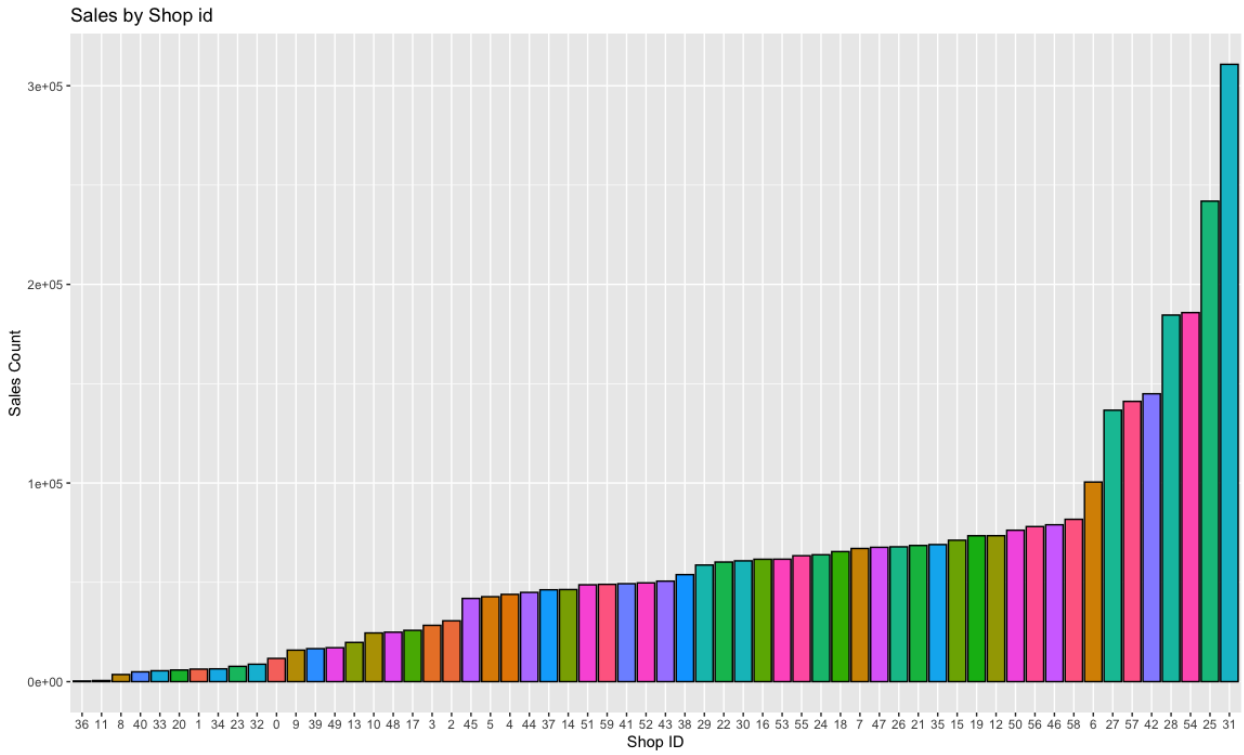
Brian Moran
May 5th 4:35pm
Partners: none

Data Preperation

```
library(dplyr)
library(ggplot2)
library(lubridate)
library(rpart)

#Adds item_catagory_id to sales_train in a new dataset.

sales_data = merge(sales_train, items[,c("item_id", "item_category_id")], by = "item_id", all.x =
T)


#makes the date an object date so it can be manipulated and adds month and day to the
sales_data colunm

sales_data$date = as.Date(sales_data$date, "%d.%m.%Y")

sales_data$month = month(sales_data$date)

sales_data$day = day(sales_data$date)
```

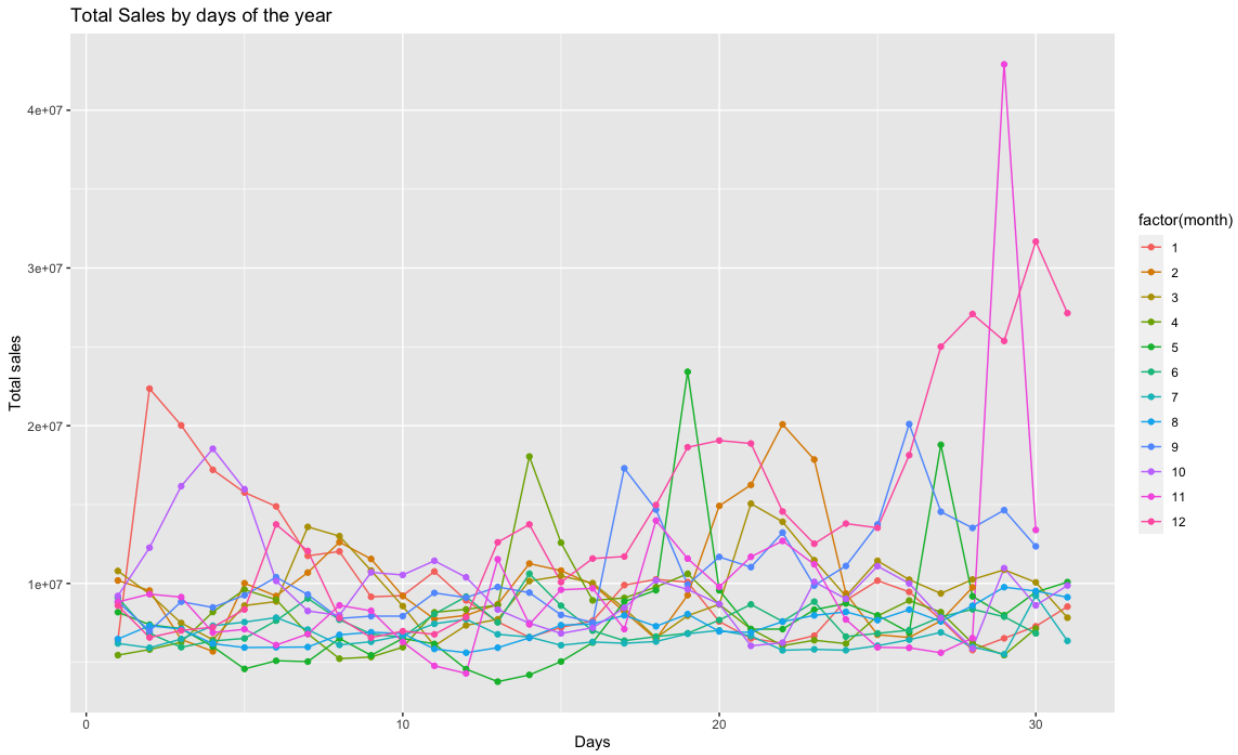
Data Exploration

```
sales_by_shop = sales_data %>%
          select(shop_id, item_cnt_day) %>%
          group_by(shop_id) %>%
          summarise(item_cnt_day =  sum(item_cnt_day))

ggplot(data =  sales_by_shop,
          mapping = aes(x = reorder(shop_id, item_cnt_day),
          y = item_cnt_day,
          fill = factor(shop_id))) +
          geom_histogram(stat = "identity", color = "black") +
          xlab("Shop ID") + ylab("Sales Count")+
          ggtitle(label = "Sales by Shop id") +
          theme(
                legend.position = "none"
          )
```

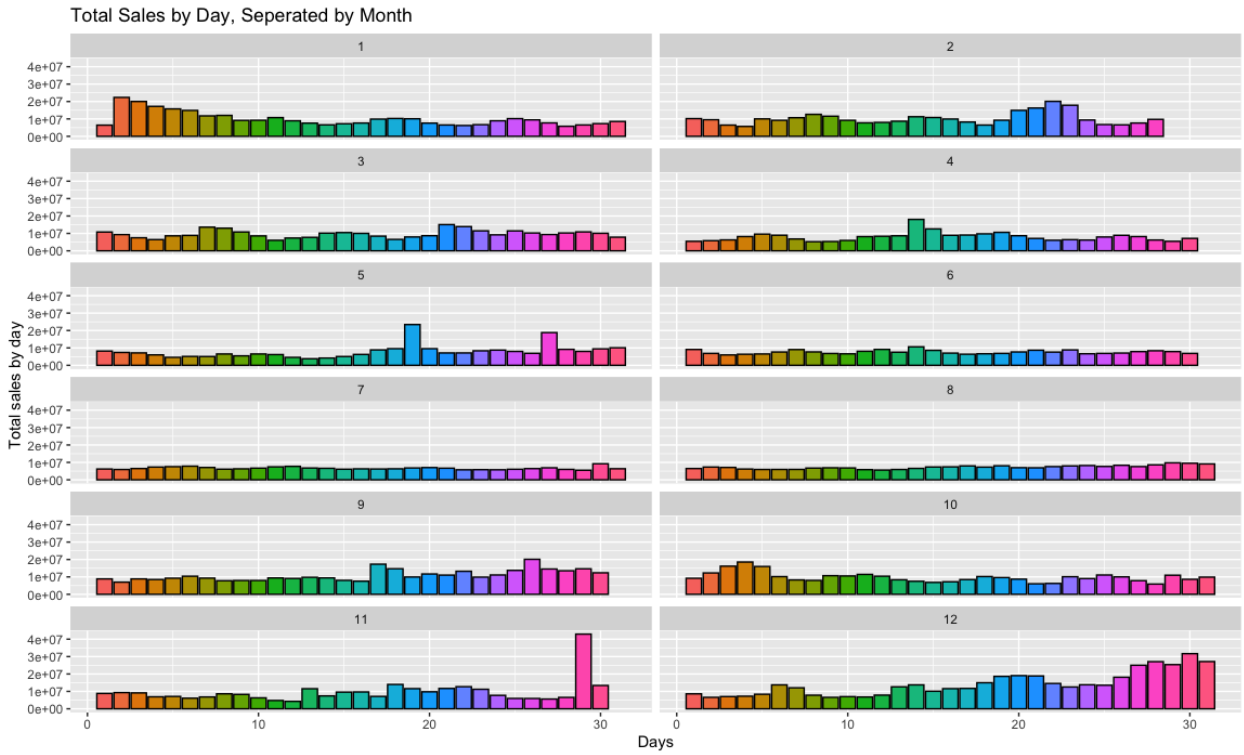



```
sales_by_daymonth =  sales_data %>%
        group_by(month, day) %>%
        summarise(total_sales =  sum(item_price * item_cnt_day))
```

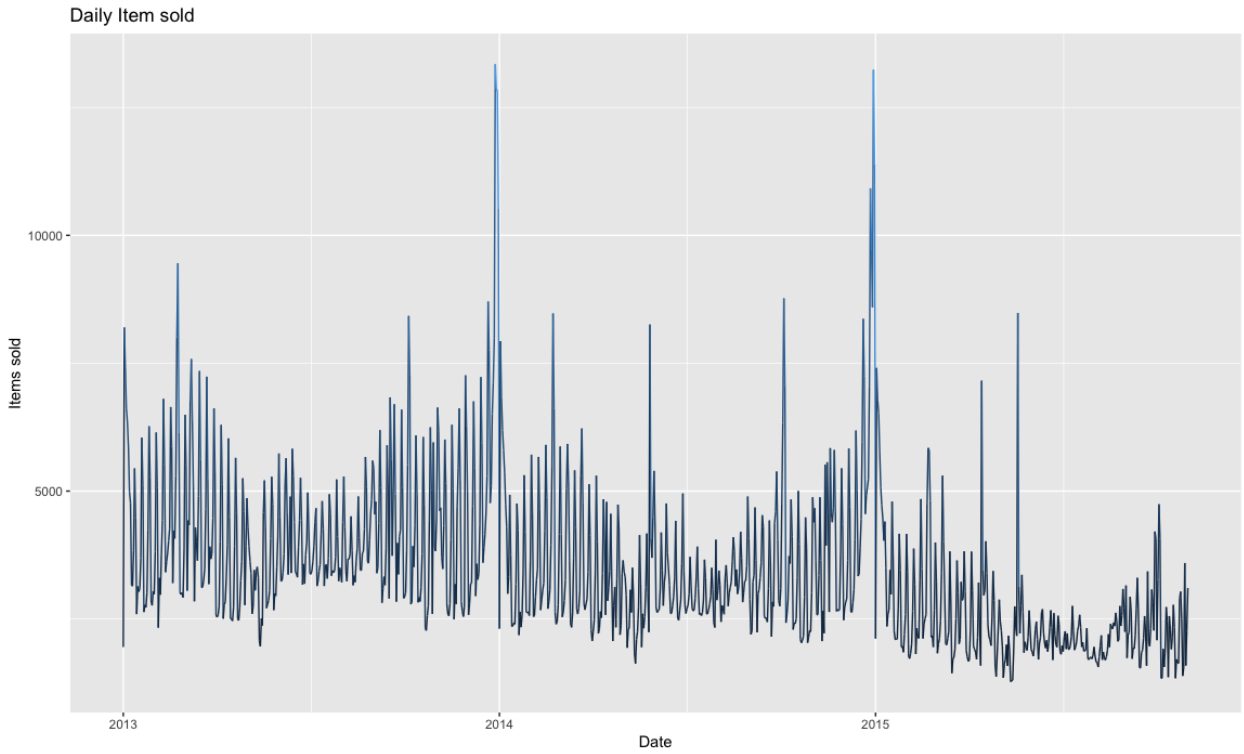
```
ggplot(sales_by_daymonth,
        aes(x = day,
        y = total_sales,
        group =  month,
        color =  factor(month))) +
        geom_line() +
        geom_point() +
        labs(title = "Total Sales by days of the year", x = "Days", y = "Total sales", fill = "Months")
```




```
ggplot(sales_by_daymonth,
        aes(x = day,
        y = total_sales,
        fill =  factor(day))) +
        geom_histogram(stat = "identity", color = "black") +
        labs(title = "Total Sales by Day, Seperated by Month", x = "Days", y = "Total sales by
                day", fill = "Days") +
        facet_wrap(~month, ncol = 2) +
        theme(
                legend.position = "none"
        )
```




```
day_sales = sales_data %>%
        group_by(date) %>%
        summarise(items_sold =  sum(item_cnt_day))

ggplot(day_sales, aes(x =  date, y = items_sold, color =  items_sold)) +
        geom_line() +
        labs(title = "Daily Item sold", x =  "Date", y = "Items sold")+
        theme(
                legend.position = "none"
        )
```

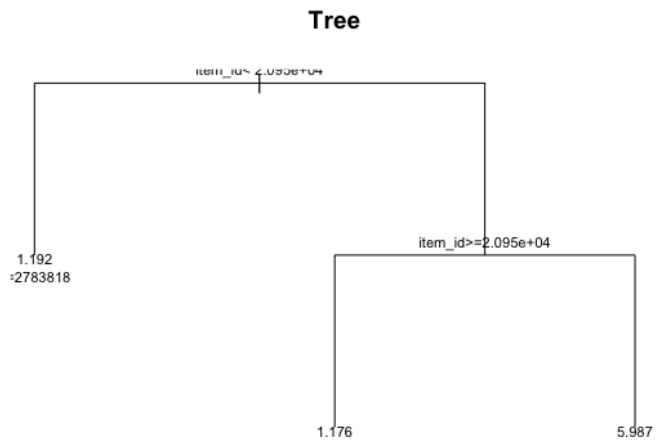


Modeling

```
linear_fit = lm(formula = item_cnt_day ~ shop_id + item_id + day + month,
                data = sales_data)

summary(linear_fit)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.104e+00  5.548e-03 199.027  < 2e-16 ***
shop_id     -9.844e-04  9.422e-05 -10.447  < 2e-16 ***
item_id      6.988e-06  2.417e-07  28.913  < 2e-16 ***
day          1.073e-03  1.719e-04   6.241 4.34e-10 ***
month        1.321e-02  4.340e-04  30.448  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.618 on 2935844 degrees of freedom
Multiple R-squared:  0.0006534,       Adjusted R-squared:  0.000652
F-statistic: 479.9 on 4 and 2935844 DF,  p-value: < 2.2e-16
```

```
linear_fit = lm(formula = item_cnt_day ~ shop_id + item_id,
                data = sales_data)

summary(linear_fit)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.202e+00  4.199e-03 286.306   <2e-16 ***
shop_id     -9.238e-04  9.422e-05  -9.805   <2e-16 ***
item_id      6.964e-06  2.417e-07  28.809   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.618 on 2935846 degrees of freedom
Multiple R-squared:  0.00031,           Adjusted R-squared:  0.0003093
F-statistic: 455.2 on 2 and 2935846 DF,  p-value: < 2.2e-16

pred = predict(linear_fit, test[,c("shop_id","item_id")])

final_result = data.frame(ID = test$ID, item_cnt_month =  pred)

write.csv(final_result, "/Users/bmoran32/Desktop/submission.csv", row.names = F)

##Decision Tree Model##

fit2 <- rpart(item_cnt_day ~ shop_id +
            item_id,
            method = "anova", data =
sales_data)

plot(fit2, uniform = TRUE,
    main = "Tree")
text(fit2, use.n = TRUE, cex = .7)

pred = predict(fit2,
test[,c("shop_id","item_id")])

final_result = data.frame(ID = test$ID,
item_cnt_month =  pred)

write.csv(final_result, "/Users/bmoran32/Desktop/submissionTree.csv", row.names = F)
```



Kaggle User: bmoran32

Kaggle Score: 1.55635 / 1.50649 on decision tree