

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
tp = pd.read_csv("tips.csv")
tp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   Unnamed: 0    244 non-null   int64   
 1   total_bill    244 non-null   float64  
 2   tip           244 non-null   float64  
 3   sex           244 non-null   object  
 4   smoker        244 non-null   object  
 5   day           244 non-null   object  
 6   time          244 non-null   object  
 7   size          244 non-null   int64   
dtypes: float64(2), int64(2), object(4)
memory usage: 15.4+ KB
```

Question 1:

Delete the Unnamed 0 column

In [3]:

```
tp = tp.drop(columns="Unnamed: 0", axis= 1)
tp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   total_bill    244 non-null   float64  
 1   tip           244 non-null   float64  
 2   sex           244 non-null   object  
 3   smoker        244 non-null   object  
 4   day           244 non-null   object  
 5   time          244 non-null   object  
 6   size          244 non-null   int64   
dtypes: float64(2), int64(1), object(4)
memory usage: 13.5+ KB
```

Question 2:

Plot the total_bill column histogram

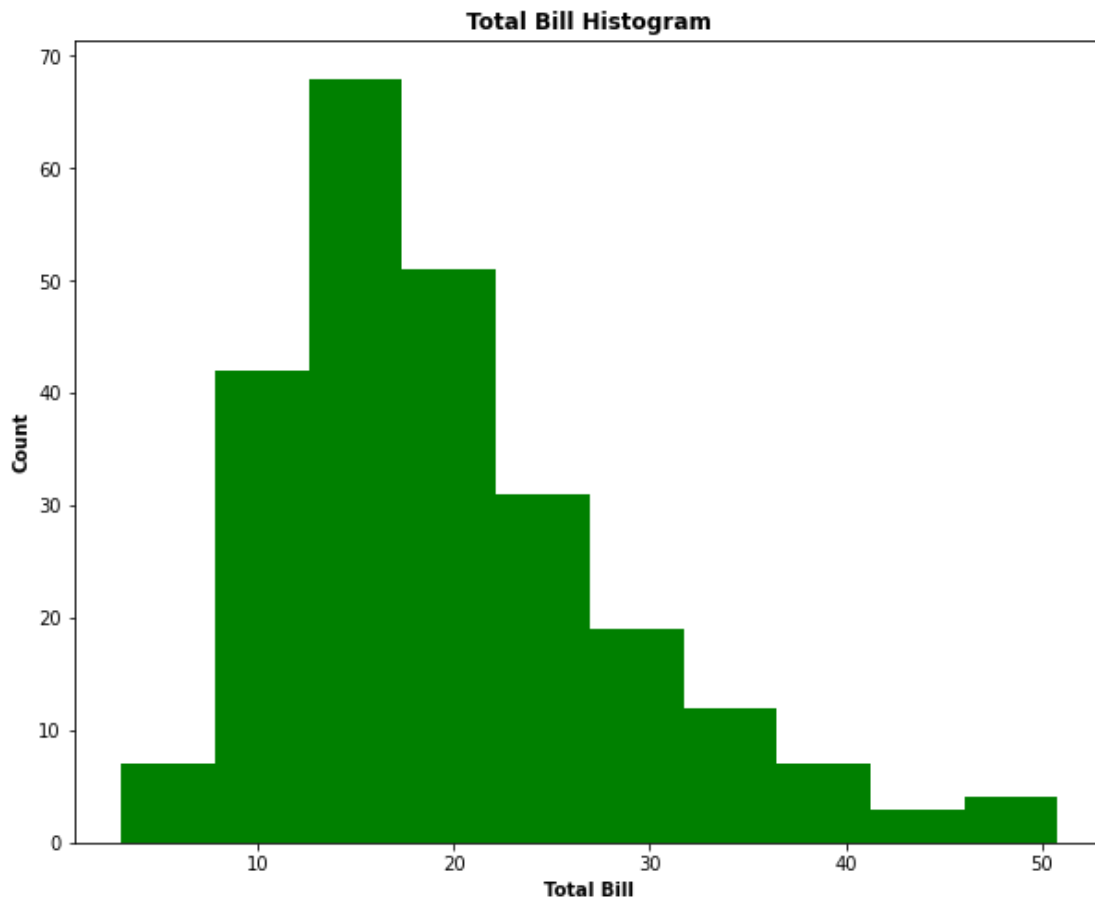
In [4]:

```
from matplotlib import pyplot as plt
```

In [5]:

```
fig = plt.figure(figsize= (10, 8))
plt.hist(data=tp, x="total_bill", color="green")
plt.xlabel("Total Bill", weight="bold")
plt.ylabel("Count", weight="bold")
plt.title("Total Bill Histogram", weight="bold")
```

```
plt.show()
```



Question 3:

Create a scatter plot presenting the relationship between `total_bill` and `tip`.

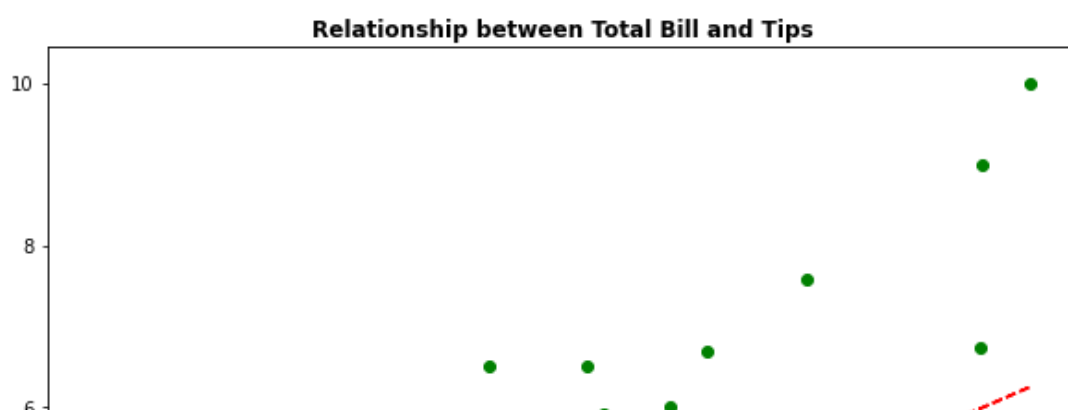
In [6]:

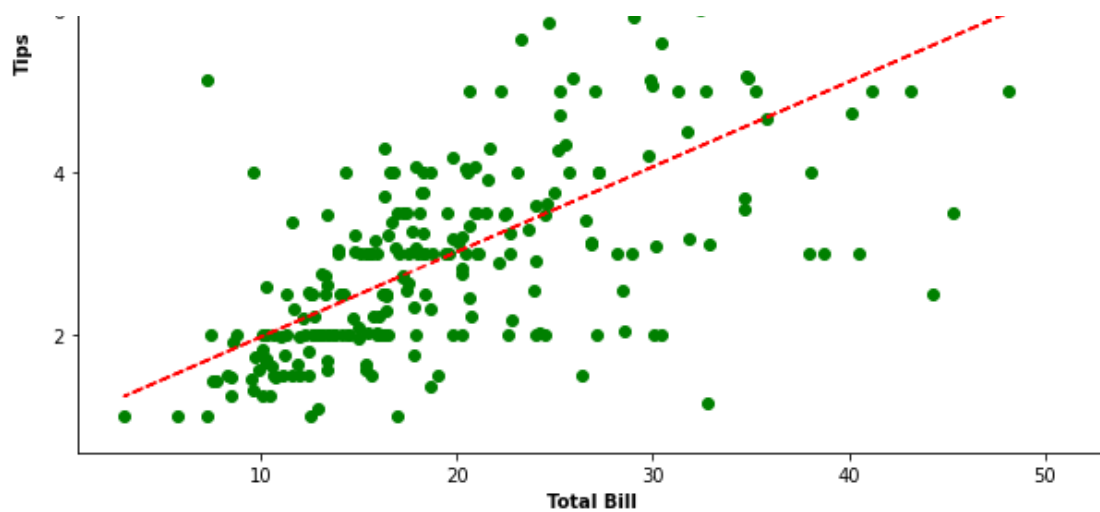
```
x = tp["total_bill"]
y = tp["tip"]

fig = plt.figure(figsize=(10, 8))
plt.scatter(x, y, color="green")
z = np.polyfit(x, y, 1)
p = np.poly1d(z)
plt.plot(x, p(x), 'r--')
corr_coef = np.corrcoef(x, y)[0, 1]

plt.xlabel('Total Bill', weight='bold')
plt.ylabel('Tips', weight='bold')
plt.title('Relationship between Total Bill and Tips', weight='bold')
plt.text(0.5, 1.1, f"Correlation coefficient = {corr_coef:.2f}", ha='center', va='center',
         weight='bold', transform=plt.gca().transAxes)
plt.show()
```

Correlation coefficient = 0.68





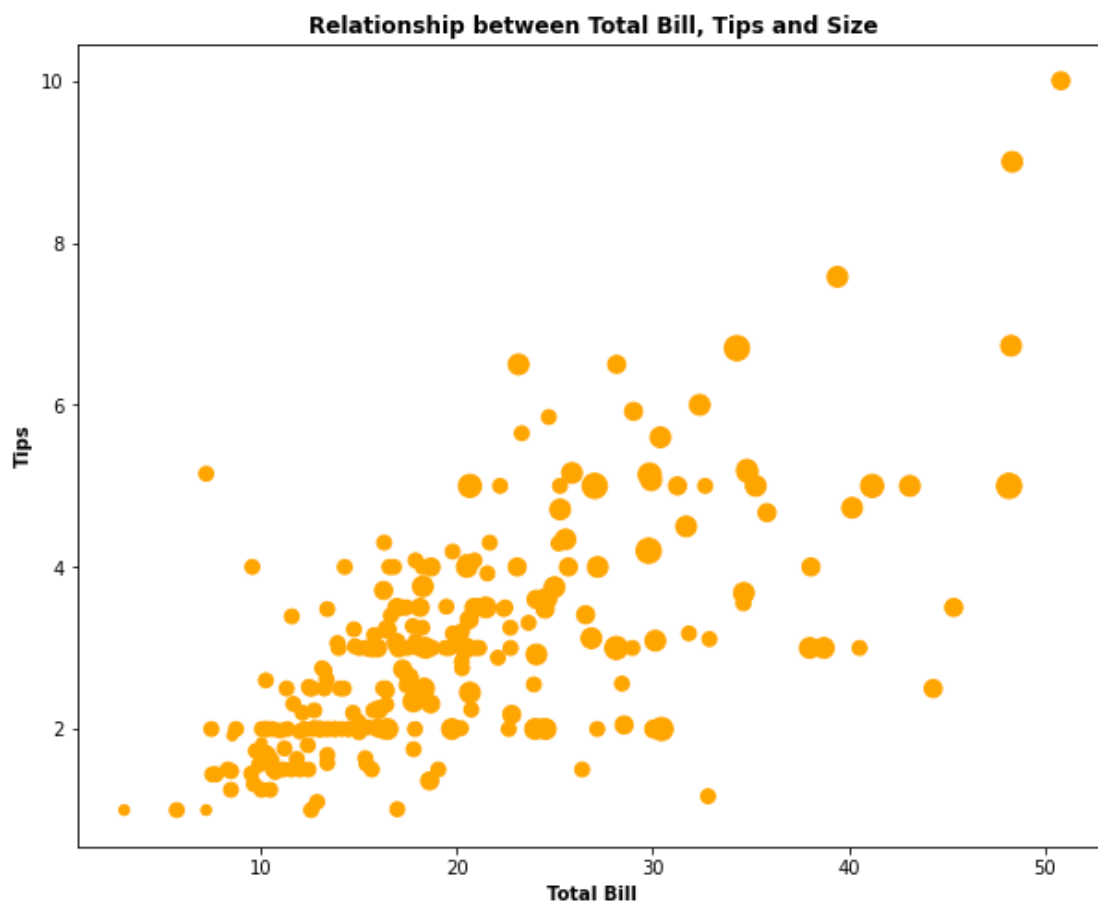
Question 4:

Create one image with the relationship of total_bill, tip and size.

In [7]:

```
x = tp["total_bill"]
y = tp["tip"]
z = tp["size"]

fig = plt.figure(figsize=(10, 8))
plt.scatter(x, y, z*30, color="orange")
plt.xlabel('Total Bill', weight='bold')
plt.ylabel('Tips', weight='bold')
plt.title('Relationship between Total Bill, Tips and Size', weight='bold')
plt.show()
```



Question 5:

Present the relationship between days and total_bill value

In [8]:

```
import seaborn as sns
```

In [9]:

```
temp1 = pd.DataFrame({
    'days': tp['day'],
    'total_bill': tp['total_bill']
})

day_order = ['Thur', 'Fri', 'Sat', 'Sun']
temp1['days'] = pd.Categorical(temp1['days'], categories=day_order, ordered=True)
temp1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   days        244 non-null   category
 1   total_bill  244 non-null   float64
dtypes: category(1), float64(1)
memory usage: 2.5 KB
```

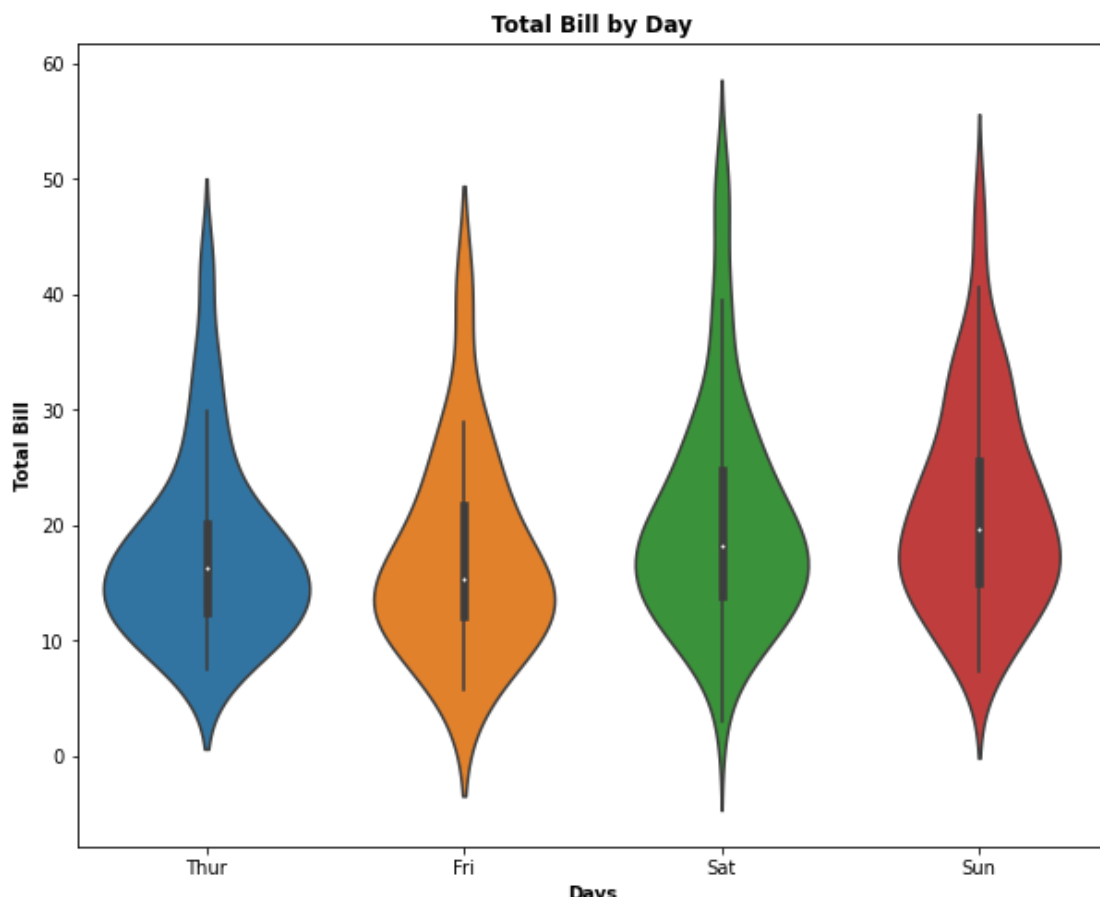
In [10]:

```
x = temp1["days"]
y = temp1["total_bill"]

fig = plt.figure(figsize=(10, 8))
sns.violinplot(x, y)
plt.title("Total Bill by Day", weight="bold")
plt.xlabel("Days", weight="bold")
plt.ylabel("Total Bill", weight="bold")
plt.show()
```

c:\Python\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```



Question 6:

Create a scatter plot with the day as the y-axis and tip as the x-axis, differ the dots by sex

In [11]:

```
temp2 = pd.DataFrame({
    'days': tp['day'],
    'tips': tp['tip'],
    'sex': tp['sex']
})

day_order = ['Sun', 'Sat', 'Fri', 'Thur']
temp2['days'] = pd.Categorical(temp2['days'], categories=day_order, ordered=True)
temp2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  -
 0   days    244 non-null      category
 1   tips    244 non-null      float64
 2   sex     244 non-null      object  
dtypes: category(1), float64(1), object(1)
memory usage: 4.4+ KB
```

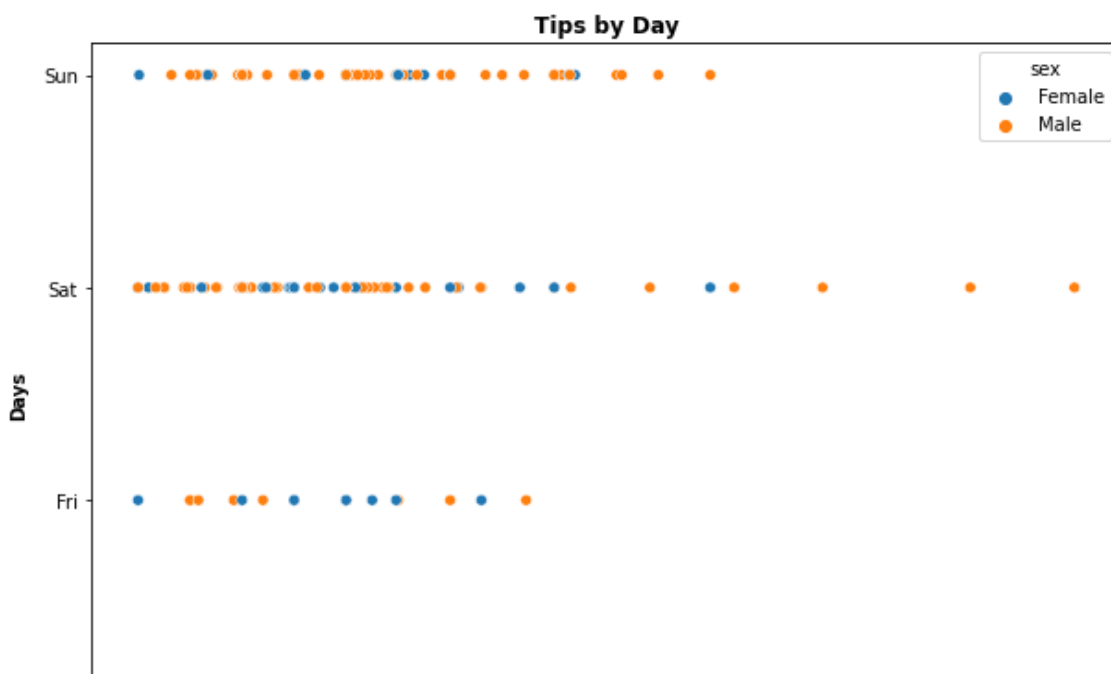
In [12]:

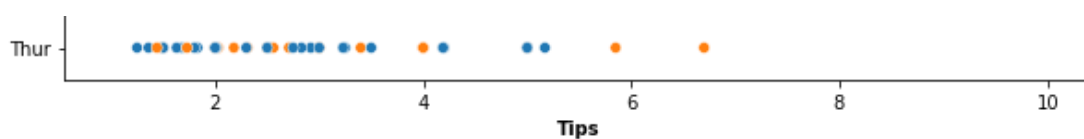
```
x = temp2['tips']
y = temp2['days']
z = temp2['sex']

fig = plt.figure(figsize=(10, 7))
sns.scatterplot(x, y, hue=z)
plt.title("Tips by Day", weight='bold')
plt.xlabel("Tips", weight='bold')
plt.ylabel("Days", weight='bold')
plt.show()
```

c:\Python\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```





Question 7:

Create a box plot presenting the total_bill per day differentiation the time (Dinner or Lunch)

In [13]:

```
temp3 = pd.DataFrame({
    'days': tp['day'],
    'total_bill': tp['total_bill'],
    'time': tp['time']
})

day_order = ['Thur', 'Fri', 'Sat', 'Sun']
temp3['days'] = pd.Categorical(temp3['days'], categories=day_order, ordered=True)
temp3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   days        244 non-null    category
1   total_bill  244 non-null    float64
2   time        244 non-null    object
dtypes: category(1), float64(1), object(1)
memory usage: 4.4+ KB
```

In [14]:

```
x = temp3['days']
y = temp3['total_bill']
z = temp3['time']

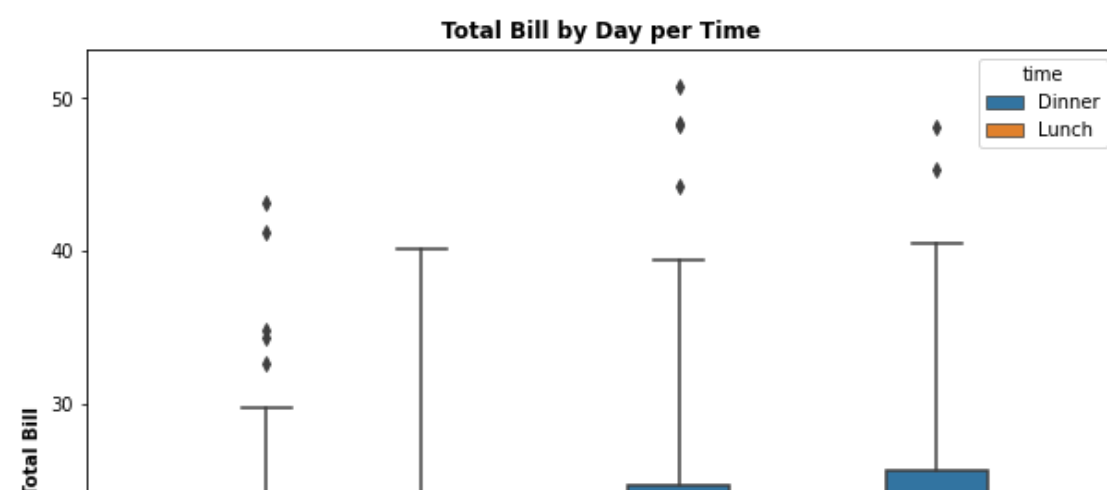
fig = plt.figure(figsize=(10,8))
sns.boxplot(x, y, hue=z)
plt.title('Total Bill by Day per Time', weight='bold')
plt.xlabel('Days', weight='bold')
plt.ylabel('Total Bill', weight='bold')
```

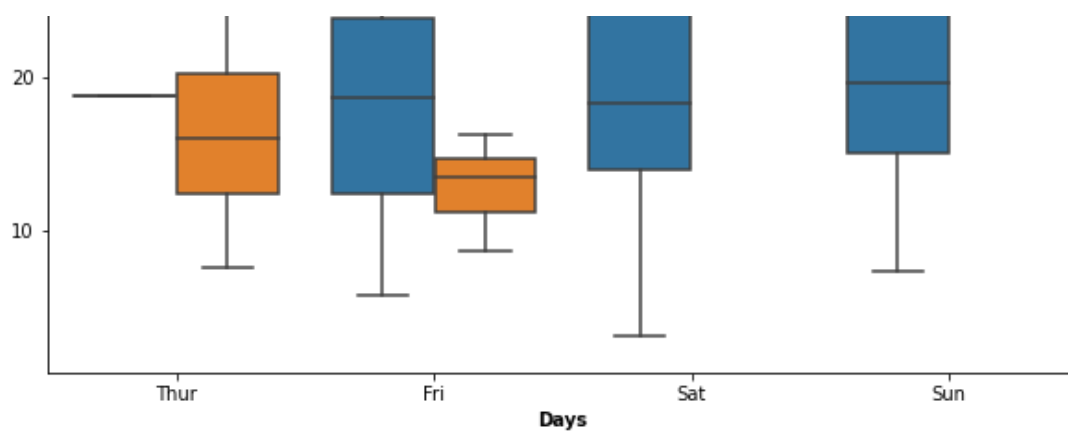
c:\Python\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[14]:

Text(0, 0.5, 'Total Bill')





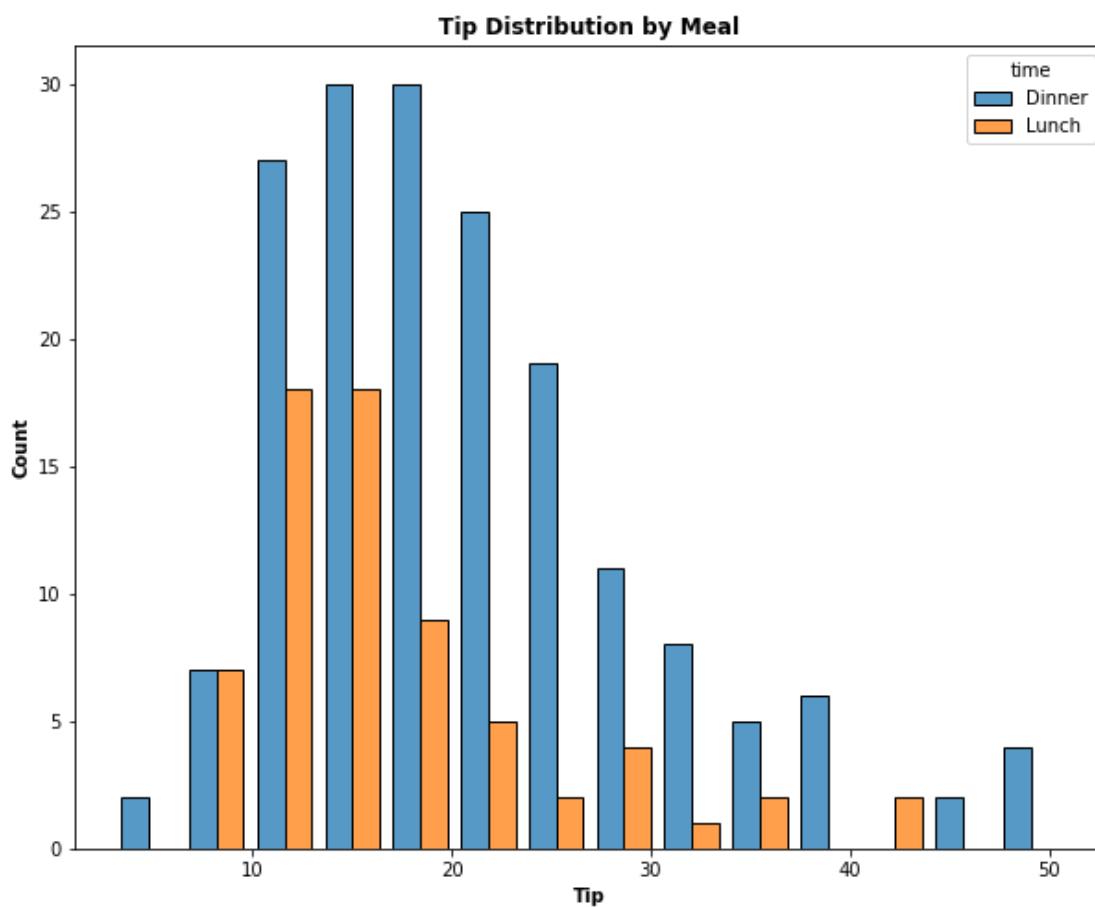
Question 8:

Create two histograms of the tip value based for Dinner and Lunch. They must be side by side.

In [15]:

```
fig = plt.figure(figsize=(10, 8))

sns.histplot(data=tp, x='total_bill', hue='time', multiple='dodge', shrink=0.8)
plt.title('Tip Distribution by Meal', weight='bold')
plt.xlabel('Tip', weight='bold')
plt.ylabel('Count', weight='bold')
plt.show()
```



Question 9:

Create two scatterplots graphs, one for Male and another for Female, presenting the total_bill value and tip relationship, differing by smoker or no smoker

In [16]:

```
tp['gender'] = tp['sex'].apply(lambda x: 'Male' if x == 'Male' else 'Female')
```

```
tp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   total_bill  244 non-null    float64
1   tip         244 non-null    float64
2   sex        244 non-null    object  
3   smoker      244 non-null    object  
4   day         244 non-null    object  
5   time       244 non-null    object  
6   size       244 non-null    int64  
7   gender     244 non-null    object  
dtypes: float64(2), int64(1), object(5)
memory usage: 15.4+ KB
```

```
In [17]:
```

```
fig = plt.figure(figsize=(10, 8))

sns.scatterplot(data=tp, x='total_bill', y='tip', hue='smoker', style='gender')
plt.title('Relation between Tips and Total Bill per Smoker and Gender', weight='bold')
plt.xlabel('Total Bill', weight='bold')
plt.ylabel('Tips', weight='bold')
plt.show
```

```
Out[17]:
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

