«Talento Tech»

Data Analytics

con Python

Clase 10

Buenos Aires Ciudad

Clase N° 10 | Exploración de Datos

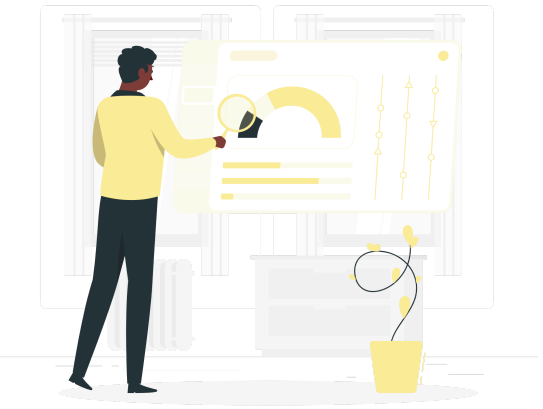
Temario:

- Técnicas de análisis exploratorio (EDA): visualización y resumen de datos.

Objetivos de la Clase:

- Comprender la importancia del análisis exploratorio de datos (EDA) en el proceso de análisis de datos.
- Aprender a utilizar librerías de Python para la visualización de datos.
- Combinar todos los conocimientos adquiridos para desarrollar habilidades en la creación de resúmenes estadísticos de conjuntos de datos.
- Aplicar técnicas de EDA a un conjunto de datos.

1. Exploración de Datos: Técnicas de Análisis Exploratorio (EDA)



La exploración de datos, o **EDA** (Exploratory Data Analysis en inglés), es un enfoque en el análisis de datos que permite a los analistas comprender con qué cuenta en cada conjunto de datos. Este proceso implica la examinación visual y resumida de nuestros datos para identificar patrones, anomalías y relaciones subyacentes.

Al comenzar un proyecto de análisis de datos, es importante realizar una **exploración exhaustiva del conjunto de datos**. Esto facilita la identificación de problemas, como los valores perdidos o las irregularidades en la distribución de los valores, y también ayuda a formular hipótesis sobre las relaciones entre diferentes variables.

En esta clase, veremos cómo se combinan dos técnicas clave para realizar un EDA: la **visualización de datos** y la **creación de resúmenes estadísticos**.

Introducción a la Visualización de Datos

La visualización de datos es una herramienta poderosa que permite **representar información de manera gráfica**. Utilizando librerías como Matplotlib y Seaborn en Python, podemos crear una variedad de gráficos que facilitan la comprensión de nuestros datos.

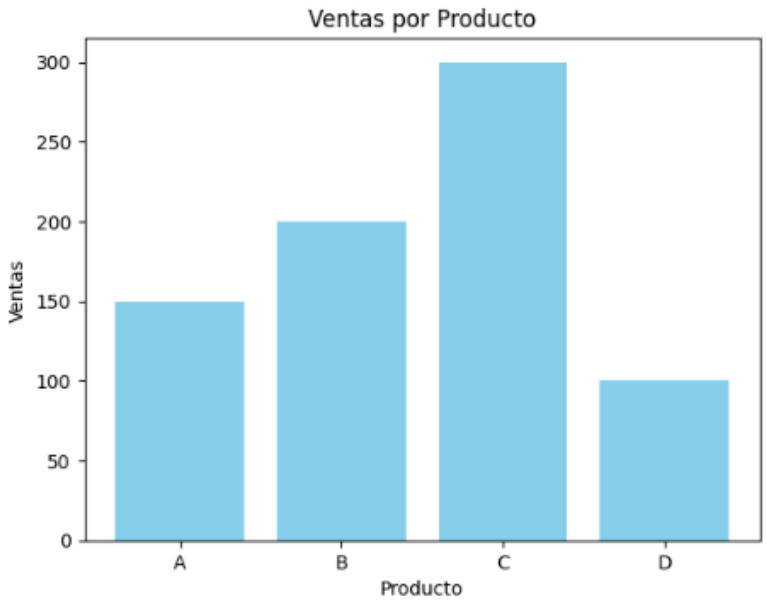
Ejemplo de Visualización con Matplotlib

Primero, presentaremos un ejemplo práctico. Supongamos que tenemos un conjunto de datos que contiene información sobre las ventas de diferentes productos. Cargaremos este conjunto de datos y crearemos un gráfico de barras para visualizar las ventas por producto.

```
import pandas as pd
import matplotlib.pyplot as plt

# Crear un DataFrame de ejemplo
data = {
    'Producto': ['A', 'B', 'C', 'D'],
    'Ventas': [150, 200, 300, 100]
}
df = pd.DataFrame(data)
```

```
# Crear un gráfico de barras
plt.bar(df['Producto'], df['Ventas'], color='skyblue')
plt.title('Ventas por Producto')
plt.xlabel('Producto')
plt.ylabel('Ventas')
plt.show()
```

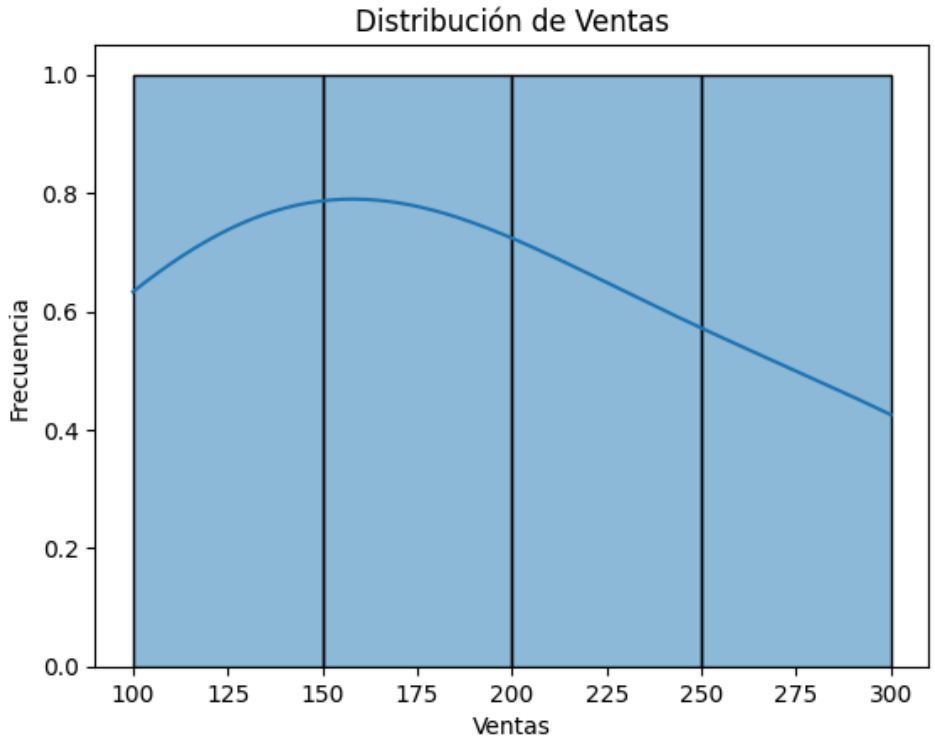


En este gráfico, cada barra representa las ventas de un producto específico. La visualización inmediata nos permite ver qué productos están vendiendo más y cuáles menos, facilitando así la toma de decisiones.

Ejemplo de Visualización con Seaborn

Seaborn, que se basa en **Matplotlib**, ofrece una interfaz más sencilla para crear gráficos estadísticos. Continuando con nuestro conjunto de datos anterior, podemos usar Seaborn para crear un gráfico de distribución.

```
import seaborn as sns
# Crear un gráfico de distribución de ventas
sns.histplot(df['Ventas'], bins=4, kde=True)
plt.title('Distribución de Ventas')
plt.xlabel('Ventas')
plt.ylabel('Frecuencia')
plt.show()
```



Aquí, el gráfico de distribución nos permite visualizar cómo se distribuyen las ventas. La línea de la densidad (kde) muestra la probabilidad de observación de las ventas en el rango representado.

Resumen de Datos

Además de la visualización, es importante **resumir nuestros datos de manera estadística**. Esto puede incluir la obtención de medidas como la media, mediana y desviación estándar, así como la identificación de valores atípicos.

Para obtener un resumen descriptivo de nuestros datos en Python, utilizamos la **función .describe() de Pandas**, que proporciona un análisis rápido y eficaz de las características de nuestros datos.

```
# Resumen estadístico del DataFrame
print(df.describe())
```

Este comando generará estadísticas clave, como la cuenta, media, desviación estándar, mínimo, máximo y los percentiles que ayudan a entender la distribución de nuestros datos.

```
            Ventas
count    4.000000
mean   187.500000
std     85.391256
min    100.000000
25%    137.500000
50%    175.000000
75%    225.000000
max    300.000000
```

2. Ejemplo de un EDA: Base de Datos "Tips" de Seaborn

En esta sección, realizaremos un análisis exploratorio utilizando la **base de datos "tips" de Seaborn**. Esta base de datos recopila información sobre las propinas que se dejan en un restaurante, incluyendo datos como el total de la cuenta, el porcentaje de propina, el día de la semana, y la cantidad de personas en la mesa.

Primero, cargaremos el conjunto de datos y realizaremos diferentes análisis y visualizaciones para extraer información significativa.

A. Cargar la Base de Datos

Comenzamos importando las librerías necesarias y cargando la base de datos "tips":

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

# Cargar la base de datos tips de Seaborn
tips = sns.load_dataset("tips")

# Visualizar las primeras filas del DataFrame
print(tips.head())
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Hallazgos Iniciales:

La **base de datos** contiene las siguientes columnas:

- **total_bill**: Monto total de la cuenta.
- **tip**: Monto de la propina.
- **sex**: Género del cliente.
- **smoker**: Indica si el cliente es fumador o no.
- **day**: Día de la semana.
- **time**: Hora de la comida (almuerzo o cena).
- **size**: Número de personas en la mesa.

B. Resumen Estadístico de los Datos

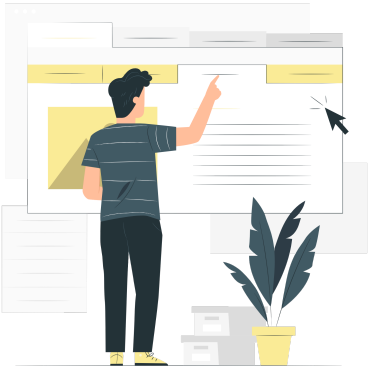
Generamos un resumen estadístico básico para comprender la distribución de los datos:

```
# Obtener un resumen estadístico
print(tips.describe())
```

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

Observaciones:

- **Total de la cuenta (total_bill)**: Varía entre 3.07 y 50.81, con un promedio de aproximadamente 19.79.
- **Propina (tip)**: Oscila entre 1.00 y 10.00, con un promedio de 2.99.
- **Número de personas en la mesa (size)**: Varía de 1 a 6, con una media de aproximadamente 2.57.



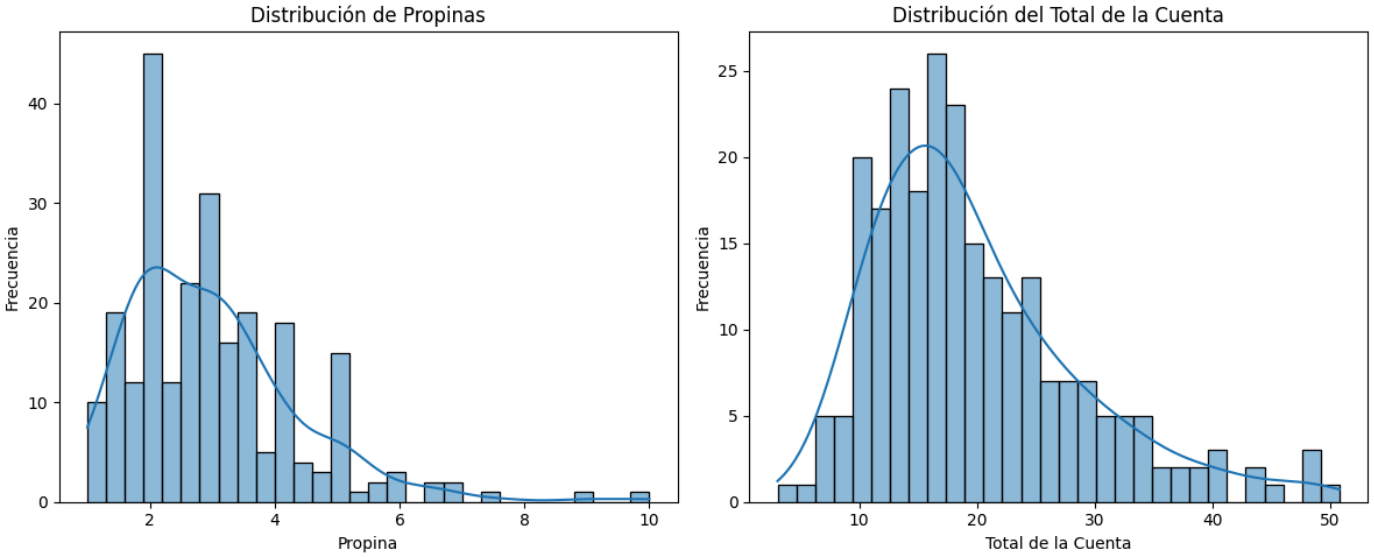
C. Análisis de Distribución

A continuación, analizamos la distribución de las propinas y el total de la cuenta utilizando histogramas:

```
# Visualizar la distribución de las propinas
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.histplot(tips['tip'], bins=30, kde=True)
plt.title('Distribución de Propinas')
plt.xlabel('Propina')
plt.ylabel('Frecuencia')

# Visualizar la distribución del total de la cuenta
plt.subplot(1, 2, 2)
sns.histplot(tips['total_bill'], bins=30, kde=True)
plt.title('Distribución del Total de la Cuenta')
plt.xlabel('Total de la Cuenta')
plt.ylabel('Frecuencia')

plt.tight_layout()
plt.show()
```



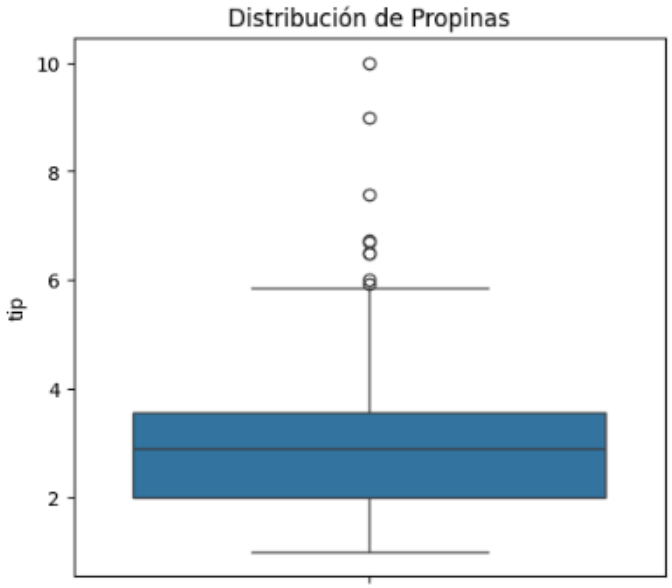
Veamos cómo examinar la distribución con BoxPlot:

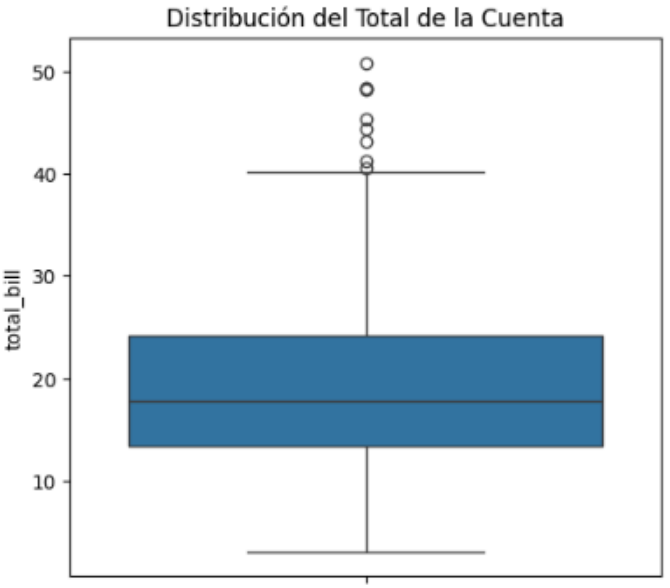
```
# Visualizar la distribución de las propinas

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.boxplot(tips['tip'])
plt.title('Distribución de Propinas')

# Visualizar la distribución del total de la cuenta
plt.subplot(1, 2, 2)
sns.boxplot(tips['total_bill'])
plt.title('Distribución del Total de la Cuenta')

plt.show()
```





Hallazgos:

- La distribución de las propinas muestra que la mayoría de las propinas son menores a 5, con un carácter sesgado hacia la izquierda.
- La distribución del total de la cuenta muestra una tendencia similar, con la mayoría de las cuentas por debajo de 25.

D. Análisis de la Relación entre Variables

Ahora, examinaremos la relación entre el total de la cuenta y la propina, utilizando un diagrama de dispersión:

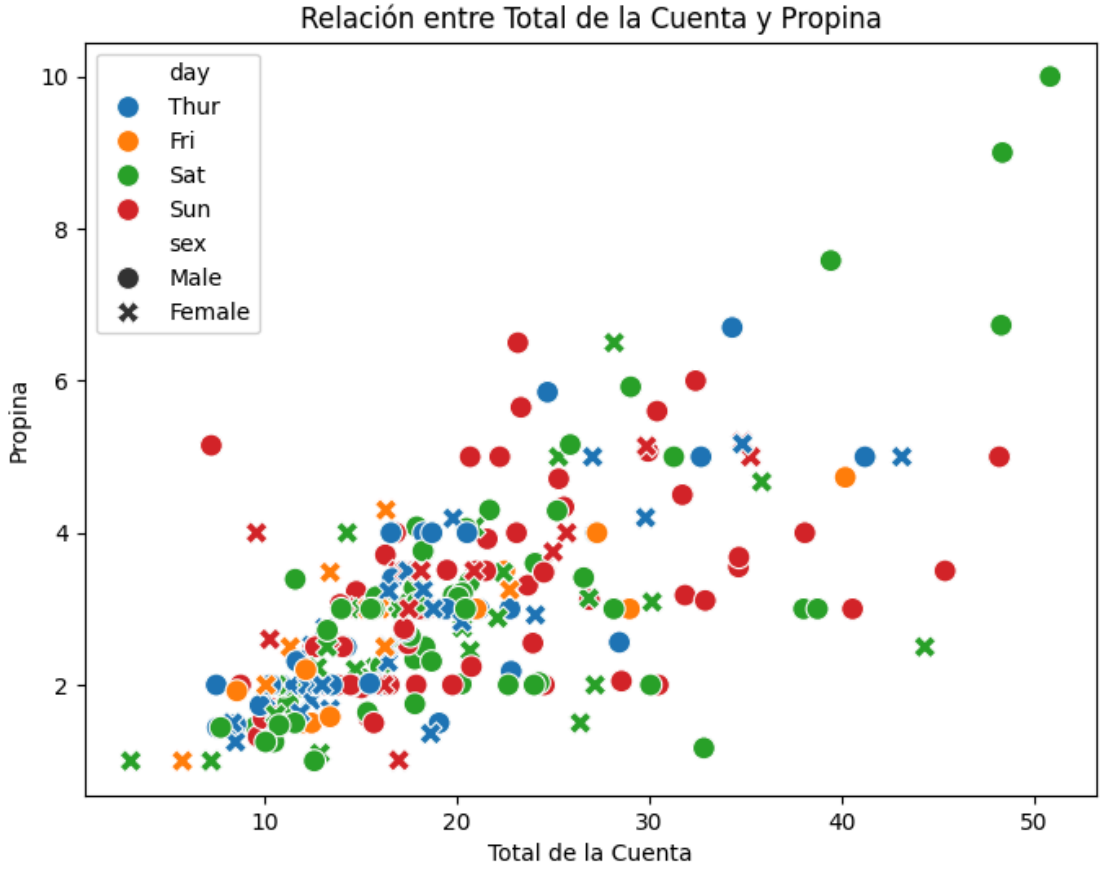
```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=tips, x='total_bill', y='tip',
hue='day', style='sex', s=100)
plt.title('Relación entre Total de la Cuenta y Propina')
plt.xlabel('Total de la Cuenta')
plt.ylabel('Propina')
plt.show()
```

Observaciones:

Hay una clara relación positiva entre el total de la cuenta y la propina: a medida que aumenta el monto total, también tiende a aumentar la propina.
Las propinas de los fines de semana (viernes y sábado) parecen ser más altas en comparación con otros días.

E. Análisis de Correlación

En la próxima clase aprenderemos a analizar la matriz de correlación, con el objetivo de ver cómo se relacionan numéricamente las diferentes variables.



Conclusiones Generales

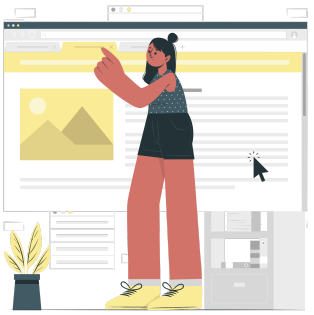
1. **Distribución de Propinas**: La mayoría de las propinas son relativamente bajas, lo que podría implicar que muchos clientes optan por dejar propinas menores a 5.
2. **Relación Total de la Cuenta y Propina**: Existe una relación positiva clara entre el total de la cuenta y la propina; donde las cuentas más altas tienden a recibir propinas más altas.
3. **Días de Mayor Actividad**: Las propinas parecen ser mayores durante los fines de semana, lo que podría relacionarse con un mayor flujo de clientes o el deseo de brindar un mejor servicio.

4. **Correlación**: La correlación observada entre el total de la cuenta y la propina es un indicador importante para entender las expectativas de los clientes respecto a la propina.

A través de esta exploración de datos, hemos sido capaces de identificar patrones y relaciones que pueden ayudar al restaurante a mejorar su servicio o a ajustar sus estrategias de marketing. Las visualizaciones y análisis estadísticos realizados son herramientas valiosas para tomar decisiones informadas en el contexto del negocio.

Reflexión final

En esta clase, aprendimos las **bases del análisis exploratorio de datos (EDA)**, incluyendo técnicas de visualización y resúmenes estadísticos. A medida que avanzamos en el curso, utilizaremos estas herramientas para profundizar en el análisis de conjuntos de datos más complejos y así mejorar nuestras habilidades en data analytics con Python. Practicar estas técnicas es fundamental para convertirte en un analista de datos efectivo.



Materiales y recursos adicionales

- Documentación de Pandas
- Documentación Matplotlib
- Documentación Seaborn
- Repositorio de datasets Seaborn

Próximos Pasos

- Introducción a la correlación y su importancia en el análisis de datos.

Ejercicios Prácticos

Actividad 1: Análisis de Ventas de Diamantes

Contexto

En esta semana como pasante en SynthData serás guiado por Matías, Data Analyst. Recientemente, la empresa ha comenzado a analizar los datos de ventas de diamantes para comprender mejor las preferencias del mercado y optimizar su estrategia de ventas. Tu tarea será realizar un análisis exploratorio de la base de datos "diamonds" de Seaborn, que contiene información sobre características de diamantes.

Objetivos

- Familiarizarte con técnicas de visualización de datos.
- Generar resúmenes estadísticos e identificar patrones en los datos.

Ejercicio práctico

1. Importar las bibliotecas necesarias y cargar el conjunto de datos “diamonds” de Seaborn.

   ```
   diamonds = sns.load_dataset("diamonds")
   ```

2. Visualizar las primeras filas del DataFrame para comprender la estructura de los datos.
3. Realizar un resumen estadístico de los datos y observar las características de las variables numéricas.
4. Crear un gráfico de distribución de los precios con Seaborn.
5. Crear un gráfico de distribución por color.
6. Interpretá los resultados e intentá justificar la elección de los gráficos.

¿Por qué importa esto en SynthData?

Este análisis nos ayuda a entender cómo las características de un producto (en este caso, los diamantes) influyen en las decisiones de compra y el precio. Pronto aprenderemos que la capacidad de visualizar relaciones entre variables será fundamental en el análisis de datos, ya que nos permite tomar decisiones dirigidas al cliente.

Actividad 2: Análisis de Supervivencia

Contexto

Después de completar tu primer ejercicio, Silvia, la Project Manager y Data Scientist, te asignó una nueva tarea relacionada con el historiador de Titanic. Usarás la base de datos "titanic" de Seaborn, que contiene datos sobre los pasajeros del famoso transatlántico para analizar si algunos factores influyeron en la tasa de supervivencia.

Objetivos

- Comprobar las relaciones entre variables categóricas y numéricas.
- Generar visualizaciones que puedan presentar hipótesis sobre las posibilidades de supervivencia.

Ejercicio práctico

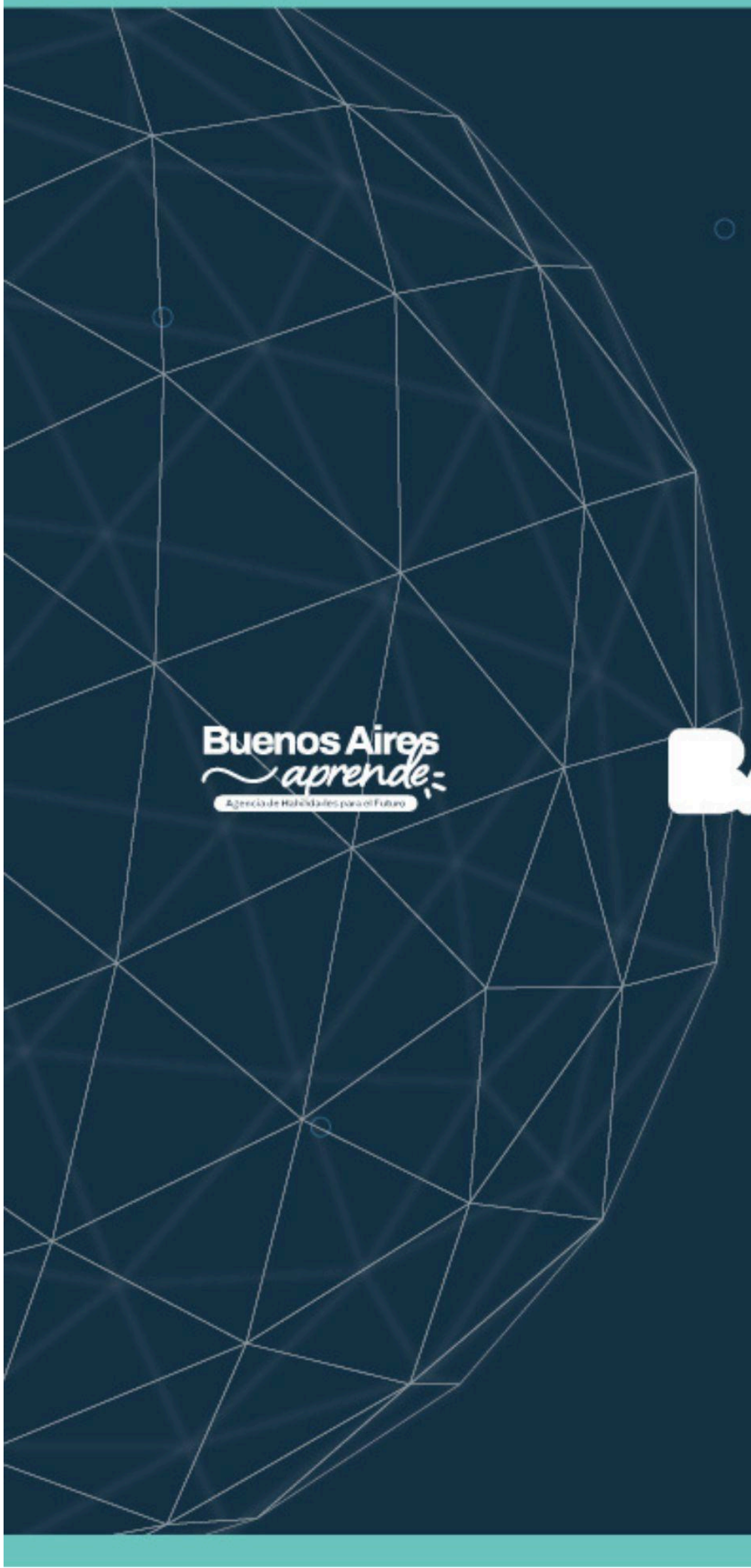
1. Cargá la base de datos “titanic” y visualizá las primeras filas.
2. Realizá un análisis estadístico básico de la tasa de supervivencia por clase de pasajero.
3. Creá un gráfico de barras que muestre la tasa de supervivencia por clase.
4. Analizá si existe alguna correlación entre la edad de los pasajeros y su tasa de supervivencia, utilizando un diagrama de caja (box plot).

¿Por qué importa esto en SynthData?

Comprender qué factores influyen en la tasa de supervivencia, te permite aplicar esta misma técnica al buscar variables que influyen, por ejemplo, en el éxito de ventas de un producto. Esto permitiría a tus potenciales clientes a diseñar productos que respondan mejor a las expectativas del público. Identificar características que correlacionan con el éxito es esencial para hacer recomendaciones informadas a los productores y a los desarrolladores de contenidos. Este tipo de análisis no solo sirve

para obtener una estadística, sino que puede aplicarse al sector del entretenimiento, y a múltiples áreas de negocio.

⛔ **Estos ejercicios son una simulación de cómo se podría resolver el problema en este contexto específico. Las soluciones encontradas no aplican de ninguna manera a todos los casos.**
Recuerda que las soluciones dependen de los sets de datos, el contexto y los requerimientos específicos de los stakeholders y las organizaciones.



Buenos Aires aprende

Agencia de Habilidades para el Futuro

Buenos Aires Ciudad