Revenue Forecast

Jaime Wu

22/08/2020

Load Libraries and Data

```
library(tidyverse)
library(ggplot2)
library(lubridate)
library(leaps)
library(readr)
library(ggplot2)
library(forecast)
library(fpp2)
library(TTR)
library(dplyr)
library(zoo)
```

Problem and Background

The online retail II data set contains all the invoice transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011. It is of interest to predict how much revenue the business will be earning on the month of December 2011, and whether the owner should purchase a new Ferrari for his partner as a Christmas gift.

Exploratory Data Analysis

```
# Load data
salesData <- read_csv("online_retail_II.csv")
head(salesData)
```

```
## # A tibble: 6 x 8
##   Invoice StockCode Description Quantity InvoiceDate         Price `Customer ID`
##   <chr>   <chr>     <chr>          <dbl> <dttm>              <dbl>         <dbl>
## 1 489434  85048     "15CM CHRI~       12 2009-12-01 07:45:00  6.95         13085
## 2 489434  79323P    "PINK CHER~       12 2009-12-01 07:45:00  6.75         13085
## 3 489434  79323W    "WHITE CHE~       12 2009-12-01 07:45:00  6.75         13085
## 4 489434  22041     "RECORD FR~       48 2009-12-01 07:45:00  2.1          13085
## 5 489434  21232     "STRAWBERR~       24 2009-12-01 07:45:00  1.25         13085
## 6 489434  22064     "PINK DOUG~       24 2009-12-01 07:45:00  1.65         13085
## # ... with 1 more variable: Country <chr>
```

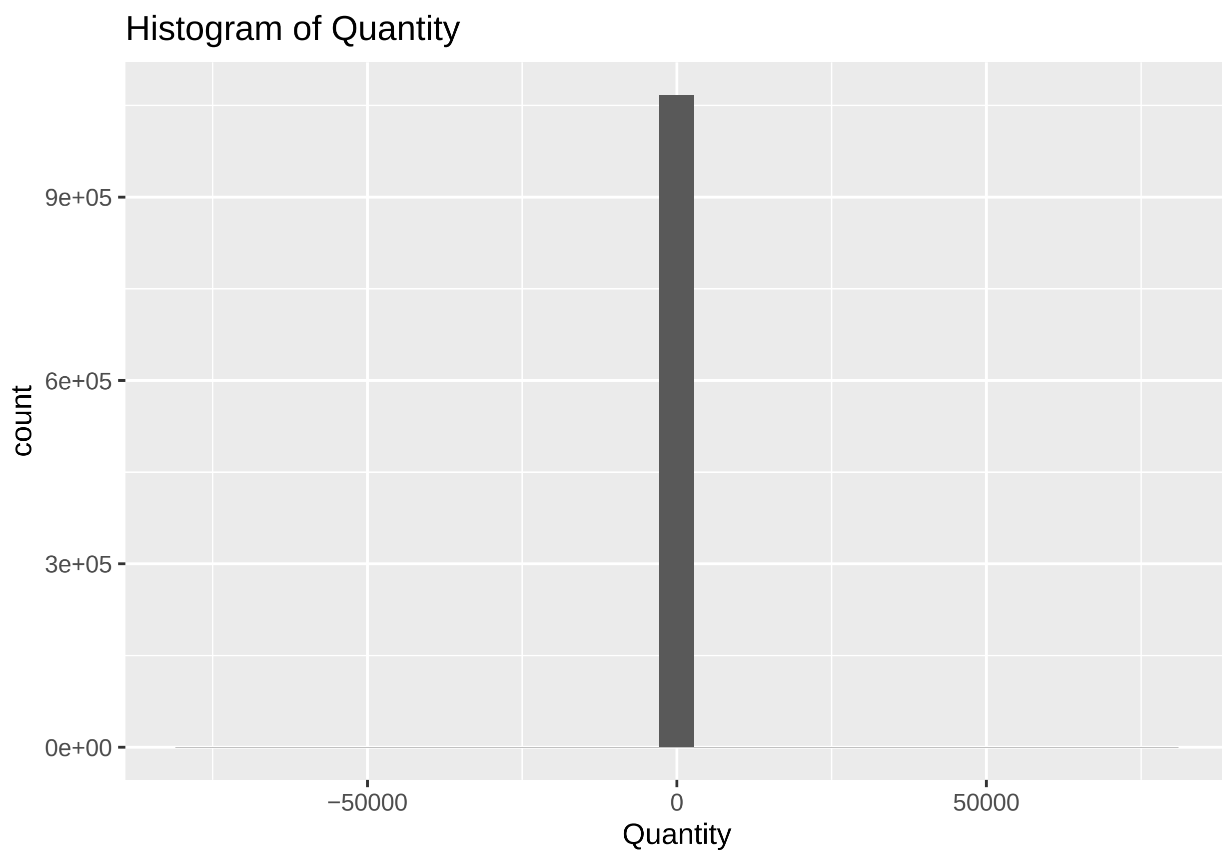
```r
summary(salesData)
```

```
##    Invoice            StockCode          Description           Quantity        
##  Length:1067371     Length:1067371     Length:1067371     Min.   :-80995.00  
##  Class :character   Class :character   Class :character   1st Qu.:     1.00  
##  Mode  :character   Mode  :character   Mode  :character   Median :     3.00  
##                                                           Mean   :     9.94  
##                                                           3rd Qu.:    10.00  
##                                                           Max.   : 80995.00  
##                                                                              
##   InvoiceDate                      Price            Customer ID    
##  Min.   :2009-12-01 07:45:00   Min.   :-53594.36   Min.   :12346   
##  1st Qu.:2010-07-09 09:46:00   1st Qu.:     1.25   1st Qu.:13975   
##  Median :2010-12-07 15:28:00   Median :     2.10   Median :15255   
##  Mean   :2011-01-02 21:13:55   Mean   :     4.65   Mean   :15325   
##  3rd Qu.:2011-07-22 10:23:00   3rd Qu.:     4.15   3rd Qu.:16797   
##  Max.   :2011-12-09 12:50:00   Max.   : 38970.00   Max.   :18287   
##                                                    NA's   :243007  
##    Country         
##  Length:1067371    
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
## 
```

A brief summary of the entire invoice data set shows that there are about 1,067,371 invoice transactions. Immediately it becomes evident that there is test data and erroneous entries within the data set since there are negative quantities and negative prices. Furthermore, the max and min values for both quantity and price are extremely large compared to the mean and median, suggesting outliers are present within the data set. Nevertheless, the following exploratory data analysis will be completed to visualise how much data cleaning is required: - Total daily, weekly and monthly sales volumes - Last months' revenue share by product and by customer - Weighted average monthly sale revenue by volume

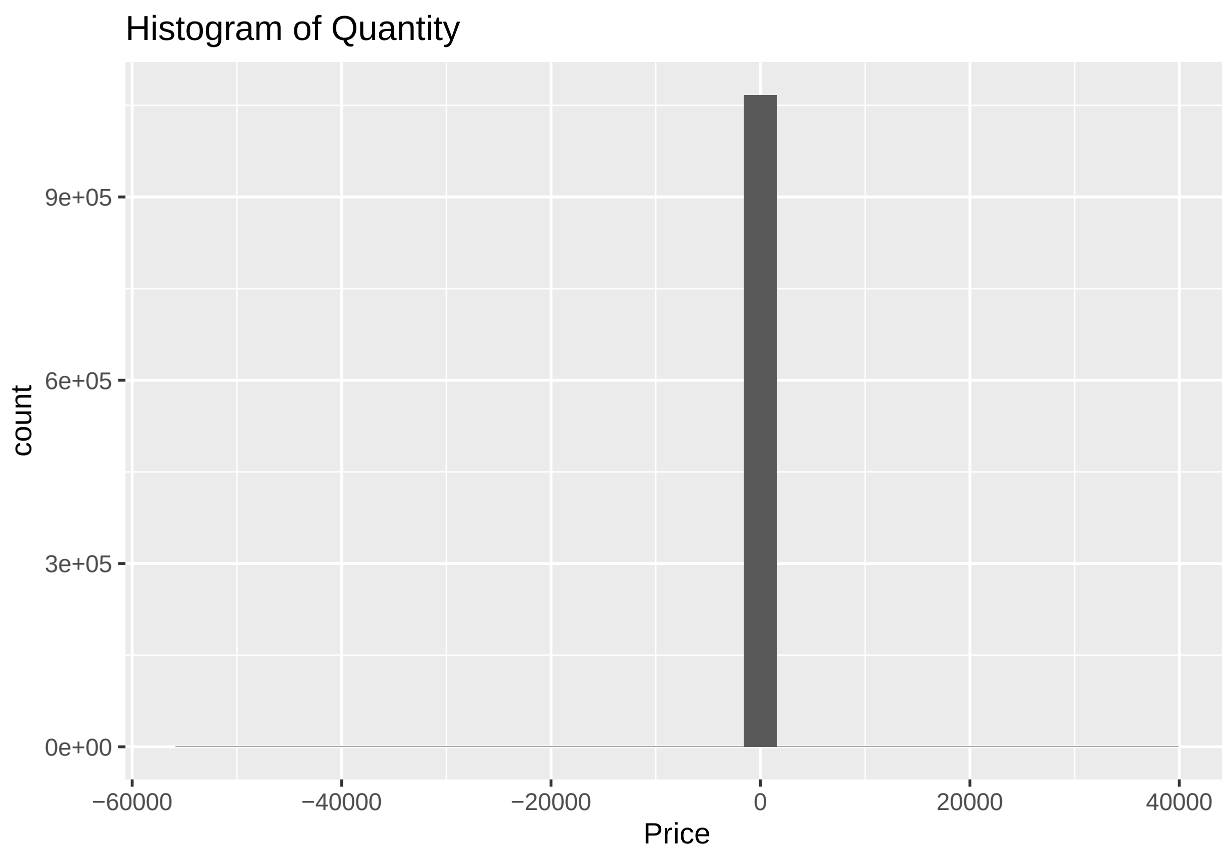
```r
# Brief data cleaning to remove outliers and erroneous entries
salesData %>%
  ggplot(aes(x = Quantity)) +
  geom_histogram() +
  labs(title = 'Histogram of Quantity')
```

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




```
salesData %>%
  ggplot(aes(x = Price)) +
  geom_histogram() +
  labs(title = 'Histogram of Quantity')
```

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



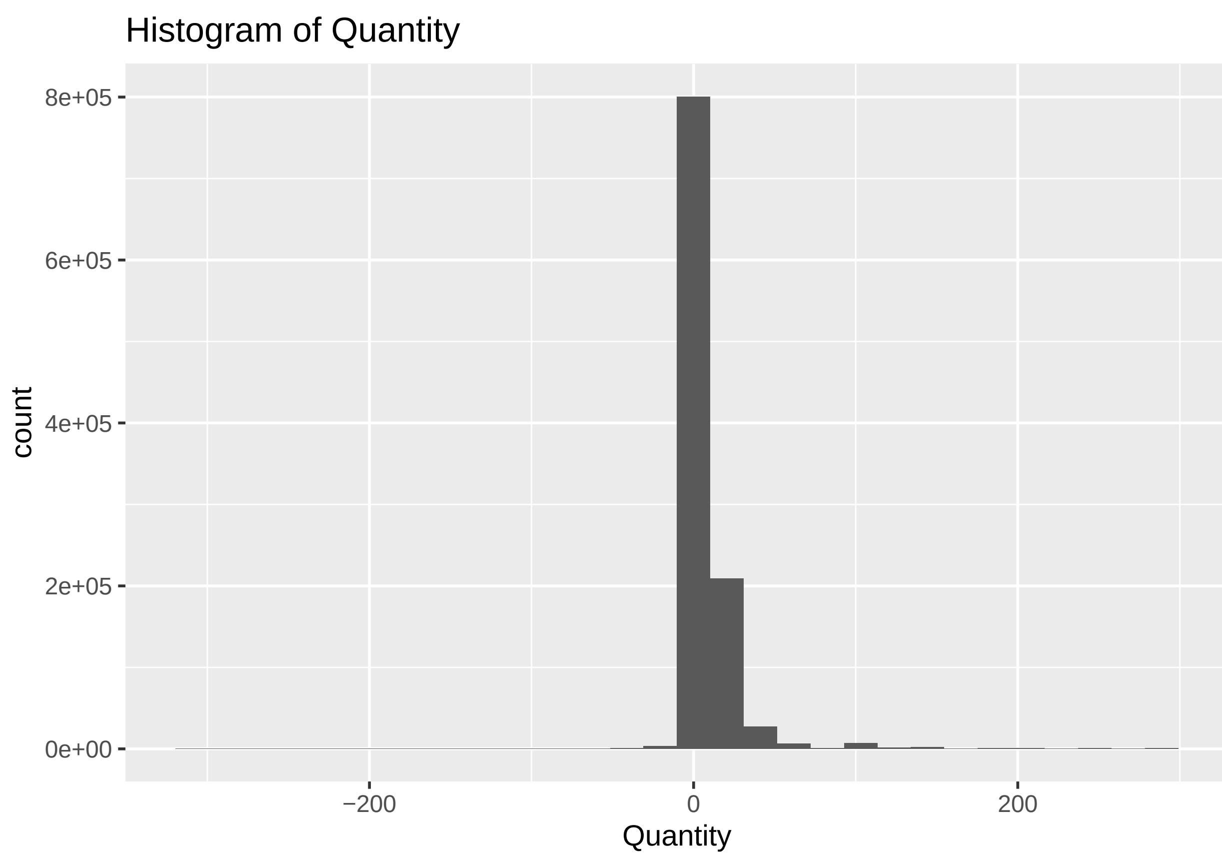

```r
salesData %>% arrange(Price)
```

```
## # A tibble: 1,067,371 x 8
##    Invoice StockCode Description Quantity InvoiceDate           Price
##    <chr>   <chr>     <chr>          <dbl> <dttm>                <dbl>
##  1 A506401 B         Adjust bad~        1 2010-04-29 13:36:00 -53594.
##  2 A516228 B         Adjust bad~        1 2010-07-19 11:24:00 -44032.
##  3 A528059 B         Adjust bad~        1 2010-10-20 12:04:00 -38926.
##  4 A563186 B         Adjust bad~        1 2011-08-12 14:51:00 -11062.
##  5 A563187 B         Adjust bad~        1 2011-08-12 14:52:00 -11062.
##  6 489464  21733     85123a mix~      -96 2009-12-01 10:52:00      0
##  7 489463  71477     short           -240 2009-12-01 10:52:00      0
##  8 489467  85123A    21733 mixed     -192 2009-12-01 10:53:00      0
##  9 489521  21646     <NA>             -50 2009-12-01 11:44:00      0
## 10 489655  20683     <NA>             -44 2009-12-01 17:26:00      0
## # ... with 1,067,361 more rows, and 2 more variables: `Customer ID` <dbl>,
## #   Country <chr>
```

An initial histogram plot of quantity and price indicates that there are outliers across both attributes and should be removed before analysis. It is important to realise that the negative quantities relate to sales returns; therefore, we should not disregard quantities less than zero. Any data with quantity less than -300 or greater than 300 will be filtered since it is possible for a customer to purchase a cheap product in bulk. In contrast, negative prices should be removed as it is likely that they relate to test data as suggested by the adjustment for bad debt entries. Therefore, data with as price less than 0 or greater than 100 will be filtered out.

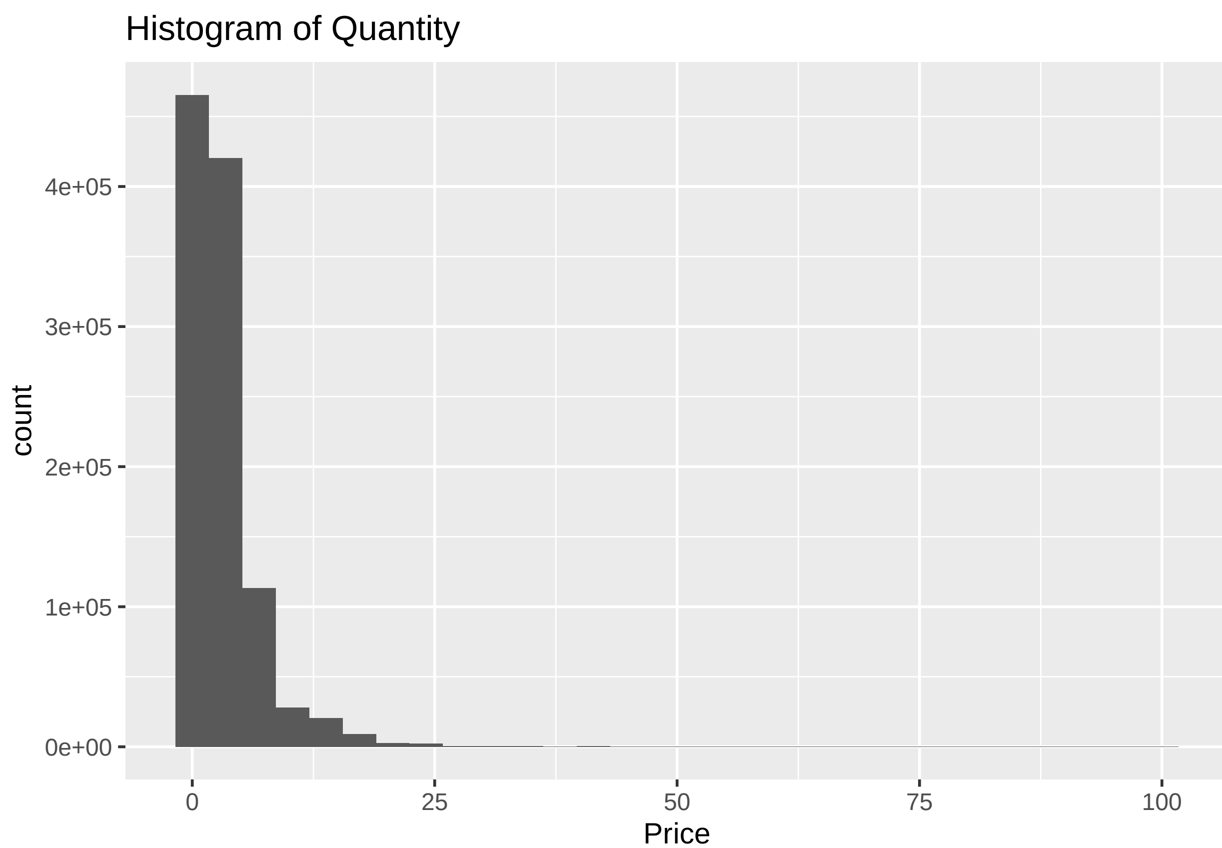
```r
salesData %>%
  filter(Quantity < 300, Quantity > -300) %>%
  ggplot(aes(x = Quantity)) +
  geom_histogram() +
  labs(title = 'Histogram of Quantity')
```

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




```r
salesData %>%
  filter(Price < 100, Price >= 0) %>%
  ggplot(aes(x = Price)) +
  geom_histogram() +
  labs(title = 'Histogram of Quantity')
```

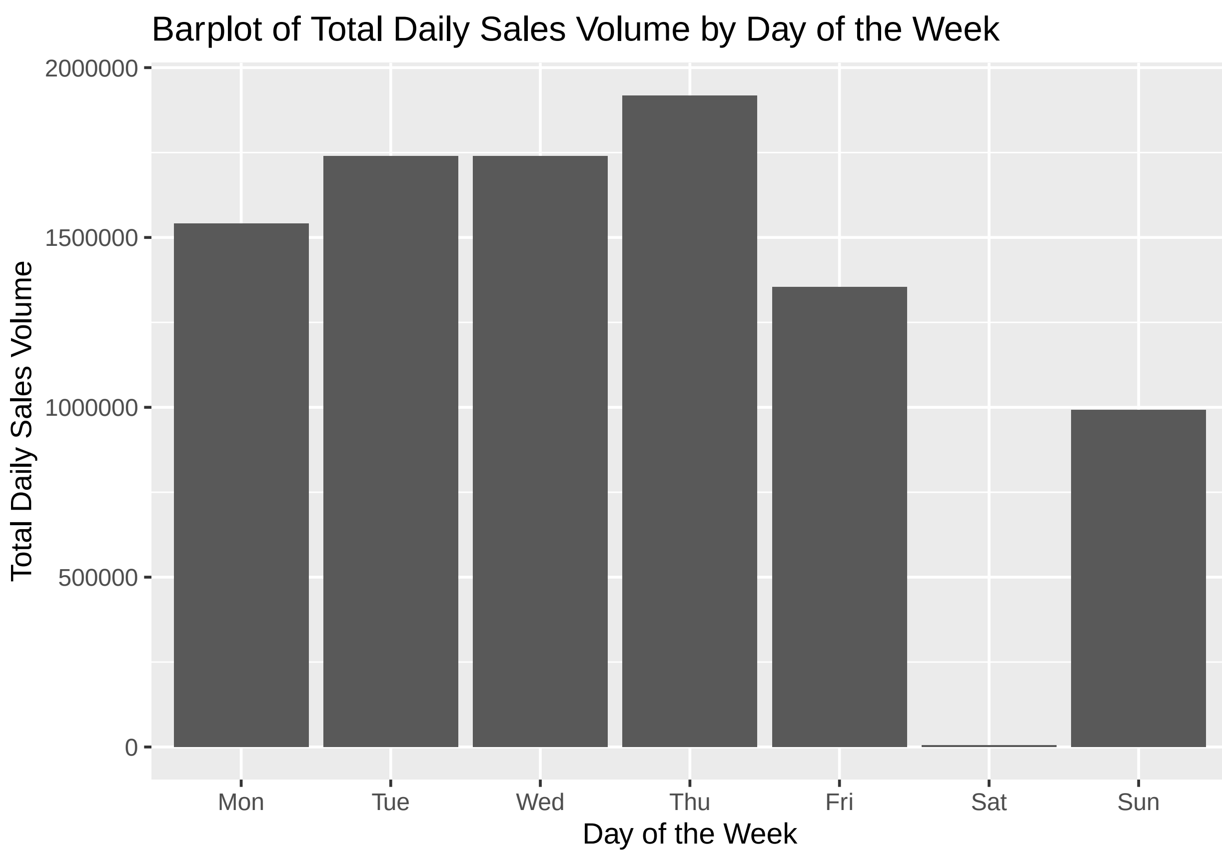
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



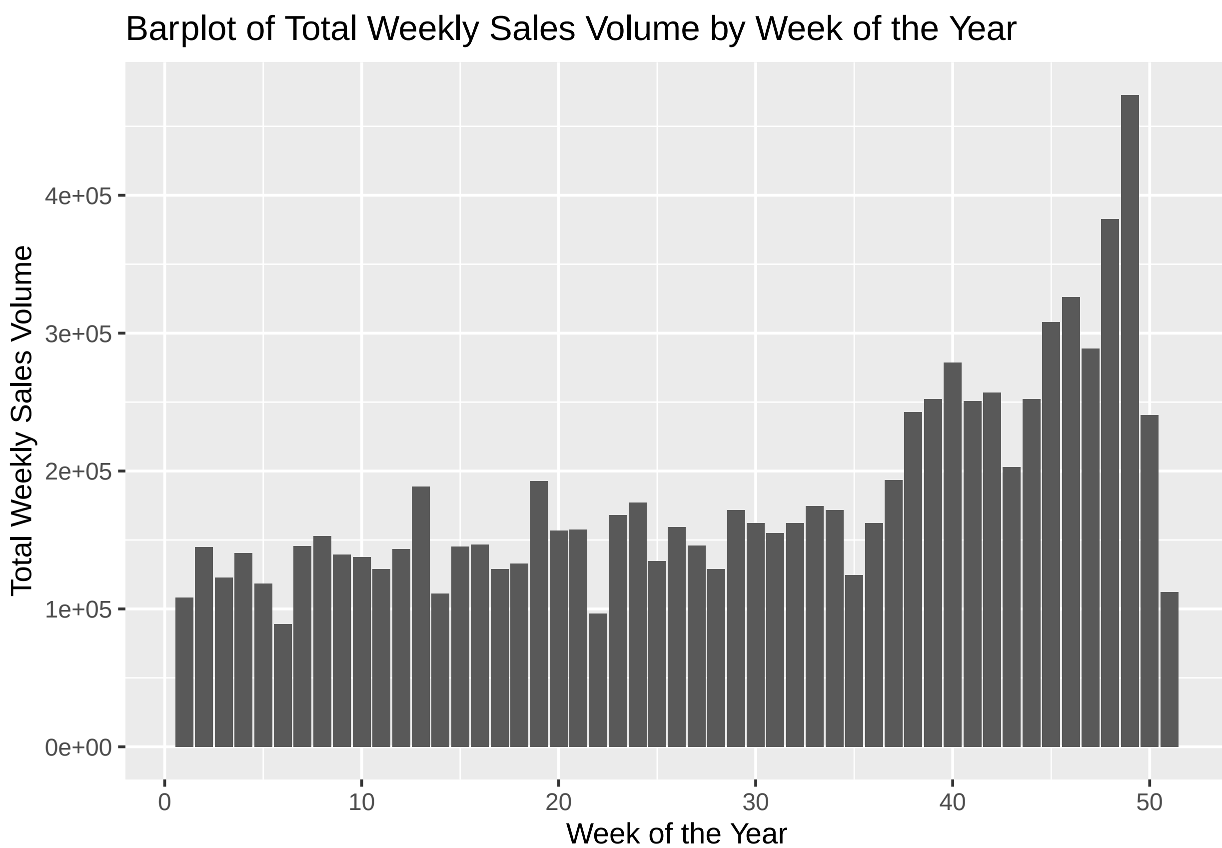
Once the erroneous data is removed, the histogram plots appear more realistic for the following exploratory data analysis: - Total daily, weekly and monthly sales volumes - Last months' revenue share by product and by customer - Weighted average monthly sale revenue by volume

```
# Exploratory Data Analysis
# Add day of the week, week number, month, and year variables to the data for plotting
salesDataC <- salesData %>%
  filter(
    Quantity > -300,
    Quantity < 300,
    Price >= 0,
    Price < 100) %>%
  mutate(
    DOW = wday(InvoiceDate, label = TRUE, abbr = TRUE, week_start = 1),
    Week = week(InvoiceDate),
    Month = month(InvoiceDate, label = TRUE, abbr = TRUE),
    Year = year(InvoiceDate)
  )

# Daily sales volume
salesDataC %>%
group_by(DOW) %>%
summarise(SalesVolume = sum(Quantity)) %>%
ggplot(aes(x = DOW, y = SalesVolume)) +
  geom_bar(stat = 'identity') +
  labs(x = 'Day of the Week',
```

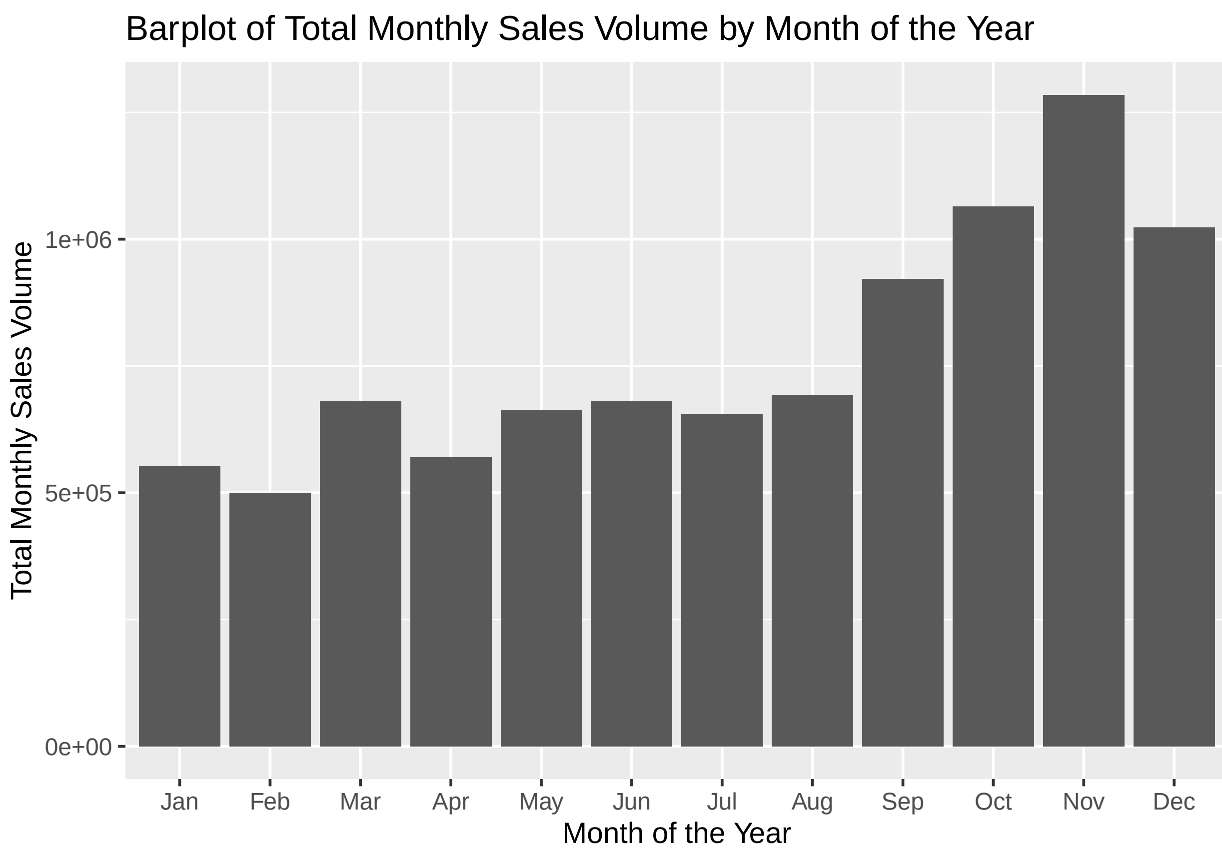
```
       y = 'Total Daily Sales Volume',
       title = 'Barplot of Total Daily Sales Volume by Day of the Week')
```



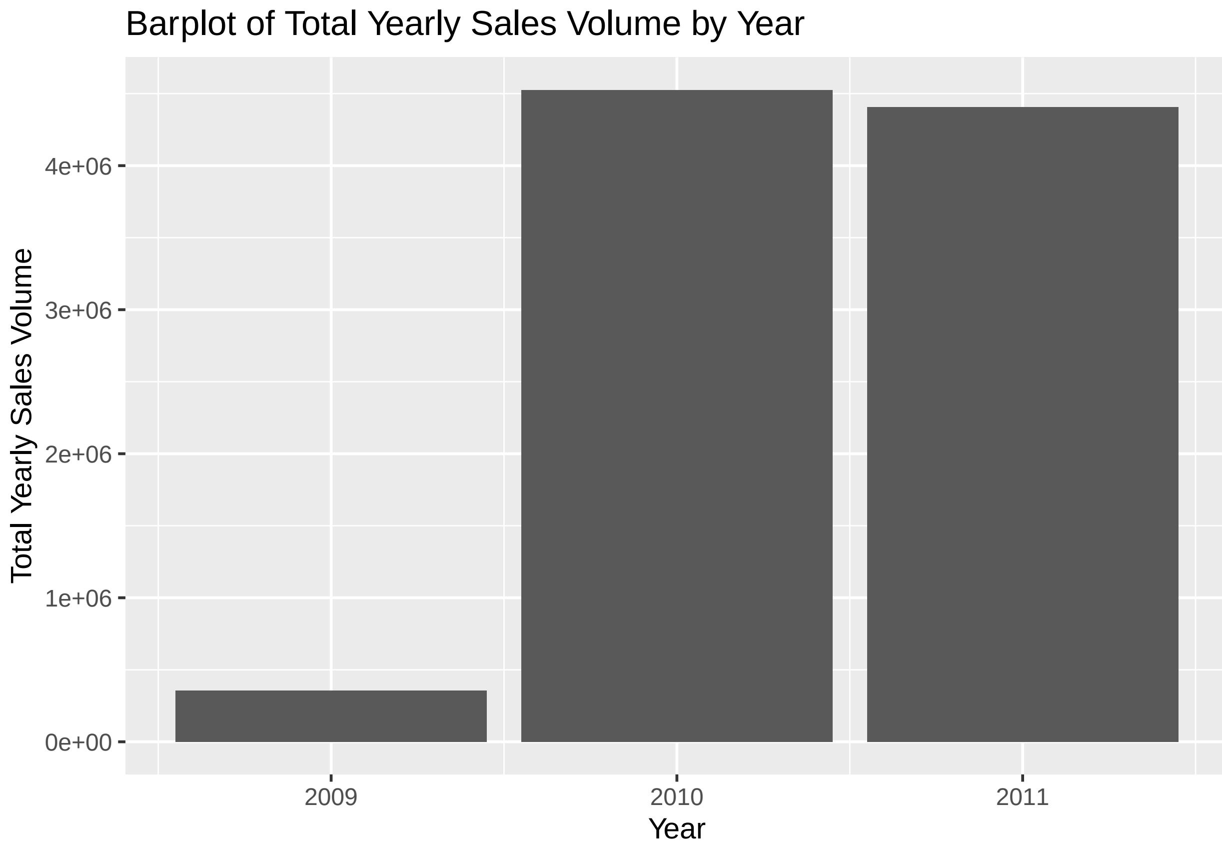

```
# Weekly sales volume
salesDataC %>%
group_by(Week) %>%
summarise(SalesVolume = sum(Quantity)) %>%
ggplot(aes(x = Week, y = SalesVolume)) +
  geom_bar(stat = 'identity') +
  labs(x = 'Week of the Year',
       y = 'Total Weekly Sales Volume',
       title = 'Barplot of Total Weekly Sales Volume by Week of the Year')
```




```
# Monthly sales volume
salesDataC %>%
group_by(Month) %>%
summarise(SalesVolume = sum(Quantity)) %>%
ggplot(aes(x = Month, y = SalesVolume)) +
  geom_bar(stat = 'identity') +
  labs(x = 'Month of the Year',
       y = 'Total Monthly Sales Volume',
       title = 'Barplot of Total Monthly Sales Volume by Month of the Year')
```




```r
# Yearly sales volume
salesDataC %>%
group_by(Year) %>%
summarise(SalesVolume = sum(Quantity)) %>%
ggplot(aes(x = Year, y = SalesVolume)) +
  geom_bar(stat = 'identity') +
  labs(x = 'Year',
       y = 'Total Yearly Sales Volume',
       title = 'Barplot of Total Yearly Sales Volume by Year')
```



The total daily sales volume against the day of the week shows an abnormally low amounts of sales on Saturdays. It is unlikely that the online retailer is closed on Saturday, and could be potentially due to an error during the data collection process. Nevertheless, the plot suggests that most of the sales is made during the middle of the week then trails off in the weekend.

The weekly and monthly sales volume plot indicates the sales volume is roughly the same for the first 7-8 months of the year, but increases near the end of the year as we approach Christmas, suggesting the data is seasonal.

The sales sales volume over each year is consistent to the amount of data we have for each year.

```
# Last months revenue shared by product and by customer
salesDataC %>%
  filter(Year == 2011, Month == 'Nov') %>%
  group_by(StockCode) %>%
  summarise(Revenue = sum(Price*Quantity), Description = first(Description)) %>%
  arrange(desc(Revenue), by_group = TRUE)
```

```
## # A tibble: 2,958 x 3
##    StockCode Revenue Description
##    <chr>       <dbl> <chr>
##  1 23084      22805. RABBIT NIGHT LIGHT
##  2 22086      18178. PAPER CHAIN KIT 50'S CHRISTMAS
##  3 22910      12832. PAPER CHAIN KIT VINTAGE CHRISTMAS
##  4 22423      12798. REGENCY CAKESTAND 3 TIER
##  5 22114      10013. HOT WATER BOTTLE TEA AND SYMPATHY
##  6 23355      10007. HOT WATER BOTTLE KEEP CALM
```

```
##  7 POST        9468. POSTAGE
##  8 79321       9243. CHILLI LIGHTS
##  9 85123A      9156. WHITE HANGING HEART T-LIGHT HOLDER
## 10 84347       8590. ROTATING SILVER ANGELS T-LIGHT HLDR
## # ... with 2,948 more rows
```

```
salesDataC %>%
  filter(Year == 2011, Month == 'Nov') %>%
  group_by(`Customer ID`) %>%
  summarise(Revenue = sum(Price*Quantity)) %>%
  arrange(desc(Revenue), by_group = TRUE)
```

```
## # A tibble: 1,702 x 2
##    `Customer ID` Revenue
##            <dbl>   <dbl>
##  1            NA 276973.
##  2         14646  24225.
##  3         14096  22622.
##  4         14911  22536.
##  5         17450  17502.
##  6         14088  16852.
##  7         17389  11505.
##  8         17511  10991.
##  9         18102  10773.
## 10         13081   9414.
## # ... with 1,692 more rows
```

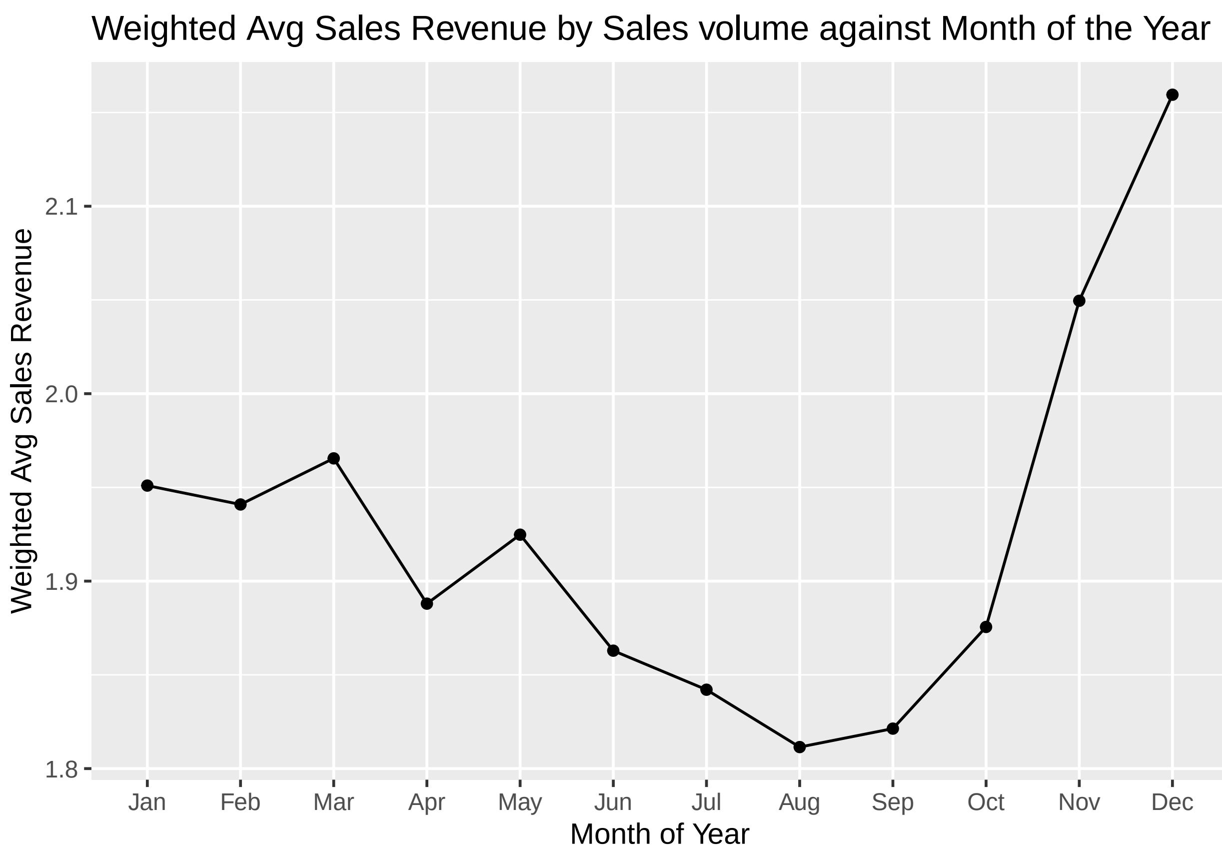
```
salesDataC %>%
  filter(Year == 2011, Month == 'Nov') %>%
  group_by(StockCode, `Customer ID`) %>%
  summarise(Revenue = sum(Price*Quantity), Description = first(Description)) %>%
  arrange(desc(Revenue), by_group = TRUE)
```

```
## # A tibble: 57,910 x 4
## # Groups:   StockCode [2,958]
##    StockCode `Customer ID` Revenue Description
##    <chr>             <dbl>   <dbl> <chr>
##  1 23084                NA   6180. RABBIT NIGHT LIGHT
##  2 22114                NA   5123. HOT WATER BOTTLE TEA AND SYMPATHY
##  3 22086                NA   4742. PAPER CHAIN KIT 50'S CHRISTMAS
##  4 84347                NA   4346. ROTATING SILVER ANGELS T-LIGHT HLDR
##  5 22910                NA   4290. PAPER CHAIN KIT VINTAGE CHRISTMAS
##  6 22355                NA   3341. CHARLOTTE BAG SUKI DESIGN
##  7 23328                NA   3148. SET 6 SCHOOL MILK BOTTLES IN CRATE
##  8 22947                NA   2993. WOODEN ADVENT CALENDAR RED
##  9 23343                NA   2879. JUMBO BAG VINTAGE CHRISTMAS
## 10 22423                NA   2735. REGENCY CAKESTAND 3 TIER
## # ... with 57,900 more rows
```

The top product on Nov of 2011 appears to be Rabbit night light at a revenue of $22,805.25. The top customer appears to be 14646, spending $24,225.33. The customer that spent the most on Nov 2011 on a product is 15061, with $1664.40 spent on Regency Cakestand 3 Tier.

```r
# Monthly weighted average sales revenue by sales volume
salesDataC %>%
  group_by(Month) %>%
  summarise(WtAvgSalePriceByVolume = sum(Price*Quantity)/sum(Quantity)) %>%
  ggplot(aes(x = Month, y = WtAvgSalePriceByVolume)) +
    geom_point() +
    geom_line(aes(group = 1)) +
    labs(x = 'Month of Year',
         y = 'Weighted Avg Sales Revenue',
         title = 'Weighted Avg Sales Revenue by Sales volume against Month of the Year')
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



The trend is quite interesting as during the holiday seasons, people are not only purchasing more products but also purchasing more expensive products compared to other months of the year. It is also interesting to see that after Christmas, people are purchasing much cheaper products relative to the amount of products being purchased.

Clean Data - Deal With Sales Returns

A negative value for quantity indicate a sales return, and some of these returns relate to products sold before the data collection date, thus should be filtered out. If a sales return is made in the current year, then there should be a corresponding sales invoice with positive quantity, and every other variable should also have the exact same value. Therefore, the sales return should be less than the quantity in sales invoice unless the customer has purchased items before data collection date in which case we should remove the sales return. Further cleaning could involve data imputation on missing values in the price or quantity attributes or searching through the descriptions of each product for obvious signs of test data.

```
# Remove observations with no quantity
missing <- is.na(salesDataC$Quantity)
salesDataC <- salesDataC[!missing, ]

# Find all sales returns
salesReturns <- salesDataC$Quantity < 0

for (i in 1:length(salesReturns)) {
  if (salesReturns[i]) {
    # For each sales return, find the corresponding sales invoice
    # If we cannot find a corresponding sales invoice then the sales return
    # relates to the period prior to data collection and should be removed
     match <- salesDataC %>%
       filter(
         StockCode == salesDataC$StockCode[i],
         `Customer ID` == salesDataC$`Customer ID`[i],
         Price == salesDataC$Price[i],
         Country == salesDataC$Country[i],
         Quantity == salesDataC$Quantity[i]*-1)
    if (dim(match)[1] > 0) {
      salesReturns[i] <- FALSE
    }
  }
}

salesDataC <- salesDataC[!salesReturns, ]
write_csv(salesDataC, "salesDataClean.csv")
```

Forecasting

To forecast the revenue for December 2011 we will need to fit the data to a model for prediction. The simplest approach would be to fit the revenue against time in a linear regression model. However, as the sales data is time series a better approach would be identifying trends from a time series regression analysis.

The metrics of interest would be the total daily revenue between the periods 01/12/2009 and 09/12/2011. We can obtain the daily total revenue by simply multiplying the price and quantity to get revenue then group by the invoice date.

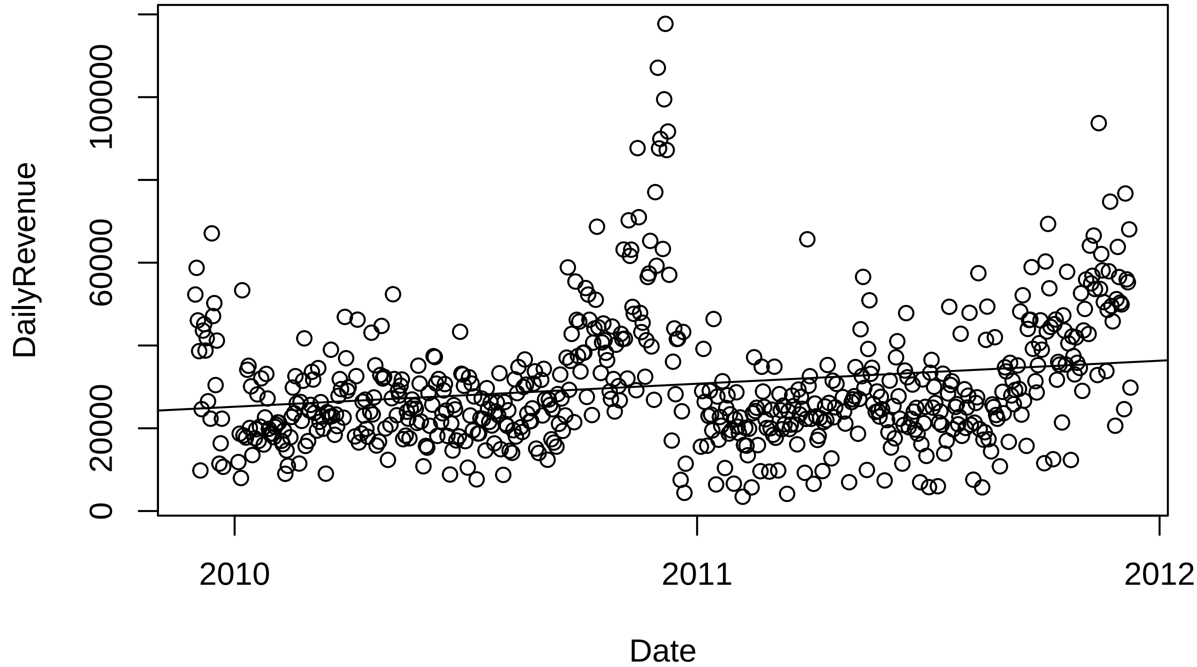
We can achieve basic forecasting using a naive model, a simple exponential smoothing model, and an ARIMA model. A naive model will use the most recent observation as the forecast for the next observation. It is not wise to assume that the future revenue will be reflective of the past revenue since seasonal effects can be seen in the exploratory analysis. A simple exponentially smoothing model could be fitted to account for the trend and seasonality of the data; however, the best model would likely to be an ARIMA model since it also takes into account for autocorrelation, the time lag between observations. As with all predictions, uncertainty will arise, so a 95% prediction interval will be used as relative good gauge of the predicted revenue, assuming the owner is risk adverse.

```
# Prepare data
salesDataClean <- read_csv("salesDataClean.csv")
```

```r
# Aggregate metrics for prediction
salesDataPred <- salesDataClean %>%
  mutate(
    Revenue = Quantity*Price,
    Date = as.Date(InvoiceDate)
  ) %>%
  filter(Revenue > 0) %>%
  group_by(Date) %>%
  summarise(DailyRevenue = sum(Revenue)) # Compute daily revenue

# Create a time series object from start date of 01/12/2009 to 09/12/2011
# and impute any missing data using zoo
dates <- seq(as.Date("2009-12-01"), as.Date("2011-12-09"), by = "day")
datesTable <- tibble("Date" = dates)
salesDataPred <- left_join(datesTable, salesDataPred, by = "Date")
salesDataPredTs <- zoo(salesDataPred$DailyRevenue, dates)
salesDataPredTs <- na.approx(salesDataPredTs)

# Fit a simple linear model for comparison
linearModel <- lm(DailyRevenue ~ Date, data = salesDataPred)
summary(linearModel)
```

```
##
## Call:
## lm(formula = DailyRevenue ~ Date, data = salesDataPred)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -28471  -9483  -3310   6945  87328
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.974e+05  4.422e+04  -4.463 9.63e-06 ***
## Date         1.524e+01  2.957e+00   5.152 3.50e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15510 on 602 degrees of freedom
##   (135 observations deleted due to missingness)
## Multiple R-squared:  0.04223,    Adjusted R-squared:  0.04064
## F-statistic: 26.54 on 1 and 602 DF,  p-value: 3.501e-07
```

```r
plot(DailyRevenue ~ Date, data = salesDataPred)
abline(linearModel$coef[1], linearModel$coef[2])
```



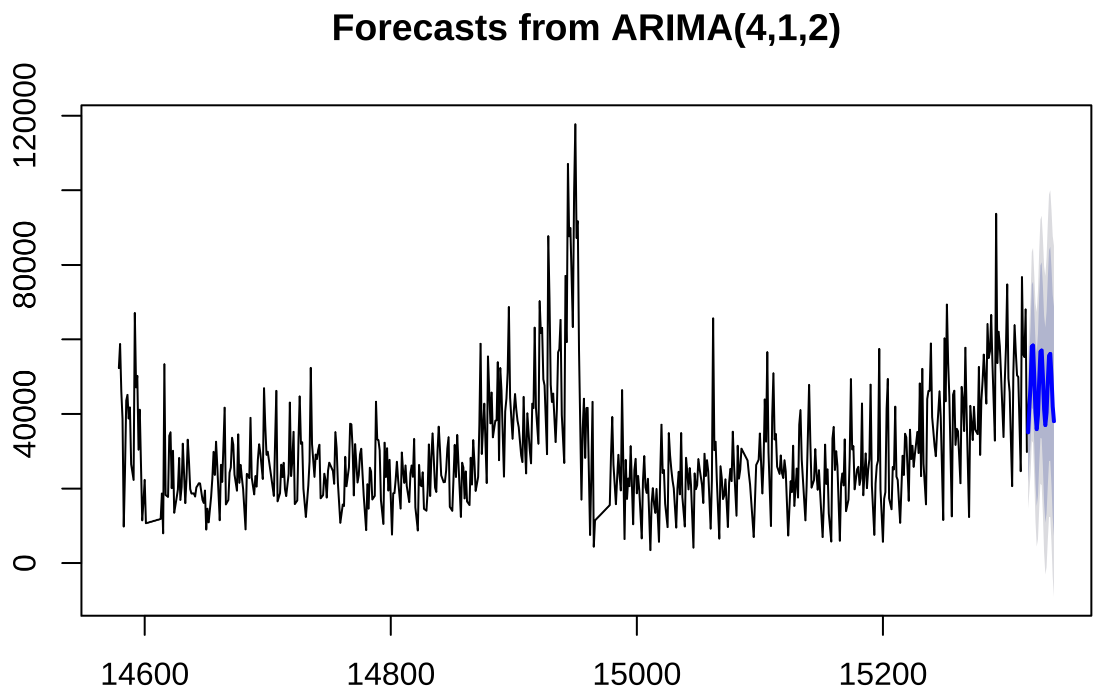

```r
pred = tibble(Date = seq(as.Date("2011-12-10"), as.Date("2011-12-31"), by = "day"))
linearModelPredictions <- predict(linearModel, pred, interval = "prediction")

# Fit an ARIMA model using the time series data
arimaModel <- auto.arima(salesDataPredTs, seasonal = TRUE)
summary(arimaModel)
```

```
## Series: salesDataPredTs
## ARIMA(4,1,2)
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ma1     ma2
##       0.6298  -0.5238  -0.2184  -0.2887  -1.1865  0.8199
## s.e.  0.0440   0.0573   0.0427   0.0415   0.0244  0.0690
##
## sigma^2 estimated as 109216664:  log likelihood=-7874.98
## AIC=15763.97   AICc=15764.12   BIC=15796.19
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -10.66552 10401.06 7639.007 -12.37161 33.13503 0.8444661
##                     ACF1
## Training set -0.04827965
```

```r
forecastArima <- forecast(arimaModel, h = 22)
summaryModel <- tibble(summary(forecastArima))
```

```
##
## Forecast method: ARIMA(4,1,2)
##
## Model Information:
## Series: salesDataPredTs
## ARIMA(4,1,2)
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ma1     ma2
##       0.6298  -0.5238  -0.2184  -0.2887  -1.1865  0.8199
## s.e.  0.0440   0.0573   0.0427   0.0415   0.0244  0.0690
##
## sigma^2 estimated as 109216664:  log likelihood=-7874.98
## AIC=15763.97   AICc=15764.12   BIC=15796.19
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -10.66552 10401.06 7639.007 -12.37161 33.13503 0.8444661
##                     ACF1
## Training set -0.04827965
##
## Forecasts:
##       Point Forecast     Lo 80    Hi 80     Lo 95     Hi 95
## 15318       35051.78 21658.694 48444.86 14568.824  55534.73
## 15319       41737.57 27087.353 56387.80 19331.994  64143.15
## 15320       47871.85 32323.608 63420.09 24092.865  71650.84
## 15321       58130.26 41560.656 74699.87 32789.235  83471.29
## 15322       58407.43 41314.049 75500.81 32265.360  84549.50
## 15323       49938.43 32049.311 67827.54 22579.385  77297.47
## 15324       40447.61 21291.078 59604.14 11150.223  69745.00
## 15325       35884.01 15324.581 56443.45  4441.075  67326.95
## 15326       39751.03 17957.224 61544.84  6420.278  73081.79
## 15327       49095.15 26461.807 71728.49 14480.440  83709.86
## 15328       56691.54 33526.745 79856.34 21264.042  92119.04
## 15329       57054.22 33425.936 80682.51 20917.877  93190.57
## 15330       50146.11 25906.261 74385.96 13074.459  87217.76
## 15331       41248.29 16123.180 66373.41  2822.748  79673.84
## 15332       36990.47 10806.440 63174.50 -3054.549  77035.49
## 15333       40373.81 13234.657 67512.97 -1131.944  81879.57
## 15334       48672.96 20854.320 76491.59  6128.024  91217.89
## 15335       55626.58 27348.780 83904.37 12379.419  98873.73
## 15336       56149.16 27468.241 84830.08 12285.479 100012.84
## 15337       50046.35 20852.657 79240.04  5398.451  94694.24
## 15338       42014.05 12098.660 71929.44 -3737.592  87765.69
## 15339       38030.17  7253.174 68807.17 -9039.185  85099.53
```

```r
plot(forecastArima)
```




```r
# Calculate expected revenue with 95% prediction intervals
earnedRevenue <- salesDataPred %>%
  filter(Date > as.Date("2011-11-30")) %>%
  .$DailyRevenue %>%
  sum(na.rm = TRUE)
expectedRevenue <- sum(summaryModel$`Point Forecast`) + earnedRevenue
expectedRevenueLow <- sum(summaryModel$`Lo 95`) + earnedRevenue
expectedRevenueHigh <- sum(summaryModel$`Hi 95`) + earnedRevenue
tibble(`Low Expected Revenue` = expectedRevenueLow,
       `Expected Revenue` = expectedRevenue,
       `High Expected Revenue` = expectedRevenueHigh)
```

```
## # A tibble: 1 x 3
##   `Low Expected Revenue` `Expected Revenue` `High Expected Revenue`
##                    <dbl>              <dbl>                   <dbl>
## 1                670198.           1440130.                2210062.
```

The expected revenue to be earned by the online retailer on December 2011 is about $1,440,130 with a low and high prediction of $670,198 and $2,210,062. Assuming that a new Ferrari is approximately $400,000 NZD, as listed on the official Auckland Ferrari dealers website, I recommend him to purchase the new Ferrari. I strongly back my recommendation as the 95% prediction interval is above the price of a new Ferrari.