R Project

Harshita Cherukuri

November 17, 2018

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

test<-fread(file = "test.csv")  
train<-fread(file = "train.csv")  
store<-fread(file = "store.csv")  
# View(train)  
# View(test)  
# View(store)  
str(train)

## Classes 'data.table' and 'data.frame': 1017209 obs. of 9 variables:  
## $ Store : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ DayOfWeek : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ Date : chr "2015-07-31" "2015-07-31" "2015-07-31" "2015-07-31" ...  
## $ Sales : int 5263 6064 8314 13995 4822 5651 15344 8492 8565 7185 ...  
## $ Customers : int 555 625 821 1498 559 589 1414 833 687 681 ...  
## $ Open : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Promo : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ StateHoliday : chr "0" "0" "0" "0" ...  
## $ SchoolHoliday: int 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

train[, Date := as.Date(Date)]  
test[, Date := as.Date(Date)]  
train<-train[order(Date)]  
#View(train)  
test<-test[order(Date)]  
summary(train)

## Store DayOfWeek Date Sales   
## Min. : 1.0 Min. :1.000 Min. :2013-01-01 Min. : 0   
## 1st Qu.: 280.0 1st Qu.:2.000 1st Qu.:2013-08-17 1st Qu.: 3727   
## Median : 558.0 Median :4.000 Median :2014-04-02 Median : 5744   
## Mean : 558.4 Mean :3.998 Mean :2014-04-11 Mean : 5774   
## 3rd Qu.: 838.0 3rd Qu.:6.000 3rd Qu.:2014-12-12 3rd Qu.: 7856   
## Max. :1115.0 Max. :7.000 Max. :2015-07-31 Max. :41551   
## Customers Open Promo StateHoliday   
## Min. : 0.0 Min. :0.0000 Min. :0.0000 Length:1017209   
## 1st Qu.: 405.0 1st Qu.:1.0000 1st Qu.:0.0000 Class :character   
## Median : 609.0 Median :1.0000 Median :0.0000 Mode :character   
## Mean : 633.1 Mean :0.8301 Mean :0.3815   
## 3rd Qu.: 837.0 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :7388.0 Max. :1.0000 Max. :1.0000   
## SchoolHoliday   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.1786   
## 3rd Qu.:0.0000   
## Max. :1.0000

dim(train)

## [1] 1017209 9

#The test set 1017209 observations.  
summary(test)

## Id Store DayOfWeek Date   
## Min. : 1 Min. : 1.0 Min. :1.000 Min. :2015-08-01   
## 1st Qu.:10273 1st Qu.: 279.8 1st Qu.:2.000 1st Qu.:2015-08-12   
## Median :20544 Median : 553.5 Median :4.000 Median :2015-08-24   
## Mean :20544 Mean : 555.9 Mean :3.979 Mean :2015-08-24   
## 3rd Qu.:30816 3rd Qu.: 832.2 3rd Qu.:6.000 3rd Qu.:2015-09-05   
## Max. :41088 Max. :1115.0 Max. :7.000 Max. :2015-09-17   
##   
## Open Promo StateHoliday SchoolHoliday   
## Min. :0.0000 Min. :0.0000 Length:41088 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:0.0000 Class :character 1st Qu.:0.0000   
## Median :1.0000 Median :0.0000 Mode :character Median :0.0000   
## Mean :0.8543 Mean :0.3958 Mean :0.4435   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## NA's :11

dim(test)

## [1] 41088 8

#The test dtaa set has 41088 observations.  
test[is.na(test$Open),]

## Id Store DayOfWeek Date Open Promo StateHoliday SchoolHoliday  
## 1: 10752 622 6 2015-09-05 NA 0 0 0  
## 2: 9040 622 1 2015-09-07 NA 0 0 0  
## 3: 8184 622 2 2015-09-08 NA 0 0 0  
## 4: 7328 622 3 2015-09-09 NA 0 0 0  
## 5: 6472 622 4 2015-09-10 NA 0 0 0  
## 6: 5616 622 5 2015-09-11 NA 0 0 0  
## 7: 4760 622 6 2015-09-12 NA 0 0 0  
## 8: 3048 622 1 2015-09-14 NA 1 0 0  
## 9: 2192 622 2 2015-09-15 NA 1 0 0  
## 10: 1336 622 3 2015-09-16 NA 1 0 0  
## 11: 480 622 4 2015-09-17 NA 1 0 0

#only store number 622 is not open.  
test$Open[test$Store == 622]

## [1] 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0  
## [24] 1 1 1 1 1 1 0 1 1 1 1 1 NA 0 NA NA NA NA NA NA 0 NA NA  
## [47] NA NA

#37 observations of store number 622.   
#Additionally the whole "customers" column is missing from the test data since this data is know only later.   
  
test[is.na(test)] <- 1  
#during the test period there are no Stateholidays  
#test$Store[test$SchoolHoliday == 1]  
991/1722

## [1] 0.5754936

# A large percent of the time (57.5%) of the time there are school holidays.

## Including Plots

You can also embed plots, for example:

## Store DayOfWeek Date Sales Customers Open Promo StateHoliday  
## 1: 1115 7 942 21734 4086 2 2 4  
## SchoolHoliday  
## 1: 2

## Id Store DayOfWeek Date Open Promo StateHoliday SchoolHoliday  
## 1: 41088 856 7 48 2 2 2 2

## [1] 856

## [1] 259

##   
## 0 1   
## 0.1698933 0.8301067

##   
## 0 1   
## 0.1456386 0.8543614

##   
## 0 1   
## 0.6184855 0.3815145

##   
## 0 1   
## 0.6041667 0.3958333

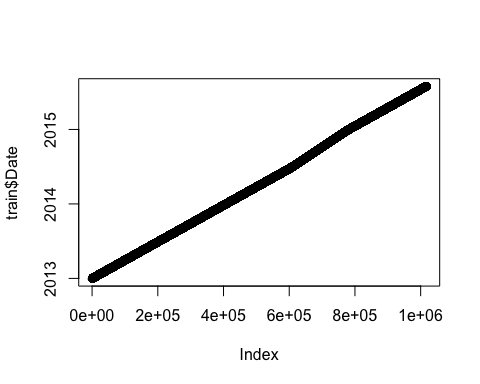
##   
## 0 a   
## 0.995619159 0.004380841

##   
## 0 a b c   
## 0.969475300 0.019917244 0.006576820 0.004030637

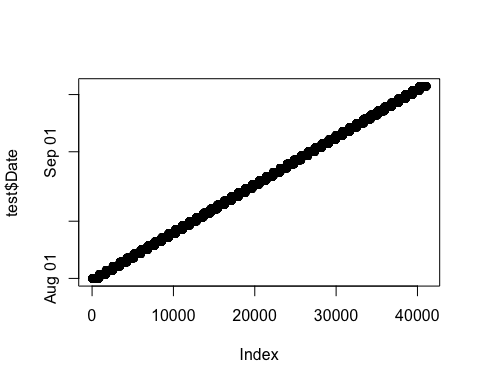
##   
## 0 1   
## 0.8213533 0.1786467

##   
## 0 1   
## 0.5565129 0.4434871

plot(train$Date)



plot(test$Date)



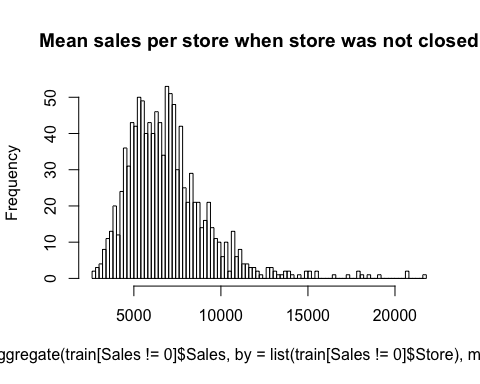
all(table(test$Date) == 856)

## [1] TRUE

#Columns that are unique to the train set  
  
hist(train$Sales, 100)



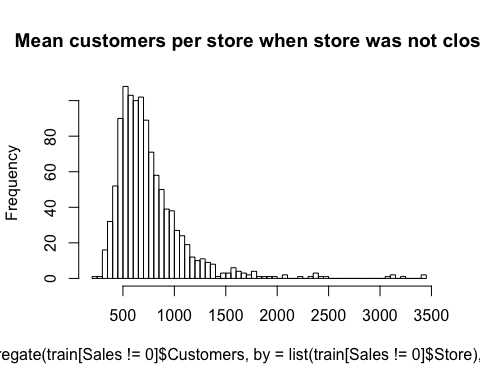
hist(aggregate(train[Sales != 0]$Sales,   
 by = list(train[Sales !=0]$Store), mean)$x, 100, main = "Mean sales per store when store was not closed")



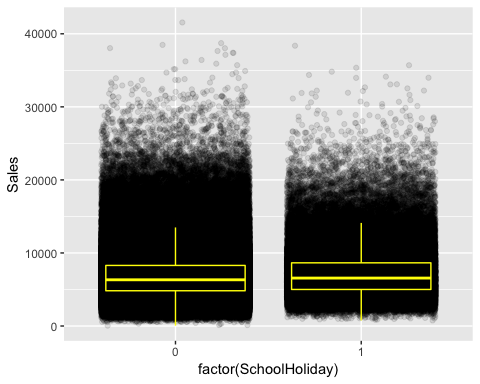
hist(train$Customers, 100)



hist(aggregate(train[Sales !=0]$Customers,  
 by = list(train[Sales !=0]$Store), mean)$x, 100, main = "Mean customers per store when store was not closed")

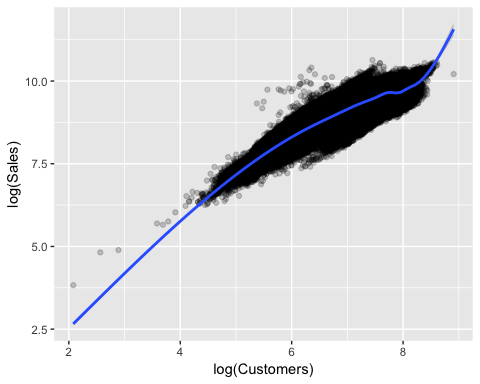


ggplot(train[Sales != 0], aes(x= factor(SchoolHoliday), y= Sales))+  
 geom\_jitter(alpha=0.1)+  
 geom\_boxplot(color = "yellow", outlier.colour = NA, fill = NA)

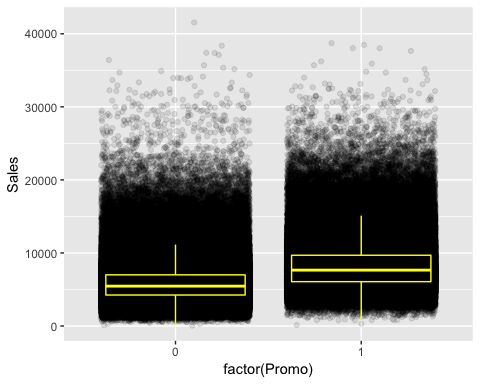


ggplot(train[train$Sales != 0 & train$Customers !=0],  
 aes(x=log(Customers), y = log(Sales)))+  
 geom\_point(alpha=0.2)+  
 geom\_smooth()

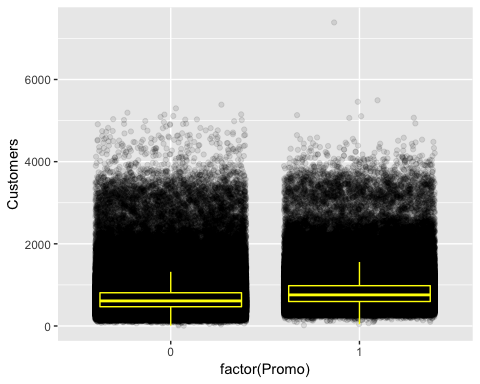
## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



ggplot(train[train$Sales !=0 & train$Customers!=0],  
 aes(x=factor(Promo), y = Sales))+  
 geom\_jitter(alpha = 0.1)+  
 geom\_boxplot(color = "yellow", outlier.color = NA, fill = NA)



ggplot(train[train$Sales !=0 & train$Customers!=0],  
 aes(x=factor(Promo), y = Customers))+  
 geom\_jitter(alpha = 0.1)+  
 geom\_boxplot(color = "yellow", outlier.color = NA, fill = NA)



#Sales is asexpected strongly correlated with the number of customers. It looks like Boxplots of customers overlap a little more than the boxplots of sales. This would mean that the promos are not mainly attracting more customers but make customer spend ore. The mean amount spent per customer is about one Euro Higher.

with(train[train$Sales != 0 & train$Promo ==0], mean(Sales/Customers))

## [1] 8.941128

with(train[train$Sales != 0 & train$Promo ==1], mean(Sales/Customers))

## [1] 10.17896

table(ifelse(train$Sales !=0, "Sales>0", "Sales = 0"),  
 ifelse(train$Promo, "Promo", "No promo"))

##   
## No promo Promo  
## Sales = 0 161666 11205  
## Sales>0 467463 376875

table(ifelse(train$Open ==1, "Opened", "Closed"),  
 ifelse(train$Sales, "Sales>0", "Sales = 0"))

##   
## Sales = 0 Sales>0  
## Closed 172817 0  
## Opened 54 844338

table(ifelse(train$Sales !=0, "Sales>0", "Sales =0"),  
 ifelse(train$Customers, "Customers>0", "Customers = 0"))

##   
## Customers = 0 Customers>0  
## Sales =0 172869 2  
## Sales>0 0 844338

summary(train$StateHoliday)

## Length Class Mode   
## 1017209 character character

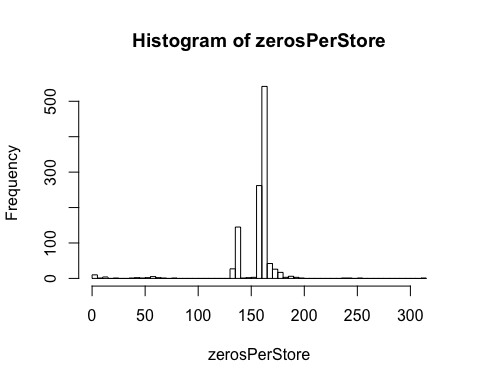
table(ifelse(train$StateHoliday !=0, "Public", "Easter"),  
 ifelse(train$Sales, "Sales>0", "Sales = 0"))

##   
## Sales = 0 Sales>0  
## Easter 142731 843428  
## Public 30140 910

table(ifelse(train$Open ==1, "Opened", "Closed"),  
 ifelse(train$Customer, "Customers>0", "Customers= 0"))

##   
## Customers= 0 Customers>0  
## Closed 172817 0  
## Opened 52 844340

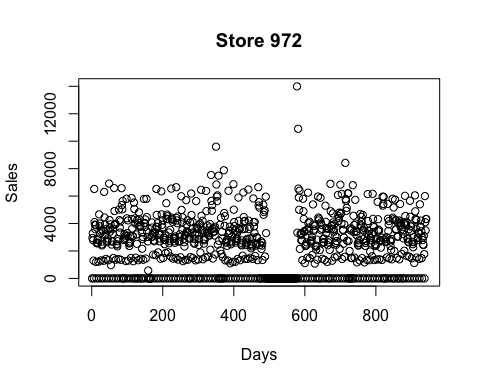
#train[Open ==1 & Sales == 0]  
  
#there are no sales when the store is closed but there are some store that are open but have no sales even if they had some customers. These observations may be error in the data or outliers.  
  
zerosPerStore <- sort(tapply(train$Sales, list(train$Store), function(x) sum(x == 0)))  
hist(zerosPerStore, 100)



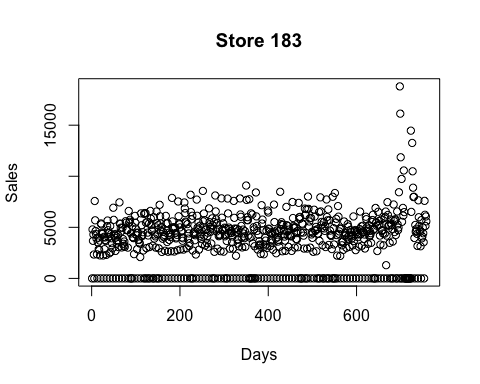
#Stores with most zero sales  
tail(zerosPerStore, 10)

## 105 339 837 25 560 674 972 349 708 103   
## 188 188 191 192 195 197 240 242 255 311

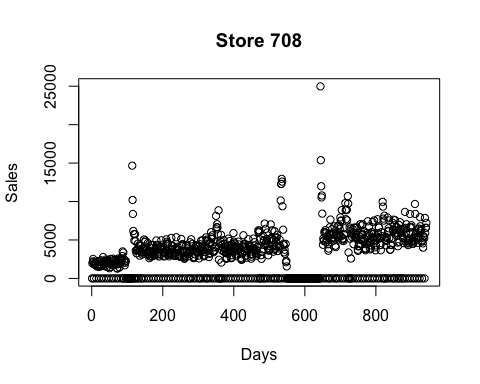
#The store have different amounts of days with zero sales. There ar espikes in e sales before the stores close and after the stoes reopen.  
plot(train[Store == 972, Sales], ylab = "Sales", xlab = "Days", main = "Store 972")



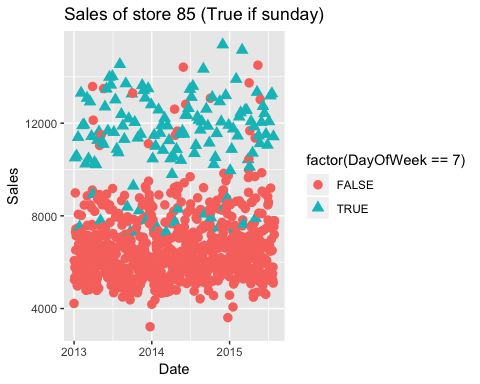
plot(train[Store == 183, Sales], ylab = "Sales", xlab = "Days", main = "Store 183")



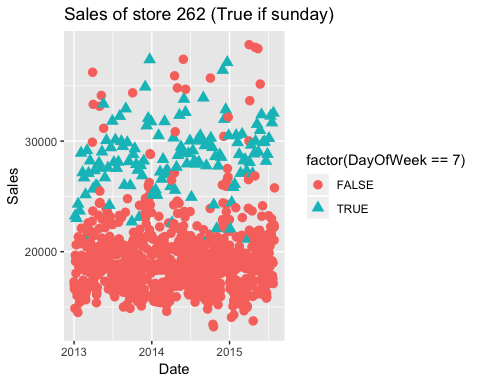
plot(train[Store == 708, Sales], ylab = "Sales", xlab = "Days", main = "Store 708")



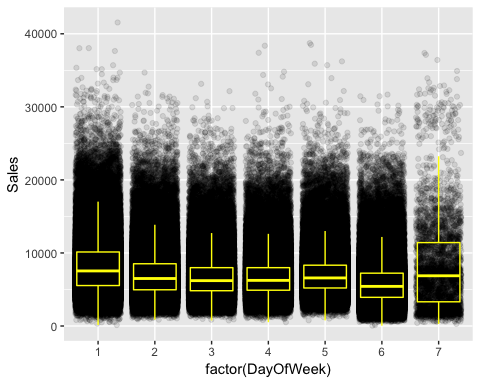
#There are stores that have no zeros in their sales. these are exceptions because they are opened also on Sundays/holidays. The Sales  
ggplot(train[Store == 85],   
 aes(x = Date, y = Sales,   
 color = factor(DayOfWeek == 7), shape = factor(DayOfWeek == 7))) +   
 geom\_point(size = 3) + ggtitle("Sales of store 85 (True if sunday)")



ggplot(train[Store == 262],   
 aes(x = Date, y = Sales,   
 color = factor(DayOfWeek == 7), shape = factor(DayOfWeek == 7))) +   
 geom\_point(size = 3) + ggtitle("Sales of store 262 (True if sunday)")



ggplot(train[Sales!=0],  
 aes(x= factor(DayOfWeek), y=Sales))+  
 geom\_jitter(alpha=0.1)+  
 geom\_boxplot(color= "yellow", outlier.colour = NA, fill=NA)



#There is a lot more varuiability on Sundys than on other days, depsite median beign almost the same.

##The store file contains information about the stores that can be linked to traina dn test datasets via store ID.  
summary(store)

## Store StoreType Assortment   
## Min. : 1.0 Length:1115 Length:1115   
## 1st Qu.: 279.5 Class :character Class :character   
## Median : 558.0 Mode :character Mode :character   
## Mean : 558.0   
## 3rd Qu.: 836.5   
## Max. :1115.0   
##   
## CompetitionDistance CompetitionOpenSinceMonth CompetitionOpenSinceYear  
## Min. : 20.0 Min. : 1.000 Min. :1900   
## 1st Qu.: 717.5 1st Qu.: 4.000 1st Qu.:2006   
## Median : 2325.0 Median : 8.000 Median :2010   
## Mean : 5404.9 Mean : 7.225 Mean :2009   
## 3rd Qu.: 6882.5 3rd Qu.:10.000 3rd Qu.:2013   
## Max. :75860.0 Max. :12.000 Max. :2015   
## NA's :3 NA's :354 NA's :354   
## Promo2 Promo2SinceWeek Promo2SinceYear PromoInterval   
## Min. :0.0000 Min. : 1.0 Min. :2009 Length:1115   
## 1st Qu.:0.0000 1st Qu.:13.0 1st Qu.:2011 Class :character   
## Median :1.0000 Median :22.0 Median :2012 Mode :character   
## Mean :0.5121 Mean :23.6 Mean :2012   
## 3rd Qu.:1.0000 3rd Qu.:37.0 3rd Qu.:2013   
## Max. :1.0000 Max. :50.0 Max. :2015   
## NA's :544 NA's :544

# View(store)  
table(store$StoreType)

##   
## a b c d   
## 602 17 148 348

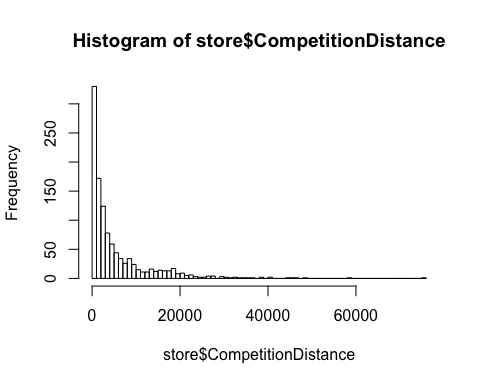
table(store$Assortment)

##   
## a b c   
## 593 9 513

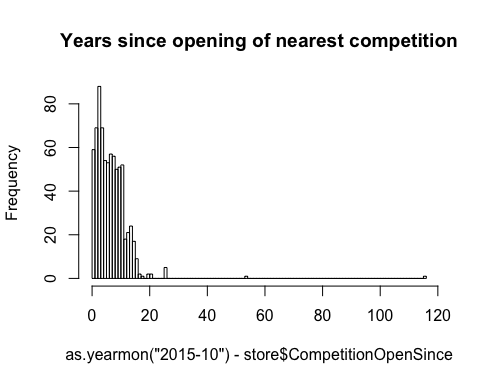
table(data.frame(Assortment = store$Assortment, StoreType = store$StoreType))

## StoreType  
## Assortment a b c d  
## a 381 7 77 128  
## b 0 9 0 0  
## c 221 1 71 220

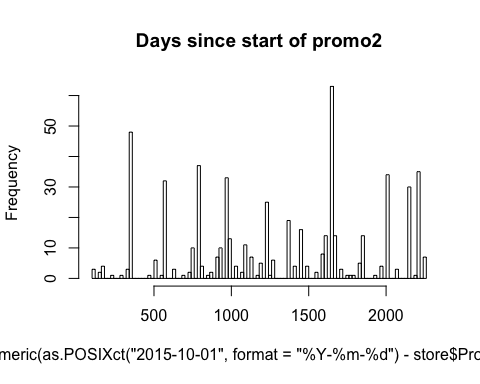
#There seems to be a connection between the StoreType and Assortment  
hist(store$CompetitionDistance,100)



#A huge number of stores are closely located.  
store$CompetitionOpenSince<- as.yearmon(paste(store$CompetitionOpenSinceYear,store$CompetitionOpenSinceMonth, sep = "-"))  
# View(store$CompetitionOpenSince)  
#One competitor opened in 1900  
hist(as.yearmon("2015-10")-store$CompetitionOpenSince, 100, main="Years since opening of nearest competition")



#Assume that the promo starts on the first day of the week.   
store$Promo2Since <- as.POSIXct(paste(store$Promo2SinceYear, store$Promo2SinceWeek, 1, sep = "-"),  
 format = "%Y-%U-%u")  
hist(as.numeric(as.POSIXct("2015-10-01", format = "%Y-%m-%d") - store$Promo2Since),   
 100, main = "Days since start of promo2")

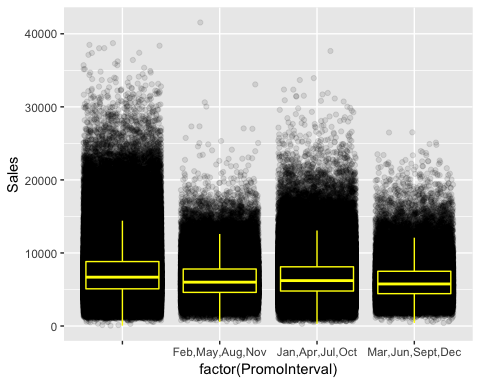


table(store$PromoInterval)

##   
## Feb,May,Aug,Nov Jan,Apr,Jul,Oct Mar,Jun,Sept,Dec   
## 544 130 335 106

#Seems like maximum number of the stores have no specific promo interval.

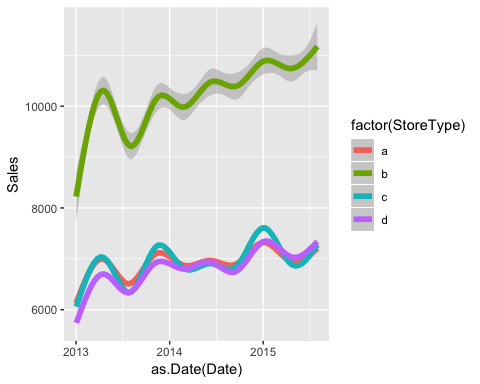
train\_store<- merge(train, store, by = "Store")  
ggplot(train\_store[Sales!=0], aes(x=factor(PromoInterval), y=Sales))+  
 geom\_jitter(alpha=0.1)+  
 geom\_boxplot(color="yellow", outlier.colour = NA, fill = NA)



#The stores with promos tend to make lower sales. This doesnt necessarily mean promos dont help or are counterproductive. They are possibly measures that are taken mainly by stores with low sales in the first place.

ggplot(train\_store[Sales != 0],   
 aes(x = as.Date(Date), y = Sales, color = factor(StoreType))) +   
 geom\_smooth(size = 2)

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



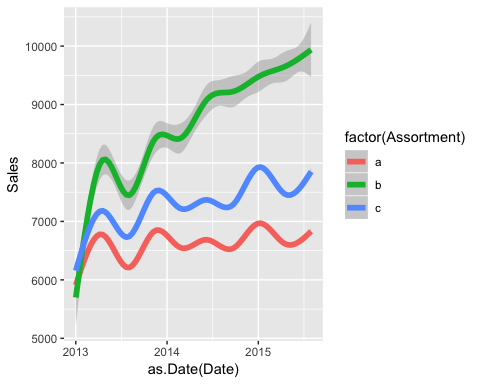
#Store with extra asssortment type tend to have the highest sales.  
ggplot(train\_store[Customers != 0],   
 aes(x = as.Date(Date), y = Customers, color = factor(StoreType))) +   
 geom\_smooth(size = 2)

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



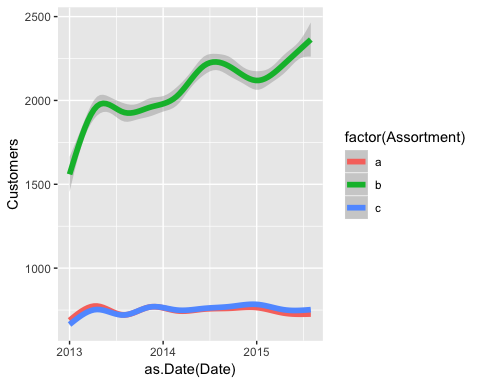
#stores with extra assortment type have the highest number of customers and its been increasing over the years.  
ggplot(train\_store[Sales != 0],   
 aes(x = as.Date(Date), y = Sales, color = factor(Assortment))) +   
 geom\_smooth(size = 2)

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



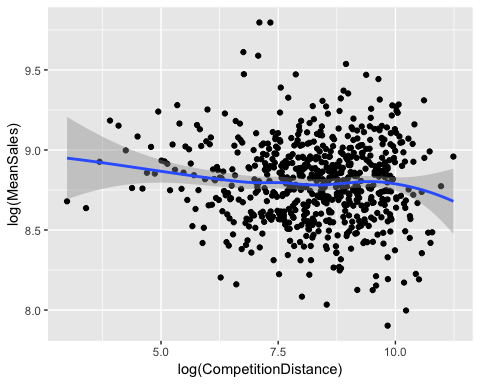
ggplot(train\_store[Sales != 0],   
 aes(x = as.Date(Date), y = Customers, color = factor(Assortment))) +   
 geom\_smooth(size = 2)

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



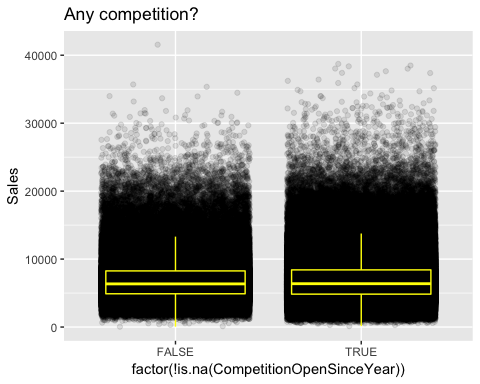
salesByDist <- aggregate(train\_store[Sales != 0 & !is.na(CompetitionDistance)]$Sales,   
 by = list(train\_store[Sales != 0 & !is.na(CompetitionDistance)]$CompetitionDistance), mean)  
colnames(salesByDist) <- c("CompetitionDistance", "MeanSales")  
ggplot(salesByDist, aes(x = log(CompetitionDistance), y = log(MeanSales))) +   
 geom\_point() + geom\_smooth()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



#The effect of the distance to the next competitor is a little counterintuitive.Lower distance to the next competitor implies (slightly, possibly not significantly)higher sales. This may occur(my assumption) because stores with a low distance to the next competitor are located in inner cities or crowded regions with higher sales in general. Maybe the effects of being in a good / bad region and having a competitor / not having a competitor cancel out:

ggplot(train\_store[Sales != 0],  
 aes(x = factor(!is.na(CompetitionOpenSinceYear)), y = Sales)) +  
 geom\_jitter(alpha = 0.1) +  
 geom\_boxplot(color = "yellow", outlier.colour = NA, fill = NA) +  
 ggtitle("Any competition?")



#There seems to be no obvious connection between sales and having NA as the competition distance.

# #Sales before and after the competitione opens.   
# train\_store$DateYearmon <- as.yearmon(train\_store$Date)  
# train\_store <- train\_store[order(Date)]  
# timespan <- 100 # Days to collect before and after Opening of competition  
# beforeAndAfterComp<- function(s){  
# x<- train\_store[Store == s]  
# daysWithComp <- x$CompetitionOpenSince >= x$DateYearmon  
# if(any(!daysWithComp)){  
# compOpening <- head(which(!daysWithComp), 1) - 1  
# if(compOpening > timespan & compOpening < (nrow(x)-timespan)){  
# x<-x[(compOpening-timespan):(compOpening+timespan), ]  
# x$Day <-1:nrow(x)  
# return(x)  
# }  
# }  
# }  
#   
# temp <- lapply(unique(train\_store[!is.na(CompetitionOpenSince)]$Store), beforeAndAfterComp)  
# temp <- do.call(rbind, temp)  
# length(unique(temp$Store))  
# ggplot(temp[train$Sales !=0], aes(x="Date", y=train$Sales))+  
# geom\_smooth()+  
# ggtitle(paste("Competition opening around day", timespan))

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.