data-analysis

August 3, 2023

1 DataFrames intro

[111]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as  sns

states = pd.read_csv("/Datasets/states.csv")
houses = pd.read_csv("/Datasets/kc_house_data.csv")
titanic = pd.read_csv("/Datasets/titanic.csv")
btc = pd.read_csv("/Datasets/coin_Bitcoin.csv")
```

“‘py pd.read_csv(states,names=***,header=***,index_col=***,sep=***)

[5]:
```
type(states)
```

[5]:
```
pandas.core.frame.DataFrame
```

[6]:
```
states.columns
```

[6]:
```
Index(['State', 'Abbrev', 'Code'], dtype='object')
```

[7]:
```
states.shape
```

[7]:
```
(51, 3)
```

[8]:
```
states.size
```

[8]:
```
153
```

[14]:
```
states.head(3)
```

[14]:

	State	Abbrev	Code
0	Alabama	Ala.	AL
1	Alaska	Alaska	AK
2	Arizona	Ariz.	AZ

[15]:
```
states.tail(3)
```

[15]:
```
            State Abbrev Code
48  West Virginia  W.Va.   WV
49      Wisconsin   Wis.   WI
50        Wyoming   Wyo.   WY
```

[16]:
```
states.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   State   51 non-null     object
 1   Abbrev  51 non-null     object
 2   Code    51 non-null     object
dtypes: object(3)
memory usage: 1.3+ KB
```

[19]:
```
states.dtypes
```

[19]:
```
State     object
Abbrev    object
Code      object
dtype: object
```

2 Basic DataFrame Analysis

- sum
- min
- max
- median
- mean
- count
- describe

[41]:
```
states.min()
```

[41]:
```
State     Alabama
Abbrev       Ala.
Code           AK
dtype: object
```

[42]:
```
states.max()
```

[42]:
```
State     Wyoming
Abbrev       Wyo.
Code           WY
```

```
dtype: object
```

[26]:
```
type(states.max())
```

[26]:
```
pandas.core.series.Series
```

[43]:
```
titanic.sum()
```

[43]:
```
pclass                                                      3004
survived                                                     500
name        Allen, Miss. Elisabeth WaltonAllison, Master. …
sex         femalemalefemalemalefemalemalefemalemalefemale…
age         290.916723025486339537147182426807245032363747…
sibsp                                                        653
parch                                                        504
ticket      24160113781113781113781113781199521350211205015…
fare        211.3375151.55151.55151.55151.5526.5577.958305…
cabin       B5C22 C26C22 C26C22 C26C22 C26E12D7A36C101?C62…
embarked    SSSSSSSSSCCCCSSSCCCCSSCCSCCCSSSCSSSCSSSCCCSCCS…
boat        211???310?D??496B??68A55548?778D?788?469???6D8…
body        ???135?????22124?????????????148?????????????…
home.dest   St Louis, MOMontreal, PQ / Chesterville, ONMon…
dtype: object
```

[44]:
```
titanic.sum(numeric_only=True)
```

[44]:
```
pclass      3004
survived     500
sibsp        653
parch        504
dtype: int64
```

[29]:
```
states.count()
```

[29]:
```
State     51
Abbrev    51
Code      51
dtype: int64
```

[45]:
```
titanic.mean(numeric_only=True)
```

[45]:
```
pclass      2.294882
survived    0.381971
sibsp       0.498854
parch       0.385027
dtype: float64
```

[46]:
```
titanic.median(numeric_only=True)
```

[46]:
```
pclass      3.0
survived    0.0
sibsp       0.0
parch       0.0
dtype: float64
```

[48]:
```
titanic.mode(numeric_only=True)
```

[48]:

	pclass	survived	sibsp	parch
0	3	0	0	0

[50]:
```
states.describe()
```

[50]:

	State	Abbrev	Code
count	51	51	51
unique	51	51	51
top	Alabama	Ala.	AL
freq	1	1	1

[52]:
```
states.describe(include=['object'])
```

[52]:

	State	Abbrev	Code
count	51	51	51
unique	51	51	51
top	Alabama	Ala.	AL
freq	1	1	1

3 Columns & Series

[58]:
```
titanic.name
```

[58]:
```
0                          Allen, Miss. Elisabeth Walton
1                         Allison, Master. Hudson Trevor
2                           Allison, Miss. Helen Loraine
3                   Allison, Mr. Hudson Joshua Creighton
4         Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
                                ...
1304                                Zabour, Miss. Hileni
1305                               Zabour, Miss. Thamine
1306                           Zakarian, Mr. Mapriededer
1307                                 Zakarian, Mr. Ortin
1308                                  Zimmerman, Mr. Leo
Name: name, Length: 1309, dtype: object
```

[60]:
```
titanic["name"]
```

[60]:
```
0                          Allen, Miss. Elisabeth Walton
1                        Allison, Master. Hudson Trevor
2                          Allison, Miss. Helen Loraine
3                  Allison, Mr. Hudson Joshua Creighton
4       Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
                               …
1304                              Zabour, Miss. Hileni
1305                             Zabour, Miss. Thamine
1306                         Zakarian, Mr. Mapriededer
1307                               Zakarian, Mr. Ortin
1308                                Zimmerman, Mr. Leo
Name: name, Length: 1309, dtype: object
```

[61]:
```
type(titanic.name)
```

[61]:
```
pandas.core.series.Series
```

[65]:
```
titanic.survived.mean()
```

[65]:
```
0.3819709702062643
```

[66]:
```
titanic.name.values
```

[66]:
```
array(['Allen, Miss. Elisabeth Walton', 'Allison, Master. Hudson Trevor',
       'Allison, Miss. Helen Loraine', …, 'Zakarian, Mr. Mapriededer',
       'Zakarian, Mr. Ortin', 'Zimmerman, Mr. Leo'], dtype=object)
```

[67]:
```
titanic.index
```

[67]:
```
RangeIndex(start=0, stop=1309, step=1)
```

[71]:
```
houses.price.describe()
```

[71]:
```
count    2.161300e+04
mean     5.400881e+05
std      3.671272e+05
min      7.500000e+04
25%      3.219500e+05
50%      4.500000e+05
75%      6.450000e+05
max      7.700000e+06
Name: price, dtype: float64
```

[72]:
```
houses["price"].describe()
```

[72]:
```
count    2.161300e+04
mean     5.400881e+05
std      3.671272e+05
min      7.500000e+04
25%      3.219500e+05
50%      4.500000e+05
75%      6.450000e+05
max      7.700000e+06
Name: price, dtype: float64
```

[74]:
```
houses["bedrooms"].unique()
```

[74]:
```
array([ 3,  2,  4,  5,  1,  6,  7,  0,  8,  9, 11, 10, 33], dtype=int64)
```

[75]:
```
houses["bedrooms"].nunique()
```

[75]:
```
13
```

[76]:
```
houses.bedrooms.nunique(dropna=False)
```

[76]:
```
13
```

[77]:
```
houses.price.nlargest(3)
```

[77]:
```
7252    7700000.0
3914    7062500.0
9254    6885000.0
Name: price, dtype: float64
```

[78]:
```
houses.price.nsmallest(3)
```

[78]:
```
1149     75000.0
15293    78000.0
465      80000.0
Name: price, dtype: float64
```

[79]:
```
houses.nlargest(3,["price"])
```

[79]:

	id	date	price	bedrooms	bathrooms	\
7252	6762700020	20141013T000000	7700000.0	6	8.00	
3914	9808700762	20140611T000000	7062500.0	5	4.50	
9254	9208900037	20140919T000000	6885000.0	6	7.75	

	sqft_living	sqft_lot	floors	waterfront	view	…	grade	sqft_above	\
7252	12050	27600	2.5	0	3	…	13	8570	
3914	10040	37325	2.0	1	2	…	11	7680	
9254	9890	31374	2.0	0	4	…	13	8860	

```
      sqft_basement  yr_built  yr_renovated  zipcode      lat     long  \
7252           3480      1910          1987    98102  47.6298 -122.323
3914           2360      1940          2001    98004  47.6500 -122.214
9254           1030      2001             0    98039  47.6305 -122.240

      sqft_living15  sqft_lot15
7252           3940        8800
3914           3930       25449
9254           4540       42730

[3 rows x 21 columns]
```

[81]:
```
states[["Abbrev","Code"]].tail(3)
```

[81]:
```
   Abbrev Code
48  W.Va.   WV
49   Wis.   WI
50   Wyo.   WY
```

[82]:
```
titanic.sex.value_counts()
```

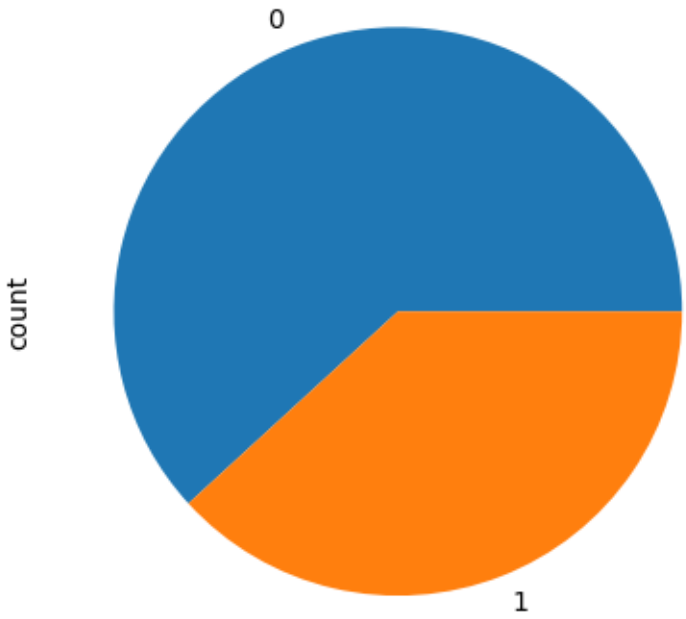
[82]:
```
sex
male      843
female    466
Name: count, dtype: int64
```

[83]:
```
titanic.sex.value_counts(ascending=True)
```

[83]:
```
sex
female    466
male      843
Name: count, dtype: int64
```

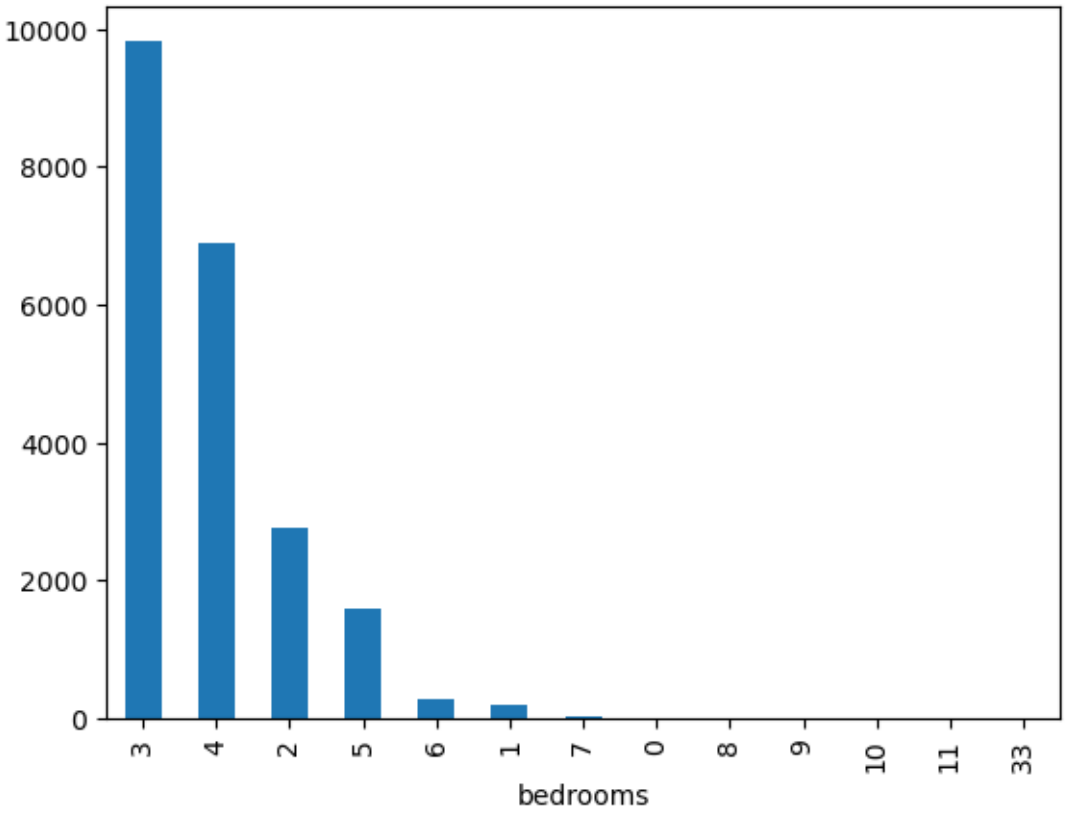
[11]:
```
titanic.survived.value_counts().plot(kind='pie')
```

[11]:
```
<Axes: ylabel='count'>
```



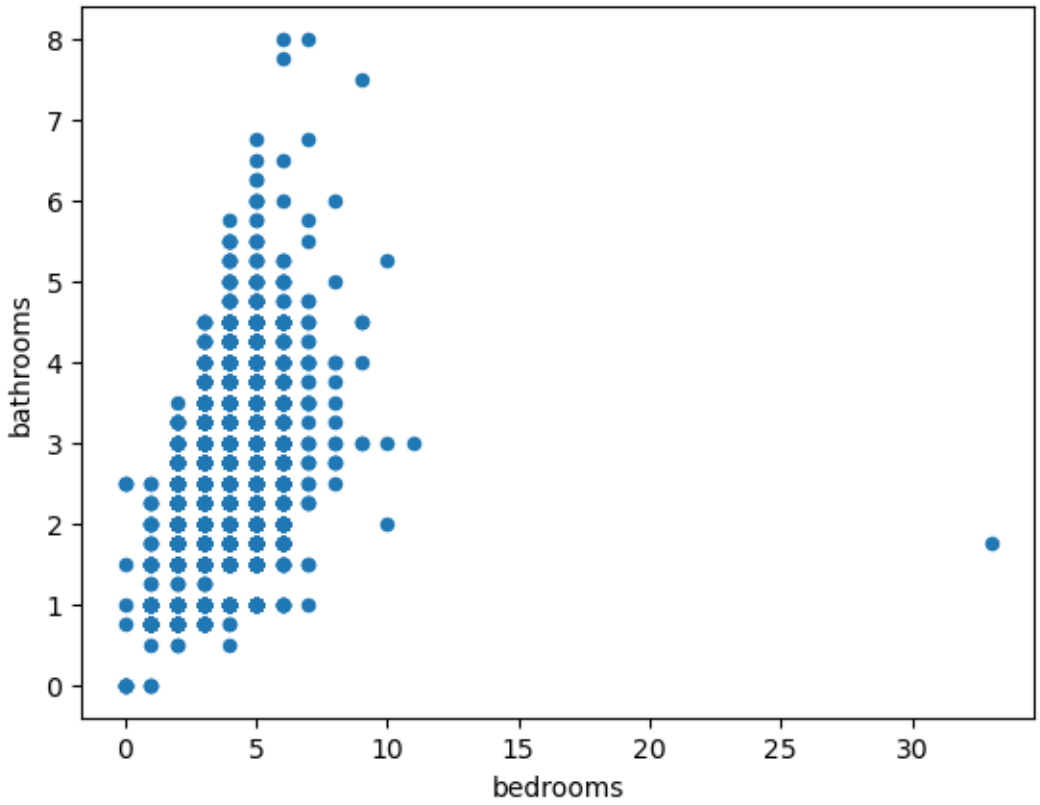
[8]:
```
houses.bedrooms.value_counts().plot(kind='bar')
```

[8]: <Axes: xlabel='bedrooms'>



[13]:
```
houses.plot(kind="scatter", x="bedrooms", y="bathrooms")
```

[13]:
```
<Axes: xlabel='bedrooms', ylabel='bathrooms'>
```



4 Indexes & Sorting

[3]:
```
states.head()
```

[3]:
```
        State  Abbrev Code
0     Alabama    Ala.   AL
1      Alaska  Alaska   AK
2     Arizona   Ariz.   AZ
3    Arkansas    Ark.   AR
4  California  Calif.   CA
```

[12]:
```
btc.head(3)
```

[12]:
```
   SNo     Name Symbol                 Date        High         Low      Open  \
0    1  Bitcoin    BTC  2013-04-29 23:59:59  147.488007  134.000000  134.444
1    2  Bitcoin    BTC  2013-04-30 23:59:59  146.929993  134.050003  144.000
2    3  Bitcoin    BTC  2013-05-01 23:59:59  139.889999  107.720001  139.000

        Close  Volume     Marketcap
0  144.539993     0.0  1.603769e+09
```

```
1  139.000000     0.0  1.542813e+09
2  116.989998     0.0  1.298955e+09
```

[13]:
```
btc.set_index("Date",inplace=True)
```

[14]:
```
btc.head(3)
```

[14]:
```
                      SNo     Name Symbol        High         Low     Open  \
Date
2013-04-29 23:59:59    1  Bitcoin    BTC  147.488007  134.000000  134.444
2013-04-30 23:59:59    2  Bitcoin    BTC  146.929993  134.050003  144.000
2013-05-01 23:59:59    3  Bitcoin    BTC  139.889999  107.720001  139.000

                           Close  Volume     Marketcap
Date
2013-04-29 23:59:59  144.539993     0.0  1.603769e+09
2013-04-30 23:59:59  139.000000     0.0  1.542813e+09
2013-05-01 23:59:59  116.989998     0.0  1.298955e+09
```

[15]:
```
btc.index
```

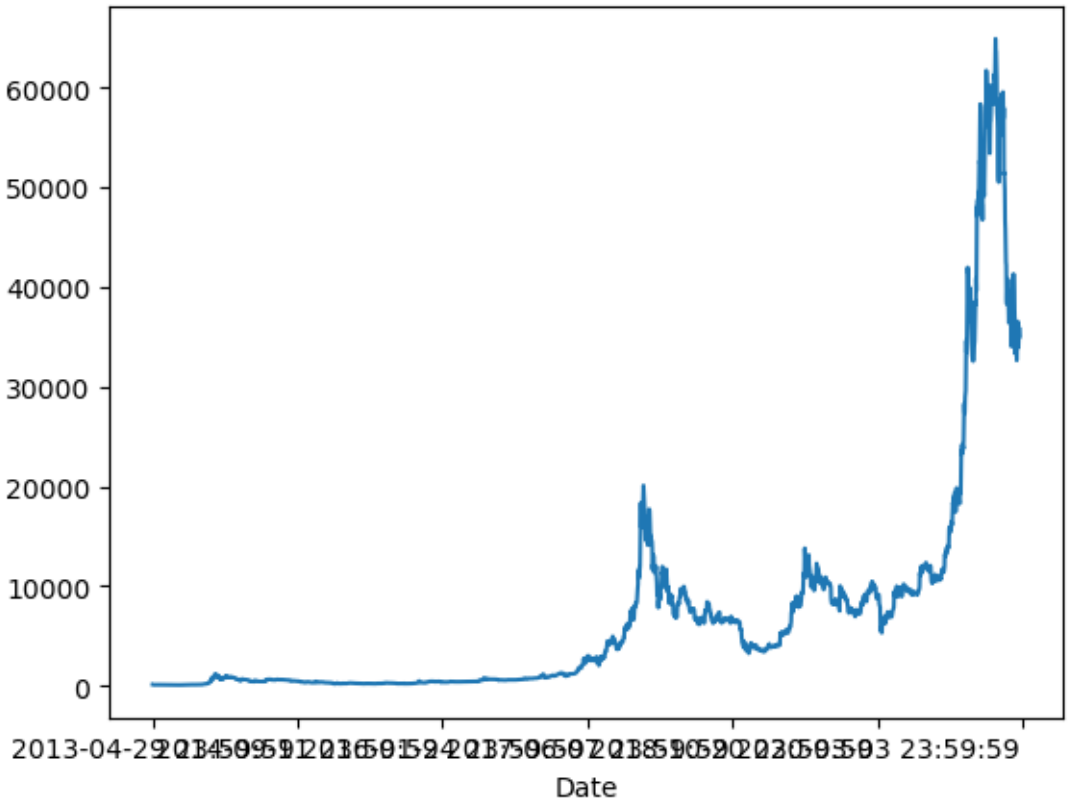
[15]:
```
Index(['2013-04-29 23:59:59', '2013-04-30 23:59:59', '2013-05-01 23:59:59',
       '2013-05-02 23:59:59', '2013-05-03 23:59:59', '2013-05-04 23:59:59',
       '2013-05-05 23:59:59', '2013-05-06 23:59:59', '2013-05-07 23:59:59',
       '2013-05-08 23:59:59',
       …
       '2021-06-27 23:59:59', '2021-06-28 23:59:59', '2021-06-29 23:59:59',
       '2021-06-30 23:59:59', '2021-07-01 23:59:59', '2021-07-02 23:59:59',
       '2021-07-03 23:59:59', '2021-07-04 23:59:59', '2021-07-05 23:59:59',
       '2021-07-06 23:59:59'],
      dtype='object', name='Date', length=2991)
```

[25]:
```
btc["High"].head(3)
```

[25]:
```
Date
2013-04-29 23:59:59    147.488007
2013-04-30 23:59:59    146.929993
2013-05-01 23:59:59    139.889999
Name: High, dtype: float64
```

[27]:
```
btc.High.plot()
```

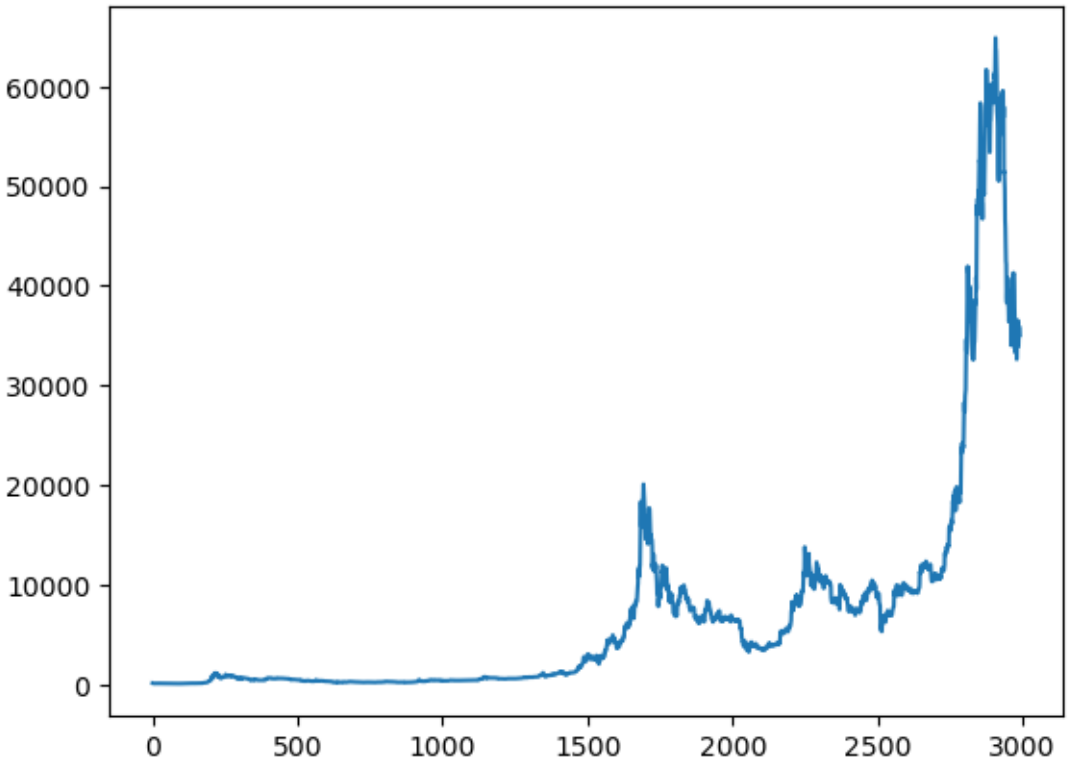
[27]:
```
<Axes: xlabel='Date'>
```



[30]:
```
btc = pd.read_csv("/Datasets/coin_Bitcoin.csv")
```

[31]:
```
btc.High.plot()
```

[31]: <Axes: >



[35]:
```
houses.sort_values("price",ascending=False).head(3)
```

[35]:

	id	date	price	bedrooms	bathrooms	\
7252	6762700020	20141013T000000	7700000.0	6	8.00	
3914	9808700762	20140611T000000	7062500.0	5	4.50	
9254	9208900037	20140919T000000	6885000.0	6	7.75	

	sqft_living	sqft_lot	floors	waterfront	view	…	grade	sqft_above	\
7252	12050	27600	2.5	0	3	…	13	8570	
3914	10040	37325	2.0	1	2	…	11	7680	
9254	9890	31374	2.0	0	4	…	13	8860	

	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	\
7252	3480	1910	1987	98102	47.6298	-122.323	
3914	2360	1940	2001	98004	47.6500	-122.214	
9254	1030	2001	0	98039	47.6305	-122.240	

	sqft_living15	sqft_lot15
7252	3940	8800
3914	3930	25449
9254	4540	42730

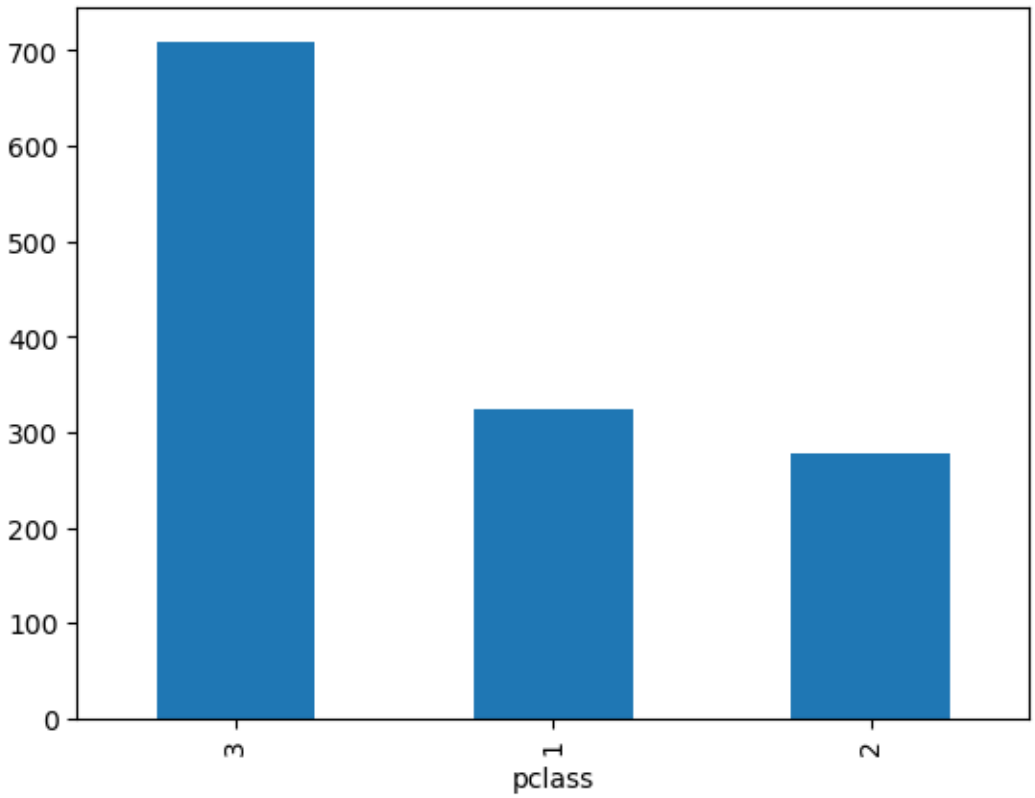
```
[3 rows x 21 columns]
```

[37]:
```
btc.sort_index().head(3)
```

[37]:
```
   SNo     Name Symbol                 Date        High         Low     Open  \
0    1  Bitcoin    BTC  2013-04-29 23:59:59  147.488007  134.000000  134.444
1    2  Bitcoin    BTC  2013-04-30 23:59:59  146.929993  134.050003  144.000
2    3  Bitcoin    BTC  2013-05-01 23:59:59  139.889999  107.720001  139.000

        Close  Volume     Marketcap
0  144.539993     0.0  1.603769e+09
1  139.000000     0.0  1.542813e+09
2  116.989998     0.0  1.298955e+09
```

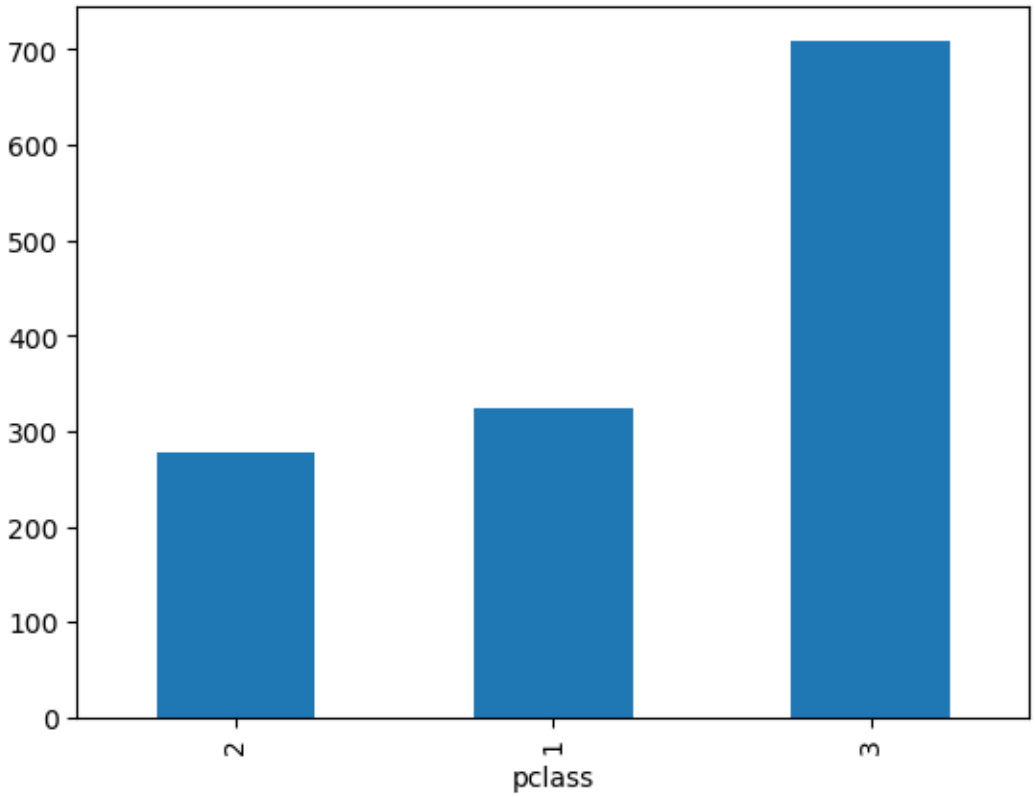
[38]:
```
titanic.pclass.value_counts().plot(kind="bar")
```

[38]:
```
<Axes: xlabel='pclass'>
```



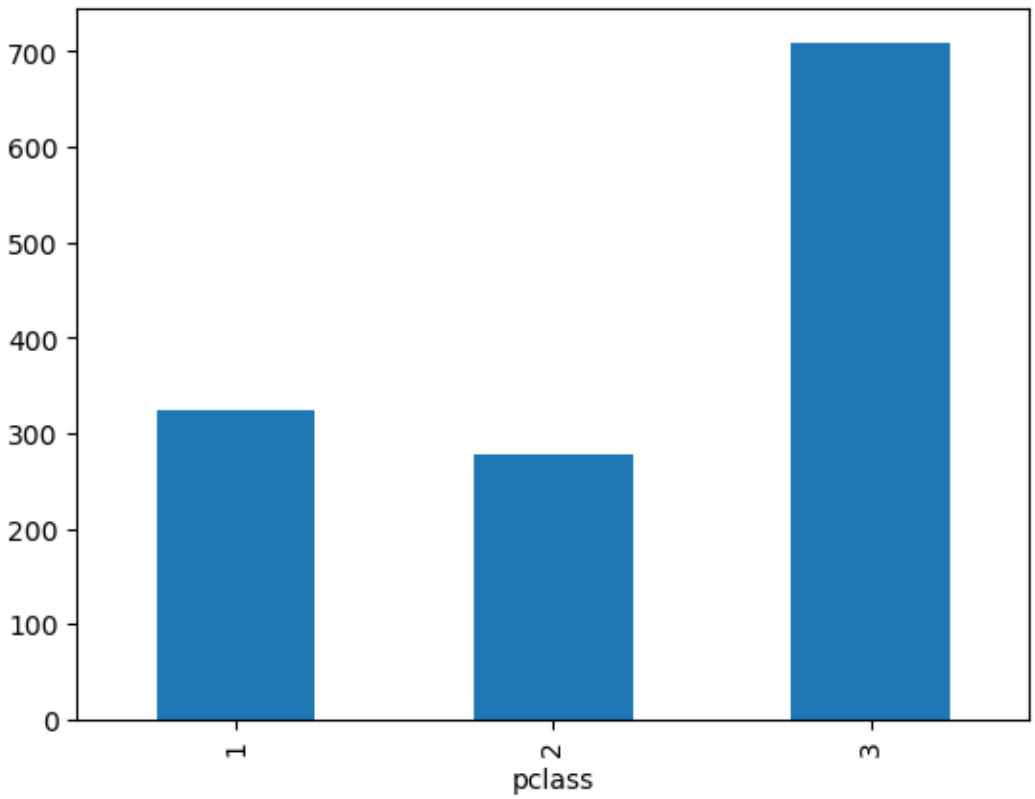
[40]:
```
titanic.pclass.value_counts().sort_values().plot(kind="bar")
```

[40]:
```
<Axes: xlabel='pclass'>
```



[41]:
```
titanic.pclass.value_counts().sort_index().plot(kind="bar")
```

[41]: <Axes: xlabel='pclass'>



[56]:
```
titanic.loc[0]
```

[56]:
```
pclass                                  1
survived                                1
name         Allen, Miss. Elisabeth Walton
sex                                 female
age                                     29
sibsp                                    0
parch                                    0
ticket                               24160
fare                                211.3375
cabin                                   B5
embarked                                 S
boat                                     2
body                                     ?
home.dest                     St Louis, MO
Name: 0, dtype: object
```

[55]:
```
titanic.loc[[0]]
```

[55]:
```
   pclass  survived                            name     sex age  sibsp  parch  \
0       1         1  Allen, Miss. Elisabeth Walton  female  29      0      0

  ticket      fare cabin embarked boat body     home.dest
0  24160  211.3375    B5        S    2    ?  St Louis, MO
```

[52]:
```
titanic.loc[0:1]
```

[52]:
```
   pclass  survived                              name     sex     age  sibsp  \
0       1         1   Allen, Miss. Elisabeth Walton  female      29      0
1       1         1  Allison, Master. Hudson Trevor    male  0.9167      1

   parch  ticket      fare    cabin embarked boat body  \
0      0   24160  211.3375       B5        S    2    ?
1      2  113781    151.55  C22 C26        S   11    ?

                        home.dest
0                    St Louis, MO
1  Montreal, PQ / Chesterville, ON
```

[53]:
```
titanic.iloc[[0]]
```

[53]:
```
   pclass  survived                            name     sex age  sibsp  parch  \
0       1         1  Allen, Miss. Elisabeth Walton  female  29      0      0

  ticket      fare cabin embarked boat body     home.dest
0  24160  211.3375    B5        S    2    ?  St Louis, MO
```

[59]:
```
titanic.loc[50:60:2, ['name', 'sex', 'age']]
```

[59]:
```
                                                 name     sex age
50  Cardeza, Mrs. James Warburton Martinez (Charlo…  female  58
52                            Carrau, Mr. Francisco M    male  28
54                Carter, Master. William Thornton II    male  11
56                         Carter, Mr. William Ernest    male  36
58                             Case, Mr. Howard Brown    male  49
60                      Cavendish, Mr. Tyrell William    male  36
```

[60]:
```
titanic["age"].value_counts().loc["18"]
```

[60]:
```
39
```

5 Filtering Data

[65]:
```
titanic.sex=='female'
```

```
[65]: 0        True
      1       False
      2        True
      3       False
      4        True
             …
      1304     True
      1305     True
      1306    False
      1307    False
      1308    False
      Name: sex, Length: 1309, dtype: bool
```

```
[63]: titanic[titanic.sex=='female'].head(3)
```

```
[63]:    pclass  survived                                             name     sex  \
      0       1         1                    Allen, Miss. Elisabeth Walton  female
      2       1         0                     Allison, Miss. Helen Loraine  female
      4       1         0  Allison, Mrs. Hudson J C (Bessie Waldo Daniels)  female

        age  sibsp  parch  ticket      fare    cabin embarked boat body  \
      0  29      0      0   24160  211.3375       B5        S    2    ?
      2   2      1      2  113781    151.55  C22 C26        S    ?    ?
      4  25      1      2  113781    151.55  C22 C26        S    ?    ?

                           home.dest
      0                    St Louis, MO
      2  Montreal, PQ / Chesterville, ON
      4  Montreal, PQ / Chesterville, ON
```

```
[66]: titanic[titanic.survived == 1].head(3)
```

```
[66]:    pclass  survived                             name     sex      age  sibsp  \
      0       1         1   Allen, Miss. Elisabeth Walton  female       29      0
      1       1         1  Allison, Master. Hudson Trevor    male   0.9167      1
      5       1         1              Anderson, Mr. Harry    male       48      0

         parch  ticket      fare    cabin embarked boat body  \
      0      0   24160  211.3375       B5        S    2    ?
      1      2  113781    151.55  C22 C26        S   11    ?
      5      0   19952     26.55      E12        S    3    ?

                           home.dest
      0                    St Louis, MO
      1  Montreal, PQ / Chesterville, ON
      5                    New York, NY
```

[67]:
```
titanic[titanic.pclass != 1].head(3)
```

[67]:
```
     pclass  survived                                     name     sex age  \
323       2         0                    Abelson, Mr. Samuel    male  30
324       2         1  Abelson, Mrs. Samuel (Hannah Wizosky)  female  28
325       2         0         Aldworth, Mr. Charles Augustus    male  30

     sibsp  parch     ticket fare cabin embarked boat body  \
323      1      0  P/PP 3381   24     ?        C    ?    ?
324      1      0  P/PP 3381   24     ?        C   10    ?
325      0      0     248744   13     ?        S    ?    ?

               home.dest
323  Russia New York, NY
324  Russia New York, NY
325   Bryn Mawr, PA, USA
```

[68]:
```
houses[houses["price"] > 5000000].head(3)
```

[68]:
```
              id             date      price  bedrooms  bathrooms  \
1164  1247600105  20141020T000000  5110800.0         5       5.25
1315  7558700030  20150413T000000  5300000.0         6       6.00
1448  8907500070  20150413T000000  5350000.0         5       5.00

      sqft_living  sqft_lot  floors  waterfront  view  …  grade  sqft_above  \
1164         8010     45517     2.0           1     4  …     12        5990
1315         7390     24829     2.0           1     4  …     12        5000
1448         8000     23985     2.0           0     4  …     12        6720

      sqft_basement  yr_built  yr_renovated  zipcode      lat     long  \
1164           2020      1999             0    98033  47.6767 -122.211
1315           2390      1991             0    98040  47.5631 -122.210
1448           1280      2009             0    98004  47.6232 -122.220

      sqft_living15  sqft_lot15
1164           3430       26788
1315           4320       24619
1448           4600       21750

[3 rows x 21 columns]
```

[69]:
```
houses[houses["bedrooms"].between(5, 7)].head(3)
```

[69]:
```
             id             date     price  bedrooms  bathrooms  sqft_living  \
14   1175000570  20150312T000000  530000.0         5       2.00         1810
22   7137970340  20140703T000000  285000.0         5       2.50         2270
42   7203220400  20140707T000000  861990.0         5       2.75         3595
```

```
    sqft_lot  floors  waterfront  view  …  grade  sqft_above  sqft_basement  \
14      4850     1.5           0     0  …      7        1810              0
22      6300     2.0           0     0  …      8        2270              0
42      5639     2.0           0     0  …      9        3595              0

    yr_built  yr_renovated  zipcode      lat     long  sqft_living15  \
14      1900             0    98107  47.6700 -122.394           1360
22      1995             0    98092  47.3266 -122.169           2240
42      2014             0    98053  47.6848 -122.016           3625

    sqft_lot15
14        4850
22        7005
42        5639

[3 rows x 21 columns]
```

[72]:
```
houses[houses["bedrooms"].isin([1,2])].head(3)
```

[72]:
```
            id             date     price  bedrooms  bathrooms  sqft_living  \
2   5631500400  20150225T000000  180000.0         2        1.0          770
11  9212900260  20140527T000000  468000.0         2        1.0         1160
18    16000397  20141205T000000  189000.0         2        1.0         1200

    sqft_lot  floors  waterfront  view  …  grade  sqft_above  sqft_basement  \
2      10000     1.0           0     0  …      6         770              0
11      6000     1.0           0     0  …      7         860            300
18      9850     1.0           0     0  …      7        1200              0

    yr_built  yr_renovated  zipcode      lat     long  sqft_living15  \
2       1933             0    98028  47.7379 -122.233           2720
11      1942             0    98115  47.6900 -122.292           1330
18      1921             0    98002  47.3089 -122.210           1060

    sqft_lot15
2         8062
11        6000
18        5095

[3 rows x 21 columns]
```

[79]:
```
women = titanic.sex == 'female'
died = titanic.survived == 0
titanic[women & died].head(3)
```

[79]:
```
     pclass  survived                                             name  \
2         1         0                        Allison, Miss. Helen Loraine
4         1         0      Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
105       1         0                           Evans, Miss. Edith Corse

        sex age  sibsp  parch    ticket     fare    cabin embarked boat body  \
2    female   2      1      2    113781   151.55  C22 C26        S    ?    ?
4    female  25      1      2    113781   151.55  C22 C26        S    ?    ?
105  female  36      0      0  PC 17531  31.6792      A29        C    ?    ?

                          home.dest
2    Montreal, PQ / Chesterville, ON
4    Montreal, PQ / Chesterville, ON
105                  New York, NY
```

[81]:
```
titanic[women | died].head(3)
```

[81]:
```
   pclass  survived                                     name     sex age  sibsp  \
0       1         1                  Allen, Miss. Elisabeth Walton  female  29      0
2       1         0                   Allison, Miss. Helen Loraine  female   2      1
3       1         0  Allison, Mr. Hudson Joshua Creighton    male  30      1

   parch  ticket      fare    cabin embarked boat body  \
0      0   24160  211.3375       B5        S    2    ?
2      2  113781    151.55  C22 C26        S    ?    ?
3      2  113781    151.55  C22 C26        S    ?  135

                        home.dest
0                     St Louis, MO
2  Montreal, PQ / Chesterville, ON
3  Montreal, PQ / Chesterville, ON
```

[80]:
```
houses[(houses["waterfront"] == 1) & (houses["price"] < 500000)].head(3)
```

[80]:
```
              id             date     price  bedrooms  bathrooms  sqft_living  \
264   2123039032  20141027T000000  369900.0         1       0.75          760
1168  3523029041  20141009T000000  290000.0         2       0.75          440
1949  1922039062  20150420T000000  480000.0         2       1.50         1008

      sqft_lot  floors  waterfront  view  …  grade  sqft_above  \
264      10079     1.0           1     4  …      5         760
1168      8313     1.0           1     3  …      5         440
1949     26487     1.0           1     4  …      6        1008

      sqft_basement  yr_built  yr_renovated  zipcode      lat     long  \
264               0      1936             0    98070  47.4683 -122.438
1168              0      1943             0    98070  47.4339 -122.512
```

```
1949              0      1943          2002    98070  47.3853 -122.479

      sqft_living15  sqft_lot15
264            1230       14267
1168            880       26289
1949           1132       24079

[3 rows x 21 columns]
```

[82]:
```
women = titanic.sex == 'female'
titanic[~women].head(3)
```

[82]:
```
   pclass  survived                                     name   sex     age  \
1       1         1          Allison, Master. Hudson Trevor  male  0.9167
3       1         0  Allison, Mr. Hudson Joshua Creighton  male      30
5       1         1                    Anderson, Mr. Harry  male      48

   sibsp  parch  ticket    fare    cabin embarked boat body  \
1      1      2  113781  151.55  C22 C26        S   11    ?
3      1      2  113781  151.55  C22 C26        S    ?  135
5      0      0   19952   26.55      E12        S    3    ?

                         home.dest
1  Montreal, PQ / Chesterville, ON
3  Montreal, PQ / Chesterville, ON
5                  New York, NY
```

[83]:
```
titanic[titanic.sex.isna()]
```

[83]:
```
Empty DataFrame
Columns: [pclass, survived, name, sex, age, sibsp, parch, ticket, fare, cabin,
embarked, boat, body, home.dest]
Index: []
```

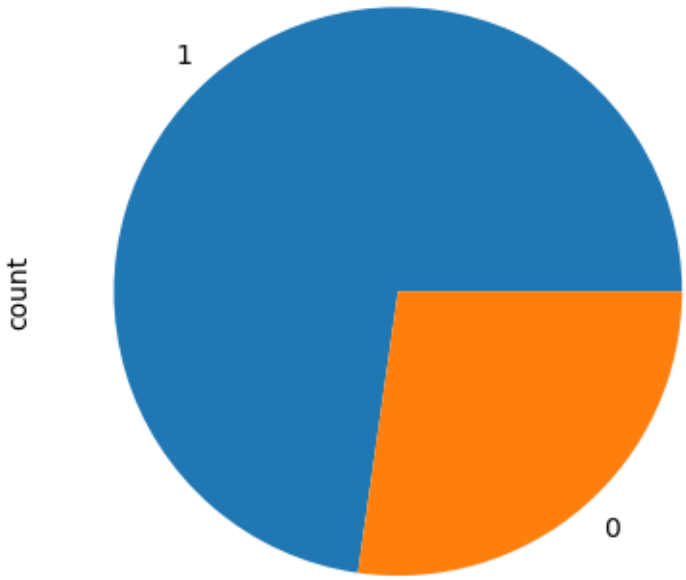
[86]:
```
titanic[titanic.sex.notna()].head(3)
```

[86]:
```
   pclass  survived                                name     sex     age  sibsp  \
0       1         1   Allen, Miss. Elisabeth Walton  female      29      0
1       1         1  Allison, Master. Hudson Trevor    male  0.9167      1
2       1         0    Allison, Miss. Helen Loraine  female       2      1

   parch  ticket      fare    cabin embarked boat body  \
0      0   24160  211.3375       B5        S    2    ?
1      2  113781    151.55  C22 C26        S   11    ?
2      2  113781    151.55  C22 C26        S    ?    ?

                         home.dest
```

```
0                     St Louis, MO
1  Montreal, PQ / Chesterville, ON
2  Montreal, PQ / Chesterville, ON
```

[87]:
```
titanic[women].survived.value_counts().plot(kind="pie")
```

[87]:
```
<Axes: ylabel='count'>
```



6 Modifying dataframes

[10]:
```
states.drop(columns='State').head(2)
```

[10]:
```
   Abbrev Code
0    Ala.   AL
1  Alaska   AK
```

[11]:
```
states.head(2)
```

[11]:
```
      State  Abbrev Code
0  Alabama    Ala.   AL
1   Alaska  Alaska   AK
```

[13]:
```
states[['Abbrev','State']].head(2)
```

[13]:
```
    Abbrev    State
0     Ala.  Alabama
1  Alaska   Alaska
```

[14]:
```
states.drop(index=0).head(2)
```

[14]:
```
      State  Abbrev Code
1   Alaska  Alaska   AK
2  Arizona   Ariz.   AZ
```

[16]:
```
states.head(2)
```

[16]:
```
      State  Abbrev Code
0  Alabama    Ala.   AL
1   Alaska  Alaska   AK
```

[21]:
```
states.insert(0,'my column name','hi')
```

[19]:
```
states.head(2)
```

[19]:
```
  my column name    State  Abbrev Code
0             hi  Alabama    Ala.   AL
1             hi   Alaska  Alaska   AK
```

[22]:
```
titanic["num_relatives"] = titanic["sibsp"] + titanic["parch"]
```

[23]:
```
titanic.head(2)
```

[23]:
```
   pclass  survived                             name     sex     age  sibsp  \
0       1         1    Allen, Miss. Elisabeth Walton  female      29      0
1       1         1   Allison, Master. Hudson Trevor    male  0.9167      1

   parch  ticket      fare    cabin embarked boat body  \
0      0   24160  211.3375       B5        S    2    ?
1      2  113781    151.55  C22 C26        S   11    ?

                        home.dest  num_relatives
0                    St Louis, MO              0
1  Montreal, PQ / Chesterville, ON              3
```

[24]:
```
solo_passengers = titanic["num_relatives"] == 0
titanic[solo_passengers].head(2)
```

[24]:
```
   pclass  survived                             name     sex age  sibsp  parch  \
0       1         1    Allen, Miss. Elisabeth Walton  female  29      0      0
```

```
5       1         1            Anderson, Mr. Harry    male  48      0      0

  ticket      fare cabin embarked boat body     home.dest  num_relatives
0  24160  211.3375    B5        S    2    ?  St Louis, MO              0
5  19952     26.55   E12        S    3    ?  New York, NY              0
```

7 Updating DataFrame Values

[25]:
```
states.rename(columns={'State':"States"}).head(2)
```

[25]:
```
  my column name   States  Abbrev Code
0             hi  Alabama    Ala.   AL
1             hi   Alaska  Alaska   AK
```

[26]:
```
titanic["sex"].replace(["female", "male"], ["F", "M"],inplace=True)
```

[27]:
```
titanic.head(2)
```

[27]:
```
   pclass  survived                             name sex      age  sibsp  parch  \
0       1         1    Allen, Miss. Elisabeth Walton   F       29      0      0
1       1         1   Allison, Master. Hudson Trevor   M   0.9167      1      2

   ticket      fare    cabin embarked boat body  \
0   24160  211.3375       B5        S    2    ?
1  113781    151.55  C22 C26        S   11    ?

                        home.dest  num_relatives
0                     St Louis, MO              0
1  Montreal, PQ / Chesterville, ON              3
```

[35]:
```
states.loc[0,['my column name']] = 'bye'
states.head(2)
```

[35]:
```
  my column name    State  Abbrev Code
0            bye  Alabama    Ala.   AL
1             hi   Alaska  Alaska   AK
```

8 Working With Types

[36]:
```
titanic["age"].replace(['?'], [None], inplace=True)
```

[37]:
```
titanic.age.value_counts(dropna=False)
```

[37]:
```
age
None      263
```

```
24          47
22          43
21          41
30          40
          …
66           1
0.6667       1
76           1
67           1
26.5         1
Name: count, Length: 99, dtype: int64
```

[38]:
```
titanic["age"].astype("float")
```

[38]:
```
0       29.0000
1        0.9167
2        2.0000
3       30.0000
4       25.0000
          …
1304    14.5000
1305        NaN
1306    26.5000
1307    27.0000
1308    29.0000
Name: age, Length: 1309, dtype: float64
```

[39]:
```
titanic["age_float"] = titanic["age"].astype("float")
titanic.head(2)
```

[39]:
```
   pclass  survived                             name sex     age  sibsp  parch  \
0       1         1    Allen, Miss. Elisabeth Walton   F      29      0      0
1       1         1   Allison, Master. Hudson Trevor   M  0.9167      1      2

   ticket      fare    cabin embarked boat body  \
0   24160  211.3375       B5        S    2    ?
1  113781    151.55  C22 C26        S   11    ?

                        home.dest  num_relatives  age_float
0                    St Louis, MO              0    29.0000
1  Montreal, PQ / Chesterville, ON              3     0.9167
```

[40]:
```
titanic["sex"] = titanic["sex"].astype("category")
```

[41]:
```
titanic.head(2)
```

[41]:
```
   pclass  survived                             name sex      age  sibsp  parch  \
0       1         1    Allen, Miss. Elisabeth Walton   F       29      0      0
1       1         1   Allison, Master. Hudson Trevor   M   0.9167      1      2

   ticket      fare    cabin embarked boat body  \
0   24160  211.3375       B5        S    2    ?
1  113781    151.55  C22 C26        S   11    ?

                        home.dest  num_relatives  age_float
0                     St Louis, MO              0    29.0000
1  Montreal, PQ / Chesterville, ON              3     0.9167
```

[44]:
```
titanic["age"] = pd.to_numeric(titanic["age"], errors="coerce")
```

[46]:
```
states.isna().head(2)
```

[46]:
```
   my column name  State  Abbrev   Code
0           False  False   False  False
1           False  False   False  False
```

[48]:
```
states['State'].dropna().head(2)
```

[48]:
```
0    Alabama
1     Alaska
Name: State, dtype: object
```

[50]:
```
states.fillna(0).head(2)
```

[50]:
```
  my column name    State  Abbrev Code
0            bye  Alabama    Ala.   AL
1             hi   Alaska  Alaska   AK
```

9 Working With Dates and Times

[52]:
```
pd.to_datetime("2022-12-31")
```

[52]:
```
Timestamp('2022-12-31 00:00:00')
```

[53]:
```
pd.to_datetime("2022/12/31")
```

[53]:
```
Timestamp('2022-12-31 00:00:00')
```

[54]:
```
pd.to_datetime("December 31st 2022 4:50am")
```

[54]:
```
Timestamp('2022-12-31 04:50:00')
```

[57]:
```
dates = ["3:55:45 Apr. 2nd 90", "3:55:45 Apr. 22nd 91", "7:55:45 Jan. 2nd 90"]
pd.to_datetime(dates)
```

```
C:\Users\Ahmed\AppData\Local\Temp\ipykernel_4932\2244665249.py:2: UserWarning:
Could not infer format, so each element will be parsed individually, falling
back to `dateutil`. To ensure parsing is consistent and as-expected, please
specify a format.
  pd.to_datetime(dates)
```

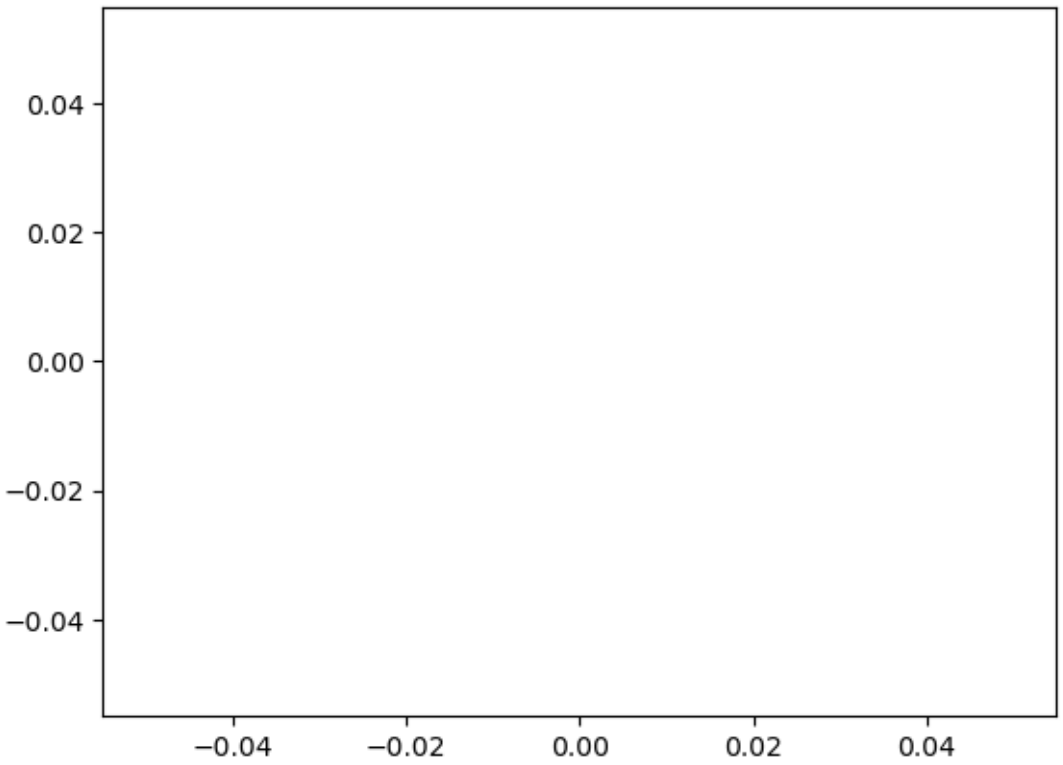
[57]:
```
DatetimeIndex(['1990-04-02 03:55:45', '1991-04-22 03:55:45',
               '1990-01-02 07:55:45'],
              dtype='datetime64[ns]', freq=None)
```

[58]:
```
pd.to_datetime("10/11/12", yearfirst=True, dayfirst=True)
```

[58]:
```
Timestamp('2010-12-11 00:00:00')
```

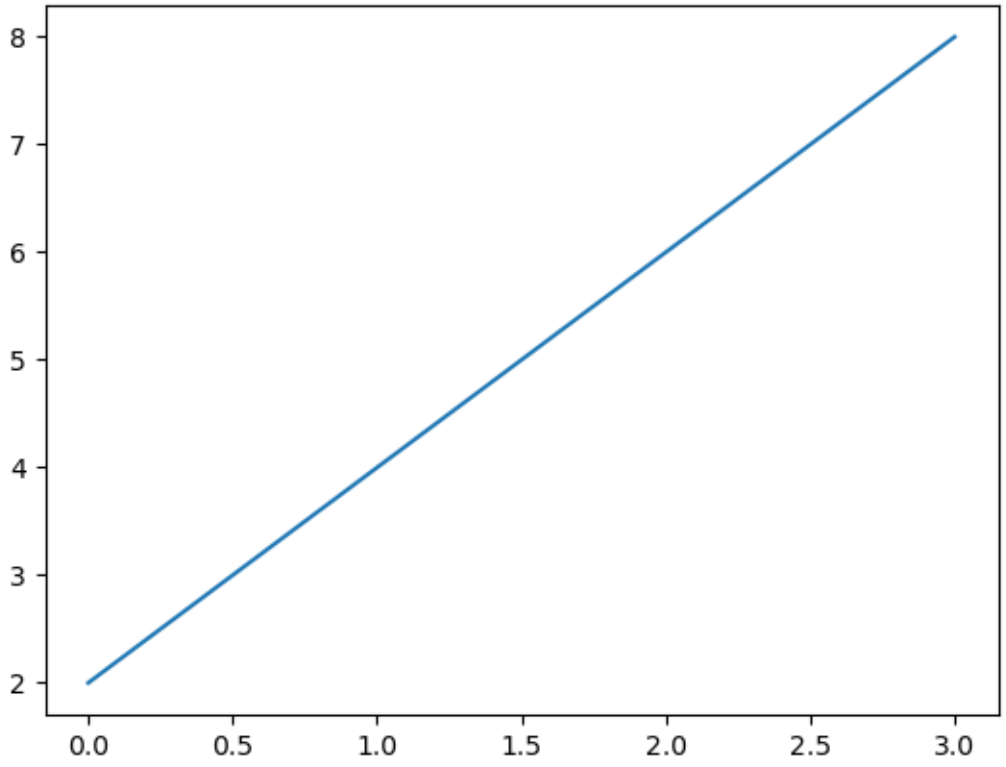
[63]:
```
plt.plot()
```

[63]:
```
[]
```



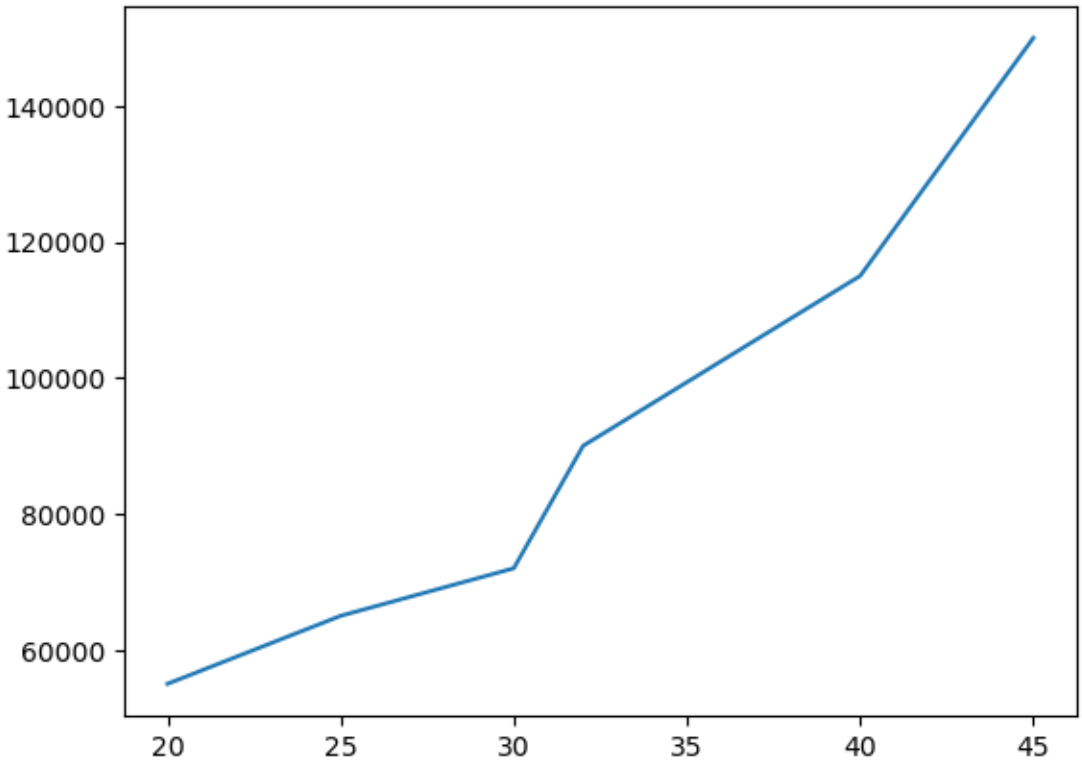
[65]:
```
plt.plot([2,4,6,8])
```

[65]: [<matplotlib.lines.Line2D at 0x2357b0e8d50>]

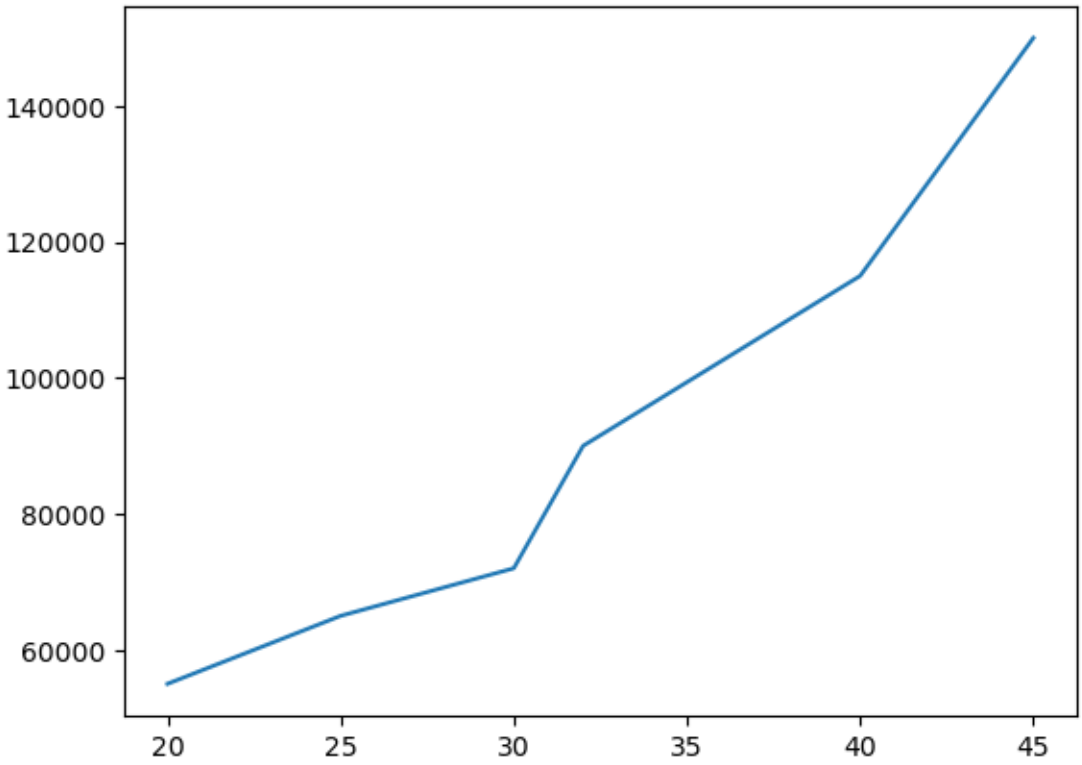


[66]:
```
salaries = [55000, 65000, 72000, 90000, 115000, 150000]
ages = [20, 25, 30, 32, 40, 45]
plt.plot(ages, salaries)
```

[66]: [<matplotlib.lines.Line2D at 0x2357b138710>]

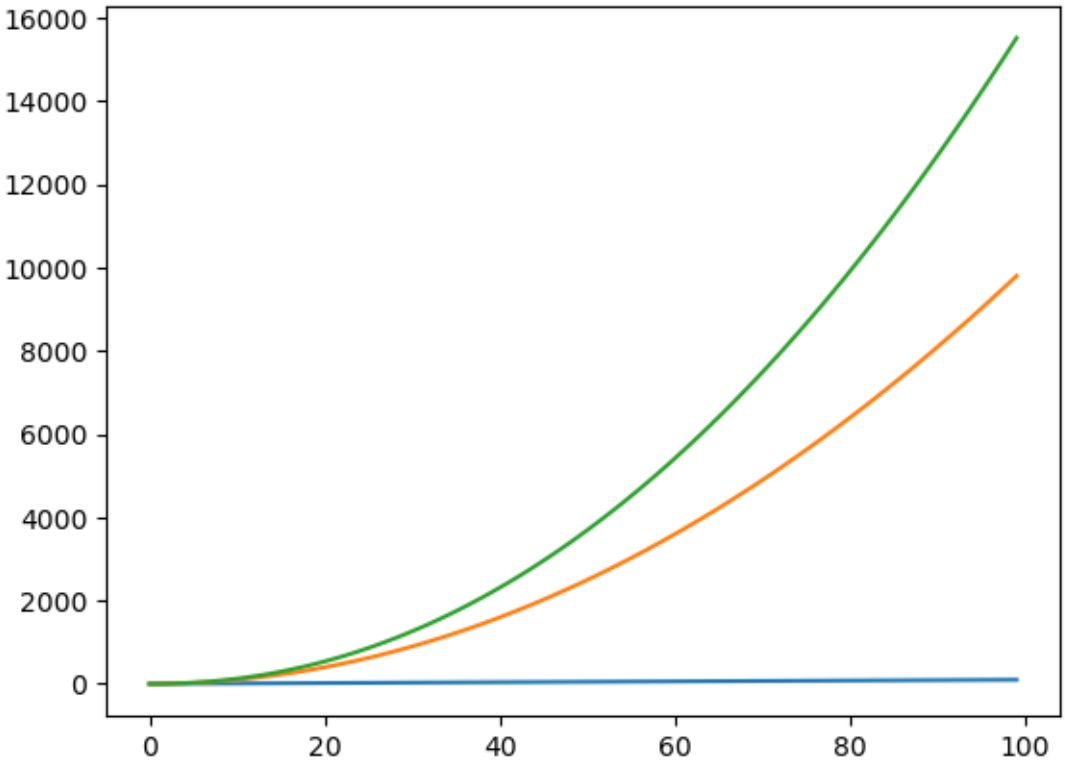


[67]:
```
plt.plot(ages, salaries)
plt.show()
```

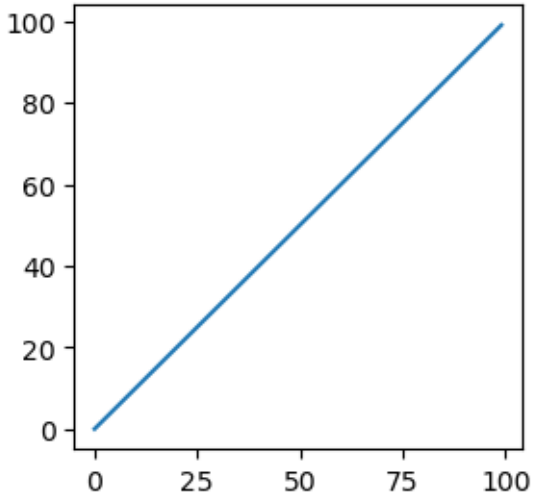


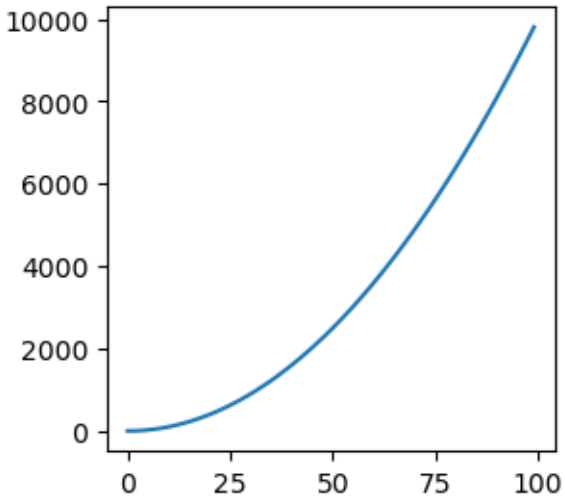
[80]:
```
nums = np.arange(100)
```

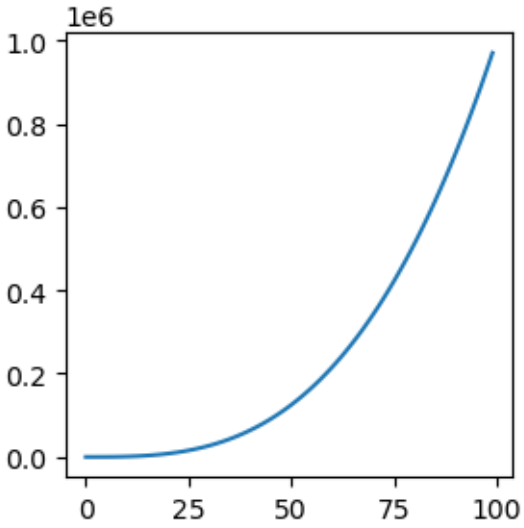
[84]:
```
plt.plot(nums,nums)
plt.plot(nums,nums**2)
plt.plot(nums,nums**2.1)
plt.show()
```



[87]:
```
plt.figure(figsize=(3,3))
plt.plot(nums, nums)
plt.figure(figsize=(3,3))
plt.plot(nums, nums*nums)
plt.figure(figsize=(3, 3))
plt.plot(nums, nums**3)
plt.show()
```







[88]:
```
plt.style.available
```

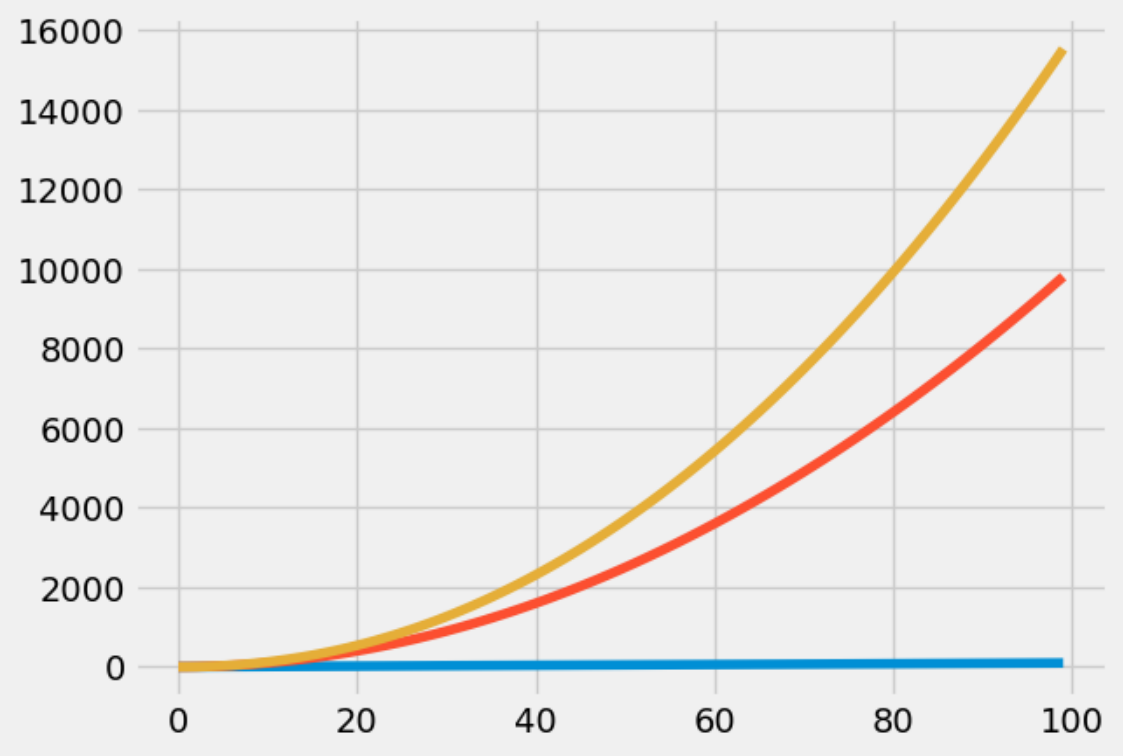
[88]:
```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
 'dark_background',
```

```
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn-v0_8',
 'seaborn-v0_8-bright',
 'seaborn-v0_8-colorblind',
 'seaborn-v0_8-dark',
 'seaborn-v0_8-dark-palette',
 'seaborn-v0_8-darkgrid',
 'seaborn-v0_8-deep',
 'seaborn-v0_8-muted',
 'seaborn-v0_8-notebook',
 'seaborn-v0_8-paper',
 'seaborn-v0_8-pastel',
 'seaborn-v0_8-poster',
 'seaborn-v0_8-talk',
 'seaborn-v0_8-ticks',
 'seaborn-v0_8-white',
 'seaborn-v0_8-whitegrid',
 'tableau-colorblind10']
```

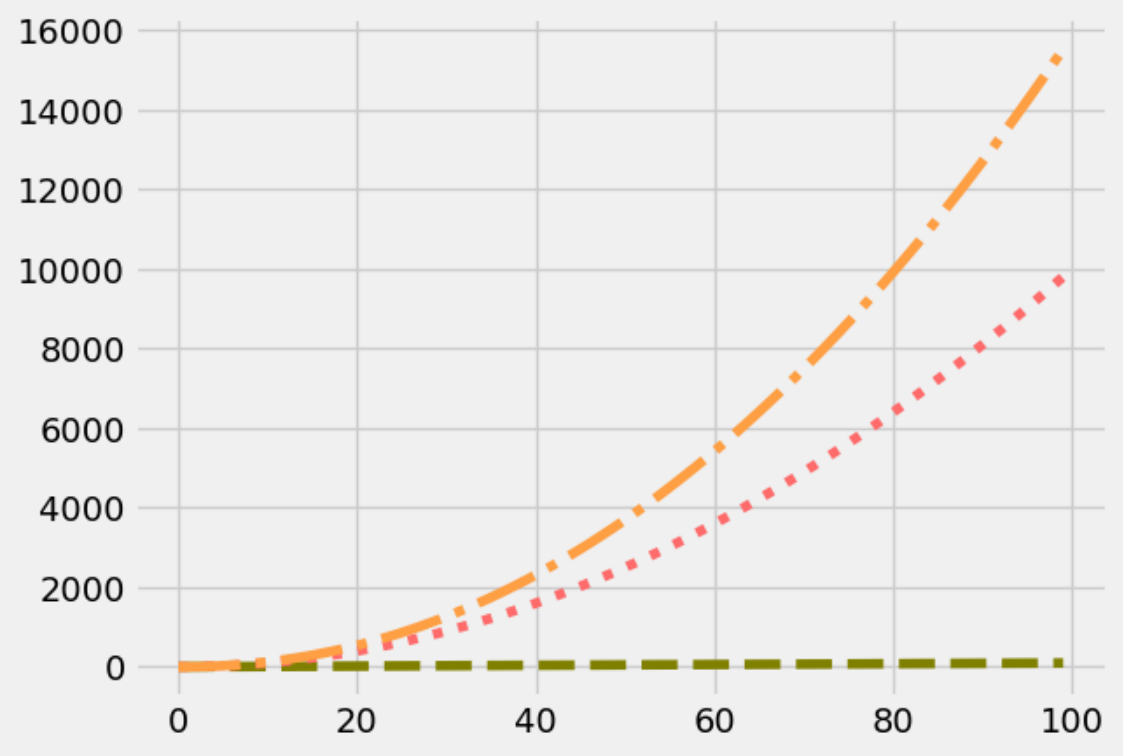
[89]:
```
plt.style.use('fivethirtyeight')
```

[99]:
```
plt.plot(nums, nums)
plt.plot(nums, nums*nums)
plt.plot(nums, nums**2.1)
```

[99]:
```
[<matplotlib.lines.Line2D at 0x2357e8af9d0>]
```

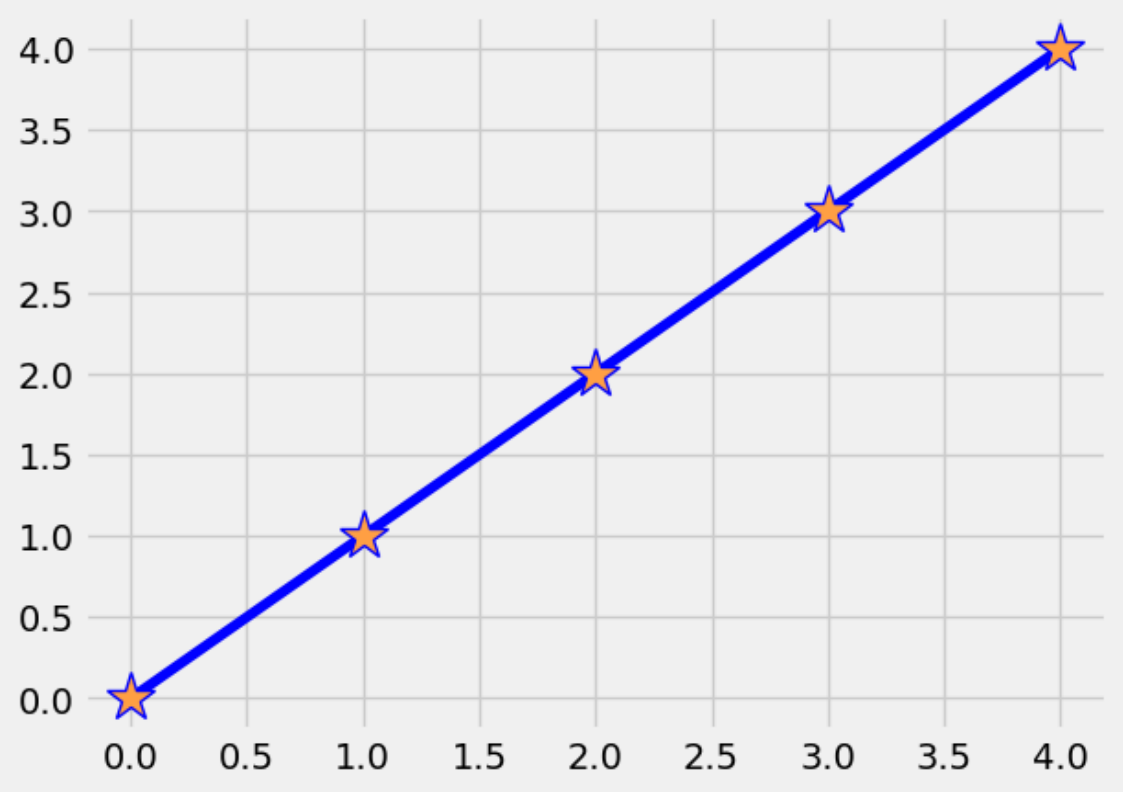


[102]:
```
plt.plot(nums, nums, color="olive", linewidth=4, linestyle="dashed")
plt.plot(nums, nums*nums, color="#ff6b6b", linewidth=4, linestyle="dotted")
plt.plot(nums, nums**2.1, c="#ff9f43", linewidth=4, linestyle="-.")
plt.show()
```

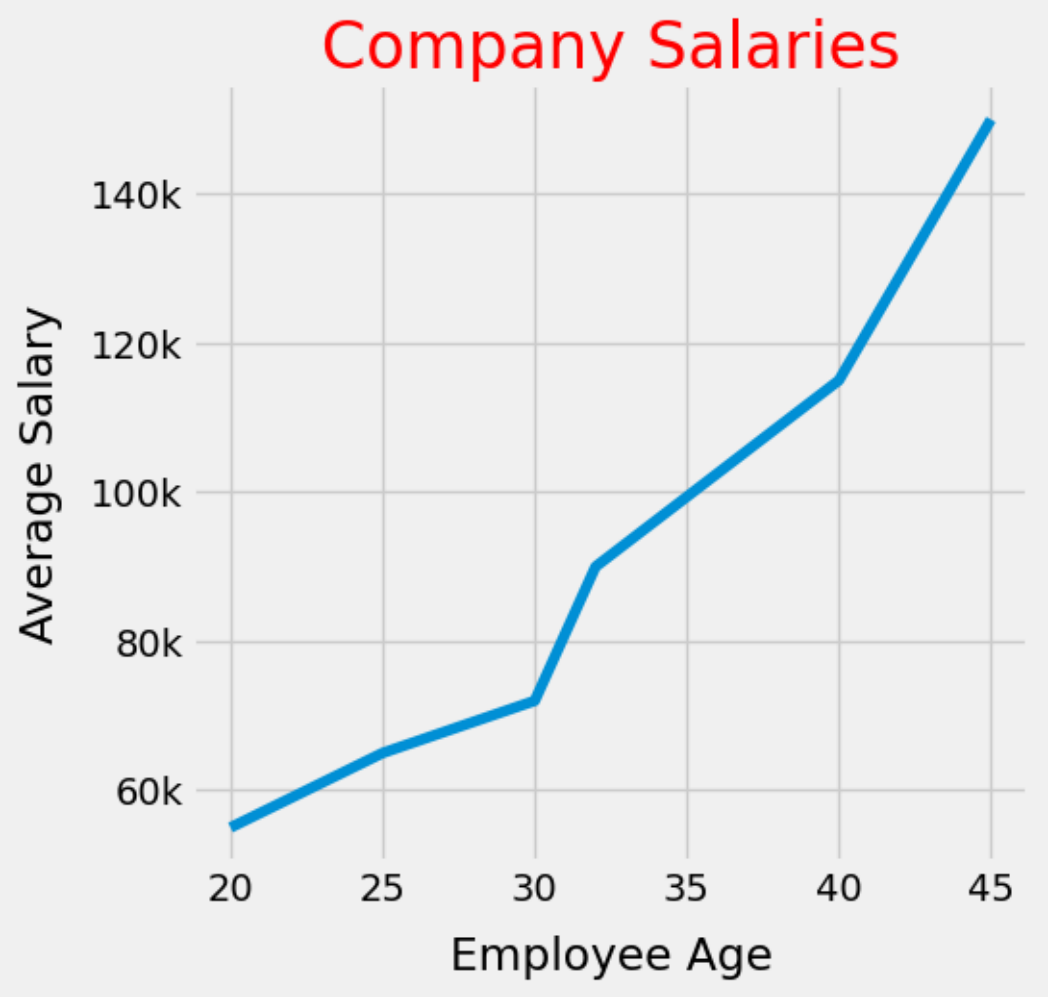


[109]:
```
nums = np.arange(5)
plt.plot(nums, nums, color="blue", marker="*",
         markersize=20, markerfacecolor="#ff9f43")
```

[109]: [<matplotlib.lines.Line2D at 0x235019a0c50>]

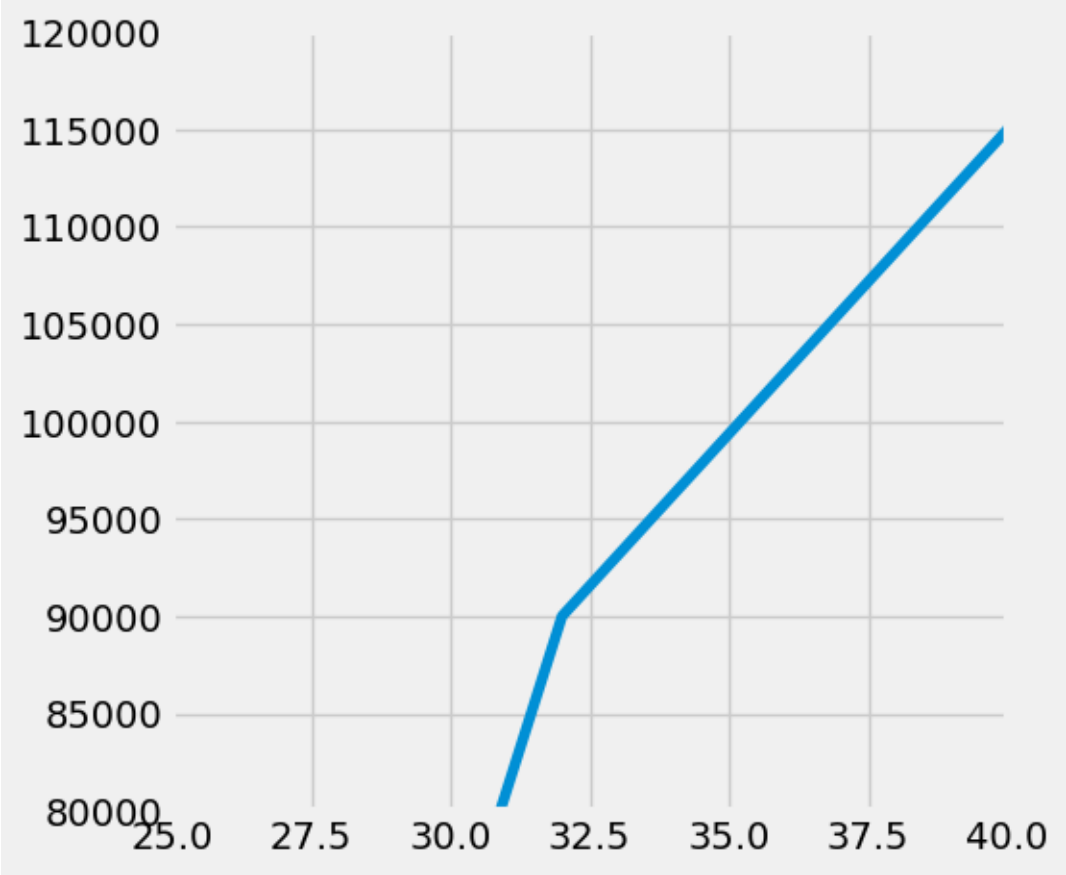


[113]:
```
plt.figure(figsize=(5,5))
salaries=[55000,65000,72000,90000,115000,150000]
ages = [20,25,30,32,40,45]
plt.plot(ages, salaries)
plt.title("Company Salaries", fontsize=24, color="red")
plt.xlabel("Employee Age", labelpad=10)
plt.ylabel("Average Salary", labelpad=10 )
plt.xticks([20,25,30,35,40,45])
plt.yticks([60000,80000,100000, 120000, 140000], labels=["60k", "80k", "100k",␣
↪"120k","140k"])
plt.show()
```



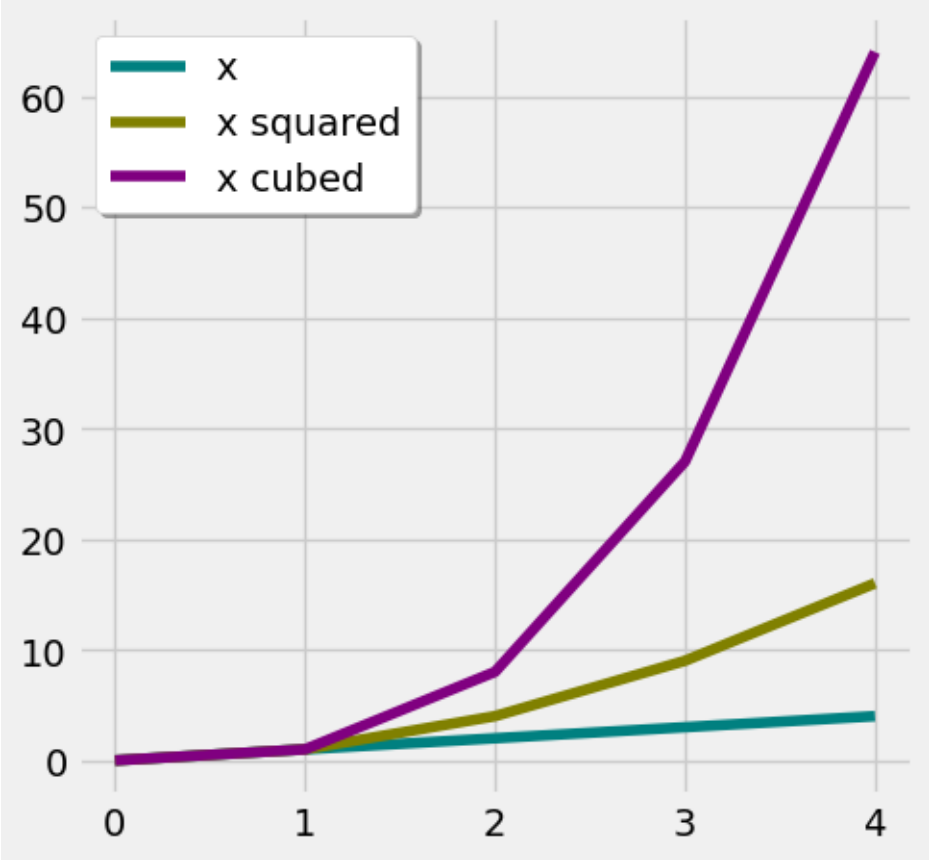
[114]:
```
plt.figure(figsize=(5, 5))
salaries = [55000, 65000, 72000, 90000, 115000, 150000]
ages = [20, 25, 30, 32, 40, 45]
plt.plot(ages, salaries)
plt.xlim(25, 40)
plt.ylim(80000, 120000)
```

[114]: (80000.0, 120000.0)



[121]:
```
plt.figure(figsize=(5, 5))
plt.plot(nums, color="teal", label="x")
plt.plot(nums**2, color="olive", label="x squared")
plt.plot(nums**3, color="purple", label="x cubed")
plt.legend(shadow=True, frameon=True, facecolor="white")
```

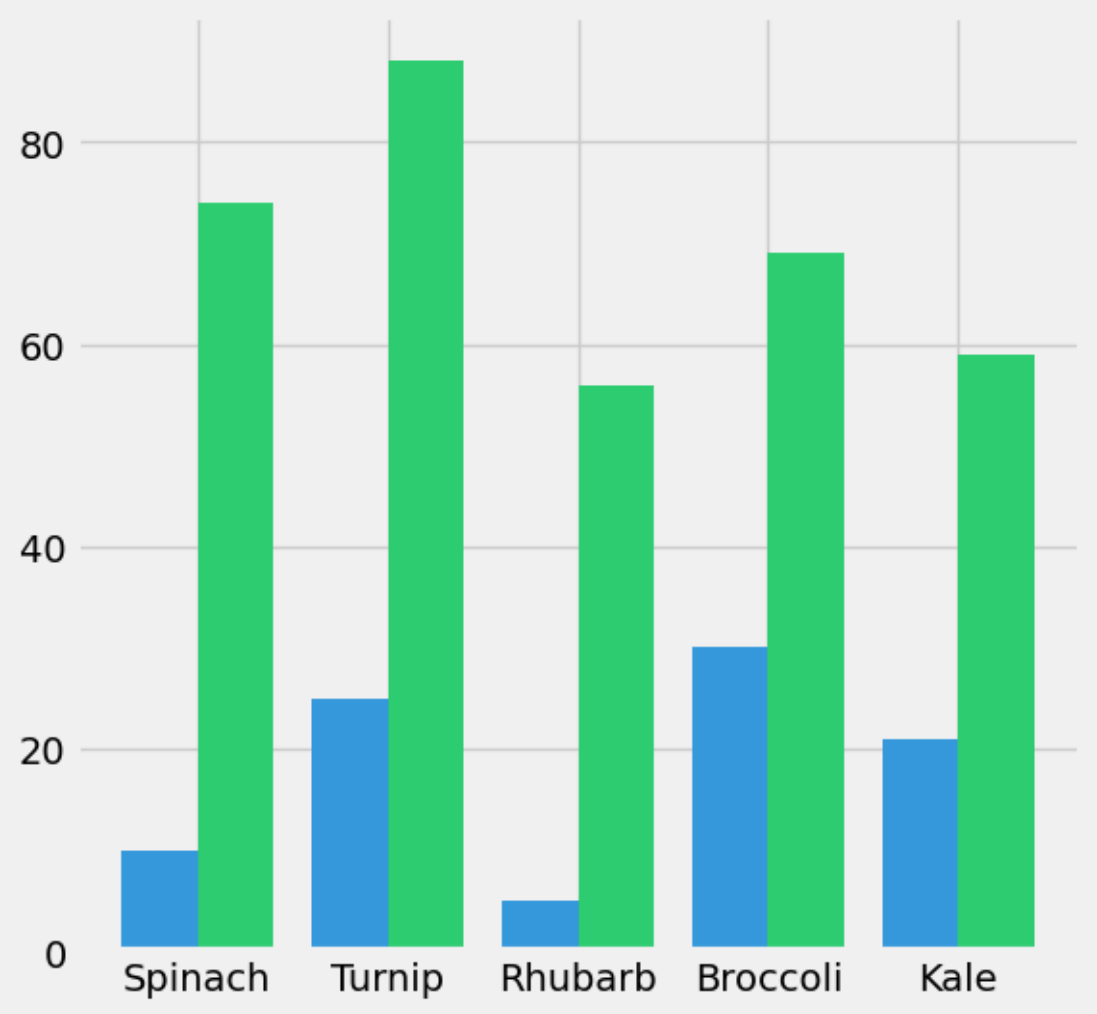
[121]: <matplotlib.legend.Legend at 0x23502c6cc90>



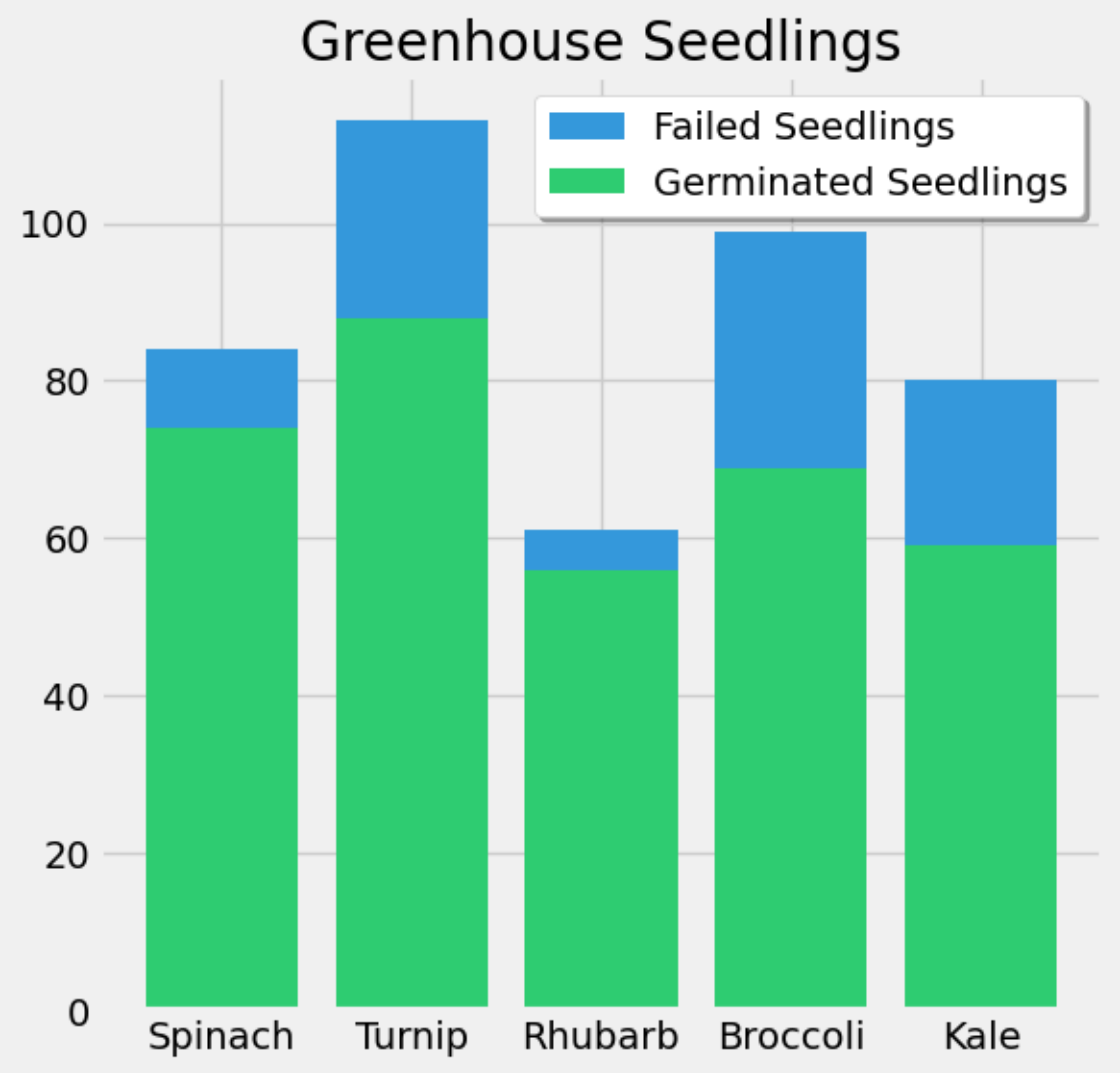
[122]:
```
plants = ['Spinach', 'Turnip', 'Rhubarb', 'Broccoli', 'Kale']
died = [10,25,5,30,21]
germinated = [74, 88, 56,69,59]
```

[123]:
```
plt.figure(figsize=(6, 6))
plt.bar(plants, died, color="#3498db")
plt.bar(plants, germinated, width=0.4, color="#2ecc71", align="edge")
```

[123]:
```
<BarContainer object of 5 artists>
```

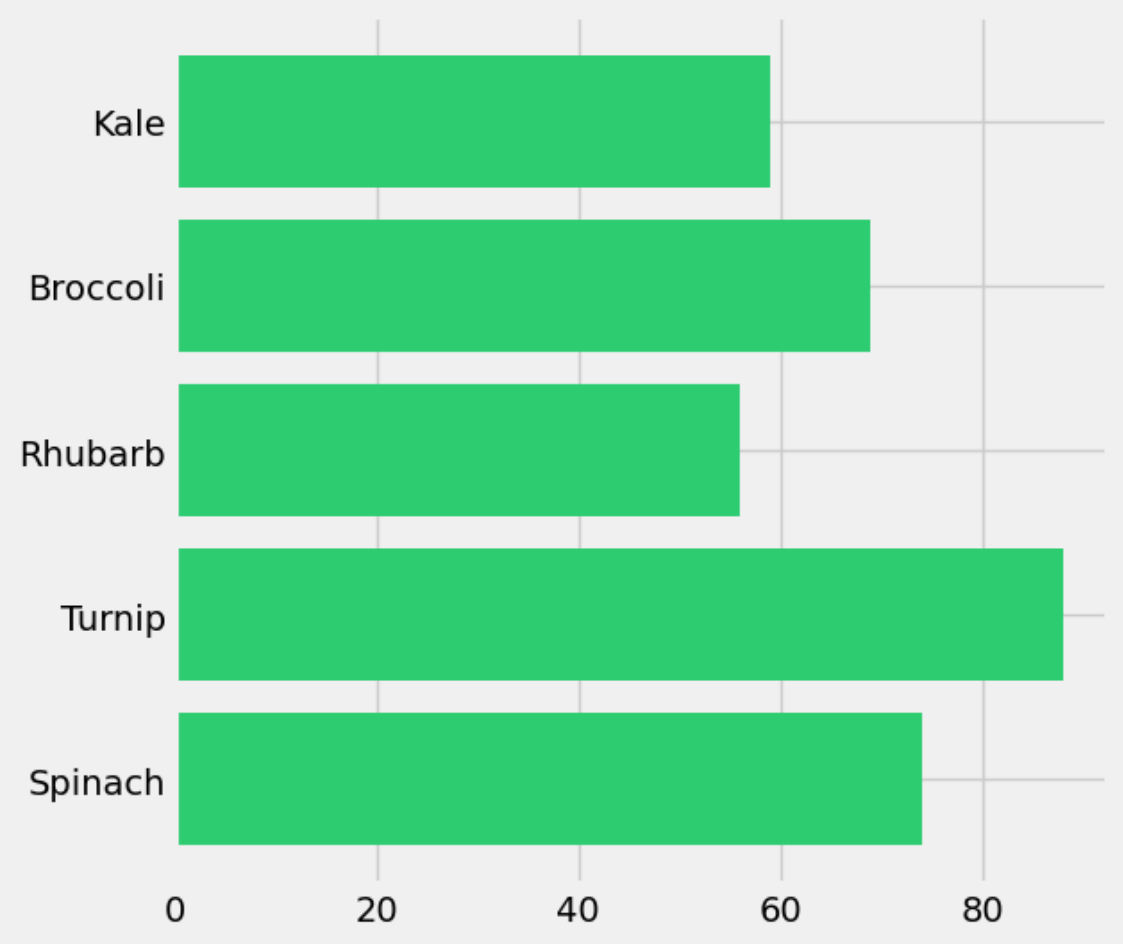


[125]:
```
plt.figure(figsize=(6, 6))
plt.bar(plants, died, color="#3498db",
        bottom=germinated, label="Failed Seedlings")
plt.bar(plants, germinated, color="#2ecc71", label="Germinated Seedlings")
plt.legend(shadow=True, frameon=True, facecolor="white")
plt.title("Greenhouse Seedlings")
plt.show()
```

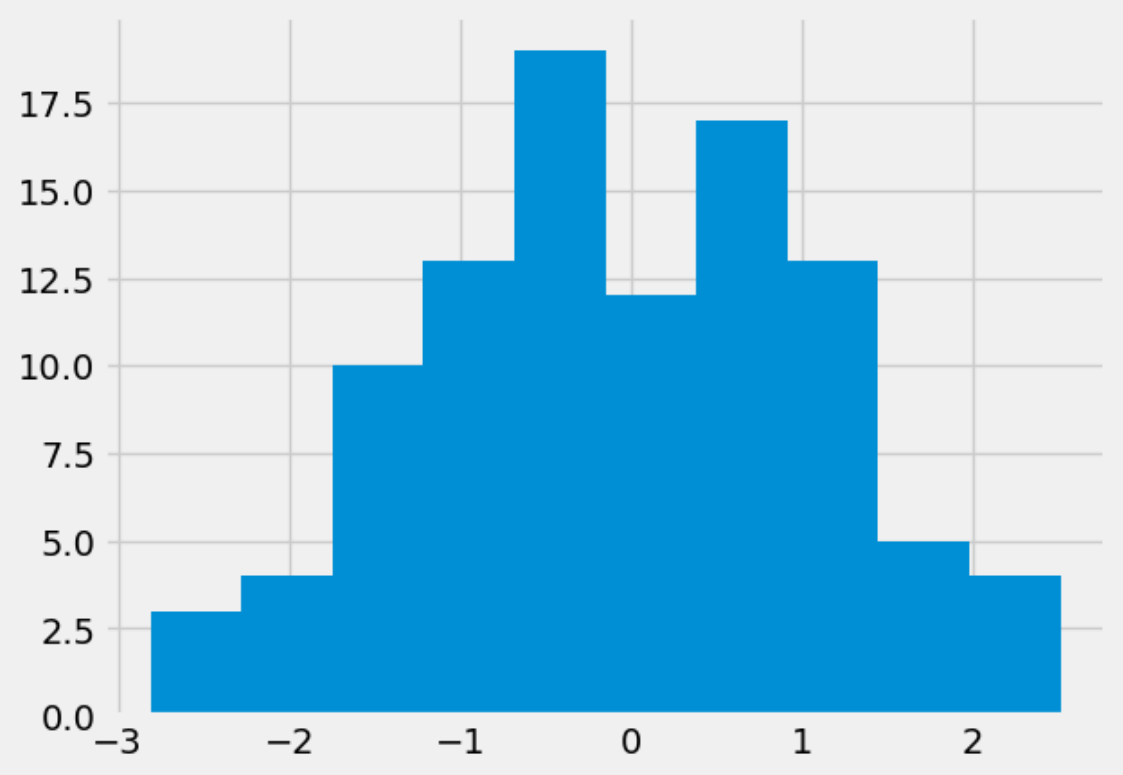


[126]:
```
plt.figure(figsize=(6, 6))
plt.barh(plants, germinated, color="#2ecc71")
```

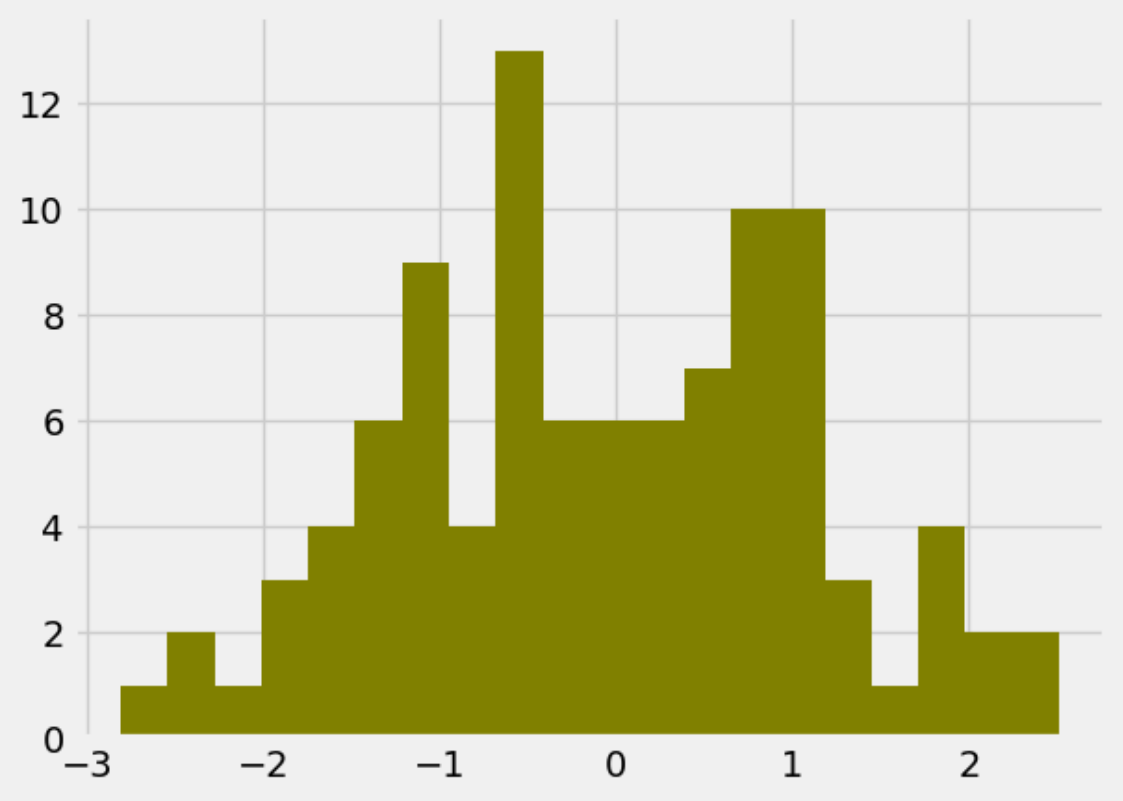
[126]: <BarContainer object of 5 artists>



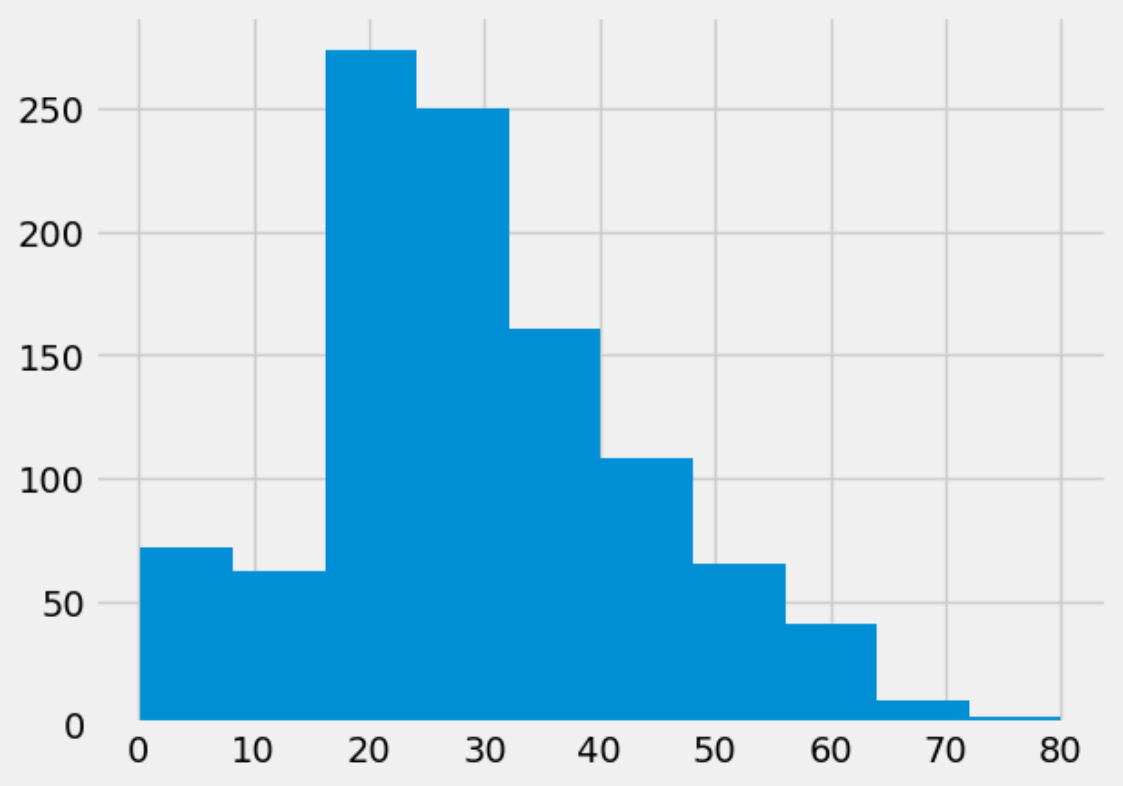
[128]:
```
nums = np.random.randn(100)
plt.hist(nums)
plt.show()
```



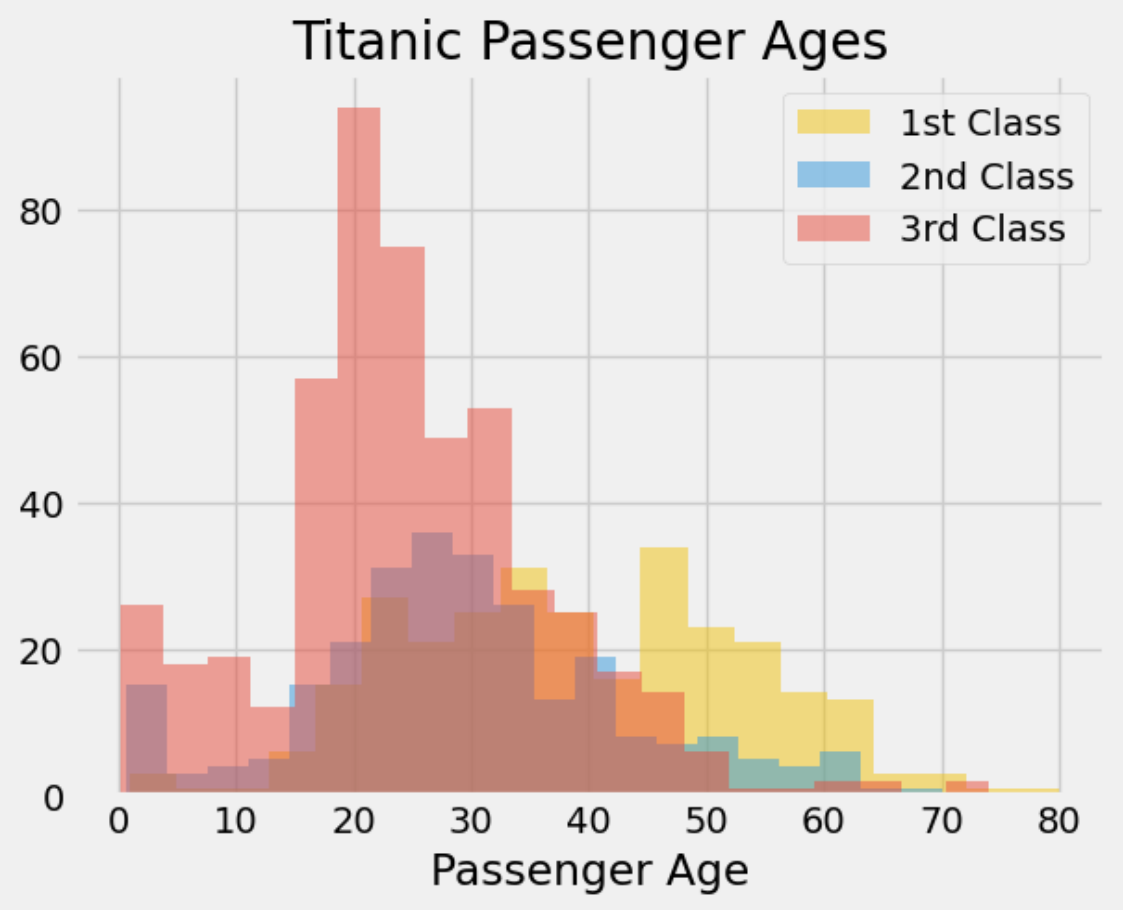
[129]:
```
plt.hist(nums, bins=20, color="olive")
plt.show()
```



[131]:
```
titanic["age"] = pd.to_numeric(titanic["age"], errors="coerce")
plt.hist(titanic["age"])
plt.show()
```



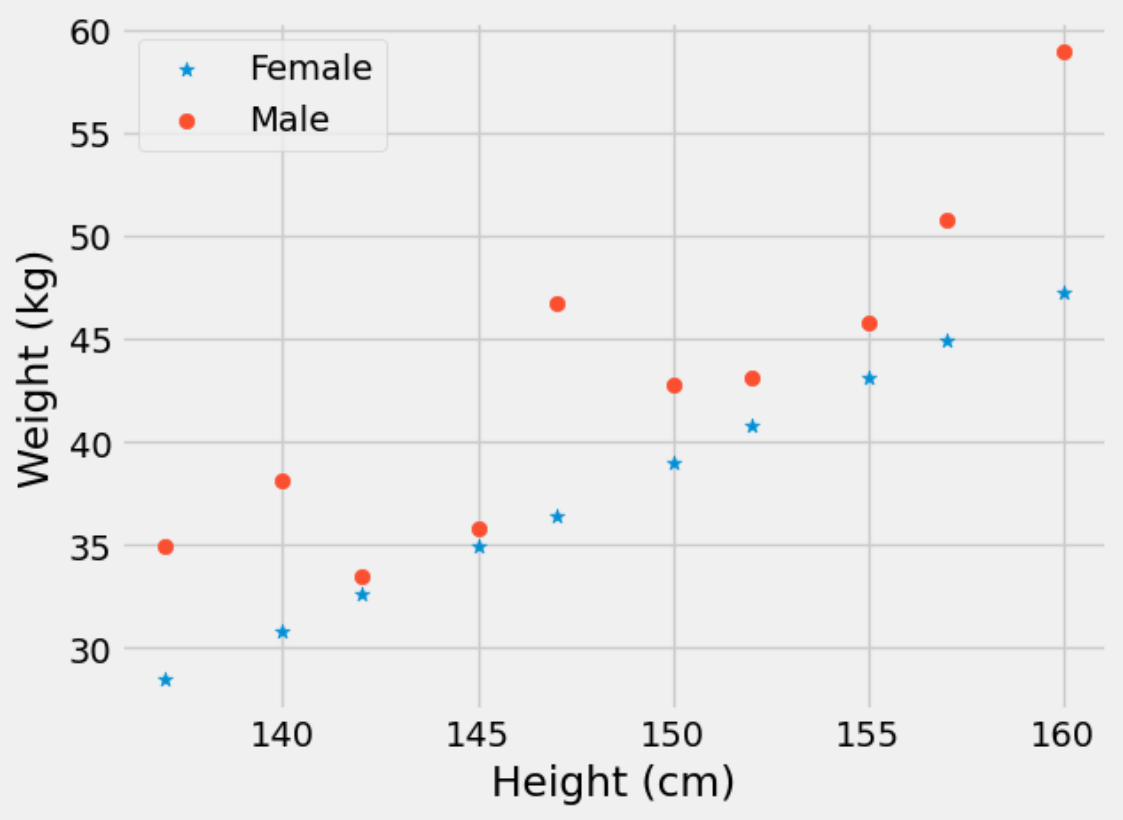
[132]:
```
first_class = titanic[titanic["pclass"] == 1]["age"]
second_class = titanic[titanic["pclass"] == 2]["age"]
third_class = titanic[titanic["pclass"] == 3]["age"]
plt.hist(first_class, label="1st Class", alpha=0.5, color="#f1c40f", bins=20)
plt.hist(second_class, label="2nd Class", alpha=0.5, color="#3498db", bins=20)
plt.hist(third_class, label="3rd Class", alpha=0.5, color="#e74c3c", bins=20)
plt.legend()
plt.title("Titanic Passenger Ages")
plt.xlabel("Passenger Age")
plt.show()
```



[133]:
```
heights = [137,140,142,145,147,150,152,155,157,160]
f_weights = [28.5,30.8,32.6,34.9,36.4,39,40.8,43.1,44.9,47.2]
m_weights = [34.9,38.1,33.5,35.8,46.7, 42.8,43.1,45.8,50.8,58.9]
```

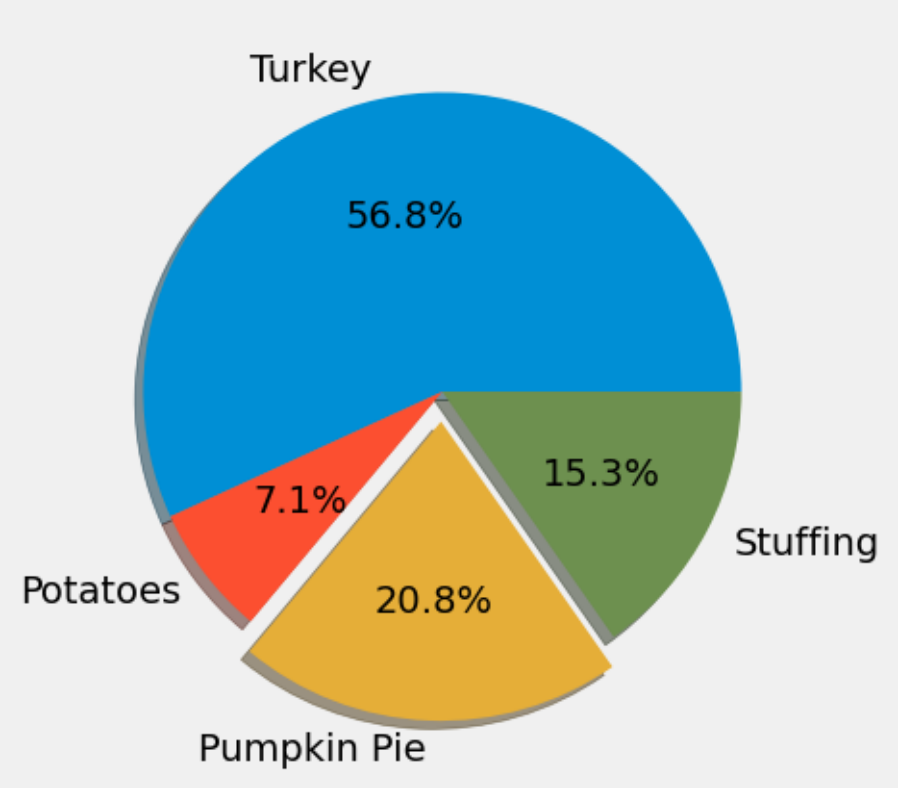
[140]:
```
plt.scatter(heights, f_weights,  marker="*", label="Female")
plt.scatter(heights, m_weights,marker="o", label="Male")
plt.legend()
plt.xlabel("Height (cm)")
plt.ylabel("Weight (kg)")
```

[140]:
```
Text(0, 0.5, 'Weight (kg)')
```

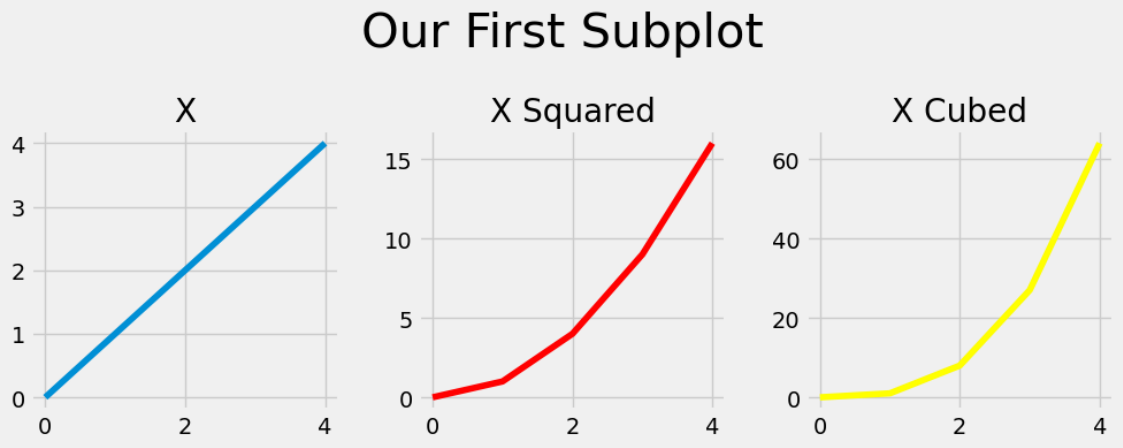


[141]:
```
labels = ["Turkey", "Potatoes", "Pumpkin Pie", "Stuffing"]
prices = [25.99,3.24,9.50, 6.99]
```

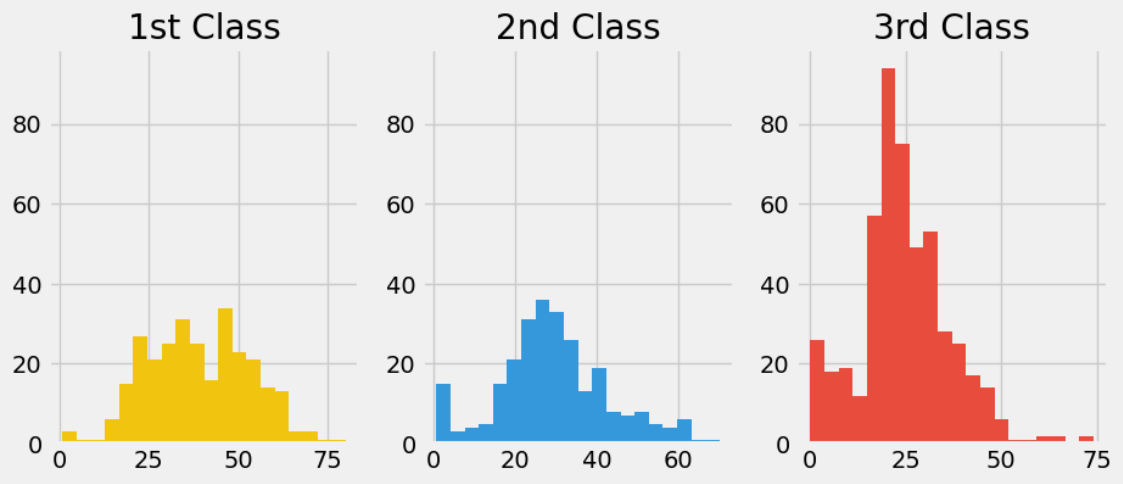
[142]:
```
plt.pie(prices, labels=labels, autopct="%1.1f%%",
        shadow=True, explode=(0, 0, 0.1, 0))
plt.show()
```



[143]:
```
nums = np.arange(5)
plt.figure(figsize=(10, 4))
plt.suptitle("Our First Subplot", fontsize=30)

plt.subplot(1, 3, 1)
plt.title("X")
plt.plot(nums, nums)

plt.subplot(1, 3, 2)
plt.plot(nums, nums**2, color="red")
plt.title("X Squared")


plt.subplot(1, 3, 3)
plt.plot(nums, nums**3, color="yellow")
plt.title("X Cubed")
plt.tight_layout()
plt.show()
```



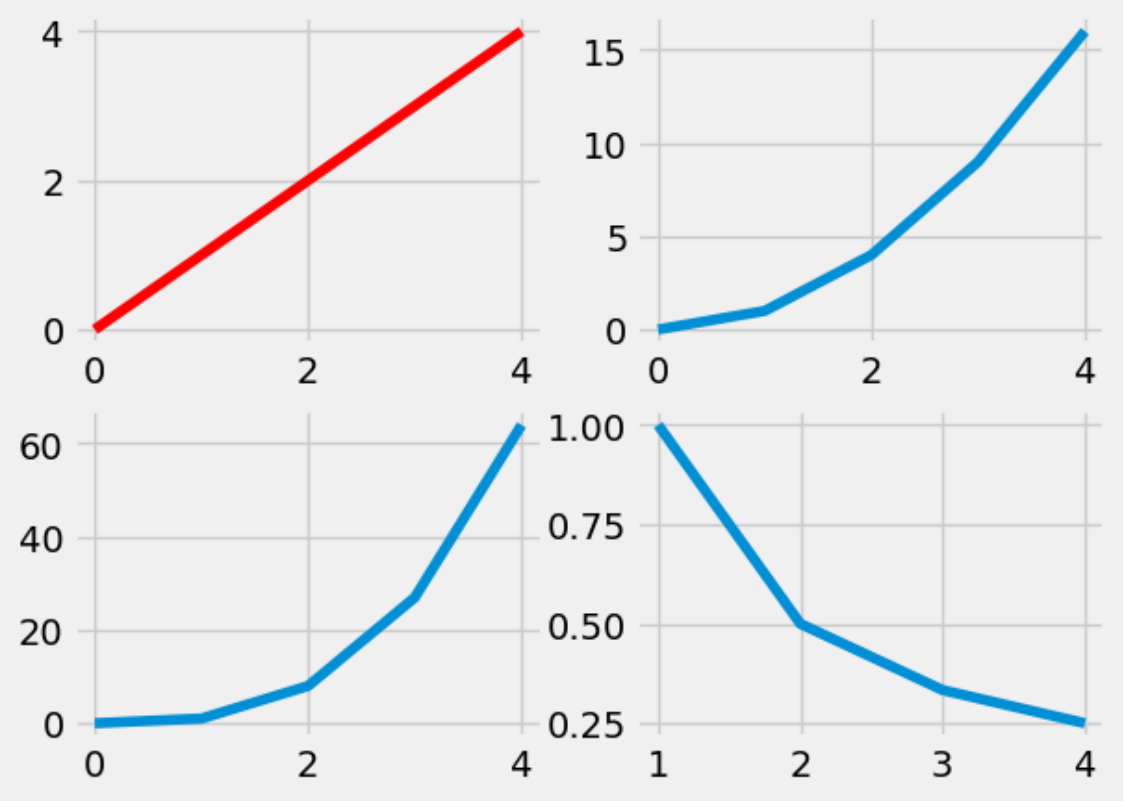
[144]:
```
first_class = titanic[titanic["pclass"] == 1]["age"]
second_class = titanic[titanic["pclass"] == 2]["age"]
third_class = titanic[titanic["pclass"] == 3]["age"]

plt.figure(figsize=(10,4))
ax = plt.subplot(1,3,1)
plt.hist(first_class, label="1st Class", color="#f1c40f", bins=20)
plt.title("1st Class")

plt.subplot(1,3,2, sharey=ax)
plt.hist(second_class, label="2nd Class", color="#3498db", bins=20)
plt.title("2nd Class")


plt.subplot(1,3,3, sharey=ax)
plt.hist(third_class, label="3rd Class", color="#e74c3c", bins=20)
plt.title("3rd Class")
plt.show()
```



[156]:
```
fig, axs = plt.subplots(2,2)
axs[0][0].plot(nums,color="red")
axs[0][1].plot(nums*nums)
axs[1][0].plot(nums**3)
axs[1][1].plot(1/nums)

axs[0][1].set_xticks([0,2,4])
```

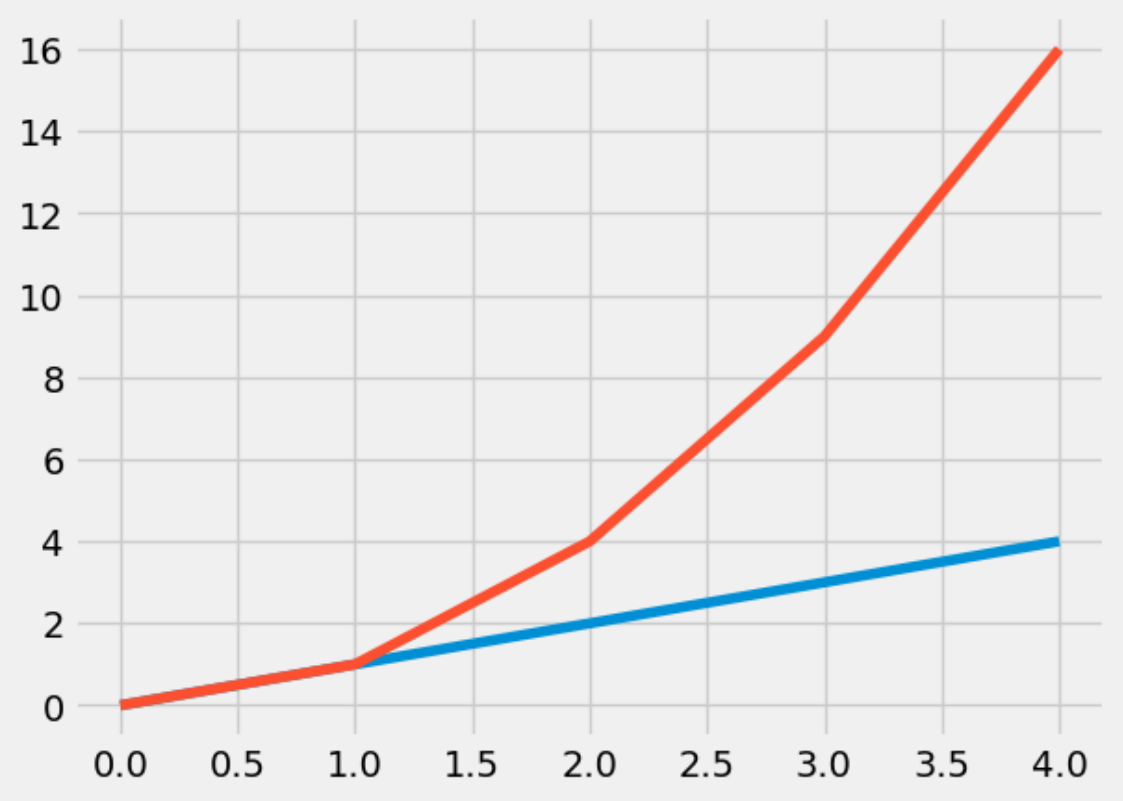
```
C:\Users\Ahmed\AppData\Local\Temp\ipykernel_4932\2433834221.py:5:
RuntimeWarning: divide by zero encountered in divide
  axs[1][1].plot(1/nums)
```

[156]:
```
[<matplotlib.axis.XTick at 0x23505ef1a50>,
 <matplotlib.axis.XTick at 0x23505f17ad0>,
 <matplotlib.axis.XTick at 0x23505f21a50>]
```



[155]:
```
fig, ax = plt.subplots()
ax.plot(nums)
ax.plot(nums*nums)
```

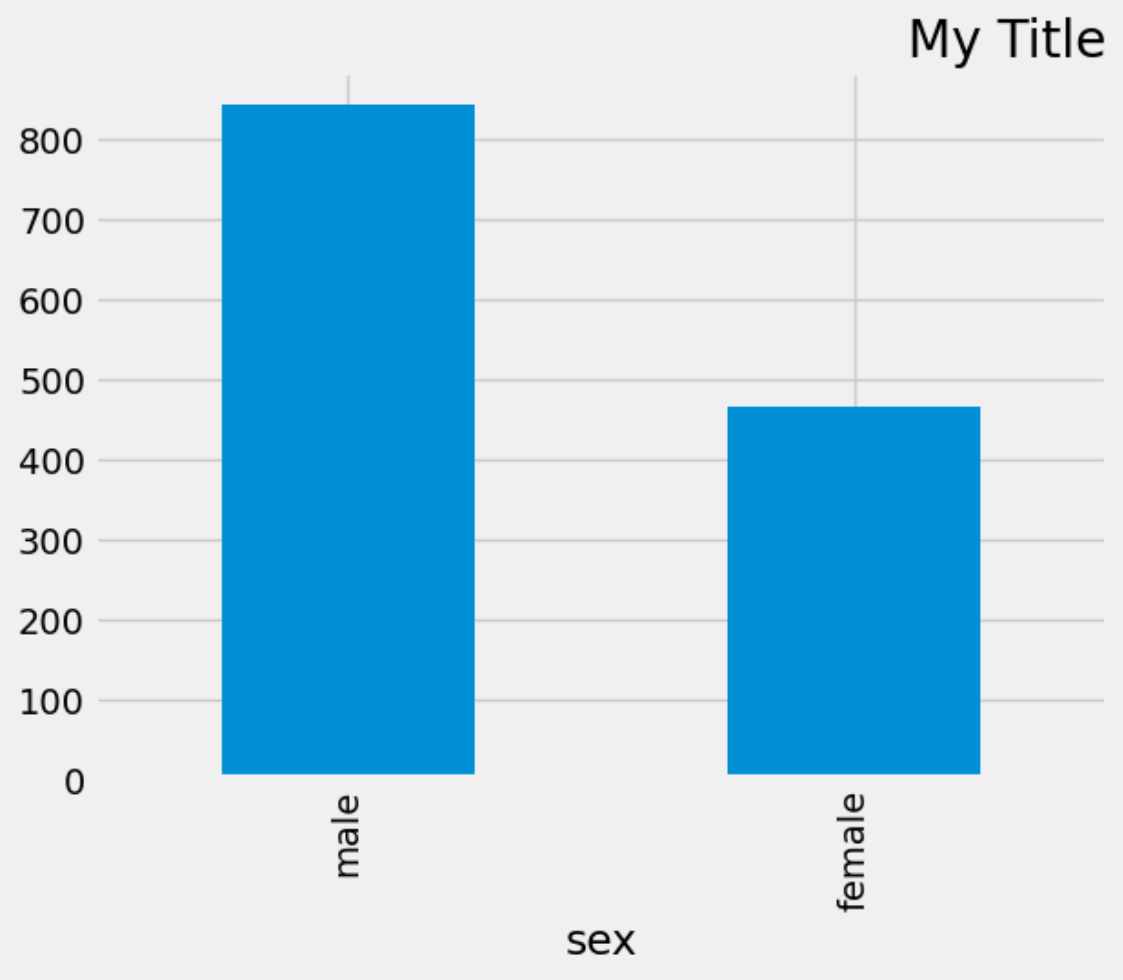
[155]: [<matplotlib.lines.Line2D at 0x23505e91710>]



10 Plotting With Pandas (and matplotlib)

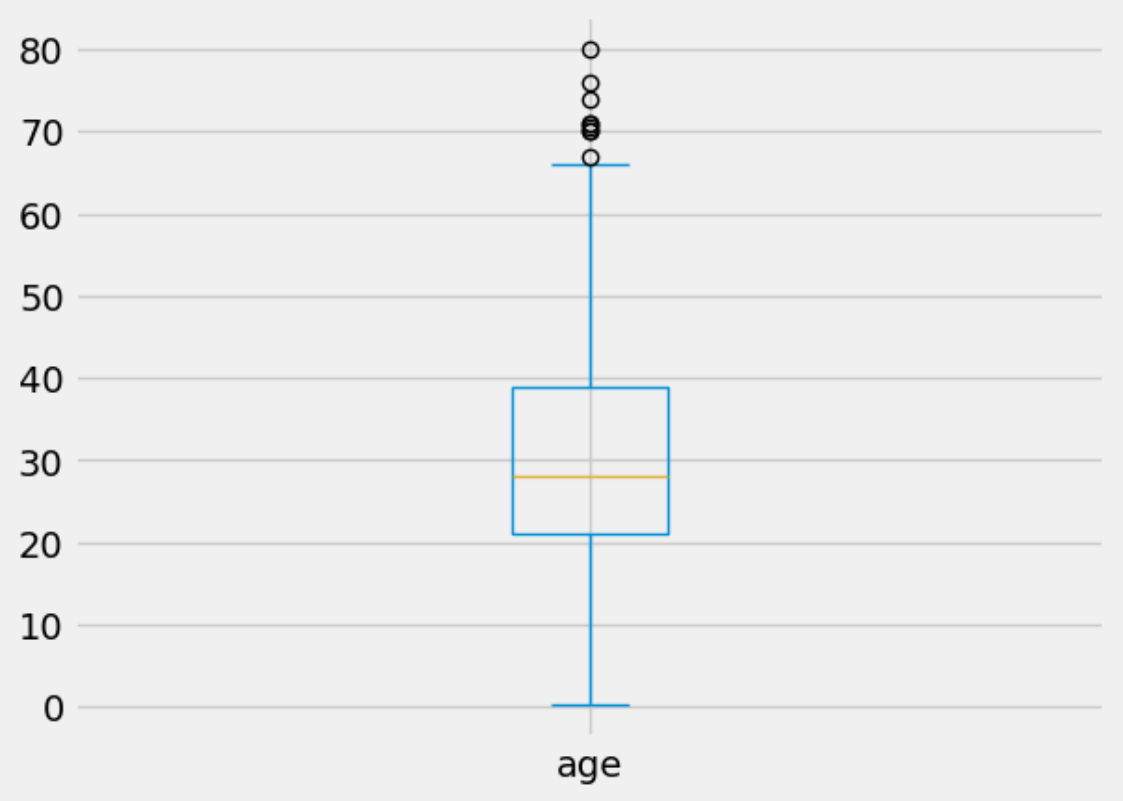
[158]:
```
titanic.sex.value_counts().plot(kind="bar")
plt.title("My Title", loc="right")
```

[158]:
```
Text(1.0, 1.0, 'My Title')
```



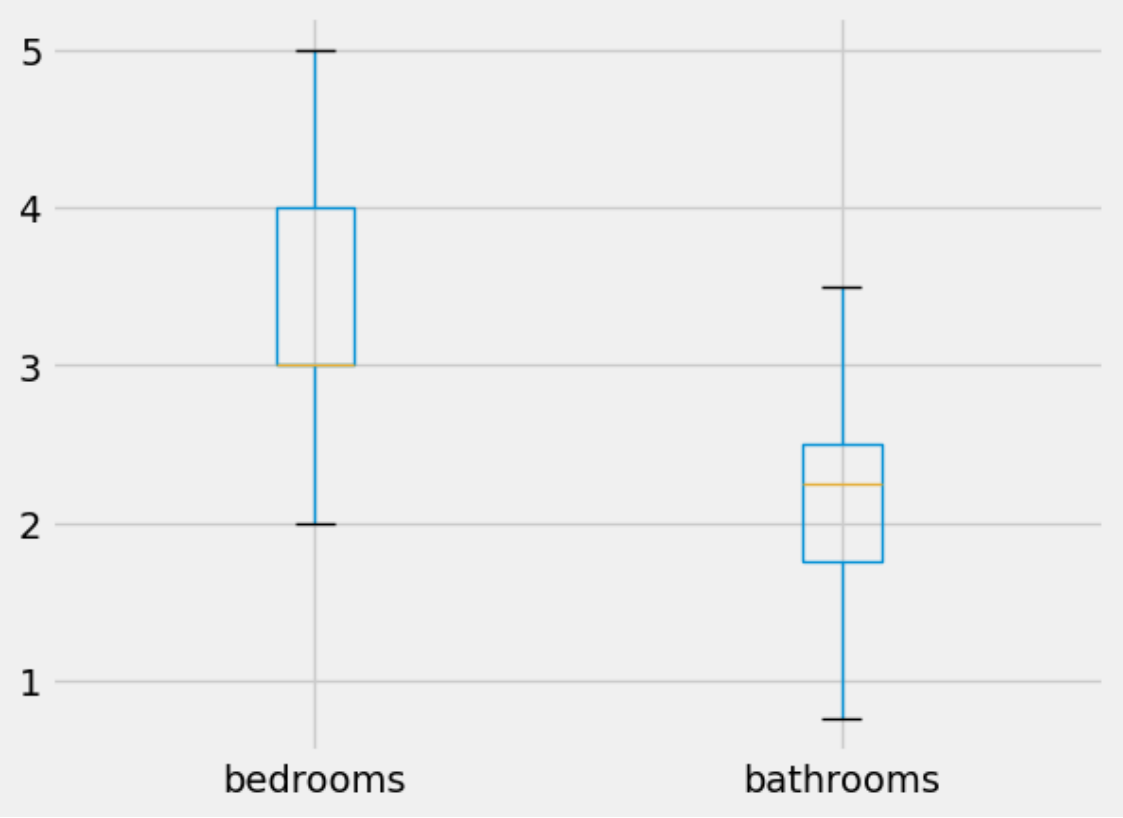
[160]:
```
titanic.age.plot(kind="box")
```

[160]: <Axes: >



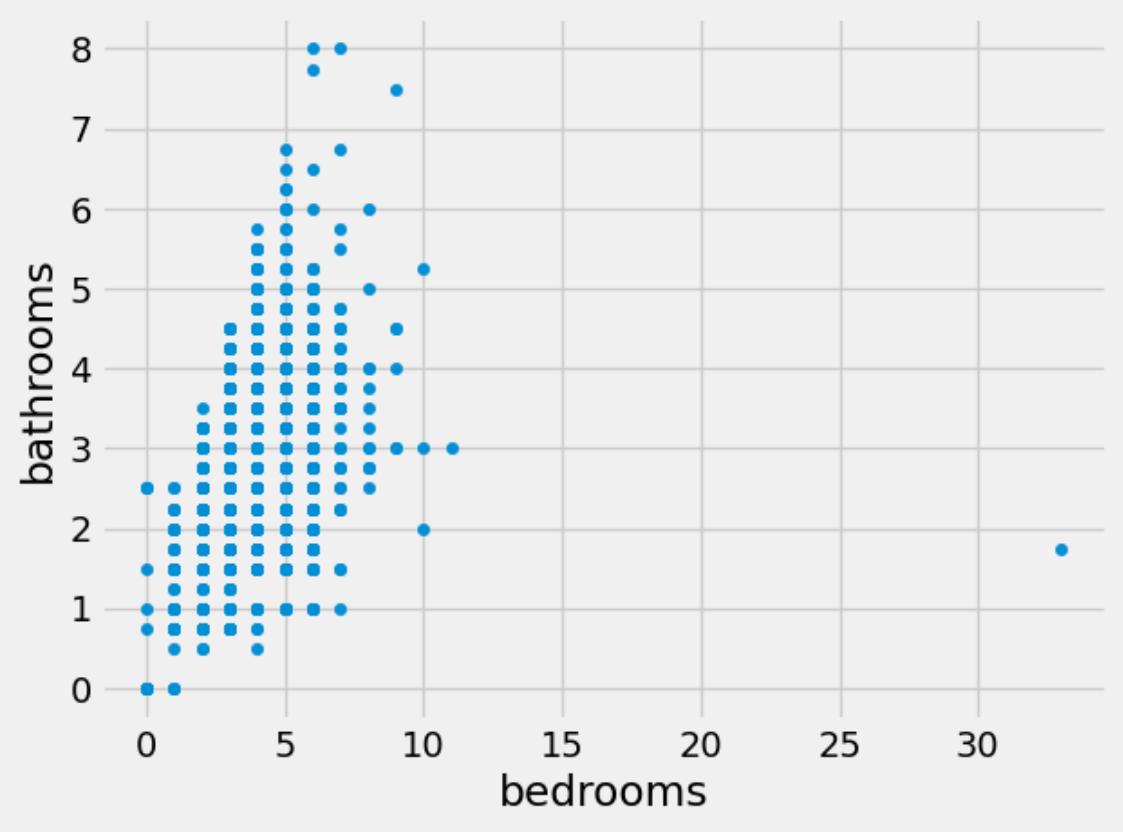
[161]:
```
houses[["bedrooms", "bathrooms"]].boxplot(showfliers=False)
```

[161]: <Axes: >



[162]:
```
houses.plot.scatter(x="bedrooms", y="bathrooms")
```

[162]:
```
<Axes: xlabel='bedrooms', ylabel='bathrooms'>
```



11 Group by

[163]:
```
df = titanic[["pclass", "survived", "sex", "age"]]
```

[165]:
```
df.groupby("sex")['survived'].mean()
```

[165]:
```
sex
female    0.727468
male      0.190985
Name: survived, dtype: float64
```

[166]:
```
df.groupby("age").first().head(2)
```

[166]:

age	pclass	survived	sex
0.1667	3	1	female
0.3333	3	0	male

[167]:
```
gbo = df.groupby(by="sex")
gbo.get_group("male").head(2)
```

[167]:
```
    pclass  survived   sex      age
1        1         1  male   0.9167
3        1         0  male  30.0000
```

[168]:
```
gbo.ngroups
```

[168]:
```
2
```

[169]:
```
gbo["age"].mean()
```

[169]:
```
sex
female    28.687071
male      30.585233
Name: age, dtype: float64
```

[170]:
```
gbo["age"].max()
```

[170]:
```
sex
female    76.0
male      80.0
Name: age, dtype: float64
```

[171]:
```
titanic.groupby("sex")["age"].agg(["min", "max", "mean", "median"])
```

[171]:
```
           min   max       mean  median
sex
female  0.1667  76.0  28.687071    27.0
male    0.3333  80.0  30.585233    28.0
```

[172]:
```
titanic.groupby("sex").agg({"age": ["min", "max"], "pclass": "mean"})
```

[172]:
```
           age          pclass
           min   max      mean
sex
female  0.1667  76.0  2.154506
male    0.3333  80.0  2.372479
```

[173]:
```
titanic.groupby(["sex", "pclass", "survived"])["age"].mean()
```

[173]:
```
sex     pclass  survived
female  1       0           35.200000
                1           37.109375
        2       0           34.090909
                1           26.711051
        3       0           23.418750
                1           20.814815
male    1       0           43.658163
```

```
                1           36.168240
        2       0           33.092593
                1           17.449274
        3       0           26.679598
                1           22.436441
Name: age, dtype: float64
```

12 MultiIndexes

[49]:
```
titanic['age'] = titanic["age"].replace(['?'], [None]).astype('float')
titanic['fare'] = titanic["fare"].replace(['?'], [None]).astype('float')
```

[9]:
```
s1 = titanic.groupby("sex")["age"].mean()
```

[10]:
```
s1.index
```

[10]:
```
Index(['female', 'male'], dtype='object', name='sex')
```

[11]:
```
s1.head(2)
```

[11]:
```
sex
female    28.687071
male      30.585233
Name: age, dtype: float64
```

[17]:
```
df = titanic.groupby(["pclass", "sex"]).mean(numeric_only=True)
```

[18]:
```
df.index
```

[18]:
```
MultiIndex([(1, 'female'),
            (1,   'male'),
            (2, 'female'),
            (2,   'male'),
            (3, 'female'),
            (3,   'male')],
           names=['pclass', 'sex'])
```

[19]:
```
titanic.index
```

[19]:
```
RangeIndex(start=0, stop=1309, step=1)
```

[21]:
```
titanic.groupby(["sex", "age"]).mean(numeric_only=True)
```

[21]:
```
                  pclass  survived     sibsp     parch       fare
sex    age
female 0.1667   3.000000  1.000000  1.000000  2.000000  20.575000
```

```
        0.7500   3.000000  1.000000  2.000000  1.000000  19.258300
        0.9167   2.000000  1.000000  1.000000  2.000000  27.750000
        1.0000   2.800000  0.800000  0.800000  1.400000  19.467500
        2.0000   2.571429  0.285714  1.428571  1.428571  39.955357
…                  …         …         …         …         …
male    70.0000  1.500000  0.000000  0.500000  0.500000  40.750000
        70.5000  3.000000  0.000000  0.000000  0.000000   7.750000
        71.0000  1.000000  0.000000  0.000000  0.000000  42.079200
        74.0000  3.000000  0.000000  0.000000  0.000000   7.775000
        80.0000  1.000000  1.000000  0.000000  0.000000  30.000000

[166 rows x 5 columns]
```

[23]:
```
titanic.set_index(['pclass','sex']).head(3)
```

[23]:
```
               survived                             name      age  sibsp  \
pclass sex
1      female         1   Allen, Miss. Elisabeth Walton  29.0000      0
       male           1  Allison, Master. Hudson Trevor   0.9167      1
       female         0    Allison, Miss. Helen Loraine   2.0000      1

               parch  ticket      fare    cabin embarked boat body  \
pclass sex
1      female      0   24160  211.3375       B5        S    2    ?
       male        2  113781  151.5500  C22 C26        S   11    ?
       female      2  113781  151.5500  C22 C26        S    ?    ?

                                   home.dest
pclass sex
1      female                     St Louis, MO
       male    Montreal, PQ / Chesterville, ON
       female  Montreal, PQ / Chesterville, ON
```

[26]:
```
titanic.sort_index().head(3)
```

[26]:
```
   pclass  survived                             name     sex      age  sibsp  \
0       1         1   Allen, Miss. Elisabeth Walton  female  29.0000      0
1       1         1  Allison, Master. Hudson Trevor    male   0.9167      1
2       1         0    Allison, Miss. Helen Loraine  female   2.0000      1

   parch  ticket      fare    cabin embarked boat body  \
0      0   24160  211.3375       B5        S    2    ?
1      2  113781  151.5500  C22 C26        S   11    ?
2      2  113781  151.5500  C22 C26        S    ?    ?

                      home.dest
0                    St Louis, MO
```

```
1  Montreal, PQ / Chesterville, ON
2  Montreal, PQ / Chesterville, ON
```

[42]:
```
titanic.set_index("cabin",inplace=True)
```

[43]:
```
titanic.head(2)
```

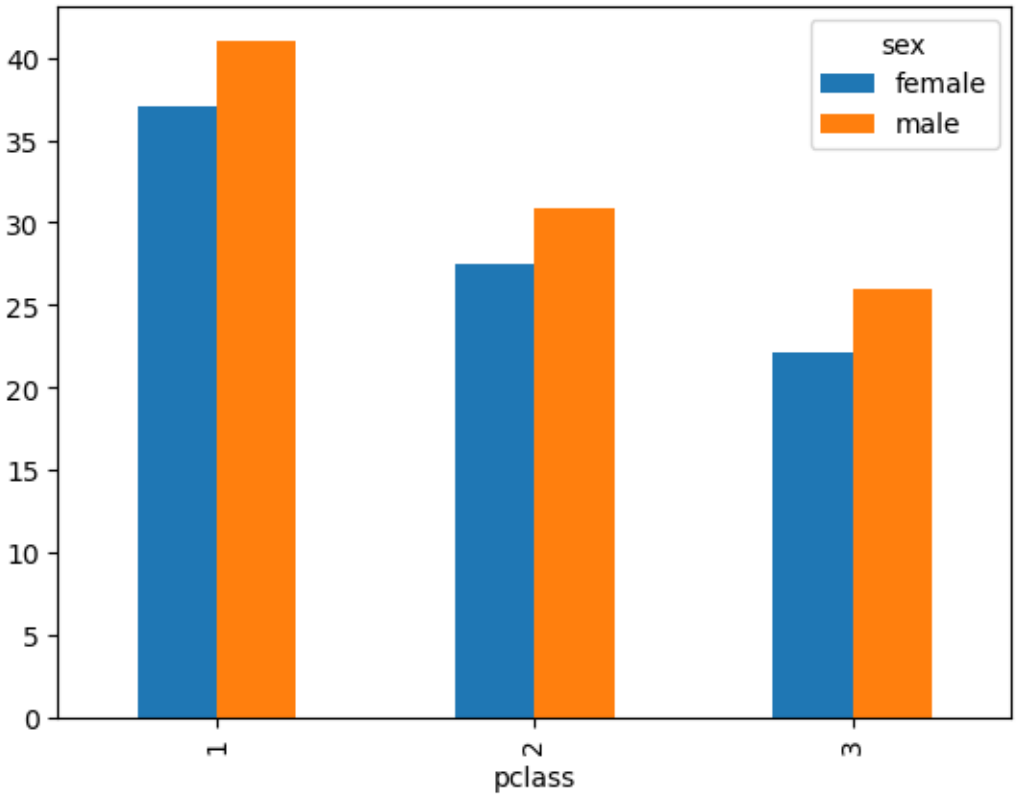
[43]:
```
         survived                             name      age  sibsp  parch  \
cabin
B5              1   Allen, Miss. Elisabeth Walton  29.0000      0      0
C22 C26         1  Allison, Master. Hudson Trevor   0.9167      1      2

         ticket      fare embarked boat body                        home.dest
cabin
B5        24160  211.3375        S    2    ?                     St Louis, MO
C22 C26  113781  151.5500        S   11    ?  Montreal, PQ / Chesterville, ON
```

[45]:
```
titanic.index.get_level_values(0)
```

[45]:
```
Index(['B5', 'C22 C26', 'C22 C26', 'C22 C26', 'C22 C26', 'E12', 'D7', 'A36',
       'C101', '?',
       …
       '?', '?', '?', '?', '?', '?', '?', '?', '?', '?'],
      dtype='object', name='cabin', length=1309)
```

[50]:
```
titanic.groupby(["pclass", "sex"])["age"].mean().unstack().plot(kind="bar")
```

[50]:
```
<Axes: xlabel='pclass'>
```



13 Working with text

[52]:
```
titanic["name"].str.upper()
```

[52]:
```
0                          ALLEN, MISS. ELISABETH WALTON
1                        ALLISON, MASTER. HUDSON TREVOR
2                          ALLISON, MISS. HELEN LORAINE
3                  ALLISON, MR. HUDSON JOSHUA CREIGHTON
4       ALLISON, MRS. HUDSON J C (BESSIE WALDO DANIELS)
                               ...
1304                             ZABOUR, MISS. HILENI
1305                            ZABOUR, MISS. THAMINE
1306                        ZAKARIAN, MR. MAPRIEDEDER
1307                              ZAKARIAN, MR. ORTIN
1308                               ZIMMERMAN, MR. LEO
Name: name, Length: 1309, dtype: object
```

[53]:
```
titanic["cabin"].str[0]
```

[53]:
```
0       B
1       C
2       C
3       C
4       C
       ..
1304    ?
1305    ?
1306    ?
1307    ?
1308    ?
Name: cabin, Length: 1309, dtype: object
```

[54]:
```
s = pd.Series(['1. Hawk.  ', '2. Pickle!\n', '3. Melonhead?\t'])
```

[55]:
```
s.str.strip(to_strip="123. \n \t")
```

[55]:
```
0          Hawk
1       Pickle!
2    Melonhead?
dtype: object
```

[57]:
```
titanic["home.dest"].str.split("/", expand=True).head(2)
```

[57]:

	0	1	2
0	St Louis, MO	None	None
1	Montreal, PQ	Chesterville, ON	None

[61]:
```
titanic['sex'].str.contains('male')
```

[61]:
```
0       True
1       True
2       True
3       True
4       True
        …
1304    True
1305    True
1306    True
1307    True
1308    True
Name: sex, Length: 1309, dtype: bool
```

14 Apply_Map

[62]:
```
def years_to_days(yrs):
    return yrs*365
titanic["age"].apply(years_to_days)
```

[62]:
```
0       10585.0000
1         334.5955
2         730.0000
3       10950.0000
4        9125.0000
            …
1304     5292.5000
1305           NaN
1306     9672.5000
1307     9855.0000
1308    10585.0000
Name: age, Length: 1309, dtype: float64
```

[63]:
```
titanic["age"] * 365
```

[63]:
```
0       10585.0000
1         334.5955
2         730.0000
3       10950.0000
4        9125.0000
            …
1304     5292.5000
1305           NaN
1306     9672.5000
1307     9855.0000
1308    10585.0000
Name: age, Length: 1309, dtype: float64
```

[64]:
```
def convert_currency(num, multiplier):
    return f"${num*multiplier}"
titanic["fare"].apply(convert_currency, args=(24,))
```

[64]:
```
0                   $5072.1
1       $3637.2000000000003
2       $3637.2000000000003
3       $3637.2000000000003
4       $3637.2000000000003
                …
1304              $346.9008
1305              $346.9008
1306    $173.39999999999998
```

```
1307    $173.39999999999998
1308                  $189.0
Name: fare, Length: 1309, dtype: object
```

[65]:
```
titanic["age"].map(lambda a: a < 18)
```

[65]:
```
0       False
1        True
2        True
3       False
4       False
        …
1304     True
1305    False
1306    False
1307    False
1308    False
Name: age, Length: 1309, dtype: bool
```

[68]:
```
titanic[["name", "sex"]].applymap(str.upper).head(2)
```

[68]:
```
                              name     sex
0   ALLEN, MISS. ELISABETH WALTON  FEMALE
1  ALLISON, MASTER. HUDSON TREVOR    MALE
```

15 Merging dataframes

[72]:
```
s1 = pd.Series(['a', 'b', 'c'])
s2 = pd.Series(['d', 'e', 'f', 'z'])
pd.concat([s1,s2],ignore_index=True)
```

[72]:
```
0    a
1    b
2    c
3    d
4    e
5    f
6    z
dtype: object
```

[73]:
```
fruits = pd.Series(
    data=["apple", "banana", "cherry", "durian"],
    index=["a", "b", "c", "d"]
)

animals = pd.Series(
```

```
    data=["badger", "cougar", "anaconda", "elk", "pika"],
    index=["b", "c", "a", "e", "p"]
)
pd.concat([animals, fruits], axis=1, join="inner",keys=['animal','fruit'])
```

[73]:

	animal	fruit
b	badger	banana
c	cougar	cherry
a	anaconda	apple

[74]:
```
teams = pd.DataFrame(
    [
        ["Suns", "Phoenix", 20, 4],
        ["Mavericks", "Dallas", 11,12],
        ["Rockets", "Houston", 7, 16],
        ['Nuggets', "Denver", 11, 12]
    ],
    columns=["team", "city", "wins", "losses"]
)
cities = pd.DataFrame(
    [
        ["Houston", "Texas", 2310000],
        ["Phoenix", "Arizona", 1630000],
        ["San Diego", "California", 1410000],
        ["Dallas", "Texas", 1310000]
    ],
    columns=["city", "state", "population"]
)
teams.merge(cities, on="city", how="inner")
```

[74]:

	team	city	wins	losses	state	population
0	Suns	Phoenix	20	4	Arizona	1630000
1	Mavericks	Dallas	11	12	Texas	1310000
2	Rockets	Houston	7	16	Texas	2310000

[77]:
```
sns.set_theme()
```

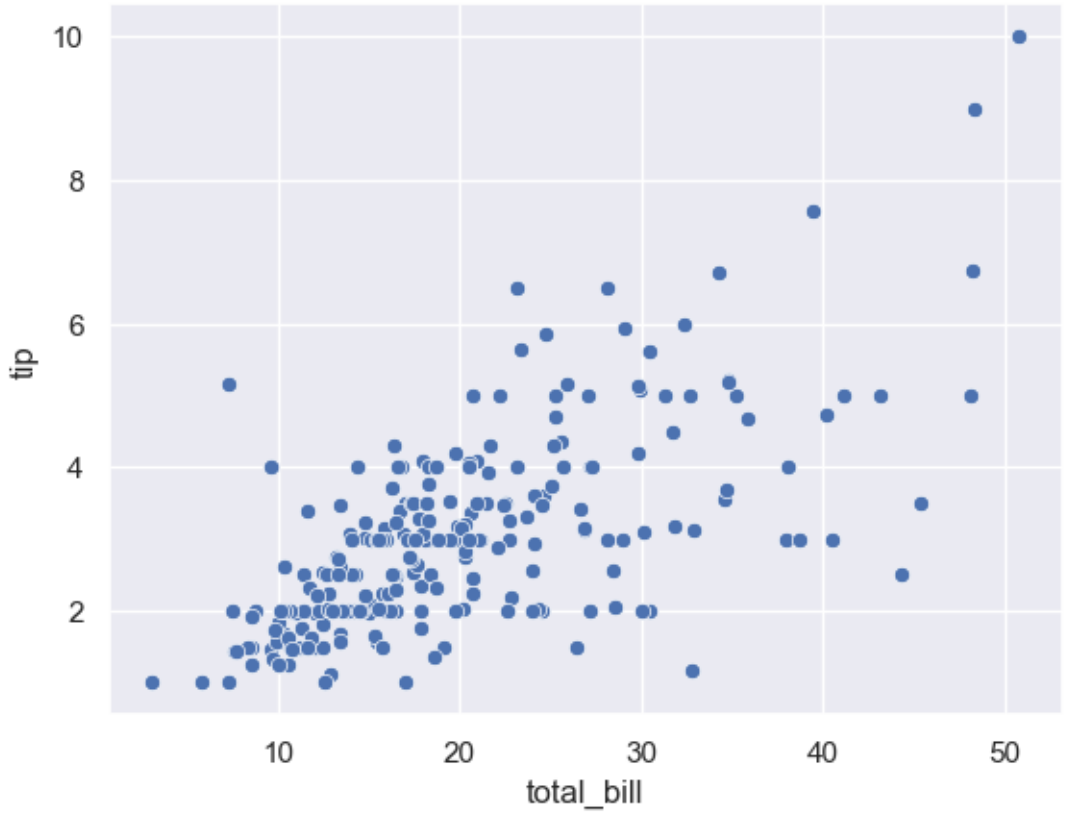
[79]:
```
tips = sns.load_dataset("tips")
penguins = sns.load_dataset("penguins")
tips.head(3)
```

[79]:

	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

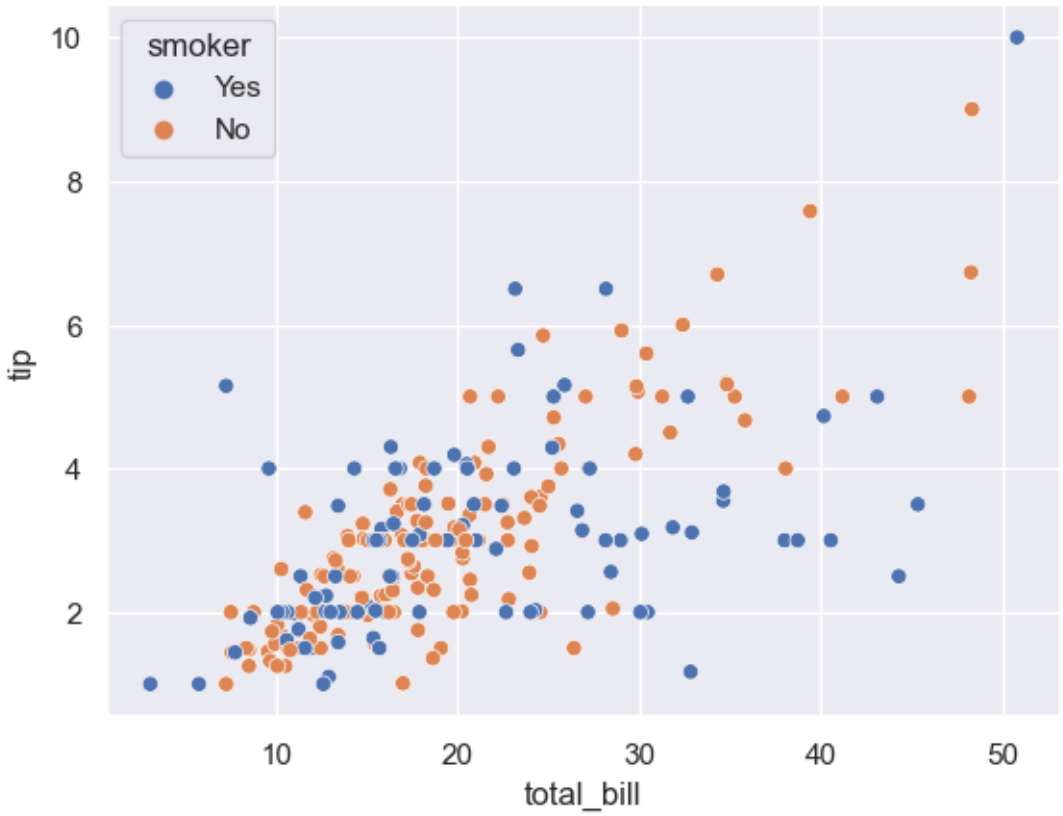
[80]:
```
sns.scatterplot(data=tips, x="total_bill", y="tip")
```

[80]: <Axes: xlabel='total_bill', ylabel='tip'>



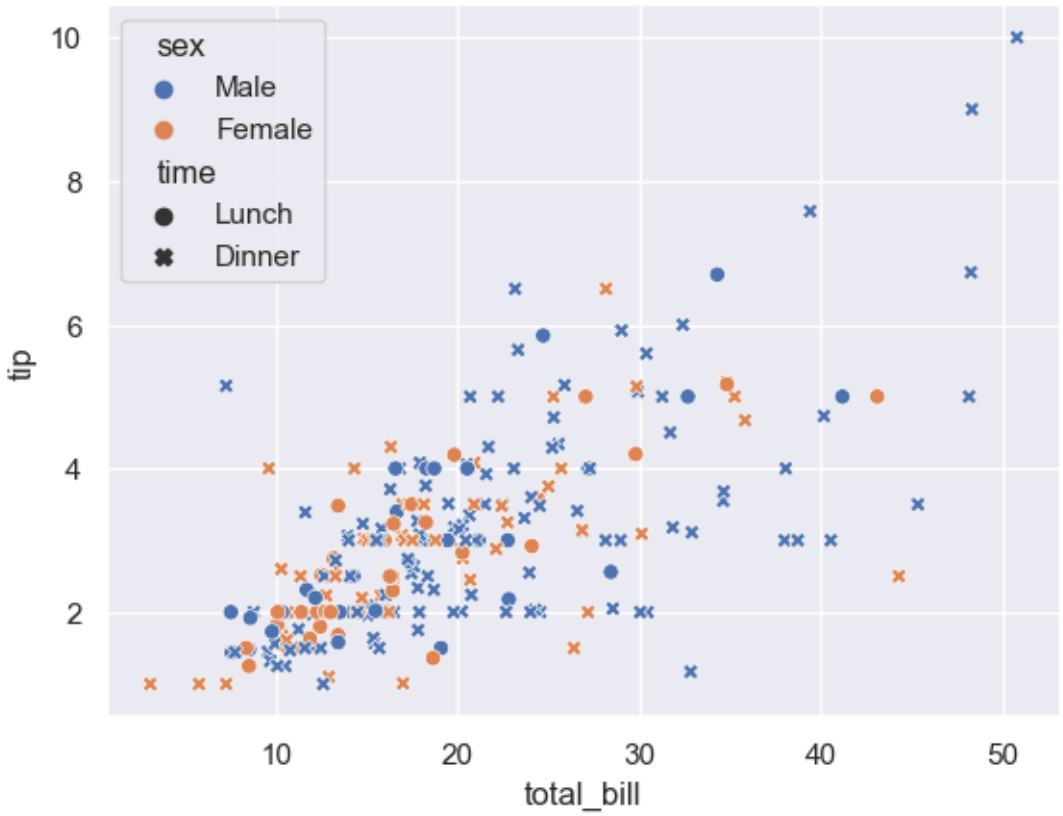
[81]:
```
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="smoker")
```

[81]: <Axes: xlabel='total_bill', ylabel='tip'>



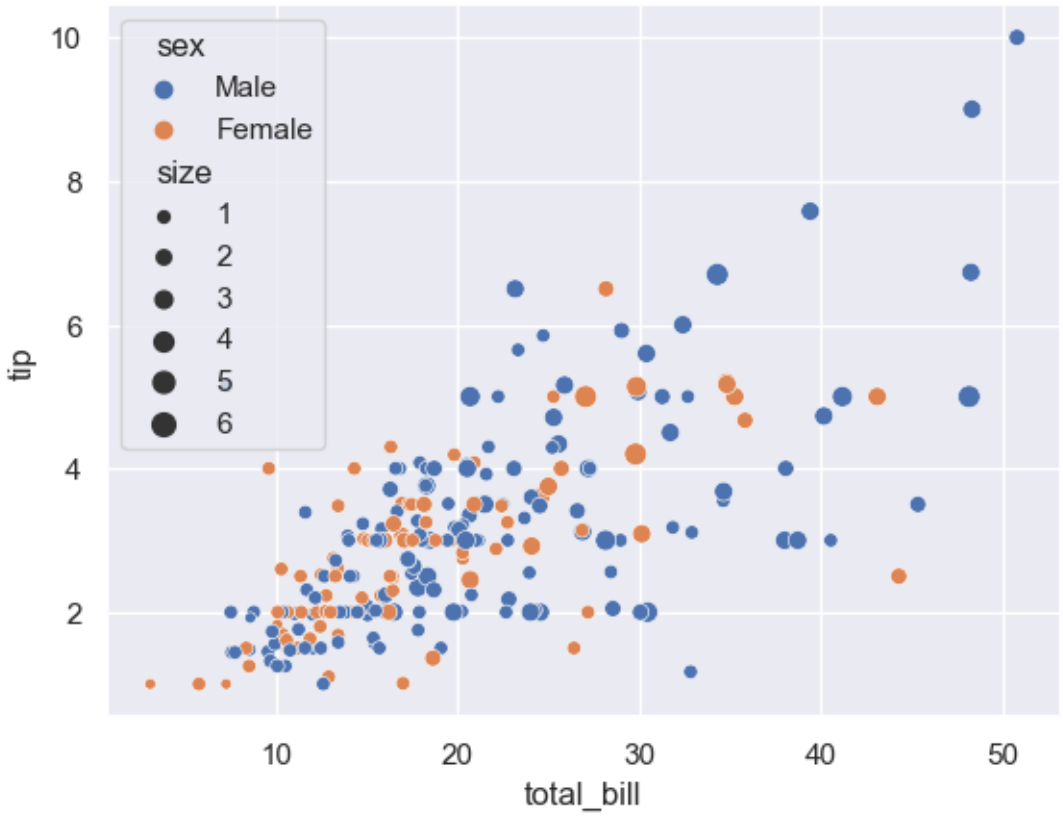
[82]:
```
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="sex", style="time")
```

[82]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



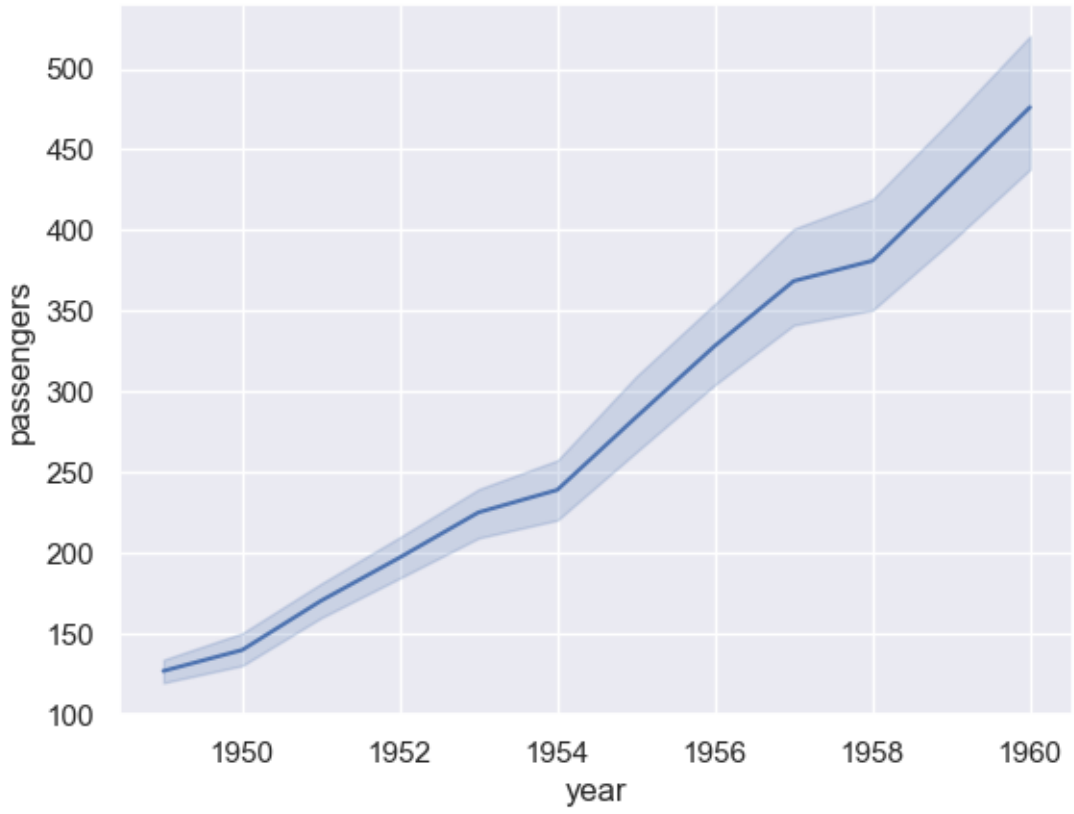
[83]:
```
sns.scatterplot(data=tips, x="total_bill", y="tip", size="size", hue="sex")
```

[83]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```



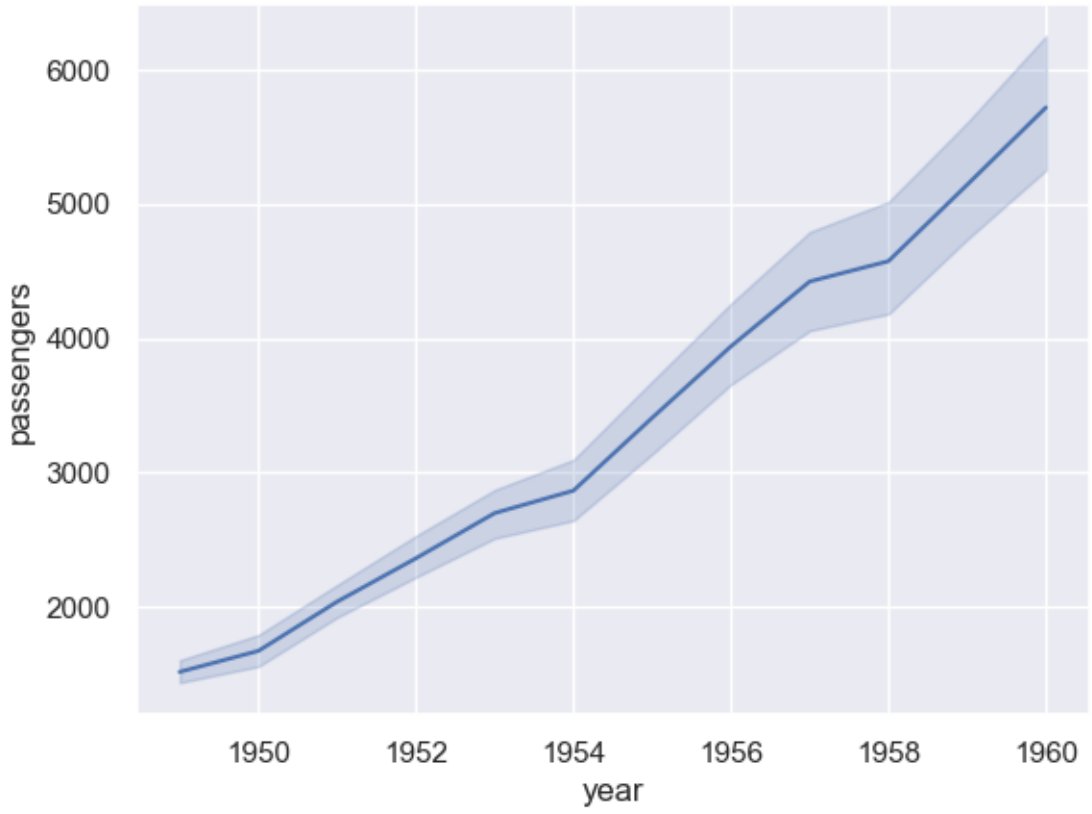
[84]:
```
flights = sns.load_dataset("flights")
sns.lineplot(data=flights, x="year", y="passengers")
```

[84]: <Axes: xlabel='year', ylabel='passengers'>



[85]:
```
sns.lineplot(data=flights, x="year", y="passengers", estimator="sum")
```

[85]:
```
<Axes: xlabel='year', ylabel='passengers'>
```



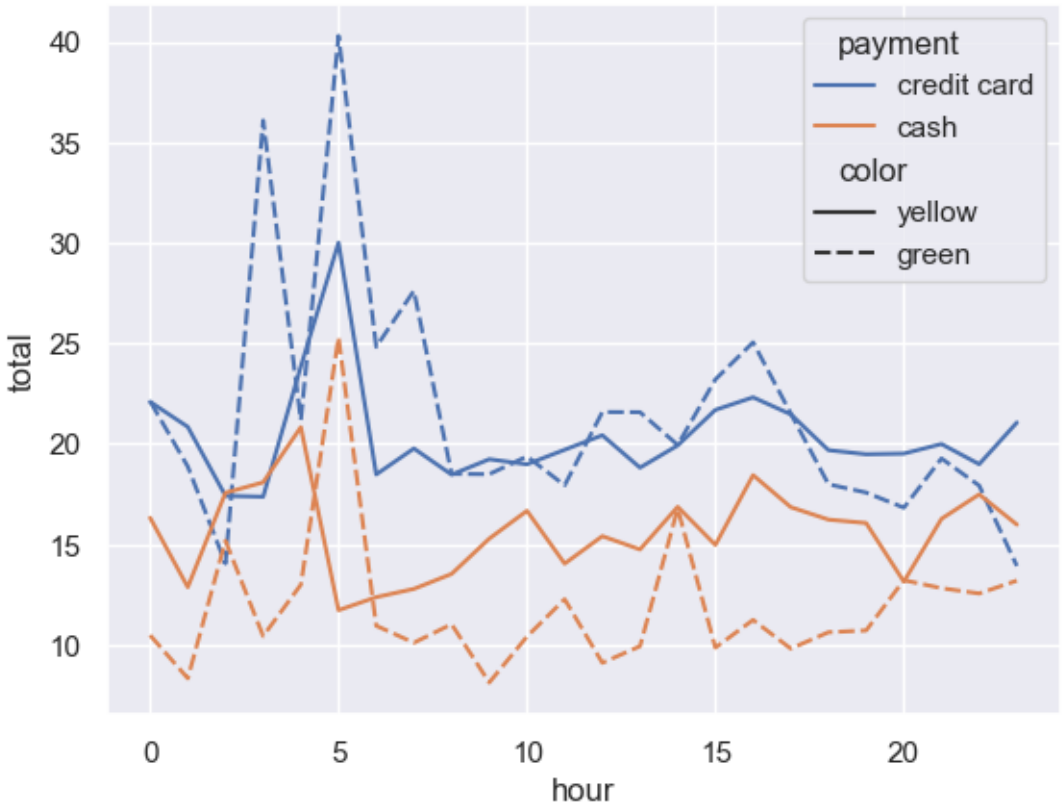
[86]:
```
trips= sns.load_dataset("taxis", parse_dates=["pickup", "dropoff"])
trips["hour"] = trips["pickup"].dt.hour
tips.head(3)
```

[86]:

	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

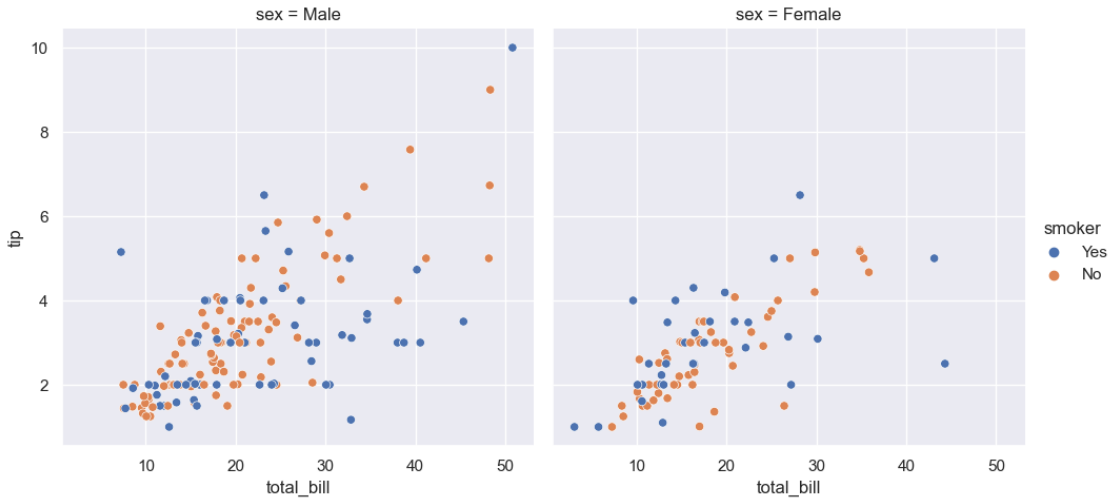
[88]:
```
sns.lineplot(data=trips, x="hour", y="total",
             hue="payment", style="color", errorbar=None)
```

[88]:
```
<Axes: xlabel='hour', ylabel='total'>
```



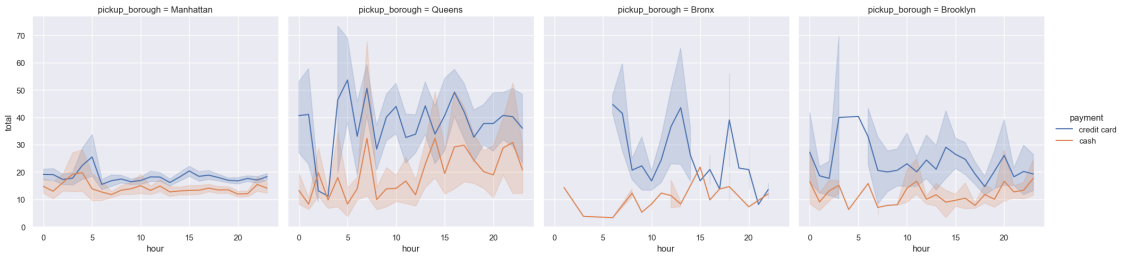
[89]:
```
sns.relplot(data=tips, x="total_bill", y="tip",
            kind="scatter", hue="smoker", col="sex")
```

[89]: <seaborn.axisgrid.FacetGrid at 0x211c72d0b10>



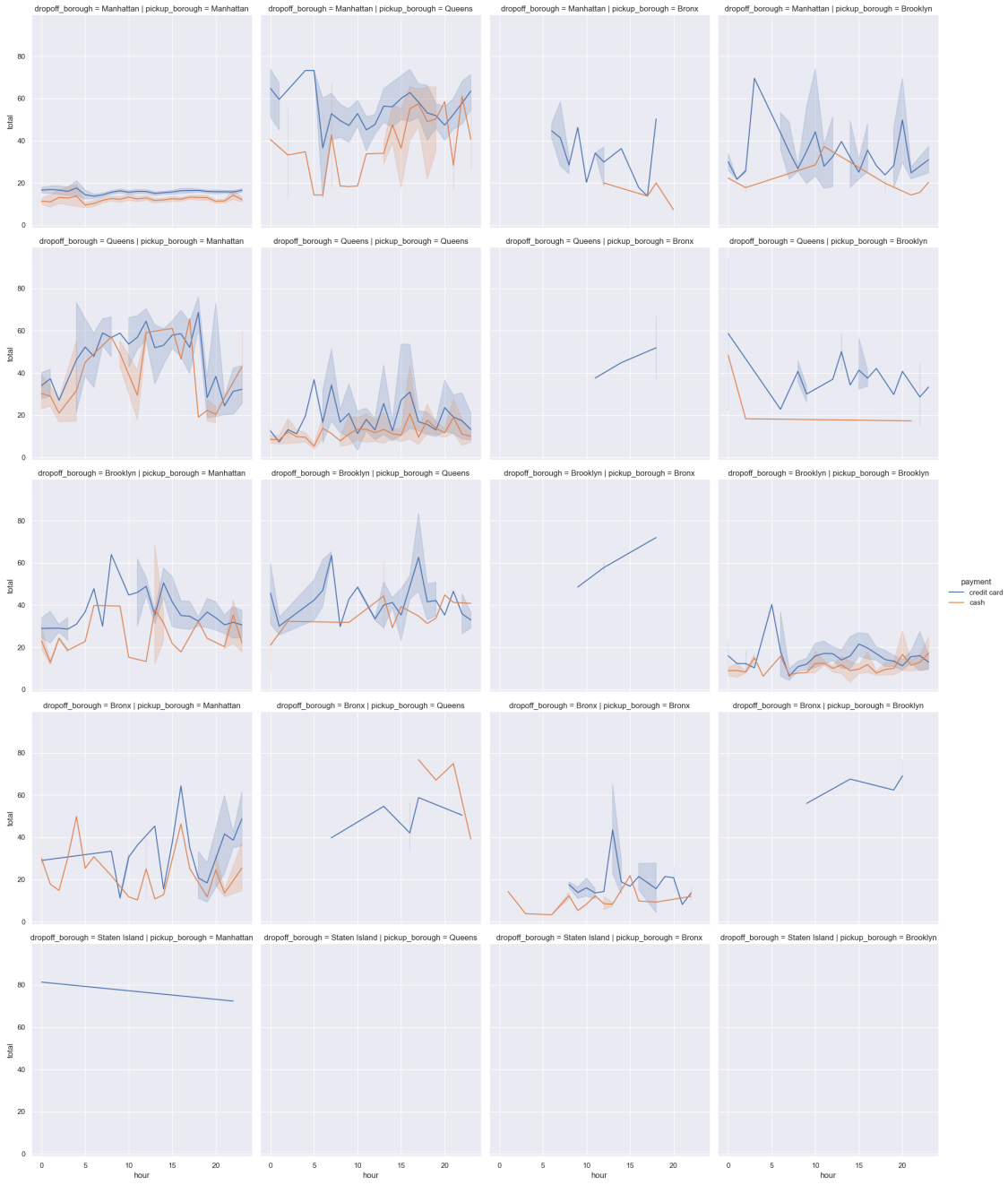
[90]:
```
sns.relplot(data=trips, x="hour", y="total", kind="line",
            col="pickup_borough", hue="payment")
```

[90]: <seaborn.axisgrid.FacetGrid at 0x211c71773d0>



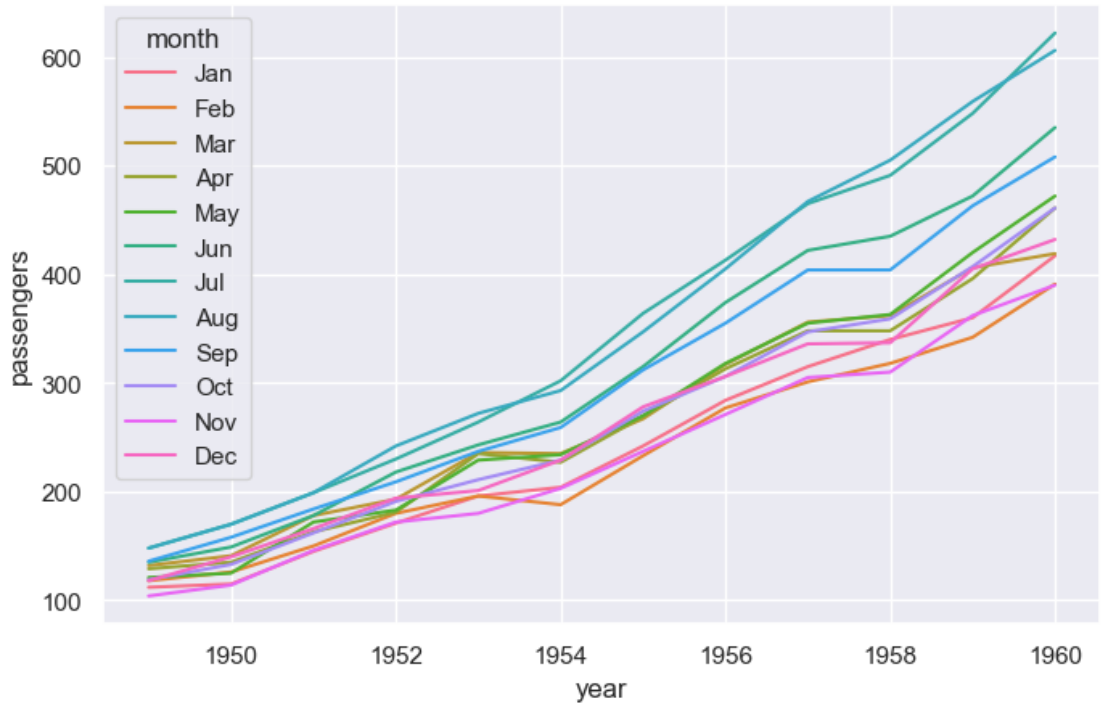
[91]:
```
sns.relplot(
    data=trips,
    x="hour",
    y="total",
    kind="line",
    col="pickup_borough",
    hue="payment",
    row="dropoff_borough"
)
```

[91]: <seaborn.axisgrid.FacetGrid at 0x211c7318750>



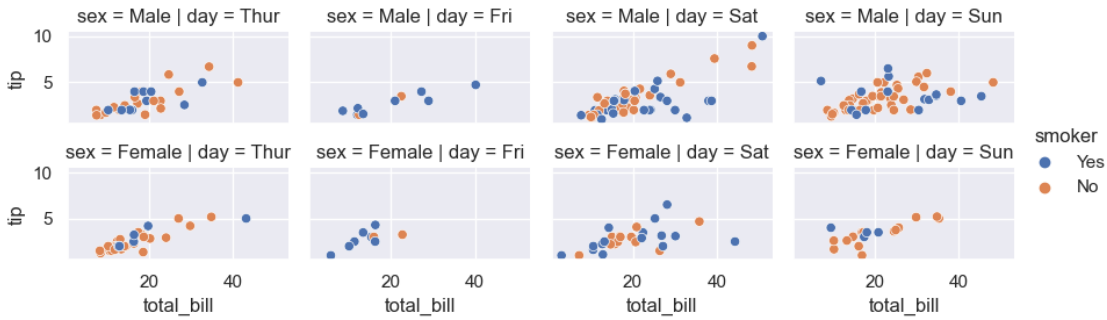
[92]:
```
plt.figure(figsize=(8, 5))
sns.lineplot(data=flights, x="year", y="passengers", hue="month")
```

[92]: <Axes: xlabel='year', ylabel='passengers'>



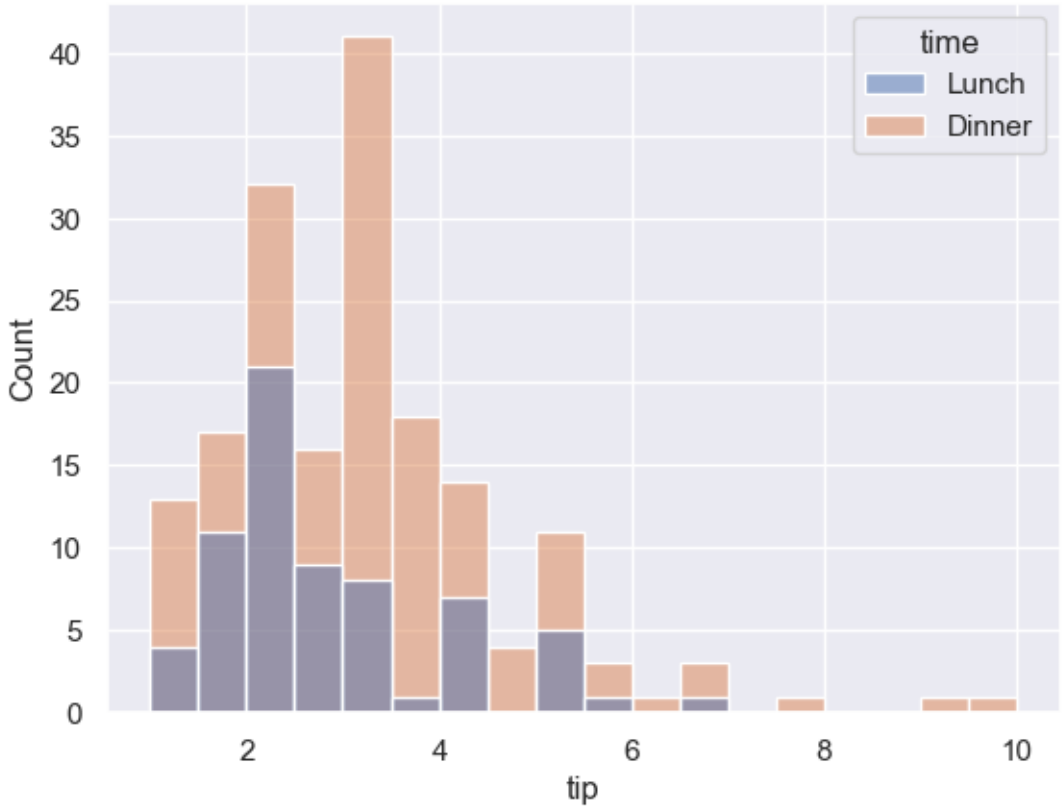
[93]:
```
sns.relplot(
    data=tips,
    x="total_bill",
    y="tip",
    kind="scatter",
    hue="smoker",
    col="day",
    row="sex",
    height=1.5,
    aspect=1.5
)
```

[93]: <seaborn.axisgrid.FacetGrid at 0x211caedfb90>



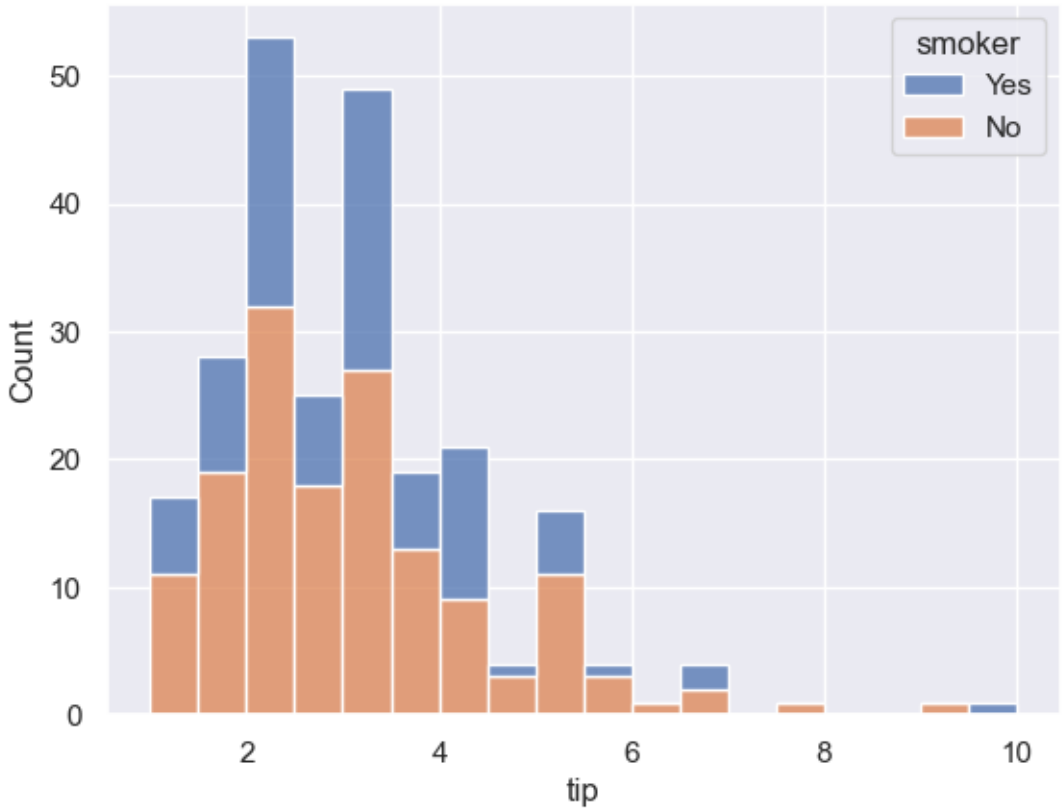
[94]:
```
sns.histplot(data=tips, x="tip", hue="time")
```

[94]: <Axes: xlabel='tip', ylabel='Count'>



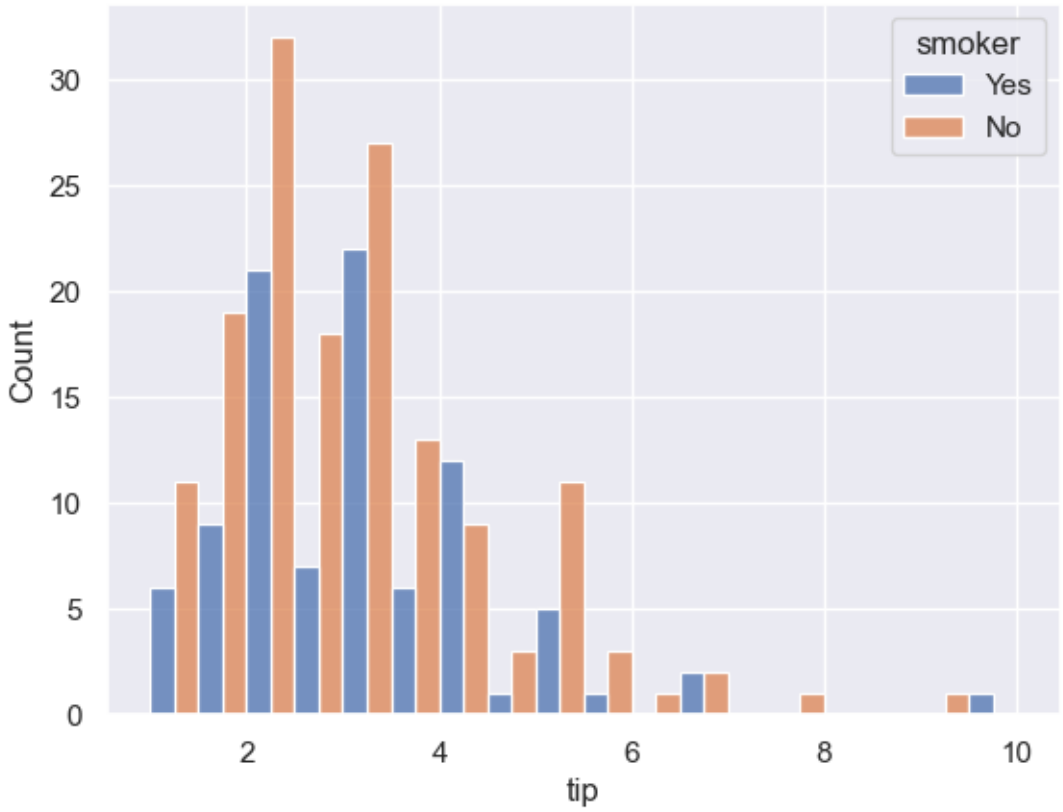
[95]:
```
sns.histplot(data=tips, x="tip", hue="smoker", multiple="stack")
```

[95]: <Axes: xlabel='tip', ylabel='Count'>



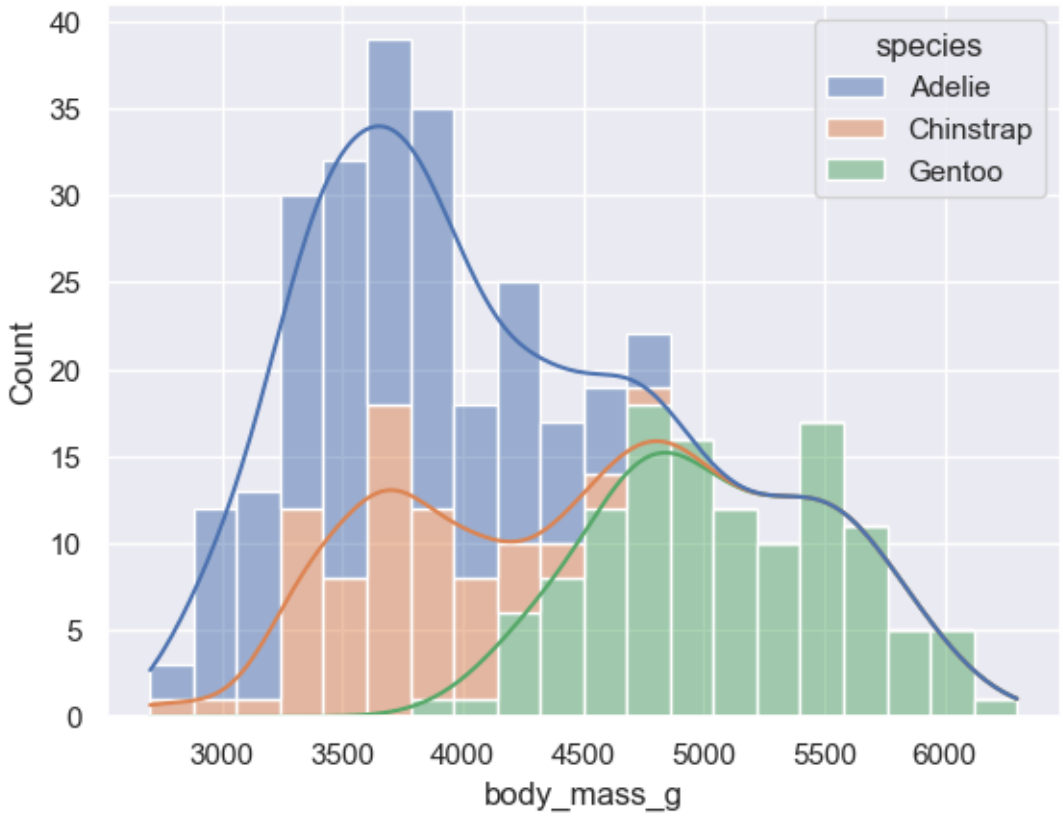
[96]:
```
sns.histplot(data=tips, x="tip", hue="smoker", multiple="dodge")
```

[96]: <Axes: xlabel='tip', ylabel='Count'>



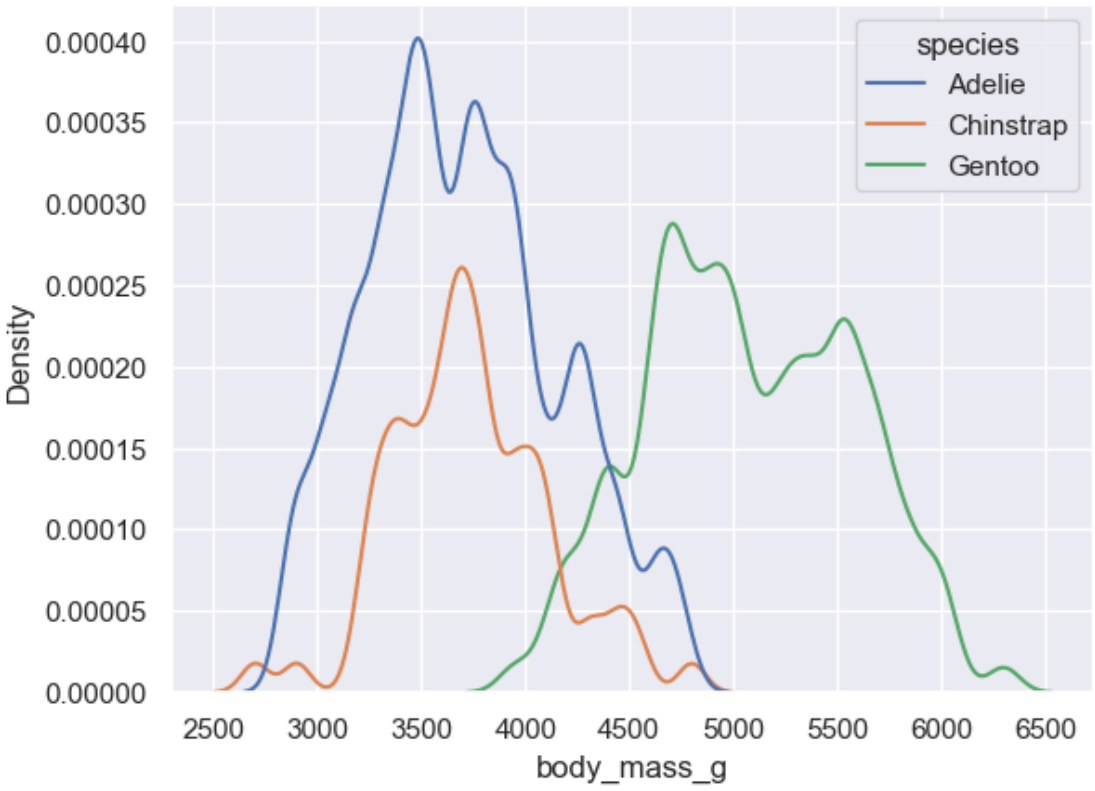
[98]:
```
sns.histplot(data=penguins, x="body_mass_g", bins=20,
             hue="species", multiple="stack", kde=True)
```

[98]: <Axes: xlabel='body_mass_g', ylabel='Count'>



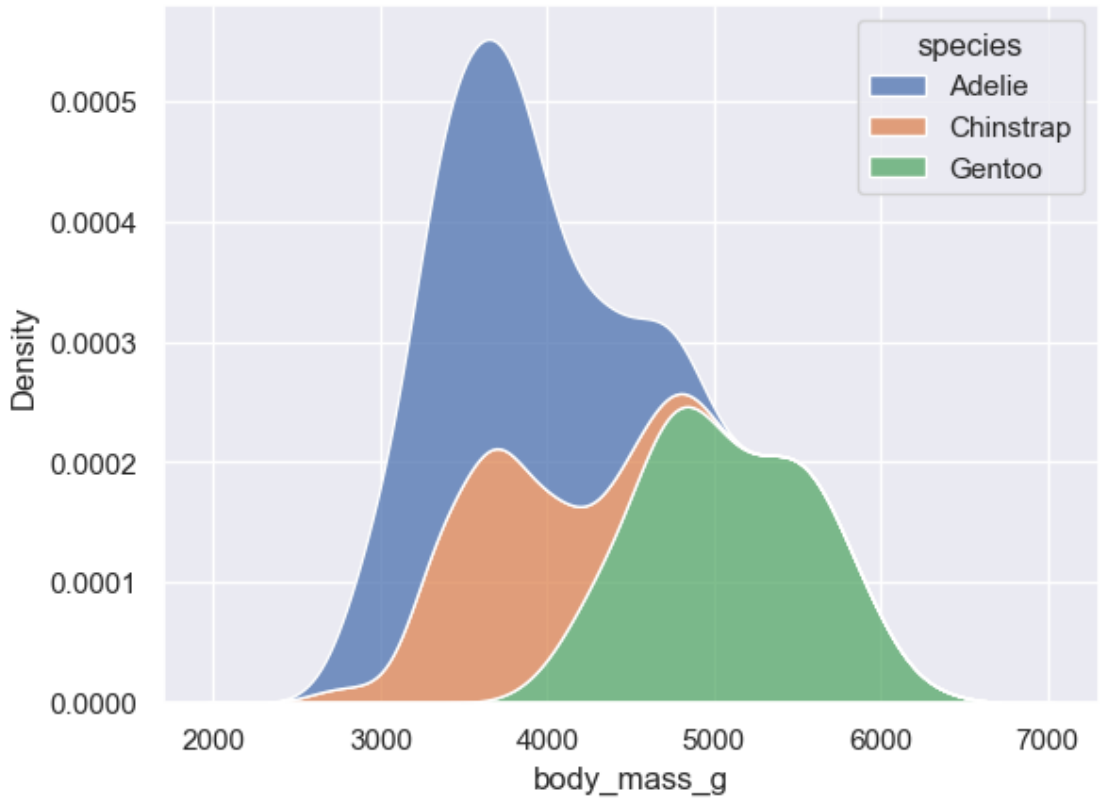
[100]:
```
sns.kdeplot(data=penguins, x="body_mass_g", hue="species", bw_adjust=0.4)
```

[100]: <Axes: xlabel='body_mass_g', ylabel='Density'>



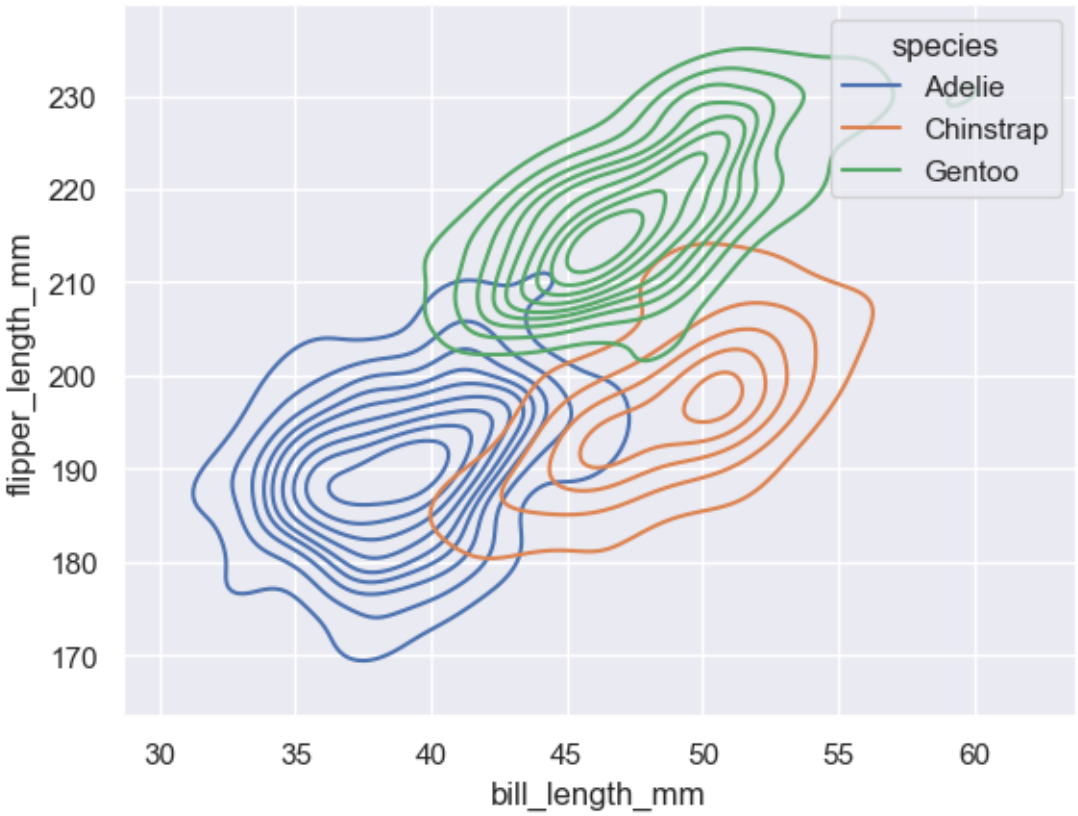
[101]:
```
sns.kdeplot(data=penguins, x="body_mass_g", hue="species", multiple="stack")
```

[101]:
```
<Axes: xlabel='body_mass_g', ylabel='Density'>
```



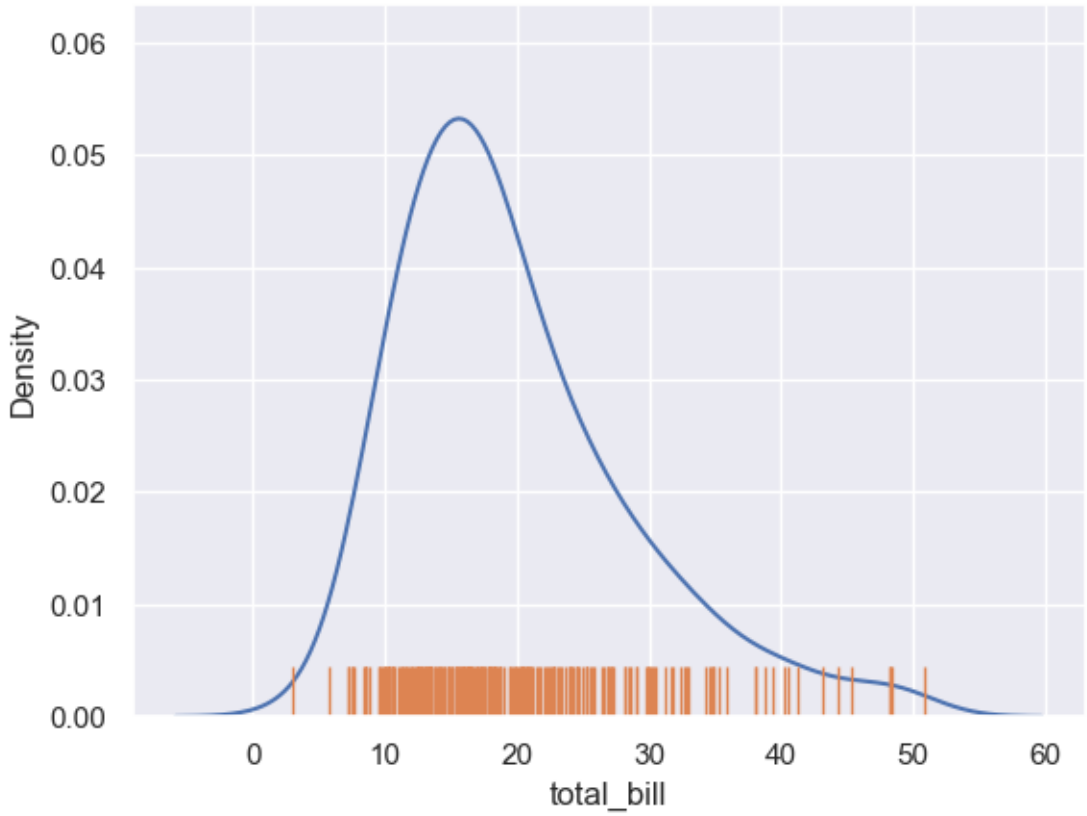
[103]:
```
sns.kdeplot(data=penguins, x="bill_length_mm",
            y="flipper_length_mm", hue="species")
```

[103]: <Axes: xlabel='bill_length_mm', ylabel='flipper_length_mm'>



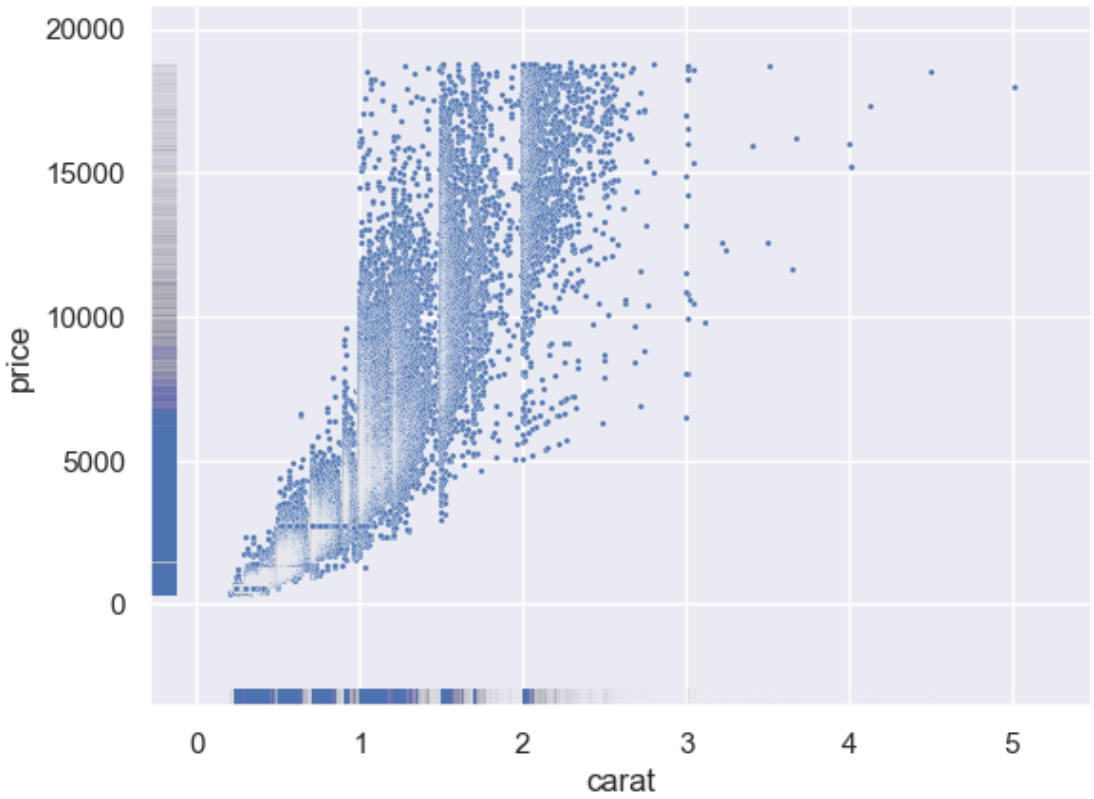
[104]:
```
sns.kdeplot(data=tips, x="total_bill")
sns.rugplot(data=tips, x="total_bill", height=0.07)
```

[104]: <Axes: xlabel='total_bill', ylabel='Density'>



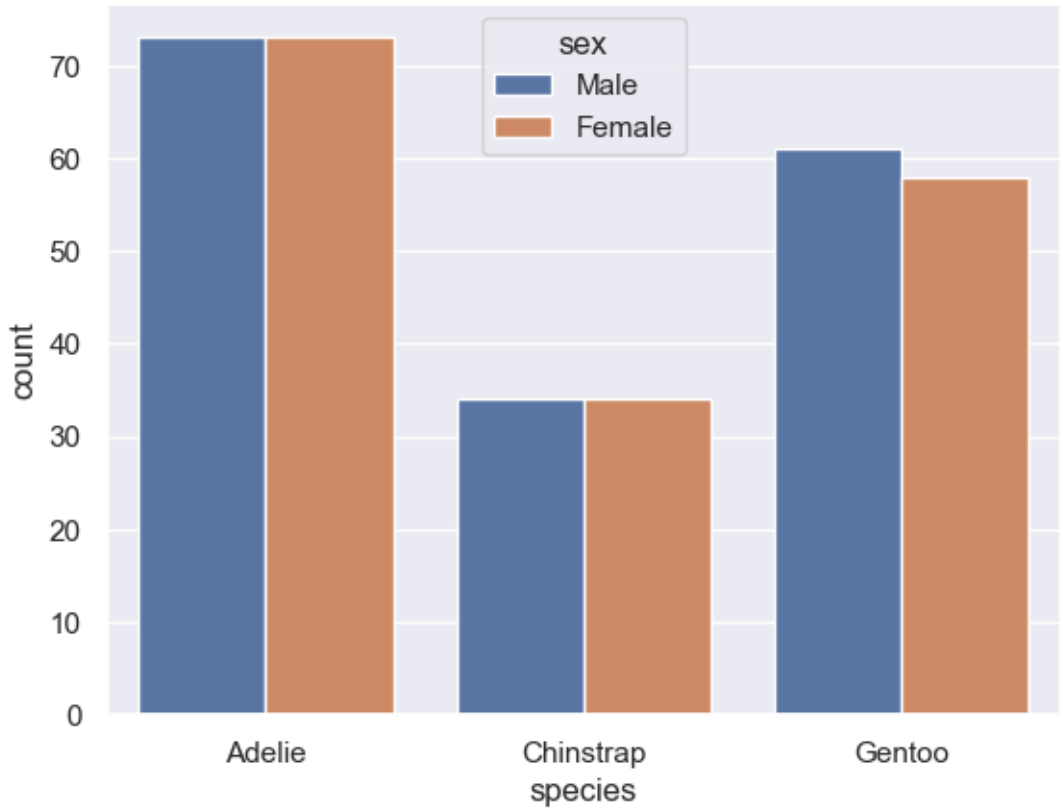
[105]:
```
diamonds = sns.load_dataset("diamonds")
sns.scatterplot(data=diamonds, x="carat", y="price", s=5)
sns.rugplot(data=diamonds, x="carat", y="price", lw=1, alpha=.005)
```

[105]: <Axes: xlabel='carat', ylabel='price'>



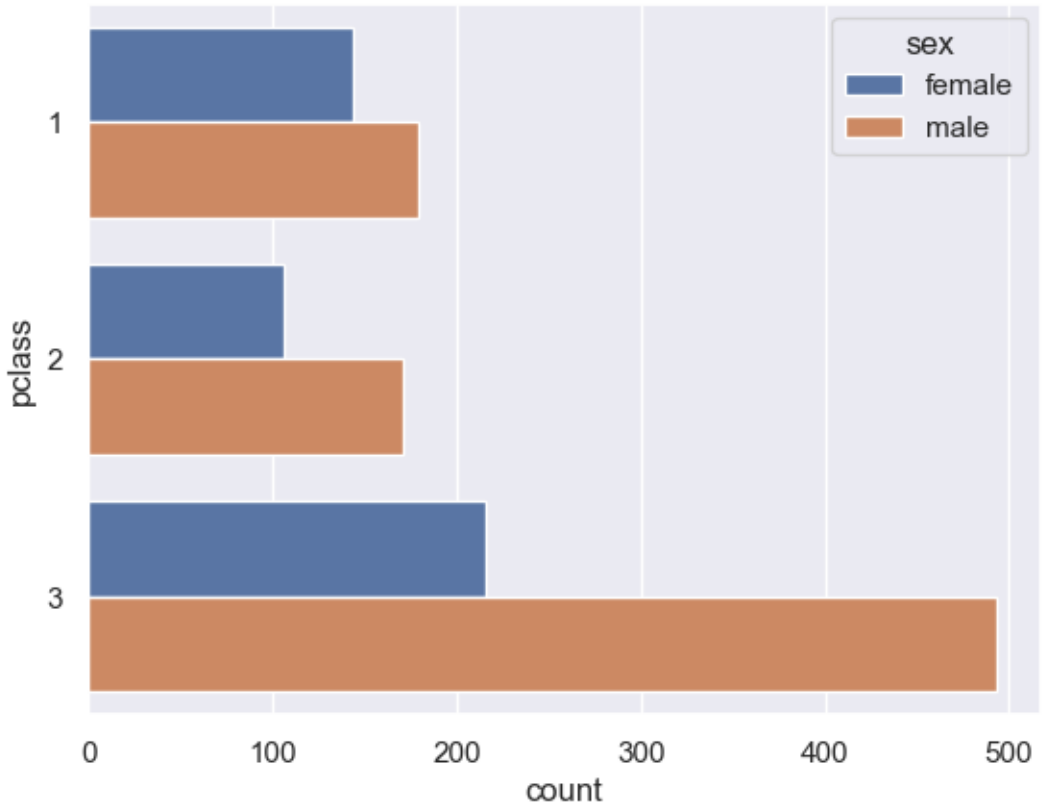
[107]:
```
sns.countplot(data=penguins, x="species", hue="sex")
```

[107]:
```
<Axes: xlabel='species', ylabel='count'>
```



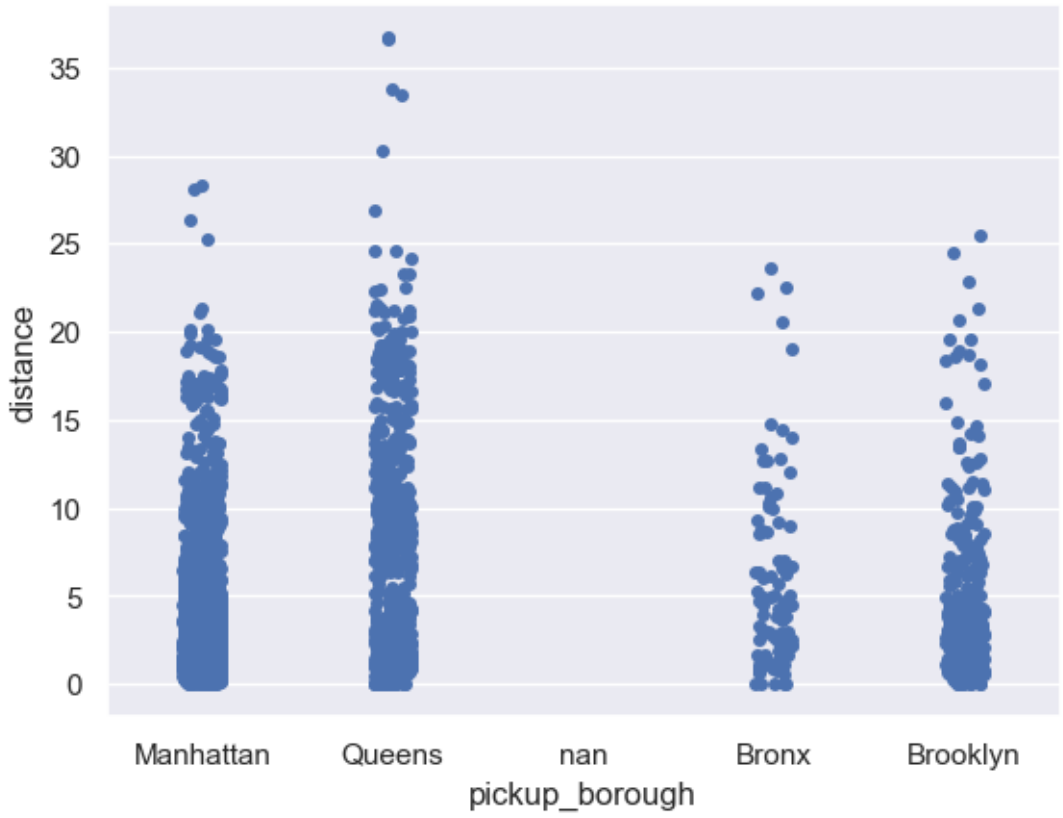
[108]:
```
sns.countplot(data=titanic, y="pclass", hue="sex")
```

[108]:
```
<Axes: xlabel='count', ylabel='pclass'>
```



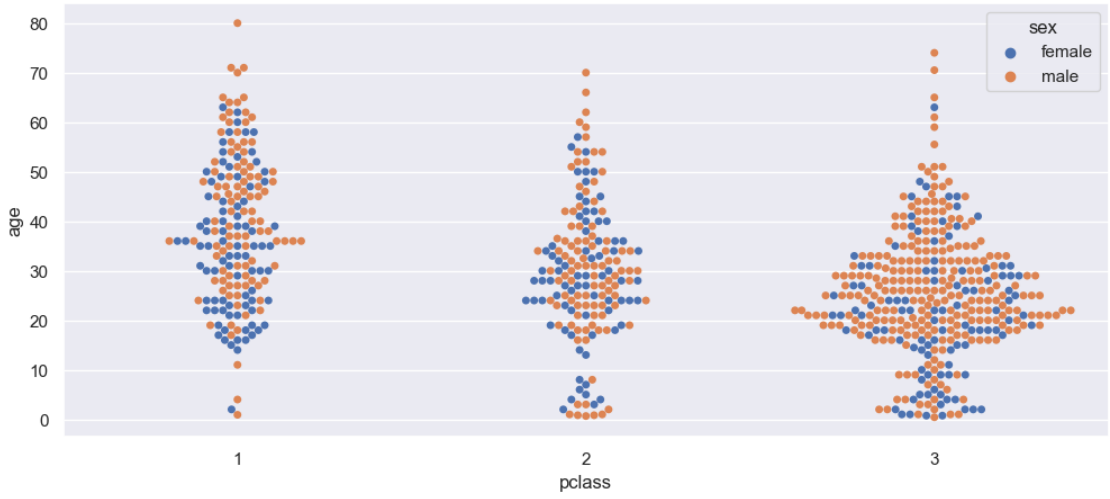
[109]:
```
sns.stripplot(data=trips, x="pickup_borough", y="distance")
```

[109]:
```
<Axes: xlabel='pickup_borough', ylabel='distance'>
```



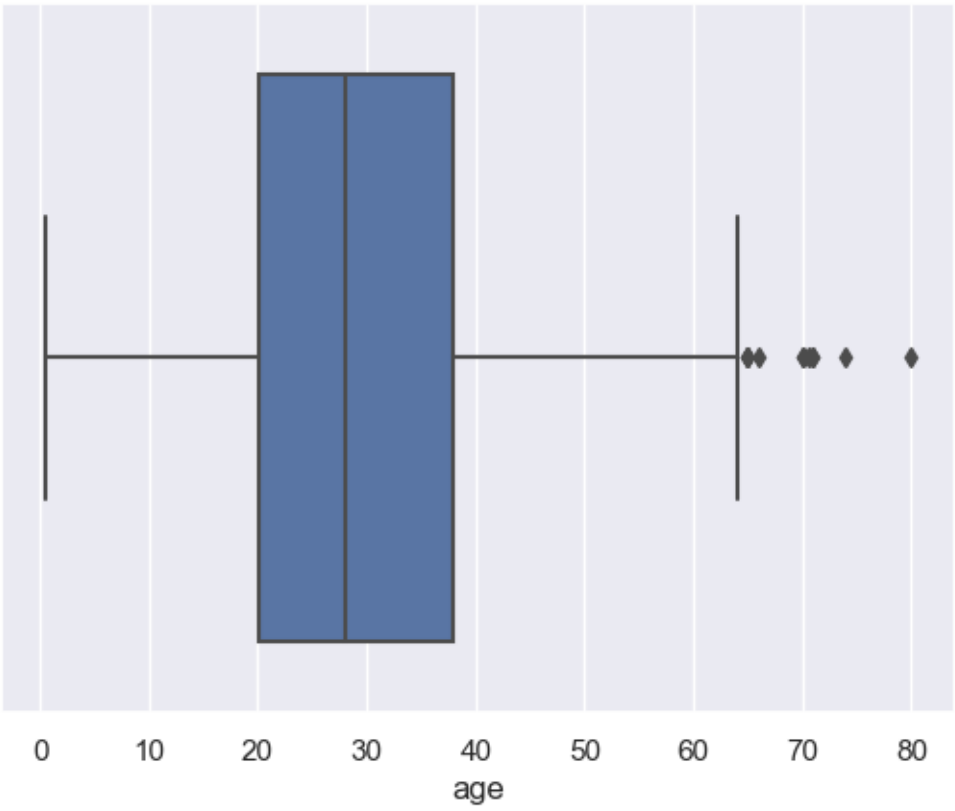
[113]:
```
plt.figure(figsize=(12, 5))
titanic = sns.load_dataset("titanic")
sns.swarmplot(data=titanic, x="pclass", y="age", hue="sex")
```

[113]: <Axes: xlabel='pclass', ylabel='age'>



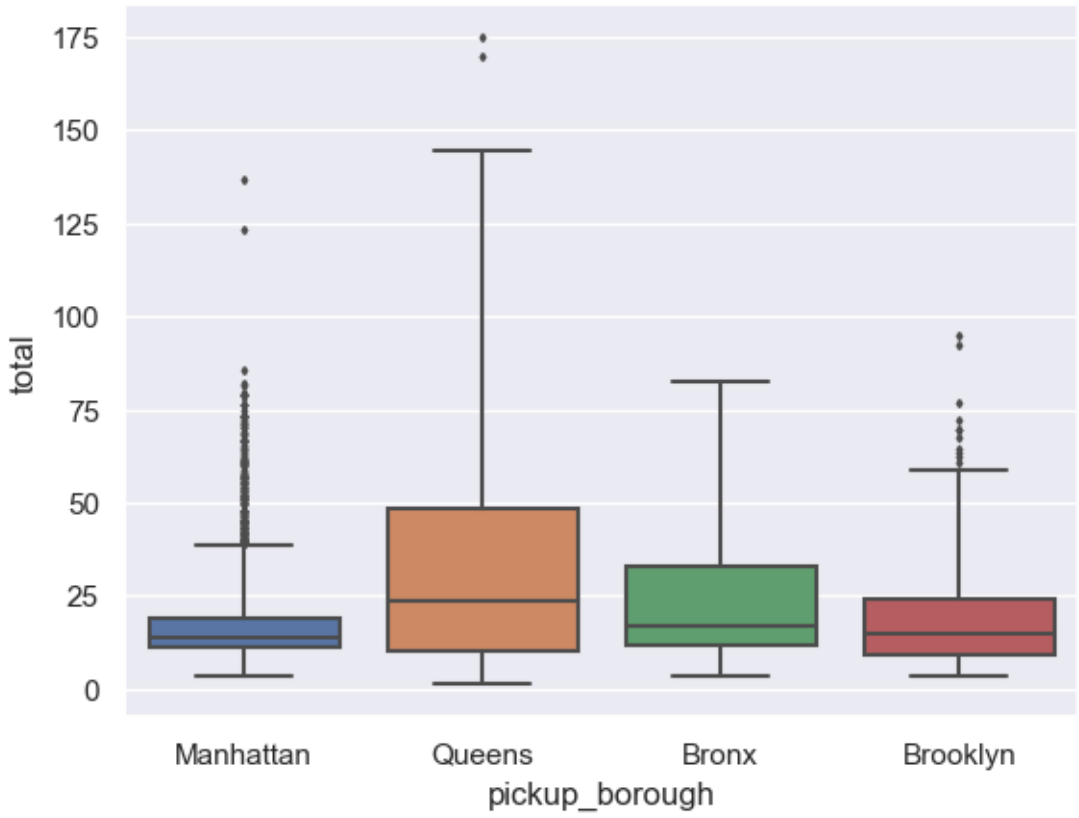
[114]:
```
sns.boxplot(data=titanic, x="age")
```

[114]: <Axes: xlabel='age'>



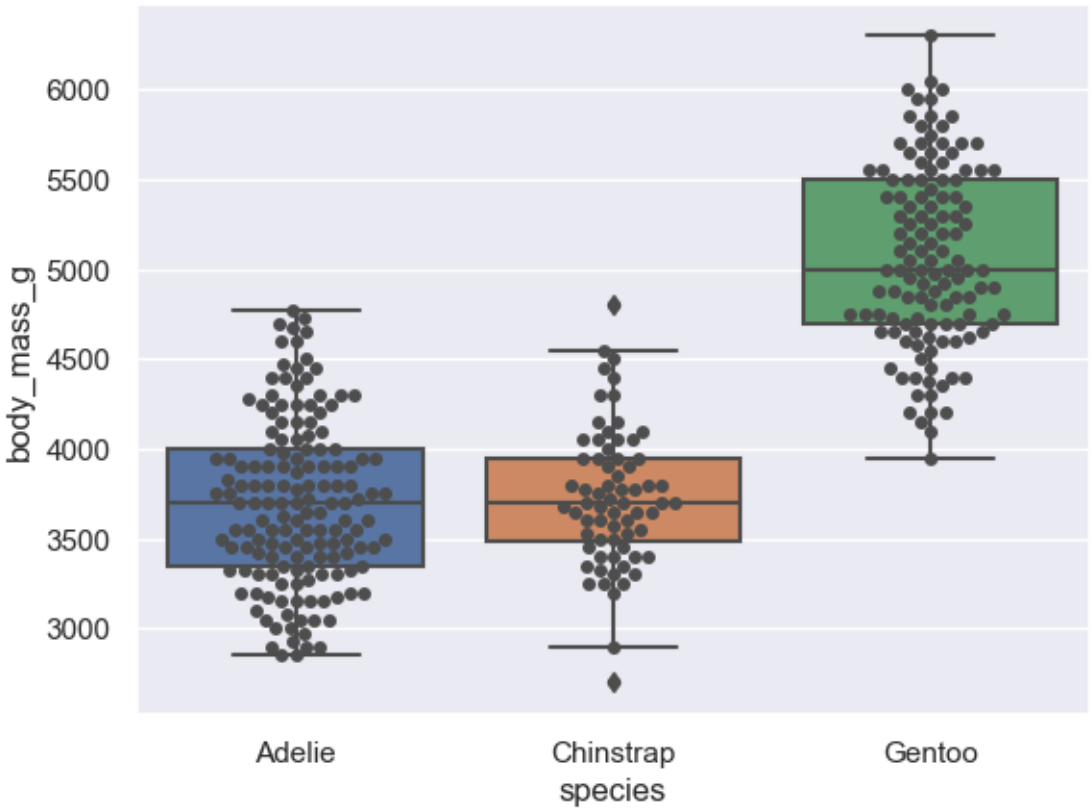
[115]:
```
sns.boxplot(data=trips, x="pickup_borough", y="total", whis=2.5, fliersize=2)
```

[115]: <Axes: xlabel='pickup_borough', ylabel='total'>



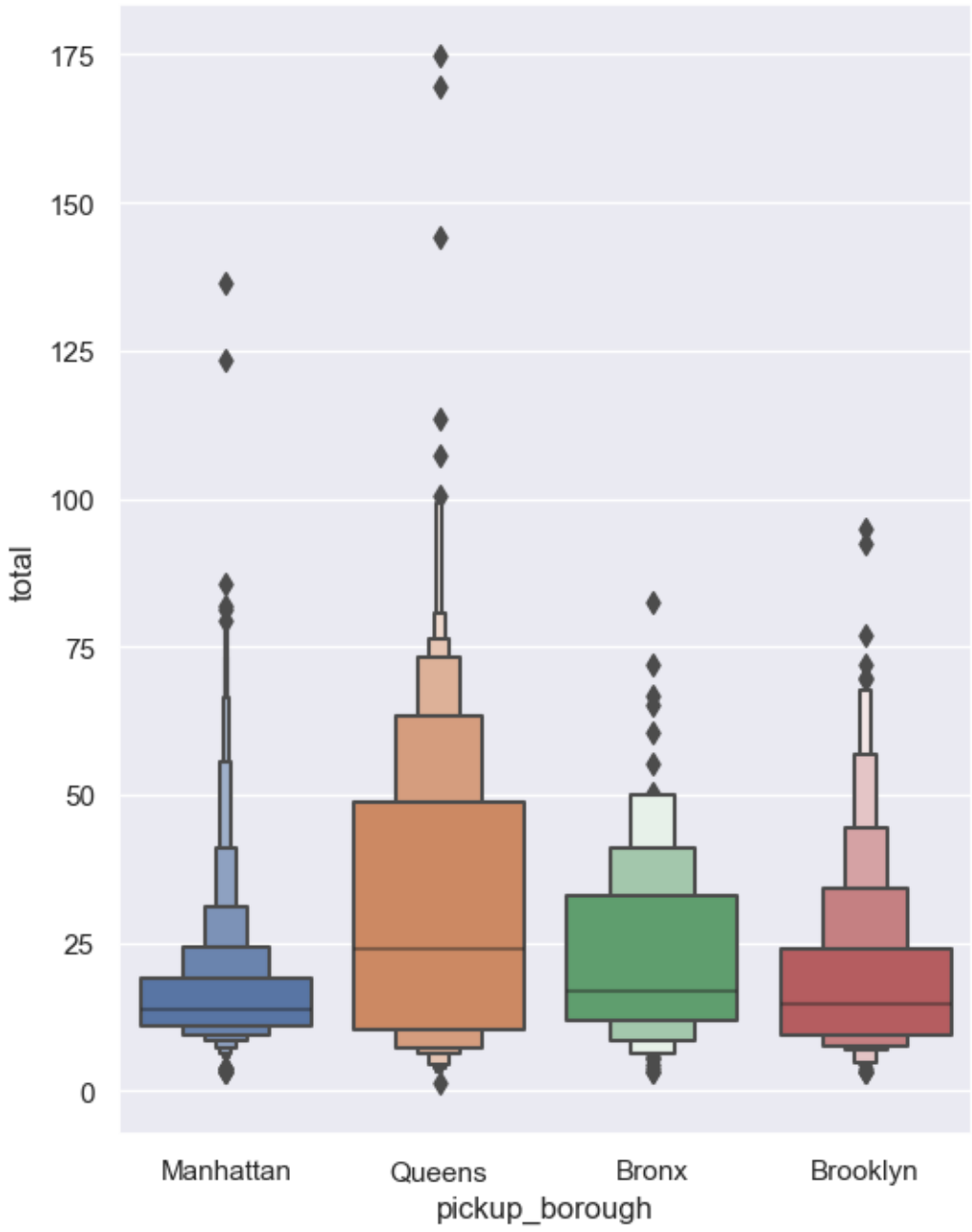
[116]:
```
sns.boxplot(data=penguins, x="species", y="body_mass_g")
sns.swarmplot(data=penguins, x="species", y="body_mass_g", color="0.3")
```

[116]: <Axes: xlabel='species', ylabel='body_mass_g'>



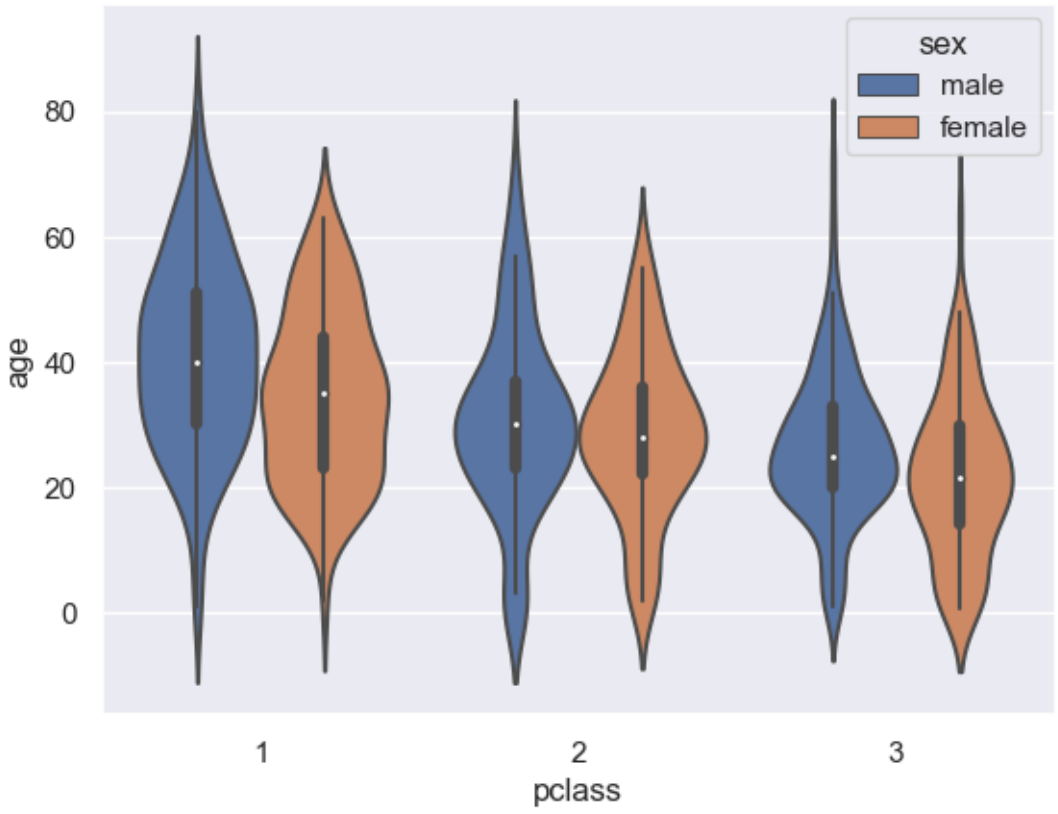
[117]:
```
plt.figure(figsize=(6, 8))
sns.boxenplot(data=trips, x="pickup_borough", y="total")
```

[117]: <Axes: xlabel='pickup_borough', ylabel='total'>



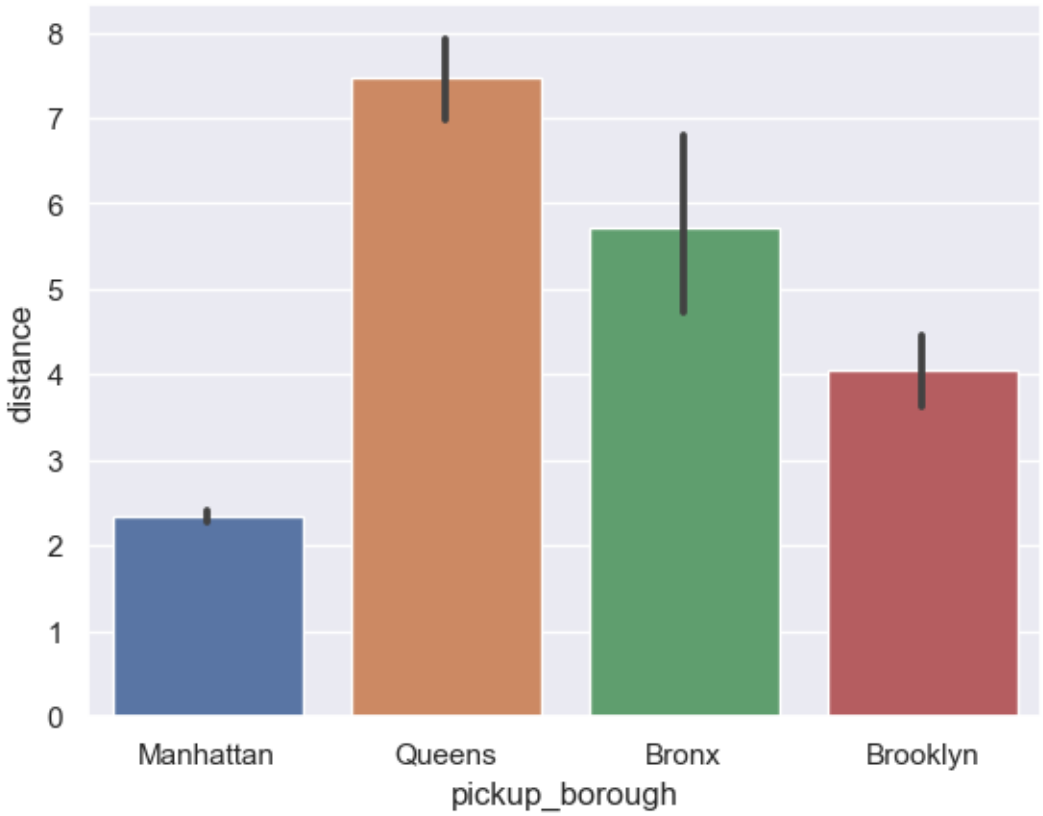
[118]:
```
sns.violinplot(data=titanic, x="pclass", y="age", hue="sex")
```

[118]: <Axes: xlabel='pclass', ylabel='age'>



[119]:
```
sns.barplot(data=trips, x="pickup_borough", y="distance")
```

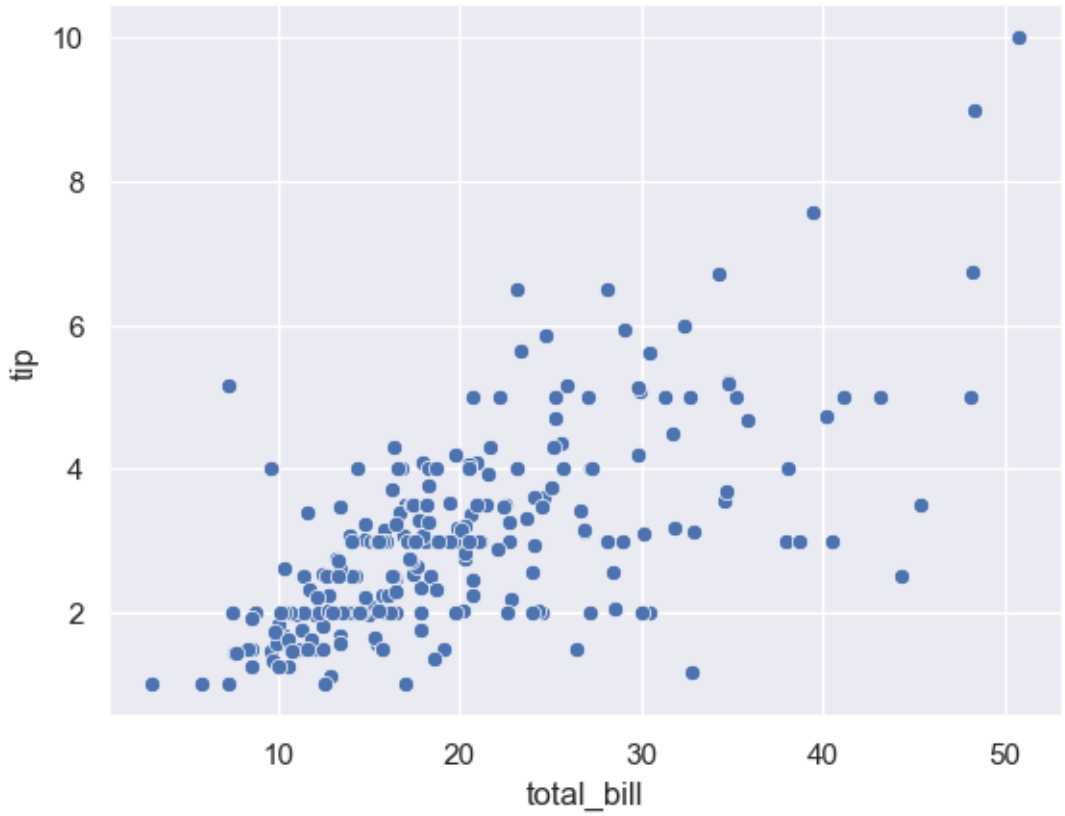
[119]:
```
<Axes: xlabel='pickup_borough', ylabel='distance'>
```



16 Seaborn Aesthetics

[120]:
```
sns.set_style("darkgrid")
sns.scatterplot(data=tips, x="total_bill", y="tip")
```

[120]:
```
<Axes: xlabel='total_bill', ylabel='tip'>
```

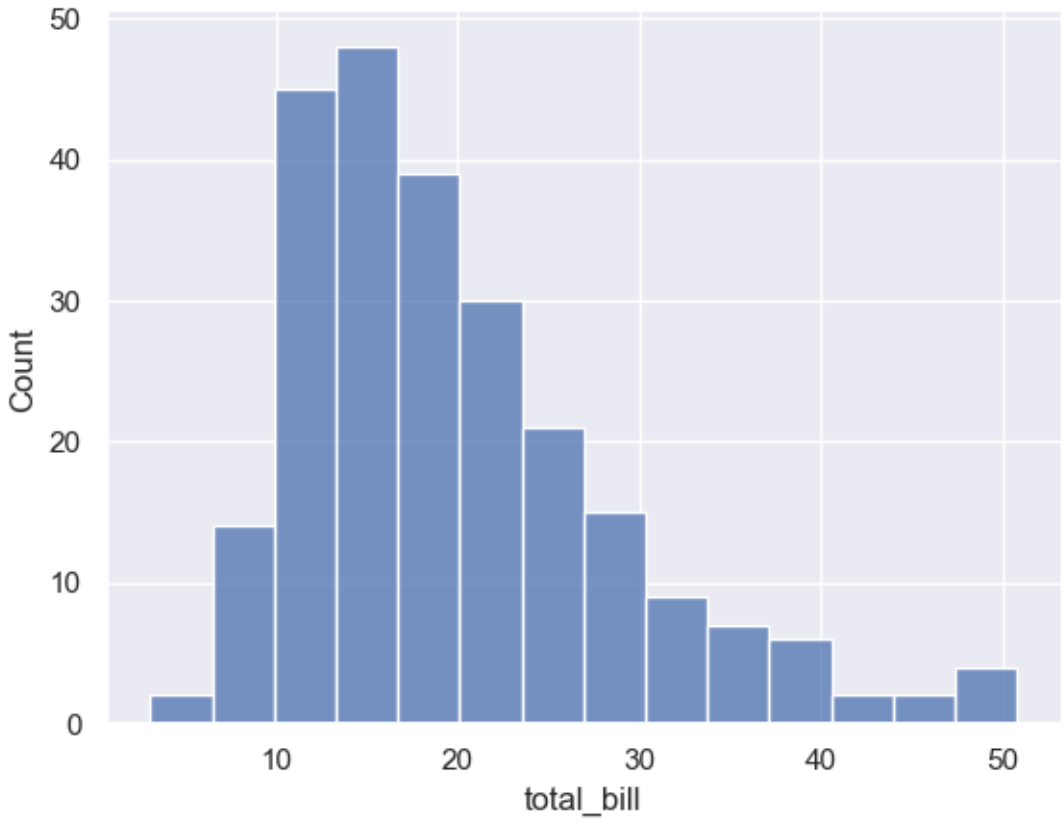


[121]:
```
sns.axes_style()
```

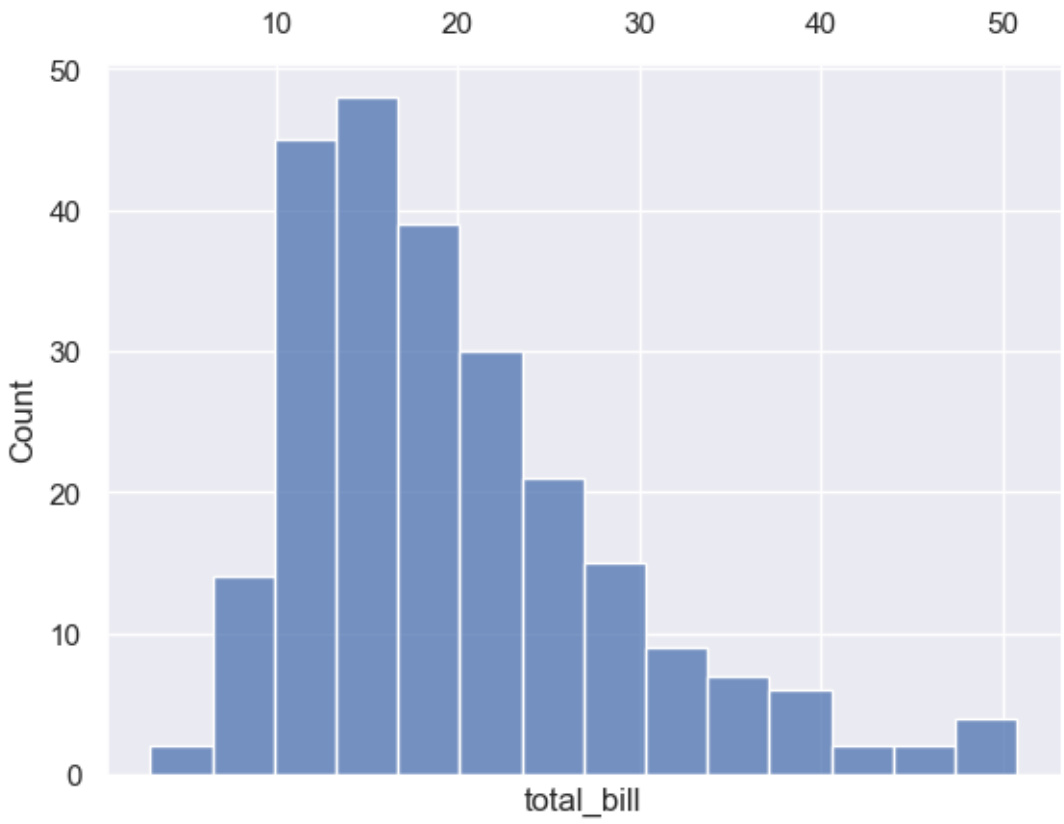
[121]:
```
{'axes.facecolor': '#EAEAF2',
 'axes.edgecolor': 'white',
 'axes.grid': True,
 'axes.axisbelow': True,
 'axes.labelcolor': '.15',
 'figure.facecolor': 'white',
 'grid.color': 'white',
 'grid.linestyle': '-',
 'text.color': '.15',
 'xtick.color': '.15',
 'ytick.color': '.15',
 'xtick.direction': 'out',
 'ytick.direction': 'out',
 'lines.solid_capstyle': <CapStyle.round: 'round'>,
 'patch.edgecolor': 'w',
 'patch.force_edgecolor': True,
 'image.cmap': 'rocket',
 'font.family': ['sans-serif'],
 'font.sans-serif': ['Arial',
```

```
 'DejaVu Sans',
 'Liberation Sans',
 'Bitstream Vera Sans',
 'sans-serif'],
'xtick.bottom': False,
'xtick.top': False,
'ytick.left': False,
'ytick.right': False,
'axes.spines.left': True,
'axes.spines.bottom': True,
'axes.spines.right': True,
'axes.spines.top': True}
```

[122]:
```
sns.histplot(data=tips, x="total_bill")
sns.despine()
```

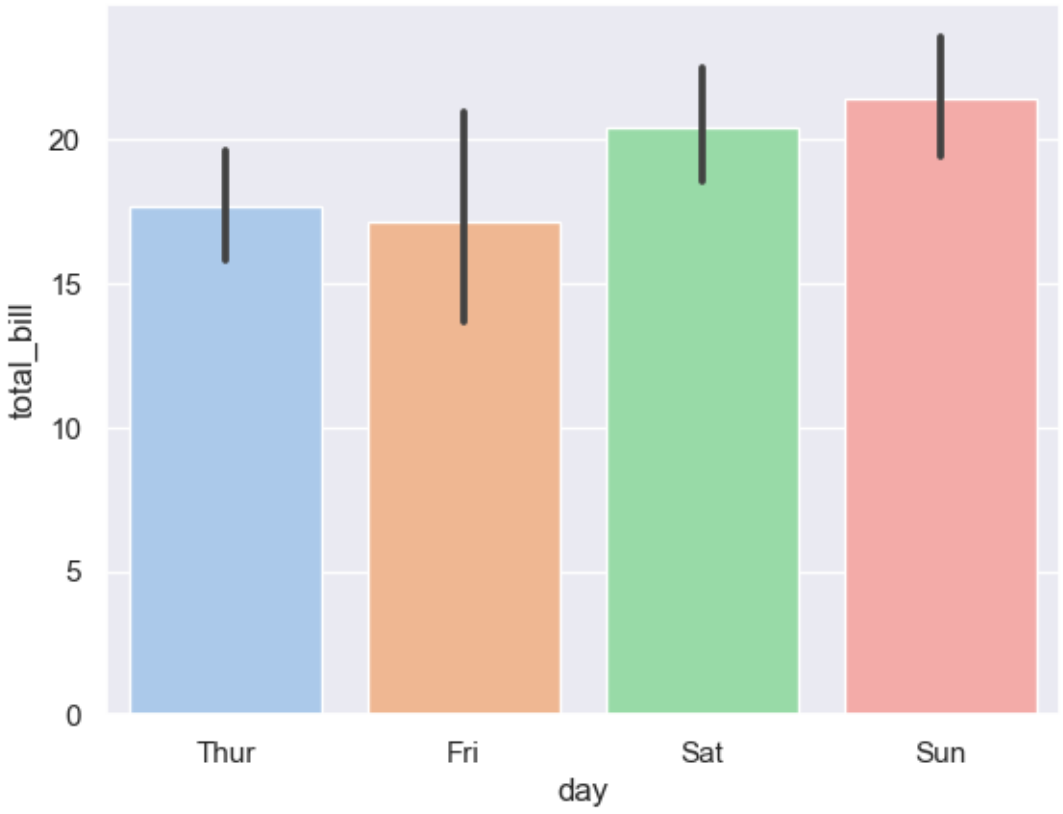


[123]:
```
sns.histplot(data=tips, x="total_bill")
sns.despine(bottom=True, top=False)
```



[124]:
```
sns.set_palette("pastel")
sns.barplot(data=tips, x="day", y="total_bill")
```

[124]: <Axes: xlabel='day', ylabel='total_bill'>



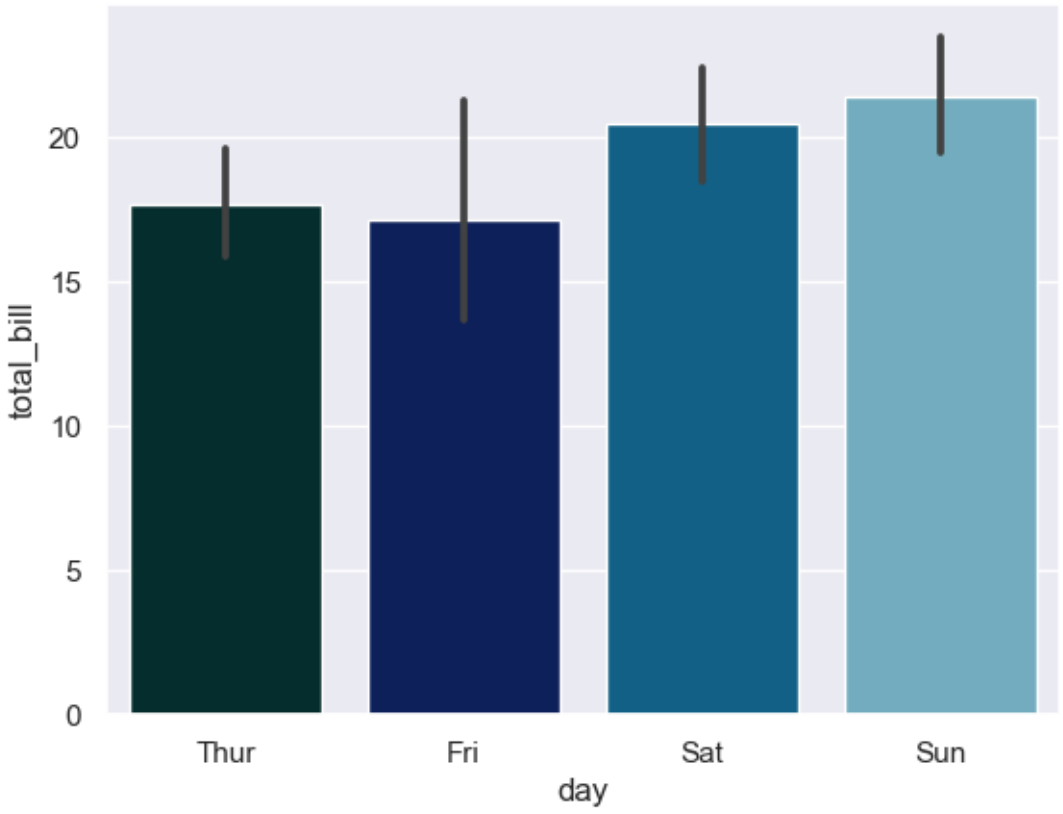
[125]:
```
sns.color_palette("gist_rainbow")
```

[125]:
```
[(1.0, 0.6009538950715422, 0.0),
 (0.6147323794382618, 1.0, 0.0),
 (0.0, 1.0, 0.14758591608686492),
 (0.0, 1.0, 0.9276829011174367),
 (0.0, 0.3058397271952257, 1.0),
 (0.4827365728900258, 0.0, 1.0)]
```

[126]:
```
sns.set_palette("terrain")
```

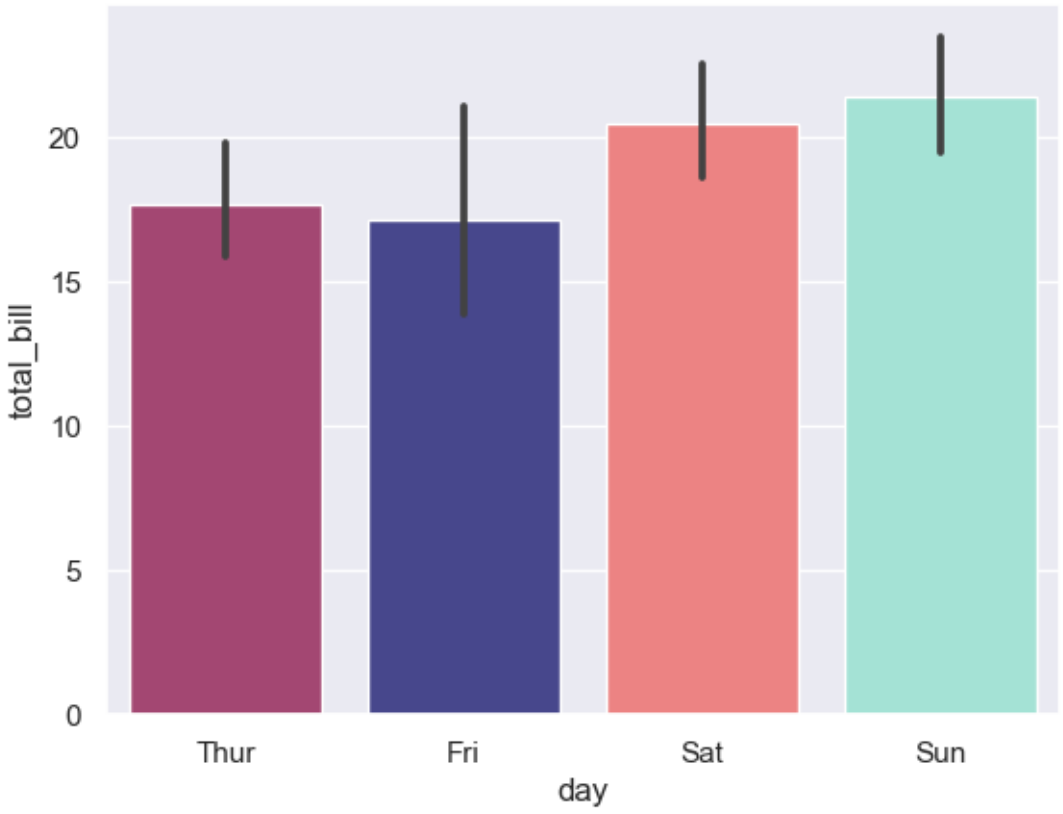
[127]:
```
sns.barplot(data=tips, x="day", y="total_bill", palette="ocean")
```

[127]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



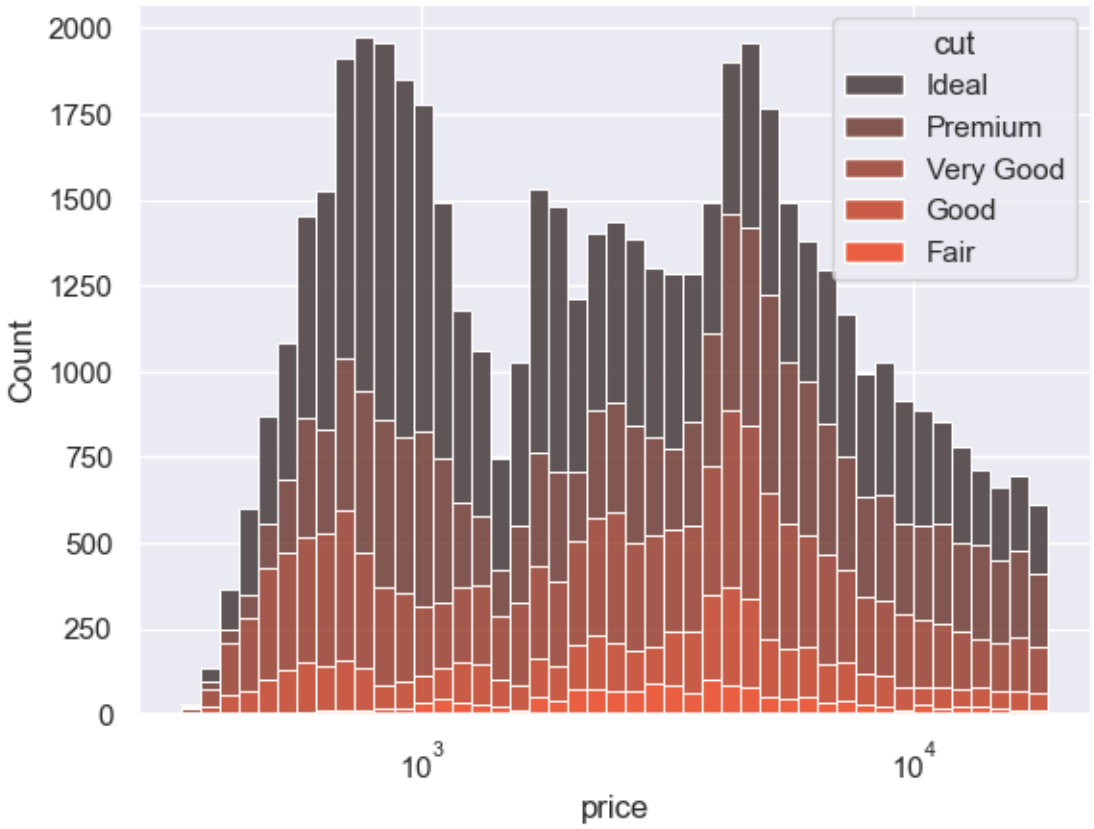
[128]:
```
codes = ["#B33771", "#3B3B98", "#FD7272", "#9AECDB", "#D6A2E8"]
custom_pal = sns.color_palette(codes)
sns.barplot(data=tips, x="day", y="total_bill", palette=custom_pal)
```

[128]: <Axes: xlabel='day', ylabel='total_bill'>



[129]:
```
sns.histplot(
    diamonds,
    x="price", hue="cut",
    multiple="stack",
    log_scale=True,
    palette=sns.dark_palette("#eb2f06", 5)
)
```

[129]: <Axes: xlabel='price', ylabel='Count'>



[130]:
```
sns.blend_palette(["#4a69bd", "#e58e26"])
```

[130]:
```
[(0.2901960784313726, 0.4117647058823529, 0.7411764705882353),
 (0.411764705882353, 0.4407843137254902, 0.6227450980392157),
 (0.5333333333333334, 0.46980392156862744, 0.5043137254901962),
 (0.6549019607843138, 0.4988235294117647, 0.3858823529411765),
 (0.7764705882352942, 0.527843137254902, 0.26745098039215687),
 (0.8980392156862745, 0.5568627450980392, 0.14901960784313725)]
```