Creado por:

Isabel Maniega

3.1.1 – Comprender y aplicar medidas estadísticas en el análisis de datos.

4.2.1 – Aplicar las estadísticas descriptivas de Python para el análisis de conjuntos de datos.

Las medidas que indican el centro aproximado de una distribución se denominan **medidas de tendencia central**.

Las medidas que describen la dispersión de los datos son **medidas de dispersión**. Estas medidas incluyen la media, la mediana, la moda, el rango, los cuartiles superior e inferior, la varianza y la desviación estándar.

In [1]:
```
# pip install scipy
```

In [2]:
```
# pip install statsmodels
```

In [3]:
```
# pip install matplotlib seaborn
```

In [4]:
```
import pandas as pd
import numpy as np

# Preprocesado y análisis
# ==============================================================================
import statsmodels.api as sm
from scipy import stats

# Gráficos
# ==============================================================
import matplotlib.pyplot as plt
import seaborn as sns
from IPython import display
```

Ejemplo 1: Notas

In [5]:
```
df = pd.DataFrame({"notas_1": [15, 16, 15, 17, 14, 14, 14, 10, 15, 25],
                   "notas_2": [16, 21, 16, 16, 13, 15, 15, 19, 22, 15],
                   "notas_3": [17, 22, 15, 22, 14, 15, 16, 15, 24, 16]})
df.head()
```

Out[5]:

	notas_1	notas_2	notas_3
0	15	16	17
1	16	21	22
2	15	16	15
3	17	16	22
4	14	13	14

Tendencia Central

Media: La media de un conjunto de datos es la suma de todos los valores de un conjunto de datos dividida por el número de valores del conjunto.

media_1 = (15 + 16 + 15 + 17 + 14) / 5

In [6]:
```
media_1 = sum(df.notas_1)/ len(df.notas_1)
media_1
```

Out[6]: 15.5

Como calcular la media de las distintas notas:

In [7]:
```
media_1 = df["notas_1"].mean()
media_1
```

Out[7]: np.float64(15.5)

In [8]:
```
media_2 = df["notas_2"].mean()
media_2
```

Out[8]: np.float64(16.8)

In [9]:
```
media_3 = df["notas_3"].mean()
media_3
```

Out[9]: np.float64(17.6)

Mediana: La mediana de un conjunto de datos es el "elemento medio" cuando los datos están organizados en orden ascendente.

mediana_1 = (10, 14, 14, 14, **15**, **15**, 15, 16, 17, 25)

mediana_1 = 15

In [10]:
```
mediana_1 = df.notas_1.sort_values(ascending=True)
mediana_1
```

Out[10]:
```
7    10
6    14
4    14
5    14
2    15
0    15
8    15
1    16
3    17
9    25
Name: notas_1, dtype: int64
```

Como calcular la mediana de las distintas notas:

In [11]:
```
mediana_1 = df["notas_1"].median()
mediana_1
```

Out[11]:
```
np.float64(15.0)
```

In [12]:
```
mediana_2 = df["notas_2"].median()
mediana_2
```

Out[12]:
```
np.float64(16.0)
```

In [13]:
```
mediana_3 = df["notas_3"].median()
mediana_3
```

Out[13]:
```
np.float64(16.0)
```

Moda: La moda es la medida que aparece con mayor frecuencia en un conjunto de datos.

moda_1 = (10, **14, 14, 14, 15, 15, 15,** 16, 17, 25)

moda_1 = (14, 15,)

Como calcular la moda de las distintas notas:

In [14]:
```
moda_1 = df["notas_1"].mode()
moda_1
```

Out[14]:
```
0    14
1    15
Name: notas_1, dtype: int64
```

In [15]:
```
moda_2 = df["notas_2"].mode()
moda_2
```

Out[15]:
```
0    15
1    16
Name: notas_2, dtype: int64
```

In [16]:
```
moda_3 = df["notas_3"].mode()
moda_3
```

Out[16]:
```
0    15
Name: notas_3, dtype: int64
```

In [17]:
```
df.notas_3.value_counts()
```

Out[17]:
```
notas_3
15    3
22    2
16    2
17    1
14    1
24    1
Name: count, dtype: int64
```

Resultados Nota_1:

In [18]:
```
print(f"Media: {media_1}, Mediana: {mediana_1}, Moda: \n{moda_1}")
```

```
Media: 15.5, Mediana: 15.0, Moda:
0    14
1    15
Name: notas_1, dtype: int64
```

Resultados Nota_2:

In [19]:
```
print(f"Media: {media_2}, Mediana: {mediana_2}, Moda: \n{moda_2}")
```

```
Media: 16.8, Mediana: 16.0, Moda:
0    15
1    16
Name: notas_2, dtype: int64
```

Resultados Nota_2:

In [20]:
```
print(f"Media: {media_3}, Mediana: {mediana_3}, Moda: \n{moda_3}")
```

```
Media: 17.6, Mediana: 16.0, Moda:
0    15
Name: notas_3, dtype: int64
```

Variabilidad

Varianza

La varianza mide el grado de dispersión de un conjunto de valores. Cuantifica la distancia que separa cada punto de datos del conjunto de la media.

Variance = (∑(Each value — Mean)^2)/ Number of values

Variance = (suma((15 -15.5)^2, (16-15.5)^2, (15-15.5)^2, (17-15.5)^2, (14-15.5)^2, (14-15.5)^2, (14-15.5)^2, (10-15.5)^2, (15-15.5)^2, (25-15.5)^2)/10)

In [21]:
```
Variance = ((15 -15.5)**2 + (16-15.5)**2 + (15-15.5)**2 + (17-15.5)**2 +
Variance
```

Out[21]:
```
13.05
```

In [22]:
```
varianza_1 = df["notas_1"].var(ddof=0)
varianza_1
```

Out[22]: np.float64(13.05)

In [23]:
```
varianza_2 = df["notas_2"].var(ddof=0)
varianza_2
```

Out[23]: np.float64(7.56)

In [24]:
```
varianza_3 = df["notas_3"].var(ddof=0)
varianza_3
```

Out[24]: np.float64(11.84)

La varianza nos ayuda a comprender en qué medida los valores individuales se desvían de la media. Una varianza más alta indica un conjunto de datos más disperso.

Desviación estándar

La desviación estándar es la raíz cuadrada de la varianza. Proporciona una medida más interpretable de la dispersión de los datos.

In [25]:
```
import math

STD = math.sqrt(((15 -15.5)**2 + (16-15.5)**2 + (15-15.5)**2 + (17-15.5)*
STD
```

Out[25]: 3.6124783736376886

Calculamos la desviación estandar de los datos:

Delta Degrees of Freedom --> ddof

https://stackoverflow.com/questions/41204400/what-is-the-difference-between-numpy-var-and-statistics-variance-in-python

In [26]:
```
std_1 = df["notas_1"].std(ddof=0)
std_1
```

Out[26]: np.float64(3.6124783736376886)

In [27]:
```
std_2 = df["notas_2"].std(ddof=0)
std_2
```

Out[27]: np.float64(2.749545416973504)

In [28]:
```
std_3 = df["notas_3"].std(ddof=0)
std_3
```

Out[28]: np.float64(3.4409301068170506)

La desviación estándar es una métrica muy utilizada en estadística. Proporciona una medida de la distancia típica entre cada punto de datos y la media, lo que permite

comprender mejor la variabilidad del conjunto de datos.

Rango intercuartílico o RIQ

El rango es la diferencia entre los valores más altos y más bajos de un conjunto de datos.
Para determinar el rango:

1. Identifique el valor más alto de su conjunto de datos. Esto se llama máximo. 10, 14, 14, 14, 15, 15, 15, 16, 17, **25**
2. Identifique el valor más bajo de su conjunto de datos. Esto se llama mínimo. **10**, 14, 14, 14, 15, 15, 15, 16, 17, 25
3. Reste el mínimo del máximo. 25-10 = **15**

Como calcular la IQR de las distintas notas:

In [29]:
```
rango_1 = df["notas_1"].max() - df["notas_1"].min()
iqr_1 = df["notas_1"].quantile(0.75) - df["notas_1"].quantile(0.25)
print(f"IQR de las notas 1: {iqr_1}, rango: {rango_1}")
```

```
IQR de las notas 1: 1.75, rango: 15
```

In [30]:
```
rango_2 = df["notas_2"].max() - df["notas_2"].min()
iqr_2 = df["notas_2"].quantile(0.75) - df["notas_2"].quantile(0.25)
print(f"IQR de las notas 2: {iqr_2}, rango: {rango_2}")
```

```
IQR de las notas 2: 3.25, rango: 9
```

In [31]:
```
rango_3 = df["notas_3"].max() - df["notas_3"].min()
iqr_3 = df["notas_3"].quantile(0.75) - df["notas_3"].quantile(0.25)
print(f"IQR de las notas 3: {iqr_3}, rango: {rango_3}")
```

```
IQR de las notas 3: 5.75, rango: 10
```

Encontrar valores atípicos

- **Notas 1:**

--> Superiores:

In [32]:
```
superiores_1 = df["notas_1"].quantile(0.75) + 1.5 * iqr_1
print(superiores_1)
```

```
18.375
```

In [33]:
```
df.notas_1
```

Out[33]:
```
0    15
1    16
2    15
3    17
4    14
5    14
6    14
7    10
8    15
9    25
Name: notas_1, dtype: int64
```

Todos los valores superiores a 18.375 son outliers, en nuestro caso es el valor 25.

--> Inferiores

In [34]:
```
inferiores_1 = df["notas_1"].quantile(0.25) - 1.5 * iqr_1
inferiores_1
```

Out[34]:
```
np.float64(11.375)
```

Todos los valores inferiores a 11.375 son considerados outliers, en este caso el valor 10.

- **Notas 2:**

--> Superiores:

In [35]:
```
superiores_2 = df["notas_2"].quantile(0.75) + 1.5 * iqr_2
print(superiores_2)
```

```
23.125
```

In [36]:
```
df.notas_2
```

Out[36]:
```
0    16
1    21
2    16
3    16
4    13
5    15
6    15
7    19
8    22
9    15
Name: notas_2, dtype: int64
```

--> Inferiores

In [37]:
```
inferiores_2 = df["notas_2"].quantile(0.25) - 1.5 * iqr_2
inferiores_2
```

Out[37]:
```
np.float64(10.125)
```

- **Notas 3:**

--> Superiores:

In [38]:
```
superiores_3 = df["notas_3"].quantile(0.75) + 1.5 * iqr_3
print(superiores_3)
```

```
29.375
```

In [39]:
```
df.notas_3
```

Out[39]:
```
0    17
1    22
2    15
3    22
4    14
5    15
6    16
7    15
8    24
9    16
Name: notas_3, dtype: int64
```

--> Inferiores

In [40]:
```
inferiores_3 = df["notas_3"].quantile(0.25) - 1.5 * iqr_3
inferiores_3
```

Out[40]:
```
np.float64(6.375)
```

Máximos, Mínimos, cuartiles (Q3, Q1), Mediana/Media (Q2)

Los **cuartiles** de un grupo de datos son las medianas de las mitades superior e inferior de ese conjunto. El cuartil inferior, Q1, es la mediana de la mitad inferior, mientras que el cuartil superior, Q3, es la mediana de la mitad superior. Si su conjunto de datos tiene una cantidad impar de puntos de datos, no tendrá en cuenta la mediana al encontrar estos valores, pero si su conjunto de datos contiene una cantidad par de puntos de datos, tendrá en cuenta ambos valores medios que utilizó para encontrar la mediana como partes de las mitades superior e inferior.

Ordene los datos de menor a mayor. 10, 14, 14, 14, 15, 15, 15, 16, 17, 25

Q = a/4 * N donde:

- a es el cuartil 1, 2, 3
- N es el numero de datos

Q = 0.25 * 10 = 2.5 tomamos los valores entre la posición 2, 3 => 14, 14

con esos datos Q1 = valor 2 + 0.25 (valor 3 - valor 2);

Q1 = 14 + 0.25(14 -14) = 14

Q = 0.75 * 10 = 7.5 buscamos los datos entre la posición 7, 8 => 15, 16

Q3 = 15 + 0.75 (16-15) = 15.75

- **Notas 1:**

In [41]:
```
max_1 = df["notas_1"].max()
min_1 = df["notas_1"].min()
q3_1 = df["notas_1"].quantile(0.75)
q1_1 = df["notas_1"].quantile(0.25)
mediana_1 = df["notas_1"].median()
media_1 = df["notas_1"].mean()
```

In [42]:
```
print(f"Maximo: {max_1}, Mínimo: {min_1}, Q3: {q3_1}, Q1: {q1_1}, Media:
```

```
Maximo: 25, Mínimo: 10, Q3: 15.75, Q1: 14.0, Media: 15.5
```

- **Notas 2:**

In [43]:
```
max_2 = df["notas_2"].max()
min_2 = df["notas_2"].min()
q3_2 = df["notas_2"].quantile(0.75)
q1_2 = df["notas_2"].quantile(0.25)
mediana_2 = df["notas_2"].median()
media_2 = df["notas_2"].mean()
```

In [44]:
```
print(f"Maximo: {max_2}, Mínimo: {min_2}, Q3: {q3_2}, Q1: {q1_2}, Media:
```

```
Maximo: 22, Mínimo: 13, Q3: 18.25, Q1: 15.0, Media: 16.8
```

- **Notas 3:**

In [45]:
```
max_3 = df["notas_3"].max()
min_3 = df["notas_3"].min()
q3_3 = df["notas_3"].quantile(0.75)
q1_3 = df["notas_3"].quantile(0.25)
mediana_3 = df["notas_3"].median()
media_3 = df["notas_3"].mean()
```

In [46]:
```
print(f"Maximo: {max_3}, Mínimo: {min_3}, Q3: {q3_3}, Q1: {q1_3}, Media:
```

```
Maximo: 24, Mínimo: 14, Q3: 20.75, Q1: 15.0, Media: 17.6
```

In [47]:
```
df.describe()
```

Out[47]:

	notas_1	notas_2	notas_3
count	10.000000	10.000000	10.000000
mean	15.500000	16.800000	17.600000
std	3.807887	2.898275	3.627059
min	10.000000	13.000000	14.000000
25%	14.000000	15.000000	15.000000
50%	15.000000	16.000000	16.000000
75%	15.750000	18.250000	20.750000
max	25.000000	22.000000	24.000000

Resultados notas 1:

In [48]:
```
print(f'Desviación estándar: {std_1}, Varianza: {varianza_1}, Rango: {ran
 Outlier Sup: {superiores_1}, Outlier Inf: {inferiores_1}')
```

```
Desviación estándar: 3.6124783736376886, Varianza: 13.05, Rango: 15,
IQR: 1.75 Outlier Sup: 18.375, Outlier Inf: 11.375
```

Resultados notas 2:

In [49]:
```
print(f'Desviación estándar: {std_2}, Varianza: {varianza_2}, Rango: {ran
 Outlier Sup: {superiores_2}, Outlier Inf: {inferiores_2}')
```

```
Desviación estándar: 2.749545416973504, Varianza: 7.56, Rango: 9,
IQR: 3.25 Outlier Sup: 23.125, Outlier Inf: 10.125
```

Resultados notas 3:

In [50]:
```
print(f'Desviación estándar: {std_3}, Varianza: {varianza_3}, Rango: {ran
 Outlier Sup: {superiores_3}, Outlier Inf: {inferiores_3}')
```

```
Desviación estándar: 3.4409301068170506, Varianza: 11.84, Rango: 10,
IQR: 5.75 Outlier Sup: 29.375, Outlier Inf: 6.375
```

Identificar distribuciones estadísticas fundamentales (gaussianas, uniformes) e interpretar sus tendencias en varios contextos (a lo largo del tiempo, univariadas, bivariadas, multivariadas).

Necesidad de distribución de probabilidad

Las distribuciones de probabilidad son herramientas versátiles que se utilizan en varios campos y aplicaciones. Principalmente modelan y cuantifican la incertidumbre y la variabilidad de los datos, lo que las hace fundamentales en la ciencia de datos, las estadísticas y los procesos de toma de decisiones. Las distribuciones de probabilidad nos permiten analizar datos y sacar conclusiones significativas al describir la probabilidad de diferentes resultados o eventos.

En el análisis estadístico, estas distribuciones desempeñan un papel fundamental en la estimación de parámetros, la prueba de hipótesis y la inferencia de datos. También se utilizan ampliamente en la evaluación de riesgos, en particular en finanzas y seguros, donde ayudan a evaluar y gestionar los riesgos financieros al cuantificar la probabilidad de varios resultados.

Los algoritmos de aprendizaje automático aprovechan las distribuciones de probabilidad para modelar la incertidumbre en las predicciones, lo que mejora su capacidad para realizar pronósticos precisos. Además, las distribuciones de probabilidad respaldan los esfuerzos de control de calidad, lo que permite monitorear y controlar los procesos al identificar desviaciones de los valores esperados.

Las distribuciones de probabilidad no se limitan únicamente al análisis de datos; también desempeñan papeles cruciales en campos como la ingeniería, la ciencia ambiental, la epidemiología y la física. En estos diversos dominios, las distribuciones de probabilidad permiten un modelado, simulación y predicción confiables, lo que en última instancia contribuye a la toma de decisiones y la resolución de problemas informadas.

Tipos de datos comunes

Antes de pasar a la explicación de las distribuciones, veamos qué tipo de datos podemos encontrar. Los datos pueden ser discretos o continuos.

- Los **datos discretos**, como sugiere el nombre, solo pueden tomar valores específicos. Por ejemplo, cuando se lanza un dado, los resultados posibles son 1, 2, 3, 4, 5 o 6, no 1,5 o 2,45. (Distribución de probabilidad discreta)
- Los **datos continuos** pueden tomar cualquier valor dentro de un rango determinado. El rango puede ser finito o infinito. Por ejemplo, el peso o la altura de una niña, la longitud de la carretera. El peso de una niña puede ser cualquier valor: 54 kg, 54,5 kg o 54,5436 kg. (Distribución de probabilidad continua)

Distribución uniforme

Cuando se lanza un dado, los resultados son de 1 a 6. Las probabilidades de obtener estos resultados son igualmente probables, lo que constituye la base de una distribución uniforme. A diferencia de la distribución de Bernoulli, todos los n resultados posibles de una distribución uniforme son igualmente probables.

Se dice que una variable X está distribuida uniformemente si la función de densidad es:

In [6]:
```
display.Image("./images/image_1.png")
```

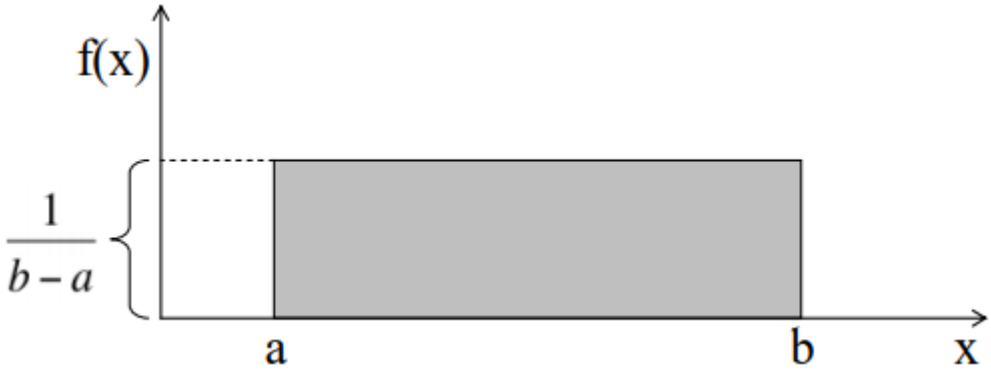
Out[6]:

$$f(x) = \frac{1}{b-a} \qquad for\ -\infty < a \leq x \leq b < \infty$$

Función de densidad El gráfico de una curva de distribución uniforme se ve así

In [7]:
```
display.Image("./images/image_2.png")
```

Out[7]:



Curva de distribución uniforme | distribución de probabilidad

Puedes ver que la forma de la curva de distribución uniforme es rectangular, por lo que la distribución uniforme se denomina distribución rectangular.

Para una distribución uniforme, a y b son los parámetros.

Ejemplo de distribución uniforme

El número de ramos que se venden diariamente en una floristería se distribuye uniformemente, con un máximo de 40 y un mínimo de 10.

Intentemos calcular la probabilidad de que las ventas diarias se encuentren entre 15 y 30.

La probabilidad de que las ventas diarias se encuentren entre 15 y 30 es (30-15)*(1/(40-10)) = 0,5

De manera similar, la probabilidad de que las ventas diarias sean mayores de 20 es = 0,667

La media y la varianza de X siguiendo una distribución uniforme son:

Media -> E(X) = (a+b)/2

Varianza -> V(X) = $(b-a)^2/n-1$

La densidad uniforme estándar tiene parámetros a = 0 y b = 1.

Distribución normal vs. distribución gaussiana

La distribución normal representa el comportamiento de la mayoría de las situaciones del universo (por eso se la llama distribución “normal”). distribución. ¡Supongo!). La gran suma de variables aleatorias (pequeñas) a menudo resulta tener una distribución normal, lo que contribuye a su amplia aplicación. Cualquier distribución se conoce como distribución normal si tiene las siguientes características:

1. La media, la mediana y la moda de la distribución coinciden.
2. La curva de la distribución tiene forma de campana y es simétrica respecto de la línea x=μ.
3. El área total bajo la curva es 1.
4. Exactamente la mitad de los valores están a la izquierda del centro y la otra mitad a la derecha.

Una distribución normal es muy diferente de la distribución binomial. Sin embargo, si el número de ensayos se acerca al infinito, las formas serán bastante similares.

La función de densidad de probabilidad de una variable aleatoria X, siguiendo una distribución normal, está dada por:

In [8]:
```
display.Image("./images/image_3.png")
```

Out[8]:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\{-\frac{1}{2}(\frac{x-u}{\sigma})^2\}} \qquad for -\infty < x < \infty.$$

La media y la varianza de una variable aleatoria X, que se dice que tiene una distribución normal, se expresan de la siguiente manera:

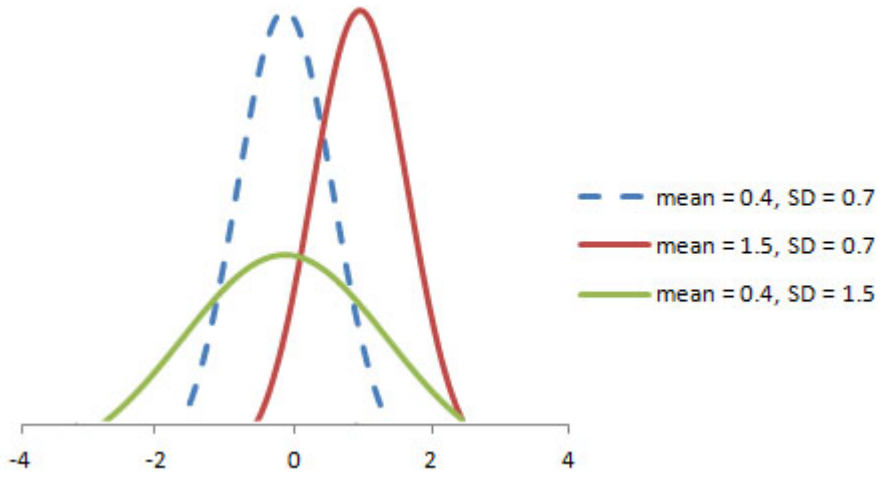
- Media -> E(X) = μ
- Varianza -> Var(X) = σ^2

Aquí, μ (media) y σ (desviación estándar) son los parámetros.

A continuación se muestra el gráfico de una variable aleatoria X ~ N (μ, σ).

In [9]:
```
display.Image("./images/image_4.png")
```

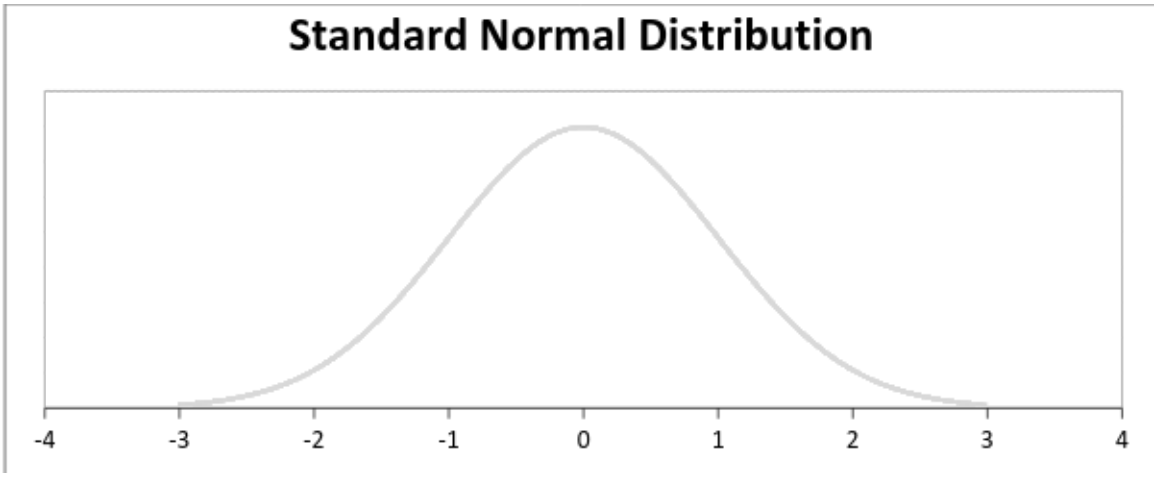
Out[9]:



Una distribución normal estándar se define como una distribución con una media de 0 y una desviación estándar de 1. Para tal caso, la PDF se convierte en:

In [10]:
```
display.Image("./images/image_5.png")
```

Out[10]:



Diferencias entre distribuciones normales y uniformes

1. Forma:

Distribución uniforme: rectangular, con todos los resultados igualmente probables.

Distribución normal: en forma de campana, con la mayoría de los resultados agrupados alrededor de la media.

2. Probabilidad:

Distribución uniforme: probabilidad constante en todo el rango.

Distribución normal: probabilidad más alta en la media, disminuyendo simétricamente a medida que se aleja de la media.

3. Dispersión:

Distribución uniforme: uniforme y constante.

Distribución normal: determinada por la desviación estándar, con la mayoría de los puntos de datos concentrados alrededor de la media.

4. Parámetros:

Distribución uniforme: definida por los valores mínimos a y máximos b.

Distribución normal: definida por la media μ y la desviación estándar σ.

Datos univariados:

Los datos univariados se refieren a un tipo de datos en el que cada observación o punto de datos corresponde a una sola variable. En otras palabras, implica la medición u observación de una sola característica o atributo para cada individuo o elemento del conjunto de datos. El análisis de datos univariados es la forma más simple de análisis en estadística.

Alturas (en cm)
164
167.3
170
174.2
178
180
186

Supongamos que se registran las alturas de siete estudiantes de una clase (tabla anterior). Solo hay una variable, que es la altura, y no se relaciona con ninguna causa o relación.

Puntos clave en el análisis univariado:

- Sin relaciones: el análisis univariado se centra únicamente en describir y resumir la distribución de la variable única. No explora las relaciones entre las variables ni intenta identificar las causas.

- Estadísticas descriptivas: las estadísticas descriptivas, como las medidas de tendencia central (media, mediana, moda) y las medidas de dispersión (rango, desviación estándar), se utilizan comúnmente en el análisis de datos univariados.
- Visualización: los histogramas, los diagramas de caja y otras representaciones gráficas se utilizan a menudo para representar visualmente la distribución de la variable única.

Datos bivariados

Los datos bivariados involucran dos variables diferentes, y el análisis de este tipo de datos se centra en comprender la relación o asociación entre estas dos variables. Un ejemplo de datos bivariados puede ser la temperatura y las ventas de helados en la temporada de verano.

Temperatura	Ventas de helados
20	2000
25	2500
35	5000

Supongamos que la temperatura y las ventas de helado son las dos variables de datos bivariados (tabla 2). Aquí, la relación es visible en la tabla: la temperatura y las ventas son directamente proporcionales entre sí y, por lo tanto, están relacionadas porque a medida que aumenta la temperatura, también aumentan las ventas.

Puntos clave en el análisis bivariado:

- Análisis de relaciones: el objetivo principal del análisis de datos bivariados es comprender la relación entre las dos variables. Esta relación puede ser positiva (ambas variables aumentan juntas), negativa (una variable aumenta mientras que la otra disminuye) o no mostrar un patrón claro.
- Diagramas de dispersión: una herramienta de visualización común para datos bivariados es un diagrama de dispersión, donde cada punto de datos representa un par de valores para las dos variables. Los diagramas de dispersión ayudan a visualizar patrones y tendencias en los datos.
- Coeficiente de correlación: una medida cuantitativa llamada coeficiente de correlación se utiliza a menudo para cuantificar la fuerza y la dirección de la relación lineal entre dos variables. El coeficiente de correlación varía de -1 a 1.

Datos multivariados

Los datos multivariados se refieren a conjuntos de datos en los que cada observación o punto de muestra consta de múltiples variables o características. Estas variables pueden representar diferentes aspectos, características o mediciones relacionadas con el fenómeno observado. Cuando se trata de tres o más variables, los datos se clasifican específicamente como multivariados.

Un ejemplo de este tipo de datos es el de un anunciante que desea comparar la popularidad de cuatro anuncios en un sitio web.

Publicidad	Género	Tasa de clics
Ad1	Masculino	80
Ad3	Femenino	55
Ad2	Femenino	123
Ad1	Masculino	66
Ad3	Masculino	35

Las tasas de clics se pueden medir tanto para hombres como para mujeres y luego se pueden examinar las relaciones entre las variables. Es similar al análisis bivariado pero contiene más de una variable dependiente.

Puntos clave en el análisis multivariado:

- Técnicas de análisis: Las formas de realizar el análisis de estos datos dependen de los objetivos que se deseen alcanzar. Algunas de las técnicas son el análisis de regresión, el análisis de componentes principales, el análisis de trayectorias, el análisis factorial y el análisis multivariado de varianza (ANOVA).
- Objetivos del análisis: La elección de la técnica de análisis depende de los objetivos específicos del estudio. Por ejemplo, los investigadores pueden estar interesados en predecir una variable en función de otras, identificar factores subyacentes que expliquen patrones o comparar medias de grupos en múltiples variables. - Interpretación: El análisis multivariado permite una interpretación más matizada de relaciones complejas dentro de los datos. Ayuda a descubrir patrones que pueden no ser evidentes al examinar las variables individualmente.

Existen muchas herramientas, técnicas y métodos diferentes que se pueden utilizar para realizar su análisis. Puede utilizar bibliotecas de software, herramientas de visualización y métodos de prueba estadística. Sin embargo, en este blog compararemos el análisis univariado, bivariado y multivariado.

Diferencia entre datos univariados, bivariados y multivariados

Univariado	Bivariado	Multivariado
Solo resume una variable a la vez.	Solo resume dos variables	Solo resume más de 2 variables.
No se ocupa de causas y relaciones.	Se ocupa de causas y relaciones y se realiza el análisis.	No se ocupa de causas y relaciones y se realiza el análisis.
No contiene ninguna variable dependiente.	Solo contiene una variable dependiente.	Es similar al bivariado pero contiene más de 2 variables.
El propósito principal es describir.	El propósito principal es explicar.	El propósito principal es estudiar la relación entre ellos.
El ejemplo de un univariado puede ser la altura.	El ejemplo de un bivariado puede ser la temperatura y las ventas de hielo en las vacaciones de verano.	Ejemplo: supongamos que un anunciante quiere comparar la popularidad de cuatro anuncios en un sitio web.

Entonces, se podrían medir sus tasas de clics tanto para hombres como para mujeres y se pueden examinar las relaciones entre las variables.

Aplicar medidas de confianza en cálculos estadísticos para evaluar la confiabilidad de los datos.

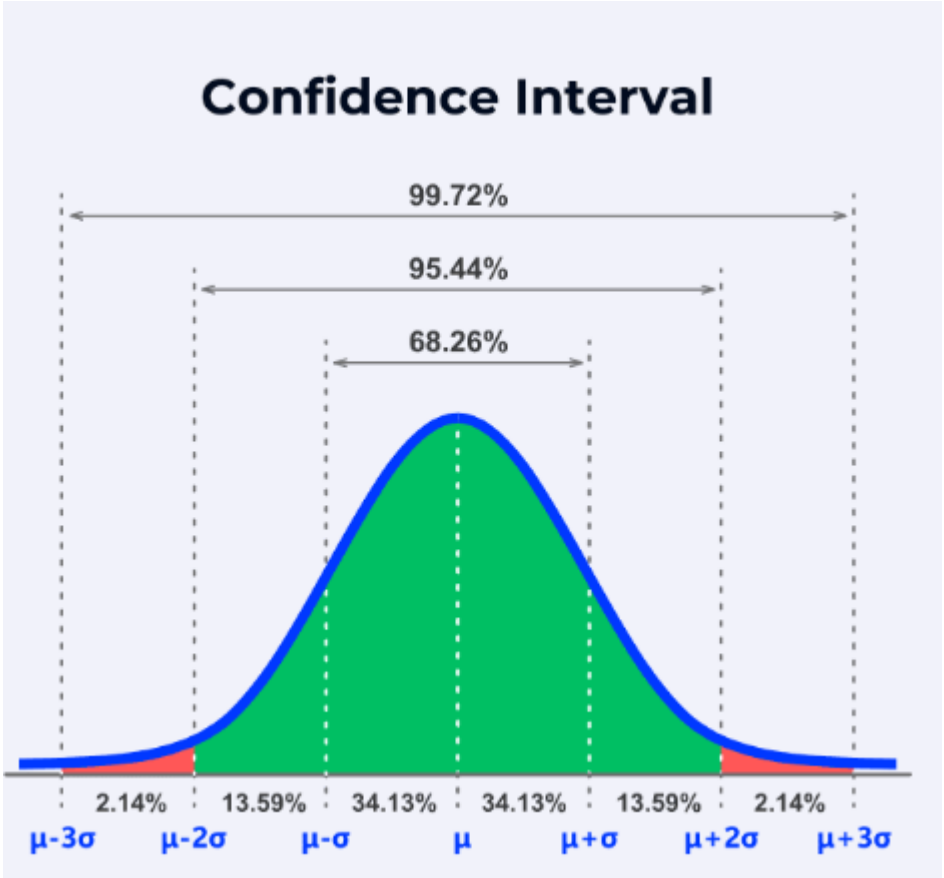
¿Qué es un intervalo de confianza?

Un intervalo de confianza es una herramienta estadística que se utiliza para estimar el rango de valores dentro del cual es probable que se encuentre un parámetro de población, como una media o proporción de población. Proporciona una medida de incertidumbre en torno a una estimación puntual derivada de datos de muestra.

Los intervalos de confianza se construyen en función de estadísticas de muestra, como la media o proporción de muestra, y suelen ir acompañados de un nivel de confianza específico, como el 95 % o el 99 %. El nivel de confianza indica la probabilidad de que el intervalo calculado contenga el parámetro poblacional verdadero en un muestreo repetido.

In [11]:
```
display.Image("./images/image_6.png")
```

Out[11]:



Importancia del intervalo de confianza en el análisis estadístico

- Cuantificación de la incertidumbre: los intervalos de confianza proporcionan una medida de la incertidumbre en torno a las estimaciones puntuales, lo que permite a los investigadores evaluar la fiabilidad y precisión de sus hallazgos. Los intervalos

de confianza ayudan a evitar el exceso de confianza en las estimaciones estadísticas al reconocer y cuantificar la incertidumbre.

- Inferencia y toma de decisiones: los intervalos de confianza desempeñan un papel crucial en la inferencia estadística y la toma de decisiones. Permiten a los investigadores realizar inferencias sobre los parámetros poblacionales basándose en datos de muestra, lo que orienta las decisiones en investigación, negocios, atención médica y formulación de políticas.
- Comparación de grupos o tratamientos: los intervalos de confianza facilitan las comparaciones entre grupos o tratamientos al proporcionar un rango de valores plausibles para los parámetros poblacionales. Ya sea que se comparen medias, proporciones u otras estadísticas, los intervalos de confianza ayudan a evaluar la importancia y la magnitud de las diferencias.
- Determinación del tamaño de la muestra: los intervalos de confianza informan sobre la determinación del tamaño de la muestra para los estudios de investigación. Al especificar el nivel deseado de precisión y confianza, los investigadores pueden calcular el tamaño de la muestra necesario para alcanzar los objetivos de su estudio y, al mismo tiempo, minimizar los costos y los recursos.
- Comunicación de resultados: los intervalos de confianza ofrecen una forma concisa de comunicar la precisión y la incertidumbre de las estimaciones estadísticas a las partes interesadas, incluidos los investigadores, los responsables de las políticas y el público en general. Proporcionan una indicación clara del rango dentro del cual es probable que se encuentre el verdadero parámetro de la población.
- Robustez ante los supuestos: a diferencia de las estimaciones puntuales, que pueden ser sensibles a los valores atípicos o las violaciones de los supuestos distributivos, los intervalos de confianza son más robustos y proporcionan una imagen más completa de la incertidumbre subyacente. Ofrecen un enfoque flexible para el análisis estadístico, en particular en situaciones en las que los supuestos paramétricos pueden no cumplirse.
- Control de calidad y mejora de procesos: en el control de calidad y la mejora de procesos, los intervalos de confianza se utilizan para monitorear y evaluar el desempeño de los sistemas y procesos. Al realizar un seguimiento de los intervalos de confianza a lo largo del tiempo, las organizaciones pueden identificar tendencias, detectar desviaciones del desempeño esperado e implementar acciones correctivas según sea necesario.
- Reproducibilidad científica: los intervalos de confianza contribuyen a la transparencia y reproducibilidad de la investigación científica al cuantificar la incertidumbre inherente a las estimaciones estadísticas. Los estudios de replicación pueden utilizar intervalos de confianza para evaluar la coherencia y la generalización de los hallazgos en diferentes muestras o entornos.
- Toma de decisiones en situaciones de incertidumbre: en contextos de toma de decisiones, los intervalos de confianza proporcionan a los responsables de la toma de decisiones un marco para considerar la incertidumbre y la variabilidad en sus elecciones. Ya sea que se evalúe la eficacia de las intervenciones, se evalúen los riesgos o se asignen recursos, los intervalos de confianza permiten tomar decisiones más fundamentadas y sólidas.

Entendimiento de los intervalos de confianza

Los intervalos de confianza son una piedra angular de la inferencia estadística, ya que nos permiten estimar parámetros de población con un cierto grado de incertidumbre. En esencia, un intervalo de confianza es un rango de valores derivados de datos de muestra que probablemente contengan el parámetro de población real.

Imagina que estás tratando de estimar la altura promedio de todos los adultos de un país. En lugar de basarte únicamente en la altura media de la muestra, que podría variar de una muestra a otra, un intervalo de confianza proporciona un rango de valores plausibles dentro de los cuales se espera que se encuentre la media de la población real. Este rango se expresa con un nivel de confianza específico, normalmente el 95 % o el 99 %.

Interpretación del intervalo de confianza

Interpretar un intervalo de confianza implica comprender qué representa el intervalo y qué no. Es fundamental comprender que el nivel de confianza asociado con un intervalo se refiere al porcentaje de intervalos de confianza, derivados de un muestreo repetido, que contendrían el parámetro de población real. Por ejemplo, si construimos 100 intervalos de confianza con un nivel de confianza del 95 %, esperaríamos que aproximadamente 95 de ellos contuvieran el parámetro de población real.

Al comunicar los resultados de un intervalo de confianza, es fundamental enfatizar que proporciona un rango de valores plausibles, no una estimación puntual específica. Además, el intervalo de confianza solo cuantifica la incertidumbre debido a la variabilidad del muestreo y no tiene en cuenta otras fuentes de incertidumbre o sesgo.

¿Cómo calcular el intervalo de confianza? El cálculo de un intervalo de confianza depende de varios factores, incluido el tamaño de la muestra, la variabilidad de la población y el nivel de confianza deseado. Para datos distribuidos normalmente con una desviación estándar poblacional conocida, la fórmula para calcular un intervalo de confianza para la media poblacional (μ) es:

$$CI = \bar{x} \pm Z(\sigma/\sqrt{n})$$

Donde:

- $\bar{x}$ es la media de la muestra.
- σ es la desviación estándar de la población.
- n es el tamaño de la muestra.
- Z es el valor crítico de la distribución normal estándar correspondiente al nivel de confianza deseado.

Para los casos en los que se desconoce la desviación estándar de la población o el tamaño de la muestra es pequeño, se utiliza la distribución t en lugar de la distribución normal estándar. Este ajuste tiene en cuenta la incertidumbre adicional introducida al estimar la desviación estándar de la población a partir de los datos de la muestra.

Tabla pequeña de valores z para intervalos de confianza

Nivel de confianza	z
0,70	1,04
0,75	1,15
0,80	1,28
0,85	1,44
0,90	1,645
0,92	1,75
0,95	1,96
0,96	2,05
0,98	2,33
0,99	2,58

Ejemplo:

Supongamos que queremos estimar el tiempo promedio que los clientes pasan en una tienda. Recopilamos una muestra de 100 clientes y descubrimos que el tiempo promedio que pasan es de 30 minutos, con una desviación estándar de 5 minutos. Si queremos construir un intervalo de confianza del 95 % para el tiempo promedio que pasa la población, podemos usar la fórmula:

`x̄ = 30`

`σ = 5`

`n = 100`

`Z = 95% --> 1.96`

```
CI = 30 ± 1.96(5/√100)

CI = 30 ± 0.98
```

Por lo tanto, el intervalo de confianza del 95 % para el tiempo medio de permanencia de los clientes en la tienda es de aproximadamente 29,02 a 30,98 minutos.

Implicaciones de los niveles de confianza

La elección del nivel de confianza tiene varias implicaciones:

- Precisión frente a certeza: un nivel de confianza más alto (p. ej., 99 %) da como resultado un intervalo de confianza más amplio, lo que refleja una mayor certeza de que el intervalo contiene el parámetro verdadero pero menos precisión sobre su valor. Por el contrario, un nivel de confianza más bajo (p. ej., 90 %) produce un intervalo más estrecho, lo que ofrece más precisión pero menos certeza.

- Significación estadística: en las pruebas de hipótesis, un intervalo de confianza que no contiene el valor de la hipótesis nula (a menudo cero) indica un resultado estadísticamente significativo en el nivel de confianza elegido. Por ejemplo, un intervalo de confianza del 95 % que no incluye el cero sugiere un efecto estadísticamente significativo con un riesgo del 5 % de un falso positivo (error de tipo I).
- Interpretación: Los niveles de confianza deben interpretarse en el contexto del estudio y del proceso de toma de decisiones. Proporcionan un rango de valores plausibles para el parámetro de interés, pero no garantizan que el valor verdadero se encuentre dentro de un intervalo único calculado a partir de una muestra.
- Falsos positivos: Incluso con un nivel de confianza alto, siempre existe la posibilidad de observar un resultado estadísticamente significativo por pura casualidad. Esto se conoce como falso positivo y el riesgo es igual al 100 % menos el nivel de confianza.

3.1.2 – Analizar y evaluar las relaciones de datos.

La correlación es el análisis estadístico de la relación o dependencia entre dos variables. La correlación nos permite estudiar tanto la fuerza como la dirección de la relación entre dos conjuntos de variables.

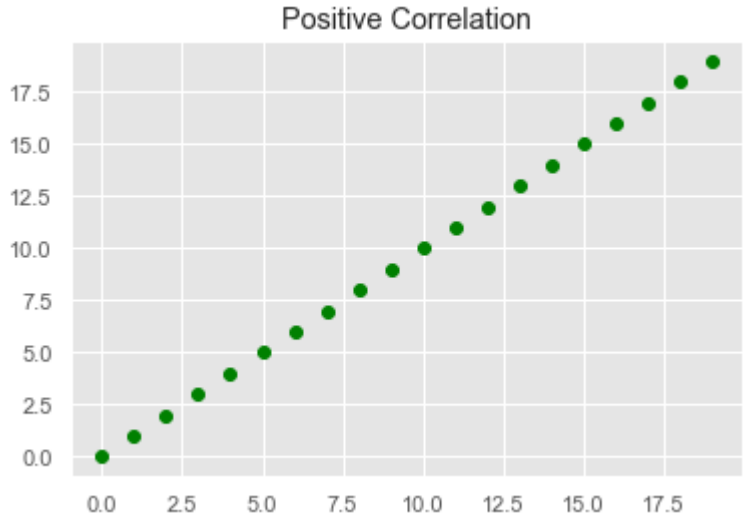
El estudio de la correlación es fundamental en el campo del aprendizaje automático. Por ejemplo, algunos algoritmos no funcionarán correctamente si dos o más variables están estrechamente relacionadas, lo que generalmente se conoce como multicolinealidad . La correlación también es la base del Análisis de Componentes Principales , una técnica de reducción de dimensionalidad lineal que es muy útil en proyectos de aprendizaje automático.

Tipos:

Correlación positiva: se dice que dos variables están correlacionadas positivamente cuando sus valores se mueven en la misma dirección. Por ejemplo, en la imagen a continuación, a medida que aumenta el valor de X, también lo hace el valor de Y a una tasa constante:

In [12]:
```
display.Image("./images/posit.png")
```

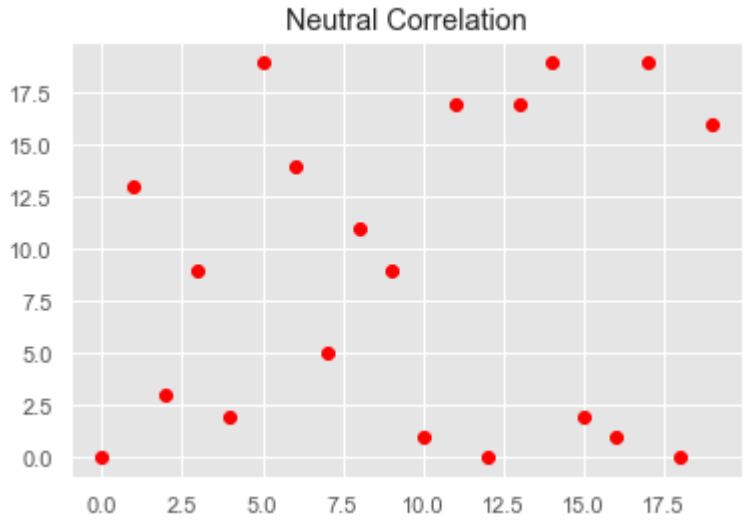
Out[12]:



Correlación Neutral: No hay relación en el cambio de las variables X e Y. En este caso los valores son completamente aleatorios y no muestran ningún signo de correlación, como se muestra en la siguiente imagen:

In [13]:
```
display.Image("./images/neutra.png")
```

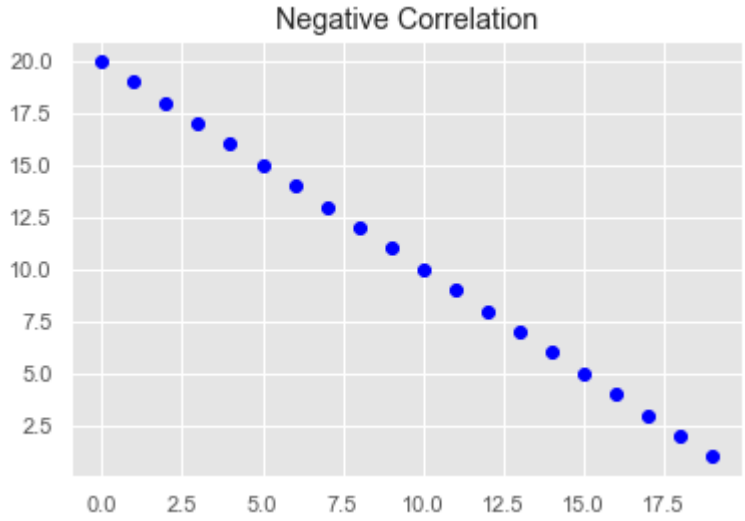
Out[13]:



Correlación negativa: finalmente, las variables X e Y estarán negativamente correlacionadas cuando sus valores cambien en direcciones opuestas, por lo que aquí, a medida que aumenta el valor de X, el valor de Y disminuye a una tasa constante:

In [14]:
```
display.Image("./images/negat.png")
```

Out[14]:



Coeficientes de correlación

Un coeficiente de correlación es un resumen estadístico que mide la fuerza y la dirección con la que se asocian dos variables entre sí.

Una de las ventajas de los coeficientes de correlación es que estiman la correlación entre dos variables de forma estandarizada, por lo que el valor del coeficiente siempre estará en la misma escala, variando de -1,0 a 1,0.

1. Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson (r) es una puntuación que mide la fuerza de una relación lineal entre dos variables. Se calcula dividiendo la covarianza de las variables X e Y por el producto de la desviación estándar de cada variable, como se muestra en la siguiente fórmula:

In [15]:
```
display.Image("./images/pearson.png")
```

Out[15]:

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

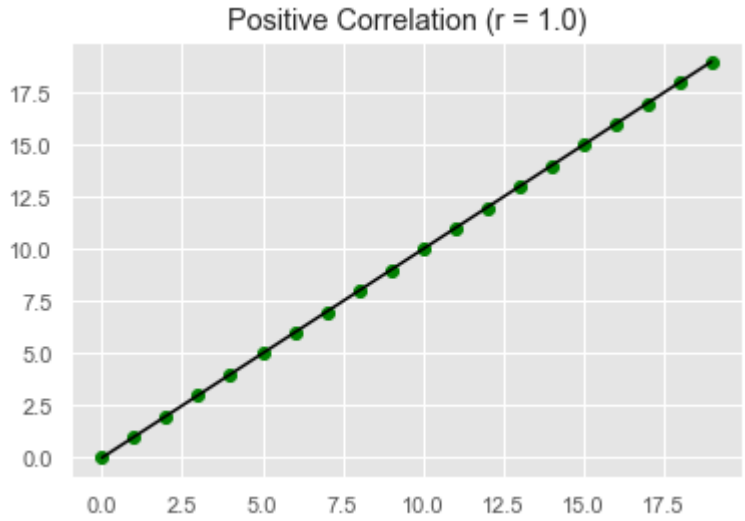
El coeficiente se basa en dos supuestos. Primero, asume que las variables siguen una distribución normal o gaussiana. Si los datos no se distribuyen normalmente, entonces otros coeficientes pueden ser más confiables.

En segundo lugar, supone que existe una relación lineal entre las dos variables, lo que significa que los cambios en los datos se pueden modelar mediante una función lineal (es decir, sus valores aumentan o disminuyen simultáneamente a una tasa constante). Si la relación entre las dos variables es más cercana a una línea recta, entonces su correlación (lineal) es más fuerte y el valor absoluto del coeficiente de correlación de Pearson es más alto. Por ejemplo, en la siguiente imagen, todos los puntos de datos se

pueden modelar perfectamente utilizando una línea recta, lo que da como resultado un coeficiente de correlación igual a 1,0.

In [16]:
```
display.Image("./images/r1.png")
```

Out[16]:



Un coeficiente de -1,0 indica una correlación negativa perfecta, mientras que un coeficiente de 1,0 muestra una correlación positiva perfecta. Por el contrario, un coeficiente de 0,0 indica que no existe una correlación lineal entre las variables.

Calculo:

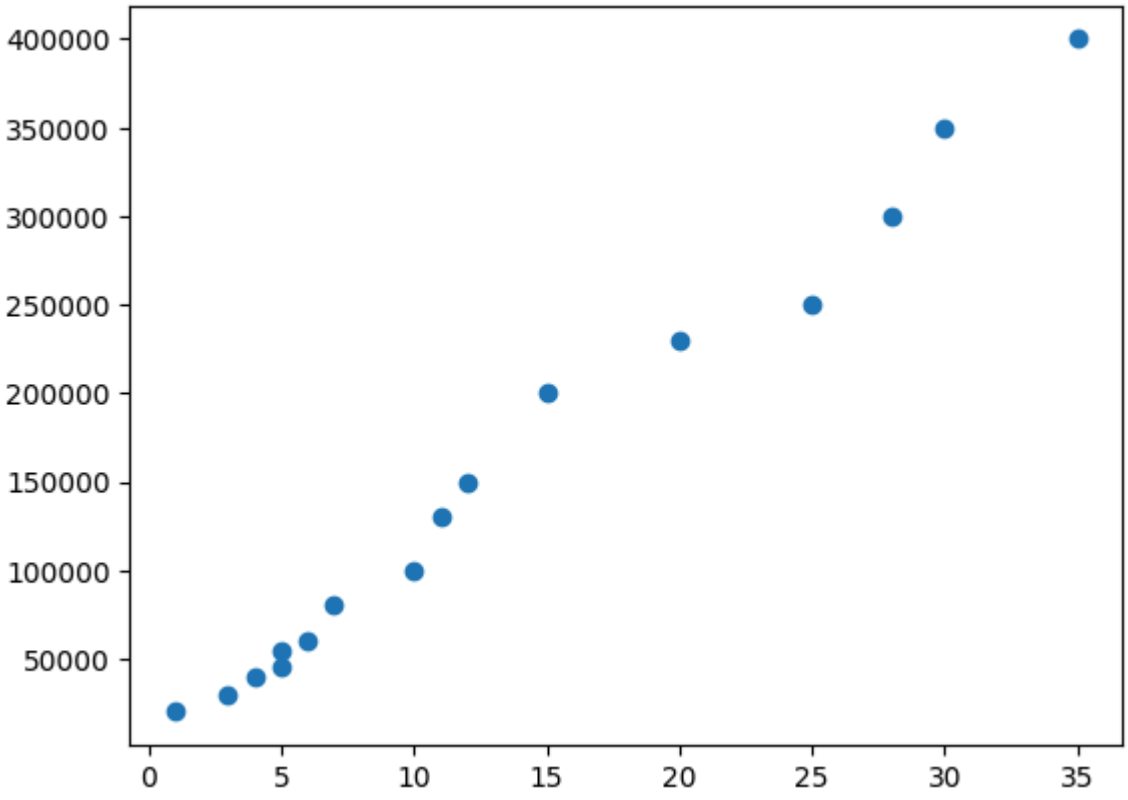
In [62]:
```
experience = [1, 3, 4, 5, 5, 6, 7, 10, 11, 12, 15, 20, 25, 28, 30,35]

salary = [20000, 30000, 40000, 45000, 55000, 60000, 80000, 100000, 130000
```

In [63]:
```
import pandas as pd

df = pd.DataFrame({"Experience": experience, "Salary": salary})
df
```

Out[63]:

	Experience	Salary
0	1	20000
1	3	30000
2	4	40000
3	5	45000
4	5	55000
5	6	60000
6	7	80000
7	10	100000
8	11	130000
9	12	150000
10	15	200000
11	20	230000
12	25	250000
13	28	300000
14	30	350000
15	35	400000

In [64]:
```
import matplotlib.pyplot as plt

plt.scatter(df.Experience, df.Salary)
#plt.plot(df.Experience, df.Salary, color='red', linewidth=2)
plt.show()
```



- scipy librería

In [65]:
```
import scipy.stats as stats

corr, _ = stats.pearsonr (experience, salary)
corr
```

Out[65]: np.float64(0.9929845761480397)

In [66]:
```
# Otros coeficientes:

spearman_corr, _ = stats.spearmanr(experience, salary)
print("spearman:", spearman_corr)

kendall_corr, _ = stats.kendalltau(experience, salary)
print("Kendall:", kendall_corr)
```

```
spearman: 0.9992644353546791
Kendall: 0.9958246164193105
```

- Numpy librería

In [67]:
```
import numpy as np

np.corrcoef(df.Experience, df.Salary)
```

Out[67]:
```
array([[1.        , 0.99298458],
       [0.99298458, 1.        ]])
```

Una matriz de correlación es una tabla que muestra los coeficientes de correlación entre variables. Cada celda de la tabla muestra la correlación entre dos variables. La diagonal de la matriz incluye los coeficientes entre cada variable y ella misma, que siempre es igual a 1,0. Los demás valores de la matriz representan la correlación entre experiencia y salario. En este caso, como solo estamos calculando correlación para dos variables, los valores son los mismos.

- Pandas librería

In [68]:
```
df['Experience'].corr(df['Salary'])
```

Out[68]: np.float64(0.9929845761480398)

In [69]:
```
print(df['Experience'].corr(df['Salary'], method='spearman'))
print(df['Experience'].corr(df['Salary'], method='kendall'))
```

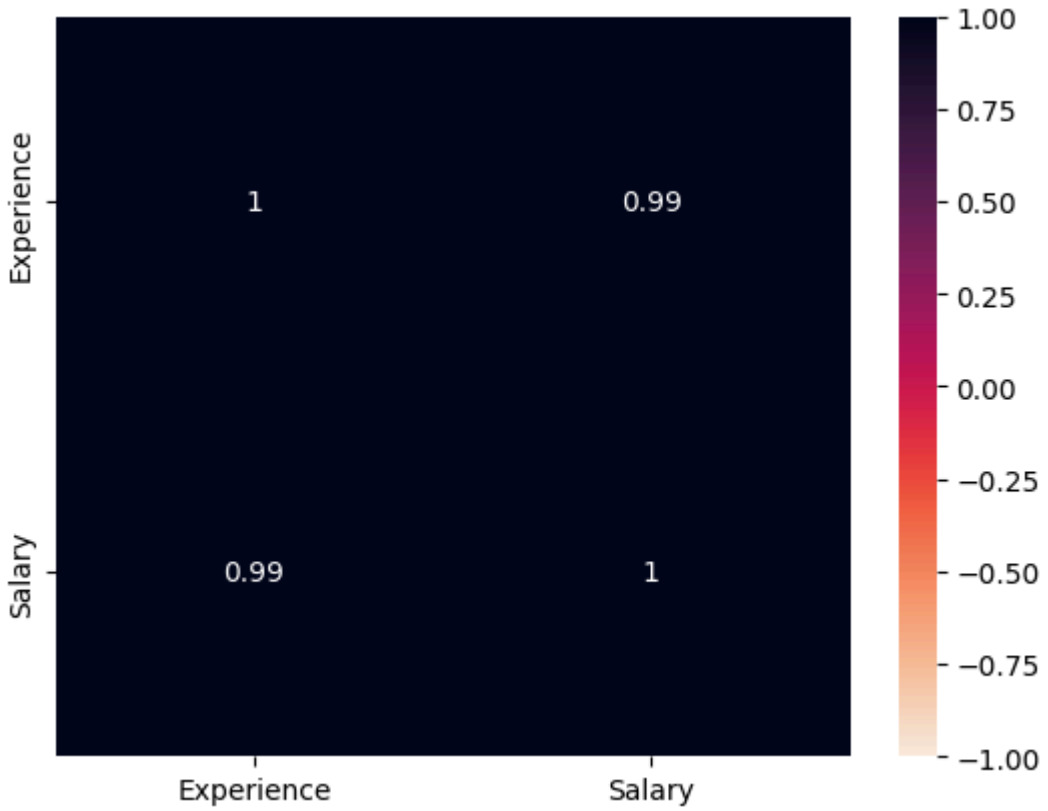
```
0.9992644353546791
0.9958246164193105
```

In [70]:
```
df.corr()
```

Out[70]:

	Experience	Salary
Experience	1.000000	0.992985
Salary	0.992985	1.000000

In [71]:
```
import seaborn as sns

sns.heatmap(df.corr(), vmin=-1, vmax=1,
annot=True,cmap="rocket_r")
plt.show()
```



Conclusión

La correlación solo cuantifica la fuerza y la dirección de la relación entre dos variables. Puede haber una fuerte correlación entre dos variables, pero no nos permite concluir que una causa la otra. Cuando las correlaciones fuertes no son causales, las llamamos correlaciones espurias.

Relación entre correlación y grado de significancia

Una correlación se considera significativa cuando su p-valor es menor que 0.05, lo que indica que la correlación observada es poco probable que se deba al azar. En otras palabras, si el p-valor es menor que 0.05, se rechaza la hipótesis nula de que no hay correlación entre las variables, y se concluye que existe una correlación significativa.

Desarrollemos este tema:

- P-valor (p-value): es una medida de la probabilidad de obtener los resultados observados (o resultados aún más extremos) si la hipótesis nula fuera verdadera. En

el contexto de la correlación, la hipótesis nula es que no hay correlación entre las variables.

- Nivel de significancia: generalmente 0.05, es un umbral predefinido para determinar si un resultado es significativo. Si el p-valor es menor o igual que el nivel de significancia, se considera que la correlación es significativa.
- Hipótesis nula: establece que no hay relación entre las variables. Si la hipótesis nula se rechaza, se concluye que hay una relación significativa entre las variables.
- Hipótesis alternativa: establece que sí hay una relación entre las variables. Si la hipótesis nula se rechaza, la hipótesis alternativa se considera válida, lo que indica que hay una correlación significativa entre las variables.

En resumen, para determinar si una correlación es significativa, se compara el p-valor con el nivel de significancia predefinido (generalmente 0.05). Si el p-valor es menor o igual que el nivel de significancia, se concluye que la correlación es significativa.

Interpretar y evaluar críticamente la información presentada en varios tipos de gráficos, incluidos diagramas de caja, histogramas, diagramas de dispersión, diagramas de líneas y mapas de calor de correlación.

Documentación de matplotlib: https://matplotlib.org/stable/users/index.html

Pagina general: https://matplotlib.org/

In [72]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits import mplot3d # librería para gráficos en 3D
```

Un posible dataframe (df)

In [73]:
```
df = pd.DataFrame({"x": [1,32,4,23,40,2,2,27,6,18,49,67,46,7,
                         20,24,35,33,40,80,26,85,77,11,92,24],
                   "y": [31,10,85,25,4,83,32,43,66,18,93,6,42,
                         27,21,42,53,32,85,32,42,58,67,17,4,5]})
```

In [74]:
```
df
```

Out[74]:

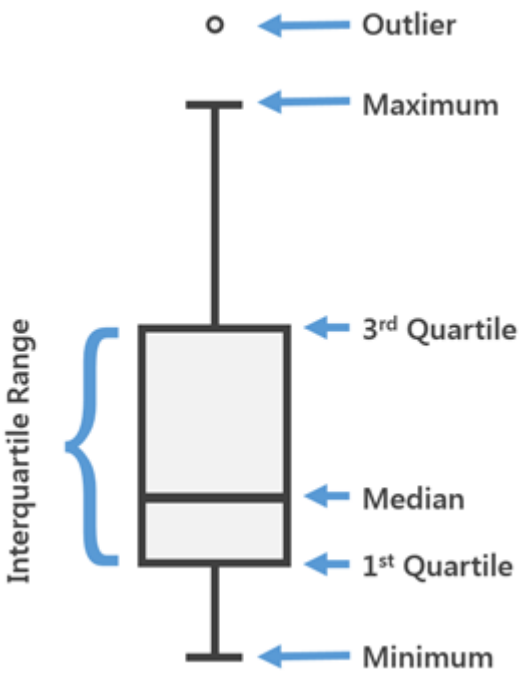
	x	y
0	1	31
1	32	10
2	4	85
3	23	25
4	40	4
5	2	83
6	2	32
7	27	43
8	6	66
9	18	18
10	49	93
11	67	6
12	46	42
13	7	27
14	20	21
15	24	42
16	35	53
17	33	32
18	40	85
19	80	32
20	26	42
21	85	58
22	77	67
23	11	17
24	92	4
25	24	5

Diagrama de cajas (box plot)

Un diagrama de caja (también, diagrama de caja y bigotes o box plot) es un método estandarizado para representar gráficamente una serie de datos numéricos a través de sus cuartiles. De esta manera, se muestran a simple vista la mediana y los cuartiles de los datos, y también pueden representarse sus valores atípicos. Nos muestra variables Discretas y Continuas.

In [17]:
```
display.Image("./images/boxplot.png")
```

Out[17]:

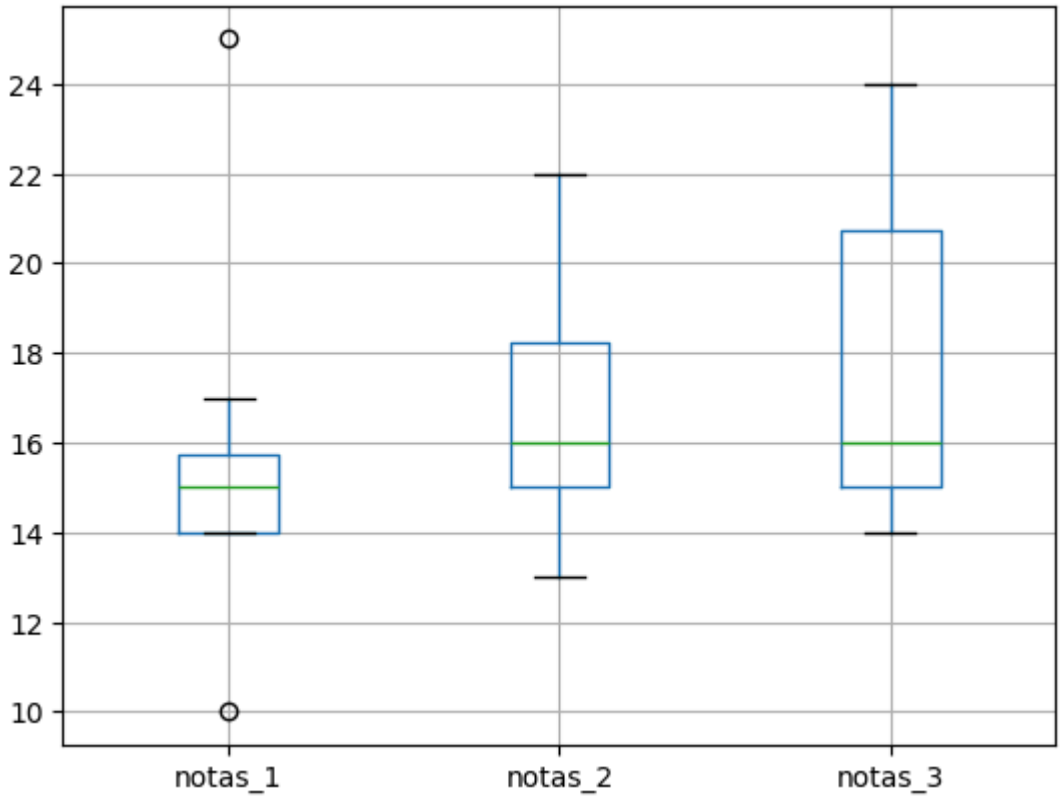


In [76]:
```
df = pd.DataFrame({"notas_1": [15, 16, 15, 17, 14, 14, 14, 10, 15, 25],
                   "notas_2": [16, 21, 16, 16, 13, 15, 15, 19, 22, 15],
                   "notas_3": [17, 22, 15, 22, 14, 15, 16, 15, 24, 16]})
df.head()
```

Out[76]:

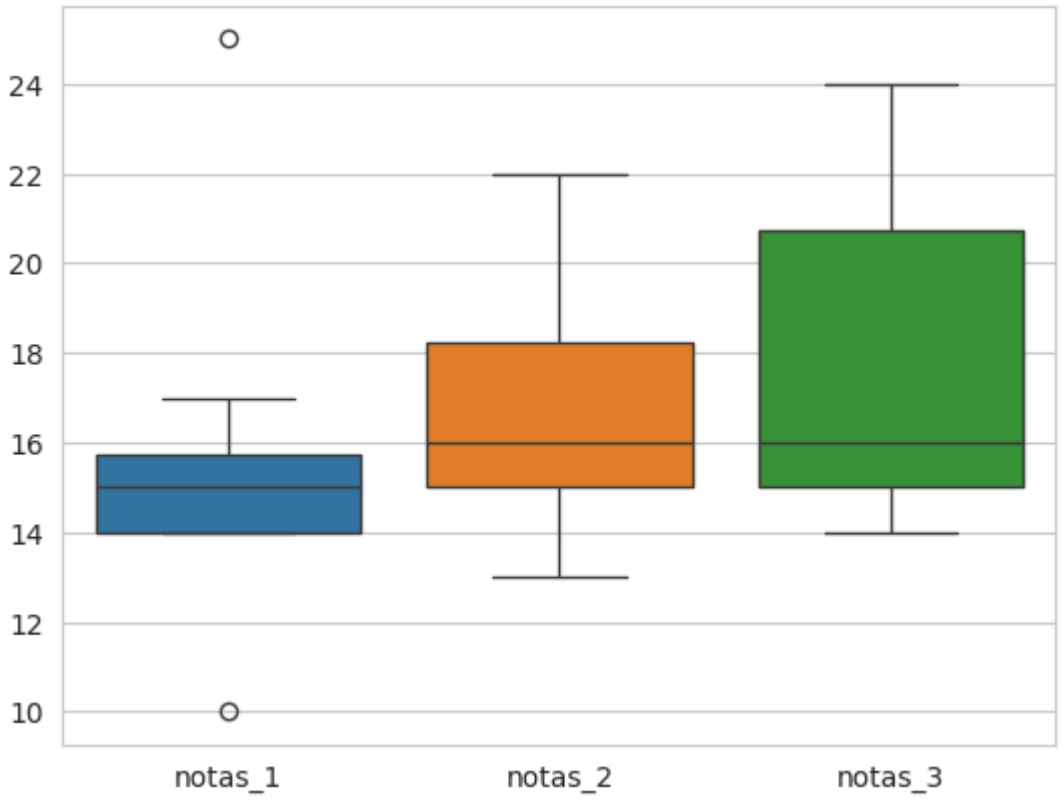
	notas_1	notas_2	notas_3
0	15	16	17
1	16	21	22
2	15	16	15
3	17	16	22
4	14	13	14

In [77]:
```
# usando pandas:
boxplot = df.boxplot(column=["notas_1", "notas_2", "notas_3"])
```



In [78]:
```
# seaborn:
sns.set_style("whitegrid")
sns.boxplot(data=df)
```

Out[78]: <Axes: >



Histograma

Un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados. Sirven para obtener una "primera vista" general, o panorama, de la distribución de la población, o de la muestra, respecto a una característica, cuantitativa y continua (como la longitud o el peso). De esta manera ofrece una visión de grupo permitiendo observar una preferencia, o tendencia, por parte de la muestra o población por ubicarse hacia una determinada región de valores dentro del espectro de valores posibles (sean infinitos o no) que pueda adquirir la característica. Así pues, podemos evidenciar comportamientos, observar el grado de homogeneidad, acuerdo o concisión entre los valores de todas las partes que componen la población o la muestra, o, en contraposición, poder observar el grado de variabilidad, y por ende, la dispersión de todos los valores que toman las partes, también es posible no evidenciar ninguna tendencia y obtener que cada miembro de la población toma por su lado y adquiere un valor de la característica aleatoriamente sin mostrar ninguna preferencia o tendencia.

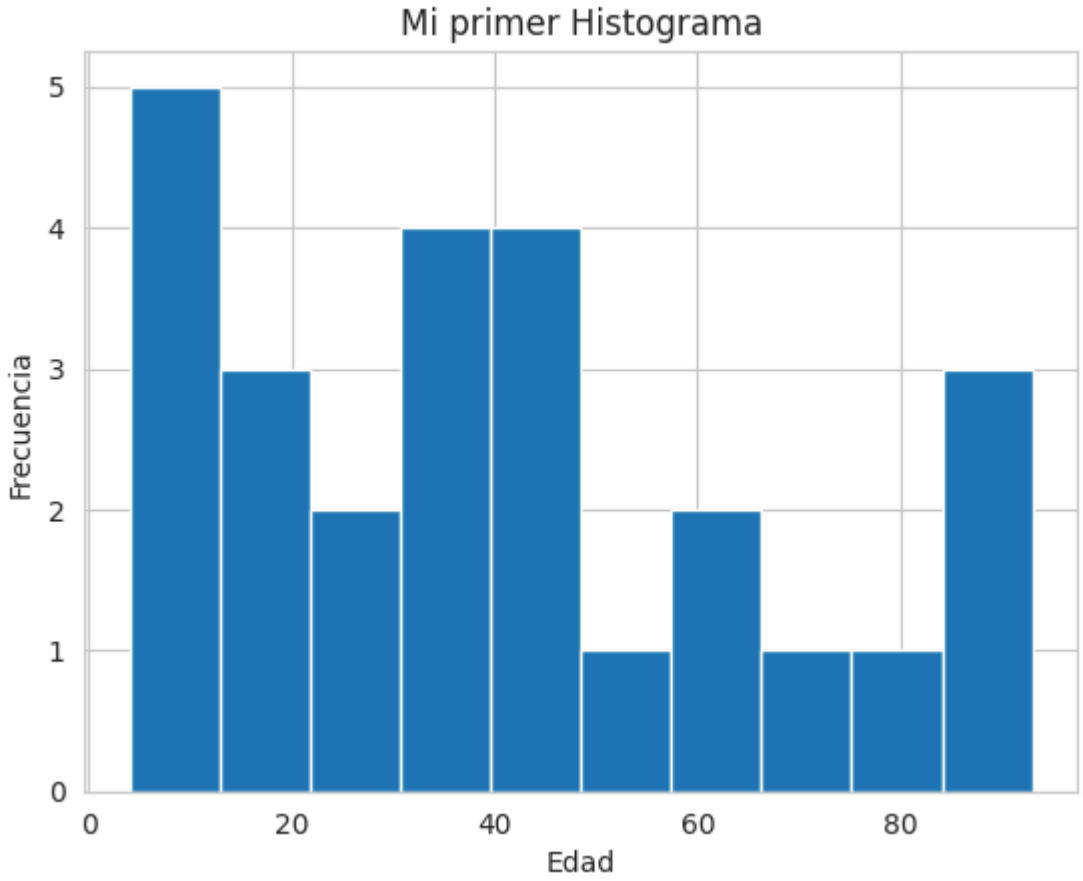
En el eje vertical se representan las frecuencias, es decir, la cantidad de población o la muestra, según sea el caso, que se ubica en un determinado valor o subrango de valores de la característica que toma la característica de interés. Evidentemente, cuando este espectro de valores es infinito o muy grande, se reduce a solo una parte que muestre la tendencia o comportamiento de la población. En otras ocasiones, este espectro es extendido para mostrar el alejamiento o ubicación de la población o la muestra analizada respecto de un valor de interés. Se utilizan para relacionar **variables cuantitativas continuas**. Para variables **cuantitativas discretas** las barras se dibujan separadas y el gráfico se llama diagrama de frecuencias, porque la variable representada en el eje horizontal ya no representa un espectro continuo de valores, sino valores cuantitativos específicos, igual que ocurre en un diagrama de barras, usado para representar una característica cualitativa o categórica. Su utilidad se hace más evidente cuando se cuenta con un gran número de datos cuantitativos y que se han agrupado en intervalos de clase.

Casi sin parametrizar

In [79]:
```
df = pd.DataFrame({"x": [1,32,4,23,40,2,2,27,6,18,49,67,46,7,
                         20,24,35,33,40,80,26,85,77,11,92,24],
                   "y": [31,10,85,25,4,83,32,43,66,18,93,6,42,
                         27,21,42,53,32,85,32,42,58,67,17,4,5]})
```

In [80]:
```
# y: [31,10,85,25,4,83,32,43,66,18,93,6,42,27,21,42,53,32,85,32,42,58,67,
# histograma
plt.hist(df.y)
# cuadrícula
plt.grid(True)
# Etiqueta eje X
plt.xlabel("Edad")
# Etiqueta eje Y
plt.ylabel("Frecuencia")
# titulo al gráfico
plt.title("Mi primer Histograma")

# mostrar el gráfico
```

```
plt.show()

# NOTA: se refiere a la frecuencia absoluta, repeticiones
```

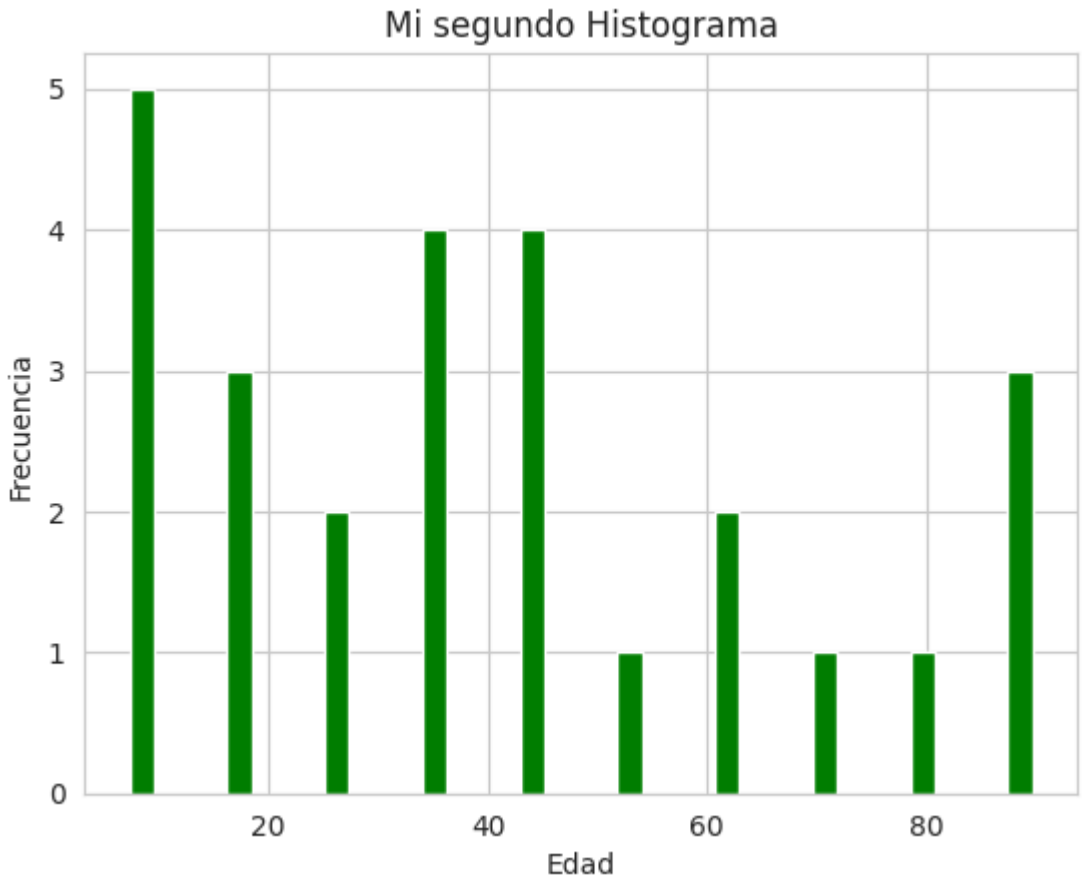


Histograma 2

Con estas características:

- 10 divisiones
- 0.25 de ancho de la barra

In [81]:
```
# realizar un histagrama:
plt.hist(df.y, bins=10,
        color="green",
        histtype="bar",
        rwidth= 0.25)
# cuadrícula
plt.grid(True)
# Etiqueta del eje de la X
plt.xlabel("Edad")
# Etiqueta del eje de la Y
plt.ylabel("Frecuencia")
# titulo:
plt.title("Mi segundo Histograma")

# Mostrar el gráfico:
plt.show()
```

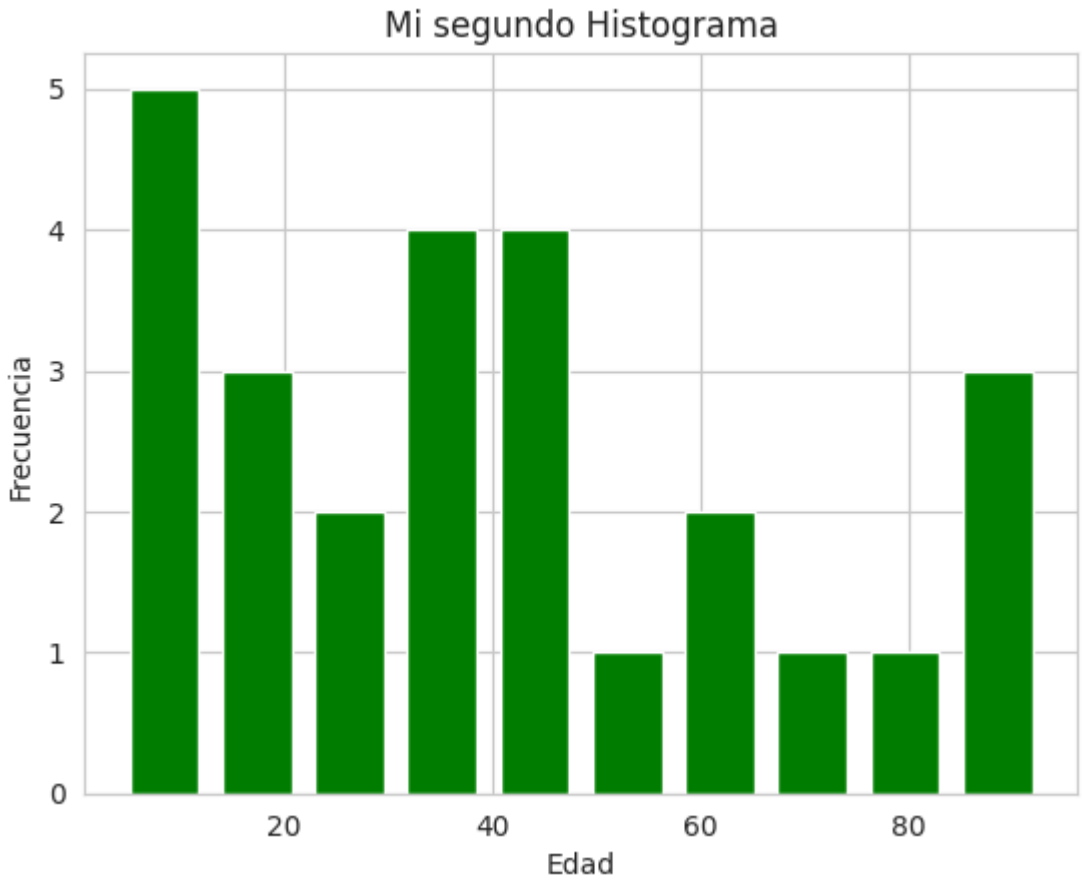


Histograma 3

Con estas características:

- 10 divisiones
- 0.75 de ancho de la barra

In [82]:
```
# realizar un histagrama:
plt.hist(df.y, bins=10,
        color="green",
        histtype="bar",
        rwidth= 0.75) # modificando solo la anchura de la barra
# cuadrícula
plt.grid(True)
# Etiqueta del eje de la X
plt.xlabel("Edad")
# Etiqueta del eje de la Y
plt.ylabel("Frecuencia")
# titulo:
plt.title("Mi segundo Histograma")

# Mostrar el gráfico:
plt.show()
```

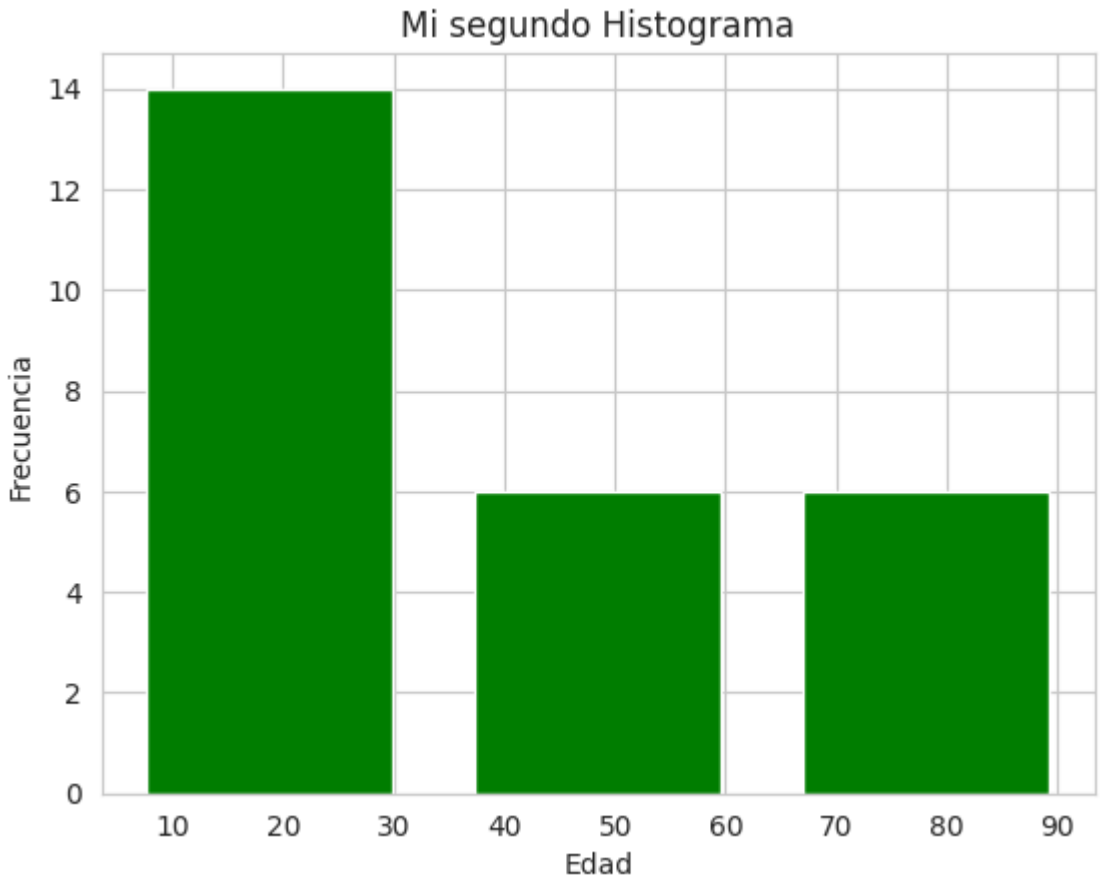


Histograma 4

Con estas características:

- 3 divisiones
- 0.75 de ancho de la barra

In [83]:
```
# realizar un histagrama:
plt.hist(df.y, bins=3, # modificando solo el número de barras
        color="green",
        histtype="bar",
        rwidth= 0.75)
# cuadrícula
plt.grid(True)
# Etiqueta del eje de la X
plt.xlabel("Edad")
# Etiqueta del eje de la Y
plt.ylabel("Frecuencia")
# titulo:
plt.title("Mi segundo Histograma")

# Mostrar el gráfico:
plt.show()
```



Scatter plot (nube de puntos)

Una nube de puntos es un conjunto de vértices en un sistema de coordenadas tridimensional. Estos vértices se identifican habitualmente como coordenadas X, Y, y Z y son representaciones de la superficie externa de un objeto.

Las nubes de puntos se crean habitualmente con un láser escáner tridimensional. Este instrumento mide de forma automática un gran número de puntos en la superficie de un objeto, y generan un fichero de datos con una nube de puntos. La nube de puntos representa el conjunto de puntos que ha medido el dispositivo.

Las nubes de puntos tienen múltiples aplicaciones, entre las que se incluyen la elaboración de modelos tridimensionales en CAD de piezas fabricadas, la inspección de calidad en metrología, y muchas otras en el ámbito de la visualización, animación, texturización y aplicaciones de personalización masiva.

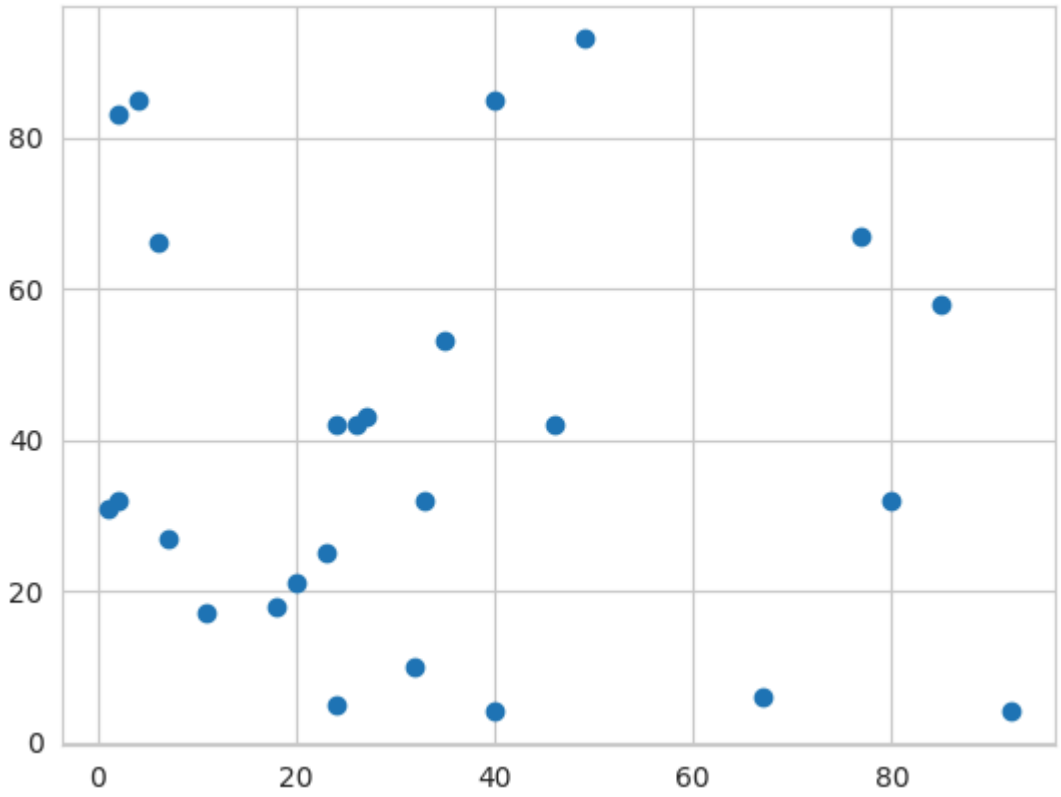
In [84]:
```
df
```

Out[84]:

	x	y
0	1	31
1	32	10
2	4	85
3	23	25
4	40	4
5	2	83
6	2	32
7	27	43
8	6	66
9	18	18
10	49	93
11	67	6
12	46	42
13	7	27
14	20	21
15	24	42
16	35	53
17	33	32
18	40	85
19	80	32
20	26	42
21	85	58
22	77	67
23	11	17
24	92	4
25	24	5

Scatter plot 1

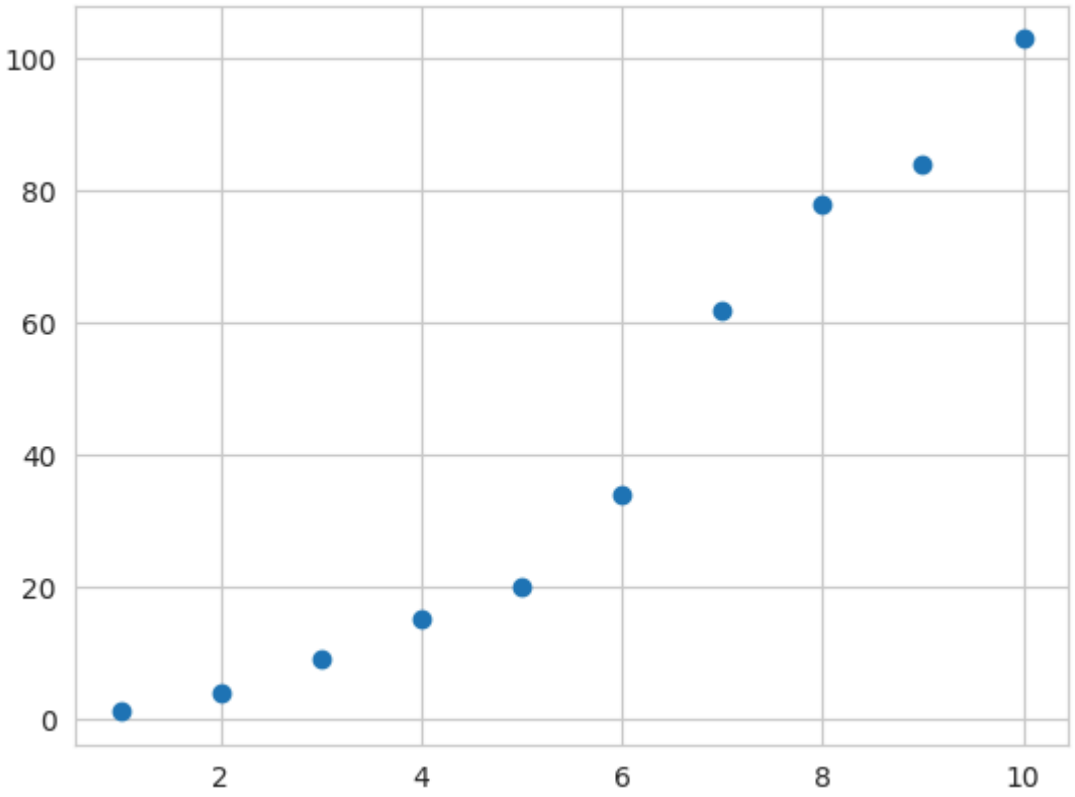
In [85]:
```
plt.scatter(df.x, df.y)
plt.show()
```



Scatter plot 2

In [86]:
```
# Con otros datos
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [1, 4, 9, 15, 20, 34, 62, 78, 84, 103]
```

In [87]:
```
plt.scatter(x, y)
plt.grid(True)
plt.show()
```

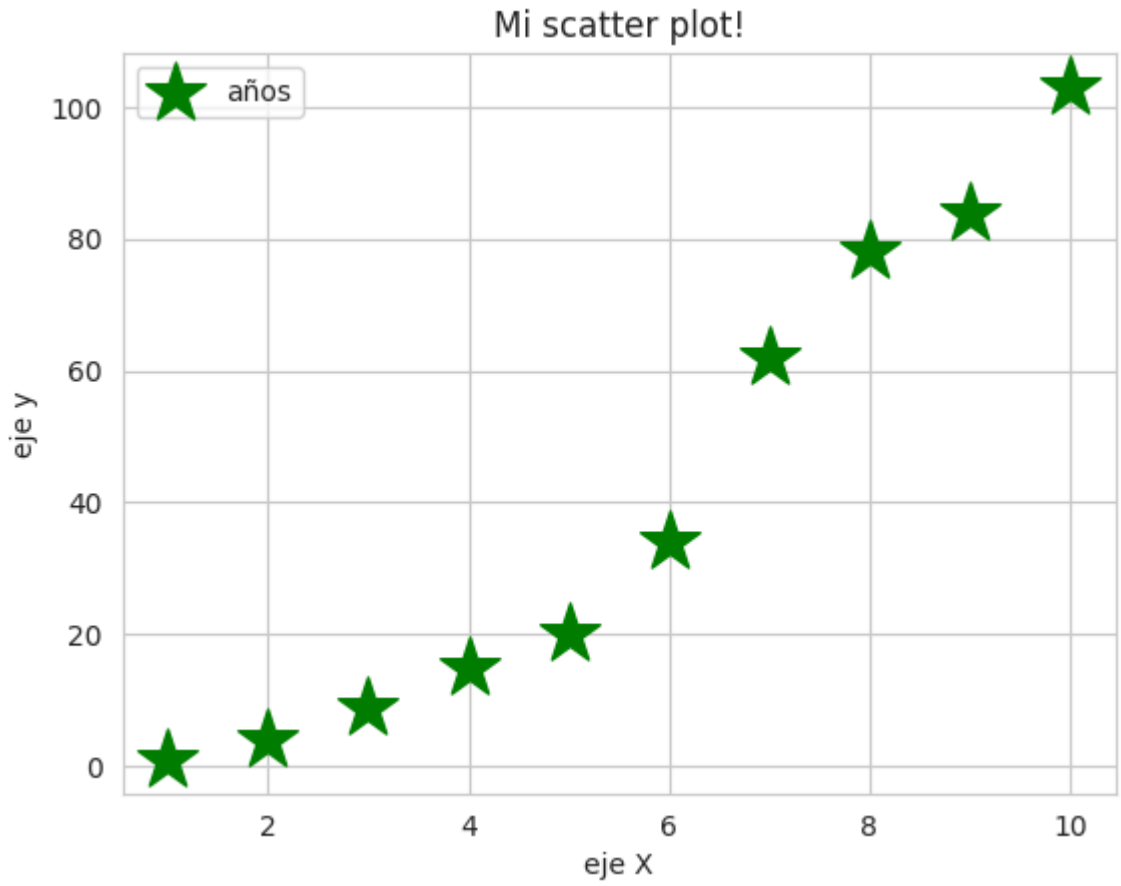


Scatter plot 3

Con las siguientes características:

- label = "años"
- color = verde
- simbolo *
- s = 500 es bastante grande

In [88]:
```
plt.scatter(x, y,
            label="años", color="green",
            marker="*", s=500)
plt.xlabel("eje X")
plt.ylabel("eje y")
plt.title("Mi scatter plot!")
plt.legend()
plt.show()
```

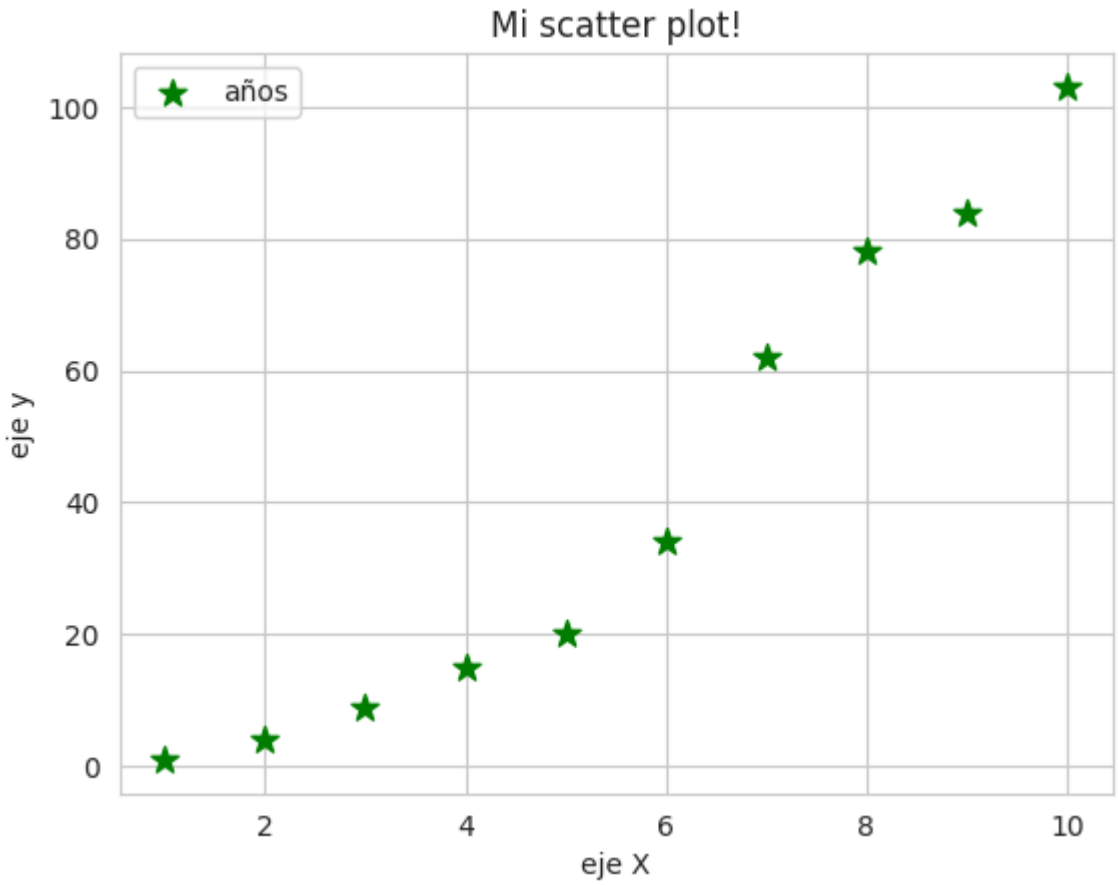


Scatter plot 4

Con las siguientes características:

- s = 100 es bastante grande

In [89]:
```
plt.scatter(x, y,
            label="años", color="green",
            marker="*", s=100) # modificamos el tamaño del punto
plt.xlabel("eje X")
plt.ylabel("eje y")
plt.title("Mi scatter plot!")
plt.legend()
plt.show()
```



Lineplots

Los gráficos lineales muestran cómo una variable continua cambia con el tiempo. La variable que mide el tiempo se representa en el eje X. La variable continua se representa en el eje Y.

In [90]:
```
import pandas as pd

df = pd.DataFrame({"Year" : [2014,2015,2016,2017,2018],
                   "Sales" : [2000, 3000, 4000, 3500, 6000]})

df
```

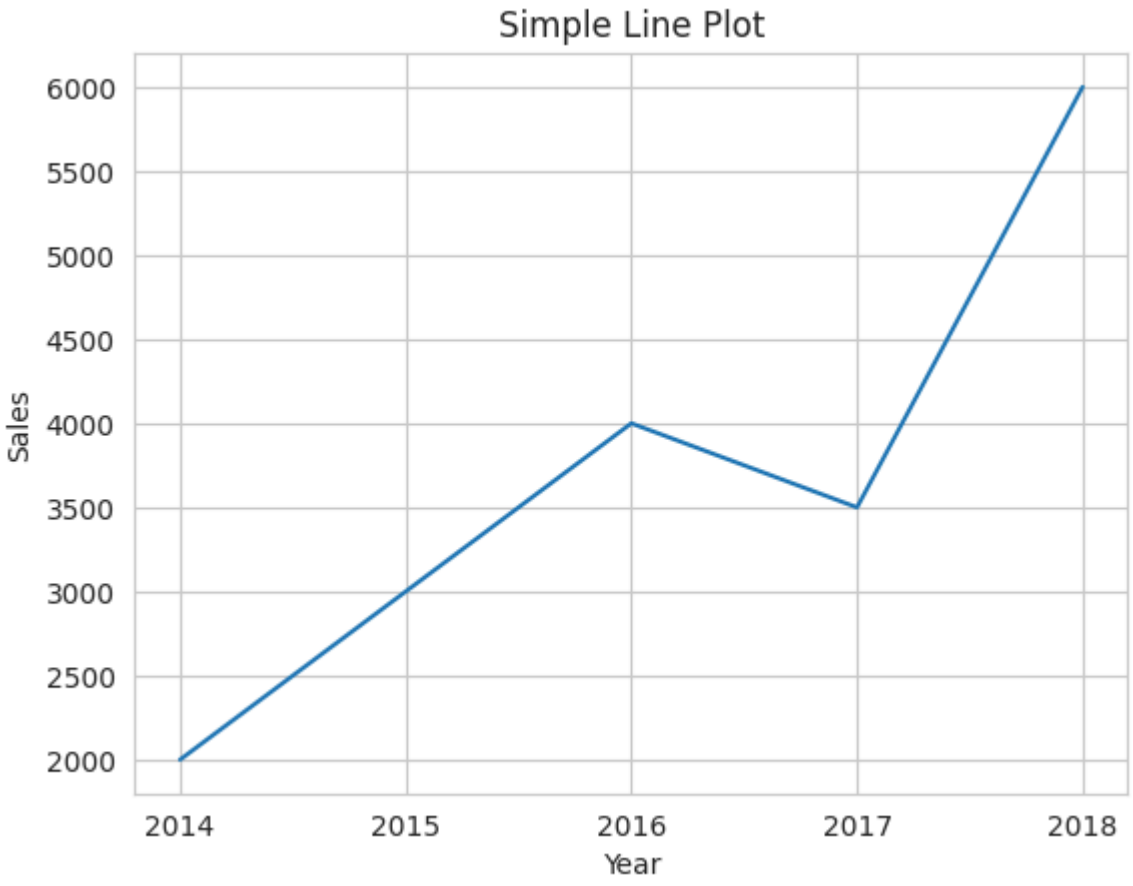
Out[90]:

	Year	Sales
0	2014	2000
1	2015	3000
2	2016	4000
3	2017	3500
4	2018	6000

In [91]:
```
# plt.plot( ) function is used for line graph. It is the default graph ty
```

In [92]:
```
# x es df.Year
# y es df.Sales
plt.plot(df["Year"], df["Sales"])
```

```
plt.title("Simple Line Plot")
plt.xlabel('Year')
plt.ylabel('Sales')

plt.style.use('fivethirtyeight')

plt.show()
```



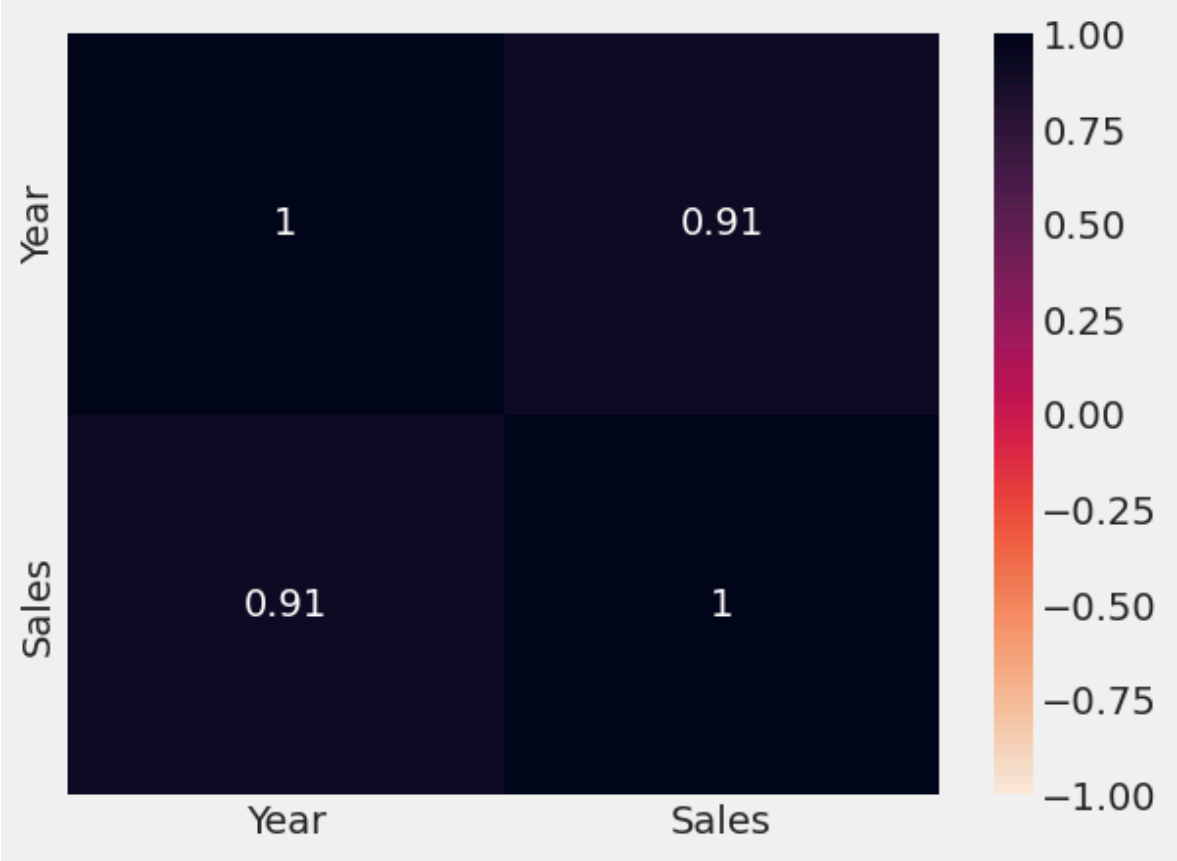
Correlation Heatmaps

La correlación de datos es una forma de entender la relación entre múltiples valores o características en su conjunto de datos.

- Cada uno de estos tipos de correlación existe en un espectro representado por valores de -1 a +1 donde las características de correlación positiva leve o alta pueden ser como 0,5 o 0,7.
- Una correlación positiva muy fuerte y perfecta está representada por una puntuación de correlación de 0,9 o 1.
- Si hay una fuerte correlación negativa, se representará con un valor de -0,9 o -1. Los valores cercanos a cero indican que no hay correlación.

In [93]:
```
import seaborn as sns

sns.heatmap(df.corr(), vmin=-1, vmax=1,
```

```
annot=True,cmap="rocket_r")
plt.show()
```



Creado por:

Isabel Maniega