

Data Analytics

–

Python

–

Visualización



TEMA 4 – Visualización

Índice

1. Librerías
2. Gráfico de barras
3. Histograma
4. Diagrama de dispersión
5. Gráficos de líneas
6. Diagramas de caja
7. Gráficos circulares
8. Mapas de calor



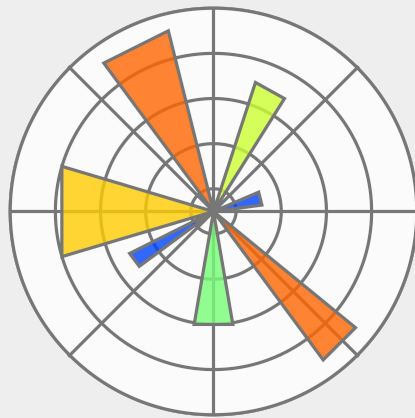
1. Librerías

Las librerías principales son **matplotlib** y **seaborn**. Hay muchas otras más como *plotly*, *bokeh*, *altair*...

Seaborn está basada en matplotlib y tiene varios datasets para practicar.

```
# instalando librerías  
pip install matplotlib  
pip install seaborn
```

```
# importando librerías  
import matplotlib.pyplot as plt  
import seaborn as sns
```



seaborn

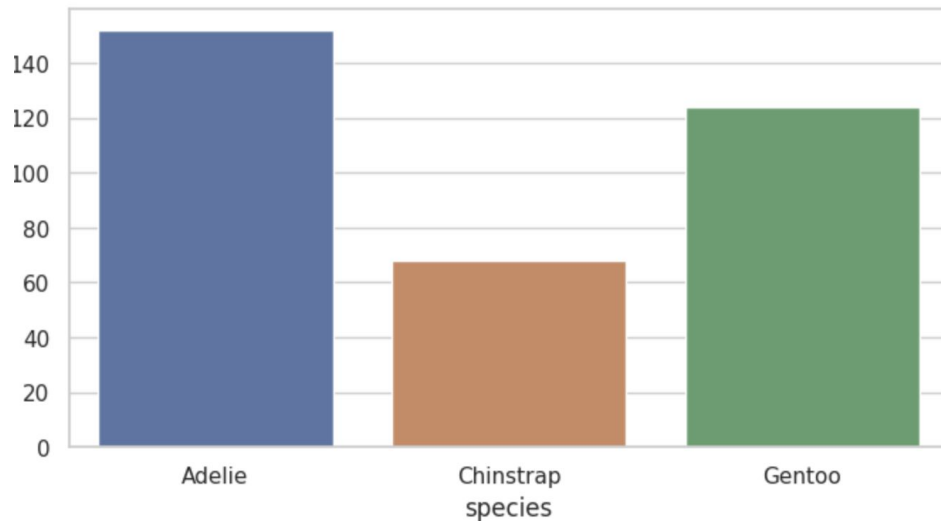


2. Gráfico de barras

Los gráficos de barras comparan cantidades de diferentes **categorías**. Son excelentes para visualizar diferencias entre grupos. Ejemplo: Comparar el número de ofertas de trabajo entre diferentes plataformas de empleo, resaltando cuáles tienen más oportunidades en análisis de datos.

```
# matplotlib  
plt.bar(data['categoria'], data['valor'])
```

```
# seaborn  
sns.barplot(x='categoria', y='valor',  
data=data)
```

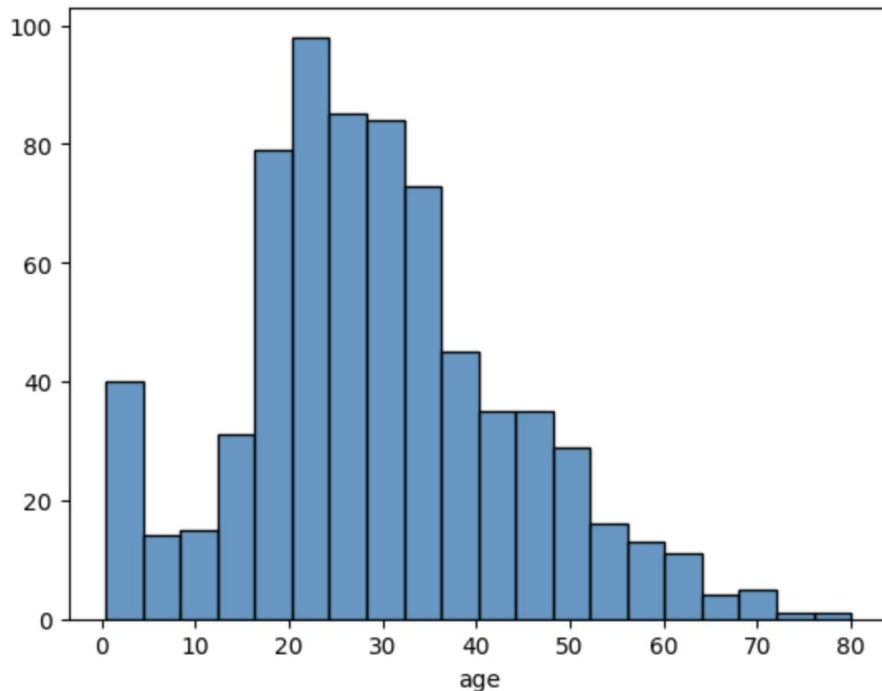


3. Histograma

Un histograma muestra la **distribución** de un conjunto de datos y es ideal para entender la frecuencia de diferentes rangos de valores en una variable. Aplicación típica: analizar la distribución salarial de roles en análisis de datos, identificando concentraciones de salarios bajos, medios y altos.

```
# matplotlib
plt.hist(data['columna'], bins=10)

# seaborn
sns.histplot(data['columna'], bins=10)
```

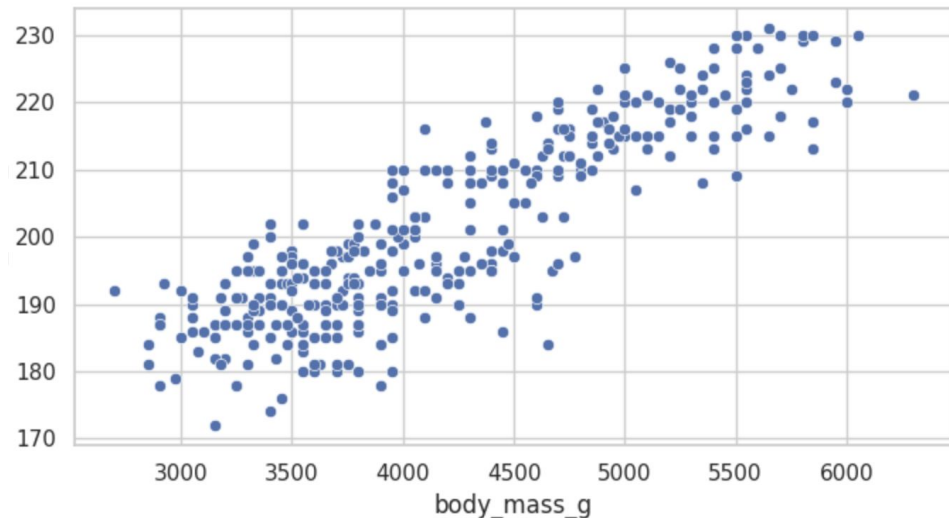


4. Diagrama de dispersión

Los diagramas de dispersión muestran la **relación** entre dos variables y pueden revelar **correlaciones**. Ejemplo: Investigar la relación entre la experiencia laboral y el salario en el campo del análisis de datos, identificando patrones de correlación positiva o negativa.

```
# matplotlib
plt.scatter(data['variable_x'],
            data['variable_y'])

# seaborn
sns.scatterplot(x='variable_x', y='variable_y',
               data=data)
```

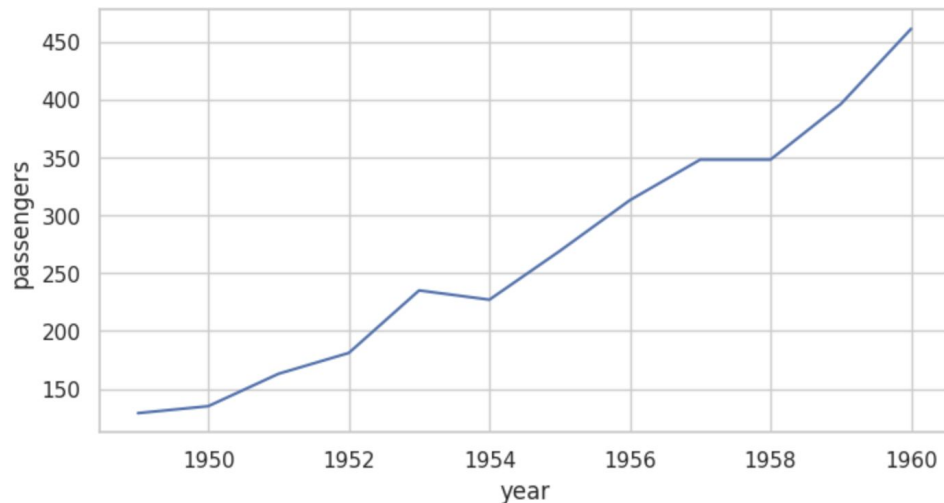


5. Gráfico de líneas

Los gráficos de líneas trazan **datos a lo largo del tiempo**, perfectos para visualizar tendencias y cambios. Aplicación: Seguimiento del crecimiento de puestos de trabajo en análisis de datos a lo largo de los años, mostrando tendencias al alza o a la baja.

```
# matplotlib
plt.plot(data['tiempo'], data['valor'])

# seaborn
sns.lineplot(x='tiempo', y='valor', data=data)
```

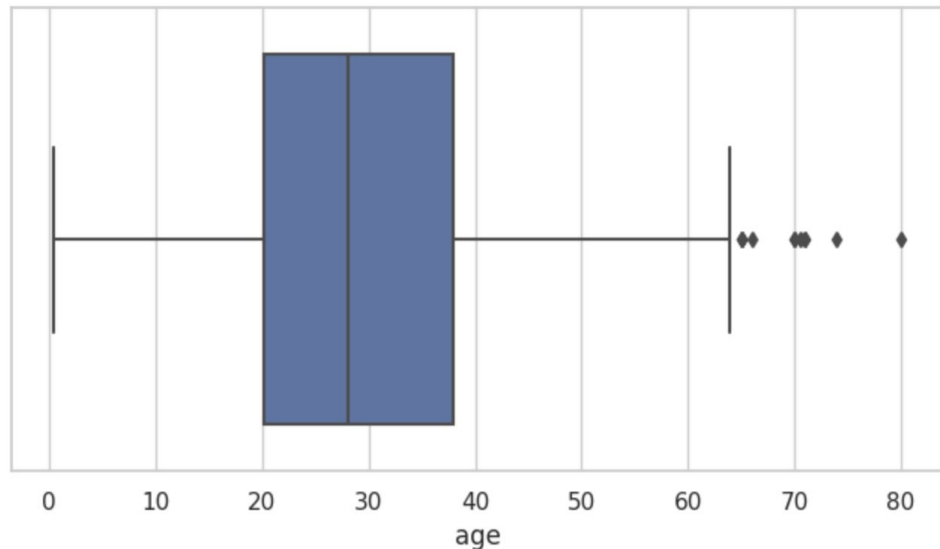


6. Diagrama de caja

Los diagramas de caja resumen **distribuciones** de datos a través de cuartiles, destacando medianas, rangos intercuartílicos y valores atípicos. Nos ayudan a identificar rápidamente la presencia de **outliers**. Ejemplo: Comparar rangos salariales entre diferentes niveles de roles en análisis de datos, desde junior hasta senior.

```
# matplotlib
plt.boxplot(data['columna'])

# seaborn
sns.boxplot(y=data['columna'])
```

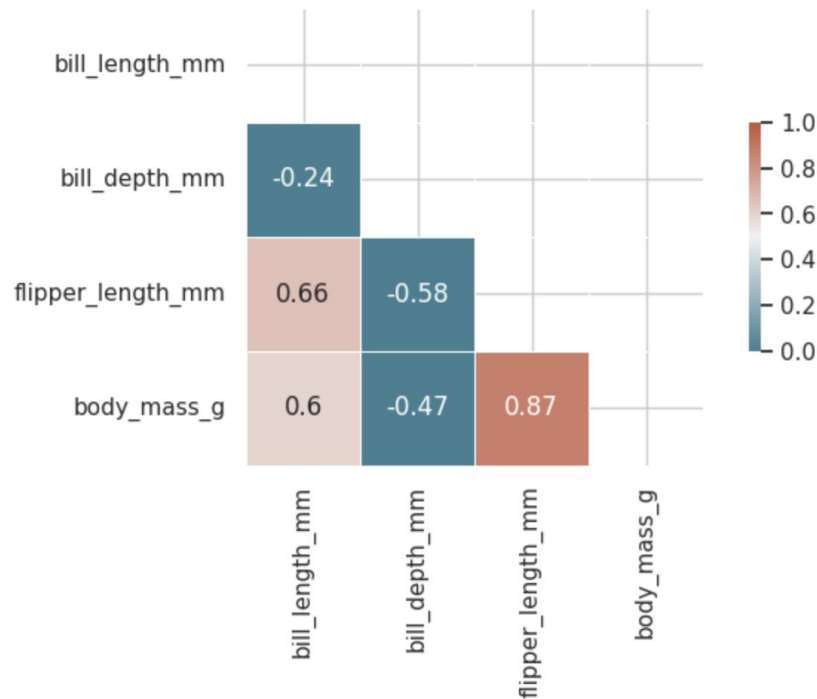


7. Mapa de calor

Los mapas de calor utilizan colores para representar magnitudes en datos bidimensionales, ideales para análisis complejos. También para ver en una matriz la correlación entre variables. Aplicación: Evaluar la densidad de ofertas de trabajo en análisis de datos por región, identificando zonas con alta y baja demanda.

```
# matplotlib
plt.imshow(data, cmap='hot',
            interpolation='nearest')
```

```
# seaborn
sns.heatmap(data)
```



8. Gráficos circulares

Los gráficos circulares ilustran **proporciones** dentro de un todo, útiles para destacar cómo se dividen los segmentos. Hay que evitar poner muchas categorías para facilitar la legibilidad. Aplicación: Mostrar la distribución de trabajos de análisis de datos por sector, evidenciando dónde hay mayor demanda.

```
# matplotlib
plt.pie(data['valor'],
labels=data['categoria'], autopct='%1.1f%%')
```

```
# seaborn
No tiene función directa para tal fin
```

