

Creado por:

Isabel Maniega

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

In [2]: # pip install statsmodels

In [3]: 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
```

Ejemplo 1: Notas

```
In [4]: 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[4]:
```

	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

Como calcular la media de las distintas notas:

```
In [5]: media_1 = df["notas_1"].mean()
media_1

Out[5]: 15.5

In [6]: media_2 = df["notas_2"].mean()
media_2

Out[6]: 16.8

In [7]: media_3 = df["notas_3"].mean()
media_3

Out[7]: 17.6
```

Mediana

Como calcular la mediana de las distintas notas:

```
In [8]: mediana_1 = df["notas_1"].median()
mediana_1

Out[8]: 15.0

In [9]: mediana_2 = df["notas_2"].median()
mediana_2

Out[9]: 16.0

In [10]: mediana_3 = df["notas_3"].median()
mediana_3

Out[10]: 16.0
```

Moda

Como calcular la moda de las distintas notas:

```
In [11]: moda_1 = df["notas_1"].mode()
moda_1

Out[11]:
```

0	14
1	15
dtype:	int64

```
In [12]: moda_2 = df["notas_2"].mode()
moda_2

Out[12]:
```

0	15
1	16
dtype:	int64

```
In [13]: moda_3 = df["notas_3"].mode()
moda_3

Out[13]:
```

0	15
dtype:	int64

Resultados Nota 1:

```
In [14]: print(f"Media: {media_1}, Mediana: {mediana_1}, Moda: \n{moda_1}")

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

Resultados Nota 2:

```
In [15]: print(f"Media: {media_2}, Mediana: {mediana_2}, Moda: \n{moda_2}")

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

Resultados Nota 3:

```
In [16]: print(f"Media: {media_3}, Mediana: {mediana_3}, Moda: \n{moda_3}")

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

RESUMEN

Notas 1 Notas 2 Notas 3Media 15.5 16.8 17.6Mediana 15.0 16.0 16.0Moda 14/15 15/16 15/0

Histograma y Curva Normal

Notas 1

```
In [17]: notas_1 = df["notas_1"]
notas_1

Out[17]:
```

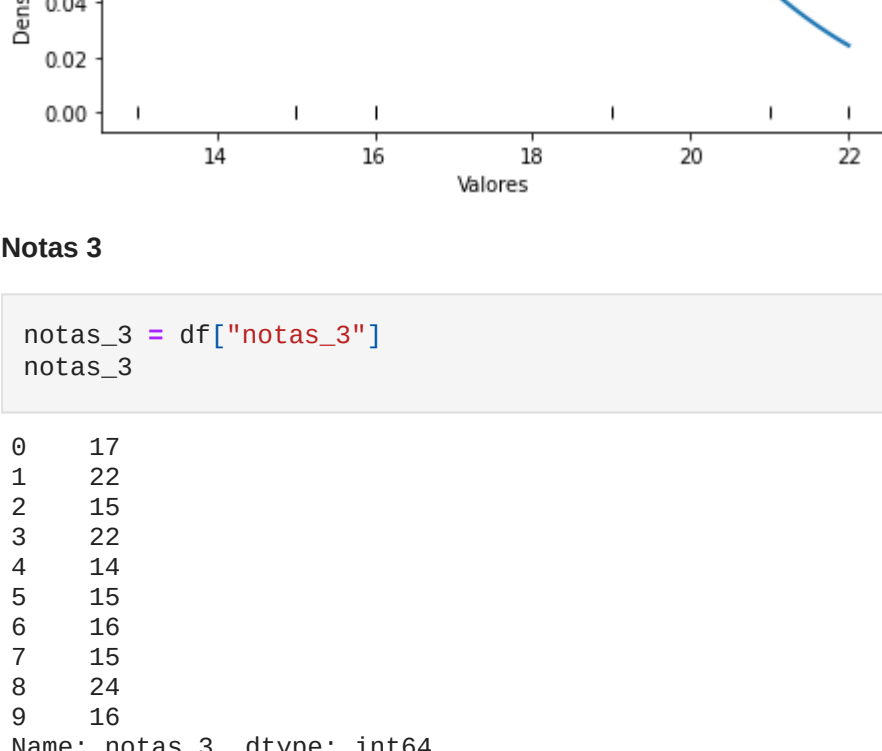
0	15
1	16
2	15
3	17
4	14
5	14
6	14
7	10
8	15
9	25
Name:	notas_1, dtype: int64

```
In [22]: # Histograma + curva normal teórica
# =====

# valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(notas_1)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(notas_1), max(notas_1), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7, 4))
ax.plot(x_hat, y_hat, linewidth=2, label="normal")
# ax.hist(x=notas_1, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(notas_1, np.full_like(notas_1, -0.01), "k", markeredgewidth=1)
ax.set_title("Distribución Notas 1")
ax.set_xlabel("Valores")
ax.set_ylabel("Densidad de probabilidad")
ax.legend()
plt.show()
```



Notas 2

```
In [23]: notas_2 = df["notas_2"]
notas_2

Out[23]:
```

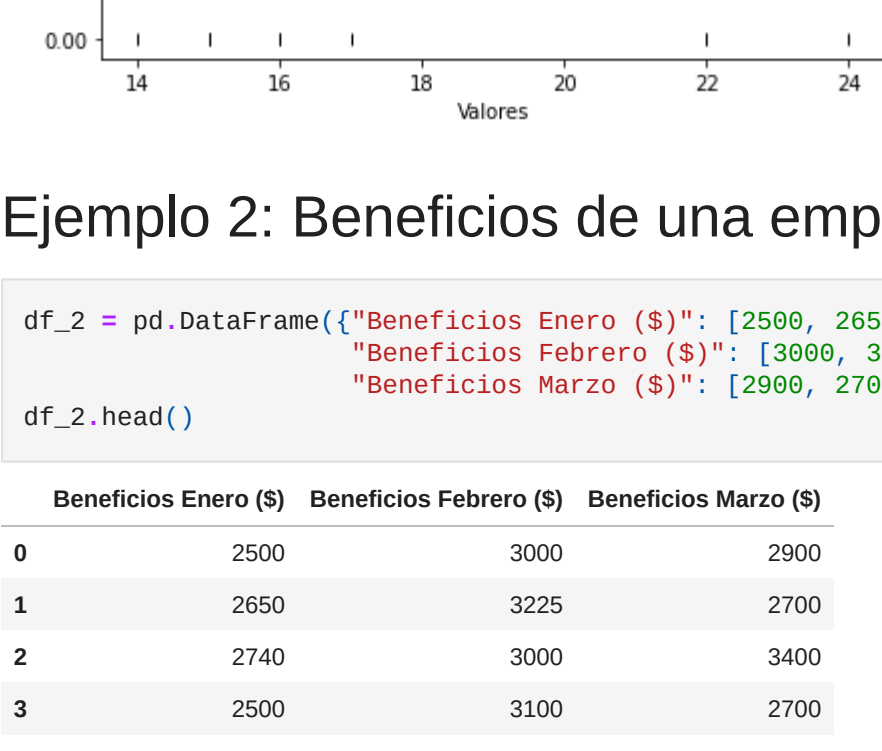
0	16
1	21
2	16
3	16
4	13
5	15
6	15
7	19
8	22
9	15
Name:	notas_2, dtype: int64

```
In [29]: # Histograma + curva normal teórica
# =====

# valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(notas_2)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(notas_2), max(notas_2), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7, 4))
ax.plot(x_hat, y_hat, linewidth=2, label="normal")
# ax.hist(x=notas_2, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(notas_2, np.full_like(notas_2, -0.01), "k", markeredgewidth=1)
ax.set_title("Distribución Notas 2")
ax.set_xlabel("Valores")
ax.set_ylabel("Densidad de probabilidad")
ax.legend()
plt.show()
```



Notas 3

```
In [30]: notas_3 = df["notas_3"]
notas_3

Out[30]:
```

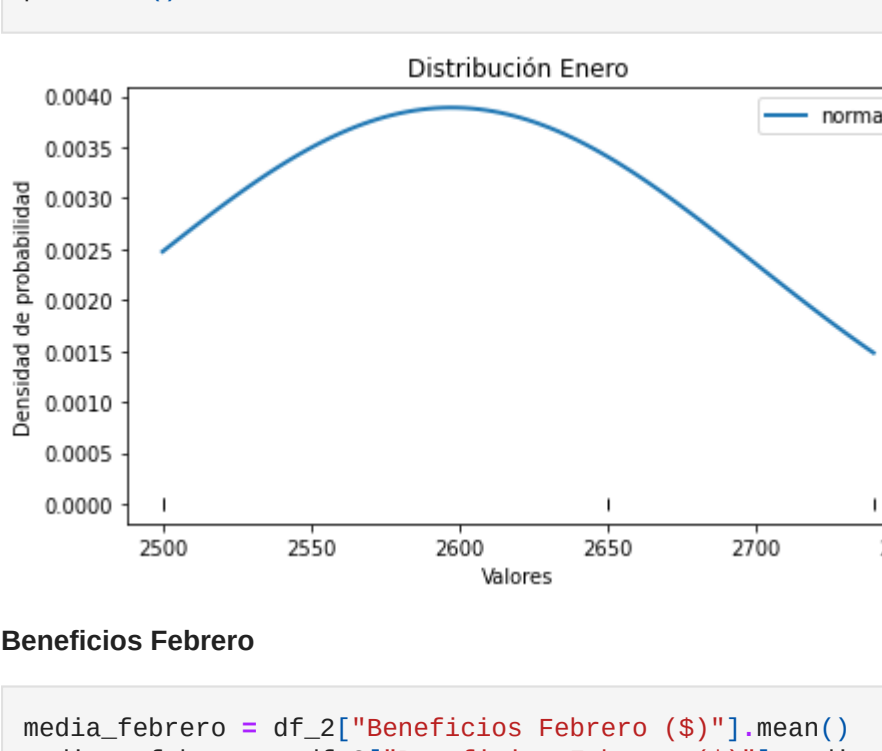
0	17
1	22
2	15
3	22
4	14
5	14
6	16
7	15
8	24
9	16
Name:	notas_3, dtype: int64

```
In [31]: # Histograma + curva normal teórica
# =====

# valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(notas_3)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(notas_3), max(notas_3), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7, 4))
ax.plot(x_hat, y_hat, linewidth=2, label="normal")
# ax.hist(x=notas_3, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(notas_3, np.full_like(notas_3, -0.01), "k", markeredgewidth=1)
ax.set_title("Distribución Notas 3")
ax.set_xlabel("Valores")
ax.set_ylabel("Densidad de probabilidad")
ax.legend()
plt.show()
```



Ejemplo 2: Beneficios de una empresa

```
In [34]: df_2 = pd.DataFrame({"Beneficios Enero ($)": [2500, 2650, 2740, 2500],
                           "Beneficios Febrero ($)": [3000, 3225, 3000, 3100],
                           "Beneficios Marzo ($)": [2900, 2700, 3400, 2700]})
df_2.head()

Out[34]:
```

	Beneficios Enero (\$)	Beneficios Febrero (\$)	Beneficios Marzo (\$)
0	2500	3000	2900
1	2650	3225	2700
2	2740	3000	3400
3	2500	3100	2700

Beneficios Enero

```
In [35]: media_enero = df_2["Beneficios Enero ($)"].mean()
mediana_enero = df_2["Beneficios Enero ($)"].median()
moda_enero = df_2["Beneficios Enero ($)"].mode()
print(f"Media: {media_enero}, Mediana: {mediana_enero}, Moda: \n{moda_enero}")

Media: 2597.5, Mediana: 2575.0, Moda:
0    2500
dtype: int64
```

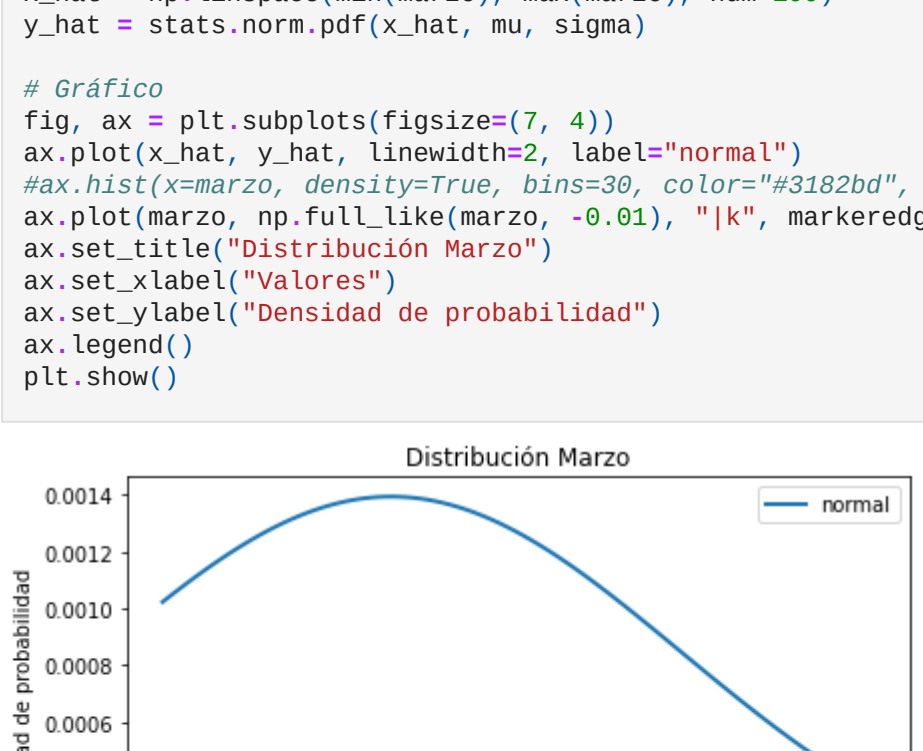
```
In [39]: enero = df_2["Beneficios Enero ($)"]
```

```
In [40]: # Histograma + curva normal teórica
# =====

# valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(enero)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(enero), max(enero), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7, 4))
ax.plot(x_hat, y_hat, linewidth=2, label="normal")
# ax.hist(x=enero, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(enero, np.full_like(enero, -0.01), "k", markeredgewidth=1)
ax.set_title("Distribución Enero")
ax.set_xlabel("Valores")
ax.set_ylabel("Densidad de probabilidad")
ax.legend()
plt.show()
```



Beneficios Febrero

```
In [41]: media_febrero = df_2["Beneficios Febrero ($)"].mean()
mediana_febrero = df_2["Beneficios Febrero ($)"].median()
moda_febrero = df_2["Beneficios Febrero ($)"].mode()
print(f"Media: {media_febrero}, Mediana: {mediana_febrero}, Moda: \n{moda_febrero}")

Media: 3081.25, Mediana: 3050.0, Moda:
0    3000
dtype: int64
```

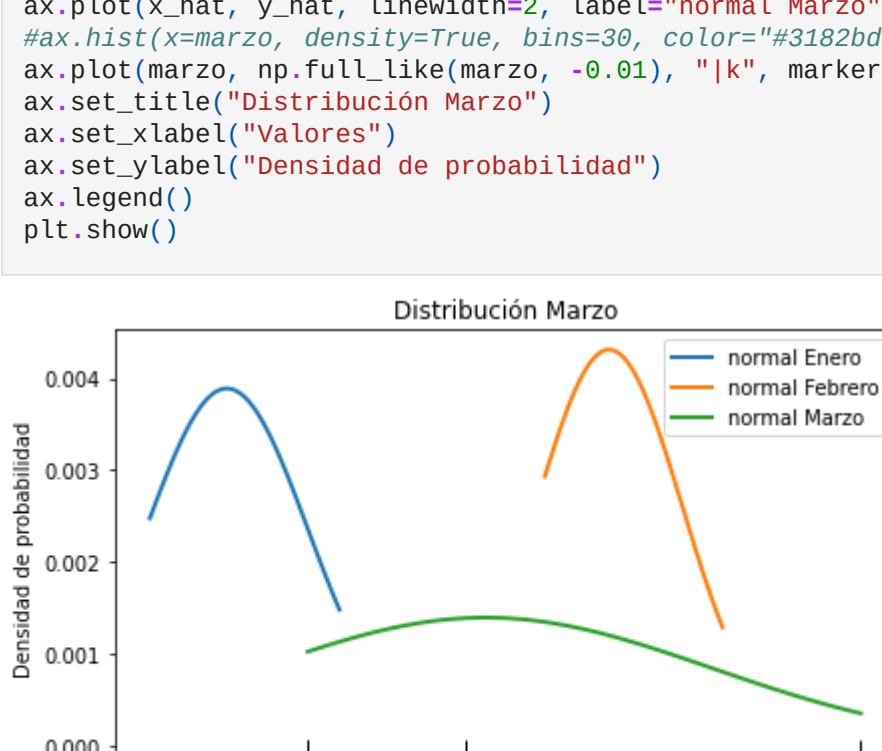
```
In [42]: febrero = df_2["Beneficios Febrero ($)"]
```

```
In [43]: # Histograma + curva normal teórica
# =====

# valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(febrero)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(febrero), max(febrero), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7, 4))
ax.plot(x_hat, y_hat, linewidth=2, label="normal")
# ax.hist(x=febrero, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(febrero, np.full_like(febrero, -0.01), "k", markeredgewidth=1)
ax.set_title("Distribución Febrero")
ax.set_xlabel("Valores")
ax.set_ylabel("Densidad de probabilidad")
ax.legend()
plt.show()
```



Beneficios Marzo

```
In [44]: media_marzo = df_2["Beneficios Marzo ($)"].mean()
mediana_marzo = df_2["Beneficios Marzo ($)"].median()
moda_marzo = df_2["Beneficios Marzo ($)"].mode()
print(f"Media: {media_marzo}, Mediana: {mediana_marzo}, Moda: \n{moda_marzo}")

Media: 2925.0, Mediana: 2800.0, Moda:
0    2700
dtype: int64
```

```
In [45]: marzo = df_2["Beneficios Marzo ($)"]
```

```
In [46]: # Histograma + curva normal teórica
# =====

# valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(marzo)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(marzo), max(marzo), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7, 4))
ax.plot(x_hat, y_hat, linewidth=2, label="normal")
# ax.hist(x=marzo, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(marzo, np.full_like(marzo, -0.01), "k", markeredgewidth=1)
ax.set_title("Distribución Marzo")
ax.set_xlabel("Valores")
ax.set_ylabel("Densidad de probabilidad")
ax.legend()
plt.show()
```


Gráfico conjunto

```
In [47]: # Histograma + curva normal teórica
# =====

# ENERO
# =====
# valores de la media (mu) y desviación típica (sigma) de los datos
mu_e, sigma_e = stats.norm.fit(enero)

# Valores teóricos de la normal en el rango observado
x_hat_e = np.linspace(min(enero), max(enero), num=100)
y_hat_e = stats.norm.pdf(x_hat_e, mu_e, sigma_e)

# FEBRERO
# =====
# valores de la media (mu) y desviación típica (sigma) de los datos
mu_f, sigma_f = stats.norm.fit(febrero)

# Valores teóricos de la normal en el rango observado
x_hat_f = np.linspace(min(febrero), max(febrero), num=100)
y_hat_f = stats.norm.pdf(x_hat_f, mu_f, sigma_f)

# MARZO
# =====
# valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(marzo)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(marzo), max(marzo), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7, 4))
ax.plot(x_hat_e, y_hat_e, linewidth=2, label="normal Enero")
ax.plot(x_hat_f, y_hat_f, linewidth=2, label="normal Febrero")
ax.plot(x_hat, y_hat, linewidth=2, label="normal Marzo")
# ax.hist(x=enero, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(marzo, np.full_like(marzo, -0.01), "k", markeredgewidth=1)
ax.set_title("Distribución Marzo")
ax.set_xlabel("Valores")
ax.set_ylabel("Densidad de probabilidad")
ax.legend()
plt.show()
```


RESUMEN

ENERO FEBRERO MARZOMedia 2597.5 3081.5 2925.0Mediana 2575.0 3050.0 2800.0Moda 2500.0 3000.0 2700.0

Creado por:

Isabel Maniega