Trabajo Práctico
Análisis y Predicción del Clima en Tandil

Científicos de Datos 2021

Integrantes:

Raineri Franco

LU: 249293

Mail: francoraineri6@gmail.com

Aranda Carlos

Lu:248832

Mail: caranda@alumnos.exa.unicen.edu.ar

Introducción

Para el desarrollo de este trabajo se nos pidió realizar un análisis de los datos meteorológicos de Tandil desde 2010 a la fecha para poder generar un modelo que ayude a los ciudadanos a contar con un mejor pronóstico del clima a 1,3, 5 y 10 días. Para realizar esta tarea haremos uso de los conocimientos aprendidos en la materia y en los cursos realizados sobre el lenguaje R en la plataforma DataCamp.

1.1 Preparación de Datos:

Carga los datos proporcionados en formato CSV en un DataFrame para cada uno de los años. Para realizar este ejercicio, lo primero que hicimos fue convertir cada hoja de cálculo correspondiente a cada año, contenida en el archivo xlsx brindado, al formato CSV.
Luego, cargamos diferentes DataFrame de climas en una lista donde su índice era el año del clima de tandil: list_df[["año clima"]] <- dataFrame_AñoClima

1.2 Despliegue en Pantalla

Despliega en pantalla los primeros diez registros de cada DataFrame.
Para este ejercicio utilizaremos la función que proporciona R llamada head(), la cual devuelve los primeros N datos que se requieran de un objeto, siendo el objeto el primer parámetro de la función y N el segundo parámetro de la función.

Despliega en pantalla los últimos diez registros de cada DataFrame.
Para este ejercicio utilizaremos la función que proporciona R llamada tail(), la cual devuelve los últimos N datos que se requieran de un objeto, siendo el objeto el primer parámetro de la función y N el segundo parámetro de la función.

Para estos últimos incisos anteriores, los dataframes con menos cantidad de meses registrados se vieron impresos completamente.

1.3 Limpieza de los datos

Limpia los datos de tal forma que todos los valores sean numéricos, o nulos en caso de que el dato no esté disponible. Para la limpieza de los datos recorremos cada data frame y a cada celda del mismo se le aplica la función **gsub**(exp , remp , celda).
El primer argumento, es una expresión regular a modificar, y las ocurrencias de esta son reemplazadas por el segundo argumento. El tercer argumento es la celda a limpiar.
En este caso reemplazamos todos aquellos caracteres que no sean numéricos por un caracter vacío.
Luego convertimos los datos a numéricos para así poder operar con ellos.

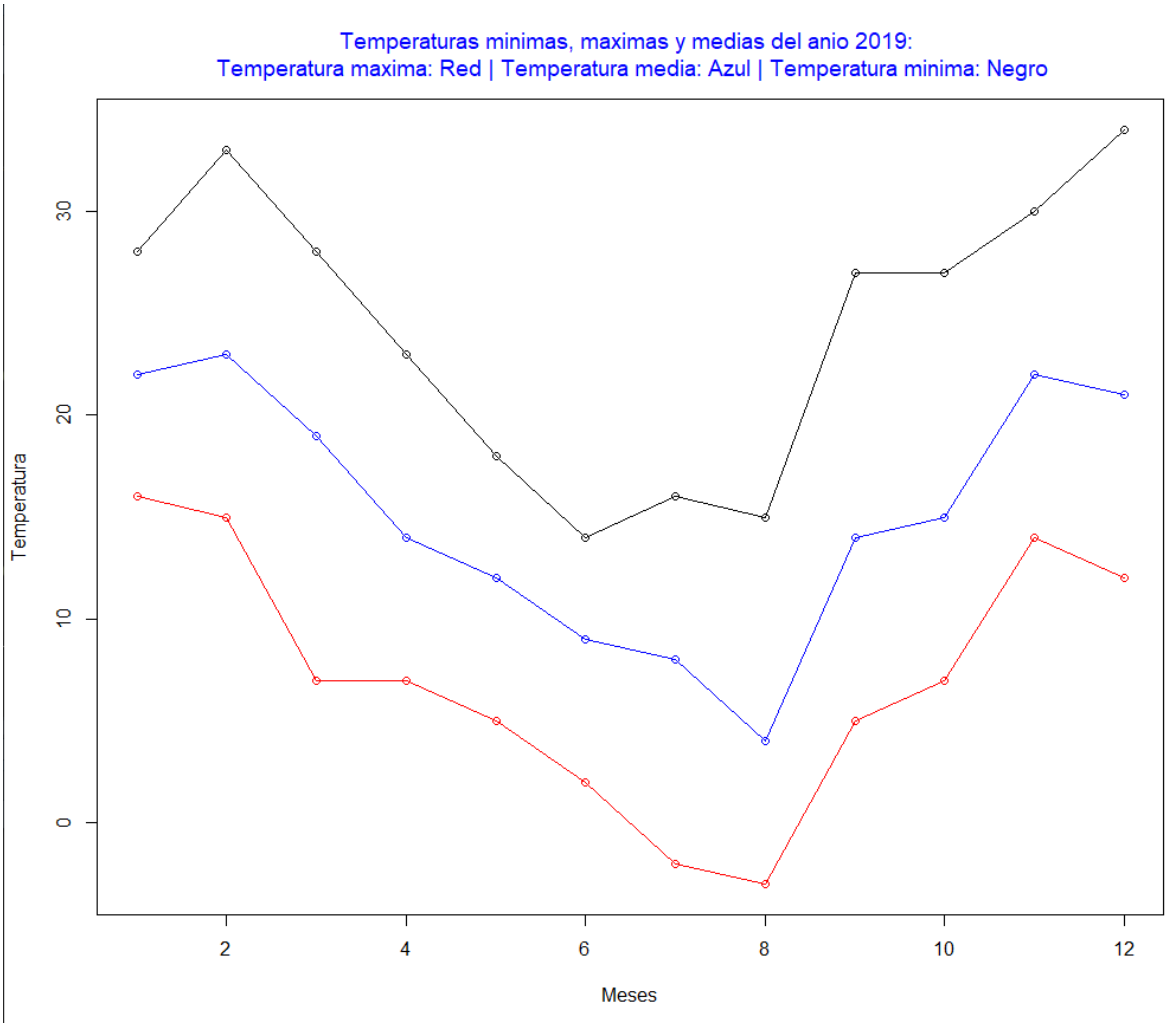
1.4 Combinar todos los data frames en uno.

Realiza una combinación de todos los datos para crear un DataFrame único con la información de todos los años.”
Para realizar este ejercicio utilizamos la función que provee R llamada rbind(df1 , df2), la cual combina dos objetos por filas, es decir, pone el segundo objeto pasado como parámetro debajo del primer objeto. Esto dentro de un bucle for por cada data frame en la lista que contiene los mismos.

2. Análisis Preliminar de los Datos

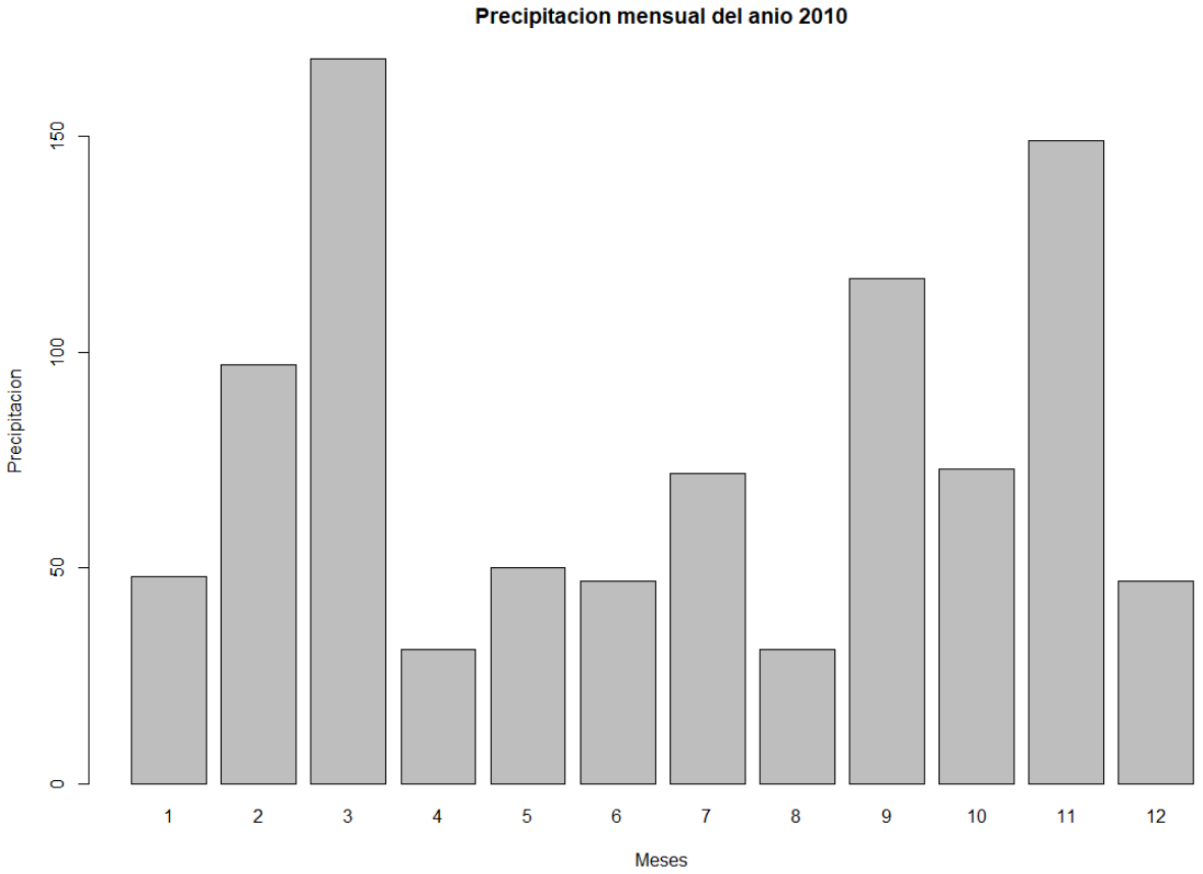
En esta sección, llevarás a cabo un análisis preliminar del clima en Tandil desde 2010 a la fecha. Para realizar cada uno de los gráficos se utilizó la función plot(), la cual nos permite realizar gráficas indicando su título, nombre de ejes, tipo de gráfico, límites máximos y mínimos de los ejes, etc.

1. Crea un gráfico de líneas que muestre la temperatura mensual máxima, mínima y media del año 2019.



En este gráfico podemos observar que las temperaturas mas altas fueron a principios de año y a fin de año y a mitad de año se mantienen temperaturas bajas.

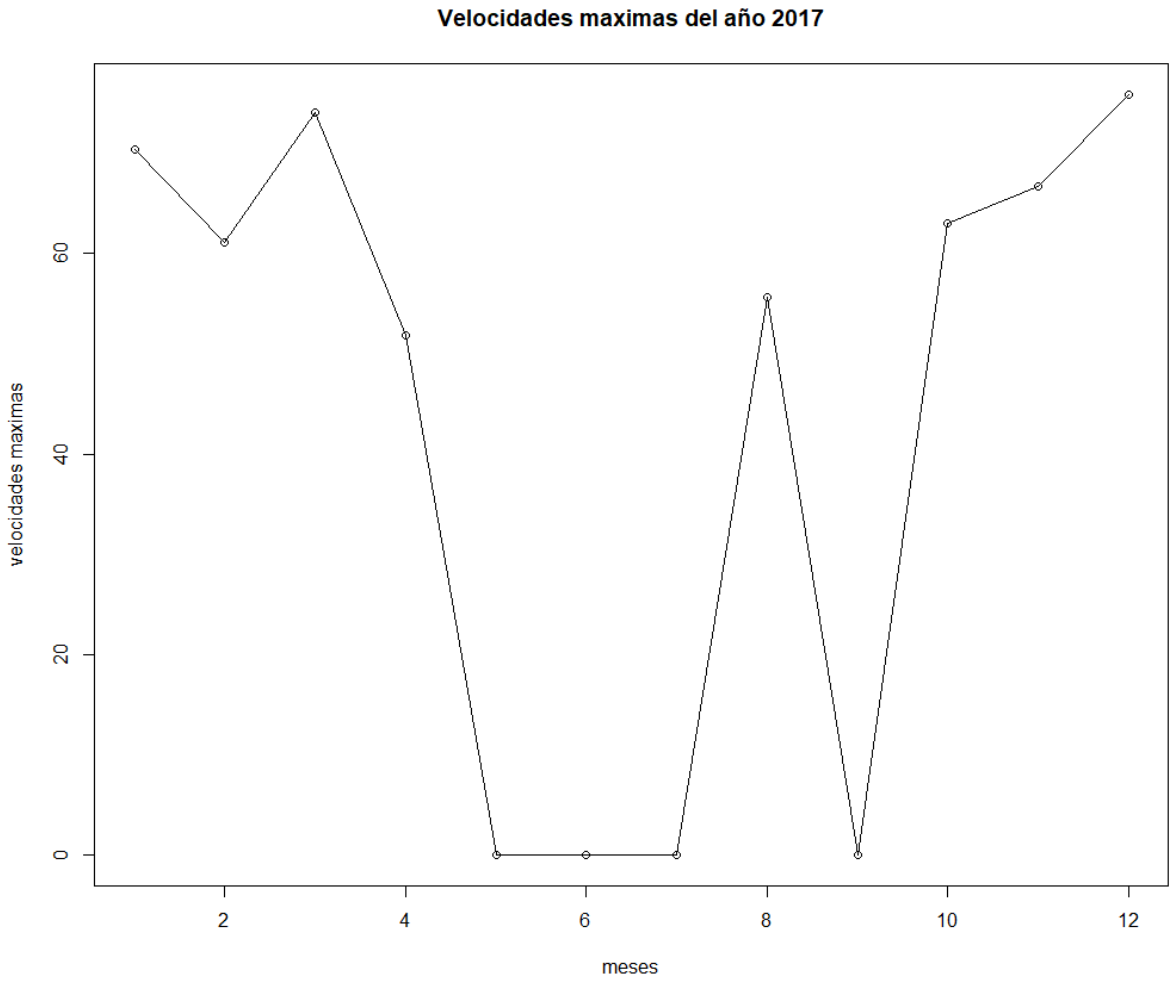
2. Crea un gráfico de barras que muestre la precipitación mensual acumulada durante el año 2010.



En este gráfico de barras vemos que en el mes de Marzo, Septiembre y Noviembre se mantuvo una precipitación de más de 100, luego el resto de meses debajo del 100.

3. Crea un gráfico de líneas que muestre la velocidad máxima mensual durante el año 2017.

Para este ejercicio se agregó una nueva limpieza de datos sobre el DataFrame del año 2017 debido a que el mismo contenía datos faltantes (NA) y no se podían graficar dichos valores a parte de que nosotros obtenemos los maximos y minimos para realizar una grafica mas exacta y acuerdo a los datos, a estos datos NA se los reemplazó con un 0.

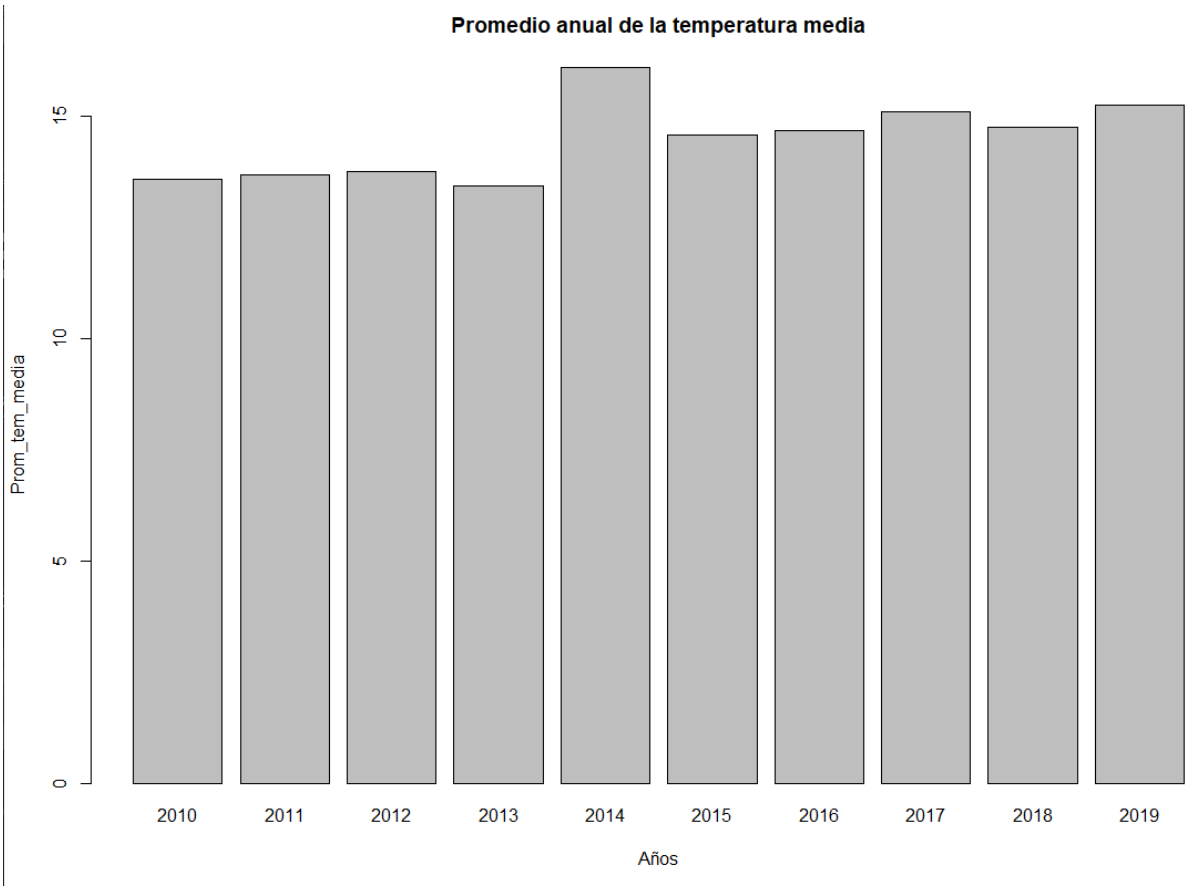


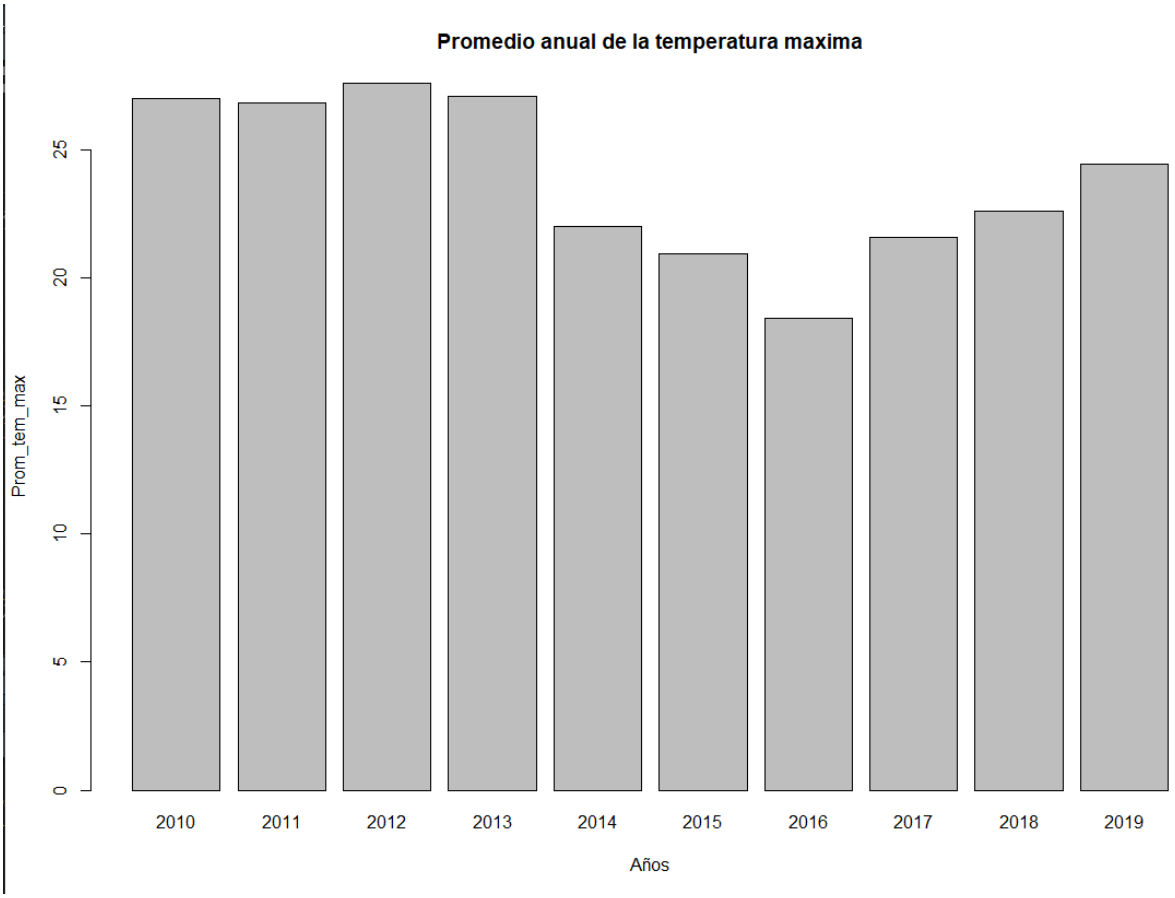
En este gráfico también observamos que en los primeros meses y en los últimos es cuando más velocidades máximas obtuvimos, aunque de todas formas en el mes de Agosto tenemos una velocidad máxima, no es el pico extremo del año.

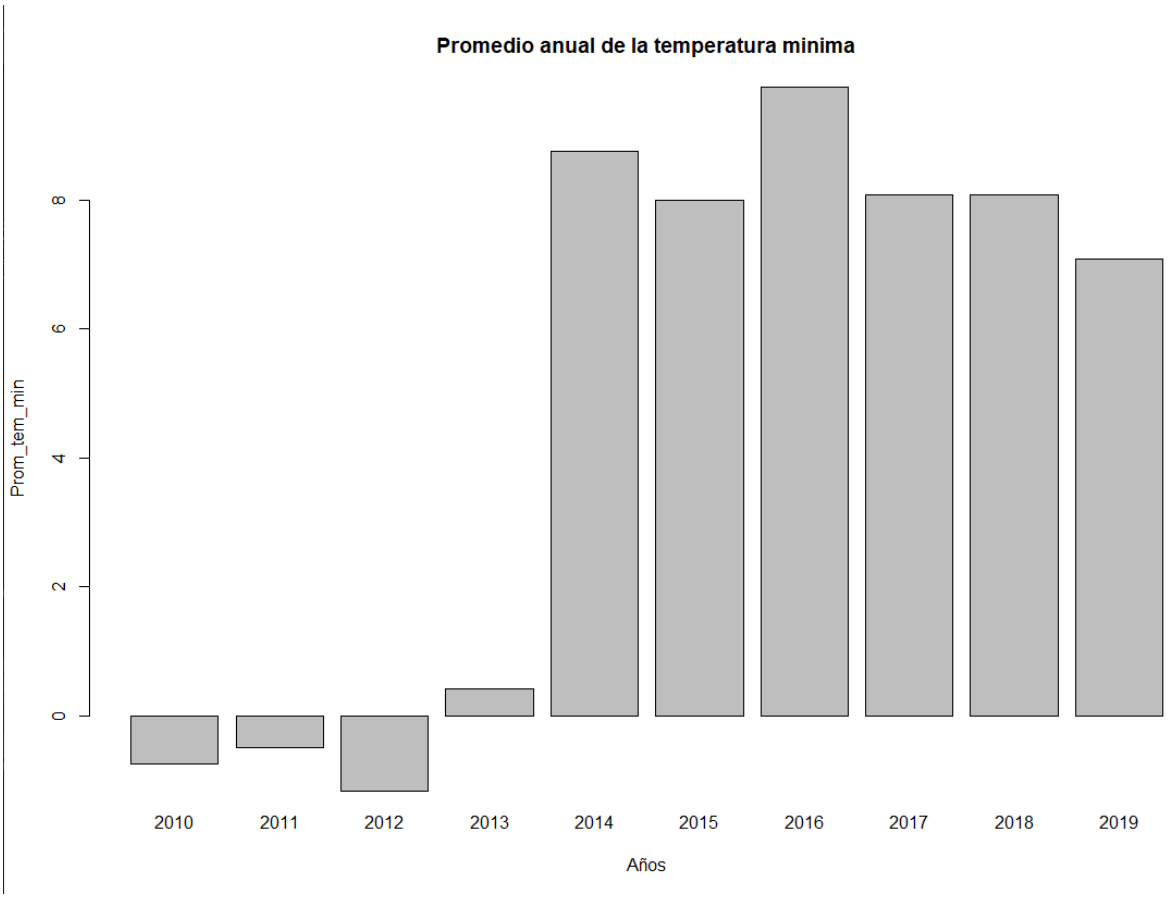
4. Crea un DataFrame que contenga la temperatura promedio anual (máxima, mínima y media) desde 2010 a 2019.

Para realizar este ejercicio, se recorrio la lista de dataFrames que tenemos de los climas y se fue cargando el promedio anual de las temperaturas máxima, mínima y media desde el año 2010 al 2019 en un nuevo data Frame llamado datos_clima_2010_2019 al cual se le agregó una nueva columna llamada Año la cual va desde el año 2010 al 2019.

5. Crea un gráfico de barras que muestre las temperaturas promedio anuales calculadas en el Paso 4.


Promedio anual de la temperatura media
Prom_tem_media
0
5
10
15
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Años


Promedio anual de la temperatura maxima
Prom_tem_max
0
5
10
15
20
25
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Años



En estos gráficos de barra vemos que las temperaturas mínimas máximas se dieron en los años 2010 al 2013, luego de eso, la temperatura mínima era mayor a 6.
Si vemos el gráfico de las temperaturas medias, se mantiene entre 17 y 35 en todos los años.
Luego en la gráfica de las temperaturas máximas observamos que la temperatura más alta fue en el año 2014.

6. Crea una función que te permita crear los gráficos de los Pasos 3 al 4, pasando como parámetros, el año, el conjunto de datos, el tipo de gráfico y el título del gráfico.

Para realizar este ejercicio, creamos una función con los siguientes parámetros:
(list_datos_clima , titulo_grafico = "gráfico" , tipo_grafico = 1 , operacion = 1 ,
anio = 2010 , anioFin = 3000)

Descripción de los parámetros:

list_datos_clima: es una lista que contiene los data frames de cada año requerido
anio: si la variable operación tiene valor 1, se utilizará la variable “anio” para ver en ese año las velocidades de los climas, si la operación tiene valor 2, se la utilizara como el inferior de fechas sobre el promedio de los años.
anioFin: rango máximo de año de las fechas a buscar por la operación 2.
tipo_grafico: si tiene valor 1 se realizará un gráfico de línea, 2 para realizar gráfico de barra

titulo_grafico: es el título del gráfico
operacion: si vale 1 se realizará la gráfica para las velocidades máximas, en cambio si el valor de la operación es 2 buscaremos en un rango de fechas los promedios de las distintas temperaturas anuales.

Casi todos los parámetros tienen valor por defecto en caso de que el usuario no los ingrese, menos la lista de dataFrames, la cual tendrá que contener la información de todos los años con sus tablas cargadas de la siguiente manera:(nos referimos a los nombres de las columnas y el orden)

	MES	T. MEDIA	T. MÁX	T. MÍN	V. MEDIA VIENTO	RACHAS MÁX	PRESIÓN MEDIA	LLUVIA
1	1	21 °C	33 °C	7 °C	13 km/h	-- km/h	1010.6 hPa	189 mm
2	2	19 °C	32 °C	5 °C	12.5 km/h	-- km/h	1015.8 hPa	53 mm
3	3	17 °C	31 °C	2 °C	10.1 km/h	-- km/h	1015.3 hPa	24 mm
4	4	14 °C	29 °C	-2 °C	11.2 km/h	-- km/h	1014.9 hPa	59 mm
5	5	10 °C	23 °C	-3 °C	11.6 km/h	-- km/h	1019.8 hPa	46 mm
6	6	7 °C	17 °C	-2 °C	10.7 km/h	-- km/h	1017.1 hPa	33 mm
7	7	7 °C	21 °C	-8 °C	12.4 km/h	-- km/h	1018.4 hPa	71 mm
8	8	8 °C	20 °C	-5 °C	15.9 km/h	-- km/h	1019.1 hPa	35 mm
9	9	11 °C	24 °C	-4 °C	13.4 km/h	-- km/h	1019.4 hPa	65 mm
10	10	13 °C	26 °C	0 °C	15 km/h	-- km/h	1016.2 hPa	27 mm
11	11	18 °C	31 °C	2 °C	13.9 km/h	-- km/h	1014.1 hPa	174 mm
12	12	19 °C	35 °C	2 °C	14.8 km/h	-- km/h	1014.1 hPa	62 mm

dejamos este formato debido a los datos dados por la cátedra.
Dejamos algunos ejemplos de el uso de la función en el código:

```
357  ##  EJEMPLOS DE LA FUNCION DEL 2.6
358  #ejemplo de funcion en linea y barra de el ejercicio 3 con funcion
359  funcion_clima(list_df,"grafico de linea operacion 1",1,1,2017)
360  funcion_clima(list_df,"grafico de barra operacion 1",2,1,2017)
361
362  ##ejemplo de las funciones de linea y barra del ejercicio 2.5 con la funcion
363  funcion_clima(list_df,"grafico de linea operacion 2",1,2,2012,2019)
364  funcion_clima(list_df,"grafico de barra operacion 2",2,2,2012,2019)
365
```

3. Creación de un Modelo para el Pronóstico del Clima

En esta sección, trabajamos en la creación de diversos modelos que te permitirán hacer un pronóstico del clima en un próximo mes.

Interpretamos para los incisos siguientes que para realizar un pronóstico del clima, era posible únicamente pronosticar el atributo Temperatura Media ‘T.Media’

1. Utilizando el histórico de datos de clima, elabora un programa en R que implemente una regresión lineal simple y te permita calcular el clima del día siguiente.

Para realizar este pronóstico utilizamos el siguiente y breve código:

```
regresion1 <- lm(get('T. MEDIA')~MES,datos_clima)
print(regresion1)
# Given Coefficients: (Intercept - MES) : 16.8787      -0.3583

getTemp <- function( mes = 1 ){
  return (16.8787 - 0.3583*mes)}
print(paste("La temperatura media en el próximo mes será: ",getTemp(7)))
```

Indicando en la función de regresión que buscamos el cálculo de la variable T.MEDIA en función del MES, que lo pasaremos por parámetro.

2. A partir del histórico de datos de clima, codifica un Modelo Monte Carlo que te permita calcular el clima en ventanas de 1, 2, 5 y 10 días.

Para la construcción de un modelo de montecarlo en este inciso, se interpretó que lo solicitado es la distribución de probabilidades del clima, basado en la temperatura media de el dataframe obtenido.

Se utilizó la librería “DisimForMixed” para calcular automáticamente las probabilidades condicionales.
Presentamos el siguiente pseudocódigo explicando lo realizado para este inciso:

```
size <- length(datos_clima$`T. MEDIA`) # cdad total de datos
#llevo a un dataframe todas las T Medias en mis datos:
dfAux <- as.data.frame(datos_clima$`T. MEDIA`)

distribucion <- dfAux$Freq/size
dfAux <- unir_filas(dfAux, distribucion)
```

```
dado <- datos_clima$`T. MEDIA` [1:(size-1)]
resultado <- datos_clima$`T. MEDIA`[2:size]
data_Condicional <- data.frame(dado, resultado)

#A partir de los datos y el tiempo en el dia siguiente, calculamos el vector de probabilidades
#condicionales, y la matriz para montecarlo
vecCond <- calcularProbCondicional(data_Condicional)
matCond <- matrix(vecCond$condProbVal,nrow = 23,ncol = 23)

#Por ultimos implementamos la simulacion montecarlo y los metodos sigDadoAnterior yconverge

simulacionMC <- function(matAcum, inicial = 1, n_dias = 10, MinTries = 100){
  muestras <- 0
  vecAnt <- replicate(nrow(matAcum),1)
  vecAct <- replicate(nrow(matAcum),0)
  salidas <- replicate(nrow(matAcum),0)
  while ((!converge(vecAct, vecAnt)) && muestras < MinTries){
    actual <- as.integer(inicial)
    muestras <- muestras + 1
    for (i in 1:n_dias){
      actual = sigDadoAnterior(matAcum, actual)
    }
    salidas[actual] <- salidas[actual] + 1
    vecAnt <- vecAct
    vecAct <- salidas/muestras
  }
  print(paste('muestras tomadas: ', muestras))
  vecAct
}
```

END

Cabe aclarar que los métodos sigDadoAnterior() y converge() no fueron incluidos ya que:

1. El primero únicamente retorna un siguiente valor del clima dado el anterior, usando como parámetro la matriz de probabilidades calculada.
2. El segundo chequea la convergencia del algoritmo a partir de la diferencia de la probabilidad actual y la probabilidad anterior, con un valor de aceptación epsilon.

De forma análoga a lo explicado, se podría plantear una predicción para los demás parámetros del clima en Tandil, y no solo basados en la temperatura.

3. Crea una red neuronal simple, que teniendo como parámetros de entrada:

La temperatura media, temperatura máxima, temperatura mínima, velocidad media del viento, velocidad máxima del viento y la presión media...
Nos permita calcular el clima del mes siguiente.

Para realizar este enunciado se empezo eliminando los espacios de los nombres de las columnas con el siguiente fragmento de codigo:

```
## modificacion de nombres de dataFrames quitando los espacios
nombres <- names(datos_nn)
for(i in 1:length(nombres)){
  nombres[i] <- gsub(" ", "", nombres[i])
}
#asigno los nuevos nombres de las columnas
names(datos_nn) <- nombres
```

Luego se rellenan los valores faltantes de cada fila de el DataSet con el promedio propio de cada columna

```
# se llenan los valores faltantes con un promedio propio de su columna.
# (Solo las columnas de rachas_max tienen NA's)
datos_nn$`RACHASMÁX`
mean(datos_nn$`RACHASMÁX`, na.rm=TRUE)
datos_nn$`RACHASMÁX`[is.na(datos_nn$`RACHASMÁX`)] <- mean(datos_nn$`RACHASMÁX`, na.r

# ademas se elimina la columna lluvia y mes ya que no son necesarias
datos_nn$LLUVIA <- NULL
```

Generamos filas aleatorias desde el dataSet para realizar pruebas sobre la Red neuronal guardados en la variable test utilizando la función sample.

```
#genero un numero aleatorio y agarro la columna perteneciente al mismo para el entre
rand_index <- sample (1:nrow (datos_nn), round(0.75*nrow(datos_nn)))
train <- datos_nn [rand_index, ]
# y utilizo el opuesto para el testeo
test <- datos_nn [-rand_index, ]
#
```

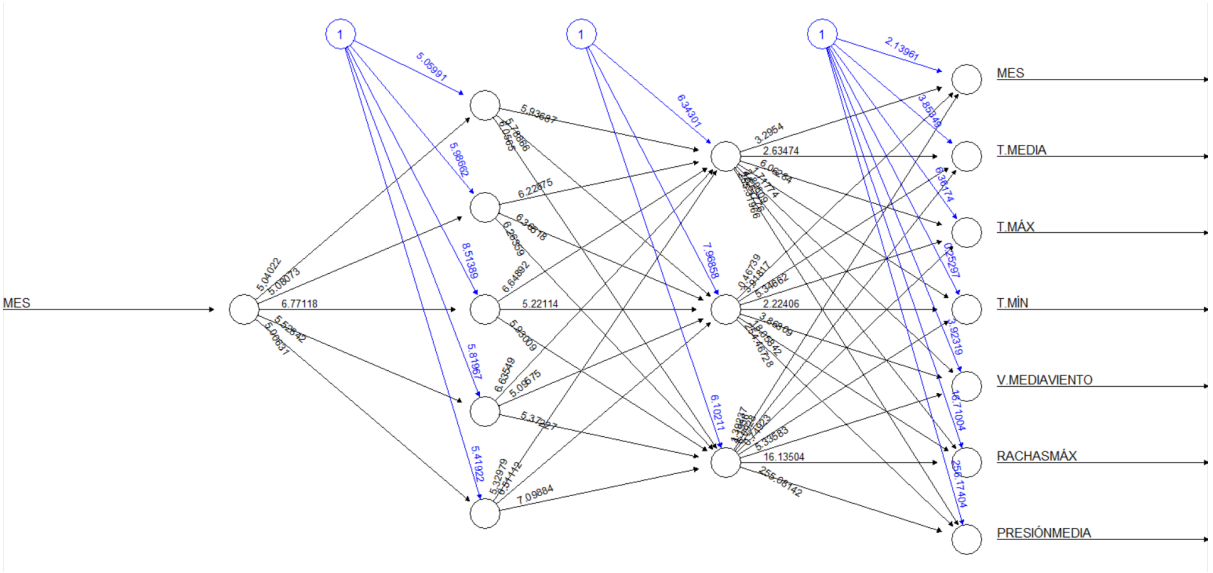
Otro tipo de testeo para realizar pruebas sobre la red neuronal:

```
#metodo maxmin elegido, con una escala de datos [0,1] para la red neuronal
maxs <- apply (datos_nn, 2, max)
mins <- apply (datos_nn, 2, min)
scaled <- as.data.frame (scale (datos_nn, center = mins, scale = maxs - mins))
train_ <- scaled [rand_index, ]
test_ <- scaled [-rand_index, ]
#red neuronal
library (neuralnet)
require(neuralnet)
```

creación de la red neuronal con los datos pedidos en el enunciado:
nn <-
neuralnet((MES+T.MEDIA+T.MÁX+T.MÍN+V.MEDIAVIENTO+RACHASMÁX+PRESIÓNMEDIA) ~ MES, data=datos_nn, hidden = c(5,3), linear.output = TRUE)

Comentario sobre lo pedido: No se entendía el hecho de poder obtener el clima del mes siguiente, en este caso puntual de las redes neuronales, tenemos un dato de entrada que seria el Mes, con los datos de salida de ese mes, suponiendo que como dato de entrada va a recibir el mes siguiente y no el mes actual.

Gráfico de la red neuronal:



En nuestro caso elegimos una red neuronal que tiene 5 neuronas en la primer capa y 3 neuronas en la segunda capa.

Para la predicción de datos, creamos un vector con los datos de meses desde el dataSet Test creado para realizar pruebas

```
##prediccion realiada basandonos en test
vMes <- as.data.frame(test$MES)
names(vMes) <- "MES"
vMes
predict = compute(nn,vMes)
result <- predict$net.result
class(result)
colnames(result) <- c("MES","T.MEDIA","T.MÁX","T.MÍN","V.MEDIAVIENTO","RACHASMÁX","PRESIÓNMEDIA")
result
```

Luego los datos obtenidos los cargamos en result cambiando los nombres de las columnas para que sea más representativo.

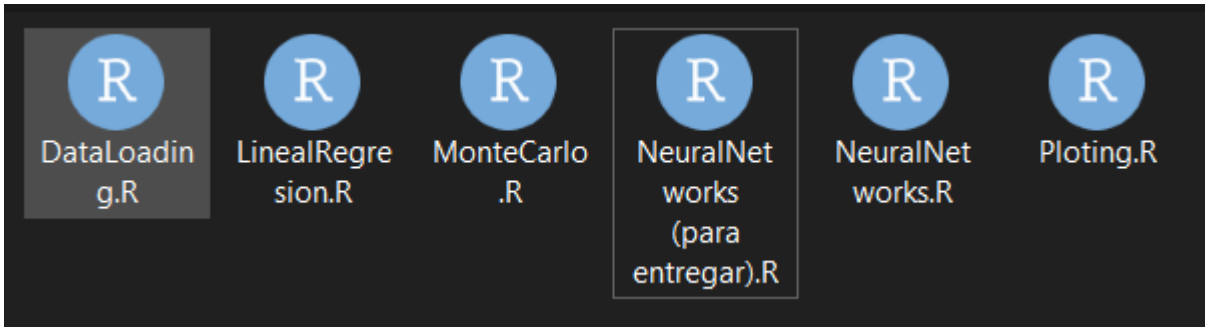
Datos generados con la red neuronal:

```
> result
       MES T.MEDIA  T.MÁX T.MÍN V.MEDIAVIENTO RACHASMÁX PRESIÓNMEDIA
 [1,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [2,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [3,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [4,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [5,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [6,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [7,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [8,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
 [9,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[10,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[11,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[12,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[13,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[14,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[15,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[16,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[17,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[18,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[19,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[20,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[21,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[22,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[23,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[24,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[25,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[26,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[27,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[28,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[29,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
[30,] 6.36    14.6 23.864 4.944       15.9352  66.58125     1021.042
```

No pudimos encontrar el error al realizar la creación de la red neuronal para que no nos genere datos repetidos, intentamos agregar más neuronas por capa o muchas capas o incluso una capa sola con muchas o pocas neuronas.

Posible solución: Suponemos que puede ser la forma en la que construimos la red neuronal o por falta de datos.

Comentario para la ejecución de los códigos:
Separamos los códigos en distintos archivos para ser más organizados a la hora de ejecutar el código de cada inciso siendo así orden de los enunciados en los archivos:



Ejercicio 1:

- DataLoading.R

Ejercicio 2:

- Ploting.R

Ejercicio 3:

- LinealRegresion.R
- MonteCarlo.R
- NeuralNetwork.R
- NeuralNetWorks(para entregar).R: Este archivo ".R" se genero por errores de nombres de variables.

Fuentes:

- Para complementar lo visto sobre montecarlo y redes neuronales Data Science con R
- Para determinar qué expresión regular utilizar en el inciso 1.3, usamos referencias de este sitio web: Expresiones regulares en R
- El resto de material fue dado por la cátedra de Científicos de datos