In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
iris = sns.load_dataset('iris')
iris.head()
```

Out[2]:

	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

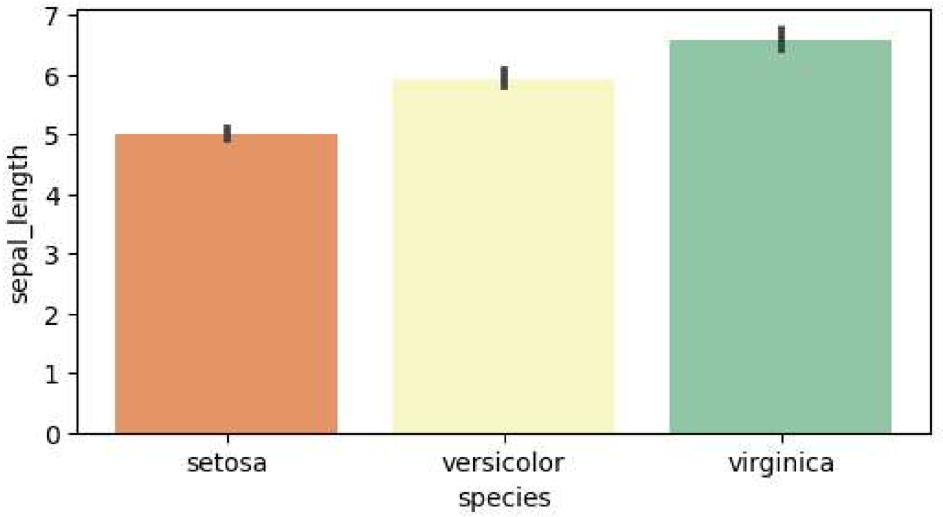
In [3]:
```
sns.get_dataset_names()
```

```
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

BARPLOT

In [4]:
```
plt.figure(figsize=(6,3))
sns.barplot(x="species", y="sepal_length", data=iris, palette="Spectral")
plt.plot()
```

Out[4]:
```
[]
```



In [5]:
```
mtcars = pd.read_csv(".//DATA//mtcars.csv")
mtcars.head()
```

Out[5]:

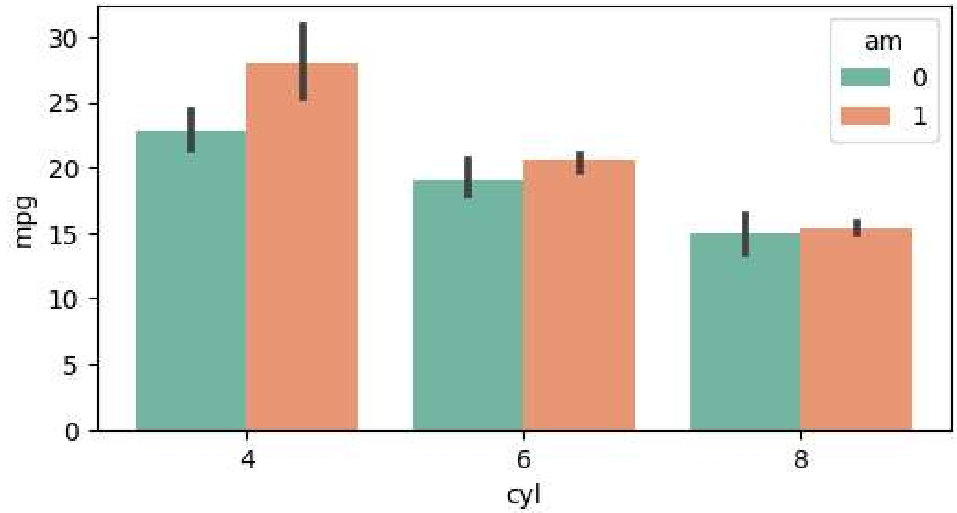
	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

In [6]:
```
groupby_cyl = mtcars.groupby("cyl")
groupby_cyl["mpg"].mean()
```

Out[6]:
```
cyl
4    26.663636
6    19.742857
8    15.100000
Name: mpg, dtype: float64
```

In [7]:
```
plt.figure(figsize=(6,3))
sns.barplot(x="cyl", y="mpg", hue="am", data=mtcars, palette="Set2")
plt.plot()
```

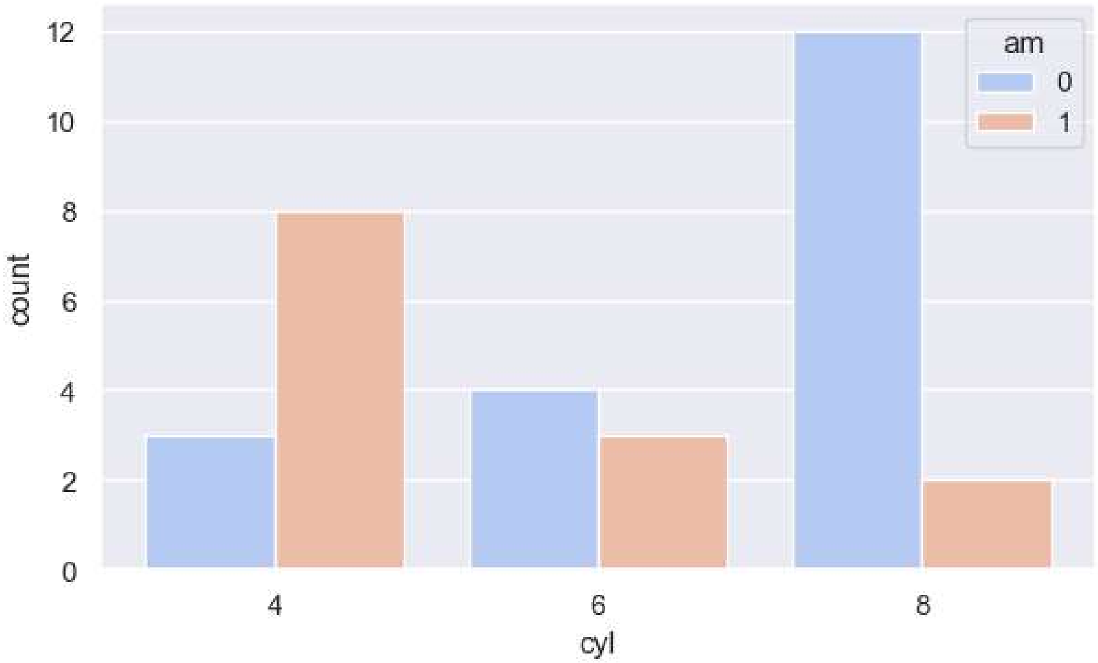
Out[7]:
```
[]
```



COUNTPLOT

In [8]:
```
sns.set_style("darkgrid")
sns.set(rc={'figure.figsize':(7,4)})
sns.countplot(x="cyl", hue="am", data=mtcars, palette="coolwarm")
plt.plot()
```

Out[8]:
```
[]
```

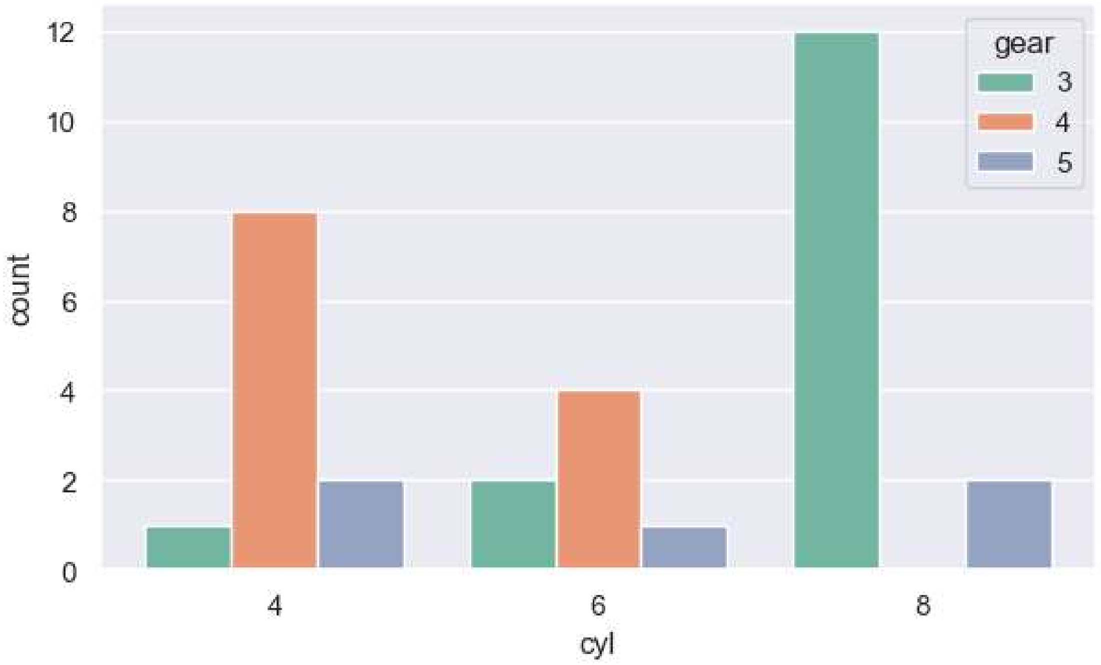


In [9]:

```
sns.countplot(x="cyl", hue="gear", data=mtcars, palette="Set2")
plt.plot()
```

Out[9]:

```
[]
```



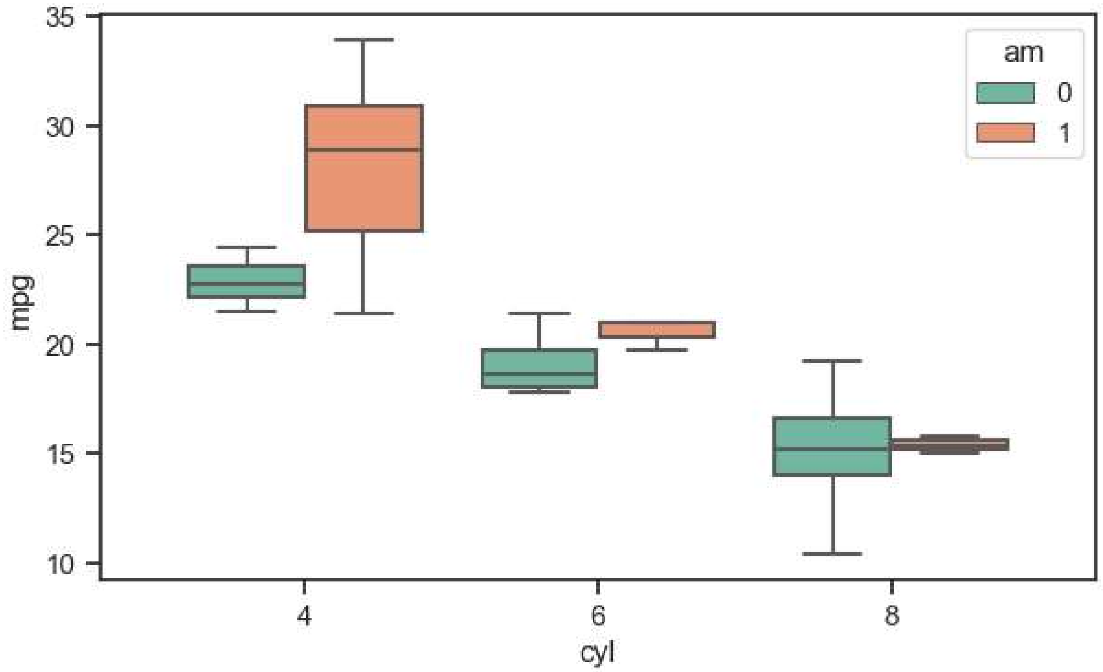
BOXPLOT

In [10]:

```
sns.set_style("ticks")
sns.boxplot(x="cyl", y="mpg", hue="am", data=mtcars, palette="Set2")
plt.plot()
```

Out[10]:

```
[]
```



In [11]:

```
tips = sns.load_dataset('tips')
tips.head()
```

Out[11]:

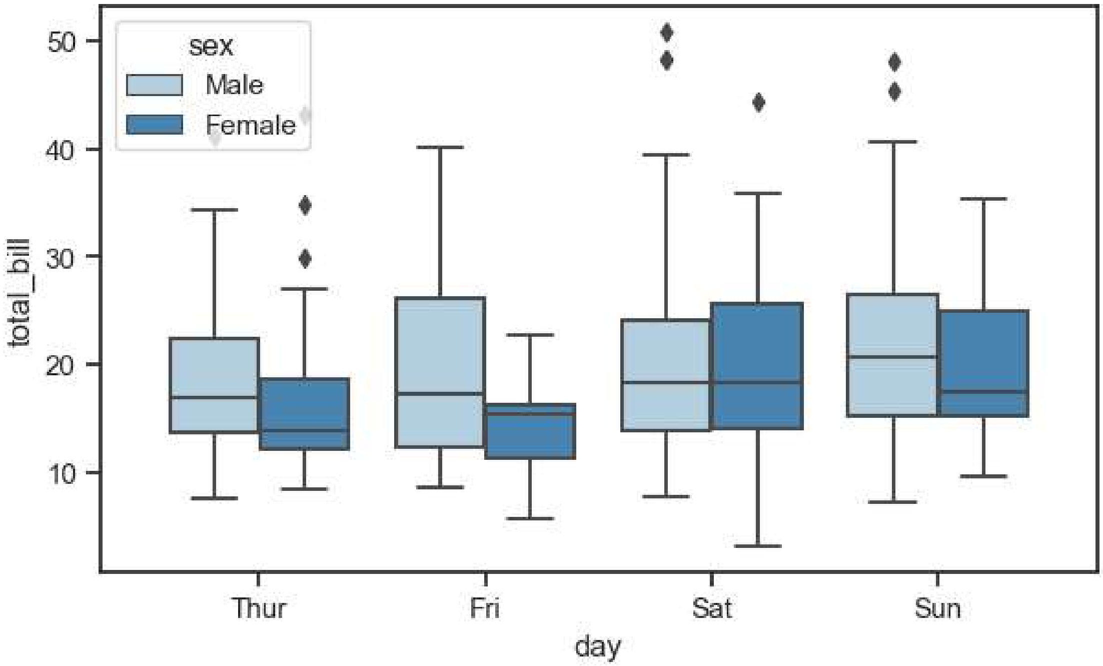
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [12]:

```
sns.boxplot(x="day", y="total_bill", hue="sex", data=tips, palette="Blues")
plt.plot()
```

Out[12]:

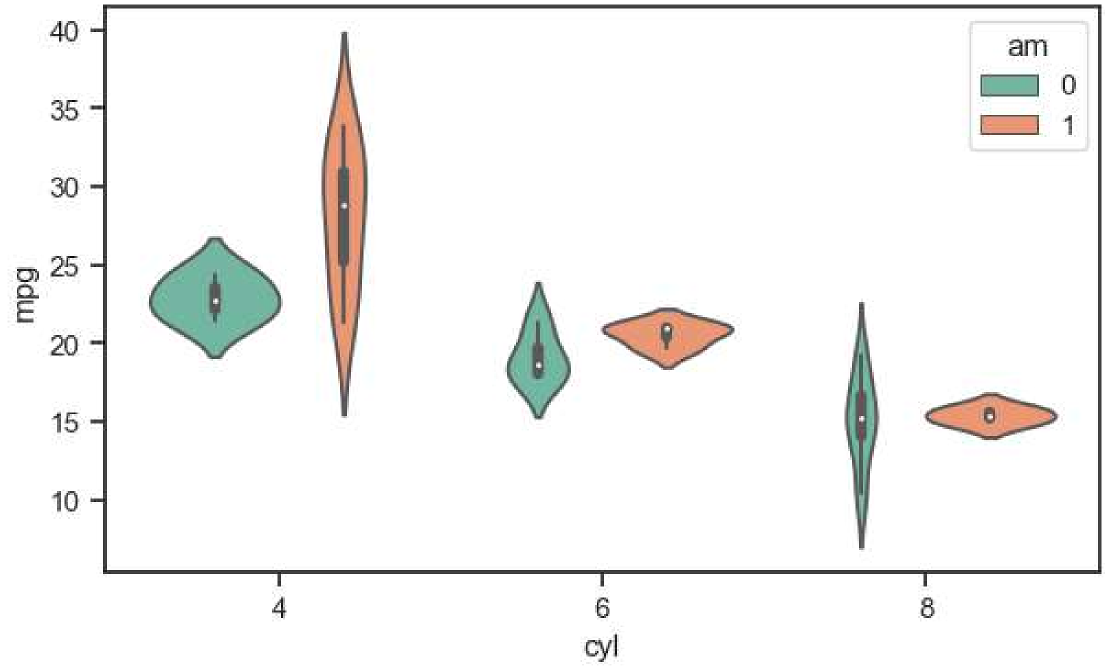
```
[]
```



VIOLINPLOT

```
In [13]: sns.violinplot(x="cyl", y="mpg", hue="am", data=mtcars, palette="Set2")
         plt.plot()
```

```
Out[13]: []
```

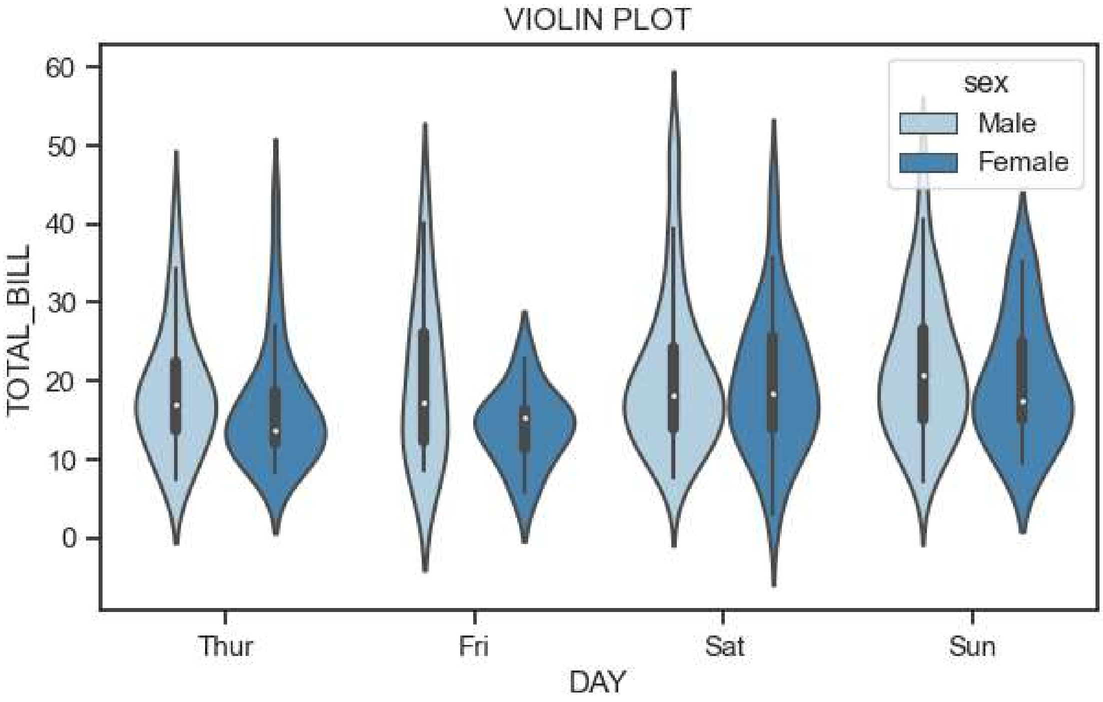


In [14]:

```
sns.set_style("ticks")
sns.violinplot(x="day", y="total_bill", hue="sex", data=tips, palette="Blues")
plt.title("VIOLIN PLOT")
plt.xlabel('DAY')
plt.ylabel('TOTAL_BILL')
```

Out[14]:

```
Text(0, 0.5, 'TOTAL_BILL')
```



CATPLOT

In [15]:

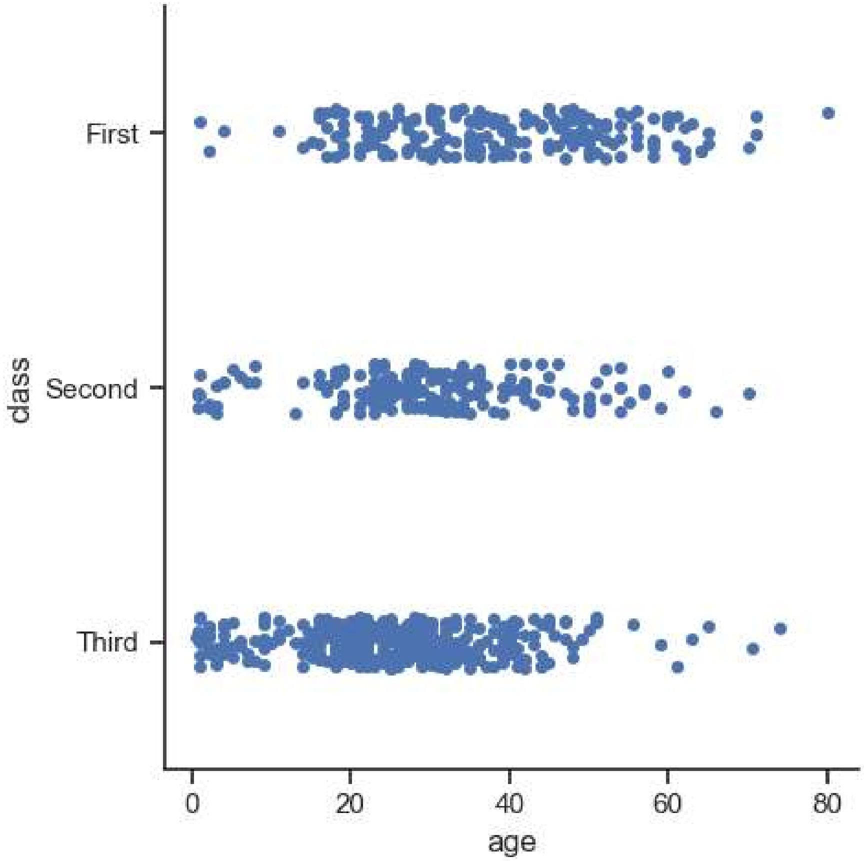
```
sns.set_style("ticks")
df = sns.load_dataset("titanic")
df.head(10)
```

Out[15]:

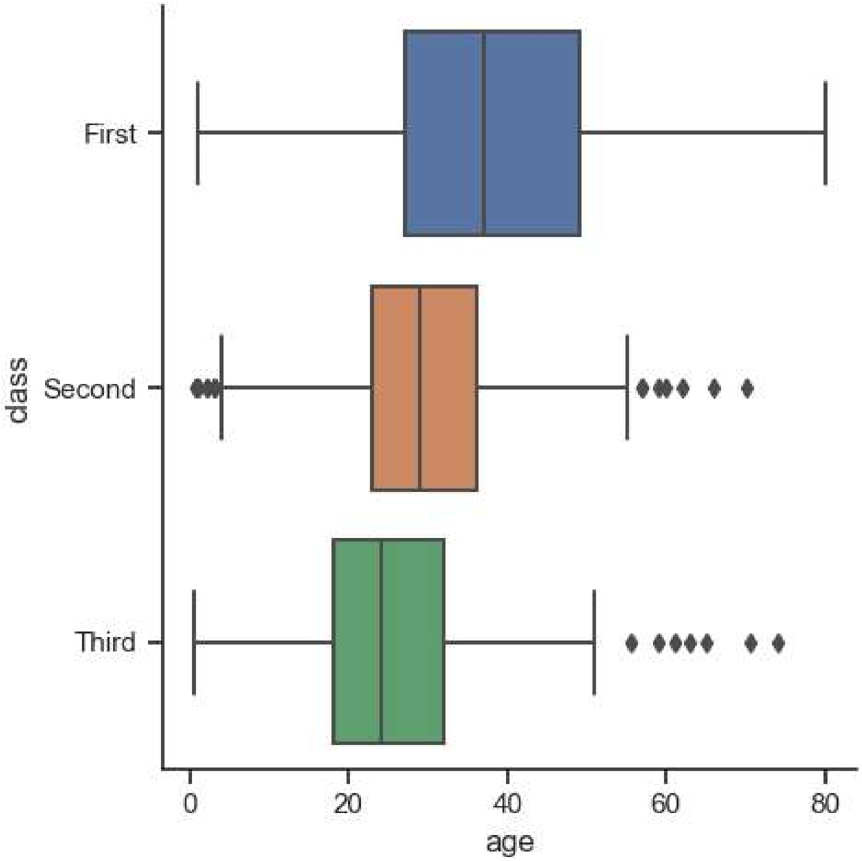
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embarl
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	South
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Che
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	South
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	South
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	South
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	Quee
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	South
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN	South
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	NaN	South
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	NaN	Che

```
In [16]: sns.catplot(data=df, x="age", y="class")
```

```
Out[16]: <seaborn.axisgrid.FacetGrid at 0x28513a0b1c0>
```

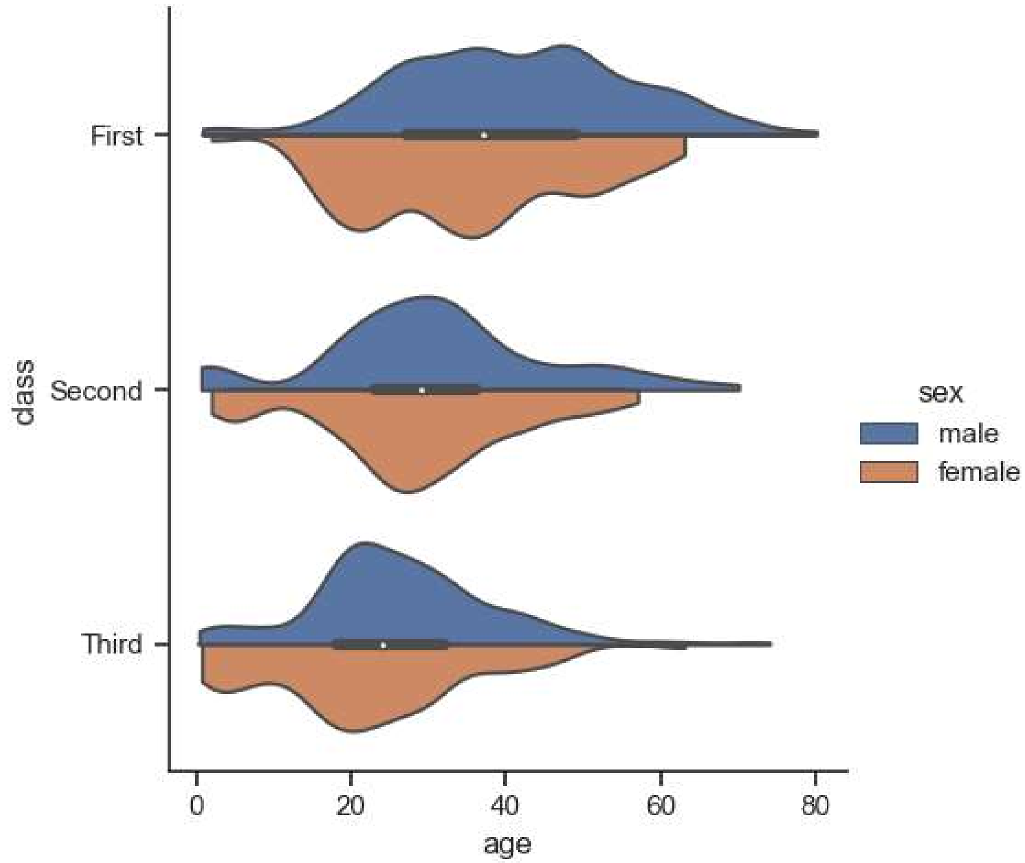


```
In [17]: sns.catplot(data=df, x="age", y="class", kind="box")

Out[17]: <seaborn.axisgrid.FacetGrid at 0x285138c5720>
```



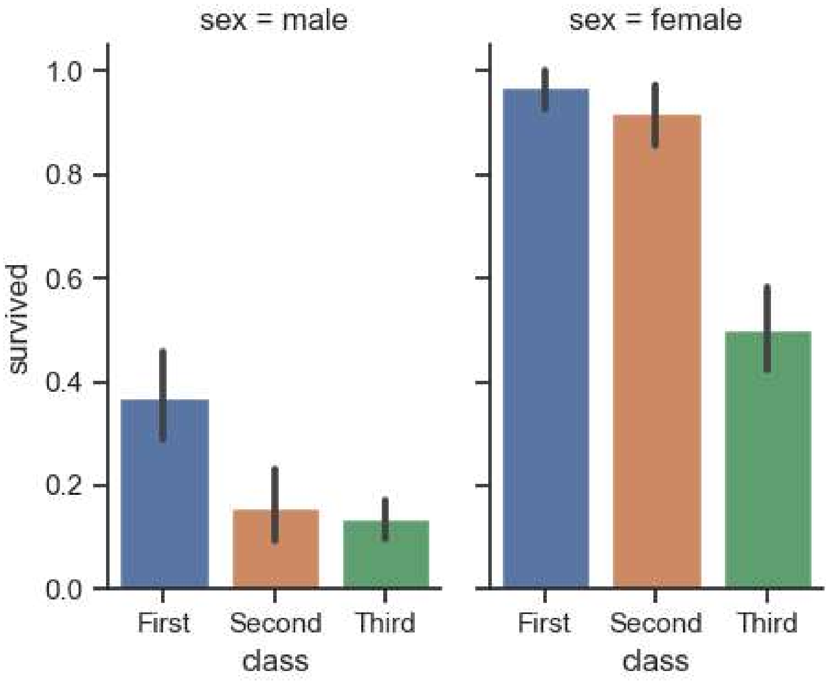
```
In [18]: sns.catplot(
             data=df, x="age", y="class", hue="sex",
             kind="violin", bw=.25, cut=0, split=True,
         )
```

```
Out[18]: <seaborn.axisgrid.FacetGrid at 0x28516cef220>
```



```
In [19]: sns.catplot(
             data=df, x="class", y="survived", col="sex",
             kind="bar", height=4, aspect=.6,
         )
```

```
Out[19]: <seaborn.axisgrid.FacetGrid at 0x28516cefbe0>
```



DISTRIBUTION PLOT

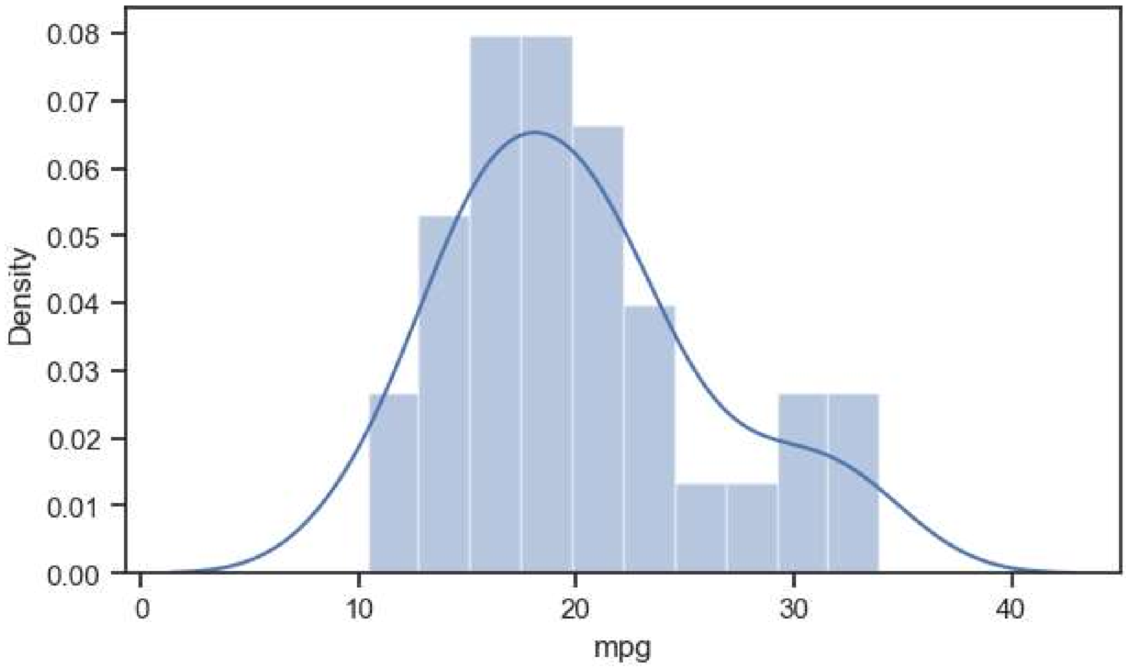
In [20]:

```
sns.distplot(mtcars.mpg, bins=10)
```

```
C:\Users\Gunjan Kumar\AppData\Local\Temp\ipykernel_14768\3310083437.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.co
m/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(mtcars.mpg, bins=10)
```

Out[20]:

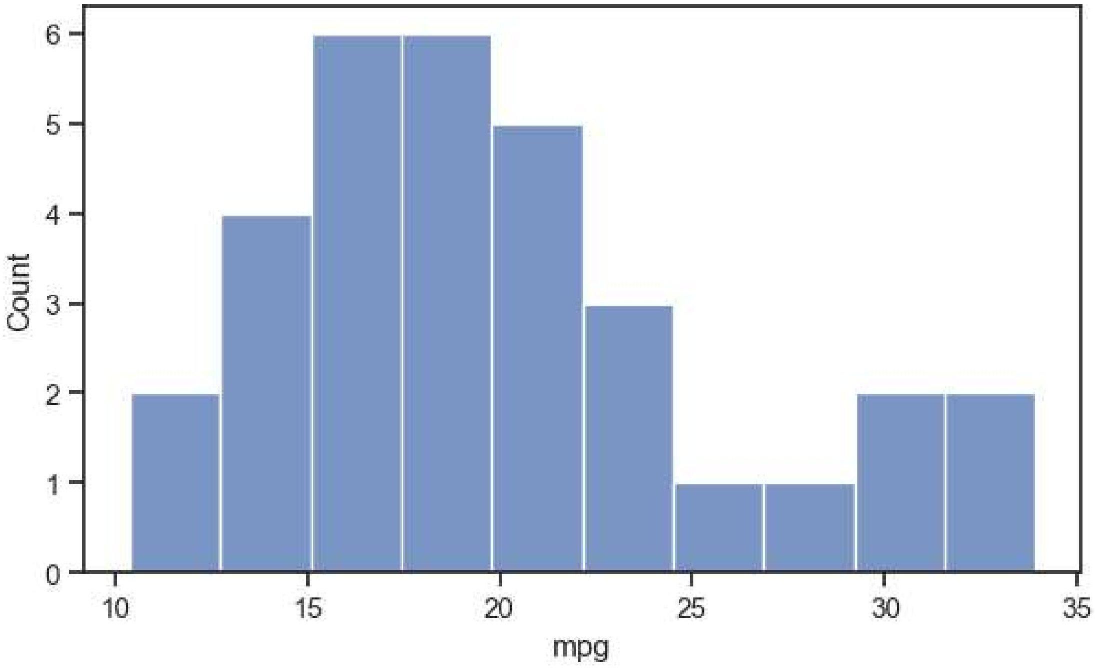
```
<Axes: xlabel='mpg', ylabel='Density'>
```



HISTOGRAM PLOT

In [21]:
```
sns.set_style("whitegrid")
sns.set_style("ticks")
sns.histplot(mtcars.mpg, bins=10)
```

Out[21]:
```
<Axes: xlabel='mpg', ylabel='Count'>
```



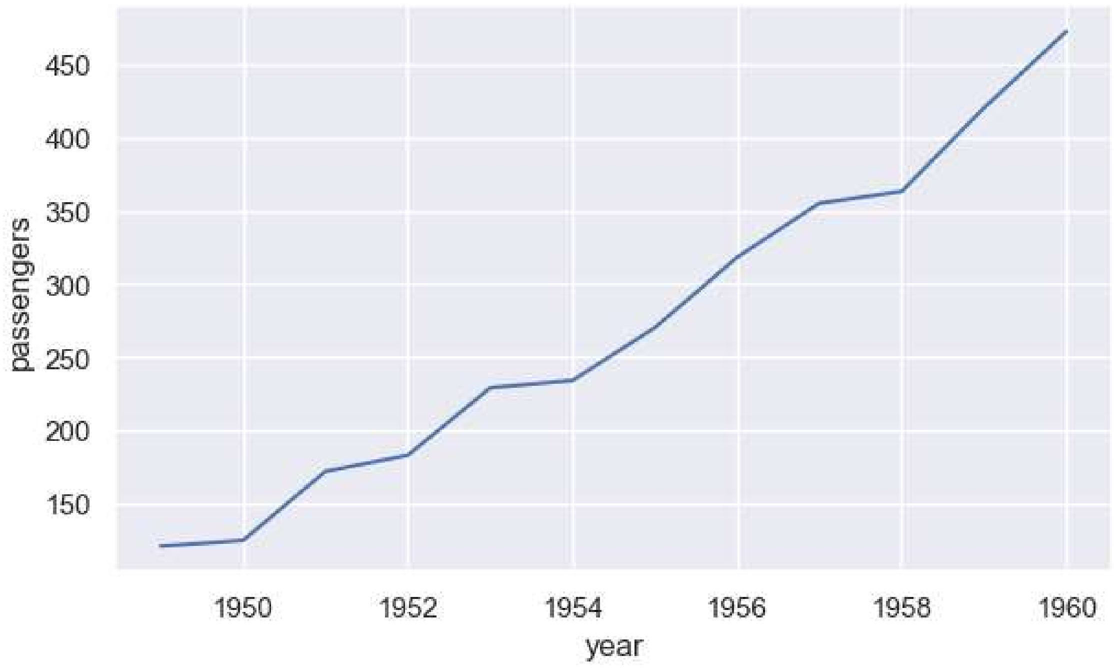
LINEPLOT

In [22]:
```
flights = sns.load_dataset("flights")
flights.head(), flights["month"].unique()
```

Out[22]:
```
(   year month  passengers
 0  1949   Jan         112
 1  1949   Feb         118
 2  1949   Mar         132
 3  1949   Apr         129
 4  1949   May         121,
 ['Jan', 'Feb', 'Mar', 'Apr', 'May', ..., 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
 Length: 12
 Categories (12, object): ['Jan', 'Feb', 'Mar', 'Apr', ..., 'Sep', 'Oct', 'Nov', 'Dec'])
```

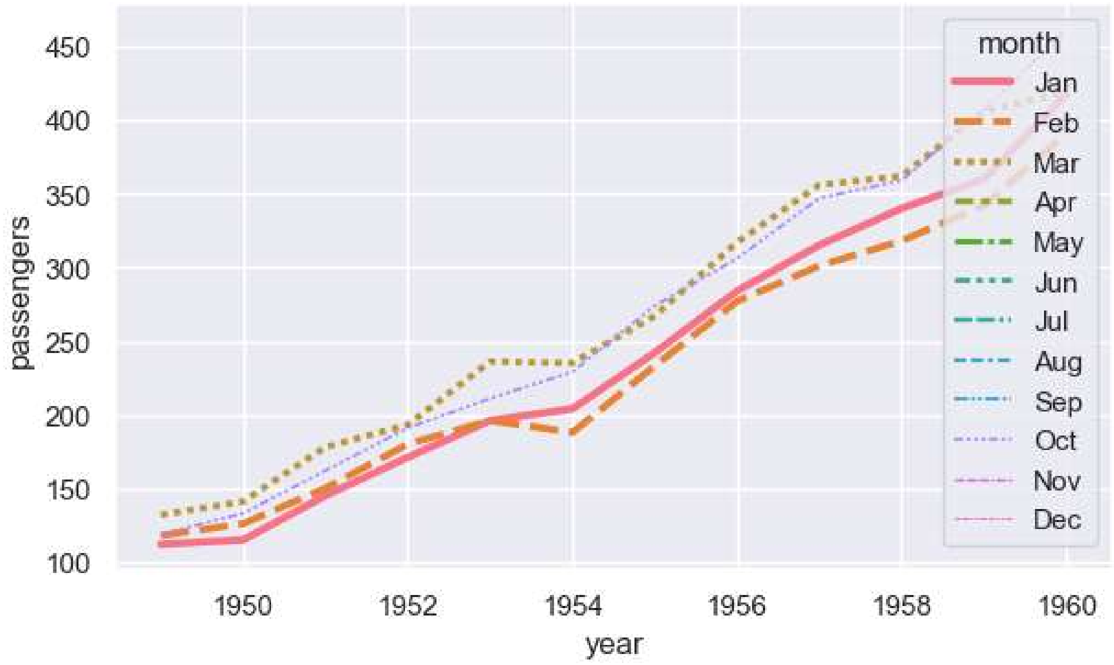
```
In [23]: sns.set_style("darkgrid")
         may_flights = flights.query("month == 'May'")
         sns.lineplot(data=may_flights, x="year", y="passengers")
```

```
Out[23]: <Axes: xlabel='year', ylabel='passengers'>
```



```
In [24]: data_flights = flights.loc[flights["month"].isin(["Oct", "Jan", "Feb", "Mar"])]
         sns.lineplot(data=data_flights,
                      x="year", y="passengers", hue="month", style="month", size="month")
```

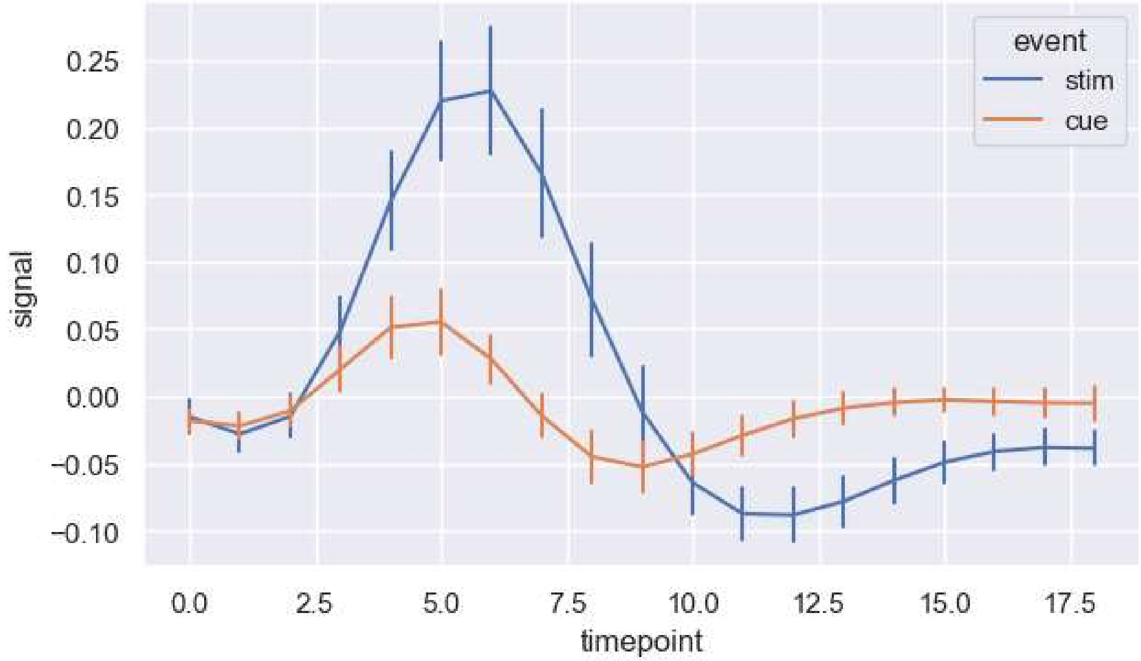
```
Out[24]: <Axes: xlabel='year', ylabel='passengers'>
```



In [25]:
```
fmri = sns.load_dataset("fmri")
print(fmri.head())
sns.lineplot(
    data=fmri, x="timepoint", y="signal", hue="event", err_style="bars", errorbar=("se", 
)
```

```
  subject  timepoint event    region    signal
0     s13         18  stim  parietal -0.017552
1      s5         14  stim  parietal -0.080883
2     s12         18  stim  parietal -0.081033
3     s11         18  stim  parietal -0.046134
4     s10         18  stim  parietal -0.037970
```

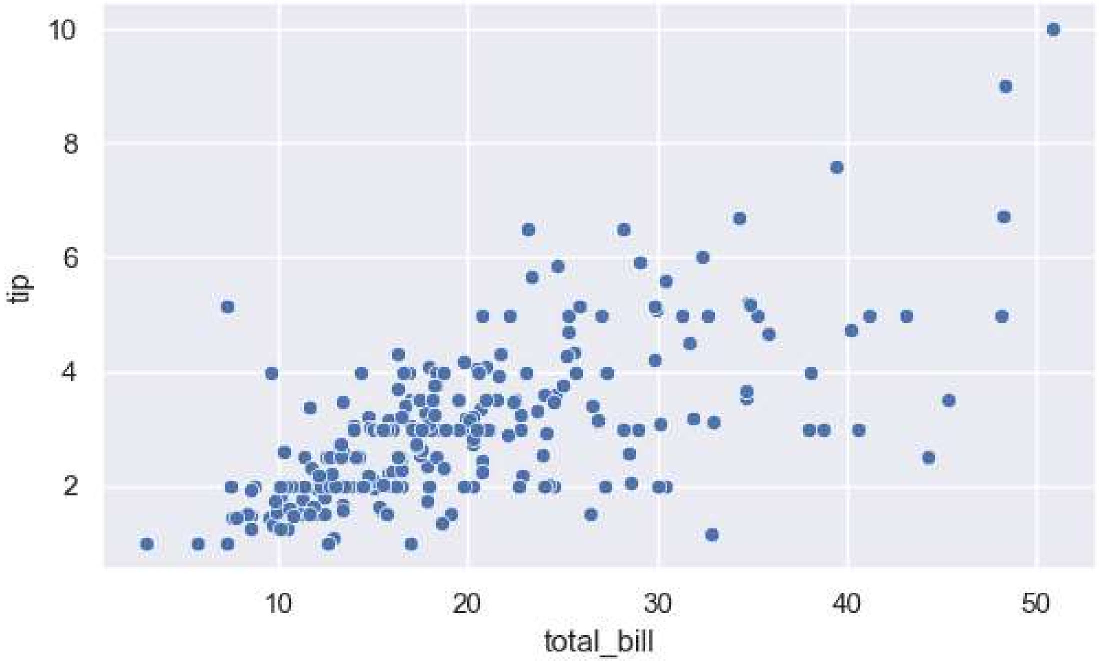
Out[25]:
```
<Axes: xlabel='timepoint', ylabel='signal'>
```



HISTOGRAM

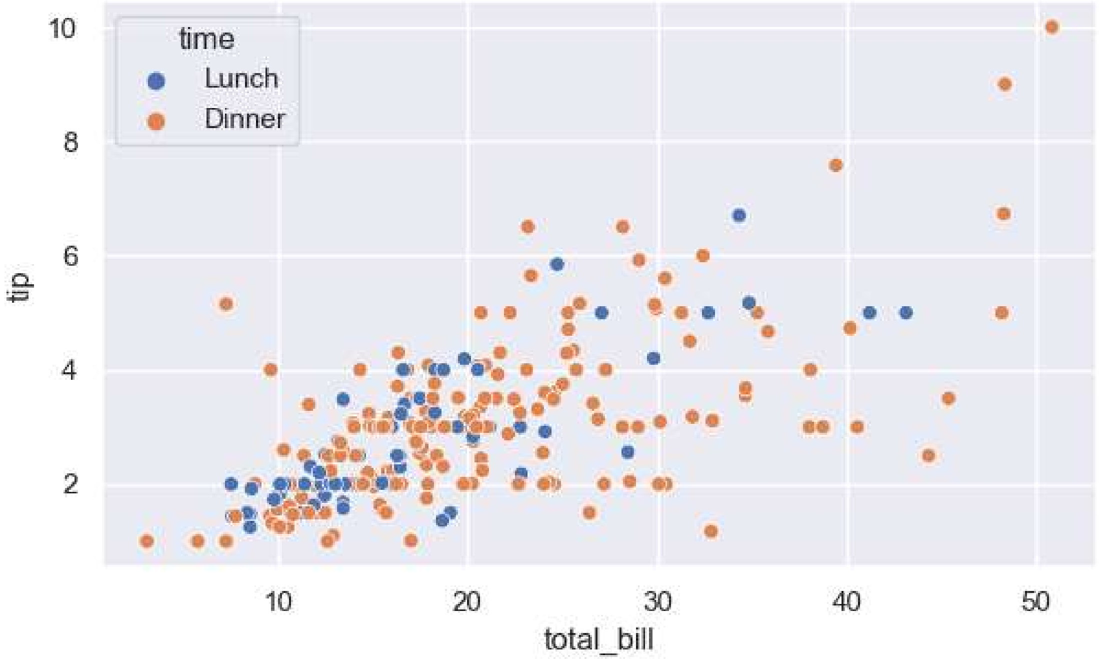
In [26]:
```
sns.scatterplot(data=tips, x="total_bill", y="tip")
```

Out[26]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



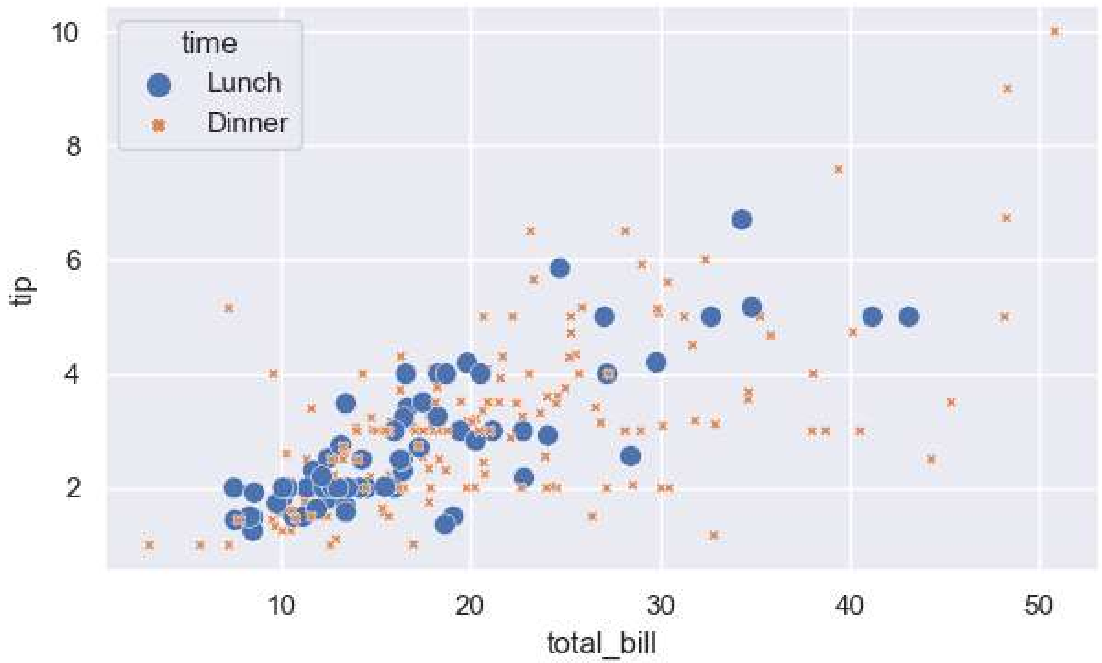
In [27]:
```
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="time")
```

Out[27]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



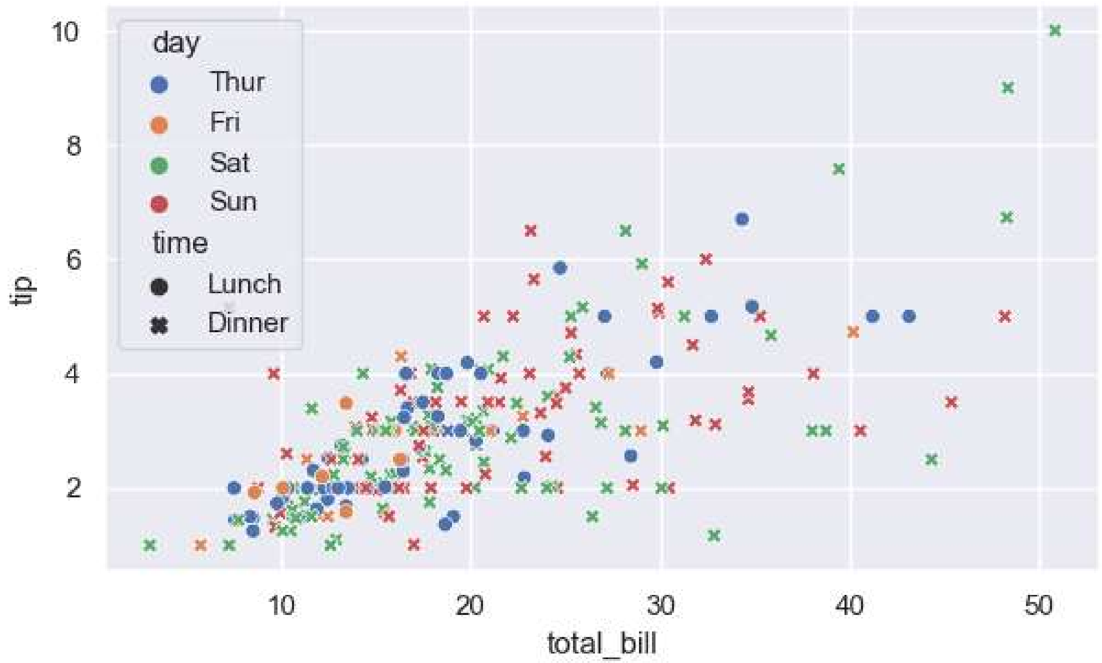
```
In [28]: sns.scatterplot(data=tips, x="total_bill", y="tip", hue="time", style="time", size="time"
```

```
Out[28]: <Axes: xlabel='total_bill', ylabel='tip'>
```



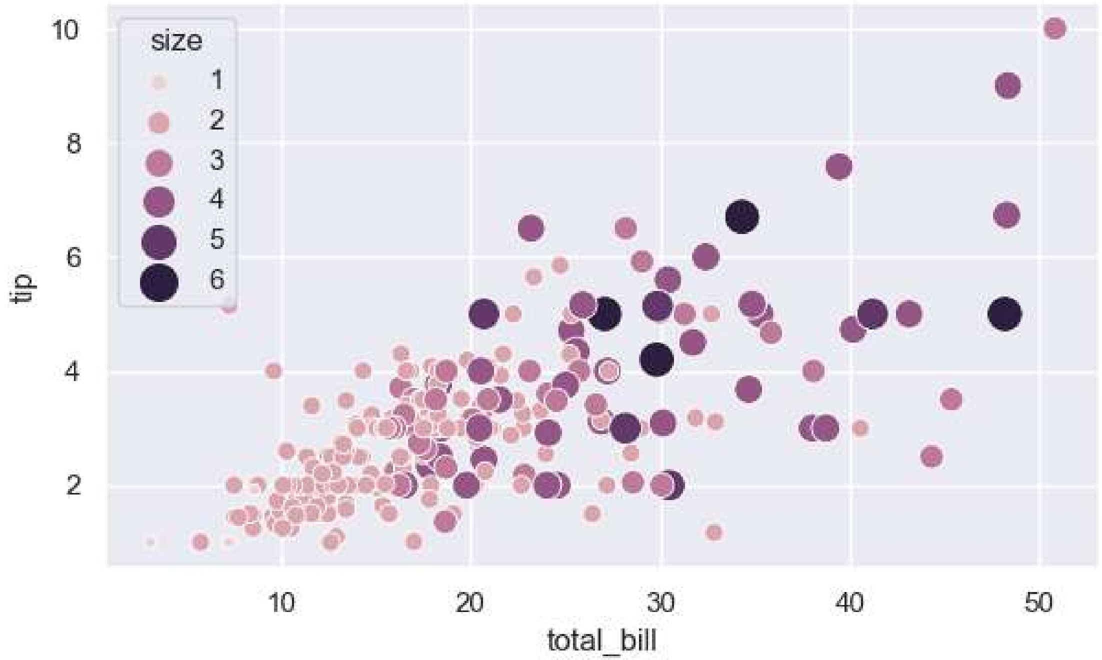
```
In [29]: sns.scatterplot(data=tips, x="total_bill", y="tip", hue="day", style="time")
```

```
Out[29]: <Axes: xlabel='total_bill', ylabel='tip'>
```



In [30]:
```
sns.scatterplot(
    data=tips, x="total_bill", y="tip", hue="size", size="size",
    sizes=(20, 200), legend="full"
)
```

Out[30]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



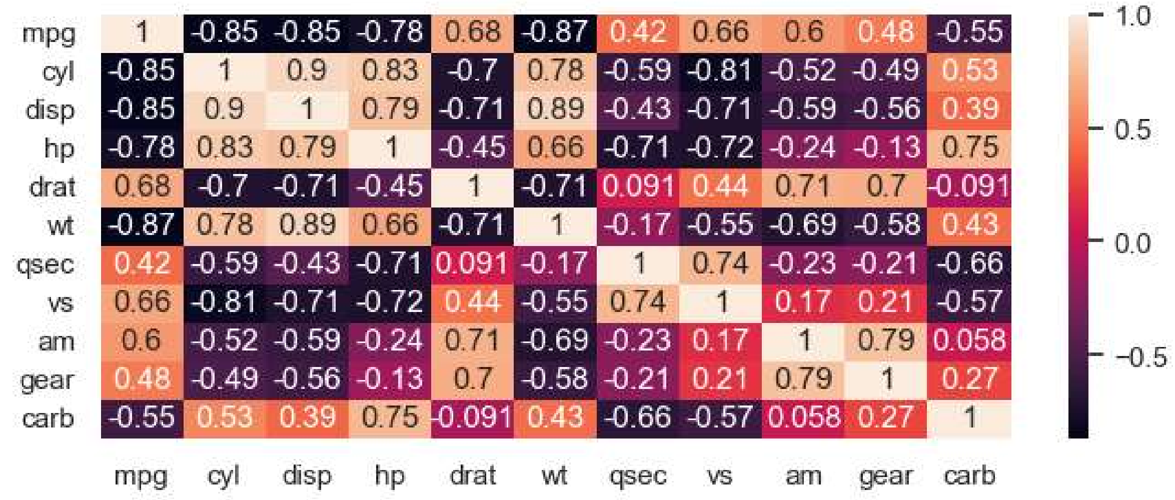
In [31]:
```
mtcars.loc[:,"mpg":"carb"].corr()
```

Out[31]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gea
mpg	1.000000	-0.852162	-0.847551	-0.776168	0.681172	-0.867659	0.418684	0.664039	0.599832	0.48028
cyl	-0.852162	1.000000	0.902033	0.832447	-0.699938	0.782496	-0.591242	-0.810812	-0.522607	-0.49268
disp	-0.847551	0.902033	1.000000	0.790949	-0.710214	0.887980	-0.433698	-0.710416	-0.591227	-0.55556
hp	-0.776168	0.832447	0.790949	1.000000	-0.448759	0.658748	-0.708223	-0.723097	-0.243204	-0.12570
drat	0.681172	-0.699938	-0.710214	-0.448759	1.000000	-0.712441	0.091205	0.440278	0.712711	0.69961
wt	-0.867659	0.782496	0.887980	0.658748	-0.712441	1.000000	-0.174716	-0.554916	-0.692495	-0.58328
qsec	0.418684	-0.591242	-0.433698	-0.708223	0.091205	-0.174716	1.000000	0.744535	-0.229861	-0.21268
vs	0.664039	-0.810812	-0.710416	-0.723097	0.440278	-0.554916	0.744535	1.000000	0.168345	0.20602
am	0.599832	-0.522607	-0.591227	-0.243204	0.712711	-0.692495	-0.229861	0.168345	1.000000	0.79405
gear	0.480285	-0.492687	-0.555569	-0.125704	0.699610	-0.583287	-0.212682	0.206023	0.794059	1.00000
carb	-0.550925	0.526988	0.394977	0.749812	-0.090790	0.427606	-0.656249	-0.569607	0.057534	0.27407

In [32]:
```
plt.figure(figsize=(8,3))
sns.heatmap(mtcars.loc[:,"mpg":"carb"].corr(), annot=True)
```

Out[32]:
```
<Axes: >
```

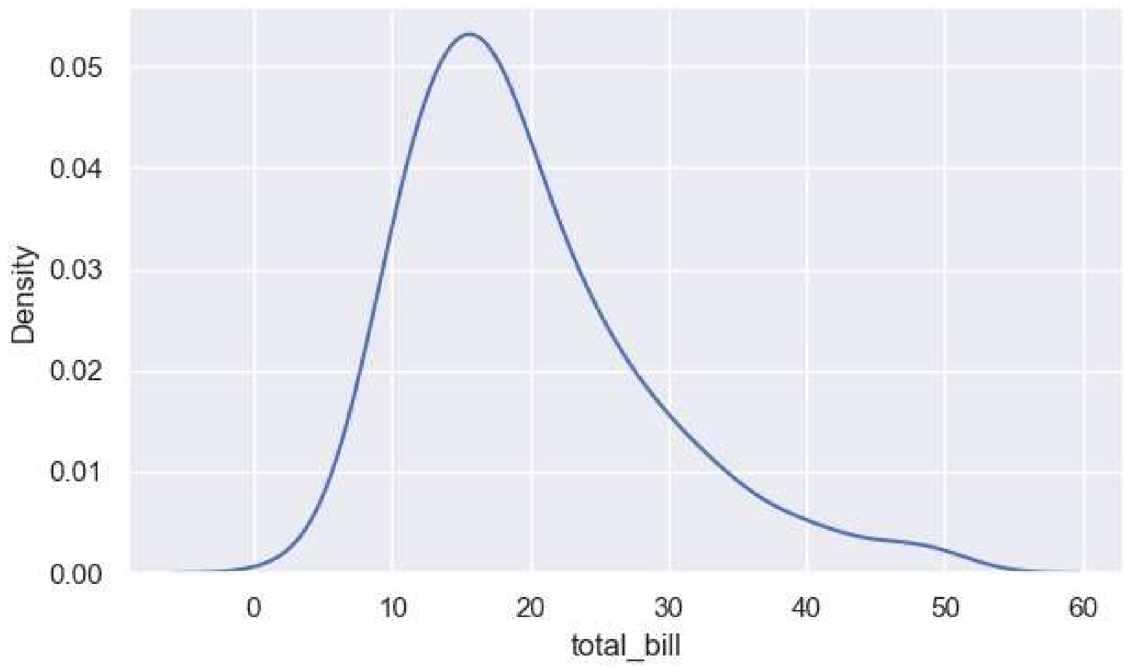


	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
mpg	1	-0.85	-0.85	-0.78	0.68	-0.87	0.42	0.66	0.6	0.48	-0.55
cyl	-0.85	1	0.9	0.83	-0.7	0.78	-0.59	-0.81	-0.52	-0.49	0.53
disp	-0.85	0.9	1	0.79	-0.71	0.89	-0.43	-0.71	-0.59	-0.56	0.39
hp	-0.78	0.83	0.79	1	-0.45	0.66	-0.71	-0.72	-0.24	-0.13	0.75
drat	0.68	-0.7	-0.71	-0.45	1	-0.71	0.091	0.44	0.71	0.7	-0.091
wt	-0.87	0.78	0.89	0.66	-0.71	1	-0.17	-0.55	-0.69	-0.58	0.43
qsec	0.42	-0.59	-0.43	-0.71	0.091	-0.17	1	0.74	-0.23	-0.21	-0.66
vs	0.66	-0.81	-0.71	-0.72	0.44	-0.55	0.74	1	0.17	0.21	-0.57
am	0.6	-0.52	-0.59	-0.24	0.71	-0.69	-0.23	0.17	1	0.79	0.058
gear	0.48	-0.49	-0.56	-0.13	0.7	-0.58	-0.21	0.21	0.79	1	0.27
carb	-0.55	0.53	0.39	0.75	-0.091	0.43	-0.66	-0.57	0.058	0.27	1

KDEPLOT

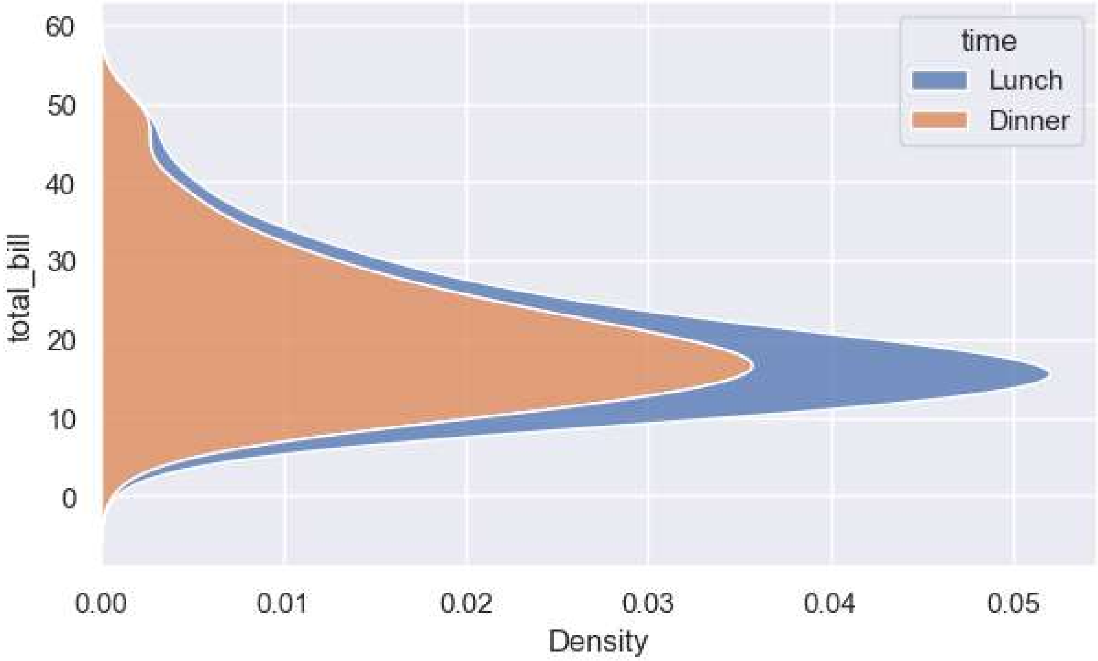
In [33]:
```
sns.kdeplot(data=tips, x="total_bill")
```

Out[33]:
```
<Axes: xlabel='total_bill', ylabel='Density'>
```



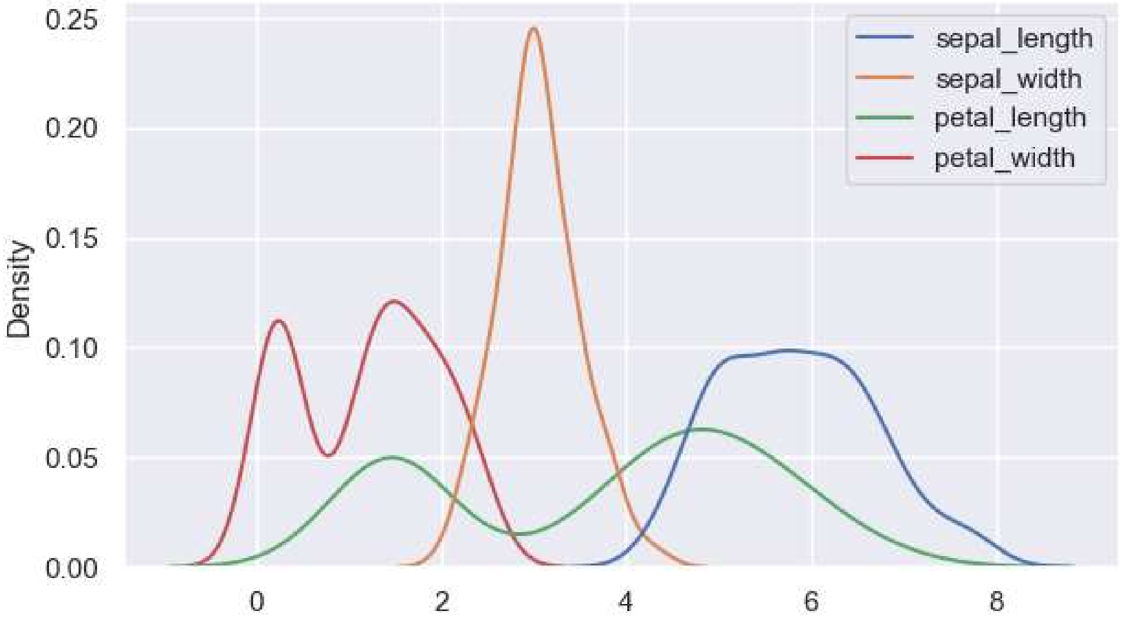
In [34]:
```
sns.kdeplot(data=tips, y="total_bill", hue="time", multiple="stack")
```

Out[34]:
```
<Axes: xlabel='Density', ylabel='total_bill'>
```



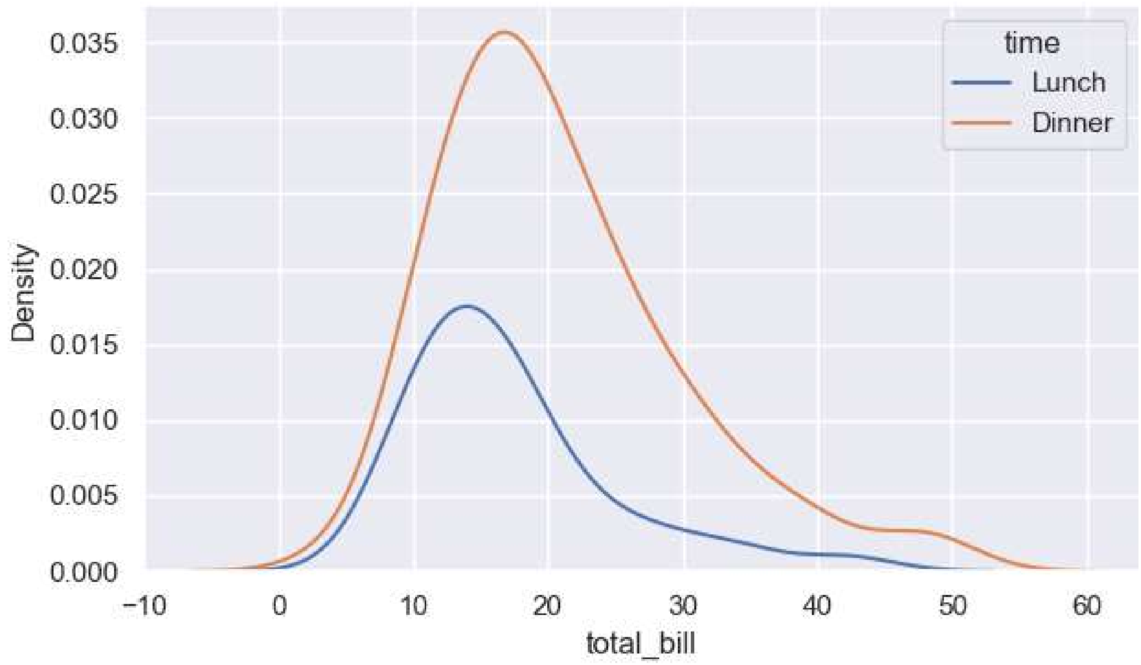
In [35]:
```
sns.kdeplot(data=iris)
```

Out[35]:
```
<Axes: ylabel='Density'>
```



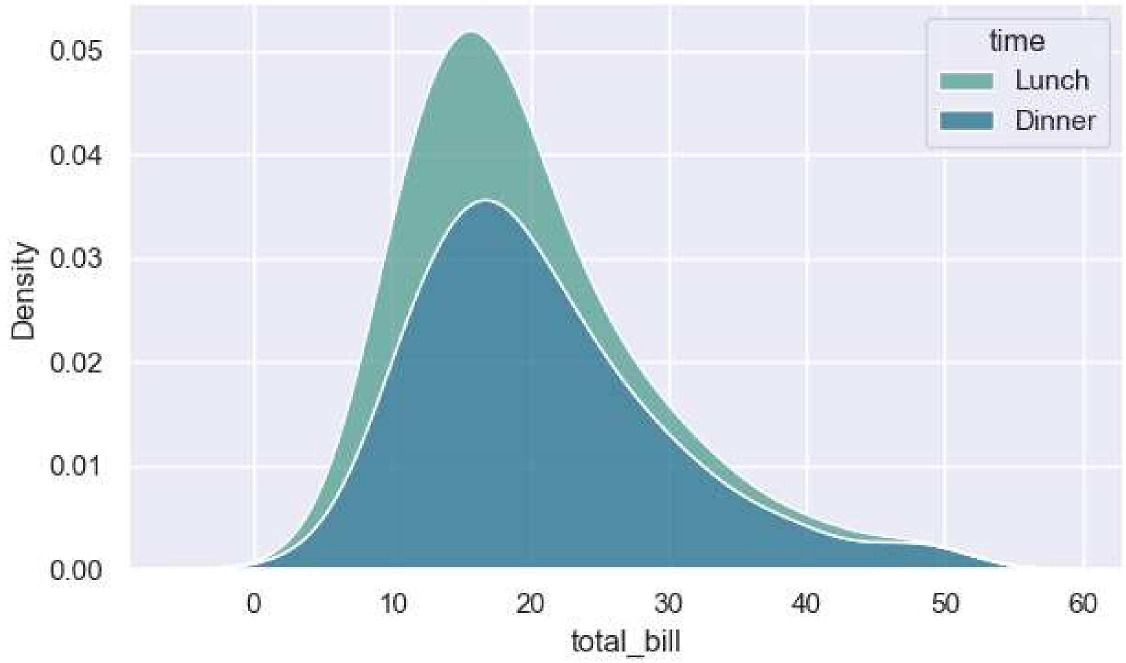
In [36]:
```
sns.kdeplot(data=tips, x="total_bill", hue="time")
```

Out[36]:
```
<Axes: xlabel='total_bill', ylabel='Density'>
```



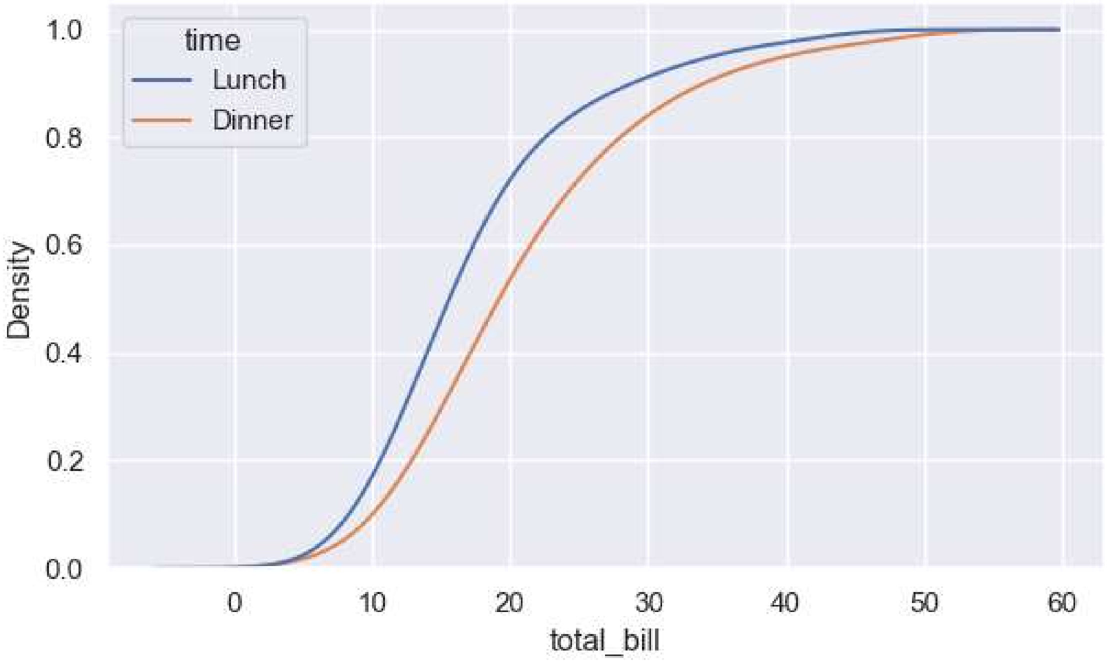
In [37]:
```
sns.kdeplot(data=tips, x="total_bill", hue="time", multiple="stack", palette="crest")
```

Out[37]:
```
<Axes: xlabel='total_bill', ylabel='Density'>
```



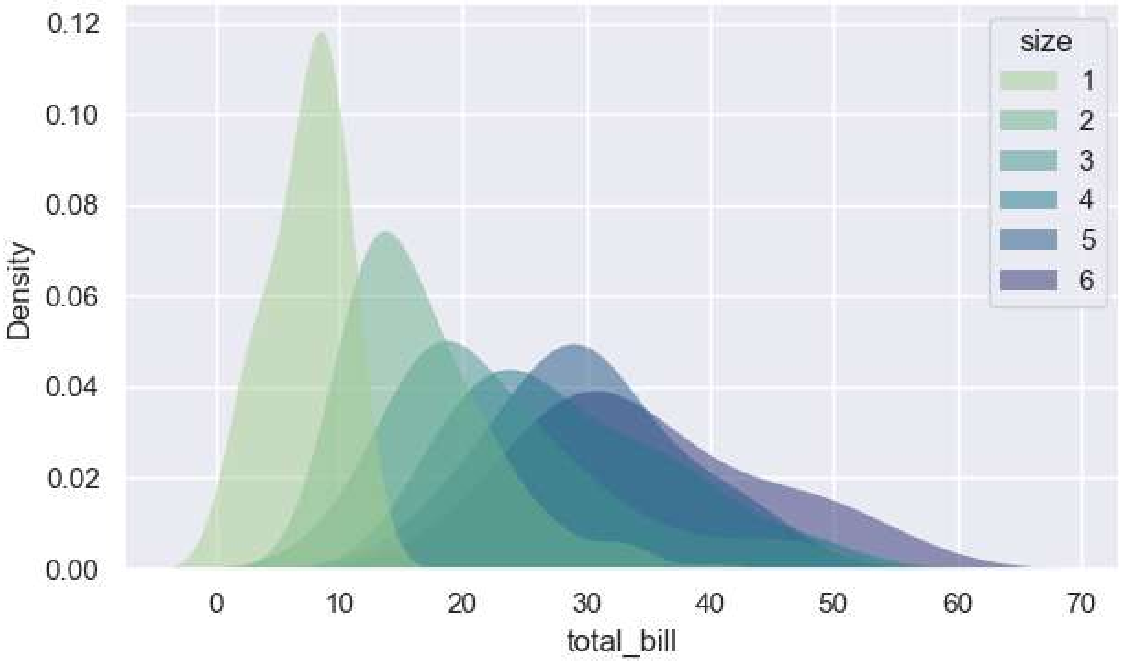
In [38]:
```
sns.kdeplot(
    data=tips, x="total_bill", hue="time",
    cumulative=True, common_norm=False, common_grid=True,
)
```

Out[38]:
```
<Axes: xlabel='total_bill', ylabel='Density'>
```



In [39]:
```
sns.kdeplot(
   data=tips, x="total_bill", hue="size",
   fill=True, common_norm=False, palette="crest",
   alpha=.5, linewidth=0,
)
```

Out[39]:
```
<Axes: xlabel='total_bill', ylabel='Density'>
```



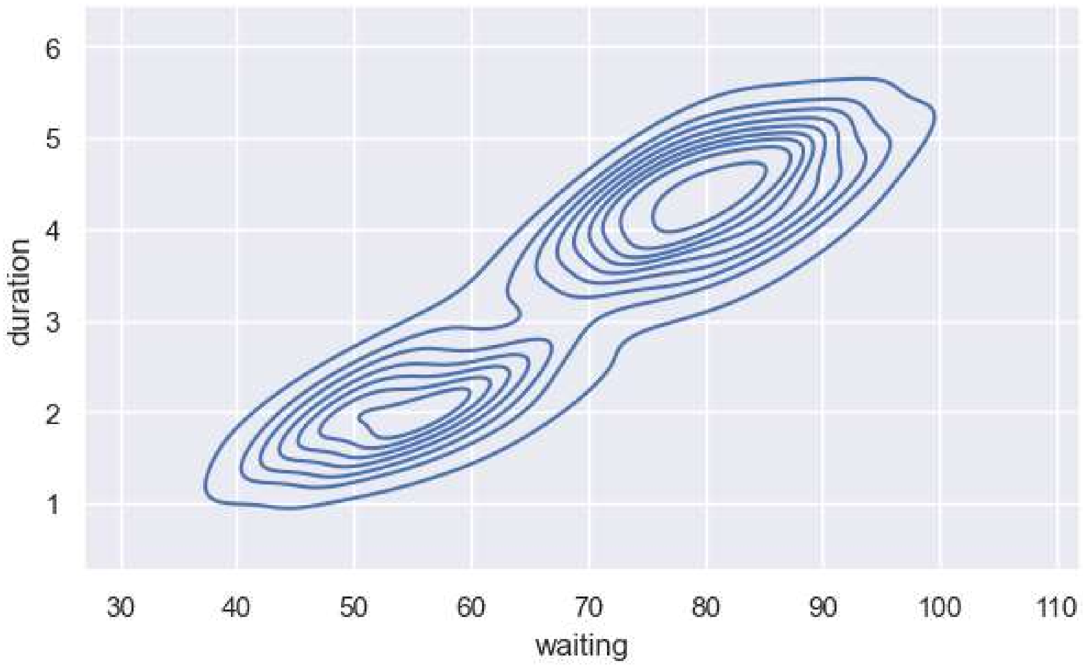
In [40]:
```
geyser = sns.load_dataset("geyser")
geyser.head()
```

Out[40]:

	duration	waiting	kind
0	3.600	79	long
1	1.800	54	short
2	3.333	74	long
3	2.283	62	short
4	4.533	85	long

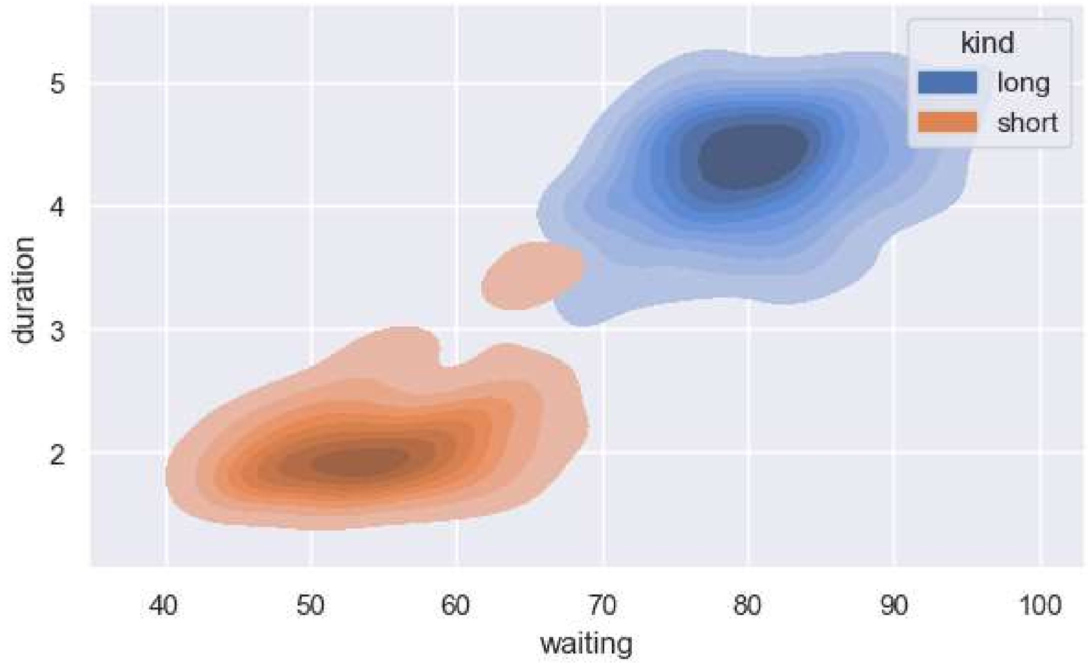
In [41]:
```
sns.kdeplot(data=geyser, x="waiting", y="duration")
```

Out[41]:
```
<Axes: xlabel='waiting', ylabel='duration'>
```



In [42]:
```
sns.kdeplot(
    data=geyser, x="waiting", y="duration", hue="kind", fill=True,
)
```

Out[42]:
```
<Axes: xlabel='waiting', ylabel='duration'>
```

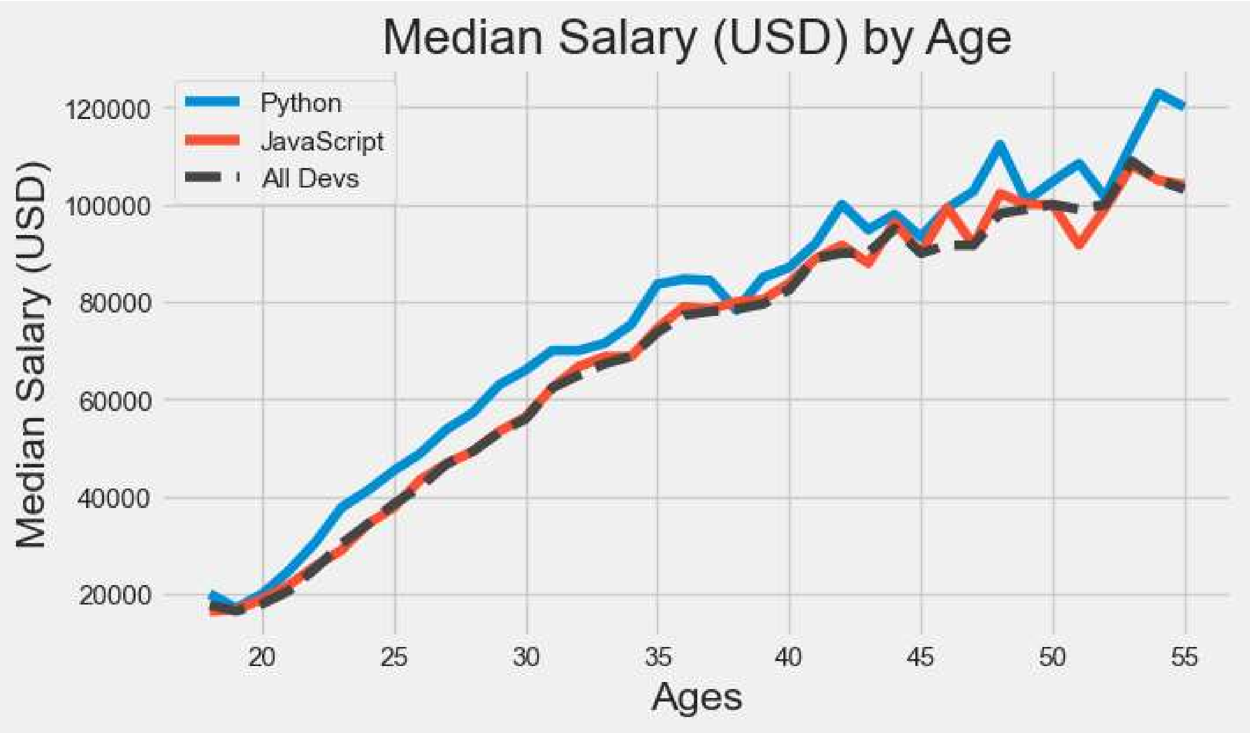


MATPLOTLIB

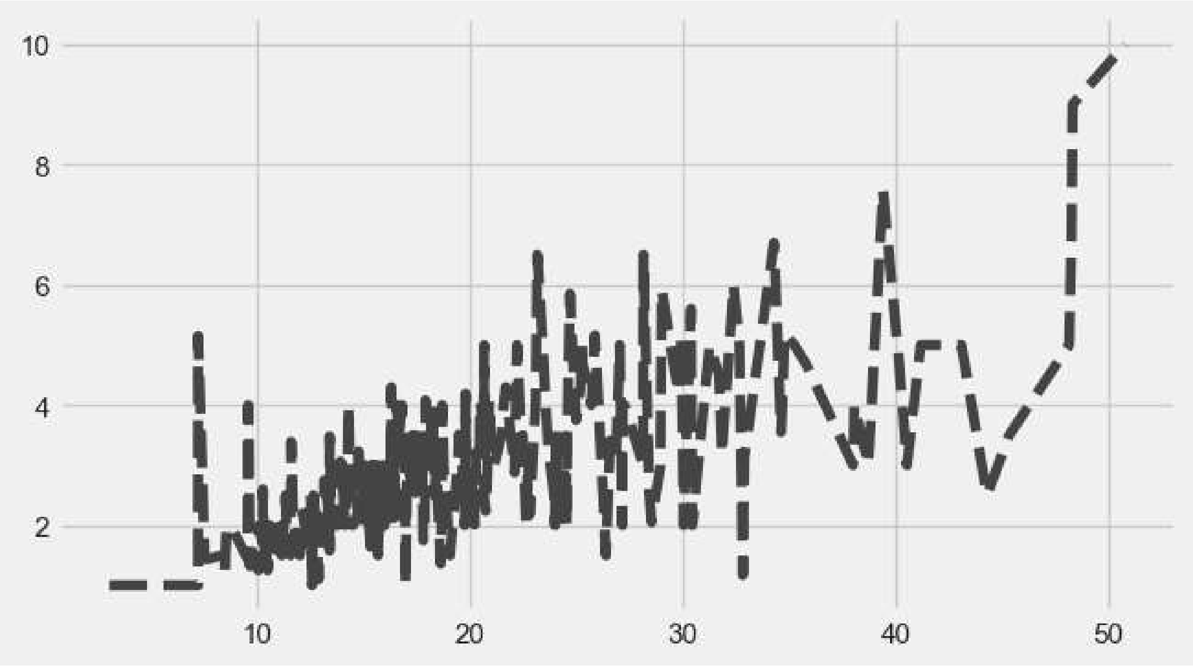
Lineplot

In [43]:

```
from matplotlib import pyplot as plt

# print(plt.style.available)
# plt.xkcd()
plt.style.use("fivethirtyeight")

ages_x = [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
          36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55]

py_dev_y = [20046, 17100, 20000, 24744, 30500, 37732, 41247, 45372, 48876, 53850, 57287, 6
            84392, 78254, 85000, 87038, 91991, 100000, 94796, 97962, 93302, 99240, 102736
plt.plot(ages_x, py_dev_y, label='Python')

js_dev_y = [16446, 16791, 18942, 21780, 25704, 29000, 34372, 37810, 43515, 46823, 49293, 5
            78508, 79996, 80403, 83820, 88833, 91660, 87892, 96243, 90000, 99313, 91660, 1
plt.plot(ages_x, js_dev_y, label='JavaScript')

dev_y = [17784, 16500, 18012, 20628, 25206, 30252, 34368, 38496, 42000, 46752, 49320, 5320
         78000, 78508, 79536, 82488, 88935, 90000, 90056, 95000, 90000, 91633, 91660, 9815
plt.plot(ages_x, dev_y, color='#444444', linestyle='--', label='All Devs')

plt.xlabel('Ages')
plt.ylabel('Median Salary (USD)')
plt.title('Median Salary (USD) by Age')

plt.legend()
# plt.tight_layout()
plt.grid(True)
# plt.savefig('plot.png')

plt.show()
```



In [44]:
```
plt.style.use("fivethirtyeight")
data_lineplot = tips.sort_values(by="total_bill")

plt.plot(list(data_lineplot["total_bill"]), list(data_lineplot["tip"]), color='#444444', 

plt.show()
```


10
8
6
4
2
10
20
30
40
50