Taller Exploración de Datos (EDA)

Minería de Datos

Jonatan Camilo Igua Contreras
ID: 808919 | NRC: 73466

2025-03-14

1. Análisis descriptivo del dataset Iris

1.1 Carga y exploración de los datos

Importar dataset Iris en R: Para importar el dataset de Iris solo se usa el comando data(), que tiene la función de cargar datasets.

```
data(iris)
```

Datos del dataset Iris: Usando el comando head(), se puede ver la estructura del dataset de Iris

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Visualizar la estructura de los datos Con el comando str() se puede visualizar la estructura del conjunto de datos,se puede apreciar que el dataset cuenta con 150 observaciones osea 150 filas y 5 variables (Sepal.Length , Sepal.Width ,

Petal.Length , Petal.With , Species). Cada una de estas variables son numéricas, execto la de species que es un factor que cuenta con tres niveles (setosa, versicolor , virginica)

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1
```

Con el comando colnames() se pueden solo ver las columnas del dataset iris.

```
colnames(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

Tipos de datos: Usando el comando sapply() con la función class() se puede visualizar el tipo de dato de cada columna del dataset, se encontró que las cuatro variables son de tipo numéricas pero la variable de species es de tipo factor que cuenta con categorias para cada especie de iris.

```
sapply(iris, class)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##    "numeric"    "numeric"    "numeric"    "numeric"     "factor"
```

Verificar dimensiones: Para poder verificar las dimensiones del dataset de iris se usa el comando dim(), este comando retorna el numero de filas y columnas del del dataset. El resultado obtenido es que el dataset tiene 150 filas y 5 columnas.

```
## [1] 150   5
```

Se puede consultar de manera individual las dimensiones con el comando nrow() se pueden visualizarlas filas y con el comando ncol() las columnas.

```
## [1] 150
```

```
## [1] 5
```

Valores faltantes: R provee comandos para consultar si en un dataset hay valores faltantes, usando el comando is.na(), consulta cuantos datos faltantes hay en el dataset. Para contar cuantos valores faltantes hay se usa el comando sum().

```
## [1] 0
```

Si la suma da un valor de cero, no hay valores faltantes en el dataset de iris.

1.2 Calculo de medidas de tendencia central y dispersion

Analisis de las variables: Para poder ver un resumen de los datos estadísticos del dataset, se puede usar summary(), en este caso se analizara el resumen estadístico de las variables del dataset.
Resumen:

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Para crear la tabla con los datos de las variables de iris se uso un dataframe llamado Resumen_Variables, este almacena vectores con el resumen de los datos obtenidos anteriormente.

Tabla de análisis

```
print(Resumen_Variables)
```

```
##       Variable Mínimo Q1_25 Mediana_Q2 Media Q3_75 Máximo
## 1 Sepal.Length    4.3   5.1       5.80  5.84   6.4    7.9
## 2  Sepal.Width    2.0   2.8       3.00  3.06   3.3    4.4
## 3 Petal.Length    1.0   1.6       4.35  3.76   5.1    6.9
## 4  Petal.Width    0.1   0.3       1.30  1.20   1.8    2.5
```

- **Variable Sepal.Length**
  Mínimo = La flor con el sépalo mas corto mide 4.3 cm
  Q1 = El 25% de los sepalos tiene una longitud menor a 5.1 cm
  Mediana = La mitad de los sepalos miden menos de 5.8 cm
  Media = El promedio de los sepalos miden 5.84 cm
  Q3 = El 75% de los sepalos tienen una longitud menor de 6.4 cm
  Máximo = El sepalo mas largo en el conjunto mide 7.9 cm

- **Variable Sepal.Width**
  Mínimo: La flor con el sépalo mas estrecho mide 2.0 cm
  Q1: El 25% de los sepalos tienen un ancho menor a 2.8 cm
  Mediana: La mitad de los sepalos tienen un ancho menor 3.0 cm
  Media: El promedio de los sepalos tienen un promedio de 3.06 cm
  Q3: El 75% de los sepalos tienen un ancho menor a 3.3 cm
  Máximo: El sepalo mas ancho mide 4.4 cm

- **Variable Petal.Length**
  Mínimo: El pétalo mas corto mide 1.0 cm
  Q1: El 25% de los pétalos miden menos de 1.6 cm
  Mediana: La mitad de los pétalos miden menos de 4.35 cm
  Media: El promedio de los pétalos miden 3.76
  Q3: El 75% de los pétalos miden menos de 5.1 cm
  Máximo: El pétalo mas largo mide 6.6 cm

- **Variable Petal.Width**
  Mínimo: El pétalo mas estrecho mide 0.1 cm
  Q1: El 25% de los pétalos tienen un ancho menor a 0.3 cm
  Mediana: La mitad de los pétalos miden menos de 1.3 cm
  Media: El promedio de los pétalos tienen un ancho de 1.20 cm
  Q3: El 75% de los pétalos miden menos de 1.8 cm
  Máximo: El pétalo mas ancho mide 2.5 cm

Cálculos por especie

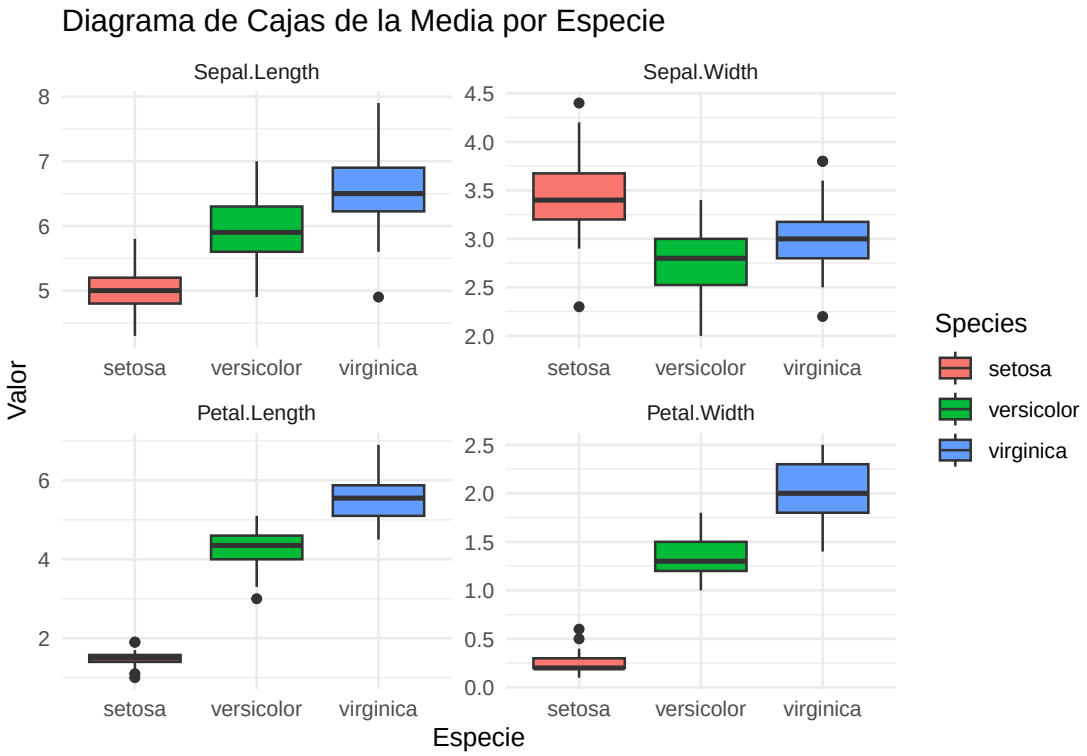
Media de las especies: Para obtener la media de las tres especies se usa la función aggregate(), para calcular la media de cada variable, se usa el argumento ~ Species, para indicar que se agruparan los datos por la columna Species y se le indica que se quiere calcular la media con la funcion FUN = mean

```
media_iris <- aggregate(. ~ Species, data = iris, FUN = mean)
```

Resultado del calculo de la Media:

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa        5.006       3.428        1.462       0.246
## 2 versicolor        5.936       2.770        4.260       1.326
## 3  virginica        6.588       2.974        5.552       2.026
```

- **Análisis media**
  Setosa: Tiene los sepalos mas anchos con un valor de 3.428 cm y sus pétalos son los mas pequeños en longitud con un valor de 1.462 y ancho 0.246 cm.
  Versicolor: Tiene los sepalos mas largos que la especie setosa pero mas cortos que virginica.
  virginica: Cuenta con los sepalos y pétalos mas largos y anchos. Sus pétalos cuentan con una medida de 5.552 cm de largo y 2.026 cm de ancho.



Mediana de las especies: La mediana representa el valor central de los datos de las tres especies, se implementa la función aggregate(), para calcular la mediana de cada variable, se usa el argumento ~ Species, para indicar que se agruparan los datos por la columna Species y se le indica que se quiere calcular la mediana con la función FUN = median

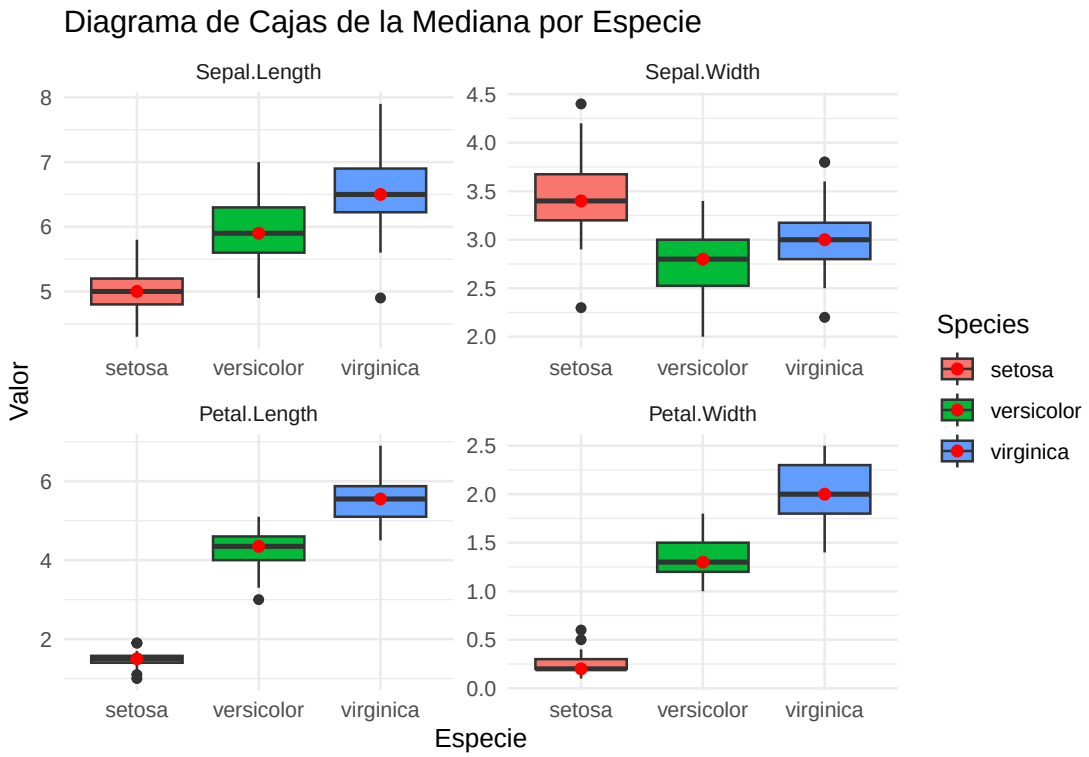
```
mediana_iris <- aggregate(. ~ Species, data = iris, FUN = median)
```

Resultado del calculo de la Mediana

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa          5.0         3.4         1.50         0.2
## 2 versicolor          5.9         2.8         4.35         1.3
## 3  virginica          6.5         3.0         5.55         2.0
```

- **Análisis mediana**
  Setosa: Es la especie con sepalos mas anchos con 3.4 cm pero pétalos mas pequeños con 1.5 cm
  Versicolor: Los datos analizados indican que tiene una longitud de pétalos

intermedia con 4.35
Virginica: Tiene los valores mas altos en todas las variables excepto en el ancho del pétalo y un poco menor en el ancho del sepalo comparado con la especie setosa.



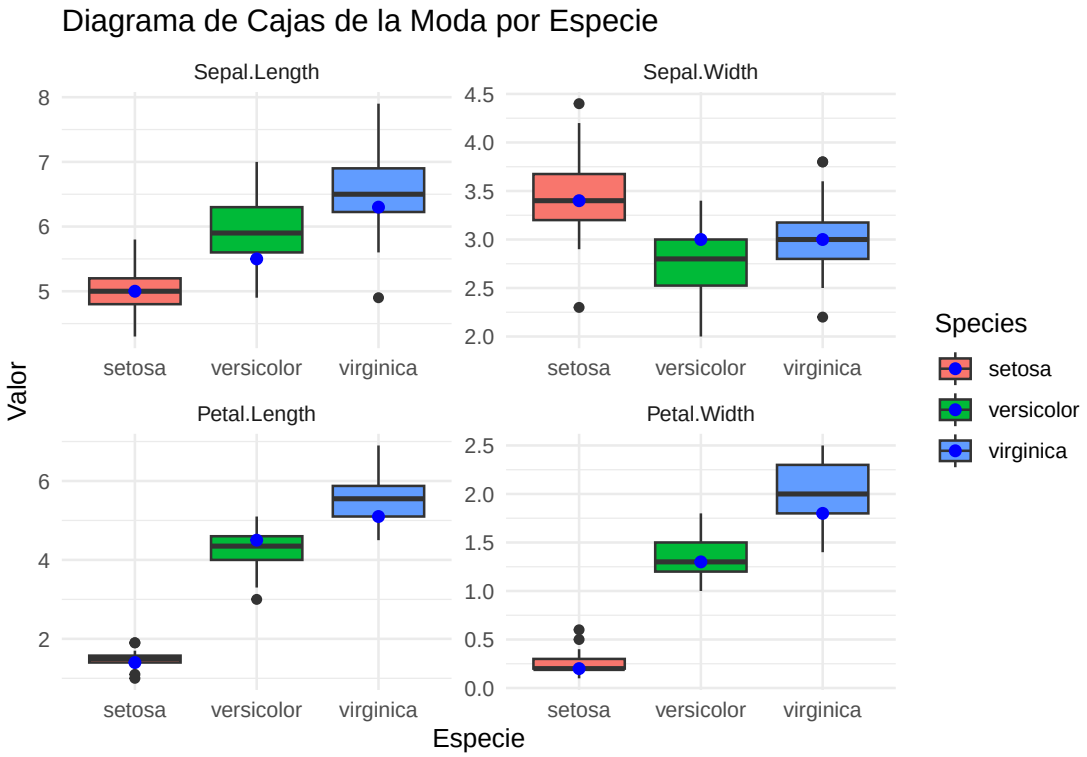
Moda de las especies En en el lenguaje de R no existe una función para calcular la moda, por eso es necesario crear una función, se crea una table(x) para los valores de x esta cuenta cuantas veces aparece un valor en la columna. Con wich.max(tab) encuentra la posición del valor con la mayor frecuencia, con names(tab), se obtiene el nombre del valor mas frecuente. Se guarda en la variable moda_iris para imprimir la moda.

```r
# Función para calcular la moda
calcular_moda <- function(x) {
  tab <- table(x)  #tabla de frecuencias
  moda <- names(tab)[which.max(tab)]  #valor con mayor frecuencia
  return(moda)
}
# Calcular la moda para cada variable
moda_iris <- aggregate(. ~ Species, data = iris, FUN = calcular_moda)
```

Resultado del calculo de la Moda

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa            5         3.4          1.4         0.2
## 2 versicolor          5.5           3          4.5         1.3
## 3  virginica          6.3           3          5.1         1.8
```

- **Análisis moda** La moda en este caso representara las medidas de los sepalos y pétalos que mas se repiten para cada especie. Con ayuda de la tabla se analizo que la especie de setosa tiende a tener pétalos pequeños y sepalos anchos, la versicolor tiene tamaños intermedios, el valor que mas se repite es el de la longitud del sepalo con 5.5 cm y la especie de virginica tiene los mayores tamaños en sepalos y pétalos.



Varianza de las especies Para obtener la varianza de las tres especies se usa la función aggregate(), para calcular la varianza de cada variable, se usa el argumento ~ Species, para indicar que se agruparan los datos por la columna Species y se le indica que se quiere calcular la varianza con la funcion FUN = var

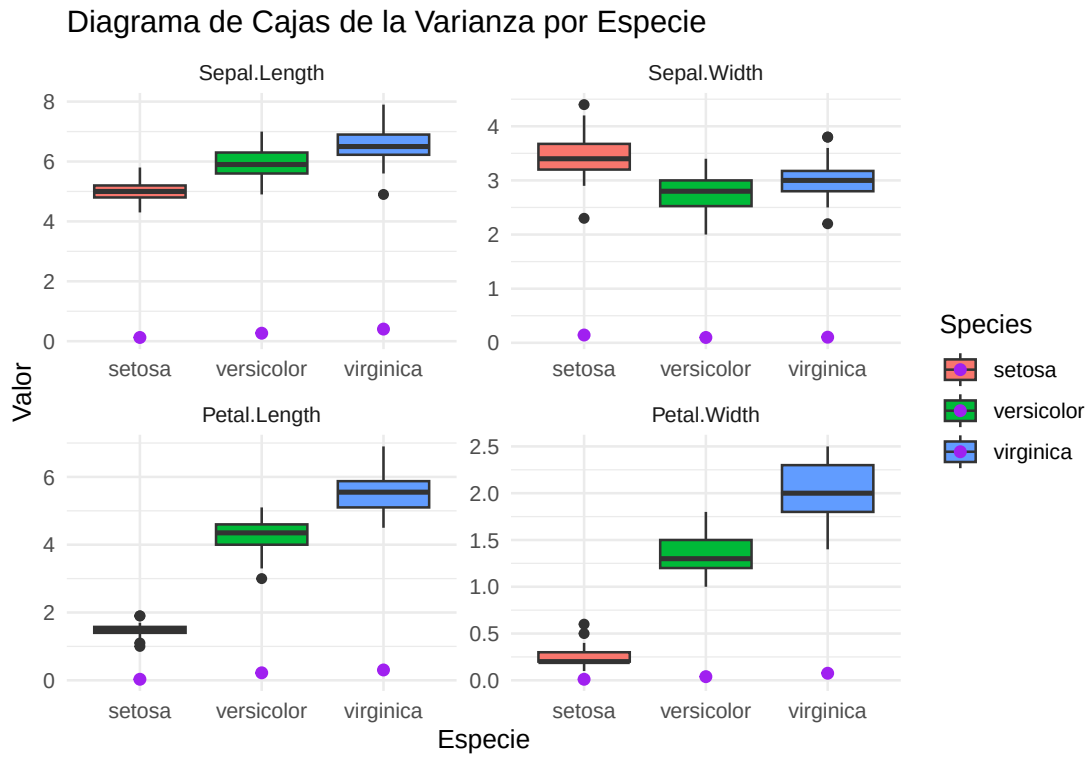
```r
varianza_iris <- aggregate(. ~ Species, data = iris, FUN = var)
```

Resultado del calculo de la Varianza

```r
print(varianza_iris)
```

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa    0.1242490  0.14368980   0.03015918  0.01110612
## 2 versicolor    0.2664327  0.09846939   0.22081633  0.03910612
## 3  virginica    0.4043429  0.10400408   0.30458776  0.07543265
```

- **Análisis varianza** La varianza se enfoca en medir cuanto se dispersan los datos con respecto a su media, la que anteriormente se calculo, entre mayor sea el valor de la varianza mas dispersos están los datos. Analizando los datos obtenidos, se puede deducir que la especie de setosa en su variable Sepal.Length tiene la varianza mas baja de 0.1242, esto quiere decir que la longitud de sepalo es mas uniforme en esta especie. Por otro lado la varianza mas alta en la misma variable es de virginica con 0.4043, quiere decir que sus sepalos son mas variables entre si.



Desviación estándar de las especies Para obtener la desviación estándar de las tres especies se usa la función aggregate(), para calcular la desviación de cada variable, se usa el argumento ~ Species, para indicar que se agruparan los datos por la columna Species y se le indica que se quiere calcular la desviación con la funcion FUN = sd

```
desviacion_iris <- aggregate(. ~ Species, data = iris, FUN = sd)
```

Resultado del calculo de la desviación estándar

```
print(desviacion_iris)
```

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa    0.3524897   0.3790644    0.1736640   0.1053856
## 2 versicolor    0.5161711   0.3137983    0.4699110   0.1977527
## 3  virginica    0.6358796   0.3224966    0.5518947   0.2746501
```

- **Análisis desviación estándar:** Con ayuda de la desviación estándar podemos medir cuanto varían los datos con respecto a la media, si los resultados son altos esto indica que la dispersión es mayor y si los resultados son bajos significa que los datos de las flores están mas agrupados alrededor de la media.
  Setosa = Según los resultados tiene la menor desviación estandar, quiere decir que las cuatro variables están mas agrupadas y menos dispersas. Las variables en donde es mas evidente ese hecho son Petal.Length con una desviación de 0.1737 y Petal.Width con una desviación de 0.1054
  Versicolor = La dispersión es mayor que en la especie de setosa pero menor que virginica.
  Virginica = Tiene la mayor desviación estándar, los datos de esta especie están mas dispersos que los de las otras especies, la variable Petal.Length tiene una desviación de 0.5519 y Petal.Width tiene 0.2747.

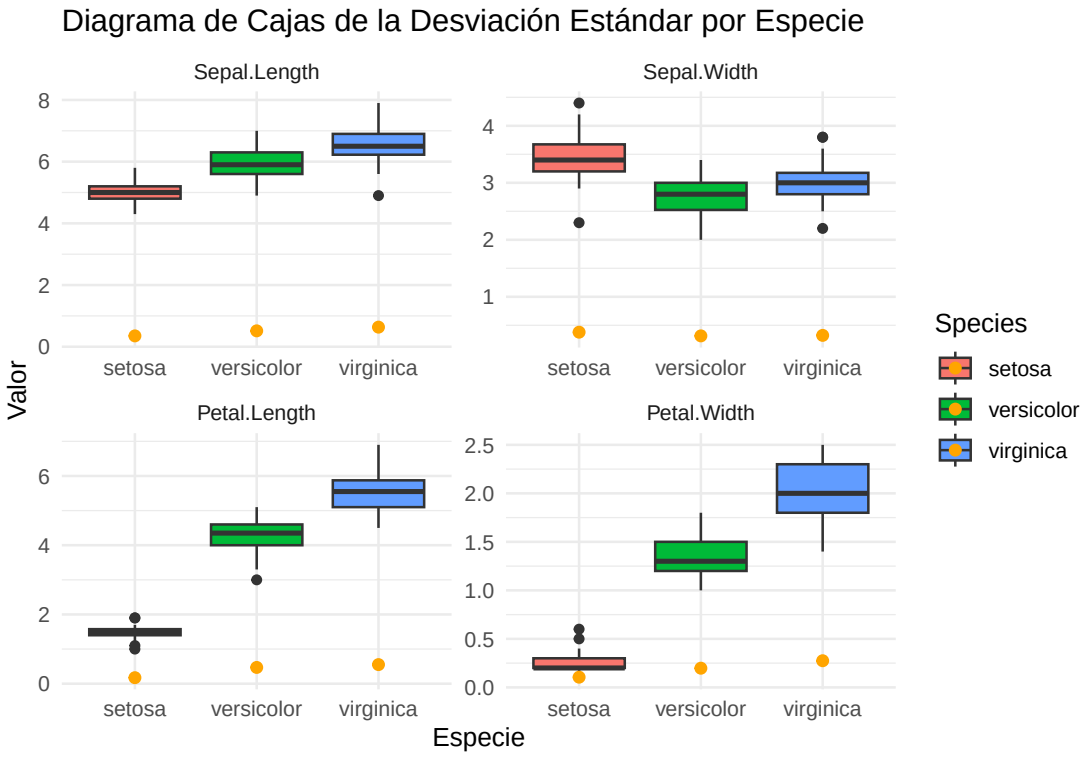
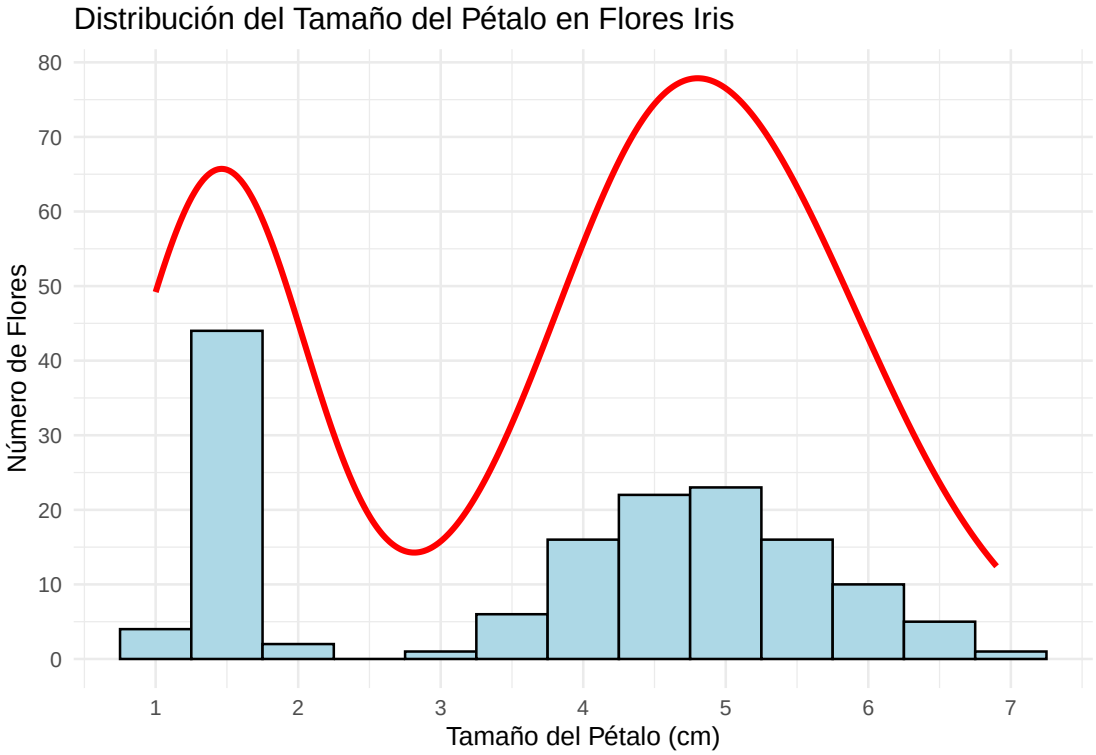


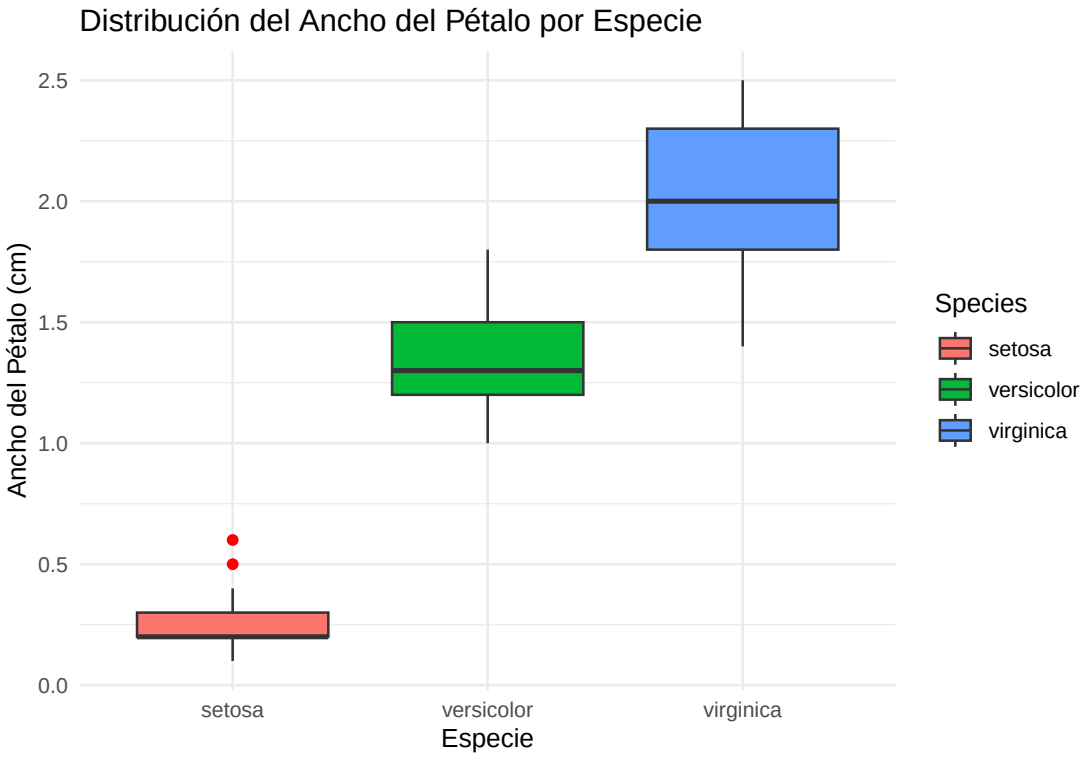
Diagrama de Cajas de la Desviación Estándar por Especie
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Valor
Especie
setosa
versicolor
virginica
Species

Histograma variable Petal.Length



El histograma muestra la frecuencia con la que aparecen los valores de la longitud del pétalo, en el gráfico las barras azules representan la distribución de los datos. Existen dos tipos de agrupación:
* Primera agrupación: Se puede apreciar una alta frecuencia de valores pequeños, esto quiere decir que los iris son cortos de 1 a 2 cm.
* Segunda agrupación: Entre 3 y 7 se puede ver una mayor dispersión, de los datos esto quiere decir que en este sector hay mas flores variadas con diferentes medidas. La curva de densidad de color rojo, permite estimar la densidad de los datos, hay una gran acumulación de datos no dispersos en la primera agrupación, pero por otro lado hay mayor densidad de datos en la agrupación dos.

Boxplots de Petal.Width



En el bloxplot se puede apreciar que el eje x muestra los tipos de flor, el eje y los intervalos del ancho de los pétalos, los puntos rojos indican valores atipicos. Se puede analizar que setosa tiene los pétalos mas pequeños, virginica posee los pétalos mas anchos y la versicolor esta en un punto medio.

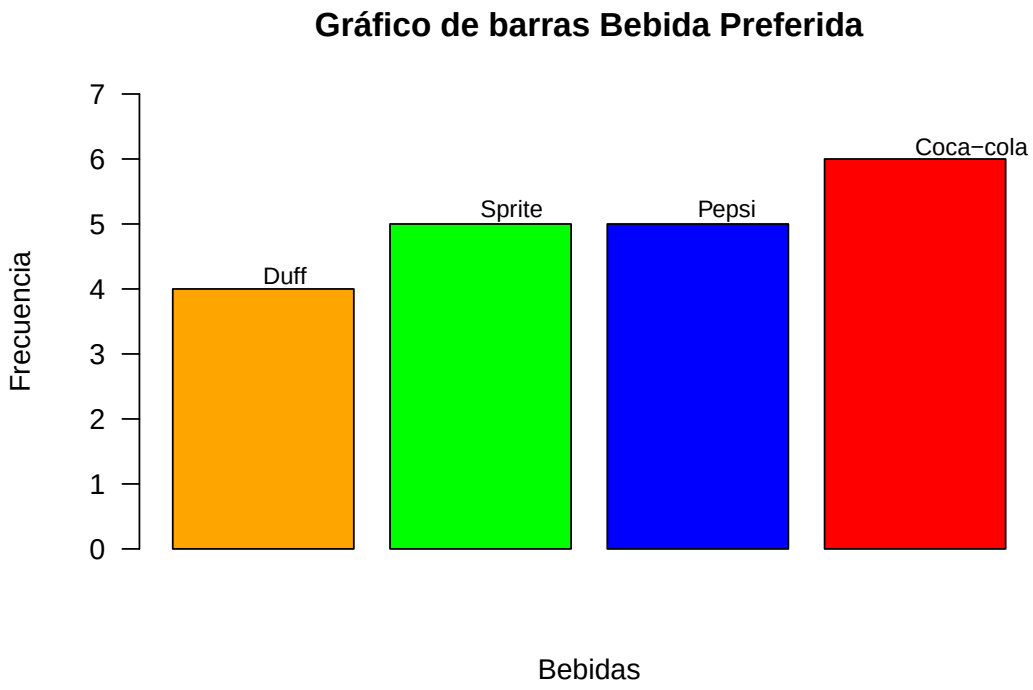
2. Tablas de frecuencias y visualización de datos

Datos de la bebida favorita: Del grupo de personas se obtuvieron los siguientes datos:
* Duff: 4
* Sprite: 5
* Pepsi: 5
* Coca-cola: 6

Para crear la tabla se usa un dataframe que muestre la bebida y la frecuencia.

Tabla de frecuencias de bebidas

```
##      Marcas Frecuencia
## 1      Duff          4
## 2    Sprite          5
## 3     Pepsi          5
## 4 Coca-cola          6
```



Como se puede apreciar en el gráfico la bebida mas preferida por el grupo de personas es la coca-cola por otro lado la Sprite y la Pepsi tiene la misma votación y la bebida menos preferida es la Duff

3. Análisis del dataset Swiss

Importar dataset swiss en R: Para importar en dataset de swiss se usa el comando data().

```
data(swiss)
```

Datos del dataset swiss: Usando el comando head(), se puede ver la estructura del dataset de swiss

```
head(swiss)
```

```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0          15        12     9.96
## Delemont          83.1        45.1           6         9    84.84
## Franches-Mnt      92.5        39.7           5         5    93.40
## Moutier           85.8        36.5          12         7    33.77
## Neuveville        76.9        43.5          17        15     5.16
## Porrentruy        76.1        35.3           9         7    90.57
##              Infant.Mortality
## Courtelary               22.2
## Delemont                 22.2
## Franches-Mnt             20.2
## Moutier                  20.3
## Neuveville               20.6
## Porrentruy               26.6
```

Visualizar la estructura de Swiss Con el comando str() se puede visualizar la estructura del conjuto de datos de swiss

```
str(swiss)
```

```
## 'data.frame':    47 obs. of  6 variables:
##  $ Fertility       : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9 ...
##  $ Agriculture     : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...
##  $ Examination     : int  15 6 5 12 17 9 16 14 12 16 ...
##  $ Education       : int  12 9 5 7 15 7 7 8 7 13 ...
##  $ Catholic        : num  9.96 84.84 93.4 33.77 5.16 ...
##  $ Infant.Mortality: num  22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...
```

Usando el comando colnames() se pueden ver las columnas del dataset

```
colnames(swiss)
```

```
## [1] "Fertility"        "Agriculture"      "Examination"      "Education"
## [5] "Catholic"         "Infant.Mortality"
```

Las columnas encontraron fueron:
Fertility : Fertilidad
Agriculture: Agricultura
Examination: Examenes
Education: Educación
Catholic: Porcentaje de Catolicos
Infant.Mortality: Mortalidad infantil

Verificación tipos de datos: Usando la función sapply() con la función class() se puede ver el tipo de dato de cada columna del dataset,se encontro que de las seis variables cuatro son del tipo numerico pero Examination y Education son de tipo integer lo que significa que sus valores son de tipo entero, los de tipo numerico permiten punto decimal.

```
sapply(swiss, class)
```

```
##        Fertility      Agriculture      Examination        Education
##        "numeric"        "numeric"        "integer"        "integer"
##         Catholic Infant.Mortality
##        "numeric"        "numeric"
```

Principales indicadores Estadisticos

Calculo del valor de la Media Para calcular la media de cada varible se usa la función mean().

Variable Fertility

```
mean(swiss$Fertility)
```

```
## [1] 70.14255
```

Variable Infant.Mortality

```
mean(swiss$Infant.Mortality)
```

```
## [1] 19.94255
```

Calculo del valor de la Mediana: Para calcular la mediana de cada variable se usa la función median().

Variable Fertility

```
median(swiss$Fertility)
```

```
## [1] 70.4
```

Variable Infant.Mortality

```
median(swiss$Infant.Mortality)
```

```
## [1] 20
```

Calculo del valor de la Varianza: Para calcular la varianza de cada variable se usa la función var().

Variable Fertility

```
var(swiss$Fertility)
```

```
## [1] 156.0425
```

Variable Infant.Mortality

```
var(swiss$Infant.Mortality)
```

```
## [1] 8.483802
```

Calculo del valor de la Desviación estándar: La desviación estandar se calcula con la función sd().

Variable Fertility

```
sd(swiss$Fertility)
```

```
## [1] 12.4917
```

Variable Infant.Mortality

```
sd(swiss$Infant.Mortality)
```

```
## [1] 2.912697
```

Calculo del valor Mínimo: Para calcular el valor minimo de las variables se usa la función min().

Variable Fertility

```
min(swiss$Fertility)
```

```
## [1] 35
```

Variable Infant.Mortality

```
min(swiss$Infant.Mortality)
```

```
## [1] 10.8
```

Calculo del valor Máximo

Para calcular el maximo se usa la función max().

Variable Fertility

```
max(swiss$Fertility)
```

```
## [1] 92.5
```

Variable Infant.Mortality

```
max(swiss$Infant.Mortality)
```

```
## [1] 26.6
```

4. Notas de estudiantes y analisis de aprobación

Información de las notas de los estudiantes

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]  4.1  2.7  3.1  3.2  3.0  3.2  2.0  2.4  1.6   3.2
##  [2,]  3.1  2.6  2.0  2.4  2.8  3.3  4.0  3.4  3.0   3.1
##  [3,]  2.7  2.7  3.0  3.8  3.2  2.2  3.5  3.5  3.8   3.5
##  [4,]  3.9  4.2  4.3  3.9  3.2  3.5  3.5  3.7  4.1   3.7
##  [5,]  3.5  3.6  3.2  3.1  3.4  3.0  3.0  3.0  2.7   1.7
##  [6,]  3.6  2.1  2.4  3.0  3.1  2.5  2.5  3.6  2.2   2.4
##  [7,]  3.1  3.3  2.7  3.7  3.0  2.7  3.0  3.2  3.1   2.4
##  [8,]  3.0  2.7  2.5  3.0  3.0  3.0  3.2  3.1  3.8   4.1
##  [9,]  3.7  3.5  3.0  3.7  3.7  4.1  3.7  3.9  3.7   2.0
```

Tabla de frecuencias absolutas Con una tabla de frecuencia absoluta se puede saber cuantas veces aparece un valor, para esto se usa la función table() que cuenta cuantas veces aparece cada valor único.

```
tabla_adsoluta <- table(notasE)
```

Resultado frecuencias absolutas

```
print(tabla_adsoluta)
```

```
## notasE
## 1.6 1.7   2 2.1 2.2 2.4 2.5 2.6 2.7 2.8   3 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9
##   1   1   3   1   2   5   3   1   7   1  14   8   8   2   2   7   3   8   3   3
##   4 4.1 4.2 4.3
##   1   4   1   1
```

Tabla de frecuencias relativas La tabla de frecuencias relativas permite ver la proporción de cada valor en relación con el total, en este caso la función prop.table(tabla_adsoluta), divide cada valor de la tabla absoluta por el total de observaciones.

```
tabla_relativa <- prop.table(tabla_adsoluta)
```

Resultado frecuencias relativas

```
print(tabla_relativa)
```

```
## notasE
##         1.6         1.7           2         2.1         2.2         2.4         2.5
## 0.01111111 0.01111111 0.03333333 0.01111111 0.02222222 0.05555556 0.03333333
##         2.6         2.7         2.8           3         3.1         3.2         3.3
## 0.01111111 0.07777778 0.01111111 0.15555556 0.08888889 0.08888889 0.02222222
##         3.4         3.5         3.6         3.7         3.8         3.9           4
## 0.02222222 0.07777778 0.03333333 0.08888889 0.03333333 0.03333333 0.01111111
##         4.1         4.2         4.3
## 0.04444444 0.01111111 0.01111111
```

Tabla de frecuencias acumuladas Permite ver el porcentaje acumulado, usando la función cumsum(), se suman progresivamente las frecuencias absolutas, permite saber cuantos estudiantes tienen una notar menor o igual a un valor que busquemos.

```
tabla_acumulada <- cumsum(tabla_adsoluta)
```

Resultado frecuencias acumuladas

```
print(tabla_acumulada)
```

```
## 1.6 1.7   2 2.1 2.2 2.4 2.5 2.6 2.7 2.8   3 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9
##   1   2   5   6   8  13  16  17  24  25  39  47  55  57  59  66  69  77  80  83
##   4 4.1 4.2 4.3
##  84  88  89  90
```

Tabla de frecuencias relativas acumuladas Con esta tabla se puede saber como se distribuyen los datos, conociendo su comportamiento y su tendencia.

```
tabla_relativa_acumulada <- cumsum(tabla_relativa)
```

Resultado frecuencias relativas acumuladas

```
print(tabla_relativa_acumulada)
```

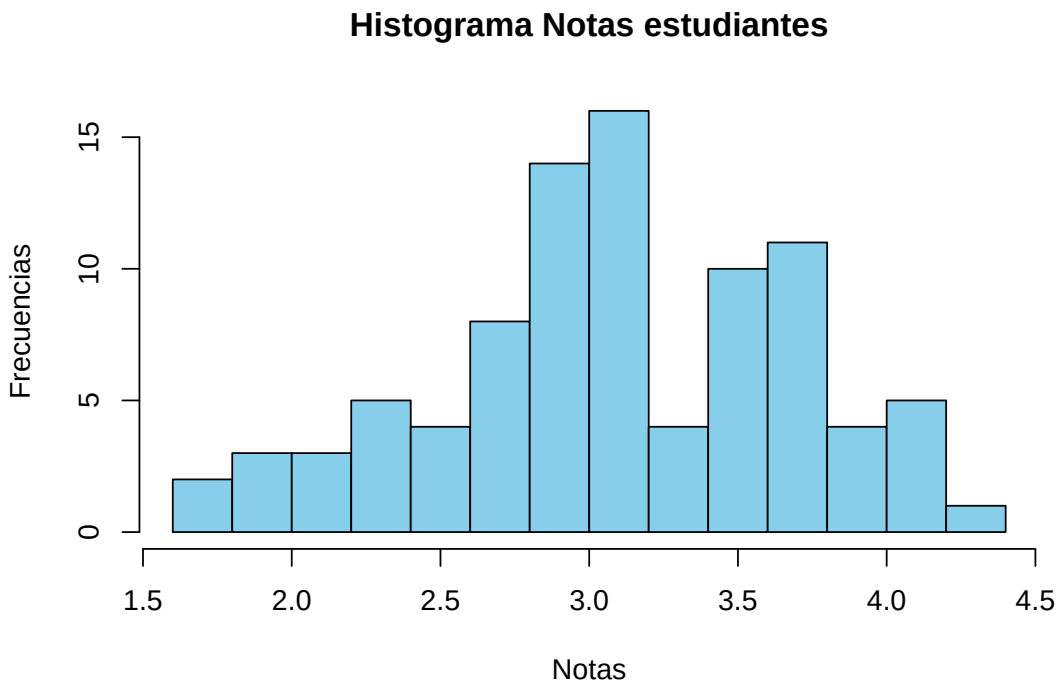
```
##        1.6        1.7          2        2.1        2.2        2.4        2.5
## 0.01111111 0.02222222 0.05555556 0.06666667 0.08888889 0.14444444 0.17777778
##        2.6        2.7        2.8          3        3.1        3.2        3.3
## 0.18888889 0.26666667 0.27777778 0.43333333 0.52222222 0.61111111 0.63333333
##        3.4        3.5        3.6        3.7        3.8        3.9          4
## 0.65555556 0.73333333 0.76666667 0.85555556 0.88888889 0.92222222 0.93333333
##        4.1        4.2        4.3
## 0.97777778 0.98888889 1.00000000
```

Tabla de distribución de frecuencias notas de estudiantes La tabla de distribución de frecuencias se crea con base a las frecuencias absolutas, relativas y acumuladas calculadas anteriormente.

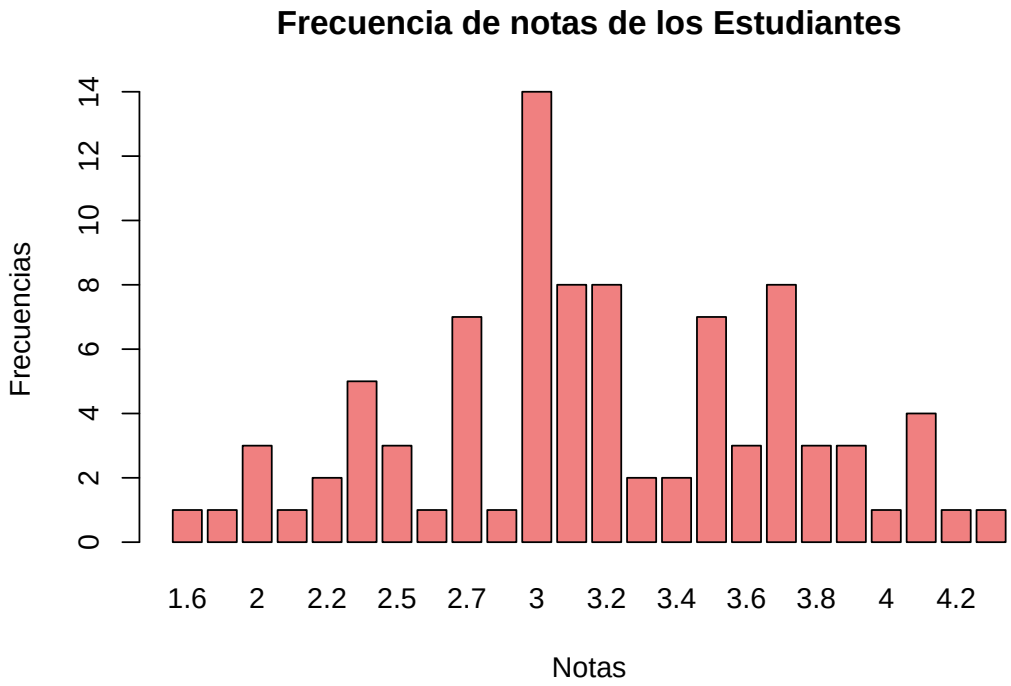
Nota	Freq	Freq..Rel	Freq..Acum	Freq..Rel..Acum
1.6	1	0.0111	1	0.0111
1.7	1	0.0111	2	0.0222
2.0	3	0.0333	5	0.0556
2.1	1	0.0111	6	0.0667
2.2	2	0.0222	8	0.0889
2.4	5	0.0556	13	0.1444
2.5	3	0.0333	16	0.1778
2.6	1	0.0111	17	0.1889
2.7	7	0.0778	24	0.2667
2.8	1	0.0111	25	0.2778
3.0	14	0.1556	39	0.4333
3.1	8	0.0889	47	0.5222
3.2	8	0.0889	55	0.6111
3.3	2	0.0222	57	0.6333
3.4	2	0.0222	59	0.6556
3.5	7	0.0778	66	0.7333
3.6	3	0.0333	69	0.7667
3.7	8	0.0889	77	0.8556
3.8	3	0.0333	80	0.8889
3.9	3	0.0333	83	0.9222
4.0	1	0.0111	84	0.9333
4.1	4	0.0444	88	0.9778
4.2	1	0.0111	89	0.9889
4.3	1	0.0111	90	1.0000

Gráficos visualizar datos de las notas

El histograma permite ver como se distribuyen las notas de los estudiantes



Con un gráfico de barras de la frecuencia absoluta se puede ver la frecuencia de cada nota, usa la función barplot para indicar a R que es este tipo de gráfico.



Indicadores estadísticos Notas estudiantes

Media de notas

```
media_notas <- mean(notasE)
print(media_notas)
```

```
## [1] 3.136667
```

Mediana de notas

```
mediana_notas <- median(notasE)
print(mediana_notas)
```

```
## [1] 3.1
```

Moda de notas

```
moda_notas <- as.numeric(names(sort(table(notasE), decreasing = TRUE)[1]))
print(moda_notas)
```

```
## [1] 3
```

Varianza de notas

```
varianza_notas <- var(notasE)
print(varianza_notas)
```

```
## [1] 0.3529101
```

Desviación estándar de notas

```
desviacion_estandar_notas <- sd(notasE)
print(desviacion_estandar_notas)
```

```
## [1] 0.5940624
```

Indicadores estadísticos de las notas:

Indicador	Valor
Media	3.1367
Mediana	3.1000
Moda	3.0000
Varianza	0.3529
Desviación estándar	0.5941

Pregunta del ejercicio ¿Que porcentaje de los estudiantes reprobaron la evaluación? Nota final < 3.0

Para calcular el porcentaje de estudiantes se puede usar la frecuencia acumulada de los estudiantes con notas inferiores a 3.0 y dividirla en el total de los estudiantes. Se usa la tabla de tabla_distribucion_frecuencia.

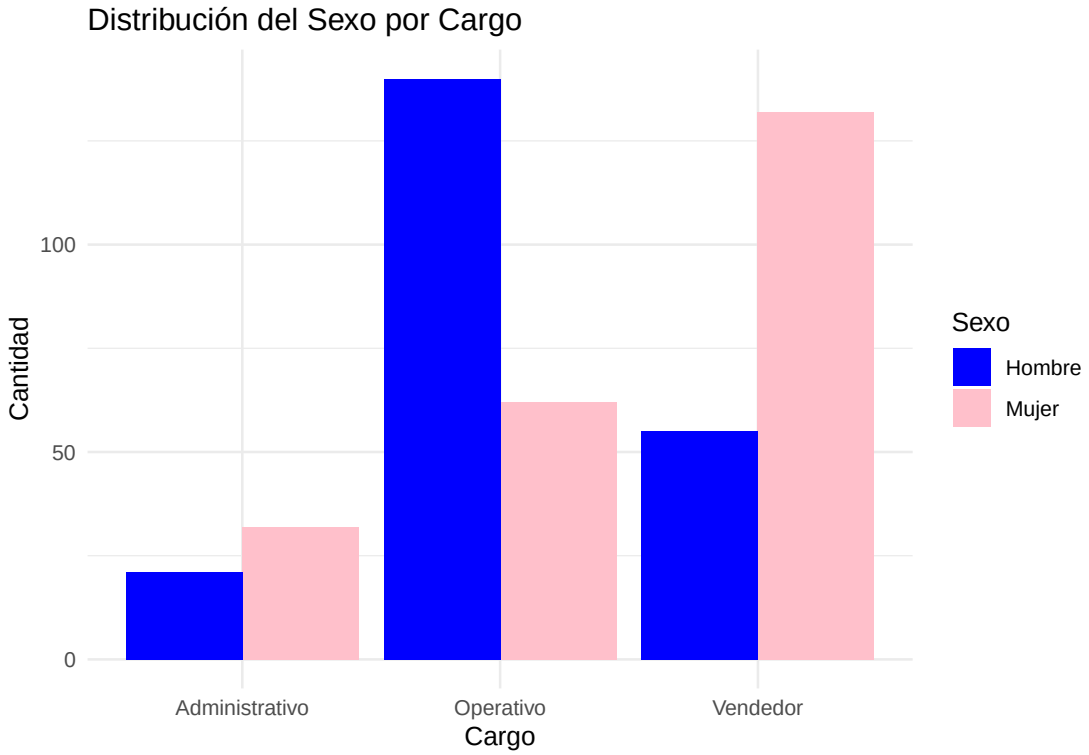
```
# Notas menores a 3.0
reprobados <- tabla_distribucion_frecuencias[tabla_distribucion_frecuencias$Nota <

# Sumar las frecuencias de los reprobados
total_reprobados <- sum(reprobados$Freq, na.rm = TRUE)

# Total de estudiantes
total_estudiantes <- sum(tabla_distribucion_frecuencias$Freq, na.rm = TRUE)

# Calcular el porcentaje
porcentaje_reprobados <- (total_reprobados / total_estudiantes) * 100

sprintf("El porcentaje de estudiantes reprobados es: %.2f%%", porcentaje_reprobado
```

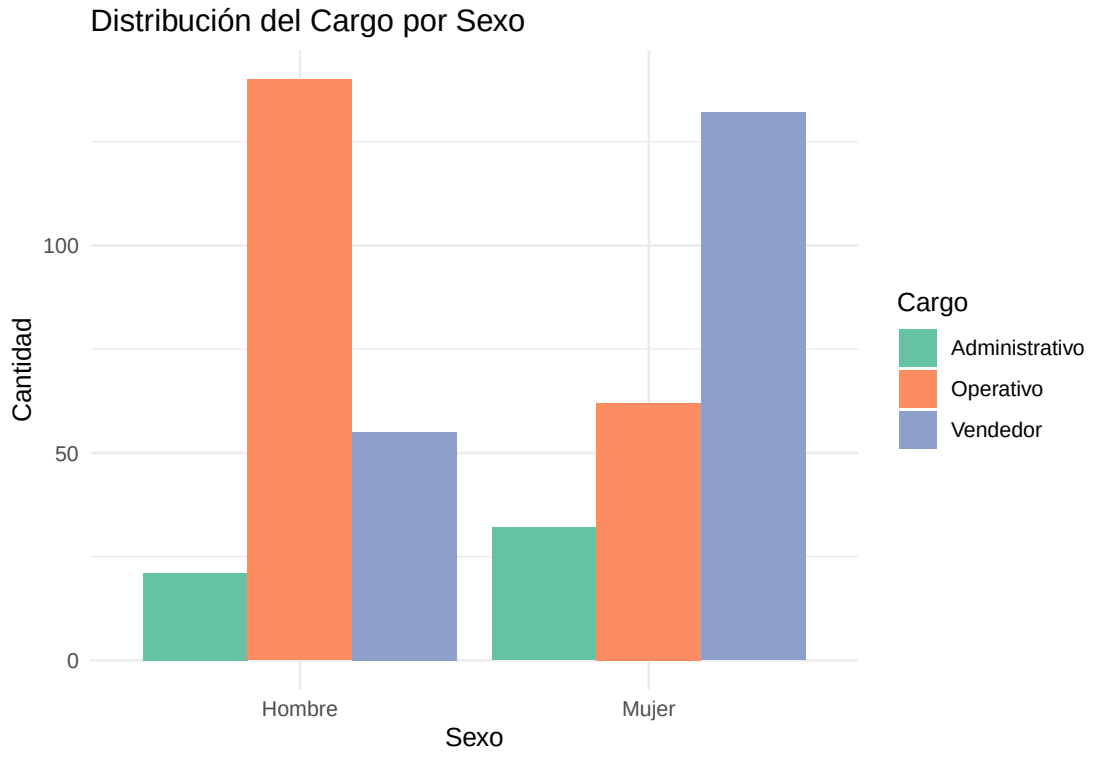
```
## [1] "El porcentaje de estudiantes reprobados es: 27.78%"
```

5. Distribución de cargos en una empresa por genero

Tabla distribución empleados por cargo y genero

Cargo	Mujer	Hombre
Administrativo	32	21
Operativo	62	140
Vendedor	132	55





6. Generación de gráficos con herramientas de IA

Para la generación de los gráficos se utilizara la herramienta Tableau AI, para hacer esto es necesario guardar los datos que se tienen de R en un formato compatible como CSV.

Con este código se toman los datos de la tabla_empleados, para guardarlos en un CSV para subirlo a Tableau AI para generar los gráficos.

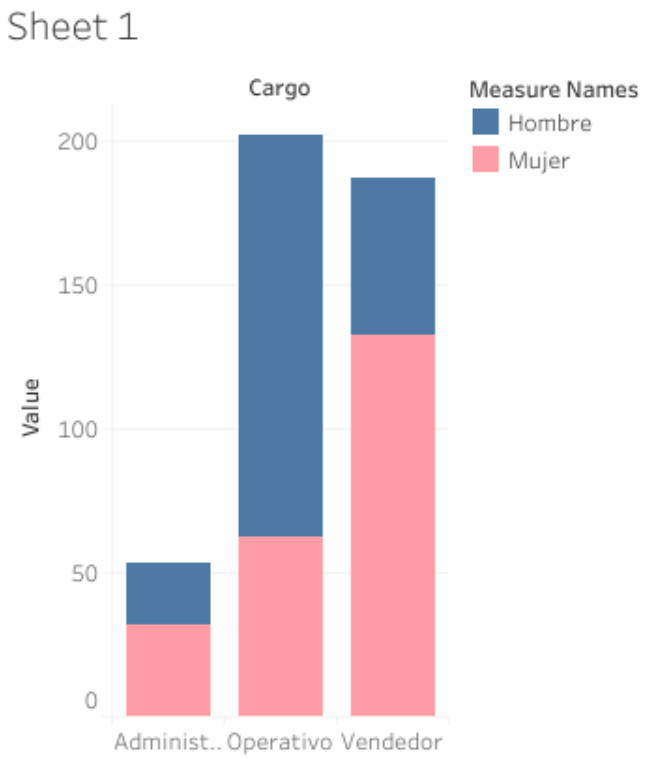
Grafico distribución del sexo por cargo Tableau AI

```
write.csv(tabla_empleados, "datos_empleados.csv", row.names = FALSE)
```

Después de subir el archivo a tableau public se descarga el gráfico generado en formato PNG, se guarda en el proyecto y para poder colocar el gráfico en R se usa readPNG(), en el cual se le indica la ruta de la imagen

```
library(png)
library(grid)
```

```
ruta_img1 <- file.path(getwd(), "Sheet 1.png")
img1 <- readPNG(ruta_img1)
grid.raster(img1)
```



Comparación de gráficos de distribución del sexo por cargo

Después de realizar el gráfico con la herramienta, se pudo apreciar diferencias significativas, la primera fue que el gráfico de barras creado con tableau tiene la distribución de las etiquetas de Hombre y mujer de forma distinta, ya que este crea 3 barras para los 3 cargos, pero agrega la diferenciación por medio de partir las barras en dos con colores distintos. Por otro lado en R la gráfica se genera con 6 barras para diferenciar los hombres y las mujeres.

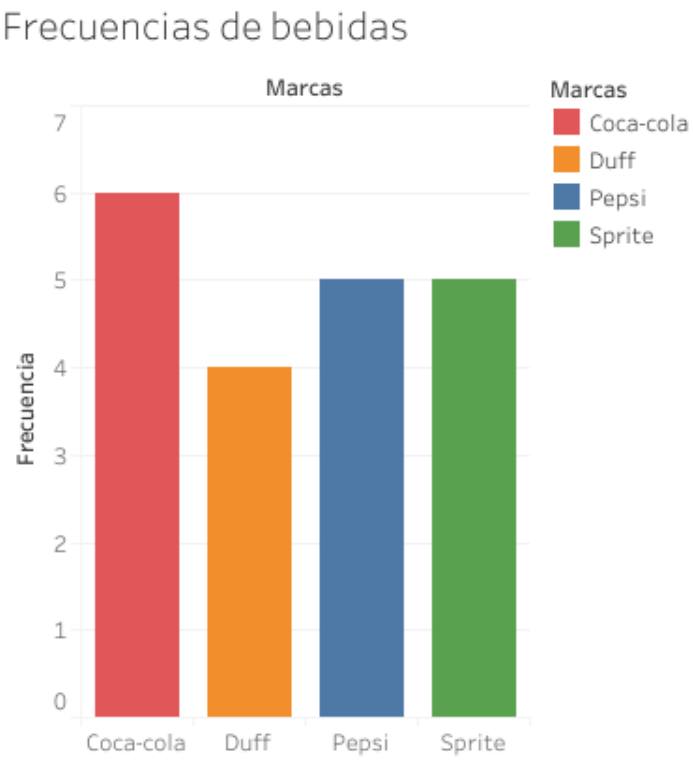
Tabla de frecuencias de bebidas

La siguiente tabla que se va a comparar es la tabla realizada en el apartado 2, la cual detalla la preferencia de un tipo de bebida por un grupo de personas. Se usa la función write.csv() para convertir los datos que se tienen en tabla_frecuencias, y poder subir este archivo en la herramienta.

```r
write.csv(tabla_frecuencias, "tabla_bebidas.csv", row.names = FALSE)
```

Después de tener el .csv se sube a tableau, se crea el gráfico con esta herramienta y se descarga en formato PNG, se guarda en el proyecto y para poder colocar el gráfico en R se usa readPNG(), en el cual se le indica la ruta de la imagen, en este caso es solo el nombre de la imagen sheet 2.png porque se agregaron las imagenes al directorio principal del proyecto, se aplico file.path para asegurar que en cualquier sistema operativo se pueda leer la imagen.

```r
library(png)
library(grid)

ruta_img2 <- file.path(getwd(), "Sheet 2.png")
img2 <- readPNG(ruta_img2)
grid.raster(img2)
```



Comparación de tabla de frecuencias de bebidas

El gráfico generado con la herramienta es muy similar en la distribución de los datos al gráfico de R, pero se diferencia en la claridad del diseño ya que las etiquetas en

el gráfico de barras de tableau son mas, claras están debajo de cada barra y se creo un pequeño index que ayuda a entender los datos, pero por otro lado el gráfico de R también cuenta con etiquetas pero están en la parte superior y no están centradas. Estos errores se pueden corregir en el código de R pero es mas engorroso y en cambio con el uso de la herramienta se creo un gráfico claro y organizado de una manera muy sencilla. Ambas tablas comparten la misma información de manera clara la única diferencia es la posición de cada barra entre las tablas, cada una expresa los mismos datos pero en distinto orden en el eje x pero no cambia la media del eje y.