seaborn

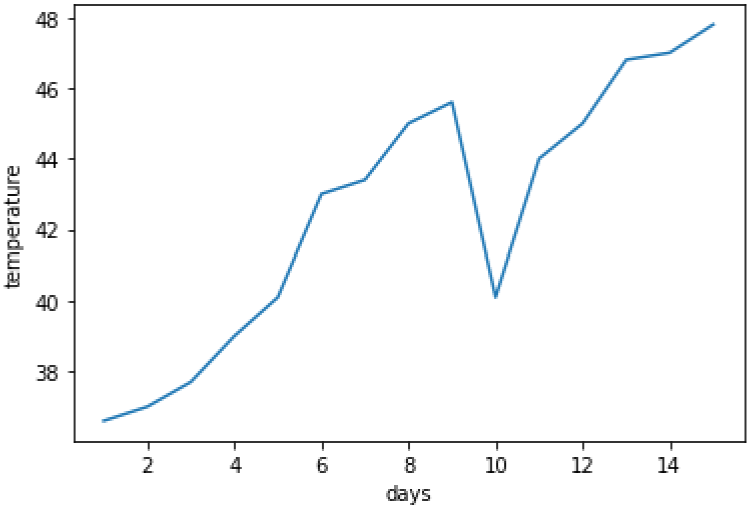
In [1]:

```
import pandas as pd
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
import seaborn as sns
```

Line Plot

In [2]:

```
days = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
temperature = [36.6,37,37.7,39,40.1,43,43.4,45,45.6,40.1,44,45,46.8,47,47.8]

temp_df = pd.DataFrame({'days':days,'temperature': temperature})
sns.lineplot(x = 'days',y = 'temperature',data = temp_df)
plt.show()
```



In [3]:

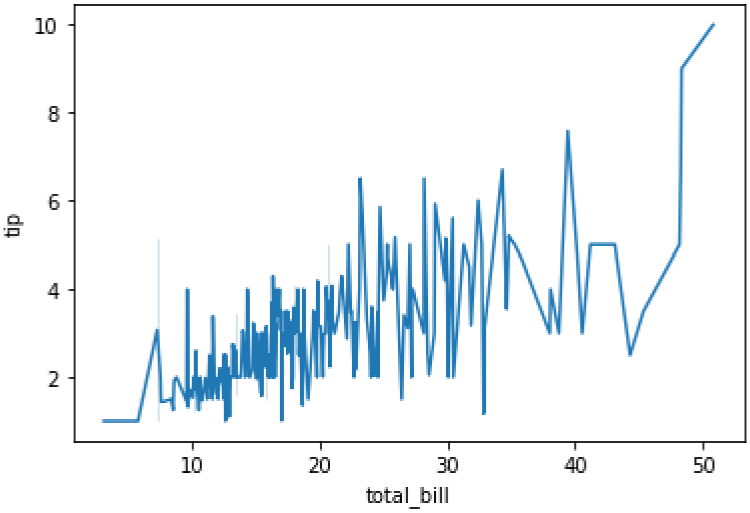
```
tips_df = pd.read_csv('tips.csv')
tips_df.head()
```

Out[3]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

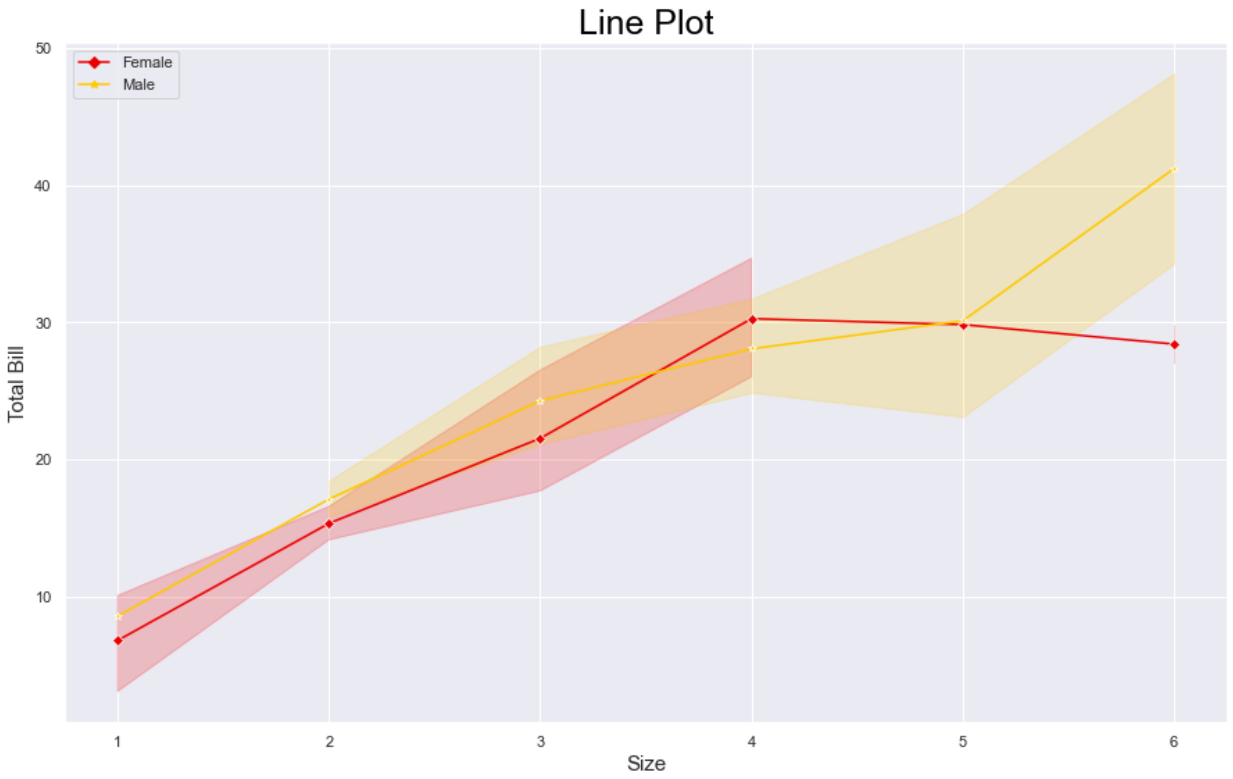
In [4]:

```
sns.lineplot(x = 'total_bill',y = 'tip',data = tips_df)
plt.show()
```

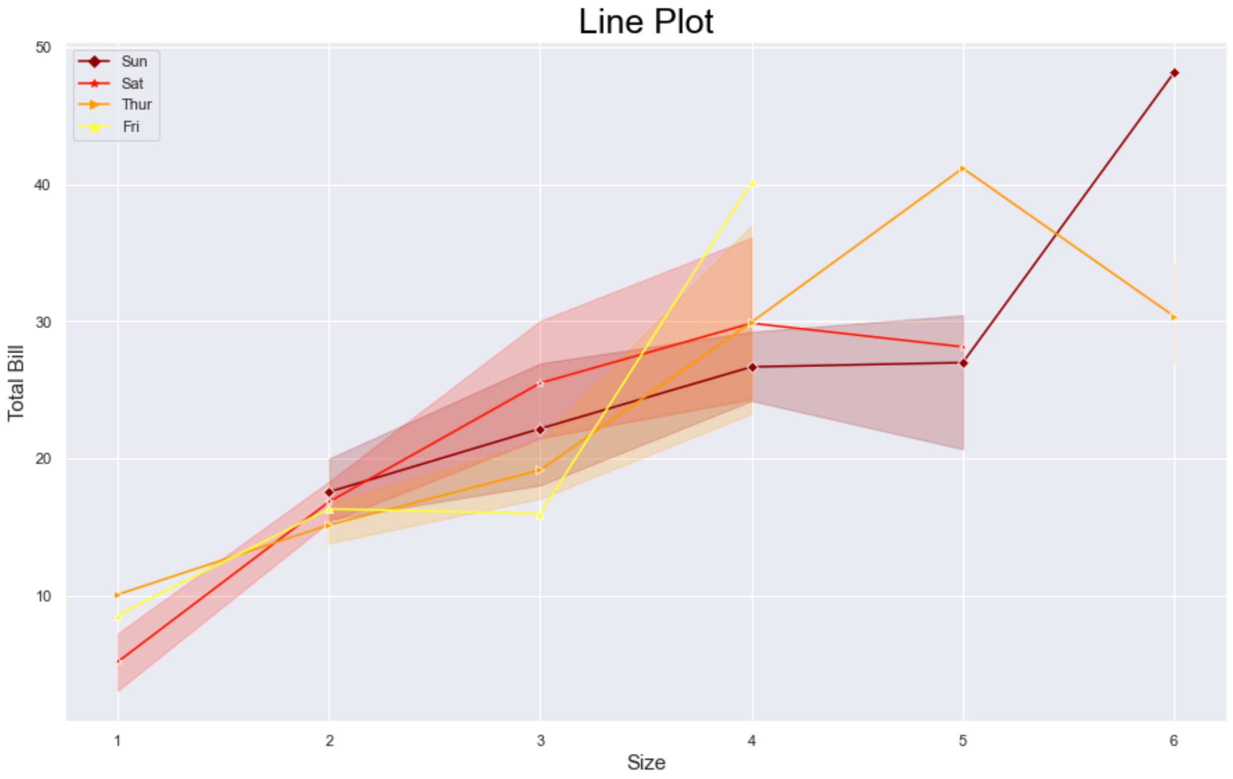


In [5]:

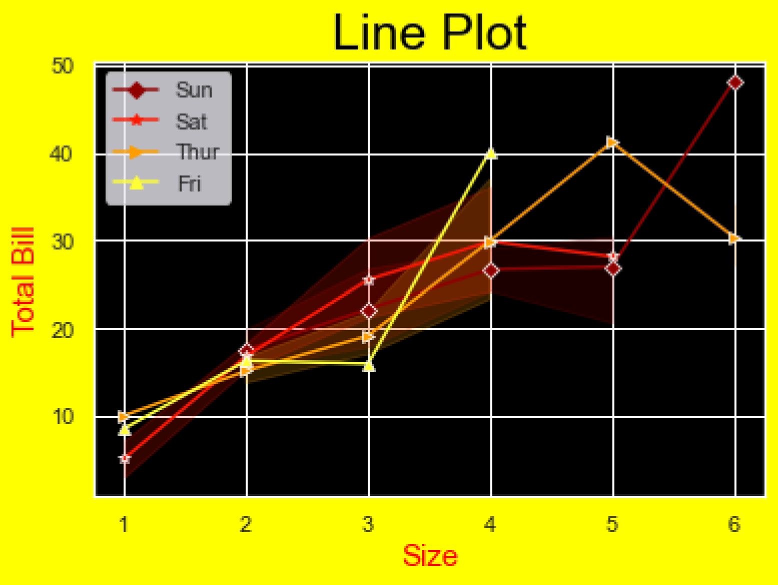
```
plt.figure(figsize = (15,9))
sns.set(style = 'darkgrid')
sns.lineplot(x = 'size',y = 'total_bill',data = tips_df,hue = 'sex',style =
# plt.title()
plt.title('Line Plot',fontsize = 25,color = 'black')
plt.xlabel('Size',fontsize = 15)
plt.ylabel('Total Bill',fontsize = 15)
plt.legend(loc = 2)
plt.show()
```



In [6]:
```
plt.figure(figsize = (15,9))
sns.set(style = 'darkgrid')
sns.lineplot(x = 'size',y = 'total_bill',data = tips_df,hue = 'day',style =
# plt.title()
plt.title('Line Plot',fontsize = 25,color = 'black')
plt.xlabel('Size',fontsize = 15)
plt.ylabel('Total Bill',fontsize = 15)
plt.legend(loc = 2)
plt.show()
```



```
In [7]: plt.figure(facecolor = 'yellow')
        ax = plt.axes()
        ax.set_facecolor('black')
        # plt.figure(figsize = (15,9))
        sns.set(style = 'darkgrid')
        sns.lineplot(x = 'size',y = 'total_bill',data = tips_df,hue = 'day',style =
        # plt.title()
        plt.title('Line Plot',fontsize = 25,color = 'black')
        plt.xlabel('Size',fontsize = 15,color = 'red')
        plt.ylabel('Total Bill',fontsize = 15,color = 'red')
        plt.legend(loc = 2)
        plt.show()
```



Histogram & Distplot

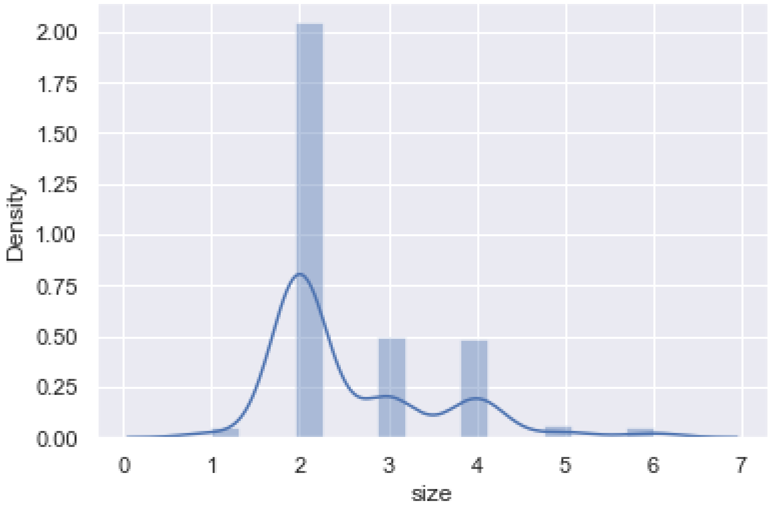
```
In [8]: tips_df.head()
```

Out[8]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

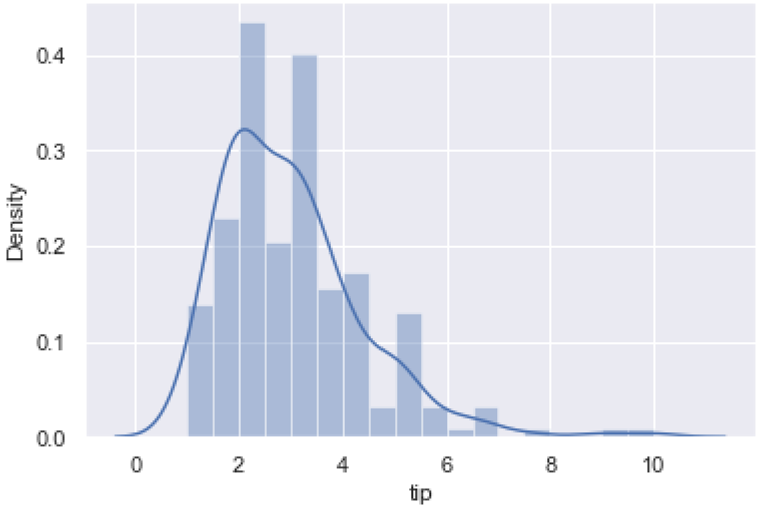
```
In [9]: sns.distplot(tips_df['size'])
        plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



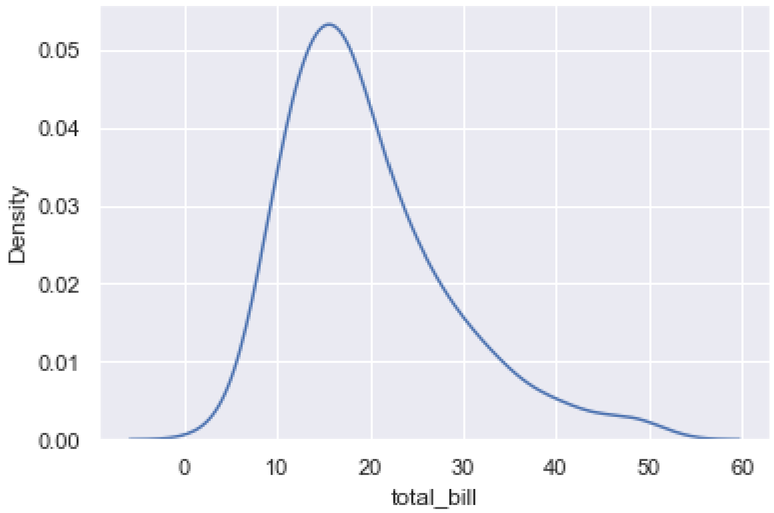
In [10]:
```
sns.distplot(tips_df['tip'])
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [11]:
```
sns.distplot(tips_df['total_bill'],hist = False)
plt.show()
```

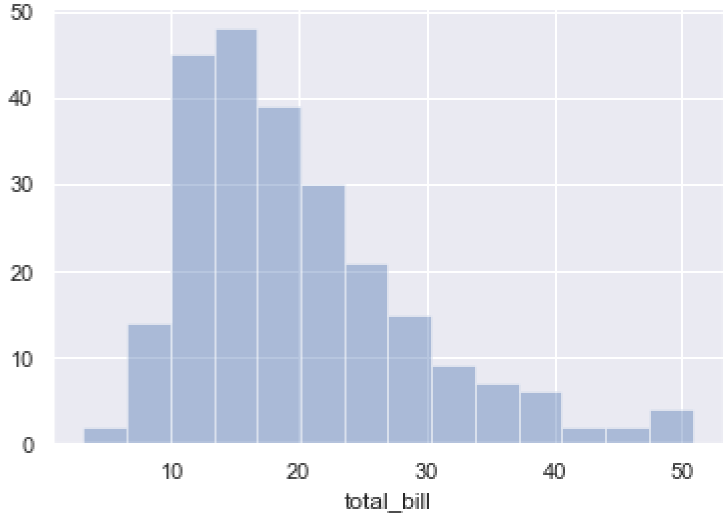
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```



In [12]:

```
sns.distplot(tips_df['total_bill'],kde = False)
plt.show()
```

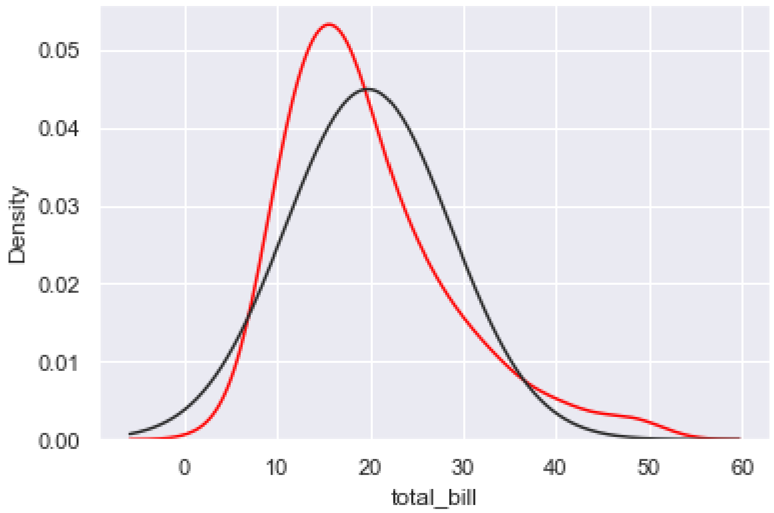
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [13]:

```
sns.distplot(tips_df['total_bill'],fit = norm,hist = False,color = 'red') #,
plt.show()
```

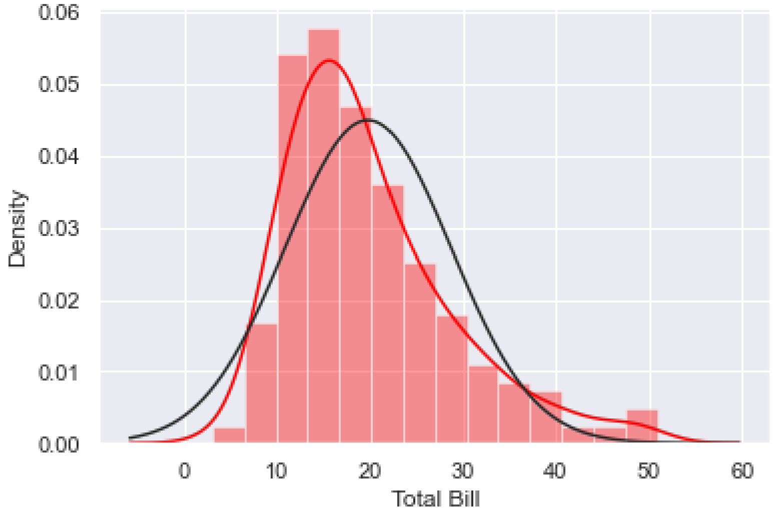
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```



In [14]:

```
sns.distplot(tips_df['total_bill'],fit = norm,hist = True,color = 'red',axlab
plt.show()
```

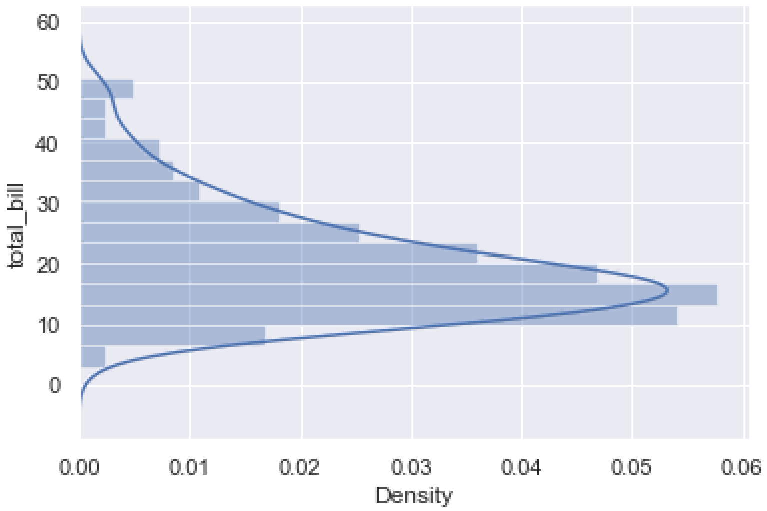
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [15]:

```
sns.distplot(tips_df['total_bill'],vertical = True)
plt.show()
```

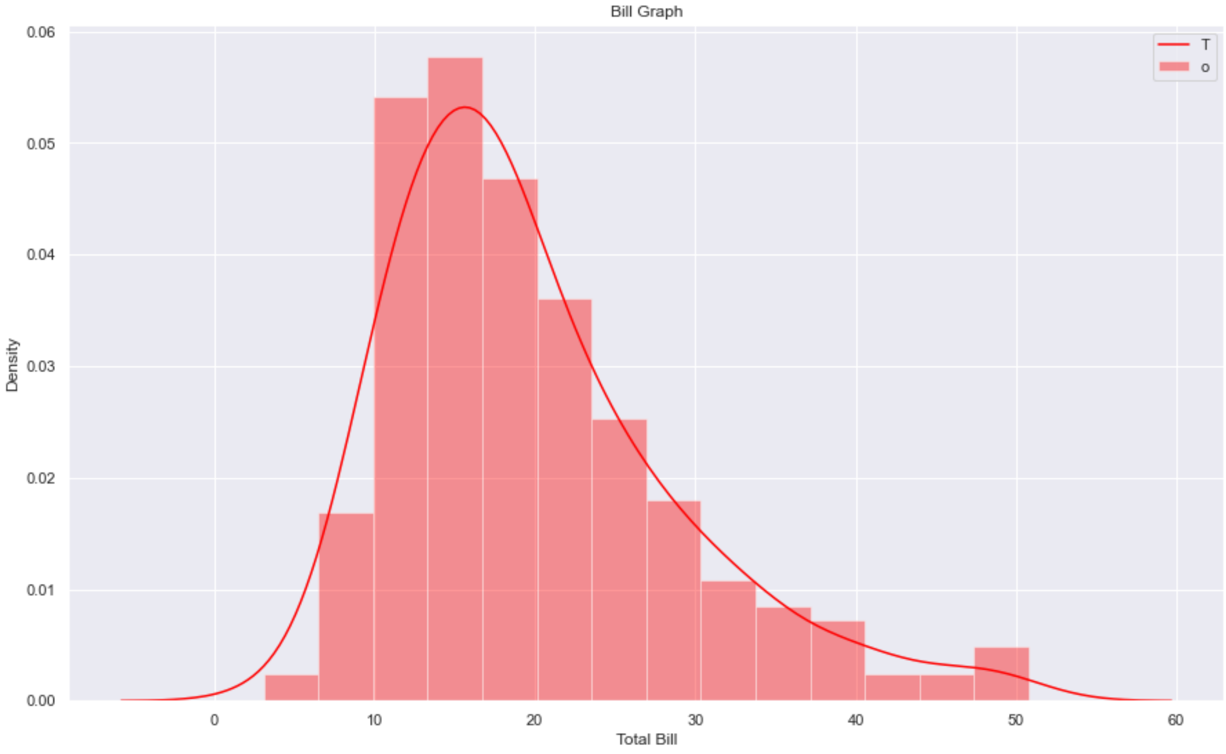
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:1689: FutureWarning: The `vertical` parameter
is deprecated and will be removed in a future version. Assign the data to th
e `y` variable instead.
  warnings.warn(msg, FutureWarning)
```



In [16]:

```
# plt.figure(facecolor = 'red')
# ax = plt.axes()
# ax.set_facecolor('black')
plt.figure(figsize = (15,9))
sns.distplot(tips_df['total_bill'],color = 'red',axlabel = 'Total Bill') #,ru
plt.title('Bill Graph')
plt.legend('Total Bill')
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

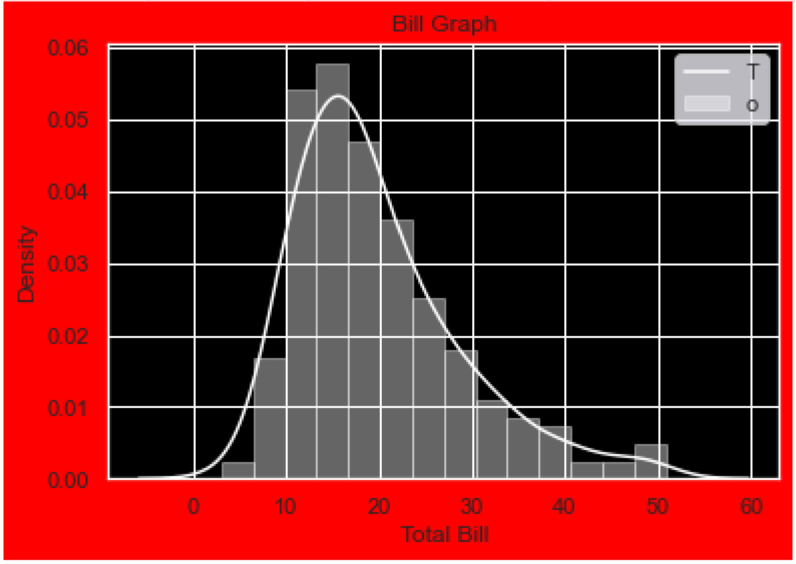


In [17]:

```
plt.figure(facecolor = 'red')
ax = plt.axes()
ax.set_facecolor('black')
```

```
# plt.figure(figsize = (15,9))
sns.distplot(tips_df['total_bill'],color = 'white',axlabel = 'Total Bill') #
plt.title('Bill Graph')
plt.legend('Total Bill')
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

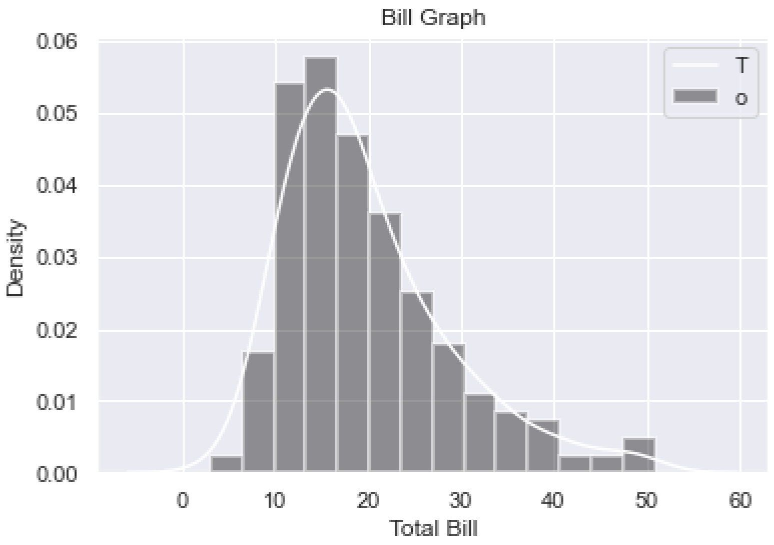


In [18]:
```
tips_df.total_bill.sort_values()
```

Out[18]:
```
67      3.07
92      5.75
111     7.25
172     7.25
149     7.51
        ...
182    45.35
156    48.17
59     48.27
212    48.33
170    50.81
Name: total_bill, Length: 244, dtype: float64
```

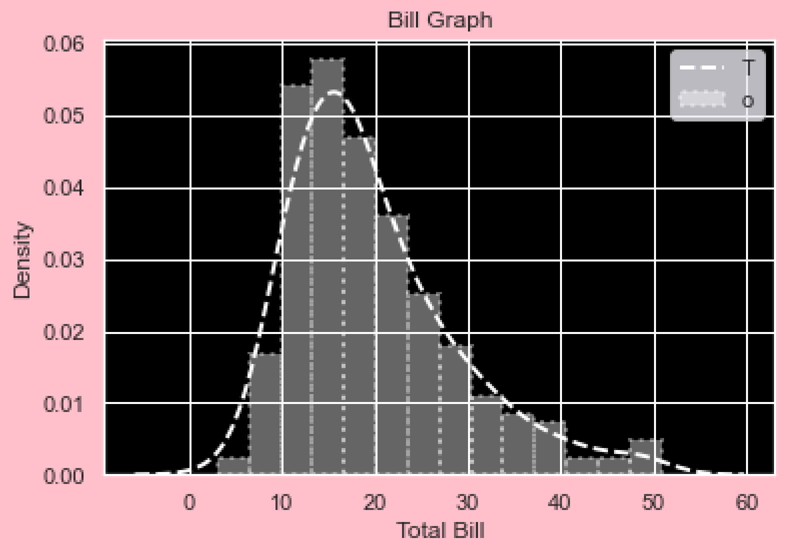
In [19]:
```
# bins = [1,5,10,15,20,25,30,35,40,45,50,55]

# plt.figure(figsize = (15,9))
sns.distplot(tips_df['total_bill'],color = 'white',axlabel = 'Total Bill',his
plt.title('Bill Graph')
# plt.xticks(bins)
plt.legend('Total Bill')
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [20]:
```
plt.figure(facecolor = 'pink')
ax = plt.axes()
ax.set_facecolor('black')
sns.distplot(tips_df['total_bill'],color = 'white',axlabel = 'Total Bill',his
plt.title('Bill Graph')
# plt.xticks(bins)
plt.legend('Total Bill')
plt.show()
```

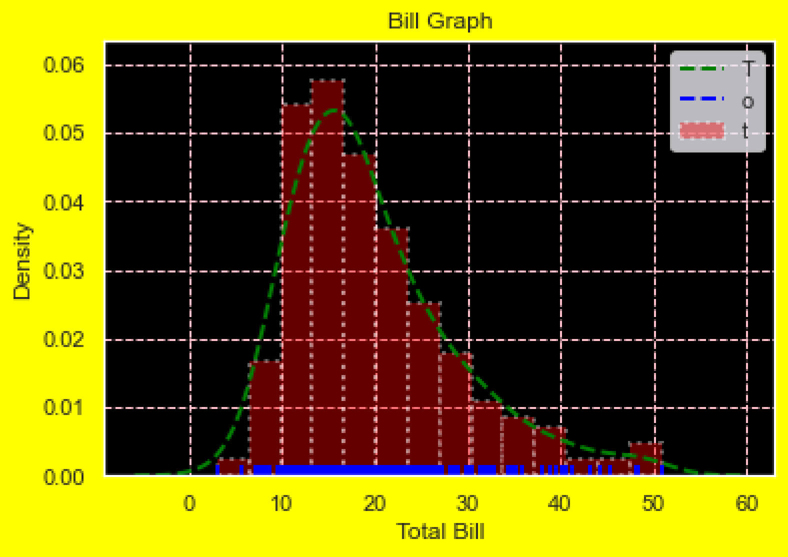
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [21]:
```
plt.figure(facecolor = 'yellow')
ax = plt.axes()
ax.set_facecolor('black')
sns.distplot(tips_df['total_bill'],color = 'red',axlabel = 'Total Bill',
                            hist_kws = {'color': 'red','linewidth': 2,'linest
                                kde_kws = {'color': 'green','linewidth': 2,'l
                                    rug = True,rug_kws = {'color': 'blue','li
plt.title('Bill Graph')
```

```
# plt.xticks(bins)
plt.legend('Total Bill')
plt.grid(color = 'pink',linestyle = '--',linewidth = 1)
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2103: FutureWarning: The `axis` variable is no
longer used and will be removed. Instead, assign variables directly to `x` o
r `y`.
  warnings.warn(msg, FutureWarning)
```



Bar Plot

In [22]:

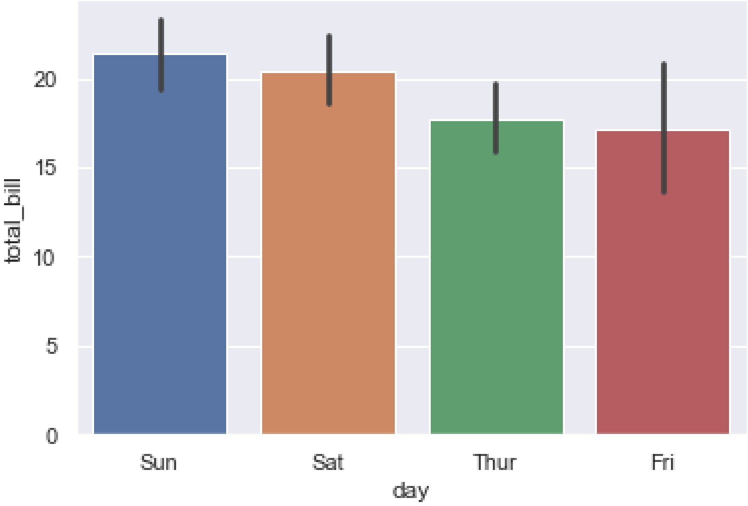
```
tips_df.head()
```

Out[22]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

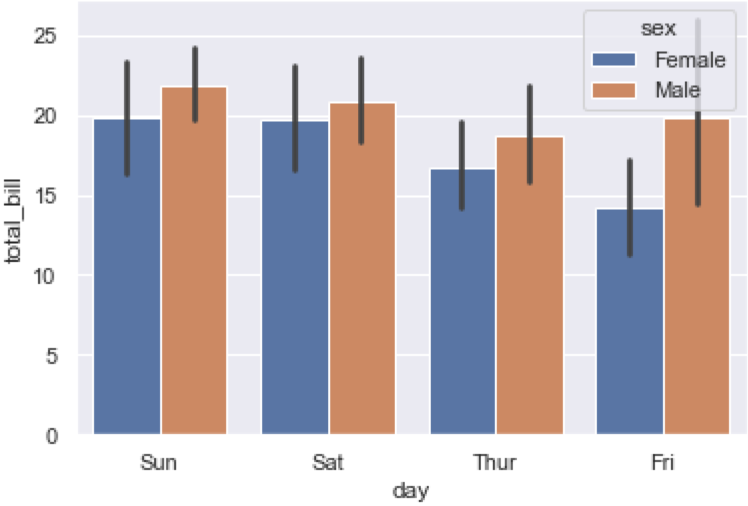
In [23]:

```
sns.barplot(x = tips_df.day,y = tips_df.total_bill)
plt.show()
```



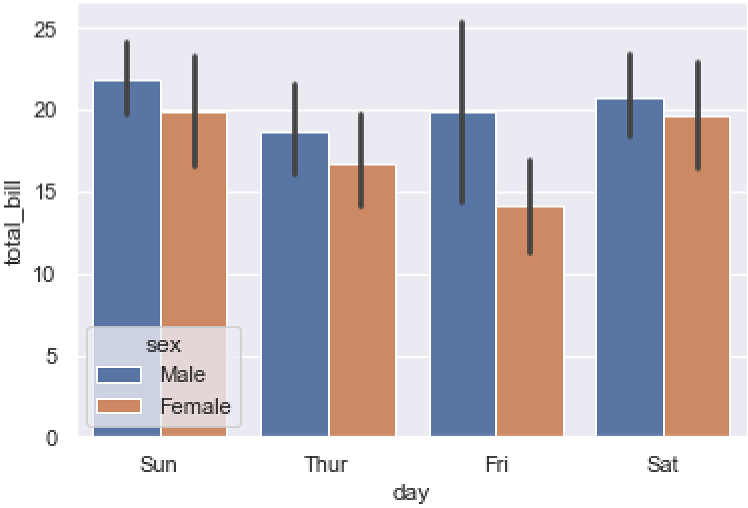
In [24]:

```
sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df)
plt.show()
```



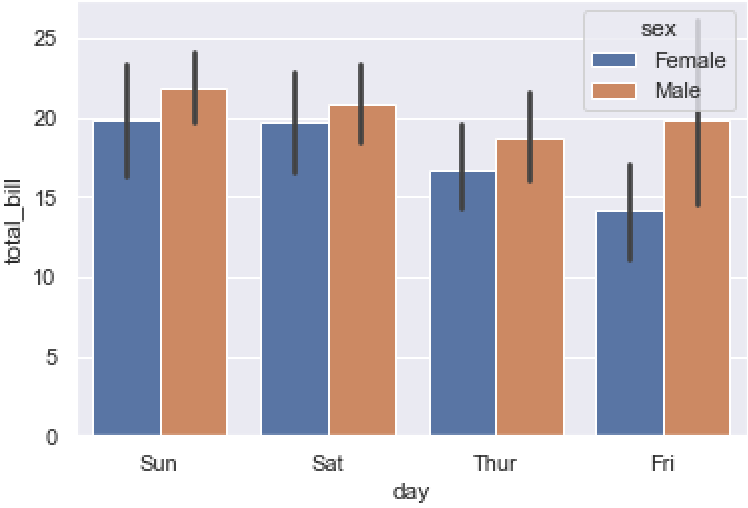
In [25]:

```
order = ['Sun','Thur','Fri','Sat']
hue_order = ['Male','Female']

sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,order = ord
plt.show()
```



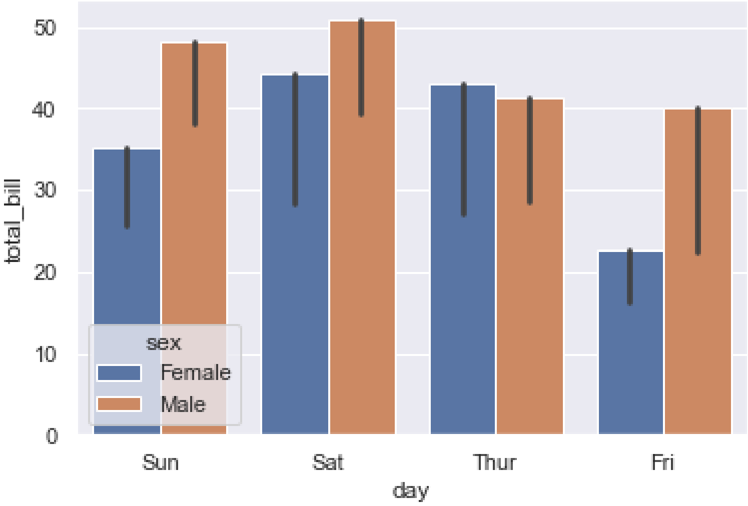
In [26]:

```
# Mean
sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,estimator =
plt.show()
```



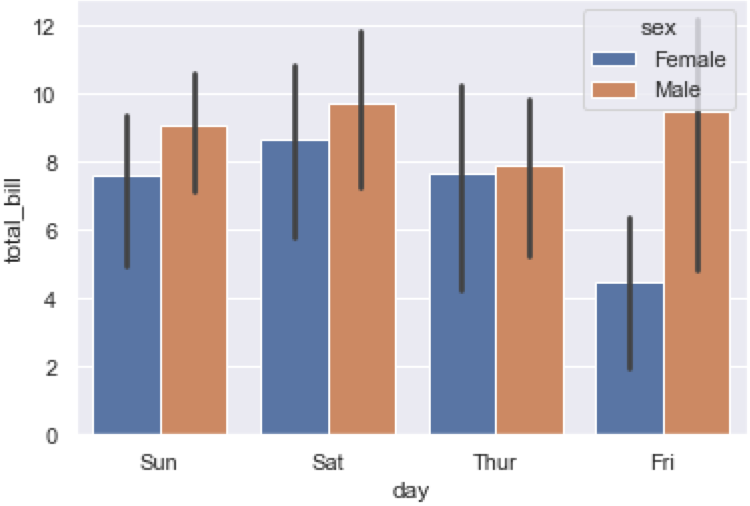
In [27]:

```
# Max
sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,estimator =
plt.show()
```

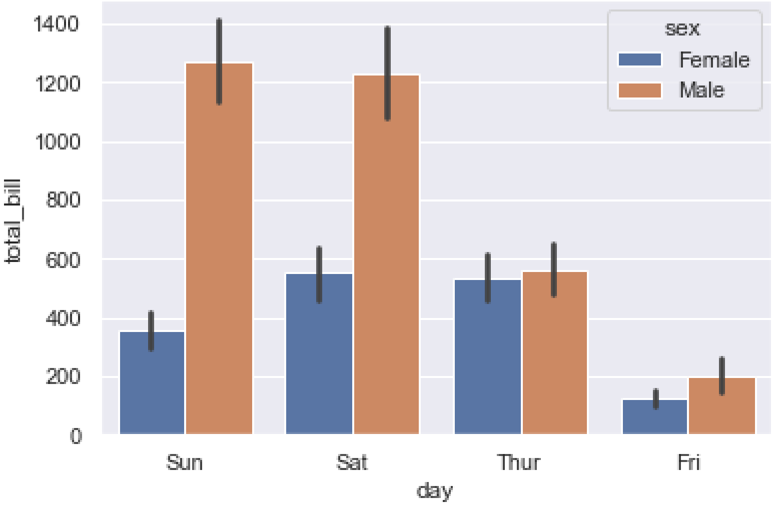


In [28]:

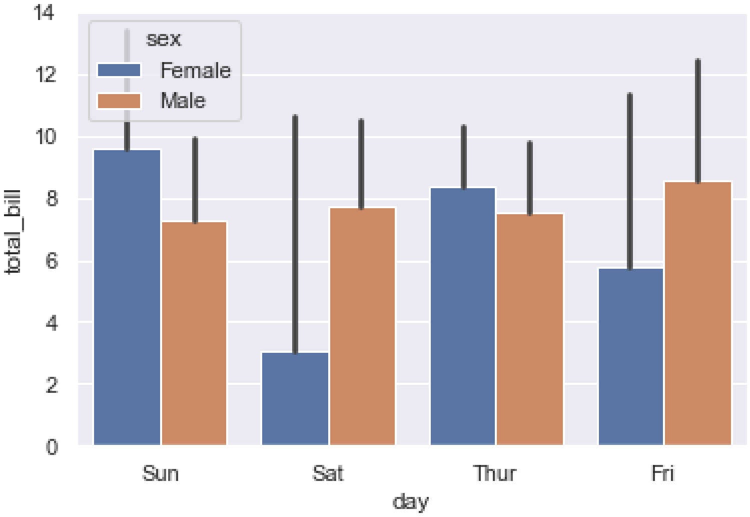
```
# STD
sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,estimator =
plt.show()
```



In [29]:
```
# Sum
sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,estimator =
plt.show()
```



In [30]:
```
# Min
sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,estimator =
plt.show()
```



In [31]:
```
# sns.barplot(x = tips_df.day,y = tips_df.total_bill,estimator = np.mean)
# plt.show()
```

In [32]:
```
# sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,ci = 10,n_
# plt.show()
```

In [33]:
```
# sns.barplot(y = 'day',x = 'total_bill',hue = 'sex',data = tips_df)
```

In [34]:
```
# sns.barplot(x = 'total_bill',y = 'size',hue = 'sex',data = tips_df,orient =
```

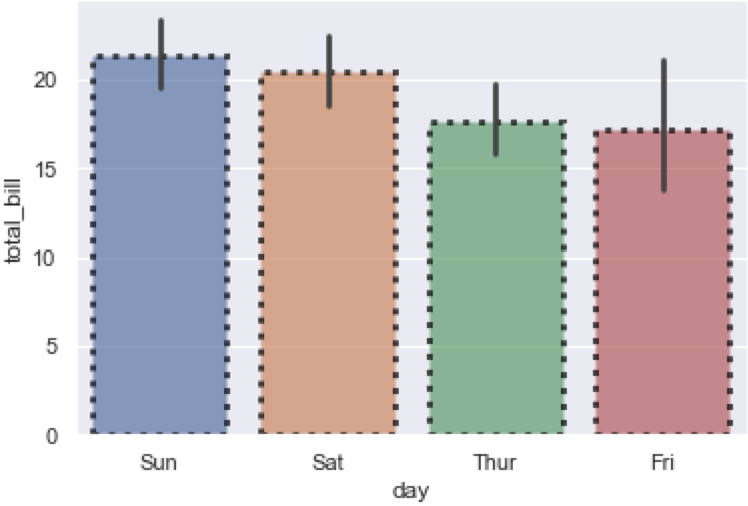
In [35]:
```
# sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,palette =
```

In [36]:
```
# sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,saturation
```

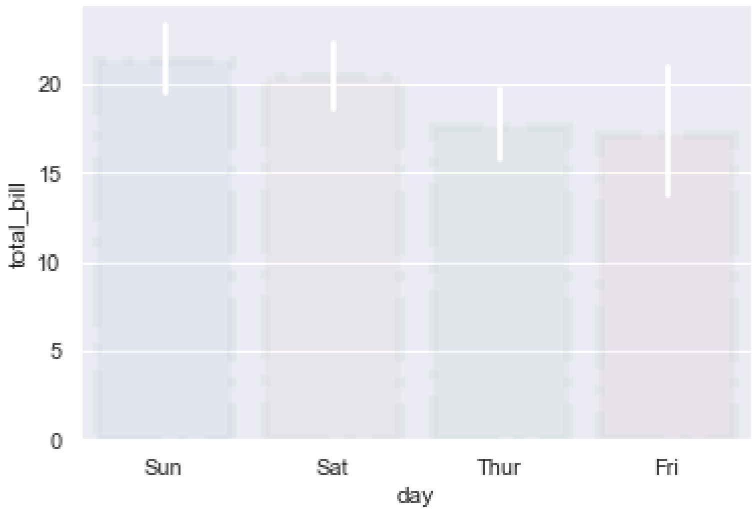
In [37]:
```
# sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,capsize =
```

In [38]:
```
# sns.barplot(x = 'day',y = 'total_bill',hue = 'sex',data = tips_df,dodge = Fa
```

In [39]:
```
kwargs = {'alpha': 0.7,'linestyle':':','linewidth':3,'edgecolor':'black'}
sns.barplot(x = 'day',y = 'total_bill',data = tips_df,**kwargs) # ,hue = 'sex
plt.show()
```

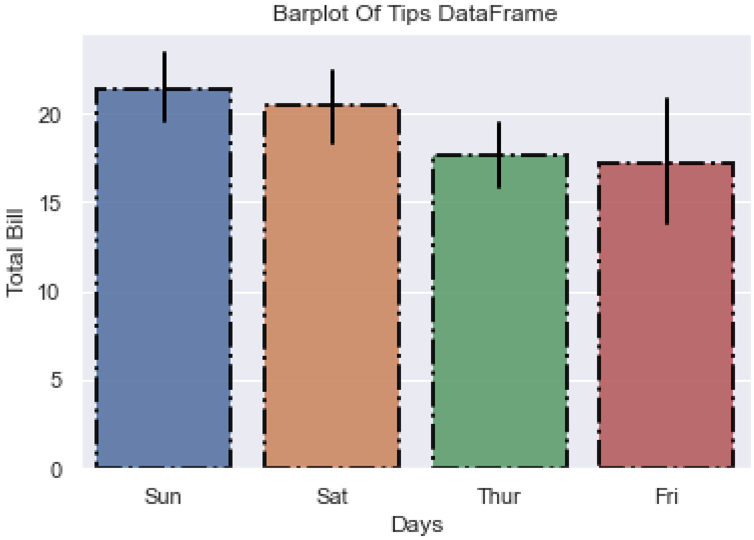


In [40]:
```
sns.barplot(x = 'day',y = 'total_bill',data = tips_df,alpha = .05,linestyle =
plt.show()
```

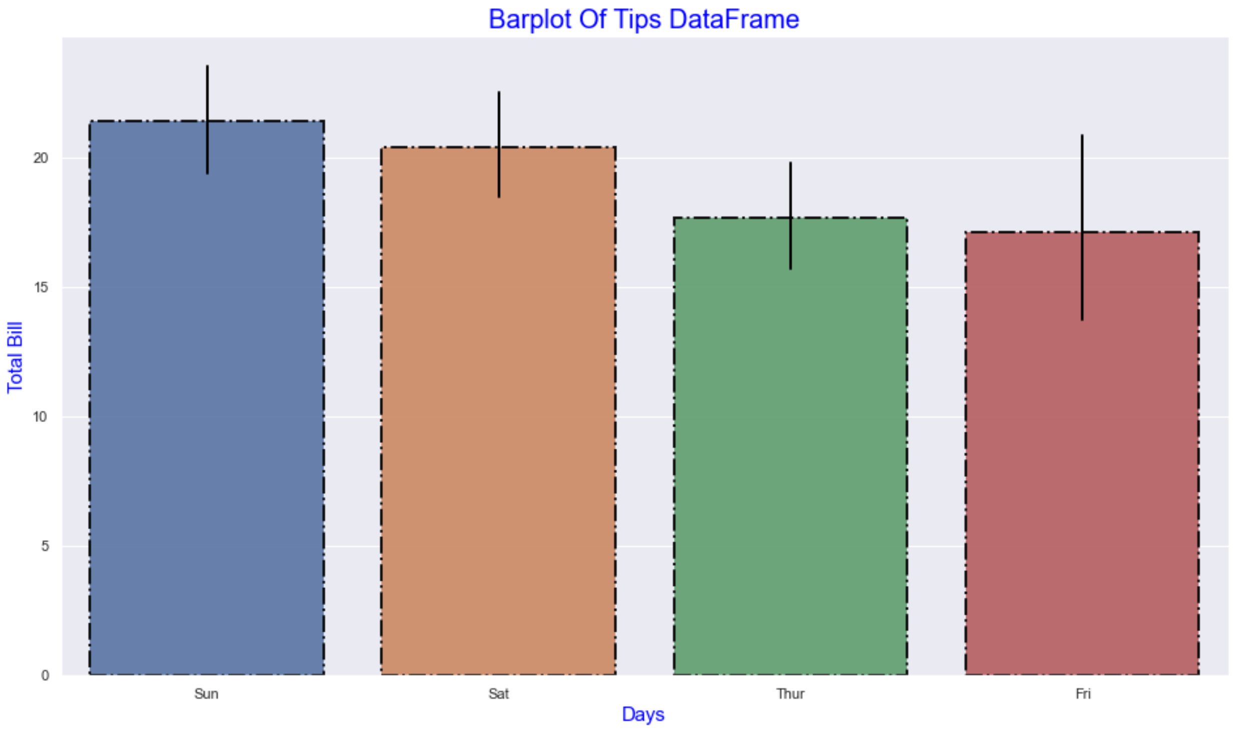


```
In [41]: ax = sns.barplot(x = 'day',y = 'total_bill',data = tips_df,
                          alpha = .9,linestyle = '-.',linewidth = 2,
                              edgecolor = 'black',errcolor = 'black',
                                  errwidth = 2)
         ax.set(title = 'Barplot Of Tips DataFrame',
                          xlabel = 'Days',
                              ylabel = 'Total Bill')
```

```
Out[41]: [Text(0.5, 1.0, 'Barplot Of Tips DataFrame'),
          Text(0.5, 0, 'Days'),
          Text(0, 0.5, 'Total Bill')]
```

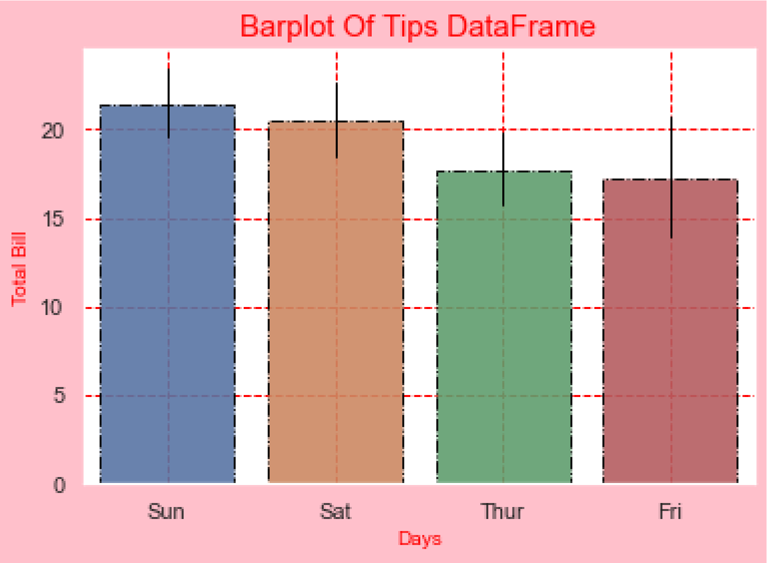


```
In [42]: plt.figure(figsize = (16,9))

         sns.barplot(x = 'day',y = 'total_bill',data = tips_df,
                          alpha = .9,linestyle = '-.',linewidth = 2,
                              edgecolor = 'black',errcolor = 'black',
                                  errwidth = 2)
         plt.title('Barplot Of Tips DataFrame',fontsize = 20,color = 'blue')
         plt.xlabel('Days',fontsize = 15,color = 'blue')
         plt.ylabel('Total Bill',fontsize = 15,color = 'blue')
         # plt.savefig()
         plt.show()
```



In [43]:

```
plt.figure(facecolor = 'pink')
ax = plt.axes()
ax.set_facecolor('white')

#plt.figure(figsize = (16,9))

sns.barplot(x = 'day',y = 'total_bill',data = tips_df,
                alpha = .9,linestyle = '-.',linewidth = 1,
                    edgecolor = 'black',errcolor = 'black',
                        errwidth = 1)
plt.title('Barplot Of Tips DataFrame',fontsize = 15,color = 'red')
plt.xlabel('Days',fontsize = 10,color = 'red')
plt.ylabel('Total Bill',fontsize = 10,color = 'red')
plt.grid(color = 'red',linestyle = '--',linewidth = 1)
# plt.savefig()
plt.show()
```

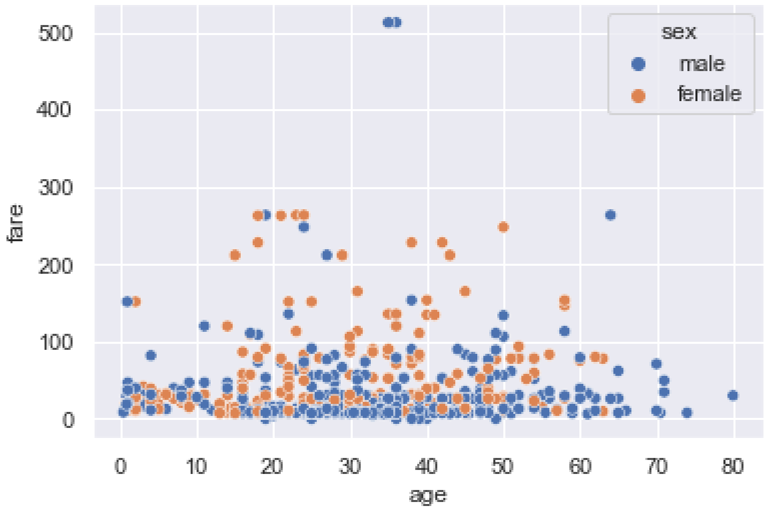


Scatter Plot

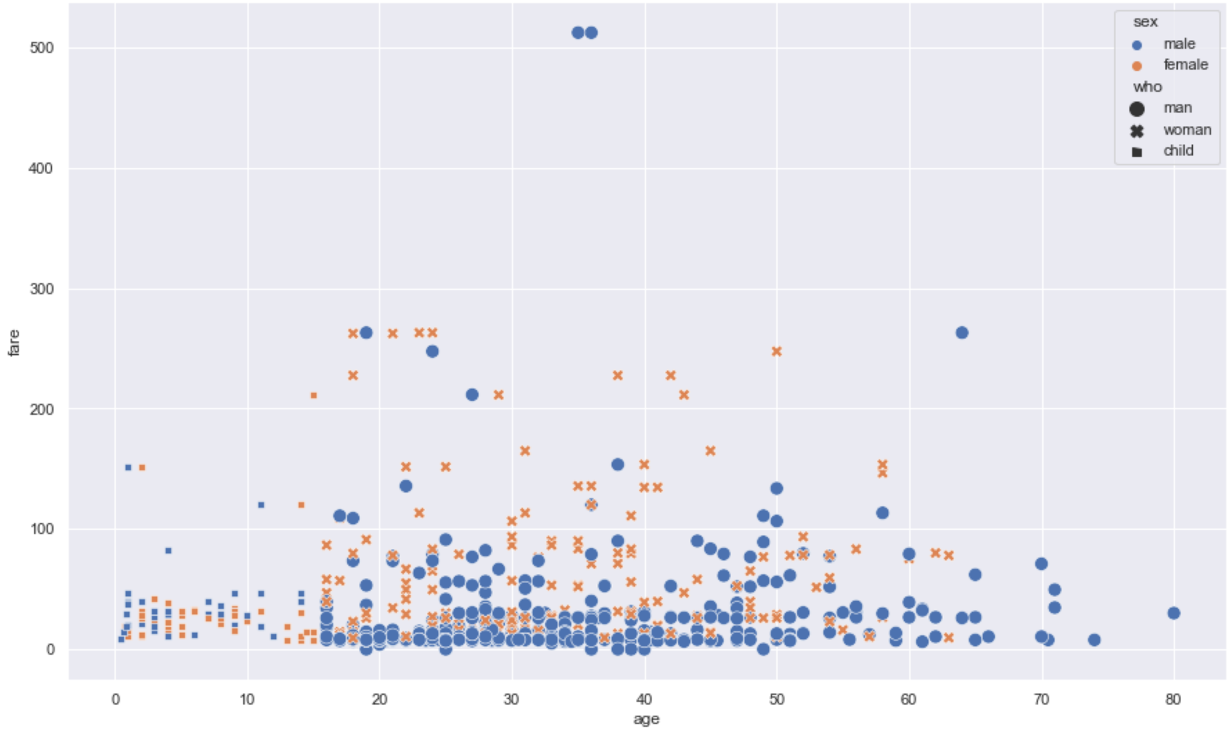
```
In [44]: titanic_df = pd.read_csv('titanic.csv')
         titanic_df.head()
```

```
Out[44]:
     survived  pclass     sex   age  sibsp  parch     fare  embarked  class    who  adult_male  de
  0         0       3    male  22.0      1      0   7.2500         S  Third    man        True  N
  1         1       1  female  38.0      1      0  71.2833         C  First  woman       False
  2         1       3  female  26.0      0      0   7.9250         S  Third  woman       False  N
  3         1       1  female  35.0      1      0  53.1000         S  First  woman       False
  4         0       3    male  35.0      0      0   8.0500         S  Third    man        True  N
```

```
In [45]: sns.scatterplot(x = 'age',y = 'fare',data = titanic_df,hue = 'sex')
         plt.show()
```

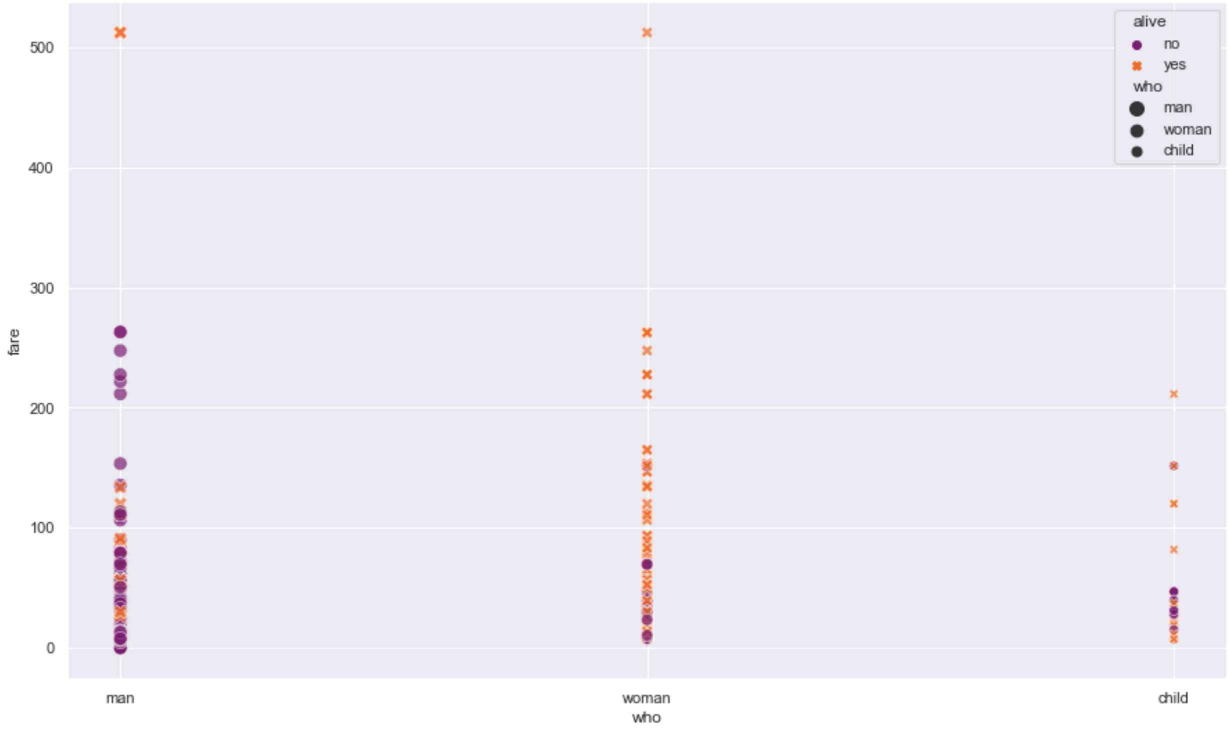


```
In [46]: plt.figure(figsize = (15,9))
         sns.scatterplot(x = 'age',y = 'fare',data = titanic_df,hue = 'sex',style = 'w
         plt.show()
```



```
In [47]: # plt.figure(figsize = (15,9))
         # sns.scatterplot(x = 'who',y = 'fare',data = titanic_df,hue = 'alive',style =
         # plt.show()
```

```
In [48]: plt.figure(figsize = (15,9))
         sns.scatterplot(x = 'who',y = 'fare',data = titanic_df,
                         hue = 'alive',style = 'alive',size = 'who',
                             sizes = (50,100),palette = 'inferno',alpha = .7)
         plt.show()
```



```
In [49]: # plt.figure(facecolor = 'pink')
```

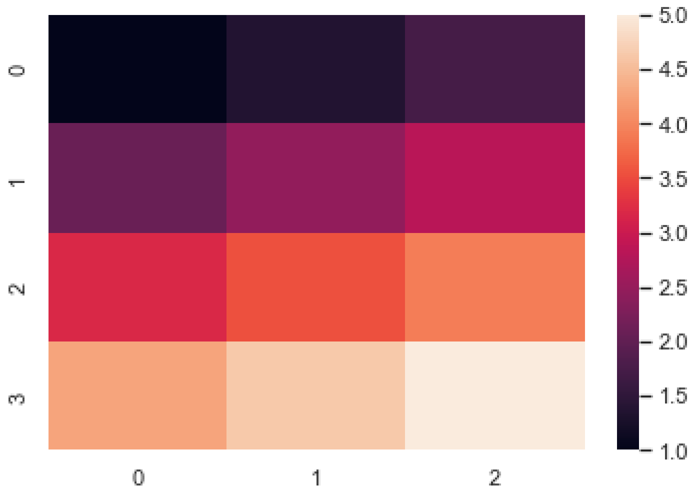
```
# ax = plt.axes()
# ax.set_facecolor('white')
#
# sns.scatterplot(x = 'age',y = 'fare',data = titanic_df,hue = 'sex')
# plt.grid(color = 'black',linestyle = '--',linewidth = 1)
# plt.show()
```

Heatmap

In [50]:
```
arr_2d = np.linspace(1,5,12).reshape(4,3)
arr_2d
```

Out[50]:
```
array([[1.        , 1.36363636, 1.72727273],
       [2.09090909, 2.45454545, 2.81818182],
       [3.18181818, 3.54545455, 3.90909091],
       [4.27272727, 4.63636364, 5.        ]])
```

In [51]:
```
sns.heatmap(arr_2d)
plt.show()
```



In [52]:
```
globalwarming_df = pd.read_csv('Who_is_responsible_for_global_warming.csv')
globalwarming_df.head()
```

Out[52]:

	Country Name	Country Code	Indicator Name	Indicator Code	2000	2001	2002	2003
0	United States	USA	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	20.178751	19.636505	19.613404	19.564105
1	United Kingdom	GBR	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	9.199549	9.233175	8.904123	9.053278
2	India	IND	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	0.979870	0.971698	0.967381	0.992392
3	China	CHN	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	2.696862	2.742121	3.007083	3.524074
4	Russian Federation	RUS	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	10.627121	10.669603	10.715901	11.090647

In [53]:

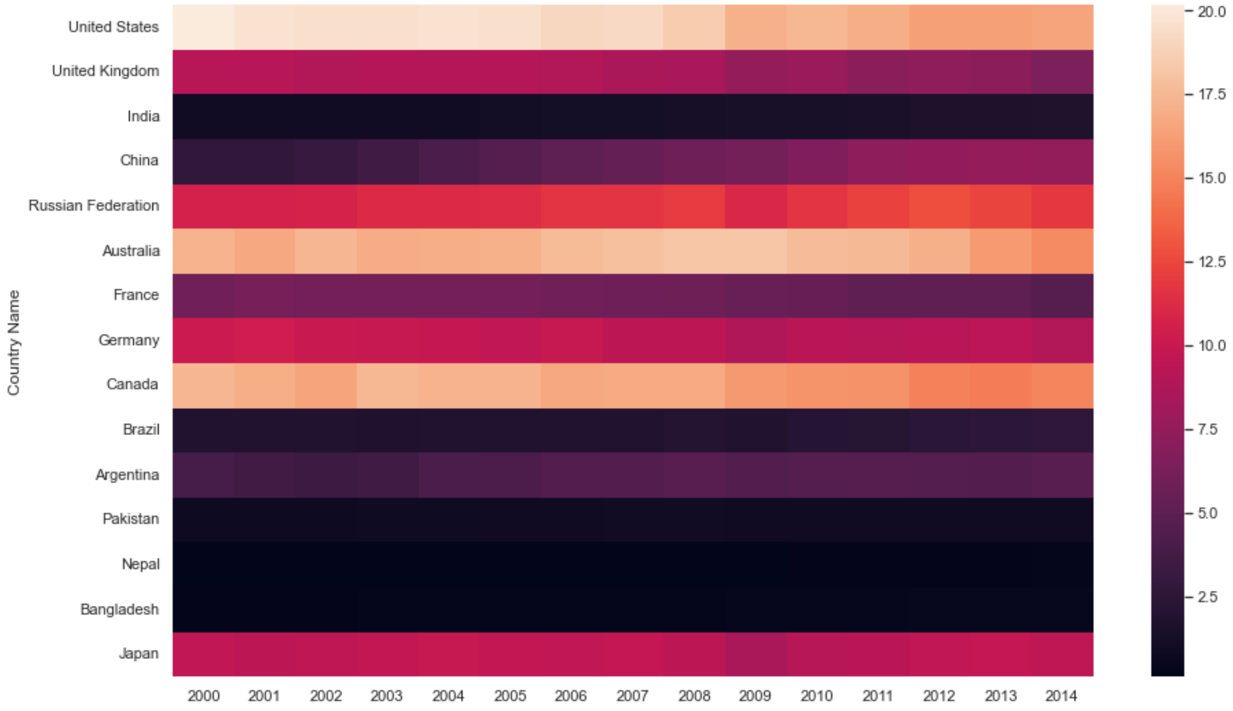
```
globalwarming_df = globalwarming_df.drop(columns =['Country Code','Indicator
globalwarming_df.head()
```

Out[53]:

	2000	2001	2002	2003	2004	2005	2006	2
Country Name								
United States	20.178751	19.636505	19.613404	19.564105	19.658371	19.591885	19.094067	19.217
United Kingdom	9.199549	9.233175	8.904123	9.053278	8.989140	8.982939	8.898710	8.617
India	0.979870	0.971698	0.967381	0.992392	1.025028	1.068563	1.121982	1.193
China	2.696862	2.742121	3.007083	3.524074	4.037991	4.523178	4.980314	5.334
Russian Federation	10.627121	10.669603	10.715901	11.090647	11.120627	11.253529	11.669122	11.672

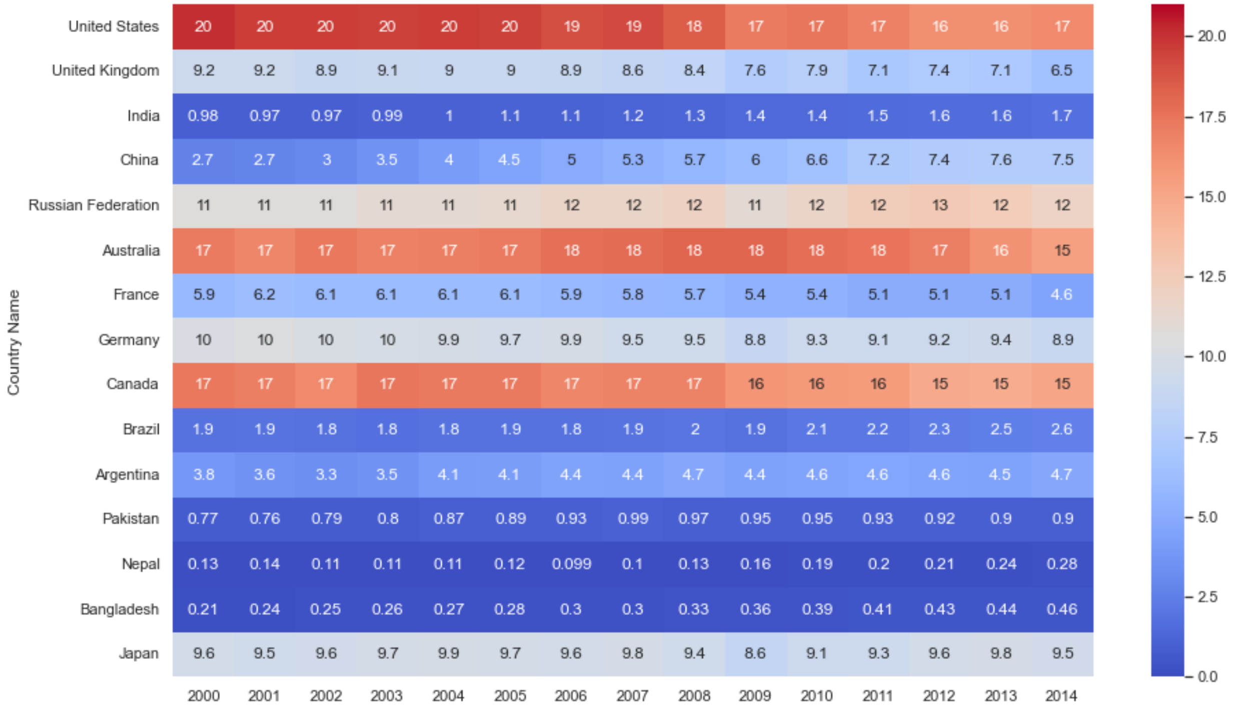
In [54]:

```
plt.figure(figsize = (15,9))
sns.heatmap(globalwarming_df)
plt.show()
```



In [55]:
```
# plt.figure(figsize = (15,9))
# sns.heatmap(globalwarming_df,vmin = 0,vmax = 21,cmap = 'coolwarm')
# plt.show()
```

In [56]:
```
plt.figure(figsize = (15,9))
sns.heatmap(globalwarming_df,vmin = 0,vmax = 21,cmap = 'coolwarm',annot = Tru
plt.show()
```

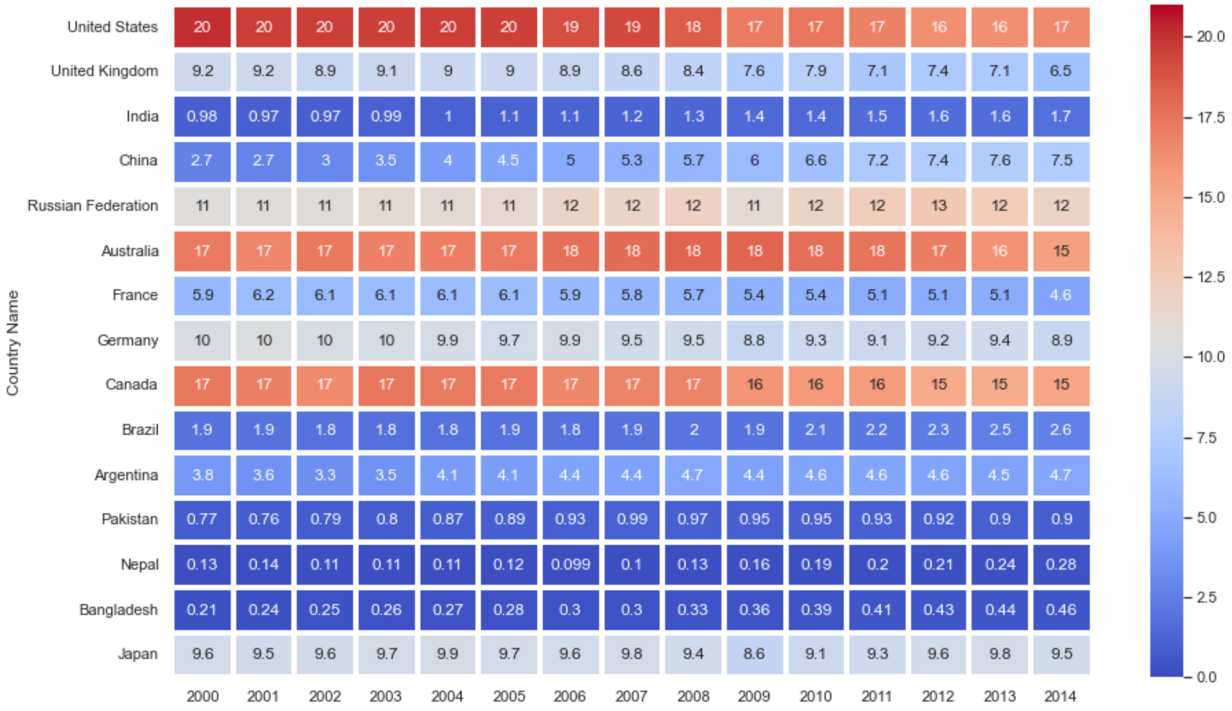


In [88]:
```
# annot_arr = np.array([['a00','a01','a02'],['a10','a11','a12'],['a20','a21','
# annot_arr
```

In [87]:
```
# sns.heatmap(arr_2d,annot = annot_arr,fmt = 's')
# plt.show()
```

In [86]:
```
# plt.figure(figsize = (15,9))
# annot_kws = {'fontsize': 10,'fontstyle': 'italic','color': 'black','alpha':
# sns.heatmap(globalwarming_df,vmin = 0,vmax = 21,cmap = 'coolwarm',annot = Tr
# plt.show()
```

In [60]:
```
plt.figure(figsize = (15,9))
sns.heatmap(globalwarming_df,vmin = 0,vmax = 21,cmap = 'coolwarm',annot = Tru
plt.show()
```



Country Name	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
United States	20	20	20	20	20	20	19	19	18	17	17	17	16	16	17
United Kingdom	9.2	9.2	8.9	9.1	9	9	8.9	8.6	8.4	7.6	7.9	7.1	7.4	7.1	6.5
India	0.98	0.97	0.97	0.99	1	1.1	1.1	1.2	1.3	1.4	1.4	1.5	1.6	1.6	1.7
China	2.7	2.7	3	3.5	4	4.5	5	5.3	5.7	6	6.6	7.2	7.4	7.6	7.5
Russian Federation	11	11	11	11	11	11	12	12	12	11	12	12	13	12	12
Australia	17	17	17	17	17	17	18	18	18	18	18	18	17	16	15
France	5.9	6.2	6.1	6.1	6.1	6.1	5.9	5.8	5.7	5.4	5.4	5.1	5.1	5.1	4.6
Germany	10	10	10	10	9.9	9.7	9.9	9.5	9.5	8.8	9.3	9.1	9.2	9.4	8.9
Canada	17	17	17	17	17	17	17	17	17	16	16	16	15	15	15
Brazil	1.9	1.9	1.8	1.8	1.8	1.9	1.8	1.9	2	1.9	2.1	2.2	2.3	2.5	2.6
Argentina	3.8	3.6	3.3	3.5	4.1	4.1	4.4	4.4	4.7	4.4	4.6	4.6	4.6	4.5	4.7
Pakistan	0.77	0.76	0.79	0.8	0.87	0.89	0.93	0.99	0.97	0.95	0.95	0.93	0.92	0.9	0.9
Nepal	0.13	0.14	0.11	0.11	0.11	0.12	0.099	0.1	0.13	0.16	0.19	0.2	0.21	0.24	0.28
Bangladesh	0.21	0.24	0.25	0.26	0.27	0.28	0.3	0.3	0.33	0.36	0.39	0.41	0.43	0.44	0.46
Japan	9.6	9.5	9.6	9.7	9.9	9.7	9.6	9.8	9.4	8.6	9.1	9.3	9.6	9.8	9.5

In [61]:
```
# plt.figure(figsize = (15,9))
# sns.heatmap(globalwarming_df,cbar = False,xticklabels = False,yticklabels =
# plt.show()
```

In [62]:
```
# plt.figure(figsize=(14,14))
#
# cbar_kws = {"orientation":"horizontal",
#             "shrink":1,
#             'extend':'min',
#             'extendfrac':0.1,
#             "ticks":np.arange(0,22),
#             "drawedges":True,
#            }
#
# sns.heatmap(globalwarming_df, cbar_kws=cbar_kws)
# plt.show()
```

In [63]:
```
# plt.figure(figsize=(16,9))
# ax = sns.heatmap(globalwarming_df,)
# ax.set(title="Heatmap",
#       xlabel="Years",
#       ylabel="Country Name",)
# sns.set(font_scale=2) # set fontsize 2
```

Correlation

In [64]:

```
globalwarming_df.corr()
```

Out[64]:

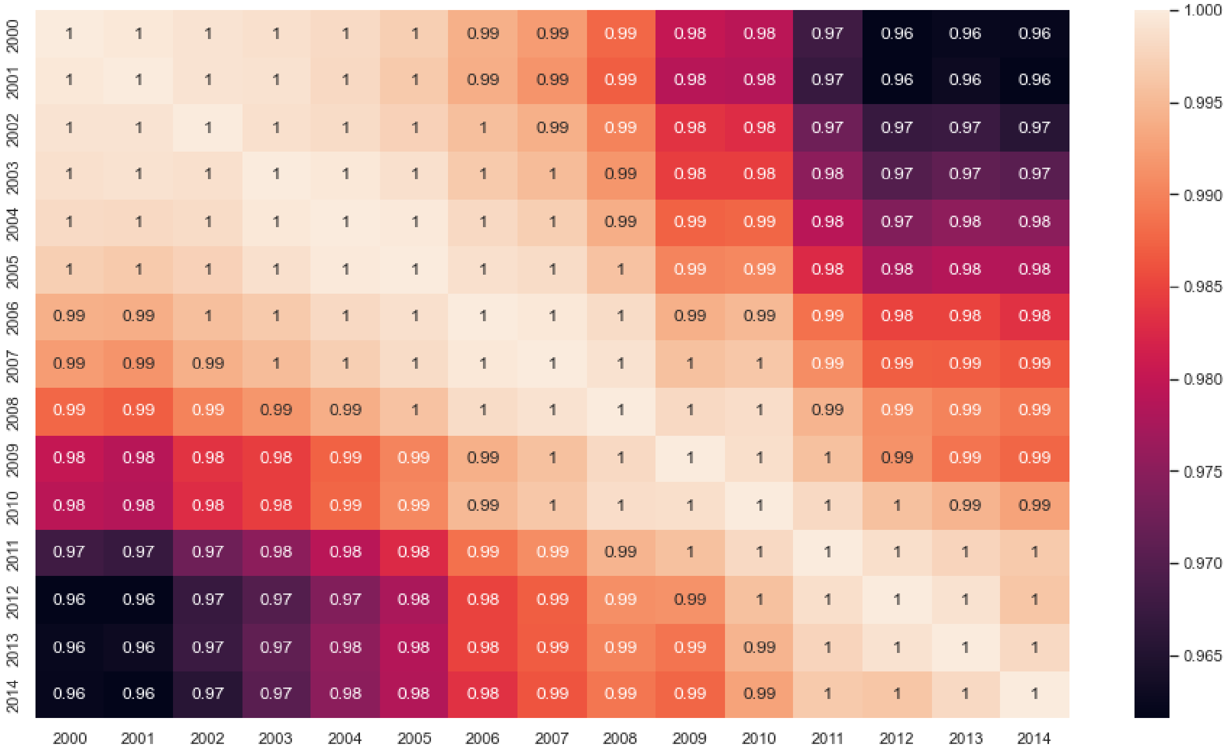
	2000	2001	2002	2003	2004	2005	2006	2007	2008
2000	1.000000	0.999632	0.999155	0.998911	0.998314	0.997008	0.994087	0.992283	0.987767
2001	0.999632	1.000000	0.999229	0.999026	0.998095	0.996628	0.993860	0.991532	0.987057
2002	0.999155	0.999229	1.000000	0.998907	0.998399	0.997391	0.995643	0.994017	0.990034
2003	0.998911	0.999026	0.998907	1.000000	0.999568	0.998887	0.996614	0.995277	0.991681
2004	0.998314	0.998095	0.998399	0.999568	1.000000	0.999701	0.998105	0.997144	0.993891
2005	0.997008	0.996628	0.997391	0.998887	0.999701	1.000000	0.998942	0.998420	0.995803
2006	0.994087	0.993860	0.995643	0.996614	0.998105	0.998942	1.000000	0.999570	0.998415
2007	0.992283	0.991532	0.994017	0.995277	0.997144	0.998420	0.999570	1.000000	0.999088
2008	0.987767	0.987057	0.990034	0.991681	0.993891	0.995803	0.998415	0.999088	1.000000
2009	0.980143	0.978912	0.983584	0.984511	0.987300	0.990125	0.994104	0.995724	0.998145
2010	0.979172	0.978562	0.982944	0.984466	0.987668	0.990498	0.994985	0.996367	0.998539
2011	0.967887	0.967206	0.972479	0.975128	0.979061	0.982646	0.988553	0.990928	0.994593
2012	0.961582	0.961625	0.967161	0.969919	0.974094	0.977758	0.984892	0.986978	0.991128
2013	0.962466	0.962827	0.967573	0.971053	0.975276	0.978611	0.984857	0.986819	0.989983
2014	0.962331	0.961622	0.965665	0.970508	0.975061	0.978521	0.983371	0.986199	0.988927

In [65]:

```
# plt.figure(figsize=(16,9))
#
# ax = sns.heatmap(globalwarming_df.corr(),annot = True,linewidths = 2)
# ax.tick_params(size = 5,color = 'white',labelsize = 5,labelcolor = 'white')
#
# plt.title('Heatmap of Who is Responsible for Global Warming',fontsize = 20)
#
# plt.show()
```

In [66]:

```
plt.figure(figsize=(16,9))

sns.heatmap(globalwarming_df.corr(), annot = True)
plt.show()
```



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
2000	1	1	1	1	1	1	0.99	0.99	0.99	0.98	0.98	0.97	0.96	0.96	0.96
2001	1	1	1	1	1	1	0.99	0.99	0.99	0.98	0.98	0.97	0.96	0.96	0.96
2002	1	1	1	1	1	1	1	0.99	0.99	0.98	0.98	0.97	0.97	0.97	0.97
2003	1	1	1	1	1	1	1	1	0.99	0.98	0.98	0.98	0.97	0.97	0.97
2004	1	1	1	1	1	1	1	1	0.99	0.99	0.99	0.98	0.97	0.98	0.98
2005	1	1	1	1	1	1	1	1	1	0.99	0.99	0.98	0.98	0.98	0.98
2006	0.99	0.99	1	1	1	1	1	1	1	0.99	0.99	0.99	0.98	0.98	0.98
2007	0.99	0.99	0.99	1	1	1	1	1	1	1	1	0.99	0.99	0.99	0.99
2008	0.99	0.99	0.99	0.99	0.99	1	1	1	1	1	1	0.99	0.99	0.99	0.99
2009	0.98	0.98	0.98	0.98	0.99	0.99	0.99	1	1	1	1	1	0.99	0.99	0.99
2010	0.98	0.98	0.98	0.98	0.99	0.99	0.99	1	1	1	1	1	1	0.99	0.99
2011	0.97	0.97	0.97	0.98	0.98	0.98	0.99	0.99	0.99	1	1	1	1	1	1
2012	0.96	0.96	0.97	0.97	0.97	0.98	0.98	0.99	0.99	0.99	1	1	1	1	1
2013	0.96	0.96	0.97	0.97	0.98	0.98	0.98	0.99	0.99	0.99	0.99	1	1	1	1
2014	0.96	0.96	0.97	0.97	0.98	0.98	0.98	0.99	0.99	0.99	0.99	1	1	1	1

In [67]:
```
breast_cancer = pd.read_csv('breast_cancer.csv')
breast_cancer.drop('Unnamed: 32',axis = 1,inplace = True)
```

In [68]:
```
breast_cancer.corr()
```

Out[68]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	sm
id	1.000000	0.074626	0.099770	0.073159	0.096893	
radius_mean	0.074626	1.000000	0.323782	0.997855	0.987357	
texture_mean	0.099770	0.323782	1.000000	0.329533	0.321086	
perimeter_mean	0.073159	0.997855	0.329533	1.000000	0.986507	
area_mean	0.096893	0.987357	0.321086	0.986507	1.000000	
smoothness_mean	-0.012968	0.170581	-0.023389	0.207278	0.177028	
compactness_mean	0.000096	0.506124	0.236702	0.556936	0.498502	
concavity_mean	0.050080	0.676764	0.302418	0.716136	0.685983	
concave points_mean	0.044158	0.822529	0.293464	0.850977	0.823269	
symmetry_mean	-0.022114	0.147741	0.071401	0.183027	0.151293	
fractal_dimension_mean	-0.052511	-0.311631	-0.076437	-0.261477	-0.283110	
radius_se	0.143048	0.679090	0.275869	0.691765	0.732562	
texture_se	-0.007526	-0.097317	0.386358	-0.086761	-0.066280	
perimeter_se	0.137331	0.674172	0.281673	0.693135	0.726628	
area_se	0.177742	0.735864	0.259845	0.744983	0.800086	
smoothness_se	0.096781	-0.222600	0.006614	-0.202694	-0.166777	
compactness_se	0.033961	0.206000	0.191975	0.250744	0.212583	
concavity_se	0.055239	0.194204	0.143293	0.228082	0.207660	
concave points_se	0.078768	0.376169	0.163851	0.407217	0.372320	
symmetry_se	-0.017306	-0.104321	0.009127	-0.081629	-0.072497	
fractal_dimension_se	0.025725	-0.042641	0.054458	-0.005523	-0.019887	
radius_worst	0.082405	0.969539	0.352573	0.969476	0.962746	
texture_worst	0.064720	0.297008	0.912045	0.303038	0.287489	
perimeter_worst	0.079986	0.965137	0.358040	0.970387	0.959120	
area_worst	0.107187	0.941082	0.343546	0.941550	0.959213	
smoothness_worst	0.010338	0.119616	0.077503	0.150549	0.123523	
compactness_worst	-0.002968	0.413463	0.277830	0.455774	0.390410	
concavity_worst	0.023203	0.526911	0.301025	0.563879	0.512606	
concave points_worst	0.035174	0.744214	0.295316	0.771241	0.722017	
symmetry_worst	-0.044224	0.163953	0.105008	0.189115	0.143570	
fractal_dimension_worst	-0.029866	0.007066	0.119205	0.051019	0.003738	

31 rows × 31 columns

In [89]:

```
# plt.figure(figsize = (30,30))
# sns.heatmap(breast_cancer.corr(),annot = True,linewidths = 2)
# plt.show()
```

Pairplot

```
In [77]: # sns.pairplot(breast_cancer)
         # plt.show()
```

```
In [80]: # sns.pairplot(breast_cancer.corr())
         # plt.show()
```

```
In [82]: breast_cancer.describe()
```

Out[82]:

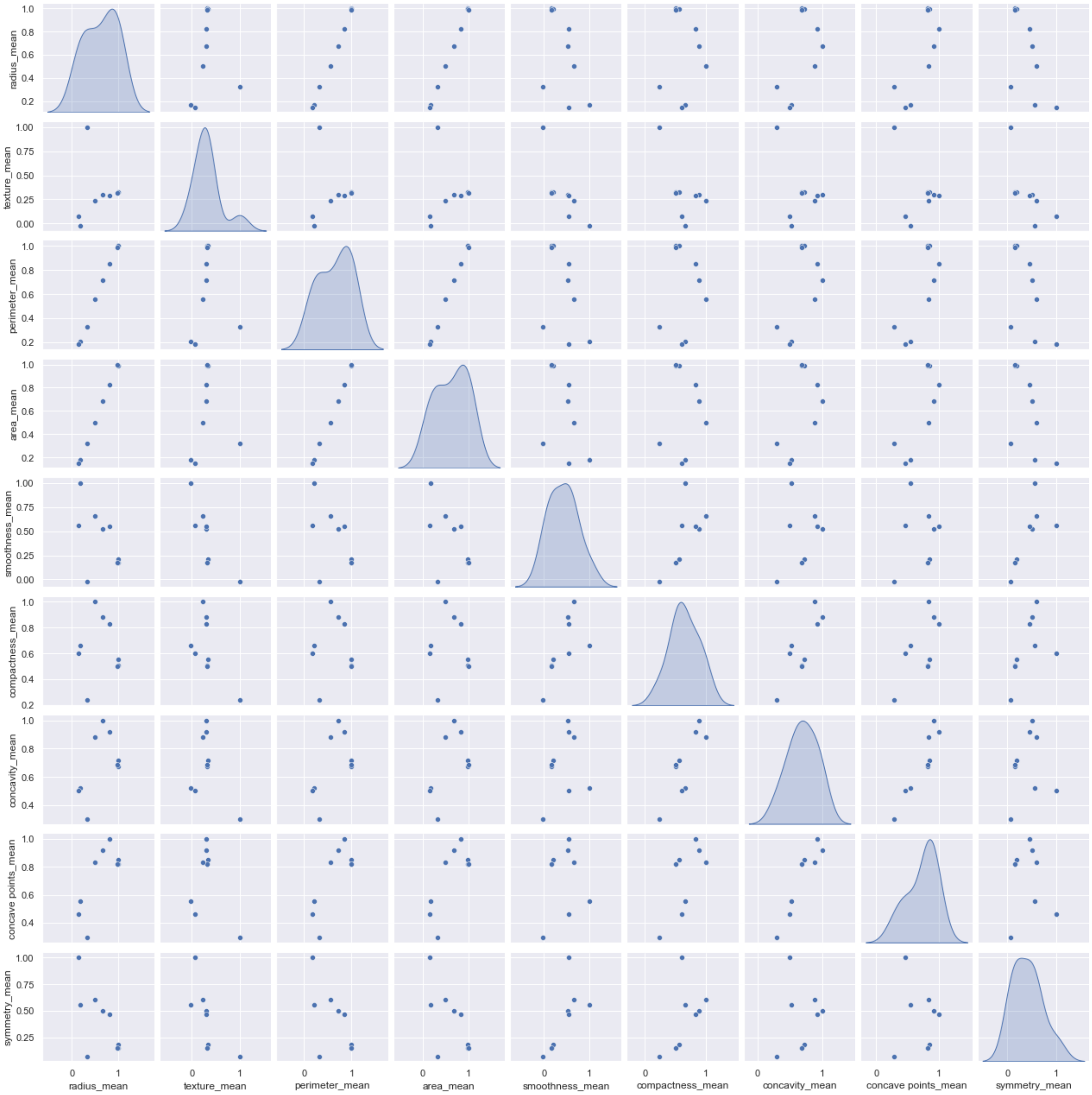
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mea
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.0000
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.0963
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.0140
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.0526
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.0863
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.0958
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.1053
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.1634

8 rows × 31 columns

```
In [83]: featureMeans = list(breast_cancer.columns[1:11])
```

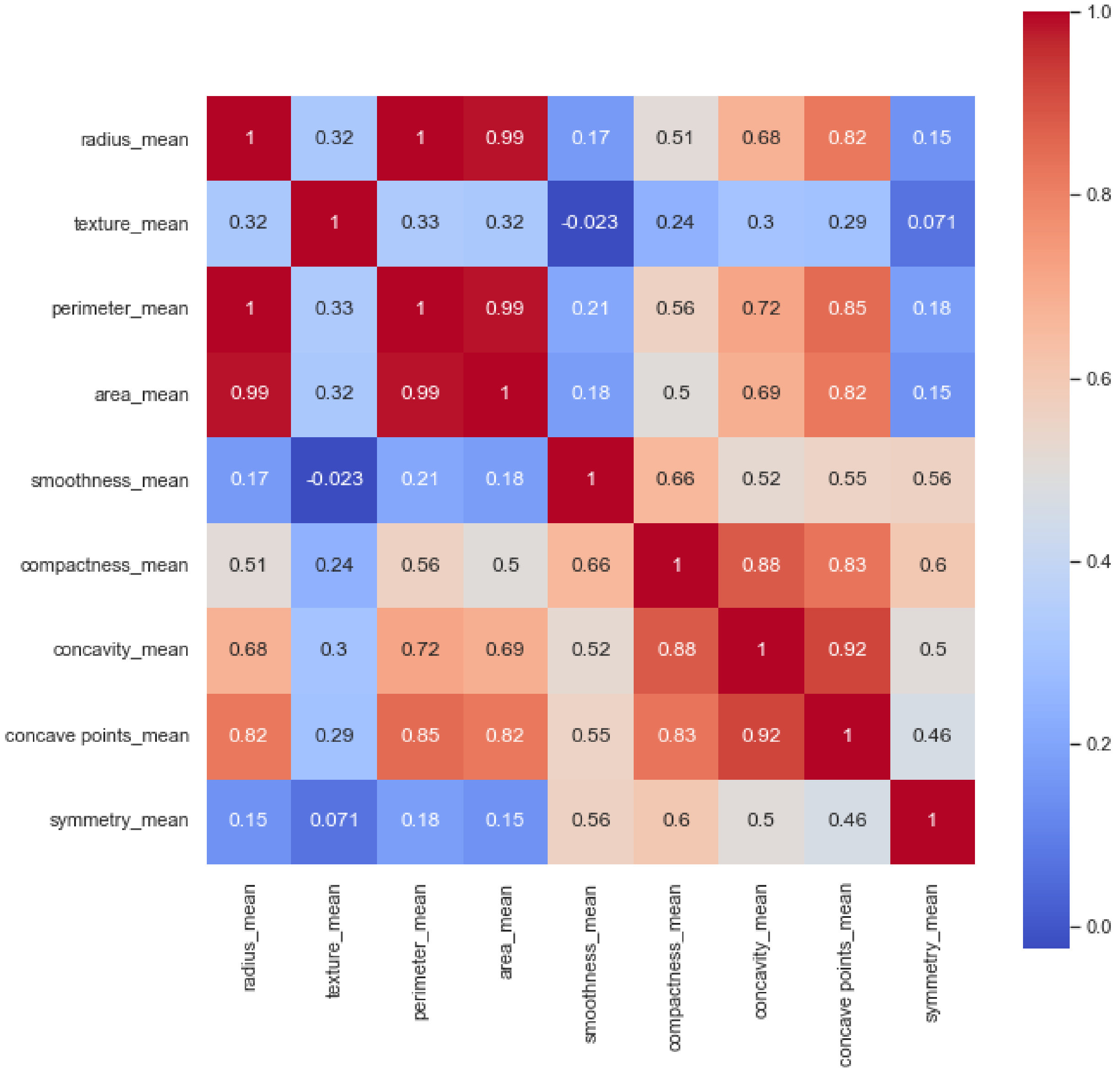
```
In [84]: correlationData = breast_cancer[featureMeans].corr()
         sns.pairplot(breast_cancer[featureMeans].corr(),diag_kind = 'kde',size = 2)
         plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\axisgrid.py:2076: UserWarning: The `size` parameter has been re
named to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



In [92]:

```
plt.figure(figsize = (10,10))
sns.heatmap(breast_cancer[featureMeans].corr(),annot = True,square = True,cma
plt.show()
```



	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean
radius_mean	1	0.32	1	0.99	0.17	0.51	0.68	0.82	0.15
texture_mean	0.32	1	0.33	0.32	-0.023	0.24	0.3	0.29	0.071
perimeter_mean	1	0.33	1	0.99	0.21	0.56	0.72	0.85	0.18
area_mean	0.99	0.32	0.99	1	0.18	0.5	0.69	0.82	0.15
smoothness_mean	0.17	-0.023	0.21	0.18	1	0.66	0.52	0.55	0.56
compactness_mean	0.51	0.24	0.56	0.5	0.66	1	0.88	0.83	0.6
concavity_mean	0.68	0.3	0.72	0.69	0.52	0.88	1	0.92	0.5
concave points_mean	0.82	0.29	0.85	0.82	0.55	0.83	0.92	1	0.46
symmetry_mean	0.15	0.071	0.18	0.15	0.56	0.6	0.5	0.46	1

CampusX

```
In [2]: import seaborn as sns
        import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd
```

```
In [3]: plt.style.use('fivethirtyeight')
```

```
In [4]: tips_df = pd.read_csv('tips.csv')
        tips_df.head()
```

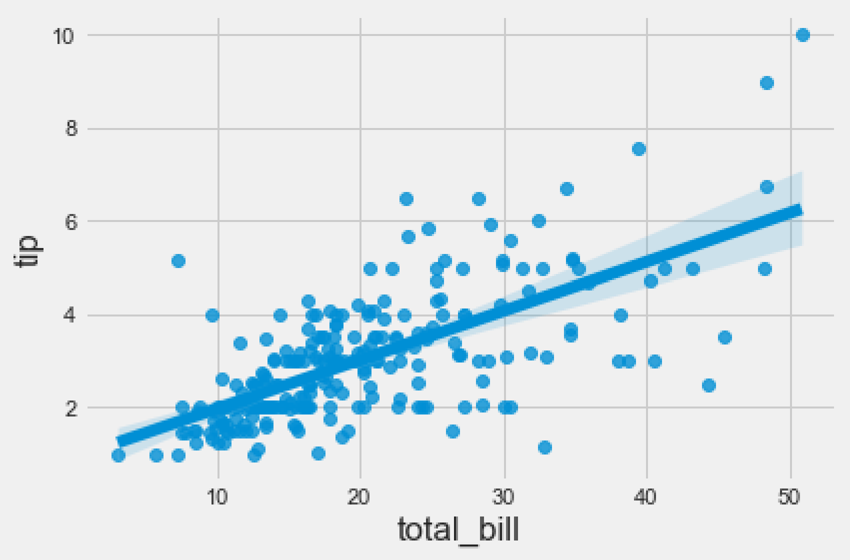
Out[4]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Scatter Plot

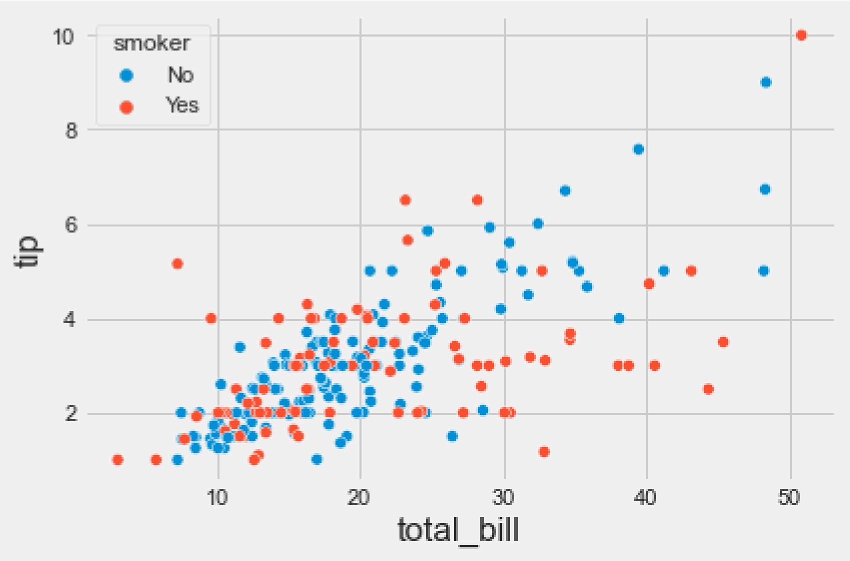
In [101…

```
sns.regplot(x = 'total_bill',y = 'tip',data = tips_df)
plt.show()
```



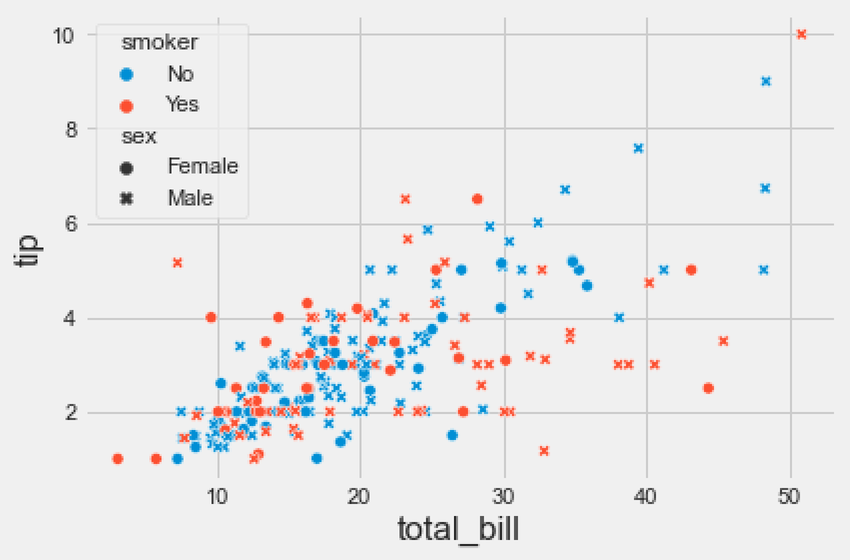
In [103…

```
sns.scatterplot(x = 'total_bill',y = 'tip',hue = 'smoker',data = tips_df)
plt.show()
```



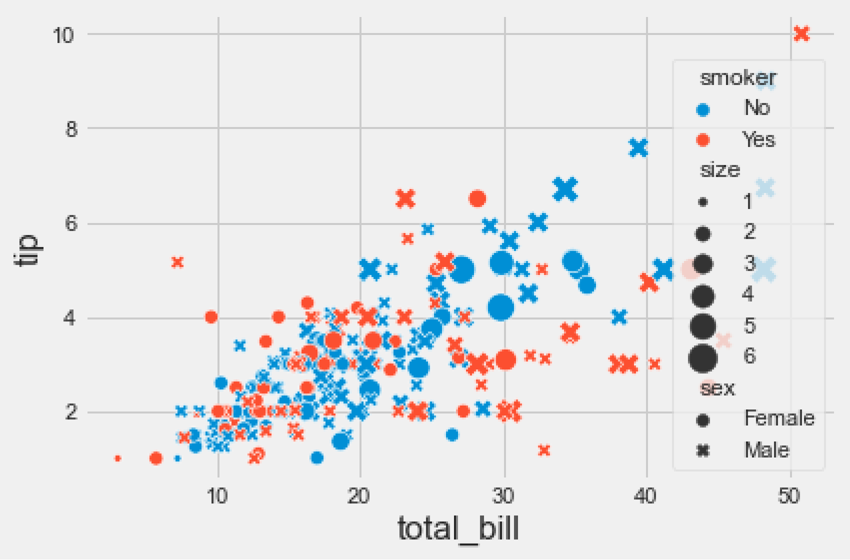
In [107…

```
sns.scatterplot(x = 'total_bill',y = 'tip',hue = 'smoker',style = 'sex',data
plt.show()
```



In [109…

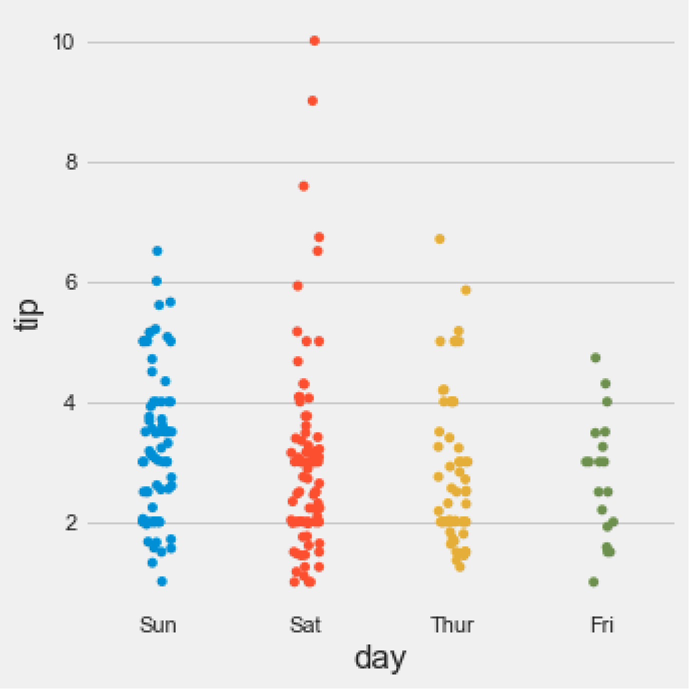
```
sns.scatterplot(x = 'total_bill',y = 'tip',hue = 'smoker',style = 'sex',size
plt.show()
```



Strip & Swarm Plot

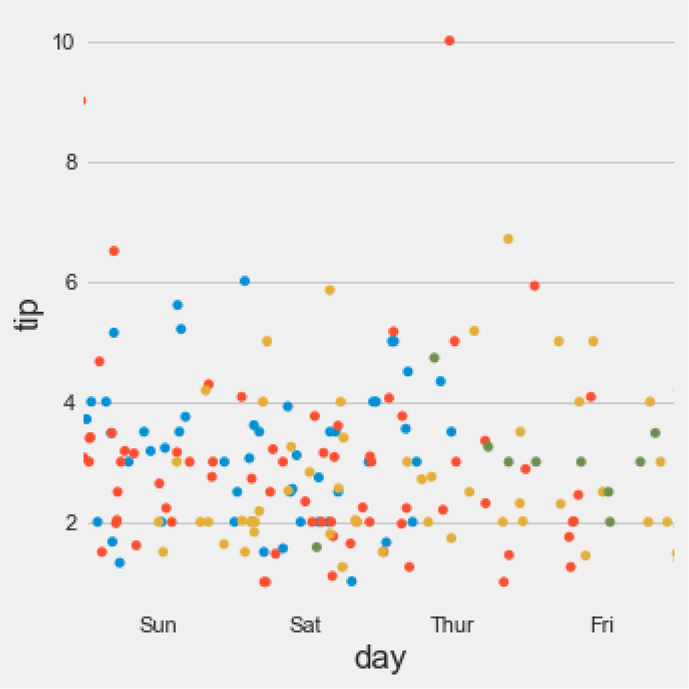
In [110…

```
sns.catplot(x = 'day',y = 'tip',kind = 'strip',data = tips_df)
plt.show()
```



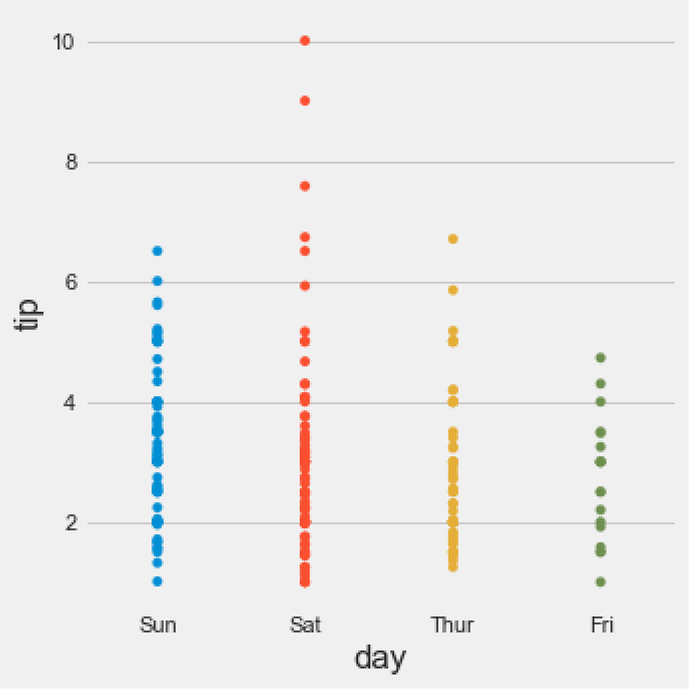
In [115…

```
sns.catplot(x = 'day',y = 'tip',kind = 'strip',jitter = 2,data = tips_df)
plt.show()
```



In [116…

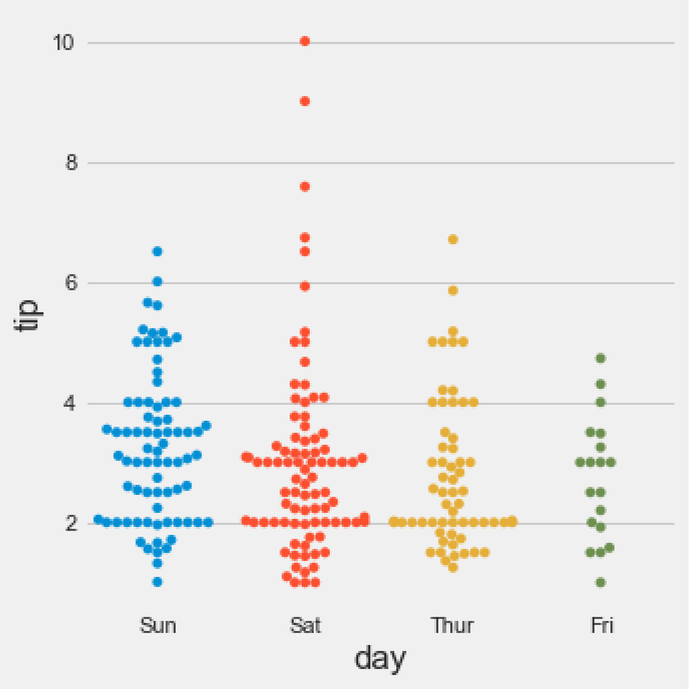
```
sns.catplot(x = 'day',y = 'tip',kind = 'strip',jitter = 0,data = tips_df)
plt.show()
```



In [117…

```
sns.catplot(x = 'day',y = 'tip',kind = 'swarm',data = tips_df)
plt.show()
```

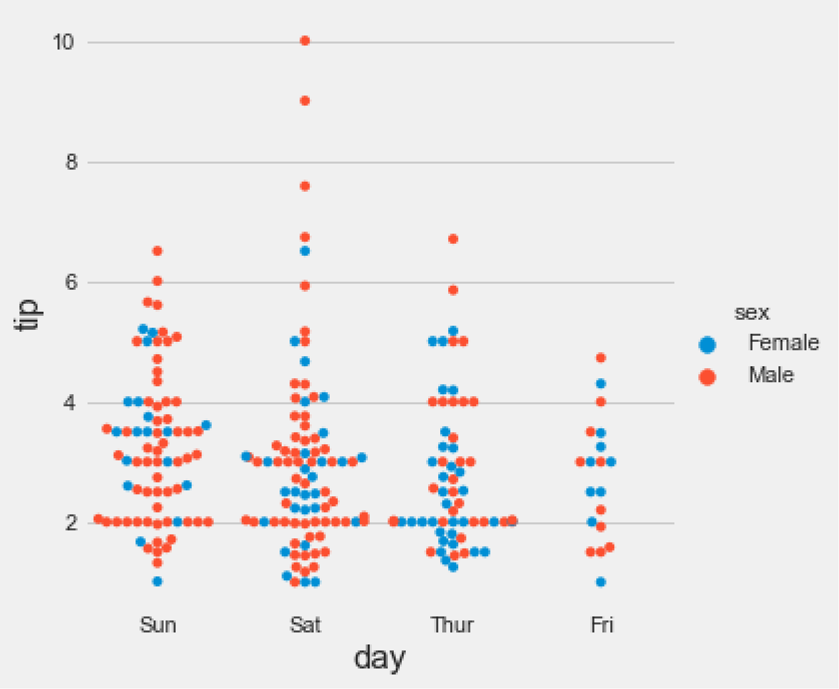
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\categorical.py:1296: UserWarning: 8.1% of the points cannot be
placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



In [118…

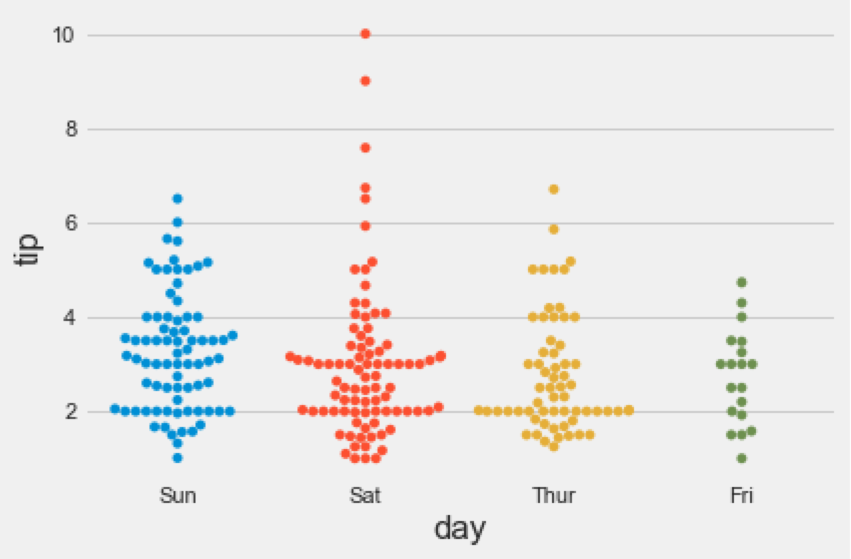
```
sns.catplot(x = 'day',y = 'tip',kind = 'swarm',hue = 'sex',data = tips_df)
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\categorical.py:1296: UserWarning: 8.1% of the points cannot be
placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



In [119…

```
# sns.swarmplot(x = 'day',y = 'tip',data = tips_df)
# plt.show()
```

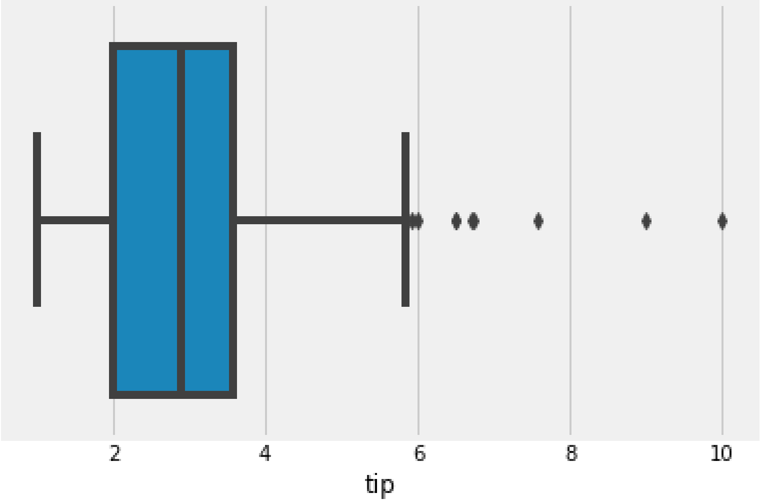


Box Plot

In [5]:

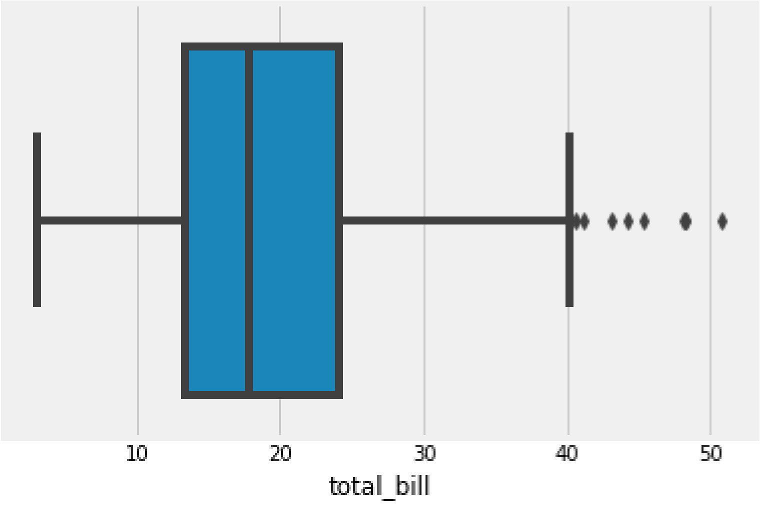
```
sns.boxplot(tips_df['tip'])
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable a
s a keyword arg: x. From version 0.12, the only valid positional argument wi
ll be `data`, and passing other arguments without an explicit keyword will r
esult in an error or misinterpretation.
  warnings.warn(
```

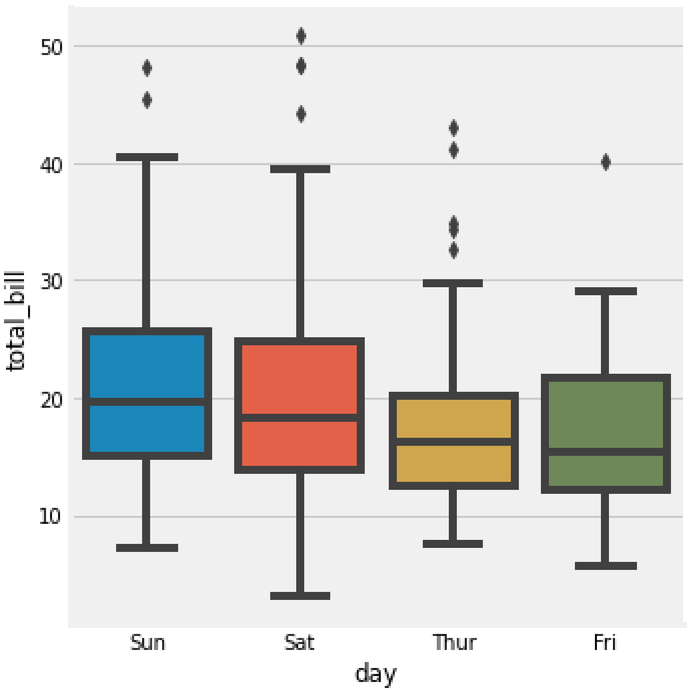


In [6]:
```
sns.boxplot(tips_df['total_bill'])
plt.show()
```

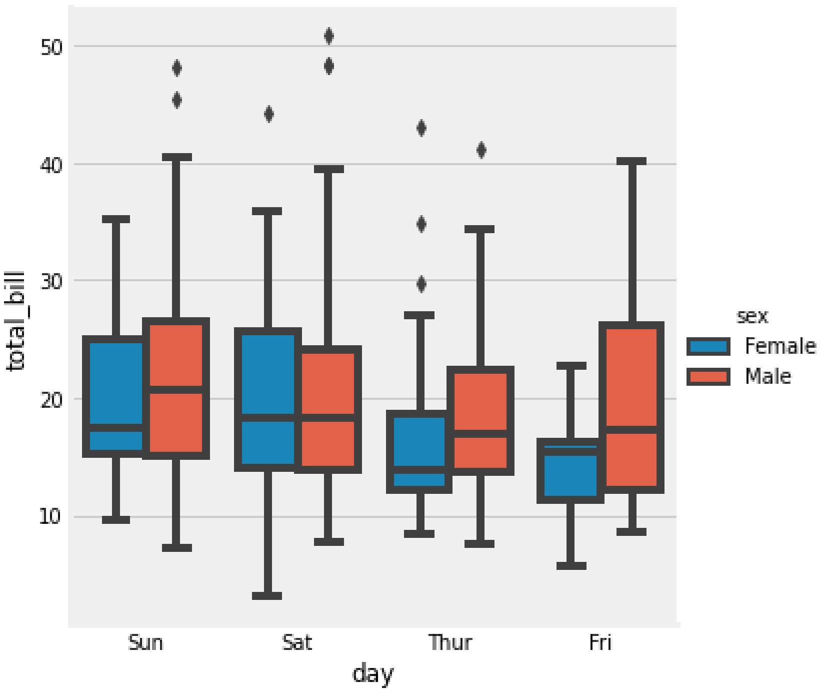
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable a
s a keyword arg: x. From version 0.12, the only valid positional argument wi
ll be `data`, and passing other arguments without an explicit keyword will r
esult in an error or misinterpretation.
  warnings.warn(
```



In [7]:
```
sns.catplot(x = 'day',y = 'total_bill',kind = 'box',data = tips_df)
plt.show()
```



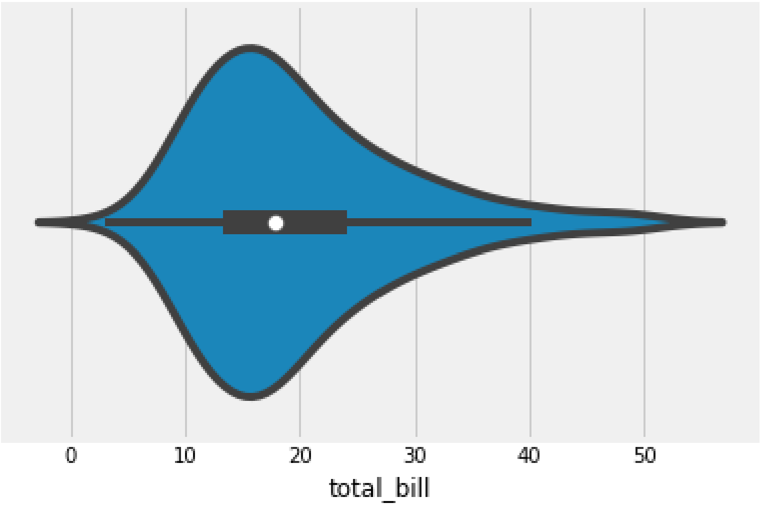
In [8]:
```
sns.catplot(x = 'day',y = 'total_bill',hue = 'sex',kind = 'box',data = tips_
plt.show()
```



Violin Plot

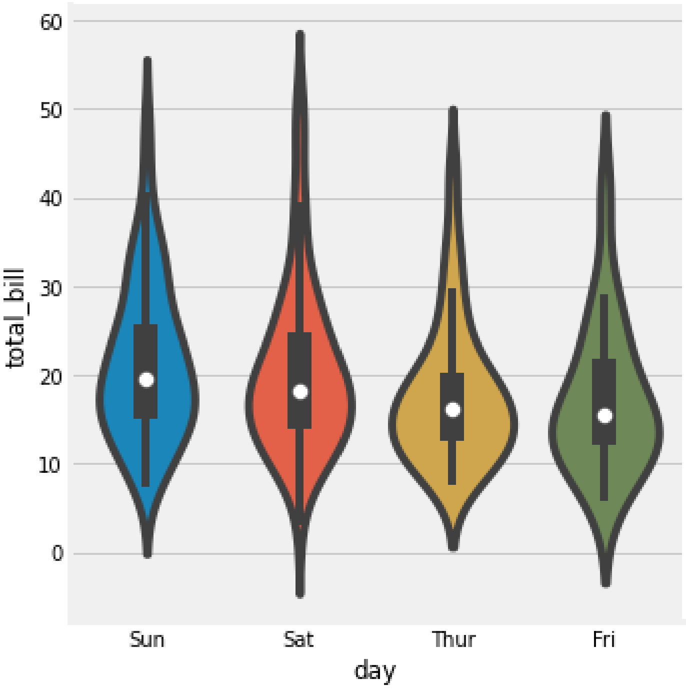
In [9]:
```
sns.violinplot(tips_df['total_bill'])
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable a
s a keyword arg: x. From version 0.12, the only valid positional argument wi
ll be `data`, and passing other arguments without an explicit keyword will r
esult in an error or misinterpretation.
  warnings.warn(
```



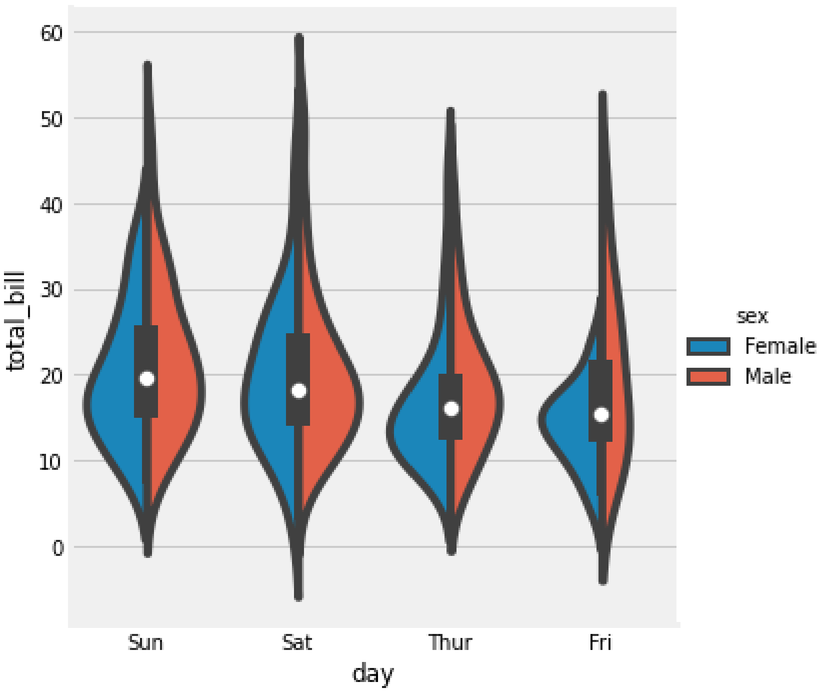
In [10]:

```
sns.catplot(x = 'day',y = 'total_bill',kind = 'violin',data = tips_df)
plt.show()
```



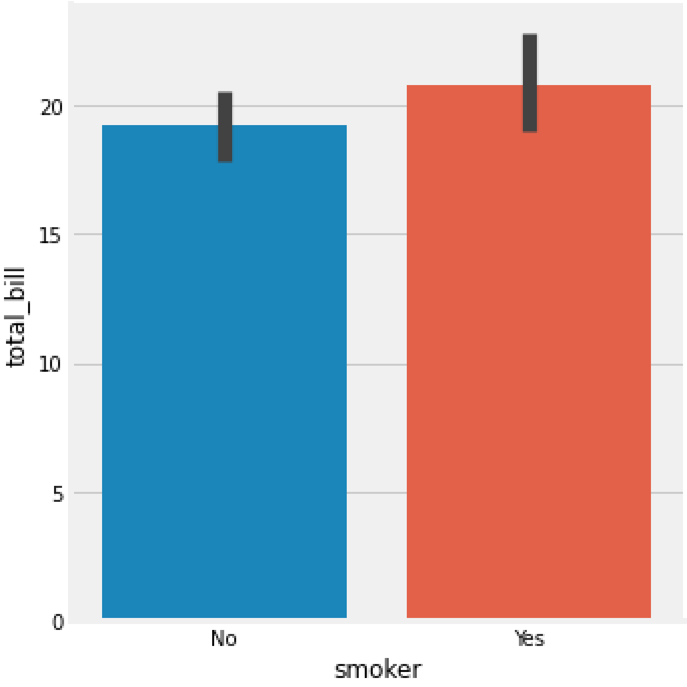
In [12]:

```
sns.catplot(x = 'day',y = 'total_bill',kind = 'violin',hue = 'sex',split = T
plt.show()
```

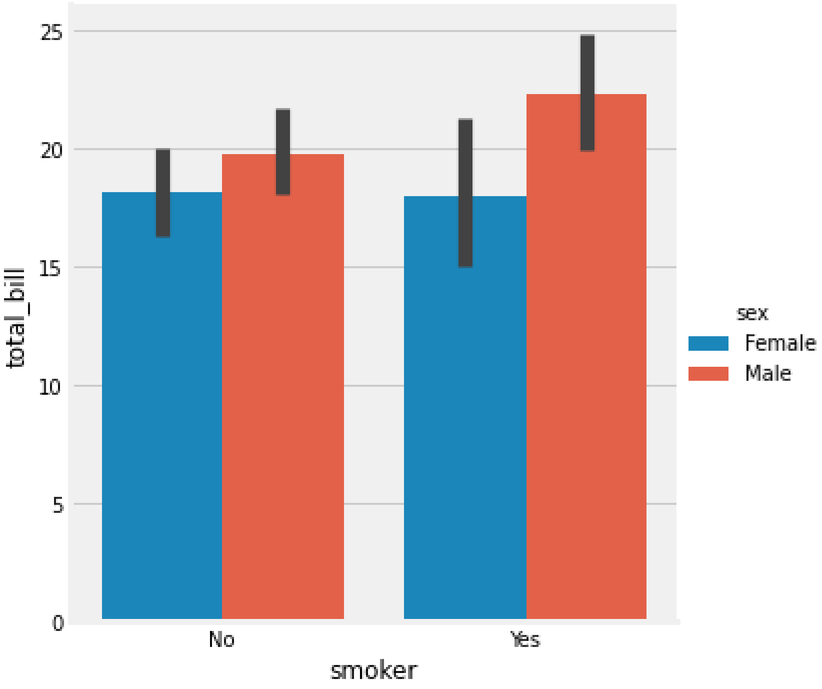


Bar & Count Plot

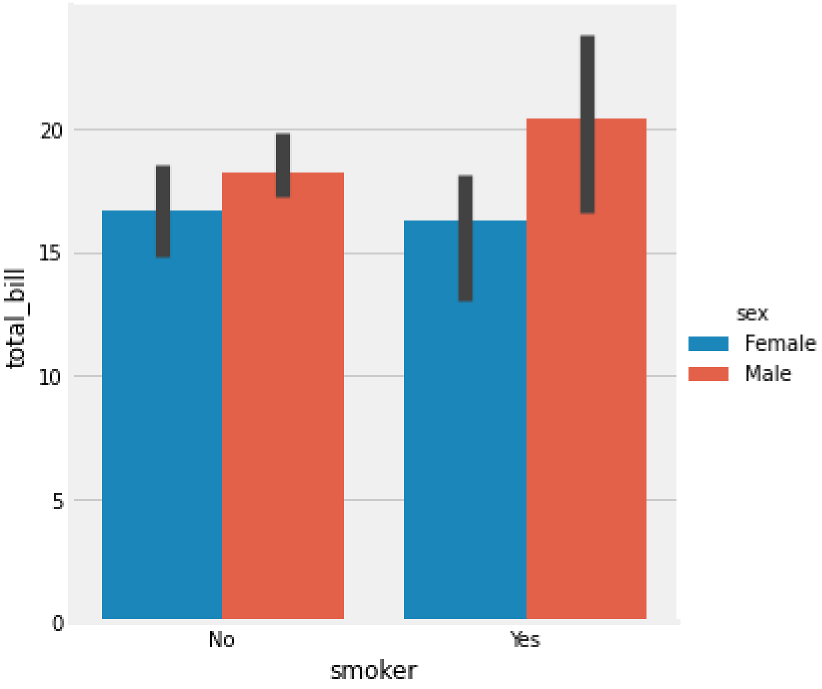
In [15]:
```
sns.catplot(x = 'smoker',y = 'total_bill',kind = 'bar',data = tips_df)
plt.show()
```



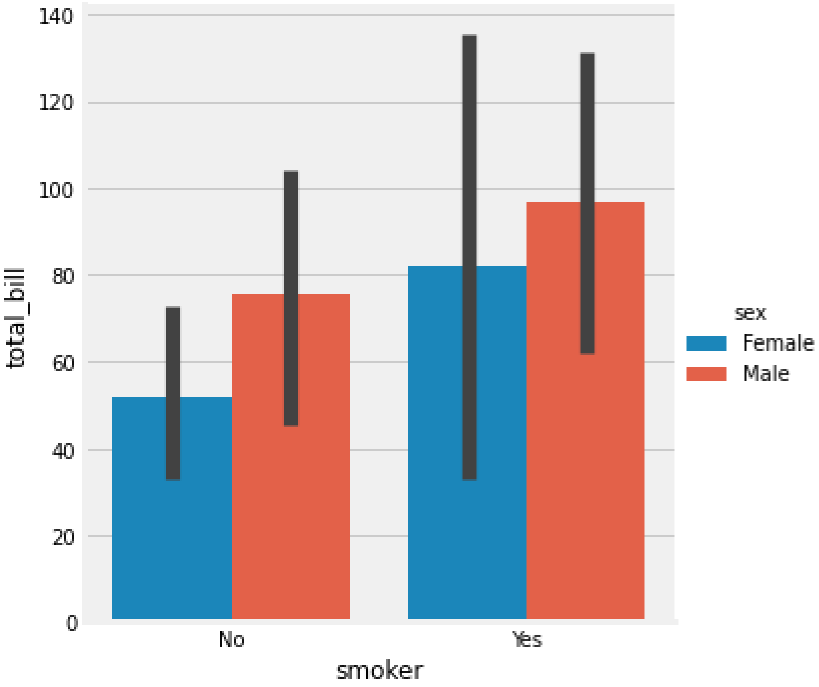
In [16]:
```
sns.catplot(x = 'smoker',y = 'total_bill',hue = 'sex',kind = 'bar',data = ti
plt.show()
```



```
In [17]: sns.catplot(x = 'smoker',y = 'total_bill',hue = 'sex',estimator = np.median,k
         plt.show()
```

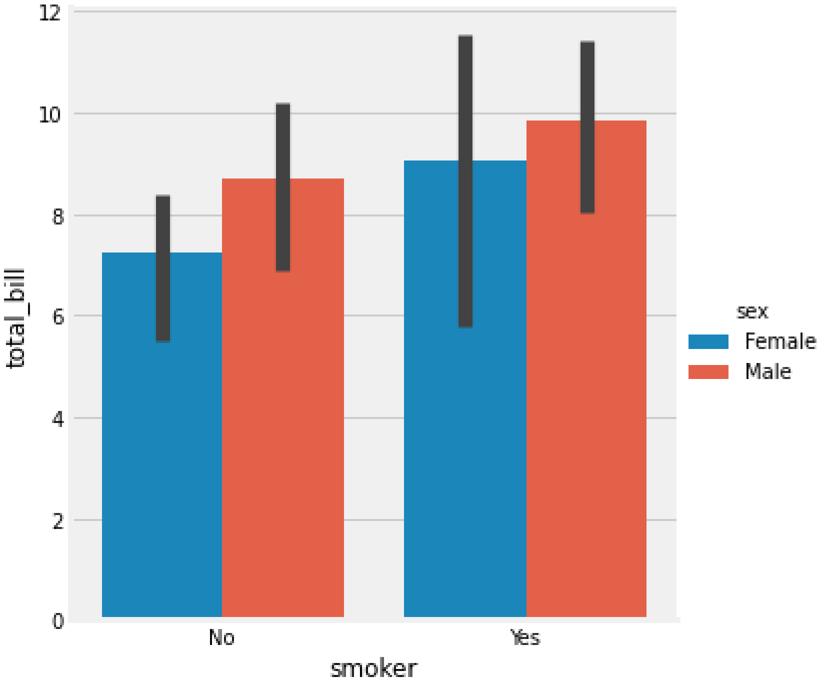


```
In [18]: sns.catplot(x = 'smoker',y = 'total_bill',hue = 'sex',estimator = np.var,kind
         plt.show()
```



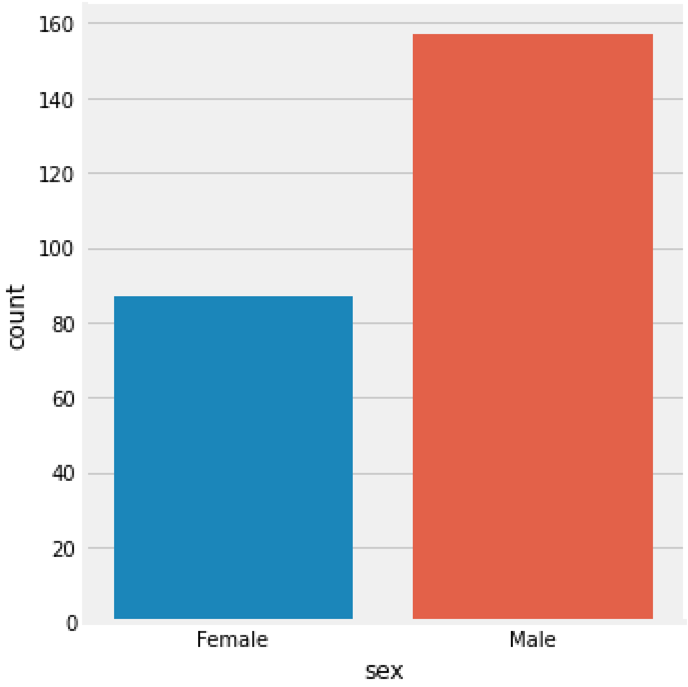
In [19]:

```
sns.catplot(x = 'smoker',y = 'total_bill',hue = 'sex',estimator = np.std,kind
plt.show()
```

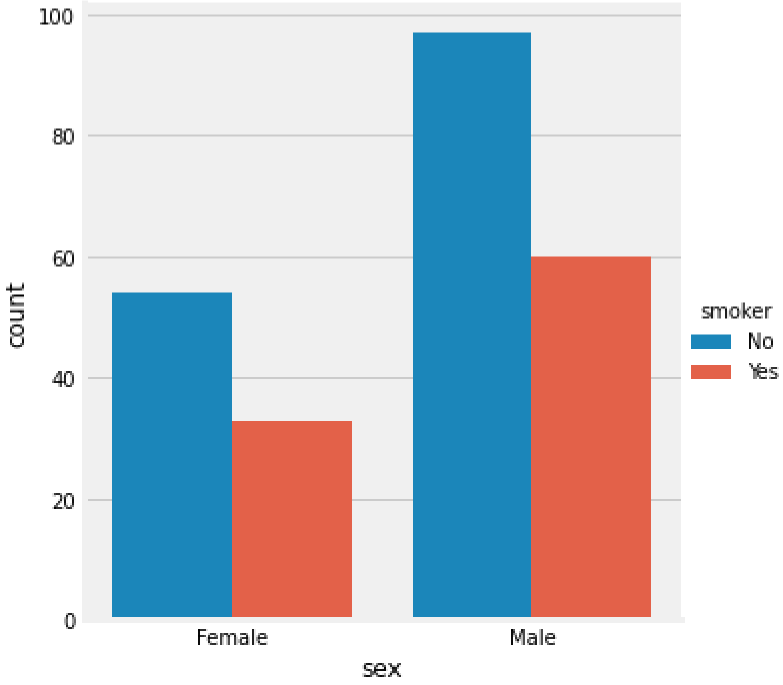


In [22]:

```
sns.catplot(x = 'sex',kind = 'count',data = tips_df)
plt.show()
```



In [23]:
```
sns.catplot(x = 'sex',hue = 'smoker',kind = 'count',data = tips_df)
plt.show()
```



Heatmap

In [33]:
```
plt.style.use('fivethirtyeight')
```

In [34]:
```
flights_df = pd.read_csv('flights.csv')
flights_df.head()
```

Out[34]:

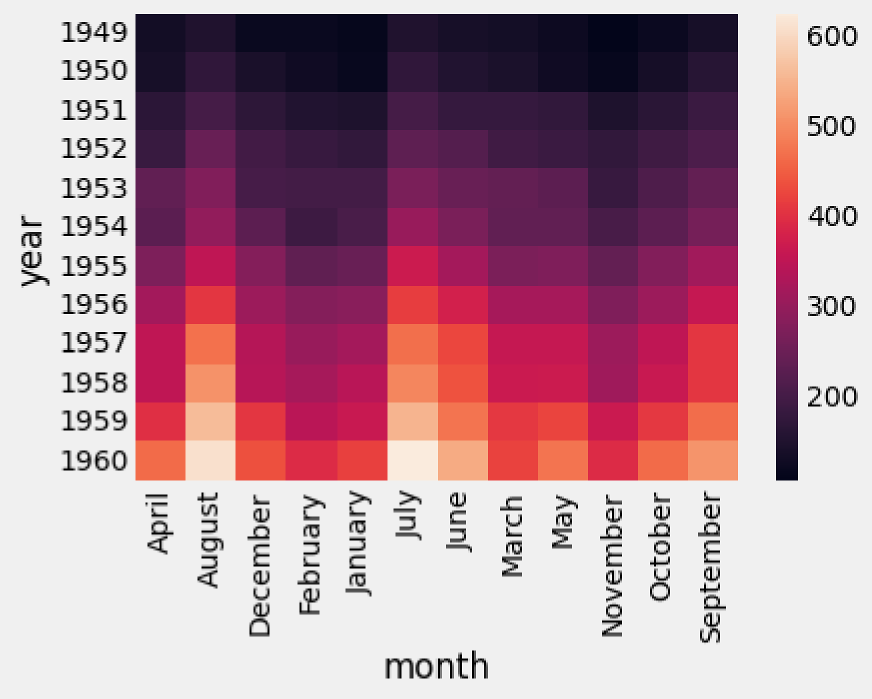
	year	month	passengers
0	1949	January	112
1	1949	February	118
2	1949	March	132
3	1949	April	129
4	1949	May	121

In [35]:

```
x = flights_df.pivot_table(index = 'year',columns = 'month',values = 'passeng
```

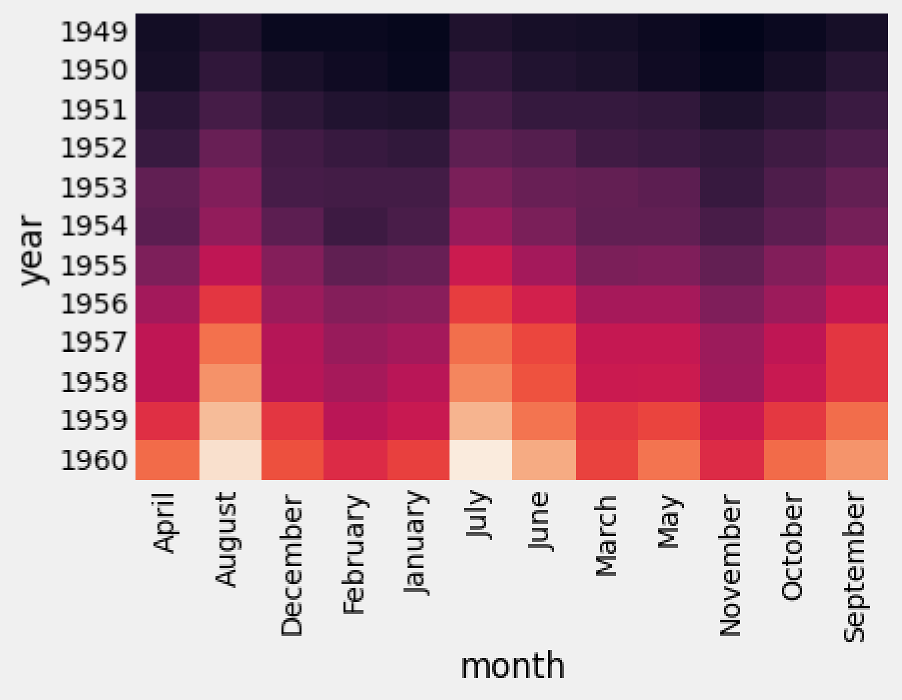
In [36]:

```
# plt.figure(figsize = (10,10))
sns.heatmap(x)
plt.show()
```



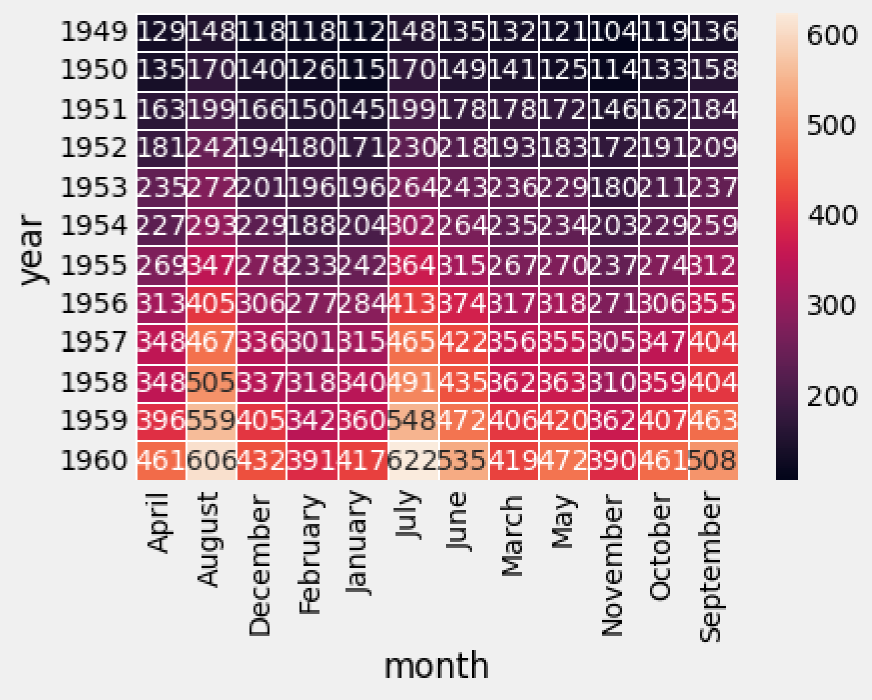
In [37]:

```
sns.heatmap(x,cbar = False)
plt.show()
```



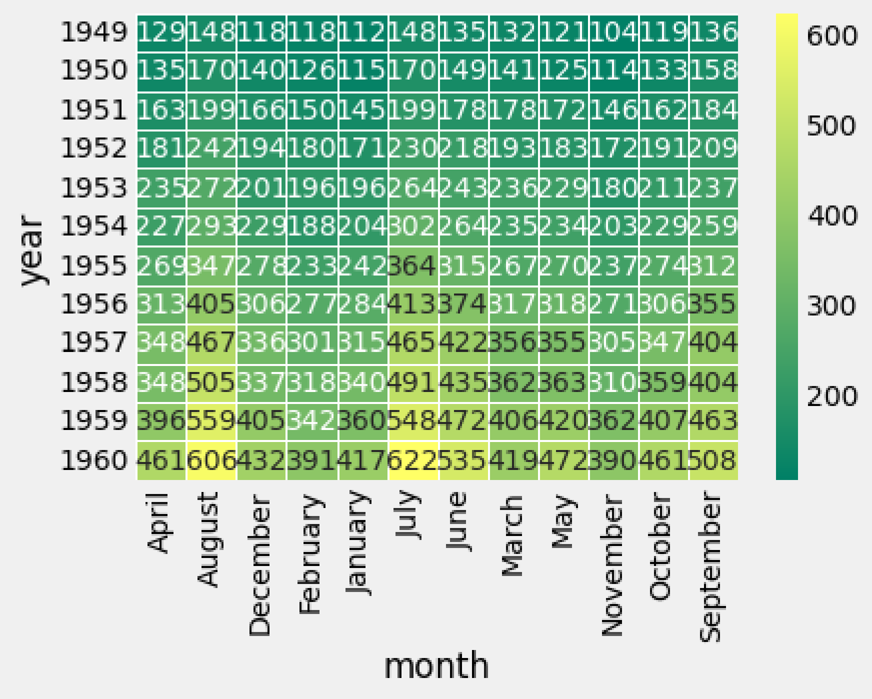
In [40]:

```
sns.heatmap(x,linewidths = 0.5,annot = True,fmt = 'd')
plt.show()
```



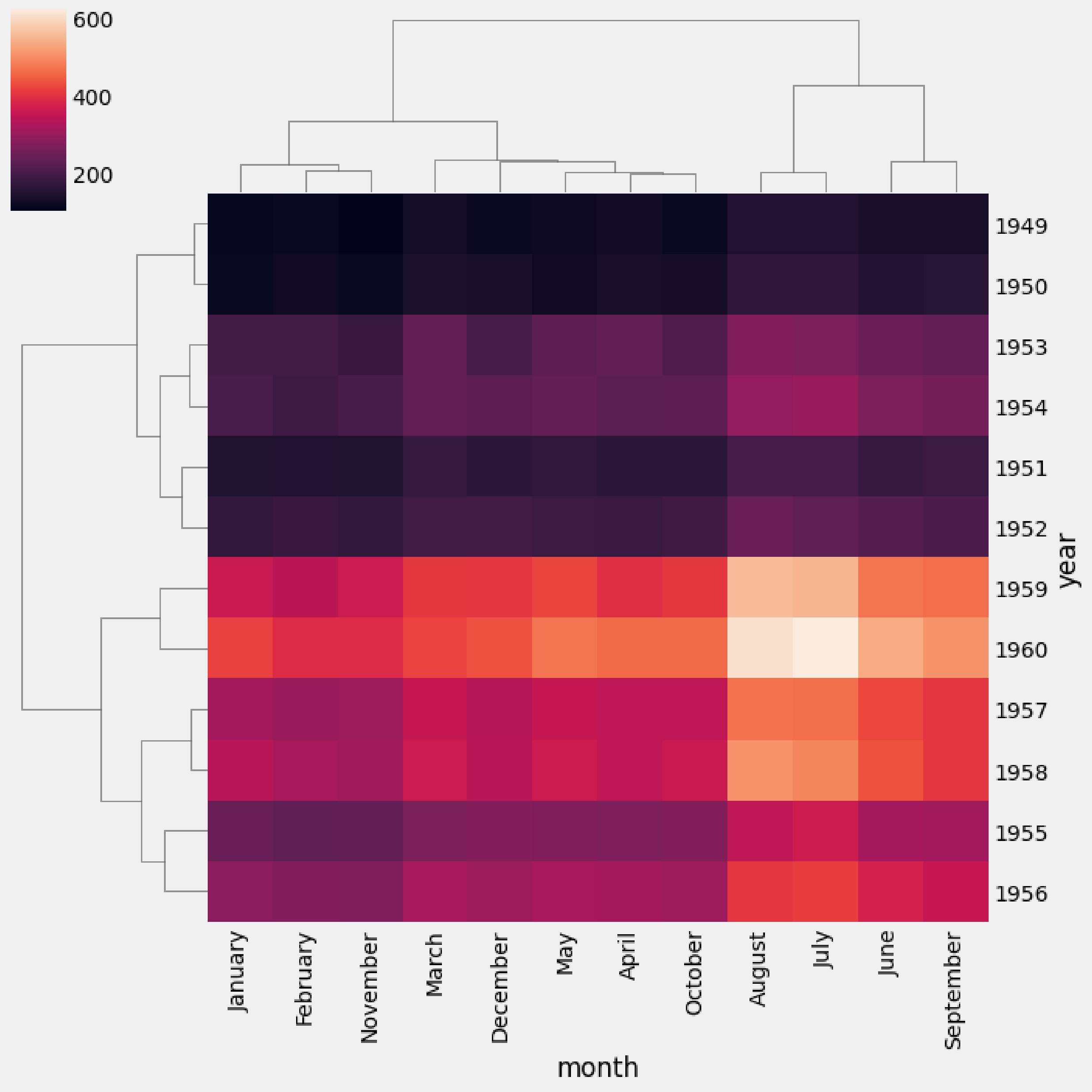
In [44]:

```
sns.heatmap(x,linewidths = 0.5,annot = True,fmt = 'd',cmap = 'summer')
plt.show()
```



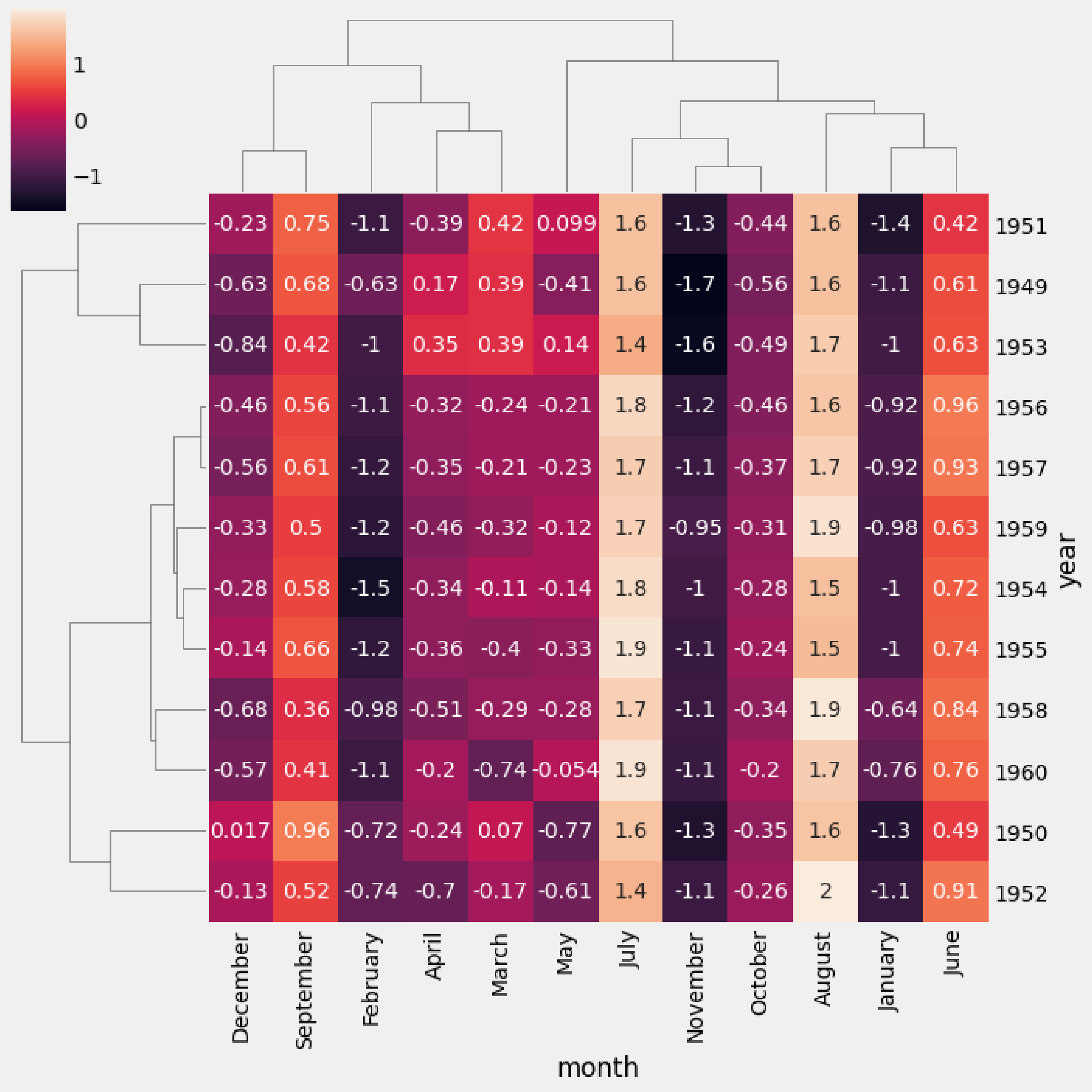
Clustermap

In [45]:
```
sns.clustermap(x)
plt.show()
```



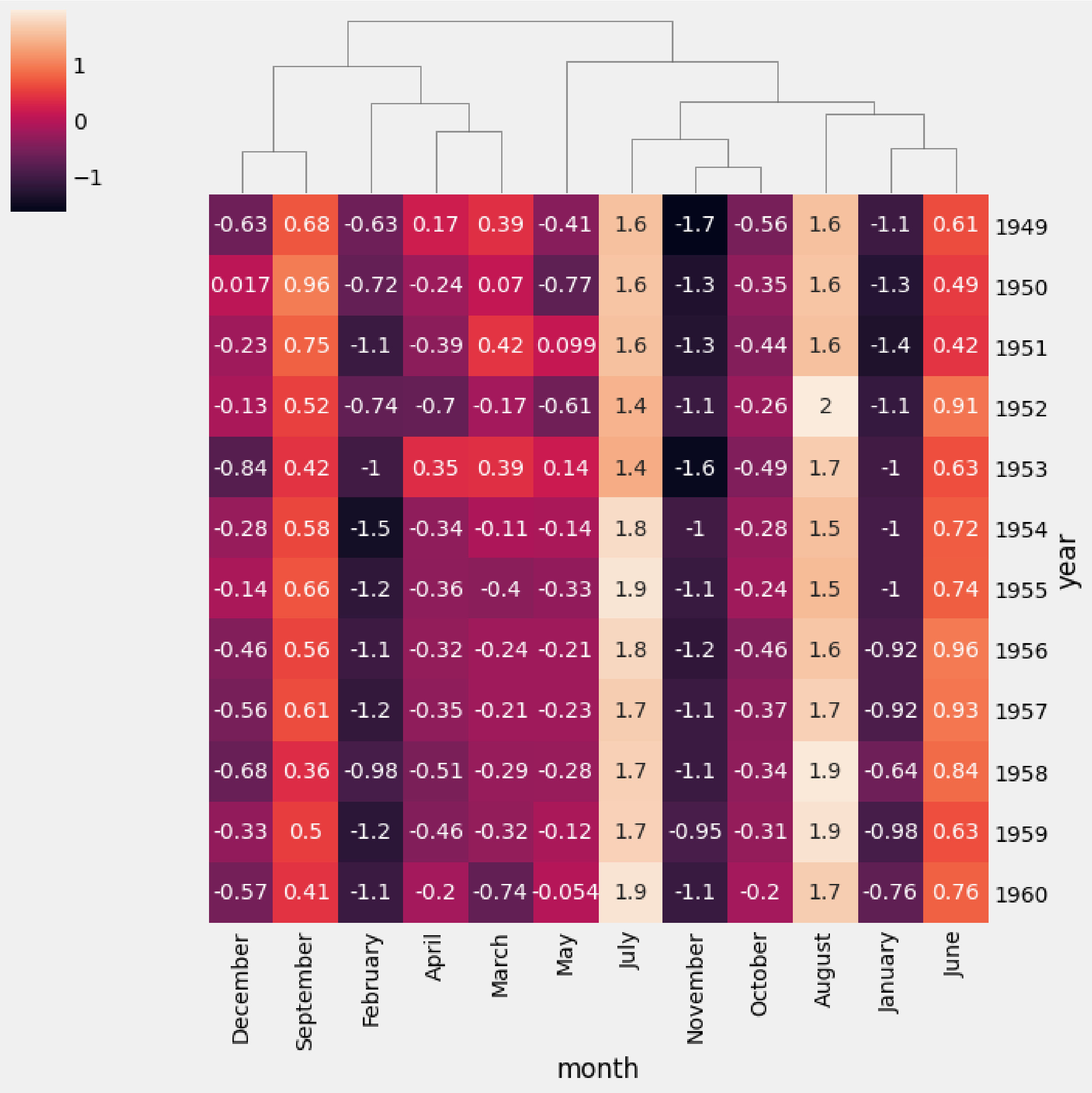
In [48]:

```
sns.clustermap(x,z_score = 0,annot = True,metric = 'correlation')
plt.show()
```



year	December	September	February	April	March	May	July	November	October	August	January	June
1951	-0.23	0.75	-1.1	-0.39	0.42	0.099	1.6	-1.3	-0.44	1.6	-1.4	0.42
1949	-0.63	0.68	-0.63	0.17	0.39	-0.41	1.6	-1.7	-0.56	1.6	-1.1	0.61
1953	-0.84	0.42	-1	0.35	0.39	0.14	1.4	-1.6	-0.49	1.7	-1	0.63
1956	-0.46	0.56	-1.1	-0.32	-0.24	-0.21	1.8	-1.2	-0.46	1.6	-0.92	0.96
1957	-0.56	0.61	-1.2	-0.35	-0.21	-0.23	1.7	-1.1	-0.37	1.7	-0.92	0.93
1959	-0.33	0.5	-1.2	-0.46	-0.32	-0.12	1.7	-0.95	-0.31	1.9	-0.98	0.63
1954	-0.28	0.58	-1.5	-0.34	-0.11	-0.14	1.8	-1	-0.28	1.5	-1	0.72
1955	-0.14	0.66	-1.2	-0.36	-0.4	-0.33	1.9	-1.1	-0.24	1.5	-1	0.74
1958	-0.68	0.36	-0.98	-0.51	-0.29	-0.28	1.7	-1.1	-0.34	1.9	-0.64	0.84
1960	-0.57	0.41	-1.1	-0.2	-0.74	-0.054	1.9	-1.1	-0.2	1.7	-0.76	0.76
1950	0.017	0.96	-0.72	-0.24	0.07	-0.77	1.6	-1.3	-0.35	1.6	-1.3	0.49
1952	-0.13	0.52	-0.74	-0.7	-0.17	-0.61	1.4	-1.1	-0.26	2	-1.1	0.91

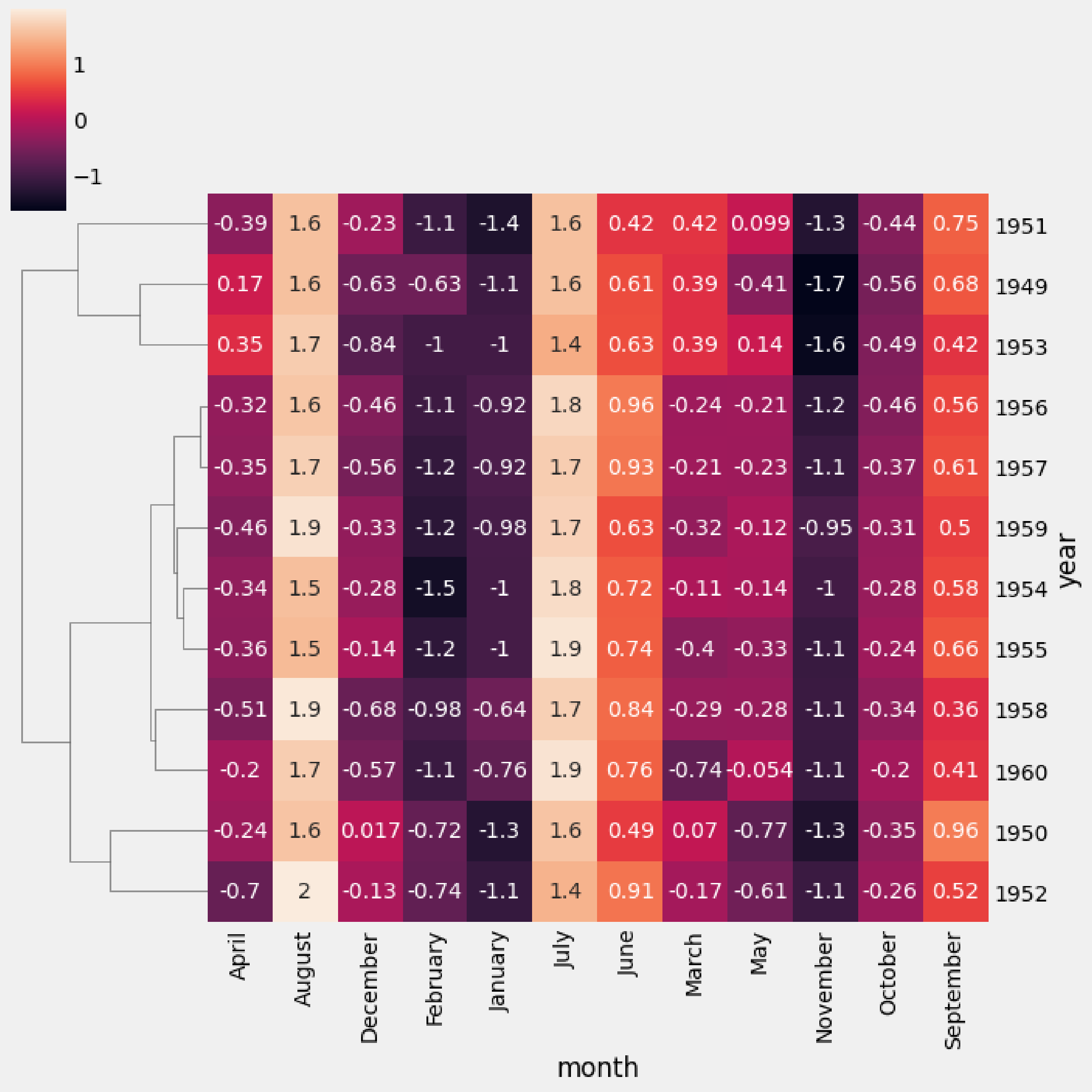
```
In [49]: sns.clustermap(x,z_score = 0,annot = True,row_cluster = False,metric = 'corre
         plt.show()
```



	December	September	February	April	March	May	July	November	October	August	January	June
1949	-0.63	0.68	-0.63	0.17	0.39	-0.41	1.6	-1.7	-0.56	1.6	-1.1	0.61
1950	0.017	0.96	-0.72	-0.24	0.07	-0.77	1.6	-1.3	-0.35	1.6	-1.3	0.49
1951	-0.23	0.75	-1.1	-0.39	0.42	0.099	1.6	-1.3	-0.44	1.6	-1.4	0.42
1952	-0.13	0.52	-0.74	-0.7	-0.17	-0.61	1.4	-1.1	-0.26	2	-1.1	0.91
1953	-0.84	0.42	-1	0.35	0.39	0.14	1.4	-1.6	-0.49	1.7	-1	0.63
1954	-0.28	0.58	-1.5	-0.34	-0.11	-0.14	1.8	-1	-0.28	1.5	-1	0.72
1955	-0.14	0.66	-1.2	-0.36	-0.4	-0.33	1.9	-1.1	-0.24	1.5	-1	0.74
1956	-0.46	0.56	-1.1	-0.32	-0.24	-0.21	1.8	-1.2	-0.46	1.6	-0.92	0.96
1957	-0.56	0.61	-1.2	-0.35	-0.21	-0.23	1.7	-1.1	-0.37	1.7	-0.92	0.93
1958	-0.68	0.36	-0.98	-0.51	-0.29	-0.28	1.7	-1.1	-0.34	1.9	-0.64	0.84
1959	-0.33	0.5	-1.2	-0.46	-0.32	-0.12	1.7	-0.95	-0.31	1.9	-0.98	0.63
1960	-0.57	0.41	-1.1	-0.2	-0.74	-0.054	1.9	-1.1	-0.2	1.7	-0.76	0.76

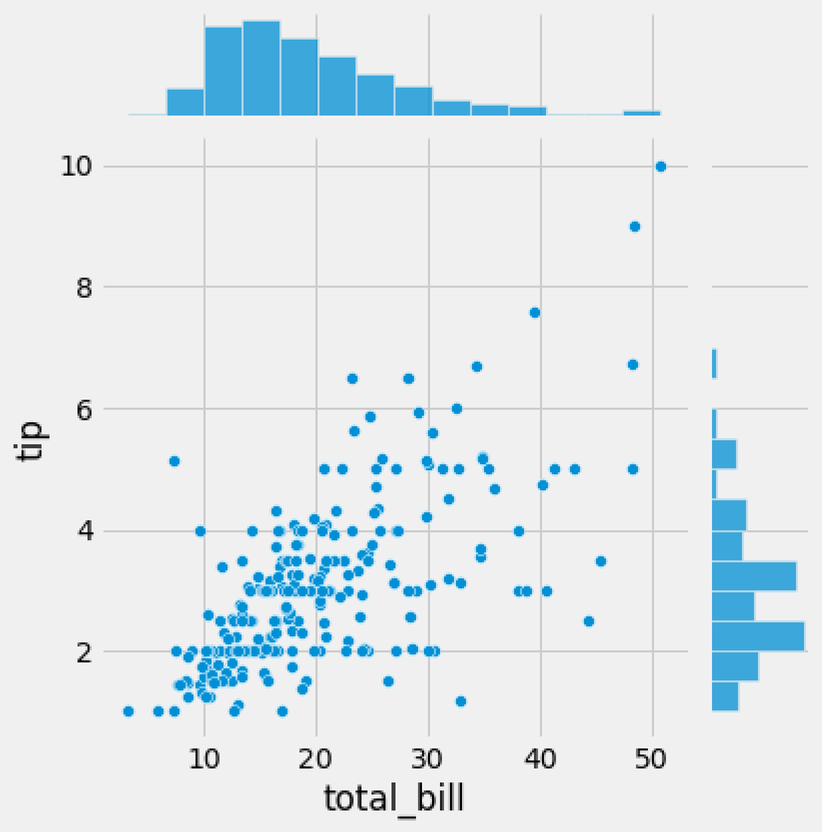
In [50]:

```
sns.clustermap(x,z_score = 0,annot = True,col_cluster = False,metric = 'corre
plt.show()
```



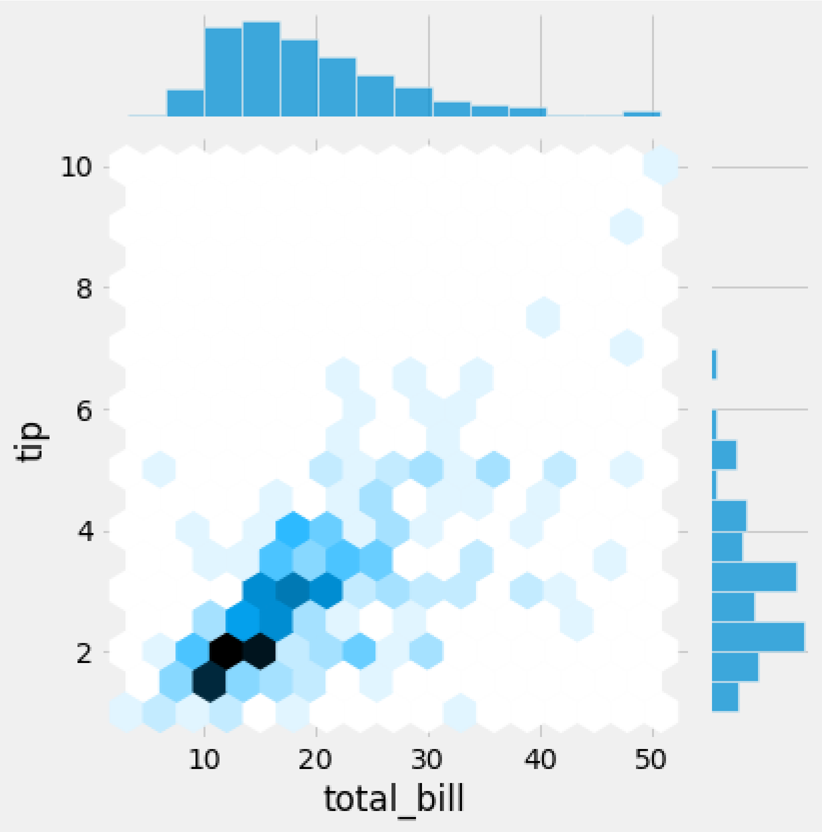
Joint Plot

```
In [51]: sns.jointplot(x = 'total_bill',y = 'tip',data = tips_df)
         plt.show()
```



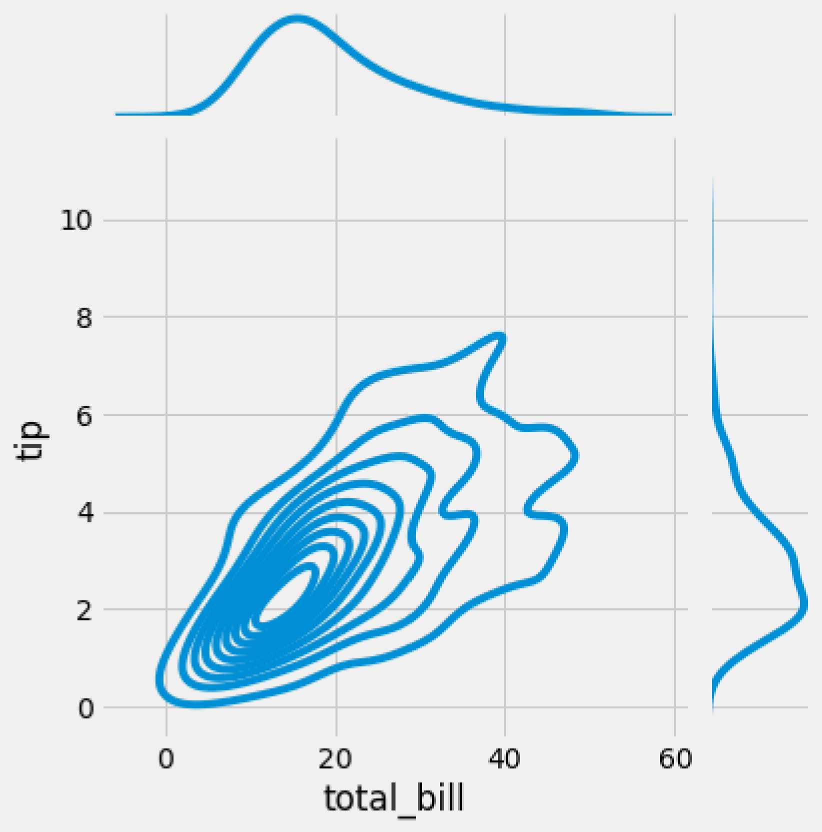
In [52]:

```
sns.jointplot(x = 'total_bill',y = 'tip',kind = 'hex',data = tips_df)
plt.show()
```

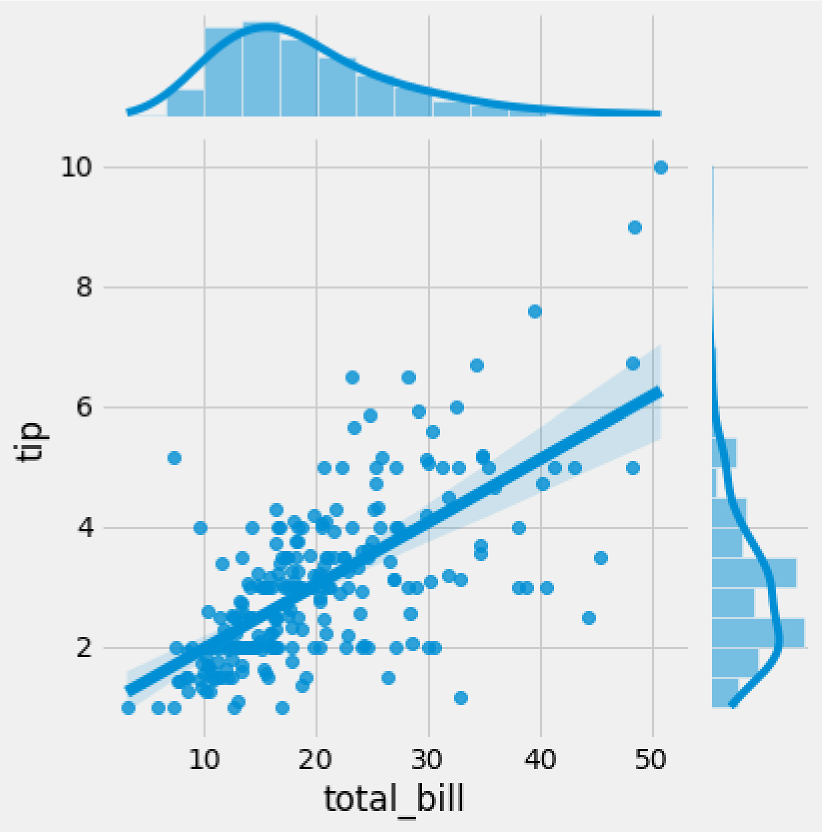


In [57]:

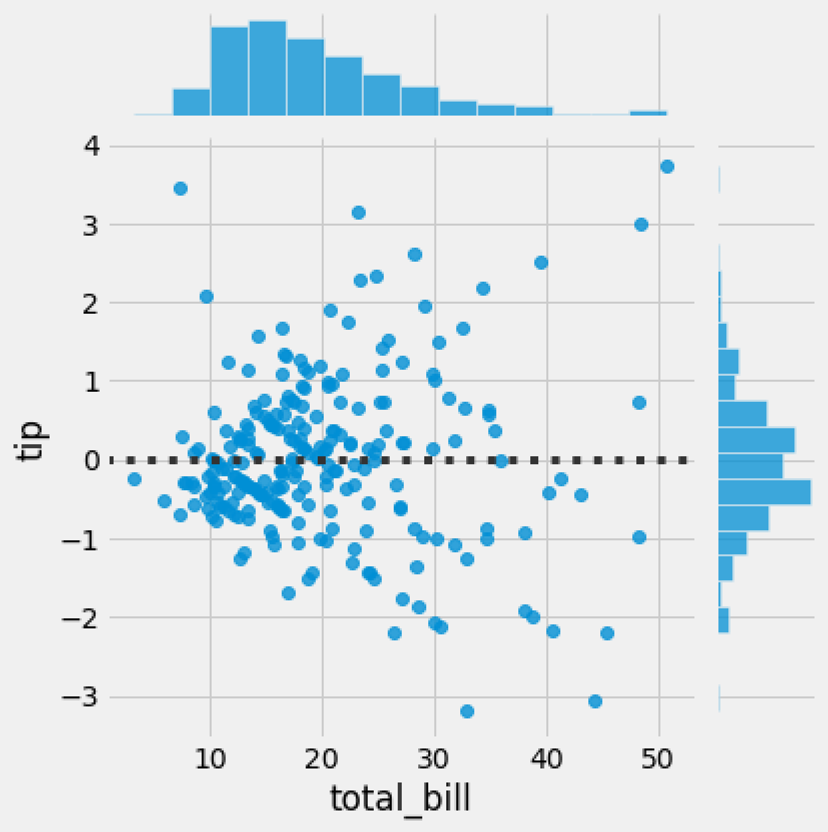
```
sns.jointplot(x = 'total_bill',y = 'tip',kind = 'kde',data = tips_df)
plt.show()
```



In [59]:
```
sns.jointplot(x = 'total_bill',y = 'tip',kind = 'reg',data = tips_df)
plt.show()
```



In [60]:
```
sns.jointplot(x = 'total_bill',y = 'tip',kind = 'resid',data = tips_df)
plt.show()
```



Pair Plot

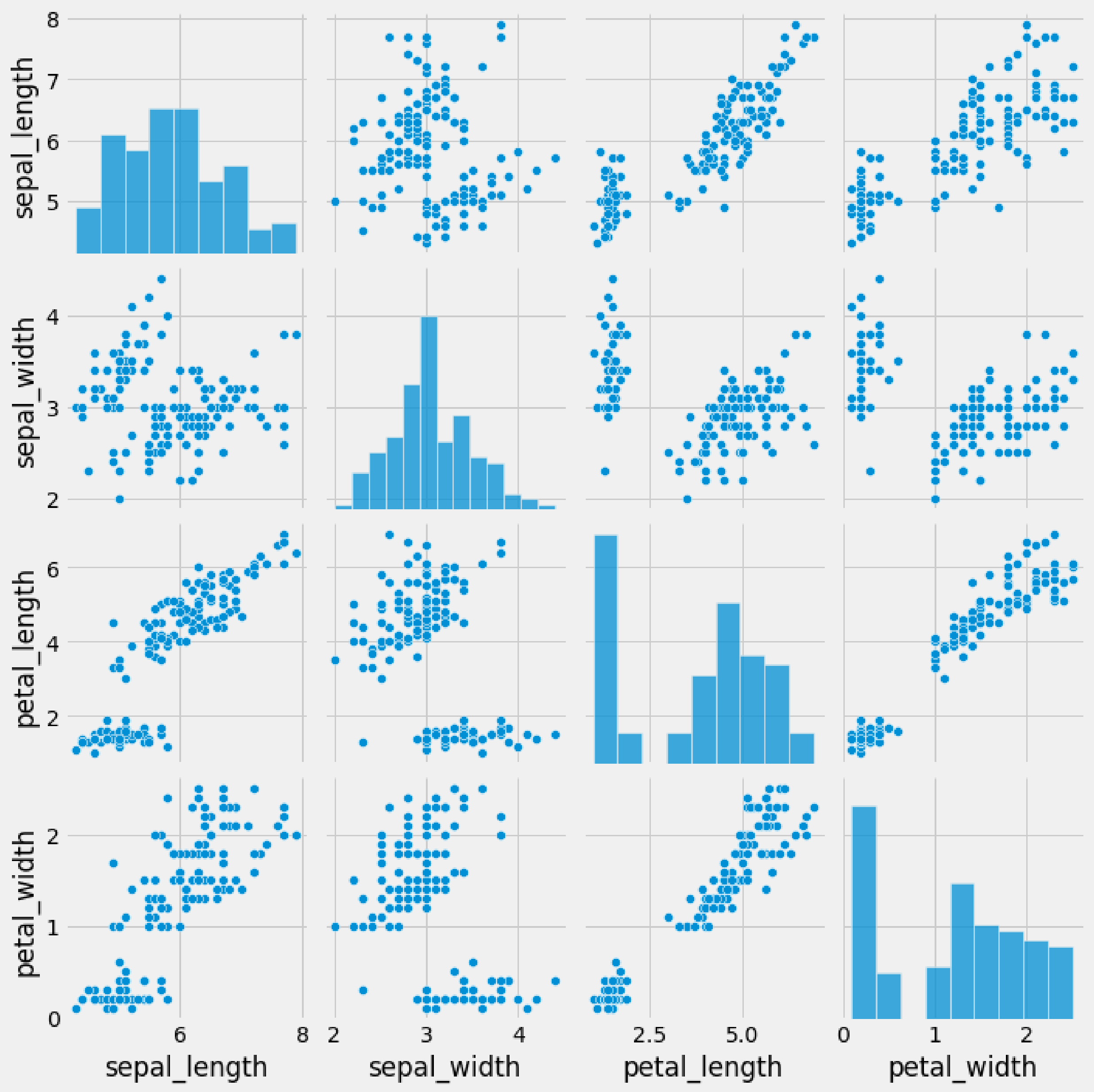
```
In [63]: plt.style.use('fivethirtyeight')
```

```
In [64]: iris_df = pd.read_csv('iris.csv')
         iris_df.head()
```

Out[64]:

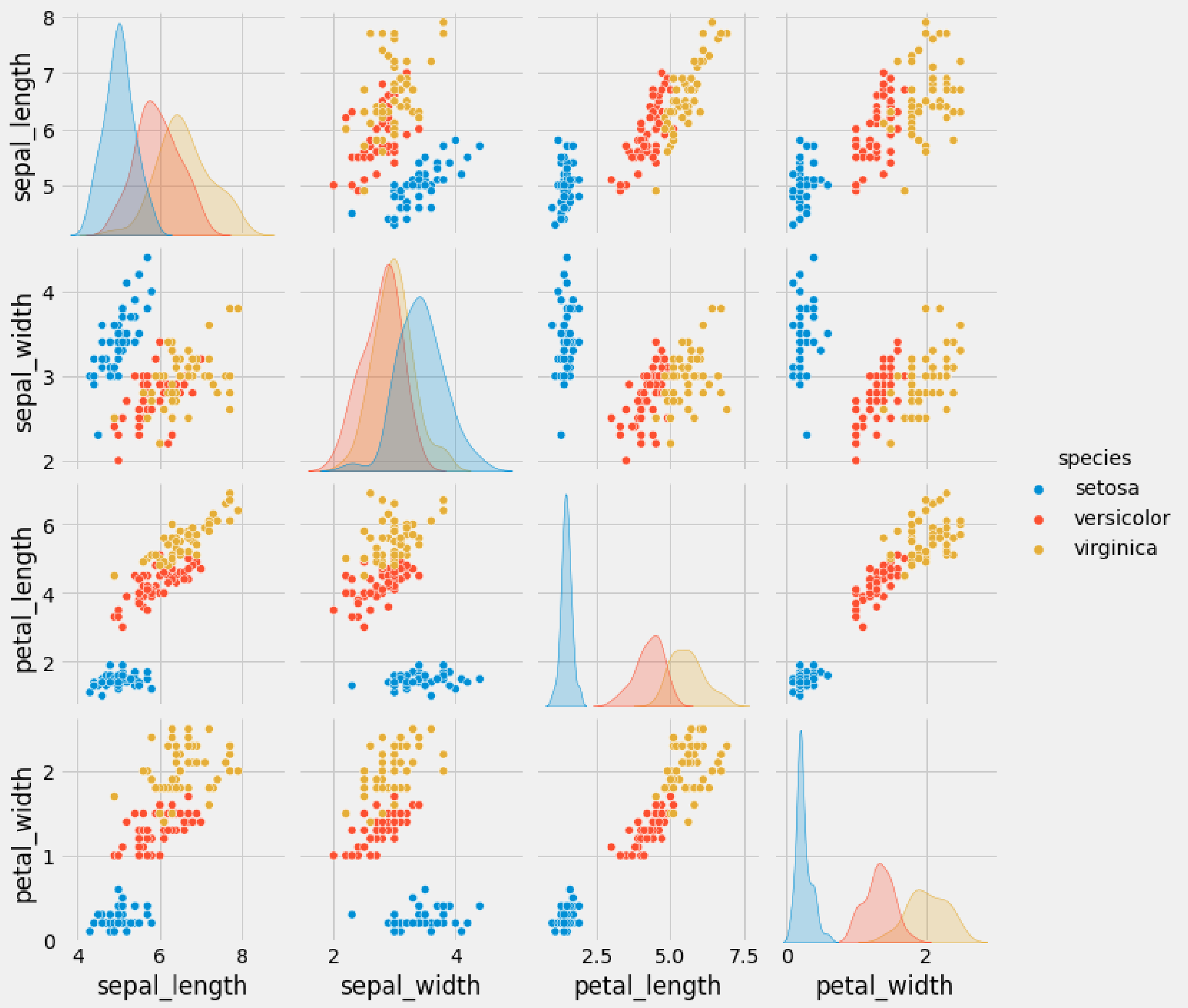
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [65]: sns.pairplot(iris_df)
         plt.show()
```



In [66]:

```
sns.pairplot(iris_df,hue = 'species')
plt.show()
```



Dist Plot

In [68]:

```
plt.style.use('fivethirtyeight')
```

In [69]:

```
titanic_df = pd.read_csv('titanic.csv')
titanic_df.head()
```

Out[69]:

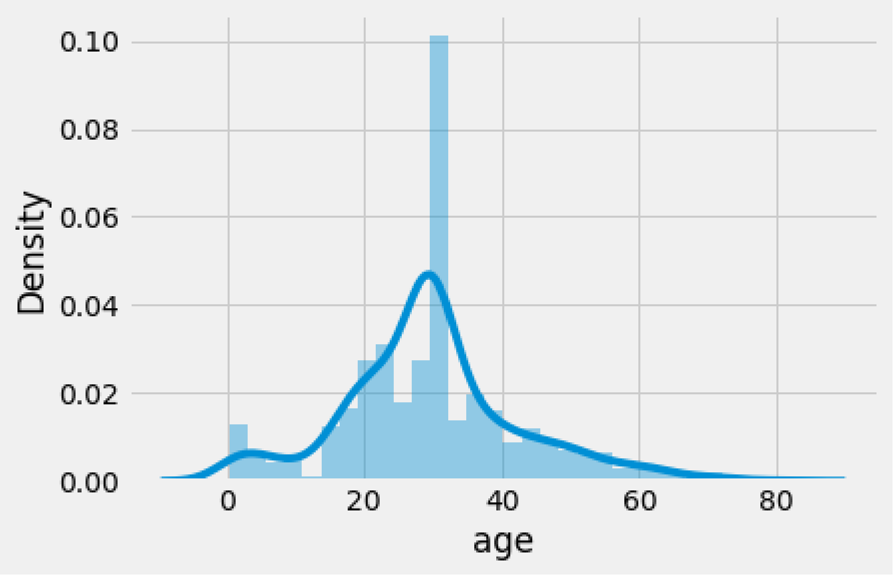
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	de
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	N
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	N
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	N

In [70]:

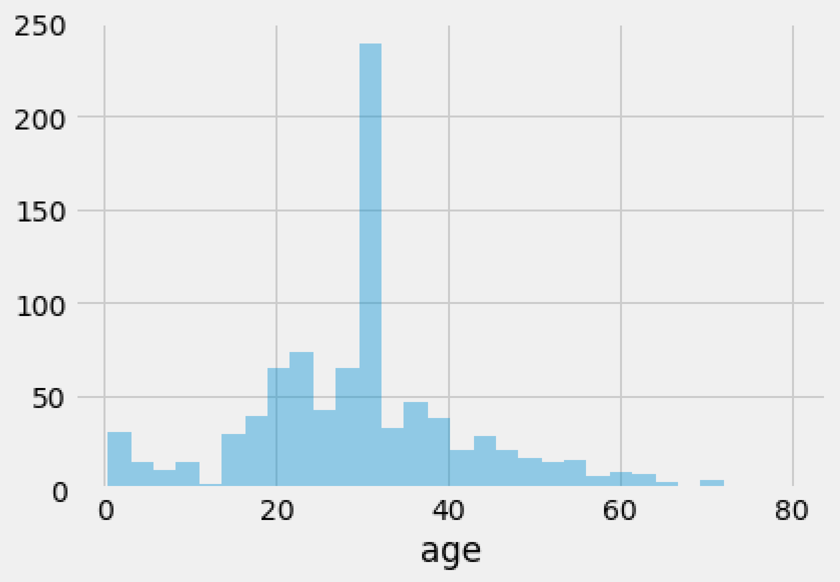
```
titanic_df['age'].fillna(titanic_df['age'].mean(),inplace = True)
```

In [76]:

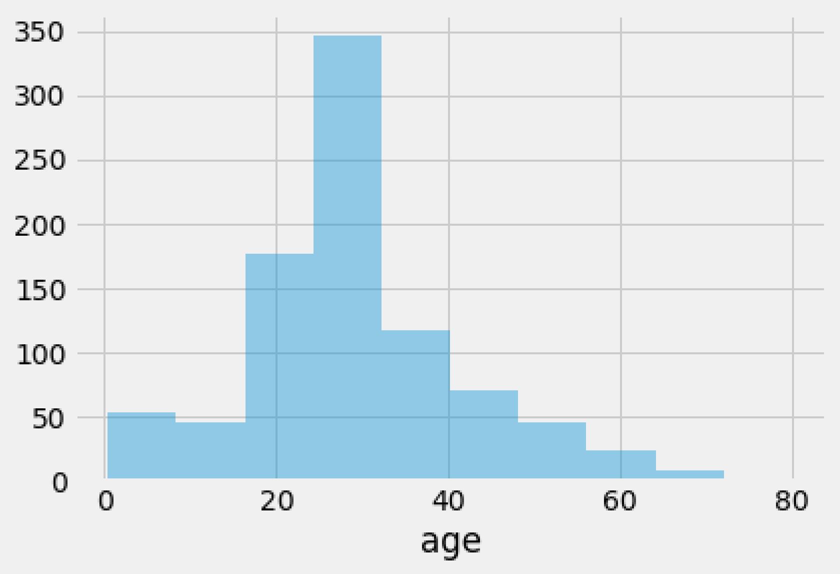
```
sns.distplot(titanic_df['age'])
plt.show()
```



In [74]:
```
sns.distplot(titanic_df['age'],kde = False)
plt.show()
```

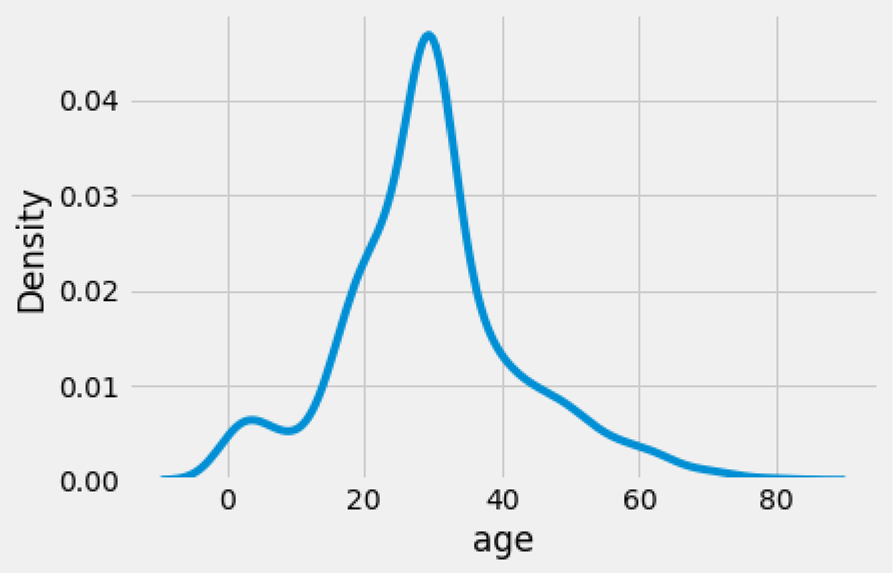


In [75]:
```
sns.distplot(titanic_df['age'],bins = 10,kde = False)
plt.show()
```



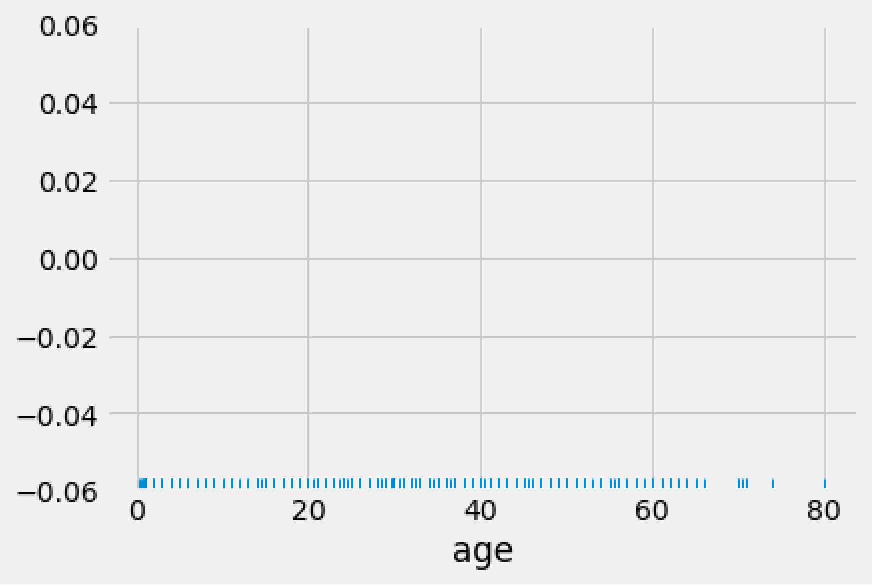
In [81]:
```
sns.distplot(titanic_df['age'],hist = False)
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```



In [83]:
```
sns.distplot(titanic_df['age'],hist = False,kde = False,rug = True)
plt.show()
```

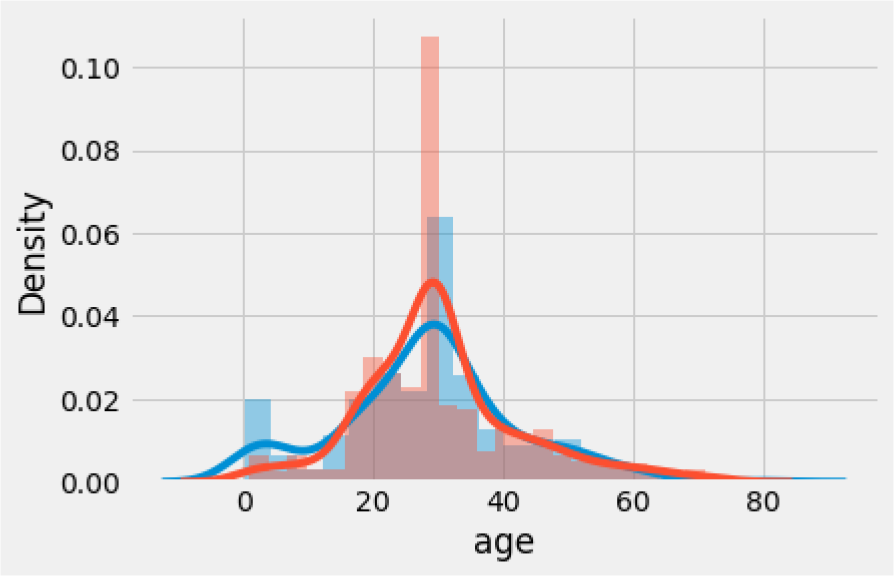
```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2103: FutureWarning: The `axis` variable is no
longer used and will be removed. Instead, assign variables directly to `x` o
r `y`.
  warnings.warn(msg, FutureWarning)
```



In [85]:

```
sns.distplot(titanic_df[titanic_df['survived']==1]['age'])
sns.distplot(titanic_df[titanic_df['survived']==0]['age'])
plt.show()
```

```
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\prasad jadhav\AppData\Local\Programs\Python\Python310\lib\site-pack
ages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



Thank You

CONNECT WITH ME :

LinkedIn GitHub kaggle Medium

PRASADMJADHAV2