Anomaly_Detection

Group B

20/02/2021

Abstract

Anomaly detection or Outlier detection identifies data points, events or observations that deviate from dataset's normal behavior. Anomalous data indicate critical incidents or potential opportunities. In order to take advantage of opportunities or fix costly problems anomaly detection has to be done in real time. Unsupervised machine learning models can be used to automate anomaly detection. Unsupervised anomaly detection algorithms scores data based on intrinsic properties of the dataset. Distances and densities are used to give an estimation what is normal and what is an outlier. Anomaly detection monitor is a tool developed for an online retailer to check product quality issues like profit opportunities and sales glitches. The application is built using R and Shinyapp following CRISP-DM framework.

Business Case

Objectives

Detect point anomalies from superstore dataset using K-NN and clustering methods

Import data

```
#load libraries
library(readxl)
library(tidyr)
library(dplyr)
library(ggplot2)
library(anomalize)
library(lemon)
library(DMwR)
```

```
#read data from file
superstore<-read_excel("superstore.xls")
```

Data Understanding

US Superstore dataset is sourced from US superstore dataset . The dataset have online orders for Superstores in U.S. from 2014-2018. Tableau community is the owner of the dataset. The dataset has 9994 records and 21 attributes.

```
data_superstore
```

Table 1: Dataset description

Attribute	Data Type	Description
Row ID	numeric	row number
Order ID	character	unique order number
Order Date	numeric	order placed date
Ship Date	numeric	order shipping date
Ship Mode	character	shipping mode of order
Customer ID	character	unique customer id for order
Customer Name	character	name of customer
Segment	character	section of product
Country	character	country based on order
City	character	city based on order
State	character	state based on order
Postal Code	numeric	pin code
Region	character	region based on order
Product ID	character	product id of product
Category	character	category of product
Sub-Category	character	sub-category of product
Product Name	character	name of product
Sales	numeric	selling price of product
Quantity	numeric	order quantity
Discount	numeric	discount on product
Profit	numeric	profit from product

Data Preparation

```
#name columns
names(superstore)<-c("rowid","orderid","order_date","ship_date","ship_mode","customer_id",
                     "customer_name","segment","country","city","state","postal_code",
                     "region","product_id","category","sub_category","product_name",
                     "sales_amt","quantity","discount","profit_amt")

#drop columns with redundant information
superstore[,c("rowid","customer_name","country")]<-NULL


#convert to date
superstore$order_date<-as.Date(superstore$order_date,format="%Y-%m-%d")
superstore$ship_date<-as.Date(superstore$ship_date,format="%Y-%m-%d")
```

Descriptive Analysis

Continous variables summary

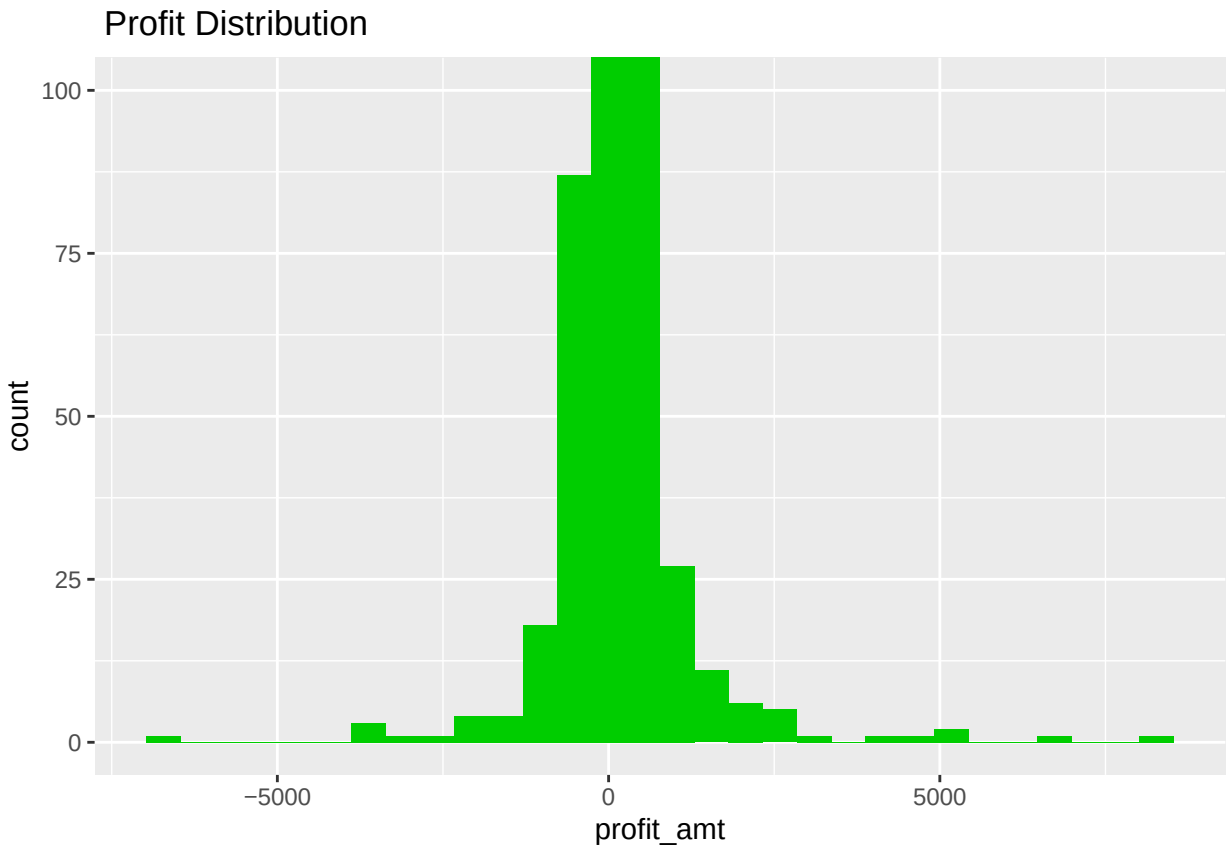
```
superstore %>%
    select_if(is.numeric)%>%
    summary()
```

```
##   postal_code      sales_amt           quantity        discount
##  Min.   : 1040   Min.   :    0.444   Min.   : 1.00   Min.   :0.0000
##  1st Qu.:23223   1st Qu.:   17.280   1st Qu.: 2.00   1st Qu.:0.0000
##  Median :56431   Median :   54.490   Median : 3.00   Median :0.2000
##  Mean   :55190   Mean   :  229.858   Mean   : 3.79   Mean   :0.1562
##  3rd Qu.:90008   3rd Qu.:  209.940   3rd Qu.: 5.00   3rd Qu.:0.2000
##  Max.   :99301   Max.   :22638.480   Max.   :14.00   Max.   :0.8000
##    profit_amt
##  Min.   :-6599.978
##  1st Qu.:    1.729
##  Median :    8.666
##  Mean   :   28.657
##  3rd Qu.:   29.364
##  Max.   : 8399.976
```

Profit

```
ggplot(data=superstore)+
  geom_histogram(mapping=aes(x=profit_amt),fill="green3")+
  coord_cartesian(ylim = c(0, 100))+
  labs(title=" Profit Distribution")
```

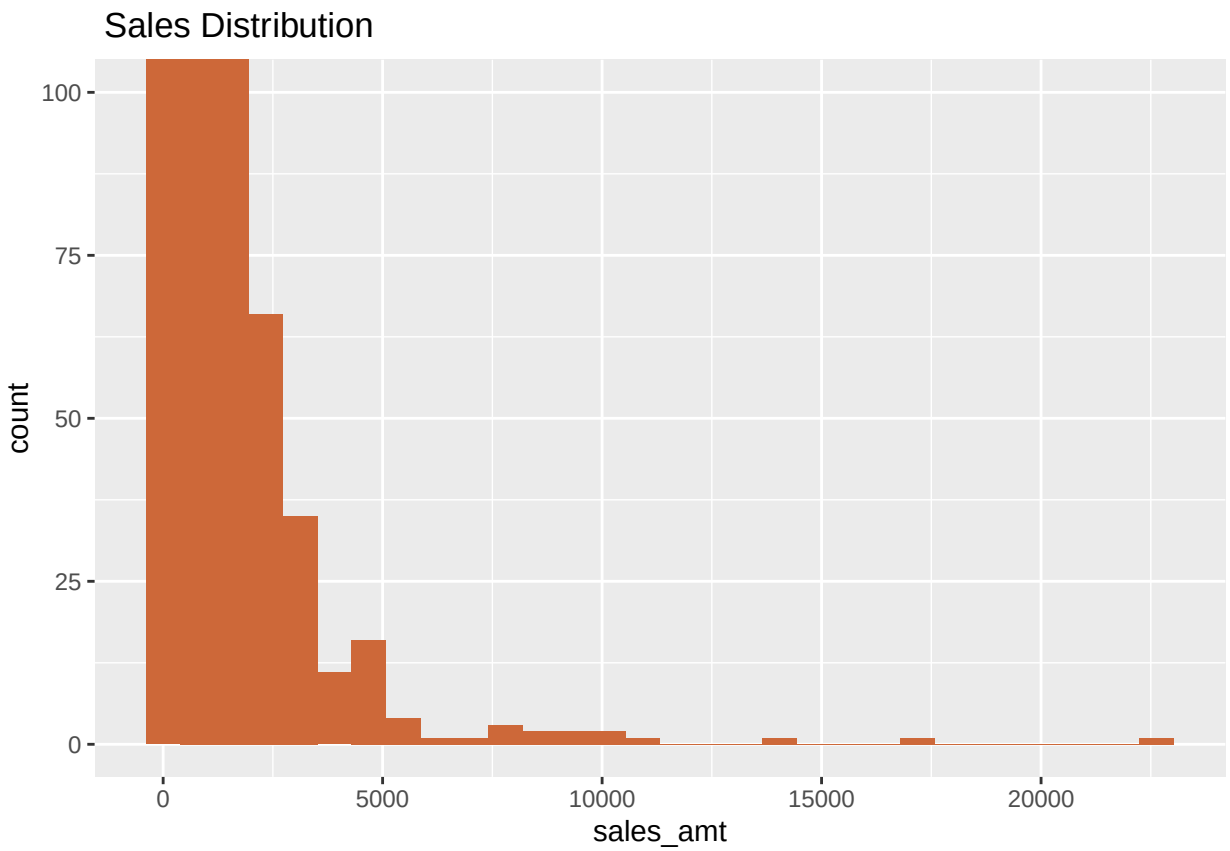
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Sales

```
ggplot(data=superstore)+
  geom_histogram(mapping=aes(x=sales_amt),fill="sienna3")+
  coord_cartesian(ylim = c(0, 100))+labs(title=" Sales Distribution")
```

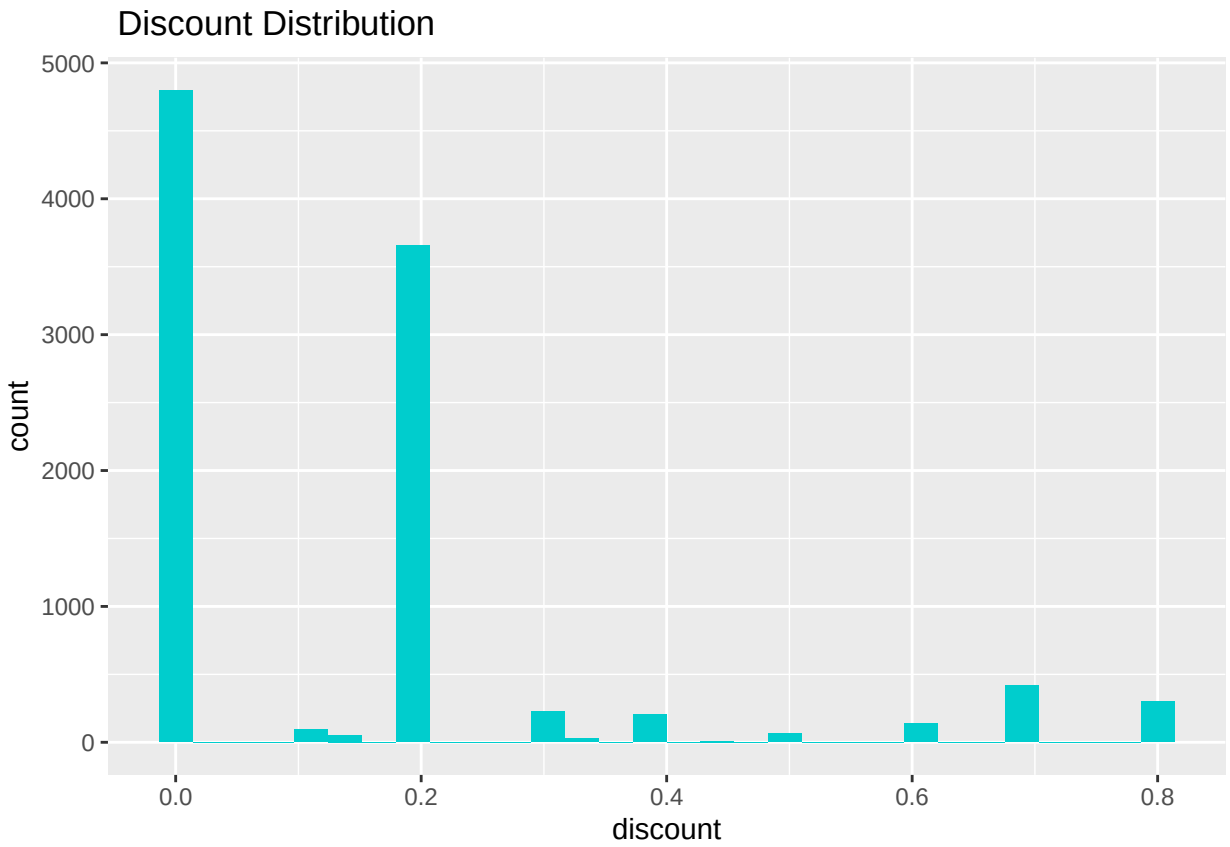
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Discount

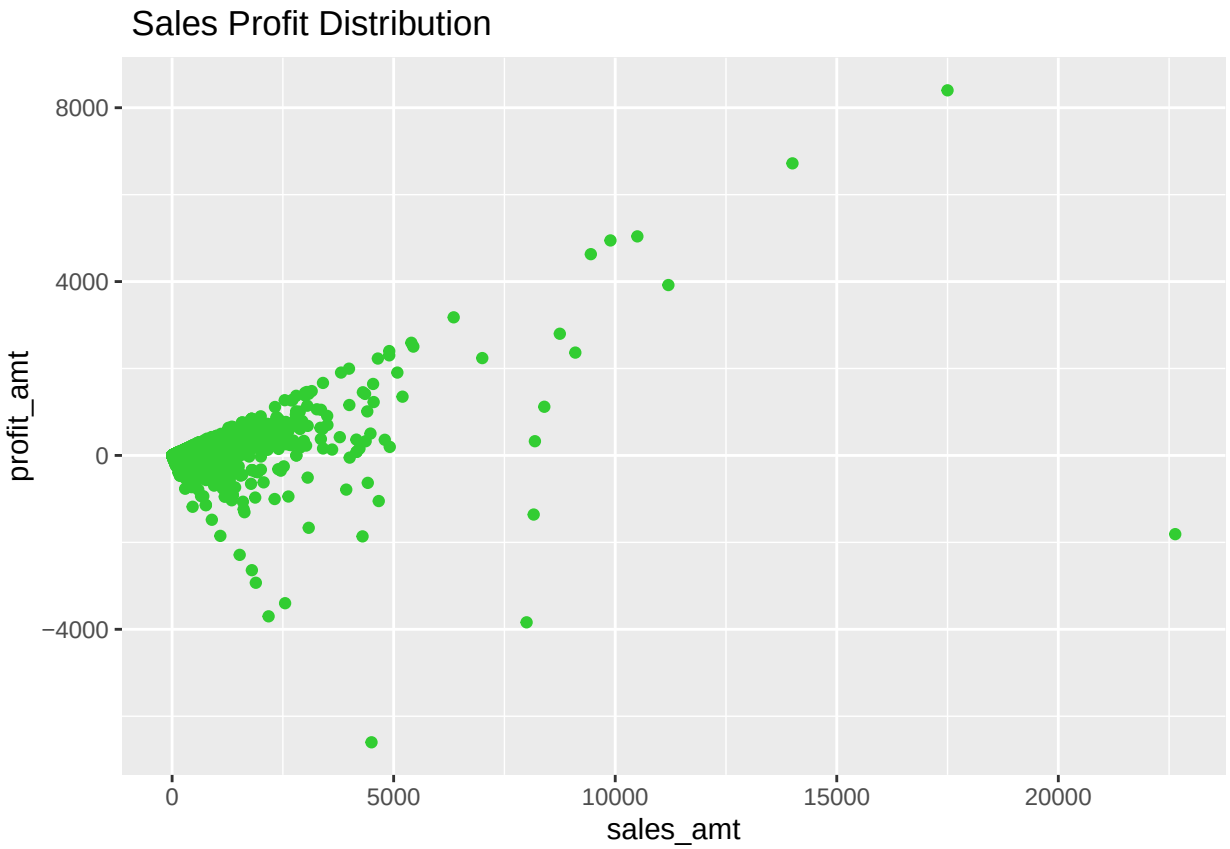
```
ggplot(data=superstore)+
  geom_histogram(mapping=aes(x=discount),fill="cyan3")+
  labs(title=" Discount Distribution")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



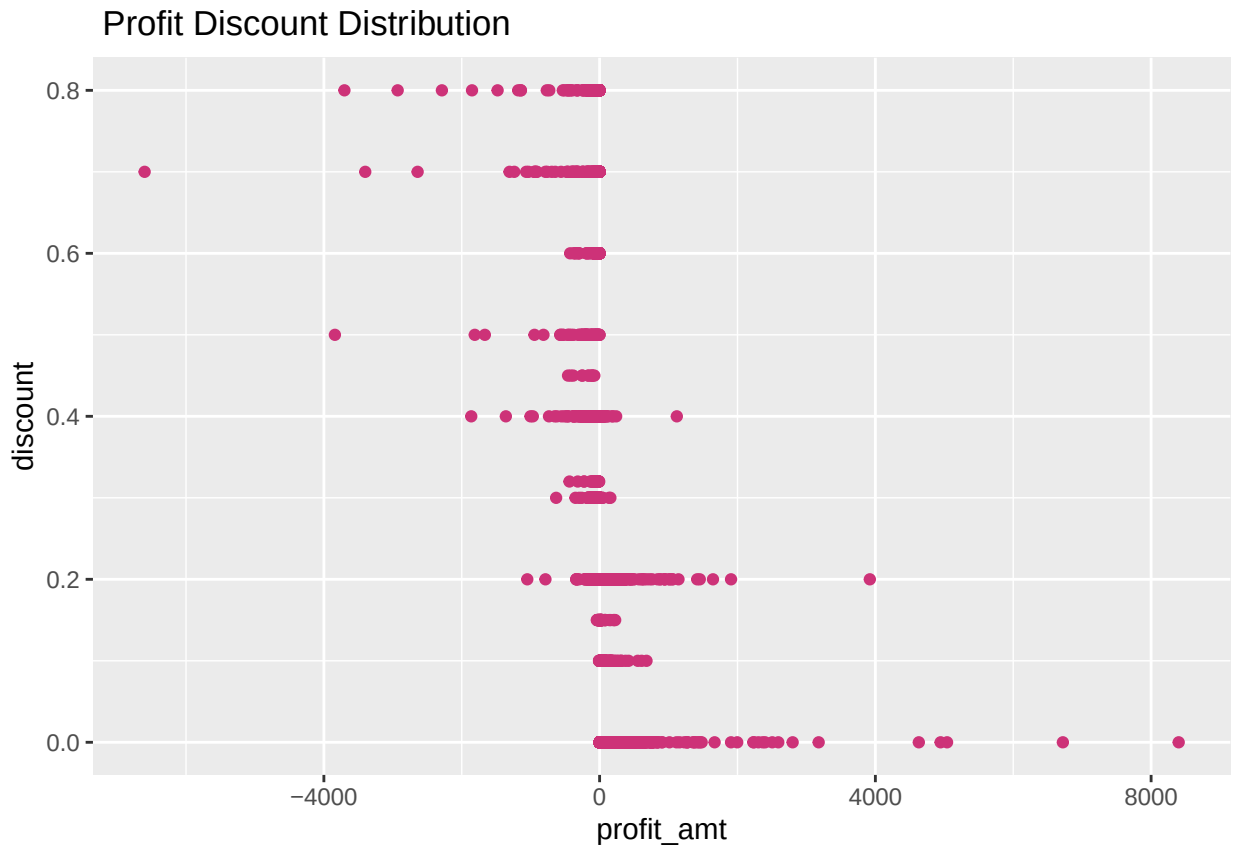
Sales Profit

```
ggplot(data = superstore) +
  geom_point(mapping = aes(x = sales_amt, y = profit_amt),colour="limegreen")+
  labs(title=" Sales Profit Distribution")
```



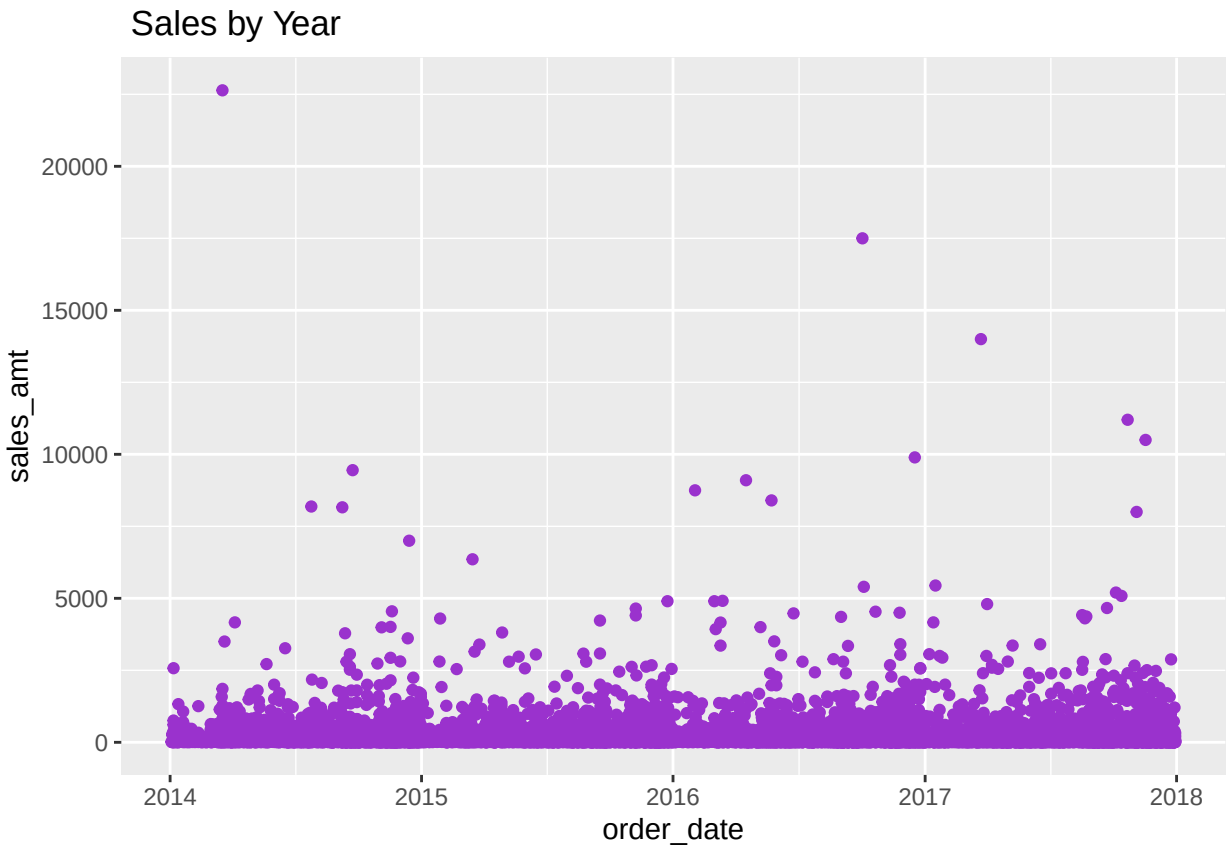
Profit Discount

```
ggplot(data = superstore) +
  geom_point(mapping = aes(x = profit_amt, y = discount),colour="violetred3")+
  labs(title=" Profit Discount Distribution")
```



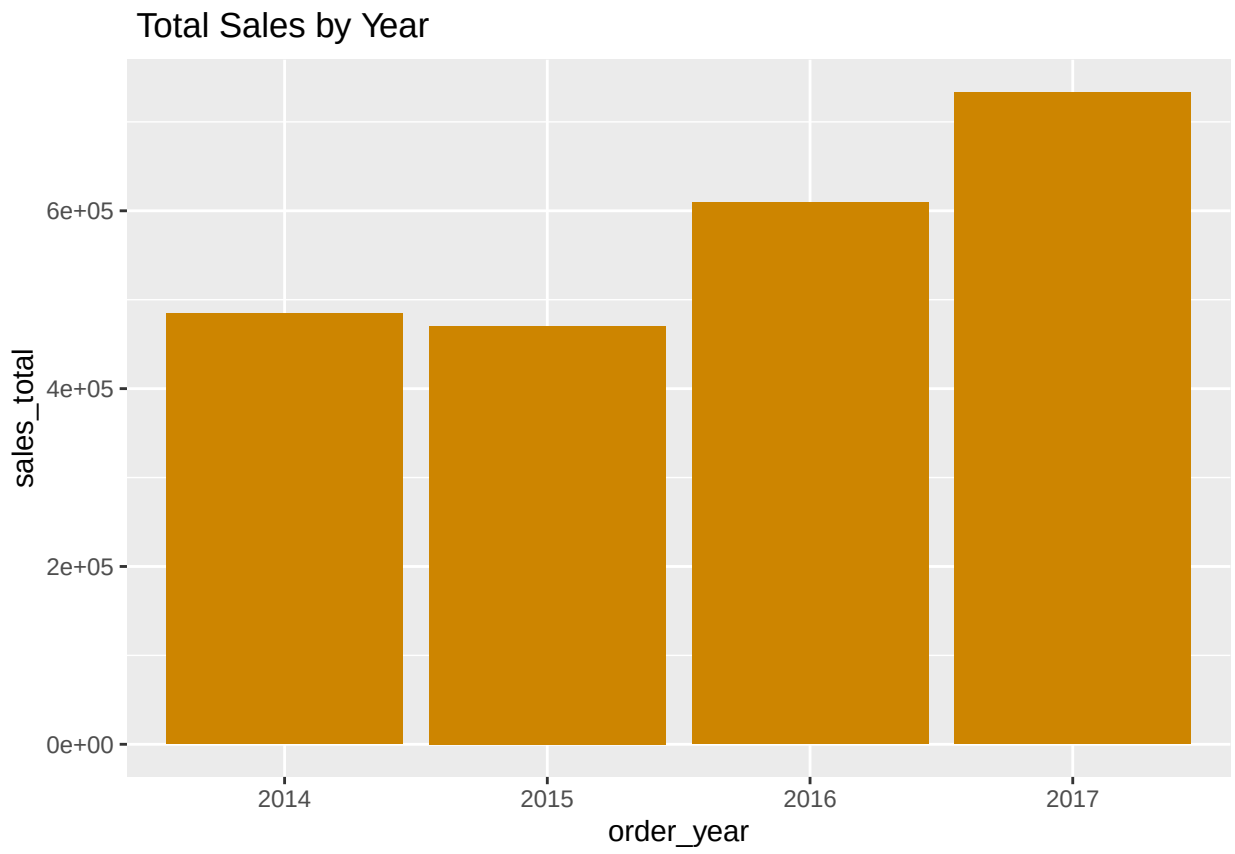
Sales by Year

```
ggplot(data=superstore,aes(x = order_date, y =sales_amt)) +
      geom_point(color = "darkorchid3") +
      labs(title=" Sales by Year")
```



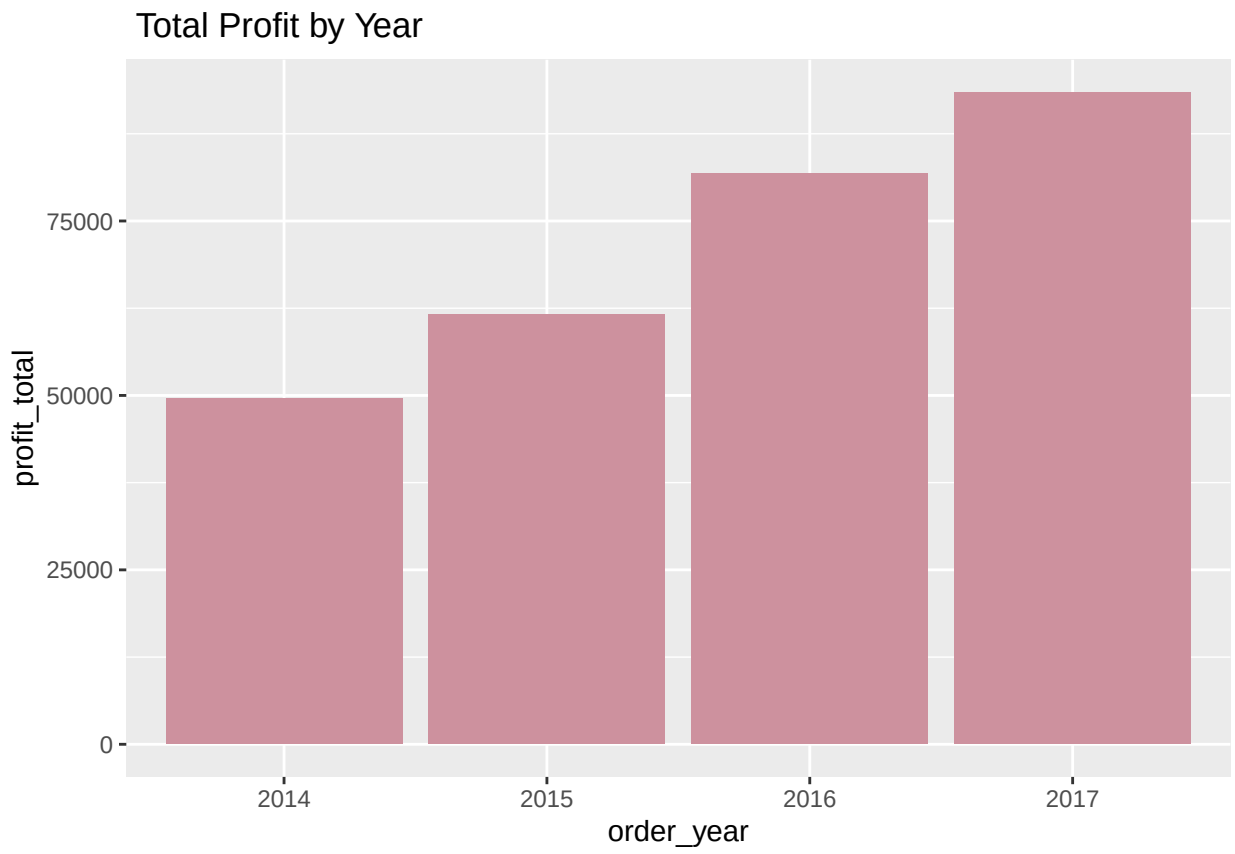
Total Sales by Year

```
sales_year<-aggregate(superstore$sales_amt,by=list(year=format(superstore$order_date, "%Y")),FUN=sum)
names(sales_year)<-c("order_year","sales_total")



ggplot(data=sales_year,aes(x = order_year, y =sales_total)) +
      geom_bar(stat="identity",fill = "orange3") +
      labs(title=" Total Sales by Year")
```



Profit by Year

```
profit_year<-aggregate(superstore$profit_amt,by=list(year=format(superstore$order_date, "%Y")),FUN=sum)
names(profit_year)<-c("order_year","profit_total")


ggplot(data=profit_year,aes(x = order_year, y =profit_total)) +
      geom_bar(stat="identity",fill = "pink3") +
      labs(title=" Total Profit by Year")
```




```
# total product id
count_product_id<-unique(superstore$product_id)
length(count_product_id)
```

```
## [1] 1862
```

```
#total product name
count_product_name<-unique(superstore$product_name)
length(count_product_name)
```

```
## [1] 1850
```

```
#product name and product id mismatch
superstore %>%
  distinct(product_name,product_id) %>%
  group_by(product_id) %>%
  filter(n()>1) %>%
  select(product_id)
```

```
## # A tibble: 64 x 1
## # Groups:   product_id [32]
##    product_id
##    <chr>
##  1 FUR-FU-10004848
```

```
##  2 FUR-CH-10001146
##  3 OFF-BI-10004654
##  4 FUR-CH-10001146
##  5 OFF-PA-10002377
##  6 OFF-AR-10001149
##  7 OFF-PA-10000659
##  8 TEC-MA-10001148
##  9 FUR-FU-10004017
## 10 TEC-AC-10003832
## # ... with 54 more rows
```

```
#total category and subcategory

count_category<-unique(superstore$category)
length(count_category)
```

```
## [1] 3
```

```
count_subcategory<-unique(superstore$sub_category)
length(count_subcategory)
```

```
## [1] 17
```

```
superstore %>%
  distinct(category,sub_category)
```

```
## # A tibble: 17 x 2
##    category        sub_category
##    <chr>           <chr>
##  1 Furniture       Bookcases
##  2 Furniture       Chairs
##  3 Office Supplies Labels
##  4 Furniture       Tables
##  5 Office Supplies Storage
##  6 Furniture       Furnishings
##  7 Office Supplies Art
##  8 Technology      Phones
##  9 Office Supplies Binders
## 10 Office Supplies Appliances
## 11 Office Supplies Paper
## 12 Technology      Accessories
## 13 Office Supplies Envelopes
## 14 Office Supplies Fasteners
## 15 Office Supplies Supplies
## 16 Technology      Machines
## 17 Technology      Copiers
```

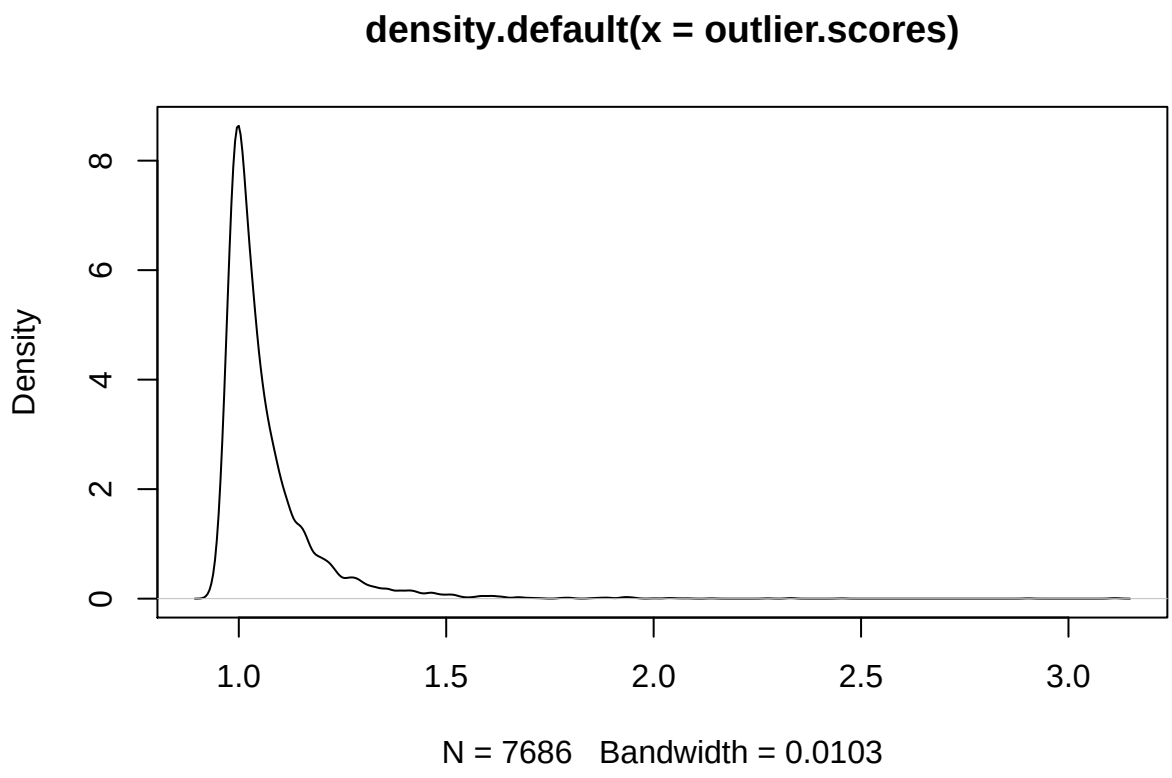
```
superstore_sales<-superstore %>%
                  select(order_date,sales_amt)


superstore_sales<-as_tibble(superstore_sales)
```

```
# superstore_sales_anomalized <- superstore_sales %>%
#     time_decompose(sales_amt, merge = TRUE) %>%
#     anomalize(remainder) %>%
#     time_recompose()
```

Model

Local Outlier Factor Algorithm -Nearest neighbour method

```
#remove duplicates rows
superstore_unq<-superstore[!duplicated(superstore[c("sales_amt","profit_amt","quantity","discount")]),]

#select numerical variables
superstore_lof<-superstore_unq[,c("sales_amt","profit_amt","quantity","discount")]



 # for k=10
 outlier.scores <- lofactor(superstore_lof, k=10)
 plot(density(outlier.scores))
```



Manual Evaluation

```
#top 5 outliers transactons
outliers <- order(outlier.scores, decreasing=T)[1:5]
print(outliers)
```

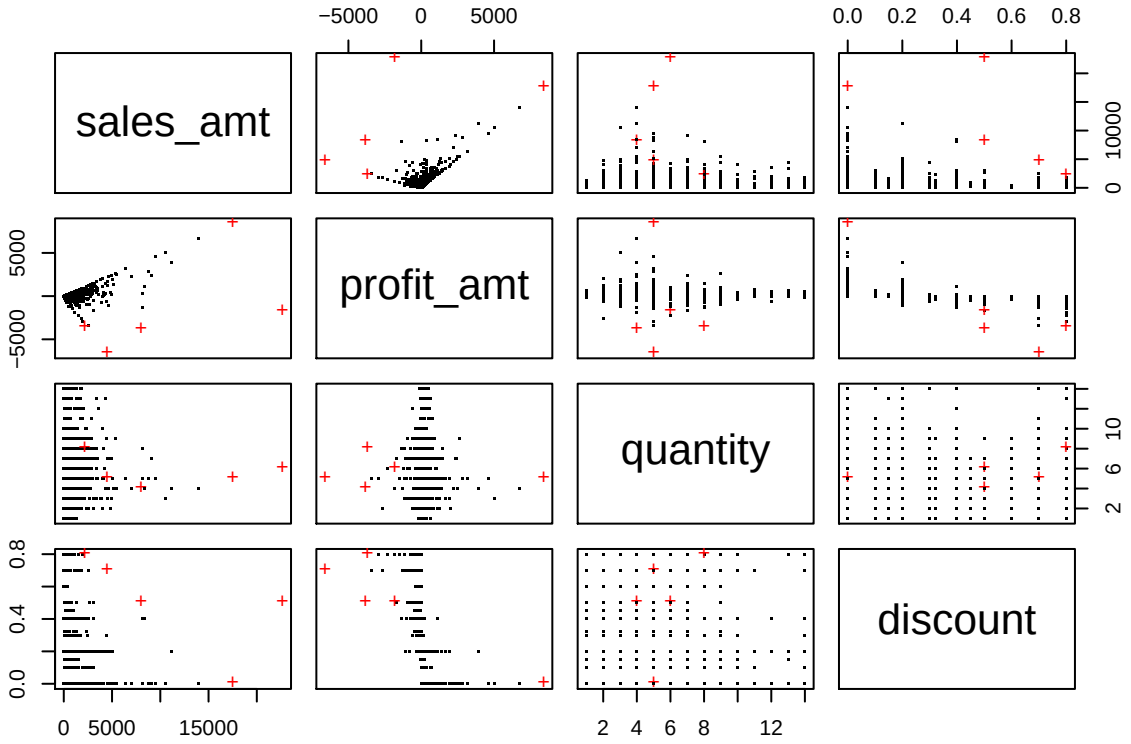
```
## [1] 2452  665 6269 7555 5613
```

```
#dataframe for outliers
superstore_unq[c(2452,665,6269,7555,5613),]
```

```
## # A tibble: 5 x 18
##   orderid order_date ship_date  ship_mode customer_id segment city  state
##   <chr>   <date>     <date>     <chr>     <chr>       <chr>   <chr> <chr>
## 1 CA-201~ 2014-03-18 2014-03-23 Standard~ SM-20320    Home O~ Jack~ Flor~
## 2 US-201~ 2017-11-04 2017-11-04 Same Day  GT-14635    Corpor~ Burl~ Nort~
## 3 CA-201~ 2016-11-25 2016-12-02 Standard~ CS-12505    Consum~ Lanc~ Ohio
## 4 CA-201~ 2014-07-26 2014-07-30 Standard~ LF-17185    Consum~ San ~ Texas
## 5 CA-201~ 2016-10-02 2016-10-09 Standard~ TC-20980    Corpor~ Lafa~ Indi~
## # ... with 10 more variables: postal_code <dbl>, region <chr>,
## #   product_id <chr>, category <chr>, sub_category <chr>, product_name <chr>,
## #   sales_amt <dbl>, quantity <dbl>, discount <dbl>, profit_amt <dbl>
```

Plot LOF outliers

```
pch <- rep(".", 7000)
pch[outliers] <- "+"
col <- rep("black",7000)
col[outliers] <- "red"
pairs(superstore_lof, pch=pch, col=col)
```



Cluster Based Local Outlier Factor

Random Forest Algorithm

Responsible ML Framework

Conclusion

Bibliography