


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
# Ignore all warnings
warnings.filterwarnings("ignore")

# Set the style of the plots to "darkgrid"
sns.set(style="darkgrid")
```

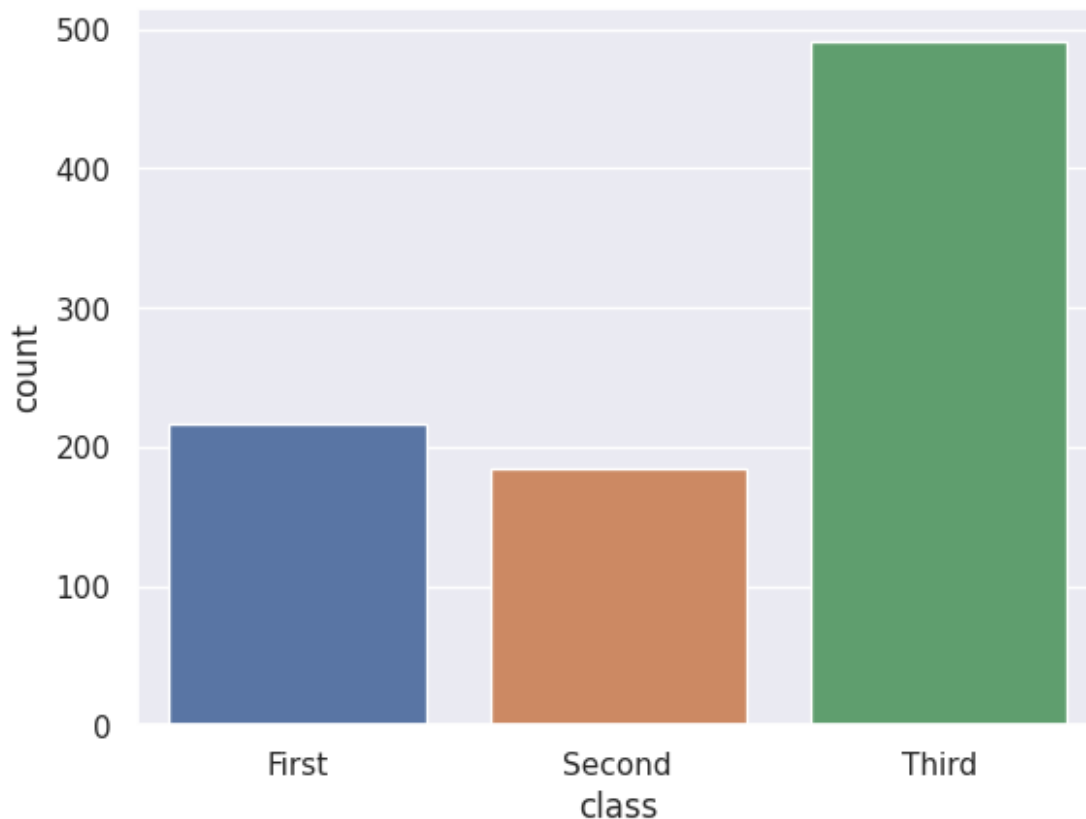
Count Plot

The count plot is a type of categorical plot in seaborn that displays the count of occurrences of each category in a categorical variable.

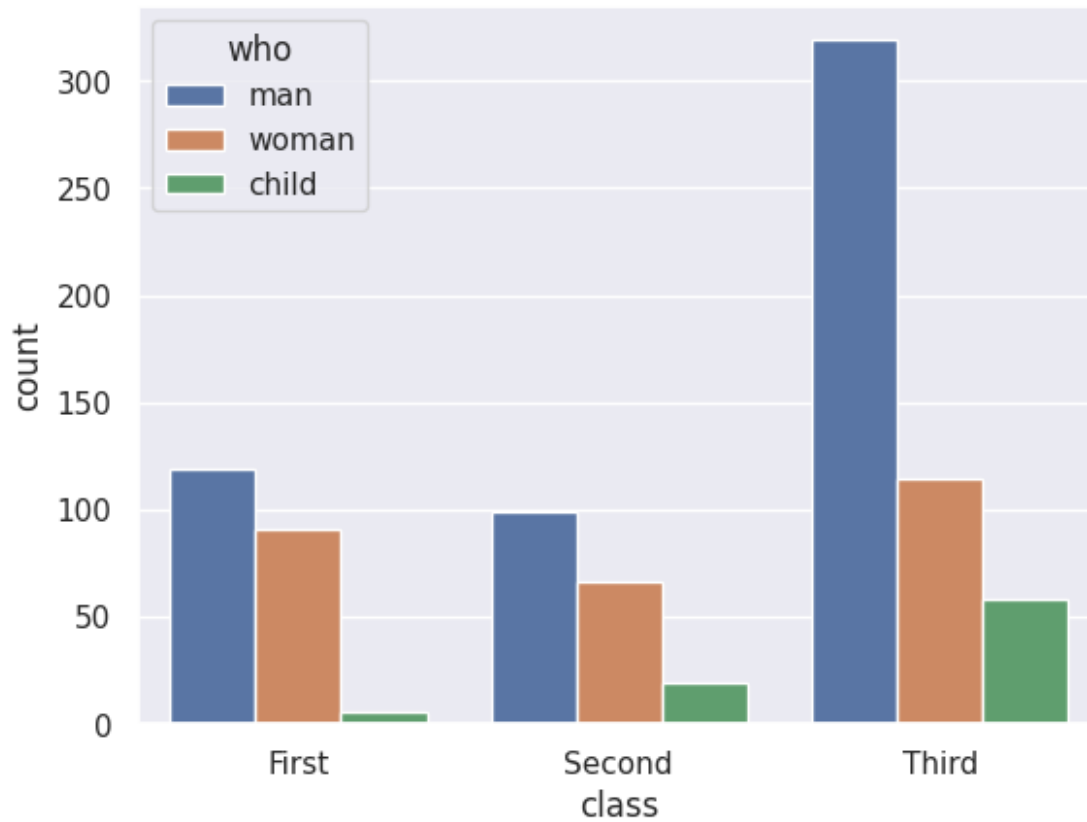
It is useful for visualizing the distribution and frequency of categorical data.

```
# Load the titanic dataset from seaborn
titanic = sns.load_dataset("titanic")

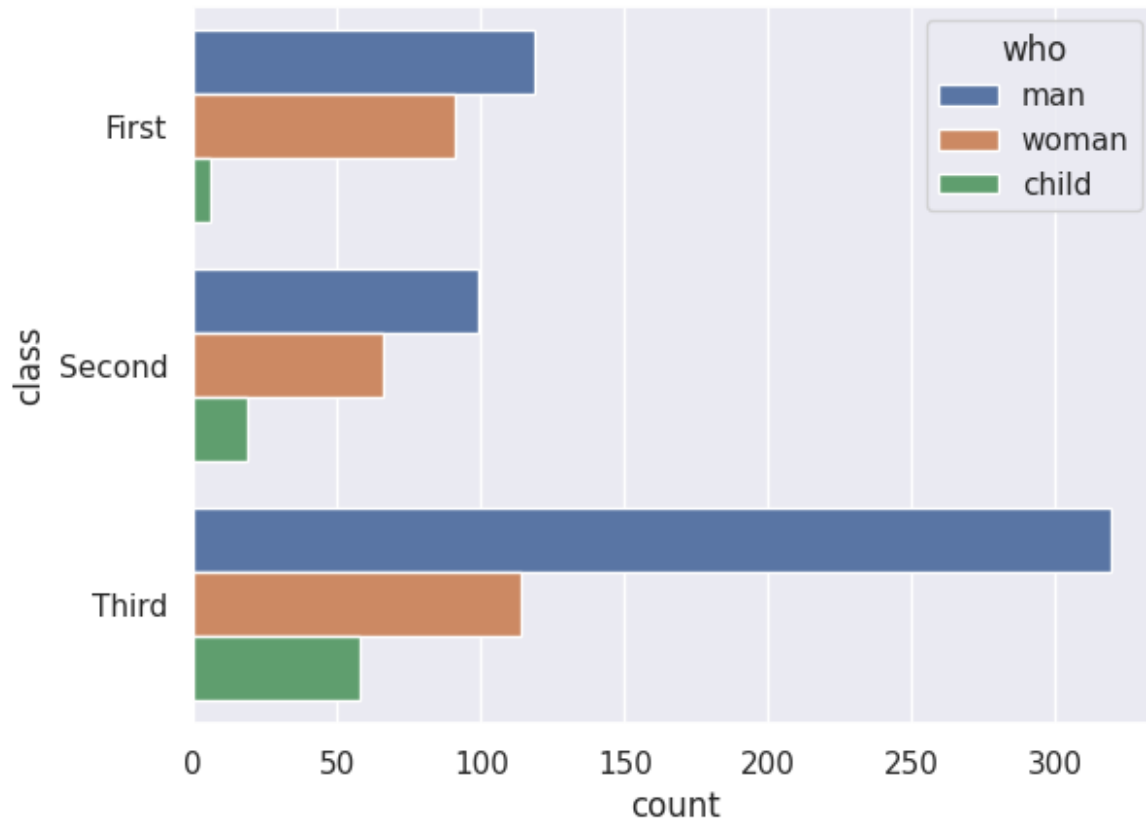
# Create a count plot of the "class" column
ax= sns.countplot(x="class", data = titanic)
plt.show()
```



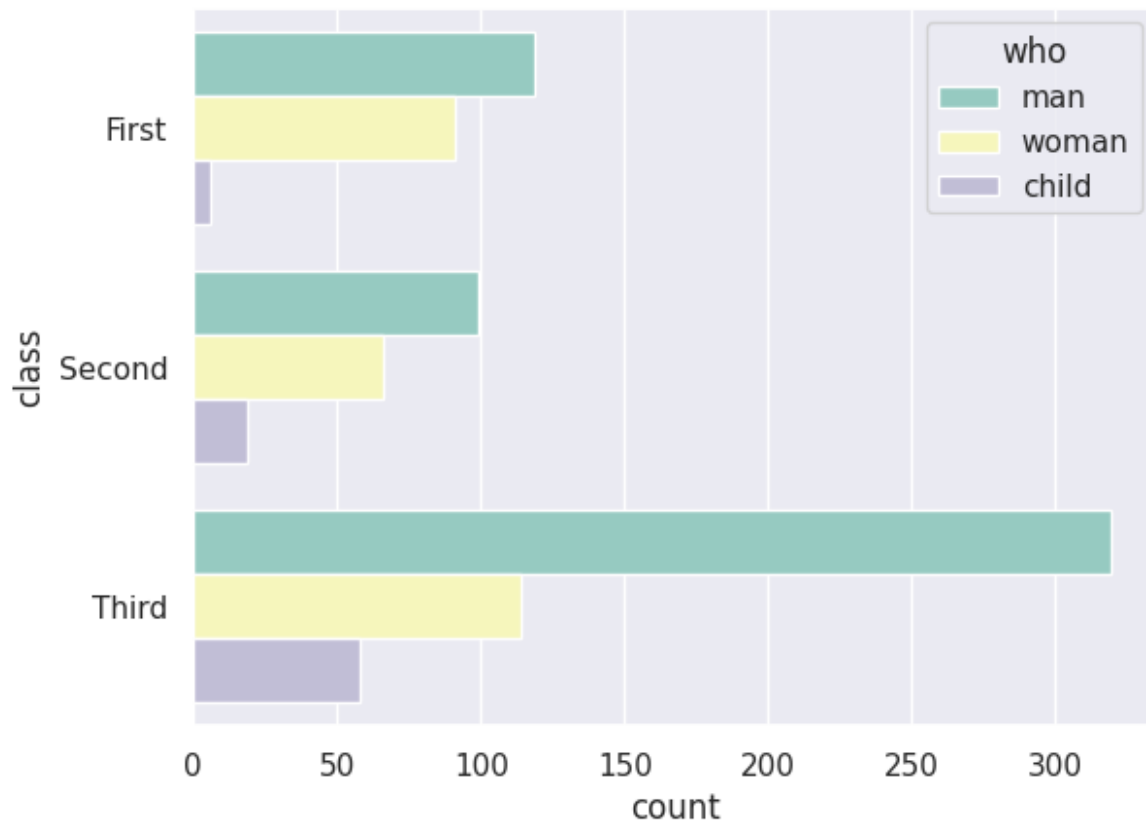
```
# Create a count plot of the "class" column, grouped by "who"
ax= sns.countplot(x="class",hue="who", data = titanic)
plt.show()
```



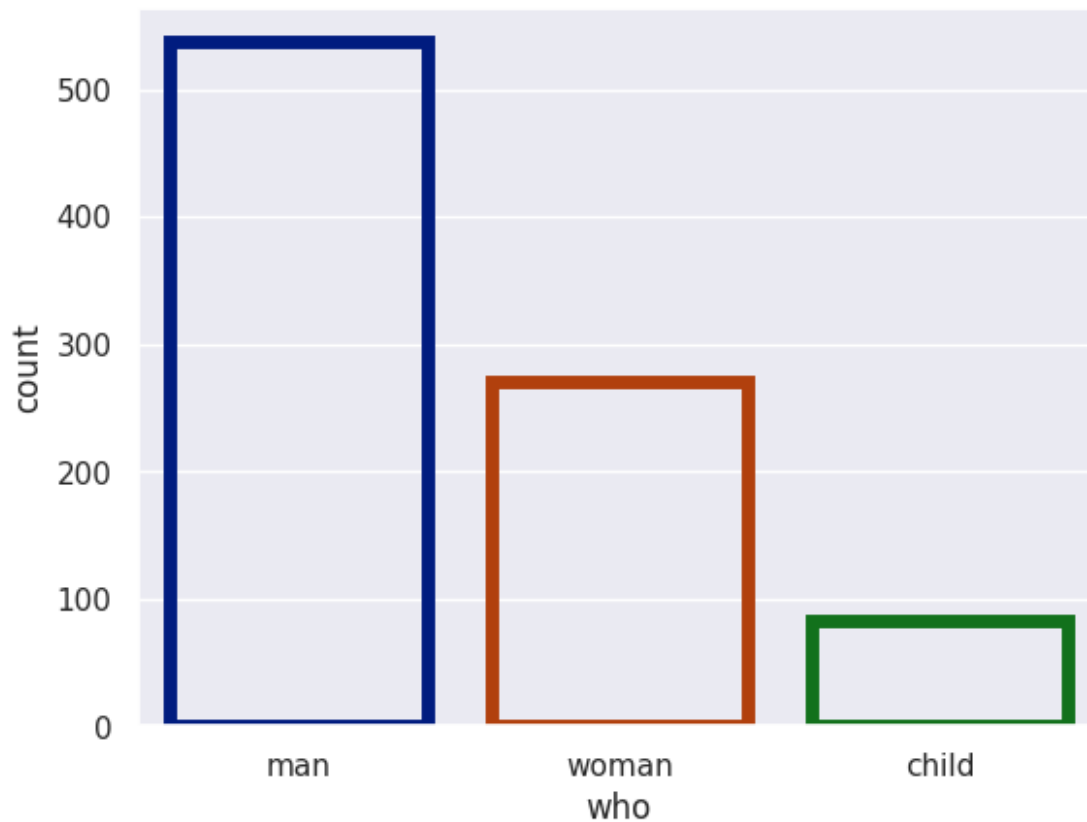
```
# Create a count plot of the "class" column (vertical orientation),
grouped by "who"
ax= sns.countplot(y="class",hue="who", data = titanic)
plt.show()
```



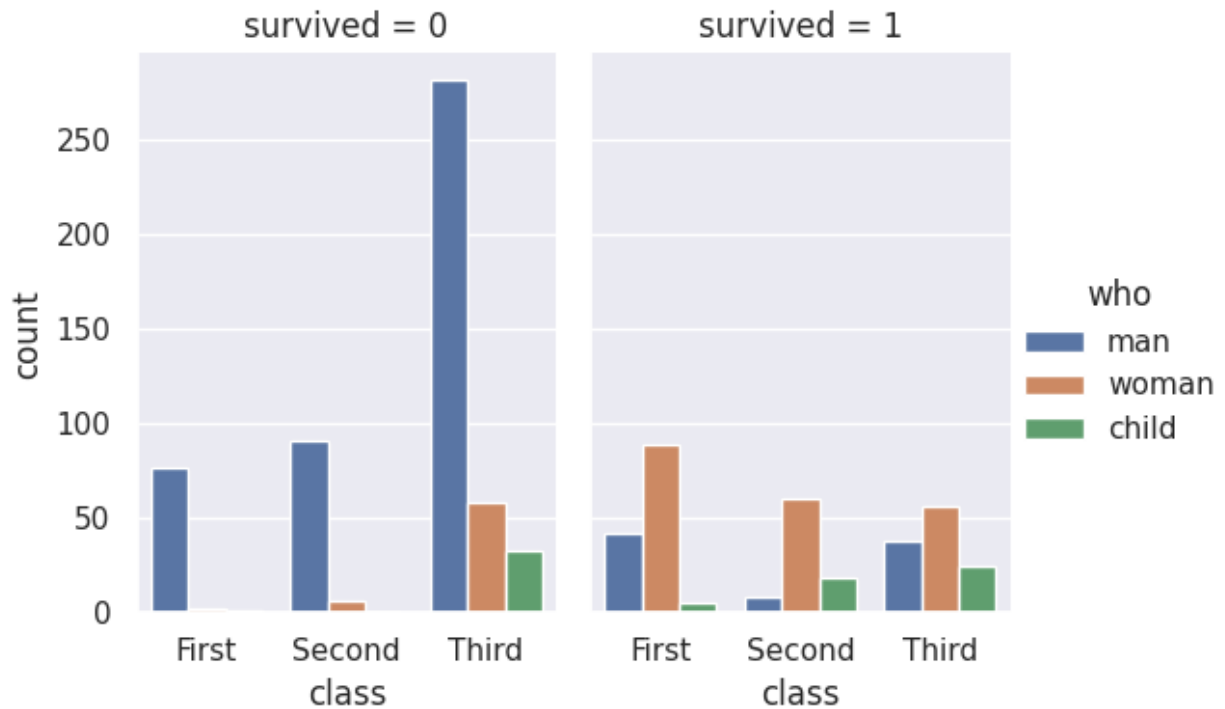
```
# Create a count plot of the "class" column (vertical orientation),  
# grouped by "who",  
# using a custom color palette ("Set3")  
ax= sns.countplot(y="class",hue="who", data = titanic,palette="Set3")  
plt.show()
```



```
# Create a count plot of the "who" column, with a customized appearance
ax= sns.countplot(x="who", data = titanic,facecolor =
(0,0,0,0),linewidth=5,edgecolor=sns.color_palette("dark",3))
plt.show()
```



```
# Create a categorical plot with multiple subplots (columns based on
"survived"),
# showing the count of each "class" group, grouped by "who"
g = sns.catplot(x="class",hue="who", data =
titanic,col="survived",kind="count",height=4,aspect=.7)
plt.show()
```



```
import seaborn as sns
import matplotlib.pyplot as plt

# Set the style of the plots to "darkgrid"
sns.set(style="darkgrid")

# Load the titanic dataset from seaborn
titanic = sns.load_dataset("titanic")

# Create a figure with 2 rows and 3 columns
fig, axes = plt.subplots(2, 3, figsize=(12, 8))

# Plot 1 - Count plot of "class" column
sns.countplot(x="class", data=titanic, ax=axes[0, 0])
axes[0, 0].set_title("Class")

# Plot 2 - Count plot of "class" column grouped by "who"
sns.countplot(x="class", hue="who", data=titanic, ax=axes[0, 1])
axes[0, 1].set_title("Class Grouped by Who")

# Plot 3 - Count plot of "class" column (vertical) grouped by "who"
sns.countplot(y="class", hue="who", data=titanic, ax=axes[0, 2])
axes[0, 2].set_title("Class (Vertical) Grouped by Who")

# Plot 4 - Count plot of "class" column (vertical) grouped by "who"
# with custom color palette
sns.countplot(y="class", hue="who", data=titanic, palette="Set3",
ax=axes[1, 0])
```

```

axes[1, 0].set_title("Class (Vertical) Grouped by Who")

# Plot 5 - Count plot of "who" column with customized appearance
sns.countplot(x="who", data=titanic, facecolor=(0, 0, 0, 0),
linewidth=5,
               edgecolor=sns.color_palette("dark", 3), ax=axes[1, 1])
axes[1, 1].set_title("Who with Customized Appearance")

# Adjust the spacing between subplots
plt.tight_layout()

# Remove the xlabel for subplots in the first row
for ax in axes[0]:
    ax.set_xlabel("")

# Display the plots
plt.show()

```



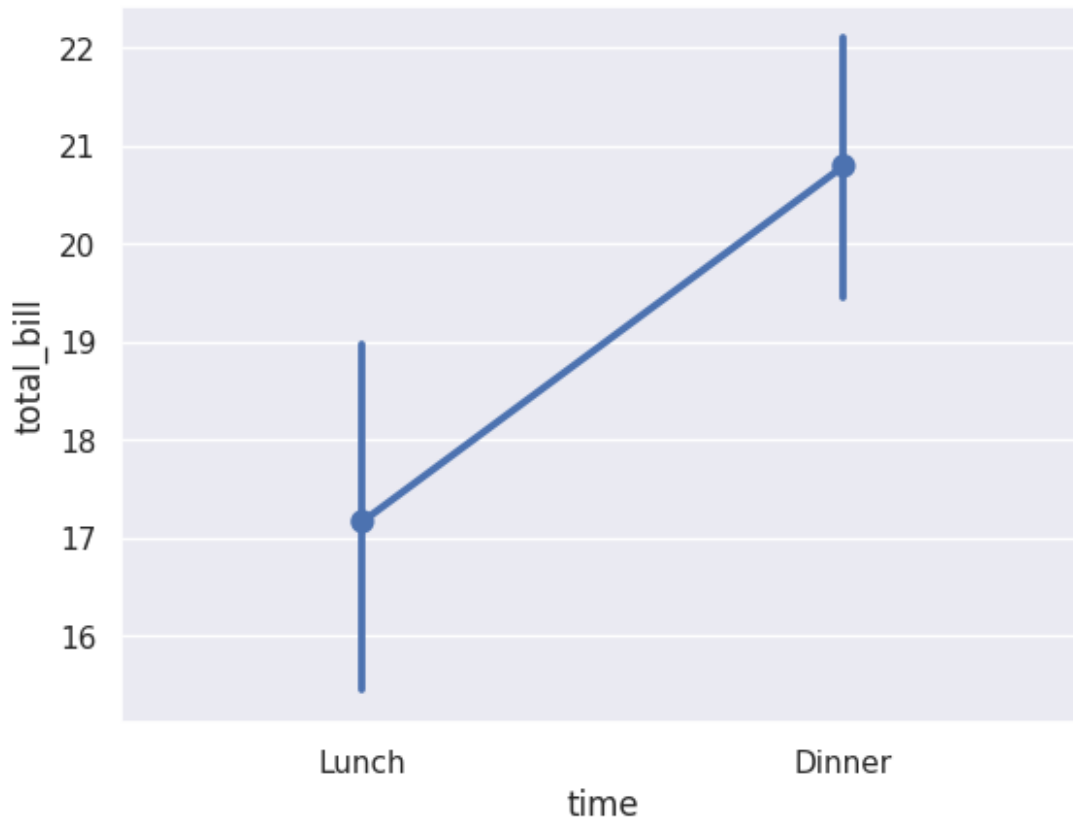
Point Plot

The point plot is a type of categorical plot in seaborn that displays an estimate of central tendency and confidence interval using points.

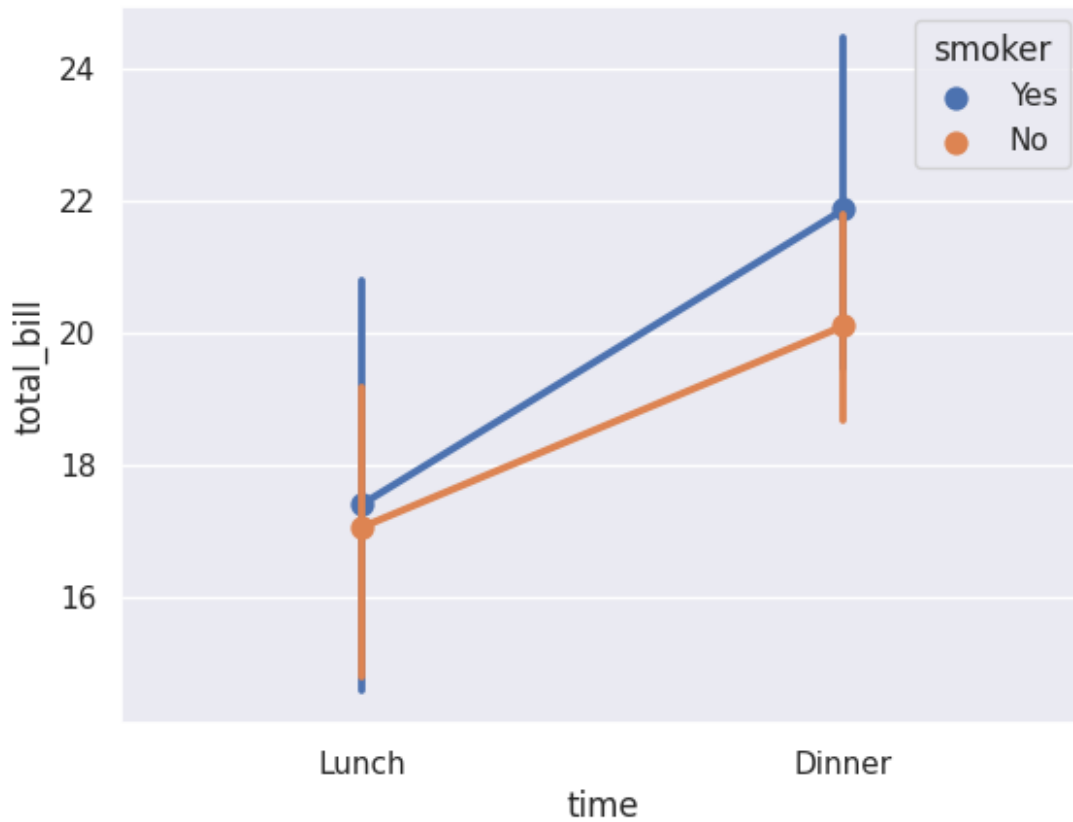
It is useful for comparing the mean or median values of a numerical variable across different categories.

```
# Load the tips dataset from seaborn
tips = sns.load_dataset("tips")

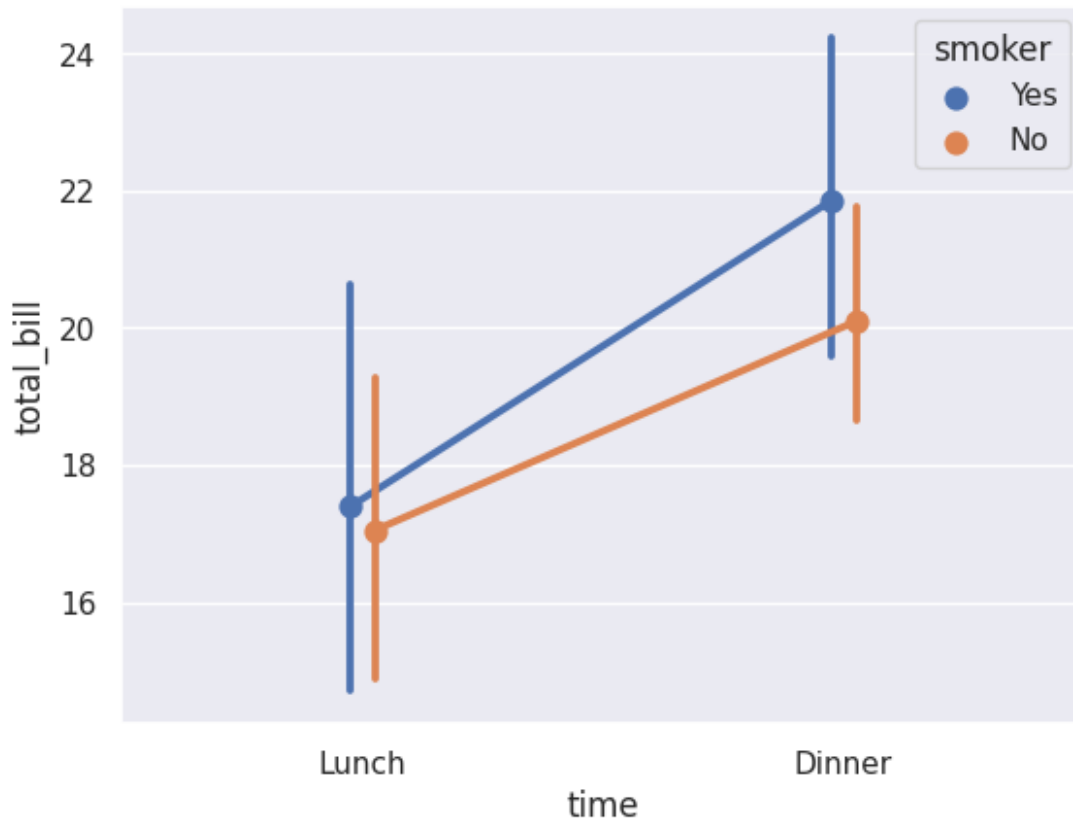
# Point plot of "total_bill" by "time"
ax = sns.pointplot(x="time", y="total_bill", data=tips)
plt.show()
```



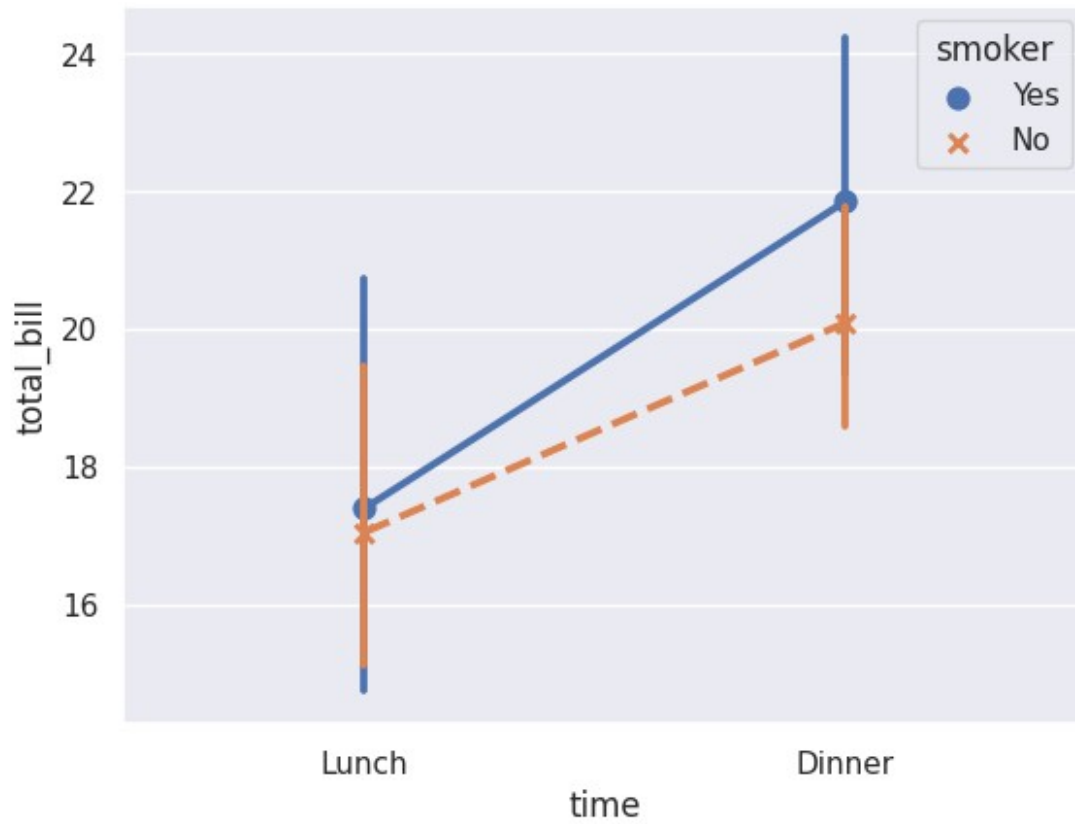
```
# Point plot of "total_bill" by "time" with hue by "smoker"
ax = sns.pointplot(x="time", y="total_bill", hue="smoker", data=tips)
plt.show()
```



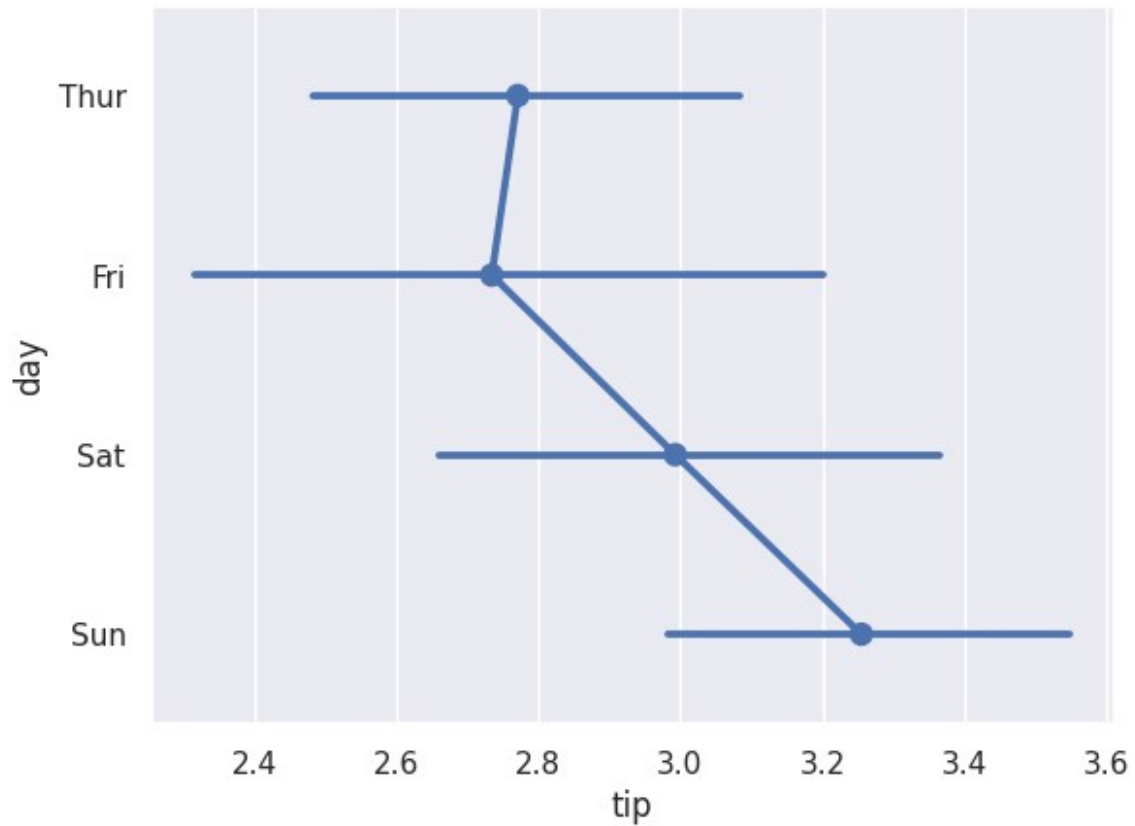
```
# Point plot of "total_bill" by "time" with hue by "smoker" and dodge enabled
ax = sns.pointplot(x="time",
y="total_bill",hue="smoker",data=tips,dodge=True)
plt.show()
```



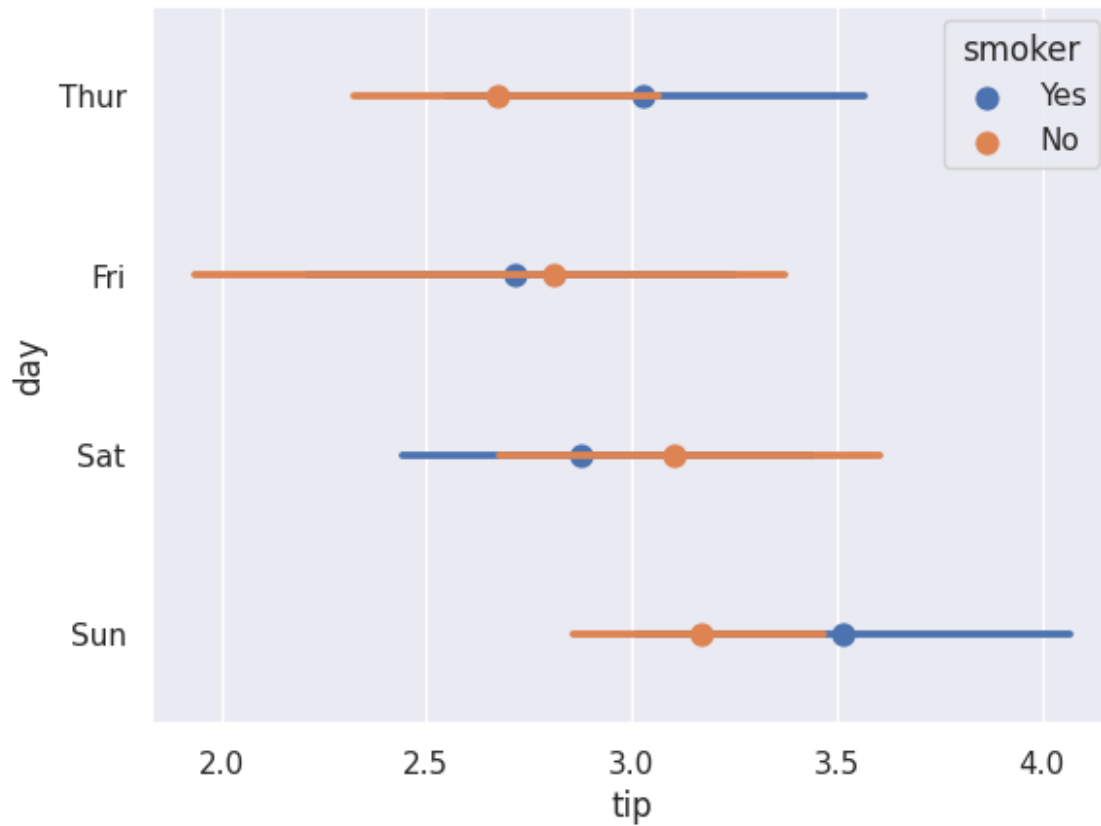
```
# Point plot of "total_bill" by "time" with hue by "smoker", custom  
markers, and linestyles  
ax = sns.pointplot(x="time",  
y="total_bill",hue="smoker",data=tips,markers=["o","x"],linestyles=["-  
","--"])  
plt.show()
```



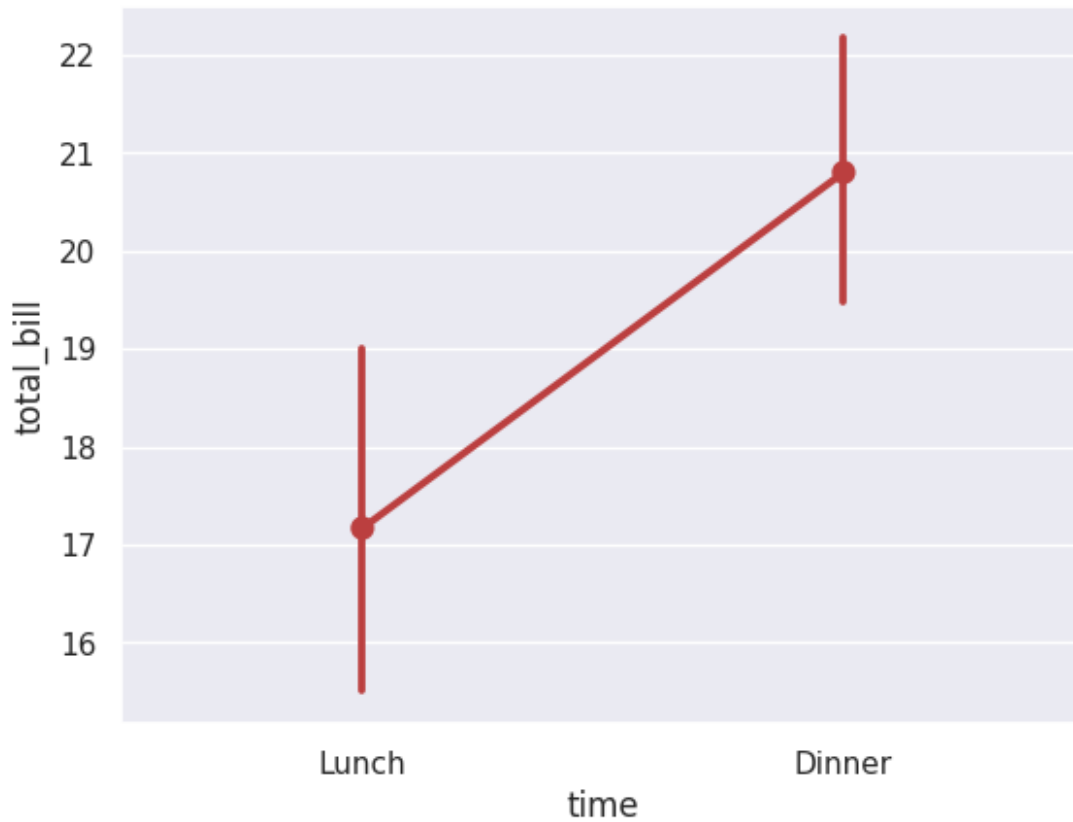
```
# Point plot of "tip" by "day"  
ax= sns.pointplot(x="tip", y="day",data=tips)  
plt.show()
```



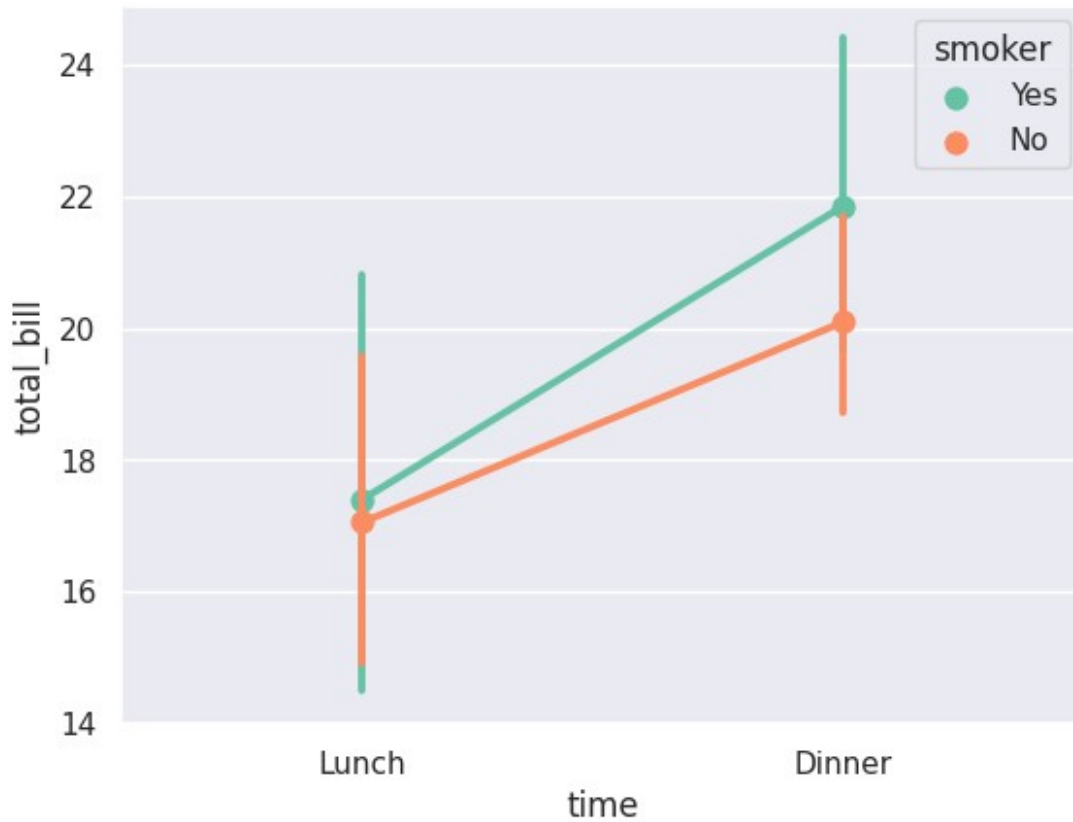
```
# Point plot of "tip" by "day" with hue by "smoker" and no line  
joining points  
ax = sns.pointplot(x="tip", y="day", hue="smoker", data=tips, join=False)  
plt.show()
```



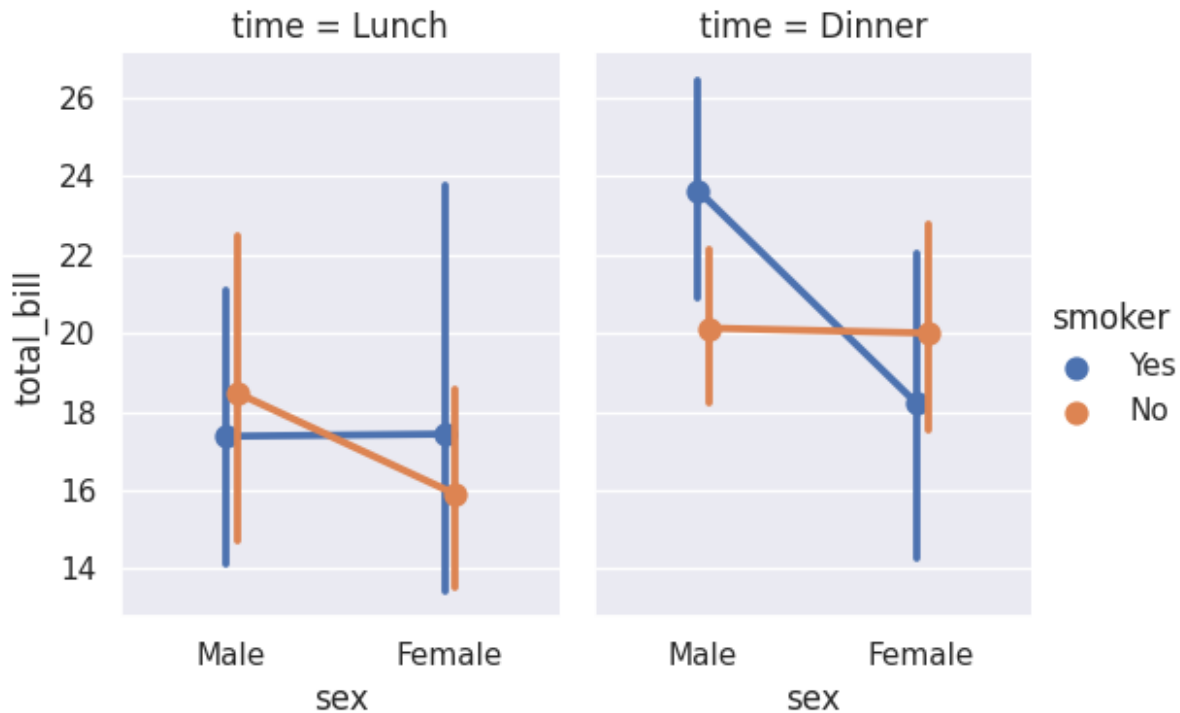
```
# Point plot of "total_bill" by "time" with custom color
ax = sns.pointplot(x="time", y="total_bill", data=tips, color="#bb3f3f")
plt.show()
```



```
# Point plot of "total_bill" by "time" with hue by "smoker" and custom  
color palette  
ax = sns.pointplot(x="time",  
y="total_bill",hue="smoker",data=tips,palette="Set2")  
plt.show()
```



```
# Categorical point plot of "total_bill" by "sex" with hue by "smoker"
and multiple subplots based on "time"
ax = sns.catplot(x="sex",
y="total_bill",data=tips,dodge=True,hue="smoker",kind="point",height=4
,aspect=.7,col="time")
plt.show()
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the tips dataset from seaborn
tips = sns.load_dataset("tips")

# Set the style of the plots to "darkgrid"
sns.set(style="darkgrid")

# Create a figure with 4 subplots arranged in 2 rows and 2 columns
fig, axes = plt.subplots(4, 2, figsize=(12, 16))

# Plot 1 - Point plot of "total_bill" by "time"
sns.pointplot(x="time", y="total_bill", data=tips, ax=axes[0, 0])
axes[0, 0].set_title("Total Bill by Time")

# Plot 2 - Point plot of "total_bill" by "time" with hue by "smoker"
sns.pointplot(x="time", y="total_bill", hue="smoker", data=tips,
ax=axes[0, 1])
axes[0, 1].set_title("Total Bill by Time (Smoker)")

# Plot 3 - Point plot of "total_bill" by "time" with hue by "smoker"
and dodge enabled
sns.pointplot(x="time", y="total_bill", hue="smoker", data=tips,
dodge=True, ax=axes[1, 0])
axes[1, 0].set_title("Total Bill by Time (Smoker, Dodge)")
```

```

# Plot 4 - Point plot of "total_bill" by "time" with hue by "smoker",
custom markers, and linestyles
sns.pointplot(x="time", y="total_bill", hue="smoker", data=tips,
markers=["o", "x"], linestyles=["-", "--"], ax=axes[1, 1])
axes[1, 1].set_title("Total Bill by Time (Smoker, Custom)")

# Plot 5 - Point plot of "tip" by "day"
sns.pointplot(x="tip", y="day", data=tips, ax=axes[2, 0])
axes[2, 0].set_title("Tip by Day")

# Plot 6 - Point plot of "tip" by "day" with hue by "smoker" and no
line joining points
sns.pointplot(x="tip", y="day", hue="smoker", data=tips, join=False,
ax=axes[2, 1])
axes[2, 1].set_title("Tip by Day (Smoker, No Join)")

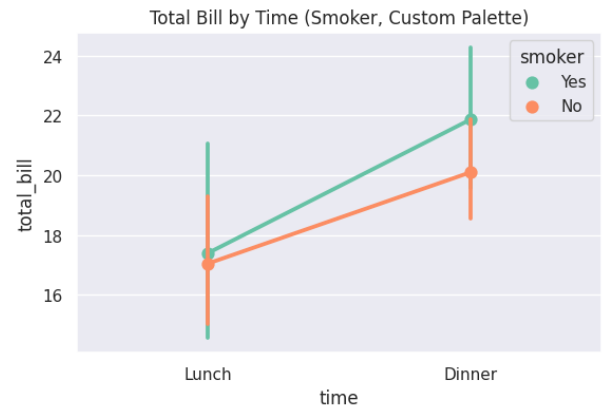
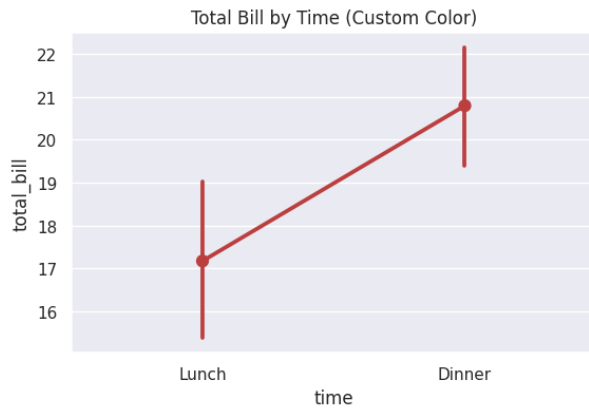
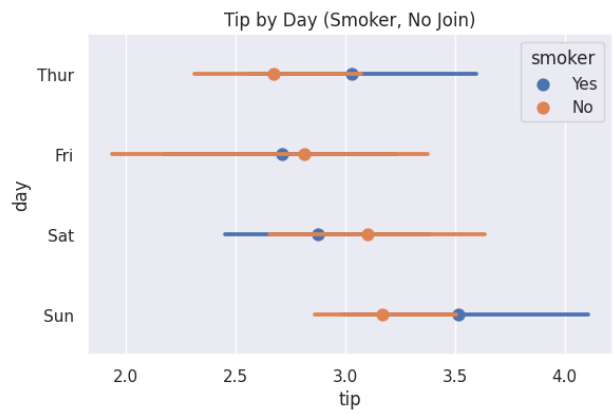
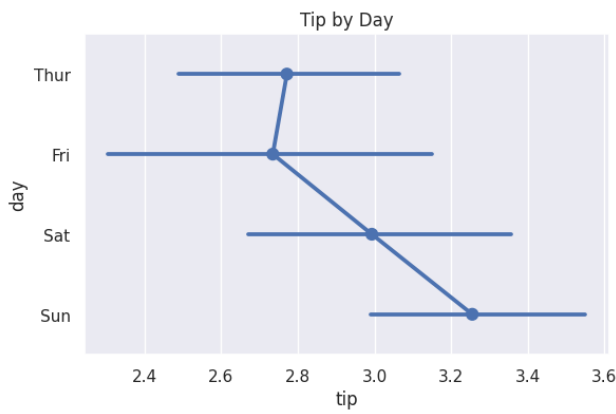
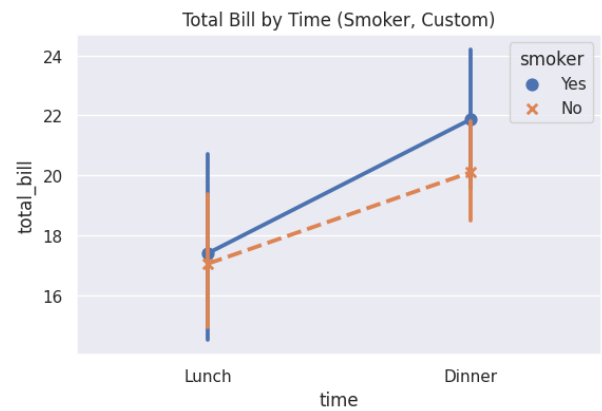
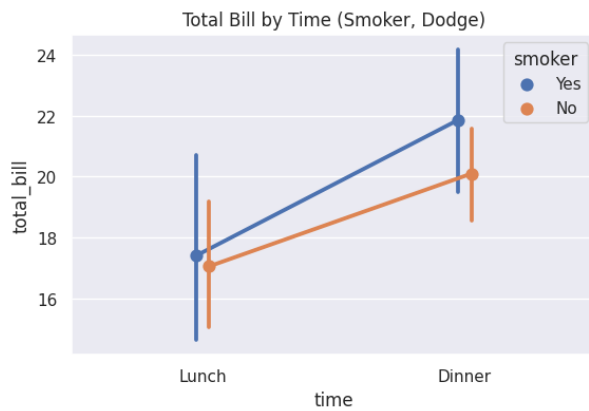
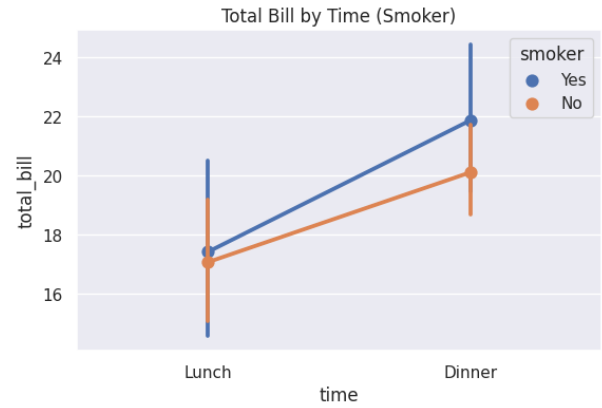
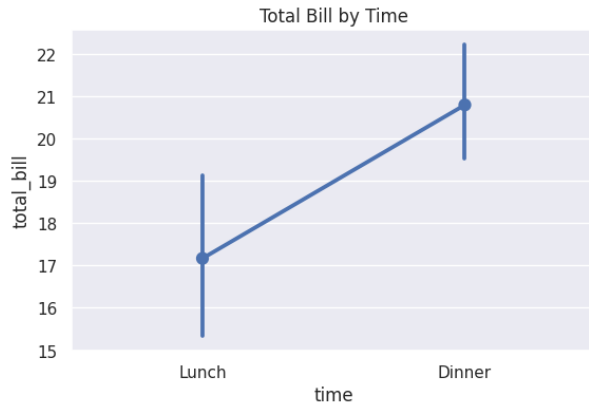
# Plot 7 - Point plot of "total_bill" by "time" with custom color
sns.pointplot(x="time", y="total_bill", data=tips, color="#bb3f3f",
ax=axes[3, 0])
axes[3, 0].set_title("Total Bill by Time (Custom Color)")

# Plot 8 - Point plot of "total_bill" by "time" with hue by "smoker"
and custom color palette
sns.pointplot(x="time", y="total_bill", hue="smoker", data=tips,
palette="Set2", ax=axes[3, 1])
axes[3, 1].set_title("Total Bill by Time (Smoker, Custom Palette)")

# Adjust the spacing between subplots
plt.tight_layout()

# Display the plots
plt.show()

```



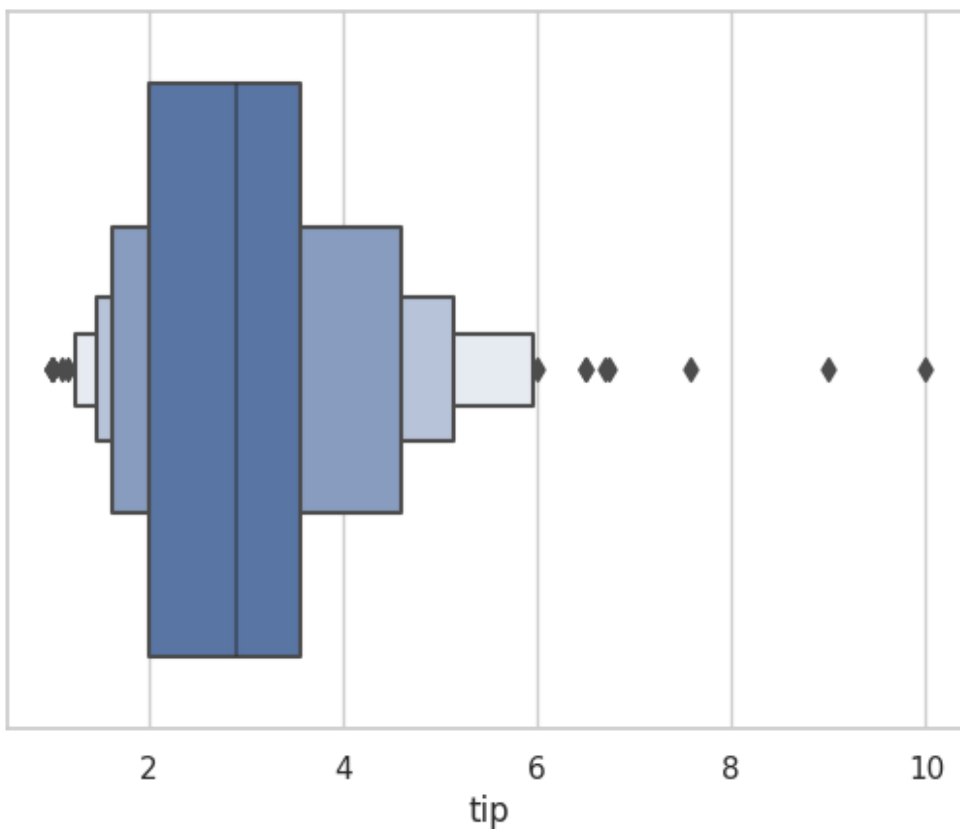
Boxenplot

The boxenplot, also known as the letter value plot or the "box plot with enhanced distribution", is a type of categorical plot in seaborn that displays the distribution of a numerical variable within different categories.

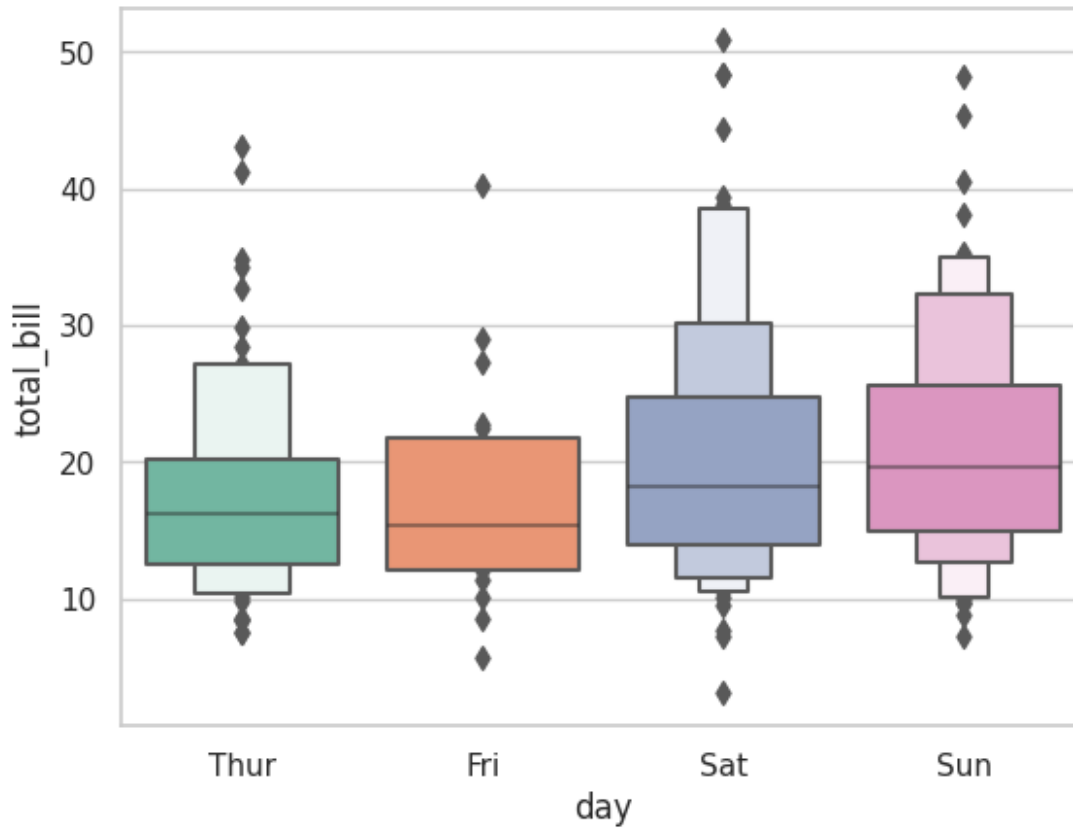
It is an enhanced version of the box plot that provides more detailed information about the distribution.

```
# Set the style of the plots to "whitegrid"
sns.set(style="whitegrid")

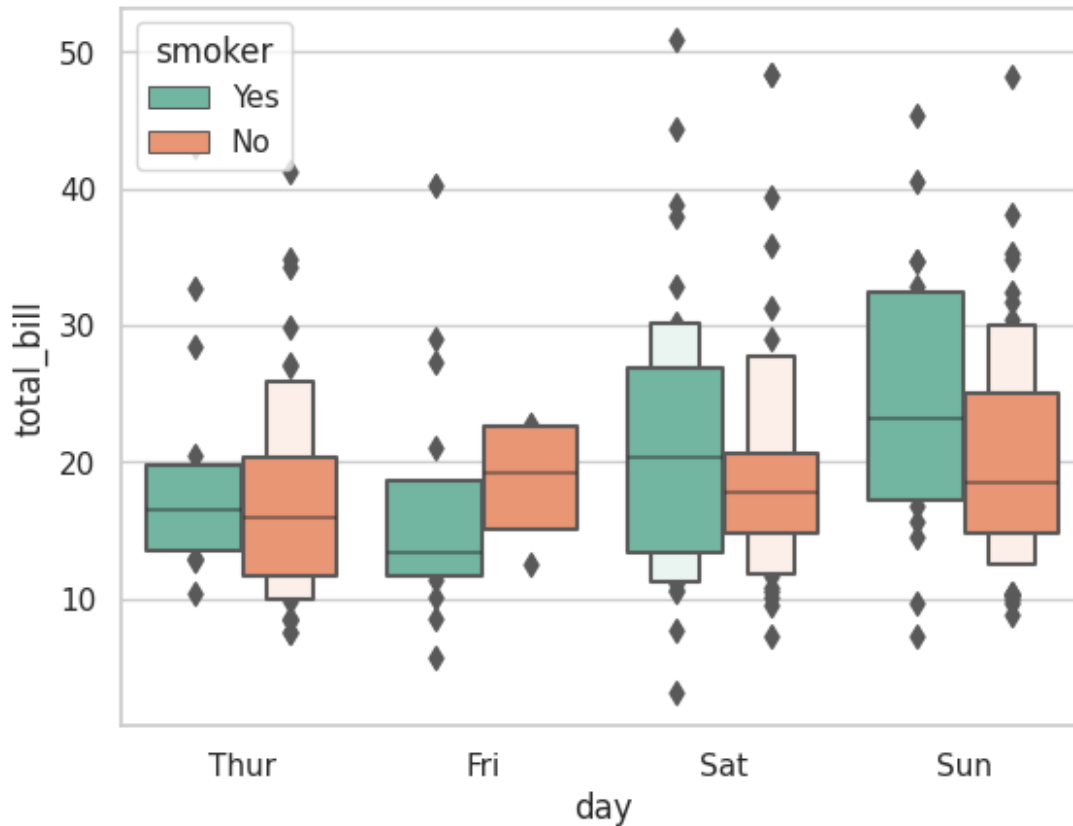
# Boxen plot of "tip" column from the tips dataset
ax=sns.boxenplot(x="tip",data=tips)
```



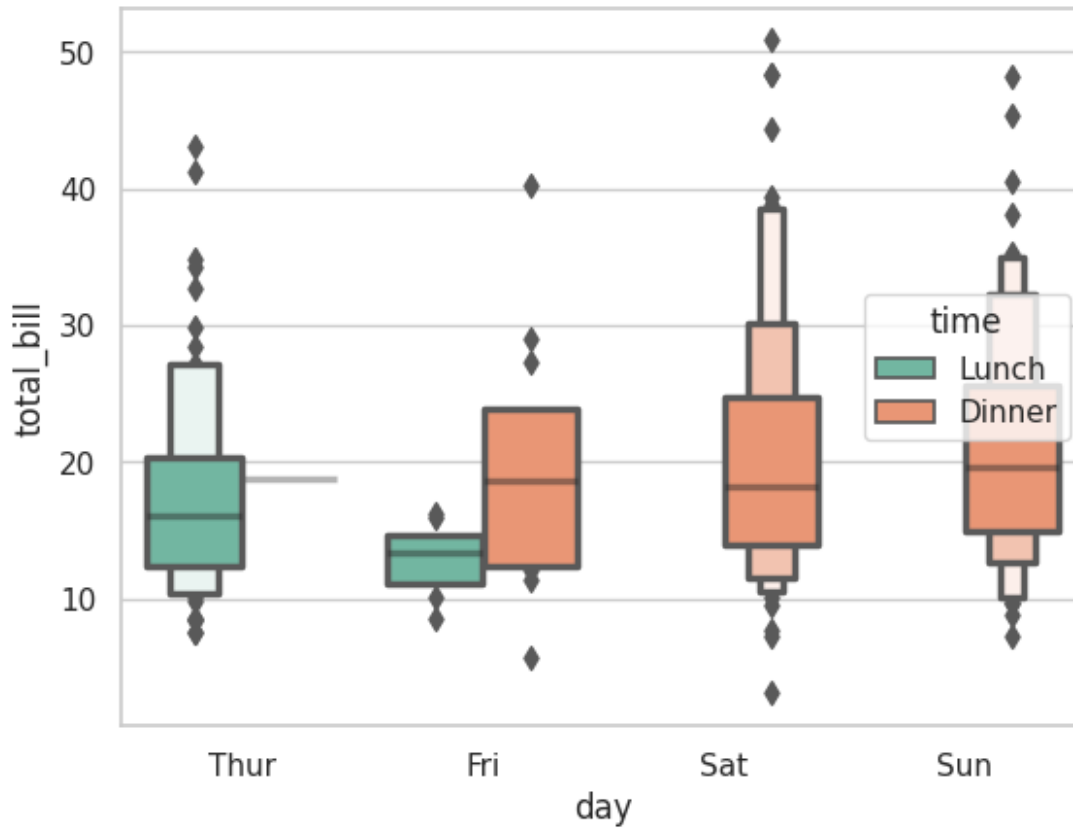
```
# Boxen plot of "total_bill" by "day" from the tips dataset with
palette "Set2"
ax = sns.boxenplot(x="day", y="total_bill",data=tips,palette="Set2")
plt.show()
```



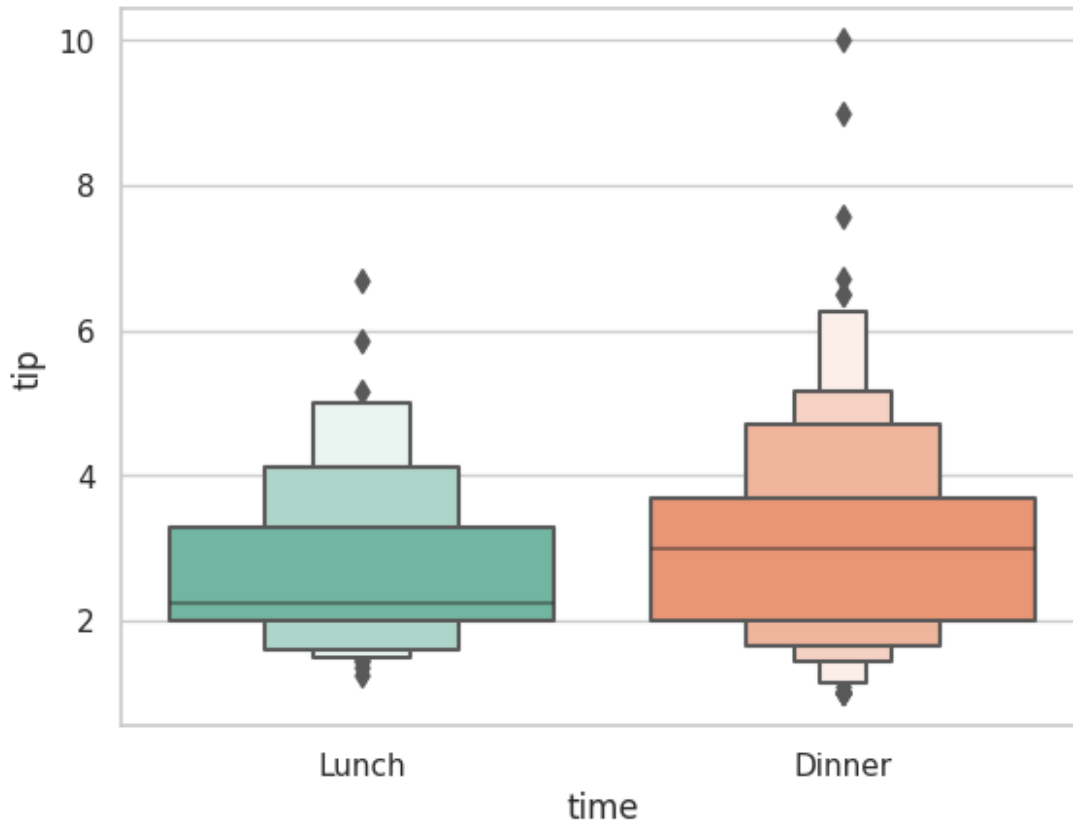
```
# Boxen plot of "total_bill" by "day" from the tips dataset with hue  
by "smoker" and palette "Set2"  
ax = sns.boxenplot(x="day",  
y="total_bill",hue="smoker",data=tips,palette="Set2")  
plt.show()
```



```
# Boxen plot of "total_bill" by "day" from the tips dataset with hue
# by "time", palette "Set2", and increased linewidth
ax = sns.boxenplot(x="day",
y="total_bill",hue="time",data=tips,palette="Set2",linewidth=2.5)
plt.show()
```

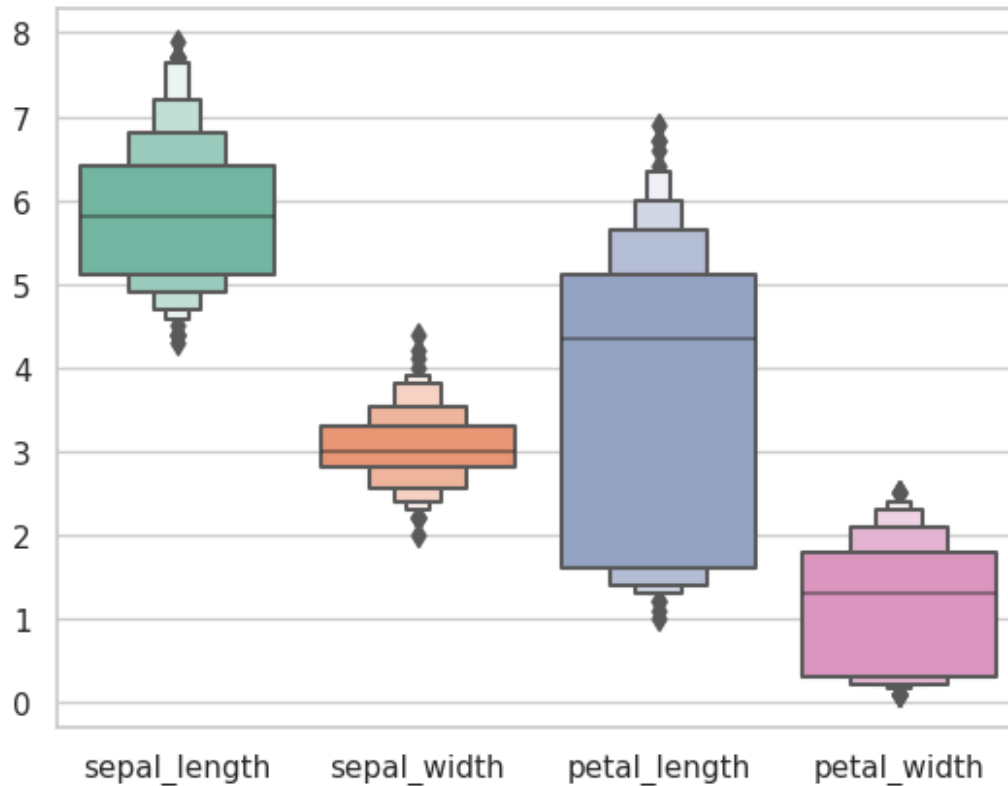


```
# Boxen plot of "tip" by "time" from the tips dataset with palette  
"Set2"  
ax = sns.boxenplot(x="time", y="tip", data=tips, palette="Set2")  
plt.show()
```

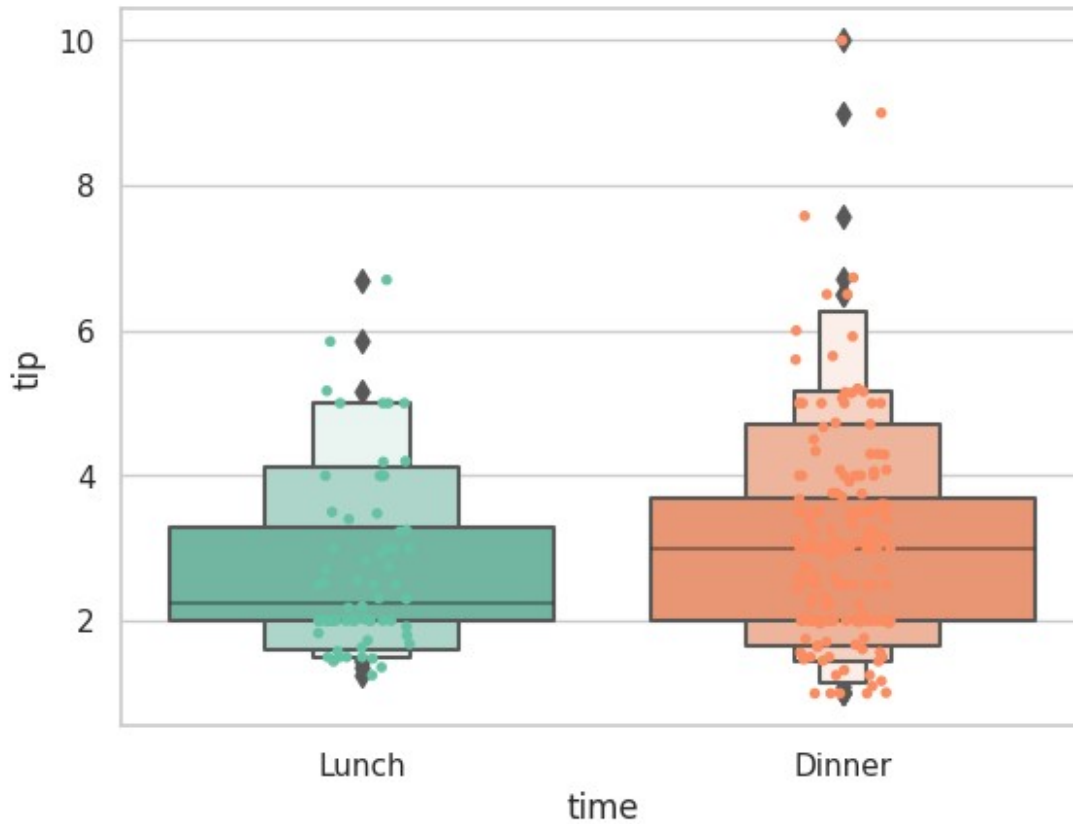


```
# load the dataset from seaborn
iris=sns.load_dataset("iris")

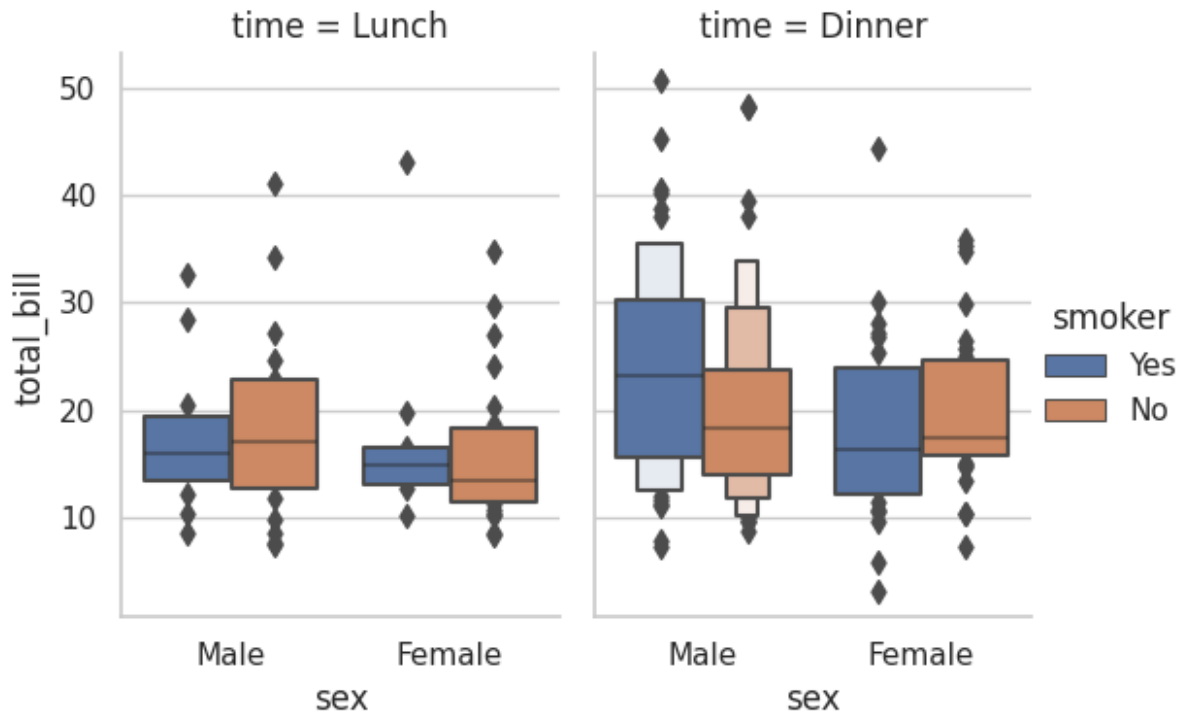
# Boxen plot of the entire iris dataset with palette "Set2"
ax = sns.boxenplot(data=iris,palette="Set2")
plt.show()
```

```
# Boxen plot of "tip" by "time" from the tips dataset with palette
"Set2"
ax = sns.boxenplot(x="time", y="tip", data=tips, palette="Set2")
# Strip plot of "tip" by "time" from the tips dataset with palette
"Set2", jittered points, and gray color
ax=sns.stripplot(x="time",
y="tip", data=tips, palette="Set2", size=4, jitter=True, color="gray")
plt.show()
```



```
# Categorical boxen plot of "total_bill" by "sex" with hue by
"smoker", multiple subplots based on "time", and customized height and
aspect ratios
g=sns.catplot(x="sex",y="total_bill",hue="smoker",col="time",data=tips
,kind="boxen",height=4,aspect=.7)
plt.show()
```



```
import seaborn as sns
import matplotlib.pyplot as plt

# Set the style of the plots to "whitegrid"
sns.set(style="whitegrid")

# Load the tips dataset from seaborn
tips = sns.load_dataset("tips")

# Create a figure with 2 subplots arranged in 2 rows and 4 columns
fig, axes = plt.subplots(2, 4, figsize=(16, 8))

# Plot 1 - Boxen plot of "tip" column from the tips dataset
sns.boxenplot(x="tip", data=tips, ax=axes[0, 0])
axes[0, 0].set_title("Tip")

# Plot 2 - Boxen plot of "total_bill" by "day" from the tips dataset
sns.boxenplot(x="day", y="total_bill", data=tips, palette="Set2",
ax=axes[0, 1])
axes[0, 1].set_title("Total Bill by Day")

# Plot 3 - Boxen plot of "total_bill" by "day" with hue by "smoker"
from the tips dataset
sns.boxenplot(x="day", y="total_bill", hue="smoker", data=tips,
palette="Set2", ax=axes[0, 2])
axes[0, 2].set_title("Total Bill by Day (Smoker)")
```

```
# Plot 4 - Boxen plot of "total_bill" by "day" with hue by "time" from
the tips dataset
sns.boxenplot(x="day", y="total_bill", hue="time", data=tips,
palette="Set2", linewidth=2.5, ax=axes[0, 3])
axes[0, 3].set_title("Total Bill by Day (Time)")

# Plot 5 - Boxen plot of "tip" by "time" from the tips dataset
sns.boxenplot(x="time", y="tip", data=tips, palette="Set2", ax=axes[1,
0])
axes[1, 0].set_title("Tip by Time")

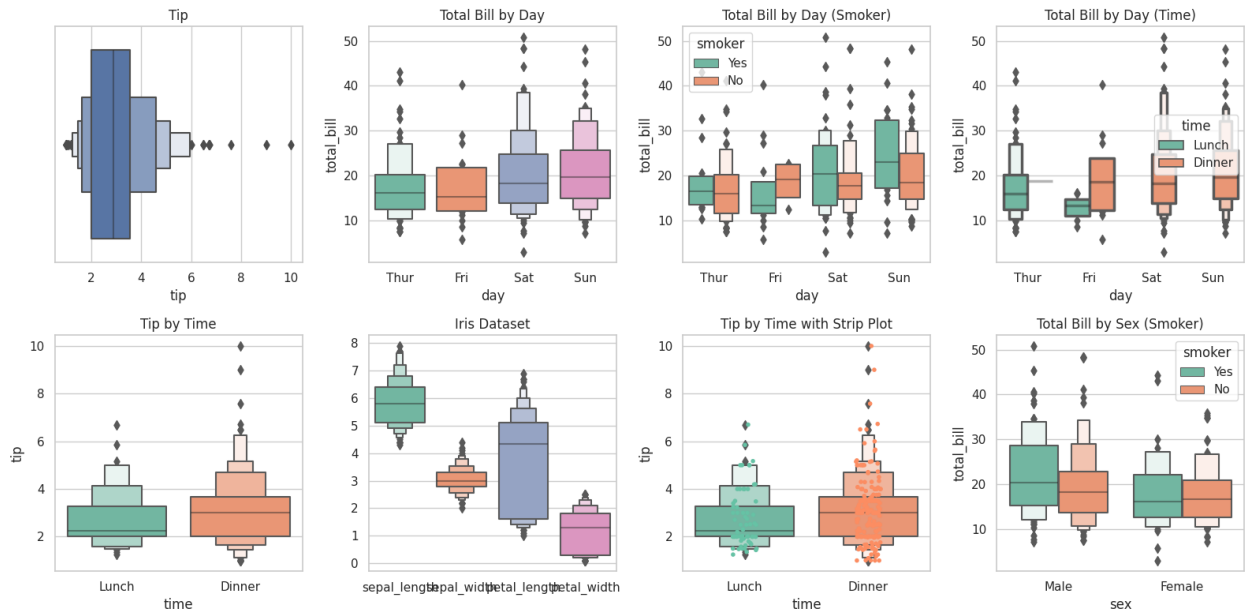
# Plot 6 - Boxen plot of the entire iris dataset
iris = sns.load_dataset("iris")
sns.boxenplot(data=iris, palette="Set2", ax=axes[1, 1])
axes[1, 1].set_title("Iris Dataset")

# Plot 7 - Boxen plot of "tip" by "time" with strip plot overlay
sns.boxenplot(x="time", y="tip", data=tips, palette="Set2", ax=axes[1,
2])
sns.stripplot(x="time", y="tip", data=tips, palette="Set2", size=4,
jitter=True, color="gray", ax=axes[1, 2])
axes[1, 2].set_title("Tip by Time with Strip Plot")

# Plot 8 - Boxen plot of "total_bill" by "sex" with hue by "smoker"
sns.boxenplot(x="sex", y="total_bill", hue="smoker", data=tips,
palette="Set2", ax=axes[1, 3])
axes[1, 3].set_title("Total Bill by Sex (Smoker)")

# Adjust the spacing between subplots
plt.tight_layout()

# Display the plots
plt.show()
```

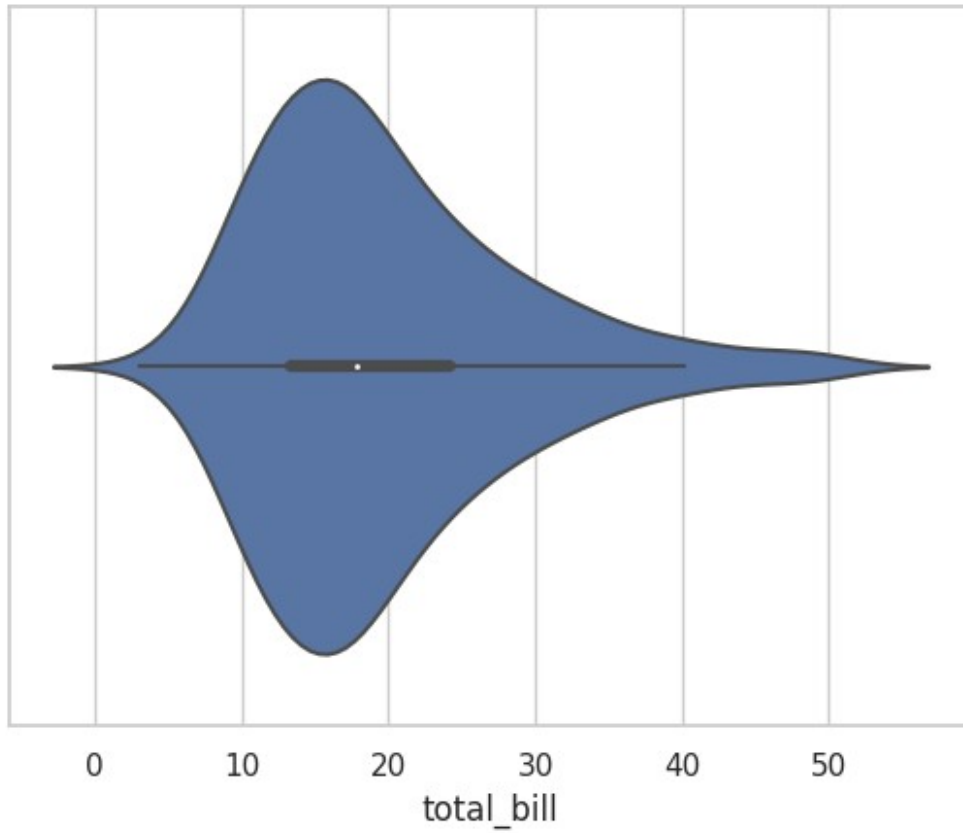


Violin Plot

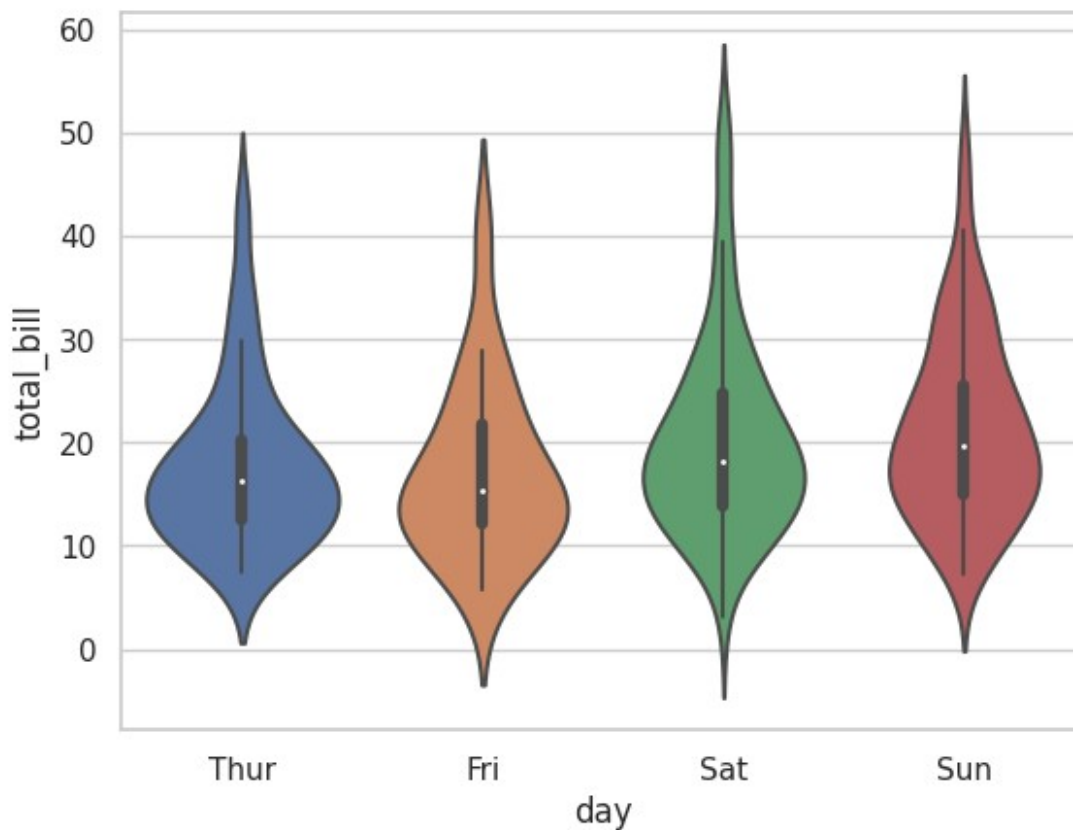
The violin plot is a type of categorical plot in seaborn that combines a box plot and a kernel density estimation plot.

It is used to visualize the distribution of a numerical variable within different categories.

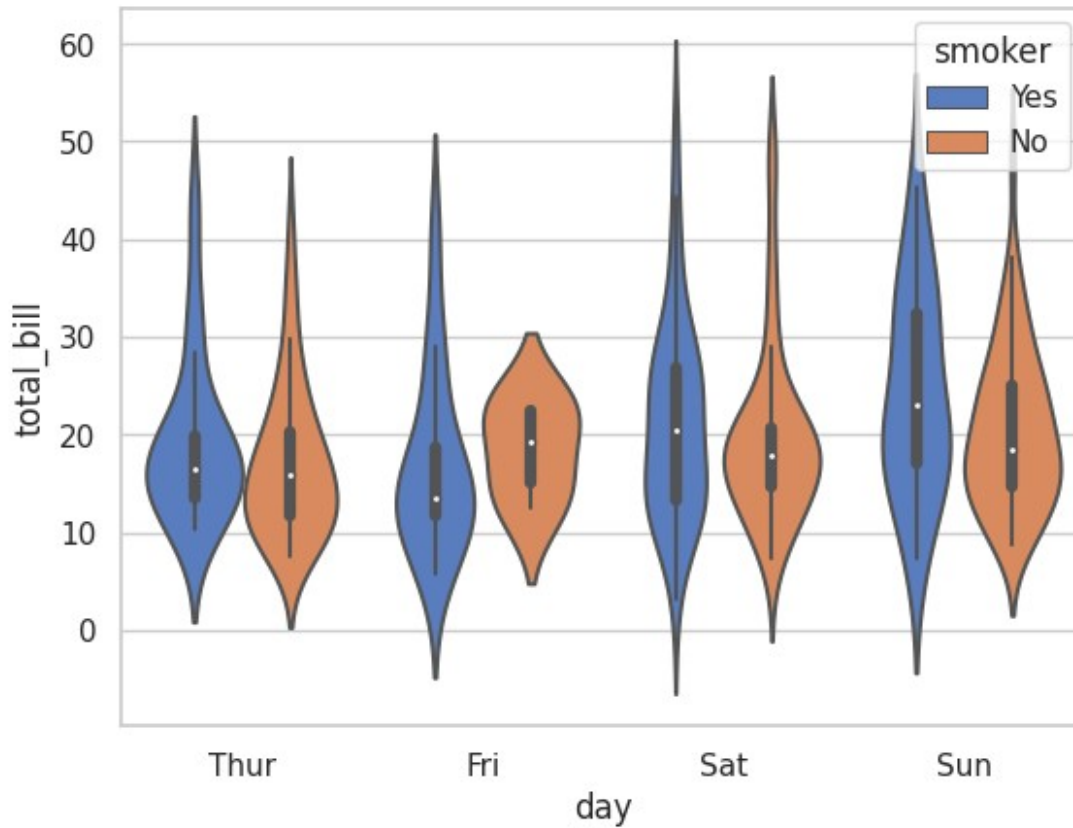
```
# Violin Plot 1 - Violin plot of "total_bill" from the tips dataset
ax= sns.violinplot(x= tips["total_bill"])
```



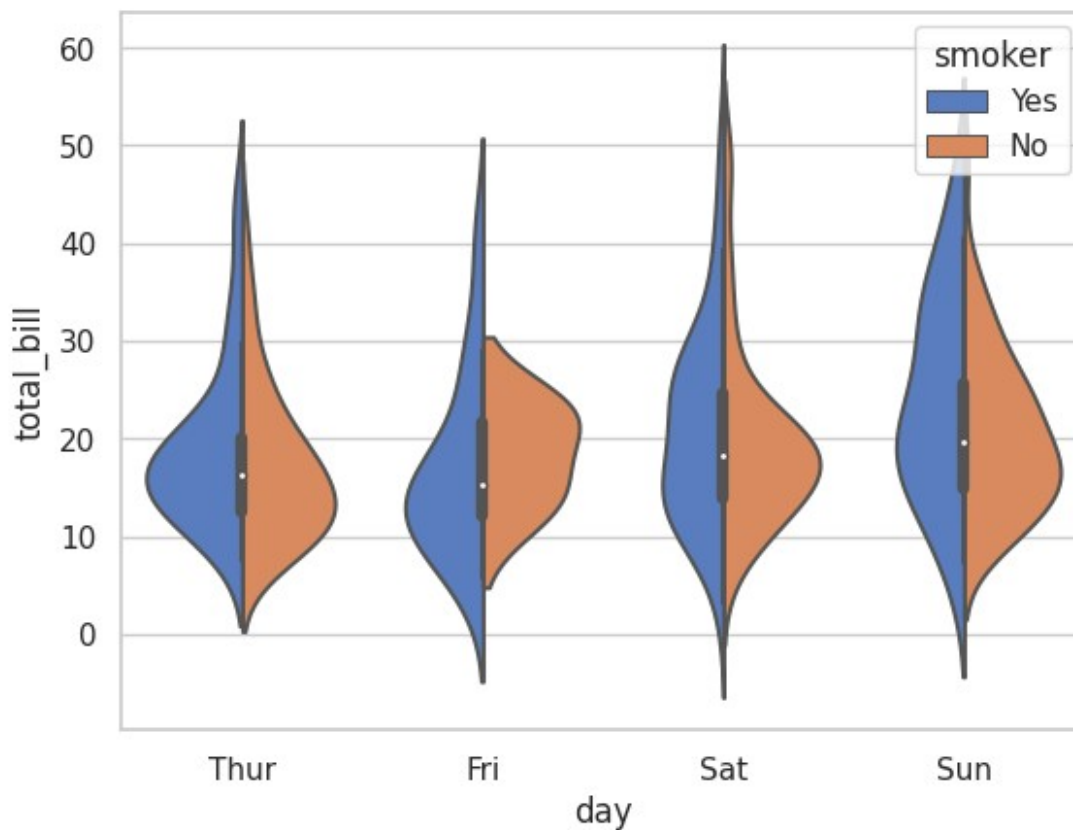
```
# Violin Plot 2 - Violin plot of "total_bill" by "day" from the tips dataset
ax= sns.violinplot(x= "day",y="total_bill",data=tips)
```



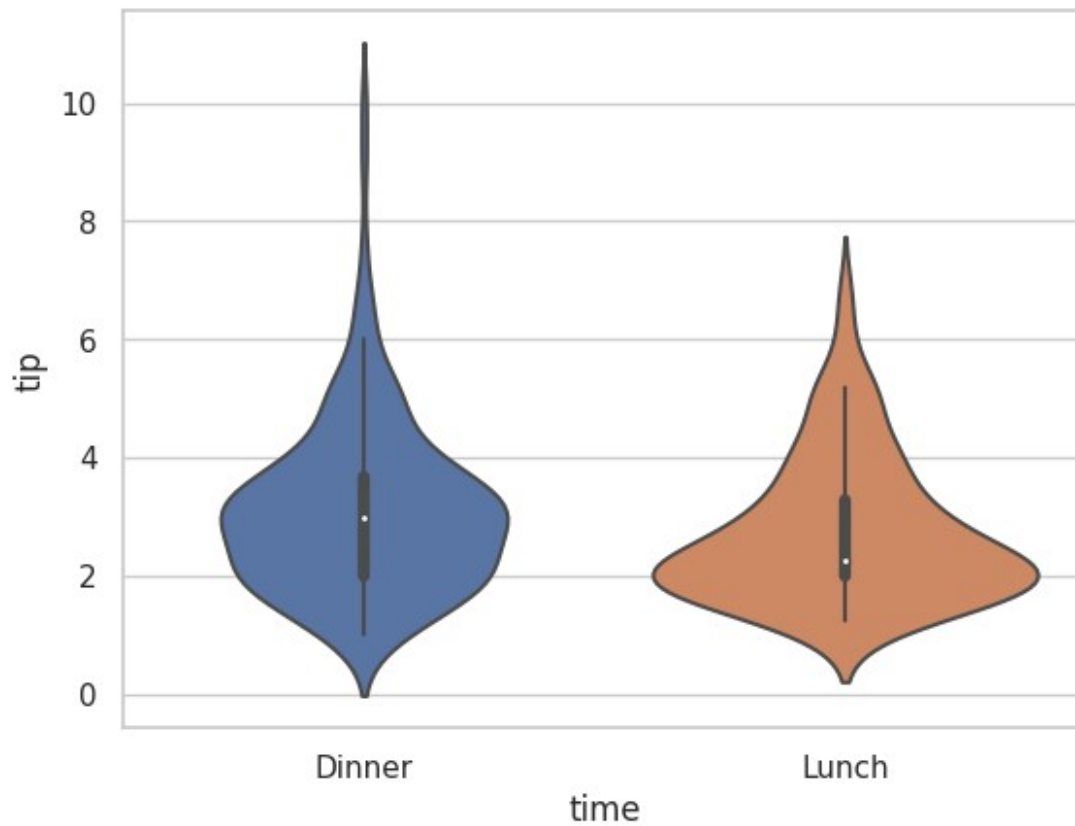
```
# Violin Plot 3 - Violin plot of "total_bill" by "day" with hue by  
"smoker" from the tips dataset  
ax= sns.violinplot(x=  
"day",y="total_bill",data=tips,hue="smoker",palette="muted")
```



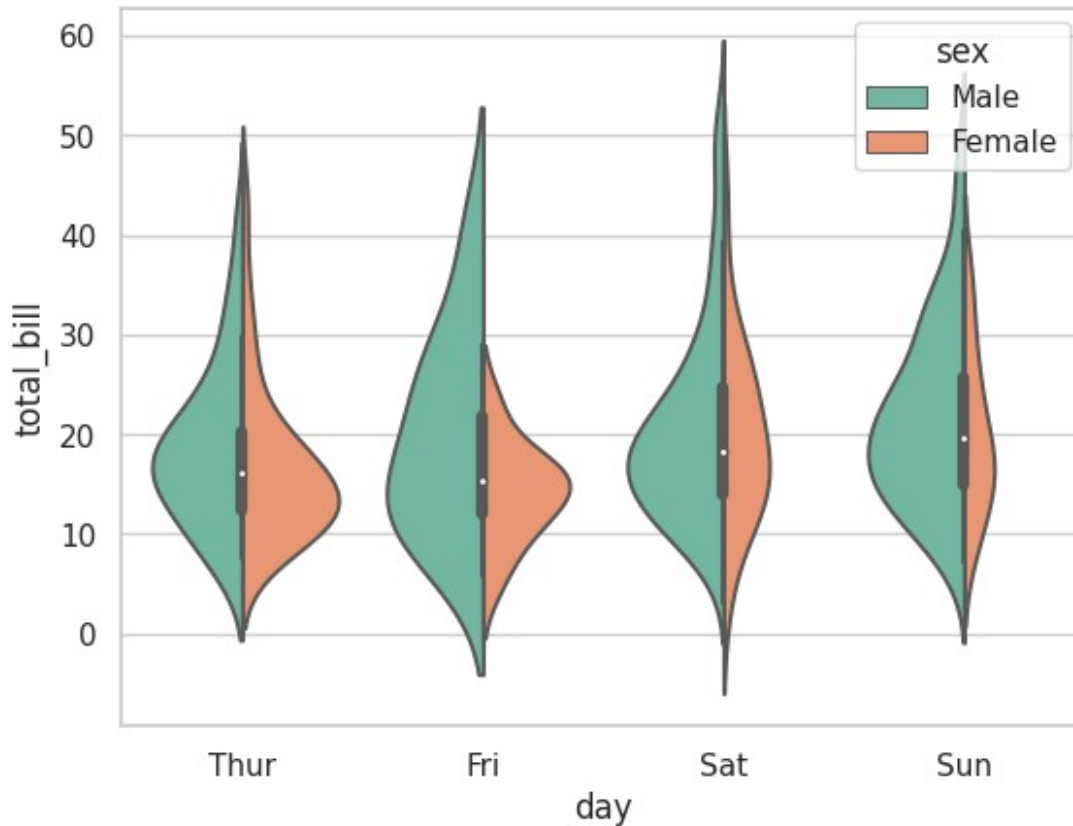
```
# Violin Plot 4 - Violin plot of "total_bill" by "day" with hue by  
"smoker" and split by hue from the tips dataset  
ax= sns.violinplot(x=  
"day",y="total_bill",data=tips,hue="smoker",split=True,palette="muted"  
)
```

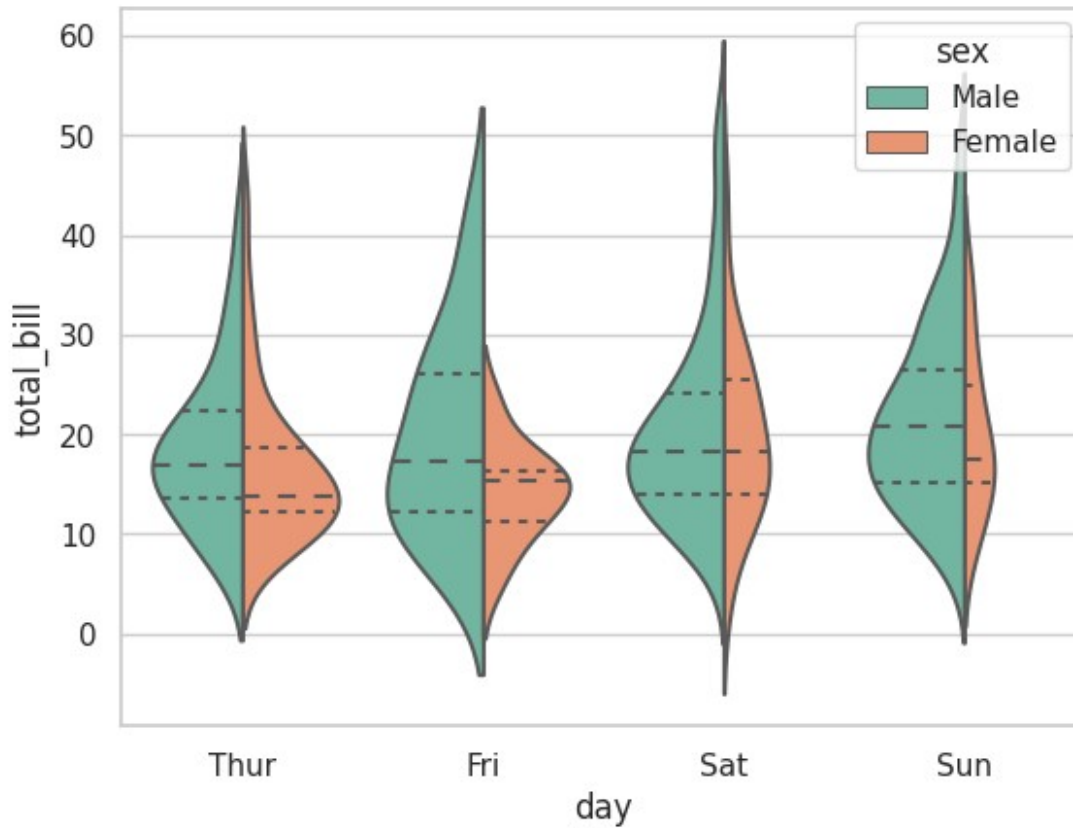
```
# Violin Plot 5 - Violin plot of "tip" by "time" from the tips dataset  
with custom order of categories  
ax= sns.violinplot(x=  
"time",y="tip",data=tips,order=["Dinner","Lunch"])
```



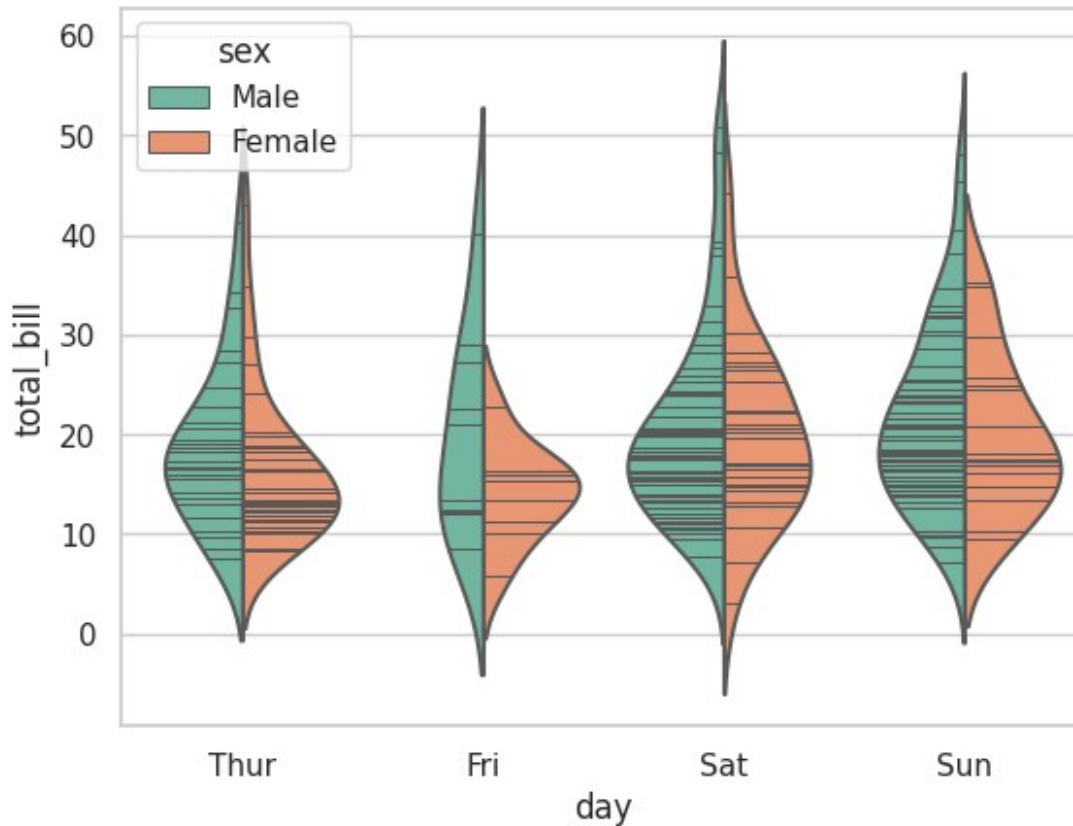
```
# Violin Plot 6 - Violin plot of "total_bill" by "day" with hue by  
"sex" and split by hue, using count scale  
ax= sns.violinplot(x=  
"day",y="total_bill",data=tips,hue="sex",split=True,palette="Set2",sca  
le="count")
```



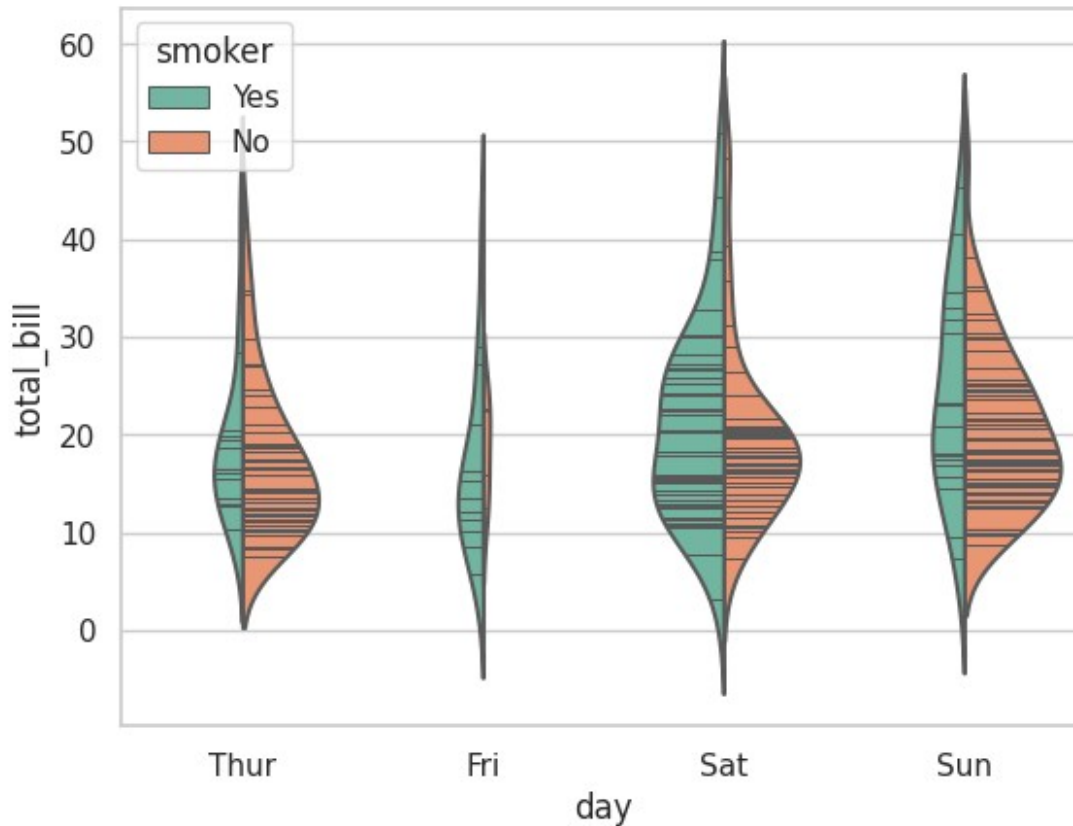
```
# Violin Plot 7 - Violin plot of "total_bill" by "day" with hue by  
"sex" and split by hue, using count scale and quartile inner  
representation  
ax= sns.violinplot(x=  
"day",y="total_bill",data=tips,hue="sex",split=True,palette="Set2",sca  
le="count",inner="quartile")
```



```
# Violin Plot 8 - Violin plot of "total_bill" by "day" with hue by  
"sex" and split by hue, using stick inner representation  
ax= sns.violinplot(x=  
"day",y="total_bill",data=tips,hue="sex",split=True,palette="Set2",inn  
er="stick")
```

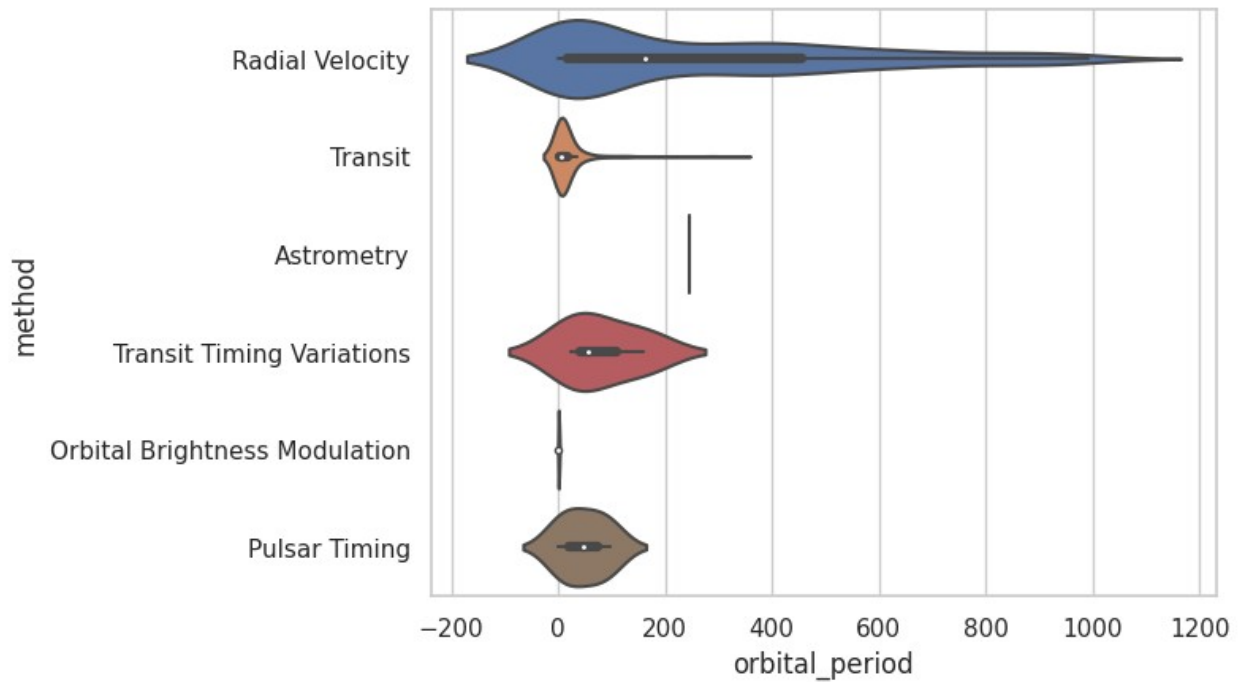


```
# Violin Plot 9 - Violin plot of "total_bill" by "day" with hue by
"smoker" and split by hue, using count scale and stick inner
representation, and disabling hue scaling
ax= sns.violinplot(x=
"day",y="total_bill",data=tips,hue="smoker",split=True,palette="Set2",
scale="count",inner="stick",scale_hue= False)
```

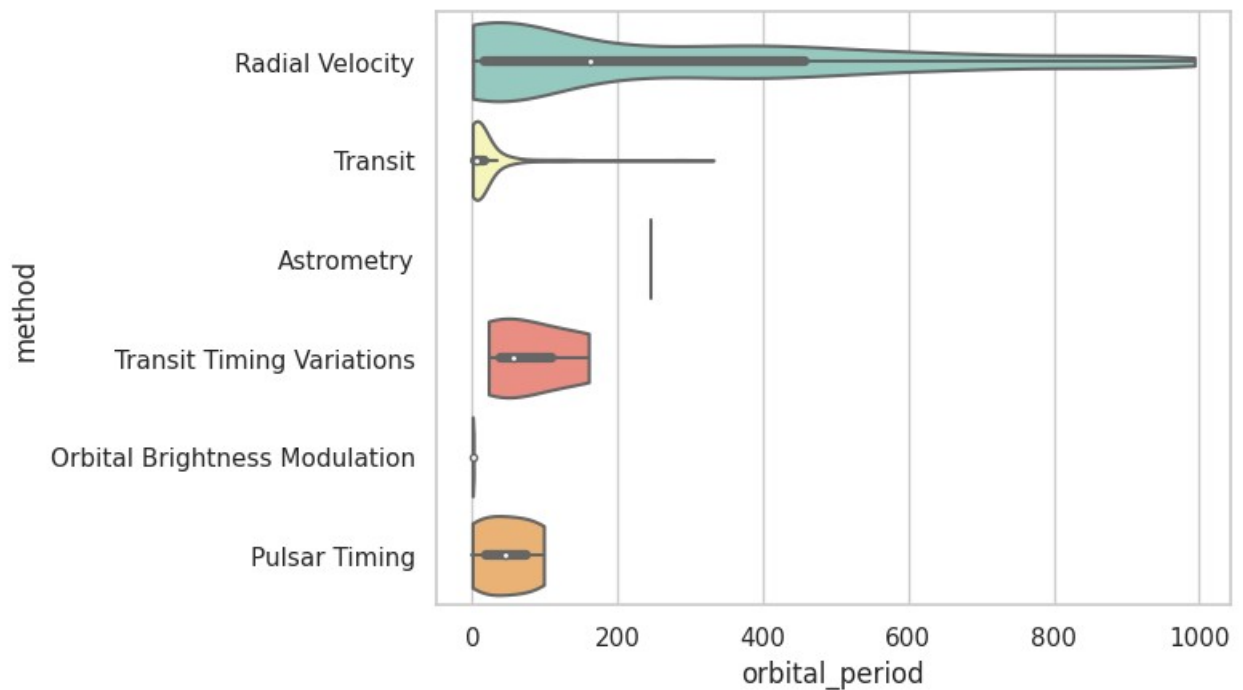


```
# Load the planets dataset from seaborn
planets = sns.load_dataset("planets")

# Violin Plot 10 - Violin plot of "orbital_period" by "method" from
the planets dataset with width scale
ax =
sns.violinplot(x="orbital_period",y="method",data=planets[planets.orbi
tal_period<1000],scale="width")
```



```
# Violin Plot 11 - Violin plot of "orbital_period" by "method" from
the planets dataset with width scale, no truncation, and custom
palette
ax =
sns.violinplot(x="orbital_period",y="method",data=planets[planets.orbi
tal_period<1000],scale="width",cut=0,palette="Set3")
```



```

import seaborn as sns
import matplotlib.pyplot as plt

# Set the style of the plots to "whitegrid"
sns.set(style="whitegrid")

# Load the tips dataset from seaborn
tips = sns.load_dataset("tips")

# Load the planets dataset from seaborn
planets = sns.load_dataset("planets")

# Create a figure with 4 subplots arranged in 2 rows and 4 columns
fig, axes = plt.subplots(3, 4, figsize=(16, 12))

# Plot 1 - Violin plot of "total_bill" from the tips dataset
ax = sns.violinplot(x=tips["total_bill"], ax=axes[0, 0])
ax.set_title("Total Bill")

# Plot 2 - Violin plot of "total_bill" by "day" from the tips dataset
ax = sns.violinplot(x="day", y="total_bill", data=tips, ax=axes[0, 1])
ax.set_title("Total Bill by Day")

# Plot 3 - Violin plot of "total_bill" by "day" with hue by "smoker"
from the tips dataset
ax = sns.violinplot(x="day", y="total_bill", hue="smoker", data=tips,
palette="muted", ax=axes[0, 2])
ax.set_title("Total Bill by Day (Smoker)")

# Plot 4 - Violin plot of "total_bill" by "day" with hue by "smoker"
and split by hue from the tips dataset
ax = sns.violinplot(x="day", y="total_bill", hue="smoker", data=tips,
palette="muted", split=True, ax=axes[0, 3])
ax.set_title("Total Bill by Day (Smoker - Split)")

# Plot 5 - Violin plot of "time" by "tip" from the tips dataset with
custom order of categories
ax = sns.violinplot(x="time", y="tip", data=tips, order=["Dinner",
"Lunch"], ax=axes[1, 0])
ax.set_title("Tip by Time")

# Plot 6 - Violin plot of "total_bill" by "day" with hue by "sex" and
split by hue, using count scale
ax = sns.violinplot(x="day", y="total_bill", hue="sex", data=tips,
palette="Set2", split=True, scale="count", ax=axes[1, 1])
ax.set_title("Total Bill by Day (Sex - Split)")

# Plot 7 - Violin plot of "total_bill" by "day" with hue by "sex" and
split by hue, using count scale and quartile inner representation
ax = sns.violinplot(x="day", y="total_bill", hue="sex", data=tips,
palette="Set2", split=True, scale="count", inner="quartile",

```



```

ax=axes[1, 2])
ax.set_title("Total Bill by Day (Sex - Split with Quartiles)")

# Plot 8 - Violin plot of "total_bill" by "day" with hue by "sex" and
split by hue, using stick inner representation
ax = sns.violinplot(x="day", y="total_bill", hue="sex", data=tips,
palette="Set2", split=True, inner="stick", ax=axes[1, 3])
ax.set_title("Total Bill by Day (Sex - Split with Sticks)")

# Plot 9 - Violin plot of "time" by "tip" from the tips dataset with
custom color
ax = sns.violinplot(x="time", y="tip", data=tips, color="#bb3f3f",
ax=axes[2, 0])
ax.set_title("Tip by Time (Custom Color)")

# Plot 10 - Violin plot of "time" by "tip" from the tips dataset with
hue by "smoker" and split by hue
ax = sns.violinplot(x="time", y="tip", hue="smoker", data=tips,
palette="Set2", split=True, ax=axes[2, 1])
ax.set_title("Tip by Time (Smoker - Split)")

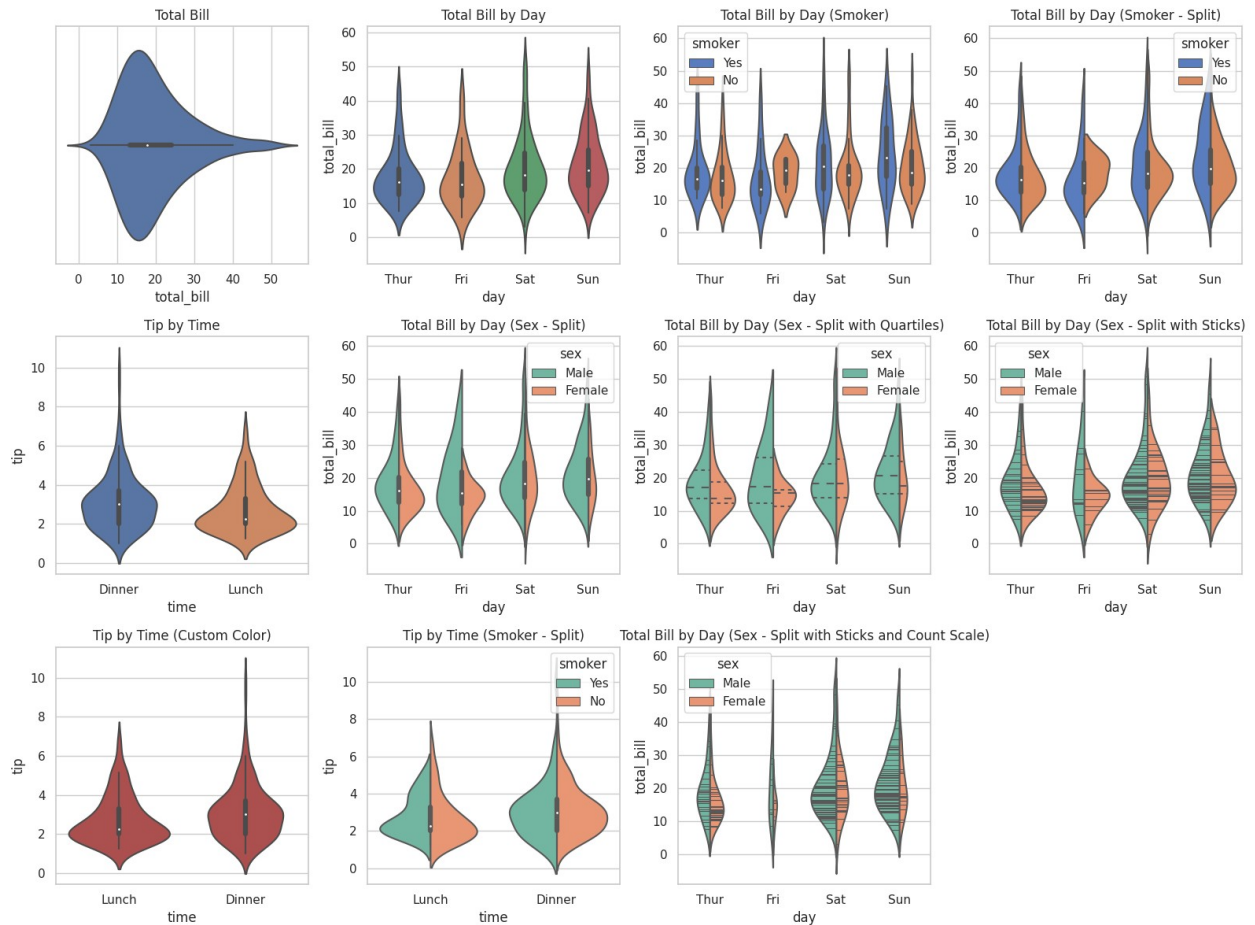
# Plot 11 - Violin plot of "total_bill" by "day" with hue by "sex" and
split by hue, using count scale, inner stick representation, and
disabling hue scaling
ax = sns.violinplot(x="day", y="total_bill", hue="sex", data=tips,
palette="Set2", split=True, scale="count", inner="stick",
scale_hue=False, ax=axes[2, 2])
ax.set_title("Total Bill by Day (Sex - Split with Sticks and Count
Scale)")

# Remove empty subplots
fig.delaxes(axes[2, 3])

# Adjust the spacing between subplots
plt.tight_layout()

# Display the plots
plt.show()

```

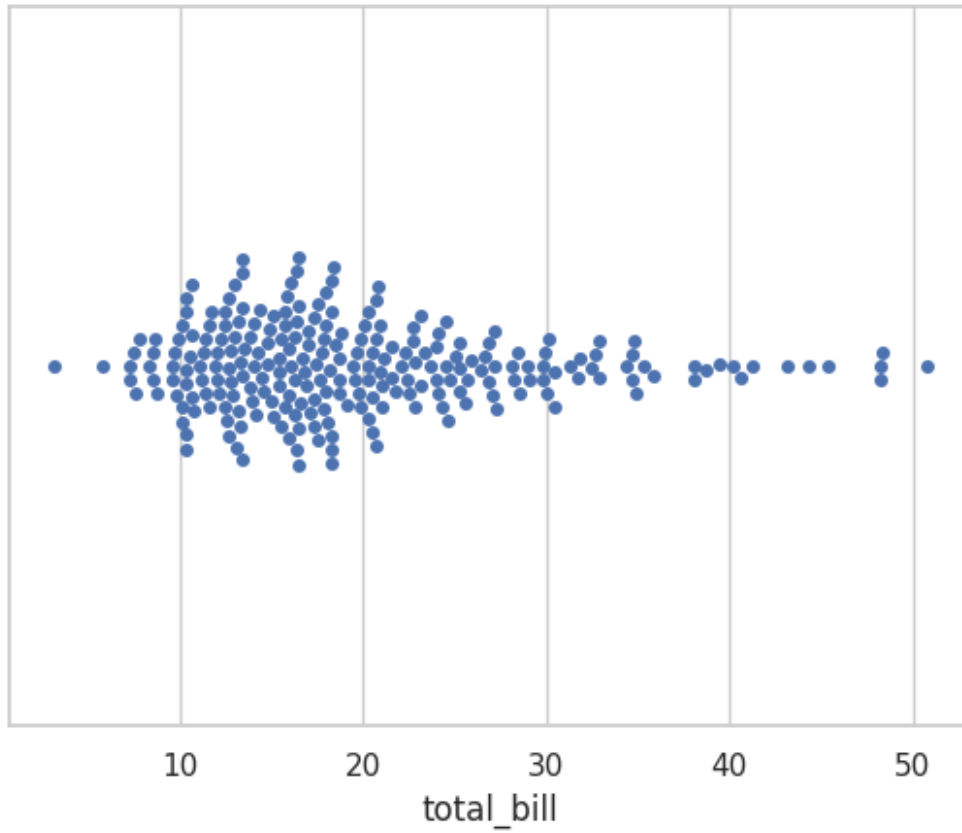


Swarm Plot

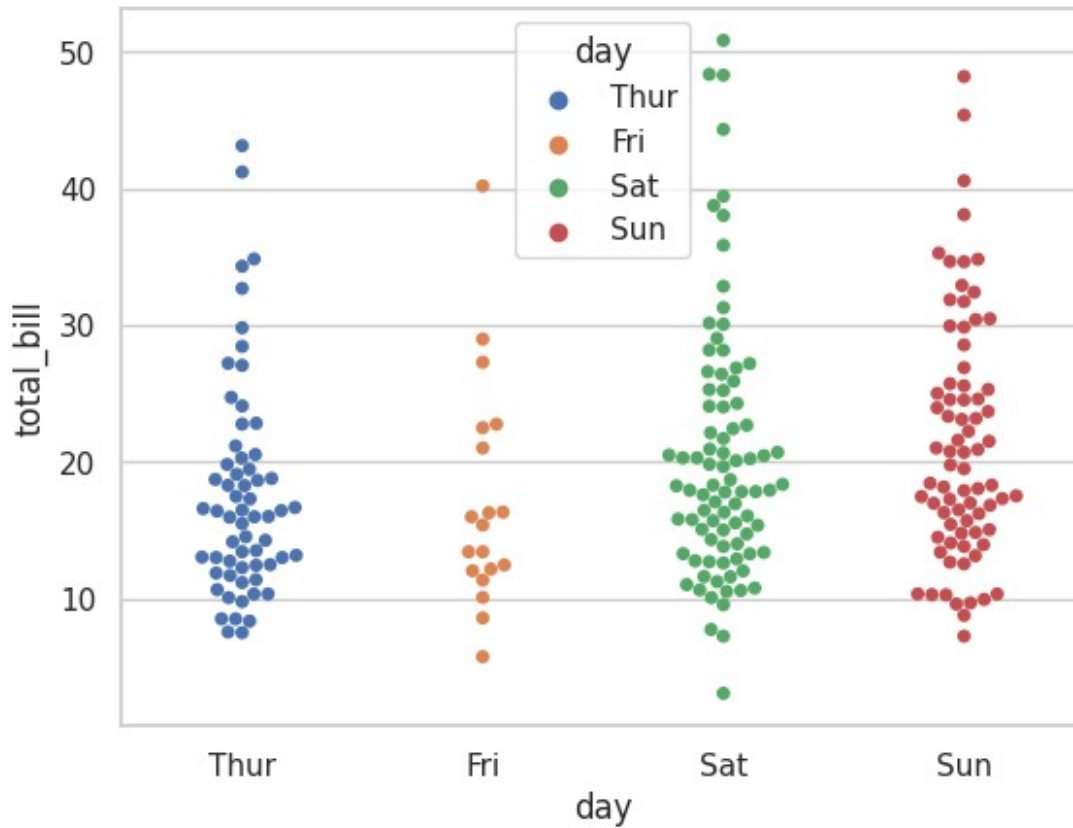
The swarm plot is a type of categorical plot in seaborn that displays the distribution of a numerical variable within different categories by plotting individual data points.

It is particularly useful when dealing with smaller datasets or when you want to show the distribution of data points more clearly.

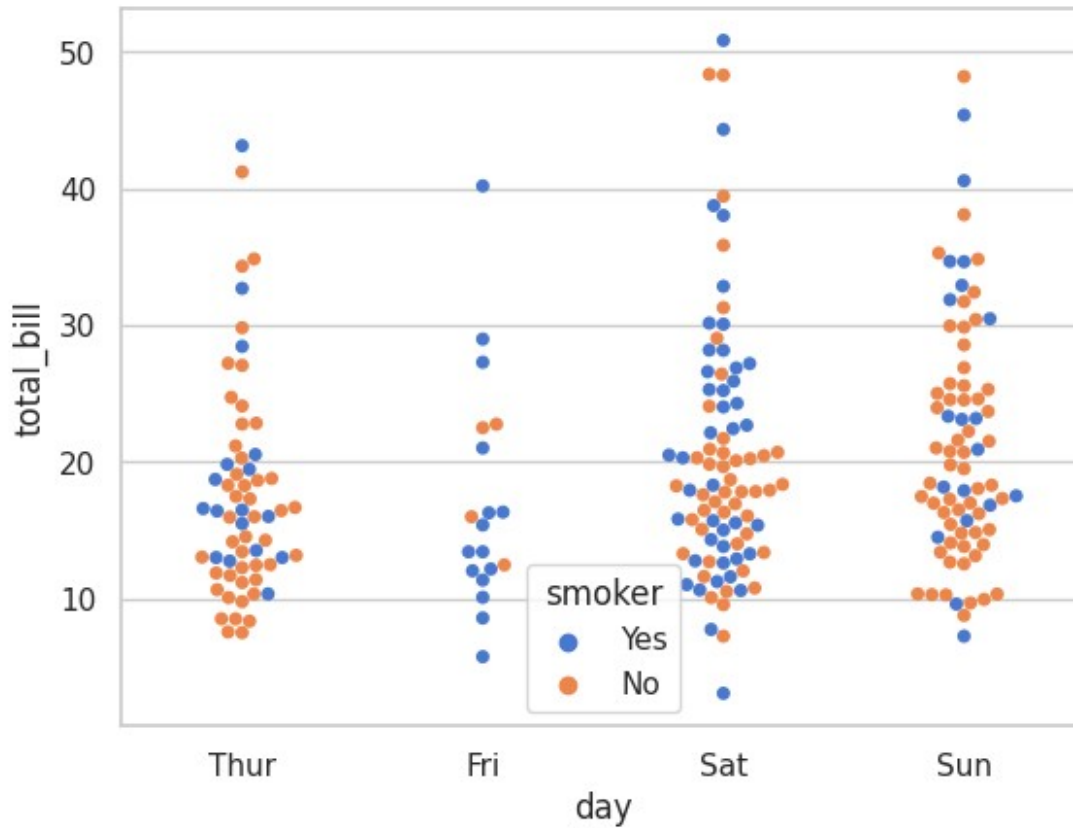
```
# Plot 1 - Swarm plot of "total_bill" from the tips dataset
ax= sns.swarmplot(x= tips["total_bill"])
```



```
# Plot 2 - Swarm plot of "total_bill" by "day" from the tips dataset  
with hue by "day"  
ax= sns.swarmplot(x= "day",y="total_bill",data=tips,hue="day")
```

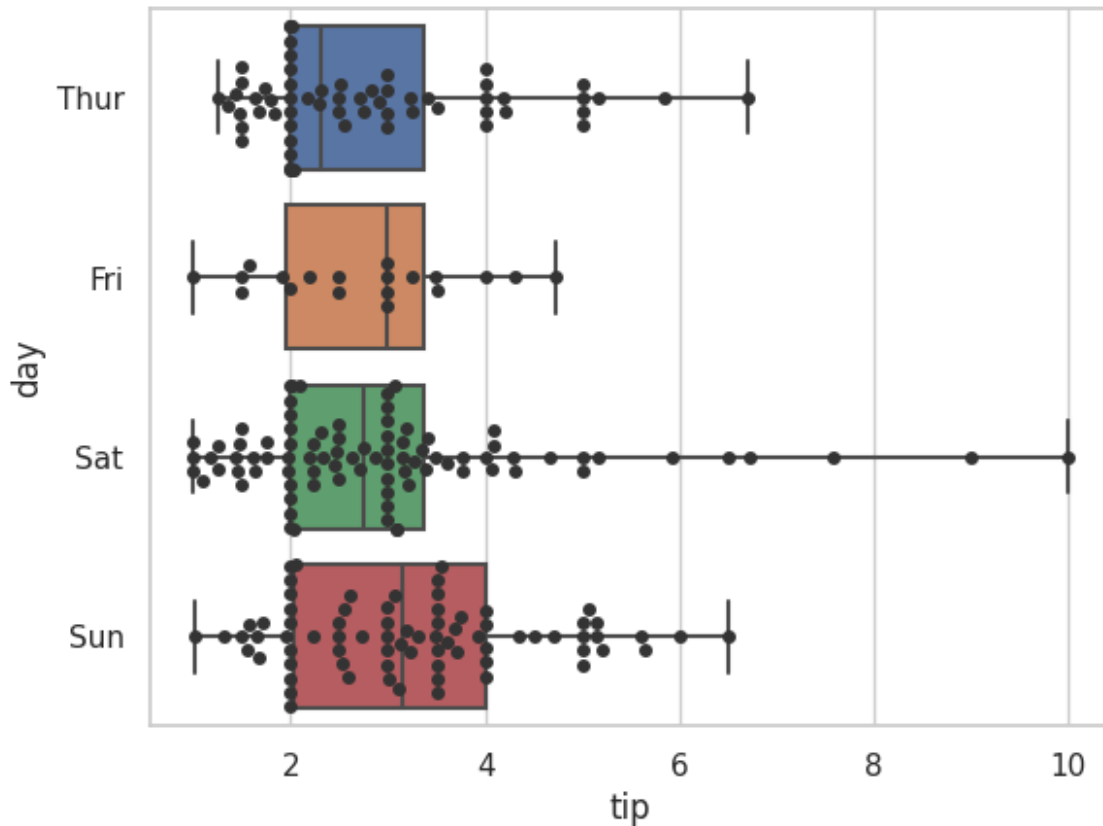


```
# Plot 3 - Swarm plot of "total_bill" by "day" from the tips dataset
with hue by "smoker" and custom palette
ax= sns.swarmplot(x=
"day",y="total_bill",data=tips,hue="smoker",palette="muted")
```

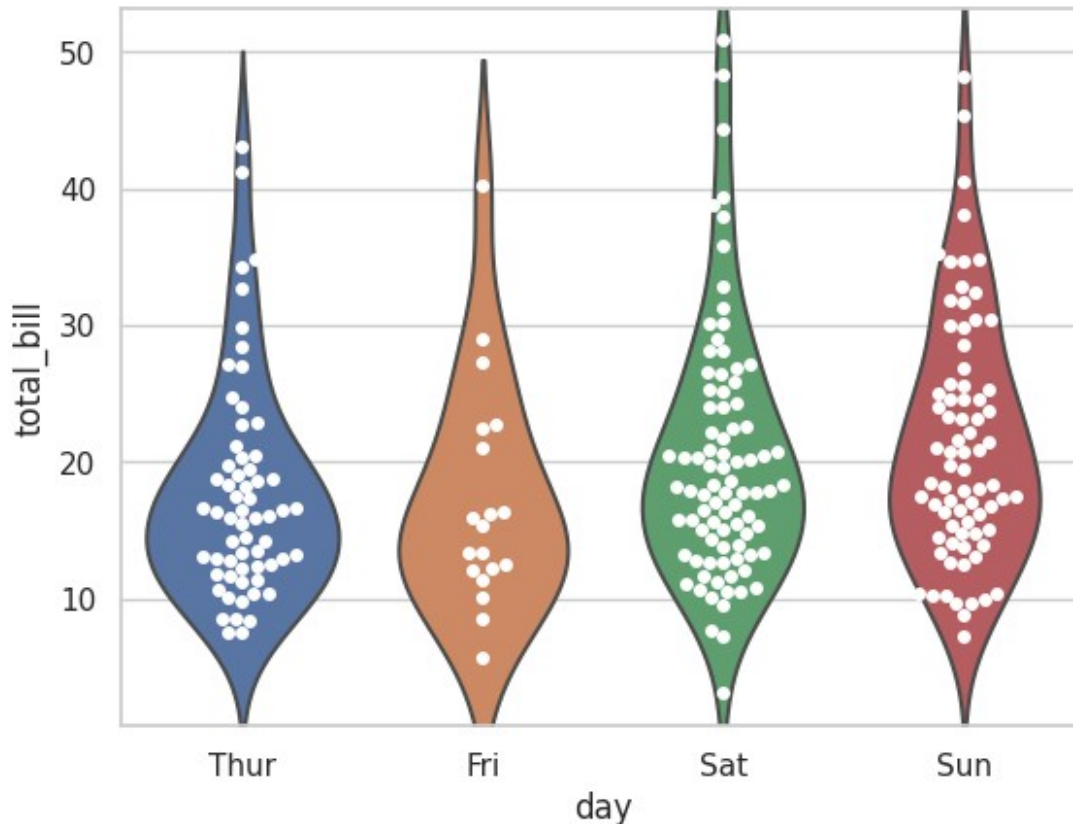


Plot 4 - Box plot of "tip" by "day" from the tips dataset with unlimited whiskers

```
ax= sns.boxplot(x="tip",y="day",data=tips,whis=np.inf)
ax = sns.swarmplot(x="tip",y="day",data=tips,color=".2")
```



```
# Plot 5 - Violin plot of "total_bill" by "day" from the tips dataset
with swarmplot on top of it
ax= sns.violinplot(x="day",y="total_bill",data=tips,inner=None)
ax =
sns.swarmplot(x="day",y="total_bill",data=tips,color="white",edgecolor
="gray")
```



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

# Set the style of the plots
sns.set(style="whitegrid")

# Create a figure and axes
fig, axes = plt.subplots(2, 2, figsize=(12, 8))

# Plot 1 - Swarm plot of "total_bill" from the tips dataset
ax = sns.swarmplot(x=tips["total_bill"], ax=axes[0, 0])
ax.set_title("Total Bill")

# Plot 2 - Swarm plot of "total_bill" by "day" from the tips dataset
# with hue by "day"
ax = sns.swarmplot(x="day", y="total_bill", data=tips, hue="day",
ax=axes[0, 1])
ax.set_title("Total Bill by Day (Hue by Day)")

# Plot 3 - Swarm plot of "total_bill" by "day" from the tips dataset
# with hue by "smoker" and custom palette
ax = sns.swarmplot(x="day", y="total_bill", data=tips, hue="smoker",
palette="muted", ax=axes[1, 0])
```

```

ax.set_title("Total Bill by Day (Hue by Smoker)")

# Plot 4 - Box plot of "tip" by "day" from the tips dataset with
unlimited whiskers
ax = sns.boxplot(x="tip", y="day", data=tips, whis=np.inf, ax=axes[1,
1])
strip_ax = ax.twinx() # Create a new axes object for the strip plot
sns.stripplot(x="tip", y="day", data=tips, color=".2", ax=strip_ax,
alpha=0.5)
strip_ax.set_ylabel("") # Remove the y-label from the strip plot axes
ax.set_title("Tip by Day (Box Plot with Strip)")

# Adjust the spacing between subplots
plt.tight_layout()

# Show the plots
plt.show()

```

