ML 1000 Assignment 2

by Anupama r.k, Queenie Tsang, Crystal (Yunan) Zhu

21/02/2021

To do list:

- **Add Pie charts! - by sub_category, region** # (done)
- **Create a Month variable - to see the change of sales/profits by month?**
- **bar charts of profits/sales by region** #(done)
- **Output the characteristics of the orders with the highest and lowest profits/sales - e.g. what made the order? when? bought what product? in which city/state/region? Any discount?**
- **relationship between discount & sales, discount & profits, sales & profits, and the role of region?**
- **from someone's analysis - there is no significant change between the four discount categories when it comes to Sales**
- **sales/profits by month, rather than by date? color by region?**

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

Objective

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

Data Understanding

US Superstore dataset is sourced from US uperstore 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.

Import data

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

## Warning: package 'dplyr' was built under R version 4.0.3

## Warning: package 'ggplot2' was built under R version 4.0.3

## Warning: package 'anomalize' was built under R version 4.0.3

## Warning: package 'lemon' was built under R version 4.0.3

## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'readr' was built under R version 4.0.3

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

```
superstore<- read_excel("US_Superstore_data.xls")
```

```
## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L2236 / R2236C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L5276 / R5276C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L8800 / R8800C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9148 / R9148C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9149 / R9149C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9150 / R9150C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9388 / R9388C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9389 / R9389C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9390 / R9390C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9391 / R9391C12: '05408'

## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Coercing text to numeric in L9743 / R9743C12: '05408'
```

```r
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

```
##  [1] "ï..Row.ID-0 missing values"     "Order.ID-0 missing values"
##  [3] "Order.Date-0 missing values"    "Ship.Date-0 missing values"
##  [5] "Ship.Mode-0 missing values"     "Customer.ID-0 missing values"
##  [7] "Customer.Name-0 missing values" "Segment-0 missing values"
##  [9] "Country-0 missing values"       "City-0 missing values"
## [11] "State-0 missing values"         "Postal.Code-0 missing values"
## [13] "Region-0 missing values"        "Product.ID-0 missing values"
## [15] "Category-0 missing values"      "Sub.Category-0 missing values"
## [17] "Product.Name-0 missing values"  "Sales-0 missing values"
## [19] "Quantity-0 missing values"      "Discount-0 missing values"
## [21] "Profit-0 missing values"        "diff_in_days-0 missing values"
```

Get a general idea of the data set.

```r
length(unique(data$Customer.ID))
```

```
## [1] 793
```

```r
#793 unique customer IDs
length(unique(data$Customer.Name))
```

```
## [1] 793
```

```r
#793 unique customer names - drop one of these two vars


length(unique(data$Order.Date))
```

```
## [1] 1237
```

```r
#1237 unique order dates
length(unique(data$Ship.Date))
```

```
## [1] 1334
```

```r
#1334 unique ship dates - more unique ship dates than order dates - orders made on the same day were sh

length(unique(data$Segment))
```

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

```r
unique(data$Segment)
```

```
## [1] "Consumer"    "Corporate"   "Home Office"
```

```r
#"Consumer"    "Corporate"   "Home Office"

unique(data$Country)
```

```
## [1] "United States"
```

```r
#all are from US - could drop this variable due to no-variation introduced by it

length(unique(data$City))
```

```
## [1] 531
```

```r
#531 different cities

length(unique(data$State))
```

```
## [1] 49
```

```r
#49 states

length(unique(data$Postal.Code))
```

```
## [1] 631
```

```r
#631 postal code - 793 unique customer IDs - some customers live very close!

unique(data$Region)
```

```
## [1] "South"   "West"    "Central" "East"
```

```r
#only 4 regions

unique(data$Category)
```

```
## [1] "Furniture"       "Office Supplies" "Technology"
```

```r
#only 3 categories - "Furniture" "Office Supplies" "Technology"

length(unique(data$Sub.Category))
```

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

```r
unique(data$Sub.Category)
```

```
##  [1] "Bookcases"   "Chairs"      "Labels"      "Tables"      "Storage"
##  [6] "Furnishings" "Art"         "Phones"      "Binders"     "Appliances"
## [11] "Paper"       "Accessories" "Envelopes"   "Fasteners"   "Supplies"
## [16] "Machines"    "Copiers"
```

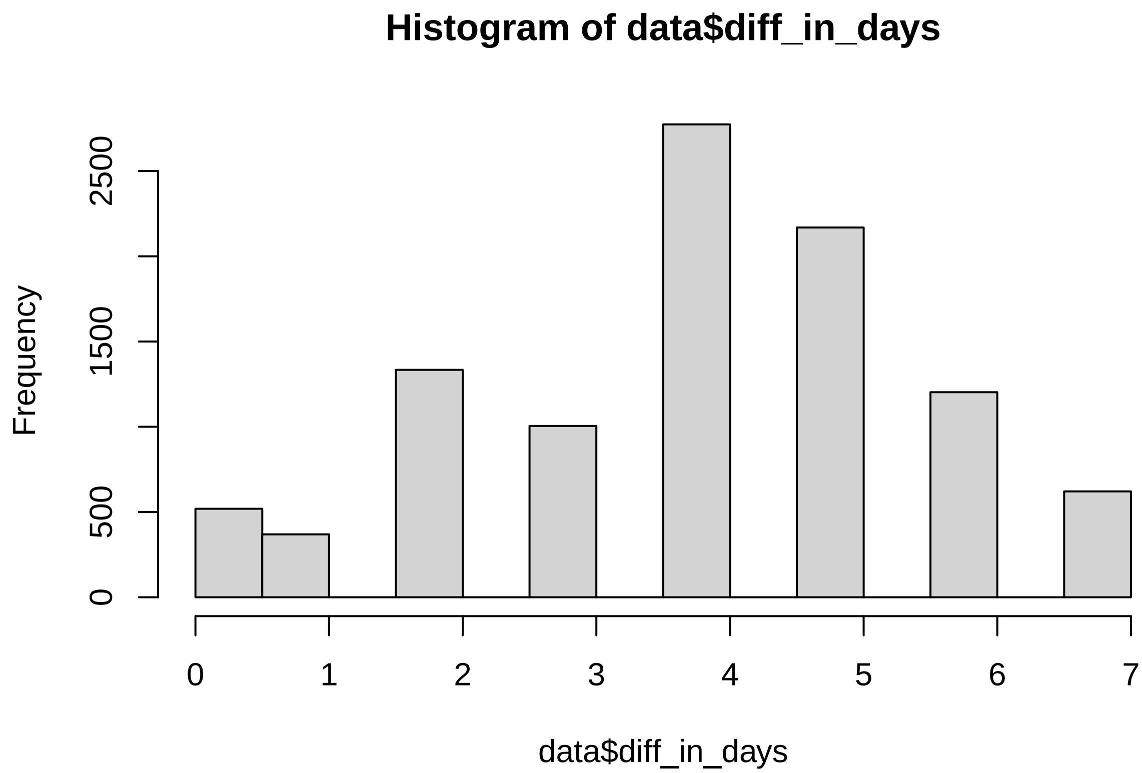
```r
#17 sub-categories

length(unique(data$Product.Name))
```

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

```r
#1850 product names
length(unique(data$Product.ID))
```

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

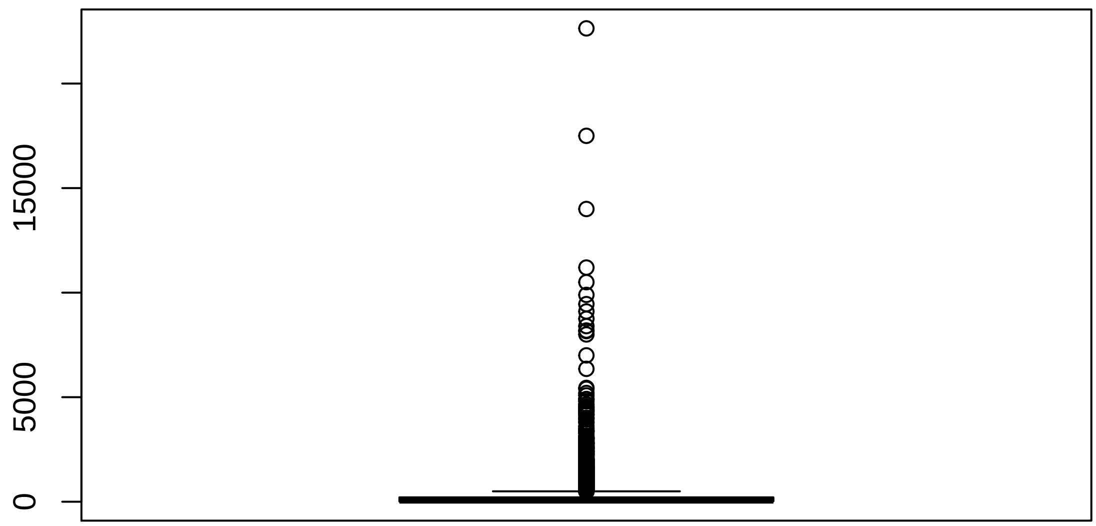
```r
#1862 product IDs - potential redundant variables!

hist(data$diff_in_days)
```



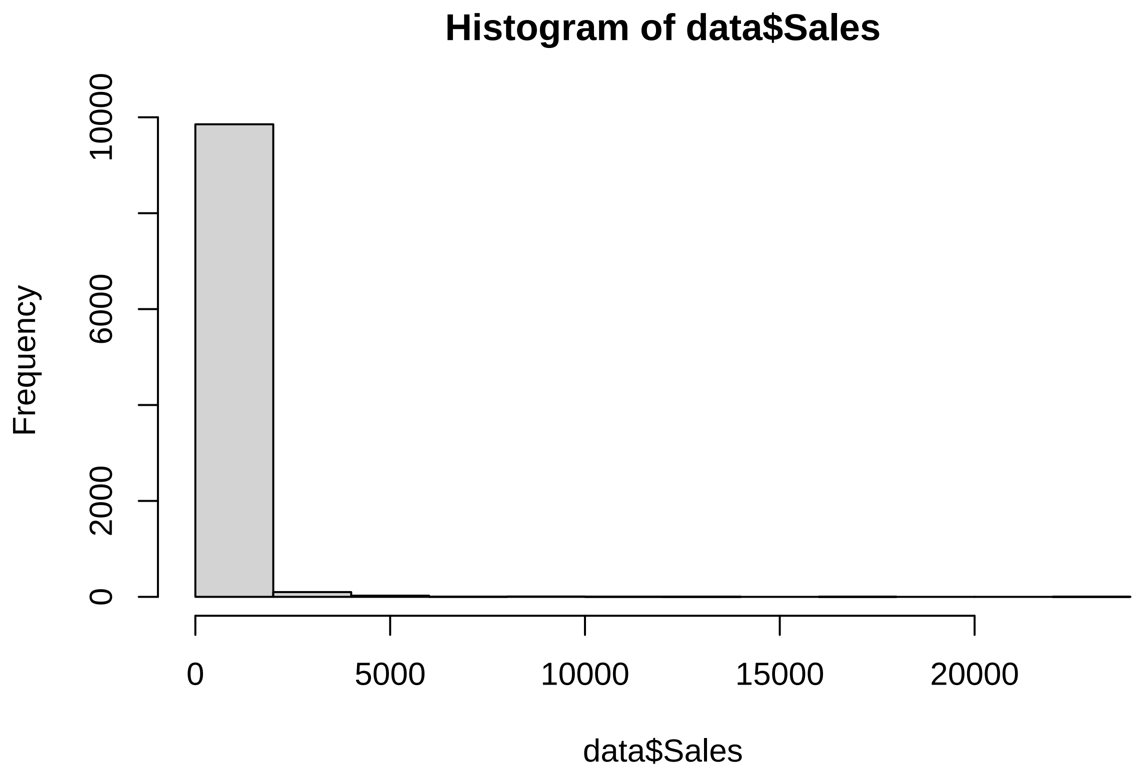

```r
#The time difference between order date and ship date typically takes 4 days.

summary(data$Sales)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     0.444    17.280    54.490   229.858   209.940 22638.480
```

```r
boxplot(data$Sales)
```



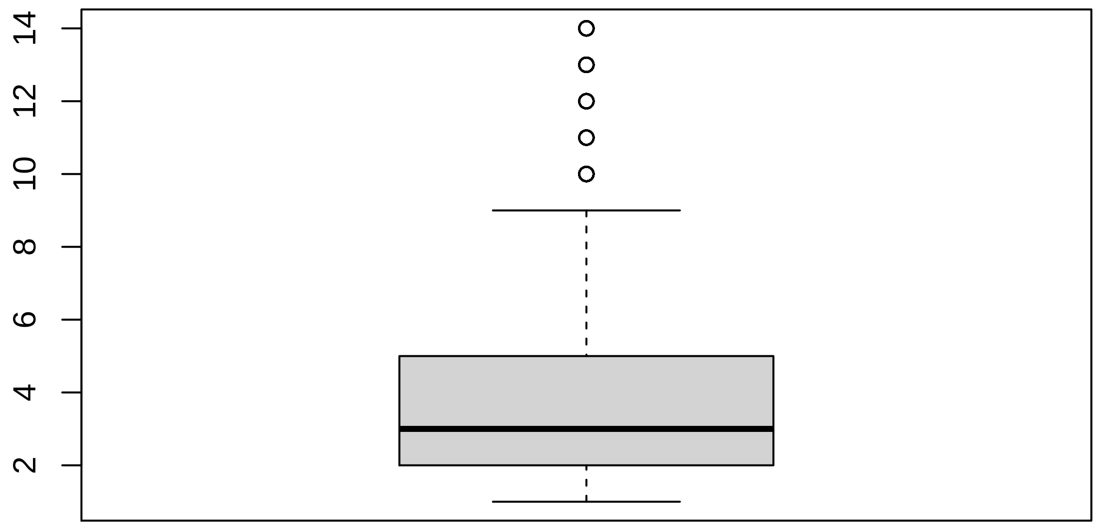

```r
hist(data$Sales)
```



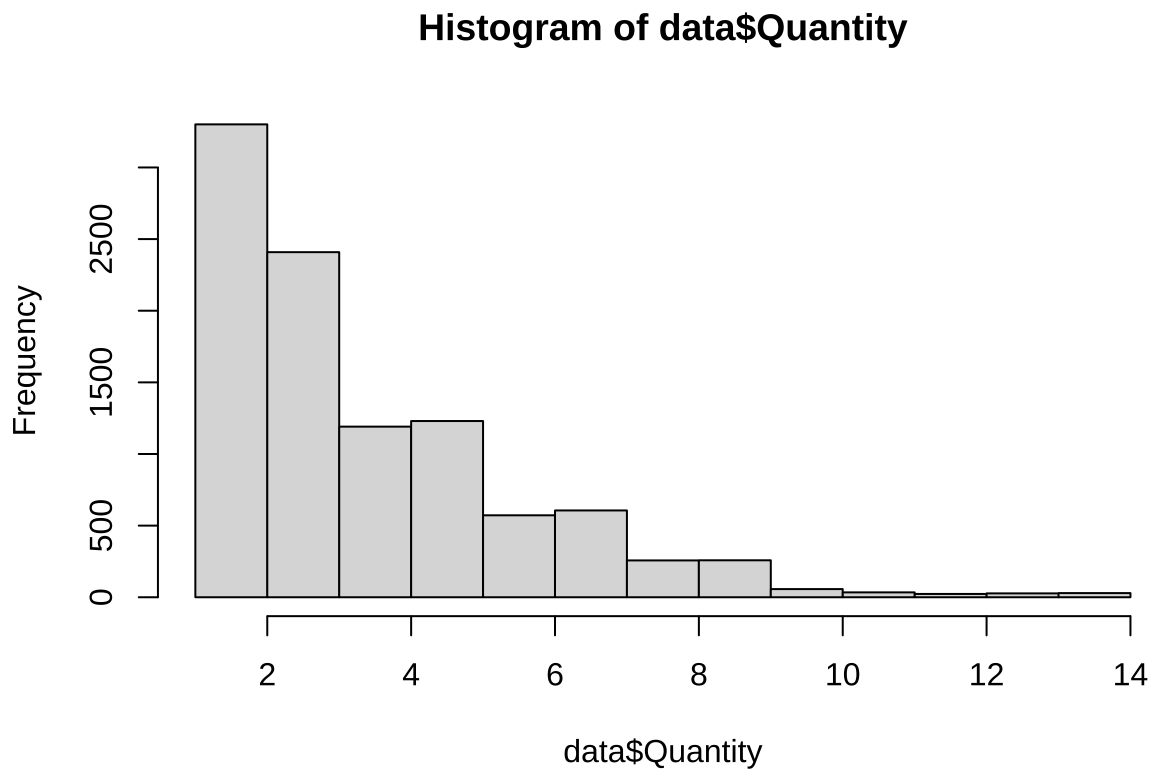

```r
#a large amount of orders with very small Sales!


summary(data$Quantity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    2.00    3.00    3.79    5.00   14.00
```

```r
boxplot(data$Quantity)
```



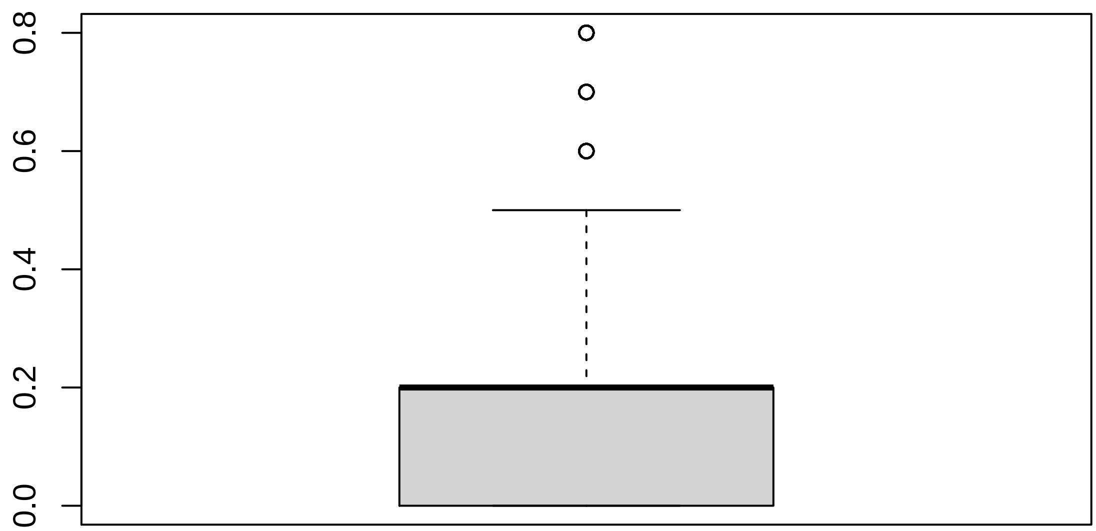

```r
#not many outliers - the #of products in each order is stable?
hist(data$Quantity)
```



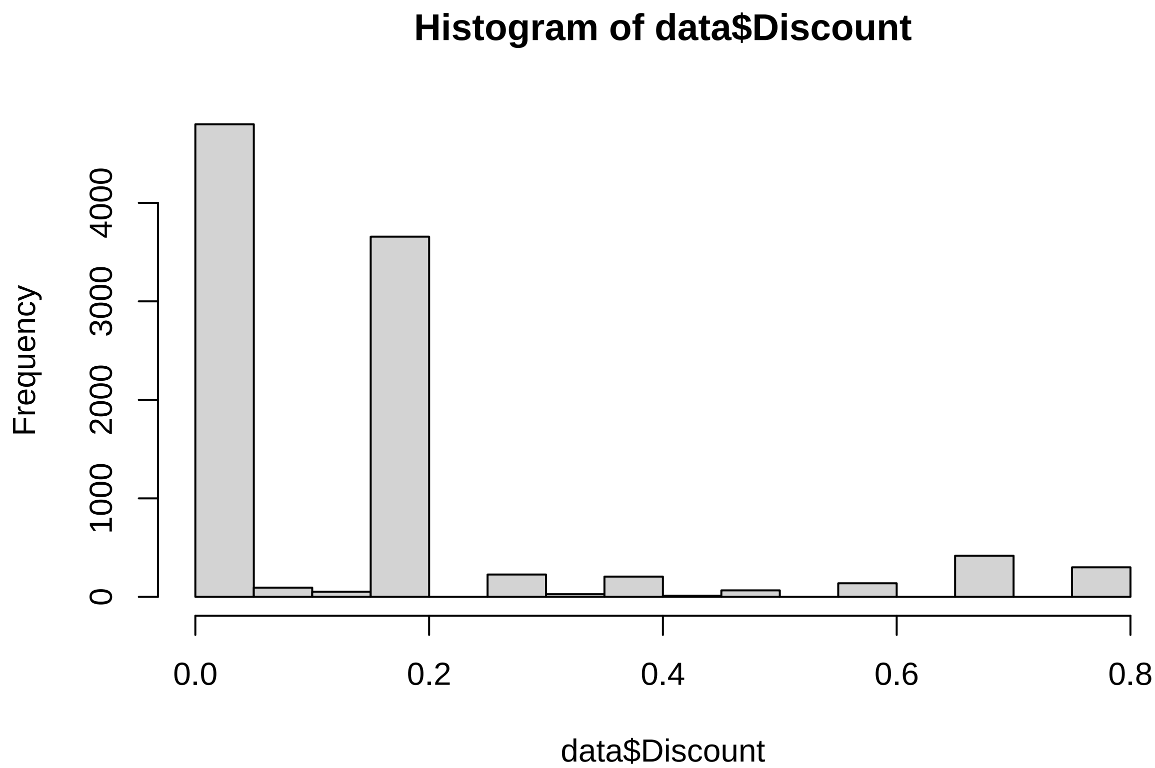

```r
#very skewed distribution - most of the orders have small #of items

summary(data$Discount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.2000  0.1562  0.2000  0.8000
```

```r
boxplot(data$Discount)
```



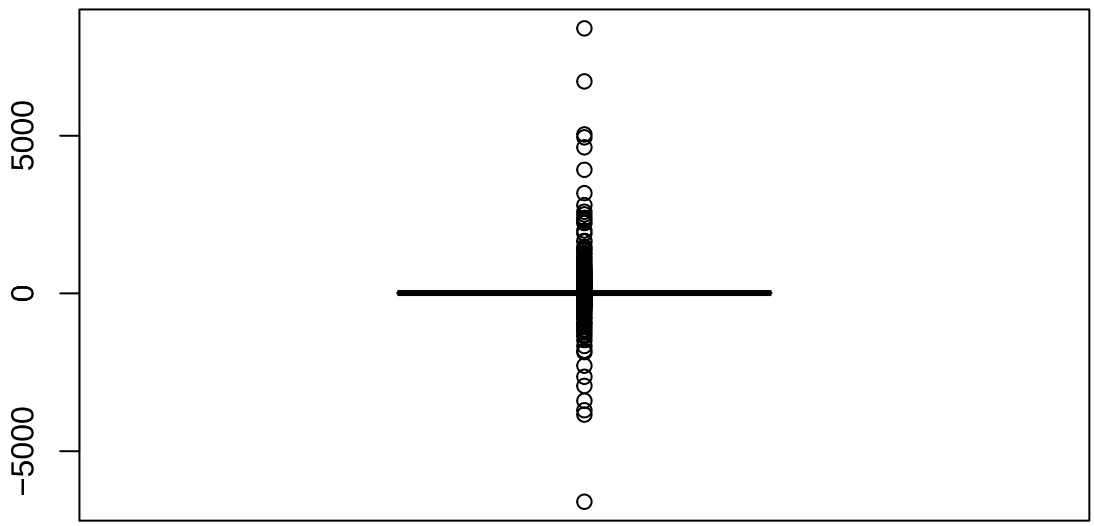

```
#a strange looking box dataplot? - median & 3rd quantile are the same (0.2) - not many orders have high
hist(data$Discount)
```



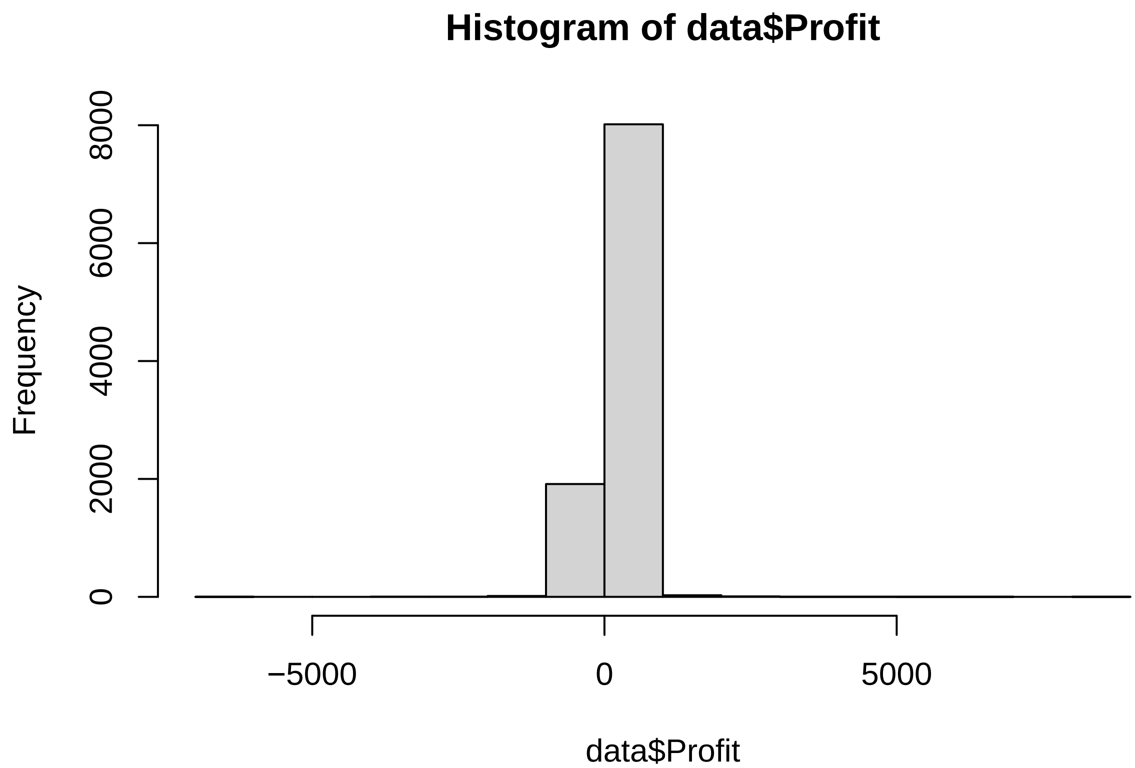

```r
#most of the orders were placed without any discounts or with 20% off

summary(data$Profit)
```

```
##       Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -6599.978     1.729     8.666    28.657    29.364  8399.976
```

```r
boxplot(data$Profit)
```




```r
#most of the profits are outside of the box - but most of them clustered close to the box(not with so e
hist(data$Profit)
```



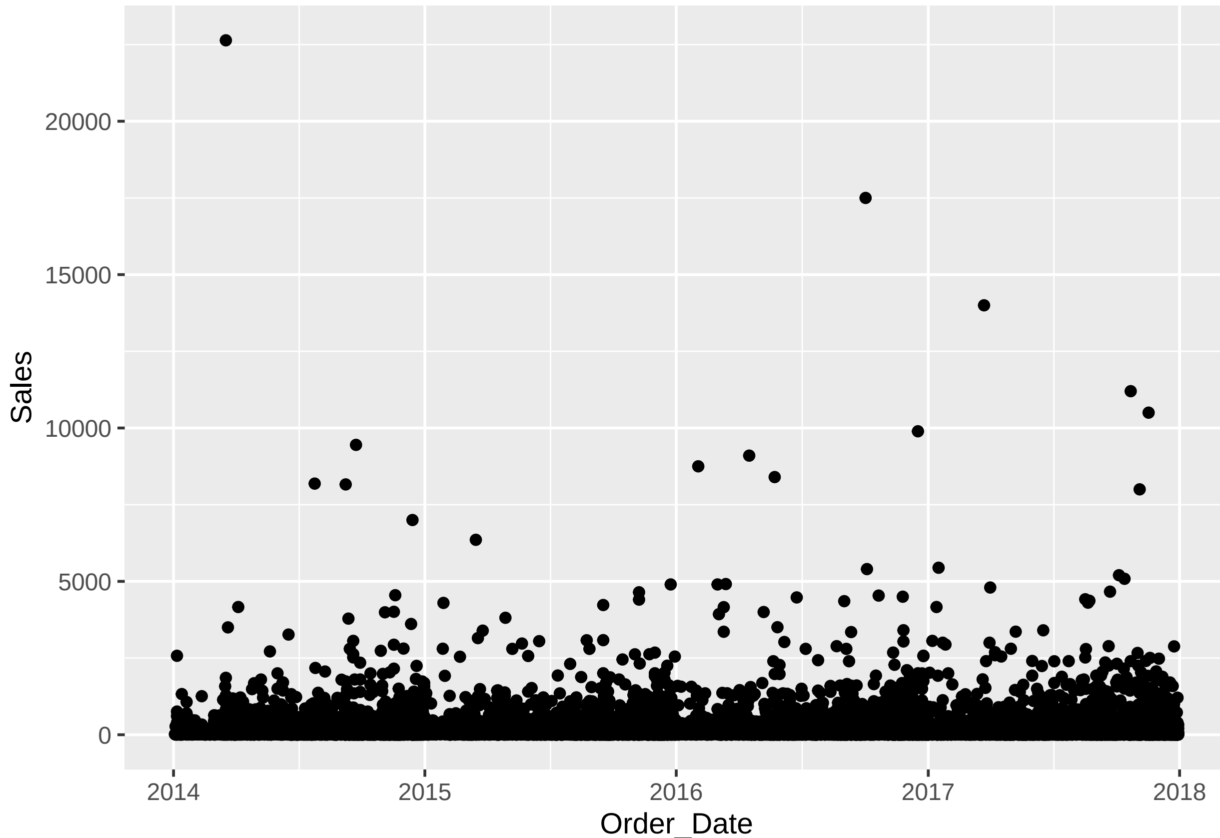

```
#most of the orders have profits ~1000 (or ~800?), and ~ -800
```

Remove the dot in the column names and replace with "_" to make variable names easier to handle:

```
##  [1] "ï__Row_ID"     "Order_ID"      "Order_Date"    "Ship_Date"
##  [5] "Ship_Mode"     "Customer_ID"   "Customer_Name" "Segment"
##  [9] "Country"       "City"          "State"         "Postal_Code"
## [13] "Region"        "Product_ID"    "Category"      "Sub_Category"
## [17] "Product_Name"  "Sales"         "Quantity"      "Discount"
## [21] "Profit"        "diff_in_days"
```

Exploratory Data Analysis

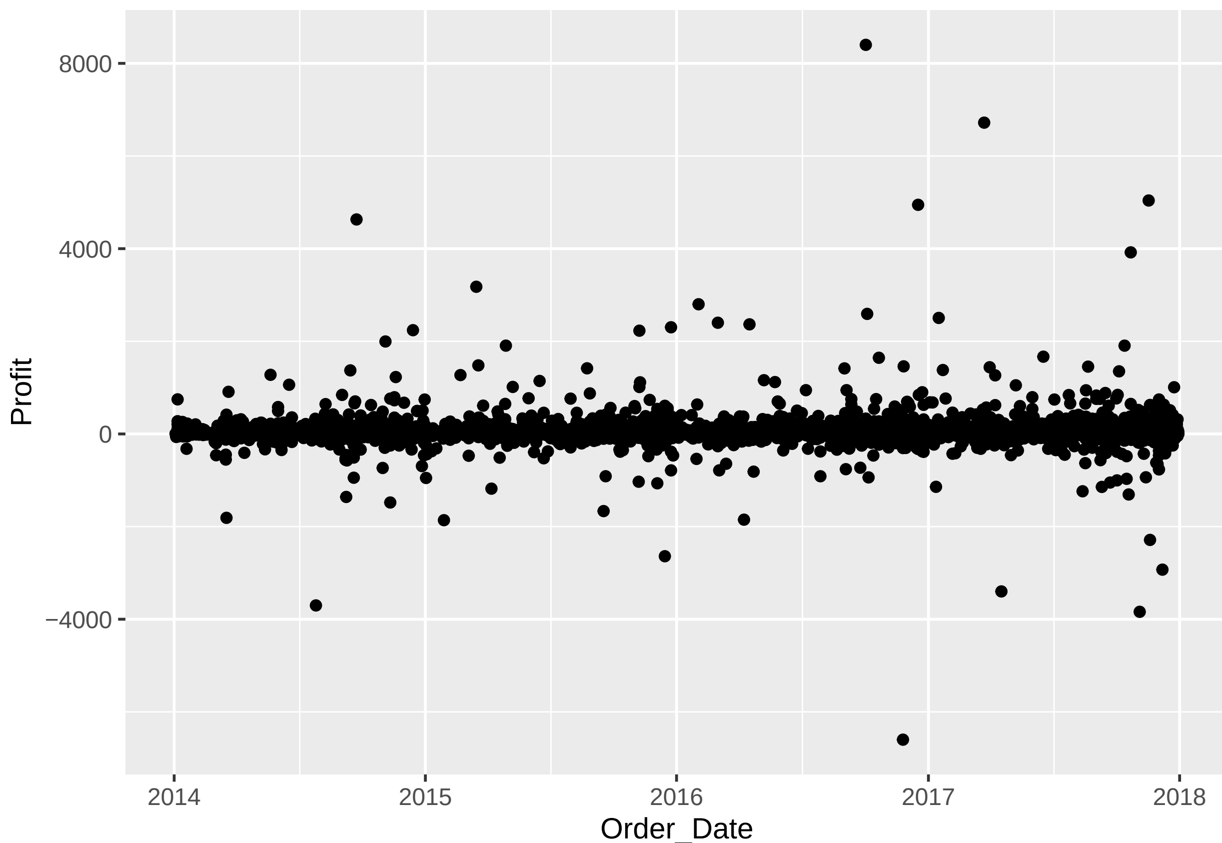
Plot Sales in relation to Order Date:



Plot Profit in relation to Order Date:

```
ggplot(data = data) +
  geom_point(mapping = aes(x = Order_Date, y = Profit), xlab="Order Date", ylab="Profit")
```

```
## Warning: Ignoring unknown parameters: xlab, ylab
```



Some outliers for certain days

```r
table(data$`Sub_Category`)
```

```
##
## Accessories  Appliances         Art     Binders   Bookcases      Chairs
##         775         466         796        1523         228         617
##     Copiers   Envelopes   Fasteners Furnishings      Labels    Machines
##          68         254         217         957         364         115
##       Paper      Phones     Storage    Supplies      Tables
##        1370         889         846         190         319
```

look at the time range for these transactions, ie. start date for Order_Date column:

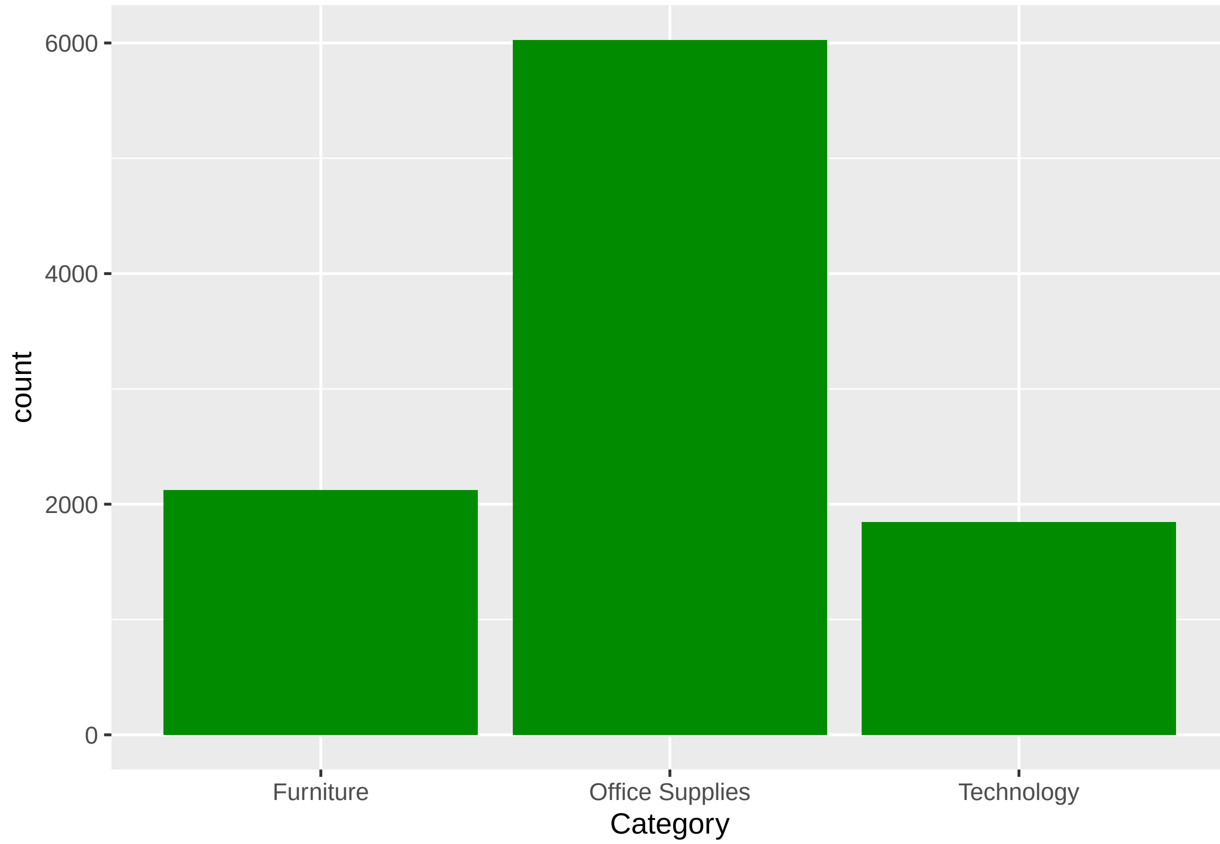
```r
summary(data$Order_Date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2014-01-03" "2015-05-23" "2016-06-26" "2016-04-30" "2017-05-14" "2017-12-30"
```

```r
#[1] min "2014-01-03", max "2017-12-30"
```

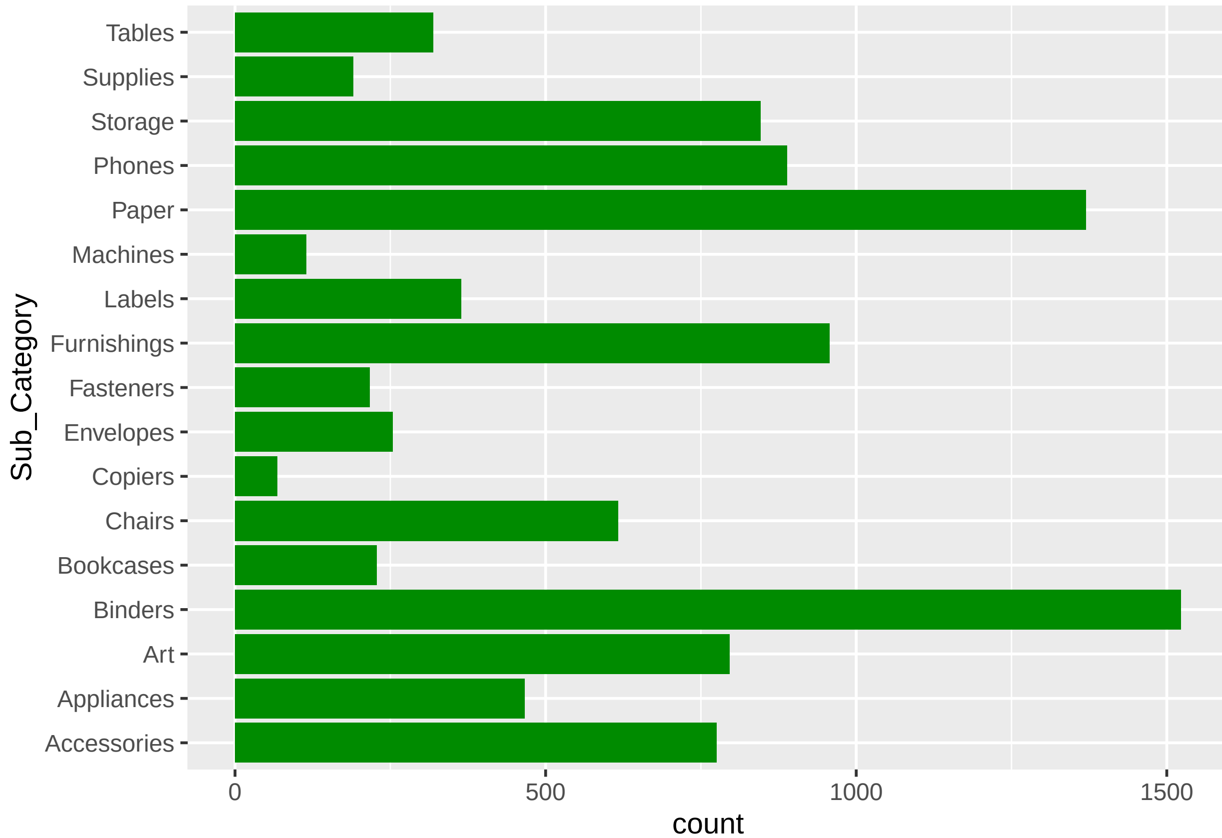
Basically this dataset covers transactions ranging from 2014-01-03 to 2017-12-30.

```
ggplot(data = data) +
  geom_bar(mapping = aes(x = Category),fill="green4")
```



Most type of products sold belong to the Office supplies category.

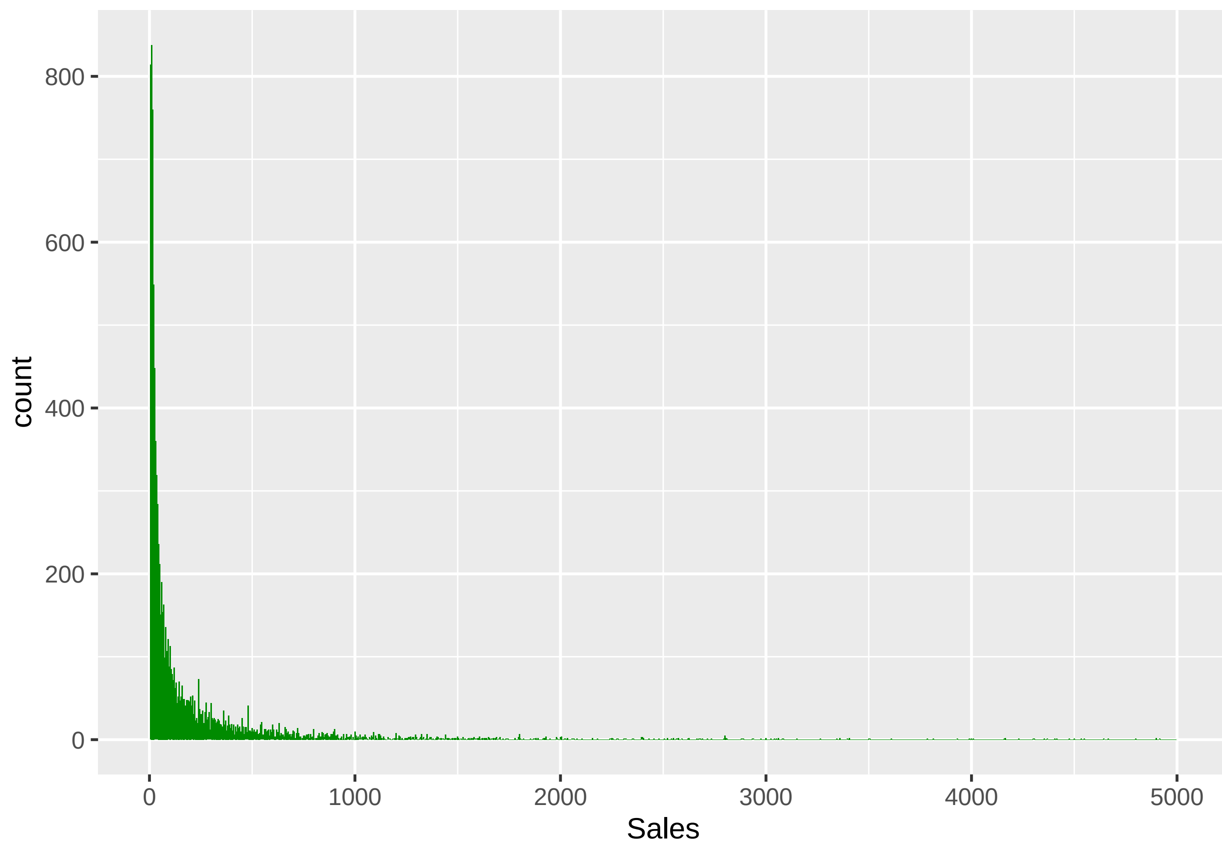
```
ggplot(data = data) +
  geom_bar(mapping = aes(y = `Sub_Category`), fill="green4")
```




```
ggplot(data = data, mapping = aes(x = Sales)) +
  xlim(0, 5000) +
  geom_histogram(binwidth = 5,fill="green4")
```

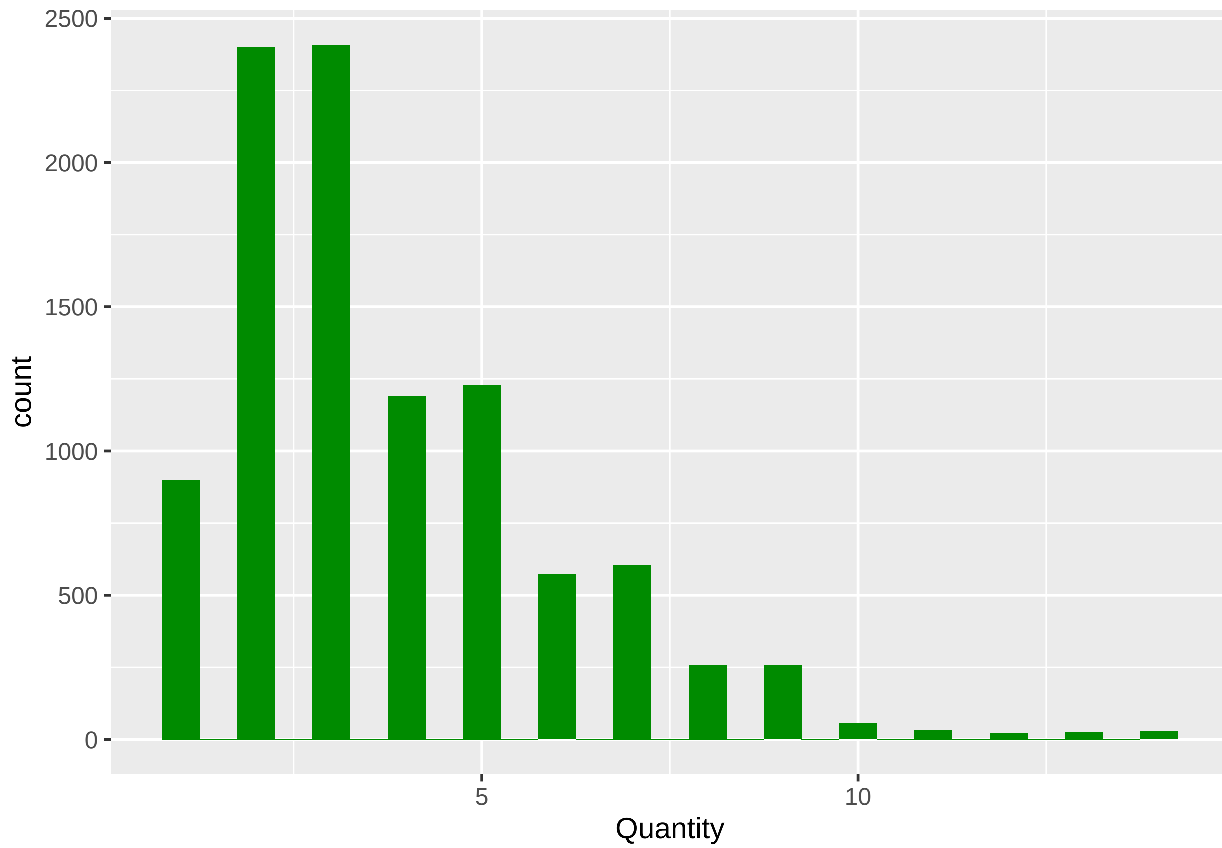
```
## Warning: Removed 19 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



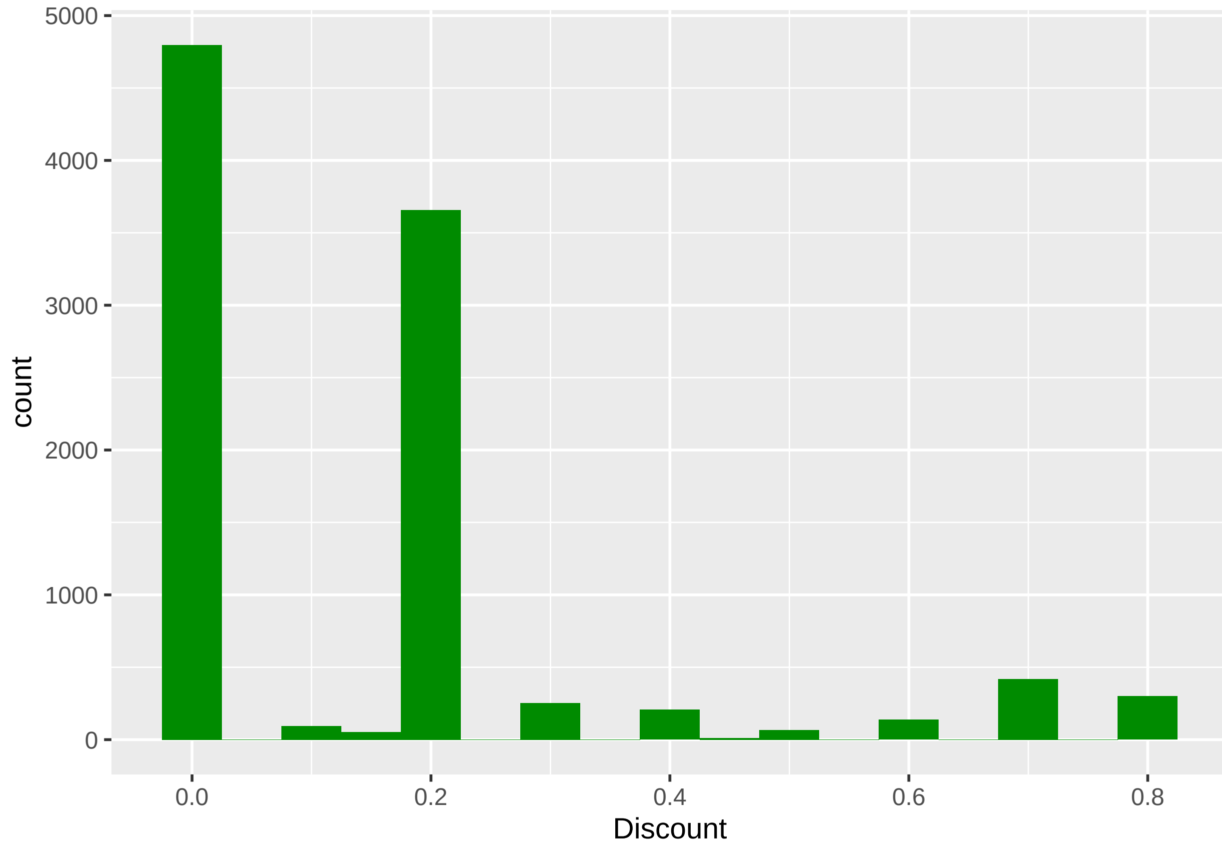
Most sales are very few items (<500).

```r
ggplot(data = data, mapping = aes(x = Quantity)) +
  geom_histogram(binwidth = 0.5,fill="green4")
```




```
ggplot(data = data) +
  geom_histogram(mapping = aes(x = Discount),
                 binwidth = 0.05,
                 xlab="Discount",
                 fill="green4")
```

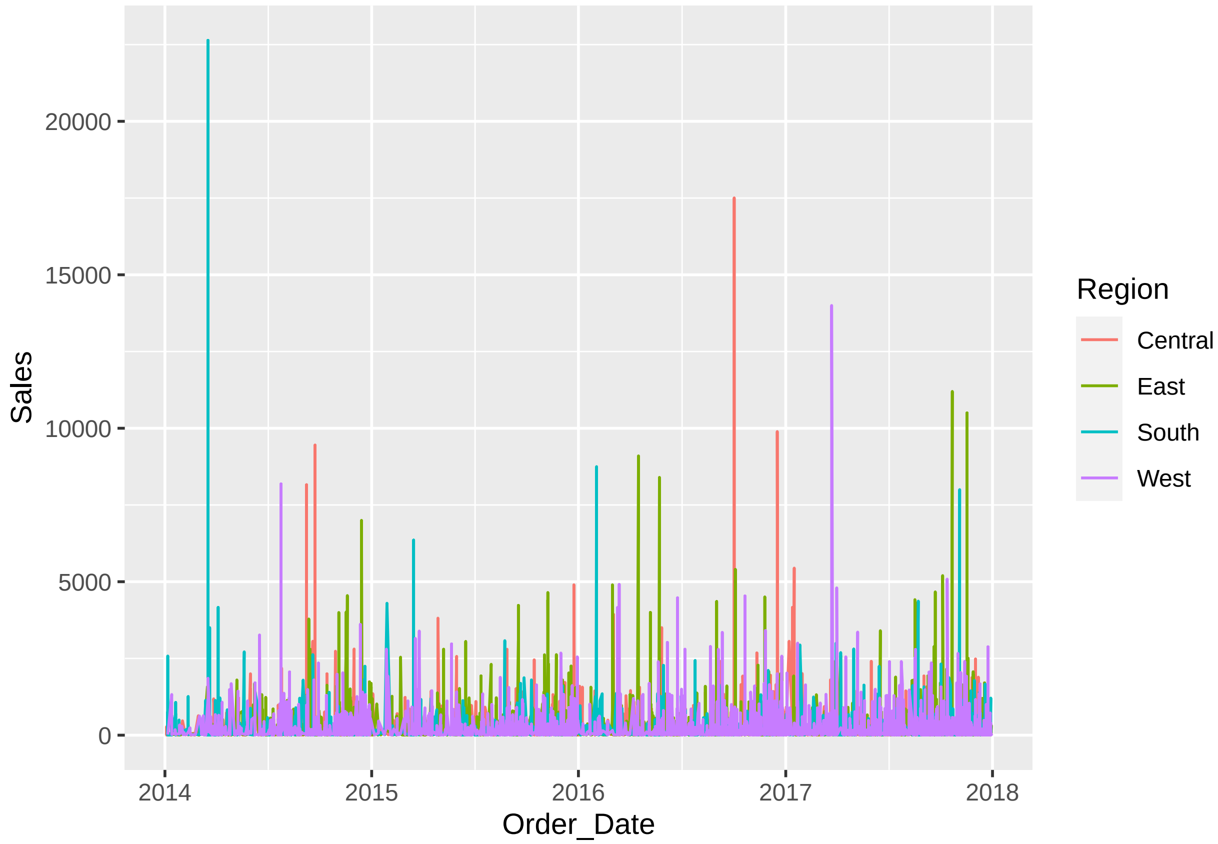
```
## Warning: Ignoring unknown parameters: xlab
```



Sales transactions mostly do not involve discounts.

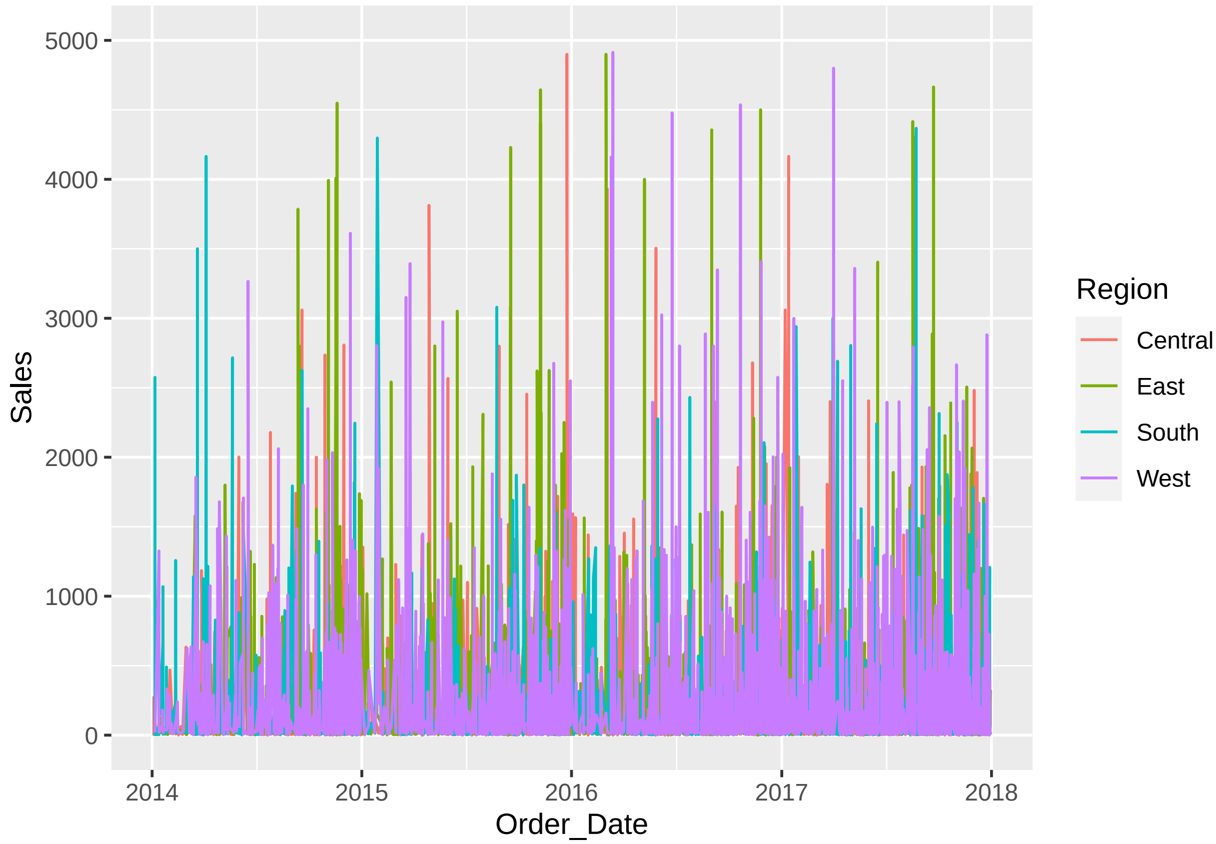
Visualise sales transactions by Region over time (order date).

```
ggplot(data, aes(Order_Date, Sales,color=Region)) +
     geom_line()
```



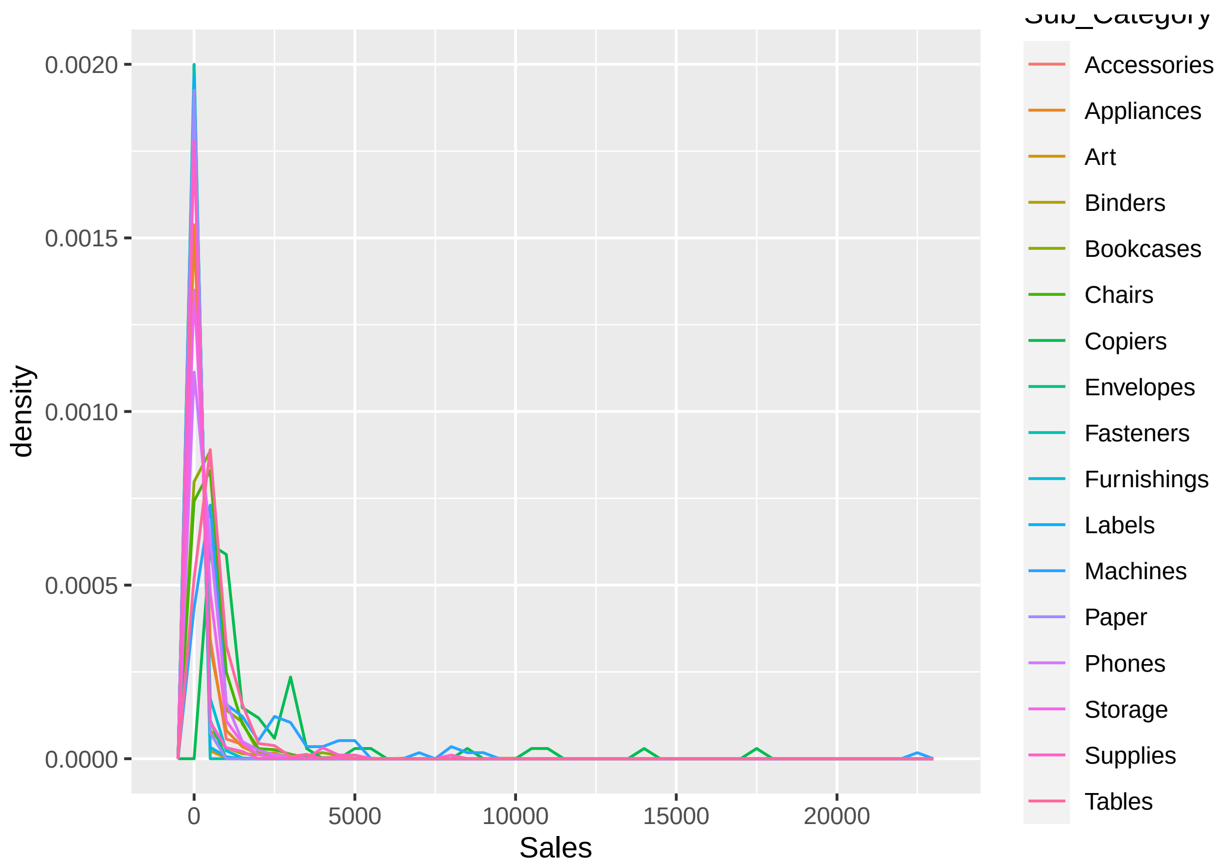
Let's zoom in a little bit - Visualise sales transactions by Region over time (order date).

```
ggplot(data, aes(Order_Date, Sales,color=Region)) +
     geom_line() +
     ylim(0,5000)
```



How does profit change with sub-category?

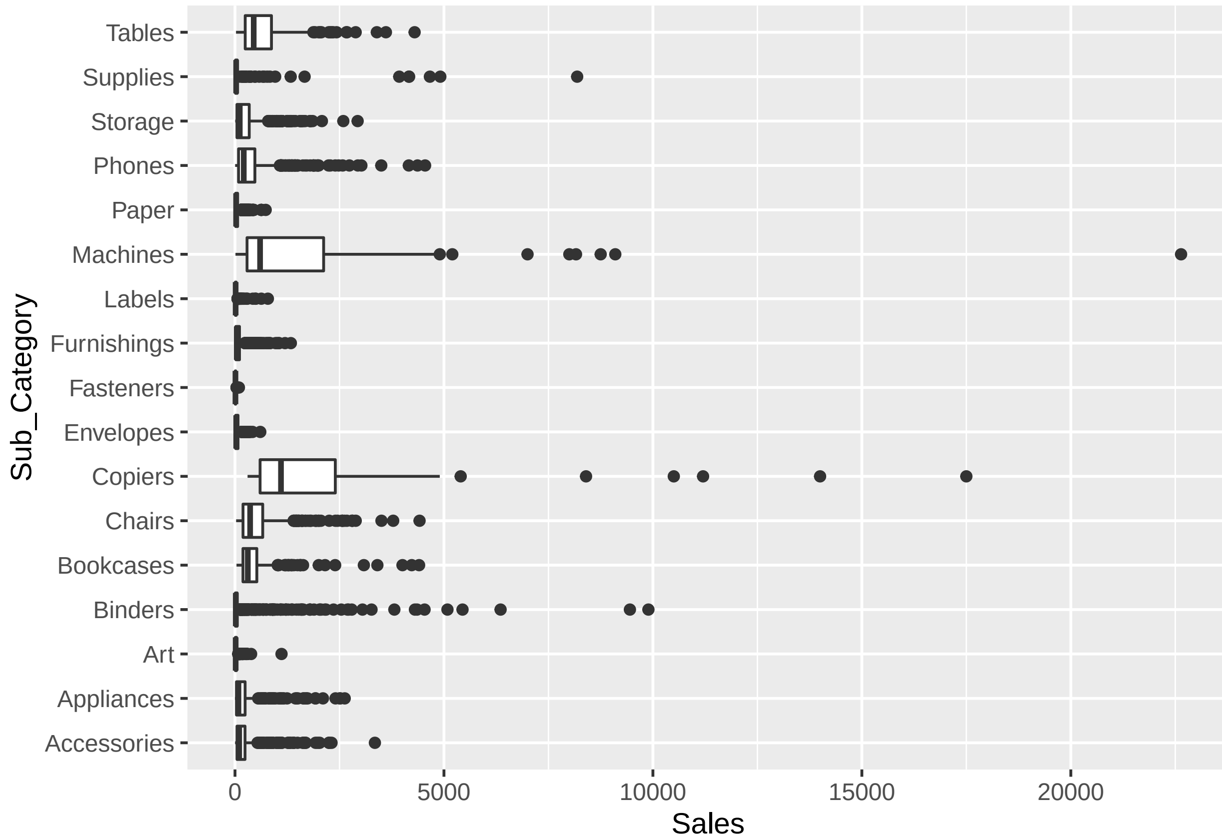
```
#density plot where the count is standardized,area under each frequency is 1
ggplot(data = data, mapping = aes(x = Sales, y = ..density..)) +
  geom_freqpoly(mapping = aes(colour = Sub_Category), binwidth = 500)
```



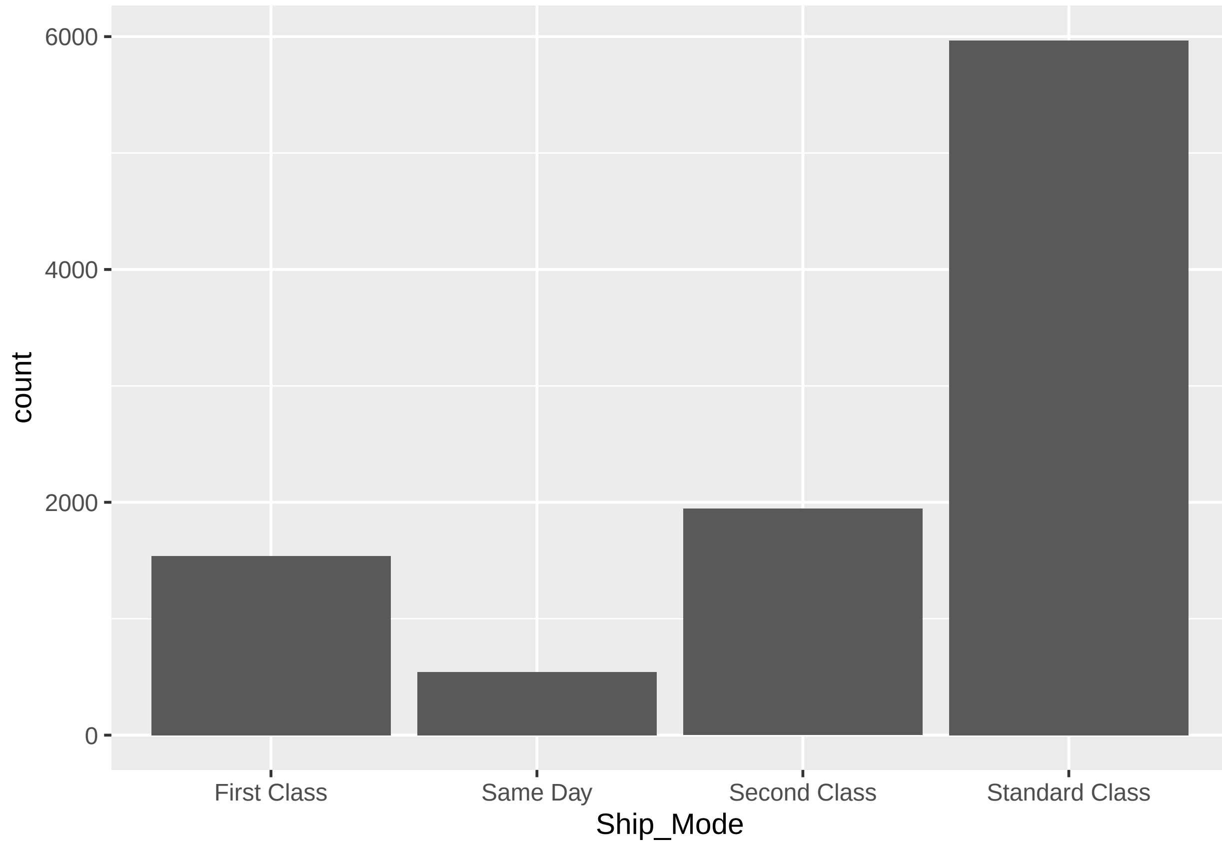
It looks like some categories of items ie. supplies or accessories have negative sales values.

How does sales vary across sub category?

```
ggplot(data = data, mapping = aes(x = Sales, y = `Sub_Category` )) +
  geom_boxplot()
```



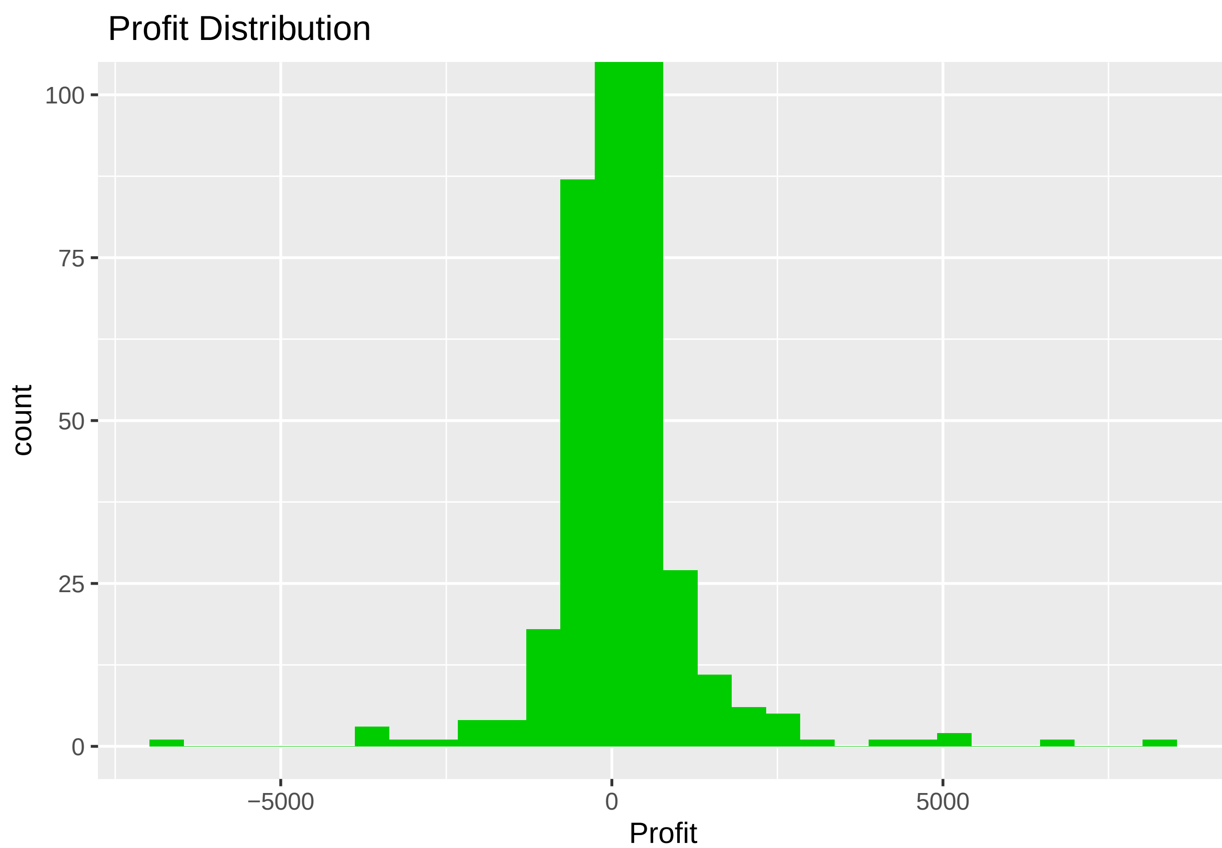

```
ggplot(data =data, mapping = aes(x = Ship_Mode)) +
  geom_bar()
```



Most transactions are shipped via Standard Class method.

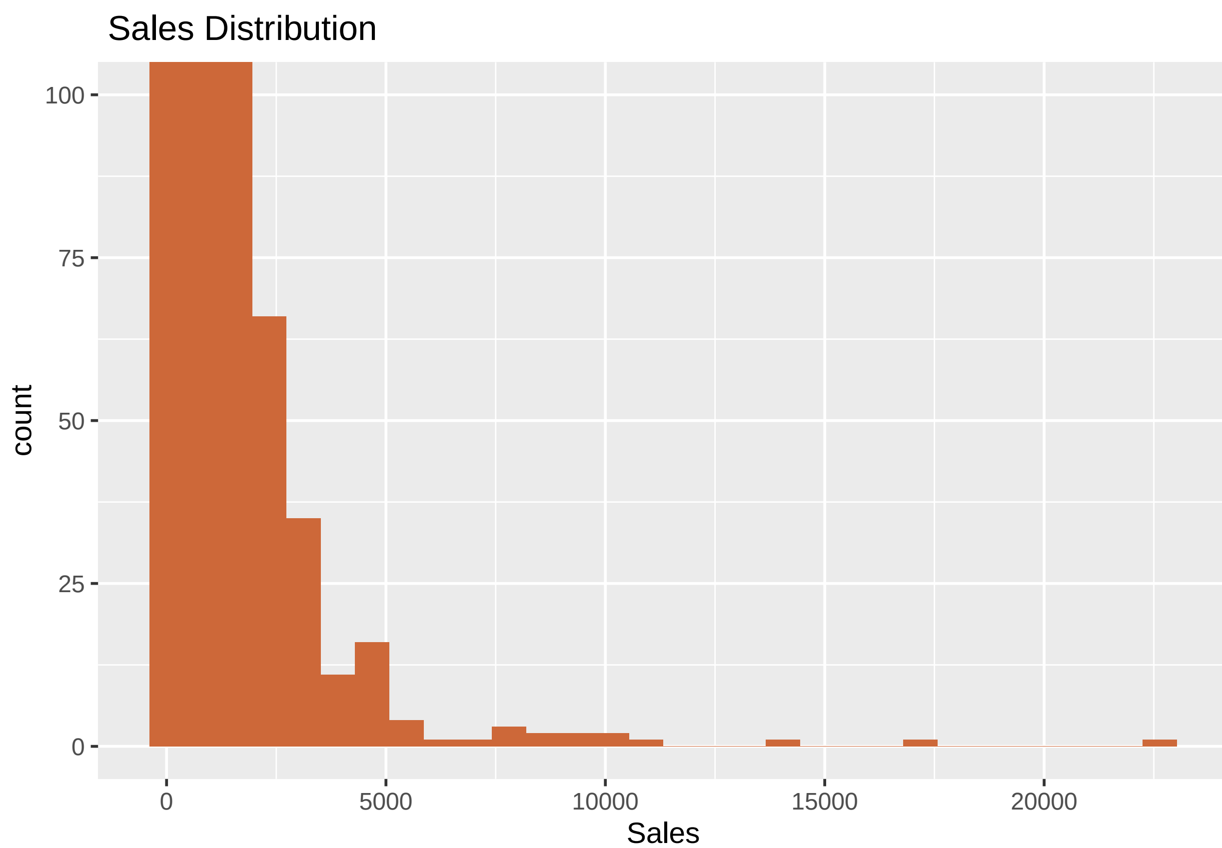
```
ggplot(data)+
geom_histogram(mapping=aes(x=Profit),fill="green3")+
coord_cartesian(ylim = c(0, 100))+
labs(title=" Profit Distribution")
```

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

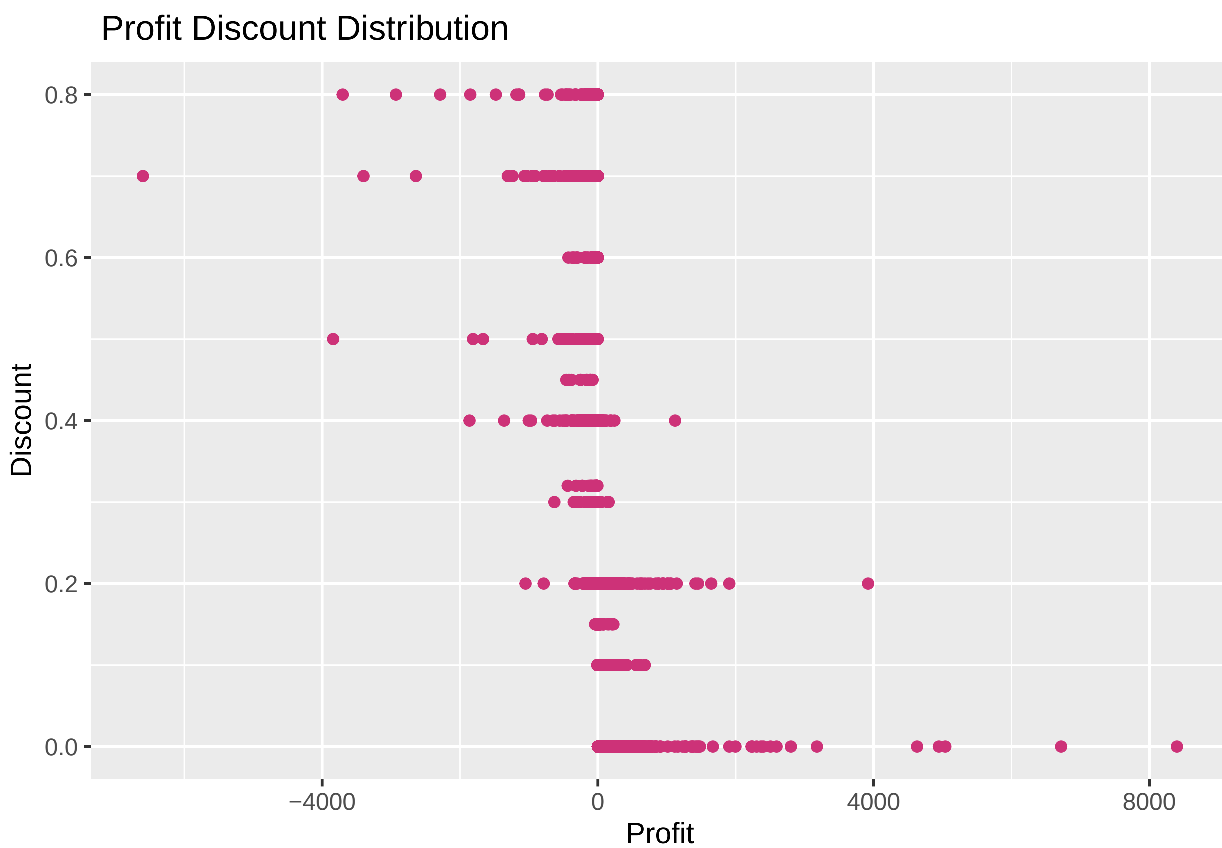



```
ggplot(data)+
geom_histogram(mapping=aes(x=Sales),fill="sienna3")+
coord_cartesian(ylim = c(0, 100))+labs(title=" Sales Distribution")
```

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

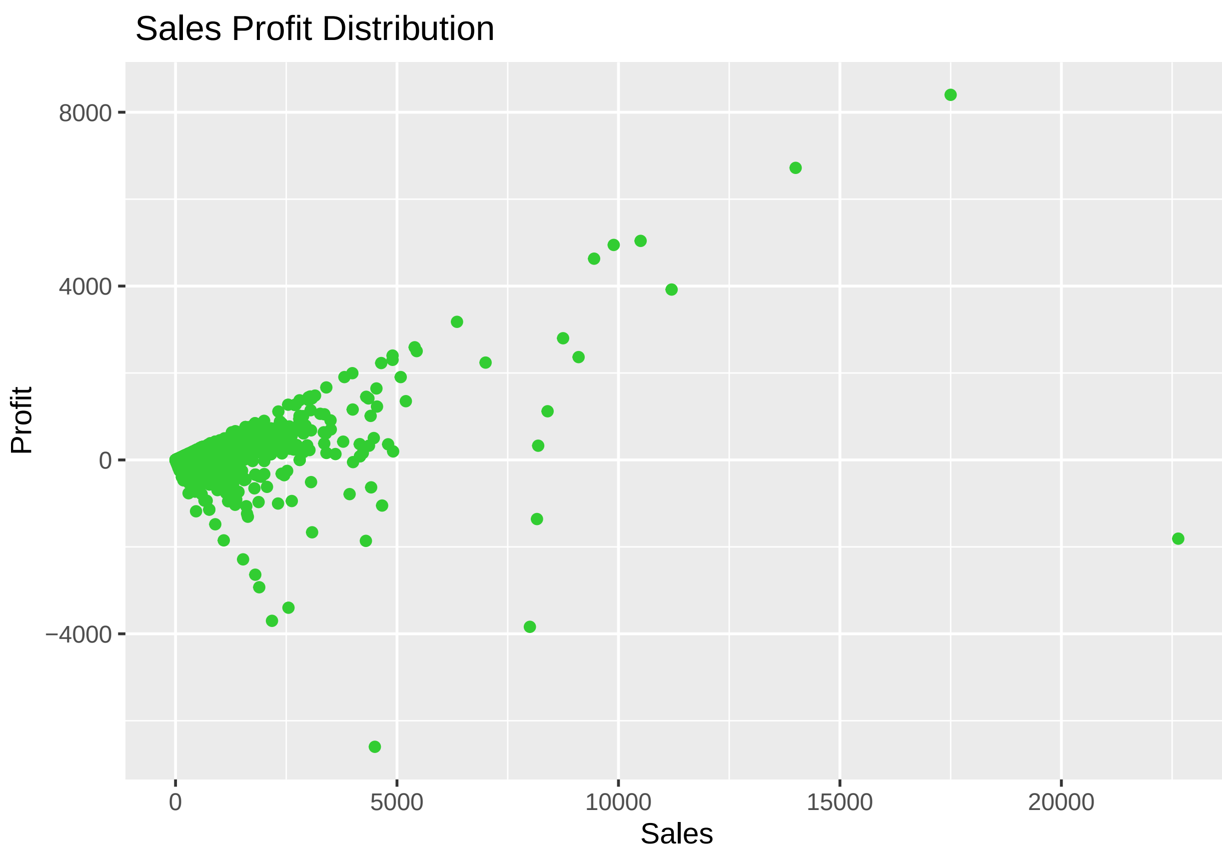



```
ggplot(data) +
geom_point(mapping = aes(x = Profit, y = Discount),colour="violetred3")+
labs(title=" Profit Discount Distribution")
```



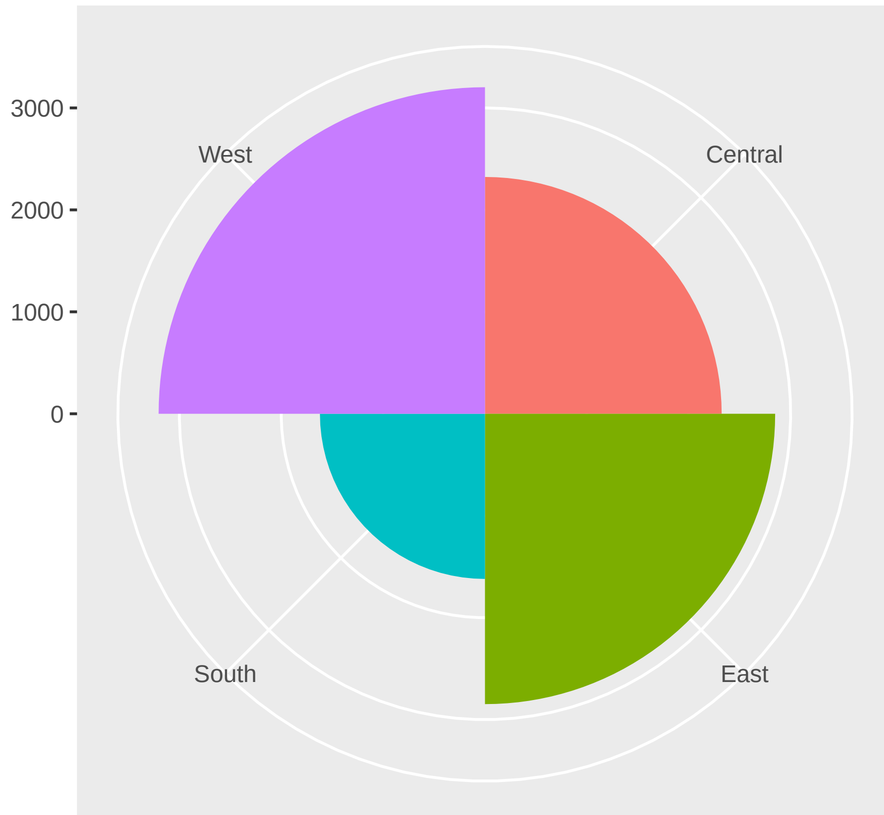
Sales Profit

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



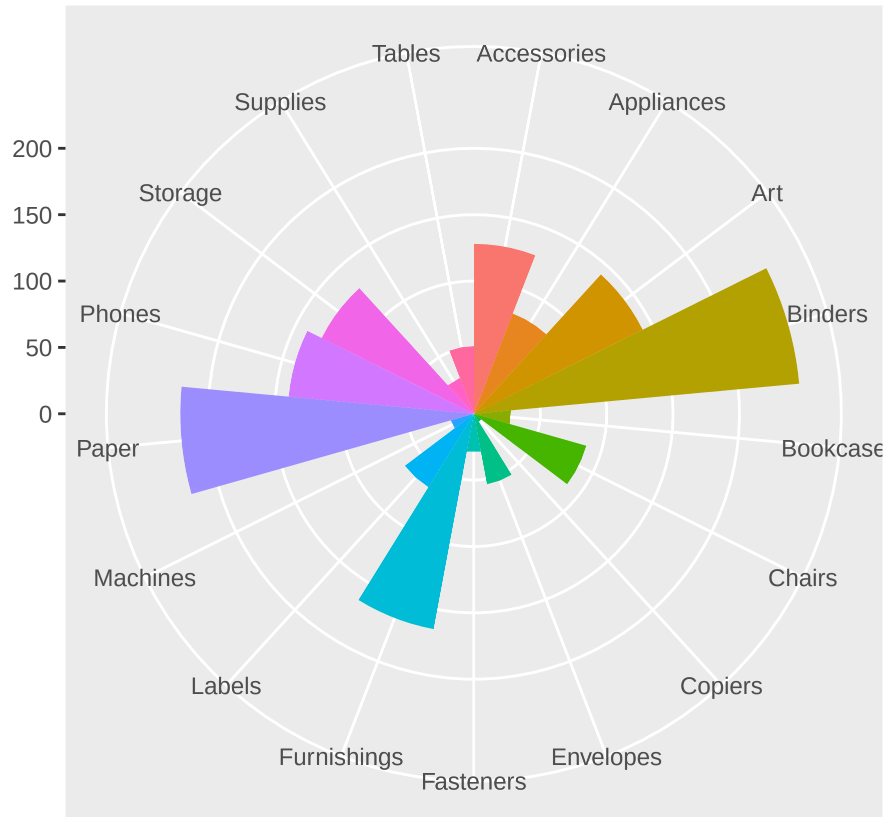
Transactions by region:

```
bar <- ggplot(data = data) +
  geom_bar(
    mapping = aes(x = Region, fill = Region),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)

bar + coord_polar()
```



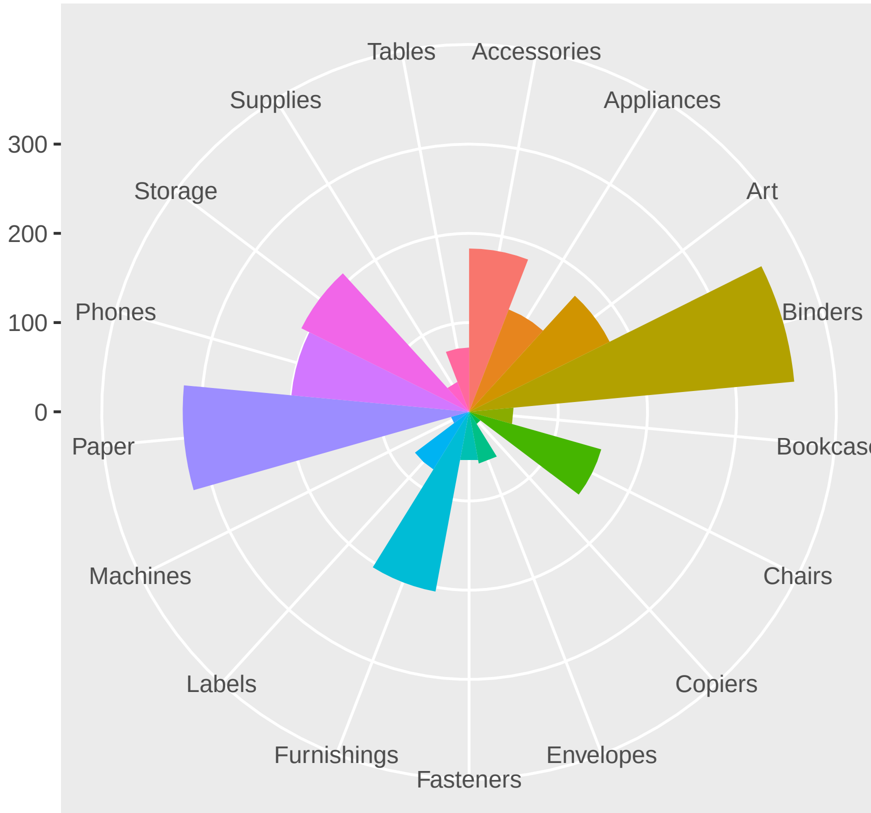
The above chart shows proportions of transactions from the different regions.

```
#Extracting the rows for South region, and sub-categories:
South <- data %>%
  select(Region, Sub_Category) %>%
  filter(Region == "South")

bar <- ggplot(data = South) +
  geom_bar(
    mapping = aes(x = Sub_Category, fill = Sub_Category),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)

bar + coord_polar()
```

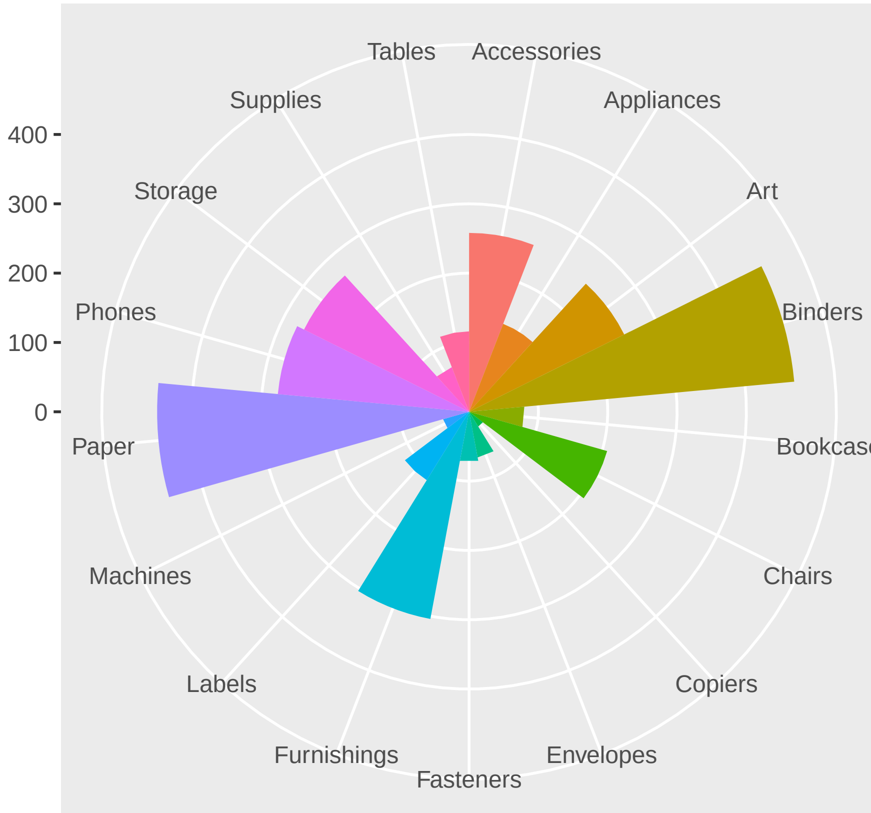


In the South, most transactions are Binders, Paper, or Furnishings.

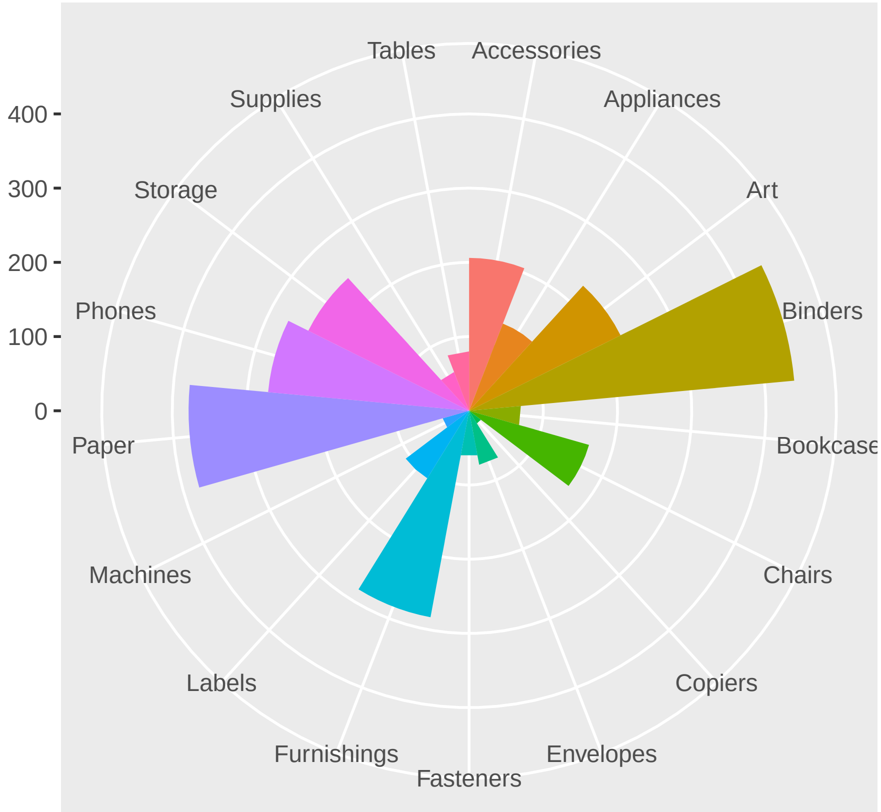
```r
#Extracting the rows for Central region, and sub-categories:
Central <- data %>%
  select(Region, Sub_Category) %>%
  filter(Region == "Central")

bar <- ggplot(data = Central) +
  geom_bar(
    mapping = aes(x = Sub_Category, fill = Sub_Category),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)

bar + coord_polar()
```




```
#Extracting the rows for West region, and sub-categories:
West <- data %>%
  select(Region, Sub_Category) %>%
  filter(Region == "West")

bar <- ggplot(data = West) +
  geom_bar(
    mapping = aes(x = Sub_Category, fill = Sub_Category),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)

bar + coord_polar()
```

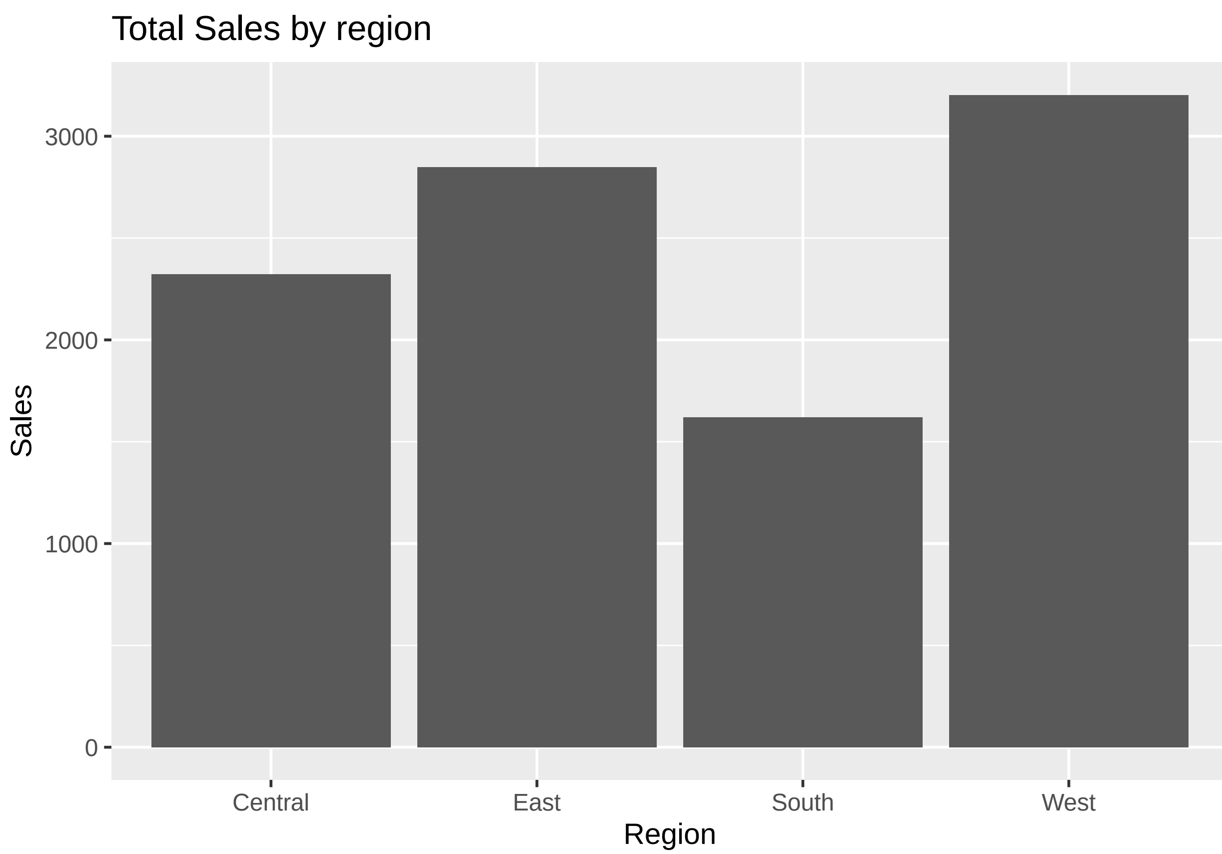



```r
#Extracting the rows for East region, and sub-categories:
East <- data %>%
  select(Region, Sub_Category) %>%
  filter(Region == "East")

bar <- ggplot(data = East) +
  geom_bar(
    mapping = aes(x = Sub_Category, fill = Sub_Category),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)

bar + coord_polar()
```



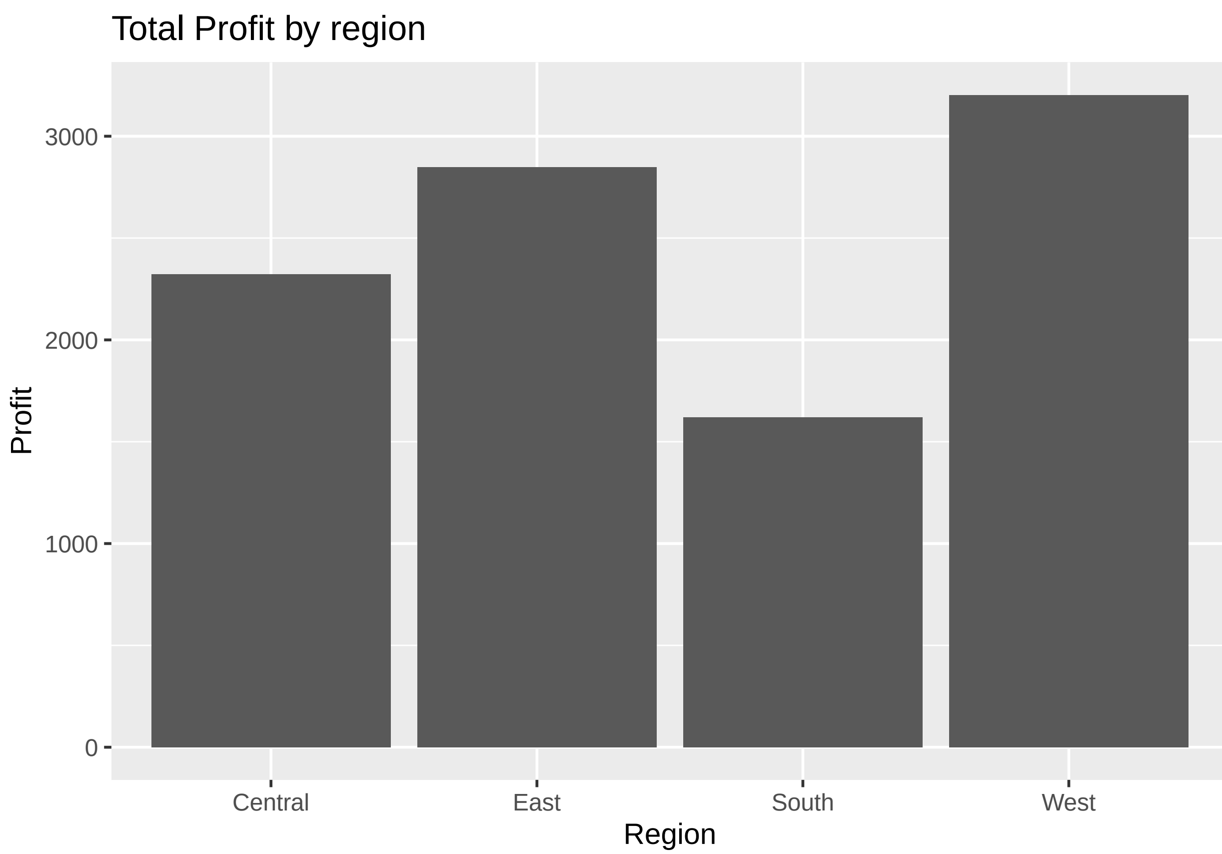
- **bar charts of profits/sales by region**

```
ggplot(data = data) +
  geom_bar(mapping = aes(x = Region, fill = Sales)) +
  ggtitle("Total Sales by region") +
  ylab("Sales")
```



Total sales per region.

```
ggplot(data = data) +
  geom_bar(mapping = aes(x = Region, fill = Profit))+
  ggtitle("Total Profit by region")+
  ylab("Profit")
```



Data Preparation

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

```
#make a copy of the original dataset and copy to data1
data1 <- data
```

drop column Row ID because it is not necessary; it is the row number from the original excel file. The country variable is also not needed because all the values are United states. Customer_Name and Customer_ID give redundant information. So we will drop the Customer_Name column and keep only the Customer_ID column.

```
data1[,c("Row_ID", "Country", "Customer_Name")]<-NULL
```

Test & Train dataset

Model

Evaluation

Deployment

Responsible ML Framework

Conclusion

References