

Visualización de datos con Python

Elmer Achalma

Este documento presenta una colección de ejemplos prácticos de visualización de datos utilizando Python y la biblioteca Matplotlib. Se incluyen gráficos de líneas, barras, histogramas, circulares, de caja y combinados, cada uno acompañado de código comentado y optimizado. El objetivo es proporcionar una guía educativa para estudiantes y profesionales interesados en representar datos de manera clara y efectiva, con énfasis en buenas prácticas de diseño y presentación.

Palabras Claves: Visualización de datos, Python, Matplotlib, Gráficos estadísticos, Análisis de datos

Tabla de contenidos		6	Grafico de cajas	6
Introduction	2	7	Grafico de barras combinadas	7
1 Grafico de lineas	2	8	Graficos combinados	8
2 Gráfico de barras	3			
2.1 horizontal	3			
2.2 vertical	4			
3 Histograma	4			
4 Grafico circular	5			
5 Grafico de Donut	5			

 [Elmer Achalma](#)

Los autores no tienen conflictos de intereses que revelar. Agradezco a mis maestros por su orientación en el aprendizaje de la programación y visualización de datos, a mis padres por su apoyo constante durante mi formación, y a Dios por la salud y fortaleza para perseguir mis metas. Me comprometo a seguir mejorando mis habilidades en la universidad y en todos los ámbitos, siempre con respeto.

La correspondencia relativa a este artículo debe dirigirse a

Visualización de datos con Python

1 Grafico de lineas

```
# Mostrar el gráfico
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para mejor visualización
plt.figure(figsize=(10, 8))

# Datos de consumo de carne bovina (kg por habitante)
consumo_bovino = [22.1, 22.1, 23.1, 23.9, 24
anios_bovino = [2001, 2002, 2003, 2004, 2005

# Datos de consumo de carne porcina (kg por
consumo_porcino = [17.9, 19.4, 19.1, 18.3, 1
anios_porcino = [2001, 2002, 2003, 2004, 200

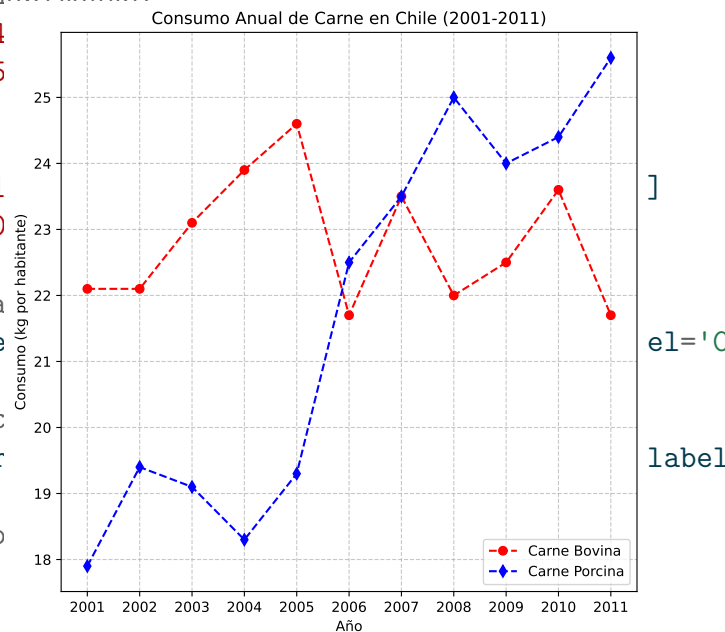
# Graficar consumo de carne bovina con marca
plt.plot(anios_bovino, consumo_bovino, marke

# Graficar consumo de carne porcina con marc
plt.plot(anios_porcino, consumo_porcino, mar

# Etiquetas de los ejes y título con formato
plt.xlabel('Año')
plt.ylabel('Consumo (kg por habitante)')
plt.title('Consumo Anual de Carne en Chile (2001-2011)')

# Añadir leyenda en la esquina inferior derecha
plt.legend(loc='lower right')

# Configurar marcas en el eje x para mostrar cada año
plt.xticks(anios_bovino)
```



```
# Añadir una cuadrícula para mejorar la legibilidad
plt.grid(True, linestyle='--', alpha=0.7)

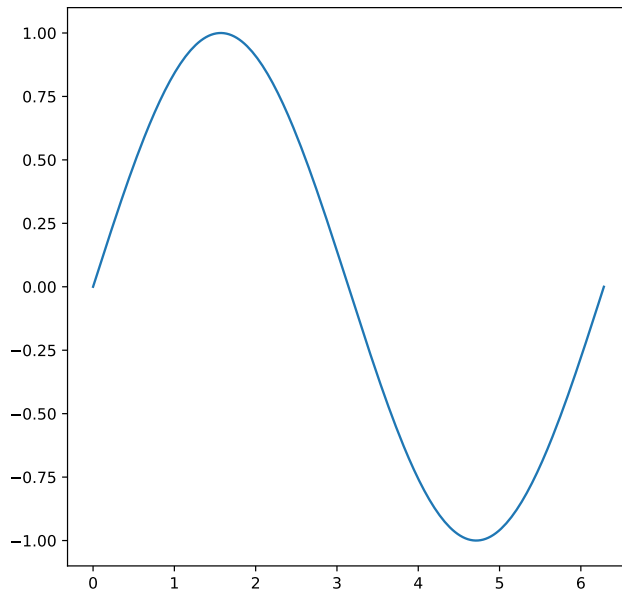
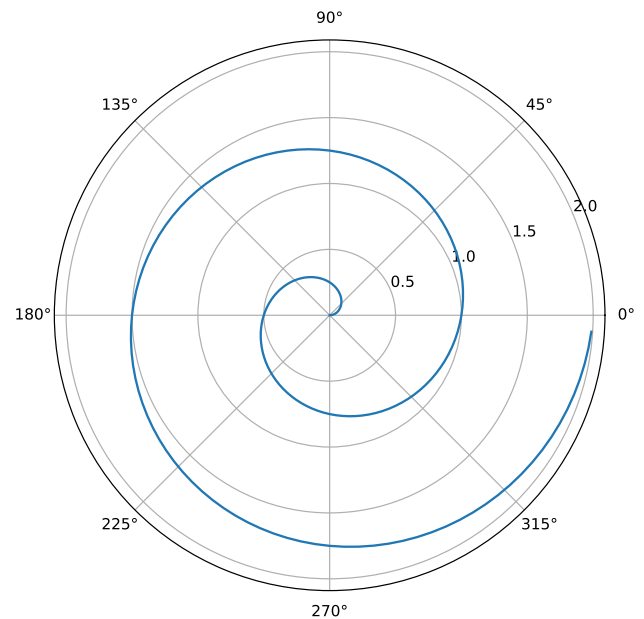
# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional)
plt.savefig('consumo_carne_chile.png')

import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2 * np.pi, 200)
y = np.sin(x)

fig, ax = plt.subplots()
ax.plot(x, y)
plt.show()
```

**Figura 1***A line plot on a polar axis*

2 Gráfico de barras

2.1 horizontal

```
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
    subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

```
import matplotlib.pyplot as plt
```

```
# Configurar el tamaño de la figura para una m
#plt.figure(figsize=(10, 8))
```

```
# Tipos de legumbres y sus respectivos consumo
tipos_legumbres = ["Poroto", "Lenteja", "Garba
consumo_legumbres = [2.1, 1.0, 0.3, 0.5]
```

```
# Crear un gráfico de barras con color persona
plt.bar(tipos_legumbres, consumo_legumbres, co
```

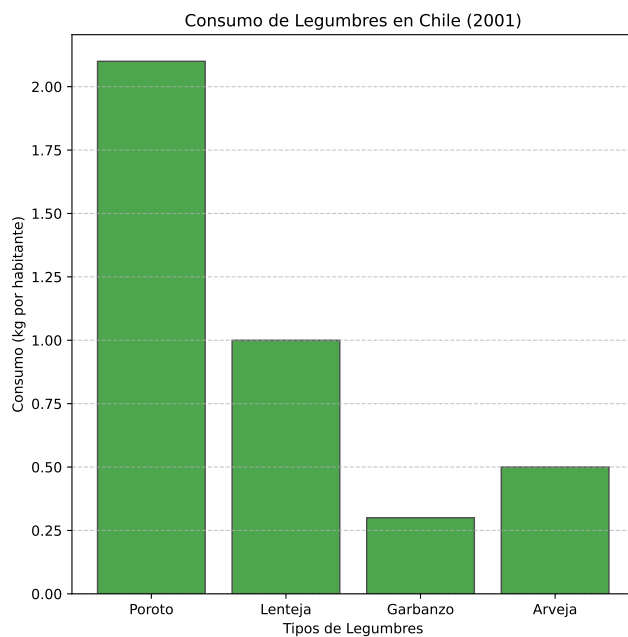
```
# Etiquetas de los ejes y título con formato a
plt.xlabel('Tipos de Legumbres')
plt.ylabel('Consumo (kg por habitante)')
plt.title('Consumo de Legumbres en Chile (2001
```

```
# Añadir una cuadrícula en el eje y para facil
plt.grid(True, axis='y', linestyle='--', alpha
```

```
# Ajustar el diseño para evitar recortes de
plt.tight_layout()

# Guardar la figura en un archivo (opcional,
# plt.savefig('consumo_legumbres_2001.png')

# Mostrar el gráfico
plt.show()
```

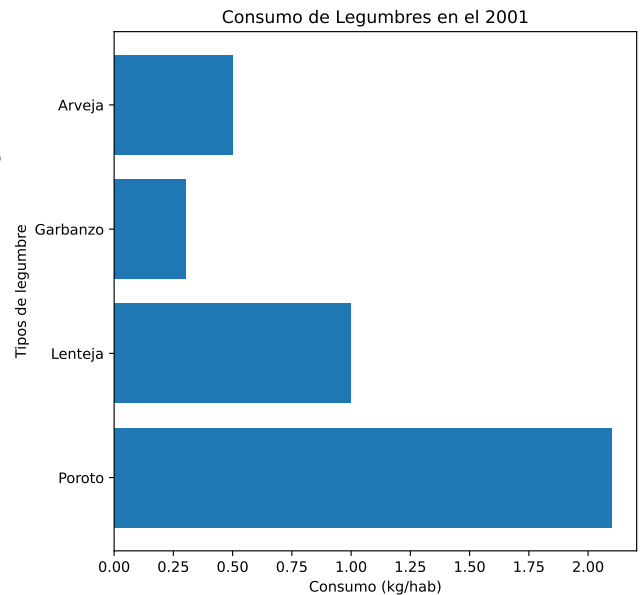


2.2 vertical

```
#plt.figure(figsize=[10,8])
#Consumo de legumbres en el 2001
legumbres=["Poroto","Lenteja","Garbanzo",
consumo=[2.1, 1.0, 0.3, 0.5]

plt.barh(legumbres,consumo)

plt.ylabel("Tipos de legumbre")
plt.xlabel("Consumo (kg/hab)")
plt.title("Consumo de Legumbres en el 2001")
plt.show()
```



3 Histograma

```
import matplotlib.pyplot as plt
```

```
# Configurar el tamaño de la figura para una m
#plt.figure(figsize=(10, 8))
```

```
# Datos de niveles de glucosa (mg/dl)
niveles_glucosa = [52, 54, 55, 57, 56, 57, 54,
76, 79, 81, 85, 88, 84, 89,
```

```
# Bordes de los intervalos para el histograma
intervalos = [50, 60, 70, 80, 90, 100, 110, 120]
```

```
# Crear el histograma con color personalizado
plt.hist(niveles_glucosa, bins=intervalos, col
```

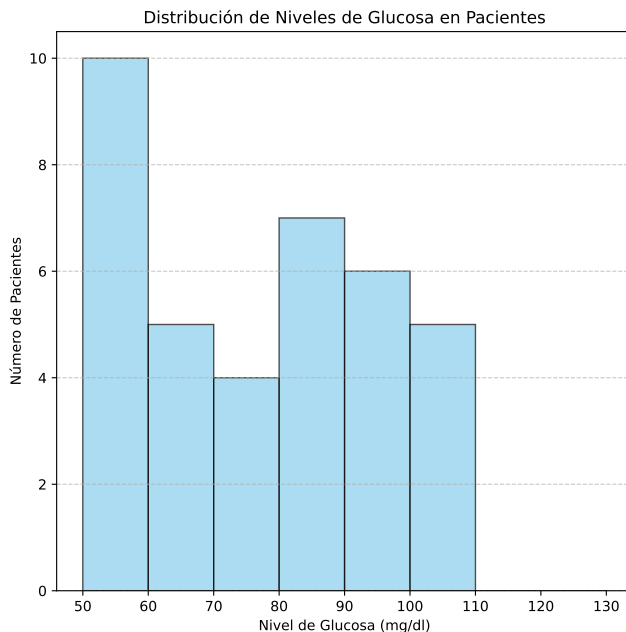
```
# Etiquetas de los ejes y título con formato a
plt.xlabel('Nivel de Glucosa (mg/dl)')
plt.ylabel('Número de Pacientes')
plt.title('Distribución de Niveles de Glucosa
```

```
# Añadir una cuadrícula en el eje y para facil
plt.grid(True, axis='y', linestyle='--', alpha
```

```
# Ajustar el diseño para evitar recortes de et
plt.tight_layout()
```

```
# Guardar la figura en un archivo (opcional)
# plt.savefig('distribucion_glucosa.png')

# Mostrar el gráfico
plt.show()
```



4 Grafico circular

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una
# plt.figure(figsize=(10, 8))

# Datos de marcas de autos y sus ventas (en
# marcas_autos = ["Kia", "Toyota", "Nissan", "
# ventas = [10.5, 15.3, 14.2, 16.1, 9.8]
# Resaltar la primera marca (Kia) ligeramente
# resaltar = [0.1, 0, 0, 0, 0]

# Aplicar un estilo visual predefinido (ggplot)
plt.style.use("ggplot")

# Crear el gráfico de pastel
plt.pie(x=ventas, explode=resaltar, labels=
```

```
# Asegurar que el gráfico sea circular
plt.axis("equal")

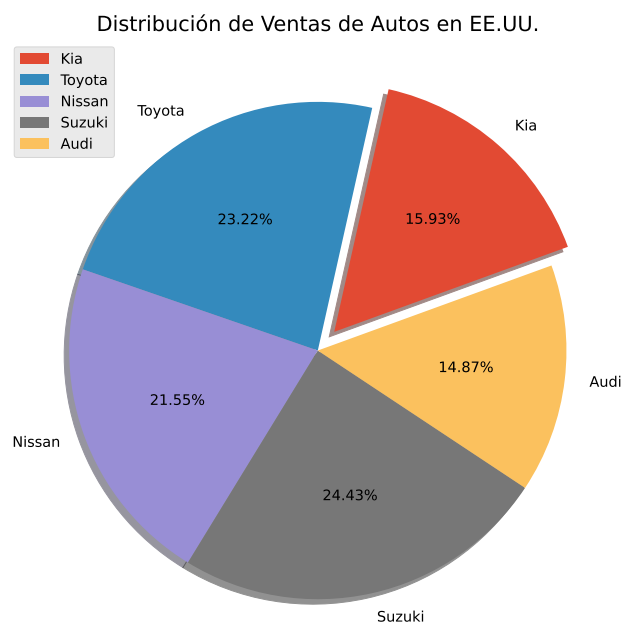
# Añadir un título descriptivo
plt.title("Distribución de Ventas de Autos en EE.UU.")

# Añadir una leyenda en la esquina superior izquierda
plt.legend(marcas_autos, loc="upper left")

# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional, d
# plt.savefig('ventas_autos_eeuu.png')

# Mostrar el gráfico
plt.show()
```



5 Grafico de Donut

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una m
# plt.figure(figsize=(10, 8))

# Crear el gráfico de donut
plt.pie(x=ventas, explode=resaltar, labels=
```

```
# Datos de marcas de autos y sus ventas (en
marcas_autos = ["Kia", "Toyota", "Nissan", "
ventas = [10.5, 15.3, 14.2, 16.1, 9.8]
# Resaltar la primera marca (Kia) ligerament
resaltar = [0.1, 0, 0, 0, 0]
```

```
# Aplicar un estilo visual predefinido (ggplot
plt.style.use("ggplot")
```

```
# Crear el gráfico de pastel (donut chart)
plt.pie(ventas, explode=resaltar, labels=mar
```

```
# Asegurar que el gráfico sea circular
plt.axis("equal")
```

```
# Añadir un título descriptivo
plt.title("Distribución de Ventas de Autos e
```

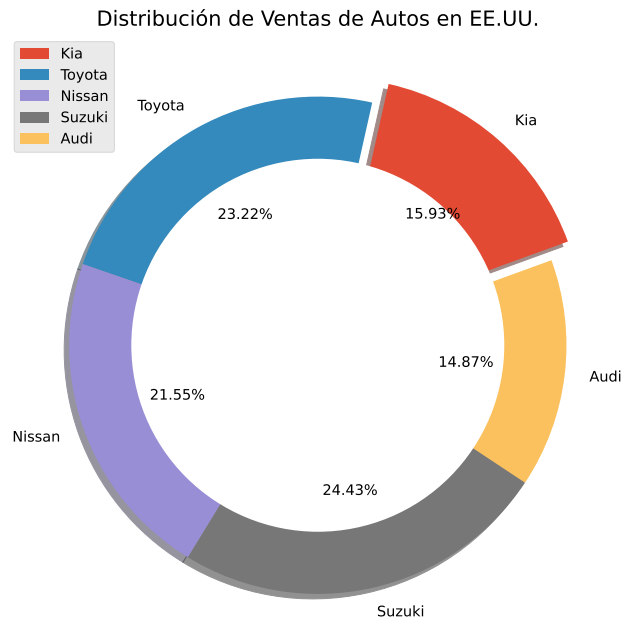
```
# Añadir una leyenda en la esquina superior izquierda
plt.legend(marcas_autos, loc="upper left")
```

```
# Añadir un círculo central para crear el
circulo_central = plt.Circle(xy=(0, 0), ra
plt.gca().add_artist(circulo_central)
```

```
# Ajustar el diseño para evitar recortes
plt.tight_layout()
```

```
# Guardar la figura en un archivo (opciona
# plt.savefig('ventas_autos_eeuu_donut.png
```

```
# Mostrar el gráfico
plt.show()
```



6 Gráfico de cajas

```
import matplotlib.pyplot as plt
```

```
# Configurar el tamaño de la figura para una m
# plt.figure(figsize=(10, 8))
```

```
# Datos de las edades de los alumnos
edades_alumnos = [12, 13, 12, 17, 16, 15, 14,
```

```
# Crear el diagrama de caja con un estilo pers
plt.boxplot(edades_alumnos, vert=True, patch_a
            boxprops=dict(facecolor='lightblue
            medianprops=dict(color='red'), whi
            capprops=dict(color='black'), flie
```

```
# Etiquetas de los ejes y título con formato a
plt.ylabel('Edad (años)')
plt.title('Distribución de Edades de Alumnos d
```

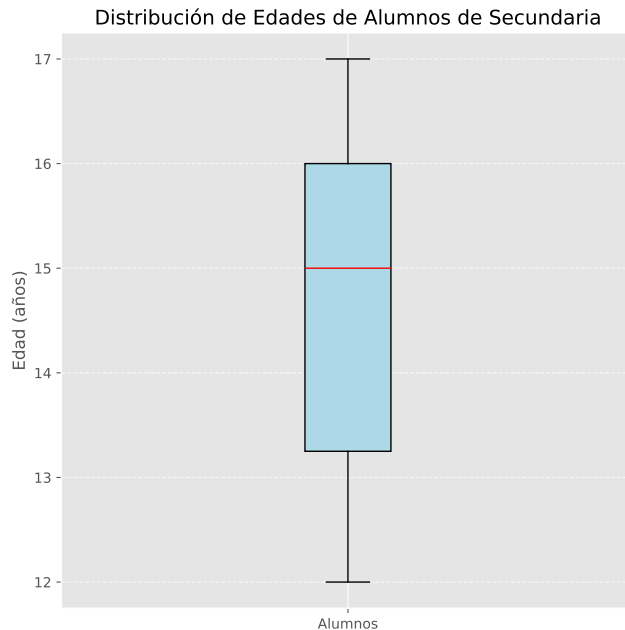
```
# Añadir una cuadrícula en el eje y para facil
plt.grid(True, axis='y', linestyle='--', alpha
```

```
# Configurar las etiquetas del eje x (opcional
plt.xticks([1], ['Alumnos'])
```

```
# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional)
# plt.savefig('edades_alumnos_secundaria.png')

# Mostrar el gráfico
plt.show()
```



```
# Configurar las etiquetas del eje x con los días
plt.xticks(dias, ['Día 1', 'Día 2', 'Día 3', 'Día 4', 'Día 5'])

# Etiquetas de los ejes y título con formato adecuado
plt.xlabel('Días')
plt.ylabel('Unidades Vendidas')
plt.title('Ventas de Autos por Día (Toyota vs. Audi)')

# Configurar las marcas del eje y para mayor claridad
plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50])

# Añadir una cuadrícula en el eje y para facilitar la lectura
plt.grid(True, axis='y', linestyle='--', alpha=0.5)

# Añadir una leyenda para identificar las marcas
plt.legend()

# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional, si se desea)
# plt.savefig('ventas_autos_toyota_audi.png')

# Mostrar el gráfico
plt.show()
```

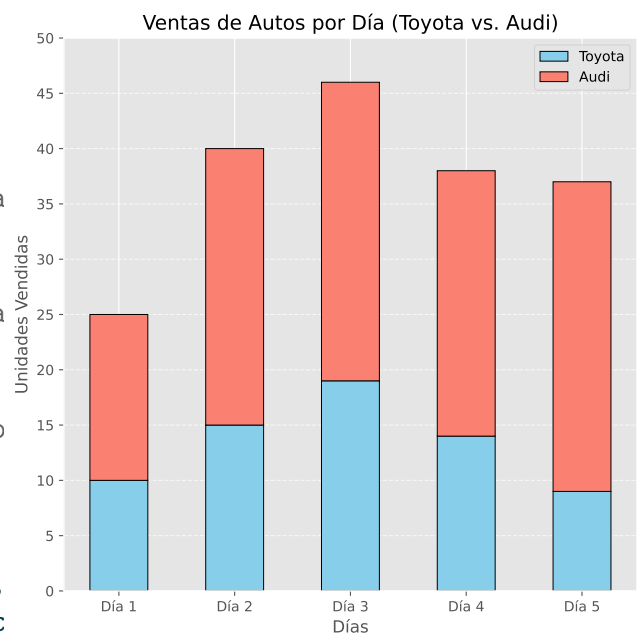
7 Gráfico de barras combinadas

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una mejor visualización
plt.figure(figsize=(10, 8))

# Datos de ventas de autos por día (en unidades)
ventas_toyota = [10, 15, 19, 14, 9]
ventas_audi = [15, 25, 27, 24, 28]
dias = [0, 1, 2, 3, 4] # Posiciones para los días
ancho_barras = 0.5 # Grosor de las barras

# Crear el gráfico de barras apiladas
barras_toyota = plt.bar(dias, ventas_toyota, width=ancho_barras)
barras_audi = plt.bar(dias, ventas_audi, width=ancho_barras)
```



kyblu
'Audi'

8 Graficos combinados

```
import matplotlib.pyplot as plt

# Configurar el tamaño de la figura para una
plt.figure(figsize=(10, 8))

# Datos de vacunación proyectada (en número
vacunacion_proyectada = [250, 120, 270, 560,
# Datos de vacunación real (en número de pac
vacunacion_real = [150, 300, 120, 550, 500,
meses = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio"]

# Graficar la vacunación proyectada como una
plt.plot(meses, vacunacion_proyectada, marke

# Graficar la vacunación real como barras
plt.bar(meses, vacunacion_real, color='skyblue', edgecolor='black', alpha=0.7, label='Va

# Etiquetas de los ejes y título con formato adecuado
plt.xlabel('Meses')
plt.ylabel('Número de Pacientes Vacunados')
plt.title('Vacunación Real vs. Proyectada (Enero - Julio)')

# Añadir una leyenda
plt.legend()

# Añadir una cuadrícula en el eje y para facilitar la lectura
plt.grid(True, axis='y', linestyle='--', alpha=0.7)

# Ajustar el diseño para evitar recortes
plt.tight_layout()

# Guardar la figura en un archivo (opcional, descomentar para usar)
# plt.savefig('vacunacion_real_vs_proyectada.png')

# Mostrar el gráfico
plt.show()
```

