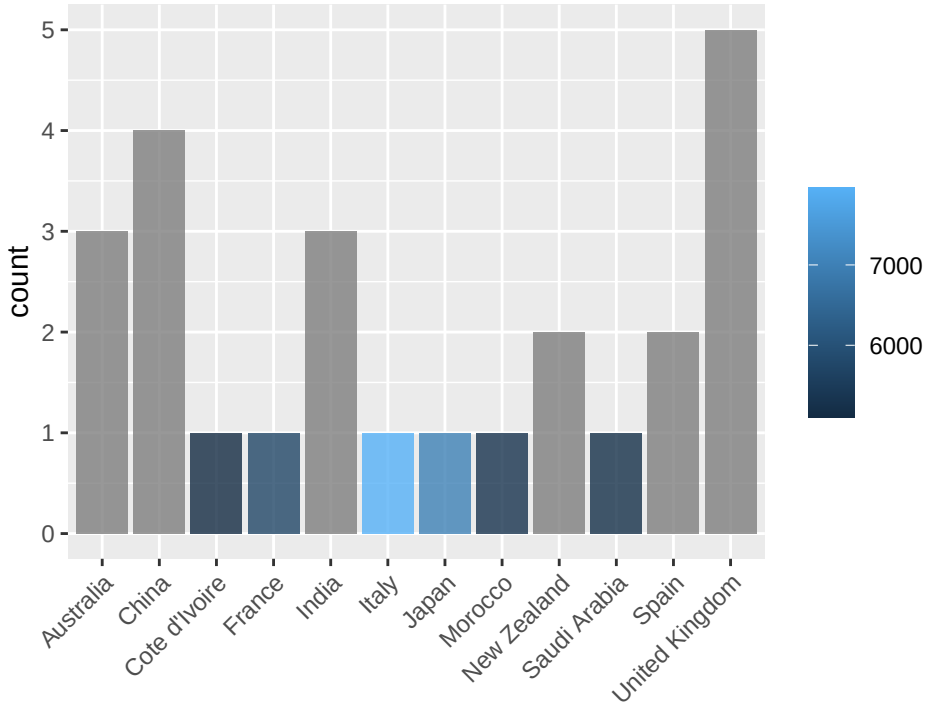
```
##      Row.ID        Order.ID Order.Date  Ship.Date      Ship.Mode     Segment
## 2256  31462  CA-2011-139892 2011-09-08 2011-09-12 Standard Class    Consumer
## 2274  13560 ES-2011-3248922 2011-09-08 2011-09-11   Second Class   Corporate
## 2757  27720   ID-2011-64599 2011-02-10       <NA> Standard Class   Corporate
## 2858  12051 ES-2011-2257437 2011-08-10       <NA> Standard Class    Consumer
## 2859  49997    IV-2011-9090 2011-08-10       <NA> Standard Class    Consumer
## 3189  27407   IN-2011-35178 2011-11-11       <NA> Standard Class Home Office
##             City    State        Country Market     Region        Category
## 2256 San Antonio    Texas  United States     US    Central      Technology
## 2274        Lugo  Galicia          Spain     EU      South Office Supplies
## 2757        Fuji Shizuoka          Japan   APAC North Asia      Technology
## 2858      London  England United Kingdom     EU      North      Technology
## 2859     Abidjan  Lagunes  Cote d'Ivoire Africa     Africa      Technology
## 3189      Suzhou    Gansu          China   APAC North Asia      Technology
##      Sub.Category    Sales Quantity Discount    Profit Shipping.Cost
## 2256     Machines 8159.952        8      0.4 -1359.992        342.11
## 2274   Appliances 6517.080       12      0.0  2476.440         28.74
## 2757       Phones 6998.640       11      0.0  2939.310        413.80
## 2858       Phones 5276.988        9      0.1  1758.888        454.81
## 2859       Phones 5100.000        8      0.0  1428.000         85.22
## 3189       Phones 5725.350        9      0.0  1602.990        302.61
##      Order.Priority
## 2256         Medium
## 2274       Critical
## 2757         Medium
## 2858         Medium
## 2859         Medium
## 3189         Medium
```

The new range of Sales and countries that have sales above 5000

```
## [1]  5049.00 22638.48
```

```
##  [1] "United States"  "Spain"          "Japan"          "United Kingdom"
##  [5] "Cote d'Ivoire"  "China"          "Australia"      "India"
##  [9] "Italy"          "New Zealand"    "France"         "Saudi Arabia"
## [13] "Morocco"
```

Sales in countries other than the US with sales >= 5000



Generate Time series plot of the entire dataset ‘

```
##      Row.ID        Order.ID Order.Date  Ship.Date      Ship.Mode     Segment
## 2256  31462  CA-2011-139892   Sep 2011 2011-09-12 Standard Class    Consumer
## 2274  13560 ES-2011-3248922   Sep 2011 2011-09-11   Second Class   Corporate
## 2757  27720   ID-2011-64599   Feb 2011       <NA> Standard Class   Corporate
## 2858  12051 ES-2011-2257437   Aug 2011       <NA> Standard Class    Consumer
## 2859  49997    IV-2011-9090   Aug 2011       <NA> Standard Class    Consumer
## 3189  27407   IN-2011-35178   Nov 2011       <NA> Standard Class Home Office
##             City    State        Country Market     Region        Category
## 2256 San Antonio    Texas  United States     US    Central      Technology
## 2274        Lugo  Galicia          Spain     EU      South Office Supplies
## 2757        Fuji Shizuoka          Japan   APAC North Asia      Technology
## 2858      London  England United Kingdom     EU      North      Technology
## 2859     Abidjan  Lagunes  Cote d'Ivoire Africa     Africa      Technology
## 3189      Suzhou    Gansu          China   APAC North Asia      Technology
##      Sub.Category    Sales Quantity Discount    Profit Shipping.Cost
## 2256     Machines 8159.952        8      0.4 -1359.992        342.11
## 2274   Appliances 6517.080       12      0.0  2476.440         28.74
## 2757       Phones 6998.640       11      0.0  2939.310        413.80
## 2858       Phones 5276.988        9      0.1  1758.888        454.81
## 2859       Phones 5100.000        8      0.0  1428.000         85.22
## 3189       Phones 5725.350        9      0.0  1602.990        302.61
##      Order.Priority
## 2256         Medium
## 2274       Critical
## 2757         Medium
## 2858         Medium
## 2859         Medium
```

```
## 3189         Medium
```

Filter the Order.Date and Sales to used for the timeseries

```
##   Order.Date    Sales
## 1   Sep 2011 8159.952
## 2   Sep 2011 6517.080
## 3   Feb 2011 6998.640
## 4   Aug 2011 5276.988
## 5   Aug 2011 5100.000
## 6   Nov 2011 5725.350
```

Check the start, end , frequency of the time series

```
## [1] 2011    1
```

```
## [1] 2014   12
```

```
## [1] 0.08333333
```

```
## [1] 12
```

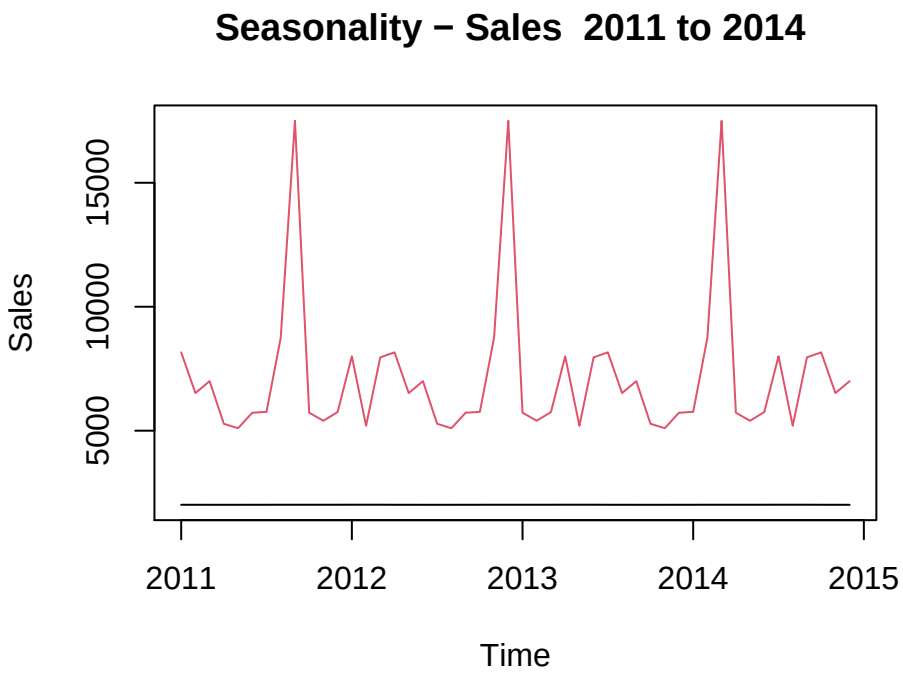
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2011 2011.000 2011.083 2011.167 2011.250 2011.333 2011.417 2011.500 2011.583
## 2012 2012.000 2012.083 2012.167 2012.250 2012.333 2012.417 2012.500 2012.583
## 2013 2013.000 2013.083 2013.167 2013.250 2013.333 2013.417 2013.500 2013.583
## 2014 2014.000 2014.083 2014.167 2014.250 2014.333 2014.417 2014.500 2014.583
##           Sep      Oct      Nov      Dec
## 2011 2011.667 2011.750 2011.833 2011.917
## 2012 2012.667 2012.750 2012.833 2012.917
## 2013 2013.667 2013.750 2013.833 2013.917
## 2014 2014.667 2014.750 2014.833 2014.917
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2011   1   2   3   4   5   6   7   8   9  10  11  12
## 2012   1   2   3   4   5   6   7   8   9  10  11  12
## 2013   1   2   3   4   5   6   7   8   9  10  11  12
## 2014   1   2   3   4   5   6   7   8   9  10  11  12
```

```
##          Order.Date     Sales
## Jan 2011   2011.667  8159.952
## Feb 2011   2011.667  6517.080
## Mar 2011   2011.083  6998.640
## Apr 2011   2011.583  5276.988
## May 2011   2011.583  5100.000
## Jun 2011   2011.833  5725.350
## Jul 2011   2012.583  5759.964
## Aug 2011   2013.083  8749.950
## Sep 2011   2013.750 17499.950
## Oct 2011   2013.417  5726.160
## Nov 2011   2013.750  5399.910
```

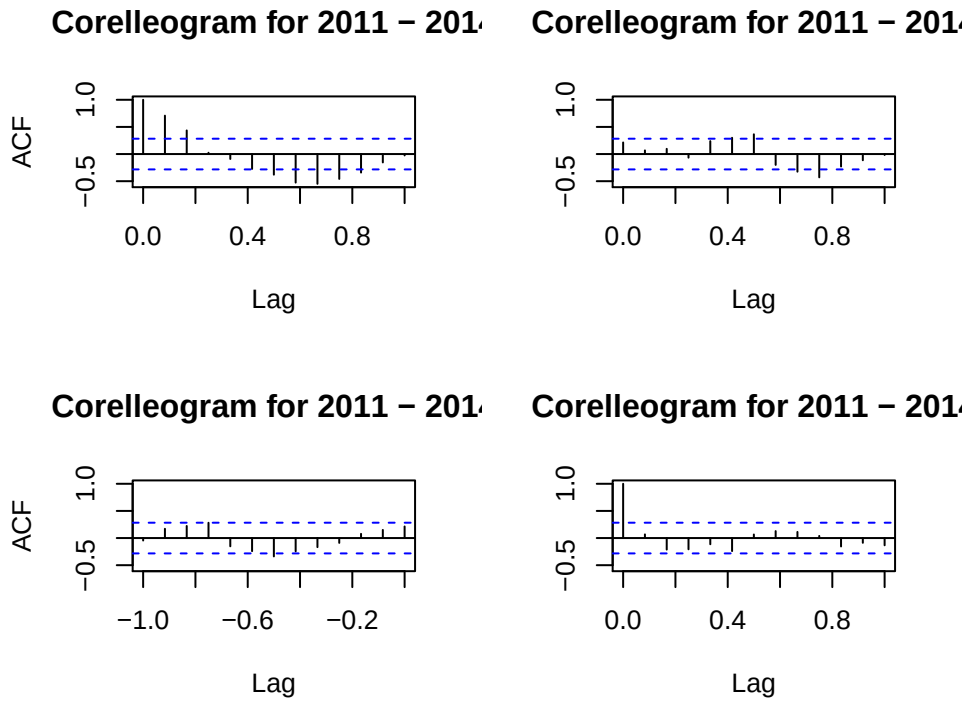
```
## Dec 2011   2013.417  5751.540
## Jan 2012   2014.833  7999.980
## Feb 2012   2014.750  5199.960
## Mar 2012   2014.667  7958.580
## Apr 2012   2011.667  8159.952
## May 2012   2011.667  6517.080
## Jun 2012   2011.083  6998.640
## Jul 2012   2011.583  5276.988
## Aug 2012   2011.583  5100.000
## Sep 2012   2011.833  5725.350
## Oct 2012   2012.583  5759.964
## Nov 2012   2013.083  8749.950
## Dec 2012   2013.750 17499.950
## Jan 2013   2013.417  5726.160
## Feb 2013   2013.750  5399.910
## Mar 2013   2013.417  5751.540
## Apr 2013   2014.833  7999.980
## May 2013   2014.750  5199.960
## Jun 2013   2014.667  7958.580
## Jul 2013   2011.667  8159.952
## Aug 2013   2011.667  6517.080
## Sep 2013   2011.083  6998.640
## Oct 2013   2011.583  5276.988
## Nov 2013   2011.583  5100.000
## Dec 2013   2011.833  5725.350
## Jan 2014   2012.583  5759.964
## Feb 2014   2013.083  8749.950
## Mar 2014   2013.750 17499.950
## Apr 2014   2013.417  5726.160
## May 2014   2013.750  5399.910
## Jun 2014   2013.417  5751.540
## Jul 2014   2014.833  7999.980
## Aug 2014   2014.750  5199.960
## Sep 2014   2014.667  7958.580
## Oct 2014   2011.667  8159.952
## Nov 2014   2011.667  6517.080
## Dec 2014   2011.083  6998.640
```

Plotting the time series for 2011-2014 to know the Seasonality




```
##
## Autocorrelations of series 'tseries', by lag
##
## , , Order.Date
##
##  Order.Date       Sales
##   1.000 ( 0.0000)  0.214 ( 0.0000)
##   0.708 ( 0.0833)  0.150 (-0.0833)
##   0.435 ( 0.1667)  0.078 (-0.1667)
##   0.024 ( 0.2500) -0.089 (-0.2500)
##  -0.091 ( 0.3333) -0.168 (-0.3333)
##  -0.274 ( 0.4167) -0.244 (-0.4167)
##  -0.380 ( 0.5000) -0.338 (-0.5000)
##  -0.526 ( 0.5833) -0.242 (-0.5833)
##  -0.549 ( 0.6667) -0.152 (-0.6667)
##  -0.459 ( 0.7500)  0.282 (-0.7500)
##  -0.341 ( 0.8333)  0.223 (-0.8333)
##  -0.155 ( 0.9167)  0.171 (-0.9167)
##  -0.026 ( 1.0000) -0.045 (-1.0000)
##   0.289 ( 1.0833)  0.075 (-1.0833)
##
## , , Sales
##
##  Order.Date       Sales
##   0.214 ( 0.0000)  1.000 ( 0.0000)
##   0.068 ( 0.0833)  0.068 ( 0.0833)
##   0.097 ( 0.1667) -0.213 ( 0.1667)
##  -0.067 ( 0.2500) -0.207 ( 0.2500)
##   0.238 ( 0.3333) -0.114 ( 0.3333)
```

```
##   0.300 ( 0.4167) -0.240 ( 0.4167)
##   0.364 ( 0.5000)  0.066 ( 0.5000)
##  -0.200 ( 0.5833)  0.128 ( 0.5833)
##  -0.325 ( 0.6667)  0.113 ( 0.6667)
##  -0.430 ( 0.7500)  0.039 ( 0.7500)
##  -0.230 ( 0.8333) -0.155 ( 0.8333)
##  -0.114 ( 0.9167) -0.088 ( 0.9167)
##  -0.017 ( 1.0000) -0.134 ( 1.0000)
##   0.089 ( 1.0833) -0.144 ( 1.0833)
```



Now we explore time series data for each year separately

Get the rows for sales in 2011

Get only the year and month from the Order Date

```
##   Order.Date    Sales
## 1   Sep 2011 8159.952
## 2   Sep 2011 6517.080
## 3   Feb 2011 6998.640
## 4   Aug 2011 5276.988
## 5   Aug 2011 5100.000
## 6   Nov 2011 5725.350
```

```
## [1] FALSE
```

Convert to time series object Time series object for 2011 sales

```
##          Order.Date    Sales
```

```
## Jan 2011   2011.667 8159.952
## Feb 2011   2011.667 6517.080
## Mar 2011   2011.083 6998.640
## Apr 2011   2011.583 5276.988
## May 2011   2011.583 5100.000
## Jun 2011   2011.833 5725.350
## Jul 2011   2011.667 8159.952
## Aug 2011   2011.667 6517.080
## Sep 2011   2011.083 6998.640
## Oct 2011   2011.583 5276.988
## Nov 2011   2011.583 5100.000
## Dec 2011   2011.833 5725.350
```

```
## [1] TRUE
```

Check the start, end , frequency of the time series

```
## [1] 2011    1
```

```
## [1] 2011   12
```

```
## [1] 0.08333333
```

```
## [1] 12
```

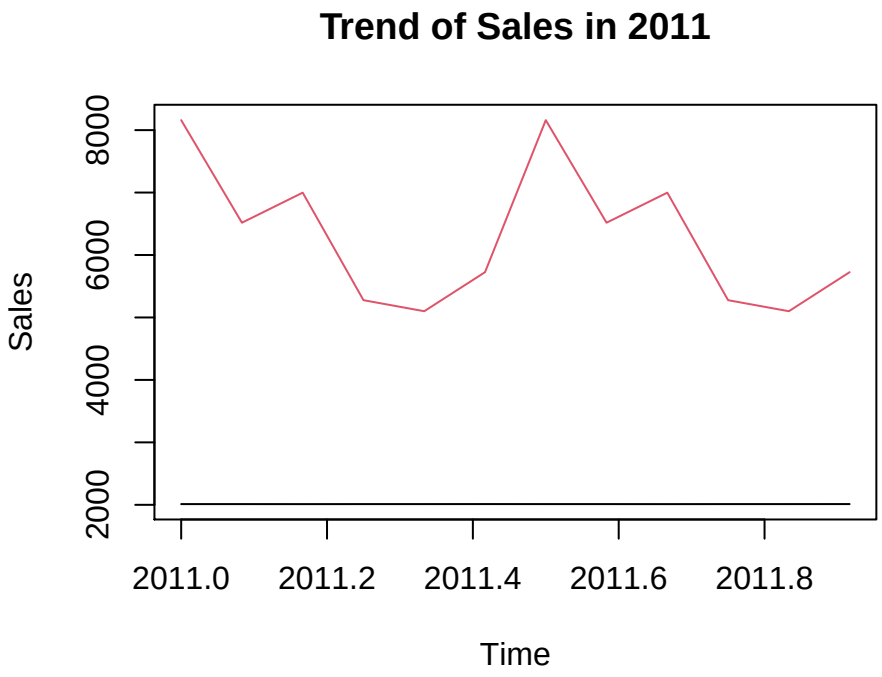
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2011 2011.000 2011.083 2011.167 2011.250 2011.333 2011.417 2011.500 2011.583
##           Sep      Oct      Nov      Dec
## 2011 2011.667 2011.750 2011.833 2011.917
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2011   1   2   3   4   5   6   7   8   9  10  11  12
```

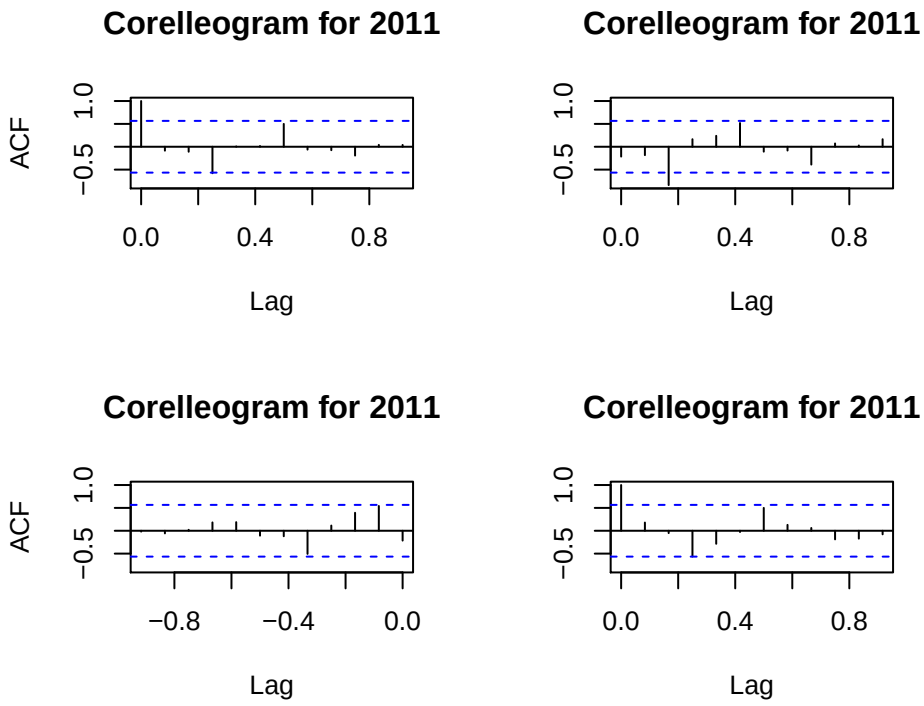
```
##          Order.Date    Sales
## Jan 2011   2011.667 8159.952
## Feb 2011   2011.667 6517.080
## Mar 2011   2011.083 6998.640
## Apr 2011   2011.583 5276.988
## May 2011   2011.583 5100.000
## Jun 2011   2011.833 5725.350
## Jul 2011   2011.667 8159.952
## Aug 2011   2011.667 6517.080
## Sep 2011   2011.083 6998.640
## Oct 2011   2011.583 5276.988
## Nov 2011   2011.583 5100.000
## Dec 2011   2011.833 5725.350
```

```
## [1] TRUE
```

Plotting the time series for 2011 to show the Trend in 2011




```
##
## Autocorrelations of series 'tseries2011', by lag
##
## , , Order.Date
##
##  Order.Date       Sales
##   1.000 ( 0.0000) -0.215 ( 0.0000)
##  -0.086 ( 0.0833)  0.542 (-0.0833)
##  -0.109 ( 0.1667)  0.393 (-0.1667)
##  -0.579 ( 0.2500)  0.114 (-0.2500)
##   0.008 ( 0.3333) -0.506 (-0.3333)
##   0.016 ( 0.4167) -0.120 (-0.4167)
##   0.500 ( 0.5000) -0.108 (-0.5000)
##  -0.063 ( 0.5833)  0.188 (-0.5833)
##
## , , Sales
##
##  Order.Date       Sales
##  -0.215 ( 0.0000)  1.000 ( 0.0000)
##  -0.184 ( 0.0833)  0.179 ( 0.0833)
##  -0.839 ( 0.1667) -0.052 ( 0.1667)
##   0.165 ( 0.2500) -0.565 ( 0.2500)
##   0.239 ( 0.3333) -0.285 ( 0.3333)
##   0.519 ( 0.4167) -0.028 ( 0.4167)
##  -0.108 ( 0.5000)  0.500 ( 0.5000)
##  -0.083 ( 0.5833)  0.129 ( 0.5833)
```



Autocorrelation measures the relationship between a variable's current values and its historical values. Trend refers to the increasing or decreasing values in a given time series.

```
##          Order.Date    Sales
## Jan 2012      15371 593.9895
## Feb 2012      15371 151.9200
## Mar 2012      15371 200.1600
## Apr 2012      15371 192.8800
## May 2012      15371  94.0200
## Jun 2012      15371  57.0000
## Jul 2012      15371  95.5170
## Aug 2012      15371  69.5544
## Sep 2012      15371 139.5900
## Oct 2012      15371  33.1200
## Nov 2012      15371   8.9640
## Dec 2012      15371  18.5400
```

Check the start, end of the time series

```
## [1] 2012    1
```

```
## [1] 2012   12
```

```
## [1] 0.08333333
```

```
## [1] 12
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2012 2012.000 2012.083 2012.167 2012.250 2012.333 2012.417 2012.500 2012.583
##           Sep      Oct      Nov      Dec
## 2012 2012.667 2012.750 2012.833 2012.917
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2012   1   2   3   4   5   6   7   8   9  10  11  12
```

```
##          Order.Date    Sales
## Jan 2012      15371 593.9895
## Feb 2012      15371 151.9200
## Mar 2012      15371 200.1600
## Apr 2012      15371 192.8800
## May 2012      15371  94.0200
## Jun 2012      15371  57.0000
## Jul 2012      15371  95.5170
## Aug 2012      15371  69.5544
## Sep 2012      15371 139.5900
## Oct 2012      15371  33.1200
## Nov 2012      15371   8.9640
## Dec 2012      15371  18.5400
```

There is only one sale in 2012 over $5000. Not enough data for the timeseries

2013

Get the rows for sales in 2013

Get only the year and month form the Order Date

```
##   Order.Date    Sales
## 1   Jan 2013 1649.214
## 2   Jan 2013 1358.280
## 3   Jan 2013  728.568
## 4   Jan 2013 2189.520
## 5   Jan 2013 1362.060
## 6   Jan 2013  299.052
```

```
## [1] FALSE
```

Convert to time series object Time series object for 2013 sales

```
##          Order.Date    Sales
## Jan 2013       2013 1649.214
## Feb 2013       2013 1358.280
## Mar 2013       2013  728.568
## Apr 2013       2013 2189.520
## May 2013       2013 1362.060
## Jun 2013       2013  299.052
## Jul 2013       2013  246.120
## Aug 2013       2013  155.034
## Sep 2013       2013  242.250
## Oct 2013       2013  416.664
## Nov 2013       2013   85.428
## Dec 2013       2013   94.980
```

```
## [1] TRUE
```

Check the start, end of the time series

```
## [1] 2013    1
```

```
## [1] 2013   12
```

```
## [1] 0.08333333
```

```
## [1] 12
```

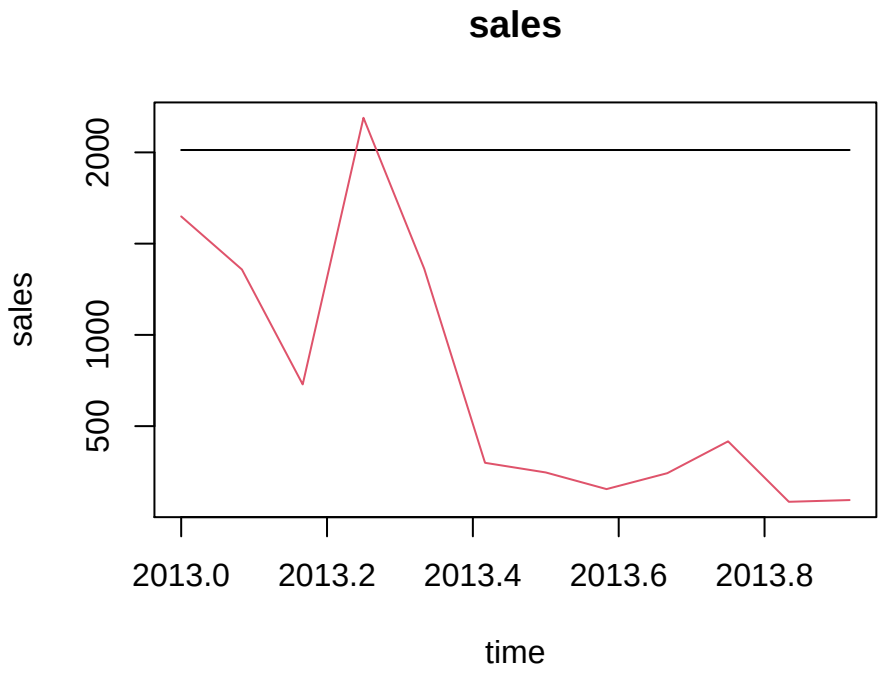
```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2013 2013.000 2013.083 2013.167 2013.250 2013.333 2013.417 2013.500 2013.583
##           Sep      Oct      Nov      Dec
## 2013 2013.667 2013.750 2013.833 2013.917
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2013   1   2   3   4   5   6   7   8   9  10  11  12
```

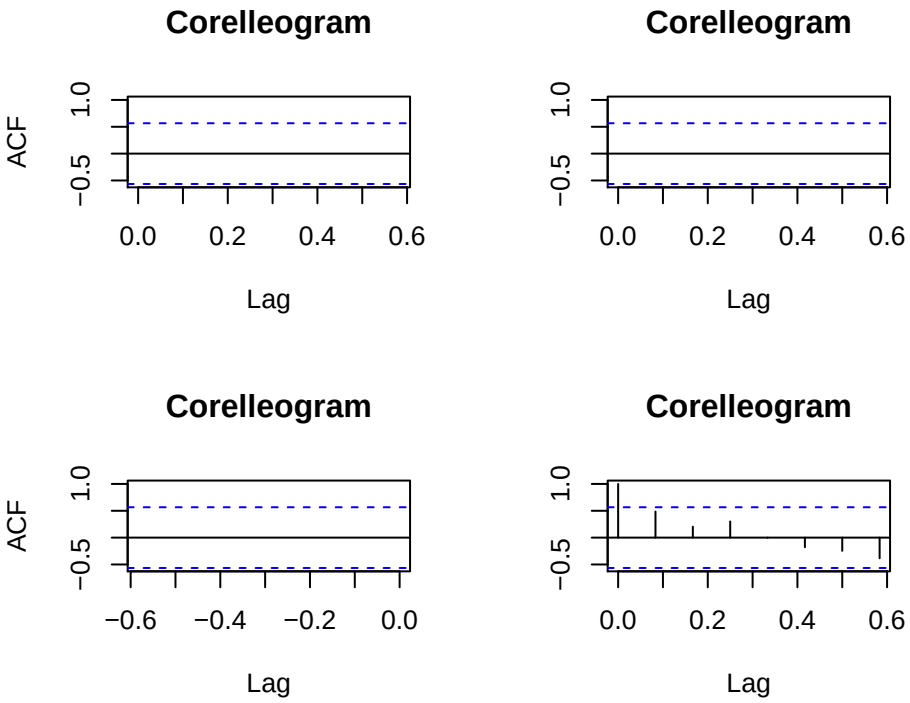
```
##          Order.Date    Sales
## Jan 2013       2013 1649.214
## Feb 2013       2013 1358.280
## Mar 2013       2013  728.568
## Apr 2013       2013 2189.520
## May 2013       2013 1362.060
## Jun 2013       2013  299.052
## Jul 2013       2013  246.120
## Aug 2013       2013  155.034
## Sep 2013       2013  242.250
## Oct 2013       2013  416.664
## Nov 2013       2013   85.428
## Dec 2013       2013   94.980
```

Check if the time series object was generated

```
## [1] TRUE
```



When the autocorrelation in a time series is high, it becomes easy to predict future values by simply referring to past values.



2014

Get the rows for sales in 2014

Get only the year and month form the Order Date

```
##   Order.Date   Sales
## 1   Nov 2014 7999.98
## 2   Oct 2014 5199.96
## 3   Sep 2014 7958.58
```

```
## [1] FALSE
```

Convert to time series object Time series object for 2011 sales

```
##          Order.Date   Sales
## Jan 2014   2014.833 7999.98
## Feb 2014   2014.750 5199.96
## Mar 2014   2014.667 7958.58
## Apr 2014   2014.833 7999.98
## May 2014   2014.750 5199.96
## Jun 2014   2014.667 7958.58
## Jul 2014   2014.833 7999.98
## Aug 2014   2014.750 5199.96
## Sep 2014   2014.667 7958.58
## Oct 2014   2014.833 7999.98
## Nov 2014   2014.750 5199.96
## Dec 2014   2014.667 7958.58
```

```
## [1] TRUE
```

Check the start, end of the time series

```
## [1] 2014    1
```

```
## [1] 2014   12
```

```
## [1] 0.08333333
```

```
## [1] 12
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2014 2014.000 2014.083 2014.167 2014.250 2014.333 2014.417 2014.500 2014.583
##           Sep      Oct      Nov      Dec
## 2014 2014.667 2014.750 2014.833 2014.917
```

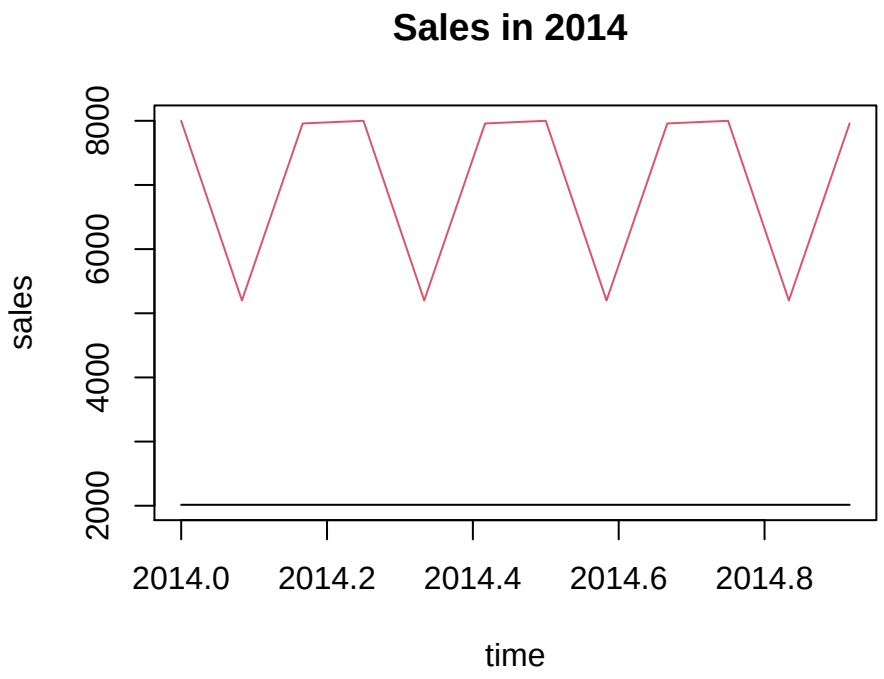
```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2014   1   2   3   4   5   6   7   8   9  10  11  12
```

```
##          Order.Date   Sales
## Jan 2014   2014.833 7999.98
## Feb 2014   2014.750 5199.96
## Mar 2014   2014.667 7958.58
## Apr 2014   2014.833 7999.98
## May 2014   2014.750 5199.96
```

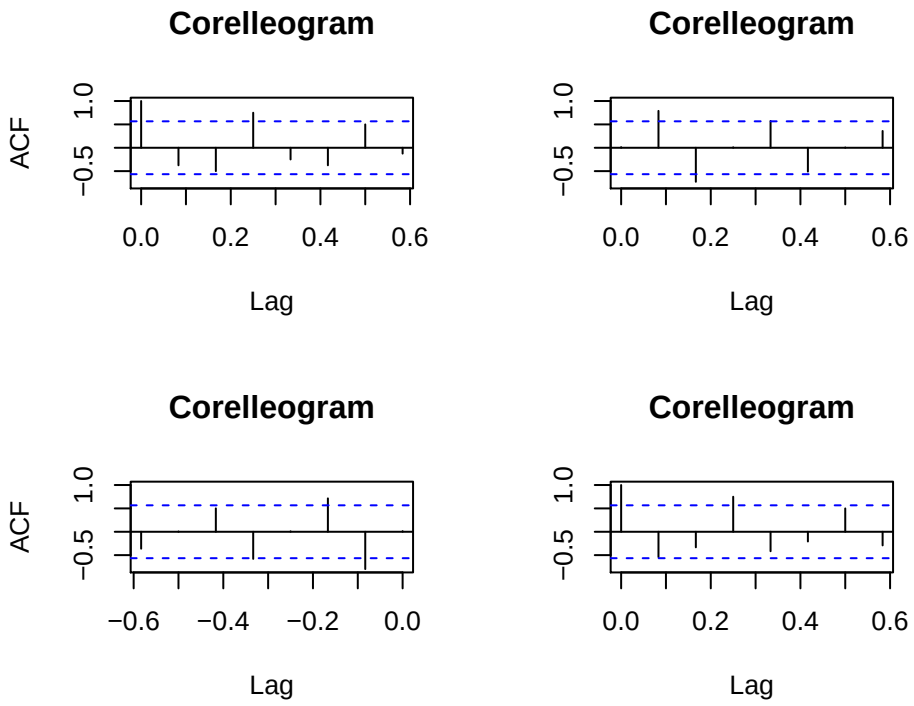
```
## Jun 2014   2014.667 7958.58
## Jul 2014   2014.833 7999.98
## Aug 2014   2014.750 5199.96
## Sep 2014   2014.667 7958.58
## Oct 2014   2014.833 7999.98
## Nov 2014   2014.750 5199.96
## Dec 2014   2014.667 7958.58
```

Check if the time series object was generated

```
## [1] TRUE
```



When the autocorrelation in a time series is high, it becomes easy to predict future values by simply referring to past values.



We have plotted the timeseries to look for Seasonality in the 4 year and trends in each on the 4 years. Also, we have plotted the autocorrelation function to try and dig more into the data. Wecould use statistical tests to know whether or not the data is auto correlated.

4.2.3 Random Forest Regression Model

Random Forest is a classification algorithm used in supervised machine learning and consists of constructing multiple decision trees during training and outputs the mode of the predicted variable of each decision tree. For the current application, the predicted variable is the dollar amount of sales. The Random Forest function in R allows the user to customize multiple input parameters, including among other, the number of trees, number of features, tree depth and the minimum leaf size. This Random Forest model uses the default parameters available in R, with the sole exception being the number of trees. The former was set to 100, as numbers above started to affect processing time. Additionally, observing the graph which shows the model error as a function of the number of trees, it could be seen that after 100 trees, the deviation in error values decreased significantly, another reason why the number of trees parameter was set to 100.

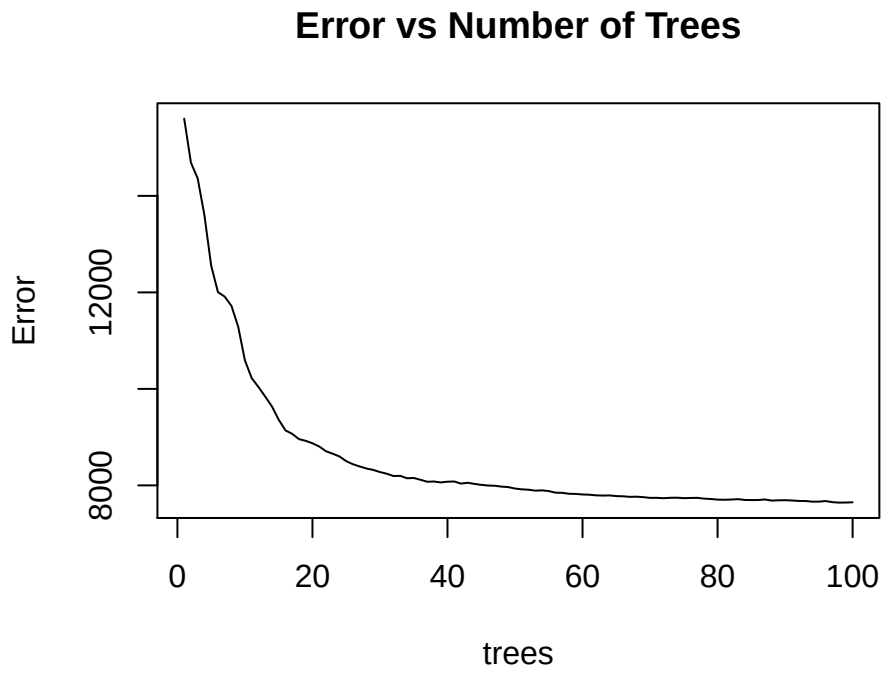
Prior to running the Random Forest model, features the following features were set to be treated as factors: Ship.mode, segment, city, state, country, market, region, category, sub.category and priority. The remaining values were left as numeric. During the initial run of the model, an error was encountered which stated that the Random Forest function in R could not handle features with more than 53 unique categories. This affected three features directly: city, state and country which had 3636,1094 and 145 unique values. Given this information, those features were excluded from the analysis. Both the market and region features provided us with location information at a higher level.

Here is a summary of the generated Random Forest model:

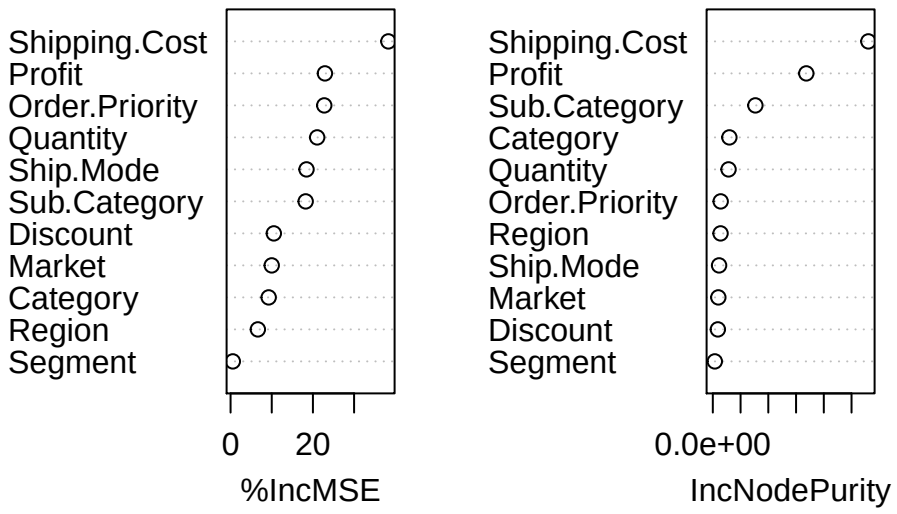
```
##
## Call:
##  randomForest(formula = train$Sales ~ ., data = train, importance = TRUE,      ntree = 100)
```

```
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 7649.933
##                     % Var explained: 89.68
```

Then, using the generated model, the mean of squared residuals error was plotted against the number of trees.



As stated previously, it can be seen that the deviation in error significantly decreases as the number of trees approaches 100. Additionally, the variable importance was determined and is shown in the figure below:



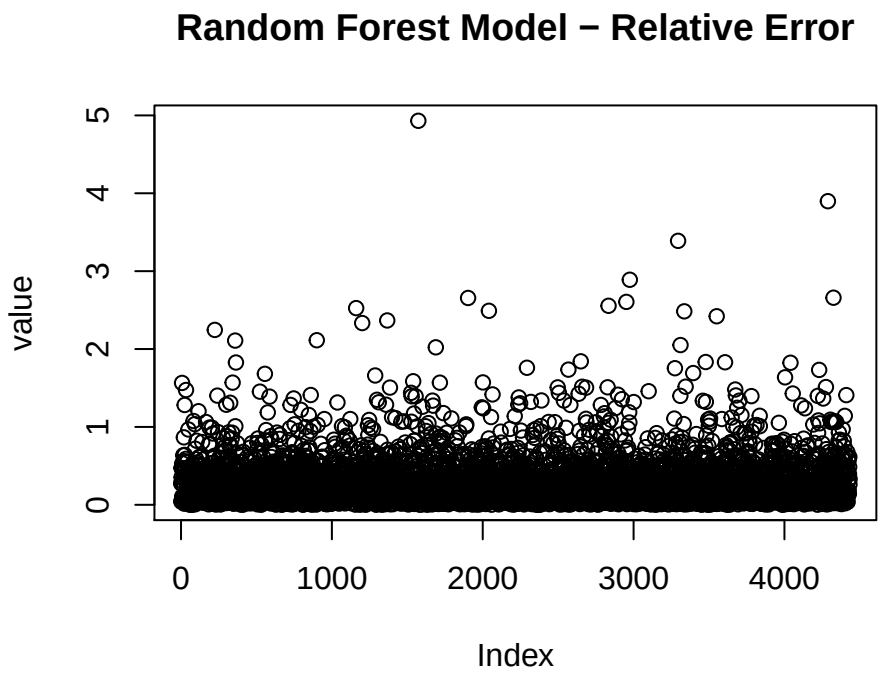
It is important to note, that the current list of variables shown reflects the most recent model. After initially running the Random Forest algorithm and visualizing the relative importance of the variables, the features that were identified as low importance, were removed from the dataset and the algorithm was re-applied in an attempt to improve the performance.

```
## Warning: package 'ROCR' was built under R version 4.0.3
```

Lastly, we needed to evaluate and measure the performance of the model. As this was a regression application, we could not use a confusion matrix to determine accuracy. Instead, we directly compared the predicted and actual values using the relative error, described by the following equation:

$$error_{relative} = \frac{|sales_{predict} - sales_{actual}|}{sales_{actual}}$$

where sales_predict is the predicted sales value obtained from the model and the sales_actual parameter, is the actual sales data from the original dataset. Once the relative error was calculated for each observation, the values where visualized on a scatter plot, shown in the figure below.



Observing the figure above, it can be seen that the Random forest model performed quite poorly, having frequent cases where the relative error exceeded 100%. After completed our analysis using the Random Forest algorithm, we determined that it was not the best choice for predicting sales and thus did not select it as our final model.