ADVANCE
Consortium

ATENEA 2025
BOOTCAMPS

Identificación Material

Bootcamp	Soluciones de Futuro con IA y Datos- **Intermedio**
Módulo	Desarrollo de soluciones basadas en datos
Eje Temático	Visualización y análisis exploratorio

ADVANCE Consortium

Visualizaciones Python

Visualizaciones

La **Visualización** es un aspecto crucial del análisis y la interpretación de datos, ya que **permite comprender fácilmente conjuntos de datos complejos**. Ayuda a **identificar patrones, relaciones y tendencias** que podrían no ser evidentes sólo a través de los datos brutos. En los últimos años, Python se ha convertido en uno de los lenguajes de programación más populares para el análisis de datos, debido a su amplia gama de bibliotecas y marcos de trabajo.

Las bibliotecas de visualización en Python permiten a los usuarios crear visualizaciones de datos intuitivas e interactivas que pueden comunicar eficazmente información a un público amplio. Algunas de las bibliotecas y marcos de visualización más populares en Python son Matplotly, Plotly y Seaborn. Cada una de estas bibliotecas tiene sus propias características y capacidades que responden a necesidades específicas.

Visualizaciones

La **visualización de datos** es crucial en el análisis por varias razones fundamentales que van más allá de la presentación estética de la información, no solo hace que los datos sean más accesibles, sino que también transforma la información en conocimiento práctico. Al facilitar la interpretación, la identificación de patrones y la toma de decisiones, las visualizaciones se convierten en una herramienta esencial para cualquier análisis efectivo Aquí algunos puntos esenciales en el análisis:

1. Comprensión Rápida y Eficiente: La mayoría de las personas procesa la información visual más rápidamente que la información textual o numérica. La visualización de datos proporciona una forma rápida y eficiente de comprender patrones, tendencias y relaciones en grandes conjuntos de datos.

2. Identificación de Patrones y Tendencias: Las visualizaciones permiten identificar patrones, tendencias y correlaciones que podrían no ser evidentes al examinar simplemente los datos en forma de números o texto. Los gráficos y las tablas facilitan la detección de comportamientos significativos.

Visualizaciones

3. Facilita la Comunicación de Resultados: Las visualizaciones son herramientas poderosas para comunicar hallazgos de manera clara y efectiva a una audiencia diversa. Un gráfico puede transmitir información compleja de manera accesible, lo que facilita la presentación de resultados a colegas, clientes o la alta dirección.

4. Identificación de Anomalías y Excepciones: La visualización de datos permite identificar rápidamente anomalías, valores atípicos y excepciones en los datos. Estos puntos destacados pueden ser críticos para la toma de decisiones y la comprensión de eventos inusuales.

5. Apoyo a la Toma de Decisiones: La visualización de datos proporciona información clave para la toma de decisiones informadas. La representación gráfica facilita la evaluación de escenarios, comparaciones y la evaluación de opciones, lo que contribuye a decisiones más fundamentadas.

6. Enfoque en la Acción: Las visualizaciones destacan la información esencial, permitiendo que los analistas se centren en la acción necesaria. Esto evita la sobrecarga de información y garantiza que la atención se centre en los aspectos más relevantes del análisis.

Visualizaciones

7. Exploración Interactiva de Datos: Herramientas de visualización como Tableau y Power BI ofrecen capacidades interactivas que permiten a los usuarios explorar datos a diferentes niveles de detalle, profundizando según sea necesario.

8. Mejora la Memoria y Retención: La información visual tiende a ser más memorable que la información textual. Las visualizaciones eficaces facilitan la retención de información, lo que es crucial para la aplicación práctica de los resultados del análisis.

Principios de visualización

1. Simplicidad: La simplicidad se refiere a la idea de presentar información de manera clara y directa, evitando el exceso de detalles innecesarios.

Ejemplos Prácticos:

- Utilizar gráficos simples y comprensibles.
- Limitar el número de colores y elementos decorativos.
- Eliminar información redundante que no contribuya a la comprensión.

Visualizaciones

Principios de visualización

2. **Claridad:** La claridad implica que la visualización comunica el mensaje de manera fácilmente comprensible, evitando confusiones o malinterpretaciones.

   Ejemplos Prácticos:

   - Etiquetar ejes y puntos clave para una fácil interpretación.
   - Asegurarse de que los colores utilizados sean distinguibles.
   - Evitar la saturación visual y el desorden en la presentación.

3. Relevancia: La relevancia se refiere a presentar solo la información esencial que contribuye al entendimiento del mensaje principal.

   Ejemplos Prácticos:

   - Centrarse en los datos más importantes para la audiencia.
   - Eliminar elementos que no aporten al propósito de la visualización.
   - Priorizar la información clave para destacar puntos significativos.

Visualizaciones

TIPOS DE GRÁFICOS

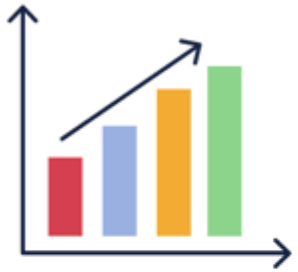


Gráfico de Barras

Gráfico de Lineas

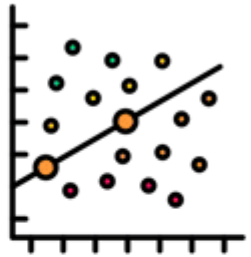


Gráfico de Dispersión

Mapa de Calor

Gráfico Circular

Diagrama de Red

Visualizaciones

TIPOS DE GRÁFICOS

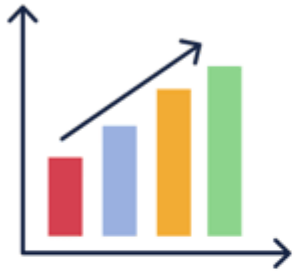


Gráfico de Barras

Gráfico de Lineas

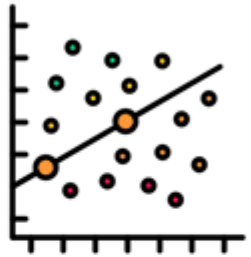


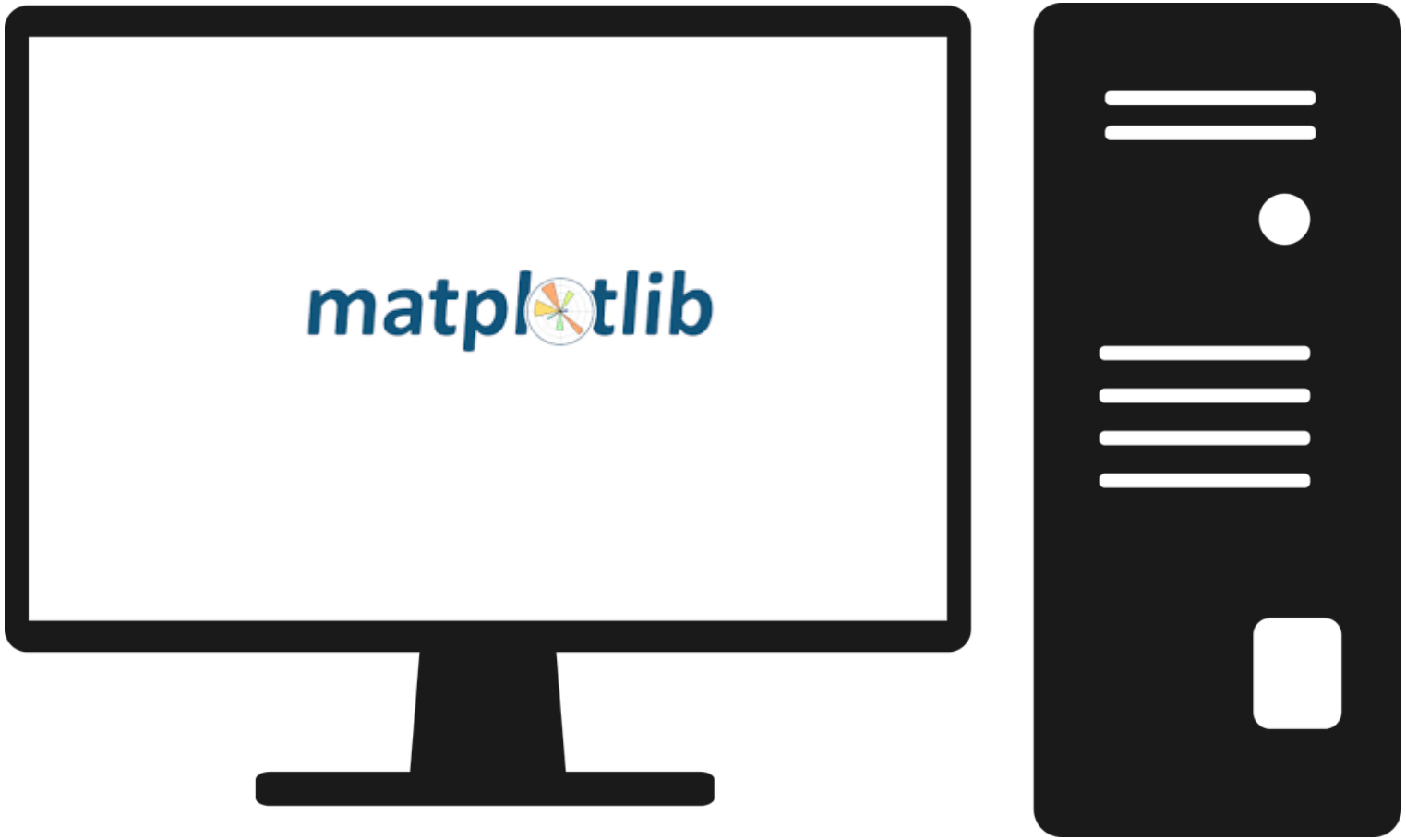
Gráfico de Dispersión

Mapa de Calor

Gráfico Circular

Diagrama de Red

INSTALAR MATPLOTLIB



NumPy

Matplotlib es una biblioteca completa para **crear visualizaciones estáticas, animadas e interactivas** en Python. Permite crear y personalizar los tipos de gráficos más comunes, entre ellos: Diagramas de barras, Histograma, Diagramas de sectores, Diagramas de caja y bigotes, Diagramas de violín, Diagramas de dispersión o puntos, Diagramas de líneas, Diagramas de áreas, Diagramas de contorno, Mapas de color

INSTALAR MATPLOTLIB

- **Paso No.1:** Nos ubicamos en la ruta de la carpeta de Python.

  Ejemplo: C:\Users\USUARIO\AppData\Local\Programs\Python\Python311

- **Paso No.2:** Abrimos **cmd** con esa ruta y ejecutamos el comando PIP:

  py -m pip install matplotlib

INSTALAR SEABORN

SeaBorn

Seaborn es una conocida **biblioteca de Python para la visualización de datos** que ofrece una interfaz fácil de usar para **producir gráficos estadísticos visualmente atractivos e informativos**. Está diseñado para trabajar con marcos de datos Pandas, facilitando la visualización y exploración de datos de forma rápida y eficaz.

Seaborn ofrece una variedad de potentes herramientas para la visualización de datos, incluyendo **gráficos de dispersión, gráficos de líneas, gráficos de barras, mapas de calor, y muchos más**. También permite realizar análisis estadísticos avanzados, como análisis de regresión, gráficos de distribución y gráficos categóricos

INSTALAR SEABORN

- **Paso No.1:** Nos ubicamos en la ruta de la carpeta de Python.

  Ejemplo: C:\Users\USUARIO\AppData\Local\Programs\Python\Python311

- **Paso No.2:** Abrimos **cmd** con esa ruta y ejecutamos el comando PIP:

  py -m pip install seaborn

Visualizaciones – Tipos de Gráficos

Pandas proporciona métodos de visualización básicos que permiten crear gráficos directamente desde un DataFrame. Estos métodos son convenientes para realizar exploración visual de los datos antes de realizar análisis más detallados con bibliotecas de visualización más avanzadas como Matplotlib y Seaborn:

Desviación

Barra divergente

Un gráfico de barras estándar simple que puede manejar valores de magnitud tanto negativos como positivos.

Tabla de columna

Divide un valor único en 2 componentes contrastables (p. ej., masculino/ femenino).

Correlación

Gráfico de dispersión

La forma estándar de mostrar la relación entre dos variables continuas, cada una de las cuales tiene su propio eje.

Línea de tiempo

Una buena manera de mostrar la relación entre una cantidad (columnas) y un ratio (línea).

Clasificación

Gráfico de barras

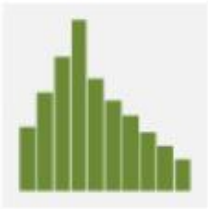
Este tipo de visualizaciones permiten mostrar los rangos de valores de forma sencilla cuando se ordenan.

Diagrama de tira de puntos

Los puntos están ordenados en una tira. Esta distribución ahorra espacio para diseñar rangos en múltiples categorías.

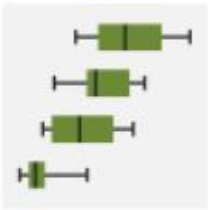
Distribución

Histograma



Es la forma más habitual de mostrar una distribución estadística. Para desarrollarlo se recomienda mantener un pequeño espacio entre las columnas para, así, resaltar la "forma" de los datos.

Gráfico de velas (o cajas)



Eficaz para visualizar distribuciones múltiples mostrando la mediana (centro) y el rango de los datos.

Visualizaciones – Tipos de Gráficos

Pandas proporciona métodos de visualización básicos que permiten crear gráficos directamente desde un DataFrame. Estos métodos son convenientes para realizar exploración visual de los datos antes de realizar análisis más detallados con bibliotecas de visualización más avanzadas como Matplotlib y Seaborn:

Cambios en el tiempo

Línea

Es la forma estándar para mostrar una serie temporal cambiante. Si los datos son muy irregulares puede ser útil emplear marcadores que ayuden a representar puntos de datos.

Mapa de calor calendario

Sirve para mostrar patrones temporales (diario, semanal, mensual). Es necesario ser muy precisos con la cantidad de datos.

Magnitud

Columnas

La forma estándar de mostrar la relación entre dos variables continuas, cada una de las cuales tiene su propio eje.

Gráfico de Marimekko

Ideal para mostrar el tamaño y la proporción de los datos al mismo tiempo, y siempre y cuando, los datos no sean muy complejos.

Parte de un todo

Gráfico de tarta

Uno de los gráficos más comunes para mostrar datos parciales o completos. Conviene tener presente que no es fácil comparar de forma precisa el tamaño de los distintos segmentos.

Diagrama de Venn

Limitado a representaciones esquemáticas que permiten mostrar interrelaciones o coincidencias.

Espacial

Mapa coroplético

Se trata del enfoque estándar para colocar datos en un mapa.

Mapa de flujo

Es utilizado para mostrar un movimiento de cualquier tipo dentro de un mismo mapa. Por ejemplo, puede emplearse para representar movimientos migratorios.

Visualizaciones con Python

Dato un DataFrame con la siguiente información (10 elementos):

- Fecha: fechas creadas al azar a partir del 2022-01-01
- Ventas: cantidades creados al azar entre 50 y 200
- Ganancias: creados al azar entre 100 y 500
- Productos: Productos en la venta de la lista (A, B, C)
- Calificación: valores creados al azar entre 0 y 5

Visualizaciones – Tipos de Gráficos

DataFrame con la fuente de datos para los gráficos:

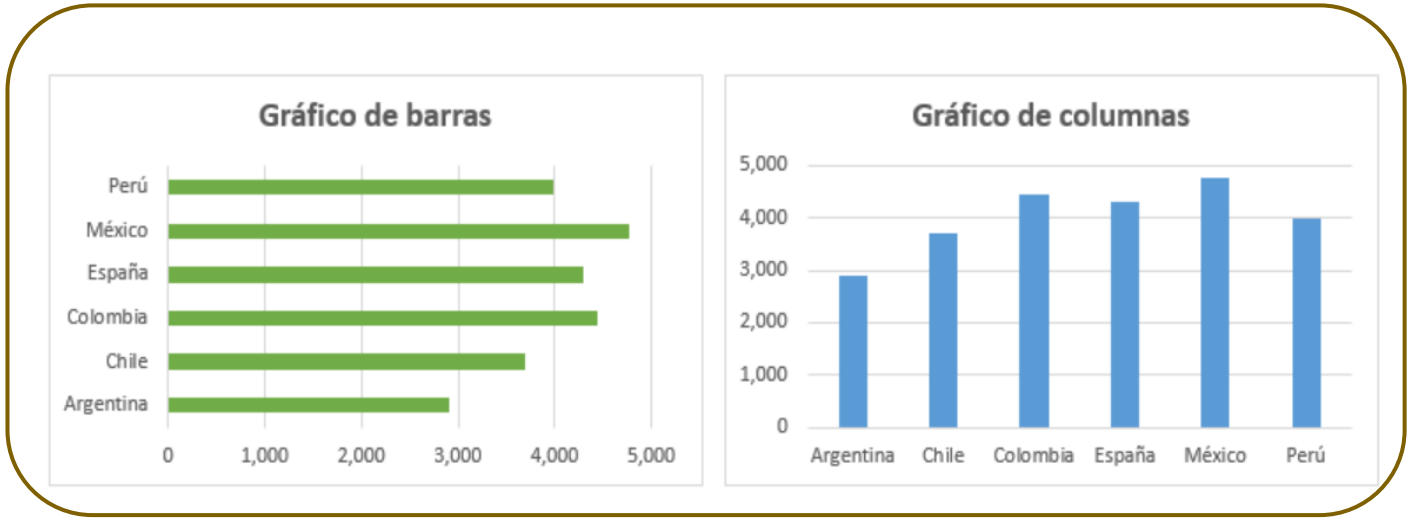
```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


# Crear un Dataframe de ejemplo
data={  'Fecha':pd.date_range(start='2022-01-01',periods=10),
            'Ventas':np.random.randint(50,200,size=10),
            'Ganancias':np.random.uniform(100,500,size=10),
            'Producto':['A','B','A','C','B','C','A','B','C','A'],
            'Calificacion':np.random.randint(1,6,size=10)
         }

df=pd.DataFrame(data)
print("\n Dataframe de ejemplo:")
print(df)
```

```
 Dataframe de ejemplo:
       Fecha  Ventas   Ganancias Producto  Calificacion
0 2022-01-01      52  253.135403        A             2
1 2022-01-02     113  271.287752        B             1
2 2022-01-03     148  181.415663        A             4
3 2022-01-04      55  330.152539        C             5
4 2022-01-05     188  399.085018        B             3
5 2022-01-06     184  422.479619        C             1
6 2022-01-07     154  276.211245        A             2
7 2022-01-08     123  299.683312        B             3
8 2022-01-09     180  149.605819        C             3
9 2022-01-10      65  296.222785        A             3
```

Visualizaciones – Tipos de Gráficos

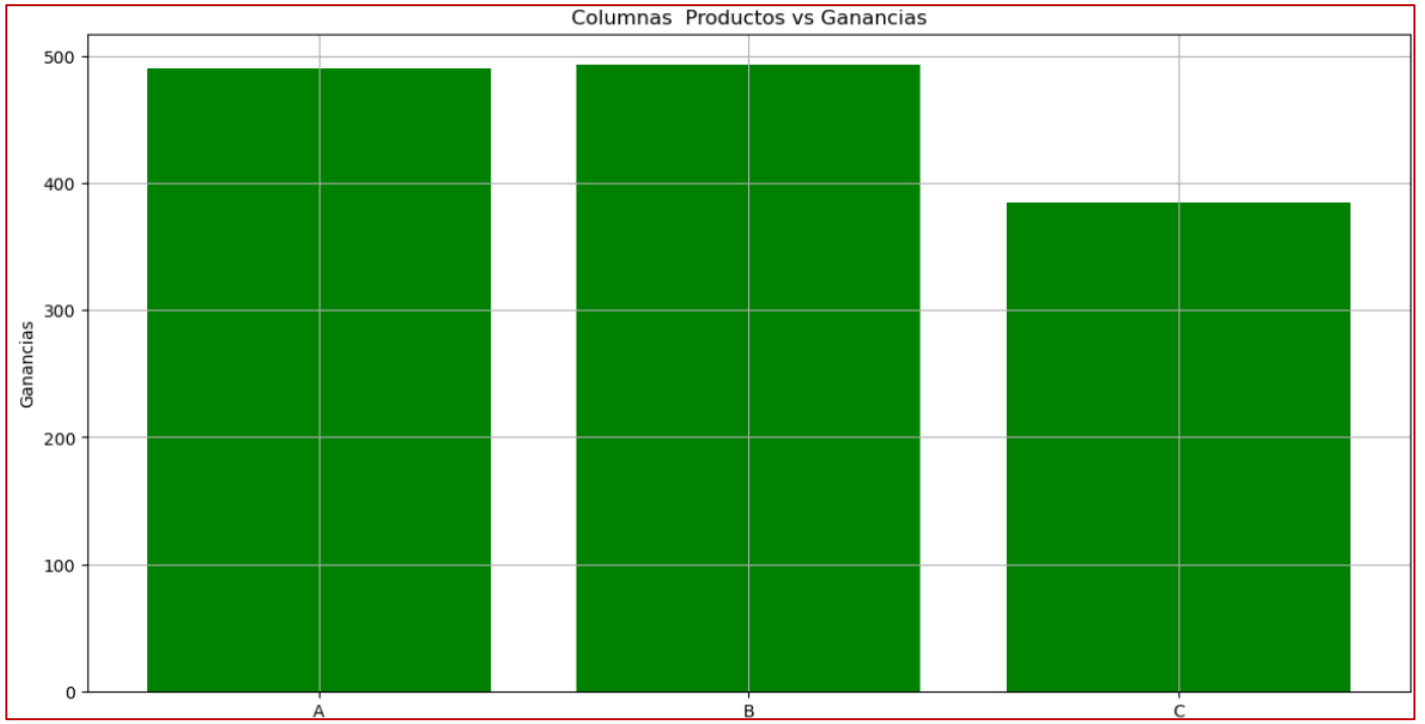
Gráfico de Barras – Gráfico de Columnas: Principios Aplicados:

- Simplicidad al utilizar un formato básico.
- Claridad mediante etiquetas y ejes bien definidos.
- Relevancia al resaltar solo datos esenciales.



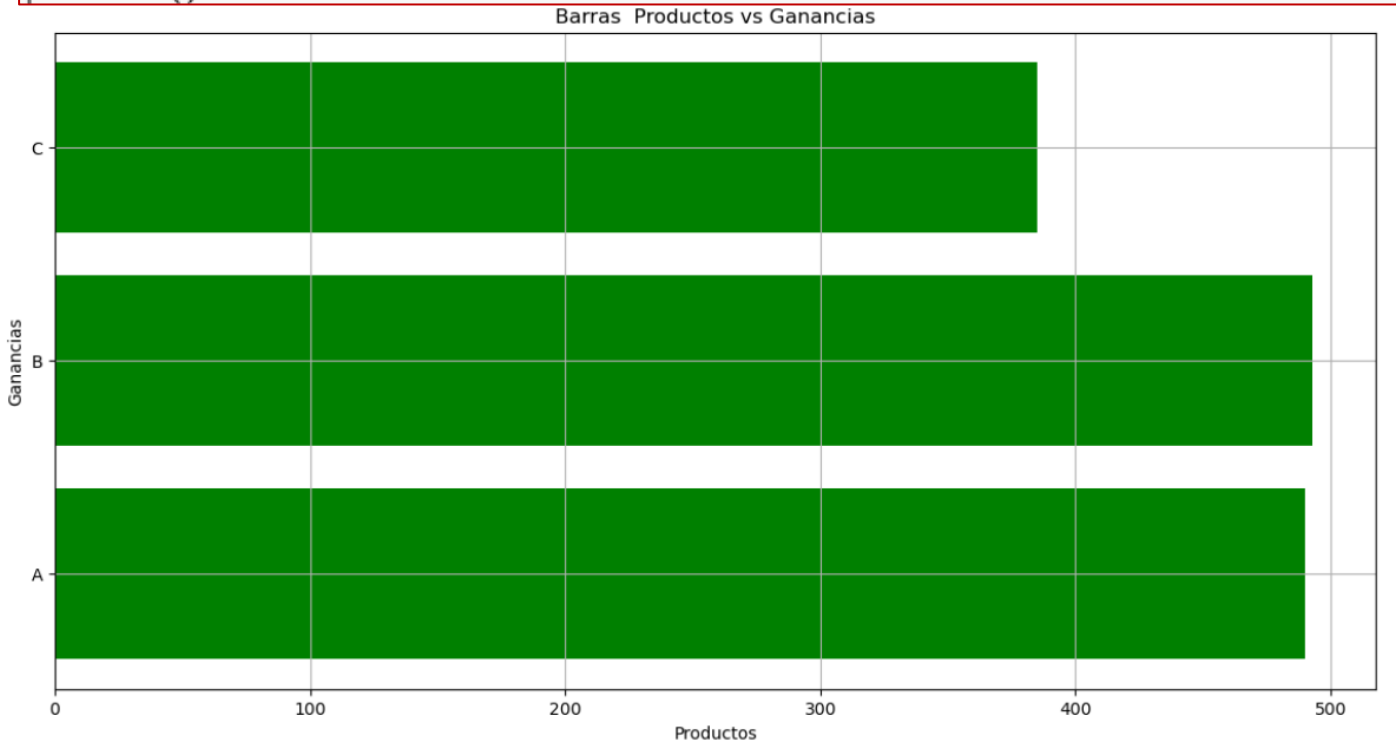
Visualizaciones – Tipos de Gráficos

```
#Crear gráfico de columnas por 'Ganancias'
plt.figure(figsize=(14, 7)) #tupla del ancho y alto en pulgadas
plt.bar(df['Producto'], df['Ganancias'], color='green')
plt.title('Columnas  Productos vs Ganancias')
plt.xlabel('Productos')
plt.ylabel('Ganancias')
plt.grid(True) # Muestra la cuadrícula
plt.savefig('C:\\figuras\\Columnas_Productos_Ganancias.png')  # Guardar la imagen
plt.show()
```



Visualizaciones – Tipos de Gráficos

```
#Crear gráfico de barras por 'Ganancias'
plt.figure(figsize=(14, 7)) #tupla del ancho y alto en pulgadas
plt.barh(df['Producto'], df['Ganancias'], color='green')
plt.title('Barras  Productos vs Ganancias')
plt.xlabel('Productos')
plt.ylabel('Ganancias')
plt.grid(True) # Muestra la cuadrícula
plt.savefig('C:\\figuras\\Barras_Productos_Ganancias.png')  # Guardar la imagen
plt.show()
```

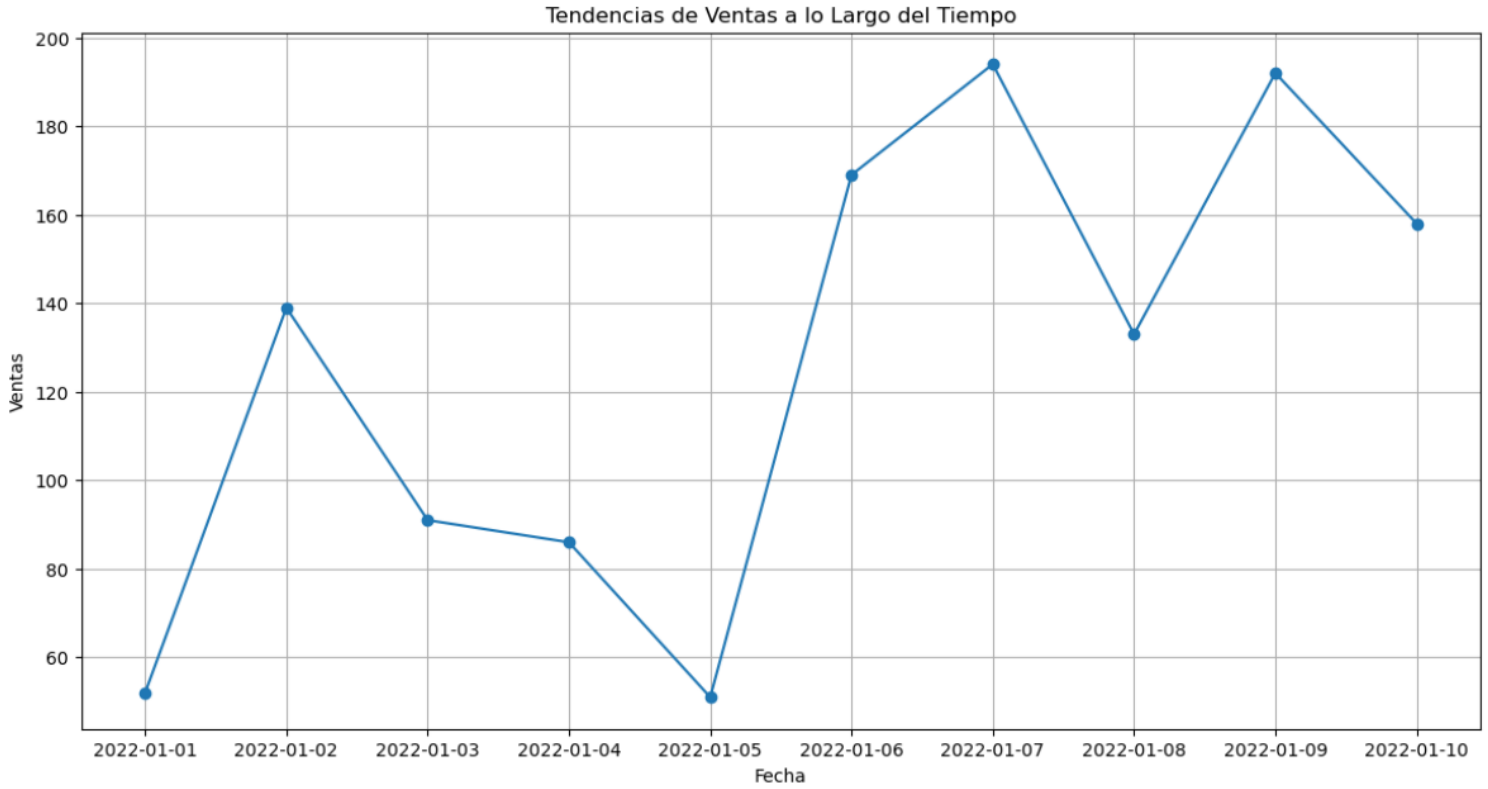


Visualizaciones – Tipos de Gráficos

Gráficos de Líneas: Los gráficos de líneas conectan puntos de datos con líneas, siendo útiles para representar tendencias y cambios en el tiempo.

Visualizaciones – Tipos de Gráficos

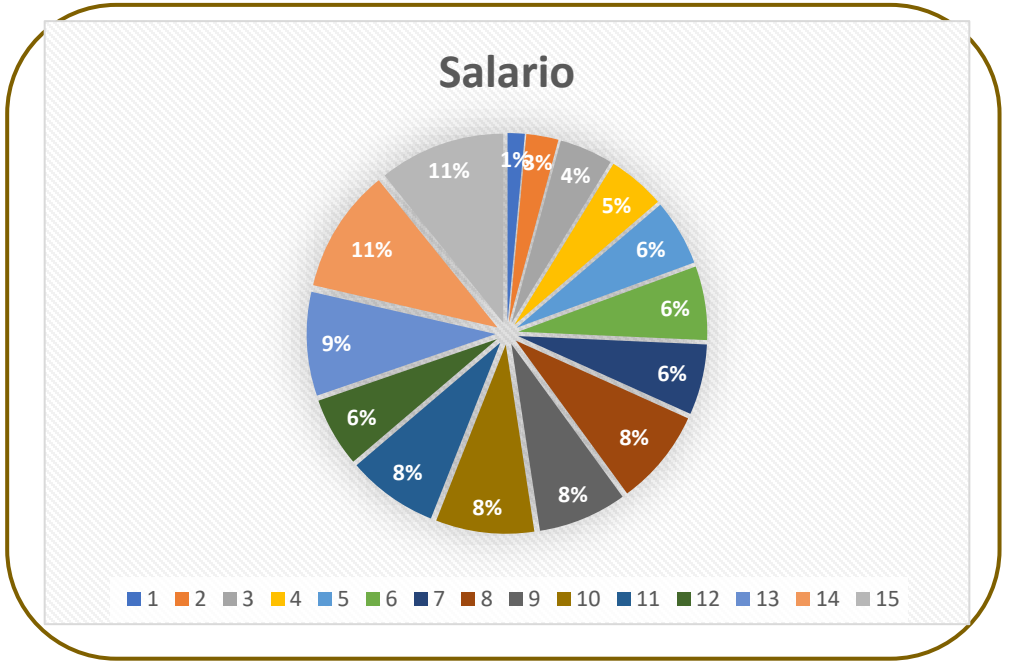
```python
#Crear gráfico de líneas por 'Ventas'
plt.figure(figsize=(14, 7))
plt.plot(df['Fecha'], df['Ventas'], marker='o')
plt.title('Tendencias de Ventas a lo Largo del Tiempo')
plt.xlabel('Fecha')
plt.ylabel('Ventas')
plt.grid(True)
plt.savefig('C:\\figuras\\tendencias_ventas.png')
plt.show()
```



Visualizaciones – Tipos de Gráficos

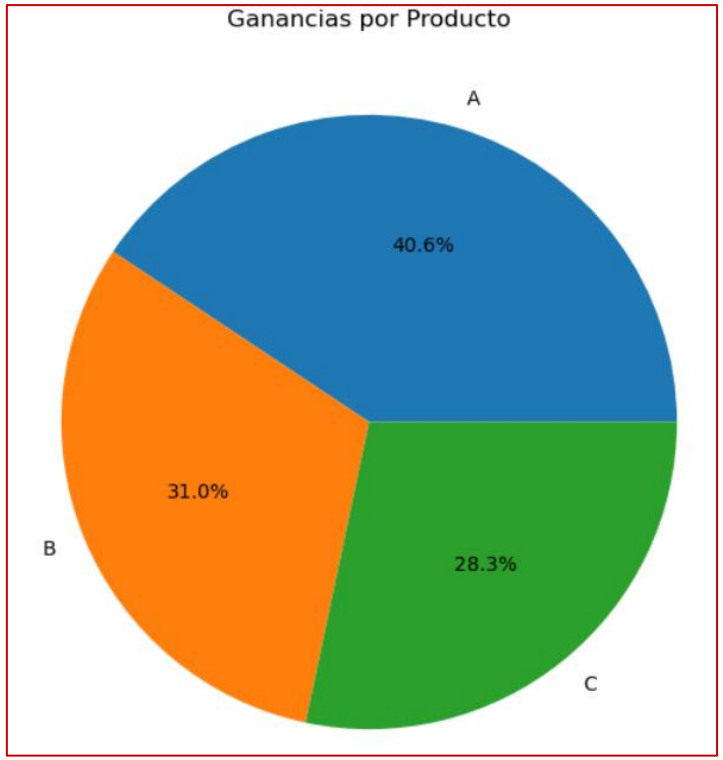
Diagrama de Pastel o Torta- Principios Aplicados:

- Simplicidad al presentar categorías clave.
- Claridad mediante etiquetas y porcentajes.
- Relevancia al destacar solo algunas secciones.



Visualizaciones – Tipos de Gráficos

```
#Crear gráfico de tortas para 'Producto'
#Agrupar por Producto y sumar las Ganancias
ventas_por_producto = df.groupby('Producto')['Ganancias'].sum().reset_index()
plt.figure(figsize=(14, 7))
plt.pie(ventas_por_producto['Ganancias'],labels=ventas_por_producto['Producto'],autopct='%.2f')
plt.title('Ganancias por Producto')
plt.grid(True)
plt.savefig('tortas_GananciasXproducto.png')
plt.show()
```

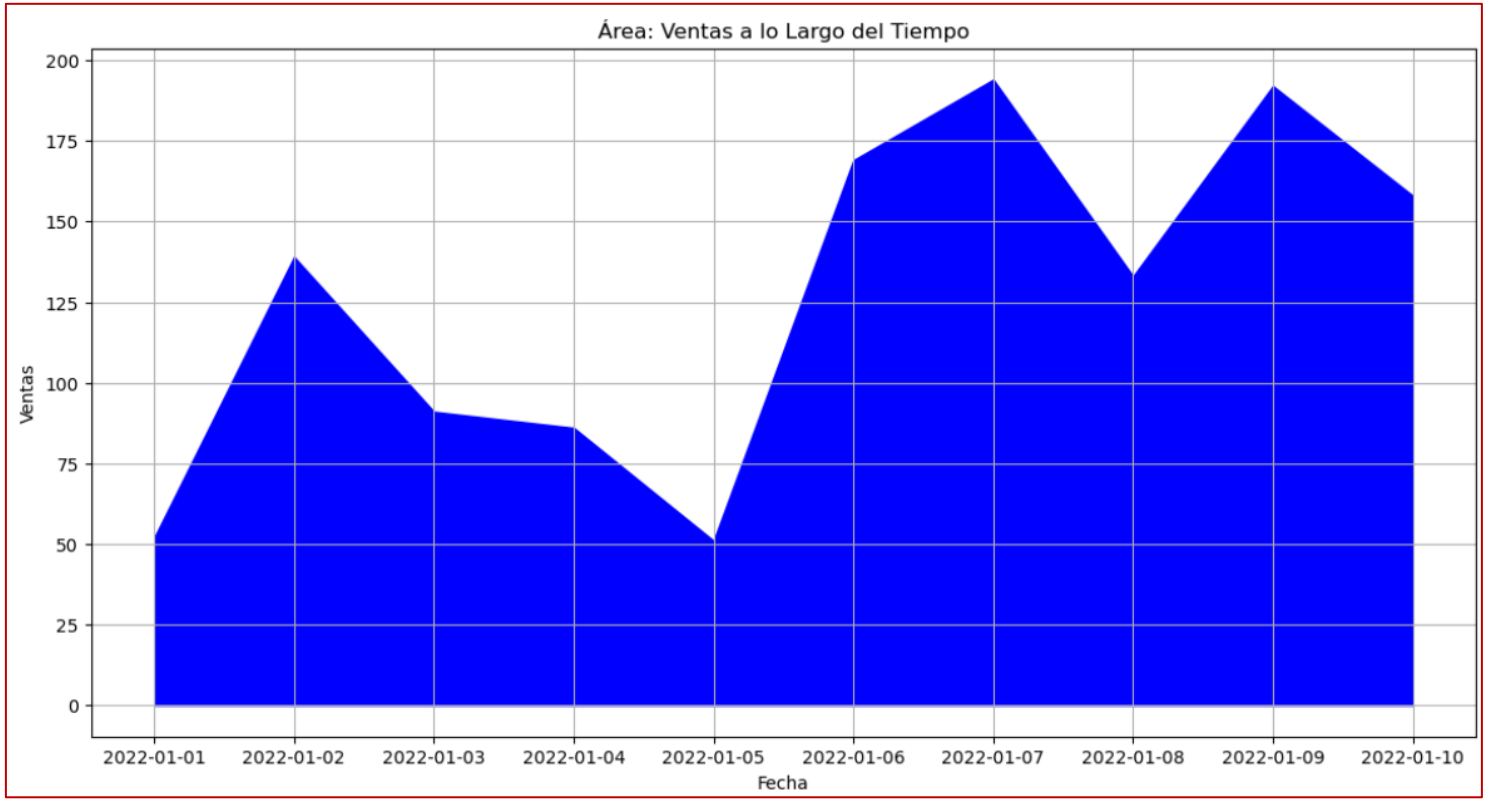


Visualizaciones – Tipos de Gráficos

Gráficos de Área: Los gráficos de área muestran la evolución de valores en un periodo de tiempo y la contribución relativa de cada categoría al total.

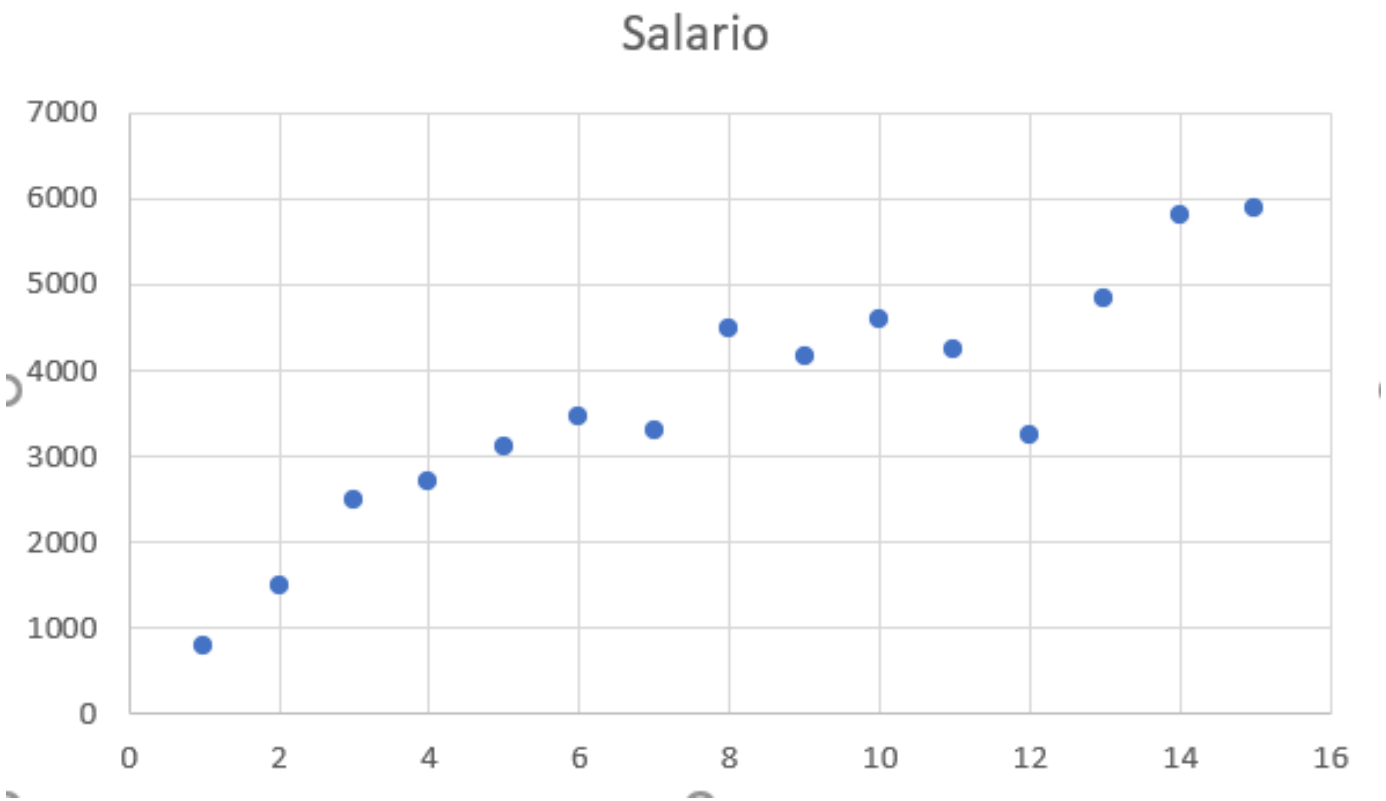
Visualizaciones – Tipos de Gráficos

```python
#Crear gráfico de áreas por 'Ventas'
plt.figure(figsize=(14, 7))
plt.fill_between(df['Fecha'],df['Ventas'],label="Ventas",color="blue")
plt.title('Área: Ventas a lo Largo del Tiempo')
plt.xlabel('Fecha')
plt.ylabel('Ventas')
plt.grid(True)
plt.savefig('C:\\figuras\\area_ventas.png')
plt.show()
```



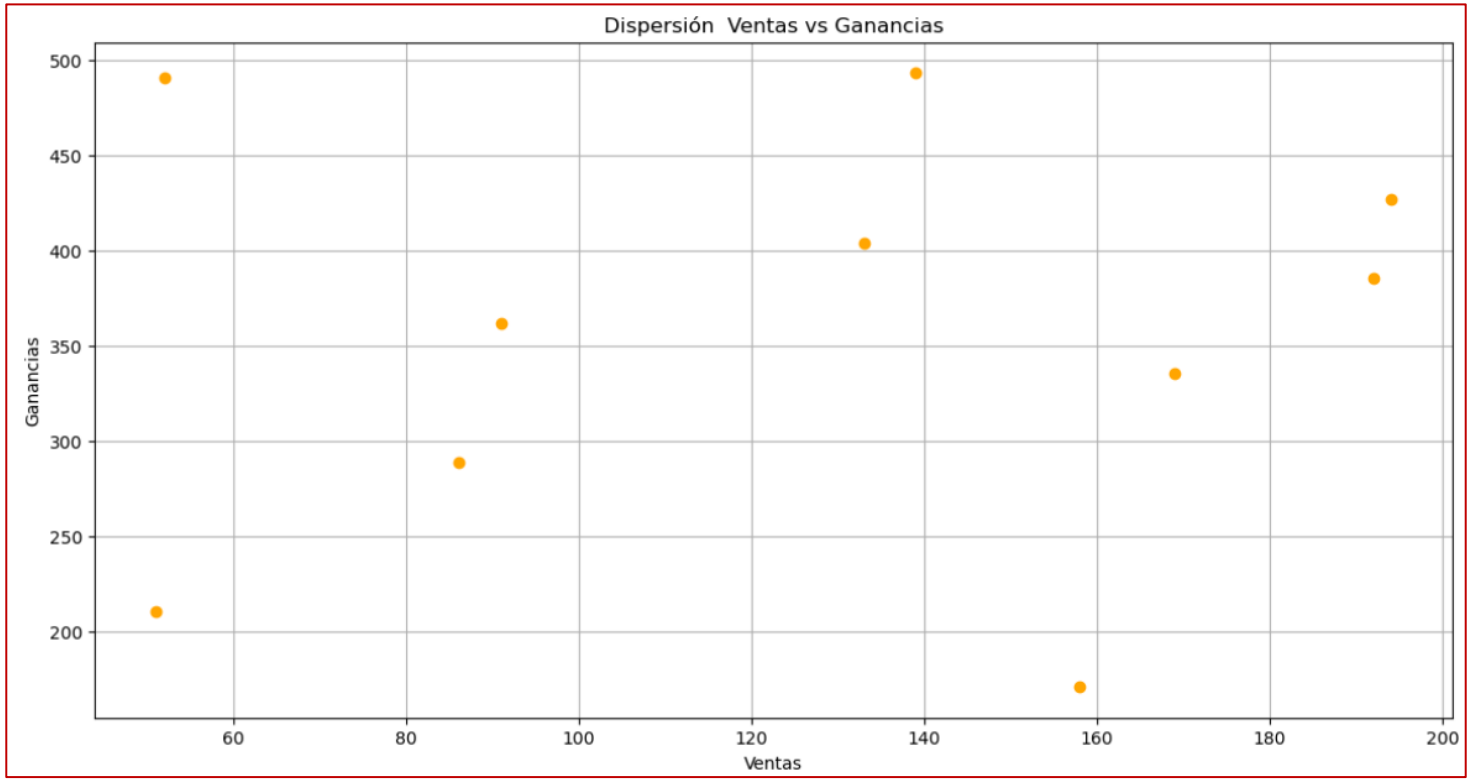
Visualizaciones – Tipos de Gráficos

Gráficos de Dispersión: Los gráficos de dispersión muestran la relación entre dos conjuntos de datos mediante puntos en un plano. Son ideales para identificar patrones, correlaciones y outliers.



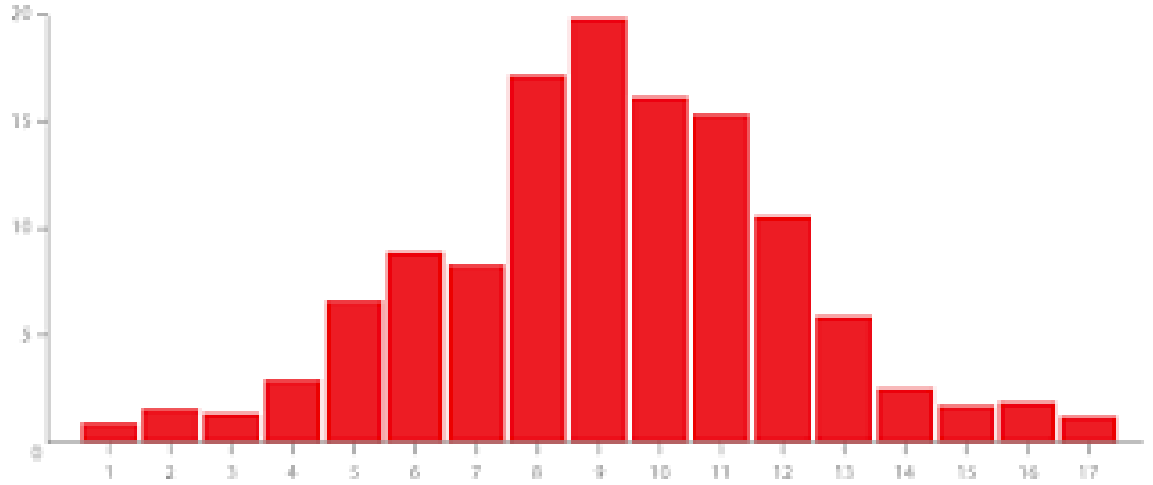
Visualizaciones – Tipos de Gráficos

```
#Crear gráfico de dispersión para 'Ventas'  vs 'Ganancias'
plt.figure(figsize=(14, 7))
plt.scatter(df['Ventas'],df['Ganancias'],color='orange')
plt.title('Dispersión  Ventas vs Ganancias')
plt.xlabel('Ventas')
plt.ylabel('Ganancias')
plt.grid(True)
plt.savefig('dispersion_ventasGanancias.png')
plt.show()
```



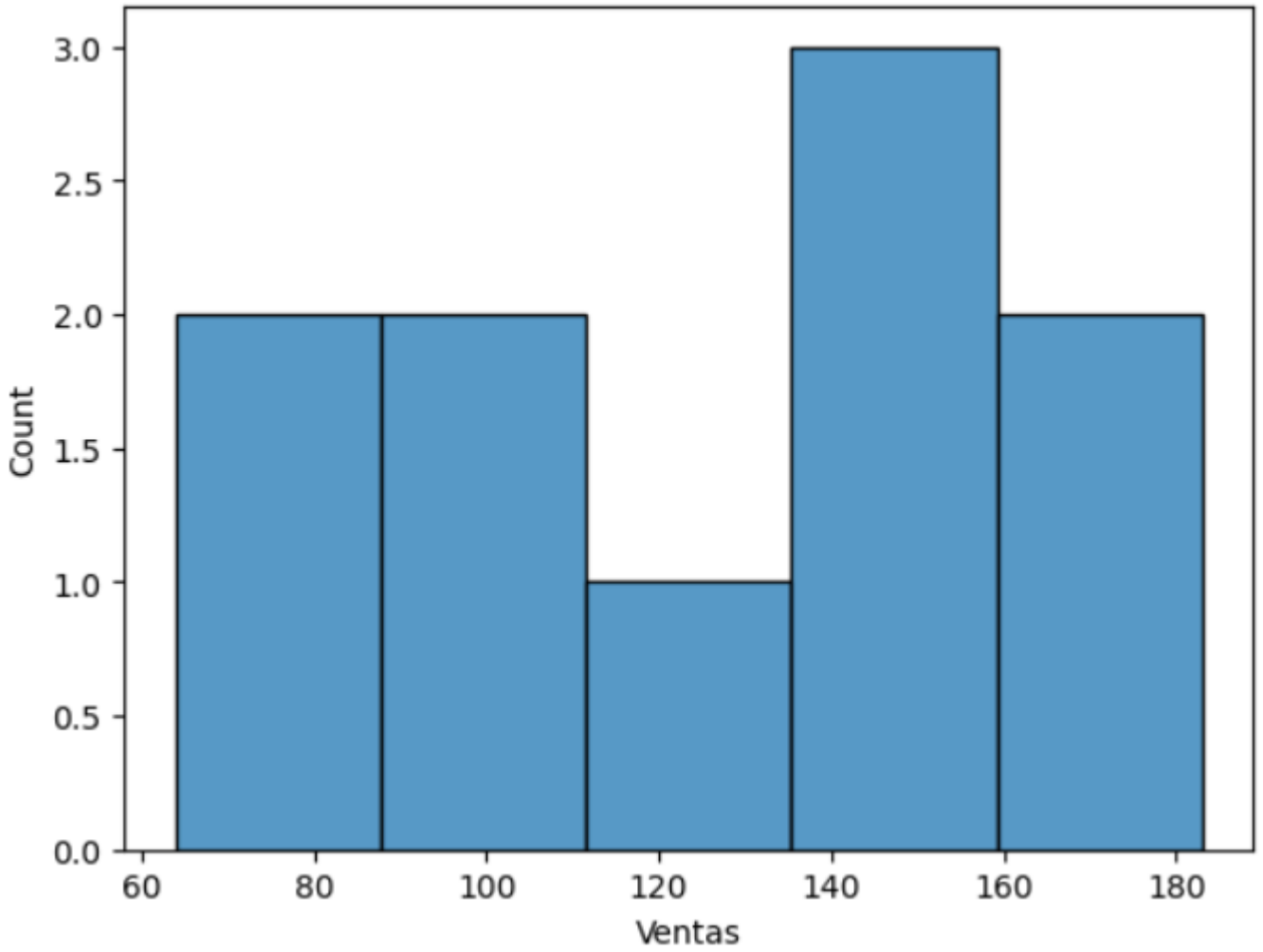
Visualizaciones – Tipos de Gráficos

Histograma: es la representación gráfica en forma de barras, que simboliza la distribución de un conjunto de datos. Sirven para obtener una "primera vista" general, o panorama, de la distribución de la población, o de la muestra, respecto a una característica, cuantitativa y continua..



Visualizaciones – Tipos de Gráficos

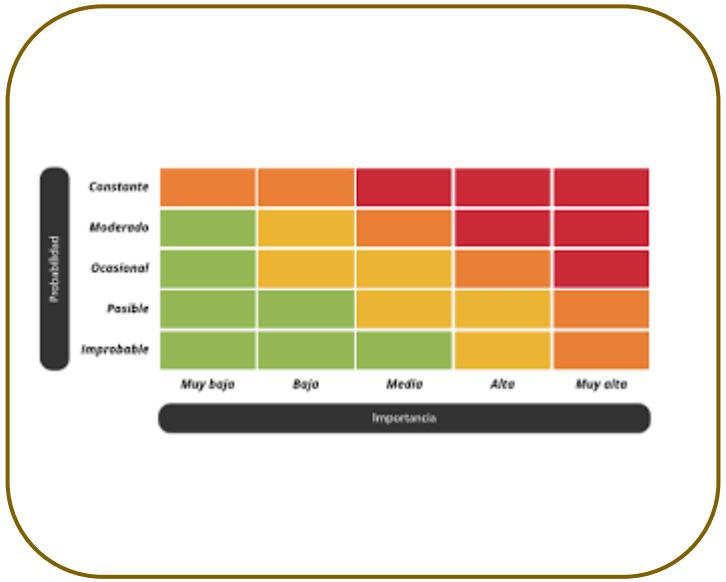
```
#Crear Histograma  para 'Ventas'
sns.histplot(data=df, x="Ventas")
```



Visualizaciones – Tipos de Gráficos

Diagrama de Calor - Principios Aplicados:

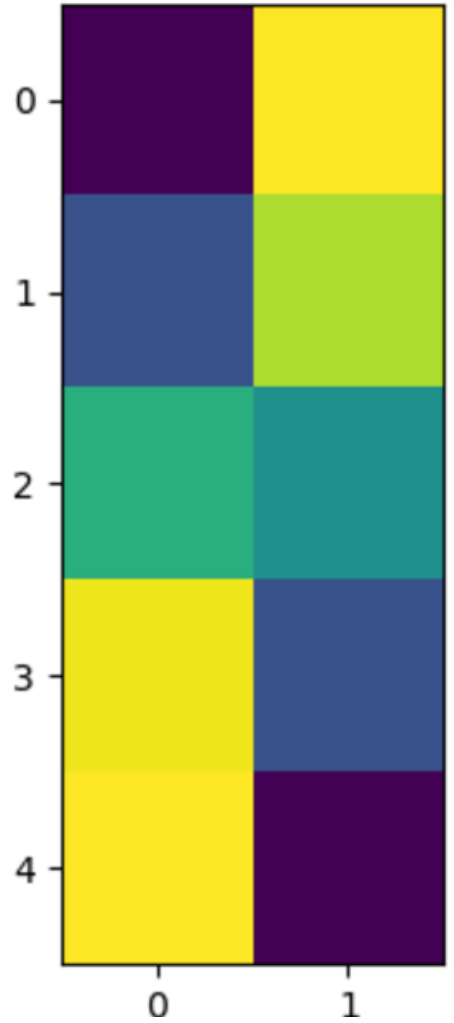
- Simplicidad en la representación de datos de manera visual.
- Claridad mediante la variación de colores.
- Relevancia al mostrar patrones significativos.



Visualizaciones – Tipos de Gráficos

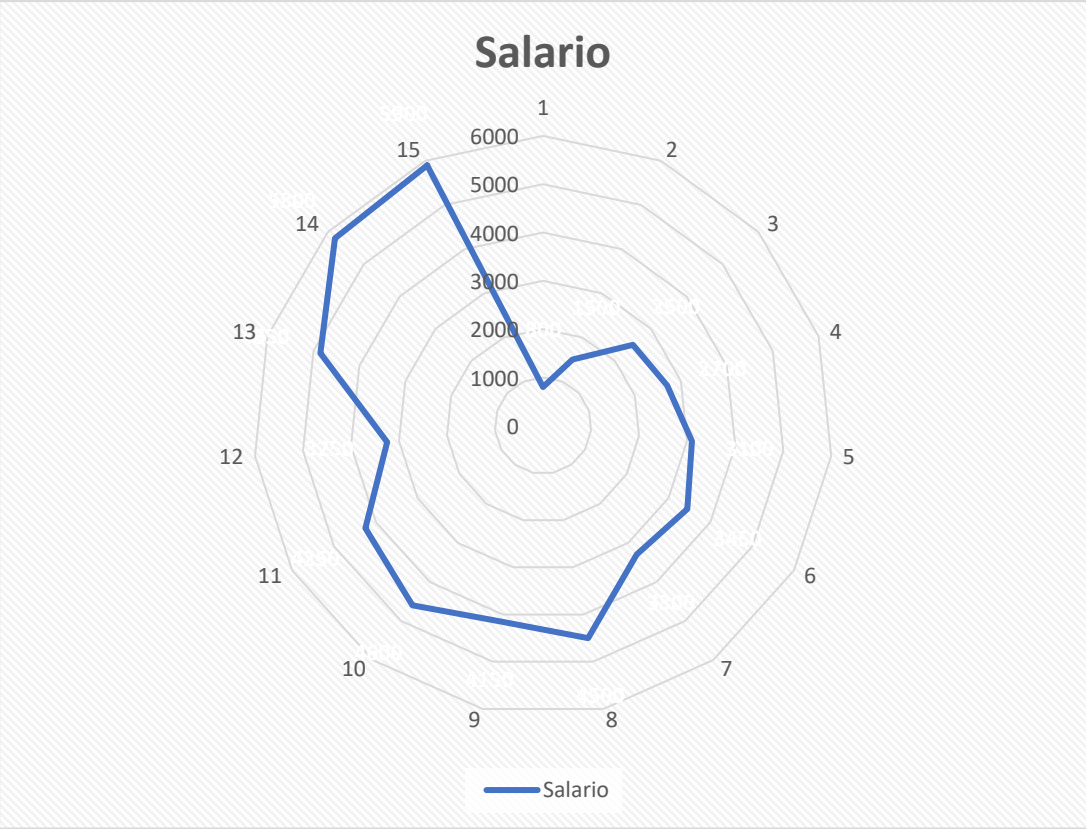
```
#Diagrama de Calor
import matplotlib.pyplot as plt1
df1=pd.DataFrame(
    {
        "Ventas":[100,200,350,490,500],
        "Ganancias":[500,450,300,200,100]
    }
)
print(df1)
plt1.imshow(df1)
```

	Ventas	Ganancias
0	100	500
1	200	450
2	350	300
3	490	200
4	500	100



Visualizaciones – Tipos de Gráficos

Gráficos de Radar (araña o radial): Los gráficos de radar representan datos en un formato circular, permitiendo la comparación de múltiples series de datos.

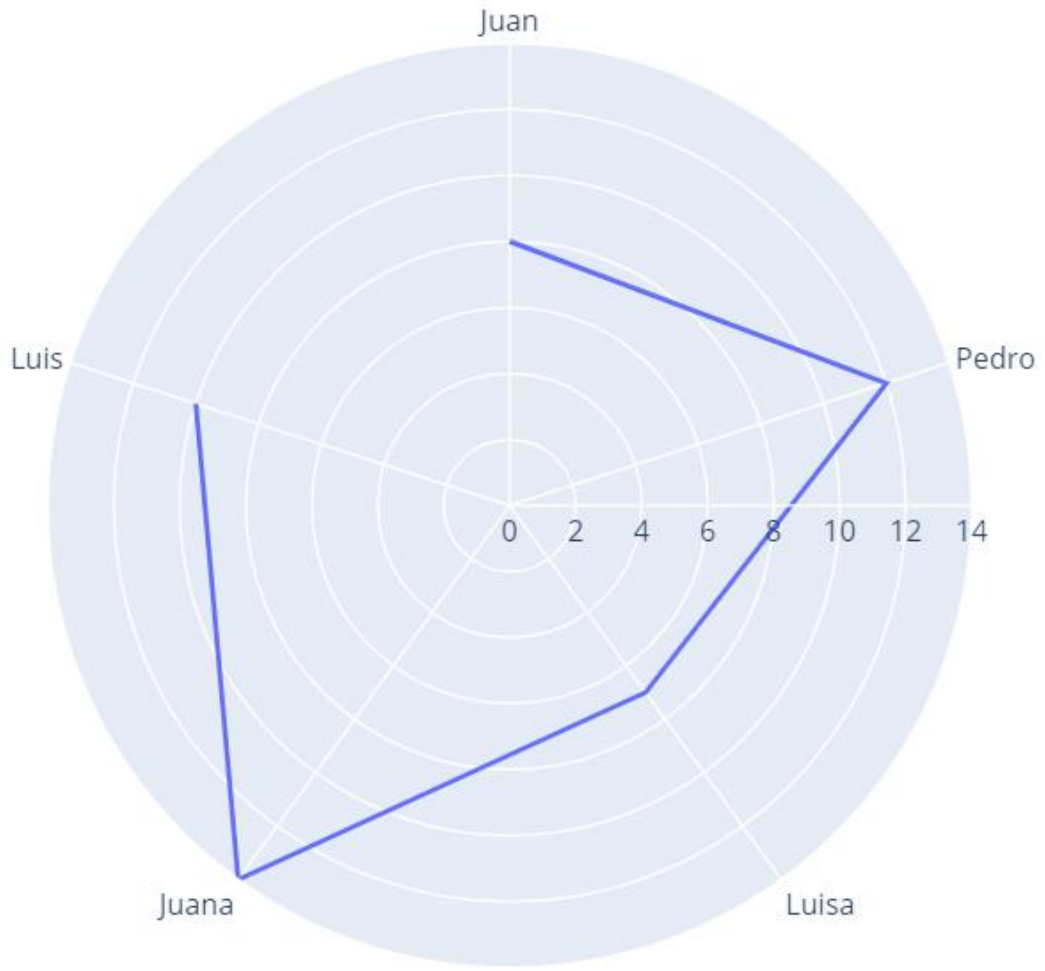


Cuanto más lejos esté la línea o punto del centro del gráfico, mayor será la puntuación del salario en ese aspecto particular.

Visualizaciones – Tipos de Gráficos

```python
# Gráfica de Radar o Araña
import plotly.express as px
import pandas as pd

# Datos de muestra
df = pd.DataFrame({ "puntaje": [8, 12, 7, 14, 10],
                    "nombre" : ['Juan', 'Pedro', 'Luisa', 'Juana', 'Luis']})
print("\n DataFrame de prueba")
print(df)
fig = px.line_polar(df, r = 'puntaje', theta = 'nombre')
fig.show()
```

```
 DataFrame de prueba
   puntaje nombre
0        8   Juan
1       12  Pedro
2        7  Luisa
3       14  Juana
4       10   Luis
```




ADVANCE
Consortium
GRACIAS