seaborn

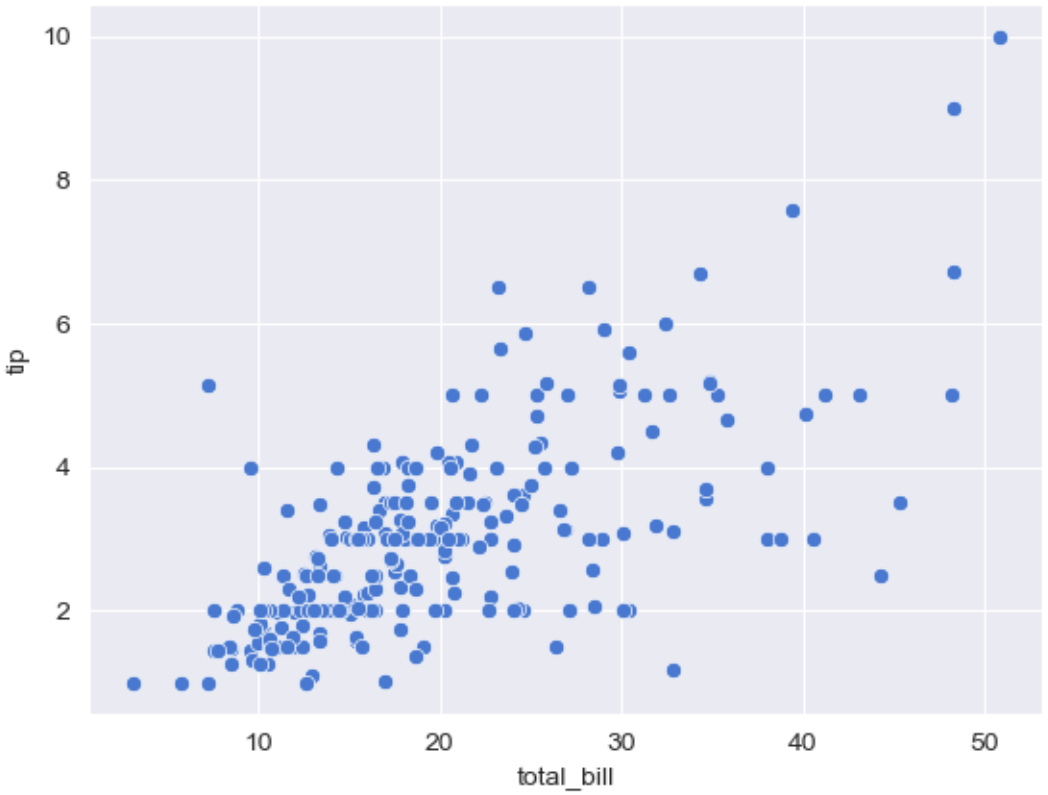
June 16, 2024

[71]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
#sns.set_style('stylename') designs colors of fig,axes: Possible stylenames:␣
 ↪white,dark,whitegrid,darkgrid,ticks
#sns.set_palette('palettename') designs colors of plots: deep: Good all-purpose␣
 ↪palette with distinct colors, muted: Softer colors reduce strain and are␣
 ↪good for detailed presentations., colorblind: Ensures accessibility for␣
 ↪viewers with color vision deficiencies., cubehelix: Perceptually uniform,␣
 ↪ideal for scientific data where precise color differentiation is important.,␣
 ↪Blues: Easy-to-interpret sequential palette, great for heatmaps or gradient␣
 ↪data., coolwarm: Diverging palette that effectively shows contrasts around a␣
 ↪central value.
#sns.set_context('name') changes scalling names: paper,notebook,talk,poster␣
 ↪default is paper
#sns.set(font_scale=1.0) sets font size
```

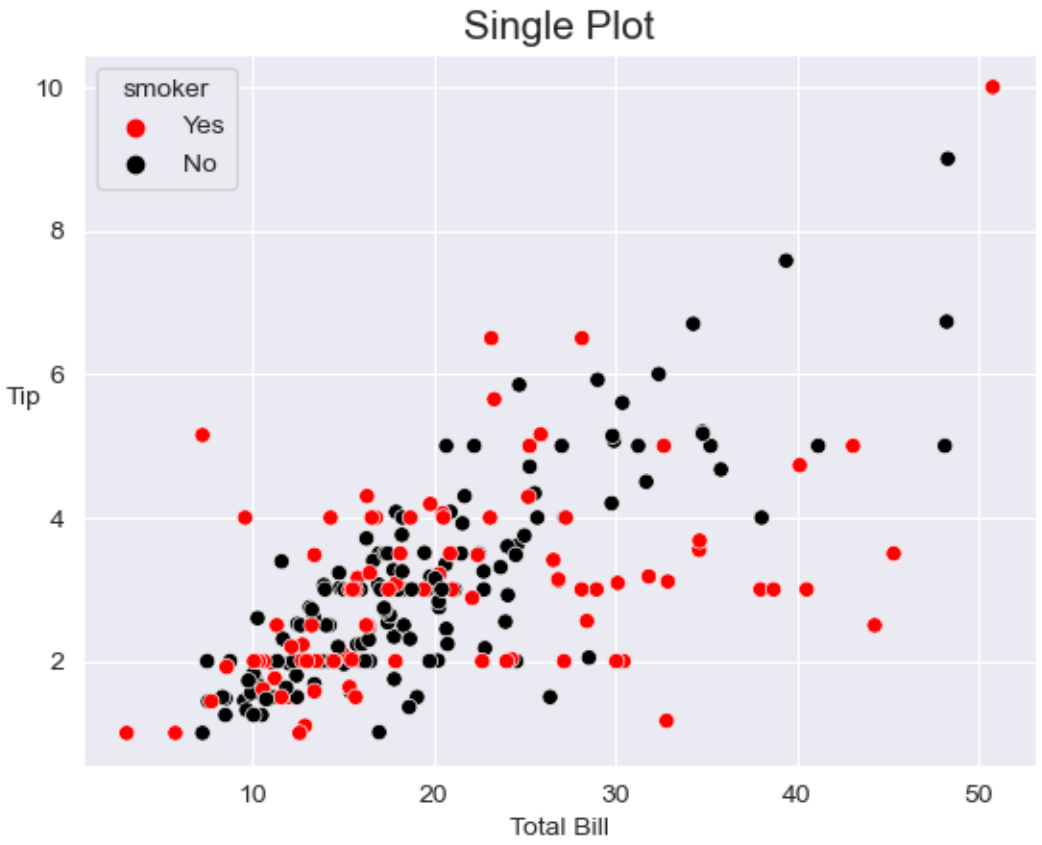
[72]:
```
#Multivariate Plots:
sns.set_style('darkgrid')
sns.set_palette('muted')
```

[73]:
```
#1. ScatterPlot:
x=np.random.randn(100)
y=np.random.randint(10,20,100)
sns.scatterplot(x=x,y=y)
plt.show()
```

[74]:
```
tips = sns.load_dataset("tips")
sns.scatterplot(x='total_bill',y='tip', data=tips)
plt.show() #Note that while applying sns on pd df we don't need to specify␣
 ↪xlabels, ylabels etc. sns automatically fetch it.
```

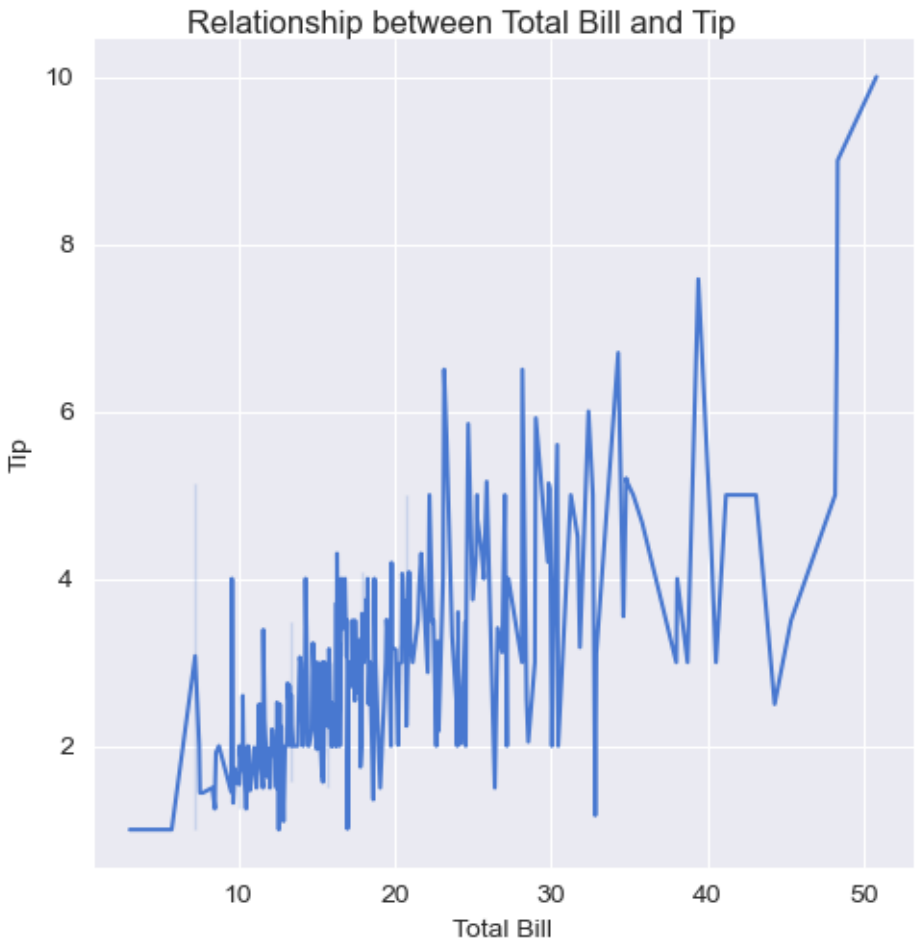


[108]:
```
g=sns.scatterplot(x='total_bill',y='tip',hue='smoker'␣
↪,hue_order=['Yes','No'],palette={'Yes':'red','No':'black'},data=tips) #Added␣
↪another variable and specified its (hue's) colors and legend order. The hue␣
↪parameter is used to add a third dimension to a plot, typically by␣
↪distinguishing data points with different colors based on the values of a␣
↪categorical variable.
g.set_title('Single Plot',fontsize=15) #we give the plot a variable name then␣
↪use syntax as in matplotlib. This is for single plot, for subplot type plots␣
↪(relplots and catplots) see next.
g.set_xlabel('Total Bill')
g.set_ylabel('Tip',rotation=0)
plt.show()
```



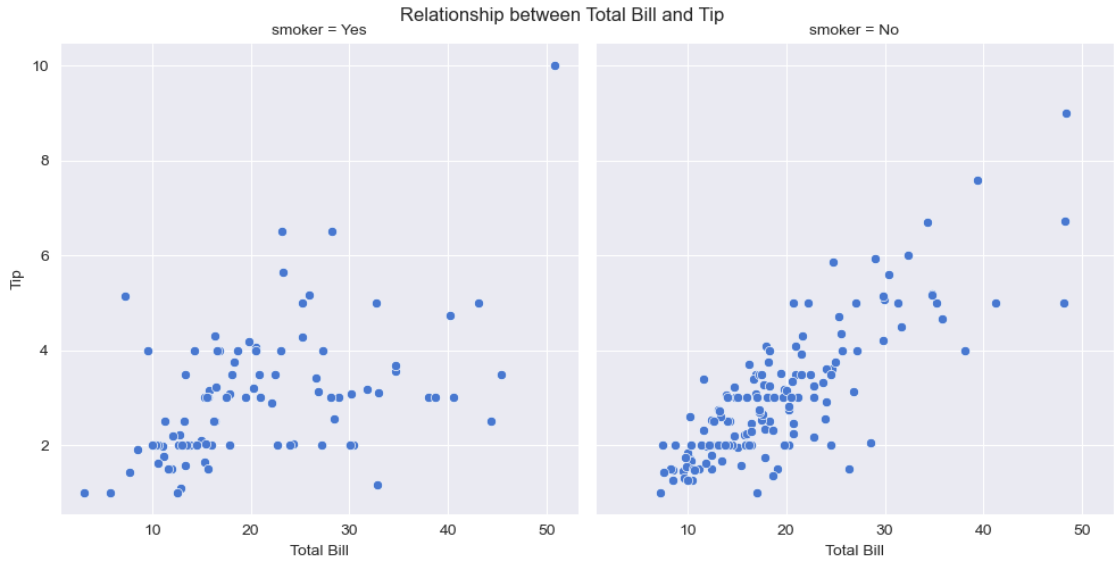
[112]:
```
#2. Relational Plot: The relplot function in Seaborn (which works well with␣
 ↪Pandas DataFrames) is a versatile function for creating relational plots␣
 ↪that visualize the relationship between multiple variables. It can generate␣
 ↪both scatter plots and line plots, depending on the kind of data you have.
g=sns.relplot(x='total_bill', y='tip', data=tips,kind='line') #Auto␣
 ↪kind='scatter'
plt.xlabel('Total Bill')
plt.ylabel('Tip')
plt.suptitle('Relationship between Total Bill and Tip',y=1) #This is for␣
 ↪subplotytpe plots
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



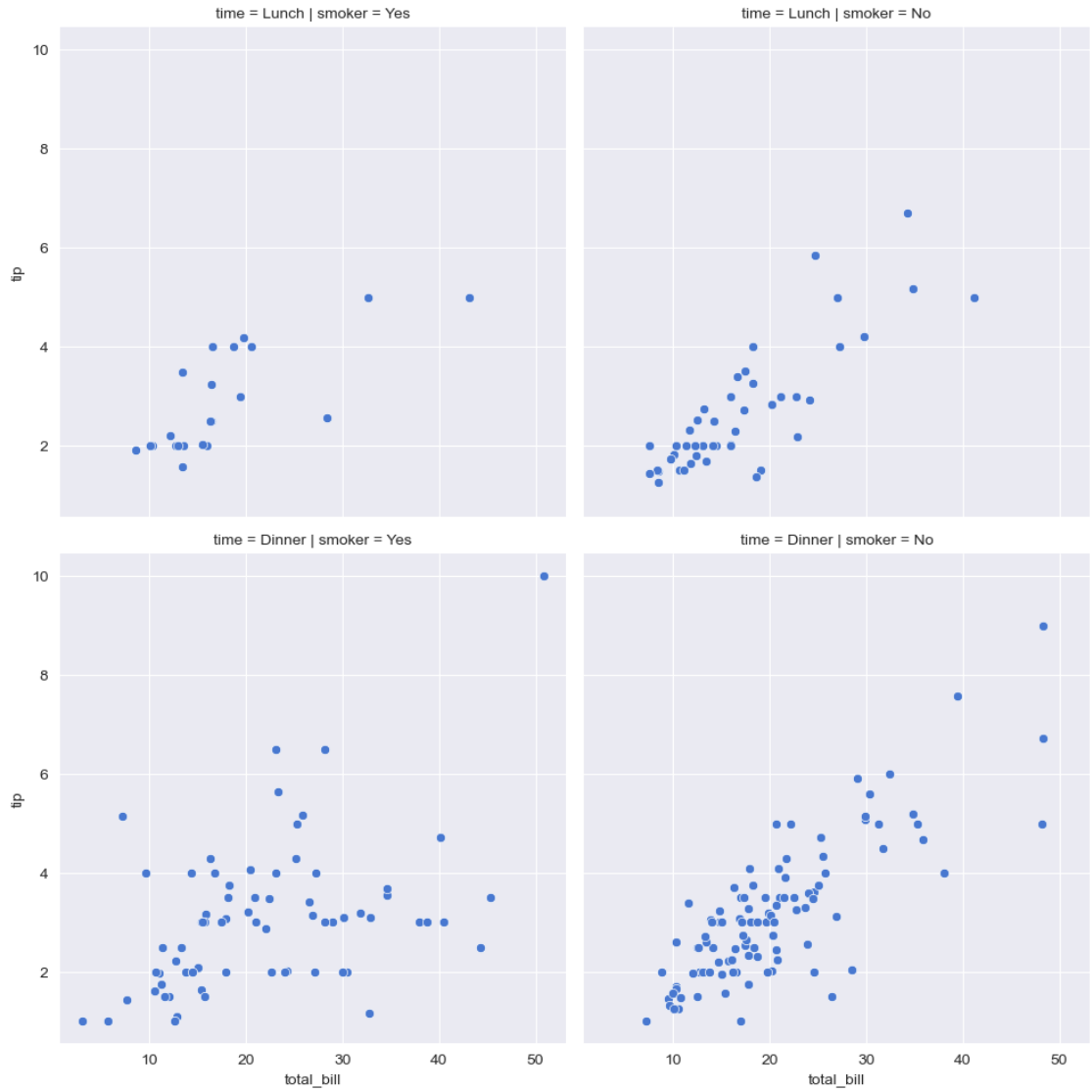
[114]:
```
#Customizing ScatterPlot:
g=sns.relplot(x='total_bill', y='tip',col='smoker', data=tips,kind='scatter')␣
  ↪#you can use row instead of col to get vertical plot
g.set(xlabel="Total Bill", ylabel="Tip")
g.fig.suptitle('Relationship between Total Bill and Tip',y=1) #For subplotype␣
  ↪plots (catplot, relplot)
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



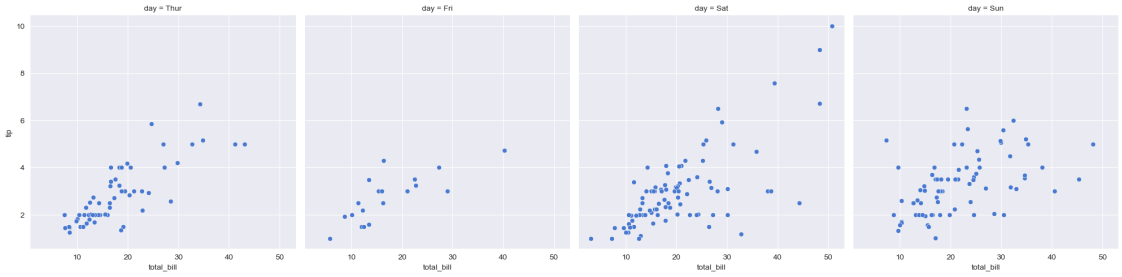
[78]:
```
sns.relplot(x='total_bill', y='tip',col='smoker',row='time',␣
 ↪data=tips,kind='scatter')
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



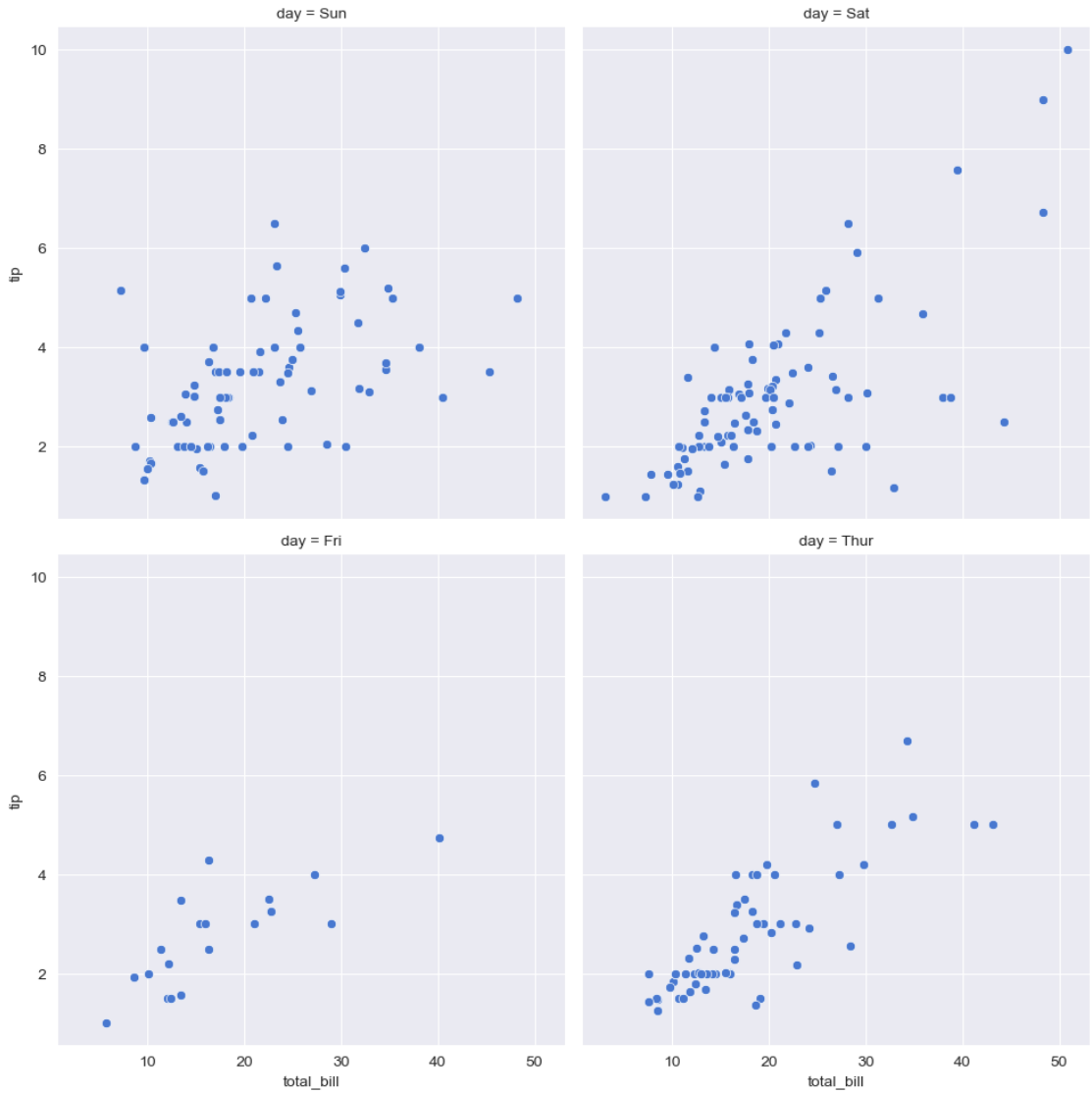
[79]:
```
sns.relplot(x='total_bill', y='tip',col='day', data=tips,kind='scatter')
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



[80]:
```
sns.relplot(x='total_bill', y='tip',col='day', col_wrap=2,␣
  ↪col_order=['Sun','Sat','Fri','Thur'],data=tips,kind='scatter')
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



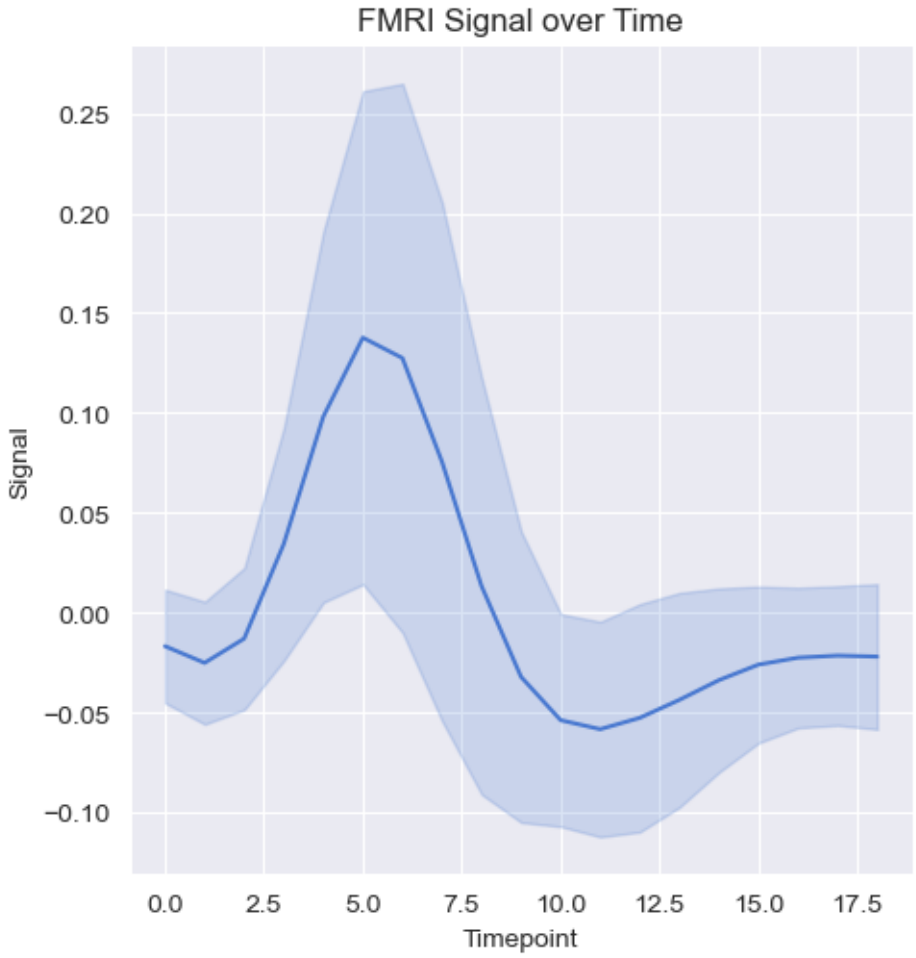
[81]:
```
sns.
 ↪relplot(x='total_bill',y='tip',hue='smoker',size='size',style='smoker',alpha=0.
 ↪8,hue_order=['Yes','No'],palette={'Yes':'red','No':'black'},data=tips)
plt.show()
#Like hue changes color of scatters based on selected column, size changes␣
 ↪size, style changes style while alpha is for transperancy.
#We can apply all these customizations on line kind plot also
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[82]:
```
#Customizing LinePlot:
#i. Basic Plot:
fmri = sns.load_dataset("fmri")
sns.relplot(x='timepoint', y='signal', kind='line', data=fmri,ci='sd')␣
 ↪#ci='mean' by default
plt.xlabel('Timepoint')
plt.ylabel('Signal')
plt.title('FMRI Signal over Time')
plt.show()
#The shaded region shows ci of std
```

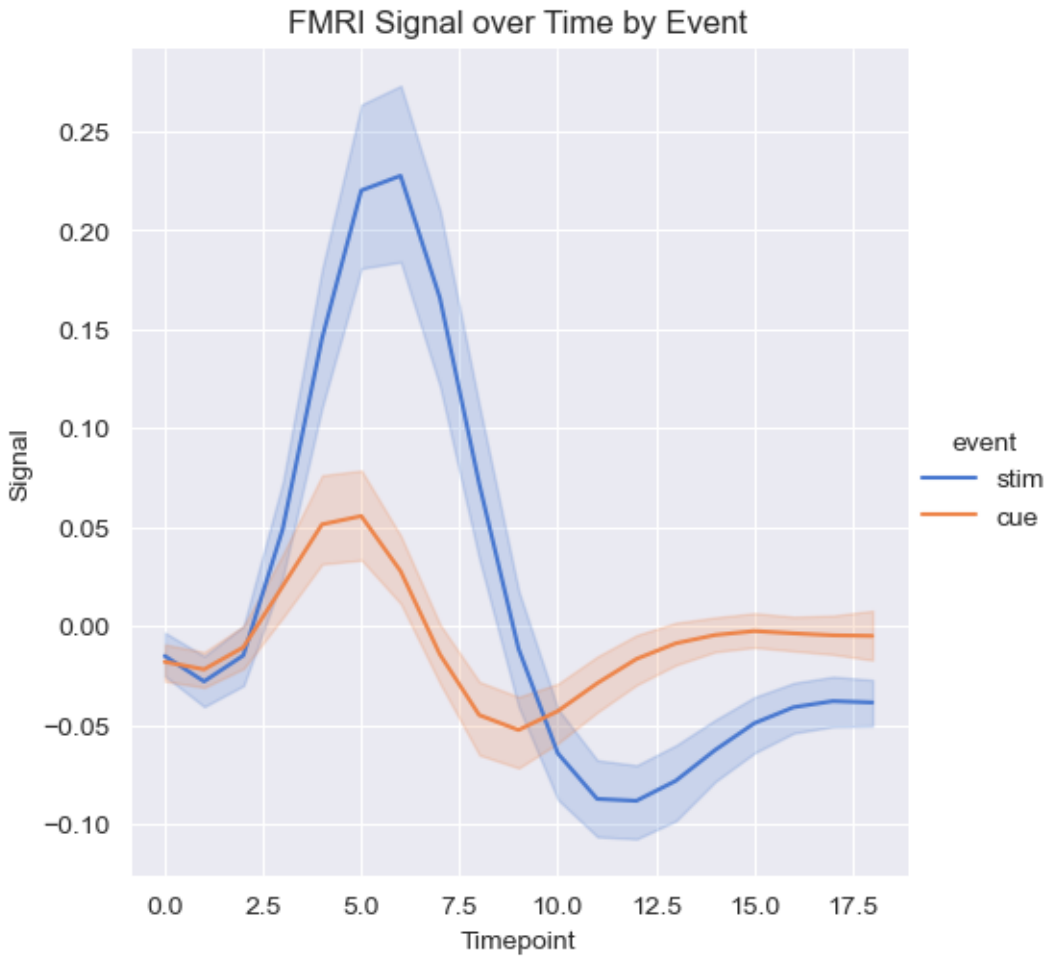
```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:848:
FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  func(*plot_args, **plot_kwargs)
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
```

```
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



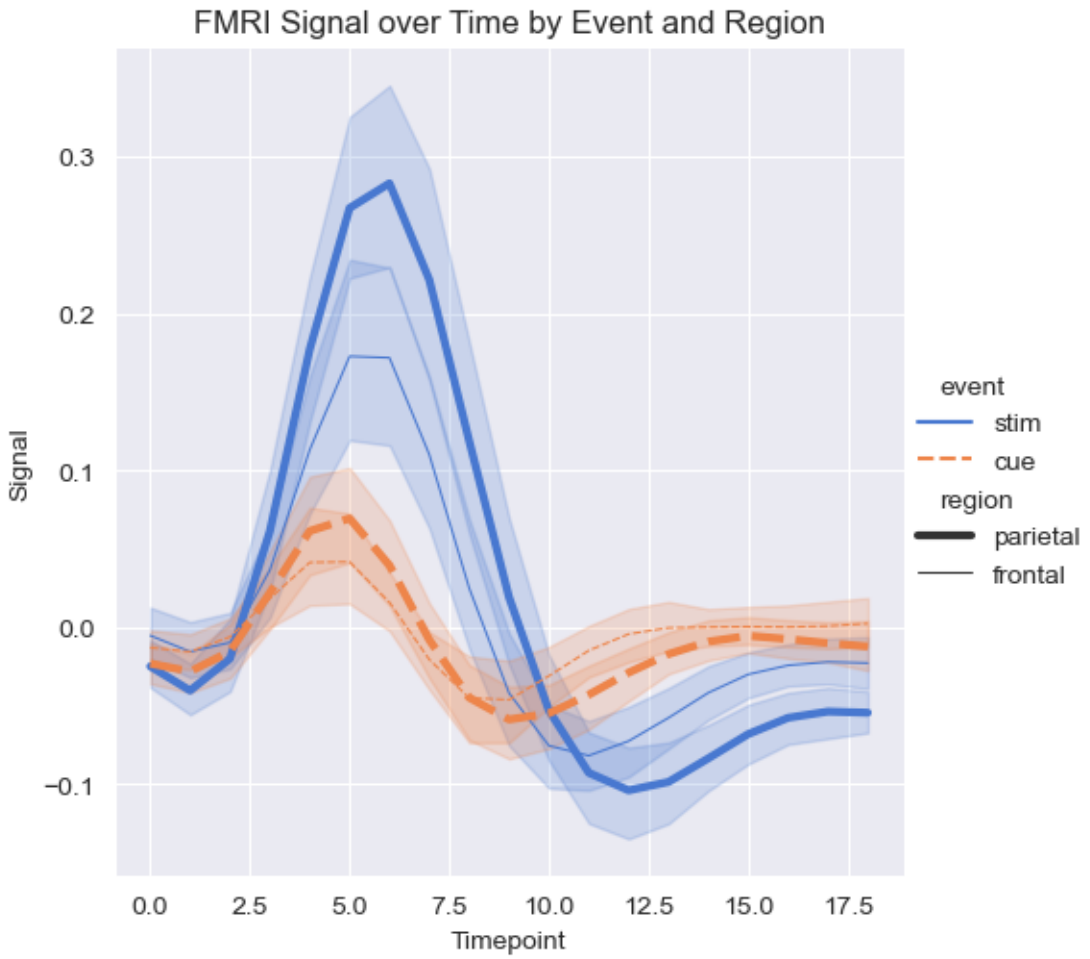
[83]:
```
#You can use the hue parameter to add color grouping to the line plot:
sns.relplot(x='timepoint', y='signal', hue='event', kind='line', data=fmri)
plt.xlabel('Timepoint')
plt.ylabel('Signal')
plt.title('FMRI Signal over Time by Event')
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



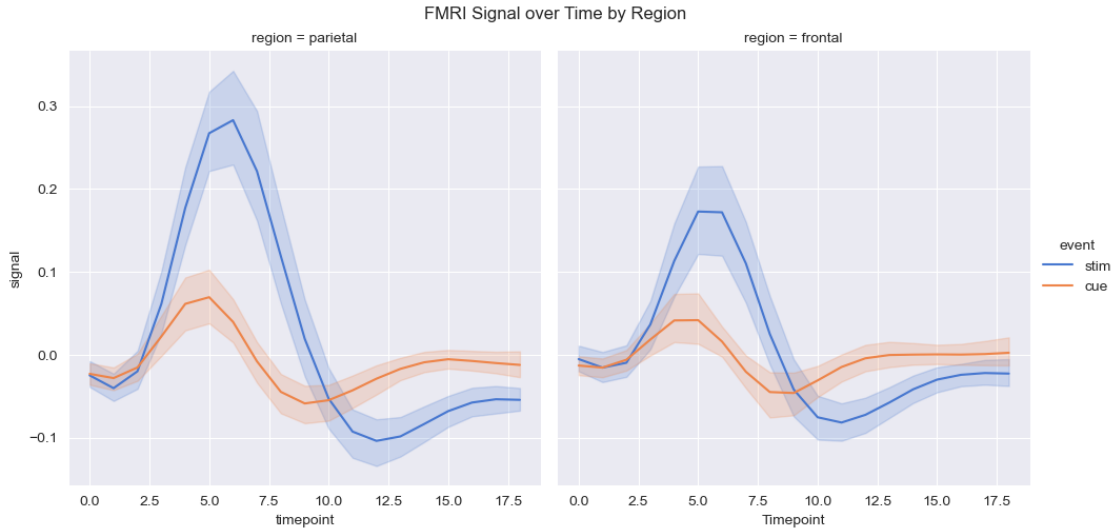
[84]:
```
#The style and size parameters can further distinguish lines based on different␣
 ↪categorical variables:
sns.relplot(x='timepoint', y='signal', hue='event', style='event',␣
 ↪size='region', kind='line', data=fmri)
plt.xlabel('Timepoint')
plt.ylabel('Signal')
plt.title('FMRI Signal over Time by Event and Region')
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



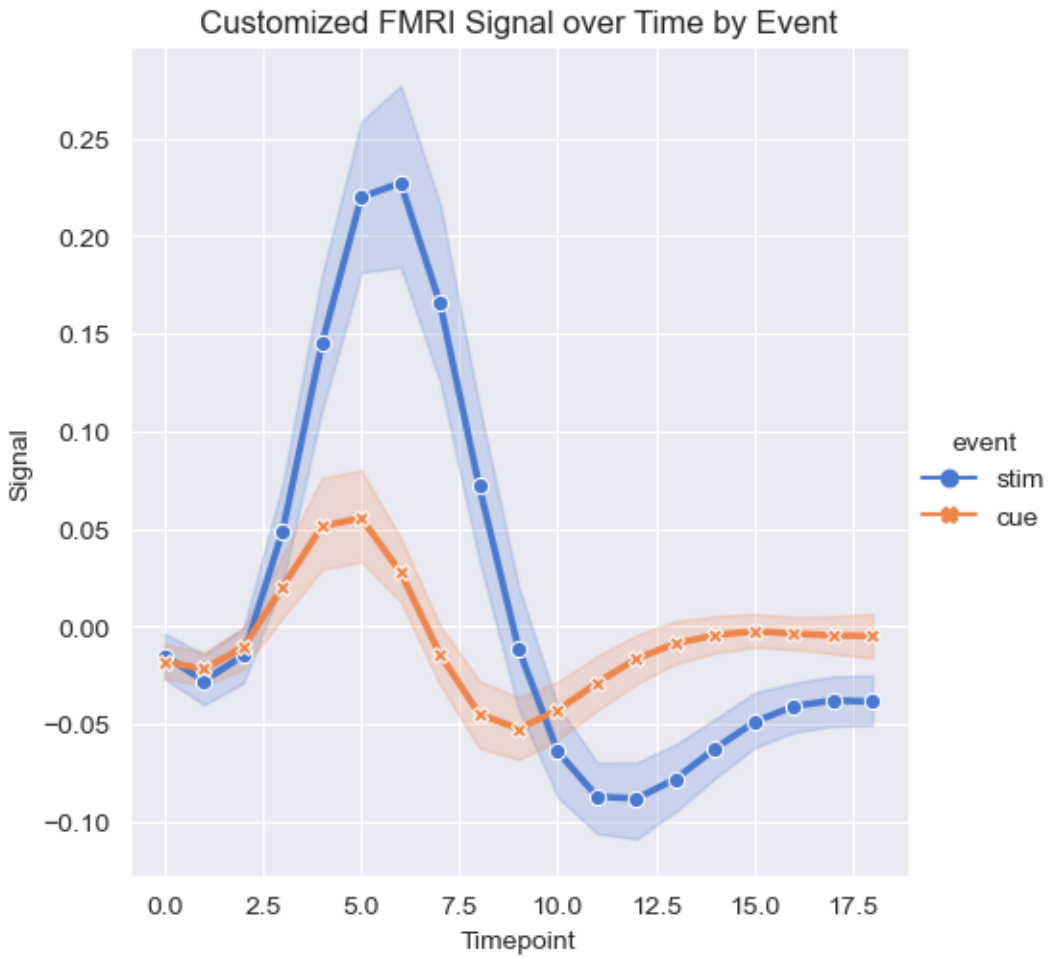
[85]:
```
#You can create multiple subplots based on another variable using col and row
sns.relplot(x='timepoint', y='signal', hue='event', kind='line', col='region',␣
 ↪data=fmri)
plt.xlabel('Timepoint')
plt.ylabel('Signal')
plt.suptitle('FMRI Signal over Time by Region', y=1.02)
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



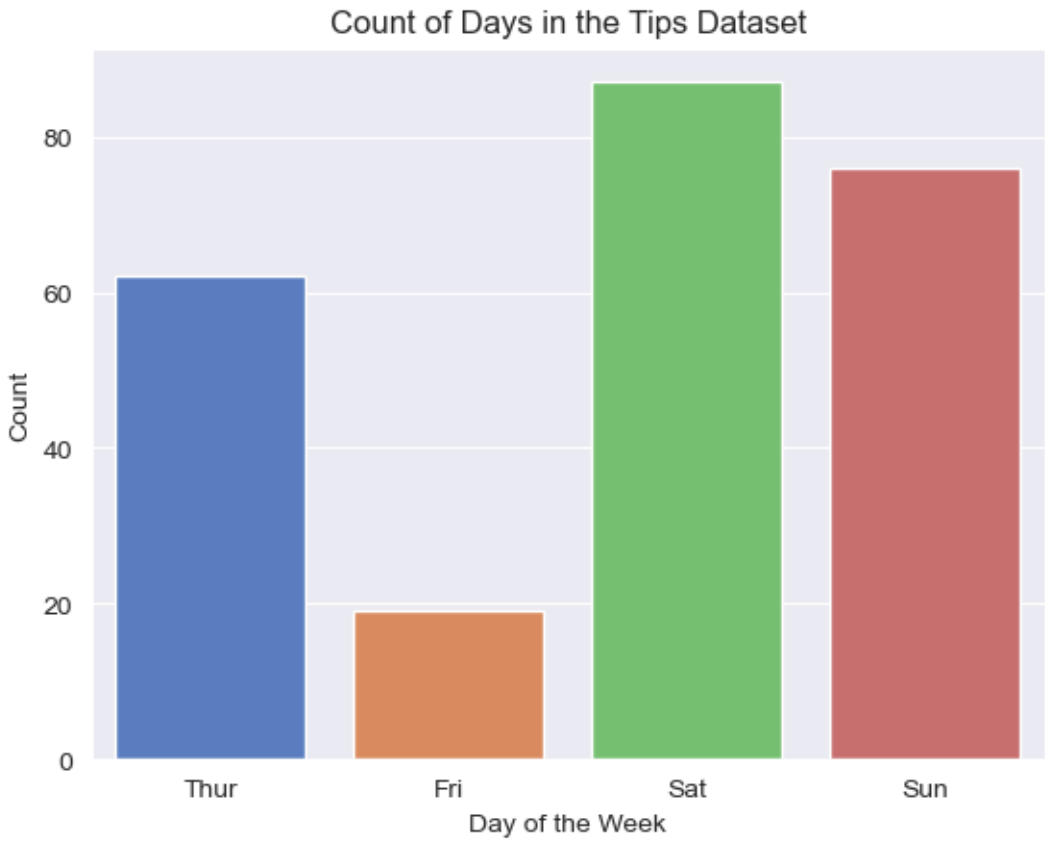
[86]:
```
#You can customize the appearance of the lines with parameters like linewidth,␣
 ↪linestyle, and markers:
sns.relplot(x='timepoint', y='signal', hue='event', kind='line',␣
 ↪data=fmri,linewidth=2.5, style='event', markers=True, dashes=False)
plt.xlabel('Timepoint')
plt.ylabel('Signal')
plt.title('Customized FMRI Signal over Time by Event')
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

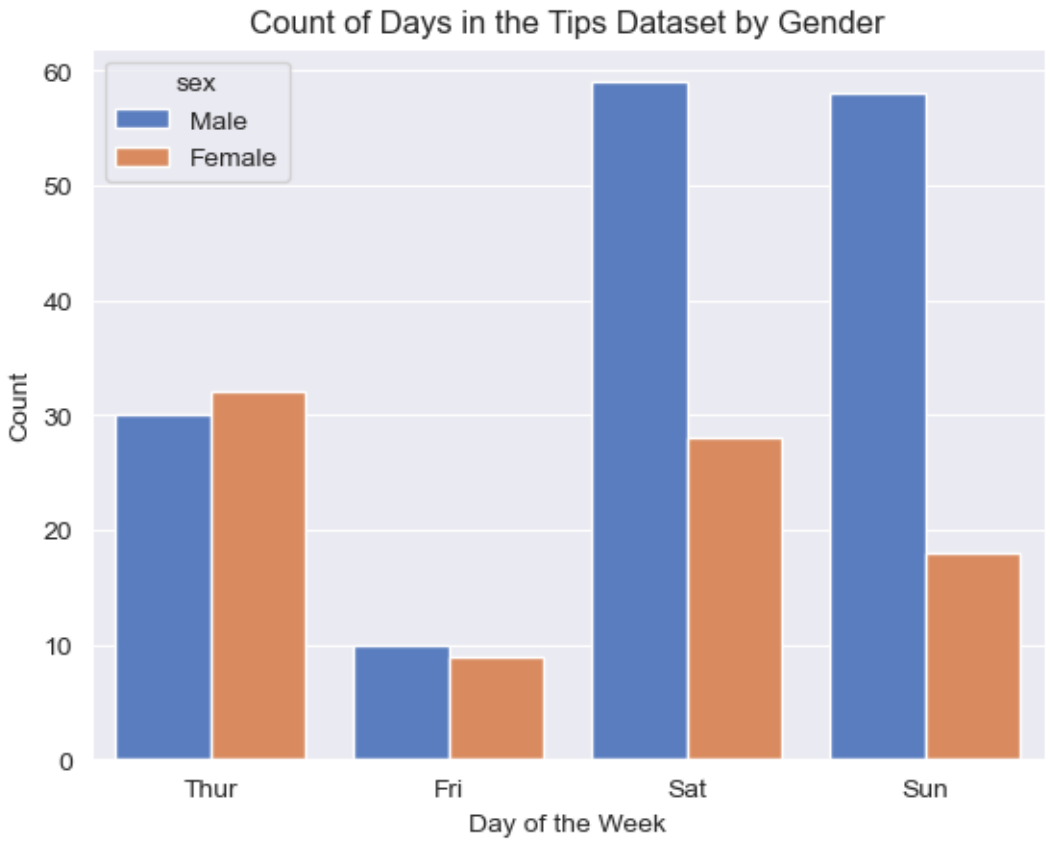


[87]:
```
#Categorical Plots:
```

[88]:
```
#1. Count plot in Seaborn is a bar plot that displays the counts of␣
 ↪observations in each categorical bin using bars
tips = sns.load_dataset("tips") #'tips' dataset (builtin)
sns.countplot(x='day', data=tips) #'x' column of 'tips' df
plt.xlabel('Day of the Week')
plt.ylabel('Count')
plt.title('Count of Days in the Tips Dataset')
plt.show()
```

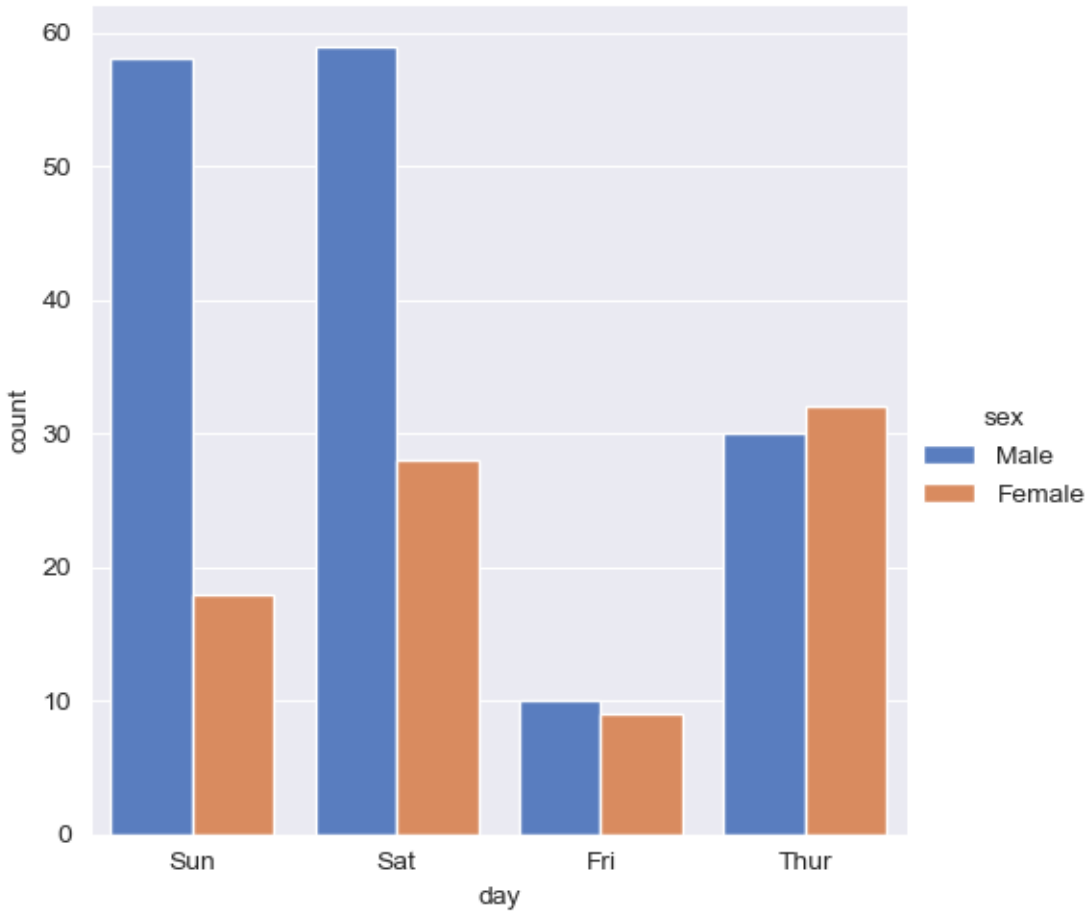


[89]:
```
sns.countplot(x='day', hue='sex', data=tips) #The hue parameter adds another␣
↪categorical variable, splitting the bars into segments colored by the hue␣
↪variable.
plt.xlabel('Day of the Week')
plt.ylabel('Count')
plt.title('Count of Days in the Tips Dataset by Gender')
plt.show()
```



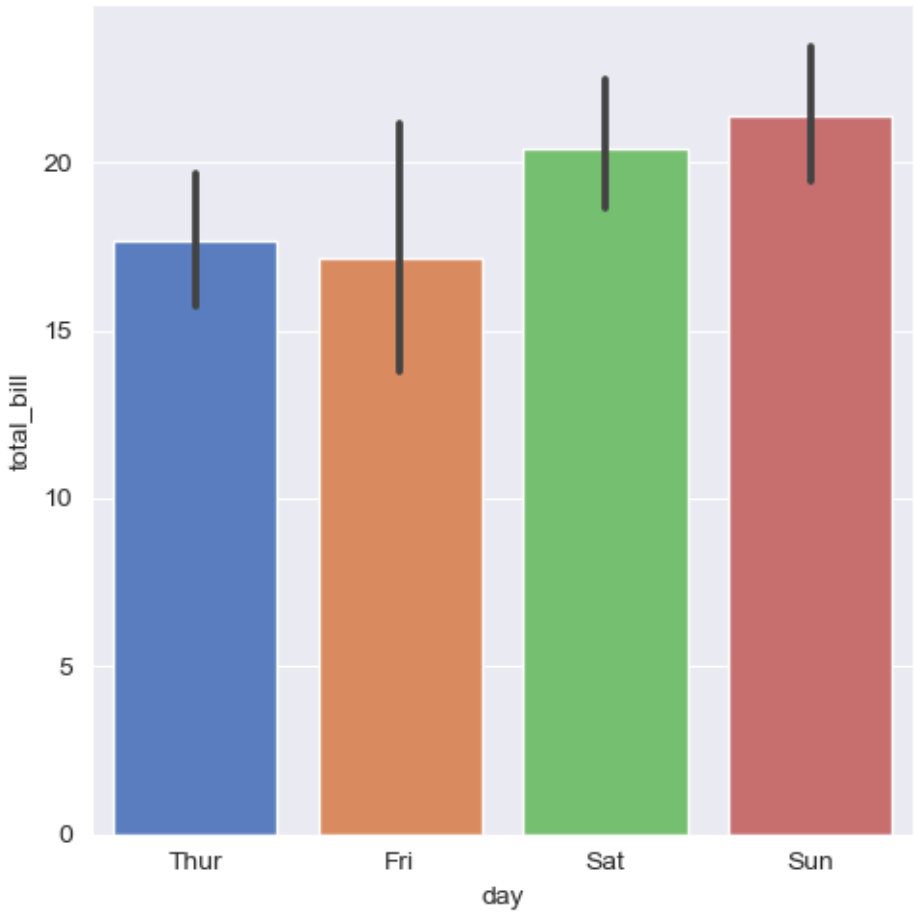
[90]:
```
#2. Cat Plot: Like relplot, A categorical plot (catplot) in Seaborn is a␣
 ↪high-level interface for drawing categorical plots. It can generate␣
 ↪different types of plots such as bar plots, box plots, and violin plots,␣
 ↪among others.
sns.catplot(x='day', hue='sex',␣
 ↪data=tips,order=['Sun','Sat','Fri','Thur'],kind='count') #Bydefault␣
 ↪kind=scatter
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



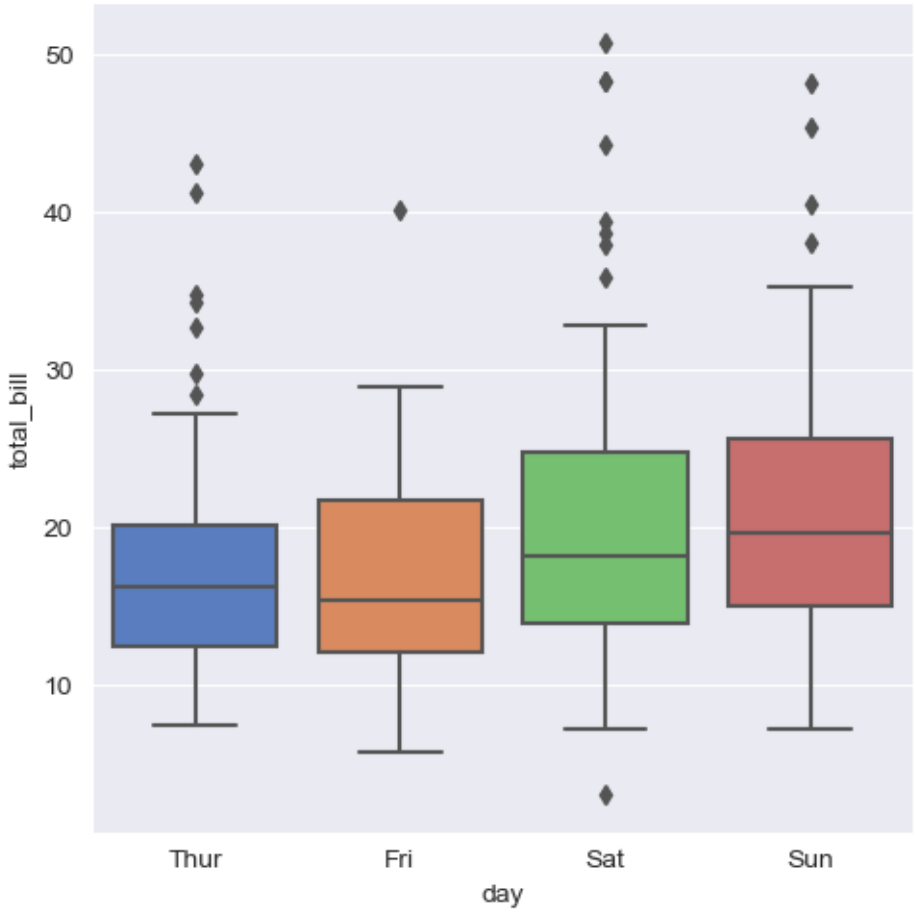
[91]:
```
sns.catplot(x='day', y='total_bill', data=tips,kind='bar')
plt.show()
#We can use row, col parameters as in relplot
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



[92]:
```
sns.catplot(x='day', y='total_bill', data=tips,kind='box',whis=1.0) #By default␣
↪whisks extend(low and up end) at 1.5*IQR (IQR-[25,75]), we changed to 1.0
plt.show()
```

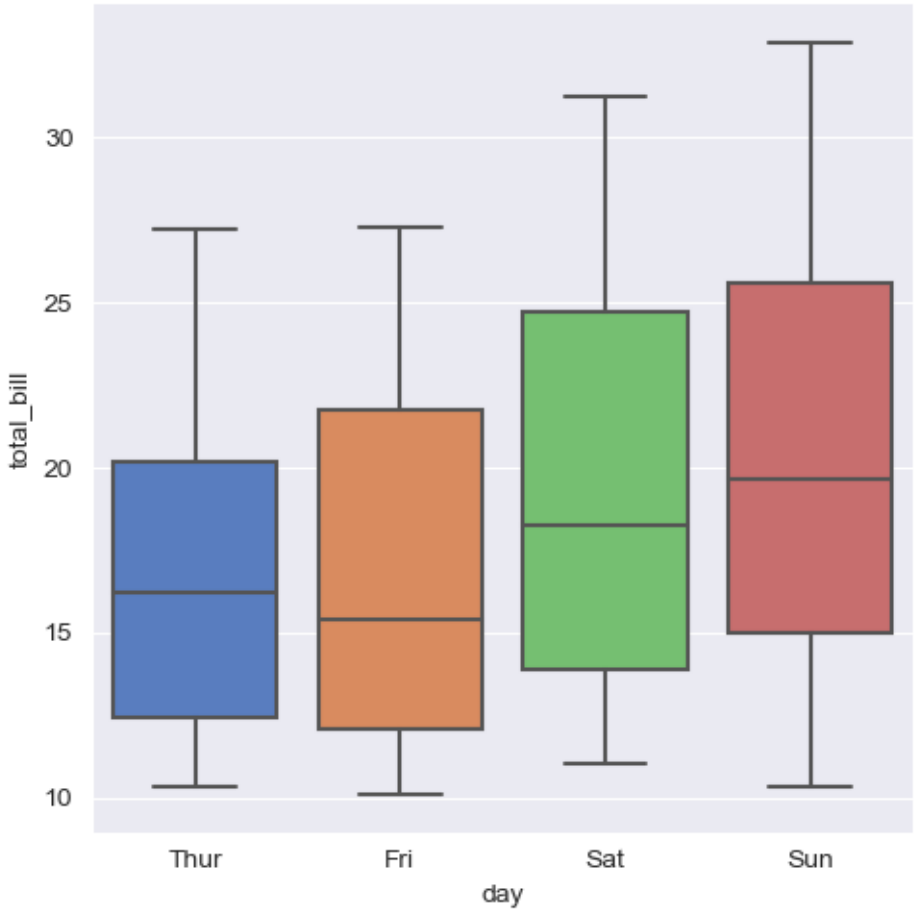
```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



[93]:
```
#In above box plots by default whiskers tell IQR, scatters represent outliers,␣
↪centre line=median,closing of box are 25th and 75th percentile
```

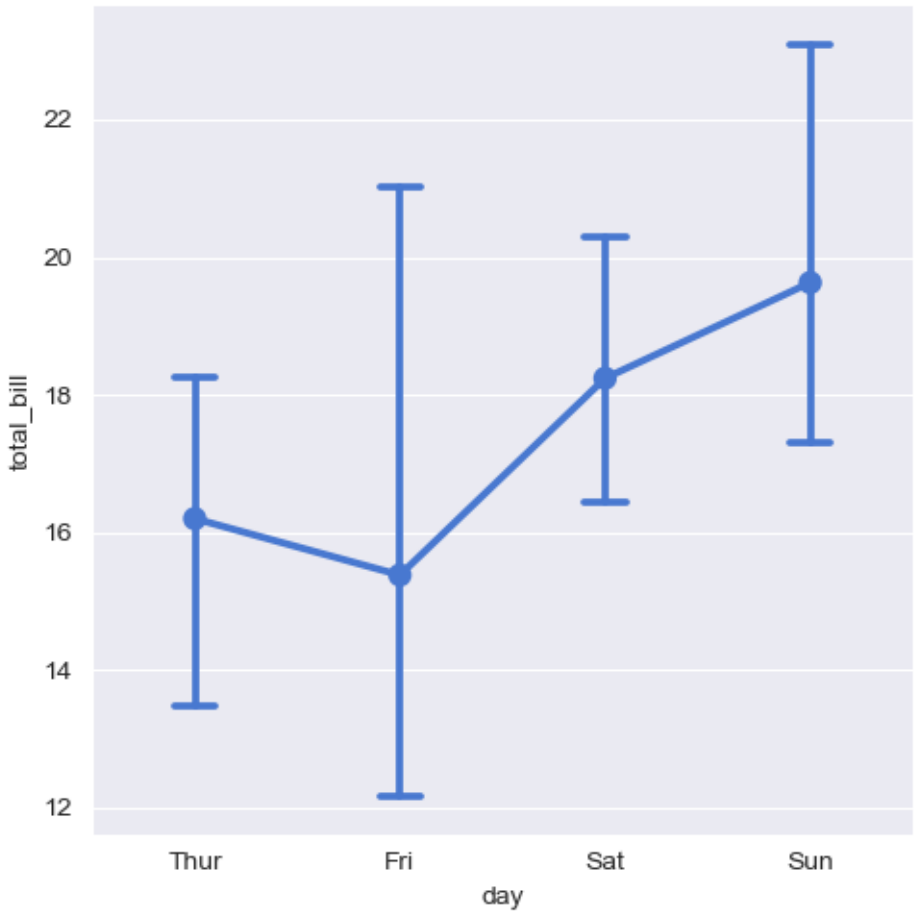
[94]:
```
sns.catplot(x='day', y='total_bill', data=tips,kind='box',sym='',whis=[10,90])␣
↪#sym='' omits outliers, whis=[10,90] changed percentiles (closings of box)
plt.show()
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



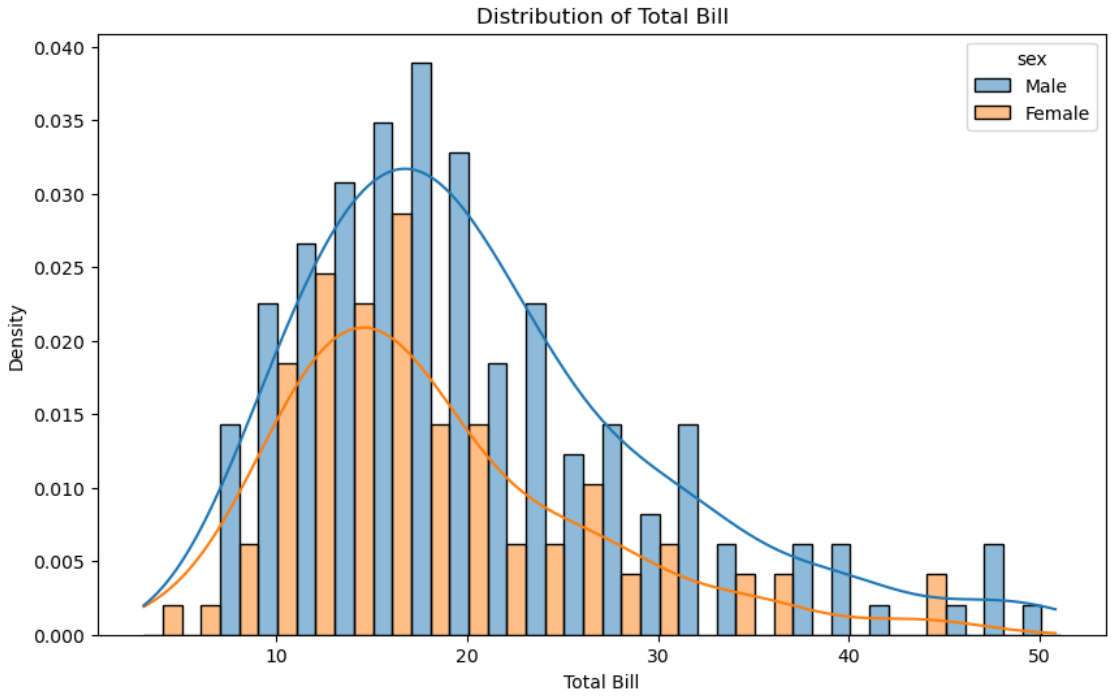
[95]:
```
#This plot is used to compare statistical metrics (mean,median, std etc.)
sns.catplot(x='day', y='total_bill',␣
 ↪data=tips,kind='point',estimator='median',capsize=0.2) #estimator='mean' by␣
 ↪default, use join=False to get rid of joining lines
plt.show() #dodge=True in pointplot makes sure that lines joining the two␣
 ↪points don't overlap
#In this plot points represent statistical metrics, vertical lines represent ci␣
 ↪or confidence intervals and horizontal lins (cap) are end points of ci
```

```
C:\Users\scimo\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



[7]:
```
#3. Histplot:
import seaborn as sns
import matplotlib.pyplot as plt
tips = sns.load_dataset("tips")
plt.figure(figsize=(10, 6))
sns.histplot(data=tips, x='total_bill', bins=20,binwidth=2, kde=True,␣
 ↪hue='sex', multiple='dodge', element='bars', stat='density')
plt.title('Distribution of Total Bill')
plt.xlabel('Total Bill')
plt.ylabel('Density')
plt.show()
'''bins: Specify the number of bins (bars) in the histogram.
kde: Add a Kernel Density Estimate (KDE) curve to the histogram.  Adds a KDE␣
 ↪line to the histogram, providing a smoothed representation of the data␣
 ↪distribution.
hue: Add another dimension by coloring the bars based on a categorical variable.
multiple: Specify how to handle multiple variables (stack, dodge, fill, layer)␣
 ↪to visualize multiple distributions within the same plot.
```

```
element: Specify the type of plot element (bars, step, poly).
stat: Specify the statistic to compute (count, frequency, density, probability).
common_norm: Normalize across the histogram bars for all levels of the hue␣
 ↪variable.'''
```



[7]: 'bins: Specify the number of bins (bars) in the histogram.\nkde: Add a Kernel Density Estimate (KDE) curve to the histogram. Adds a KDE line to the histogram, providing a smoothed representation of the data distribution.\nhue: Add another dimension by coloring the bars based on a categorical variable.\nmultiple: Specify how to handle multiple variables (stack, dodge, fill, layer) to visualize multiple distributions within the same plot.\nelement: Specify the type of plot element (bars, step, poly).\nstat: Specify the statistic to compute (count, frequency, density, probability).\ncommon_norm: Normalize across the histogram bars for all levels of the hue variable.'