Importación de datos desde mysql a R

```
library(RMySQL)
library(DBI)
con_sql <- dbConnect(MySQL(),user="gabriel",password="XXX",dbname="proyecto",host="
127.0.0.1",port=3306)
```

Luego de que se conecto exitosamente comence a hacer las queries para importarlas a un dataframe

```
dbListTables(con_sql)
dbListFields(con_sql,"cb_ipos")
ipos <- dbSendQuery(con_sql,"select * from cb_ipos")
relaciones <- dbSendQuery(con_sql,"select * from cb_relationships")
acquisitions <- dbSendQuery(con_sql,"select * from cb_acquisitions")
data_acquisitions <- fetch(acquisitions,n=-1)
data_relaciones <- fetch(relaciones, n = -1)
data_ipos <- fetch(ipos,n=-1)
degrees <-dbSendQuery(con_sql,"select * from cb_degrees")
data_degrees <- fetch(degrees, n = -1)
funding_rounds <- dbSendQuery(con_sql,"select * from cb_funding_rounds")
data_funding_rounds <- fetch(funding_rounds,n = -1)
funds <- dbSendQuery(con_sql,"select * from cb_funds")
data_funds <- fetch(funds, n = -1)
investments <- dbSendQuery(con_sql,"select * from cb_investments")
data_investments <-fetch(investments, n = -1)
milestones <- dbSendQuery(con_sql,"select * from cb_milestones")
data_milestones <- fetch(milestones, n = - 1)
objects1 <- dbSendQuery(con_sql,"select * from cb_objects")
data_objects <- fetch(objects1, n = - 1)
offices <- dbSendQuery(con_sql,"select * from cb_offices")
data_offices <- fetch(offices, n = - 1)
```

Para poder trabajar los archivos en python se escribio las tablas en csv con un bucle for

```
nombres <- list(data_acquisitions=data_acquisitions,data_degrees=data_degrees,data_
funding_rounds=data_funding_rounds, data_funds=data_funds,
                data_investments=data_investments,data_ipos=data_ipos,data_mileston
es=data_milestones,data_objects=data_objects, data_offices=data_offices,
                data_relaciones=data_relaciones)

for (i in 1:10) {
  write.csv(nombres[[i]], file= paste(names(nombres)[i],".csv"))

}
```

Limpieza de datos

Limpieza tabla degrees

Para trabajar la tabla degrees se decidio que ibamos a categorizar las universidades por tipos. Las primeras 50 universidades seran categoria 1 y el resto de universidades seran categoria 2 Para ello encontre un dataset de las 50 principales universidades de kaggle

Importe las universidades y filtre solamente las 50 primeras

```
library(readr)
library(stringr)
library(dplyr)
universities <- read_csv("~/tfm/cwurData.csv")
universities_top <- universities[1:50,]
head(universities_top)
```

Realice limpieza de datos, utilice la funcion str_splt_fixed para deja solo el nombre de la universidad quitando primero caracteres despues de la coma

```
limpieza <- str_split_fixed(universities_top$institution,",",2)
universities_top$instituion2 <- limpieza[,1]
```

En esta linea quite todos los caracteres que venian despues del “at”

```
limpieza2 <- str_split_fixed(universities_top$instituion2,"at",2)
universities_top$instituion2 <- limpieza2[,1]
```

por ultimo transforme Harvard Business School a Harvard

```
universities_top$instituion2[1]<- "Harvard"
```

En la tabla degrees cree una columna llamada “category”. Con esta linea primero se creo un vector de las 50 universidades con un “|” que significa o.
Grepl compara este vector de universidades y devuelve true si encuentra una coincidencia. Con el ifelse evalua si es TRUE y le pone un 1 y 2 si es FALSE

```
data_degrees$category <- ifelse(grepl(paste(universities_top$instituion2,collapse =
"|"),
                                      data_degrees$institution)==TRUE,1,2)


universties_category <- summarise(group_by(data_degrees,object_id),
                                  category_university = min(category))
```

Cada persona miembro de una empresa tiene un codigo de personay el codigo de su empresa asi que cree una tabla con el id de la persona y el id de la persona

```
cruce_empresa_persona <- data_relaciones[,3:4]
cruce_empresa_persona$categoria_degree <-
  universties_category$category_university[match(cruce_empresa_persona$person_objec
t_id,universties_category$object_id)]
```

Luego cree una tabla en donde este agrupado por empresa la categoria de la universidad. En una empresa podian existir varias personas y por ende varias categorias de universidad. Con la funcion summarize me quedo con la minima categoria es decir 1. Es decir si hay una persona en la empresa que ha estudiado en una universidad categoria 2 y otra persona que haya estudiado en la categoria 1 la formula escoge a la persona de la categoria 1

```
degree_empresa <- summarise(group_by(cruce_empresa_persona,relationship_object_id),
categoria = min(categoria_degree))
```

Los que esten NA los pongo como categoria 2

```
degree_empresa$categoria <- ifelse(is.na(degree_empresa$categoria)=="TRUE",2,degree
_empresa$categoria)

write.csv(degree_empresa, file= "degree_empresa.csv")
```

Limpieza Data objects

La tabla objects es donde se encontraba la columna status que nos va a servir para correr los algoritmos. Es en esta tabla que realice todos los cambios y se fueron adjuntando las variables para crear la tabla pra los algoritmos

Para poder trabajar con la tabla de las empresas filtre el entity type a Company

```
data_objects_company <- filter(data_objects,entity_type == "Company")
```

Luego elimine las columnas que no sirven para trabajar los algoritmos

```
data_objects_company <- subset(data_objects_company, select =-c(normalized_name, pa
rent_id, permalink,domain, homepage_url,  twitter_username, logo_url, logo_width, l
ogo_height,    short_description, first_milestone_at, last_milestone_at,
milestones, relationships, created_at, created_by,updated_at))
```

Transformar la columna de fundacion a fecha

```
data_objects_company$founded_at <- as.Date(data_objects_company$founded_at)
```

Calcular la cantidad de directivos de una empresa

```
directivos <- summarise(group_by(data_relaciones,relationship_object_id),
cuenta = sum(!is.na(person_object_id)))
```

Match con tabla objects de empresas con cantidad de directivos

```
data_objects_company$directivos <- directivos$cuenta[match(data_objects_company$id,
directivos$relationship_object_id)]
```

Match con tabla de categoria de universidad

```
data_objects_company$universidad <- degree_empresa$categoria[match(data_objects_com
pany$id,                          degree_empresa$relationship_object_id)]
```

Calcular la cantidad de inversores

```
inversores <- summarise(group_by(data_investments,funded_object_id),
                        cuenta = sum(!is.na(investor_object_id)))
```

Match cantidad de inversores con tabla objects

```
data_objects_company$inversores <- inversores$cuenta[match(data_objects_company$id,
                                                           inversores$fun
ded_object_id)]
```

Agrupar cantidad de oficinas

```
oficinas <- summarise(group_by(data_offices,object_id),
                      cuenta=(sum(!is.na(id))))
```

Match cantidad de oficinas

```
data_objects_company$oficinas <- oficinas$cuenta[match(data_objects_company$id,
                                                       oficinas$object_id)]
```

Limpieza NAs

```
data_objects_company2 <- data_objects_company
attach(data_objects_company2)
```

Calcular la media para el numero de oficinas. La media es 1 Se reemplazo los NAs por la media

```
data_objects_company2$oficinas[is.na(data_objects_company2$oficinas)]<- round(mean(
data_objects_company2$oficinas,na.rm = TRUE))
```

Reemplazar los NAs por la categoria menor en la universidad es decir 2

```
data_objects_company2$universidad[is.na(data_objects_company2$universidad)] <- 2
```

Calcular la media para cantidad de directivos o fundadores

```
data_objects_company2$directivos[is.na(data_objects_company2$directivos)]<- round(m
ean(data_objects_company2$directivos,na.rm = TRUE))
```

Eliminar columnas no necesarias

```


data_objects_company2[,4] <- NULL

data_objects_company3 <- data_objects_company2

data_objects_company3[,18:21]<- NULL

write.csv(data_objects_company3, file="tabla_sucia.csv")
```

Match con la longitud y latitud

```
tabla_final$longitud <- oficinas2$longitude[match(tabla_final$id,oficinas2$object_i
d)]
tabla_final$latitud <- oficinas2$latitude[match(tabla_final$id,oficinas2$object_id)
]

tabla_final$coordenadas <- paste(tabla_final$longitud,",",tabla_final$latitud)

write.csv(tabla_final, file = "tabla_coordenadas.csv")

tabla_coordenadas <- tabla_final[complete.cases(tabla_final),]

tabla_coordenadas$latitud <- NULL
tabla_coordenadas$longitud <- NULL
write.csv(tabla_coordenadas, file ="tabla_coordenadas_final.csv")
```