

ESTADÍSTICA DESCRIPTIVA

Librerías:

```
from scipy.stats import shapiro, kstest
import warnings
warnings.filterwarnings("ignore")
from itertools import combinations
```

PARA VARIABLES NUMÉRICAS

Sintaxis:

```
media = df['columna'].mean()
print(f"La media de los ingresos de los clientes es: {np.round(media_ingresos, 2)}")
```

MEDIDAS DE CENTRALIZACIÓN

Media: df['columna'].mean()

Mediana: df['columna'].median() Visualización → **sns.boxplot**

Moda: df['columna'].mode() Visualización → **sns.countplot** (categóricas)
sns.histplot (numéricas)

MEDIDAS DE DISPERSIÓN

Varianza: df['columna'].var() Visualización → **sns.histplot**

Indica cuán dispersos están los valores alrededor de su valor promedio.

Desviación estándar: df['columna'].std() Visualización → **sns.histplot**

Indica cuánto varían los valores respecto a la media. Raíz cuadrada de la varianza.

Está en la misma medida que la variable original.

MEDIDAS DE CORRELACIÓN

Relación entre dos variables Visualización → **sns.scatterplot**

El método **combinations** hace todas las combinaciones de columnas de un DF

```
numericas = df.select_dtypes(include = np.number).columns
combinaciones = list(combinations(numericas, 2))
```

Gráfica de correlación:

Sintaxis: `df_correlaciones = df.corr(method = "spearman")`

- Spearman: no lineales
- Pearson: lineales

Interpretación: relación de todas las combinaciones de columnas.

- Correlación débil: $< 0,3$
- Correlación moderada: $0,3 - 0,7$
- Correlación fuerte: $> 0,7$

Visualización:

`sns.heatmap(df_correlaciones, sobre qué datos queremos hacer el heatmap`
`annot = True,` para que nos muestre los valores de cada una de las celdas del heatmap
`fmt = ".2f",` para que nos redondee a dos decimales
`cmap = "mako",` para establecer el color de la gráfica
`vmax = 1,` para indicar cual es el valor máximo de nuestra escala de colores
`vmin = -1);` para identificar el valor mínimo de nuestra escala de colores

Si la tabla tiene los mismos valores, como un espejo, eliminar una de las partes.

Creamos una matriz booleana de máscara:

`mask = np.triu(np.ones_like(df_correlaciones, dtype=bool))`

Creamos un heatmap con la máscara

`sns.heatmap(df_correlaciones, annot = True, fmt = ".2f", cmap = "mako", vmax = 1, vmin = -1, mask = mask);`

MEDIDAS DE POSICIÓN RELATIVAS

Comprender la distribución de los datos e identificar valores atípicos.

Visualización → boxplot

Sintaxis:

`percentile_25_75 = np.percentile(df['age'], [25, 75])`

`print(f'El percentil 25 de la columna 'age' es: {percentile_25_75[0]}') → 33`

`print(f'El percentil 75 de la columna 'age' es: {percentile_25_75[1]}') → 46`

Percentiles:

Percentil 25: el 25% de los clientes tienen una edad igual o inferior a 33

Percentil 75: el 75% de los clientes tienen una edad igual o inferior a 46

$IQR(\text{rango intercuartílico}) = Q3 - Q1$ Salto = $IQR \times 1,5$

`Q1/Q3 = np.nanpercentile(df["columna"], 25/75) →` Donde empiezan los bigotes

`bigote_izq=Q1- salto` `bigote_der=Q3+salto →` Los outliers por abajo y por arriba

MEDIDAS DE RESUMEN CATEGÓRICAS

Frecuencia absoluta: cuántos valores hay por categoría.

```
frecuencia_absoluta = df['columna'].value_counts()
```

Frecuencia relativa: porcentaje de valores por cada categoría.

```
frecuencia_relativa = df['columna'].value_counts(normalize=True) * 100
```

Crear un DataFrame para mostrar la tablas:

```
tabla_frecuencias = pd.DataFrame({  
    'Estado Civil': frecuencia_absoluta.index,  
    'Frecuencia Absoluta': frecuencia_absoluta.values,  
    'Frecuencia Relativa (%)': frecuencia_relativa.values})
```

Visualización → **sns.barplot**

TABLA DE CONTINGENCIA

Porcentaje de valores para cada combinación de los valores únicos de dos columnas categóricas.

```
tabla_contingencia = pd.crosstab(df['education'], df['marital'], normalize=True) *  
100
```

	marital	divorced	married	single
education				
basic 4y	1.254955	8.186395	1.138215	
basic 6y	0.462096	4.482331	0.841501	
basic 9y	1.439794	10.533356	3.348980	
high school	3.015784	13.065156	8.021013	
illiterate	0.004864	0.036481	0.002432	
professional course	1.675706	8.013717	3.614077	
university degree	3.383029	16.219569	11.260549	