Análisis Exploratorio

de Datos

Sesión 2

Análisis de Datos: Tipos de Variables y Medidas Estadísticas

- En el análisis de datos, es fundamental identificar correctamente el tipo de variable con el que trabajamos, ya que esto determina cómo podemos analizar, visualizar y manipular la información.

- Es esencial aprender a clasificar variables, calcular medidas de tendencia central y dispersión, y visualizar datos mediante histogramas y diagramas de caja.

Tipos de Variables en Análisis de Datos

1 **Variables Categóricas**

Representan atributos o características sin valor numérico asociado. No tienen un orden cuantificable.

Ejemplos:

- Género: Masculino, Femenino, Otro.

- Color favorito: Rojo, Azul, Verde.

- Tipo de transporte: Auto, Bicicleta, Autobús.

Tipos de Variables en Análisis de Datos

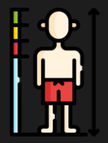
2 **Variables Cuantitativas**

Representan cantidades numéricas que pueden ser discretas o continuas.

Ejemplos: Ejemplos:

- Edad en años (15, 22, 30).

- Altura en cm (175, 182, 160),



- Ingresos mensuales en dólares (1500, 2800, 3200).

Subtipos de Variables

Variables Categóricas

- ✓ Nominales: Sin orden específico (colores)
- ✓ Ordinales: Con orden implícito, pero diferencias no medibles (nivel de satisfacción: bajo, medio, alto)

Ejemplo en Python

```
import pandas as pd

df = pd.DataFrame({"Género": ["Masculino", "Femenino", "Masculino", "Otro"]})
print(df["Género"].value_counts())  # Cuenta la frecuencia de cada categoría
```

Subtipos de Variables

Variables Cuantitativas

- ✓ Discretas: Valores enteros contables, sin decimales (número de hijos: 0, 1, 2)
- ✓ Continuas: Cualquier valor dentro de un rango, incluyendo decimales (temperatura: 36.5°C, 22.3°C)

Ejemplo en Python

```python
import numpy as np

# Variable Discreta
num_hijos = np.array([0, 1, 2, 3, 2, 1, 0, 3])
print("Promedio de hijos:", np.mean(num_hijos))

# Variable Continua
pesos = np.array([65.2, 70.1, 68.5, 72.3])
print("Peso máximo:", np.max(pesos))
```

Tablas de Frecuencia

Una tabla de frecuencia es una herramienta estadística utilizada para organizar y resumir datos, mostrando cuántas veces aparece cada valor dentro de una variable. Ayuda a visualizar la distribución de los datos y a identificar patrones de frecuencia.

Tipos

1. Frecuencia absoluta: Número de veces que aparece cada categoría.
2. Frecuencia relativa: Proporción de cada categoría en relación con el total.
3. Frecuencia acumulada: Suma progresiva de las frecuencias.

Ejemplo en Python

```python
import pandas as pd

# Datos de ejemplo: Calificaciones de un grupo de estudiantes
datos = [7, 8, 6, 9, 7, 6, 8, 9, 7, 8, 7, 9, 6, 7, 8]

# Crear un DataFrame
df = pd.DataFrame(datos, columns=["Calificación"])

# Crear la tabla de frecuencias
tabla_frecuencia = df["Calificación"].value_counts().sort_index().to_frame("Frecuencia Absoluta")
tabla_frecuencia["Frecuencia Relativa"] = tabla_frecuencia["Frecuencia Absoluta"] / tabla_frecuencia["Frecuencia Absoluta"].sum()
tabla_frecuencia["Frecuencia Acumulada"] = tabla_frecuencia["Frecuencia Absoluta"].cumsum()
tabla_frecuencia["Frecuencia Relativa Acumulada"] = tabla_frecuencia["Frecuencia Relativa"].cumsum()

# Mostrar la tabla de frecuencias
print(tabla_frecuencia)
```

Interpretación de la tabla

Calificación	Frecuencia Absoluta	Frecuencia Relativa	Frecuencia Acumulada	Frecuencia Relativa Acumulada
6	3	0.20	3	0.20
7	5	0.33	8	0.53
8	4	0.27	12	0.80
9	3	0.20	15	1.00

Frecuencia relativa: Proporción de cada categoría en relación con el total.

✓ Frecuencia Absoluta: La calificación 7 apareció 5 veces.

✓ Frecuencia Relativa: El 20% de los estudiantes obtuvo una calificación de 6.

✓ Frecuencia Acumulada: Hasta la calificación 8, ya se han registrado 12 estudiantes.

✓ Frecuencia Relativa Acumulada: El 80% de los estudiantes obtuvo una calificación de 8 o menor.

Medidas de Tendencia Central

Media (Promedio)

Es el valor promedio de un conjunto de datos. Se calcula sumando todos los valores y dividiendo entre el número total de elementos. Es útil cuando los datos son simétricos, pero sensible a valores atípicos.

Mediana

Es el valor central de un conjunto de datos ordenados. Si el número de elementos es impar, será el valor del medio. Si es par, es el promedio de los dos valores centrales. No se ve afectada por valores extremos.

Moda

Es el valor que aparece con mayor frecuencia. Puede no existir o tener más de un valor. Es útil para identificar categorías o valores más frecuentes, especialmente en datos cualitativos.

Medidas de Dispersión

1

Rango de Variación

Es la diferencia entre el valor máximo y mínimo de un conjunto de datos. Aunque es fácil de calcular, puede verse afectado por valores atípicos, por lo que no siempre refleja con precisión la dispersión real.

$$Rango = \text{Valor Máximo} - \text{Valor Mínimo}$$

2

Varianza

Indica cuánto se alejan los valores respecto a la media. Se calcula como el promedio de las diferencias al cuadrado entre cada valor y la media. Una varianza alta indica mayor dispersión de los datos.

$$\sigma^2 = \frac{\sum (X_i - \mu)^2}{N}$$

3

Desviación Estándar

Es la raíz cuadrada de la varianza y mide cuánto se alejan los valores de la media. Está expresada en las mismas unidades que los datos originales, lo que facilita su interpretación.

$$\sigma = \sqrt{\text{Varianza}}$$

Población, Muestra y Corrección de Bessel

1

Población

Es el conjunto completo de datos sobre el cual se desea realizar un estudio. Incluye a todas las personas, objetos o eventos que cumplen con ciertos criterios de interés para la investigación.

Ejemplo

```
import numpy as np

# Definir la población completa
poblacion = np.array([5, 10, 15, 20, 25, 30, 35, 40])
```

2

Muestra

Es un subconjunto de la población seleccionado para análisis debido a limitaciones de tiempo, costo o accesibilidad. Debe ser representativa para que los resultados puedan generalizarse a toda la población.

Ejemplo

```
# Seleccionar una muestra aleatoria de 5 elementos sin reemplazo
muestra = np.random.choice(poblacion, size=5, replace=False)
```

Población, Muestra y Corrección de Bessel

3 **Corrección de Bessel**

Se usa al calcular la varianza y desviación estándar de una muestra. Se divide entre N-1 en lugar de N para corregir la subestimación de la variabilidad poblacional, proporcionando una estimación más precisa.

Ejemplo

```
# Cálculo de la varianza con y sin corrección de Bessel
varianza_poblacional = np.var(muestra)  # División entre N
varianza_muestral = np.var(muestra, ddof=1)  # División entre N-1

print("Varianza poblacional:", varianza_poblacional)
print("Varianza muestral con corrección de Bessel:", varianza_muestral)
```

Medidas de Posición

Cuartiles
Dividen los datos en cuatro partes iguales: Q1 (25%), Q2 (50%, mediana) y Q3 (75%). Son útiles para entender la distribución de los datos y detectar valores atípicos.

Quintiles
Dividen los datos en cinco partes iguales. Cada quintil representa un 20% de los datos, permitiendo un análisis más detallado de la distribución.

Deciles
Dividen los datos en diez partes iguales. Cada decil representa un 10% de los datos, ofreciendo una visión más granular de la distribución.

Percentiles
Dividen los datos en cien partes iguales. Cada percentil representa un 1% de los datos, proporcionando el análisis más detallado de la distribución.

Ejemplo en Python

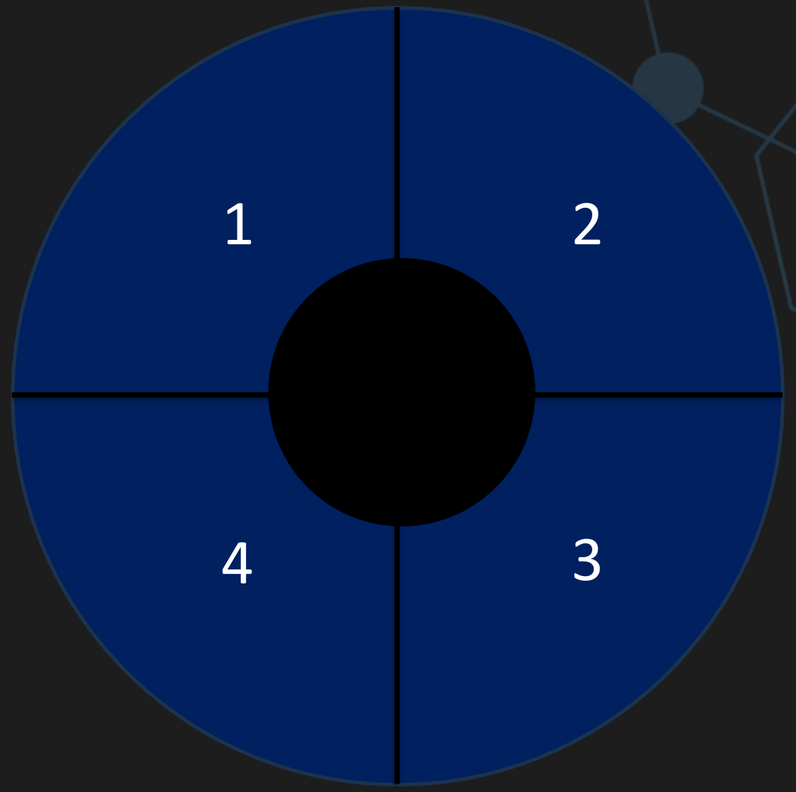
```python
import numpy as np

# Datos de ejemplo
datos = np.array([5, 10, 15, 20, 25, 30, 35, 40])

# Cálculo de los cuartiles
Q1 = np.percentile(datos, 25)
Q2 = np.percentile(datos, 50)  # Mediana
Q3 = np.percentile(datos, 75)

print("Primer cuartil (Q1):", Q1)
print("Mediana (Q2):", Q2)
print("Tercer cuartil (Q3):", Q3)
```

Explicación:
Se usa np.percentile() para calcular cada cuartil. Estos valores ayudan a entender cómo se distribuyen los datos y detectar posibles valores atípicos.

Puntos Atípicos (Outliers)

Definición

Un outlier es un valor que se aleja significativamente del resto de las observaciones, siendo inusualmente alto o bajo en comparación.



Causas

Pueden aparecer por errores de medición, variabilidad natural de los datos o eventos excepcionales.

Manejo

Se pueden eliminar, transformar los datos o usar métodos estadísticos robustos menos sensibles a outliers.

Detección

El método del Rango Intercuartílico (IQR) es común: se calculan límites usando Q1-1.5*IQR y Q3+1.5*IQR.

Ejemplo en Python:

```python
import numpy as np

# Datos de ejemplo con un posible outlier
datos = np.array([10, 12, 15, 14, 13, 102, 11, 12, 14, 13])

# Cálculo de cuartiles e IQR
Q1 = np.percentile(datos, 25)
Q3 = np.percentile(datos, 75)
IQR = Q3 - Q1

# Definir límites para detectar outliers
limite_inferior = Q1 - 1.5 * IQR
limite_superior = Q3 + 1.5 * IQR

# Identificar valores atípicos
outliers = datos[(datos < limite_inferior) | (datos > limite_superior)]

print("Datos originales:", datos)
print("Q1:", Q1, "| Q3:", Q3, "| IQR:", IQR)
print("Límite inferior:", limite_inferior, "| Límite superior:", limite_superior)
print("Outliers detectados:", outliers)
```

Explicación:

- Se define un conjunto de datos donde 102 es un posible outlier.
- Se calculan los cuartiles Q1 y Q3, y luego el IQR.
- Se establecen los límites de detección basados en la regla de 1.5 × IQR.
- Se identifican los valores fuera de estos límites como outliers.

Visualización de Datos: Histograma y Boxplot

Histograma

El histograma muestra la distribución de frecuencias de una variable numérica, agrupando datos en intervalos o "bins". Permite observar la forma de la distribución, dispersión y tendencia central.

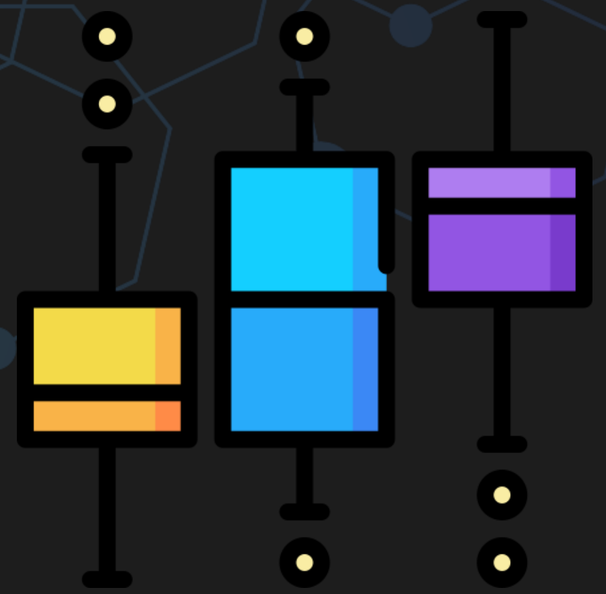
Ejemplo en Python

```python
import matplotlib.pyplot as plt

plt.hist(datos, bins=5, edgecolor="black")
plt.xlabel("Valores")
plt.ylabel("Frecuencia")
plt.title("Histograma de Datos")
plt.show()
```

Visualización de Datos: Histograma y Boxplot

Boxplot (Diagrama de Caja)

El boxplot (diagrama de caja y bigotes) resume la distribución de datos mostrando la mediana, cuartiles y valores atípicos. Es ideal para identificar la dispersión, detectar outliers y comparar distribuciones de diferentes categorías.



Ejemplo en Python

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

# Suponiendo que 'datos' es una lista o un array de datos
sns.boxplot(x=datos)

# Títulos y etiquetas
plt.title("Boxplot de Datos")

# Mostrar gráfico
plt.show()
```

Actividad Práctica Guiada

Objetivo: Identificar y analizar outliers en un conjunto de datos utilizando Python

Requisitos:

1. **Importar librerías.**
2. **Crear o cargar un conjunto de datos (ver guía de estudio).**
3. **Visualizar los datos.**
4. **Identificar outliers usando el rango intercuartílico (IQR).**
5. **Analizar los outliers.**
6. **Manejar los outliers: eliminar si son errores, transformar datos, usar métodos robustos.**
7. **Visualizar los datos sin outliers.**

El detalle de la actividad se encuentra en la guía de estudio de la sesión.

Preguntas

Sección de preguntas

Análisis Exploratorio

de Datos

Continúe con las actividades