Descriptive-Statistics—Penguins

Say OL

October 14, 2023

Contents

1 Descriptive Statistics

1.1 Loading Libraries

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

1.2 The Penguins Dataset

[2]:
```
penguins = sns.load_dataset(name="penguins")
penguins.head()
```

[2]:
```
  species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0  Adelie  Torgersen            39.1           18.7              181.0
1  Adelie  Torgersen            39.5           17.4              186.0
2  Adelie  Torgersen            40.3           18.0              195.0
3  Adelie  Torgersen             NaN            NaN                NaN
4  Adelie  Torgersen            36.7           19.3              193.0

   body_mass_g     sex
0       3750.0    Male
1       3800.0  Female
2       3250.0  Female
3          NaN     NaN
4       3450.0  Female
```

1.3 Last Five Rows

[3]:
```
penguins.tail()
```

[3]:
```
     species  island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
339  Gentoo  Biscoe             NaN            NaN                NaN
340  Gentoo  Biscoe            46.8           14.3              215.0
341  Gentoo  Biscoe            50.4           15.7              222.0
342  Gentoo  Biscoe            45.2           14.8              212.0
343  Gentoo  Biscoe            49.9           16.1              213.0

     body_mass_g     sex
339          NaN     NaN
340       4850.0  Female
341       5750.0    Male
342       5200.0  Female
343       5400.0    Male
```

1.4 Dataset Index

[4]:
```
penguins.index.values
```

[4]:
```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
        13,  14,  15,  16,  17,  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,  56,  57,  58,  59,  60,  61,  62,  63,  64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
```

```
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
       143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
       156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
       169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
       182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194,
       195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
       208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220,
       221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233,
       234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
       247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259,
       260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272,
       273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285,
       286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298,
       299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311,
       312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324,
       325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337,
       338, 339, 340, 341, 342, 343], dtype=int64)
```

1.5 Dataset Columns

[5]:
```
penguins.columns.values
```

[5]:
```
array(['species', 'island', 'bill_length_mm', 'bill_depth_mm',
       'flipper_length_mm', 'body_mass_g', 'sex'], dtype=object)
```

1.6 Data Types

[6]:
```
penguins.dtypes
```

[6]:
```
species              object
island               object
bill_length_mm      float64
bill_depth_mm       float64
flipper_length_mm   float64
body_mass_g         float64
sex                  object
dtype: object
```

1.7 Short Information

[7]:
```
penguins.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
```

```
 0   species            344 non-null    object
 1   island             344 non-null    object
 2   bill_length_mm     342 non-null    float64
 3   bill_depth_mm      342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                333 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

1.8 Count Duplicated Rows

[8]:
```
penguins.duplicated().sum()
```

[8]:
```
0
```

1.9 Count Missing Data

[9]:
```
penguins.isnull().sum()
```

[9]:
```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
dtype: int64
```

1.10 List Rows Contained Missing Values

[10]:
```
penguins[penguins.isnull().any(axis=1)]
```

[10]:
```
     species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
3     Adelie  Torgersen             NaN            NaN                NaN
8     Adelie  Torgersen            34.1           18.1              193.0
9     Adelie  Torgersen            42.0           20.2              190.0
10    Adelie  Torgersen            37.8           17.1              186.0
11    Adelie  Torgersen            37.8           17.3              180.0
47    Adelie      Dream            37.5           18.9              179.0
246   Gentoo     Biscoe            44.5           14.3              216.0
286   Gentoo     Biscoe            46.2           14.4              214.0
324   Gentoo     Biscoe            47.3           13.8              216.0
336   Gentoo     Biscoe            44.5           15.7              217.0
339   Gentoo     Biscoe             NaN            NaN                NaN

     body_mass_g  sex
3            NaN  NaN
8         3475.0  NaN
```

```
9          4250.0  NaN
10         3300.0  NaN
11         3700.0  NaN
47         2975.0  NaN
246        4100.0  NaN
286        4650.0  NaN
324        4725.0  NaN
336        4875.0  NaN
339           NaN  NaN
```

1.11 Drop Rows Contained Missing Values

[11]:
```
df = penguins.dropna().copy()
df.head()
```

[11]:
```
  species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0  Adelie  Torgersen            39.1           18.7              181.0
1  Adelie  Torgersen            39.5           17.4              186.0
2  Adelie  Torgersen            40.3           18.0              195.0
4  Adelie  Torgersen            36.7           19.3              193.0
5  Adelie  Torgersen            39.3           20.6              190.0

   body_mass_g     sex
0       3750.0    Male
1       3800.0  Female
2       3250.0  Female
4       3450.0  Female
5       3650.0    Male
```

1.12 Descriptive Statistics of Numerical Columns

[12]:
```
df.describe()
```

[12]:
```
       bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g
count      333.000000     333.000000         333.000000   333.000000
mean        43.992793      17.164865         200.966967  4207.057057
std          5.468668       1.969235          14.015765   805.215802
min         32.100000      13.100000         172.000000  2700.000000
25%         39.500000      15.600000         190.000000  3550.000000
50%         44.500000      17.300000         197.000000  4050.000000
75%         48.600000      18.700000         213.000000  4775.000000
max         59.600000      21.500000         231.000000  6300.000000
```

1.13 Descriptive Statistics of Categorical Columns

[13]:
```
df.describe(include="object")
```

[13]:
```
        species  island   sex
count       333     333   333
unique        3       3     2
top     Adelie  Biscoe  Male
freq        146     163   168
```

1.14 Distribution of Numerical Columns

[14]:
```
pd.cut(x=df["bill_depth_mm"],
       bins=10,
       right=False)\
    .value_counts(sort=False)\
    .to_frame()
```

[14]:
```
                 count
bill_depth_mm
[13.1, 13.94)       20
[13.94, 14.78)      33
[14.78, 15.62)      32
[15.62, 16.46)      33
[16.46, 17.3)       43
[17.3, 18.14)       53
[18.14, 18.98)      55
[18.98, 19.82)      39
[19.82, 20.66)      15
[20.66, 21.508)     10
```

[15]:
```
pd.cut(x=df["bill_length_mm"],
       bins=10,
       right=False)\
    .value_counts(sort=False)\
    .to_frame()
```

[15]:
```
                 count
bill_length_mm
[32.1, 34.85)        8
[34.85, 37.6)       39
[37.6, 40.35)       55
[40.35, 43.1)       47
[43.1, 45.85)       47
[45.85, 48.6)       53
[48.6, 51.35)       61
[51.35, 54.1)       16
[54.1, 56.85)        5
[56.85, 59.628)      2
```

[16]:
```
pd.cut(x=df["body_mass_g"],
       bins=10,
       right=False)\
```

```
    .value_counts(sort=False)\
    .to_frame()
```

[16]:
body_mass_g	count
[2700.0, 3060.0)	14
[3060.0, 3420.0)	42
[3420.0, 3780.0)	69
[3780.0, 4140.0)	52
[4140.0, 4500.0)	41
[4500.0, 4860.0)	39
[4860.0, 5220.0)	27
[5220.0, 5580.0)	27
[5580.0, 5940.0)	16
[5940.0, 6303.6)	6

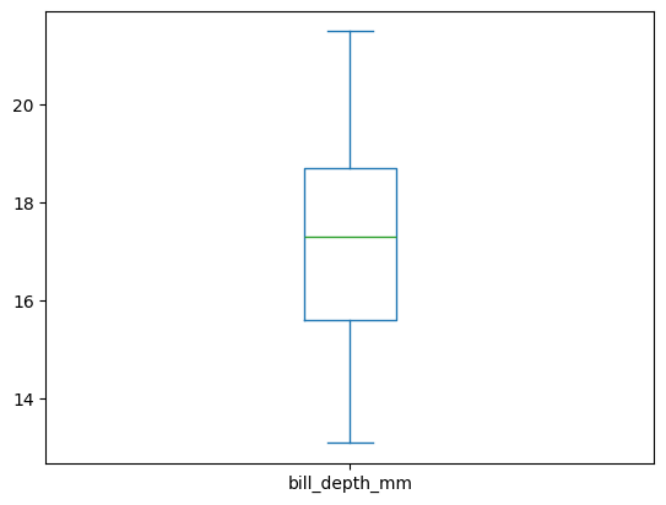
[17]:
```
pd.cut(x=df["flipper_length_mm"],
       bins=10,
       right=False)\
    .value_counts(sort=False)\
    .to_frame()
```

[17]:
flipper_length_mm	count
[172.0, 177.9)	3
[177.9, 183.8)	20
[183.8, 189.7)	51
[189.7, 195.6)	77
[195.6, 201.5)	44
[201.5, 207.4)	15
[207.4, 213.3)	42
[213.3, 219.2)	38
[219.2, 225.1)	28
[225.1, 231.059)	15

1.15 Distribution of Numerical Columns using Box Plot

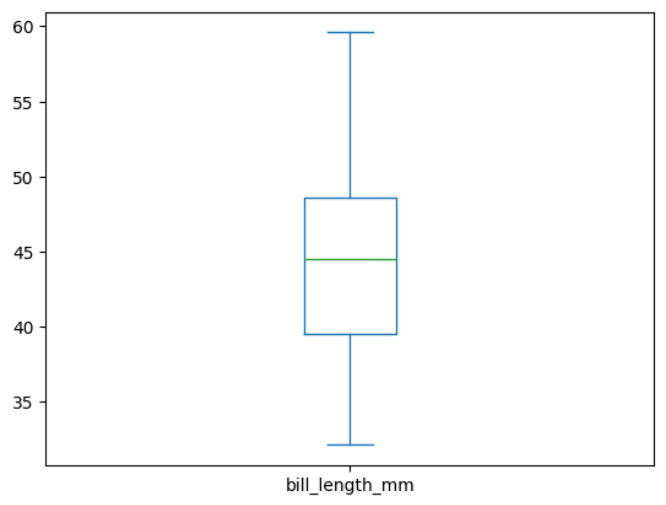
[18]:
```
df["bill_depth_mm"].plot.box()
```

[18]: <Axes: >



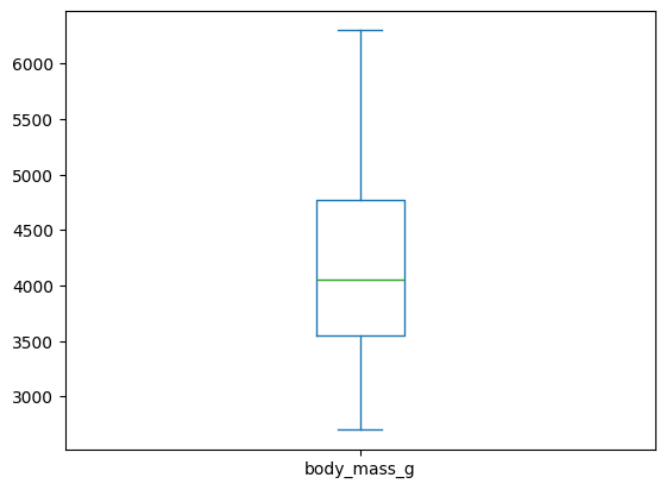
[19]:
```
df["bill_length_mm"].plot.box()
```

[19]: <Axes: >



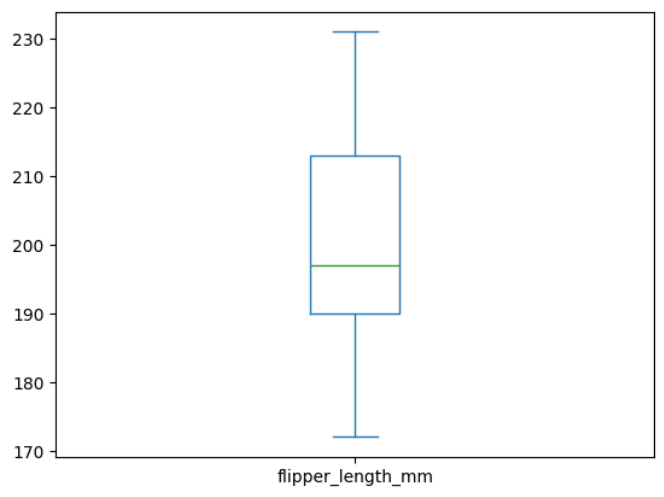
[20]:
```
df["body_mass_g"].plot.box()
```

[20]: <Axes: >



[21]:
```
df["flipper_length_mm"].plot.box()
```

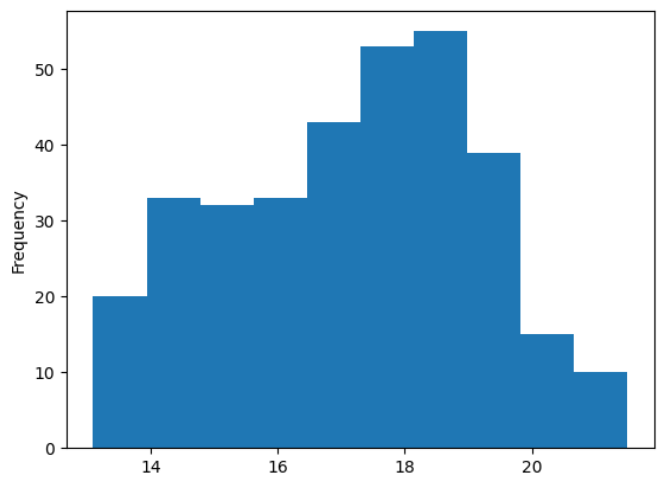
[21]:
```
<Axes: >
```



1.16 Distribution of Numerical Columns using Histogram

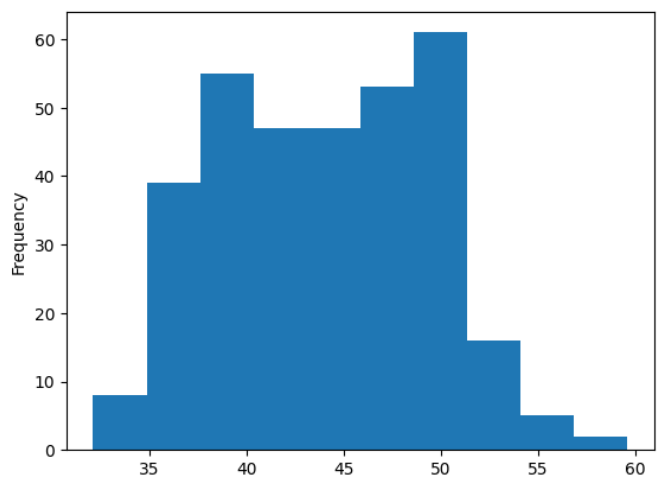
[22]:
```
df["bill_depth_mm"].plot.hist()
```

[22]:
```
<Axes: ylabel='Frequency'>
```



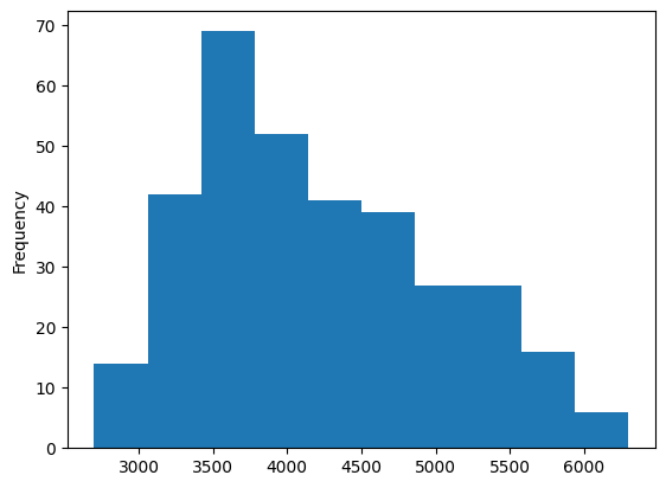
[23]:
```
df["bill_length_mm"].plot.hist()
```

[23]: <Axes: ylabel='Frequency'>



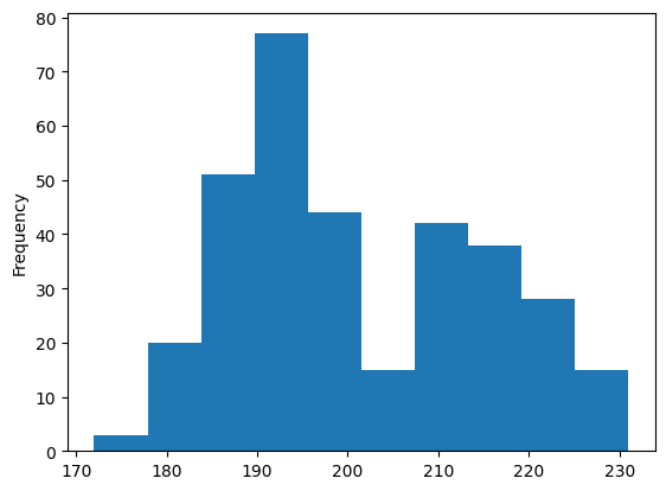
[24]:
```
df["body_mass_g"].plot.hist()
```

[24]: <Axes: ylabel='Frequency'>



[25]:
```
df["flipper_length_mm"].plot.hist()
```

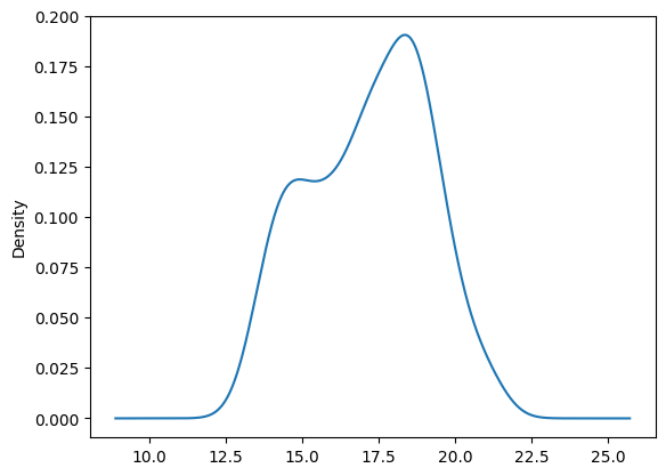
[25]:
```
<Axes: ylabel='Frequency'>
```



1.17 Distribution of Numerical Columns using Kernel Density Estimation

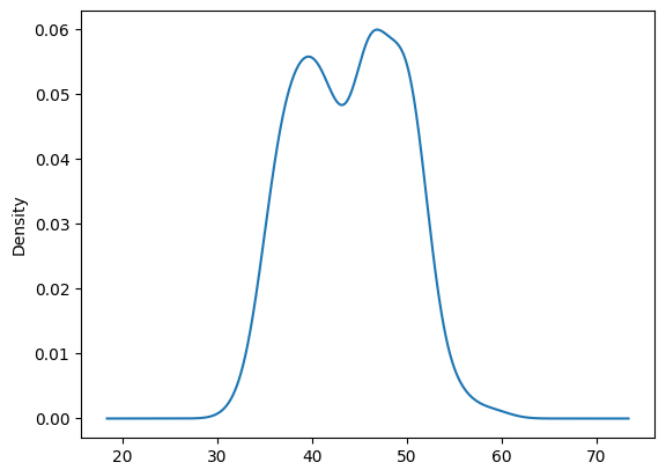
[26]:
```
df["bill_depth_mm"].plot.kde()
```

[26]:
```
<Axes: ylabel='Density'>
```



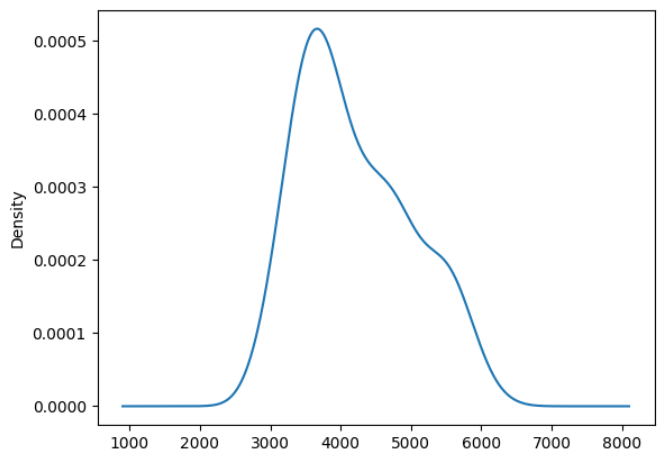
[27]:
```
df["bill_length_mm"].plot.kde()
```

[27]:
```
<Axes: ylabel='Density'>
```



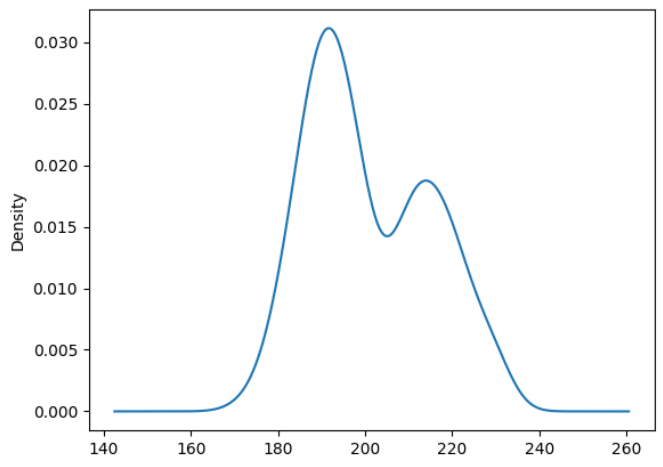
[28]:
```
df["body_mass_g"].plot.kde()
```

[28]:
```
<Axes: ylabel='Density'>
```



[29]:
```
df["flipper_length_mm"].plot.kde()
```

[29]:
```
<Axes: ylabel='Density'>
```



1.18 Distribution of Categorical Columns

[30]:
```
df["island"].value_counts(sort=False).to_frame()
```

[30]:

island	count
Torgersen	47
Biscoe	163
Dream	123

[31]:
```
df["island"].value_counts(sort=False, normalize=True).to_frame()
```

[31]:
```
           proportion
island
Torgersen    0.141141
Biscoe       0.489489
Dream        0.369369
```

[32]:
```
df["island"].value_counts(sort=False, normalize=True)\
    .mul(other=100)\
    .to_frame(name="percentage")
```

[32]:
```
           percentage
island
Torgersen   14.114114
Biscoe      48.948949
Dream       36.936937
```

[33]:
```
df["sex"].value_counts(sort=False).to_frame()
```

[33]:
```
        count
sex
Male      168
Female    165
```

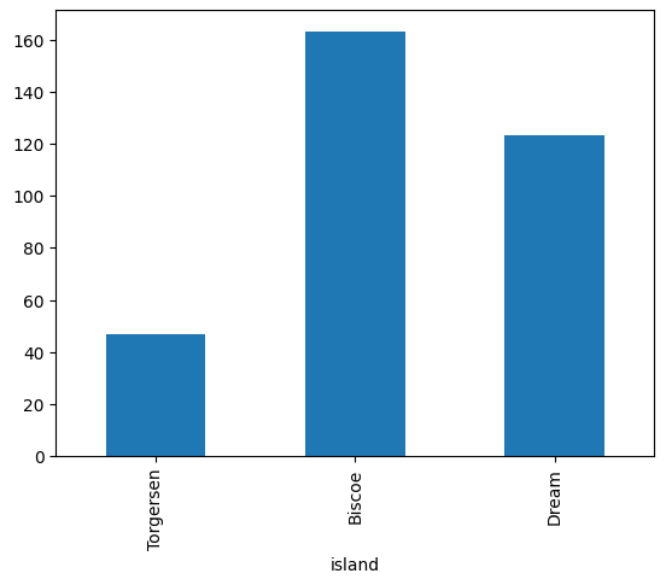
[34]:
```
df["species"].value_counts(sort=False).to_frame()
```

[34]:
```
           count
species
Adelie       146
Chinstrap     68
Gentoo       119
```

1.19 Distribution of Categorical Columns using Bar Graph

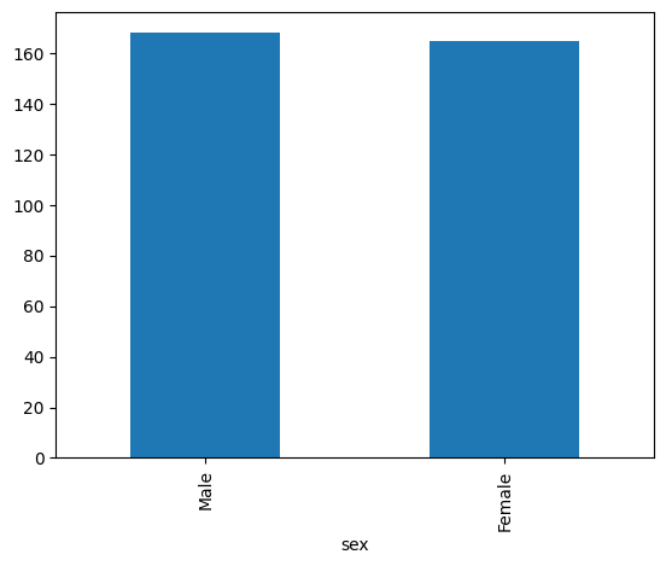
[35]:
```
df["island"].value_counts(sort=False).plot.bar()
```

[35]:
```
<Axes: xlabel='island'>
```



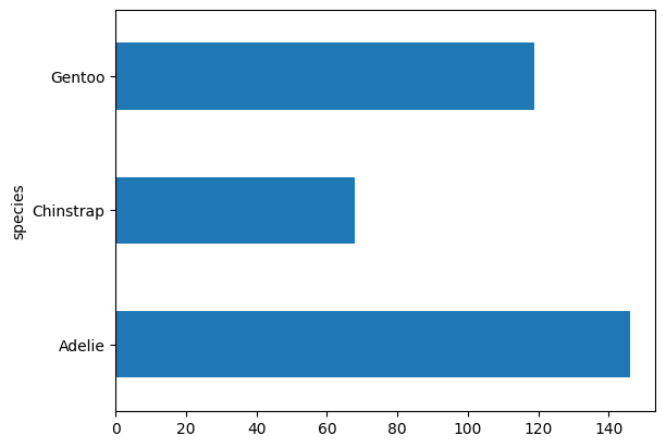
[36]:
```
df["sex"].value_counts(sort=False).plot.bar()
```

[36]: <Axes: xlabel='sex'>



[37]:
```
df["species"].value_counts(sort=False).plot.barh()
```

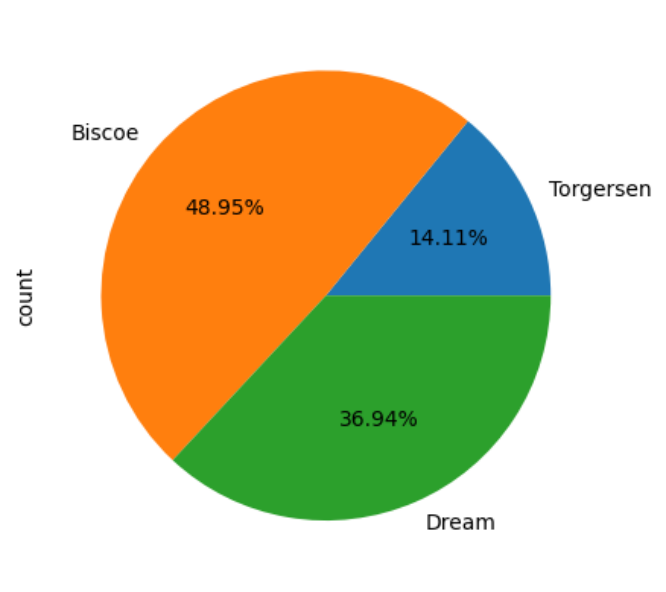
[37]: <Axes: ylabel='species'>



1.20 Distribution of Categorical Columns using Pie Chart

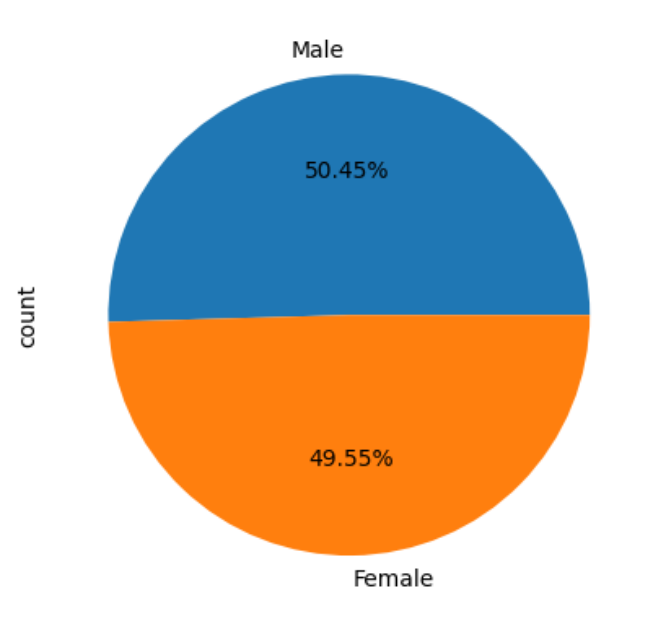
[38]:
```
df["island"].value_counts(sort=False).plot.pie(autopct="%0.2f%%")
```

[38]:
```
<Axes: ylabel='count'>
```



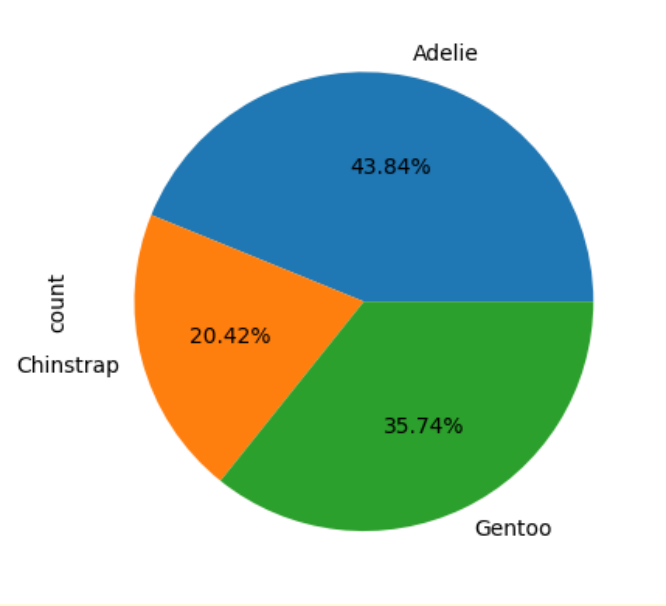
[39]:
```
df["sex"].value_counts(sort=False).plot.pie(autopct="%0.2f%%")
```

[39]:
```
<Axes: ylabel='count'>
```



[40]:
```
df["species"].value_counts(sort=False).plot.pie(autopct="%0.2f%%")
```

[40]:
```
<Axes: ylabel='count'>
```



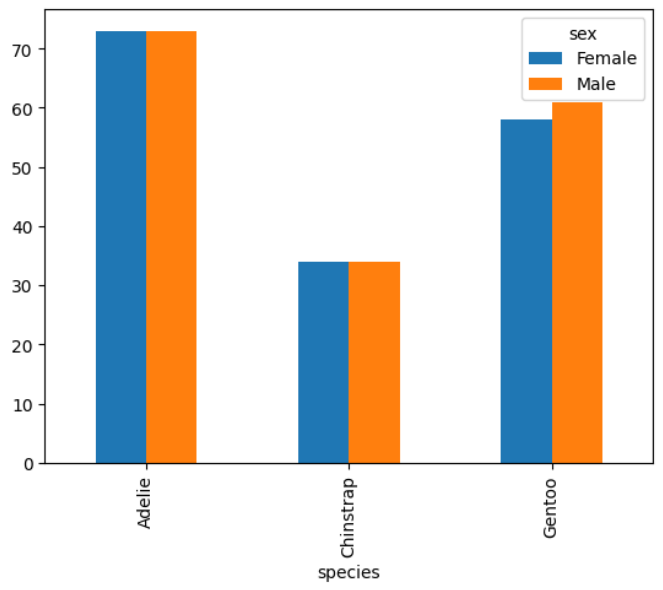
1.21 Distribution of Categorical Columns by Species

[41]:
```
df.pivot_table(index="species",
               columns="sex",
               values="body_mass_g",
               aggfunc="count")
```

[41]:
```
sex        Female  Male
species
Adelie         73    73
Chinstrap      34    34
Gentoo         58    61
```

[42]:
```
df.pivot_table(index="species",
               columns="sex",
               values="body_mass_g",
               aggfunc="count").plot.bar()
```

[42]:
```
<Axes: xlabel='species'>
```



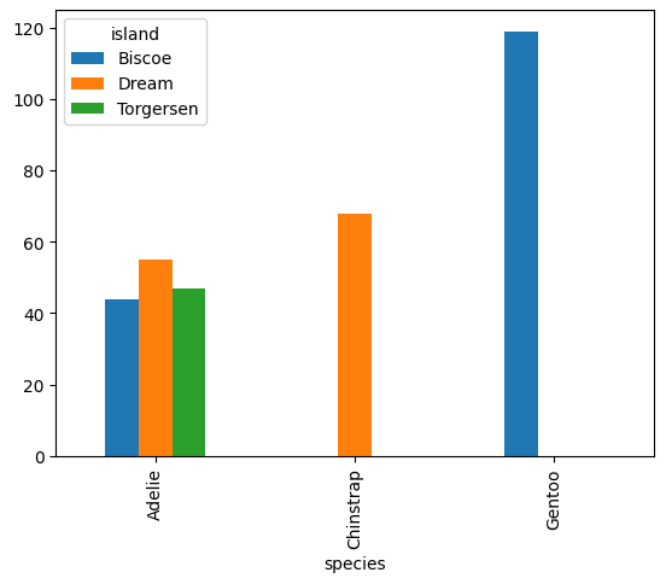
[43]:
```
df.pivot_table(index="species",
               columns="island",
               values="body_mass_g",
               aggfunc="count",
               fill_value=0)
```

[43]:
```
island     Biscoe  Dream  Torgersen
species
Adelie         44     55         47
Chinstrap       0     68          0
```

```
Gentoo        119      0          0
```

[44]:
```
df.pivot_table(index="species",
               columns="island",
               values="body_mass_g",
               aggfunc="count",
               fill_value=0).plot.bar()
```

[44]:
```
<Axes: xlabel='species'>
```



[45]:
```
df.pivot_table(index=["species", "sex"],
               columns="island",
               values="body_mass_g",
               aggfunc=["count", "mean"],
               fill_value=0)
```

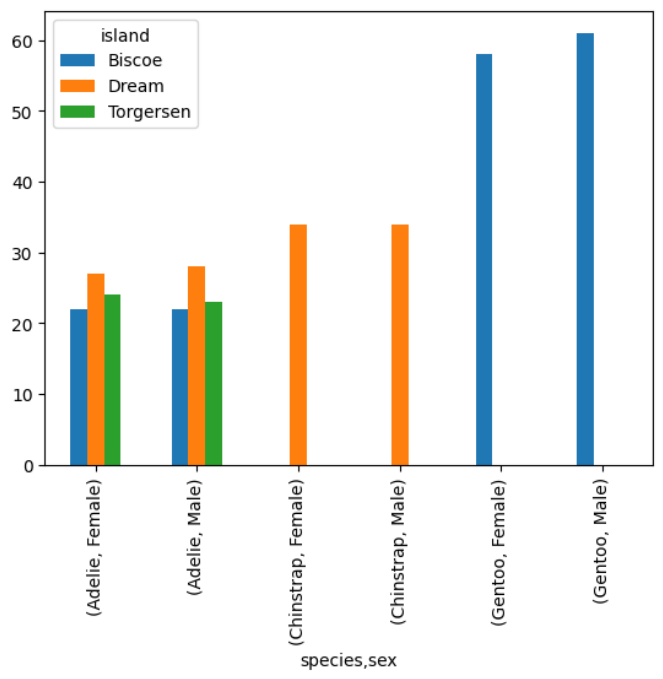
[45]:

		count			mean		
island		Biscoe	Dream	Torgersen	Biscoe	Dream	Torgersen
species	sex						
Adelie	Female	22	27	24	3369.318182	3344.444444	3395.833333
	Male	22	28	23	4050.000000	4045.535714	4034.782609
Chinstrap	Female	0	34	0	0.000000	3527.205882	0.000000
	Male	0	34	0	0.000000	3938.970588	0.000000
Gentoo	Female	58	0	0	4679.741379	0.000000	0.000000
	Male	61	0	0	5484.836066	0.000000	0.000000

[46]:
```
df.pivot_table(index=["species", "sex"],
               columns="island",
               values="body_mass_g",
               aggfunc="count",
```

```
                fill_value=0).plot.bar()
```

[46]: <Axes: xlabel='species,sex'>



1.22 Export Cleaned Penguins Dataset to CSV File

[47]:
```
df.to_csv(path_or_buf="penguins_clean.csv", index=False)
```