ITESO

Rossman Store Sales

Exploración de datos

Sara Eugenia Rodríguez Reyes

21/10/2015

Introducción

Rossman es una empresa que opera más de 3,000 farmacias en 7 países europeos. Actualmente los gerentes de las tiendas Rossman tienen la tarea de predecir sus ventas diarias con seis semanas de antelación.

Las ventas son influenciadas por muchos factores, ya sean promociones, la competencia, las escuelas, días festivos, estacionalidad y localidad.

El desafío es predecir seis semanas de ventas diarias de 1,115 tiendas ubicadas en Alemania. Estas predicciones permiten la creación de horarios eficaces del personal que aumenten la productividad y la motivación; así como enfocarse a una mejor atención al cliente y sus equipos.

Para poder predecir esto, primero es necesario conocer los datos y explorarlos. Que es lo que se hará en esta primera etapa.

Se tienen 3 archivos:

- train.csv – datos históricos incluyendo ventas
- test.csv – datos históricos excluyendo ventas
- store.csv – información complementaria sobre las tiendas

A continuación se presenta el código con el que se hizo la exploración de datos; este mismo es comentado. Se presentan gráficas con explicaciones de patrones encontrados en los datos.

```
##Fijar el directorio de trabajo
setwd("C:/Users/Sara/Documents/ITESO/PAP2/Rossman")
#Cargar librerias
library(plyr)
library(dplyr)
library(ggplot2)
library(forecast)
library(data.table)
library(zoo)
library(missForest)
##Cargar archivos
train <- read.csv("train.csv",header=TRUE)
test <- read.csv("test.csv",header=TRUE)
store <- read.csv("store.csv",header=TRUE)
```

Análisis de NA's

Al archivo store.csv le faltan muchos datos, es por esto que el hacer una predicción sin antes tomar acción, puede que no sea la mejor opción y los resultados se distorsionen. Por lo tanto se hará un análisis de los NA para tomar la decisión de eliminar esas filas o utilizar algún método para reemplazarlos.

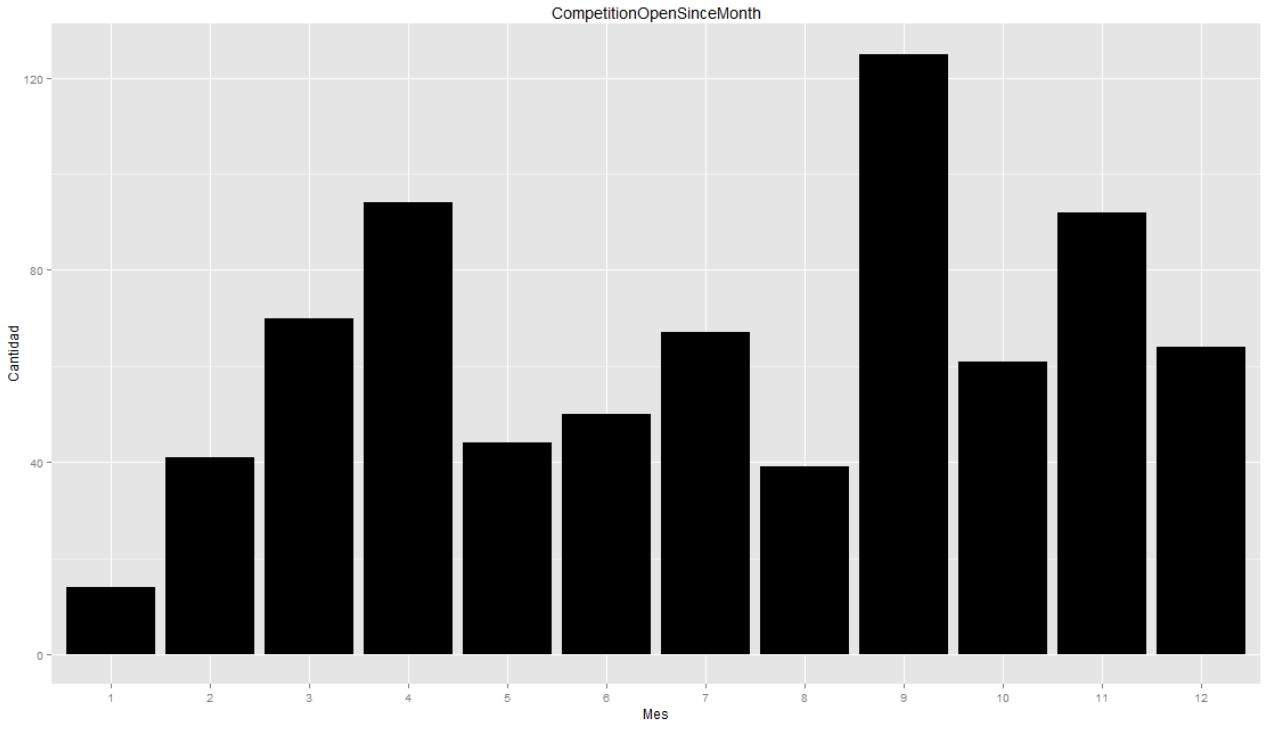
```
###ANALISIS DE LOS NA
##Cuantos NA hay en cada variable del data set store
na<- 0
for (i in 1:length(names(store))){
na[i] <- sum(is.na(store[,i]))
}
nas <- as.matrix(cbind(names(store),na))
```

Store	0
StoreType	0
Assortment	0
CompetitionDistance	3
CompetitionOpenSinceMonth	354
CompetitionOpenSinceYear	354
Promo2	0
Promo2SinceWeek	544
Promo2SinceYear	544
PromoInterval	0

Las variables “CompetitionOpenSinceMonth” y “CompetitionOpenSinceYear” tienen ambas 354 NAs; por lo tanto vamos a analizarlas.

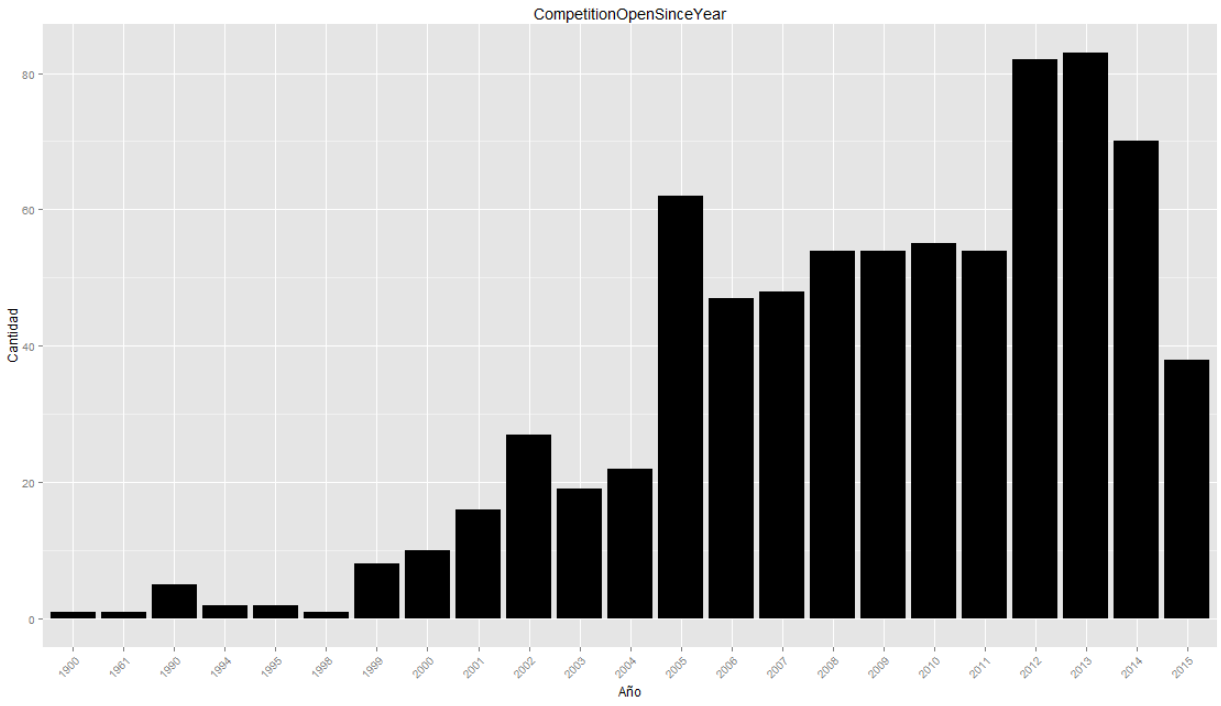
```
###CompetitionOpenSinceMonth
compMes <-
as.data.frame(table(store$CompetitionOpenSinceMonth))#Competencia por
mes
colnames(compMes) <- c("Mes", "Cantidad")
ggplot(compMes, aes(x = Mes, y = Cantidad)) +
geom_bar(stat = "identity",
fill = "black") +
labs(title = "CompetitionOpenSinceMonth")
```

Se observa que el mes 9 (Septiembre) es cuando más competidores cercanos abrieron tiendas.



```
### CompetitionOpenSinceYear
compAno <- as.data.frame(table(store$CompetitionOpenSinceYear))
colnames(compAno) <- c("Año", "Cantidad")
ggplot(compAno, aes(x = Año, y = Cantidad)) +
geom_bar(stat = "identity",
fill = "black") +
labs(title = "CompetitionOpenSinceYear") +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

En el año 2013 fue donde se abrieron más tiendas de competidores.



Ahora analizamos las promociones. Cuando se empieza una promoción, lo esperado es que las ventas aumenten.

Las variables “Promo2SinceWeek” y “Promo2SinceYear” tienen ambas 544 NAs.

```
table(store$Promo2SinceWeek, store$Promo2SinceYear)
table(store$Promo2)
```

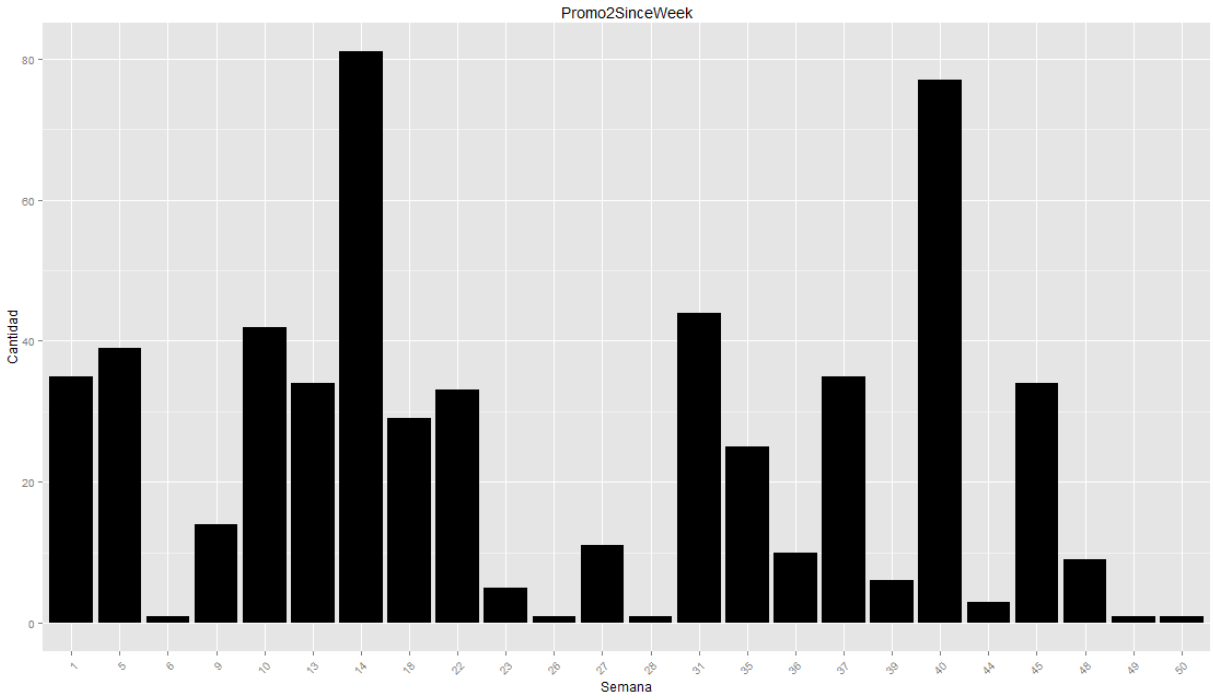
	2009	2010	2011	2012	2013	2014	2015
1	0	0	0	19	13	3	0
5	0	3	3	0	33	0	0
6	0	0	0	0	0	0	1
9	0	0	14	0	0	0	0
10	0	0	0	0	10	32	0
13	0	34	0	0	0	0	0
14	0	0	63	6	7	1	4
18	0	4	14	1	2	6	2
22	0	0	8	25	0	0	0
23	0	0	0	0	1	1	3
26	0	1	0	0	0	0	0
27	0	0	2	5	4	0	0
28	0	0	0	1	0	0	0
31	7	0	0	0	37	0	0
35	0	14	4	7	0	0	0
36	0	0	0	0	10	0	0
37	35	0	0	0	0	0	0
39	1	5	0	0	0	0	0
40	0	0	16	11	2	48	0
44	0	1	0	2	0	0	0
45	30	0	0	0	1	3	0
48	0	1	4	4	0	0	0
49	0	0	0	0	0	1	0
50	0	1	0	0	0	0	0

Se observa que en la semana 18, en casi todos los años fue cuando las tiendas empezaron a participar en la Promo2. En el año 2011 y 2014 fue cuando más tiendas empezaron a participar.

No participan	Participan
544	571

El 51.21% de promociones están continua y consecutivamente en tiendas, mientras que el resto no; como se mencionó anteriormente, se esperaría que si se quiere aumentar las ventas, mayor número de tiendas deberían participar en las promociones.

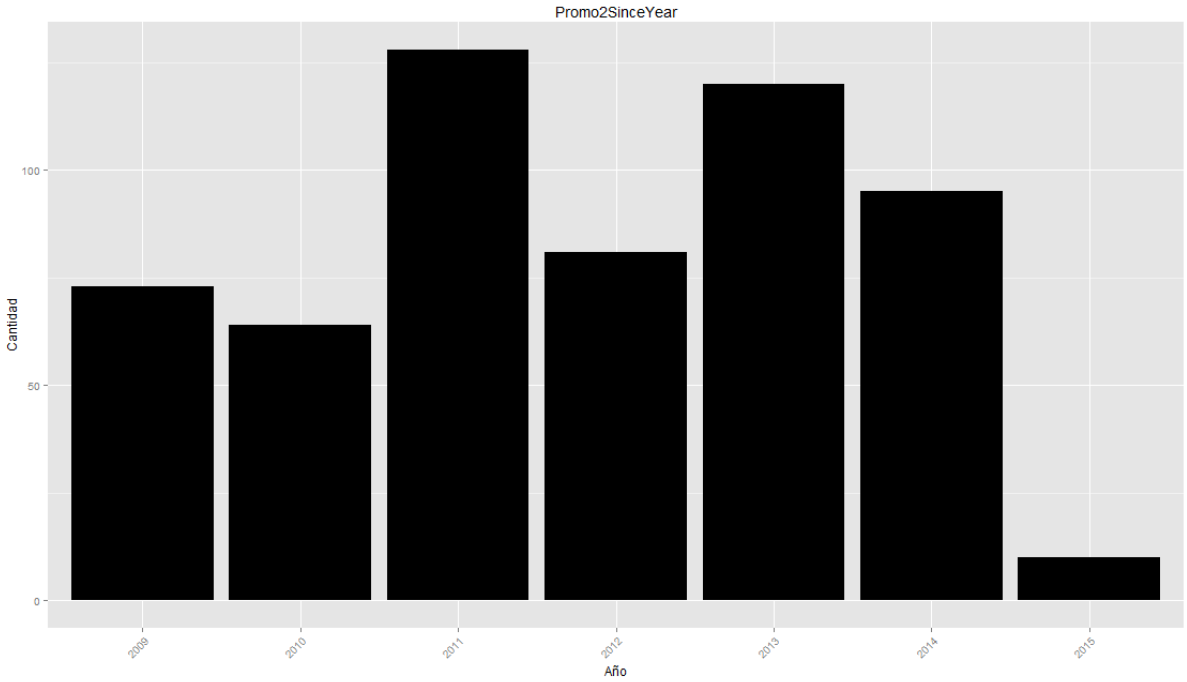
```
##Promo2SinceWeek
promoSem <- as.data.frame(table(store$Promo2SinceWeek))
colnames(promoSem) <- c("Semana", "Cantidad")
ggplot(promoSem, aes(x = Semana, y = Cantidad)) +
geom_bar(stat = "identity",
fill = "black") +
labs(title = "Promo2SinceWeek") +
theme(axis.text.x = element text(angle = 45, hjust = 1))
```



En la semana 14 fue cuando más tiendas empezaron a participar en la Promo2.

```
## Promo2SinceYear
promoAño <- as.data.frame(table(store$Promo2SinceYear))
colnames(promoAño) <- c("Año", "Cantidad")
ggplot(promoAño, aes(x = Año, y = Cantidad)) +
geom_bar(stat = "identity",
fill = "black") +
labs(title = "Promo2SinceYear") +
theme(axis.text.x = element text(angle = 45, hjust = 1))
```

En el 2011 fue cuando más tiendas comenzaron a participar en la Promo2.



Imputación de los NA's

A continuación se imputan los datos NA en el conjunto de datos Store, utilizando la librería MissForest. Esta librería se utiliza para imputar valores faltantes en el caso donde hay datos de tipo mixto. Puede ser utilizada para imputar los datos continuos y/o categóricos. Se puede ejecutar en paralelo para ahorrar tiempo de cálculo.

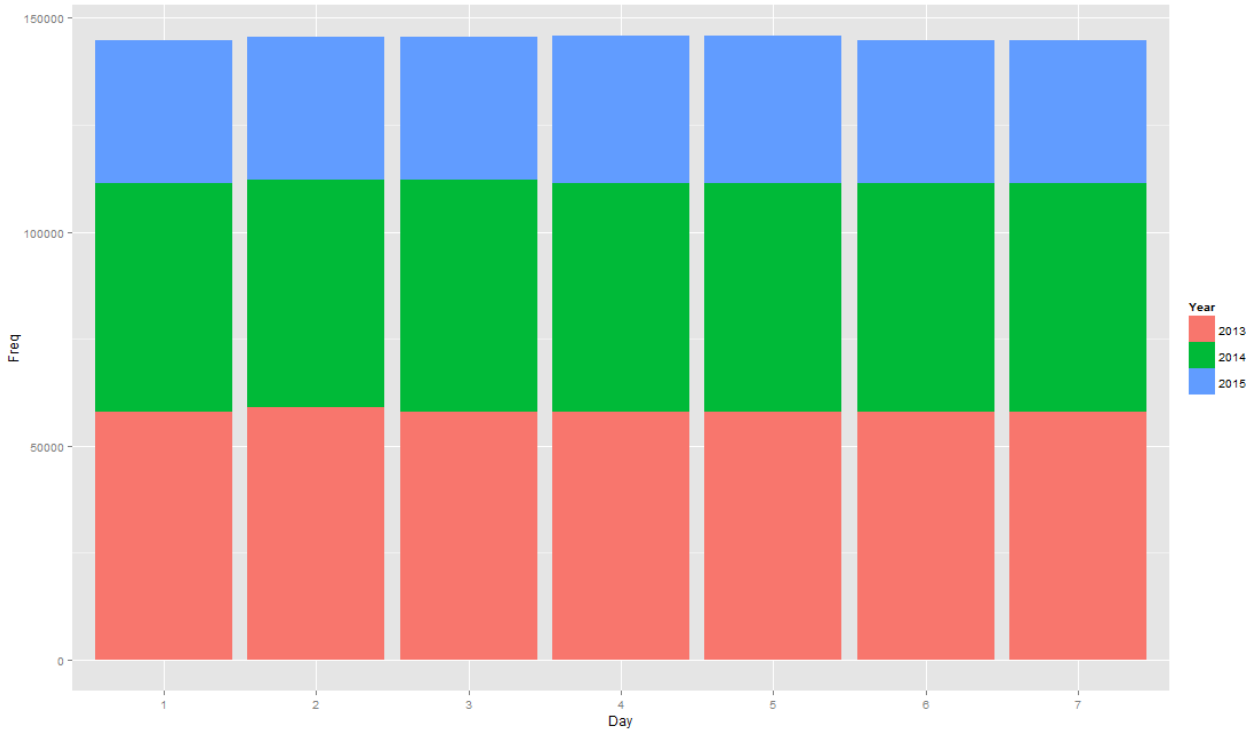
```
## Imputacion de NA's en el conjunto de datos Store
summary(store)
##El conjunto de datos contiene 7 variables continuas y 3 variables
categoricas
set.seed(1)
##verbose=FALSE para no ver lo que pasa en cada iteracion:
store.imp <- missForest(store, xtrue = store, verbose = FALSE)
##Unir dataset train y store
data <- left_join(train, store, by = "Store")
data <- filter(train, Open == 1) #Para analizar cuando las tiendas estan
abiertas
#Separar el año
train$year <- 1:nrow(train)
train$year[grep("2013",train$Date)] <- 2013
train$year[grep("2014",train$Date)] <- 2014
train$year[grep("2015",train$Date)] <- 2015
```

Exploración de datos

Se comienza observando cómo se comportan las ventas en los diferentes días de la semana durante los años.

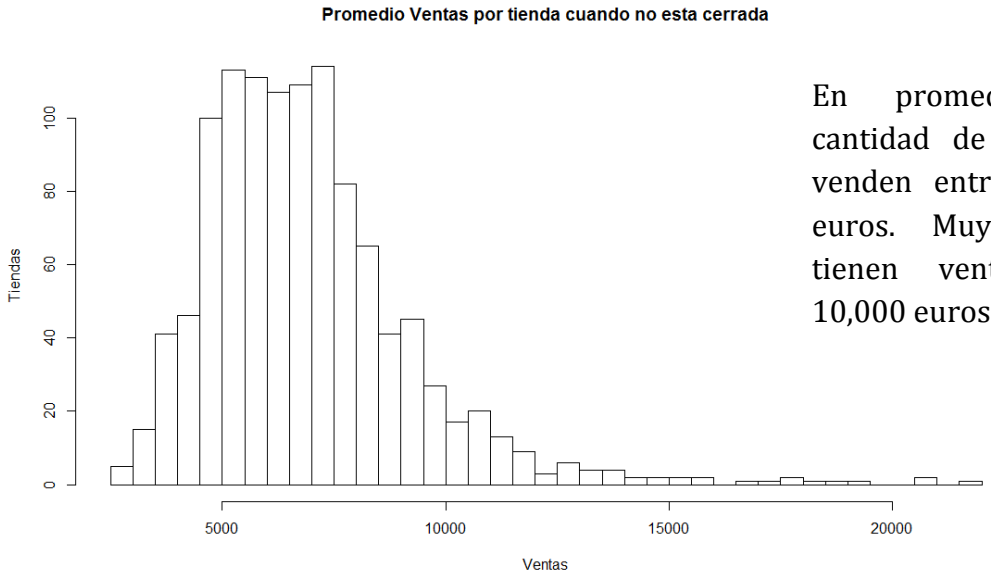
```
xyTable <- table(train$DayOfWeek, train$year)
xyTable <- as.data.frame(xyTable)
names(xyTable) <- c("Day", "Year", "Freq")
ggplot(xyTable, aes(x=Day, y = Freq, fill=Year)) +
geom_bar(stat="identity")
```

Se observa que es muy similar el comportamiento de las ventas en los 7 días de la semana durante los tres años.



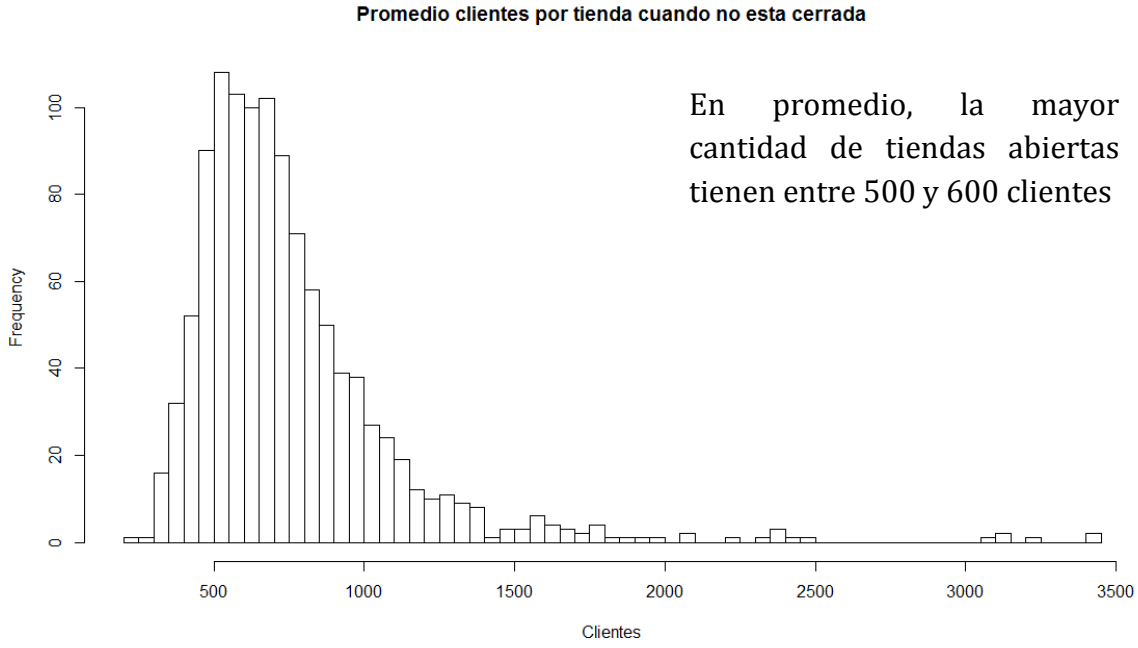
Analizamos las ventas, utilizando solo las tiendas que están abiertas.

```
##Histograma de las ventas
ventTienda <- filter(train, Sales != 0)
hist(aggregate(ventTienda$Sales, by = list(ventTienda$Store), mean)$x,
50, main = "Promedio Ventas por tienda cuando no esta cerrada",
xlab="Ventas", ylab="Tiendas")
```



En promedio, la mayor cantidad de tiendas abiertas venden entre 7,500 y 8,000 euros. Muy pocas tiendas tienen ventas mayores a 10,000 euros.

```
##Histograma de clientes
hist(train$Customers, 100, xlab="Clientes", ylab="Frecuencia",
main="Histograma de clientes")
hist(aggregate(ventTienda$Customers, by = list(ventTienda$Store),
mean)$x, 100,main = "Promedio clientes por tienda cuando no esta
cerrada", xlab="Clientes")
```

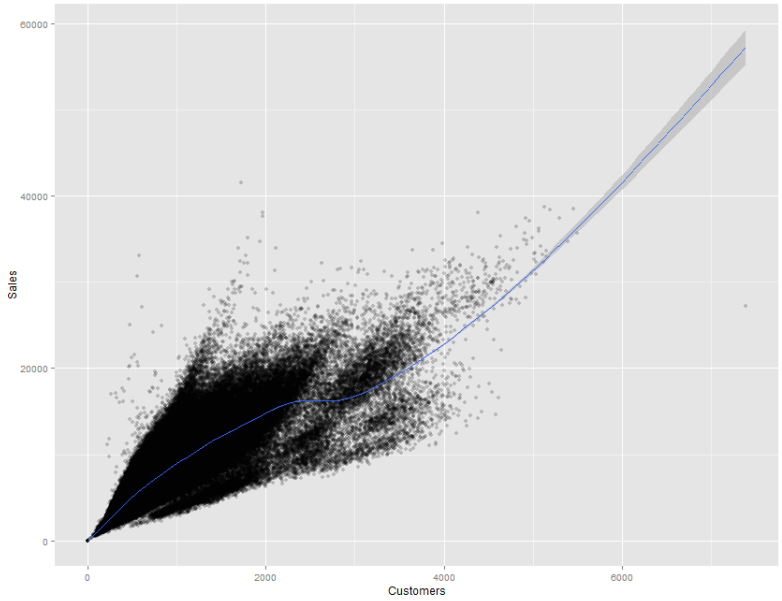


En promedio, la mayor cantidad de tiendas abiertas tienen entre 500 y 600 clientes

Las ventas están altamente correlacionadas con el número de clientes. Se obtiene una correlación del 0.999993.

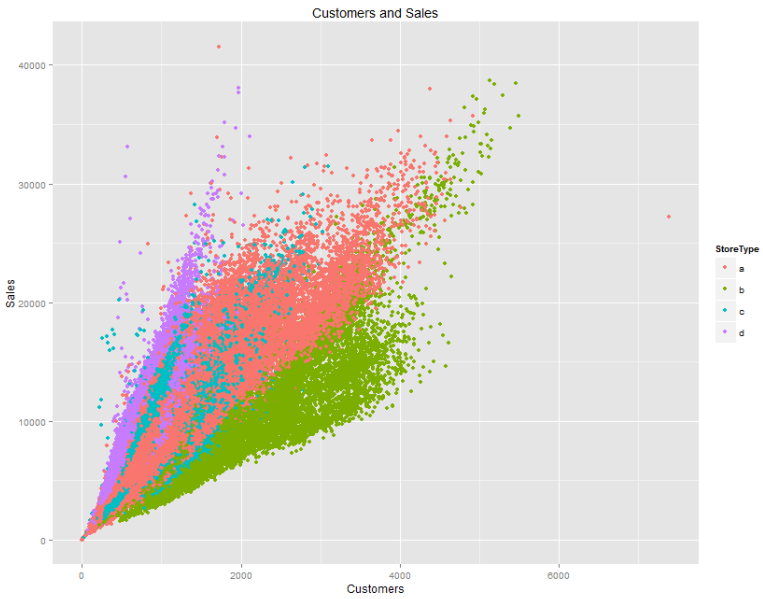
```
cor(train$Sales!=0, train$Customers!=0)
ggplot(train[train$Sales != 0 && train$Customers != 0], aes(x =
Customers, y = Sales)) + geom_point(alpha = 0.2) + geom_smooth()
```

La gráfica muestra una correlación positiva entre el número de clientes y las ventas.



Se analizan las ventas dependiendo del tipo de tienda

```
data1 <- merge(store, train, by="Store")
data1 <- filter(data1, Open == 1) #solo si las tiendas estan abiertas
ggplot(data1, aes(x = Customers, y = Sales)) + geom_point(aes(colour =
StoreType)) + labs(title = "Customers and Sales")
```



Para el mismo nivel de ventas, el tipo de tienda “d” requiere un menor número de clientes que el tipo de tienda "b". El tipo de tienda es una variable importante, aunque el tipo de tienda a y c tienden a revolverse.

Hay ocasiones en las que la tienda está cerrada y hay promociones. Si fueran ventas por internet, se justificarían las promociones, pero no.

```
table(ifelse(train$Open != 0, "Open ", "Close"), ifelse(train$Promo,
"Promo", "No promo"))
```

	No Promo	Promo
Close	161633	11184
Open	467496	376896

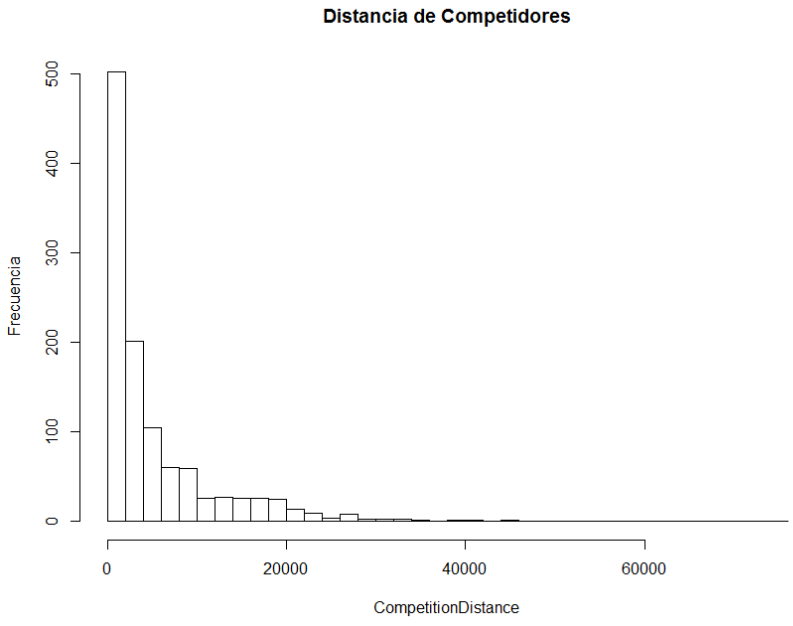
No hay ventas cuando las tiendas están cerradas; hay coherencia. Aunque hay tiendas que no vendieron a pesar de estar abiertas y tenían clientes.

```
table(ifelse(train$Open == 1, "Open", "Close"), ifelse(train$Sales > 0,
"Hay ventas", "No hay ventas"))
```

	Hay Ventas	No hay ventas
Close	0	172817
Open	844338	54

Los competidores abren sus sucursales a distancias demasiado cerca de las tiendas Rossman.

```
as.data.frame(store.imp$ximp)
hist(store.imp$ximp$CompetitionDistance, 50, xlab="CompetitionDistance",
ylab="Frecuencia", main="Distancia de Competidores")
```

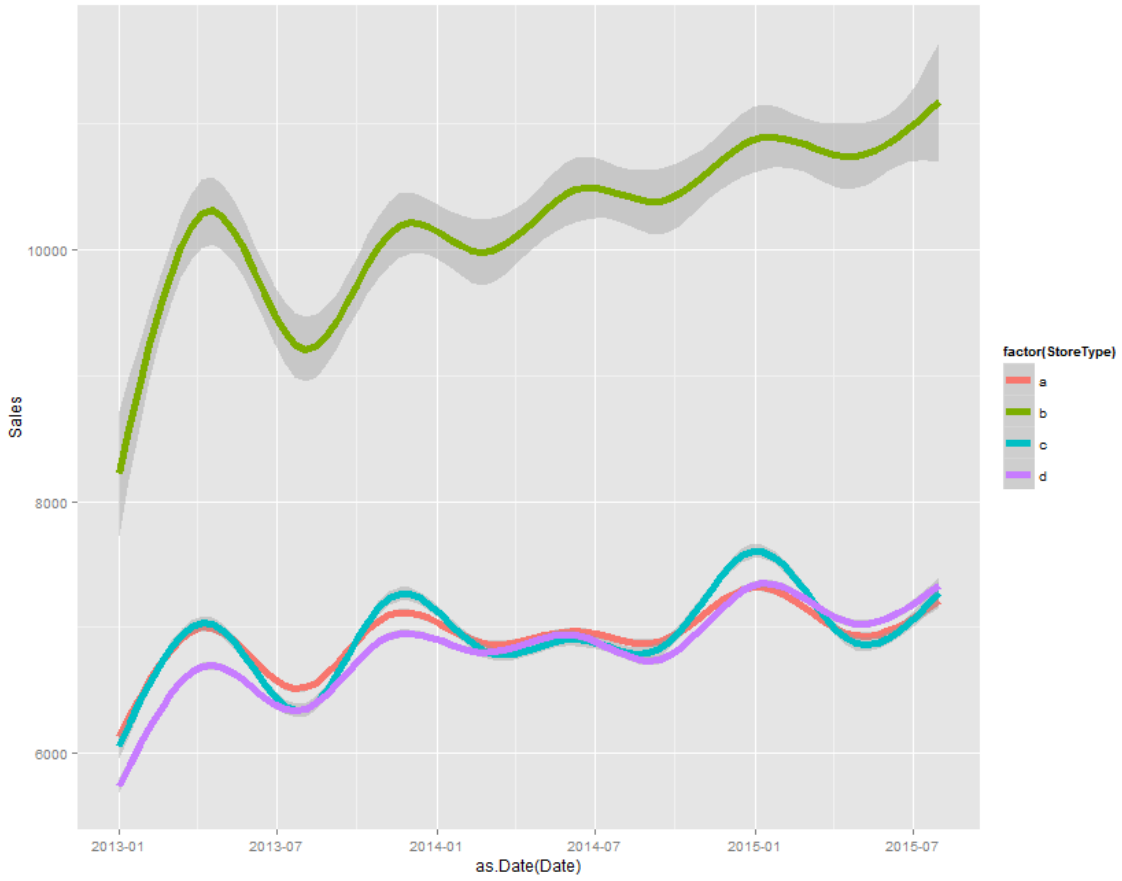


Los diferentes tipos de tiendas y tipos de surtidos implican diferentes niveles de ventas.

```
train_store <- merge(train, store.imp$ximp, by = "Store")
train_store2 <- filter(train_store, Sales != 0)
```

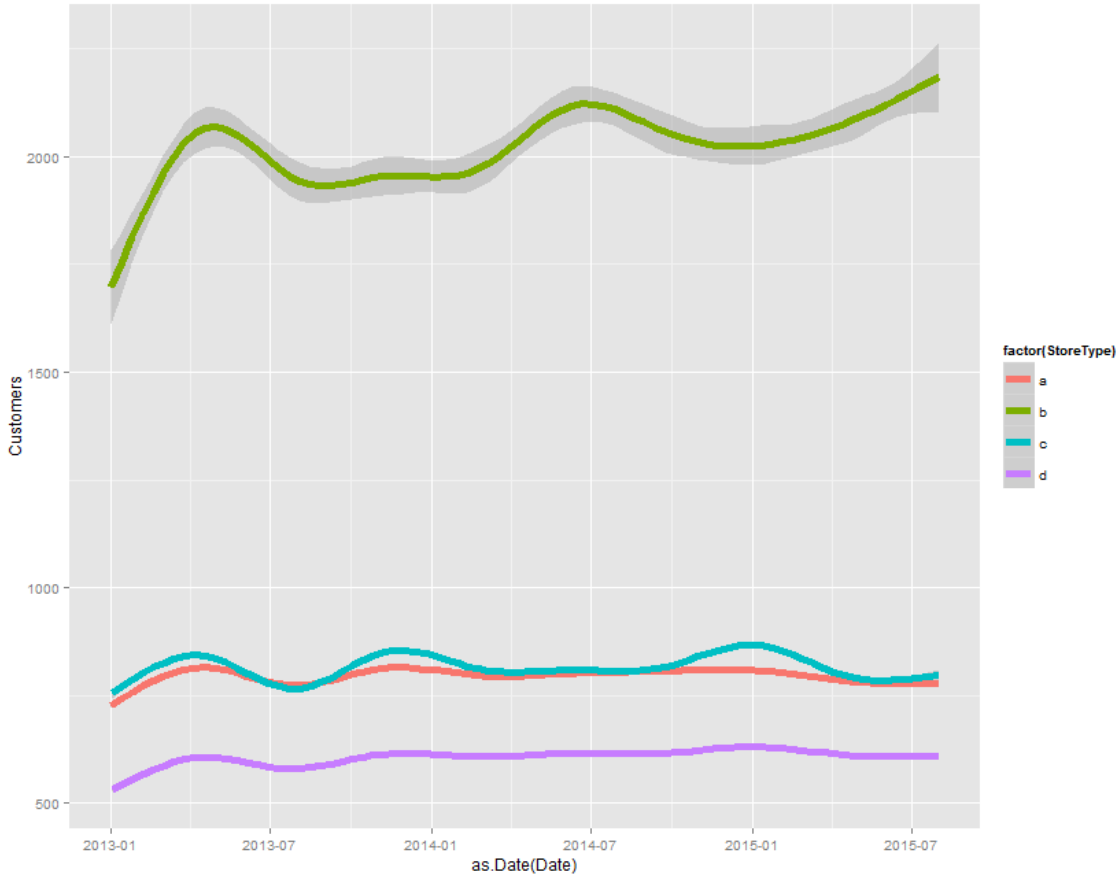
Análisis por tipo de tienda

```
#Ventas
ggplot(train_store2, aes(x = as.Date(Date), y = Sales, color =
factor(StoreType))) + geom_smooth(size = 2)
```



Las ventas han tenido una tendencia a la alza en todas las tiendas en el transcurso del tiempo; la tienda tipo “b” es la que ha tenido mayores ventas.

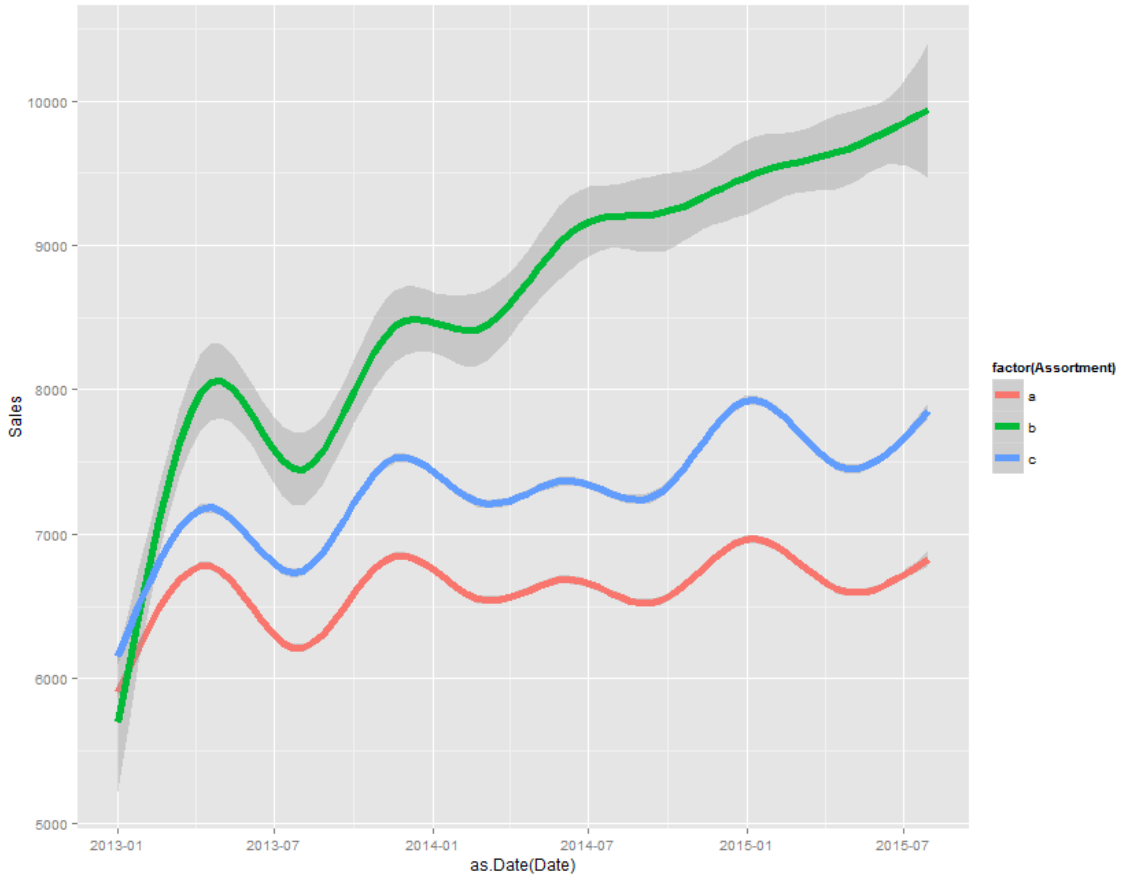
```
#Clientes
train_store3 <- filter(train_store, Customers != 0)
ggplot(train_store3, aes(x = as.Date(Date), y = Customers, color =
factor(StoreType))) + geom_smooth(size = 2)
```



En este caso los clientes de todos los tipos de tienda no han tenido una tendencia a la alza con el transcurso del tiempo. El único tipo de tienda que ha aumentado su clientela es la tipo “b”, mientras que las demás se han mantenido constantes.

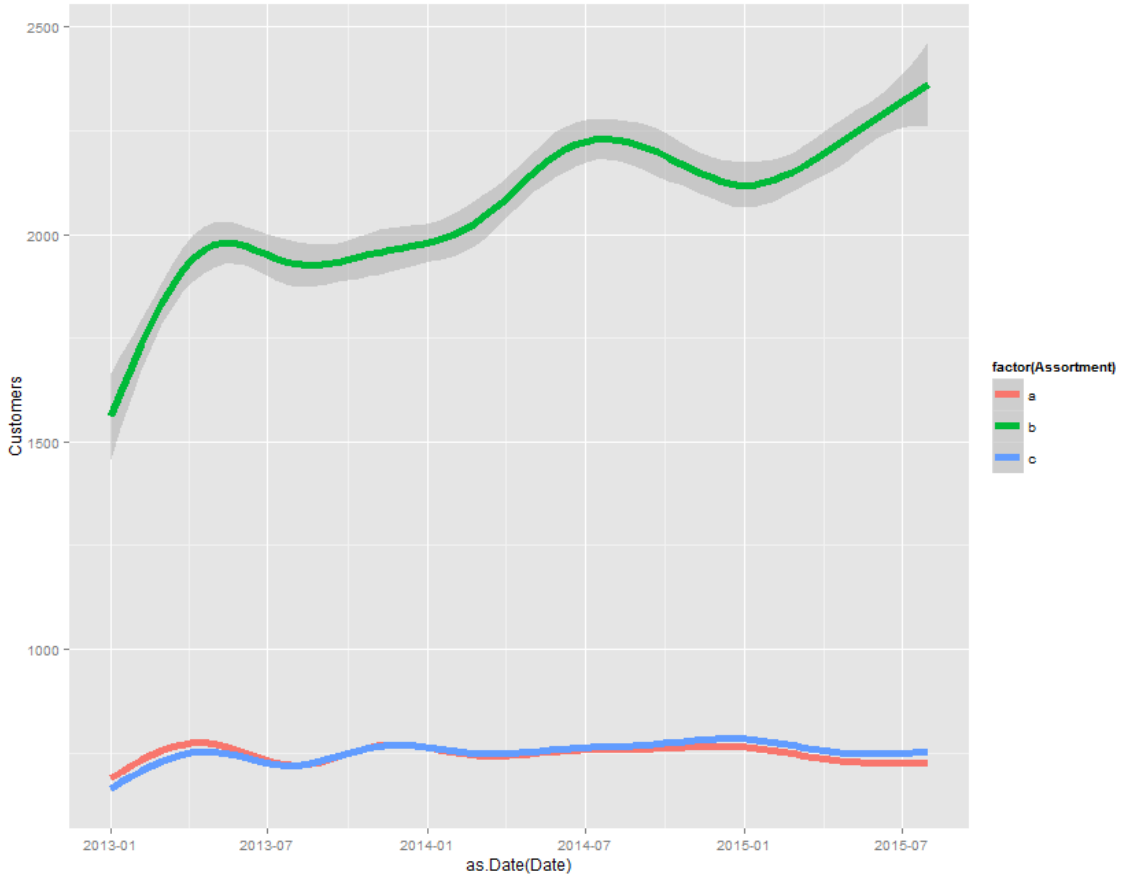
Análisis por tipo de surtido

```
#Ventas
ggplot(train_store2, aes(x = as.Date(Date), y = Sales, color =
factor(Assortment))) +
geom_smooth(size = 2)
```



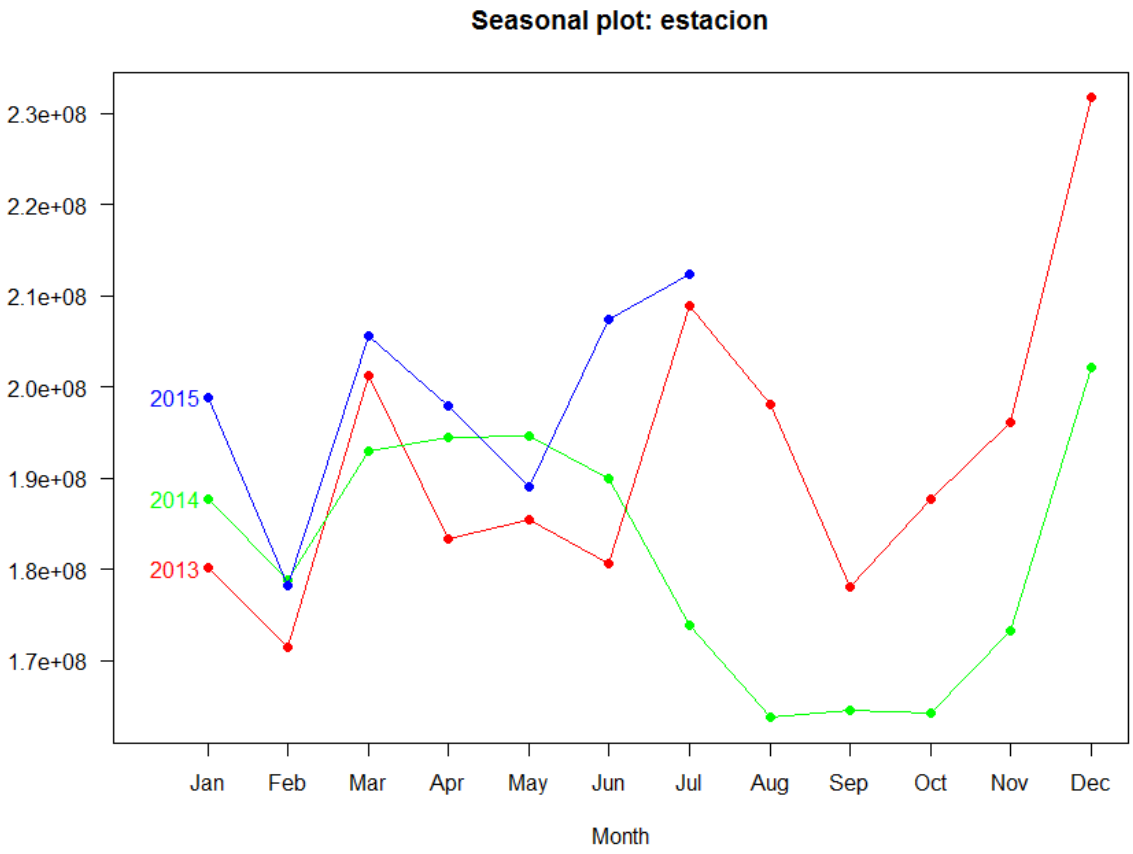
El surtido tipo “b” o “extra” es el que ha tenido un mayor aumento de ventas a través de los años. Mientras que el tipo “a/básico” y “c/extendido”, han permanecido relativamente constantes.

```
#Clientes
ggplot(train_store3, aes(x = as.Date(Date), y = Customers, color =
factor(Assortment))) +
geom_smooth(size = 2)
```



Siguiendo el comportamiento de las ventas, el tipo de surtido “b” es el que tiene más consumidores. Mientras que el surtido “b” y “c” han permanecido constantes; les falta promoción.

```
#Ventas por estación de año
train$Date<-as.Date(as.character(train$Date))
train$year<-as.factor(format(train$Date, "%Y"))
train$month<-as.factor(format(train$Date, "%m"))
ventasTiempo<-aggregate(Sales ~ ., data=train[, c("Sales", "month"
,"year")], FUN=sum)
estacion<-ts(ventasTiempo$Sales, start=2013, frequency=12)
col = rainbow(3)
seasonplot(estacion, col=col, year.labels.left = TRUE, pch=19, las=1)
```



Las ventas aumentan en Diciembre y en Marzo. Se puede suponer que en esas fechas, son los cambios de clima y mucha gente se enferma; por lo que la venta de medicinas se incrementa. Se podría esperar que igualmente para el 2015 aumenten las ventas en Diciembre y que decaigan en Septiembre.

Predicción

Para realizar la predicción, se utilizarán dos algoritmos: Random forest y Boosted Trees; la variable a predecir serán las ventas.

Se va a utilizar un data set en el que sólo se analice cuando las tiendas estén abiertas; las variables que no se van a utilizar son: Store, Open, StoreType, Assortment y PromoInterval.

Para estas predicciones se utilizó un conjunto de entrenamiento del 1% de los datos y uno para la prueba con el 99% restante. Se utilizaron 100 árboles para ambos casos y 10 predictores para el Random Forest.

Random Forest

```
#Randomforest
library(randomForest)
store.imp <- missForest(store, xtrue = store, verbose = FALSE)
store <- as.data.frame(store.imp$ximp)
data <- left_join(train, store, by = "Store")
data <- filter(data, Open == 1) #Para analizar cuando las tiendas estan
abiertas
data$Date <- NULL
data$Customers<-NULL
data$StoreType<-NULL
data$Assortment<-NULL
data$PromoInterval<-NULL
set.seed(1)
train <- sample(1:nrow(data), nrow(data)*.01) #Entrenamiento
Test <- data[-train,'Sales']
Test <- sample(1:length(Test), nrow(data)*.99) #Prueba

    rf <- randomForest(Sales ~. -Store -Sales -Open ,data=data [train
,],mtry=10,ntree=100,importance=TRUE)
    pred <- predict (rf,newdata=data[Test,], n.trees=100)
    error<-sqrt(mean(pred-Test)^2)
```

Con Random Forest el error es de 410964.5

Boosted Trees

```
store.imp <- missForest(store, xtrue = store, verbose = FALSE)
store <- as.data.frame(store.imp$ximp)
data <- left_join(train, store, by = "Store")
data <- filter(data, Open == 1) #Para analizar cuando las tiendas están
abiertas
data$Date <- NULL
set.seed(1)
train <- sample(1:nrow(data), nrow(data)*.01) #Entrenamiento
Test <- data[-train,'Sales']
Test <- sample(1:length(Test), nrow(data)*.99) #Prueba
Boost<- gbm(Sales ~. -Store -Sales -Open -StoreType -Assortment -
              PromoInterval, data = data [train ,], distribution =
"gaussian",
            n.trees = 100, interaction.depth = 4, shrinkage = 0.25)
yhatBoost <- predict(Boost, newdata=data[Test,], n.trees=100)
errorBoost<-sqrt(mean(yhatBoost-Test)^2)
```

Con Boosted Trees el error es de 411014.6

Conclusiones Predicciones

- El método predictivo con menor error fue el de Random Forest con 100 árboles y 10 predictores.
- No se usaron las variables de StoreType, Assortment, Date y PromoInterval.
- No se utilizó la variable Customers al estar altamente ligada con las ventas
- Se podría mejorar las predicciones usando otro número de árboles y predictores, pero al ser una base de datos de gran tamaño, el costo computacional aumenta, así como el tiempo de procesamiento. Es por esto que se utilizó un porcentaje de datos para el entrenamiento reducido. Una solución sería usar computación en paralelo.

Unsupervised Learning

Para este tipo de aprendizaje, no interesa la predicción, ya que no se tiene una variable de respuesta Y. El objetivo es descubrir características interesantes en las variables independientes. Se verá en este data set el aprendizaje no supervisado: principal components analysis (una herramienta para visualizar datos antes de que las técnicas supervisadas sean aplicadas.

Principal Component Analysis (PCA)

Es una forma de simplificar un conjunto de datos complejo. Ayuda a exponer las fuentes subyacentes de la variación en los datos.

El data set sólo debe contener variables numéricas. Si hay variables no-numéricas, se deben excluir. Se puede ejecutar un PCA con la función "princomp"; esta función muestra las desviaciones estándar de los componentes.

```
data$Date <- NULL
data$Assortment <- NULL
data$PromoInterval <- NULL
data$StoreType <- NULL
data$Open <- NULL
data[,2]<- as.numeric(data[,2])
data[,3]<- as.numeric(data[,3])
data[,4]<- as.numeric(data[,4])
data[,6]<- as.numeric(data[,6])
data[,7]<- as.numeric(data[,7])
data[,8]<- as.numeric(data[,8])
pca <- princomp(data)
```

La función "summary" muestra la varianza explicada por cada componente.

```
summary(pca)
```

```
Importance of components:
                             Comp.1
Standard deviation     7801.1380107
Proportion of Variance    0.8603098
Cumulative Proportion     0.8603098
                             Comp.2
Standard deviation     3119.1421139
Proportion of Variance    0.1375338
Cumulative Proportion     0.9978436
                             Comp.3
Standard deviation     3.218866e+02
Proportion of Variance 1.464687e-03
Cumulative Proportion  9.993083e-01
```

Ahora lo que queremos saber es cuánto de cada nueva variable tiene el poder de explicar la información que la variable original tiene.

Se puede observar que el 86.03% de la variación en el data set está explicada por el primer componente, y que el 99.78% está explicada por los primeros dos componentes.

Para ver las cargas y el score de cada componente usamos:

```
pca$loadings
pca$scores
```

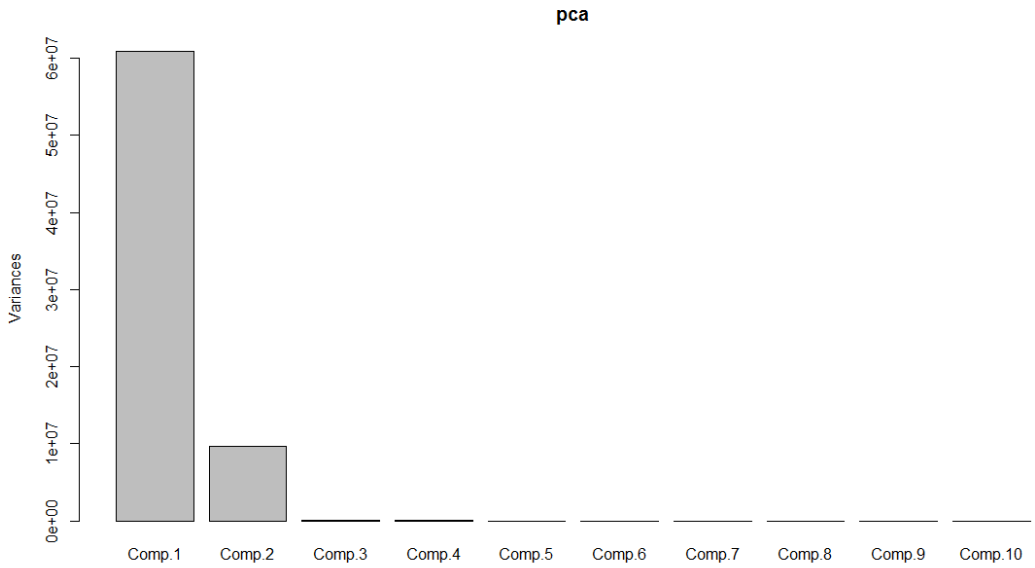
```
Loadings:
                        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
Store                                  0.998
DayOfWeek                                                                 0.997
Sales                           -0.994         0.106
Customers                       -0.106        -0.993
Promo
StateHoliday
SchoolHoliday
CompetitionDistance        1.000
CompetitionOpenSinceMonth                                        -0.998
CompetitionOpenSinceYear                                  -0.997
Promo2
Promo2SinceWeek                                    -0.997
Promo2SinceYear
```

El PCA encontró 13 variables nuevas que pueden explicar la misma información que las 13 variables originales (Store, DayOfWeek, Sales, Customers, Promo, StateHoliday, SchoolHoliday, CompetitionDistance, CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo2, Promo2SinceWeek, Promo2SinceYear) y se calculan de la siguiente forma:

Comp.2=-0.994*Sales -0.106*Customers

La gráfica “scree plot” muestra la proporción de la variación total en un data set que es explicada por cada componente en el PCA. Ayuda a identificar cuántos de los componentes son necesitados para resumir los datos.

```
screeplot(pca)
```



Se puede observar en la gráfica que la cantidad de la variación explicada decae drásticamente después del primer componente. Esto sugiere que con solo un componente es suficiente para resumir los datos.

Conclusiones

De las promociones existentes, sólo el 50% de éstas se encuentran continua y consecutivamente en tiendas; si se quiere aumentar las ventas, se debería de poner atención en que estas promociones estén durante más tiempo.

Las ventas se comportan de forma similar durante los 7 días de la semana; esto tiene coherencia, ya que las enfermedades (y por consecuente la venta de medicinas), no tienen un patrón semanal, éstas siempre existen.

Hay ocasiones en las que existen promociones mientras las tiendas están cerradas; al haber una alta correlación entre el número de clientes y las ventas, se recomienda prestar atención a que las promociones se implementen mientras las tiendas están abiertas y haya clientes.

Se recomienda observar la logística de las tiendas tipo “b”, ya que ésta es la única que ha aumentado tanto sus ventas como su clientela, para poder implementarlo en los otros tipos de tienda. Se sugiere hacer lo mismo con el tipo de producto “b”.