



Estadística Descriptiva Básica - Día 2

Change Status

in-progress

40 min

Learning Objectives

- 1 Calcular e interpretar medidas de tendencia central (media, mediana, moda)
- 2 Comprender y aplicar medidas de dispersión (varianza, desviación estándar, rango)
- 3 Utilizar estadísticos de posición (percentiles, quartiles) para análisis de distribución

Theory

Practice

Quiz

Evidence

<> **Ejercicio**: Análisis estadístico descriptivo completo de dataset de empleados

Ejercicio práctico para aplicar los conceptos aprendidos.

Crear y explorar dataset de empleados:

```
import pandas as pd
import numpy as np

# Crear dataset de empleados
np.random.seed(42)
n_empleados = 500

df = pd.DataFrame({
    'id_empleado': range(1, n_empleados + 1),
    'edad': np.random.normal(35, 8, n_empleados).clip(22, 65).astype(int),
    'salario': np.random.lognormal(10.5, 0.4, n_empleados).round(0),
    'años_experiencia': np.random.normal(8, 4, n_empleados).clip(0, 30).astype(int),
    'departamento': np.random.choice(['IT', 'Ventas', 'Marketing', 'RRHH', 'Finanzas'], n_empleados),
    'nivel': np.random.choice(['Junior', 'Senior', 'Lead', 'Manager'], n_empleados, p=[0.4, 0.4, 0.15, 0.05])
})
```



us English



Sign Out





```
print(f"Dataset creado: {df.shape[0]} empleados")
print(f"Columns: {list(df.columns)}")
```

Análisis de medidas de tendencia central:

```
# Estadísticos básicos
print("MEDIDAS DE TENDENCIA CENTRAL")
print("=" * 40)

# Variables numéricas
for col in ['edad', 'salario', 'años_experiencia']:
    print(f"\n{col.upper()}:")
    print(f"  Media: {df[col].mean():.2f}")
    print(f"  Mediana: {df[col].median():.2f}")
    print(f"  Moda: {df[col].mode().iloc[0] if len(df[col].mode()) > 0 else 'Sin moda única'}")

# Variables categóricas
print(f"\nDEPARTAMENTO MÁS COMÚN: {df['departamento'].mode().iloc[0]}")
print(f"NIVEL MÁS COMÚN: {df['nivel'].mode().iloc[0]}")
```

Análisis de dispersión:

```
print("\n\nMEDIDAS DE DISPERSIÓN")
print("=" * 30)

for col in ['edad', 'salario', 'años_experiencia']:
    print(f"\n{col.upper()}:")
    print(f"  Rango: {df[col].max() - df[col].min()}")
    print(f"  Varianza: {df[col].var():.2f}")
    print(f"  Desviación estándar: {df[col].std():.2f}")
    print(f"  Coeficiente de variación: {df[col].std() / df[col].mean():.3f}")
```

Análisis de percentiles y distribución:

```
print("\n\nANÁLISIS DE PERCENTILES")
print("=" * 25)

for col in ['edad', 'salario', 'años_experiencia']:
    print(f"\n{col.upper()}:")
```



```
percentiles = df[col].quantile([0.1, 0.25, 0.5, 0.75, 0.9])
for p, v in percentiles.items():
    print(f"  P{int(p*100)}: {v:.2f}")

# Rango intercuartílico
q1, q3 = df[col].quantile([0.25, 0.75])
iqr = q3 - q1
print(f"  IQR: {iqr:.2f}")

# Límites para outliers
limite_inf = q1 - 1.5 * iqr
limite_sup = q3 + 1.5 * iqr
outliers = ((df[col] < limite_inf) | (df[col] > limite_sup)).sum()
print(f"  Outliers (IQR): {outliers}")
```

Análisis por categorías:

```
print("\n\nANÁLISIS POR DEPARTAMENTO")
print("=" * 30)

dept_stats = df.groupby('departamento').agg({
    'salario': ['mean', 'median', 'std', 'count'],
    'edad': ['mean', 'min', 'max']
}).round(2)

print(dept_stats)

print("\n\nANÁLISIS POR NIVEL")
print("=" * 20)





nivel_stats = df.groupby('nivel').agg({
    'salario': 'mean',
    'años_experiencia': 'mean',
    'edad': 'mean'
}).round(2)

print(nivel_stats)
```

Verificación: Compara cómo las medidas de tendencia central (media vs mediana) difieren en variables con outliers, y explica por qué elegirías una u otra para diferentes tipos de análisis.

Requerimientos:



-  Dashboard
-  Career Path
-  Forms
-  Profile



Python con Pandas y NumPy
Dataset numérico para análisis estadístico
matplotlib opcional para histogramas básicos