Walmart Data Challenge

William Eerdmans

January 31, 2017

Read in the total data and subset the 1MM row data file

```
#read in sales_cust data
sales_cust_tot <- read.csv("sales_cust.csv")
sales_cust_tot <- tbl_df(sales_cust_tot)

#read in store
store_comp <- read.csv("store.csv")
store_comp <- tbl_df(store_comp)
```

Check for missing values, inconsistencies, and column data types

It can be seen in sales_cust that there are 59 NA values in Open and 103 NA values in SchoolHoliday

When looking at Open, NAs are only during 7/5/15 & 7/6/15. However, some of the stores have no sales, whereas some have sales

What can also be seen is that store 384-398 repeat for these NA values

When observing other Day = 7 dates, it was found that they too were all 0. Thus, those NAs can be determined to be Open = 0, whereas, the NAs that actually have sales will be assumed to be Open = 1.

When looking at the School Holiday NA values and observing other values between StateHoliday and SchoolHoliday, I didn't see any repeatable pattern between them, beyond that when there are certain StateHolidays, SchoolHolidays may occur. However, I believe the best course of action would be to delete these rows due to 103 values out of ~1MM is negligible. I created a new dataframe in order to seperate the prior values that were not NA in sc_open_NAs.

What can be observed with the Store_competition data is that 1) There are stores with no competitors 2) There are stores with no promotions/promotion intervals.

From the summary, it shows that there are 354 instances where stores do not have competitors (or close enough to provide any effect) and 544 instances of no Promotions.

Beyond these two sets of NAs being consistent between related columns there are 3 NAs in Competition Distance. Looking at the results, beyond the StoreType and Assortment, there is little information. Thus, I would suggest to delete the 3 rows from the analysis due to 3 being negligible than 1115 entries. Now using store_comp2

```
#Summarize the data for initial look
head(sales_cust_tot)
```

```
## # A tibble: 6 × 9
##   Store DayOfWeek    Date Sales Customers  Open Promo StateHoliday
##   <int>     <int>  <fctr> <int>     <int> <int> <int>       <fctr>
## 1     1         5 7/31/15  5263       555     1     1            0
## 2     2         5 7/31/15  6064       625     1     1            0
## 3     3         5 7/31/15  8314       821     1     1            0
## 4     4         5 7/31/15 13995      1498     1     1            0
## 5     5         5 7/31/15  4822       559     1     1            0
## 6     6         5 7/31/15  5651       589     1     1            0
## # ... with 1 more variables: SchoolHoliday <int>
```

```
summary(sales_cust_tot)
```

```
##      Store          DayOfWeek          Date             Sales
##  Min.   :   1.0   Min.   :1.000   1/1/14 :   1115   Min.   :    0
##  1st Qu.: 280.0   1st Qu.:2.000   1/1/15 :   1115   1st Qu.: 3727
##  Median : 558.0   Median :4.000   1/10/13:   1115   Median : 5744
##  Mean   : 558.4   Mean   :3.998   1/10/14:   1115   Mean   : 5774
##  3rd Qu.: 838.0   3rd Qu.:6.000   1/10/15:   1115   3rd Qu.: 7856
##  Max.   :1115.0   Max.   :7.000   1/11/13:   1115   Max.   :41551
##                                   (Other):1010519
##    Customers           Open            Promo        StateHoliday
##  Min.   :   0.0   Min.   :0.0000   Min.   :0.0000   0:986159
##  1st Qu.: 405.0   1st Qu.:1.0000   1st Qu.:0.0000   a: 20260
##  Median : 609.0   Median :1.0000   Median :0.0000   b:  6690
##  Mean   : 633.1   Mean   :0.8301   Mean   :0.3815   c:  4100
##  3rd Qu.: 837.0   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :7388.0   Max.   :1.0000   Max.   :1.0000
##                   NA's   :59
##  SchoolHoliday
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1786
##  3rd Qu.:0.0000
##  Max.   :1.0000
##  NA's   :103
```

```
#visually observe the data where there are NAs
#Start with Open
sc_open_NAs <- sales_cust_tot %>% filter(is.na(Open))

#Take a look at the values
print(sc_open_NAs, n=59)
```

```
## # A tibble: 59 × 9
##    Store DayOfWeek   Date Sales Customers  Open Promo StateHoliday
##    <int>     <int> <fctr> <int>     <int> <int> <int>       <fctr>
## 1    384         1 7/6/15 10337      1214    NA     0            0
## 2    385         1 7/6/15  6951       620    NA     0            0
## 3    386         1 7/6/15  6479       545    NA     0            0
## 4    387         1 7/6/15  8817      1172    NA     0            0
```

```
## 5    388         1 7/6/15  9596      1032    NA     0            0
## 6    389         1 7/6/15 11928      1407    NA     0            0
## 7    390         1 7/6/15 11380      1083    NA     0            0
## 8    391         1 7/6/15  5293       668    NA     0            0
## 9    392         1 7/6/15  7580       721    NA     0            0
## 10   393         1 7/6/15  5780       568    NA     0            0
## 11   394         1 7/6/15  7824       673    NA     0            0
## 12   395         1 7/6/15  4153       505    NA     0            0
## 13   396         1 7/6/15  9726      1038    NA     0            0
## 14   397         1 7/6/15  5947       842    NA     0            0
## 15   398         1 7/6/15  4426       577    NA     0            0
## 16   384         7 7/6/14     0         0    NA     0            0
## 17   386         7 7/6/14     0         0    NA     0            0
## 18   387         7 7/6/14     0         0    NA     0            0
## 19   388         7 7/6/14     0         0    NA     0            0
## 20   389         7 7/6/14     0         0    NA     0            0
## 21   390         7 7/6/14     0         0    NA     0            0
## 22   391         7 7/6/14     0         0    NA     0            0
## 23   392         7 7/6/14     0         0    NA     0            0
## 24   393         7 7/6/14     0         0    NA     0            0
## 25   394         7 7/6/14     0         0    NA     0            0
## 26   395         7 7/6/14     0         0    NA     0            0
## 27   396         7 7/6/14     0         0    NA     0            0
## 28   397         7 7/6/14     0         0    NA     0            0
## 29   398         7 7/6/14     0         0    NA     0            0
## 30   384         6 7/6/13  4369       495    NA     0            0
## 31   385         6 7/6/13  6634       625    NA     0            0
## 32   386         6 7/6/13  6860       619    NA     0            0
## 33   387         6 7/6/13  5880       938    NA     0            0
## 34   388         6 7/6/13  7409       866    NA     0            0
## 35   389         6 7/6/13  6537       861    NA     0            0
## 36   390         6 7/6/13  9231       952    NA     0            0
## 37   391         6 7/6/13  2677       389    NA     0            0
## 38   392         6 7/6/13  4035       431    NA     0            0
## 39   393         6 7/6/13  5879       533    NA     0            0
## 40   394         6 7/6/13  7993       728    NA     0            0
## 41   395         6 7/6/13  2536       365    NA     0            0
## 42   396         6 7/6/13  4889       554    NA     0            0
## 43   397         6 7/6/13  3993       588    NA     0            0
## 44   398         6 7/6/13  4484       492    NA     0            0
## 45   384         5 7/5/13  7874       886    NA     1            0
## 46   385         5 7/5/13  8277       740    NA     1            0
## 47   386         5 7/5/13  9652       798    NA     1            0
## 48   387         5 7/5/13  9207      1234    NA     1            0
## 49   388         5 7/5/13 11064      1194    NA     1            0
## 50   389         5 7/5/13  9751      1239    NA     1            0
## 51   390         5 7/5/13 11856      1102    NA     1            0
## 52   391         5 7/5/13  6099       831    NA     1            0
## 53   392         5 7/5/13  7515       755    NA     1            0
## 54   393         5 7/5/13  6146       620    NA     1            0
## 55   394         5 7/5/13 10338       786    NA     1            0
## 56   395         5 7/5/13  5274       635    NA     1            0
## 57   396         5 7/5/13 10749      1092    NA     1            0
## 58   397         5 7/5/13  6447       851    NA     1            0
```

```
## 59   398         5 7/5/13  6271       668    NA     1            0
## # ... with 1 more variables: SchoolHoliday <int>
```

```
#For the Open NAs, impute 1's for the Open column and for those where the Day of the week is 7,
#impute Open = 0
sales_cust_tot[which(is.na(sales_cust_tot$Open) & sales_cust_tot$DayOfWeek == 7),]$Open <- 0
sales_cust_tot[which(is.na(sales_cust_tot$Open) & sales_cust_tot$Sales > 0),]$Open <- 1

#Now turn attention to the 103 NAs in SchoolHoliday
sc_holiday_NAs <- sales_cust_tot %>% filter(is.na(SchoolHoliday))

#Take a look at the values
print(sc_holiday_NAs, n=59)
```

```
## # A tibble: 103 × 9
##    Store DayOfWeek     Date Sales Customers  Open Promo StateHoliday
##    <int>     <int>   <fctr> <int>     <int> <dbl> <int>       <fctr>
## 1    398         6  7/25/15  5044       538     1     0            0
## 2    384         5  7/24/15  7459       869     1     0            0
## 3    385         5  7/24/15  5328       560     1     0            0
## 4    386         5  7/24/15  5582       489     1     0            0
## 5    387         5  7/24/15  7513       963     1     0            0
## 6    388         5  7/24/15  7672       913     1     0            0
## 7    389         5  7/24/15  9265      1124     1     0            0
## 8    390         5  7/24/15  9514       899     1     0            0
## 9    391         5  7/24/15  4536       606     1     0            0
## 10   392         5  7/24/15  5718       581     1     0            0
## 11   393         5  7/24/15  5081       515     1     0            0
## 12   394         5  7/24/15  7615       669     1     0            0
## 13   395         5  7/24/15  3177       446     1     0            0
## 14   396         5  7/24/15  7972       862     1     0            0
## 15   397         5  7/24/15  4512       668     1     0            0
## 16   398         5  7/24/15  4427       579     1     0            0
## 17   384         4  7/23/15  8856      1050     1     0            0
## 18   385         4  7/23/15  5549       519     1     0            0
## 19   386         4  7/23/15  4766       450     1     0            0
## 20   387         4  7/23/15  7631       979     1     0            0
## 21   388         4  7/23/15  8329       968     1     0            0
## 22   389         4  7/23/15  9681      1216     1     0            0
## 23   390         4  7/23/15  9138       859     1     0            0
## 24   391         4  7/23/15  4733       590     1     0            0
## 25   392         4  7/23/15  5870       609     1     0            0
## 26   393         4  7/23/15  4877       504     1     0            0
## 27   394         4  7/23/15  7948       648     1     0            0
## 28   395         4  7/23/15  3261       392     1     0            0
## 29   396         4  7/23/15  9983      1081     1     0            0
## 30   397         4  7/23/15  4550       662     1     0            0
## 31   398         4  7/23/15  4585       517     1     0            0
## 32   384         3  7/22/15  6738       804     1     0            0
## 33   385         3  7/22/15  5280       490     1     0            0
## 34   386         3  7/22/15  5305       475     1     0            0
## 35   387         3  7/22/15  7560       997     1     0            0
## 36   388         3  7/22/15  7198       837     1     0            0
## 37   389         3  7/22/15  8989      1118     1     0            0
## 38   390         3  7/22/15  8938       852     1     0            0
```

```
## 39   391         3  7/22/15  3766       523     1     0            0
## 40   392         3  7/22/15  5354       576     1     0            0
## 41   393         3  7/22/15  4364       438     1     0            0
## 42   394         3  7/22/15  6649       554     1     0            0
## 43   395         3  7/22/15  2545       391     1     0            0
## 44   396         3  7/22/15  8392       848     1     0            0
## 45   397         3  7/22/15  4649       677     1     0            0
## 46   398         3  7/22/15  3999       456     1     0            0
## 47   384         5 10/10/14  7986       896     1     1            0
## 48   386         5 10/10/14  7254       595     1     1            0
## 49   387         5 10/10/14 14584      1862     1     1            0
## 50   388         5 10/10/14  9744      1101     1     1            0
## 51   389         5 10/10/14 12469      1454     1     1            0
## 52   390         5 10/10/14 11495      1105     1     1            0
## 53   391         5 10/10/14  6965       865     1     1            0
## 54   392         5 10/10/14  7175       731     1     1            0
## 55   393         5 10/10/14  6197       624     1     1            0
## 56   394         5 10/10/14 10234       792     1     1            0
## 57   395         5 10/10/14  3862       559     1     1            0
## 58   396         5 10/10/14  9502      1029     1     1            0
## 59   397         5 10/10/14  5893       788     1     1            0
## # ... with 44 more rows, and 1 more variables: SchoolHoliday <int>
```

```
#delete the rows in School Holiday where NA
#make a new dataframe without the rows with NA
sales_cust <- sales_cust_tot[which(!is.na(sales_cust_tot$SchoolHoliday)),]
sales_cust$Store <- as.factor(sales_cust$Store)
sales_cust$DayOfWeek <- as.factor(sales_cust$DayOfWeek)
sales_cust$Date <- as.Date(sales_cust$Date, "%m/%d/%y")

#Look at the store_comp data now
head(store_comp)
```

```
## # A tibble: 6 × 10
##   Store StoreType Assortment CompetitionDistance CompetitionOpenSinceMonth
##   <int>    <fctr>     <fctr>               <int>                     <int>
## 1     1         c          a                1270                         9
## 2     2         a          a                 570                        11
## 3     3         a          a               14130                        12
## 4     4         c          c                 620                         9
## 5     5         a          a               29910                         4
## 6     6         a          a                 310                        12
## # ... with 5 more variables: CompetitionOpenSinceYear <int>, Promo2 <int>,
## #   Promo2SinceWeek <int>, Promo2SinceYear <int>, PromoInterval <fctr>
```

```
summary(store_comp)
```

```
##      Store        StoreType Assortment CompetitionDistance
##  Min.   :   1.0   a:602     a:593      Min.   :   20.0
##  1st Qu.: 279.5   b: 17     b:  9      1st Qu.:  717.5
##  Median : 558.0   c:148     c:513      Median : 2325.0
##  Mean   : 558.0   d:348                Mean   : 5404.9
##  3rd Qu.: 836.5                        3rd Qu.: 6882.5
##  Max.   :1115.0                        Max.   :75860.0
##                                        NA's   :3
```

```
##  CompetitionOpenSinceMonth CompetitionOpenSinceYear     Promo2
##  Min.   : 1.000            Min.   :1900              Min.   :0.0000
##  1st Qu.: 4.000            1st Qu.:2006              1st Qu.:0.0000
##  Median : 8.000            Median :2010              Median :1.0000
##  Mean   : 7.225            Mean   :2009              Mean   :0.5121
##  3rd Qu.:10.000            3rd Qu.:2013              3rd Qu.:1.0000
##  Max.   :12.000            Max.   :2015              Max.   :1.0000
##  NA's   :354               NA's   :354
##  Promo2SinceWeek Promo2SinceYear          PromoInterval
##  Min.   : 1.0    Min.   :2009                    :544
##  1st Qu.:13.0    1st Qu.:2011    Feb,May,Aug,Nov :130
##  Median :22.0    Median :2012    Jan,Apr,Jul,Oct :335
##  Mean   :23.6    Mean   :2012    Mar,Jun,Sept,Dec:106
##  3rd Qu.:37.0    3rd Qu.:2013
##  Max.   :50.0    Max.   :2015
##  NA's   :544     NA's   :544
```

```
#Observe the 3 rows where Competition Distance is NA
store_comp %>% filter(is.na(CompetitionDistance))
```

```
## # A tibble: 3 × 10
##   Store StoreType Assortment CompetitionDistance CompetitionOpenSinceMonth
##   <int>    <fctr>     <fctr>               <int>                     <int>
## 1   291         d          a                  NA                        NA
## 2   622         a          c                  NA                        NA
## 3   879         d          a                  NA                        NA
## # ... with 5 more variables: CompetitionOpenSinceYear <int>, Promo2 <int>,
## #   Promo2SinceWeek <int>, Promo2SinceYear <int>, PromoInterval <fctr>
```

```
#Delete the 3 rows from the data
store_comp2 <- store_comp %>% filter(!is.na(CompetitionDistance))
```

Next step is to determine initial exploratory analysis

This is very quick and is only used to gauge what is occuring in the data and to also determine a direction/business problem to explore.

Total sales comes out to the weekly sales over time.

The next graph wanted to see if any of the low points were State Holidays. Only one appears to be right when 2013 began.

The next graph illustrates whether the low points were School Holidays. In this case, all of the lower points were found to be School Holidays beyond the 1 points which was a State Holiday.

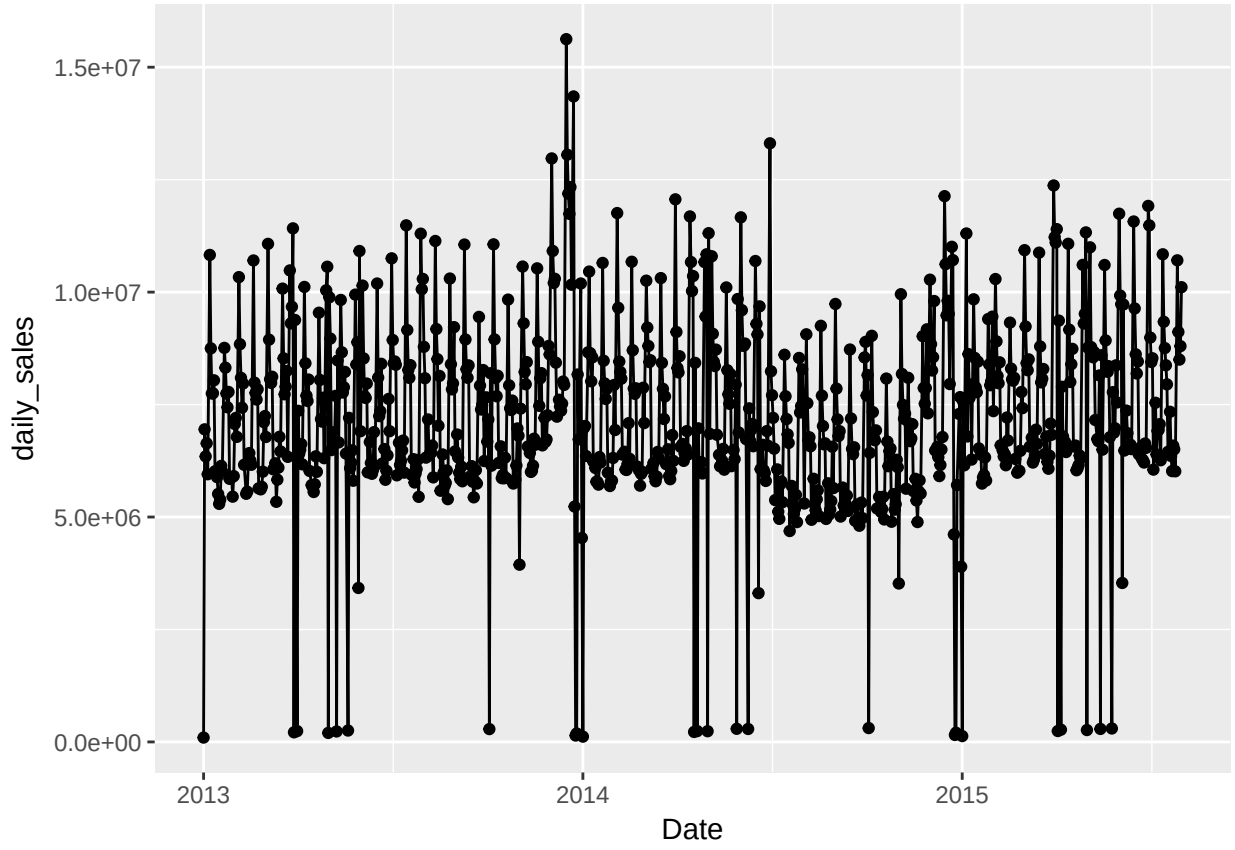
The next plot only shows the School Holiday sales from the previous graph. Similar behavior occurs in 2013 and 2014 in the data.

The next few graphs looked at the average and sum of each Week and Month over time for different years. This is to illustrate the seasonality during the week or year.

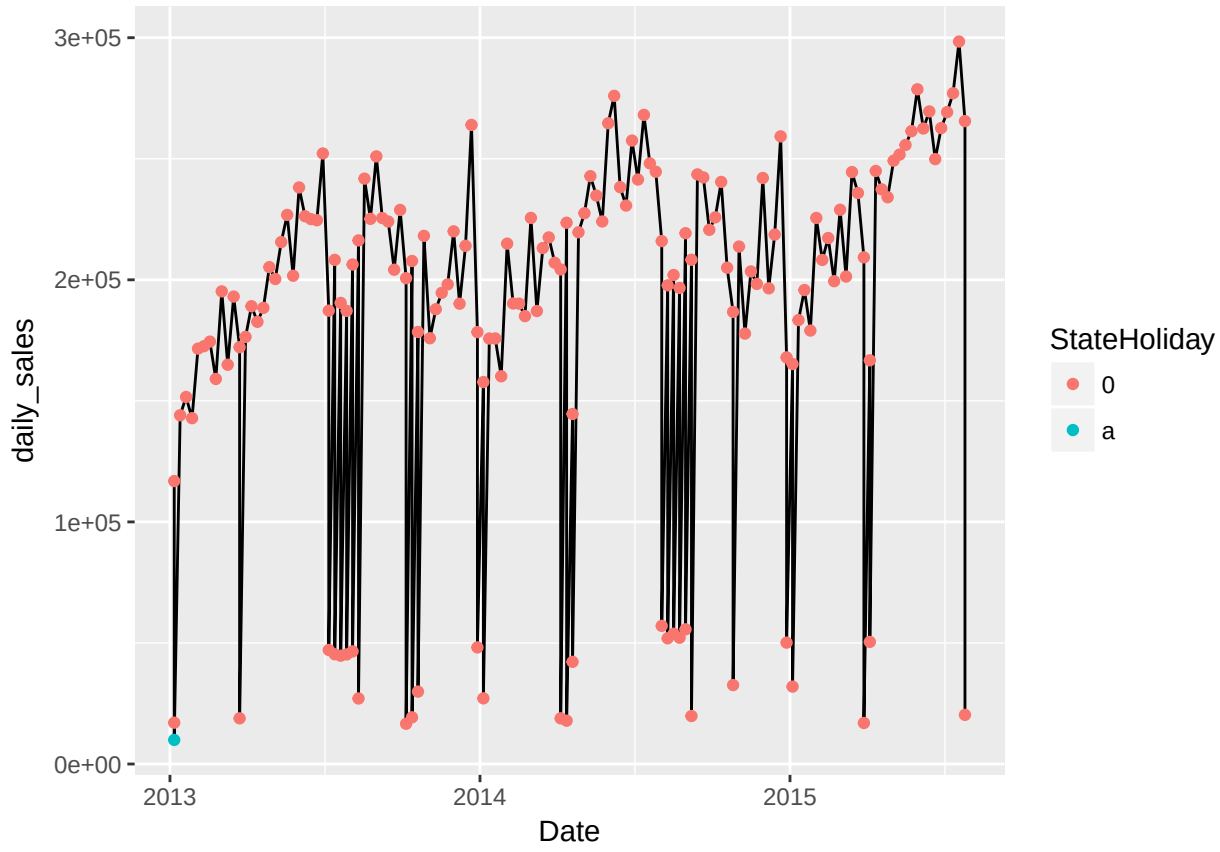
The sales during the days of the week in both the average and sum graphs show similar tendencies for the years 2013, 2014, and 2015.

Surprisingly, 2015 seems to be outperforming the prior two years in the monthly seasonal graphs in both average and sum.

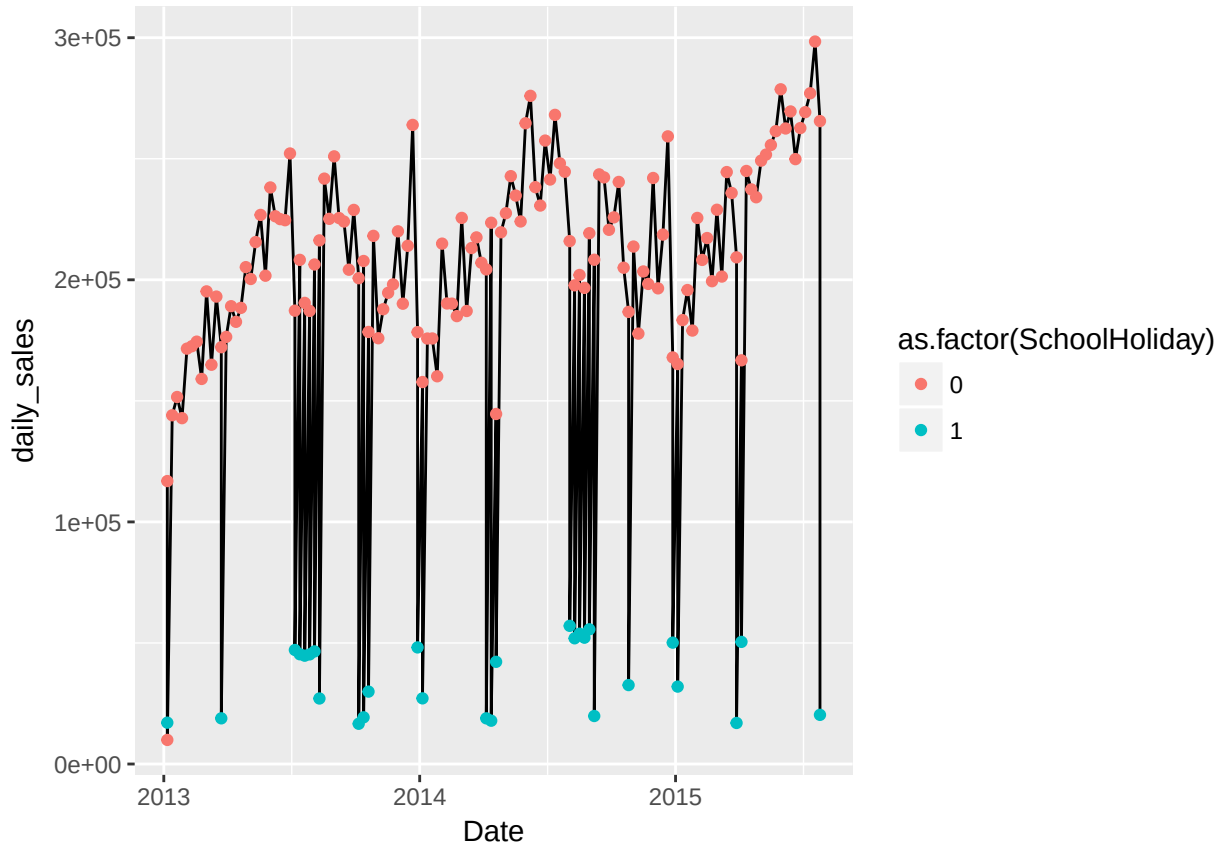
```
#calculate total sales per day for the stores provided
tot_sales <- sales_cust %>%
  filter(Open == 1 & DayOfWeek != 7) %>%
  group_by(Date) %>%
  summarize(daily_sales =sum(Sales))

#plot total sales per day
ggplot(tot_sales) + geom_line(aes(x=Date, y=daily_sales))+geom_point(aes(x=Date, y=daily_sales))
```



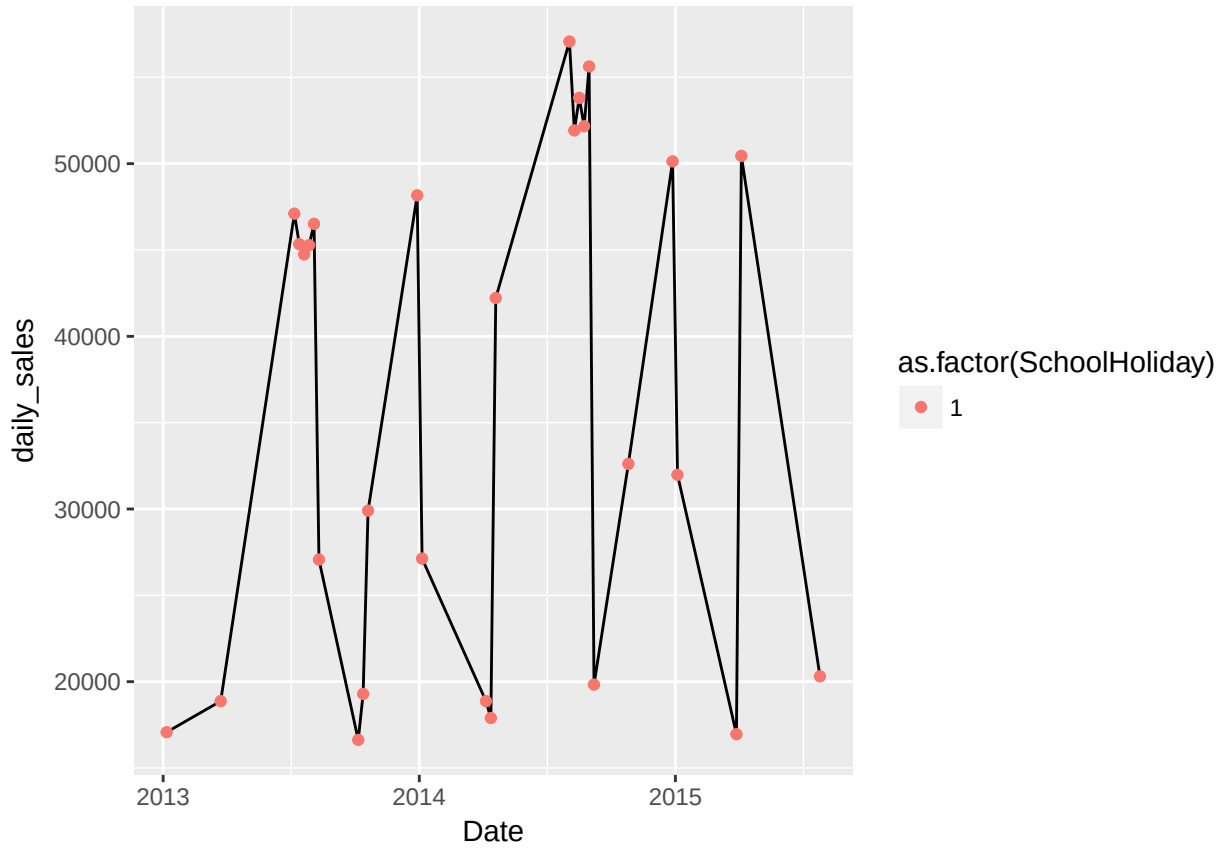

```
#Consider Holidays in the picture
tot_sales_hol <- sales_cust %>%
  filter(Open == 1 & DayOfWeek == 7) %>%
  group_by(Date, StateHoliday, SchoolHoliday) %>%
  summarize(daily_sales =sum(Sales))

#color the holiday points
ggplot(tot_sales_hol) + geom_line(aes(x=Date, y=daily_sales))+geom_point(aes(x=Date, y=daily_sales, col
```



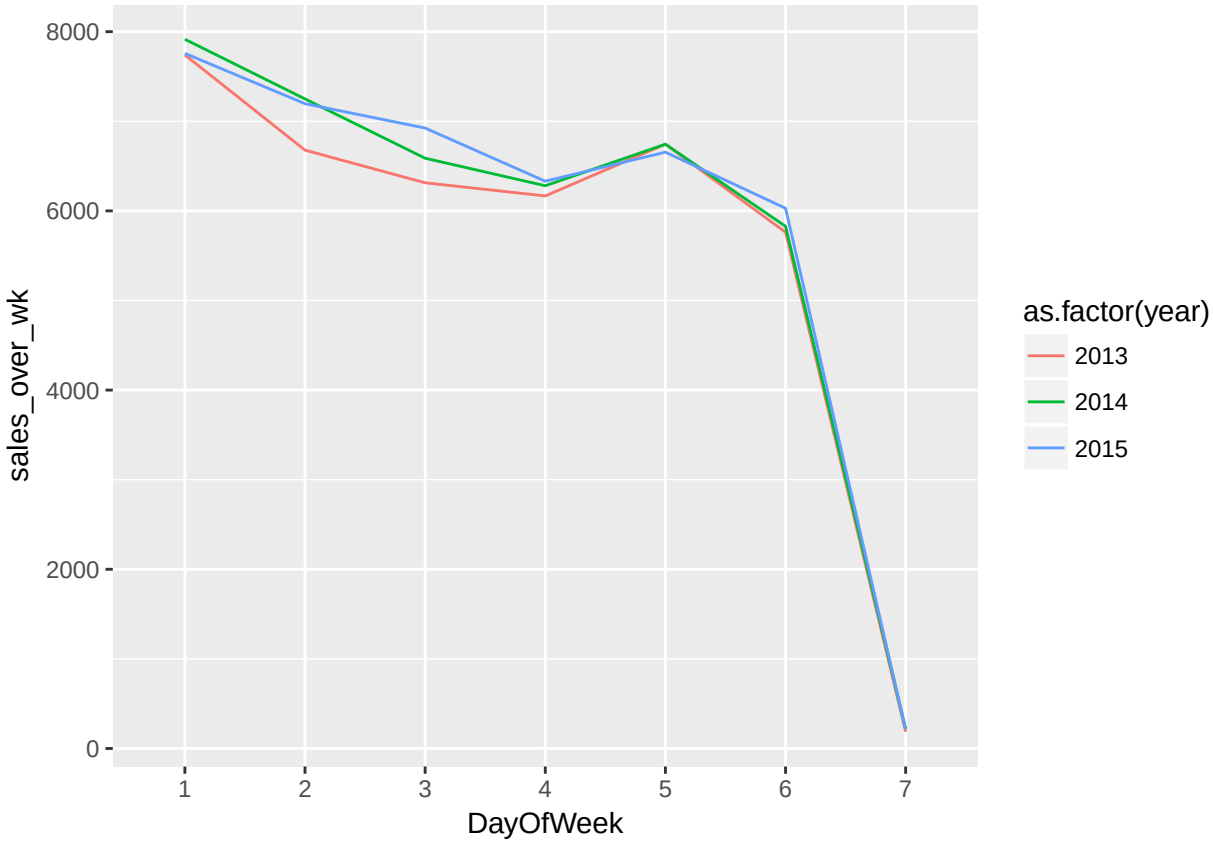

```
#color the school holiday points
ggplot(tot_sales_hol) + geom_line(aes(x=Date, y=daily_sales))+geom_point(aes(x=Date, y=daily_sales, col
```



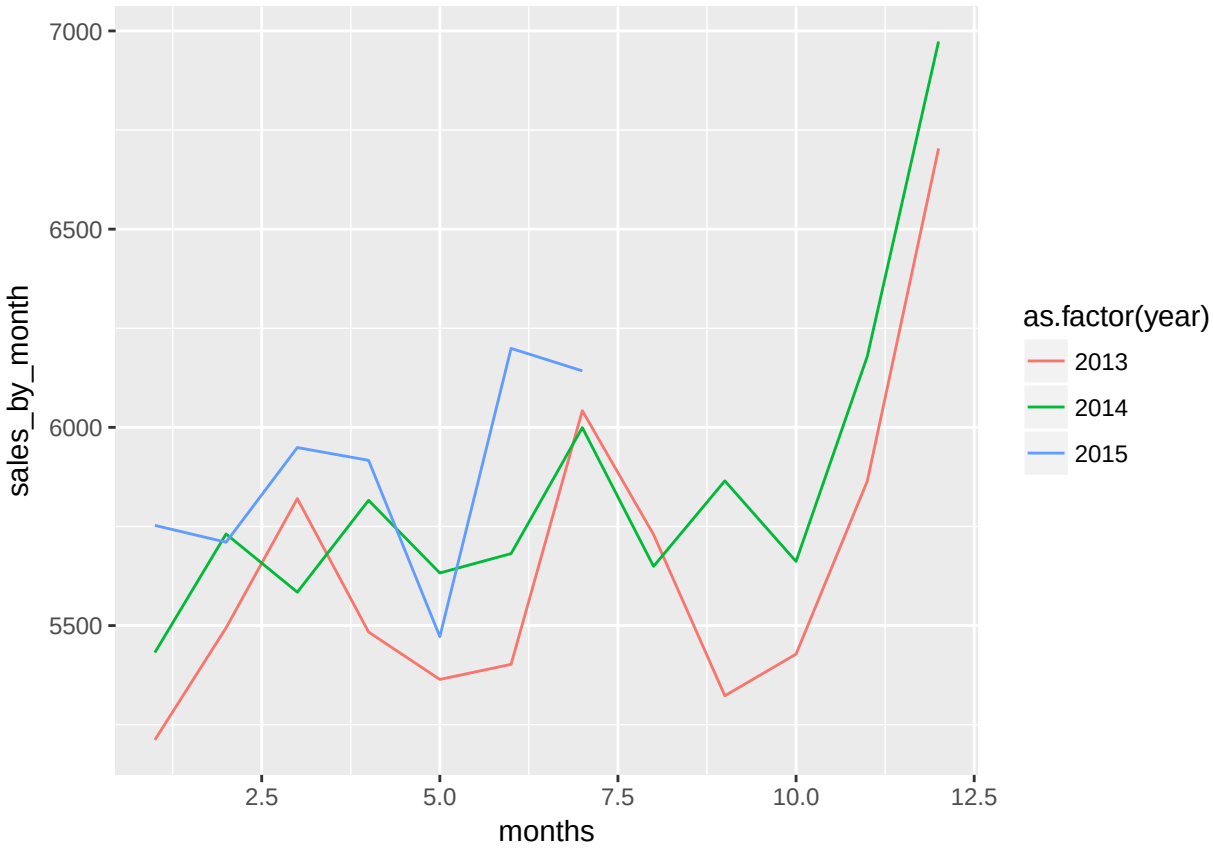

```
#Look at school holidays
ggplot(tot_sales_hol[which(tot_sales_hol$SchoolHoliday == 1),]) + geom_line(aes(x=Date, y=daily_sales))+
```



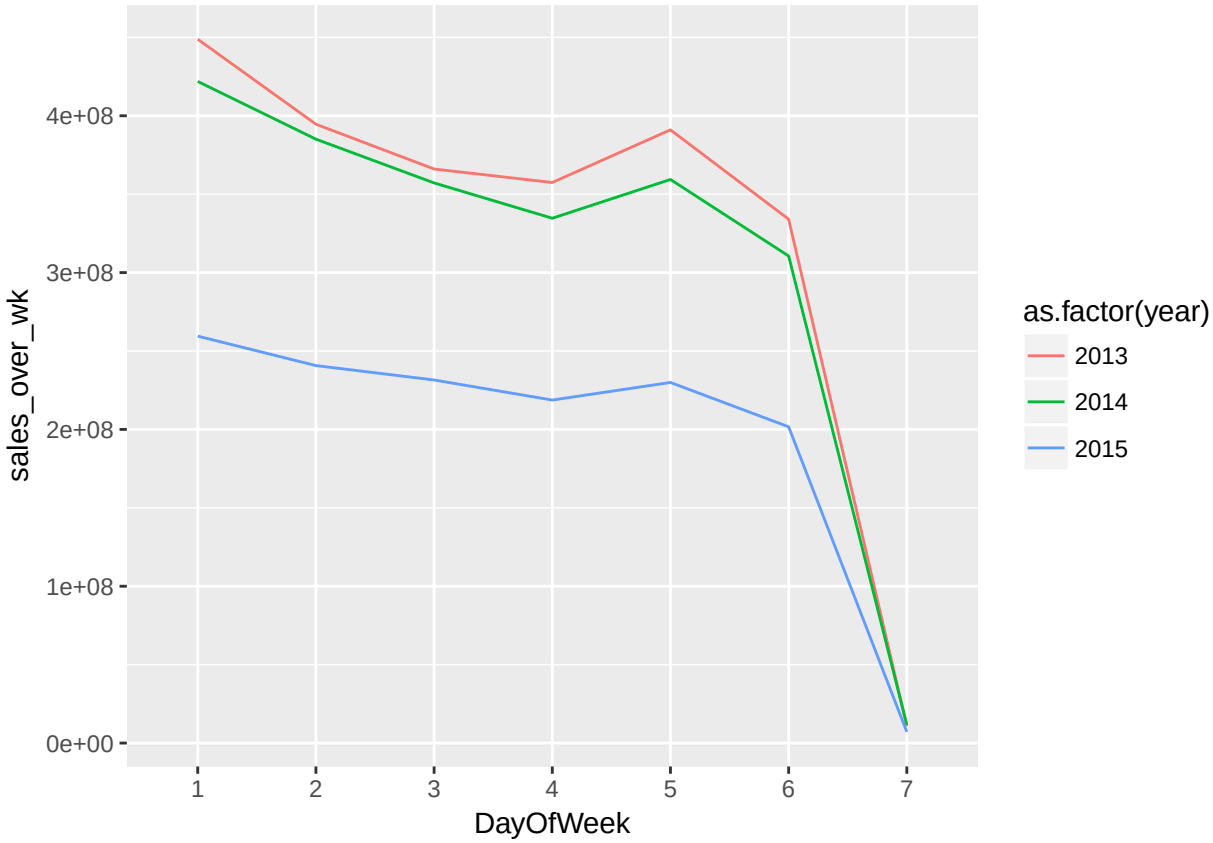

```
#create weekly seasonal plot and monthly seasonplot
#Remeber these are the averages, may want to look into sums
seasonal_weeks <- sales_cust %>%
  group_by(year = year(Date),DayOfWeek) %>%
  summarize(sales_over_wk = mean(Sales))

seasonal_months <- sales_cust %>%
  group_by(year = year(Date), months = month(Date)) %>%
  summarize(sales_by_month = mean(Sales))

#plot seasonal weeks and months
ggplot(seasonal_weeks) + geom_line(aes(x=DayOfWeek, y= sales_over_wk, group=year, color=as.factor(year)
```



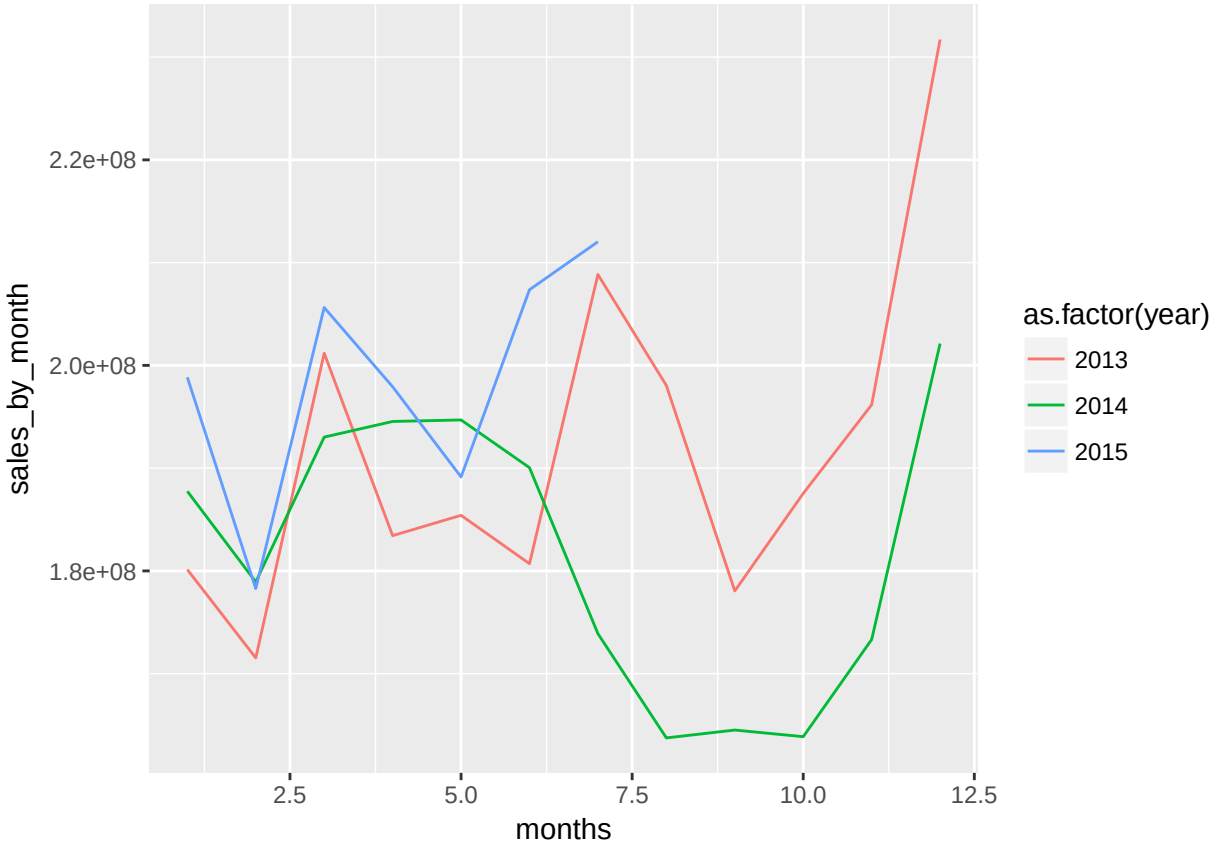

```
ggplot(seasonal_months) + geom_line(aes(x=months, y= sales_by_month, group = year, color=as.factor(year
```




```
#create weekly seasonal plot and monthly seasonplot
#THESE ARE SUMS
seasonal_weeks <- sales_cust %>%
  group_by(year = year(Date),DayOfWeek) %>%
  summarize(sales_over_wk = sum(Sales))

seasonal_months <- sales_cust %>%
  group_by(year = year(Date), months = month(Date)) %>%
  summarize(sales_by_month = sum(Sales))

#plot seasonal weeks and months
ggplot(seasonal_weeks) + geom_line(aes(x=DayOfWeek, y= sales_over_wk, group=year, color=as.factor(year)
```




```
ggplot(seasonal_months) + geom_line(aes(x=months, y= sales_by_month, group = year, color=as.factor(year
```



After looking at the visualizations above, it became clear that the data contained outliers overall and also the the state holiday data should possible be removed. However, in order to further investigate this idea, I decided to use a time series anomaly detection algorithm created by Twitter. Moving forward, I would like to determine why certain events over performed and underperformed during the year and then try to determine if this was due to promotions or the competitors.

Find anomalies in the time series and investigate whether they are occuring due to promotions or other factors

```
library(AnomalyDetection)
library(dygraphs)
library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

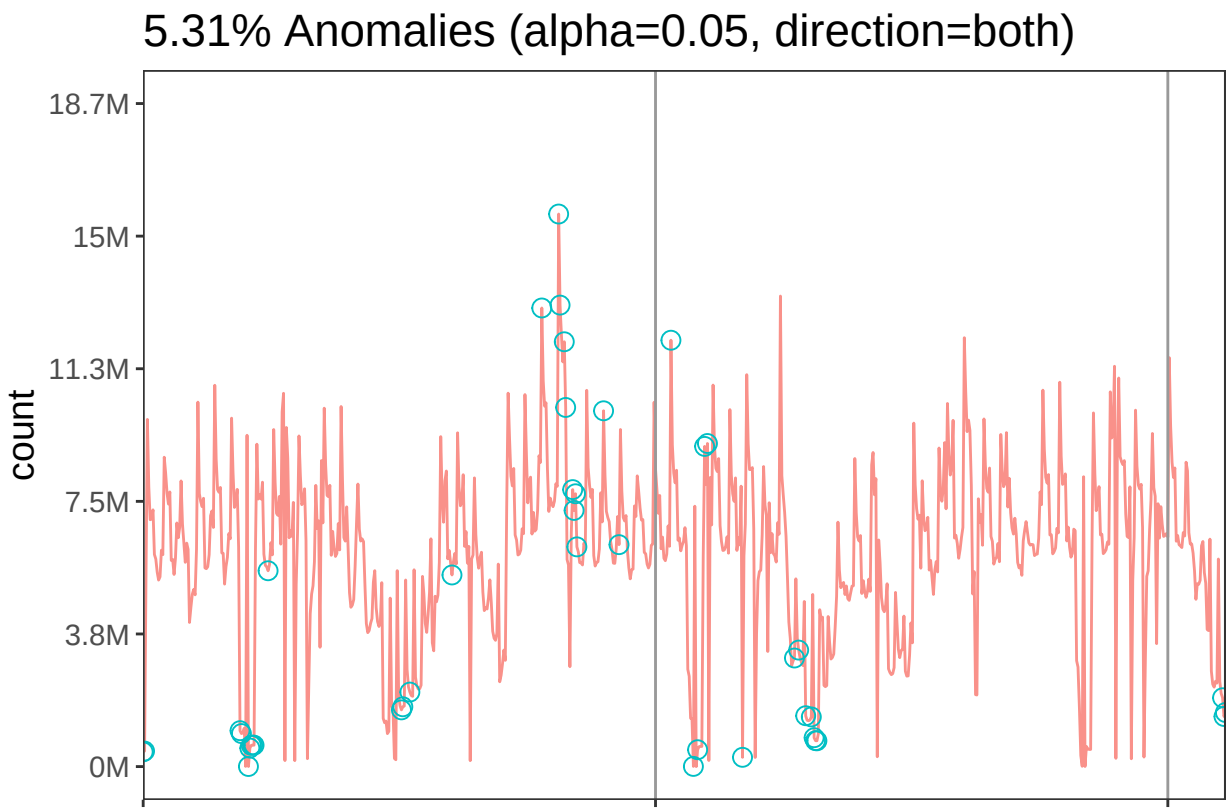
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
```

```
##     first, last
```

```
#Add day, month, and year columns to sales_cust
#Enables for grouping
sales_cust$day <- as.numeric(format(as.Date(sales_cust$Date,format="%Y-%m-%d"), "%d"))
sales_cust$month <- as.numeric(format(as.Date(sales_cust$Date,format="%Y-%m-%d"), "%m"))
sales_cust$year <- as.numeric(format(as.Date(sales_cust$Date,format="%Y-%m-%d"), "%Y"))

#Only have data where the Store is open, where it is not Sunday, and when there is no School Holidays.
daily_sales <- sales_cust %>%
  filter(Open == 1 & DayOfWeek != 7 & SchoolHoliday == 0) %>%
  group_by(year, month, day) %>%
  summarize(daily_sales =sum(Sales)) %>%
  arrange(year, month, day)


res <- AnomalyDetectionVec(daily_sales[,4], alpha=0.05, period=365, direction='both', only_last=FALSE, p
res$plot
```

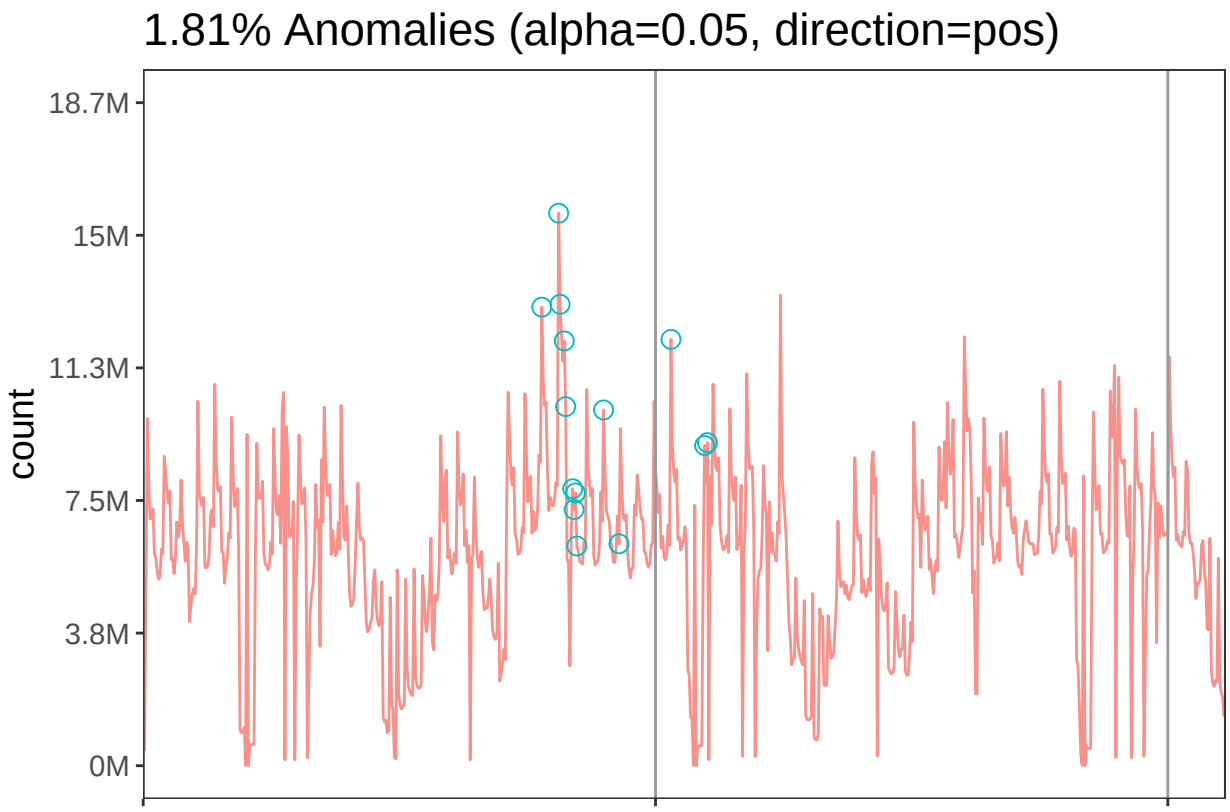



```
daily_sales[res$anoms$index,]
```

```
## Source: local data frame [41 x 4]
## Groups: year, month [13]
##
##     year month   day daily_sales
##    <dbl> <dbl> <dbl>       <int>
## 1   2013     1     3      445340
```

```
## 2   2013     1     4      468261
## 3   2013     3    25     1043558
## 4   2013     3    26      962435
## 5   2013     4     1       33326
## 6   2013     4     2      552637
## 7   2013     4     3      631299
## 8   2013     4     4      636404
## 9   2013     4     5      635104
## 10  2013     4    17     5556697
## # ... with 31 more rows
```

```
positive_outlier <- AnomalyDetectionVec(daily_sales[,4], alpha=0.05, period=365, direction='pos', only_l
positive_outlier$plot
```

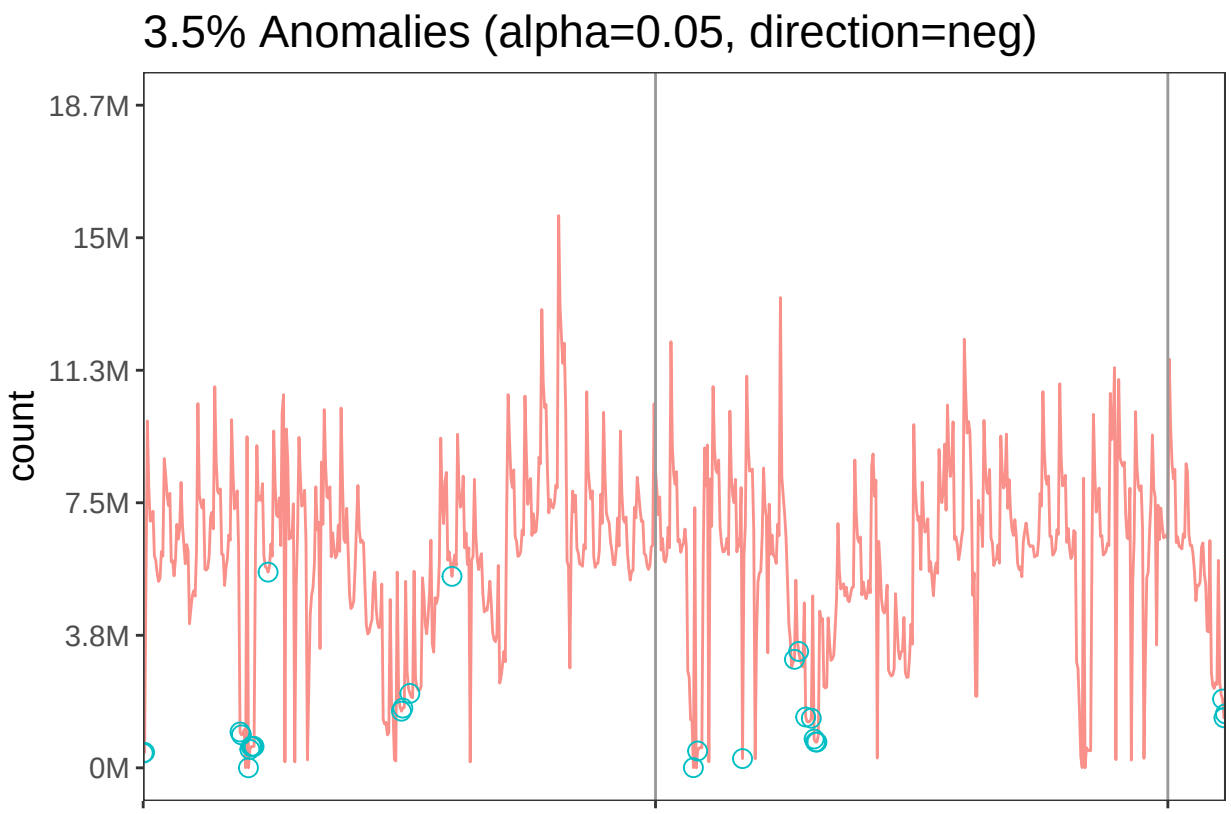



```
pos_out <- positive_outlier$anoms$index
daily_sales[positive_outlier$anoms$index,]
```

```
## Source: local data frame [14 x 4]
## Groups: year, month [5]
##
##     year month   day daily_sales
##    <dbl> <dbl> <dbl>       <int>
## 1   2013    12     2    12972440
## 2   2013    12    16    15623548
## 3   2013    12    17    13053993
## 4   2013    12    20    12019100
## 5   2013    12    21    10165366
```

```
## 6   2014     1      8     7846977
## 7   2014     1      9     7260282
## 8   2014     1     10     7729836
## 9   2014     1     11     6231179
## 10  2014     2      3    10071075
## 11  2014     2     15     6293698
## 12  2014     3     31    12062933
## 13  2014     4     28     9066052
## 14  2014     4     30     9148283
```

```
negative_outlier <- AnomalyDetectionVec(daily_sales[,4], alpha=0.05, period=365, direction='neg', only_l
negative_outlier$plot
```




```
neg_out <- negative_outlier$anoms$index
daily_sales[negative_outlier$anoms$index,]
```

```
## Source: local data frame [27 x 4]
## Groups: year, month [9]
##
##     year month   day daily_sales
##    <dbl> <dbl> <dbl>       <int>
## 1   2013     1     3      445340
## 2   2013     1     4      468261
## 3   2013     3    25     1043558
## 4   2013     3    26      962435
## 5   2013     4     1       33326
## 6   2013     4     2      552637
```

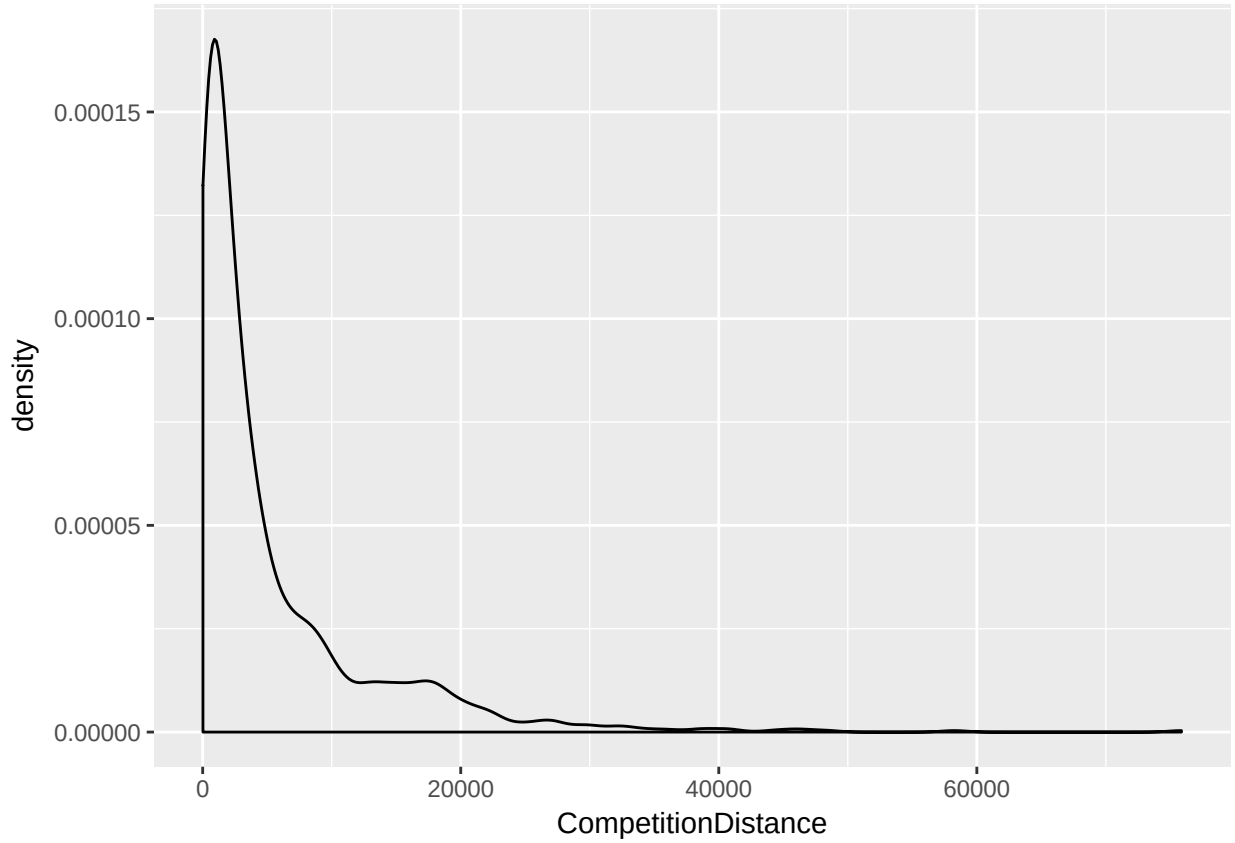
```
## 7   2013     4     3      631299
## 8   2013     4     4      636404
## 9   2013     4     5      635104
## 10  2013     4    17     5556697
## # ... with 17 more rows
```

Now that the indexes were found of these anomalies, it is important to determine what to do going forward. In order to better understand why these are anomalies, I want to see if Competitors or Promotions have an effect on these major events.

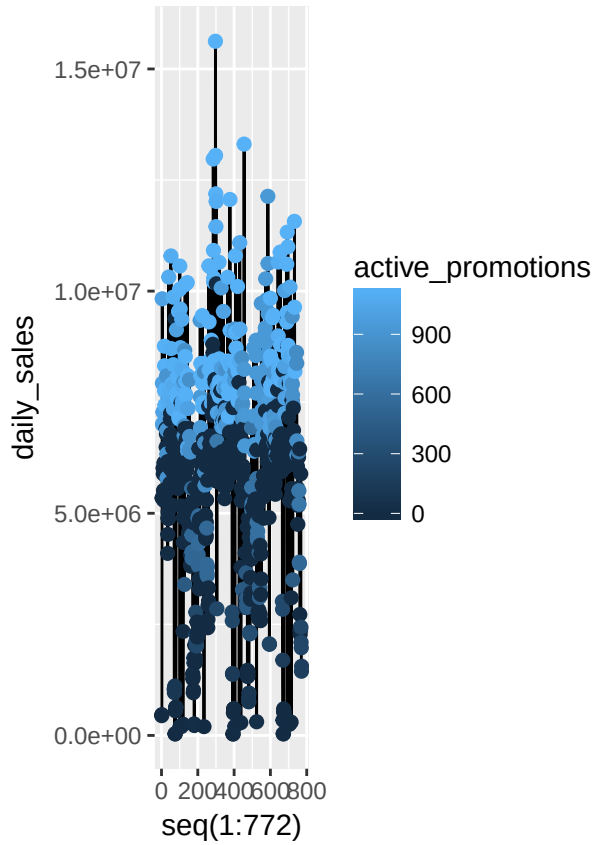
For a proof of concept, I will try the measures of total promotions and the average distance to competitor. I would assume that at times with more promotions, positive over-performing events may occur more frequently. Also, I would assume that as the mean distance from competitor decreases, the under performing events may occur more. Though the density graph

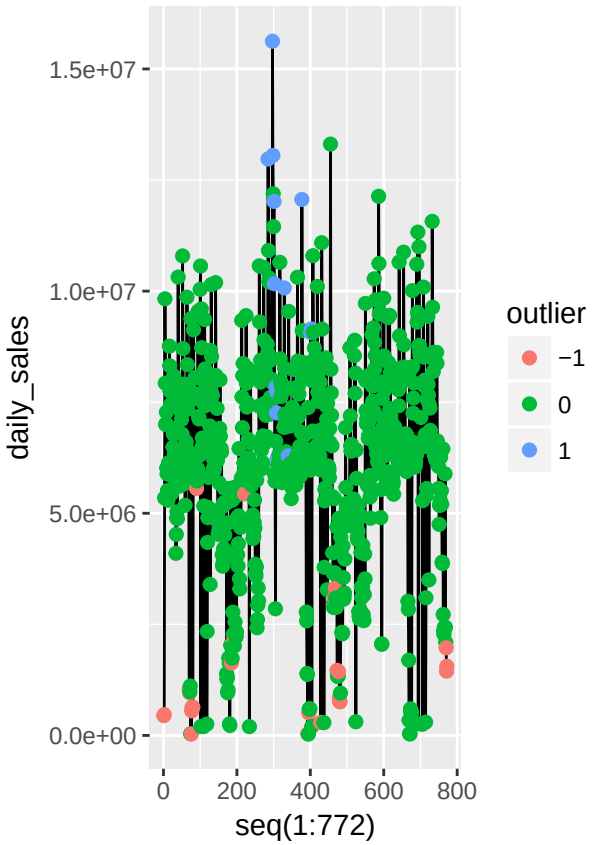
A difficulty with the competitors is that they are introduced at different times during the course of time. Thus, the mean will change over time.

```
promo_sales <- sales_cust %>%
  filter(Open == 1 & DayOfWeek != 7 & SchoolHoliday == 0) %>%
  group_by(year, month, day) %>%
  summarize(daily_sales =sum(Sales), active_promotions = sum(Promo))

#Density plot of Competitor distances
ggplot(store_comp2)+geom_density(aes(x=CompetitionDistance))
```




```
#Plot both the number of combined promotions and the sales over time
a <- ggplot(promo_sales) + geom_line(aes(x=seq(1:772), y=daily_sales))+geom_point(aes(x=seq(1:772), y=da

#Create an outlier identifier column in promo sales
#Have -1 be negative performer, 0 be normal, and 1 be positive
promo_sales$outlier <- 0
promo_sales[pos_out,]$outlier <- 1
promo_sales[neg_out,]$outlier <- -1
promo_sales$outlier <- as.factor(promo_sales$outlier)

#Color by promotions and outlier
b <- ggplot(promo_sales) + geom_line(aes(x=seq(1:772), y=daily_sales))+geom_point(aes(x=seq(1:772), y=da

gridExtra::grid.arrange(a, b, ncol=2) #suggest zooming to see full visual
```






```
#Combine the Competitor mean by month and year
#remove 1900 and 1961 for moving average
comp_dist <-store_comp2 %>% filter(CompetitionOpenSinceYear >=1990) %>%
  arrange(CompetitionOpenSinceYear, CompetitionOpenSinceMonth) %>%
  mutate(rolling_mean = cummean(CompetitionDistance)) %>%
  group_by(CompetitionOpenSinceMonth, CompetitionOpenSinceYear) %>%
  summarize(rolling_mean = mean(rolling_mean)) %>%
  arrange(CompetitionOpenSinceYear, CompetitionOpenSinceMonth)

#merge the comp_dist dataframe with promo_sales

promo_comp <- merge(promo_sales, comp_dist, by.x= c("year", "month"), by.y=c("CompetitionOpenSinceYear"
```

Now create a decision tree to see any decision rules for the outlier points

```
tree <- rpart(outlier ~ active_promotions + rolling_mean + daily_sales, data = promo_comp, method = "cl

printcp(tree)
```

```
##
## Classification tree:
## rpart(formula = outlier ~ active_promotions + rolling_mean +
##     daily_sales, data = promo_comp, method = "class")
##
## Variables actually used in tree construction:
```

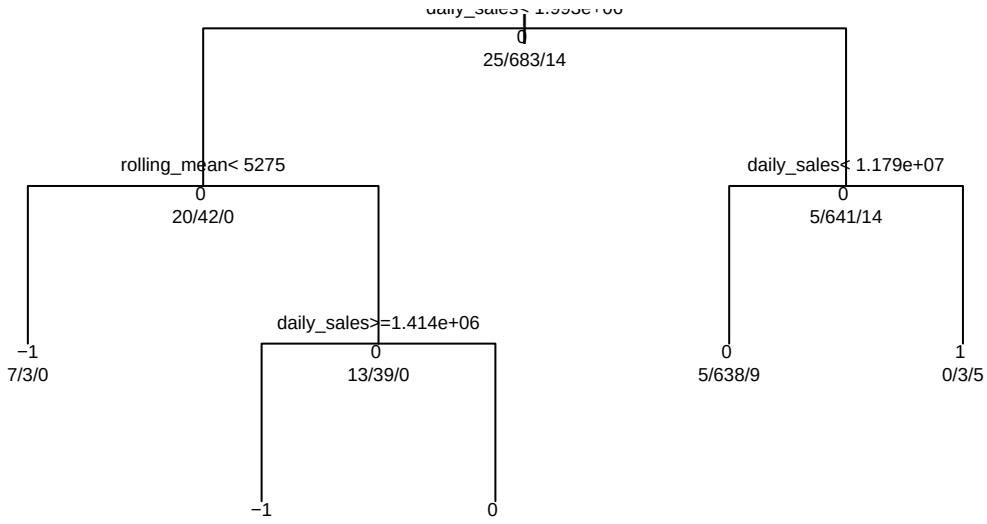
```
## [1] daily_sales  rolling_mean
##
## Root node error: 39/722 = 0.054017
##
## n= 722
##
##         CP nsplit rel error xerror    xstd
## 1 0.051282      0   1.00000 1.0000 0.15574
## 2 0.010000      4   0.79487 1.0256 0.15761
```

```
summary(tree)
```

```
## Call:
## rpart(formula = outlier ~ active_promotions + rolling_mean +
##     daily_sales, data = promo_comp, method = "class")
##   n= 722
##
##           CP nsplit rel error   xerror      xstd
## 1 0.05128205      0 1.0000000 1.000000 0.1557433
## 2 0.01000000      4 0.7948718 1.025641 0.1576119
##
## Variable importance
##       daily_sales      rolling_mean active_promotions
##                78                16                 7
##
## Node number 1: 722 observations,    complexity param=0.05128205
##   predicted class=0   expected loss=0.05401662  P(node) =1
##     class counts:    25   683    14
##    probabilities: 0.035 0.946 0.019
##   left son=2 (62 obs) right son=3 (660 obs)
##   Primary splits:
##       daily_sales       < 1992893  to the left,  improve=10.5412800, (0 missing)
##       rolling_mean      < 5460.151 to the left,  improve= 1.4019270, (0 missing)
##       active_promotions < 418.5    to the left,  improve= 0.8414247, (0 missing)
##
## Node number 2: 62 observations,    complexity param=0.05128205
##   predicted class=0   expected loss=0.3225806  P(node) =0.08587258
##     class counts:    20    42     0
##    probabilities: 0.323 0.677 0.000
##   left son=4 (10 obs) right son=5 (52 obs)
##   Primary splits:
##       rolling_mean      < 5274.75  to the left,  improve=3.396774, (0 missing)
##       daily_sales       < 616147.5 to the right, improve=2.457276, (0 missing)
##       active_promotions < 67       to the right, improve=1.357644, (0 missing)
##
## Node number 3: 660 observations,    complexity param=0.05128205
##   predicted class=0   expected loss=0.02878788  P(node) =0.9141274
##     class counts:     5   641    14
##    probabilities: 0.008 0.971 0.021
##   left son=6 (652 obs) right son=7 (8 obs)
##   Primary splits:
##       daily_sales       < 11794600 to the left,  improve=5.8313720, (0 missing)
##       rolling_mean      < 5460.151 to the left,  improve=1.6681820, (0 missing)
##       active_promotions < 934.5    to the left,  improve=0.6956612, (0 missing)
##
```

```
## Node number 4: 10 observations
##   predicted class=-1  expected loss=0.3  P(node) =0.01385042
##     class counts:     7     3     0
##    probabilities: 0.700 0.300 0.000
##
## Node number 5: 52 observations,    complexity param=0.05128205
##   predicted class=0   expected loss=0.25  P(node) =0.07202216
##     class counts:    13    39     0
##    probabilities: 0.250 0.750 0.000
##   left son=10 (12 obs) right son=11 (40 obs)
##   Primary splits:
##       daily_sales       < 1413570  to the right, improve=3.466667, (0 missing)
##       active_promotions < 67       to the right, improve=1.950000, (0 missing)
##       rolling_mean      < 5432.719 to the right, improve=1.208141, (0 missing)
##   Surrogate splits:
##       active_promotions < 149      to the right, agree=0.885, adj=0.500, (0 split)
##       rolling_mean      < 5310.06  to the left,  agree=0.808, adj=0.167, (0 split)
##
## Node number 6: 652 observations
##   predicted class=0   expected loss=0.02147239  P(node) =0.9030471
##     class counts:     5   638     9
##    probabilities: 0.008 0.979 0.014
##
## Node number 7: 8 observations
##   predicted class=1   expected loss=0.375  P(node) =0.01108033
##     class counts:     0     3     5
##    probabilities: 0.000 0.375 0.625
##
## Node number 10: 12 observations
##   predicted class=-1  expected loss=0.4166667  P(node) =0.0166205
##     class counts:     7     5     0
##    probabilities: 0.583 0.417 0.000
##
## Node number 11: 40 observations
##   predicted class=0   expected loss=0.15  P(node) =0.05540166
##     class counts:     6    34     0
##    probabilities: 0.150 0.850 0.000
```

```r
plot(tree, uniform=TRUE,
    main="Classification Tree for Outliers")
text(tree, use.n=TRUE, all=TRUE, cex=.6)
```


Classification Tree for Outliers
daily_sales< 1.993e+06
0
25/683/14
rolling_mean< 5275
0
20/42/0
daily_sales< 1.179e+07
0
5/641/14
−1
7/3/0
daily_sales>=1.414e+06
0
13/39/0
0
5/638/9
1
0/3/5
−1
0