

Tarea1: Análisis de Datos de Calidad del Aire

Pasos iniciales: Instalación de paquetes y carga de librerías

Se procede con la instalación de paquetes y librerías que serán necesarias para el desarrollo de este trabajo, usando el comando:

```
install.packages(c("mice", "factoextra", "gridExtra"))
```

Considerando que `mice` será utilizada para la imputación de datos, `factoextra` para análisis factoriales, y `gridExtra` para la disposición gráfica.

Luego se añaden a la carga de las librerías a usar en el código:

```
library("mice")
library("factoextra")
library("gridExtra")
library("tidyverse")
library("visdat")
library("dlookr")
library("e1071")
library("flextable")
library("inspectdf")
library("qqplotr")
library("ggpmisc")
library("fdth")
library("PerformanceAnalytics")
library("corrplot")
```

Carga de archivo y guardado en dataframe

Se inicia el trabajo de análisis de datos mediante la carga del archivo CSV, el cual se almacena en un dataframe llamado `air`, utilizando la función `read.csv()`.

```
air <- read.csv("/Users/nicolelastraquiroz/Documents/R studio/airq_dt.csv", header = T, sep = ",", dec = ",")
head(air)
```

```
##   Ozono RadSol Vient Temp Mes Dia
## 1    21    259  15.5   76   9  12
## 2    32     92  12.0   61   5  24
## 3    20     37   9.2   65   6  18
## 4    NA    139   8.6   82   7  11
## 5    28    273  11.5   82   8  13
## 6    NA    266  14.9   58   5  26
```

Con esto ya se pueden apreciar algunas partes de la estructura básica del dataframe, como nombres de columnas y algunas filas de datos, sin embargo, en la búsqueda de lo específico y prolijo, a continuación se procederá a hacer un análisis con más detalle.

Análisis de la estructura del data frame

```
dim(air)
```

```
## [1] 100  6
```

```
str(air)
```

```
## 'data.frame':  100 obs. of  6 variables:
## $ Ozono : int  21 32 20 NA 28 NA NA 14 89 19 ...
## $ RadSol: int  259 92 37 139 273 266 98 334 229 99 ...
## $ Vient : num  15.5 12 9.2 8.6 11.5 14.9 11.5 11.5 10.3 13.8 ...
## $ Temp  : int  76 61 65 82 82 58 80 64 90 59 ...
## $ Mes   : int  9 5 6 7 8 5 6 5 8 5 ...
## $ Dia   : int  12 24 18 11 13 26 28 16 8 8 ...
```

En general, se puede observar que el dataframe air contiene 100 observaciones (filas), correspondientes a 6 variables (columnas) del tipo numéricas (5 int y 1 num para ser específicos).

Estudio de valores faltantes

Es importante destacar que la presencia de NA pueden interferir con los análisis posteriores, por lo que no se pueden dejar de considerar al momento de realizar estudios y observaciones primarias del dataframe, de ahí la importancia de generar imputaciones coherentes y adecuadas al set de datos trabajado.

En este caso, si bien ya se veían valores faltantes (NA) en la revisión anterior del dataframe, ahora se utilizará `summary()` para obtener un conteo rápido y verificar si hay valores nulos en el dataset (considerando que inicialmente se toman las 6 variables como numéricas, tal y como se pudo ver en la revisión anterior).

```
summary(air)
```

```
##      Ozono      RadSol      Vient      Temp
## Min.   : 1.00   Min.   : 7.00   Min.   : 2.30   Min.   :56.00
## 1st Qu.:16.00   1st Qu.: 98.75   1st Qu.: 7.40   1st Qu.:72.00
## Median :29.00   Median :209.50   Median :10.30   Median :78.00
## Mean   :41.07   Mean   :185.48   Mean   :10.21   Mean   :77.52
## 3rd Qu.:51.50   3rd Qu.:259.00   3rd Qu.:12.15   3rd Qu.:84.25
## Max.   :168.00   Max.   :334.00   Max.   :20.70   Max.   :96.00
## NA's   :18      NA's    :4
##      Mes      Dia
## Min.   :5.00   Min.   : 1.00
## 1st Qu.:6.00   1st Qu.: 9.00
## Median :7.00   Median :15.00
## Mean   :7.06   Mean   :15.39
## 3rd Qu.:8.00   3rd Qu.:22.00
## Max.   :9.00   Max.   :31.00
##
```

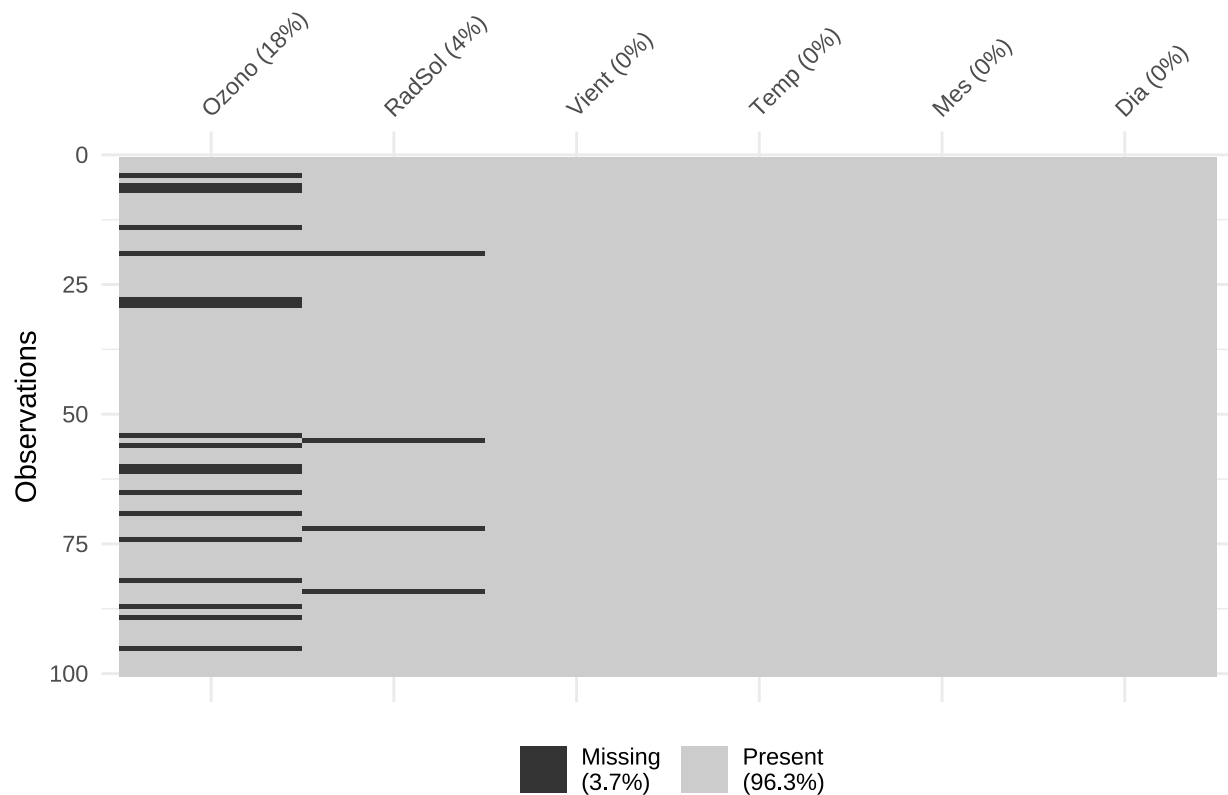
Como complemento a esto, y considerando otras formas que también vimos en clases, se añade una revisión detallada usando el conteo por columna de los NA con `colSums(is.na())`.

```
colSums(is.na(air))
```

```
##  Ozono RadSol  Vient  Temp  Mes  Dia  
##   18     4     0     0    0    0
```

Y también una visualización gráfica de los valores nulos con `visdat::vis_miss()`.

```
visdat::vis_miss(air)
```



De los 3 métodos expuestos, se puede afirmar que sí existen columnas con datos NA, estas son: **Ozono** y **RadSol**, con 18 y 4 valores faltantes respectivamente, lo que hace un total de un 3.7% de datos faltantes versus un 96,3% de datos válidos.

Imputación de valores faltantes

Los valores faltantes se podrían imputar usando la media de cada variable, ya que esta es una estrategia simple y efectiva cuando el porcentaje de datos faltantes es bajo, como en este caso, sin embargo, y considerando aplicar al máximo lo visto en clases, utilizaremos la librería `mice`: *Multivariable Imputation by Chained Equations*, que ofrece un método destacado y altamente efectivo para la mayoría de los casos a imputar.

```
imp_dt <- mice(air, m = 1, seed = 100)
```

```
##
## iter imp variable
## 1 1 Ozono RadSol
## 2 1 Ozono RadSol
## 3 1 Ozono RadSol
## 4 1 Ozono RadSol
## 5 1 Ozono RadSol
```

```
air_impt <- complete(imp_dt, 1)
```

El resultado de la imputación fue almacenado en un nuevo dataframe denominado `air_impt`.

Verificación de valores faltantes

Para descartar cualquier posible error humano en el proceso de imputación (error de código, mal guardado de los datos imputados, entre otros), se revisará nuevamente la presencia de NA en la data y se generará una breve comparativa.

```
colSums(is.na(air_impt))
```

```
## Ozono RadSol Vient Temp Mes Dia
## 0 0 0 0 0 0
```

Antes de la imputación:

```
str(air)
```

```
## 'data.frame': 100 obs. of 6 variables:
## $ Ozono : int 21 32 20 NA 28 NA NA 14 89 19 ...
## $ RadSol: int 259 92 37 139 273 266 98 334 229 99 ...
## $ Vient : num 15.5 12 9.2 8.6 11.5 14.9 11.5 11.5 10.3 13.8 ...
## $ Temp : int 76 61 65 82 82 58 80 64 90 59 ...
## $ Mes : int 9 5 6 7 8 5 6 5 8 5 ...
## $ Dia : int 12 24 18 11 13 26 28 16 8 8 ...
```

Después de la imputación:

```
str(air_impt)
```

```
## 'data.frame': 100 obs. of 6 variables:
## $ Ozono : int 21 32 20 16 28 19 28 14 89 19 ...
## $ RadSol: int 259 92 37 139 273 266 98 334 229 99 ...
## $ Vient : num 15.5 12 9.2 8.6 11.5 14.9 11.5 11.5 10.3 13.8 ...
## $ Temp : int 76 61 65 82 82 58 80 64 90 59 ...
## $ Mes : int 9 5 6 7 8 5 6 5 8 5 ...
## $ Dia : int 12 24 18 11 13 26 28 16 8 8 ...
```

En síntesis, se confirma que no existen datos faltantes posterior al proceso de imputación, por lo que ahora se podrá implementar el resto de los análisis requeridos.

Diagnóstico completo de las variables categóricas.

Aunque el dataset parece contener principalmente variables numéricas, Mes y Día pueden tratarse como variables categóricas dado su contexto, debido a esto, en este caso primero las transformaremos a categorías, y luego las analizaremos como tal.

Transformación de las variables Mes y Dia a categóricas.

```
air_impt$Mes<-factor(air_impt$Mes)
air_impt$Dia<-factor(air_impt$Dia)
```

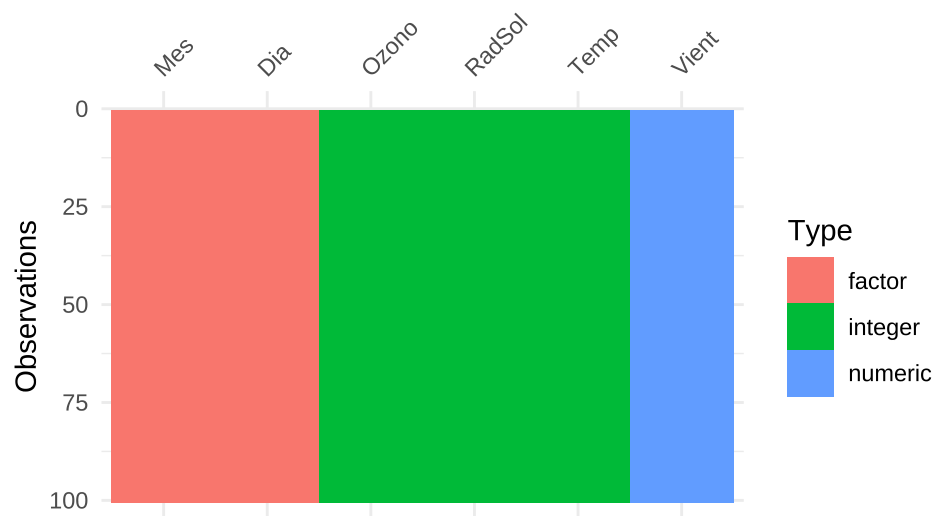
Revisión de la transformación

```
str(air_impt)
```

```
## 'data.frame':    100 obs. of  6 variables:
## $ Ozono : int  21 32 20 16 28 19 28 14 89 19 ...
## $ RadSol: int  259 92 37 139 273 266 98 334 229 99 ...
## $ Vient : num  15.5 12 9.2 8.6 11.5 14.9 11.5 11.5 10.3 13.8 ...
## $ Temp  : int  76 61 65 82 82 58 80 64 90 59 ...
## $ Mes   : Factor w/ 5 levels "5","6","7","8",...: 5 1 2 3 4 1 2 1 4 1 ...
## $ Dia   : Factor w/ 31 levels "1","2","3","4",...: 12 24 18 11 13 26 28 16 8 8 ...
```

Visualiación gráfica de los tipos de datos

```
visdat::vis_dat(air_impt,sort_type = TRUE)
```



Análisis de las variables categóricas

```
ft1 <- flextable(diagnose_category(air_impt))
ft1 <- set_caption(ft1, caption = "Detalle de variables categóricas")
ft1
```

Table 1: Detalle de variables categóricas

variables	levels	N	freq	ratio	rank
Mes	8	100	24	24	1
Mes	5	100	22	22	2
Mes	9	100	21	21	3
Mes	7	100	17	17	4
Mes	6	100	16	16	5
Dia	2	100	5	5	1
Dia	9	100	5	5	1
Dia	14	100	5	5	1
Dia	29	100	5	5	1
Dia	4	100	4	4	5
Dia	7	100	4	4	5
Dia	10	100	4	4	5
Dia	12	100	4	4	5
Dia	15	100	4	4	5
Dia	18	100	4	4	5

Estudio de normalidad en variables cuantitativas

```
numeric_vars <- select_if(air_impt, is.numeric)
summary(numeric_vars)
```

```
##      Ozono      RadSol      Vient      Temp
## Min.   : 1.00   Min.   : 7   Min.   : 2.30   Min.   :56.00
## 1st Qu.: 16.00  1st Qu.: 92   1st Qu.: 7.40   1st Qu.:72.00
## Median : 30.00  Median :204   Median :10.30   Median :78.00
## Mean   : 42.01  Mean   :182   Mean   :10.21   Mean   :77.52
## 3rd Qu.: 64.25  3rd Qu.:259   3rd Qu.:12.15   3rd Qu.:84.25
## Max.   :168.00  Max.   :334   Max.   :20.70   Max.   :96.00
```

Análisis por método gráfico

Histograma de frecuencias con abline de media y mediana

Para analizar las variables numéricas (Ozono, RadSol, Vient, Temp), se utilizarán histogramas. Éstos ayudan a visualizar la distribución de los datos y permiten, desde lo visual, identificar si las variables siguen una distribución simétrica o están sesgadas.

```
plots <- list()

for (col in colnames(numeric_vars)) {
  mean_val <- mean(numeric_vars[[col]], na.rm = TRUE)
  median_val <- median(numeric_vars[[col]], na.rm = TRUE)

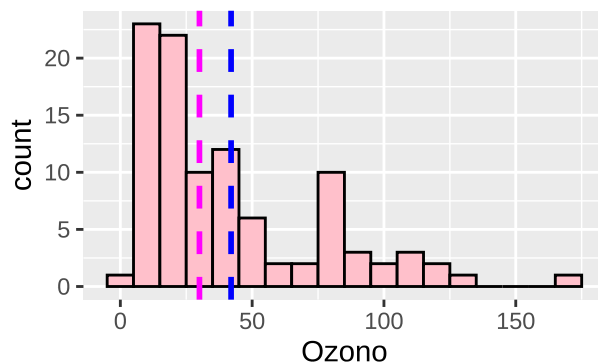
  p <- ggplot(numeric_vars, aes_string(x = col)) +
    geom_histogram(binwidth = 10, fill = "pink", color = "black") +
    geom_vline(xintercept = mean_val, color = "blue", linetype = "dashed", size = 1) +
    geom_vline(xintercept = median_val, color = "magenta", linetype = "dashed", size = 1) +
    ggtitle(paste("Histograma de", col))

  plots[[col]] <- p
}

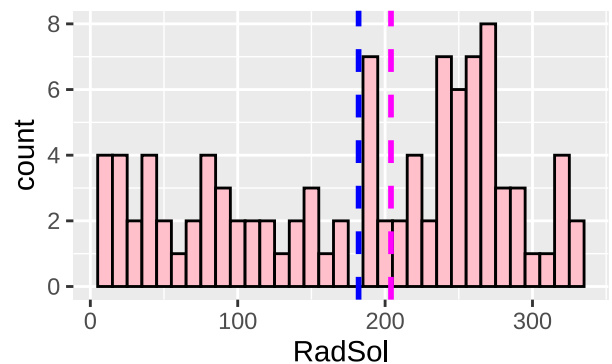
n_col <- 2

do.call("grid.arrange", c(plots, ncol = n_col))
```

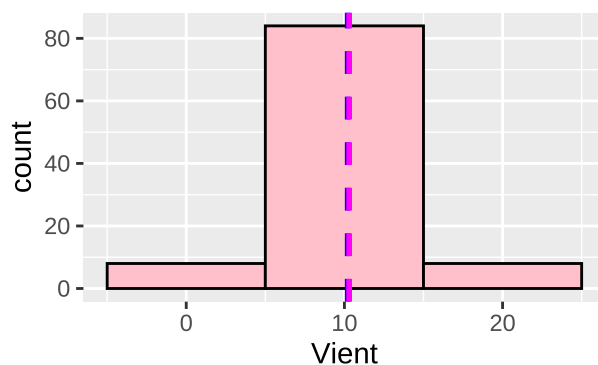
Histograma de Ozono



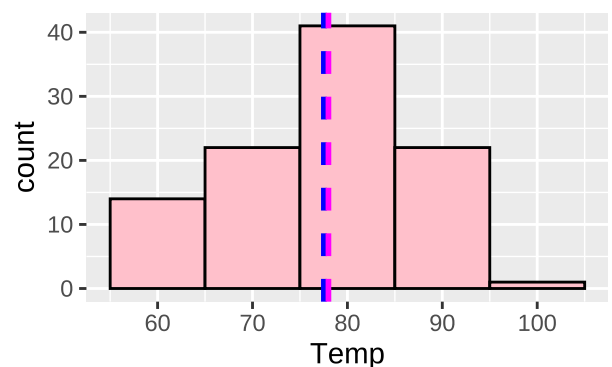
Histograma de RadSol



Histograma de Vient



Histograma de Temp



Además, se añadieron líneas verticales que indican la media (en azul) y la mediana (en magenta) para cada variable, lo que facilita la comparación entre ambas medidas y ayuda a identificar la presencia de asimetría. Se debe destacar, que si la media y la mediana están muy alejadas, es probable que la distribución sea asimétrica, como son el caso de la variables Ozono y RadSol.

Grafico de distribución de probabilidad normal

Primero usamos los recursos auxiliares aprendidos en clases para generar las tablas de frecuencias:

```
tb.freq <- function(x){
  f_i <- as.vector(table(x)) # freq absoluta
  F_i <- cumsum(f_i) # freq acumulada
  h_i <- f_i / length(x) # freq relativa
  H_i <- F_i / length(x) # freq relativa acumulada
  tf <- cbind(f_i, F_i, h_i, H_i)
  row.names(tf) <- names(table(x))
  tf
}
```

Luego, con la función ya creada, generamos las tablas de frecuencias usando sturges. También se definirán los intervalos, se generarán los cortes y agrupaciones y finalmente se visualizarán las tablas.

```
# Tablas de distribución de frecuencias
for (var_name in colnames(numeric_vars)) {
  print(paste("Tabla de distribución de frecuencias para:", var_name))
  tabla <- fdt(numeric_vars[[var_name]], breaks = "Sturges")
  print(tabla)

  # Calculo del intervalo
  intervalo <- (max(numeric_vars[[var_name]]) - min(numeric_vars[[var_name]])) / (1 + 3.322 * log10(nrow(numeric_vars)))
  print(paste("Intervalo calculado para", var_name, ":", intervalo))

  # Cortes y agrupación de los datos
  cut_var <- cut(numeric_vars[[var_name]], breaks = seq(min(numeric_vars[[var_name]]), max(numeric_vars[[var_name]]), length.out = 10))
  numeric_vars[[paste(var_name, "cut", sep = ".")]] <- cut_var

  # Tabla de frecuencias
  print(tb.freq(cut_var))
}
```

```
## [1] "Tabla de distribución de frecuencias para: Ozono"
##      Class limits  f   rf rf(%)  cf cf(%)
## [0.99,22.0762) 40 0.40    40  40    40
## [22.0762,43.1625) 23 0.23    23  63    63
## [43.1625,64.2488) 12 0.12    12  75    75
## [64.2488,85.335) 13 0.13    13  88    88
## [85.335,106.421)  5 0.05     5  93    93
## [106.421,127.507)  5 0.05     5  98    98
## [127.507,148.594)  1 0.01     1  99    99
## [148.594,169.68)  1 0.01     1 100   100
## [1] "Intervalo calculado para Ozono : 21.847200418629"
##      f_i F_i  h_i H_i
## (1,22.8]    39  39 0.39 0.39
```



```

## (22.8,44.7] 26 65 0.26 0.65
## (44.7,66.5] 11 76 0.11 0.76
## (66.5,88.4] 11 87 0.11 0.87
## (88.4,110] 6 93 0.06 0.93
## (110,132] 4 97 0.04 0.97
## (132,154] 1 98 0.01 0.98
## [1] "Tabla de distribución de frecuencias para: RadSol"
##      Class limits  f   rf rf(%)  cf cf(%)
##      [6.93,48.2312) 15 0.15    15 15   15
##      [48.2312,89.5325) 8 0.08    8 23   23
##      [89.5325,130.834) 9 0.09    9 32   32
##      [130.834,172.135) 7 0.07    7 39   39
##      [172.135,213.436) 13 0.13   13 52   52
##      [213.436,254.737) 18 0.18   18 70   70
##      [254.737,296.039) 22 0.22   22 92   92
##      [296.039,337.34) 8 0.08    8 100  100
## [1] "Intervalo calculado para RadSol : 42.7786499215071"
##      f_i F_i  h_i  H_i
## (7,49.8]    15 15 0.15 0.15
## (49.8,92.6] 10 25 0.10 0.25
## (92.6,135]  7 32 0.07 0.32
## (135,178]   8 40 0.08 0.40
## (178,221]  11 51 0.11 0.51
## (221,264]  25 76 0.25 0.76
## (264,306]  16 92 0.16 0.92
## [1] "Tabla de distribución de frecuencias para: Vient"
##      Class limits  f   rf rf(%)  cf cf(%)
##      [2.277,4.6057) 8 0.08    8 8    8
##      [4.6057,6.9345) 15 0.15   15 23   23
##      [6.9345,9.2632) 17 0.17   17 40   40
##      [9.2632,11.592) 32 0.32   32 72   72
##      [11.592,13.921) 10 0.10   10 82   82
##      [13.921,16.25) 13 0.13   13 95   95
##      [16.25,18.578) 3 0.03    3 98   98
##      [18.578,20.907) 2 0.02    2 100  100
## [1] "Intervalo calculado para Vient : 2.40711669283098"
##      f_i F_i  h_i  H_i
## (2.3,4.71]   7 7 0.07 0.07
## (4.71,7.11] 15 22 0.15 0.22
## (7.11,9.52] 17 39 0.17 0.39
## (9.52,11.9] 32 71 0.32 0.71
## (11.9,14.3] 15 86 0.15 0.86
## (14.3,16.7] 10 96 0.10 0.96
## (16.7,19.1] 1 97 0.01 0.97
## [1] "Tabla de distribución de frecuencias para: Temp"
##      Class limits  f   rf rf(%)  cf cf(%)
##      [55.44,60.63) 6 0.06    6 6    6
##      [60.63,65.82) 8 0.08    8 14   14
##      [65.82,71.01) 10 0.10   10 24   24
##      [71.01,76.2) 20 0.20   20 44   44
##      [76.2,81.39) 22 0.22   22 66   66
##      [81.39,86.58) 15 0.15   15 81   81
##      [86.58,91.77) 11 0.11   11 92   92
##      [91.77,96.96) 8 0.08    8 100  100

```

```
## [1] "Intervalo calculado para Temp : 5.23286237571952"
##           f_i F_i h_i H_i
## (56,61.2]    7   7 0.07 0.07
## (61.2,66.5]  6  13 0.06 0.13
## (66.5,71.7] 10  23 0.10 0.23
## (71.7,76.9] 20  43 0.20 0.43
## (76.9,82.2] 27  70 0.27 0.70
## (82.2,87.4] 13  83 0.13 0.83
## (87.4,92.6] 12  95 0.12 0.95
```

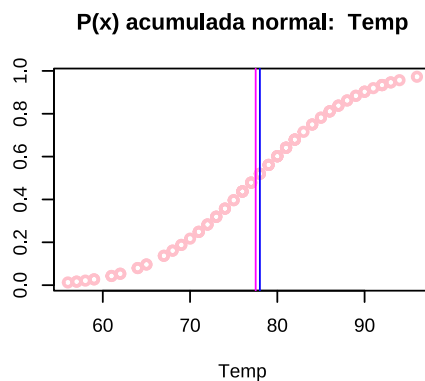
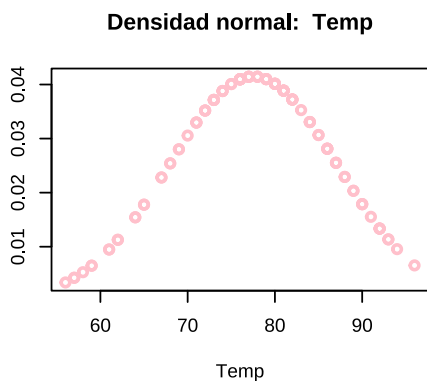
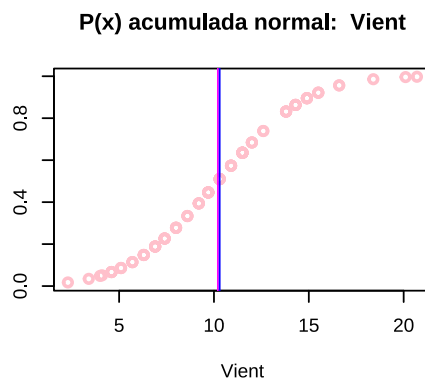
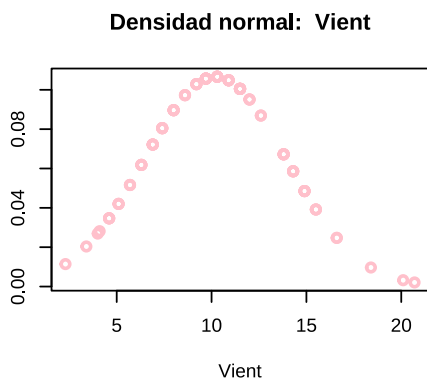
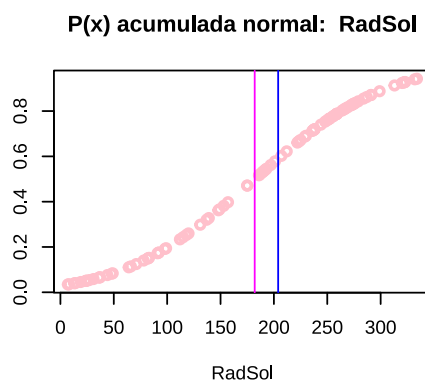
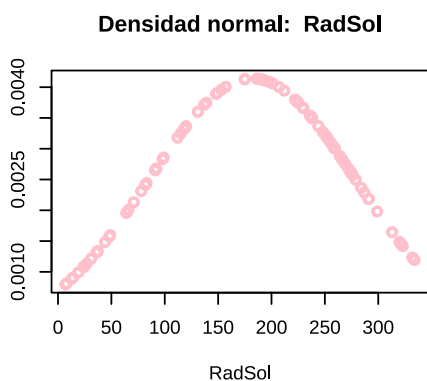
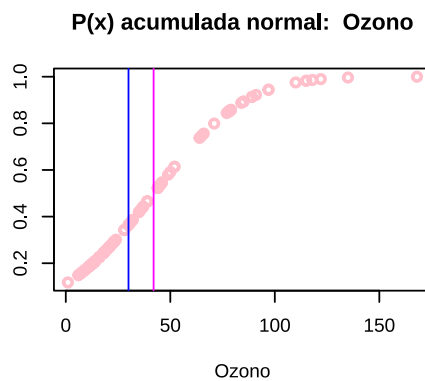
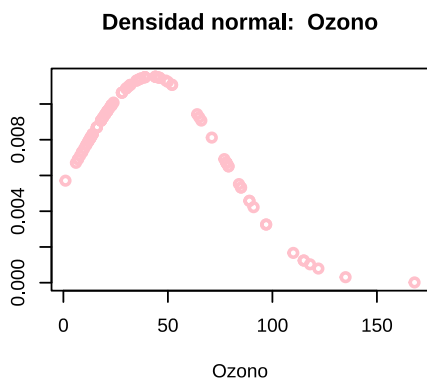
Finalmente se procederá con la generación de los gráficos de densidad normal y de probabilidad acumulada normal, para cada una de las variables numéricas.

```
par(mfrow = c(4, 2))
for (var_name in colnames(numeric_vars)[1:4]) {

# Gráfico de densidad vs distribución normal
plot(numeric_vars[[var_name]],
      dnorm(numeric_vars[[var_name]],
            mean = mean(numeric_vars[[var_name]]),
            na.rm = TRUE),
      sd = sd(numeric_vars[[var_name]],
              na.rm = TRUE)),
ylab = "", xlab = var_name,
lwd = 2, col = "pink", main = paste("Densidad normal: ", var_name))

# Gráfico de probabilidad acumulada vs distribución normal
plot(numeric_vars[[var_name]],
      pnorm(numeric_vars[[var_name]],
            mean = mean(numeric_vars[[var_name]]),
            na.rm = TRUE),
      sd = sd(numeric_vars[[var_name]],
              na.rm = TRUE)),
xlab = var_name, ylab = "",
lwd = 2, col = "pink",
main = paste("P(x) acumulada normal: ",
              var_name))

# Media y mediana como líneas verticales
abline(v = mean(numeric_vars[[var_name]]),
       na.rm = TRUE), col = "magenta")
abline(v = median(numeric_vars[[var_name]]),
       na.rm = TRUE), col = "blue")
}
```



Análisis por método matemático:

A continuación, se complementa el análisis gráfico con un enfoque matemático, para lo cual se definirán las hipótesis, se calculará la asimetría (*skewness*) y la curtosis (*kurtosis*) de cada variable.

Formulación de hipótesis

Primero se establece las respectivas hipótesis del caso:

- H_0 : La variable está distribuida normalmente.
- H_1 : La variable no está distribuida normalmente.

Se considera una significancia del 0.05, lo que implica que si el valor p es mayor a este valor, no se rechazará la hipótesis nula y, por lo tanto, se podría afirmar que la variable está distribuida normalmente.

Test de Shapiro-Wilk

se realiza la prueba de Shapiro-Wilk para evaluar la normalidad de cada variable. Esta prueba es estadísticamente robusta para tamaños de muestra pequeños, como es este caso.

```
for (col in colnames(air_impt)[1:4]) {  
  cat("\nVariable:", col)  
  cat("\nPrueba de Shapiro-Wilk:\n")  
  print(shapiro.test(air_impt[[col]]))  
}
```

```
##  
## Variable: Ozono  
## Prueba de Shapiro-Wilk:  
##  
## Shapiro-Wilk normality test  
##  
## data: air_impt[[col]]  
## W = 0.85664, p-value = 2.102e-08  
##  
##  
## Variable: RadSol  
## Prueba de Shapiro-Wilk:  
##  
## Shapiro-Wilk normality test  
##  
## data: air_impt[[col]]  
## W = 0.92599, p-value = 2.968e-05  
##  
##  
## Variable: Vient  
## Prueba de Shapiro-Wilk:  
##  
## Shapiro-Wilk normality test  
##  
## data: air_impt[[col]]
```

```
## W = 0.9834, p-value = 0.242
##
##
## Variable: Temp
## Prueba de Shapiro-Wilk:
##
## Shapiro-Wilk normality test
##
## data: air_impt[[col]]
## W = 0.97598, p-value = 0.06438
```

Resumen de normalidad para las variabes numéricas

```
ft2 <- flextable(normality(numeric_vars))
ft2 <- set_caption(ft2, caption = "Tabla de Normalidad por variables")

ft2
```

Table 2: Tabla de Normalidad por variables

vars	statistic	p_value	sample
Ozono	0.8566350	0.00000002101889	100
RadSol	0.9259890	0.00002968172350	100
Vient	0.9833954	0.24199785241031	100
Temp	0.9759791	0.06437900956539	100

Cesgo

Se debe tener en cuenta que `skewness` mide la simetría de la distribución. Los valores cercanos a 0 indicarán que la distribución de los datos es bastante simétrica.

```
for (col in colnames(air)[1:4]) {
  cat("\nVariable:", col)
  cat("\nSkewness:", skewness(air_impt[[col]], na.rm = TRUE))
}
```

```
##
## Variable: Ozono
## Skewness: 1.257036
## Variable: RadSol
## Skewness: -0.3720618
## Variable: Vient
## Skewness: 0.3007779
## Variable: Temp
## Skewness: -0.3287795
```

Ozono es la variable que mas presenta cesgo, considerando que posee un Skewness del orden de 1.26 aproximadamente. Las otras variables presentan cesgos bajos y en un orden parecido (al rededor del 0.3), sólo que para Vient es del tipo positivo y para RadSol y Temp es negativo.

Curtosis

La curtosis mide la “altura” y “anchura” de las colas de la distribución. Una curtosis positiva indica colas más pesadas que las de una distribución normal, mientras que una curtosis negativa indica colas más ligeras que las de una distribución normal.

Aquí falta añadir algo de datos bibliográficos que tengan cita en APA para decir los tipos de de curtosis si es menor a cero igual o mayor... onda platycurtica, mesocurtica y leptocurtica

```
for (col in colnames(air_impt)[1:4]) {  
  cat("\nVariable:", col)  
  cat("\nKurtosis:", kurtosis(air_impt[[col]], na.rm = TRUE))  
}
```

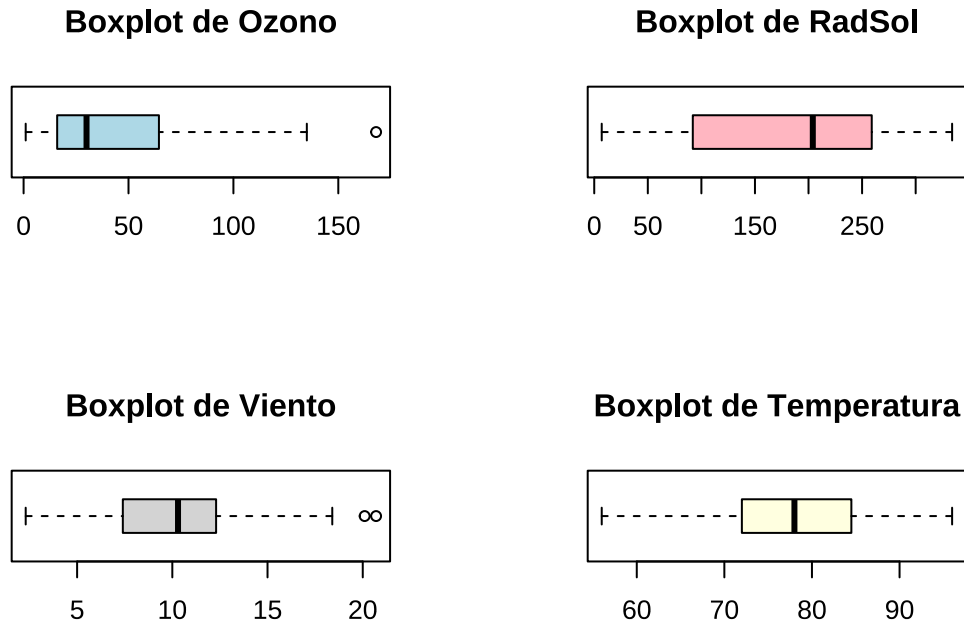
```
##  
## Variable: Ozono  
## Kurtosis: 1.053549  
## Variable: RadSol  
## Kurtosis: -1.161496  
## Variable: Vient  
## Kurtosis: -0.1396004  
## Variable: Temp  
## Kurtosis: -0.4922435
```

Se observan tendencias platycurticas en las variables RadSol, Vient y Temp, siendo la primera la que posee una tendencia más marcada que las otras, mientras que en el caso de Ozono su tendencia es del tipo leptocurtica.

Estudio de valores atípicos (outliers)

Los boxplots permiten identificar valores atípicos (outliers) que se representan como puntos fuera de los “bigotes” del gráfico.

```
par(mfrow = c(2, 2))  
boxplot(air_impt$Ozono, main = "Boxplot de Ozono", col = "lightblue", horizontal = TRUE)  
boxplot(air_impt$RadSol, main = "Boxplot de RadSol", col = "lightpink", horizontal = TRUE)  
boxplot(air_impt$Vient, main = "Boxplot de Viento", col = "lightgrey", horizontal = TRUE)  
boxplot(air_impt$Temp, main = "Boxplot de Temperatura", col = "lightyellow", horizontal = TRUE)
```



Mediante los diagramas “boxplot” se puede evidenciar visualmente la existencia de “outliers” en las variables Ozono y viento, pero para un estudio preciso, revisaremos el detalle de estos valores atípicos con la función `flextable(diagnose_outlier())`, como se muestra a continuación:

```
ft3 <- flextable::flextable(diagnose_outlier(air_impt))
ft3 <- set_table_properties(ft3, width = 0.75, layout = "autofit")
ft3 <- align(ft3, align = "center", part = "all")
ft3 <- set_caption(ft3, caption = "Detalles de valores atípicos en las variables")

ft3
```

Table 3: Detalles de valores atípicos en las variables

variables	outliers_cnt	outliers_ratio	outliers_mean	with_mean	without_mean
Ozono	1	1	168.0	42.010	40.737374
RadSol	0	0		182.010	182.010000
Vient	2	2	20.4	10.206	9.997959
Temp	0	0		77.520	77.520000

Ahora se cuenta con una mayor cantidad de información concreta respecto a estos valores atípicos, considerando por ejemplo el número y la media de éstos:

- Ozono: 1 valor atípico con una media de 168
- Vient: 2 valores atípicos con una media de 20.4

Este tipo de información es altamente importante para una posterior toma de decisiones respecto a si eliminar o no estos valores, entre otras posibles acciones a realizar.