

DATA ANALYSIS USING R

Importing necessary libraries

```
#####Install necessary libraries

install.packages("ggplot2")

library(ggplot2)

install.packages("dplyr")

library(dplyr)

install.packages("tidyverse") # Installs ggplot2 and other packages

install.packages("readxl") # Installs readxl for reading Excel files

library(tidyverse)

library(readxl)
```

output:

```
> library(ggplot2)

> library(dplyr)

> library(tidyverse)

> library(readxl)
```

Data loading

```
#####Data loading

data <- read_excel("C:/Users/karth/Downloads/Sample -
Superstore.xls")

#It displays the entire dataset

View(data)
```

Understanding the data

```
#####Understanding the data
```

```
#structure of the dataset
```

```
str(data)
```

output:

```
#####Understanding the data
```

```
> #structure of the dataset
```

```
> str(data)
```

```
tibble [9,994 × 21] (S3: tbl_df/tbl/data.frame)
```

```
$ Row ID      : num [1:9994] 1 2 3 4 5 6 7 8 9 10 ...
```

```
$ Order ID    : chr [1:9994] "CA-2016-152156" "CA-2016-152156" "CA-2016-138688" "US-2015-108966" ...
```

```
$ Order Date  : POSIXct[1:9994], format: "2016-11-08" ...
```

```
$ Ship Date   : POSIXct[1:9994], format: "2016-11-11" ...
```

```
$ Ship Mode   : chr [1:9994] "Second Class" "Second Class" "Second Class" "Standard Class" ...
```

```
$ Customer ID : chr [1:9994] "CG-12520" "CG-12520" "DV-13045" "SO-20335" ...
```

```
$ Customer Name: chr [1:9994] "Claire Gute" "Claire Gute" "Darrin Van Huff" "Sean O'Donnell" ...
```

```
$ Segment     : chr [1:9994] "Consumer" "Consumer" "Corporate" "Consumer" ...
```

```
$ Country      : chr [1:9994] "United States" "United States" "United States" "United States" ...
```

```
$ City         : chr [1:9994] "Henderson" "Henderson" "Los Angeles" "Fort Lauderdale" ...
```

```
$ State        : chr [1:9994] "Kentucky" "Kentucky" "California" "Florida" ...
```

```
$ Postal Code  : num [1:9994] 42420 42420 90036 33311 33311 ...
```

```
$ Region       : chr [1:9994] "South" "South" "West" "South" ...
```

```
$ Product ID   : chr [1:9994] "FUR-BO-10001798" "FUR-CH-10000454" "OFF-LA-10000240" "FUR-TA-10000577" ...
```

```
$ Category     : chr [1:9994] "Furniture" "Furniture" "Office Supplies" "Furniture" ...
```

```
$ Sub-Category : chr [1:9994] "Bookcases" "Chairs" "Labels" "Tables" ...
```

```
$ Product Name : chr [1:9994] "Bush Somerset Collection Bookcase" "Hon Deluxe Fabric Upholstered Stacking Chairs  
Rounded Back" "Self-Adhesive Address Labels for Typewriters by Universal" "Bretford CR4500 Series Slim Rectangular  
Table" ...
```

```
$ Sales       : num [1:9994] 262 731.9 14.6 957.6 22.4 ...
```

```
$ Quantity    : num [1:9994] 2 3 2 5 2 7 4 6 3 5 ...
```

```
$ Discount    : num [1:9994] 0 0 0 0.45 0.2 0 0 0.2 0.2 0 ...
$ Profit      : num [1:9994] 41.91 219.58 6.87 -383.03 2.52 ...
```

#It displays the first 6 rows

```
head(data)
```

output:

#It displays the first 6 rows

```
> head(data)
```

```
# A tibble: 6 × 21
```

```
  `Row ID` `Order ID` `Order Date`    `Ship Date`
    <dbl> <chr>      <dtm>          <dtm>
1     1 CA-2016-1... 2016-11-08 00:00:00 2016-11-11 00:00:00
2     2 CA-2016-1... 2016-11-08 00:00:00 2016-11-11 00:00:00
3     3 CA-2016-1... 2016-06-12 00:00:00 2016-06-16 00:00:00
4     4 US-2015-1... 2015-10-11 00:00:00 2015-10-18 00:00:00
5     5 US-2015-1... 2015-10-11 00:00:00 2015-10-18 00:00:00
6     6 CA-2014-1... 2014-06-09 00:00:00 2014-06-14 00:00:00
# 17 more variables: `Ship Mode` <chr>,
# `Customer ID` <chr>, `Customer Name` <chr>,
# `Segment` <chr>, `Country` <chr>, `City` <chr>, `State` <chr>,
# `Postal Code` <dbl>, `Region` <chr>, `Product ID` <chr>,
# `Category` <chr>, `Sub-Category` <chr>,
# `Product Name` <chr>, `Sales` <dbl>, `Quantity` <dbl>,
# `Discount` <dbl>, `Profit` <dbl>
```

#It displays the column names

```
names(data)
```

output:

#It displays the column names

```
> names(data)
[1] "Row ID"      "Order ID"      "Order Date"
[4] "Ship Date"   "Ship Mode"     "Customer ID"
[7] "Customer Name" "Segment"       "Country"
[10] "City"        "State"         "Postal Code"
[13] "Region"      "Product ID"    "Category"
[16] "Sub-Category" "Product Name"  "Sales"
[19] "Quantity"    "Discount"      "Profit"
```

#It displays the dimensions of rows and columns

```
dim(data)
```

output:

```
> #It displays the dimensions of rows and columns
```

```
> dim(data)
```

```
[1] 9994 21
```

#It displays the sum of null values

```
sum(is.na(data))
```

output:

> #It displays the sum of null values

```
> sum(is.na(data))
```

```
[1] 0
```

Summarize the dataset

```
#####Summarize the dataset
```

```
# Get a summary of each column
```

```
summary(data)
```

output:

```
> # Get a summary of each column
```

```
> summary(data)
```

Row ID Order ID

Min. : 1 Length:9994

1st Qu.:2499 Class :character

Median :4998 Mode :character

Mean :4998

3rd Qu.:7496

Max. :9994

Order Date

Min. :2014-01-03 00:00:00.00

1st Qu.:2015-05-23 00:00:00.00

Median :2016-06-26 00:00:00.00

Mean :2016-04-30 00:07:12.25

3rd Qu.:2017-05-14 00:00:00.00

Max. :2017-12-30 00:00:00.00

Ship Date Ship Mode

Min. :2014-01-07 00:00:00.00 Length:9994

1st Qu.:2015-05-27 00:00:00.00 Class :character

Median :2016-06-29 00:00:00.00 Mode :character

Mean :2016-05-03 23:06:58.56

3rd Qu.:2017-05-18 00:00:00.00

Max. :2018-01-05 00:00:00.00

Customer ID Customer Name Segment

Length:9994 Length:9994 Length:9994

Class :character Class :character Class :character

Mode :character Mode :character Mode :character

Country City State

Length:9994 Length:9994 Length:9994

Class :character Class :character Class :character
Mode :character Mode :character Mode :character

Postal Code Region Product ID
Min. : 1040 Length:9994 Length:9994
1st Qu.:23223 Class :character Class :character
Median :56431 Mode :character Mode :character
Mean :55190
3rd Qu.:90008
Max. :99301

Category Sub-Category Product Name
Length:9994 Length:9994 Length:9994
Class :character Class :character Class :character
Mode :character Mode :character Mode :character

Sales Quantity Discount
Min. : 0.444 Min. : 1.00 Min. : 0.0000
1st Qu.: 17.280 1st Qu.: 2.00 1st Qu.: 0.0000
Median : 54.490 Median : 3.00 Median : 0.2000
Mean : 229.858 Mean : 3.79 Mean : 0.1562
3rd Qu.: 209.940 3rd Qu.: 5.00 3rd Qu.: 0.2000
Max. : 22638.480 Max. : 14.00 Max. : 0.8000

Profit
Min. : -6599.978
1st Qu.: 1.729
Median : 8.666

Mean : 28.657

3rd Qu.: 29.364

Max. : 8399.976

#checking null values

colSums(is.na(data))

output:

> #checking null values

> colSums(is.na(data))

Row ID	Order ID	Order Date	Ship Date
0	0	0	0

Ship Mode	Customer ID	Customer Name	Segment
0	0	0	0

Country	City	State	Postal Code
0	0	0	0

Region	Product ID	Category	Sub-Category
0	0	0	0

Product Name	Sales	Quantity	Discount
0	0	0	0

Profit
0

#mean for sales

mean(data\$Sales)

output:

> #mean for sales

> mean(data\$Sales)

[1] 229.858

#median for sales

median(data\$Sales)

output:

```
> #median for sales  
> median(data$Sales)  
[1] 54.49
```

```
#mean for profit  
mean(data$Profit)
```

output:

```
> #mean for profit  
> mean(data$Profit)  
[1] 28.6569
```

```
#median for profit  
median(data$Profit)
```

output:

```
> #median for profit  
> median(data$Profit)  
[1] 8.6665
```

Checking missing values:

```
#####Check for missing values  
sum(is.na(data))
```

output:

```
> #####Check for missing values  
> sum(is.na(data))  
[1] 0
```

```
# Unique values in a specific column  
unique(data$Sales)
```

output:

```
> # Unique values in a specific column  
> unique(data$Sales)  
[1] 261.9600 731.9400 14.6200 957.5775 22.3680  
[6] 48.8600 7.2800 907.1520 18.5040 114.9000  
[11] 1706.1840 911.4240 15.5520 407.9760 68.8100  
[16] 2.5440 665.8800 55.5000 8.5600 213.4800  
[21] 22.7200 19.4600 60.3400 71.3720 1044.6300  
[26] 11.6480 90.5700 3083.4300 9.6180 124.2000  
[31] 3.2640 86.3040 6.8580 15.7600 29.4720  
[36] 1097.5440 190.9200 113.3280 532.3992 212.0580
```


[41]	371.1680	147.1680	77.8800	95.6160	45.9800
[46]	17.4600	211.9600	45.0000	21.8000	38.2200
[51]	75.1800	6.1600	89.9900	15.2600	1029.9500
[56]	208.5600	32.4000	319.4100	14.5600	30.0000
[61]	48.4800	1.6800	13.9800	25.8240	146.7300
[66]	79.7600	213.1150	1113.0240	167.9680	75.8800
[71]	4.6160	19.0500	831.9360	97.0400	72.7840
[76]	1.2480	9.7080	27.2400	19.3000	208.1600
[81]	16.7400	14.9000	21.3900	200.9840	230.3760
[86]	301.9600	19.9900	158.3680	20.1000	73.5840
[91]	6.4800	12.9600	53.3400	32.9600	5.6820
[96]	96.5300	51.3120	64.6240	95.9760	1.7880
[101]	23.9200	238.8960	102.3600	36.8820	74.1120
[106]	27.9920	3.3040	339.9600	41.9600	75.9600
[111]	40.0960	4.7200	23.9760	130.4640	787.5300
[116]	157.7940	47.0400	30.8400	226.5600	115.0200
[121]	68.0400	600.5580	617.7000	2.3880	243.9920
[126]	81.4240	238.5600	59.9700	78.3040	21.4560
[131]	20.0400	35.4400	11.5200	4.0200	76.1760
[136]	65.8800	43.1200	82.8000	8.8200	10.8600
[141]	143.7000	839.4300	671.9300	93.8880	384.4500
[146]	149.9700	1951.8400	171.5500	157.9200	203.1840
[151]	58.3800	105.5200	80.8800	6.6300	457.5680
[156]	944.9300	5.9800	54.3840	28.4000	27.6800
[161]	9.9360	8159.9520	275.9280	1740.0600	32.0640
[166]	177.9800	143.9760	20.9400	110.9600	340.1440
[171]	52.4480	20.1600	97.2640	396.8020	15.8800
[176]	3.2800	24.8160	408.7440	503.9600	149.9500
[181]	29.0000	7.1600	176.8000	37.2240	20.0160
[186]	899.1360	71.7600	51.8400	626.3520	19.9000
[191]	14.2800	7.4080	6.0480	46.2600	2.9460
[196]	16.0560	21.7440	218.7500	2.6000	66.2840
[201]	35.1680	444.7680	83.9200	131.9800	15.9200
[206]	52.2900	91.9900	20.8000	23.6800	452.4500
[211]	62.9820	1188.0000	89.5840	93.0600	302.3760
[216]	5.5840	22.7040	19.7760	72.7040	479.9880
[221]	27.1680	2.2000	622.4500	21.9800	161.5680
[226]	389.6960	18.6480	233.8600	620.6145	5.3280
[231]	258.0720	617.9760	10.5600	25.9200	419.6800
[236]	11.6880	31.9840	177.2250	4.0440	2001.8600
[241]	166.7200	47.8800	1503.2500	25.9200	321.5680
[246]	7.6100	3347.3700	80.5800	361.9200	12.1320
[251]	82.3680	53.9200	647.9040	20.3700	221.5500
[256]	17.5200	1.6240	3059.9820	2519.9580	328.2240
[261]	79.9000	14.0160	7.5600	37.2080	57.5760
[266]	725.8400	209.9300	5.2800	10.9200	18.1760
[271]	59.7120	24.8400	2.0800	1114.4000	1038.8400

[276]	141.7600	239.8000	31.1040	254.0580	194.5280
[281]	961.4800	19.0960	18.4960	255.9840	86.9700
[286]	300.4160	230.3520	218.3520	78.6000	27.5520
[291]	1082.4800	56.9100	77.6000	14.2800	219.0750
[296]	26.8000	9.8400	45.4800	289.2000	4.8900
[301]	15.1360	466.7680	15.2320	6.2640	87.5400
[306]	178.3840	99.1360	135.8820	3991.9800	275.9400
[311]	360.0000	43.5700	251.5200	99.9900	15.9920
[316]	290.8980	54.2240	786.7440	100.2400	37.7640
[321]	82.8000	20.7240	4.8960	4.7520	959.9840
[326]	14.3680	7.7120	698.3520	4.9600	17.8560
[331]	509.9700	30.9920	71.9280	88.8000	47.9760
[336]	24.5600	6.7900	3.0480	49.1200	4355.1680
[341]	388.7040	8.2600	17.0400	34.4000	36.2400
[346]	647.8400	20.7000	488.6460	5.5600	47.1200
[351]	23.2000	7.3600	104.7900	1043.9200	53.4240
[356]	8.1600	1023.9360	9.2400	479.0400	1488.4240
[361]	8.6520	23.8320	12.1760	50.9600	49.5360
[366]	41.9000	375.4575	83.9760	482.3400	2.9600
[371]	2.6240	23.3600	39.9800	246.3840	1799.9700
[376]	12.4620	75.7920	49.9600	70.1200	35.9520
[381]	2396.2656	131.1360	57.5840	9.5680	39.0720
[386]	35.9100	179.9500	1199.9760	27.1500	1004.0240
[391]	9.6800	28.3500	55.9800	1336.8290	113.5680
[396]	139.8600	307.1360	95.9200	383.8000	5.7800
[401]	9.3200	15.2500	196.7520	56.5600	32.7000
[406]	866.4000	287.9200	69.9900	6.6720	189.5880
[411]	291.9600	4.7680	714.3000	4.8120	247.8000
[416]	1007.9790	313.4880	31.8720	207.8460	12.2200
[421]	194.9400	70.9500	91.3600	242.9400	22.0500
[426]	2.9100	59.5200	161.9400	263.8800	30.4800
[431]	35.1200	284.3640	665.4080	63.8800	129.5680
[436]	747.5580	8.9280	103.9200	899.9100	23.5600
[441]	1272.6300	28.4850	185.3760	78.2720	254.7440
[446]	205.3328	4.7880	55.4800	340.9200	222.6660
[451]	703.9680	92.5200	62.6500	94.8500	95.7600
[456]	40.2000	14.7000	704.2500	9.0900	5.9600
[461]	159.9800	29.6000	514.1650	279.9600	2735.9520
[466]	7.9920	63.9840	70.3680	449.1500	11.0700
[471]	93.9800	189.8820	105.4200	119.6160	255.7600
[476]	241.5680	69.3000	22.6200	14.9520	801.5680
[481]	2.3760	32.7920	15.9200	2.7400	8.3400
[486]	46.7400	6354.9500	126.3000	38.0400	7.1520
[491]	5.8800	2999.9500	51.4500	11.9600	1126.0200
[496]	18.3920	14.1120	210.9800	55.1760	66.2600
[501]	22.2000	683.9520	45.6960	36.3360	666.2480
[506]	52.5120	190.7200	47.9400	979.9500	22.7500

[511]	16.7680	42.6160	10.7520	152.9400	283.9200
[516]	468.9000	380.8640	646.7760	58.1120	100.7920
[521]	66.1120	41.2800	13.3600	250.2720	11.3640
[526]	8.7200	1121.5680	34.5040	10.8240	1295.7800
[531]	19.4560	1335.6800	42.6000	84.0560	13.0000
[536]	13.1280	3.9600	2.6100	374.3760	91.8400
[541]	81.0880	19.4400	451.1520	72.4500	13.9600
[546]	33.2640	14.8500	160.7200	19.9200	7.3000
[551]	69.7120	8.7920	51.5200	470.3760	105.5840
[556]	31.1520	6.7830	406.3680	70.9800	294.9300
[561]	84.7840	20.7360	16.8210	10.3680	9.3440
[566]	31.2000	76.1200	445.9600	327.7600	11.6320
[571]	143.9820	494.3760	5.8400	142.7760	7.2180
[576]	43.1880	131.9040	3.2820	21.1680	55.1880
[581]	369.5760	15.7120	8.4480	728.9460	119.9400
[586]	3.6480	40.4800	9.9400	107.4240	37.9100
[591]	88.0200	8.6900	555.2100	523.4800	161.8200
[596]	35.5600	97.1600	15.2400	13.2300	243.3840
[601]	119.8000	300.7680	17.8800	235.9440	392.9400
[606]	18.8820	122.3280	1049.2000	15.4240	18.8400
[611]	330.4000	26.2500	132.5200	209.3000	31.5600
[616]	30.1440	14.8000	316.0000	379.4000	97.8200
[621]	103.1200	113.5520	3.3180	134.2880	701.3720
[626]	2.3080	999.4320	724.0800	918.7850	2.7240
[631]	459.9500	10.7400	23.7600	85.0560	381.5760
[636]	30.3600	108.9250	36.3520	19.5600	61.4400
[641]	38.9000	99.3900	2.6880	27.8160	82.5240
[646]	182.9940	14.3520	64.9600	68.6000	7999.9800
[651]	167.4400	479.9700	191.9840	104.0100	284.8200
[656]	36.8400	166.2400	33.4000	198.2720	47.3600
[661]	97.6960	2.6960	18.5880	15.0720	209.8800
[666]	369.9120	166.8400	15.2160	119.9600	883.9200
[671]	46.7200	24.4480	281.3400	307.9800	299.9700
[676]	103.0560	59.8080	73.3200	146.8200	1652.9400
[681]	296.3700	129.9200	45.5840	17.5680	55.9920
[686]	182.7200	400.0320	33.6300	542.6460	6.3000
[691]	179.9700	99.6960	27.9360	84.9800	18.7200
[696]	49.9800	11.7840	272.7360	3.5400	51.5200
[701]	3.5280	4.6240	55.1680	567.1200	359.3200
[706]	11.9920	58.0500	157.7400	56.9800	2.8800
[711]	79.9200	383.4380	119.8000	58.3200	12.3900
[716]	107.9820	11.3600	50.9400	646.7400	5.6400
[721]	572.5800	310.8800	641.9600	18.2800	207.0000
[726]	32.3500	7.7100	40.3000	34.5800	32.7600
[731]	544.0080	59.9400	4.2800	32.0700	24.0000
[736]	35.4900	47.9840	186.6900	17.4560	348.9280
[741]	143.9600	15.4200	43.0400	332.9400	1363.9600

[746]	9.9600	21.7200	132.7900	21.5600	22.2300
[751]	215.9680	355.3200	43.1760	1983.9680	1298.5500
[756]	213.9200	25.7800	1399.9300	5.3440	41.4720
[761]	3.1680	1228.4650	31.0860	335.5200	239.9700
[766]	9.8200	67.8000	167.9700	35.0000	37.2400
[771]	15.2800	180.6600	191.9800	65.9900	35.2160
[776]	23.6960	265.4750	51.1840	9.6640	21.0720
[781]	60.4500	186.0480	37.4400	26.9760	83.7200
[786]	287.9400	48.8960	115.3600	5.1600	38.8800
[791]	185.8800	44.4600	39.9600	102.3000	21.3600
[796]	23.1000	191.4720	5.2480	59.1840	2.8900
[801]	51.9400	15.9360	44.9100	1141.4700	280.7820
[806]	34.4400	106.3440	192.1600	322.5900	29.9900
[811]	371.9700	5.8920	68.4720	1242.9000	13.4800
[816]	31.4000	13.9440	83.7600	37.6600	34.6800
[821]	4.5400	543.9200	155.8200	70.0080	15.6480
[826]	103.6000	46.9600	8.9040	10.4400	18.3360
[831]	323.9760	323.1360	90.9300	52.7760	1199.8000
[836]	1928.7800	352.3800	46.9400	143.7300	99.9180
[841]	797.9440	8.5680	149.3520	12.9920	85.1400
[846]	21.9900	406.6000	841.5680	252.0000	46.2000
[851]	28.8400	14.5920	89.8560	13.8720	12.1920
[856]	45.0560	29.7180	447.6960	159.9900	134.4800
[861]	17.1200	6.0960	1114.2720	540.5700	167.7600
[866]	393.1650	516.4880	1007.2320	2065.3200	25.3440
[871]	25.4720	78.8528	173.8000	29.5920	204.6000
[876]	6.2400	21.8800	4.6080	35.9700	191.6000
[881]	8.6400	501.8100	127.1040	124.2000	30.0720
[886]	160.9300	1.0800	34.0200	599.2920	3.3920
[891]	559.9840	603.9200	7.9680	27.9680	336.5100
[896]	1.1120	520.0500	17.9700	1166.9200	14.6240
[901]	10.2300	154.9000	2715.9300	617.9700	10.6700
[906]	36.6300	24.1000	33.1100	44.0200	2309.6500
[911]	1090.7820	484.6500	115.2960	7.0800	4.4010
[916]	44.7500	95.9840	151.7200	155.2500	14.0300
[921]	1618.3700	99.6000	155.8200	124.9500	601.6500
[926]	22.7400	1267.5300	1379.9200	6.2080	11.8080
[931]	63.3120	15.5880	177.2000	197.9700	854.9400
[936]	124.1100	14.4000	15.6960	2.6280	14.4270
[941]	86.6200	36.6240	23.9680	28.7280	697.1600
[946]	31.8600	722.3520	8.8400	58.4640	254.6040
[951]	1113.5040	168.4640	6.7200	282.8880	11.1600
[956]	108.4000	82.3440	9.0880	19.9360	65.5680
[961]	4.4160	107.7720	45.2160	10.4160	7.8720
[966]	118.7820	1.4480	55.4700	194.8480	1.7440
[971]	25.1760	6.2700	4.3680	40.8800	31.4400
[976]	6.8800	1106.9100	836.5920	26.3800	362.9200

```
[981] 4899.9300    71.9760     3.1500    71.9840    120.1500
[986]  219.1840    28.9000   355.9600   348.2080    35.7840
[991]  447.8400     7.0400     8.7300    29.2900     2.6940
[996]    2.9340    22.9200   100.7040     2.3280    10.7800
[ reached getOption("max.print") -- omitted 5144 entries ]
```

```
#It check the type of data
class(data$Sales)
class(data$`Customer Name`)
class(data$Category)
```

output:

```
> #It check the type of data
> class(data$Sales)
[1] "numeric"
> class(data$`Customer Name`)
[1] "character"
> class(data$Category)
[1] "character"
```

```
# Frequency of categories in a specific column
table(data$Category)
```

output:

```
> # Frequency of categories in a specific column
> table(data$Category)
```

```
Furniture Office Supplies    Technology
      2121         6026         1847
```

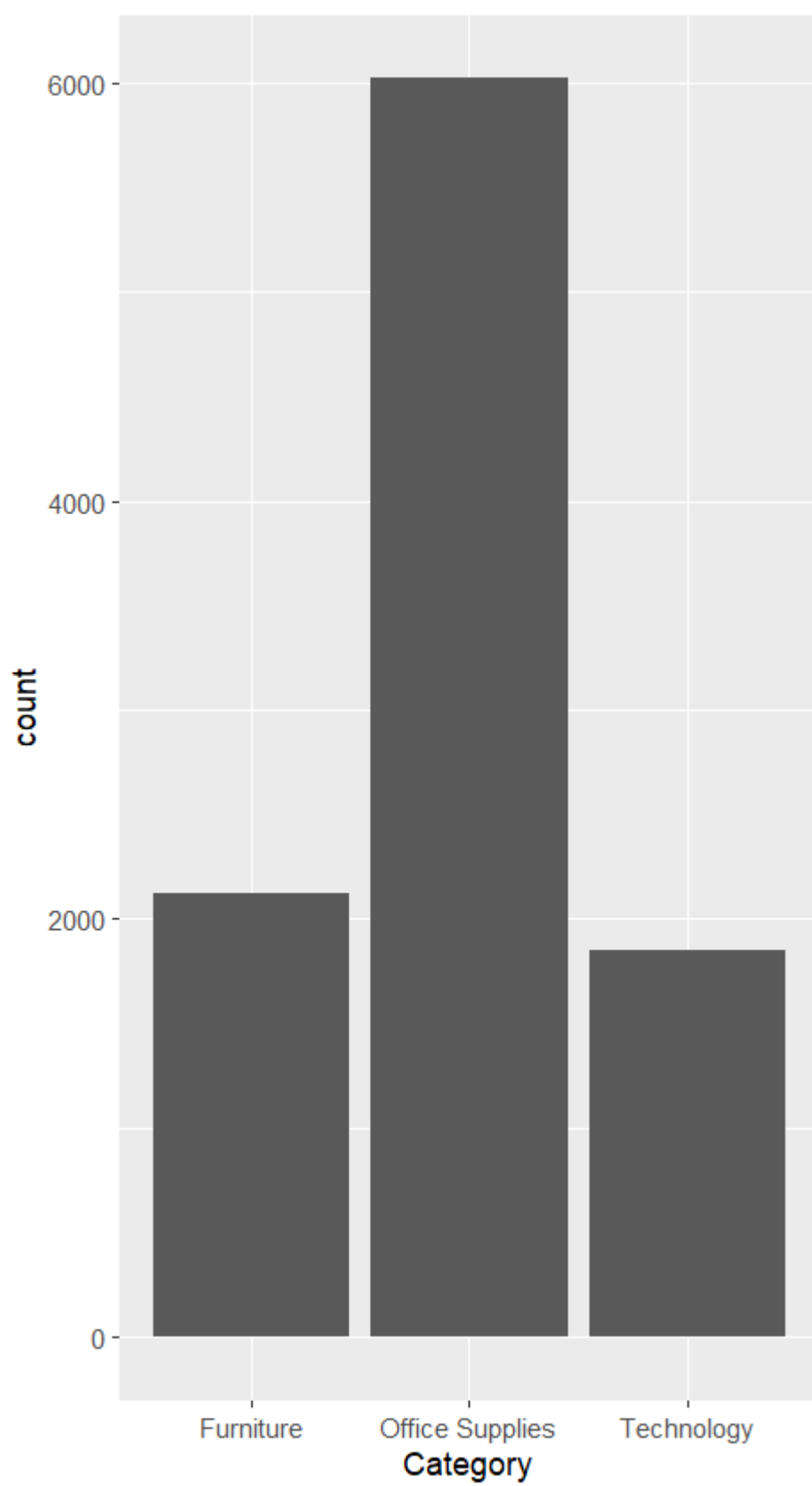
Data visualization:

```
#####Data visualization
```

```
###barplot for categorical data
```

```
ggplot(data,aes(x=Category))+geom_bar()
```

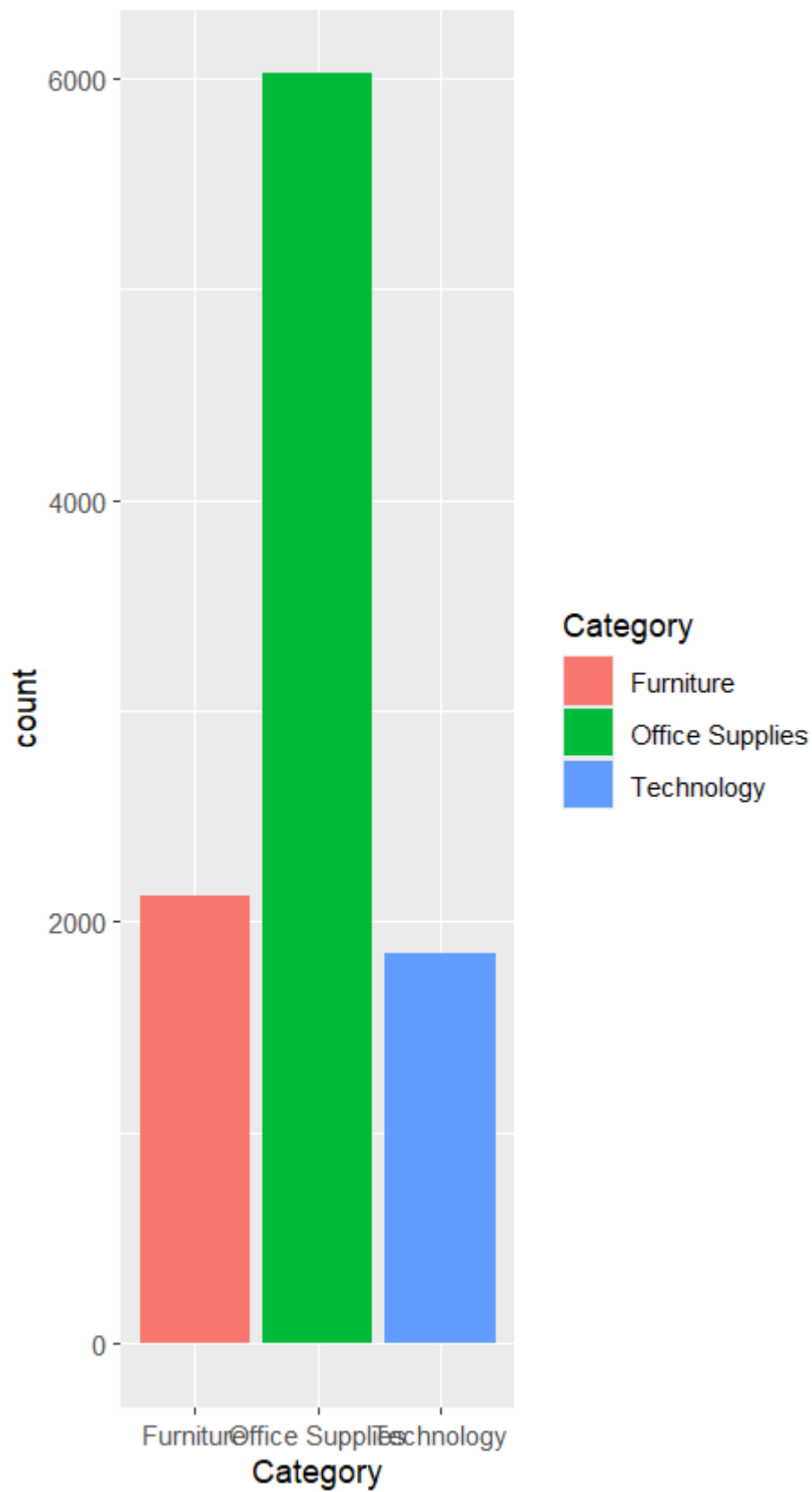
output:



```
##adding color
```

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

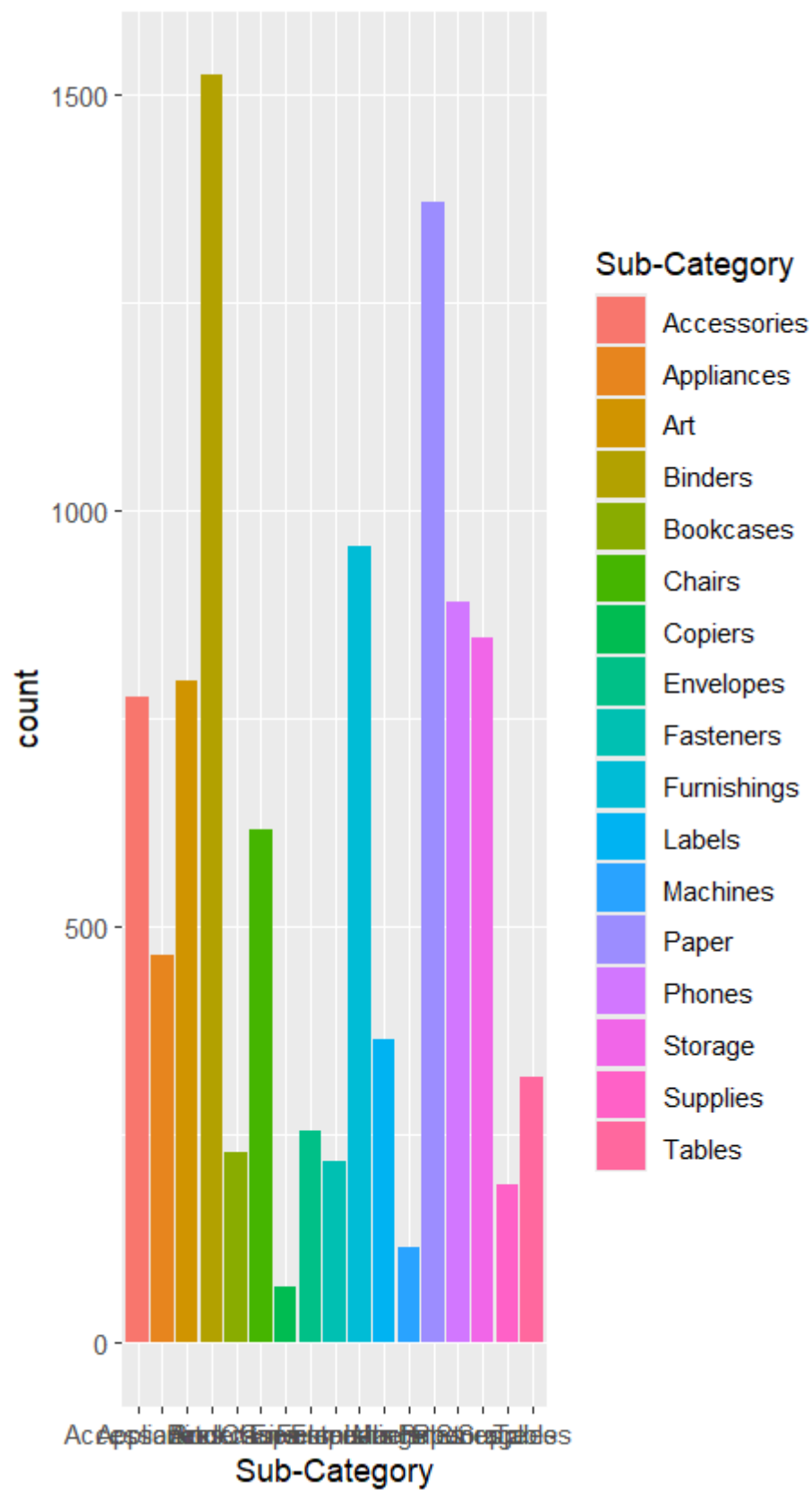
output:



#barplot for sub category

```
ggplot(data,aes(x=`Sub-Category`,fill=`Sub-Category`))+geom_bar()
```

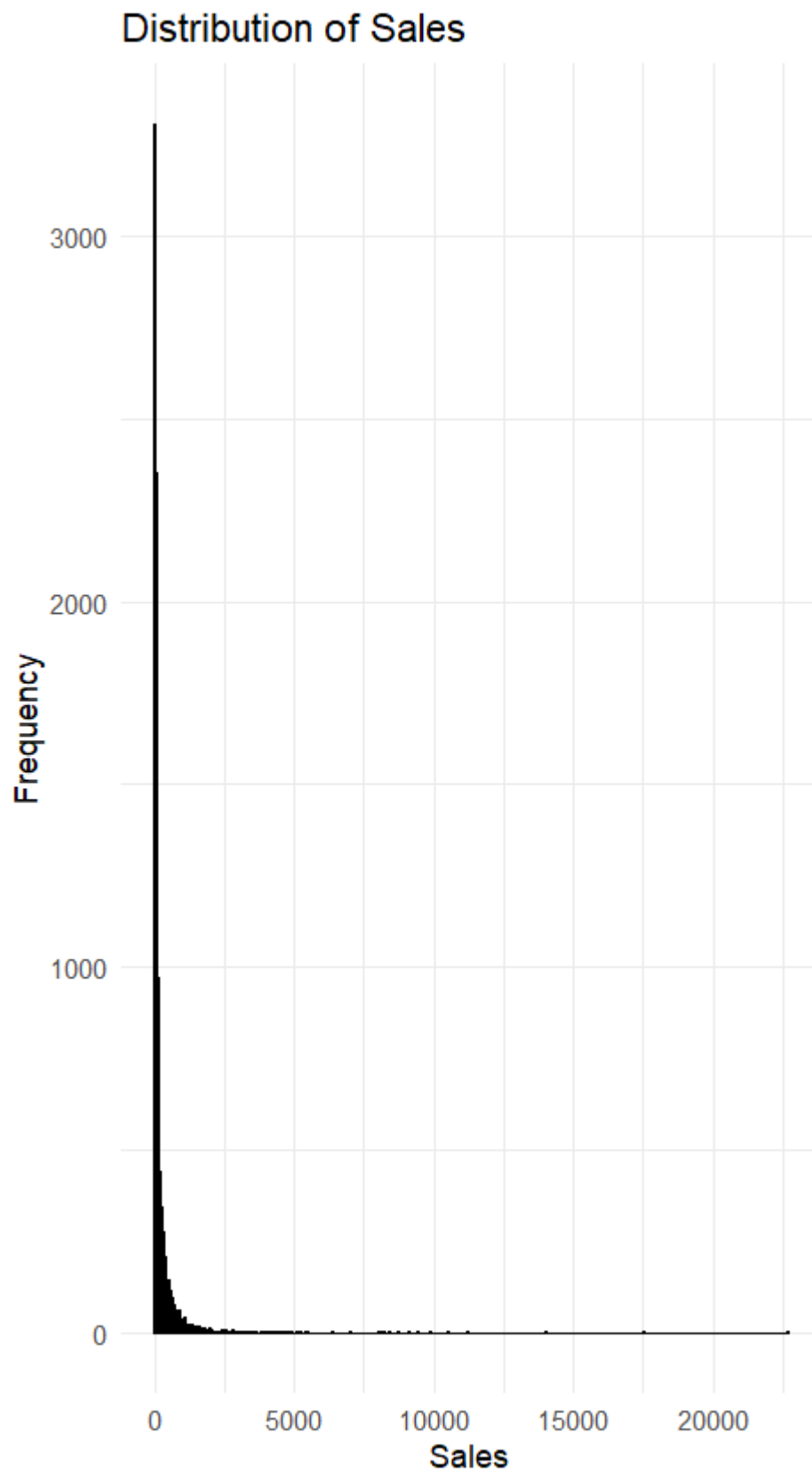

output:



Create a histogram for Sales

```
ggplot(data, aes(x = Sales)) +  
  geom_histogram(binwidth = 50, fill = "blue", color = "black",) +  
  labs(title = "Distribution of Sales", x = "Sales", y = "Frequency") +  
  theme_minimal()
```

output:

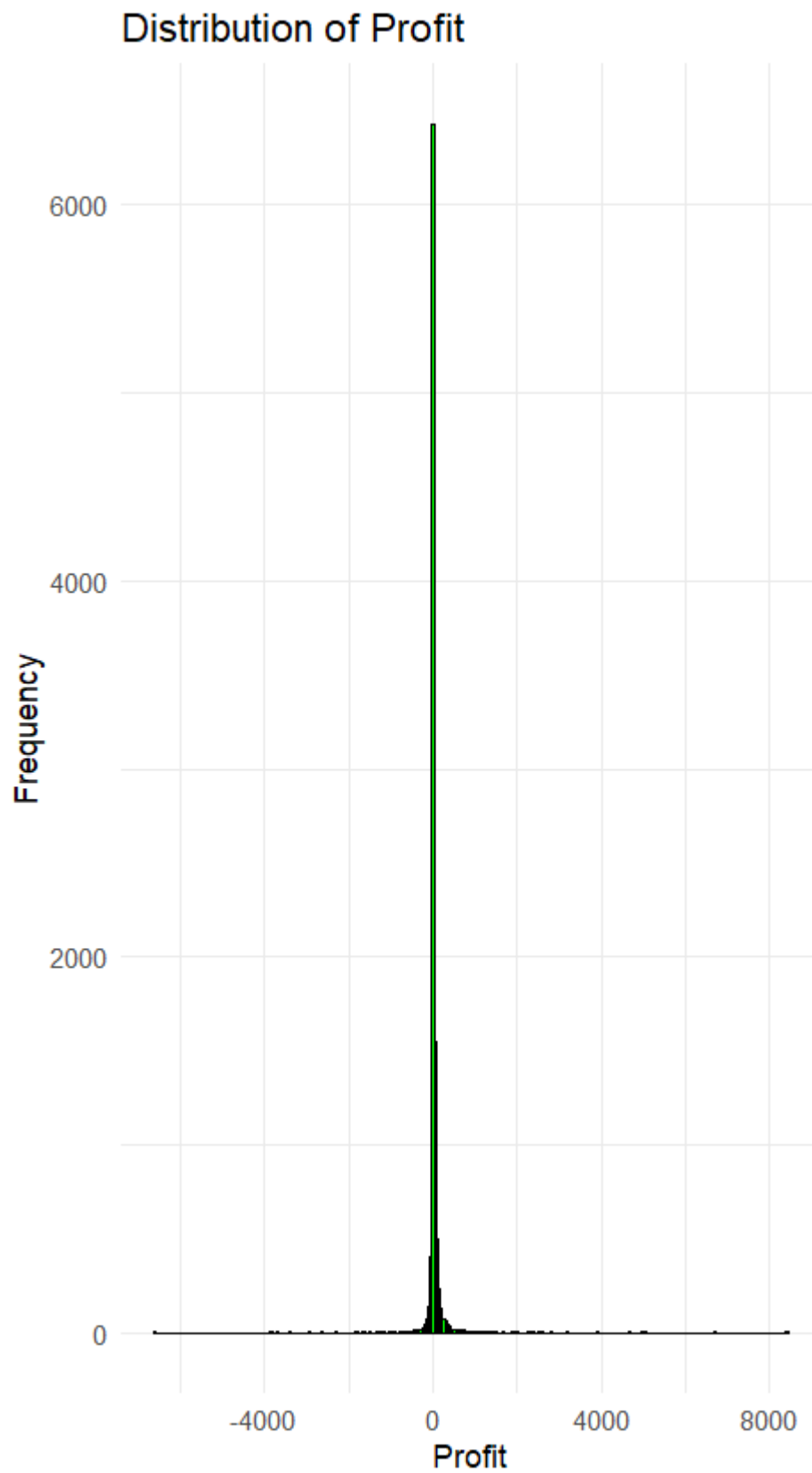


```
# Create a histogram for Profit
```

```
ggplot(data, aes(x = Profit)) +
```

```
geom_histogram(binwidth = 50, fill = "green", color = "black") +  
labs(title = "Distribution of Profit", x = "Profit", y = "Frequency") +  
theme_minimal()
```

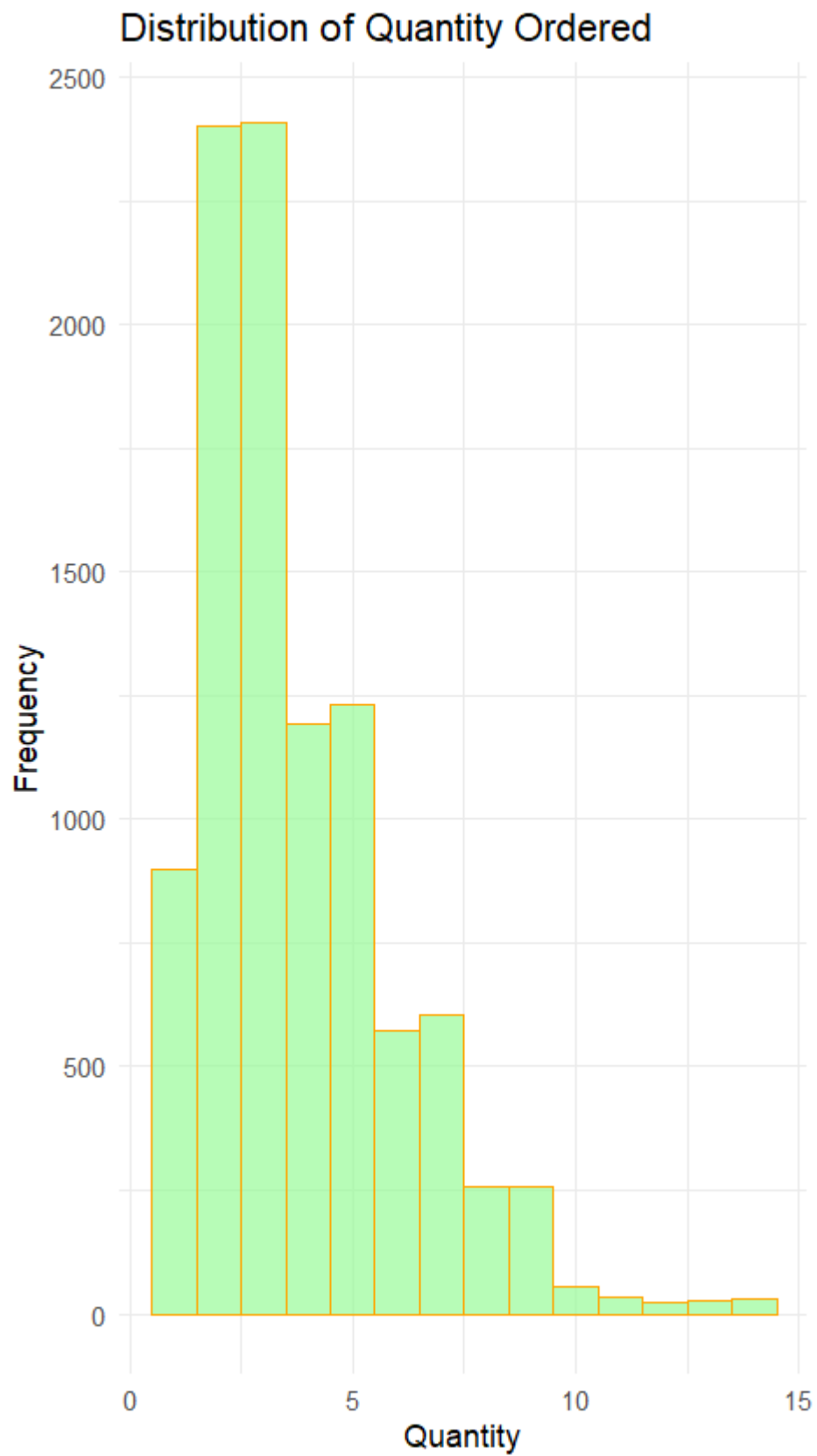
output:



```
# Create a histogram for Quantity  
ggplot(data, aes(x = Quantity)) +
```

```
geom_histogram(binwidth = 1, fill = "palegreen", color = "orange", alpha = 0.7) +  
labs(title = "Distribution of Quantity Ordered", x = "Quantity", y = "Frequency") +  
theme_minimal()
```

output:



Scatter plot of Sales vs Profit

```
ggplot(data, aes(x = Sales, y = Profit)) +
```

```
geom_point(color = "aquamarine") +  
ggtitle("Sales vs Profit") +  
xlab("Sales") +  
ylab("Profit")+theme_minimal()
```

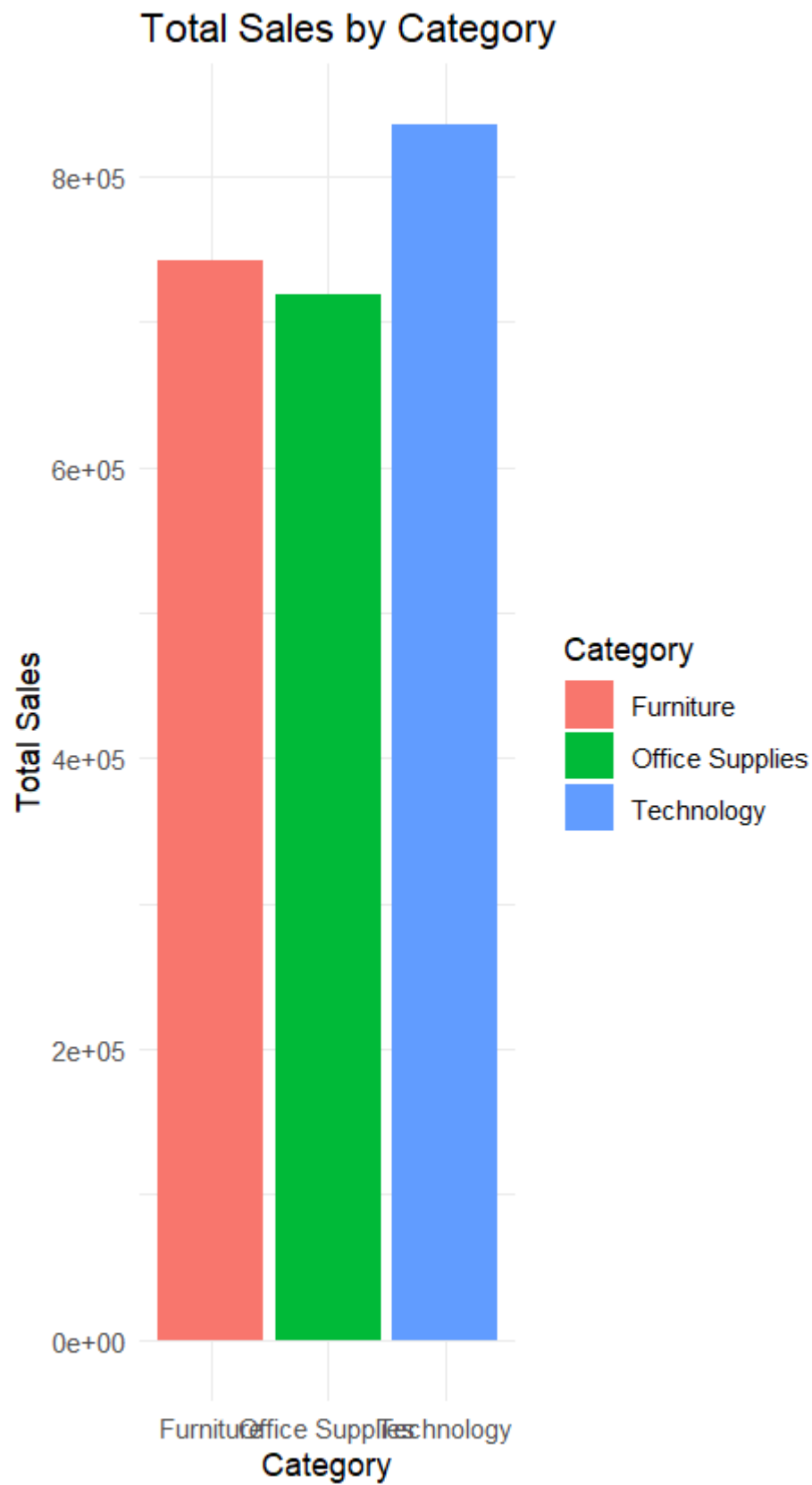
output:



Summarize total sales by category

```
category_sales <- data %>%  
  group_by(Category) %>%  
  summarise(Total_Sales = sum(Sales))  
  
# Create a bar graph for total sales by category  
ggplot(category_sales, aes(x = Category, y = Total_Sales, fill = Category)) +  
  geom_bar(stat = "identity") +  
  labs(title = "Total Sales by Category", x = "Category", y = "Total Sales")+  
  theme_minimal()
```

output:

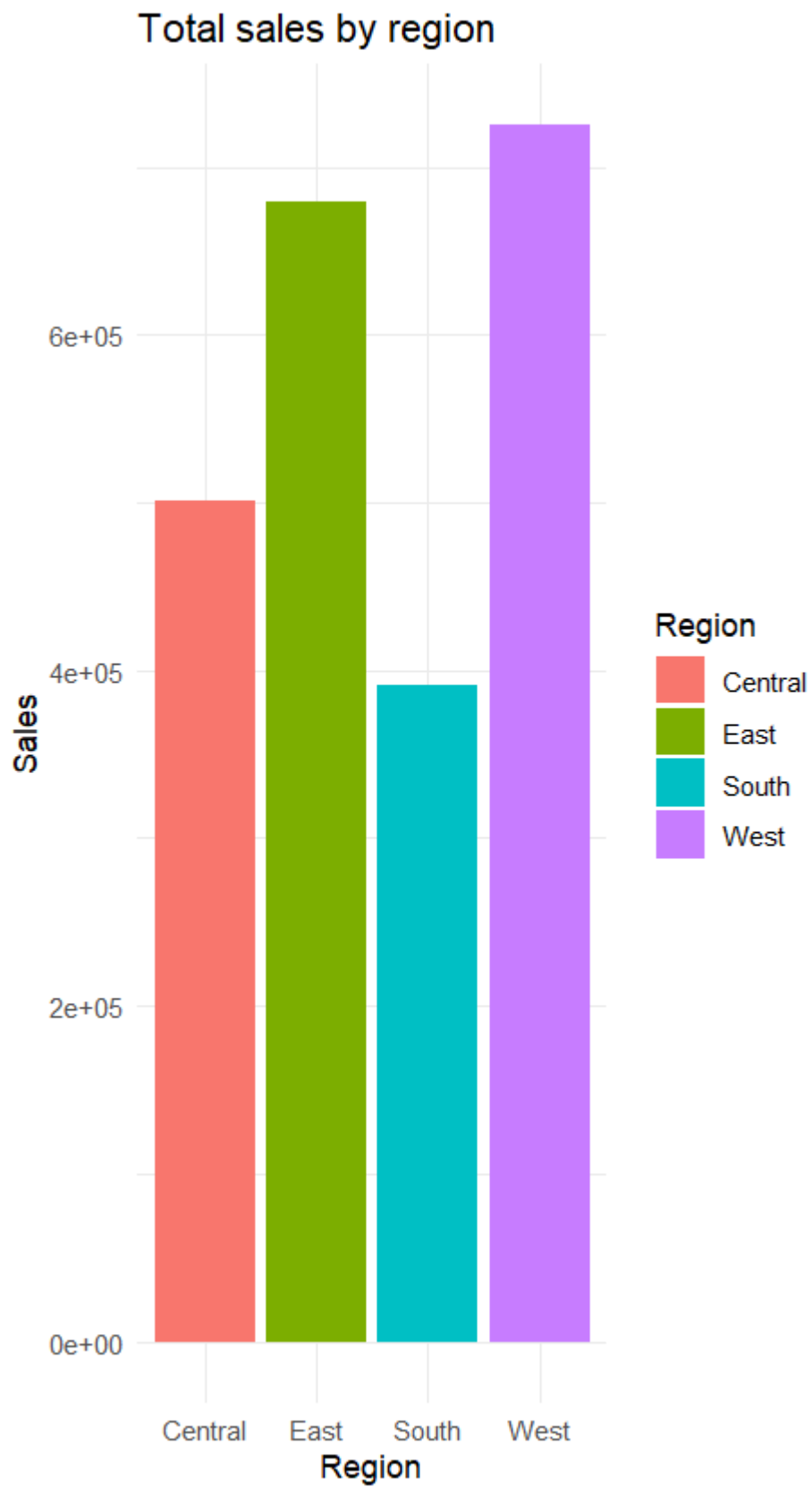


#Regional sales

```
ggplot(data,aes(x=Region,y=Sales,fill=Region))+geom_bar(stat='identity')+
```

```
labs(title = "Total sales by region",x="Region",y="Sales")+theme_minimal()
```

output:

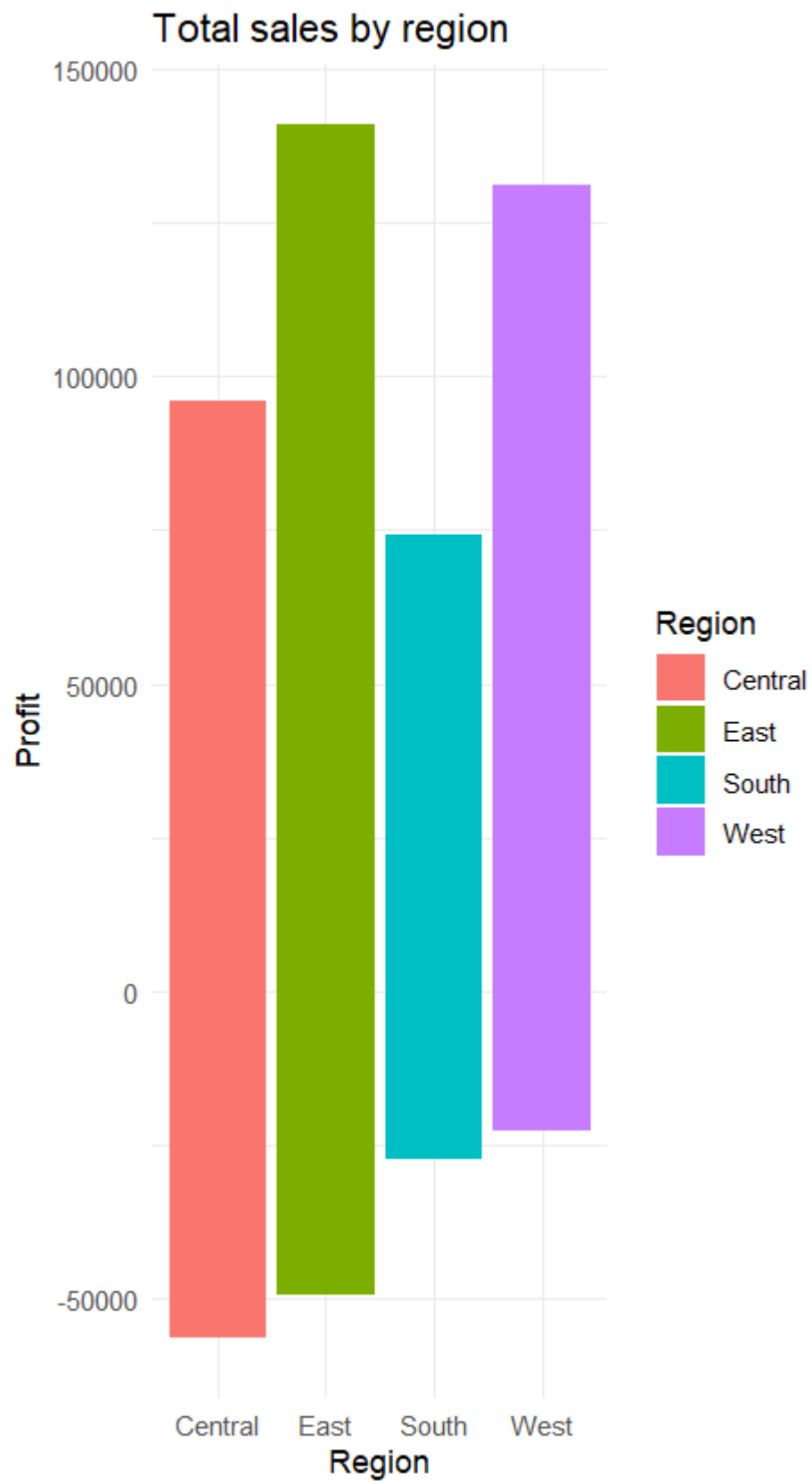


#Regional profit

```
ggplot(data,aes(x=Region,y=Profit,fill=Region))+geom_bar(stat = 'identity')+
```

```
labs(title = "Total sales by region",x="Region",y="Profit")+theme_minimal()
```

output:



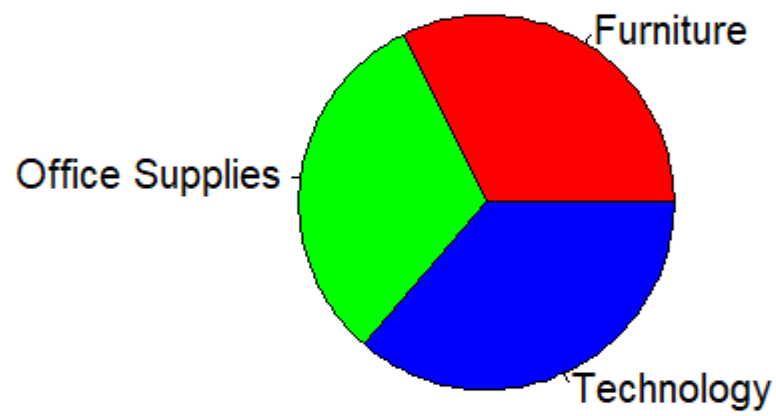
Pie chart of sales by Category

```
sales_by_category <- aggregate(Sales ~ Category, data, sum)
```

```
pie(sales_by_category$Sales, labels = sales_by_category$Category,  
    main = "Sales Distribution by Category",  
    col = rainbow(length(sales_by_category$Category)))
```

output:

Sales Distribution by Category

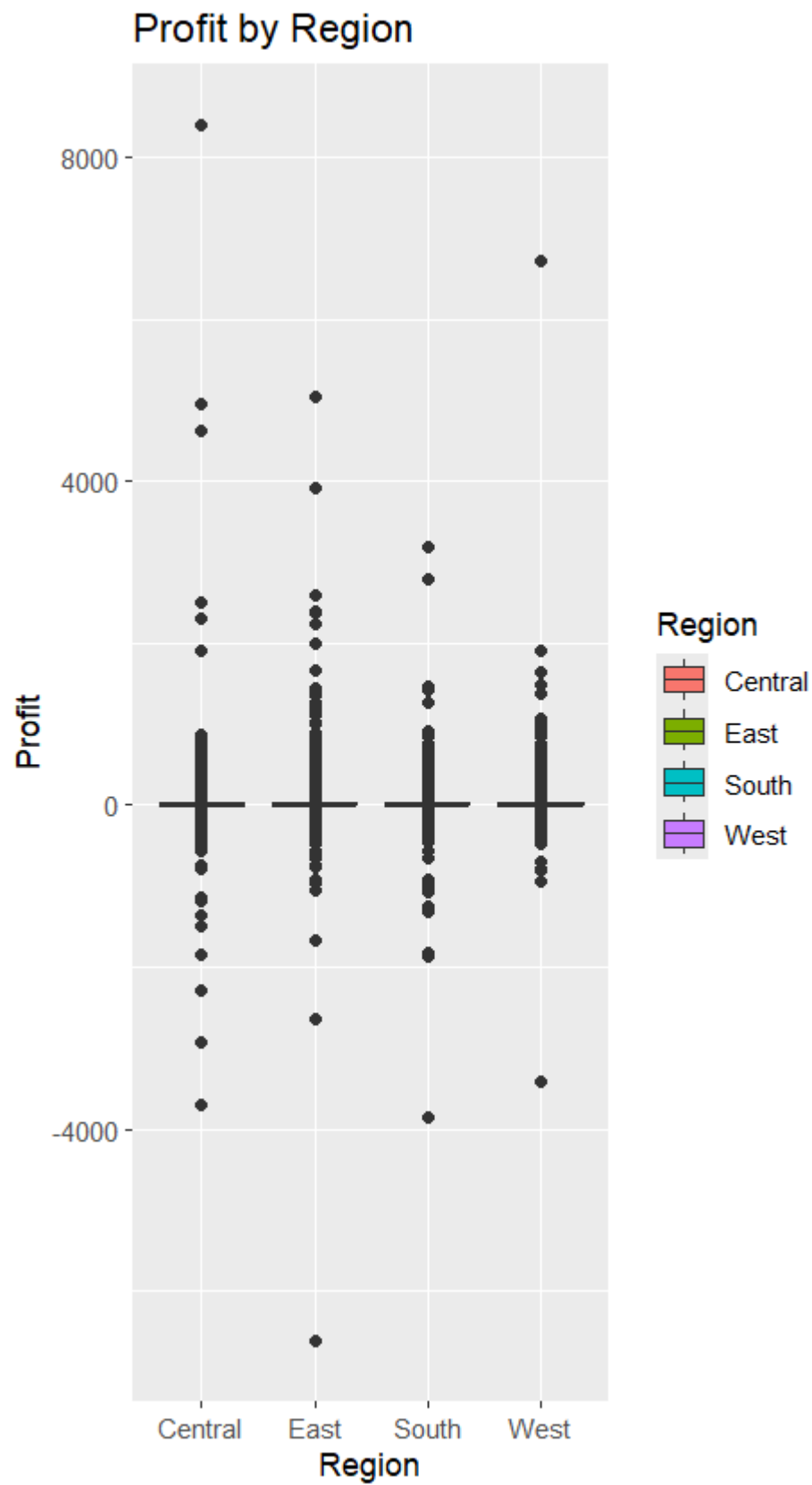


Boxplot of Profit by Region

```
ggplot(data, aes(x = Region, y = Profit, fill = Region)) +
```

```
geom_boxplot() +  
ggtitle("Profit by Region") +  
xlab("Region") +  
ylab("Profit")
```

output:

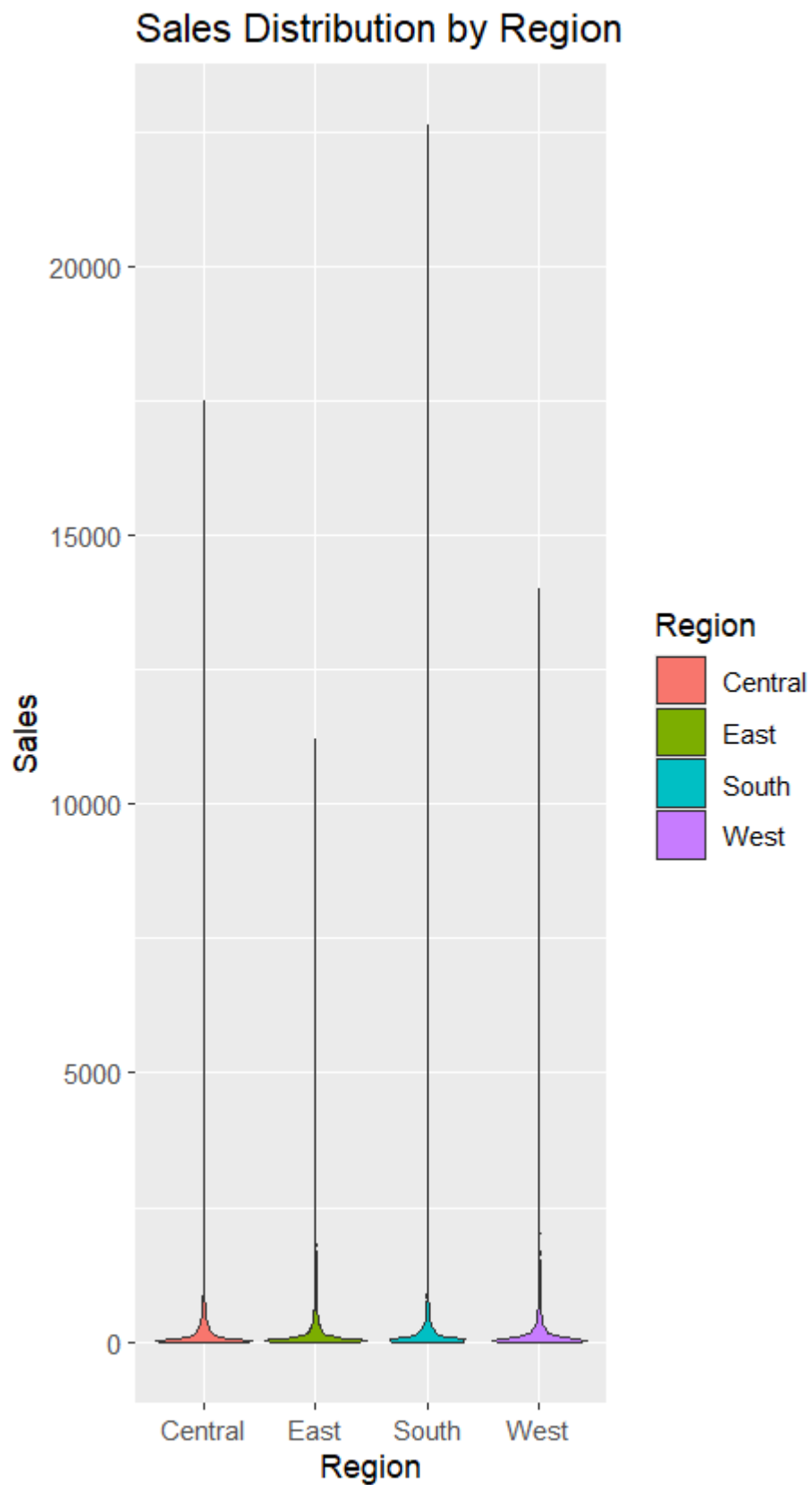


#violin plot sales over region

ggplot(data, aes(x = Region, y = Sales, fill = Region)) +

```
geom_violin() +  
ggtitle("Sales Distribution by Region") +  
xlab("Region") +  
ylab("Sales")
```

output:



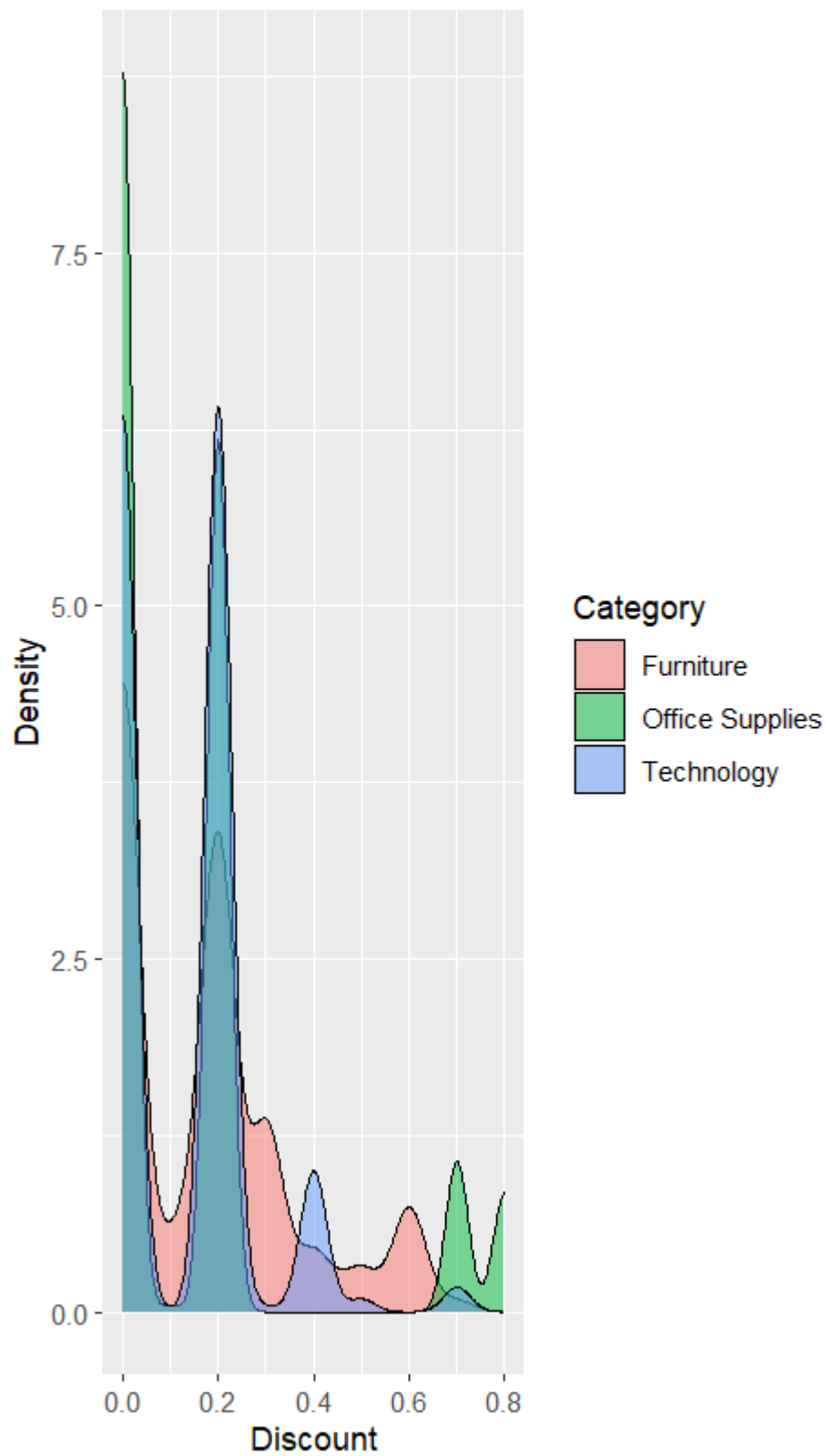
#Density plot over Discount Distribution by Category

ggplot(data, aes(x = Discount, fill = Category)) +

```
geom_density(alpha = 0.5) +  
ggtitle("Discount Distribution by Category") +  
xlab("Discount") +  
ylab("Density")
```

output:

Discount Distribution by Category

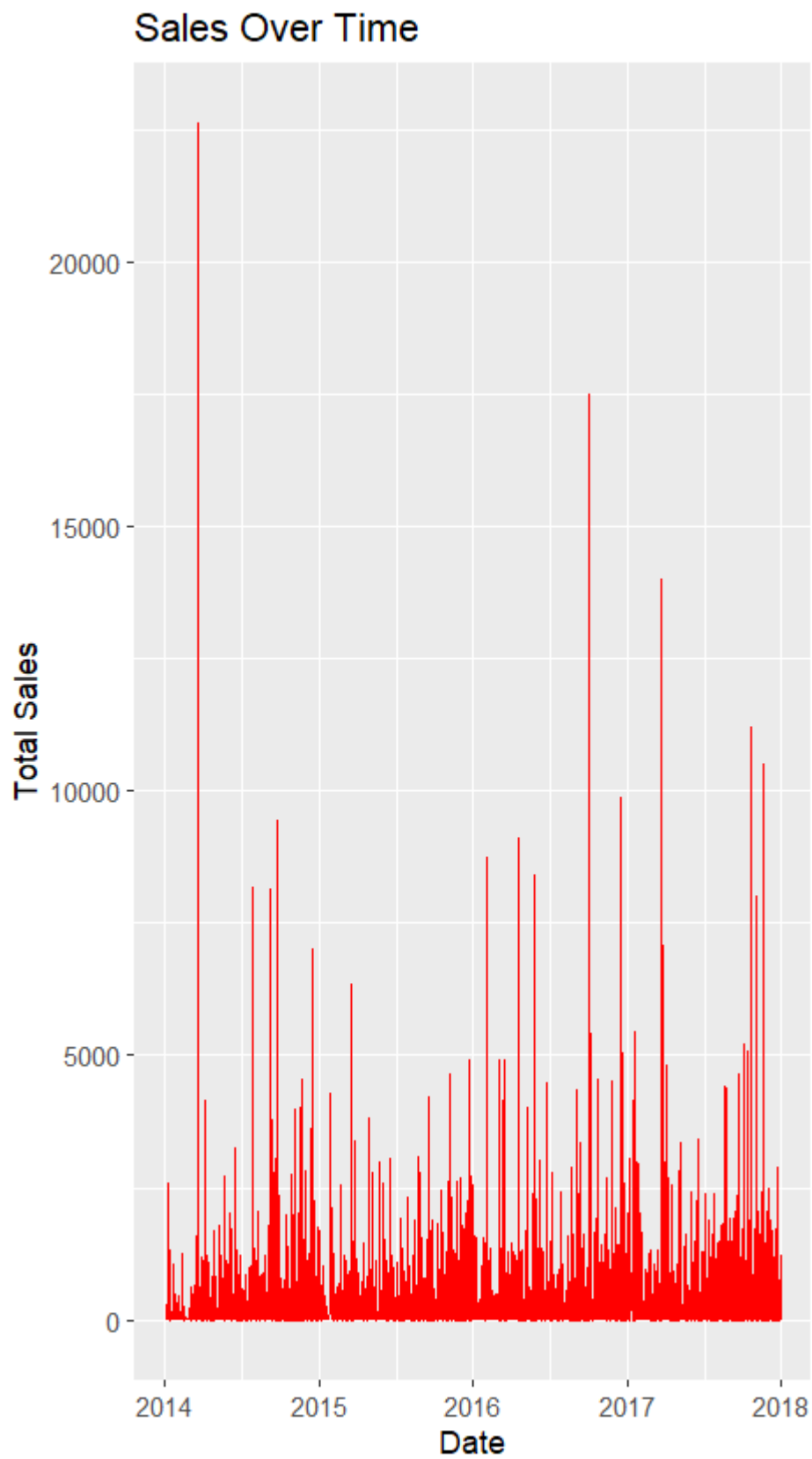


Line plot of Sales over time

ggplot(data, aes(x = `Order Date`, y = Sales)) +

```
geom_line(color = "red") +  
ggtitle("Sales Over Time") +  
xlab("Date") +  
ylab("Total Sales")
```

output:



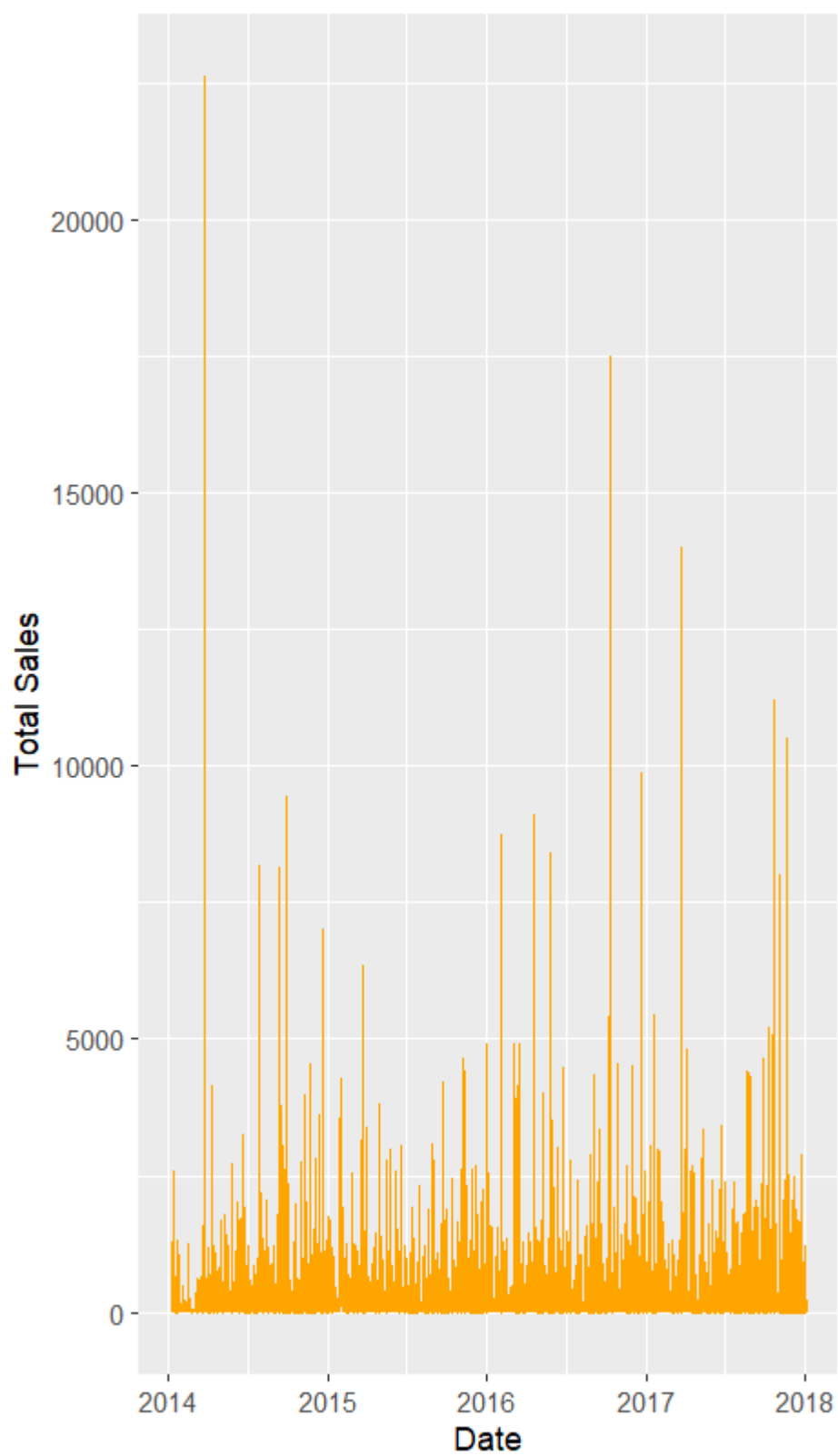
Line plot of Sales shipping time

```
ggplot(data, aes(x = `Ship Date`, y = Sales)) +
```

```
geom_line(color = "orange") +  
ggtitle("Sales Over Time") +  
xlab("Date") +  
ylab("Total Sales")
```

output:

Sales Over Time



Conclusion

Conclusion of the Superstore Data Analysis

1. No missing values were found in the dataset, ensuring completeness for analysis.

```
no_missing_values <- sum(is.na(data)) == 0
```

2. Category-wise sales analysis shows variation in sales across different categories.

```
category_sales_distribution <- ggplot(data, aes(x=Category, fill=Category)) +  
  geom_bar() +  
  labs(title="Category Sales Distribution")
```

3. The profit distribution indicates variability, with some transactions being more profitable.

```
profit_distribution <- ggplot(data, aes(x=Profit)) +  
  geom_histogram(binwidth = 50, fill = "green", color = "black") +  
  labs(title = "Distribution of Profit")
```

4. Sales and profit analysis by region shows distinct performance levels across regions.

```
regional_sales_profit <- list(  
  sales = ggplot(data, aes(x=Region, y=Sales, fill=Region)) + geom_bar(stat='identity') +  
    labs(title = "Total Sales by Region"),  
  profit = ggplot(data, aes(x=Region, y=Profit, fill=Region)) + geom_bar(stat='identity') +  
    labs(title = "Total Profit by Region")  
)
```

5. Time-series analysis reveals sales trends over time, potentially highlighting seasonal effects.

```
sales_trends_over_time <- ggplot(data, aes(x = `Order Date`, y = Sales)) +  
  geom_line(color = "red") +  
  labs(title = "Sales Over Time")
```

6. Discounts are applied variably across categories, as shown by the density plot.

```
discount_distribution_by_category <- ggplot(data, aes(x = Discount, fill = Category)) +  
  geom_density(alpha = 0.5) +  
  labs(title = "Discount Distribution by Category")
```

7. Regional performance variability is captured by the violin plot of sales distribution.

```
regional_sales_variability <- ggplot(data, aes(x = Region, y = Sales, fill = Region)) +  
  geom_violin() +  
  labs(title = "Sales Distribution by Region")
```

Key Insights

1. The superstore should focus on regions with higher sales and profit for targeted marketing.

2. Time-series trends could inform stocking and promotional strategies.

3. Discount strategies could be optimized based on their impact on sales and profit margins.