# Proyecto de Aplicación Profesional

## Modelos de predición en empresas y gobierno mediante aprendizaje estadístico.

Profesor: Dr. Páblo Dávalos de la Peña

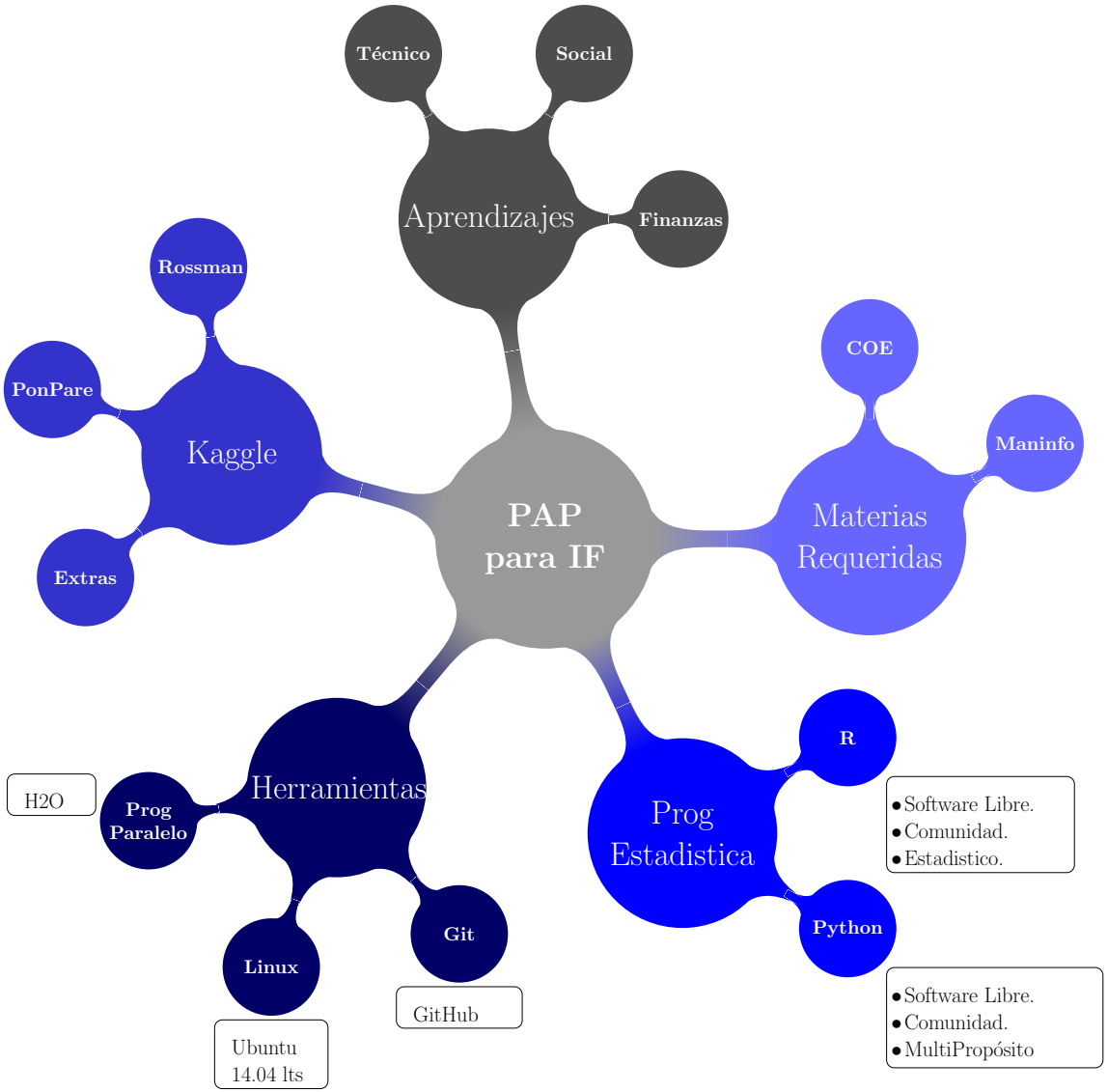
**Juan Francisco Muñoz Elguezábal**
Alumno Ingeniería Financiera
IF149833@iteso.mx

**Rodrigo Ledesma Elorriaga**
Alumno Ingeniería Financiera
IF149833@iteso.mx

**Zurisadai Velazquez Manzanero**
Alumno Ingeniería Financiera
IF149833@iteso.mx

3.Diciembre.2015

# Mapa de contenidos



# Un Proyecto de Aplicación Profesional

- **Creditos Aprobados:** 70 % antes de inscribirlo.
- **Materia Requerida 1:** Manejo de informacón y datos numéricos.
- **Materia Requerida 2:** Comunicación oral y escríta.
- **Especiales:** Pre-requisitos de Carrera y Pre-requisitos de Proyecto.

*Fuente: Video Tutorial PAP - canal oficial Pap Iteso*
https://www.youtube.com/watch?v=LFYJOpu97m8

# Ciencia de datos en PAP

**Colaboración KUESKI**: *En espera*

Seguridad de información comprometida.

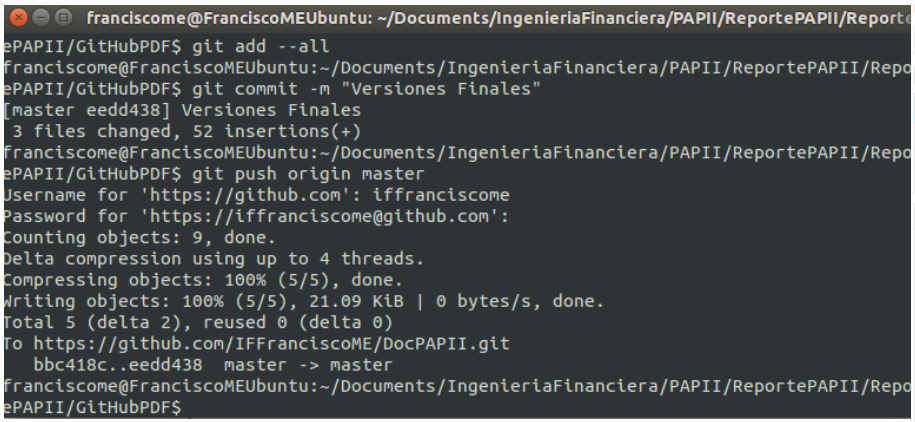
**Colaboración ITESO**: *En espera*

Seguridad de informaciý beneficio contrario a la naturaleza del PAP.

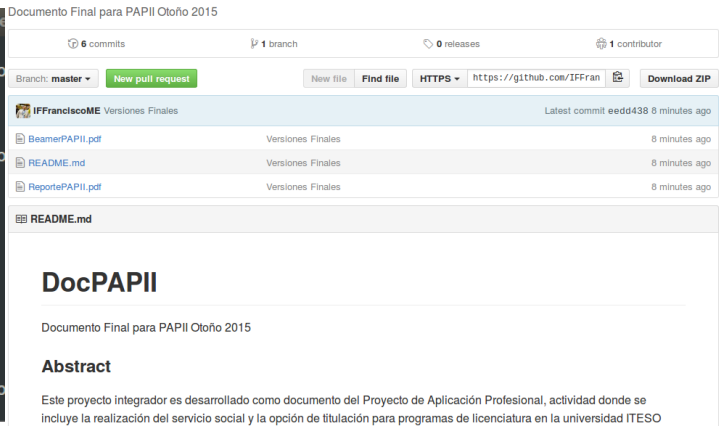
**Aprendizaje competitivo KAGGLE**: *Utilizado*

Un sitio donde se publican retos competitivos, para recompensa y/o incluso reclutamiento a través de análisis de datos para clasificación y predicción. Frecuentado por científicos de datos profesionales y por organizaciones como *Netflix*, *Airbnb*, *Walmart*, *Hillary Clinton's Emails*, entre otros.

# Herramientas Computacionales I: GITHUB en LINUX




(a) GitHub en LINUX




(b) GitHub en WEB

Set up Git on your machine if you haven't already.

```
$ mkdir /path/to/your/project
$ cd /path/to/your/project
$ git init
$ git remote add origin https://IFFranciscoME@bitbucket.org/IFFranciscoME/fds.git
```

**Create your first file, commit, and push**

```
$ echo "Francisco Muñoz Elguezabal" >> contributors.txt
$ git add contributors.txt
$ git commit -m 'Initial commit with contributors'
$ git push -u origin master
```

# Herramientas Computacionales I: Lenguajes, IDEs, S.O.

Se utilizó una computadora Toshiba U940 i5 8gb, 1.7Mhz. Las herramientas computacionales instaladas en esta y que fueron utilizadas durante el transcurso del proyecto fueron las siguientes:

## Lenguajes de Programación

Versión de *R*:
Versión de *Python*:

## IDEs (Integrated Development Environment)

Versión *RStudio*:
Versión *Pycharm*:

## Sistema Operativo:

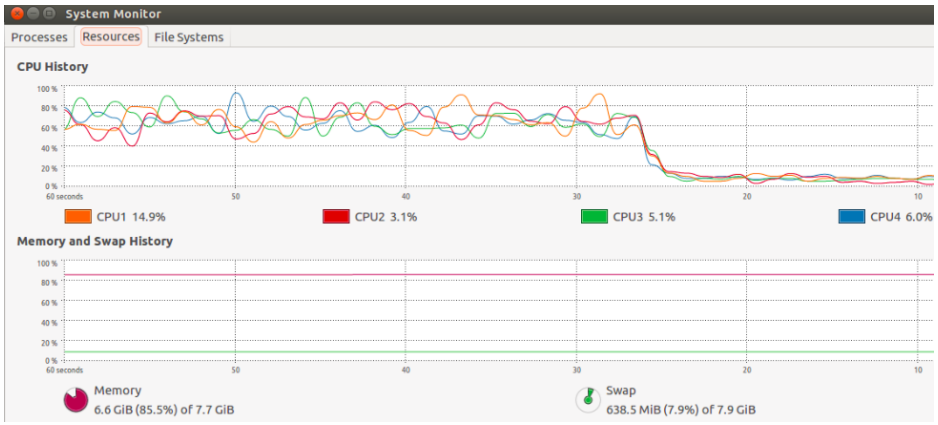
Linux: Ubuntu 14.04 LTS

# Herramientas Computacionales II: Librerias Especiales

- **R**: *data.table* Paquetería para tratamiento de *BigData*
- **Software**: Cluster de procesamiento en paralelo Online/Local *H2O*

```
> cat("fread se tarda, en segundos:")
fread se tarda, en segundos:
> Final1-Inicial1
[1] 6.969136
>
> cat("read.csv se tarda, en segundos:")
read.csv se tarda, en segundos:
> Final2-Inicial2
[1] 117.5787
>
```

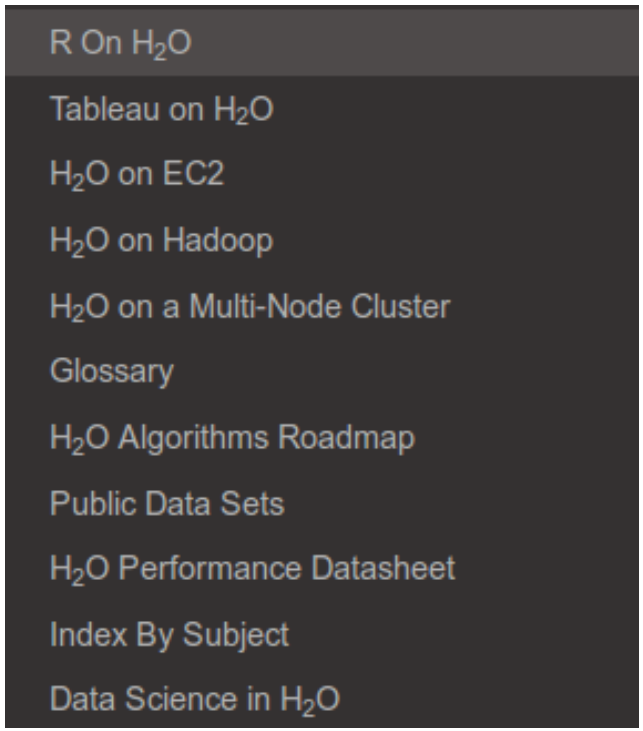
(a) fread VS readcsv



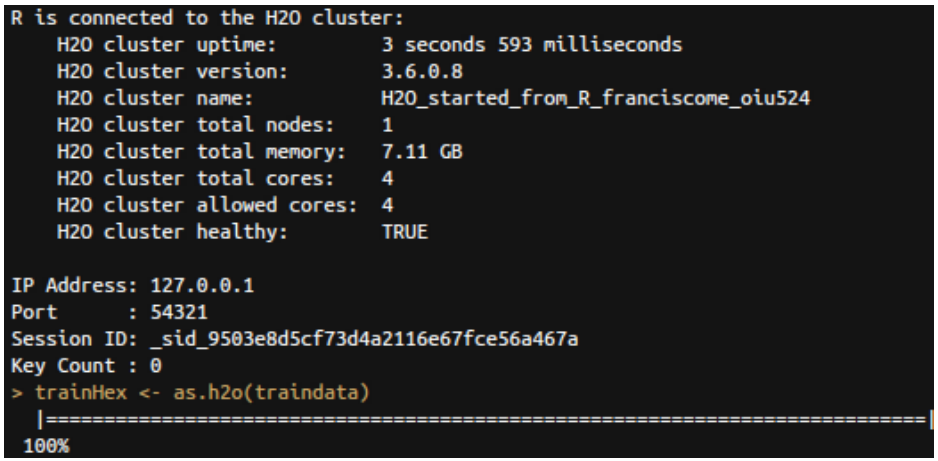

(b) 4 Procesadores y 90 % Ram

Figura : Recursos en Lectura y Procesamiento de datos

# Herramientas Computacionales III: Librerias Especiales




(a) H2O Integraciones




(b) En máquina local

Figura : Recursos en Lectura y Procesamiento de datos

# APLICACIÓN: Ciencia de Datos para Series de Tiempo I

Predicción para series de tiempo financieras mediante la implementación y clasificación de estudios de análisis técnico.

**Third-party Samples**

- OANDA Ruby Wrapper - submitted by nukeproof
- Matlab REST Wrapper - submitted by tradeasystems
- OTest (C++ on Windows) - submitted by StevenABrown
- OANDA For Go (Go programming language) - submitted by santegoeds
- OANDA Adapter - Node.js adapter for OANDA's REST and streaming API - submitted by Cloud9Trader
- OANDAWrap - Php interface for Oanda API - submitted by tavurth
- ARGO - Argo is an open source trading platform, connecting directly with OANDA through the powerful submitted by albertosantini
- pyoanda - Python library that wraps Oanda API. Built on top of requests, it's easy to use and makes s
- morgentau - Interface to the oanda REST API using ruby by morgentau
- Scalanda - Scala/Akka wrapper for Oanda REST and Stream API - submitted by msilb
- AKka-trading - Scala Backtesting + Oanda REST API Trading Framework built on top of Akka/Spray - s
- cloanda - A closure wrapper for OANDA REST API - submitted by yellowbean
- oanda-rest-java - OANDA REST api wrapper for java - submitted by rabun
- oanda-rest-cs - OANDA REST api wrapper for C# - submitted by rabun
- ROandaAPI - OANDA REST API wrapper for R - submitted by FranciscoME

(a) API en R

**Oanda Corporation**

Foreign exchange company · oanda.com

OANDA is a Canadian-based foreign exchange company providing currency conversion, online retail foreign exchange trading, online foreign currency transfers, and forex information. Wikipedia

**Country of origin:** United States of America

**CEO:** Edmond Eger III

**Headquarters:** New York City, New York, United States

(b) Broker Online de Forex

Figura : Recursos en Lectura y Procesamiento de datos

# APLICACIÓN: Ciencia de Datos para Series de Tiempo II

Código básico para utilizar la API de *ROanda* y obtener información del mercado en tiempo real, histórica, grautita y de 120 instrumentos financieros.

```
GitHub <- "https://raw.githubusercontent.com/IFFranciscoME"
D1 <- paste(GitHub,"/ROandaAPI/master/ROandaAPI.R", sep="")
downloader::source_url(D1,prompt=FALSE,quiet=TRUE)
AccountID   <- 1438853    # ID de cuenta (Ver Manual ROandaAPI)
TimeAlign   <- "America%2FMexico_City" # Uso horario
Token       <- "c567fab3522f33fda6a91dbfee0522f6-cdbba372874e6e69e4694f050f890273"
Ini <- '2010-01-01'
Fin <- '2015-11-01'

DayAlign    <- 14         # Hora para considerar el cierre diario
AccountType <- "practice" # Tipo de cuenta.
Granularity <- "D"        # Frecuencia de muestre de precio.
TInstrument <- "EUR_USD"  # Instrumento Financiero a utilizar.

ListaInst <- data.frame(InstrumentsList(AccountType,Token,AccountID))[,c(1,3)]
Precios   <- HisPrices(AccountType,Granularity,DayAlign,TimeAlign,Token,
                       TInstrument,Ini,Fin)
Precios$TimeStamp <- as.POSIXct(Precios$TimeStamp, origin = "1970-01-01")
Precios$TimeStamp <- as.Date(Precios$TimeStamp)
ACTIVO <- xts(Precios[,2:5], order.by = Precios[,1])
```

# APLICACIÓN: Ciencia de Datos para Series de Tiempo III

Realizar ciencia de datos para observaciones no secuenciales o distintas de series de tiempo es lo que aprendimos, pero también a trasladar técnicas y metodologías al caso de series de tiempo financieras, como el caso de realizar los siguiente:

- CrossValidation K-Fold para *series de tiempo* ?

```
Fold <- trunc(length(Tdata.train[,1])/10)
Train1 <- Tdata.train[1:Fold,]
Train2 <- Tdata.train[1:(Fold*2),]
Train3 <- Tdata.train[1:(Fold*3),]
Train4 <- Tdata.train[1:(Fold*4),]
Train5 <- Tdata.train[1:(Fold*5),]
Train6 <- Tdata.train[1:(Fold*6),]
Train7 <- Tdata.train[1:(Fold*7),]
Train8 <- Tdata.train[1:(Fold*8),]
Train9 <- Tdata.train[1:(Fold*9),]
Train10 <- Tdata.train[1:(Fold*10),]
```

# Análisis Técnico y Ciencia de Datos.

```
223 <<Ini8, eval = TRUE, echo = TRUE, include = TRUE, result
224 myATR <- function(x) ATR(HLC(x))[,'atr']
225 mySMI <- function(x) SMI(HLC(x))[,'SMI']
226 myADX <- function(x) ADX(HLC(x))[,'ADX']
227 myBB  <- function(x) BBands(HLC(x))[,'pctB']
228 myMACD  <- function(x) MACD(Cl(x))[,2]
229 myEMA10 <- function(x) EMA(Cl(x),n=10)[,1]
230 myEMA20 <- function(x) EMA(Cl(x),n=20)[,1]
231 myEMA30 <- function(x) EMA(Cl(x),n=30)[,1]
232 myEMA40 <- function(x) EMA(Cl(x),n=40)[,1]
233 myEMA50 <- function(x) EMA(Cl(x),n=50)[,1]
234 myEMA60 <- function(x) EMA(Cl(x),n=60)[,1]
235 myEMA70 <- function(x) EMA(Cl(x),n=70)[,1]
236 mySAR   <- function(x) SAR(x[,c('High','Close')]) [,1]
237 @
```

(a) Funciones Genéricas

```
242 <<Ini9, eval = TRUE, echo = TRUE, include = TRUE, results='hide'>>=
243 data.model <- specifyModel(Delt(Cl(ACTIVO)) ~
244 myATR(ACTIVO) + mySMI(ACTIVO) + myADX(ACTIVO) +
245 myBB(ACTIVO)  + myMACD(ACTIVO) + myEMA10(ACTIVO) +
246 myEMA20(ACTIVO) + myEMA30(ACTIVO) + myEMA40(ACTIVO) +
247 myEMA50(ACTIVO) + myEMA60(ACTIVO) + myEMA70(ACTIVO) +
248 EMA(Delt(Cl(ACTIVO))) + RSI(Cl(ACTIVO)) + mySAR(ACTIVO) +
249 runMean(Cl(ACTIVO)) + runSD(Cl(ACTIVO)))
250 @
```

(b) Modelo General

Figura : Análisis técnico y Machine Learning I

# Análisis exploratorio

Se establecieron los criterios de toma de postura de la siguiente manera:

- $x >= -0.005$ - "hold"
- $x <= 0.005$ - "hold"
- $x > 0.005$ - "buy"
- $x < -0.005$ - "sell"

```
<<Ini10, eval = TRUE, echo = TRUE, include = TRUE, results='hide'>
signals <- function(x) {
if(x >= -0.005 && x <= 0.005) {resultado <- "hold"} else
if(x >   0.005) {resultado <- "buy" } else
if(x <  -0.005) {resultado <- "sell"}
resultado }

Clase <- sapply(Tdata.train$Delt.Cl.ACTIVO, signals)
traindata <- cbind(Tdata.train,Clase)
traindata$Delt.Cl.ACTIVO <- NULL

Clase <- sapply(Tdata.eval$Delt.Cl.ACTIVO, signals)
testdata  <- cbind(Tdata.eval, Clase)
testdata$Delt.Cl.ACTIVO  <- NULL
@
```

(a) Consideración para señales

```
<<Ini9a, eval = TRUE, echo = TRUE, include = TRUE, results
Tdata.train <- as.data.frame(modelData(data.model,
data.window=c('2010-01-01','2015-01-01')))

Tdata.eval <- na.omit(as.data.frame(modelData(data.model,
data.window=c('2015-01-02','2015-11-01'))))
@
```

(b) Evaluación y Prueba

# Modelos de Machine Learning

Se ajustaron dos modelos, el primero fue el visto en el PAP, *Random Forest*, utilizando la librería de R con el mismo nombre. El segundo a manera de comparación fue una red neuronal del tipo perceptrón multicapa.

```
NTREES   <- 10000
NODESIZE <- 10

TiempoSimple1 <- Sys.time()
ModeloRF <- randomForest(Clase~., data = traindata, nodesize = NODESIZE,
                         importance = FALSE, mtry = 4, ntree = NTREES)
TiempoSimple2 <- Sys.time()

TiempoSimple2 - TiempoSimple1

TablaRF  <- table(actual=testdata$Clase,
                  predicted=predict(ModeloRF,newdata=testdata,type="class"))

AciertosRF <- round((TablaRF[1,1]+TablaRF[2,2]+TablaRF[3,3])/length(testdata[,1]),2)

set.seed(134)
ModeloNN <- nnet(Clase~., traindata, size = 3, rang = 0.1,
                 decay = 0.001, maxit = 3000, trace="F")
TablaNN  <- table(actual=testdata$Clase,
                  predicted=predict(ModeloNN,newdata=testdata,type="class"))
AciertosNN <- round(TablaNN[1,1]+TablaNN[2,2]+TablaNN[3,3])/length(testdata[,1]),2)
```

(a) Red Neuronal y Random Forest

```
> TablaNN
      predicted
actual buy hold sell
  buy   32   13    1
  hold   9  133   14
  sell   1   17   39

> AciertosNN
[1] 0.79

> TablaRF
      predicted
actual buy hold sell
  buy   37    9    0
  hold  52   90   14
  sell  14   15   28

> AciertosRF
[1] 0.6
```

(b) Tablas de resultados

# Kaggle *The Home of Data Science*

- PonPare - Prediccion de cupones a comprar por clientes.
- Rossman - Prediccion de ventas de farmacias en Alemania.
- IMDB - Análisis de sentimiento en reseñas de peliculas.
- Bike Sharing - Predicción de demanda de bicicletas.
- Otros...

# Caso Rossman

ROSSMANN es una cadena 3000 farmacias en 7 países europeos. Actualmente Rossmann tiene la tarea de predecir la cantidad de ventas diarias en un máximo de seis semanas con anticipación, por lo que le ha pedido ayuda a la página de Kaggle para un sin número de participantes analicen y pronostique el nivel de ventas de las farmacias.

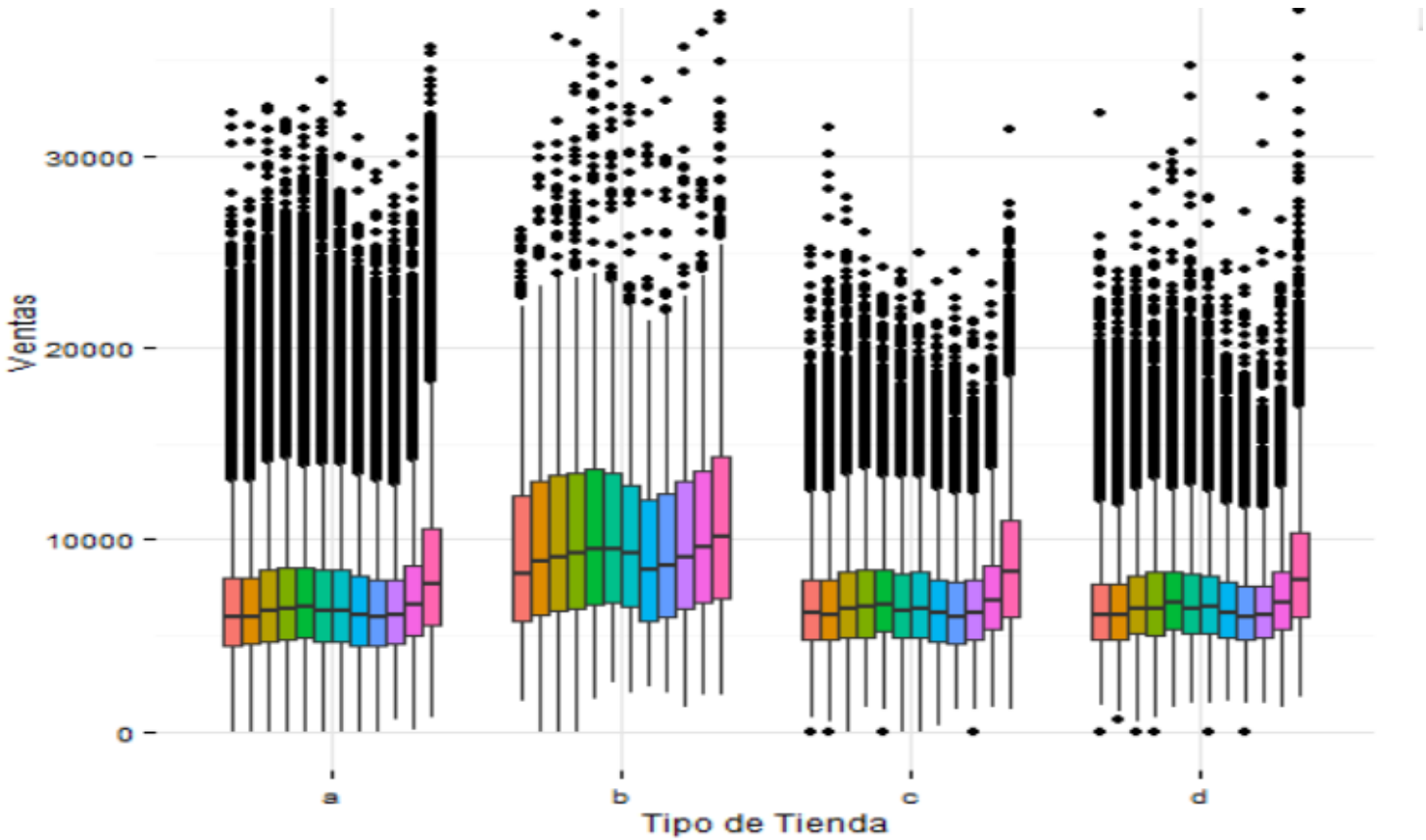
# Caso Rossman

De acuerdo a los datos que se irán obteniendo a través de la metodología se organizaran de la siguiente manera:

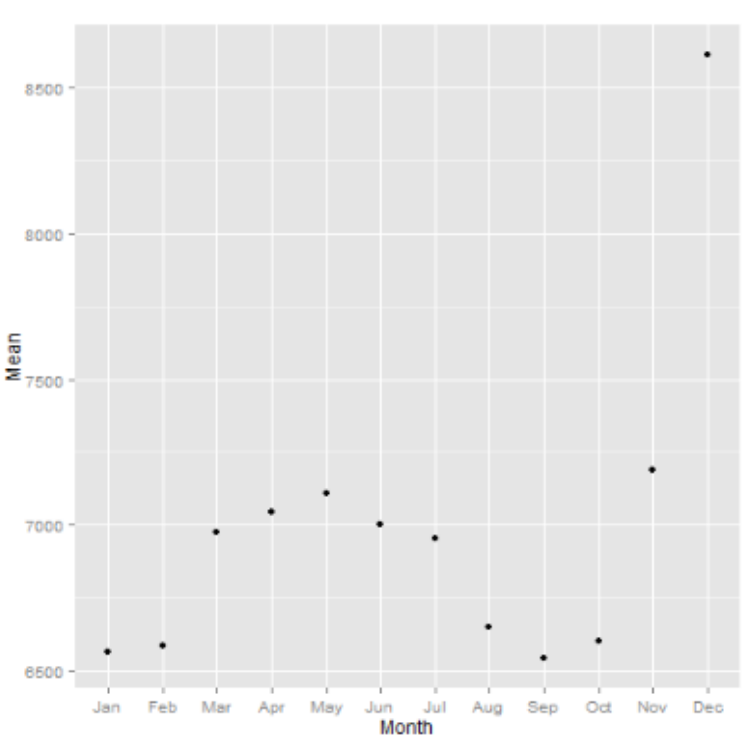
1. Descripción de variables
2. Exploración de datos.
3. Procedimiento de estimación (Logistic Regression, Tree, Random Forest
4. Bagging, Boosting)
5. Conclusiones de Análisis exploratorio
6. Predicciones.

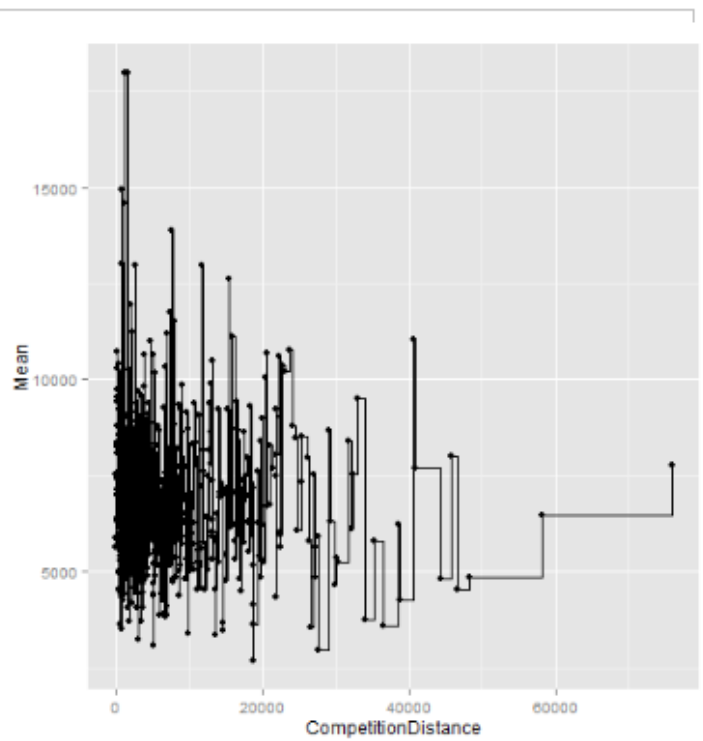
Variables: Id(tienda, Fecha), Costumer, SchoolHoliday, Promo, Store, Open, Assortment (a,b,c), Promo2, Sales, StoreType, Competition Distance, Promo Interval

# Caso Rossman



# Caso Rossman

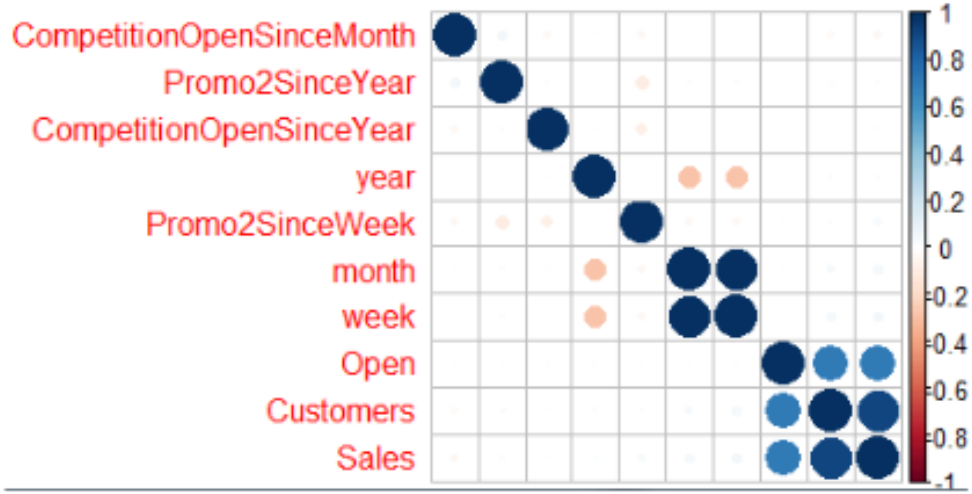


# Conclusiones Caso Rossman

Se observó claramente que el tipo de tienda es una variable que aunque parecida entre ellas, existen diferencias que pueden ser cruciales para que los algoritmos aprendan sobre ella y hagan una predicción más certera.
También se vio que las fechas en meses también son variables que ayudan señalar el nivel de ventas que tendrá por temporada.
Uno de los puntos que también fue interesante fue el hecho de que mientras más cerca haya un competidor, el nivel de ventas crece o por lo menos hay una concentración.

Por lo que puedo inferir sobre ello es que tener tiendas alrededor de lo mismo le es más fácil al cliente porque sabe donde estan toda las farmacias.




**Resultados de RMSPE**:

- Regresión Lineal: 2735.217
- Tree: 2823.370
- Random Forest: 2488.109
- Bosting: 2196.695

# Preguntas ?