**##Solucion Lab3**

**require(dplyr)**

**require(readxl)**

**require(tidyverse)**

**require(leaflet)**

**require(plotly)**

setwd("~/GitHub/data-wrangling/data/01")

inv <- read.csv("tabla\_completa.csv")

**#Consideraciones**

dias\_mes = 26

dias\_anio = 286

crecimiento = 0.05

**#Estado actual de la empresa**

**#Si esta en blanco, es despacho**

inv$Tipo <- ifelse(str\_detect(inv$CLIENTE, "Faltante")==TRUE,

"Faltante",

ifelse(str\_detect(inv$CLIENTE,"DEVOLUCION")==TRUE,

"Devolucion",

"Despacho"))

inv %>%

dplyr::group\_by(Tipo) %>%

dplyr::summarise(Cantidad=sum(CANTIDAD),

Total=sum(Q),

Pilotos=length(unique(PILOTO)),

Viajes=n()) %>%

dplyr::mutate(PorcentajeQ = Total/sum(Total),

mensualQ = Total/11)

**#promedio de viajes para devolucion, faltante y despacho**

inv %>%

dplyr::group\_by(Tipo) %>%

dplyr::summarise(Viajes=n()/11)

actual <- inv %>%

dplyr::group\_by(MES,Tipo) %>%

dplyr::summarise(Viajes=n())

plot\_ly(

data=actual,

x=~MES,

y=~Viajes,

color=~Tipo,

type="bar"

) %>%

layout(barmode="stack")

**#Pregunta 1**

inv %>%

dplyr::group\_by(MES, PILOTO) %>%

dplyr::summarise(OCUPACION\_PILOTOS = n()/dias\_mes)

inv %>%

dplyr::group\_by(MES) %>%

dplyr::summarise(viajes = n(),

viajes\_dia = viajes/dias\_mes,

pilotos = length(unique(PILOTO)))

**#Pregunta 2**

**#Codigo de viaje unico?**

unique(inv$COD\_VIAJE) %>% length()

inv %>% nrow()

inv %>%

dplyr::group\_by(UNIDAD) %>%

dplyr::summarise(Freq=n()) %>%

dplyr::mutate(Porcentaje = Freq/sum(Freq))

inv %>%

dplyr::group\_by(MES, UNIDAD) %>%

dplyr::summarise(OCUPACION = n()/dias\_mes)

inv %>%

dplyr::group\_by(UNIDAD) %>%

dplyr::summarise(OCUPACION = quantile(n()/dias\_anio, 0.75)) %>%

dplyr::mutate(OCUPACION\_NUEVA = OCUPACION\*(1+crecimiento),

VEHICULOS\_NUEVOS = ceiling(OCUPACION\_NUEVA)+1)

inv %>%

dplyr::group\_by(MES, UNIDAD) %>%

dplyr::summarise(OCUPACION = quantile(n()/dias\_mes, 0.75)) %>%

dplyr::mutate(OCUPACION\_NUEVA = OCUPACION\*(1+crecimiento),

VEHICULOS\_NUEVOS = ceiling(OCUPACION\_NUEVA)+1)

**#Pregunta 3 y 4**

inv %>%

dplyr::group\_by(MES,CLIENTE) %>%

dplyr::summarise(sum(Q)/sum(CANTIDAD)) %>%

dplyr::arrange(CLIENTE) %>% View()

**##Pregunta 5**

inv %>%

dplyr::mutate(COSTOxUNIDAD = ifelse())

dplyr::group\_by(UNIDAD) %>%

dplyr::summarise(sum(Q),

sum(CANTIDAD))

**#tarifa es igual por diferentes destinos entonces puedo diferenciarla depende de donde este**

inv %>% group\_by(UBICACION) %>% summarise(sum(Q)/sum(CANTIDAD))

**require(plotly)**

inv2 <- inv %>% group\_by(MES) %>% summarise(CANTIDAD=sum(CANTIDAD))

prom <- inv2 %>% summarise(Promedio=mean(CANTIDAD),

percentil75=quantile(CANTIDAD, 0.75))

plot\_ly(

data=inv2,

x=~inv2$MES,

y=~inv2$CANTIDAD,

type="scatter",

mode="line+markers",

name="CANTIDAD"

) %>%

add\_lines(data=prom,

y=~Promedio,

name="Promedio"

) %>%

add\_lines(data=prom,

y=~percentil75,

name="75p"

)

**#tabla para meses**

inv3 <- inv %>%

dplyr::group\_by(MES, UNIDAD) %>%

dplyr::summarise(CANTIDAD=sum(CANTIDAD),

TOTAL=sum(Q))

**#total por tipo de camion por mes**

plot\_ly(

data=inv3,

x=~MES,

y=~TOTAL,

color=~UNIDAD,

type="scatter",

mode="line+markers"

)

**#cantidad por tipo de camion por mes**

plot\_ly(

data=inv3,

x=~MES,

y=~CANTIDAD,

color=~UNIDAD,

type="scatter",

mode="line+markers"

)

**#La tendencia crece para camiones grandes y va bajando para pequenios y paneles entonces se pueden crear campanias para los que**

**#bajan y seguir bien con los grandes o al criterio**

**#cantidad de viajes por piloto por origen**

inv4 <- inv %>% group\_by(PILOTO) %>% summarise(Freq=n())

inv5 <- inv %>% group\_by(PILOTO) %>% summarise(CANTIDAD=sum(CANTIDAD), TOTAL=sum(Q))

ay <- list(

tickfont = list(color = "orange"),

overlaying = "y",

side = "right",

title = "Total"

)

plot\_ly(

data=inv4,

x=~PILOTO,

y=~Freq,

type="bar",

name="Viajes"

) %>%

add\_trace(

data=inv5,

x=~inv5$PILOTO,

y=~inv5$TOTAL,

type="scatter",

mode="line+markers",

name="TOTAL",

yaxis="y2"

) %>%

layout(yaxis = list(title = 'Count'),

yaxis2 = ay)

**#Indice de piloto**

inv %>%

dplyr::group\_by(PILOTO) %>%

dplyr::summarise(Indice = n()/sum(Q))

**###Entendiendo la ubicacion**

para\_grafica <- inv %>%

dplyr::group\_by(MES, UBICACION) %>%

dplyr::summarise(CANTIDAD=sum(CANTIDAD, na.rm=TRUE),

Total=sum(Q, na.rm=TRUE))

plot\_ly(

data=para\_grafica,

x=~MES,

y=~Total,

color=~UBICACION,

colors=c("navy", "skyblue"),

type="bar"

) %>%

layout(yaxis = list(title = 'Total'),

barmode="stack")

para\_grafica2 <- inv %>%

dplyr::group\_by(MES, UBICACION) %>%

dplyr::summarise(Total=sum(Q, na.rm=TRUE)) %>%

dplyr::mutate(Porcentaje=Total/sum(Total))

plot\_ly(

data=para\_grafica2,

x=~MES,

y=~Porcentaje,

color=~UBICACION,

colors=c("navy", "skyblue"),

type="bar"

) %>%

layout(yaxis = list(title = 'Total'),

barmode="stack")

**######PARA MAPAS**

inv\_nuevo <- inv

inv\_nuevo$Departamento <- ifelse(inv\_nuevo$UBICACION==76001,"Guatemala","Escuintla")

data <- inv\_nuevo %>%

dplyr::group\_by(Departamento) %>%

dplyr::summarise(Total=sum(Q))

**#tabla de departamentos**

norm <- read\_excel("C:/Users/JR29/Dropbox (Improgress)/PUM PUR Public/R/Modelo\_costos/PUM PUR Deptos.xlsx")

names(norm) <- c("Departamento", "DeptoN")

data <- merge(data,norm,by="Departamento", all.x=TRUE)

data <- data %>% select(Departamento=DeptoN,Total)

**#Mapa por departamento**

GT <- readr::read\_rds("C:/Users/JR29/Dropbox (Improgress)/EEG LOG Public/Mapas/ANC GOB/GTM\_adm1.rds")

GT$NAME\_1[13]<-"Quetzaltenango"

nombres <- GT$NAME\_1 %>% as.data.frame()

names(nombres)<-"Departamento"

data <- merge(nombres, data, by="Departamento", all.x=TRUE)

variable <- "Total"

GT$Casos <- data[,variable]

b <- quantile(data[,variable], c(0,.2, .4, .6, .8, 1), na.rm = T)

qpal <- colorBin(palette = "Reds", domain = GT$Casos, pretty = F, reverse = F, bins = b,na.color = "#a4a9a6")

labels <- sprintf(

"<strong>%s</strong><br/>%g </sup>",

paste(GT$NAME\_1,GT$NAME\_2, sep = "-"), GT$Casos

) %>% lapply(htmltools::HTML)

leaflet(GT) %>% addPolygons(

fillColor = ~qpal(Casos),

weight = 2,

opacity = 1,

color = "white",

dashArray = "3",

fillOpacity = 0.7,

highlight = highlightOptions(

weight = 5,

color = "#666",

dashArray = "",

fillOpacity = 0.7,

bringToFront = TRUE),

label = labels,

labelOptions = labelOptions(

style = list("font-weight" = "normal", padding = "3px 8px"),

textsize = "15px",

direction = "auto"))%>%

leaflet::addLegend(pal = qpal, values = ~Casos, opacity = 1, title = "Mapa de facturacion 2017")