**Científicos de Datos**

**TP 2 - Pronóstico de clima - R**

**Integrantes:**

Bertino, Ariel

Manzalini, Andrés

**LU:**

247865

247729

**Mails**:

abertino@alumnos.exa.unicen.edu.ar

amanzalini@alumnos.exa.unicen.edu.ar

**Grupo:** 5

**Docentes:**

Illescas, Gustavo

Mora Soto, José Arturo

**Introducción**

En el presente trabajo práctico de la cátedra Científicos de Datos, con los datos históricos del clima de Tandil provistos por Meteored Argentina, se procederá al análisis de las temperaturas desde el año 2010 hasta el 2020. Para ellos es necesario limpiar los datos de entrada, procesarlos y generar algunas herramientas que nos ayuden tanto en el análisis como en la visualización de los datos.

En la segunda parte, la idea es predecir el clima en el futuro. Todo este trabajo se realizará en el lenguaje de programación R.

**Desarrollo**

El objetivo es transformar los datos que nos brinda Meteored Argentina en una representación interna propia de R (dataframe). Una vez logrado esto, se procederá a una serie de procesamientos sobre los datos con objetivo de dar soporte a las funcionalidades solicitadas por la cátedra. (aca podriamos agregar un par de lineas mas en lo que seria desarrollo)

*Aclaración 1: Todo lo mostrado en este informe si bien son capturas del programa desarrollado en R, las explicaciones se realizarán siguiendo una mecánica de pseudocódigo. El código estará en el anexo 1 Y anexo 2 ,al final del informe. Estos dos anexos muestran dos enfoques que decidimos tomar para la implementación. Uno de ellos se procede a armar cada estructura si cabe el término de una manera más manual y progresiva(anexo 1) y la otra usando funciones de librerías puntuales al cuales contribuyen a tener un código más compacta y hasta en cierto punto más legible.*

*Aclaración 2: Este informe fue estructurado teniendo como núcleo el código detallado en el anexo 1*

**Preparación de los datos**

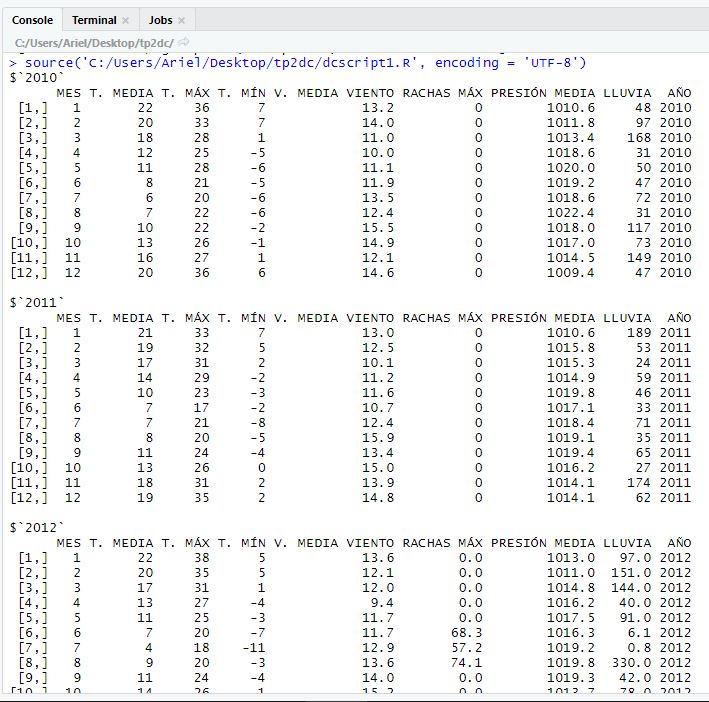
Para cargar los datos desde Meteored Argentina se descarga el archivos separado conformada por una planilla de cálculo excel, la cuando dispone de una serie de folios donde cada uno contiene los datos del clima para todos los meses de un año dado.

Este archivo se carga dentro del ambiente de desarrollo RStudio y desde ese momento se lo puede empezar a procesar.

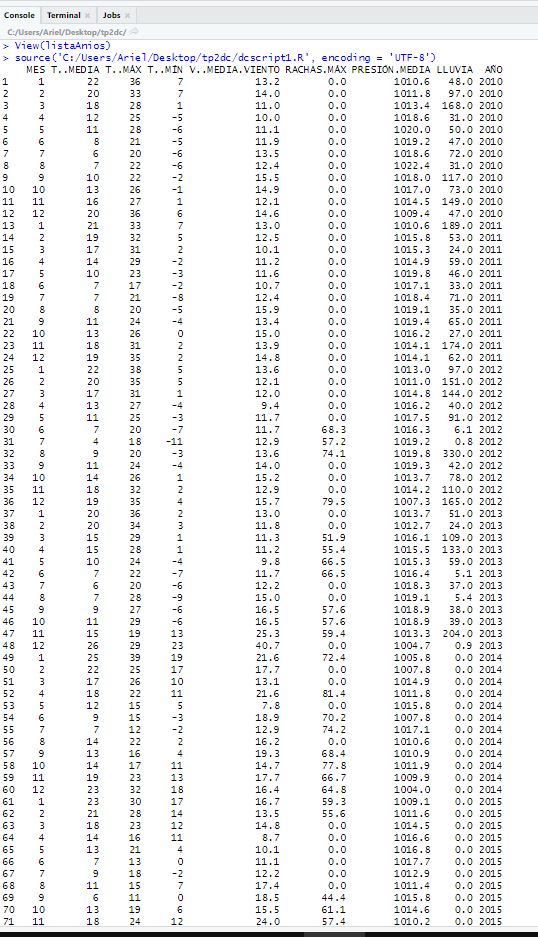
La siguiente imagen muestra una parte de una de las dos estructuras que usamos para manipular la información. Esta estructura es una lista de dataframes, más concretamente cada posicion de la lista consta de dos elementos, un campo que funciona como el año de registro de los datos y como segundo campo es el dataframe en sí con los diferentes valores de MES, T. MEDIA, T.MÁX, T. MÍN, V. MEDIA VIENTO, RACHAS MÁX, PRESIÓN MEDIA Y LLUVIA Y por último un campo correspondiente al AÑO.

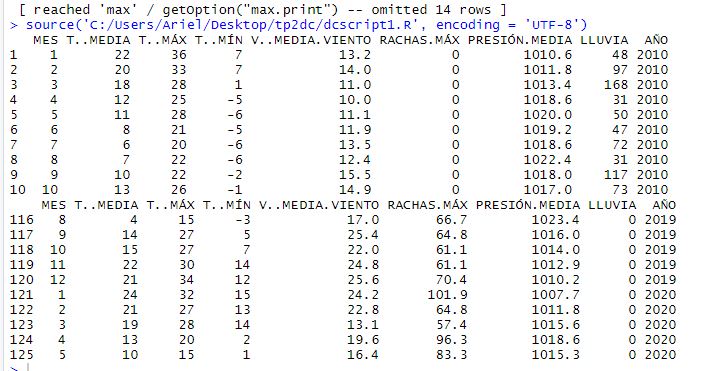
A su vez otro de la requisitos era tener todos los datos en una sola estructura. Ambas estructuras se muestran a continuación.

Estructura de cada planilla, por año



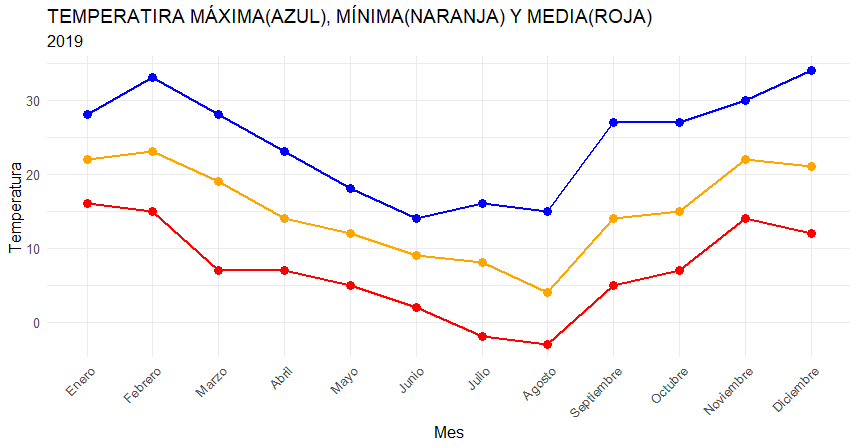
Estructura unificada/completa



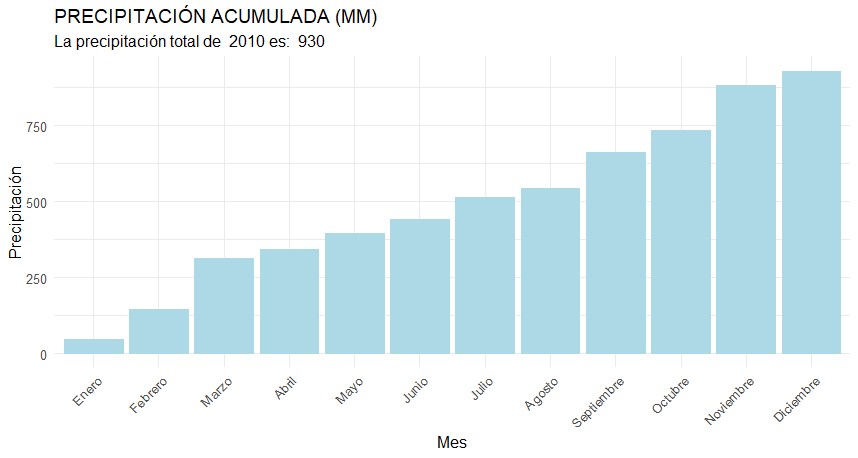
funcion: mostrar los primeros y los últimos diez registros del clima

**Análisis preliminar de los datos**

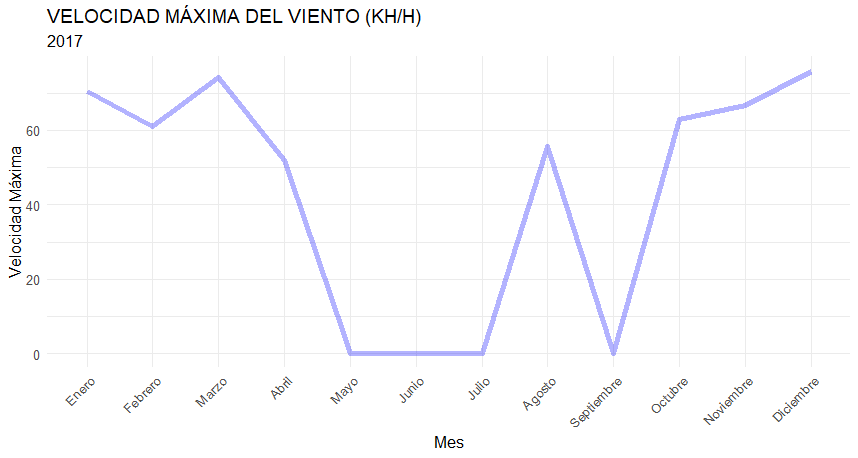
Para la creación del siguiente gráfico, dado un año determinado se muestra un gráfico de líneas múltiple en el cual se distinguen las tres temperaturas solicitadas(T..MÁX, T..MEDIA Y T..MÍN). Esto se logra aislando dichas temperaturas del subframes que corresponde a los datos del año solicitado.



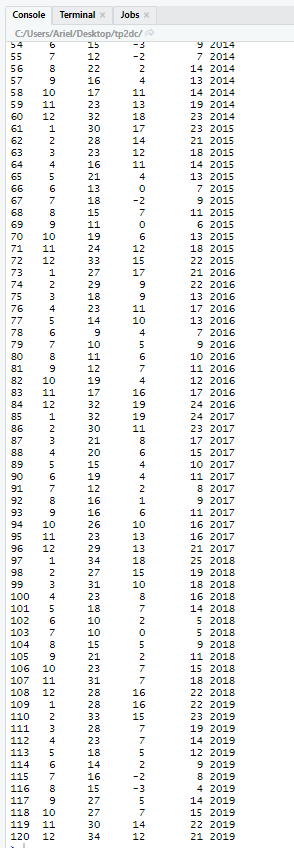
Para generar los datos necesarios para la disposición de un gráfico de barras que indique las precipitaciones acumuladas de un año en particular, se ubica en la lista de dataframes por año, el año solicitado y sobre el campo de “LLUVIA” y se arma un dataframe nuevo el cual contendrá en la posición siguiente la suma de las precipitaciones anteriores para luego generar el gráfico.



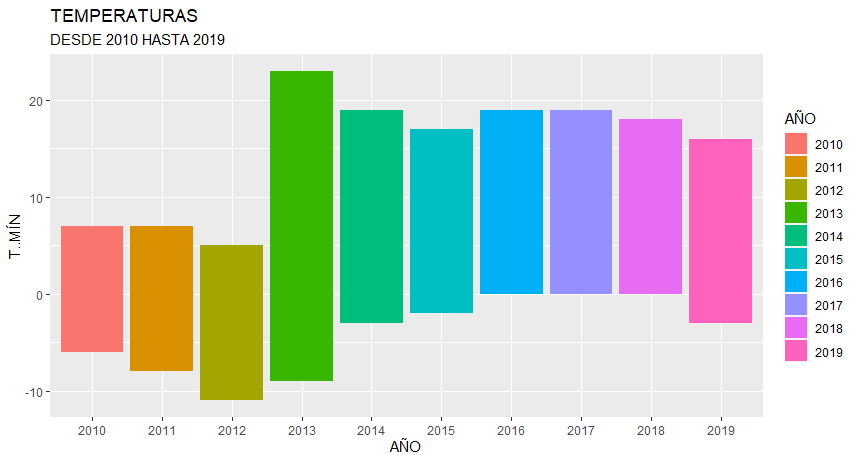
El siguiente requisito es generar un gráfico de línea que dado un año retorne la velocidad máxima que el viento alcanzó en cada mes. Para esto se seleccionan como ejes las columnas “MES” y ”RACHA MÁX” como ejes x e y respectivamente y se muestra el resultado



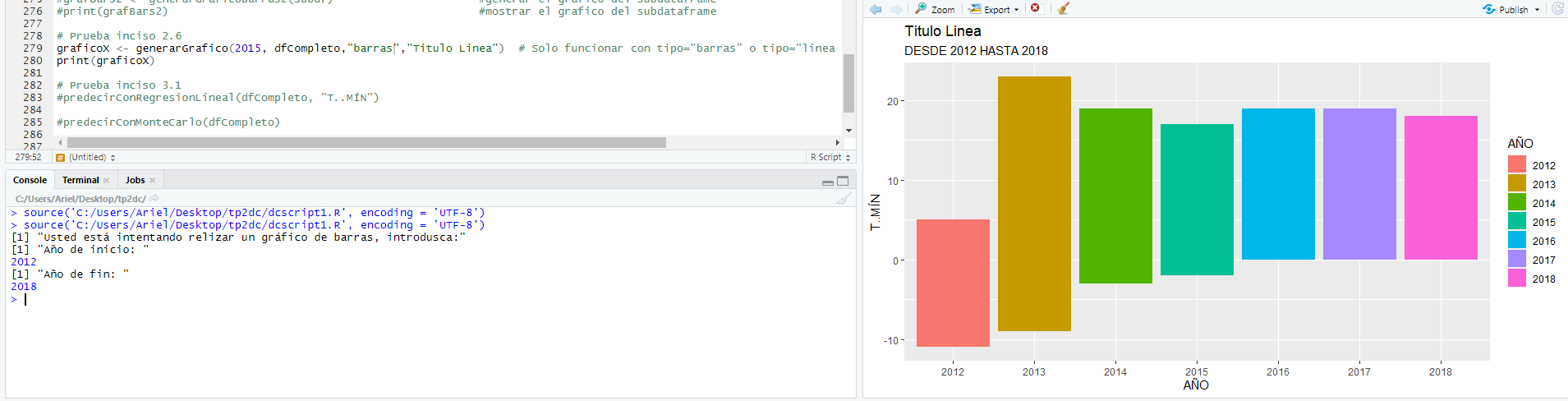
El apartado siguiente solicitaba generar un dataframe que a partir de un rango de años y el dataframe completo arroje como resultado un subdataframe que somo contenga las columnas “T..MÁX”, “T.MÍN” y “T..MEDIA”. A continuación una captura de la salida de la consola tras ejecutar las funciones correspondientes a este servicio

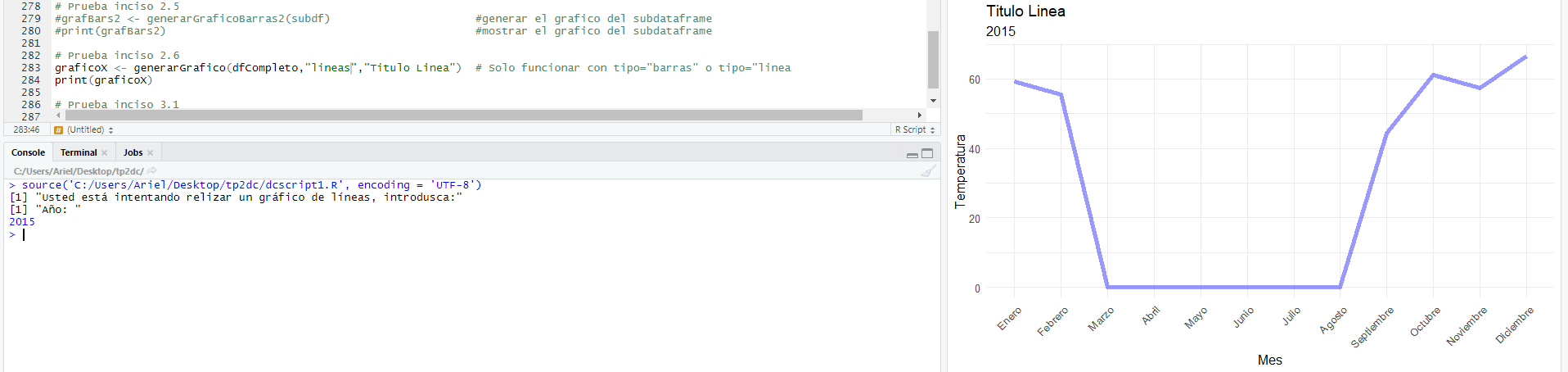


Siguiendo se pedía que a partir de dataframe generado anteriormente se bosqueje un gráfico de barras que muestre estas temperaturas despliegas para todos los años. En este punto cabe aclarar que para que no sea tan engorroso el gráfico final o que resulte estar muy cargado de información, permitimos seleccionar qué temperatura se desea mostrar en el gráfico. Como último punto antes de mostrar una captura del gráfico remarcamos que el mismo muestra de cada año el rango de temperatura minimo y maximo del campo de interes. Para este caso y como ejemplo se mostró desde el año 2010 hasta el 2019 la temperatura mínima de cada mes.



Como punto opcional de este apartado(analisis preliminar de los datos), se pedía armar una función que permita dados el tipo de gráfico que se desea crear(barras o líneas), el año (único o rango), el conjunto de datos y el título del mismo, crear el gráfico correspondiente. Llegados a este punto decidimos reutilizar parte de las funciones realizadas para incisos anteriores con lo cual. SI se desea crear un gráfico de barras pedimos por consola un rango de años y si queremos una de líneas solicitamos de que año puntual hacerlo, se carga por teclado y se genera el gráfico. a continuación un ejemplo de ambos casos.



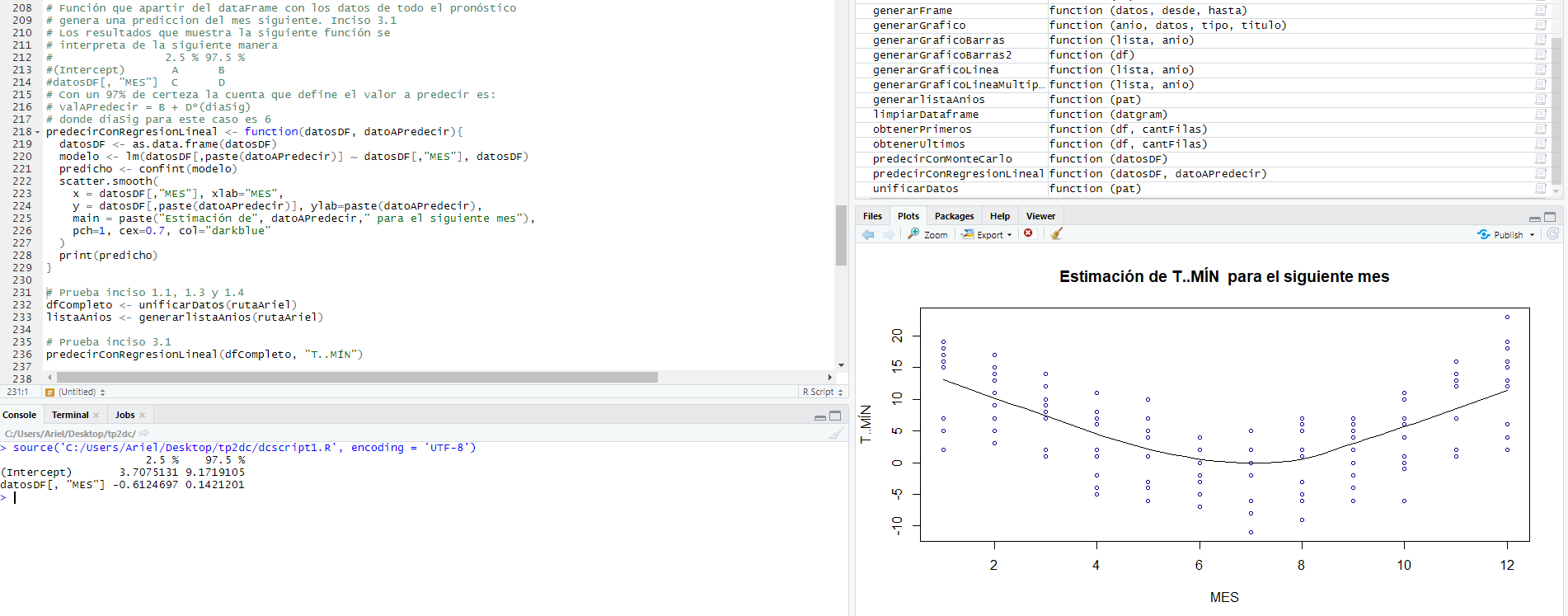


**Creación de un modelo para el pronóstico del clima**

A continuación se implementó una regresión lineal simple que a partir del dataframe con los datos de todo el pronóstico genera una predicción del mes siguiente. En la imagen se muestra que se utilizó la función “confint”, esta función recibe un modelo que contiene la variable a predecir y en qué otra variables basar la predicción y arroja como resultado una tabla con los siguientes valores. Utilizamos letras en el informe para simplificar la explicación.

|  |  |  |
| --- | --- | --- |
|  | 2.5 % | 97.5 % |
| (Intercept) | A | B |
| datosDF[, "MES"] | C | D |

Estos datos se interpretan de la siguiente manera: con un 97.5 % de certeza la cuenta que define la prediccion del valor a predecir es : valAPredecir = B + D\*(diaSig), donde diaSig para este caso es 6 y valAPredecir es “T..MÍN”



**Bibliografía consultada**

<https://rstudio.com/wp-content/uploads/2015/04/ggplot2-spanish.pdf>

<https://rstudio-pubs-static.s3.amazonaws.com/385451_2683733d503746acb3c135db8692f1bc.html>

<https://rpubs.com/Rortizdu/140196>

<https://rpubs.com/Rortizdu/140189>

<https://rpubs.com/joser/RegresionSimple>

<https://rpubs.com/Joaquin_AR/223351>

<https://rpubs.com/daniballari/ggplot>

<https://rpubs.com/Rortizdu/140203>

<https://luisxsuper.github.io/m_ggplot2.html>

<https://www.datanalytics.com/2011/02/15/como-reordenar-niveles-de-factores-en-r/>

<https://www.datanalytics.com/libro_r/elementos-de-un-grafico-en-ggplot2.html>

<https://www.datanalytics.com/libro_r/regresion-lineal.html>

<https://pmoracho.github.io/blog/2019/03/10/sample-ggplot-plots/>

<https://sites.google.com/site/digitcognem/editcolors>

<https://stackoverrun.com/es/q/5606040>

<https://ggplot2.tidyverse.org/reference/geom_text.html>

<https://progexpertos.com/q/9338/how-can-i-sort-by-month-and-year>

<https://guamandseduardo.rbind.io/slides/plots-ggplot2.html#1>

<http://www.estadistica-dma.ulpgc.es/cursoR4ULPGC/9b-grafBarplot.html>

<https://www.institutomora.edu.mx/testU/SitePages/martinpaladino/graficos_con_r_y_ggplot.html>

<https://www.diegocalvo.es/ejemplo-de-grafico-de-lineas-en-r/>

<https://www.it-swarm.dev/es/r/como-encajar-una-curva-suave-mis-datos-en-r/969118184/>

<https://www.cs.us.es/~fran/TIB/semana5.pdf>

<https://www.statmethods.net/graphs/scatterplot.html>

<http://www.sthda.com/english/wiki/r-plot-pch-symbols-the-different-point-shapes-available-in-r>

**Anexo 1**

El código está estructurado de manera tal que se pueda probar cada inciso por separado viendo así los resultados ya sean por consola y/o gráficos que se vayan generando. Con lo cual habría que ir comentando y descomentando aquello que se desee ir testeando su ejecución.

# Librerías usadas

# Instalar librerías

#install.packages("readxl")

#install.packages("tidyverse")

#install.packages("ggplot2")

#install.packages("caret")

# Cargar librerías

library(readxl)

library(tidyverse)

library(ggplot2)

library(caret)

# Definición de variables

cantAnios <- 10

anioBase <- 2010

rutaAndy <- "~/Documentos/Cientifico de Datos/Practicos/datos-clima-tandil.xlsx"

rutaAriel <- "C:/Users/Ariel/Desktop/datos-clima-tandil.xlsx"

dfCompleto <- data.frame()

listaAnios <- list()

meses<- c("Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

ordenMeses <- factor(meses)

options(digits=5,default=0)

# Definición de funciones

#-----------------------------------------------------------------------------------------------------------------

# 1. PREPARACION DE DATOS

# Función que convierte un dataframe a numeric

convertirDF <- function(df){

dfConvertido <- apply(df,c(1,2),as.numeric)

return(dfConvertido)

}

# Función que limpia un dataframe de ruido

# aclaración: las celdas sin valor las completamos con 0 (cero)

limpiardataframe <- function(datgram){

datgram <- as.data.frame(datgram)

# Elimino las unidades dejando solo los numeros pero sigue siendo tipo string o character es lo mismo en R

for (j in 1:nrow(datgram)){

for (k in 1:ncol(datgram)) {

if (class(datgram[j,k]) == "character")

if (str\_detect(datgram[j,k], "°C") | str\_detect(datgram[j,k], "mm"))

datgram[j,k] <- substr(datgram[j,k], 0, (nchar(datgram[j,k])-3 ))

if (str\_detect(datgram[j,k], "km"))

datgram[j,k] <- substr(datgram[j,k], 0, (nchar(datgram[j,k])-5 ))

if (str\_detect(datgram[j,k], "hPa"))

datgram[j,k] <- substr(datgram[j,k], 0, (nchar(datgram[j,k])-4 ))

if (str\_detect(datgram[j,k], "--"))

datgram[j,k] <- "0" # O nulo en su defecto

}

}

#convierto todo el dataframe a numerico

datgram <- convertirDF(datgram)

return(datgram)

}

# Función que genera una lista donde cada posicion es un datagrama con los datos de cada año

generarlistaAnios <- function(pat){

auxL <- list()

for (i in 0:cantAnios) {

tablaExcel <- read\_excel(pat, sheet=i+1)

anio <- anioBase+i

auxL[[paste(anio)]] <- limpiardataframe(tablaExcel)

auxL[[paste(anio)]] <- cbind(auxL[[paste(anio)]], "AÑO"=anio)

}

return(auxL)

}

# Funcion que unifica

unificarDatos <- function(pat){

aux <- data.frame()

resul <- data.frame()

for (i in 0:cantAnios) {

tablaExcel <- read\_excel(rutaAriel, sheet=i+1)

anio <- anioBase+i

aux <- data.frame(tablaExcel, "AÑO"=anio) #esta linea esta agregando una columna más al final

aux <- limpiardataframe(aux)

resul <- rbind(resul, aux)

}

return(resul)

}

# Funcion que muestra los primeros "CantFilas" registros de un datagrama

obtenerPrimeros <- function(df, cantFilas){

if (class(df) == "data.frame" && cantFilas <= nrow(df) ) {

resul <- head(df, cantFilas)

} else if (class(df) == "list") {

for (i in 1:length(df)){

dfaux <-as.data.frame(df[[paste(anioBase+i)]])

resul <- head(dfaux, cantFilas)

}

}

return(resul)

}

# Funcion que muestra los ultimos "cantFilas" registros de un datagrama

obtenerUltimos <- function(df, cantFilas){

if (class(df) == "data.frame" && cantFilas <= nrow(df) ) {

resul <- tail(df, cantFilas)

} else if (class(df) == "list") {

for (i in 0:length(df)){

dfaux <-as.data.frame(df[[paste(anioBase+i)]])

resul <- tail(dfaux, cantFilas)

}

}

return(resul)

}

# 2. ANALISIS PRELIMINAR DE LOS DATOS

# Tuvimos muchos problemas a la hora de generar los gráficos

# También quisimos realizar la parte de pasarle por parámetro

# el tipo de gráfico pero con ggplot se nos hizo muy complicado

# optamos por la siguiente función que directamente te genera

# el grafico con las tres líneas inciso 2.1

generarGraficoLineaMultiple <- function(lista, anio){

b <- lista[[paste(anio)]][,"T. MÍN"]

c <- lista[[paste(anio)]][,"T. MÁX"]

d <- lista[[paste(anio)]][,"T. MEDIA"]

datu <- data.frame(meses,b,c,d)

p <- ggplot(datu, aes(x=ordered(meses,ordenMeses), y=b, group=NA)) + geom\_point(colour="red", size = 3) + geom\_line(colour="red", size=1 )

p <- p + labs(title = "TEMPERATIRA MÁXIMA(AZUL), MÍNIMA(NARANJA) Y MEDIA(ROJA)", subtitle = anio) + theme\_minimal() + ylab("Temperatura")

p <- p + theme(text = element\_text(size=12), axis.text.x = element\_text(angle=45, hjust=1)) + xlab("Mes")

p <- p + geom\_line(aes(y = c), color = "blue", size=1) + geom\_point(aes(y = c),color="blue", size=3) #azul

p <- p + geom\_line(aes(y = d), color = "orange", size=1) + geom\_point(aes(y = d),color="orange", size=3) #naranja

return(p)

}

# Función que genera el grafico de barras inciso 2.2

generarGraficoBarras <- function(lista, anio){

b <- lista[[paste(anio)]][,"LLUVIA"]

for (i in 2:length(b)){

b[i] <- b[i] + b[i-1]

}

frape <- data.frame(meses,b)

p <- ggplot(frape, aes(x=reorder(meses,b), y=b)) + theme\_minimal() + ggtitle("PRECIPITACIÓN ACUMULADA (MM)")

p <- p + theme(text = element\_text(size=12), axis.text.x = element\_text(angle=45, hjust=1))

p <- p + labs(subtitle=paste("La precipitación total de ", anio, "es: ", b[length(b)] ))+ xlab("Mes") + ylab("Precipitación")

p <- p + geom\_bar(stat="identity", position="stack",color="lightblue" , fill="lightblue")

return(p)

}

# Función que genera el grafico de barras inciso 2.3

generarGraficoLinea <- function(lista, anio){

b <- lista[[paste(anio)]][,"RACHAS MÁX"]

df <- data.frame(meses, b)

p <- ggplot(df, aes(x=ordered(meses,ordenMeses), y=b, group=NA)) + geom\_line(colour="blue", size=2, alpha=0.3 )

p <- p + labs(title = "VELOCIDAD MÁXIMA DEL VIENTO (KH/H)", subtitle = anio) + theme\_minimal() + ylab("Velocidad Máxima")

p <- p + theme(text = element\_text(size=12), axis.text.x = element\_text(angle=45, hjust=1)) + xlab("Mes")

return(p)

}

# Función que extrea un subdataframe inciso 2.4

generarFrame <- function(datos, desde, hasta){

datos <- as.data.frame(datos)

df <- datos[datos$AÑO>=desde&datos$AÑO<=hasta, c("MES","T..MÁX","T..MÍN","T..MEDIA","AÑO")]

return(df)

}

# Función que grafica un subdataframe inciso 2.5

generarGraficoBarras2 <- function(df){

#no puedo hacer que me ponga las tres temperauras

#en su defecto explicar que hay que cambiar y=T..MÁX O y=T..MEDIA

#y lo que está mostrando es la temp minima y maxima de la T..MÍN

df <- as.data.frame(df)

df[,"AÑO"] <- as.character(df[ ,"AÑO"])

p <- ggplot(data=df, aes(x=AÑO, y=T..MÍN, fill=AÑO)) + geom\_bar(stat="identity", position="dodge")

p <- p +theme(text = element\_text(size=9), axis.text.x = element\_text(angle=45, hjust=1))

p <- p + labs(title = paste("TEMPERATURAS "), subtitle = paste("DESDE",min(df$AÑO),"HASTA",max(df$AÑO))) + theme\_grey()

return(p)

}

# Función que apatir del año, el conjunto de datos, el tipo de gráfico y el título del gráfico

# genera el grafico de los incisos 2.3 y 2.4

# Función muy precaria inciso 2.6

generarGrafico <- function(datos, tipo, titulo){

#datos <- as.data.frame(datos)

#df <- datos[datos$AÑO==anio, c("MES","T..MÁX","T..MÍN","T..MEDIA","RACHAS.MÁX","AÑO")]

#pFinal <- ggplot(data=df, aes(x=AÑO, y=T..MÍN, fill=AÑO))

a <- numeric()

if (tipo == "barras"){

print("Usted está intentando relizar un gráfico de barras, introdusca:")

print("Año de inicio: ")

a <- readline(a)

print("Año de fin: ")

b <- numeric()

b <- readline(b)

df2 <- generarFrame(datos, a, b)

grafico <- generarGraficoBarras2(df2) + labs(title=paste(titulo))

}

if (tipo == "lineas"){

datos <- as.data.frame(datos)

print("Usted está intentando relizar un gráfico de líneas, introdusca:")

print("Año: ")

a <- readline(a)

b <- datos[datos$AÑO==a,c("RACHAS.MÁX")]

df <- data.frame(meses, b)

grafico <- ggplot(df, aes(x=ordered(meses,ordenMeses), y=b, group=NA)) + geom\_line(colour="blue", size=2, alpha=0.4)

grafico <- grafico + labs(title = paste(titulo), subtitle = a) + theme\_minimal() + ylab("Temperatura")

grafico <- grafico + theme(text = element\_text(size=12), axis.text.x = element\_text(angle=45, hjust=1)) + xlab("Mes")

}

return(grafico)

}

# CREACIÓN DE UN MODELO PARA EL PRONÓSTICO DEL CLIMA

# Función que apartir del dataframe con los datos de todo el pronóstico

# genera una prediccion del mes siguiente. Inciso 3.1

# Los resultados que muestra la siguiente función se

# interpreta de la siguiente manera

# 2.5 % 97.5 %

#(Intercept) A B

#datosDF[, "MES"] C D

# Con un 97% de certeza la cuenta que define el valor a predecir es:

# valAPredecir = B + D\*(diaSig)

# donde diaSig para este caso es 6

predecirConRegresionLineal <- function(datosDF, datoAPredecir){

datosDF <- as.data.frame(datosDF)

modelo <- lm(datosDF[,paste(datoAPredecir)] ~ datosDF[,"MES"], datosDF)

predicho <- confint(modelo)

scatter.smooth(

x = datosDF[,"MES"], xlab="MES",

y = datosDF[,paste(datoAPredecir)], ylab=paste(datoAPredecir),

main = paste("Estimación de", datoAPredecir," para el siguiente mes"),

pch=1, cex=0.7, col="darkblue"

)

print(predicho)

}

# MonteCarlo

# Inciso 3.2

# Incompleto

predecirConMonteCarlo <- function(datosDF){

datosDF <- as.data.frame(datosDF)

print(datosDF)

}

#-----------------------------------------------------------------------------------------------------------------

#-----------------------------------------------------------------------------------------------------------------

# Parte de Código que simula el main donde se llaman a todas las funciones

# Prueba inciso 1.1, 1.3 y 1.4

dfCompleto <- unificarDatos(rutaAriel)

listaAnios <- generarlistaAnios(rutaAriel)

#print(listaAnios) #mostrar estructuras

#print(dfCompleto)

# Prueba inciso 1.2

#primeros <- obtenerPrimeros(dfCompleto,10) #obtener primeros 10 registro

#print(primeros) #mostrar primeros 10 registro

#ultimos <- obtenerUltimos(dfCompleto,10) #obtener últimos 10 registro

#print(ultimos) #mostrar últimos 10 registro

# Prueba inciso 2.1

#grafMulLines <- generarGraficoLineaMultiple(listaAnios,2019) #generar graficos

#print(grafMulLines) #mostrar graficos

# Prueba inciso 2.2

#grafBars <- generarGraficoBarras(listaAnios, 2010)

#print(grafBars)

# Prueba inciso 2.3

#grafLinea <- generarGraficoLinea(listaAnios, 2017)

#print(grafLinea)

# Prueba inciso 2.4

#subdf <- generarFrame(dfCompleto, 2010, 2019) #extraer subdataframe

#print(subdf) #mostrar subdataframe

# Prueba inciso 2.5

#grafBars2 <- generarGraficoBarras2(subdf) #generar el grafico del subdataframe

#print(grafBars2) #mostrar el grafico del subdataframe

# Prueba inciso 2.6

#graficoX <- generarGrafico(dfCompleto,"lineas","Titulo Linea") # Solo funcionar con tipo="barras" o tipo="linea

#print(graficoX)

# Prueba inciso 3.1

predecirConRegresionLineal(dfCompleto, "T..MÍN")

**Anexo 2**

Esta versión del código se centra mucho en la función mutate de la librería tidyverse, esta función se encarga a partir de un dataframe y las columnas involucrada en la operación a realizar agrega y/o modifica el dataframe ahorrando el tratamiento manual de las columnas y por sobre todo ahorra en general un orden de complejidad temporal por ejemplo algoritmos que normalmente influyen en un complejidad temporal O(n^3) a O(n^2) incluso en casos puntuales la logra reducir a O(n) o O(n log (n)), con ”n” siendo el tamaño de la entrada(en estos casos cantidad de registros de dataframe).

library(readxl)

library(tidyverse)

# 1. PREPARACION DE DATOS

df\_clima <- data.frame()

meses<- c("Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

options(digits=5,default=0)

for (i in 0:10) { # para cada planilla de clima

datos\_clima\_tandil <- read\_excel("~/Documentos/Cientifico de Datos/Practicos/datos-clima-tandil.xlsx", sheet=i+1)

year <- 2010+i

print(paste0("AÑO ",year))

df\_clima\_tandil <- data.frame(datos\_clima\_tandil, "YEAR"=year)

#print(df\_clima\_tandil) # df crudo

#limpieza de caracteres

# limpio temperaturas

df\_clima\_tandil <- df\_clima\_tandil %>% mutate(T..MEDIA = as.numeric(substr(T..MEDIA, 1,2)),

T..MÁX = as.numeric(substr(T..MÁX, 1, 2)),

T..MÍN = as.numeric(substr(T..MÍN, 1, 2)),

MES = as.integer(MES))

# limpio vientos

v <- substr(df\_clima\_tandil$V..MEDIA.VIENTO, 1,5)

v <- str\_replace(v,"k","")

v <- str\_replace(v,"m","")

df\_clima\_tandil <- df\_clima\_tandil %>% mutate(V..MEDIA.VIENTO = as.numeric(v))

# limpio rachas maximas

v\_max <- substr(df\_clima\_tandil$RACHAS.MÁX,1,5)

v\_max <- str\_replace(v\_max,"km","")

v\_max <- str\_replace(v\_max,"--","") #dejo como NA.

# Ver que valor usar como defecto! Si la media, una constante, 0 o dejarlo asi

df\_clima\_tandil <- df\_clima\_tandil %>% mutate(RACHAS.MÁX = as.numeric(v\_max))

# limpio presion media (hPa)

p <- substr(df\_clima\_tandil$PRESIÓN.MEDIA,1,7)

p <- str\_replace(p,"hP","")

df\_clima\_tandil <- df\_clima\_tandil %>% mutate(PRESIÓN.MEDIA = as.numeric(p))

# limpio lluvia (mm)

ll <- substr(df\_clima\_tandil$LLUVIA,1,6)

ll <- str\_replace(ll,"mm","")

ll <- str\_replace(ll,"--","")

df\_clima\_tandil <- df\_clima\_tandil %>% mutate(LLUVIA = as.numeric(ll))

print(df\_clima\_tandil)

df\_clima <- rbind(df\_clima, df\_clima\_tandil)

print("---------------------------")

}

# 2. ANALISIS PRELIMINAR DE LOS DATOS

df\_clima\_2019 <- df\_clima %>% filter(YEAR==2019) #%>% mutate(MES = as.character(meses))

glimpse(df\_clima\_2019)

# 2.1 GRAFICO DE LINEAS DE TEMPERATURA MIN, MAX Y MEDIA DEL AÑO 2019

p <- ggplot(df\_clima\_2019, aes(x=MES))

p <- p + geom\_line(aes(y=T..MEDIA), color="green") + geom\_point(aes(y = T..MEDIA), color="green", size=2)

p <- p + geom\_line(aes(y=T..MÍN), color="blue") + geom\_point(aes(y = T..MÍN), color="blue", size=2)

p <- p + geom\_line(aes(y=T..MÁX), color="red") + geom\_point(aes(y = T..MÁX), color="red", size=2)

p <- p + labs(title = "TEMPERATURA MÁXIMA, MÍNIMA Y MEDIA") + theme\_minimal() + ylab("Temperatura") + xlab("Mes")

p <- p + theme(text = element\_text(size=12), axis.text.x = element\_text(angle=45, hjust=1))

p <- p + scale\_x\_continuous(breaks = df\_clima\_2019$MES, labels=meses)

p

#2.2 GRAFICO DE BARRAS QUE MUESTRE PRECIPITACIONES EN 2010

df\_clima\_2010 <- df\_clima %>% filter(YEAR==2010)

p2 <- ggplot(df\_clima\_2010, aes(x=MES, y=LLUVIA)) + theme\_minimal() + ggtitle("PRECIPITACIÓN ACUMULADA (MM)")

p2 <- p2 + theme(text = element\_text(size=12), axis.text.x = element\_text(angle=45, hjust=1))

p2 <- p2 + geom\_bar(stat="identity", position="stack",color="lightblue" , fill="lightblue")

p2 <- p2 + scale\_x\_continuous(breaks = df\_clima\_2010$MES, labels=meses)

p2

#2.3 GRAFICO DE BARRAS QUE MUESTRE VELOCIDAD MAXIMA EN 2017

df\_clima\_2017 <- df\_clima %>% filter(YEAR==2017)

#completo los NA con la media de las velocidades maximas de viento del año

df\_clima\_2017 <- df\_clima\_2017 %>% mutate(RACHAS.MÁX = replace\_na(RACHAS.MÁX, as.numeric(mean(df\_clima\_2017$RACHAS.MÁX, na.rm=TRUE))))

df\_clima\_2017

p3 <- ggplot(df\_clima\_2017, aes(x=MES, y=RACHAS.MÁX)) + geom\_line(colour="blue", size=2, alpha=0.3)

p3 <- p3 + labs(title = "VELOCIDAD MÁXIMA DEL VIENTO (KH/H)") + theme\_minimal() + ylab("Temperatura")

p3 <- p3 + theme(text = element\_text(size=12), axis.text.x = element\_text(angle=45, hjust=1)) + xlab("Mes")

p3 <- p3 + scale\_x\_continuous(breaks = df\_clima\_2017$MES, labels=meses)

p3

# 2.4 dataframe que contiene temperatura media de 2010 a 2019

df\_temp\_media\_2010\_2019 <- df\_clima %>% filter(YEAR %in% 2010:2019) %>% group\_by(YEAR) %>% select(MES, YEAR, T..MEDIA, T..MÁX, T..MÍN)

# 2.5

p4 <- ggplot(df\_temp\_media\_2010\_2019, aes(x=MES))

p4 <- p4 + geom\_col(aes(y=T..MEDIA))

p4 <- p4 + geom\_col(aes(y=T..MÁX))

p4 <- p4 + geom\_col(aes(y=T..MÍN))

#p4 <- p4 + scale\_x\_continuous(breaks = df\_temp\_media\_2010\_2019$MES, labels=meses)

p4 <- p4 + facet\_wrap(~ YEAR)

p4

# 3. CREACION DE MODELO PARA PRONOSTICAR

# usa como data de entrenamiento df\_temp\_media\_2010\_2019

# como data de validacion 2020 el df\_clima\_2020

df\_clima\_train <- df\_clima %>% filter(YEAR %in% 2010:2019)

df\_clima\_2020 <- df\_clima %>% filter(YEAR==2020)

df\_clima\_2020

df\_clima\_train

# 3.1 modelo de regresion lineal

modelo\_rl <- lm( T..MÁX ~ MES, # ( y ~ x ) , con y={T..MEDIA, T..MÍN, T..MAX}

data = df\_clima\_train )

summary( modelo\_rl )

df\_clima\_2020 #esta funcion no hace montecarlo

# 3.2 predice el clima maximo de 2020

predict(modelo\_rl, df\_clima\_2020, interval="predict") #interval = "confidence")