

Introduction

Este trabajo presenta un análisis visual comprehensivo utilizando gráficos estadísticos descriptivos aplicados al dataset "tips". El objetivo principal es demostrar la utilidad de diferentes tipos de visualizaciones para el análisis exploratorio de datos, implementando diez gráficos estadísticos fundamentales con explicaciones detalladas y código replicable.

El dataset "tips" fue seleccionado por su versatilidad analítica, conteniendo variables tanto cualitativas (género, día, fumador, turno) como cuantitativas (factura, propina, tamaño de grupo) que permiten ejemplificar una amplia gama de gráficos estadísticos. Su contexto de restaurante facilita la interpretación y ofrece insights prácticos sobre patrones de consumo y comportamiento de clientes.

GRAFICO BARRAS HORIZONTALES

Definición

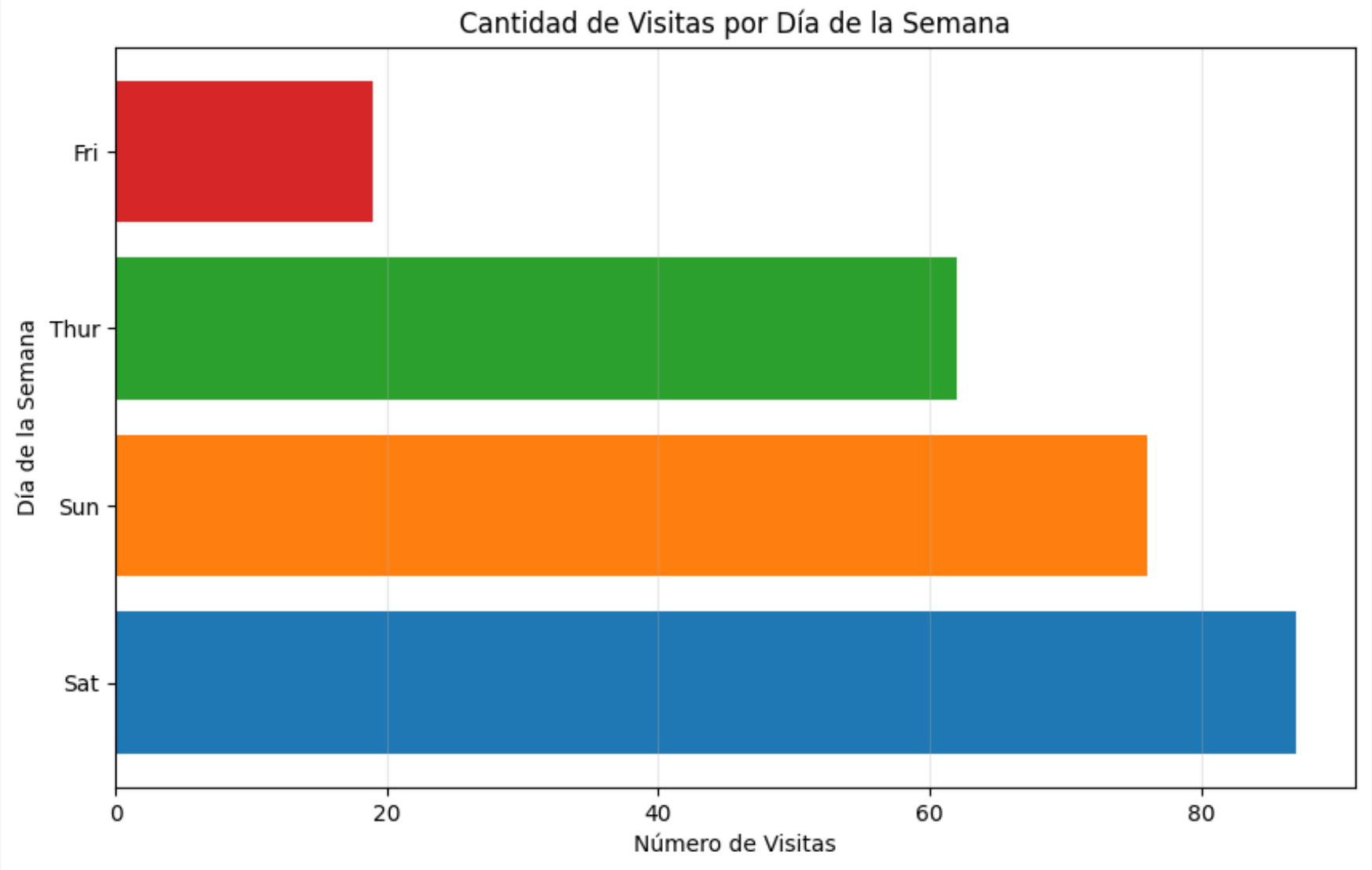
Muestra la frecuencia o cantidad de observaciones para cada categoría de manera horizontal.

Muestra

Categorías en el eje Y y frecuencias en el eje X.

Uso

Se usa con variables cualitativas nominales u ordinales.



Codigo

```
df_tips = sns.load_dataset('tips')plt.figure(figsize = (10, 6))

dia_counts = df_tips['day'].value_counts()
plt.barh(dia_counts.index, dia_counts.values, color = ['1f77b4', 'ff7f0e', '2ca02c', 'd62728'])
plt.title('CantidaddeVisitasporDadelaSemana')
plt.xlabel('NmerodeVisitas')
plt.ylabel('DadelaSemana')
plt.grid(axis = 'x', alpha = 0.3)
plt.show()
```

BARRAS VERTICALES

Definición

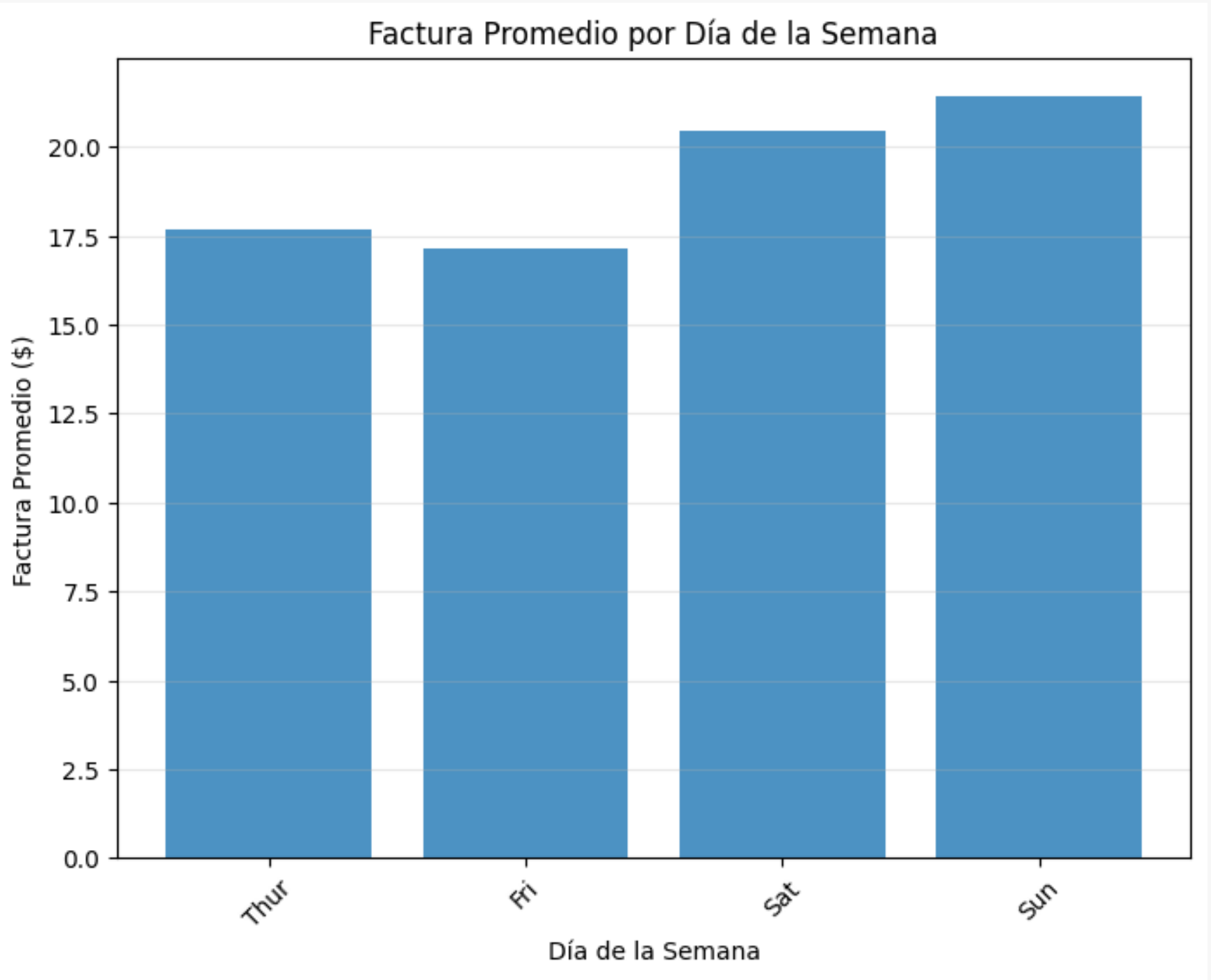
Representa valores numéricos asociados a categorías mediante barras verticales.

Muestra

Categorías en el eje X y valores numéricos en el eje Y.

Uso

Se usa para comparar magnitudes entre diferentes categorías



Codigo

```
plt.figure(figsize=(8, 6))
promedio_por_dia = df_tips.groupby('day')['total_bill'].mean()
plt.bar(promedio_por_dia.index, promedio_por_dia.values, color = '1f77b4', alpha = 0.8)
plt.title('FacturaPromedioporDadelaSemana')
plt.xlabel('DadelaSemana')
plt.ylabel('FacturaPromedio()')
plt.xticks(rotation = 45)
plt.grid(axis = 'y', alpha = 0.3)
plt.show()
```

Gráfico

Definición

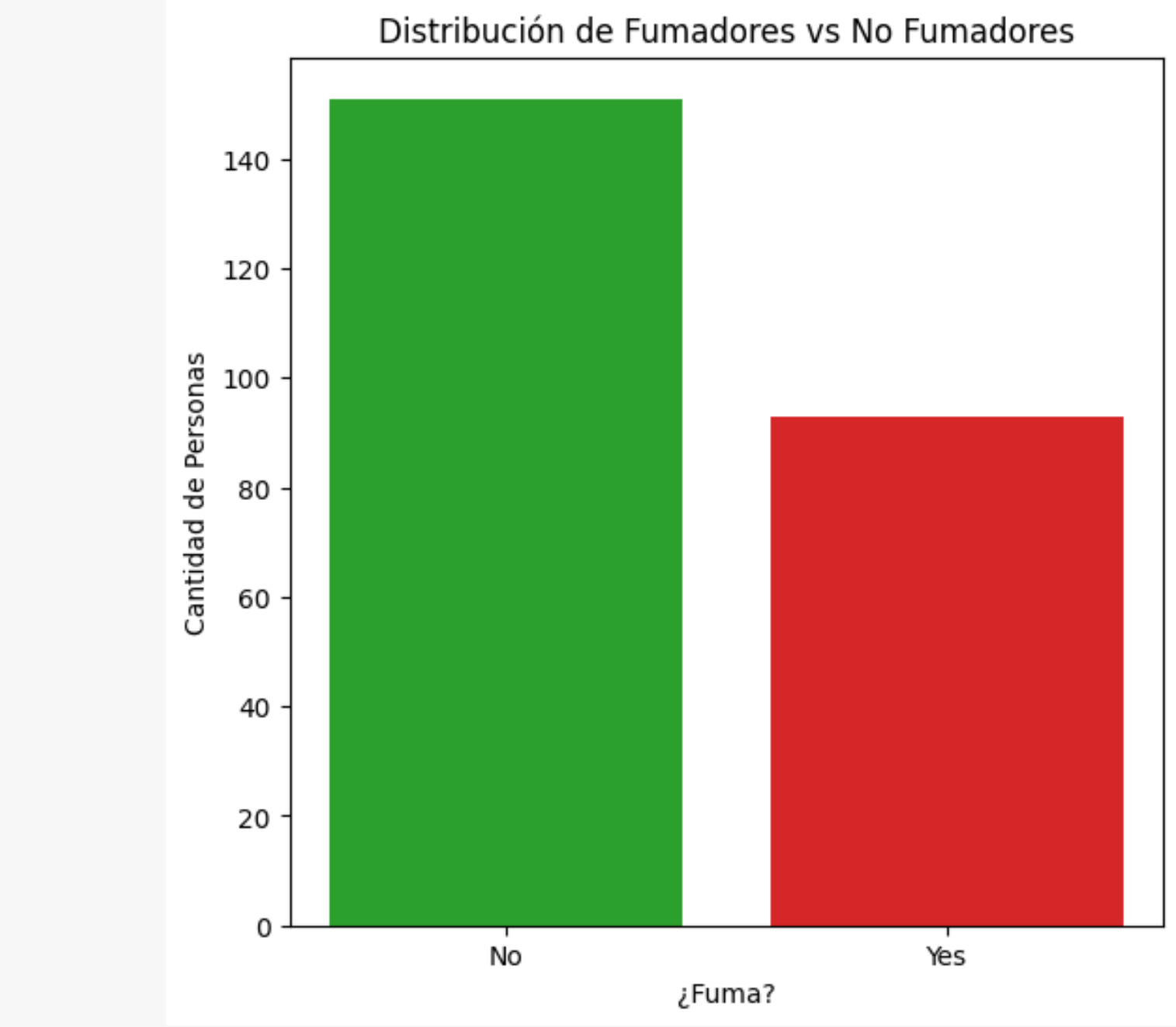
Presenta la distribución de una variable categórica con una sola serie de datos.

Muestra

Frecuencias o conteos para cada categoría de la variable

Uso

Se usa para mostrar la distribución de una variable cualitativa



Código

```
plt.figure(figsize=(6, 6))
smoker_counts = df_tips['smoker'].value_counts()
colors = ['2ca02c', 'd62728'] VerdeparaNo, RojoparaS
plt.bar(smoker_counts.index, smoker_counts.values, color = colors)
plt.title('DistribucindeFumadoresvsNoFumadores')
plt.xlabel('Fuma?')
plt.ylabel('CantidaddePersonas')
plt.show()
```

PALETA DE COLORES

Este trabajo utiliza colores distintivos asignados estratégicamente según el tipo de variable analizada.

- Azul (#1f77b4)
- Naranja (#ff7f0e)
- Verde (#2ca02c)
- Rojo (#d62728)

BARRAS COMPUESTAS

Definición

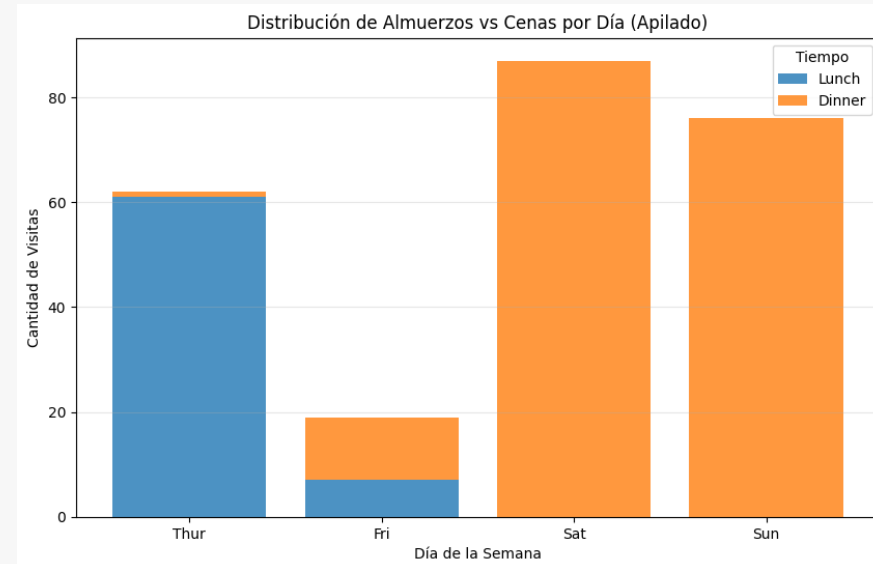
Muestra la composición interna de cada categoría mediante barras apiladas.

Muestra

Subcategorías dentro de cada categoría principal, apiladas verticalmente.

Uso

Se usa para analizar la estructura interna de cada categoría.



Codigo

```
tabla_cruzada = pd.crosstab(df_tips['day'], df_tips['time'])
colores = ['#1f77b4', '#ff7f0e']
plt.figure(figsize=(10, 6))
bottom_vals = np.zeros(len(tabla_cruzada))
for i, columna in enumerate(tabla_cruzada.columns):
    plt.bar(tabla_cruzada.index,
            tabla_cruzada[columna],
            bottom=bottom_vals,
            label=columna,
            color=colores[i],
            alpha=0.8)
    bottom_vals += tabla_cruzada[columna].values
plt.title('Distribución de Almuerzos vs Cenas por Día (Apilado)')
plt.xlabel('Día de la Semana')
plt.ylabel('Cantidad de Visitas')
plt.legend(title='Tiempo')
plt.grid(axis='y', alpha=0.3)
plt.show()
```

GRAFICO BARRAS COMPARATIVAS

Definición

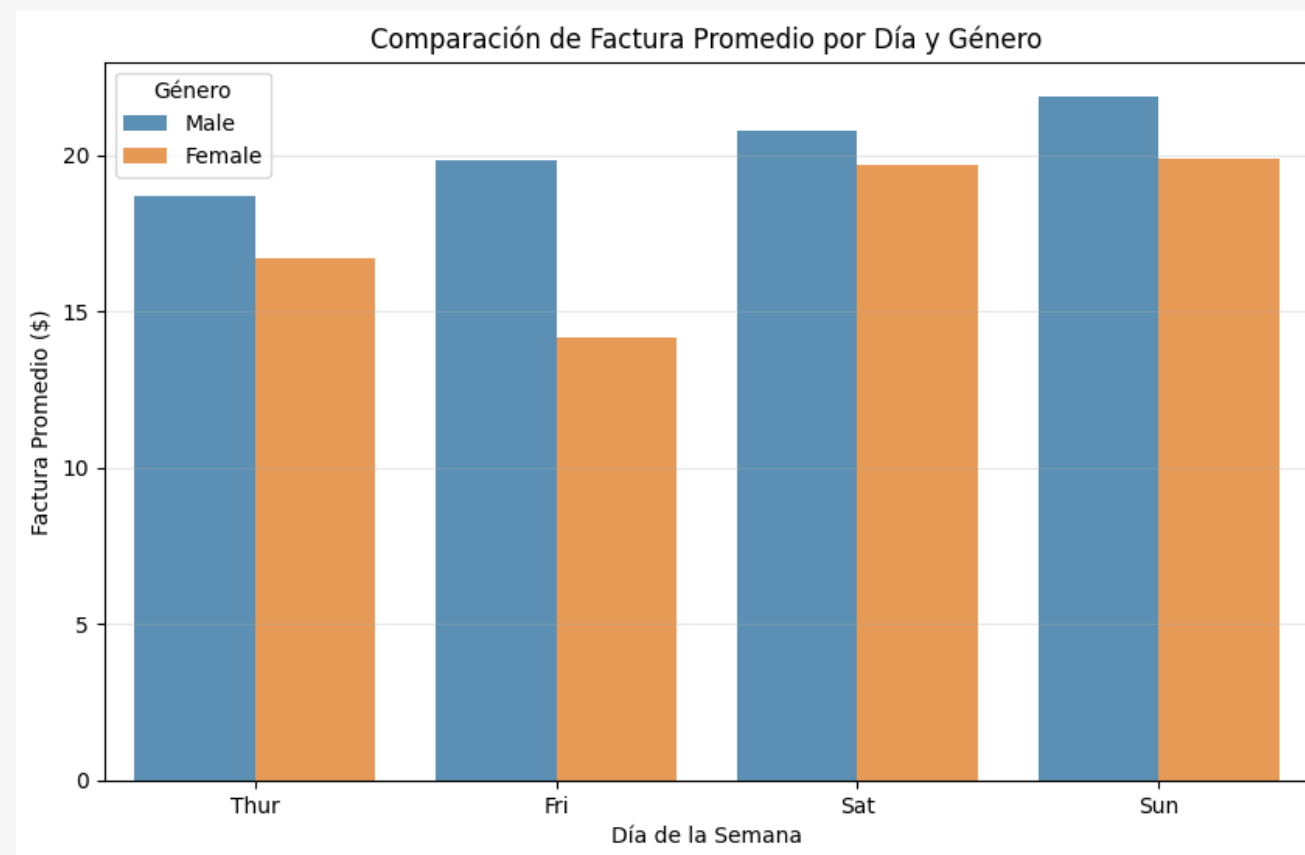
Compara múltiples series de datos para las mismas categorías mediante barras agrupadas.

Muestra

Diferentes grupos o series lado a lado para cada categoría.

Uso

Se usa para comparar diferentes grupos dentro de las mismas categorías.



Codigo

```
plt.figure(figsize=(10, 6))
sns.barplot(data=df_tips, x='day', y='total_bill', hue='sex', palette=['#1f77b4', '#ff7f0e'], alpha=0.8, errorbar=None)
plt.title('Comparación de Factura Promedio por Día y Género')
plt.xlabel('Día de la Semana')
plt.ylabel('Factura Promedio ($)')
plt.legend(title='Género')
plt.grid(axis='y', alpha=0.3)
plt.show()
```

GRÁFICO DE PIRÁMIDE

Definición

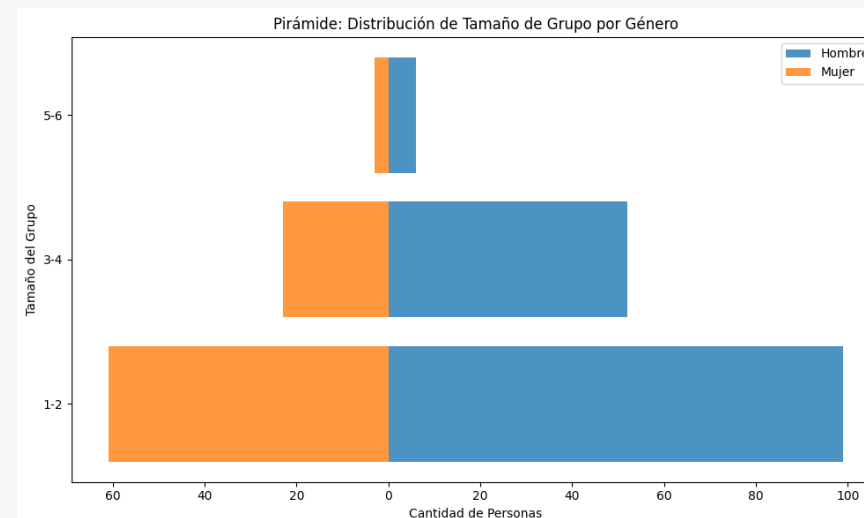
Representa la distribución de una variable dividida en dos grupos en forma simétrica.

Muestra

Dos histogramas horizontales enfrentados, usualmente por género.

Uso

Se usa para comparar la distribución de dos grupos en una misma variable.



Codigo

```
df_tips['size_category'] = pd.cut(df_tips['size'], bins=[0, 2, 4, 6], labels=['1-2', '3-4', '5-6'])
```

```
piramide_ata = pd.crosstab(df_tips['size_category'], df_tips['sex'])
```

```
fig, ax = plt.subplots(figsize=(10, 6))
```

```
y_pos = np.arange(len(piramide_ata.index))
```

```
ax.barh(y_pos, piramide_ata['Male'], color='#1f77b4', alpha=0.8, label='Hombre')
ax.barh(y_pos, -piramide_ata['Female'], color='#ff7f0e', alpha=0.8, label='Mujer')
```

```
ax.set_yticks(y_pos)
ax.set_yticklabels(piramide_ata.index)
ax.set_xlabel('Cantidad de Personas')
ax.set_ylabel('Tamaño del Grupo')
ax.set_title('Pirámide: Distribución de Tamaño de Grupo por Género')
ax.legend()
```

```
xticks = ax.get_x_ticks()
ax.set_xticklabels([f'abs(int(x))' for x in xticks])
```

```
plt.tight_layout()
plt.show()
```

Gráfico pastel

Definición

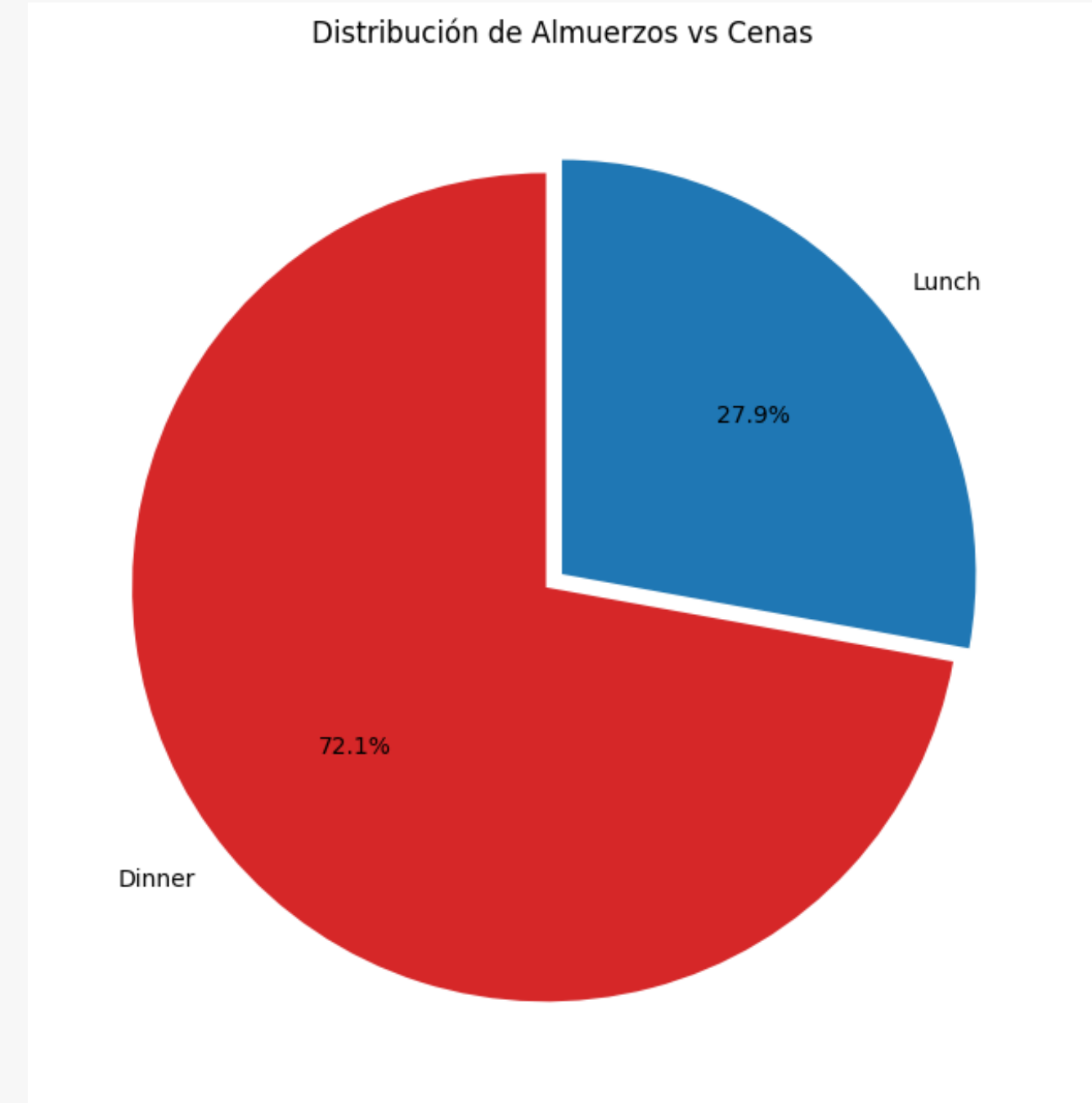
Muestra la proporción de cada categoría como partes de un todo circular.

Muestra

Porcentajes o proporciones de cada categoría en un círculo dividido.

Uso

Se usa para representar composiciones porcentuales o proporciones.



Código

```
tiempo_counts = df_tips['time'].value_counts()
colores_pastel = ['#ff9999', '#66b3ff']
plt.figure(figsize=(8, 8))
plt.pie(tiempo_counts.values, labels=tiempo_counts.index,
        autopct='%1.1f%%', colors=colores_pastel, startangle=90, explode=(0.05, 0))
plt.title('Distribución de Almuerzos vs Cenas')
plt.show()
```


Tercera parte

BARRAS 100

Definición

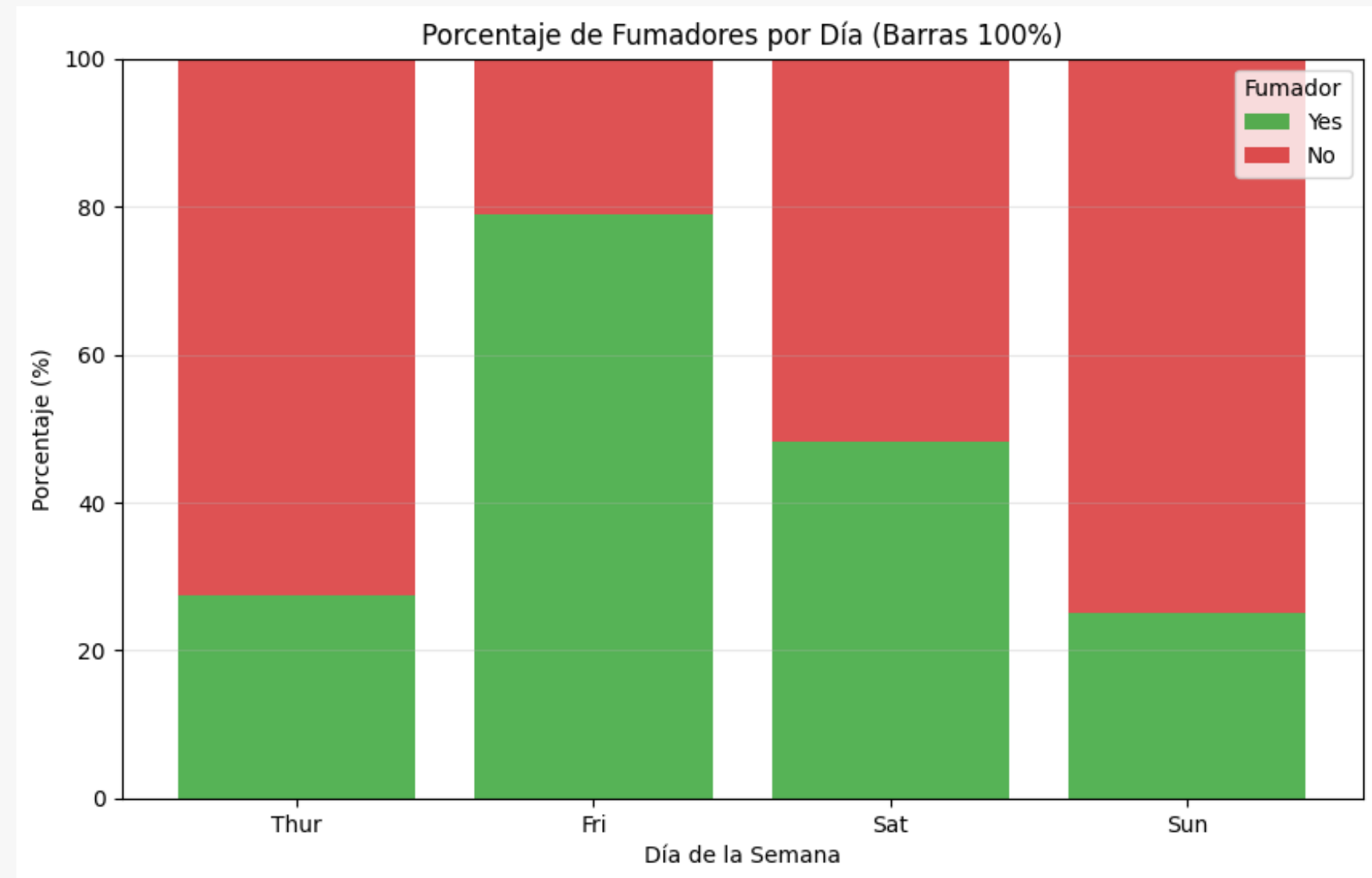
Presenta la composición porcentual de cada categoría con barras de igual altura.

Muestra

Porcentajes acumulados que suman 100% para cada categoría.

Uso

Se usa para comparar estructuras porcentuales entre diferentes categorías.



Codigo

```
tabla_porcentaje = pd.crosstab(df_ips['day'], df_ips['smoker'], normalize = 'index') * 100
plt.figure(figsize = (10, 6))
bottom = np.zeros(len(tabla_porcentaje))
for i, column in enumerate(tabla_porcentaje.columns):
    plt.bar(tabla_porcentaje.index, tabla_porcentaje[column], bottom=bottom,
            label = column, color = ['2ca02c', 'd62728'][i], alpha = 0.8)
    bottom += tabla_porcentaje[column].values
plt.title('Porcentaje de Fumadores por Día (Barras 100)')
plt.xlabel('Día de la Semana')
plt.ylabel('Porcentaje (%)')
plt.legend(title = 'Fumador')
plt.ylim(0, 100)
plt.grid(axis = 'y', alpha = 0.3)
plt.show()
```

GRÁFICO LINEAL

Definición

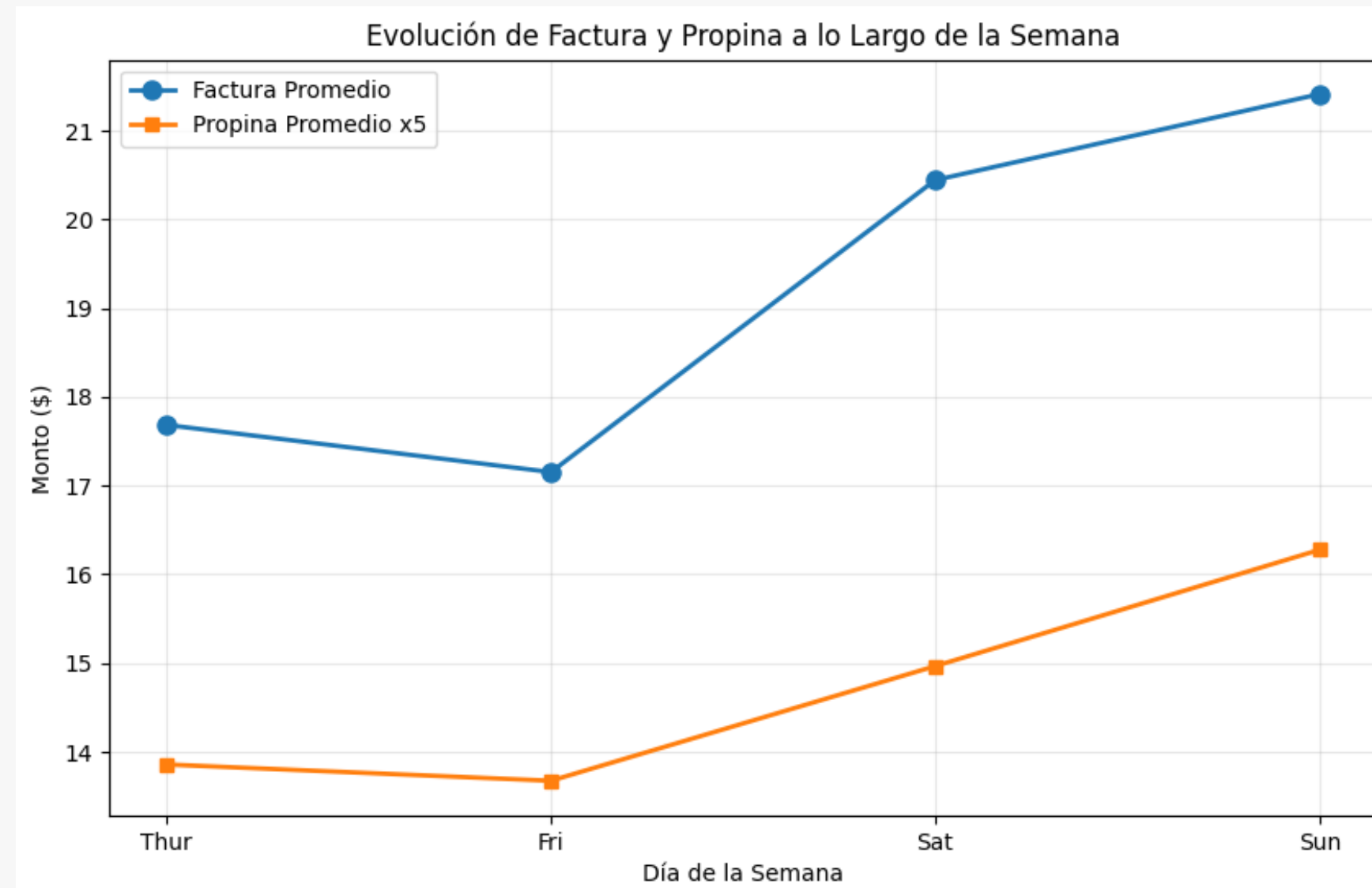
Muestra la evolución o tendencia de una variable a lo largo del tiempo o secuencia.

Muestra

Puntos conectados por líneas que indican la progresión de los valores.

Uso

Se usa para analizar tendencias, patrones y cambios a lo largo del tiempo.



Codigo

```
plt.figure(figsize=(10, 6))
plt.plot(tendencia_dia['day'], tendencia_dia['total_bill'], marker = 'o',
         linewidth = 2, markersize = 8, label = 'Factura Promedio', color = '1f77b4')
plt.plot(tendencia_dia['day'], tendencia_dia['tip'] * 5,
         marker = 's', linewidth = 2, markersize = 6, label = 'Propina Promedio x5', color = 'ff7f0e')
plt.title('Evolución de Factura y Propina a lo Largo de la Semana')
plt.xlabel('Día de la Semana')
plt.ylabel('Monto ($)')
plt.legend()
plt.grid(True, alpha = 0.3)
plt.show()
```

GRÁFICO RADAR

Definición

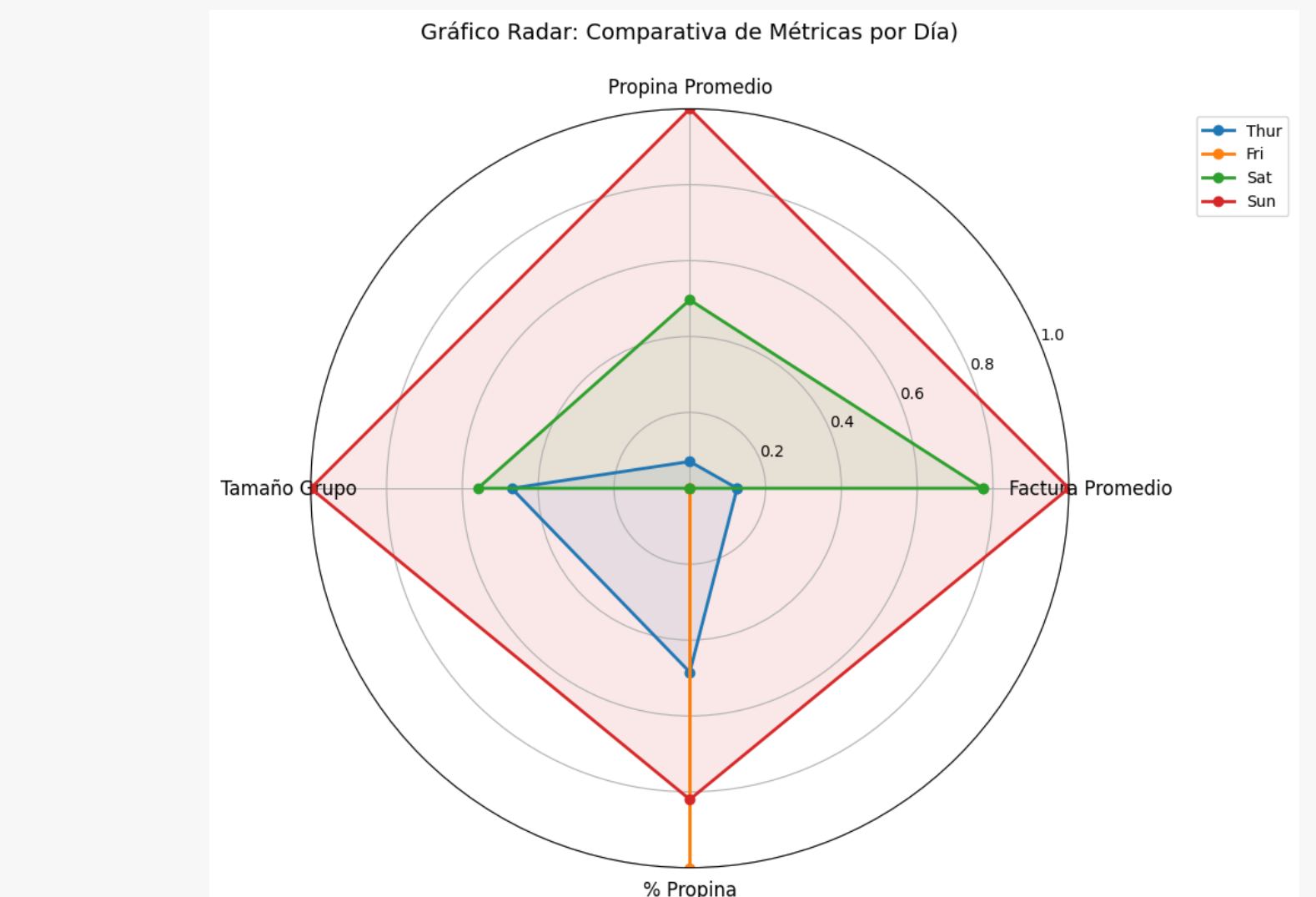
Representa múltiples variables cuantitativas en ejes radiales que parten de un centro común.

Muestra

Valores normalizados de diferentes variables conectadas formando un polígono.

Uso

Se usa para comparar perfiles multivariados entre diferentes grupos o categorías.



Código

```
metrics_day = df_ips.groupby('day').agg(['total_bill' : 'mean', 'tip' : 'mean', 'size' : 'mean']).reset_index()

df_ips['tip_percentage'] = (df_ips['tip'] / df_ips['total_bill']) * 100
metrics_day = df_ips.groupby('day')['tip_percentage'].mean().reset_index()

metrics_day = metrics_day.merge(tip_pct_day, on = 'day')
metrics_day = metrics_day[metrics_day['day'].isin(['Thur', 'Fri', 'Sat', 'Sun'])]
radar_metrics = metrics_day[['total_bill', 'tip', 'size', 'tip_percentage']].copy()
def normalize(series):
    return (series - series.min()) / (series.max() - series.min())
radar_normalized = radar_metrics.apply(normalize)
categories = ['Factura Promedio', 'Propina Promedio', 'Tamaño Grupo', 'N = len(categories)']
angles = [n / float(N) * 2 * pi for n in range(N)]
angles += angles[:1]
fig, ax = plt.subplots(figsize=(10, 10), subplot_kw=dict(projection = 'polar'))
colors = ['1f77b4', 'ff7f0e', '2ca02c', 'd62728']
days = metrics_day['day'].values
for i, day in enumerate(days):
    values = radar_normalized.iloc[i].values.flatten().tolist()
    values += values[:1]
    ax.plot(angles, values, 'o-', linewidth=2, label=day, color=colors[i])
    ax.fill(angles, values, alpha=0.1, color=colors[i])
    ax.set_xticks(angles[:-1])
    ax.set_xticklabels(categories, fontsize = 12)
    ax.set_ylim(0, 1)
    ax.set_yticks([0.2, 0.4, 0.6, 0.8, 1.0])
    ax.set_yticklabels(['0.2', '0.4', '0.6', '0.8', '1.0'], fontsize = 10)
    ax.grid(True)
plt.title('Gráfico Radar: Comparativa de Métricas por Día (Normalizado 0-1)', size=14, y=1.08)
plt.legend(loc='upper right', bbox_to_anchor = (1.3, 1.0))
plt.tight_layout()
plt.show()
```