IRIS

DATASET

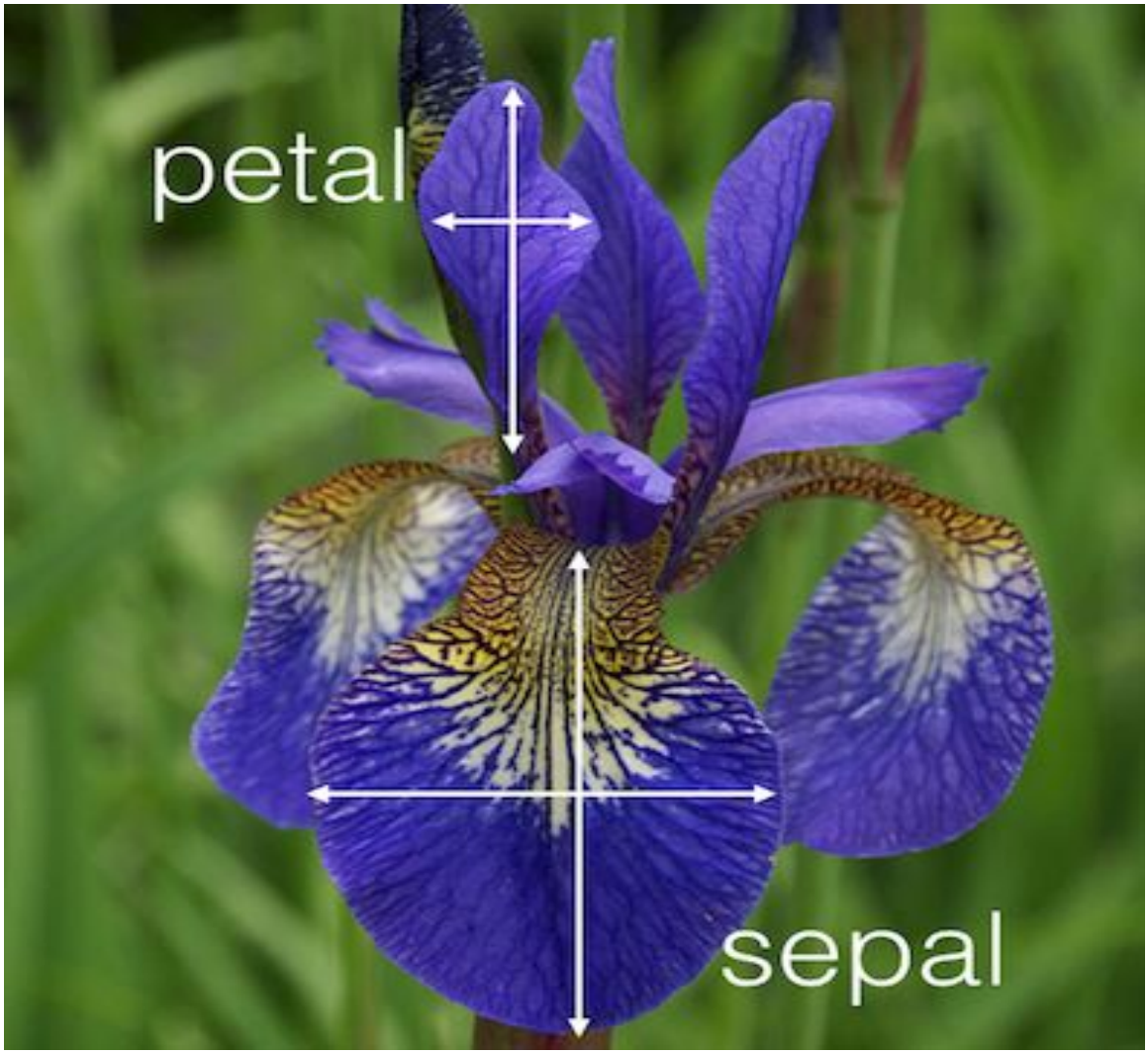
Iris

- standard dataset for analysis

Iris Dataset

- 4 numeric features:
  1. sepal-length
  2. sepal-width
  3. petal-length
  4. petal-width
- 3 labels:
  1. Iris-setosa
  2. Iris-versicolor
  3. Iris-virginica

Numeric Features



Exploring IRIS

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"
data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])
```

```
> data.head(5)
   sepal-length  sepal-width  petal-length  petal-width        Class
0           5.1          3.5           1.4          0.2  Iris-setosa
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
3           4.6          3.1           1.5          0.2  Iris-setosa
4           5.0          3.6           1.4          0.2  Iris-setosa

> len(data)
150
> set(data['Class'])
{'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
```

IRIS Feature Set

```
> data.head(5)
   sepal-length  sepal-width  petal-length  petal-width        Class
0           5.1          3.5           1.4          0.2  Iris-setosa
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
3           4.6          3.1           1.5          0.2  Iris-setosa
4           5.0          3.6           1.4          0.2  Iris-setosa

> set(data['Class'])
{'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}
```

$$X = \begin{pmatrix} x_{1_{\text{sepal-length}}} & x_{1_{\text{sepal-width}}} & x_{1_{\text{petal-length}}} & x_{1_{\text{petal-width}}} \\ x_{2_{\text{sepal-length}}} & x_{2_{\text{sepal-width}}} & x_{2_{\text{petal-length}}} & x_{2_{\text{petal-width}}} \\ \cdots & \cdots & \cdots & \cdots \\ x_{150_{\text{sepal-length}}} & x_{150_{\text{sepal-width}}} & x_{150_{\text{petal-length}}} & x_{150_{\text{petal-width}}} \end{pmatrix}$$

and

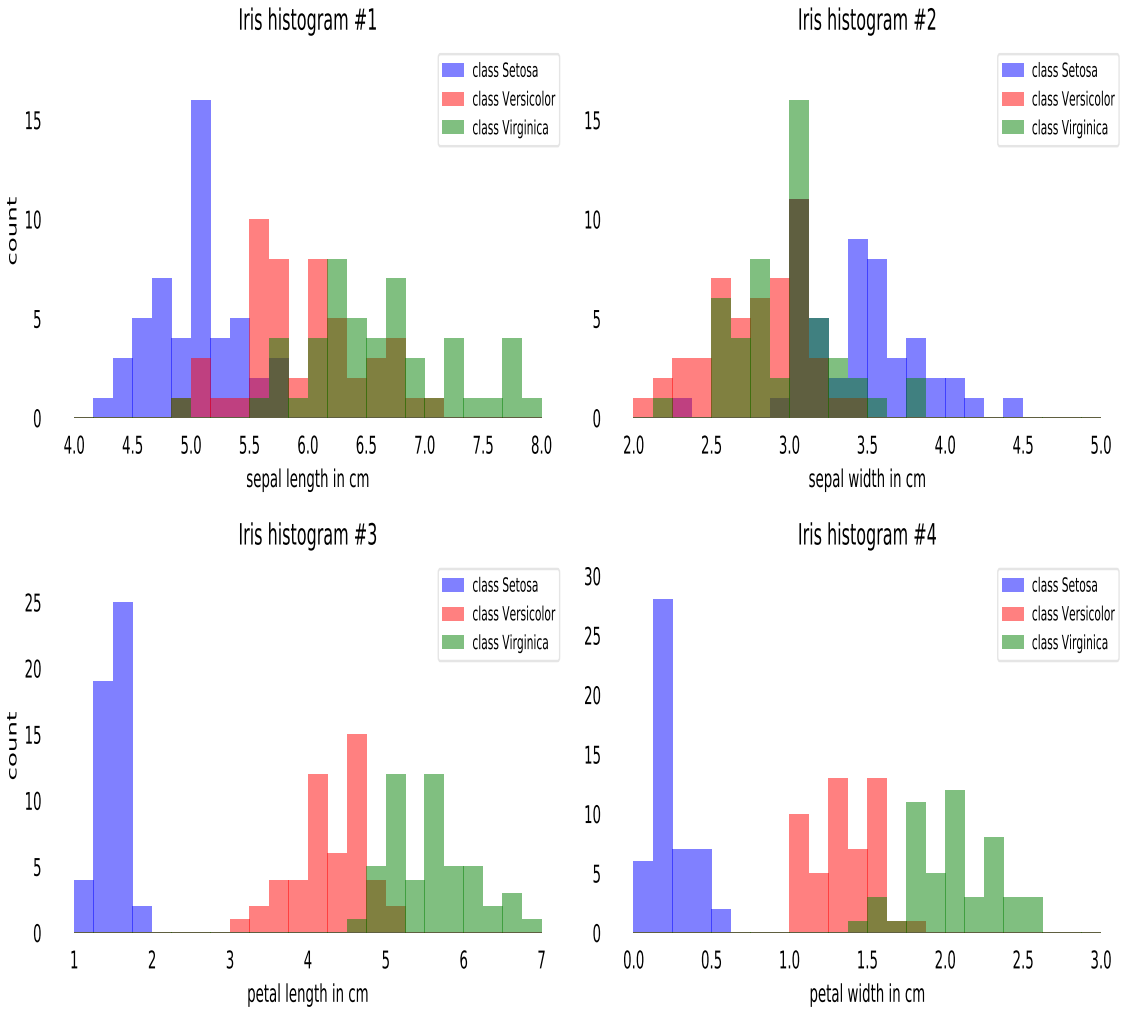
$$Y = \begin{pmatrix} \text{Iris-setosa} \\ \text{Iris-setosa} \\ \cdots \\ \text{Iris-virginica} \end{pmatrix}$$

Iris Statistics

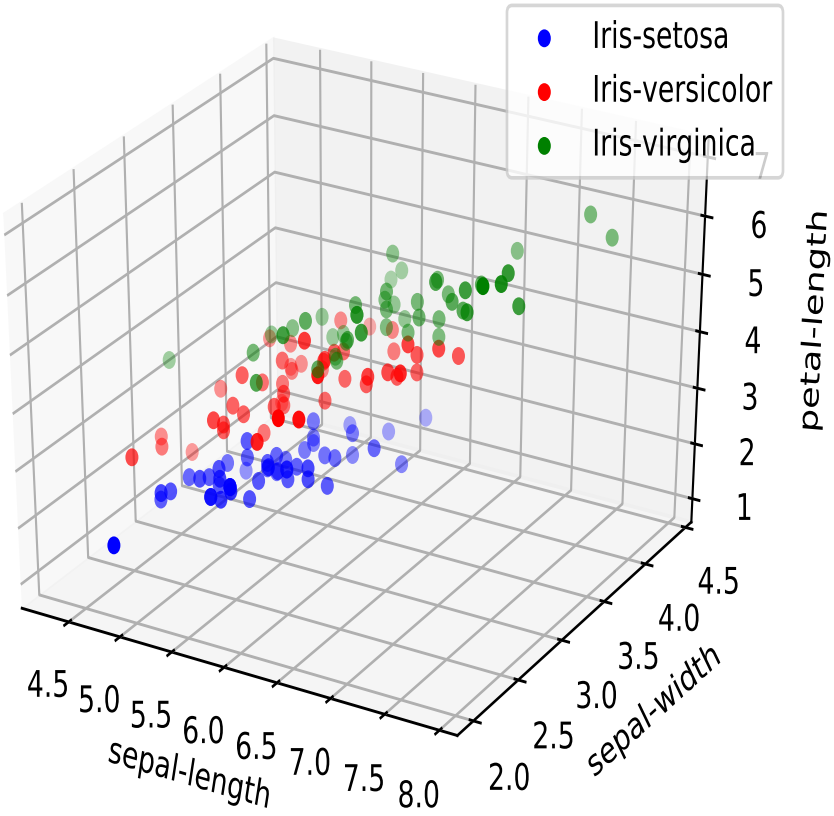
```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])
```

```
> data.describe()
ipdb> data.describe()
       sepal-length  sepal-width  petal-length  petal-width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.054000      3.758667     1.198667
std        0.828066     0.433594      1.764420     0.763161
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
ipdb>
```

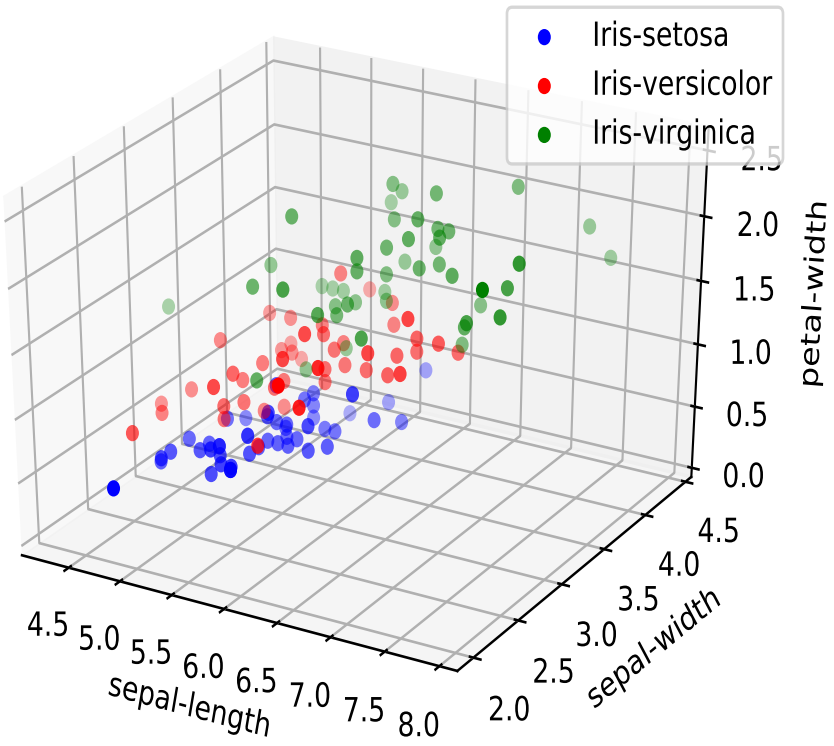
Iris Histograms



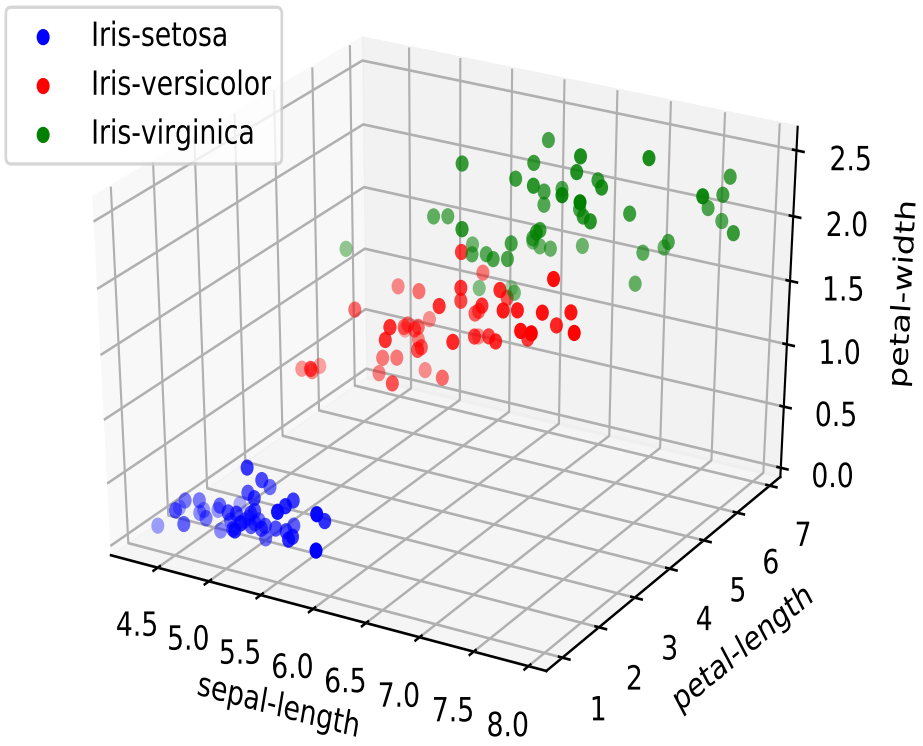
Iris Dataset:



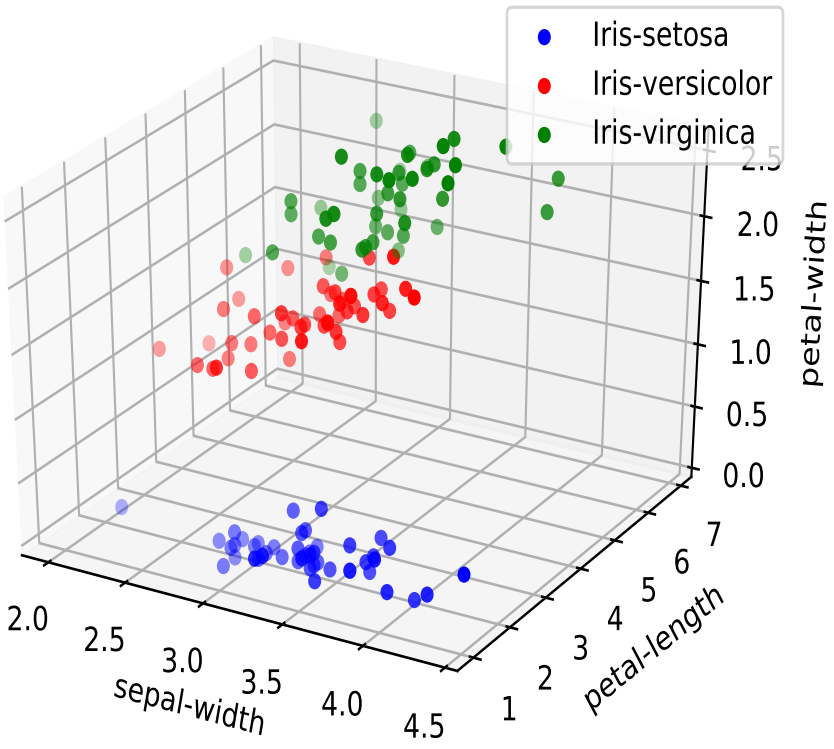
Iris Dataset:



Iris Dataset:



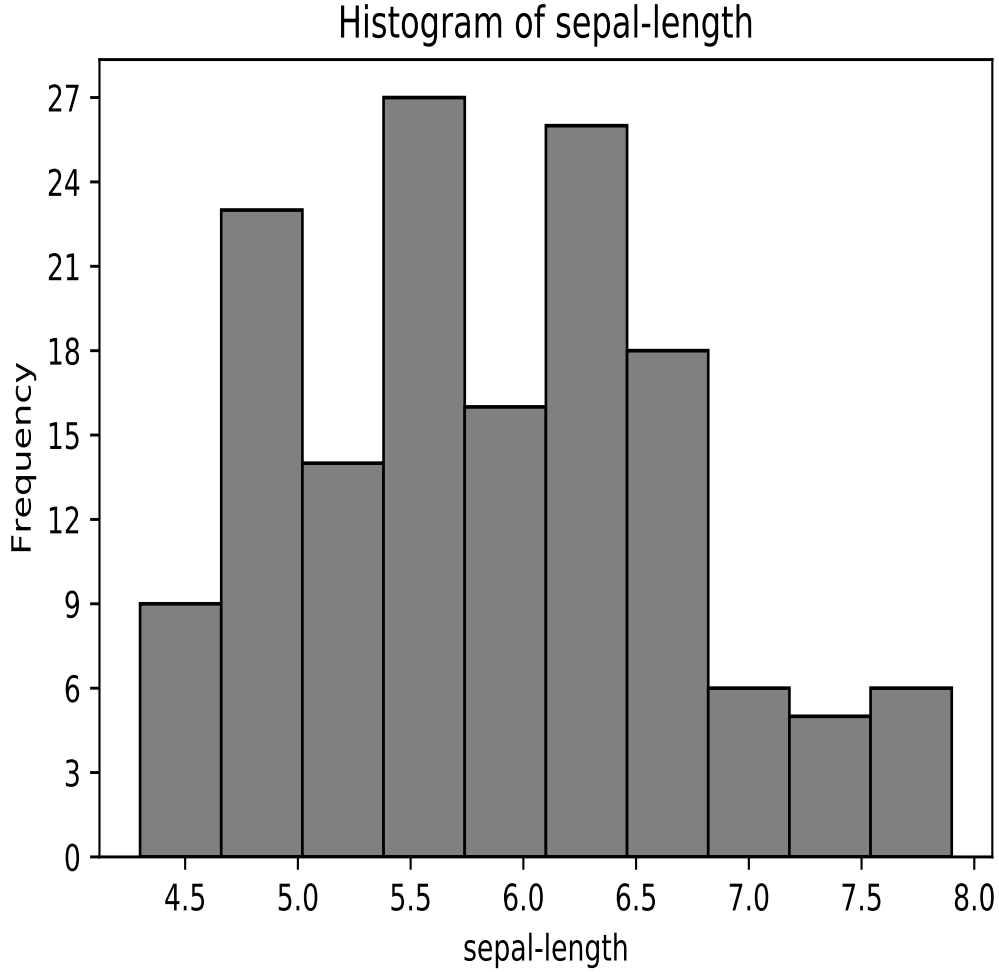
Iris Dataset:



A Simple Histogram

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

fig = plt.figure()
axes1 = fig.add_subplot(1,1,1)
axes1.hist(data["sepal-length"], bins = 10,
    histtype="bar", ec="black", color="grey")
axes1.set_title("Histogram of sepal-length")
axes1.set_xlabel("sepal-length")
axes1.set_ylabel("Frequency")
axes1.yaxis.set_major_locator(
                    MaxNLocator(integer=True))
fig.show()
```

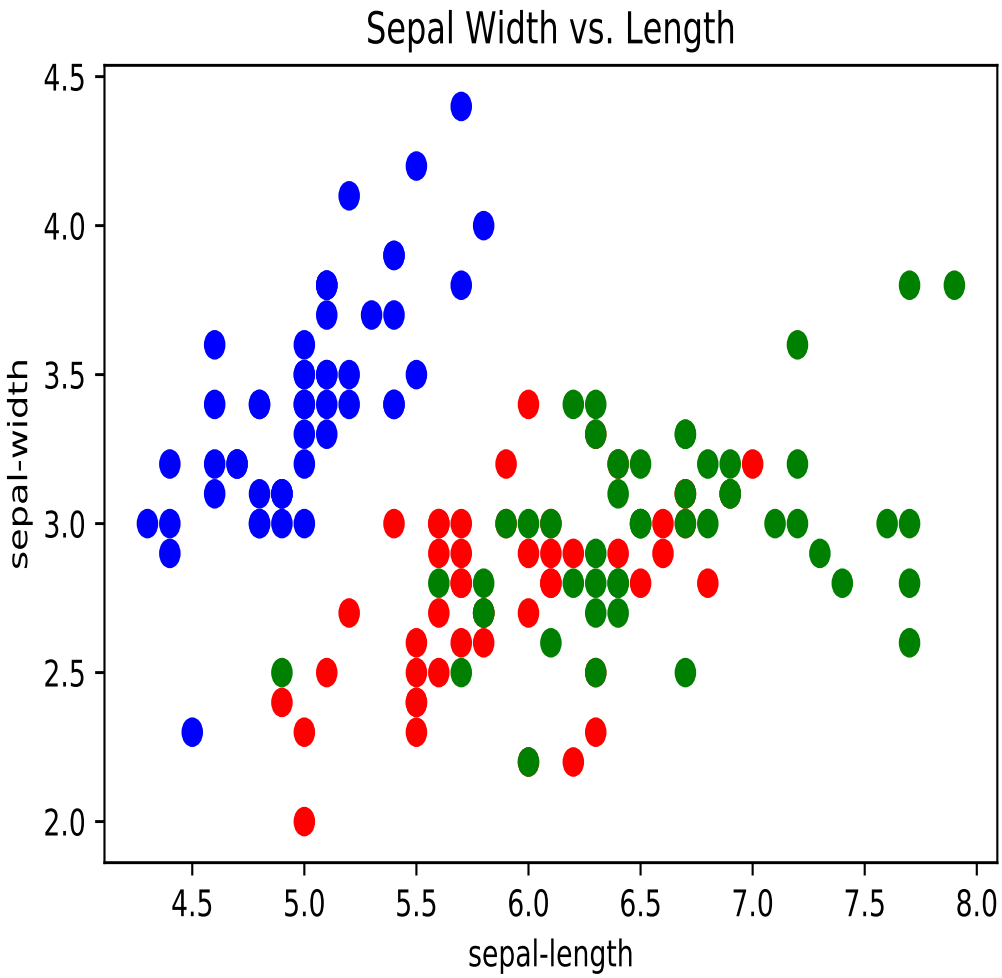
Histogram Illustration



A Simple Scatter Plot

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"
data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

color_dict = {"Iris-setosa" : "Blue",
           "Iris-versicolor" : "Red",
            "Iris-virginica" : "Green"}

data["color"] = data["Class"].map(color_dict)
scatter_plot = plt.figure()
axes1 = scatter_plot.add_subplot(1,1,1)
axes1.scatter(data["sepal-length"],
   data["sepal-width"],color=data["color"],s=50)
axes1.set_title("Sepal Width vs. Length")
axes1.set_xlabel("sepal-length")
axes1.set_ylabel("sepal-width")
scatter_plot.show()
```

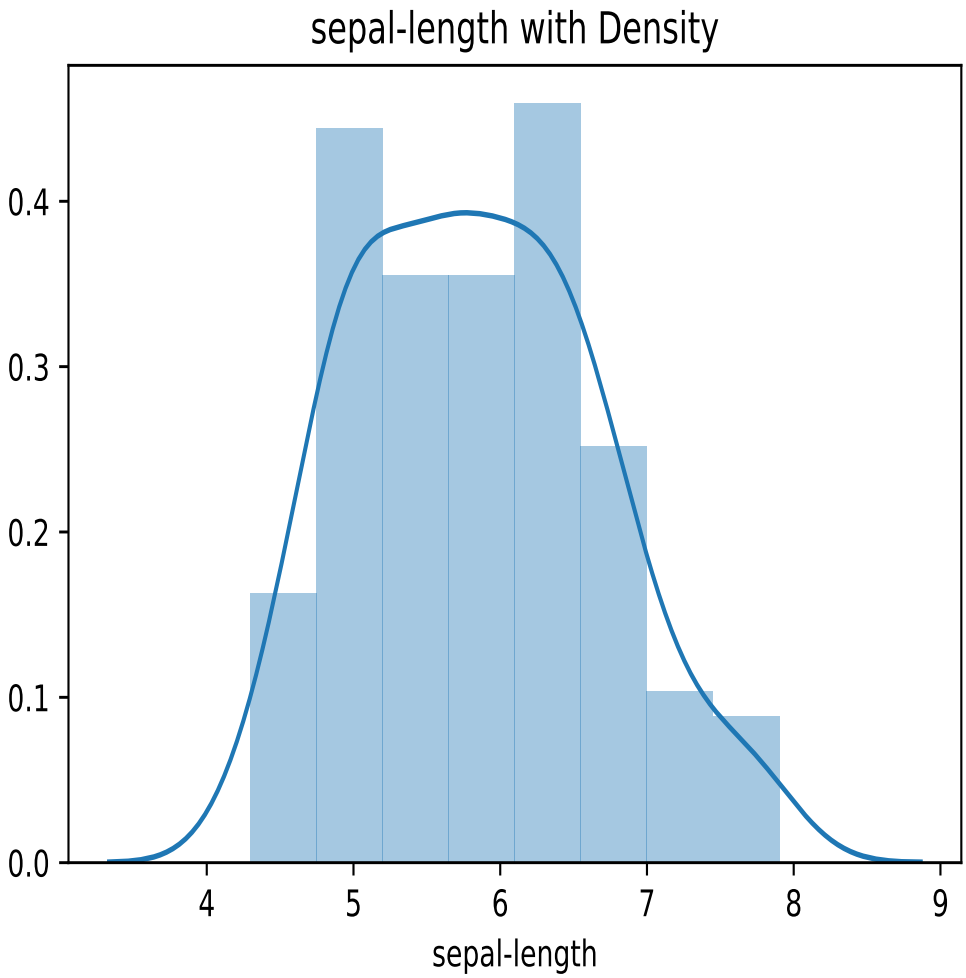
A Scatterplot Illustration



Histogram With Density

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"
data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])


hist, ax = plt.subplots()
ax = sns.distplot(data["sepal-length"])
ax.set_title("sepal-length with Density")
plt.show()
```

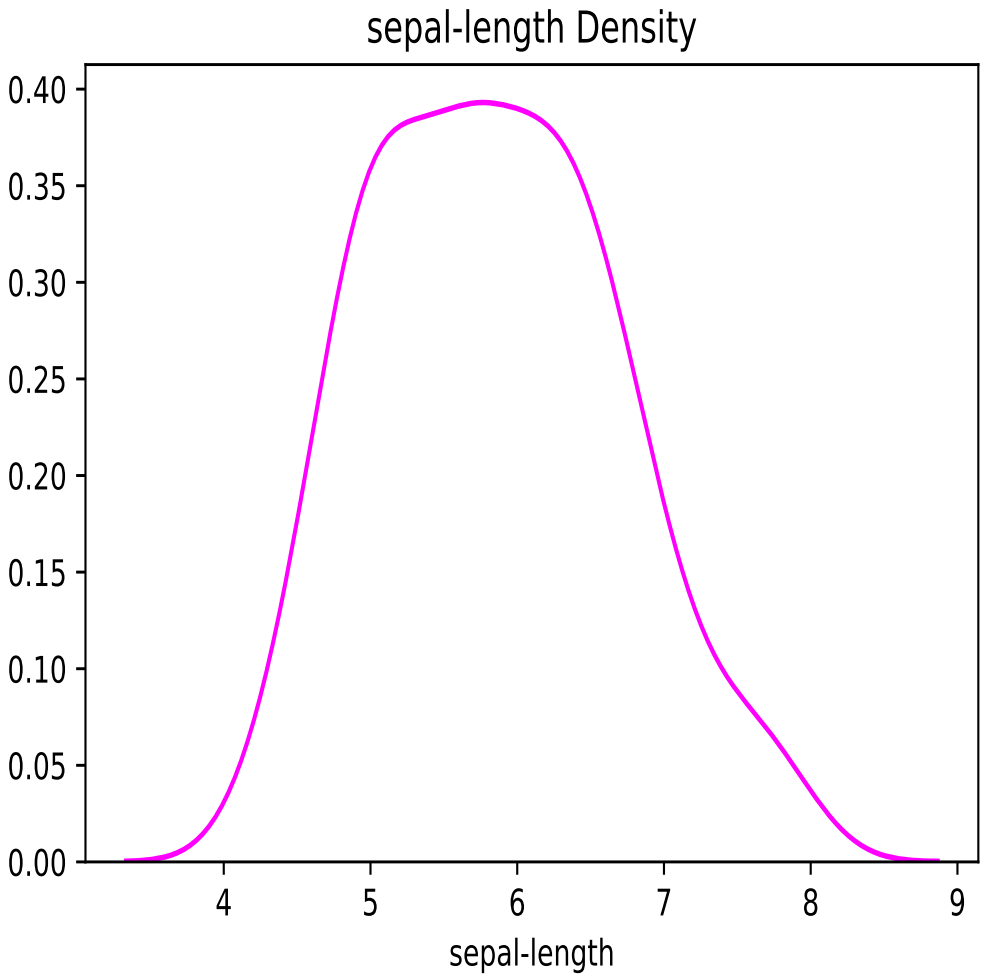
Histogram with Density Illustration



Density

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

hist, ax = plt.subplots()
ax=sns.distplot(data["sepal-length"],
   hist=False, color="magenta")
ax.set_title("sepal-length Density")
plt.show()
```

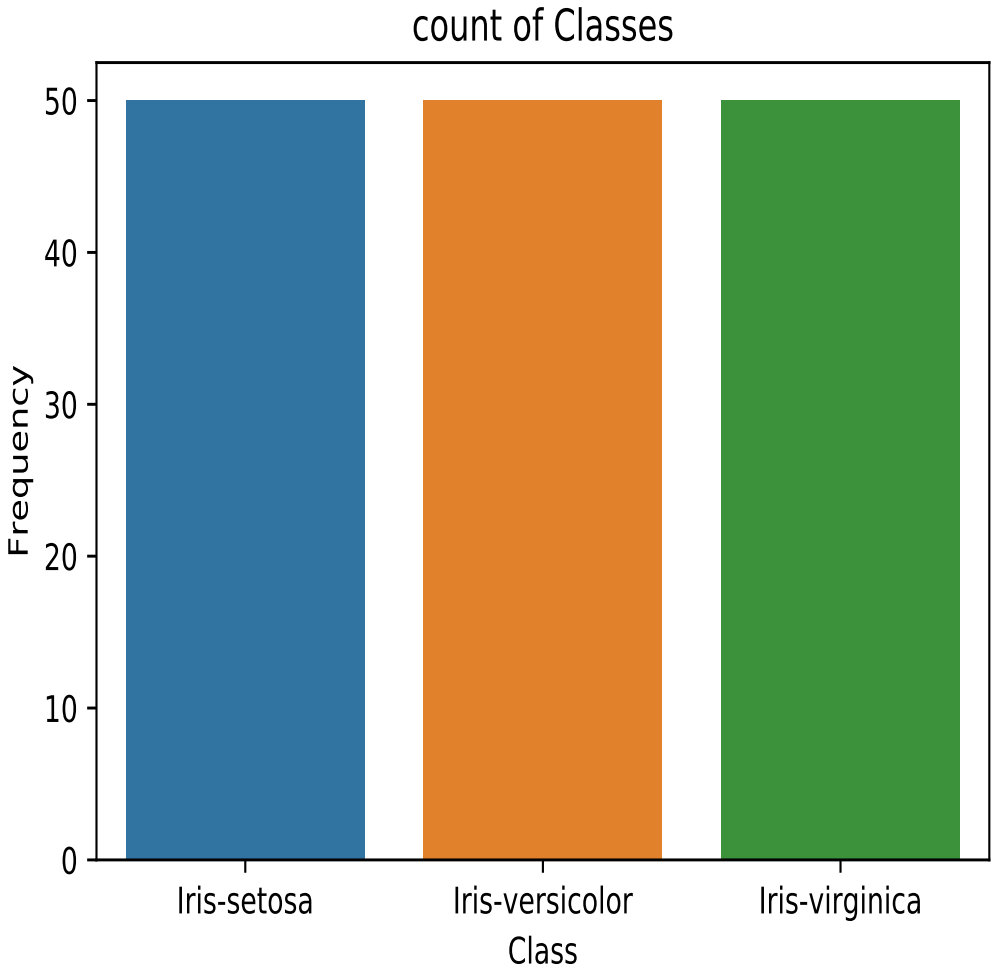
Density Illustration



Counting

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

count, ax = plt.subplots()
ax = sns.countplot("Class", data=data)
ax.set_title("count of Classes")
ax.set_xlabel("Class")
ax.set_ylabel("Frequency")
axes1.yaxis.set_major_locator(
              MaxNLocator(integer=True))
plt.show()
```

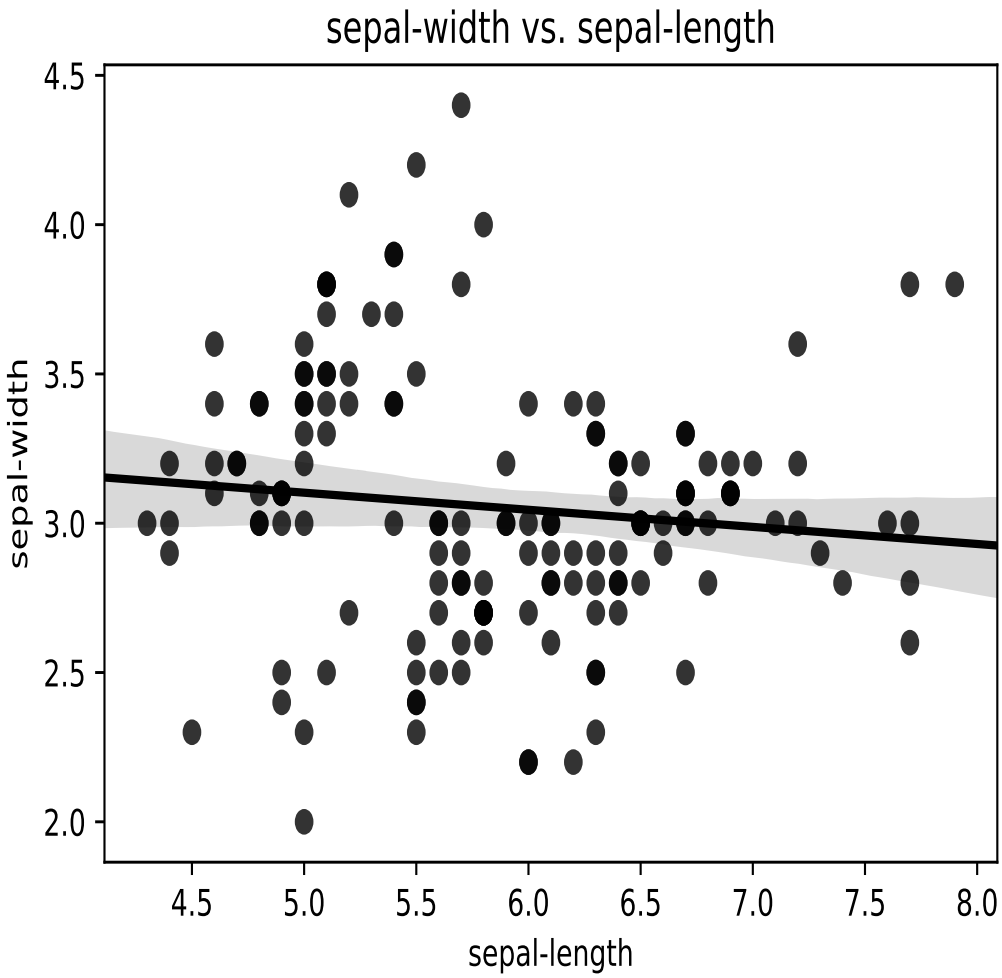
Counting Illustration



Scatterplot With Regression

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

scatter, ax = plt.subplots()
ax = sns.regplot(x = "sepal-length",
                 y = "sepal-width",
                 data = data, color = "Black")
ax.set_title("sepal-width vs. sepal-length")
ax.set_xlabel("sepal-length")
ax.set_ylabel("sepal-width")
plt.show()
```

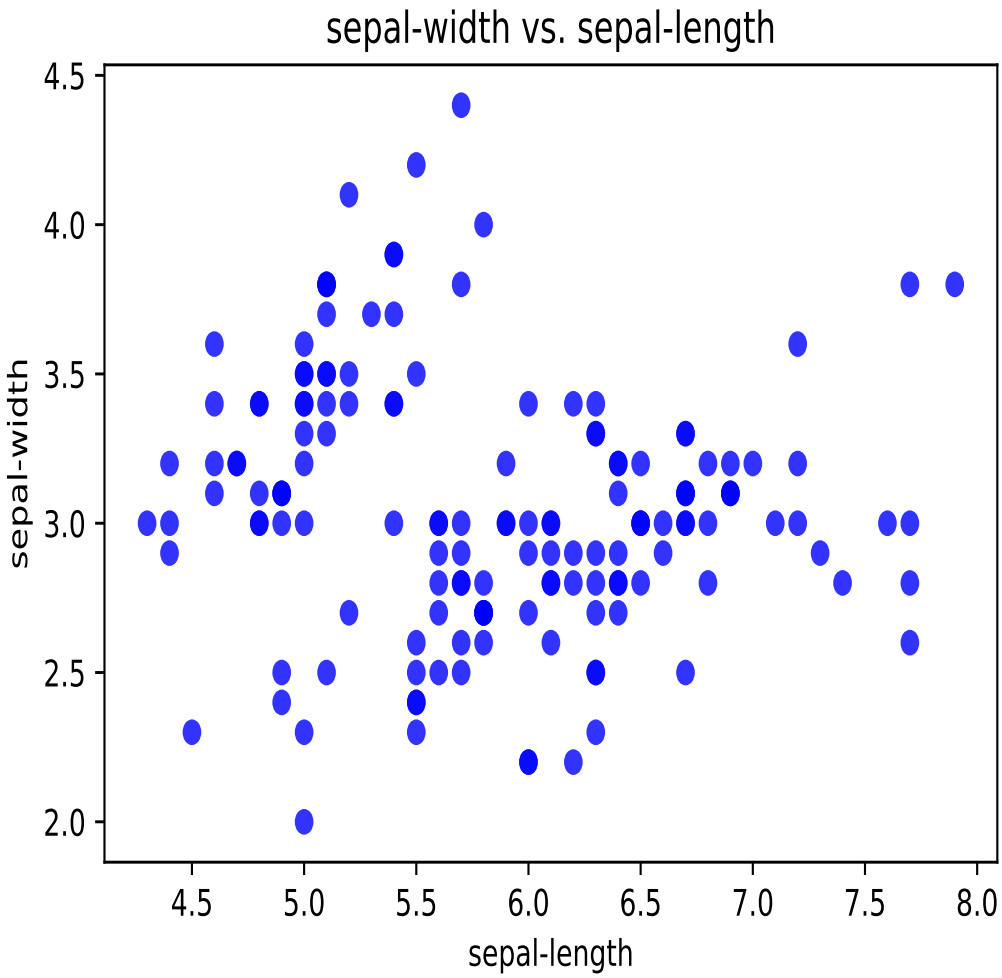
Scatterplot with Regression Illustration



Scatterplot Without Regression

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

scatter, ax = plt.subplots()
ax = sns.regplot(x = "sepal-length",
                 y = "sepal-width",
                 data =d ata, color = "blue",
                 fit_reg = False)
ax.set_title("sepal-width vs. sepal-length")
ax.set_xlabel("sepal-length")
ax.set_ylabel("sepal-width")
plt.show()
```

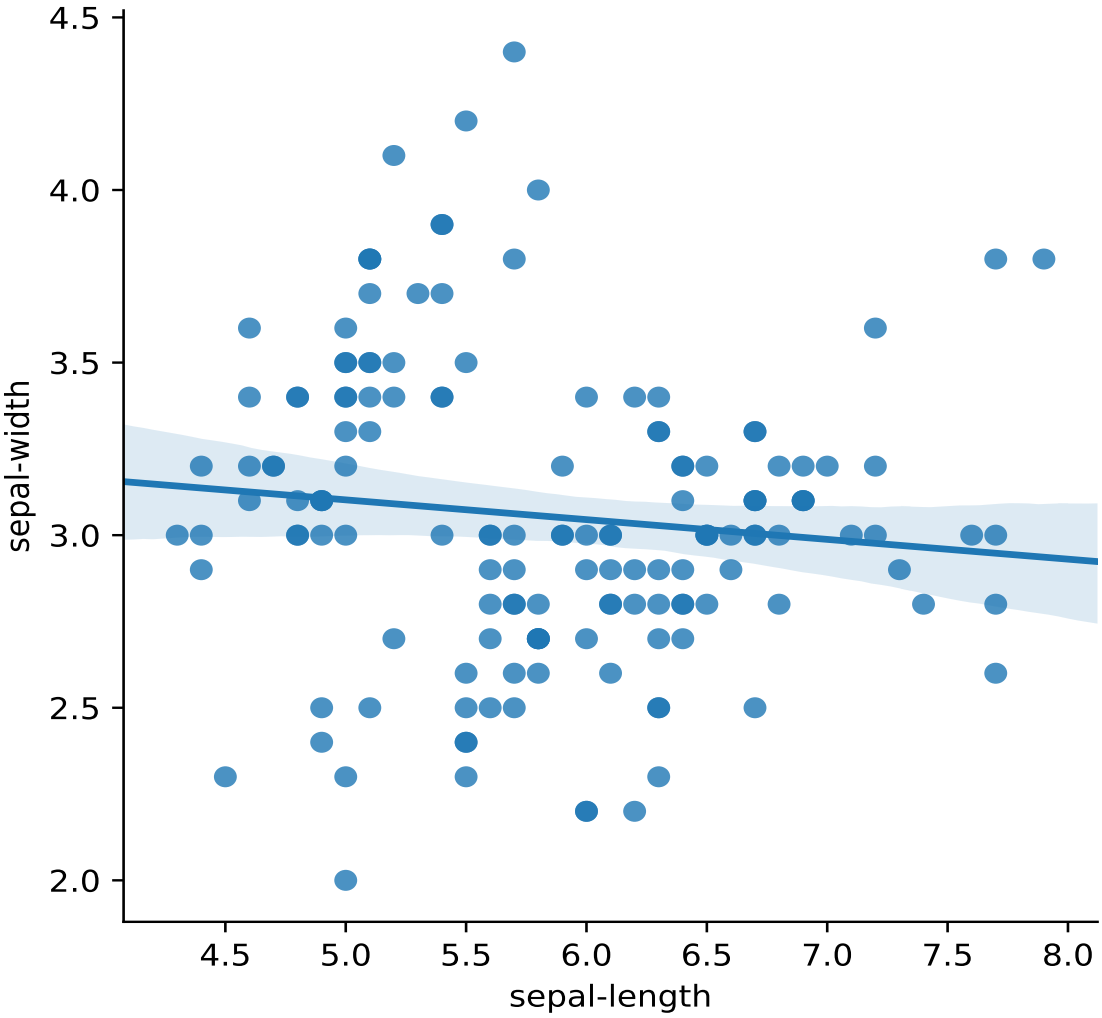
Scatterplot Without Regression



Creating a Figure

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

fig = sns.lmplot(x = "sepal-length",
                 y = "sepal-width",
                 data = data)

plt.show()
```

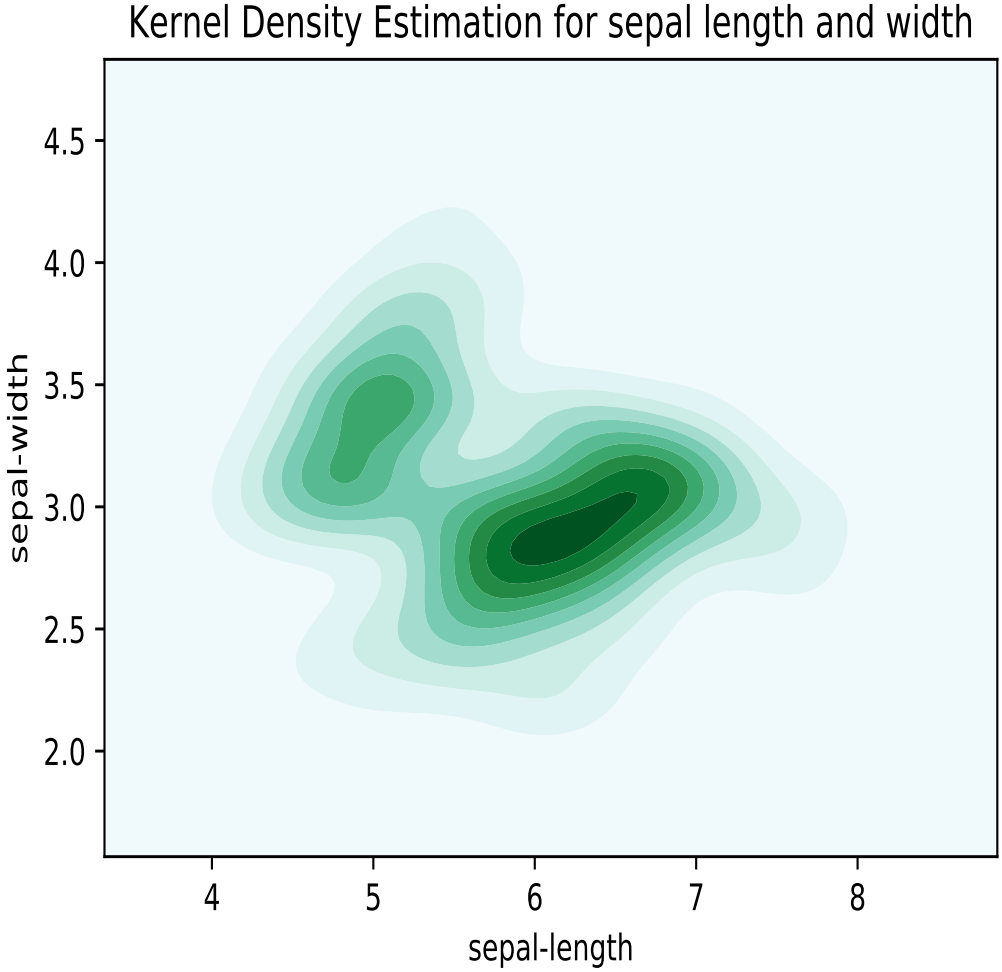
Illustration



Density for Two Variables

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

kde, ax = plt.subplots()
ax = sns.kdeplot(data=data["sepal-length"],
      data2=data["sepal-width"], shade=True)
ax.set_title("Kernel Density Estimation \
              for sepal length and width")
ax.set_xlabel("sepal-length")
ax.set_ylabel("sepal-width")
plt.show()
```

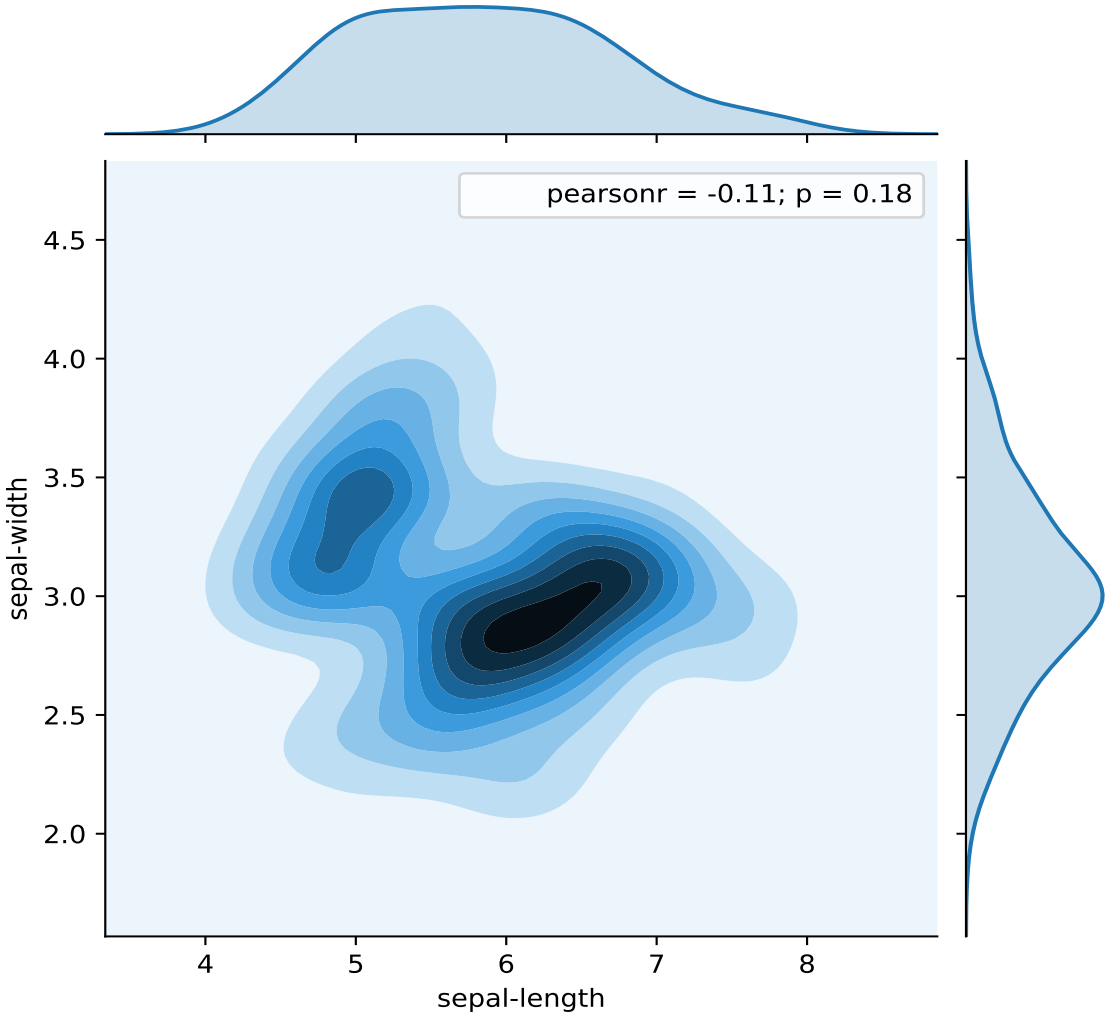
Density for Two Variables



Joint Density

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"
data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

kde_joint = sns.jointplot(x = "sepal-length",
                          y = "sepal-width",


plt.show()
```

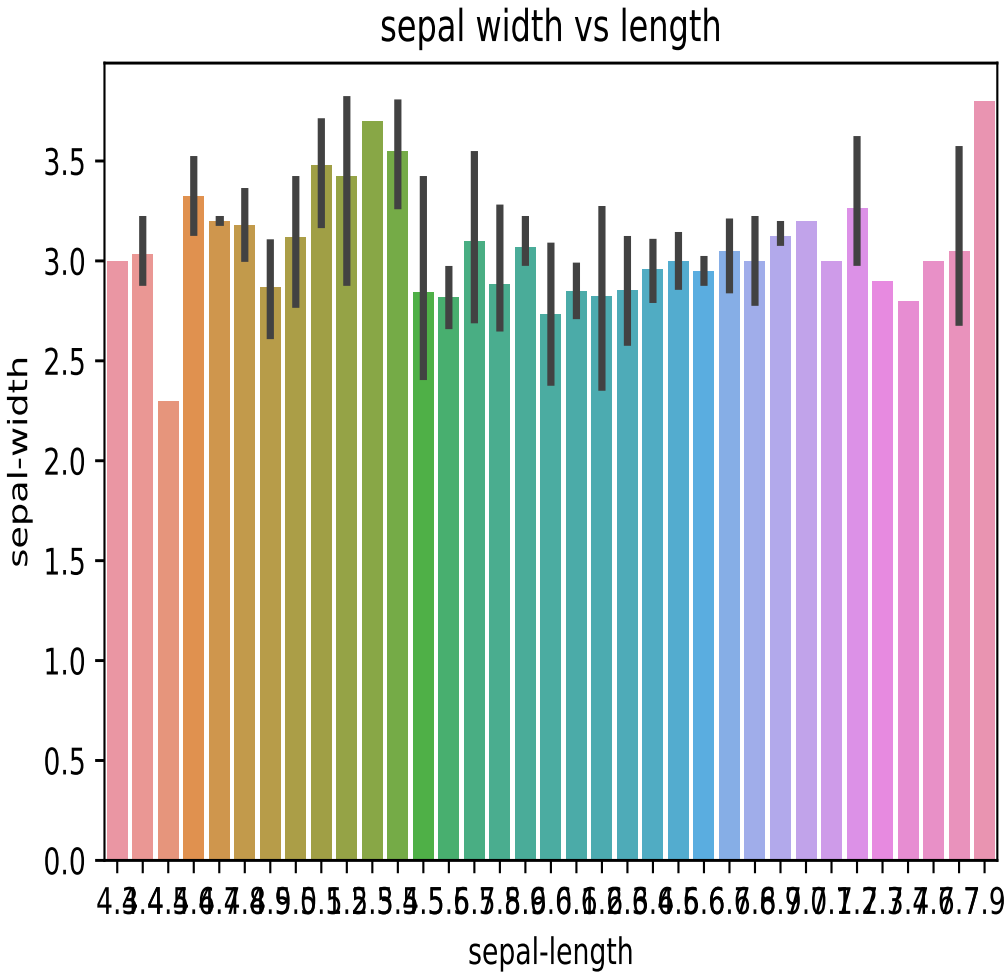
Joint Density



Bar Plots

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

bar, ax = plt.subplots()
ax = sns.barplot(x = "sepal-length",
                 y = "sepal-width",
                 data = data)
ax.set_title("sepal width vs length")
ax.set_xlabel("sepal-length")
ax.set_ylabel("sepal-width")
plt.show()
```

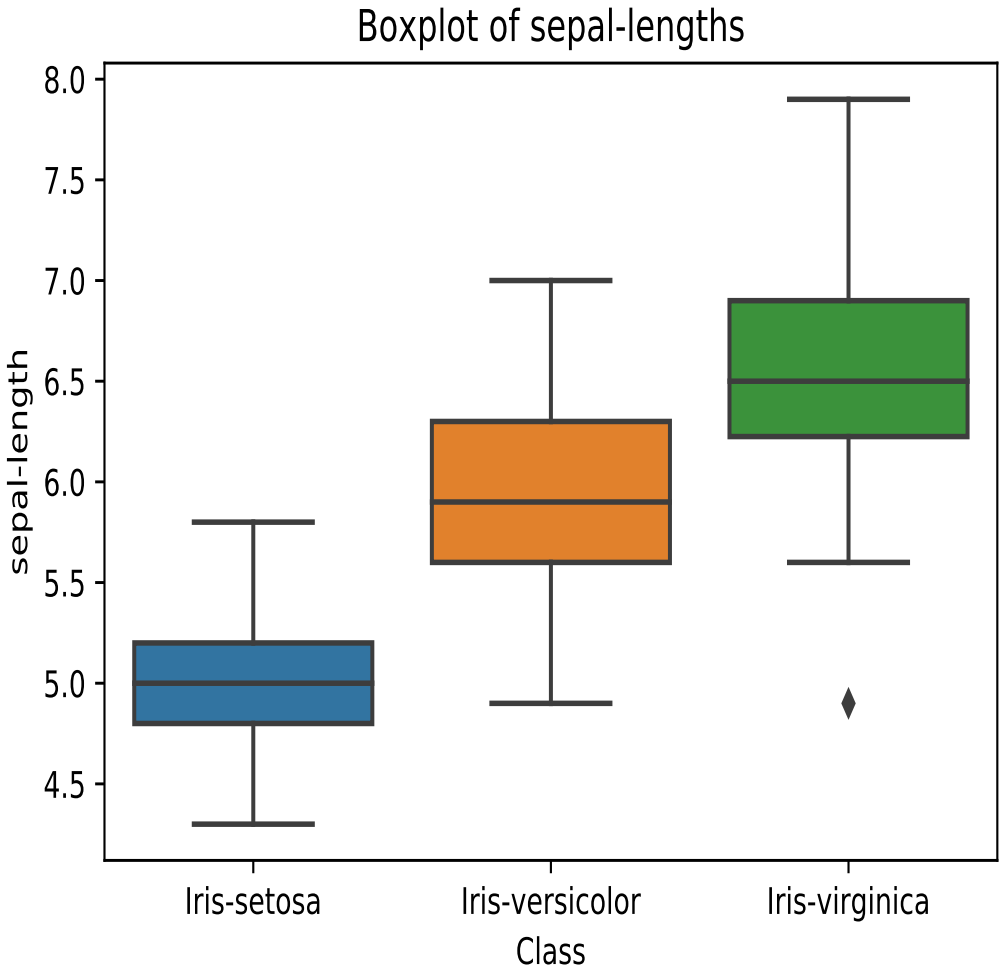
Bar Plots


sepal width vs length
sepal-width
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
sepal-length

Box Plots

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

box, ax = plt.subplots()
ax = sns.boxplot(x = "Class",
                 y = "sepal-length",
                 data = data)
ax.set_title("Boxplot of sepal-lengths")
ax.set_xlabel("Class")
ax.set_ylabel("sepal-length")
plt.show()
```

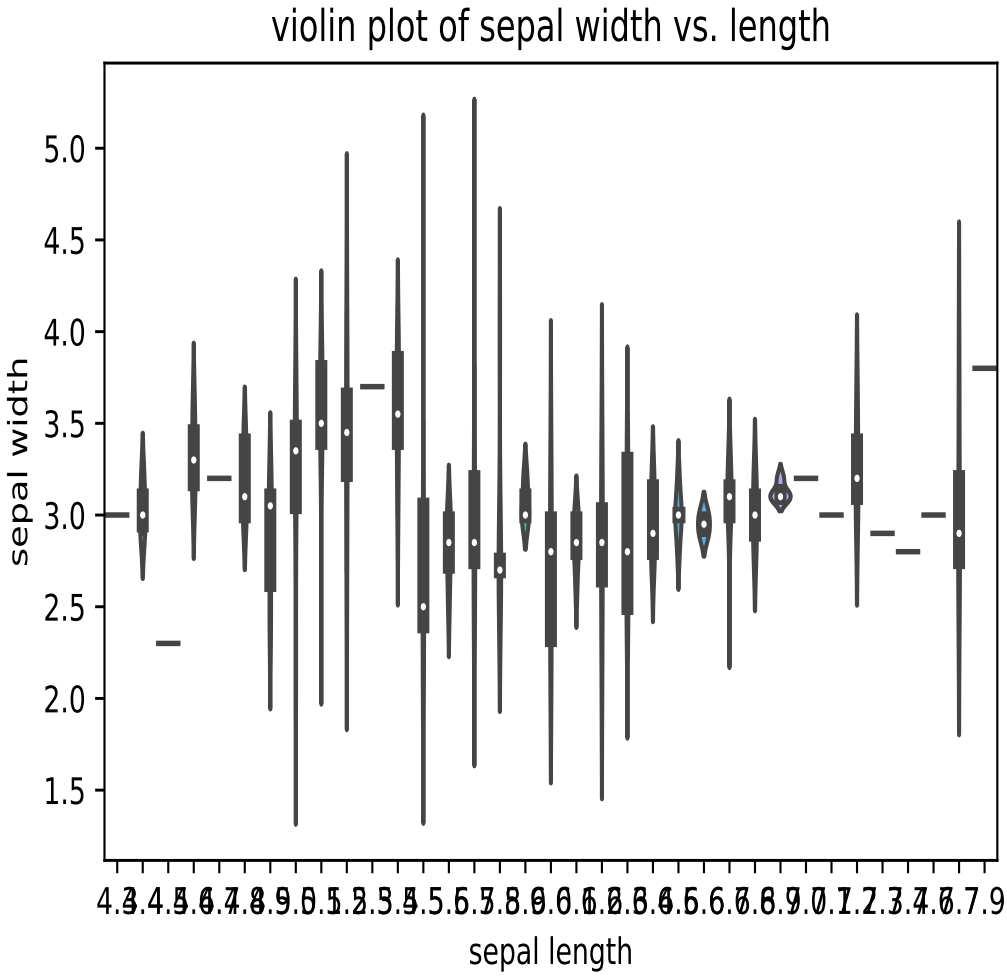
Box Plots


Boxplot of sepal-lengths
sepal-length
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
Iris-setosa
Iris-versicolor
Iris-virginica
Class

Violin Plots

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

violin, ax = plt.subplots()
ax = sns.violinplot(x = "sepal-length",
                    y = "sepal-width",
                    data = data)
ax.set_title("violin plot of sepal \
              width vs. length")
ax.set_xlabel("sepal length")
ax.set_ylabel("sepal width")
plt.show()
```

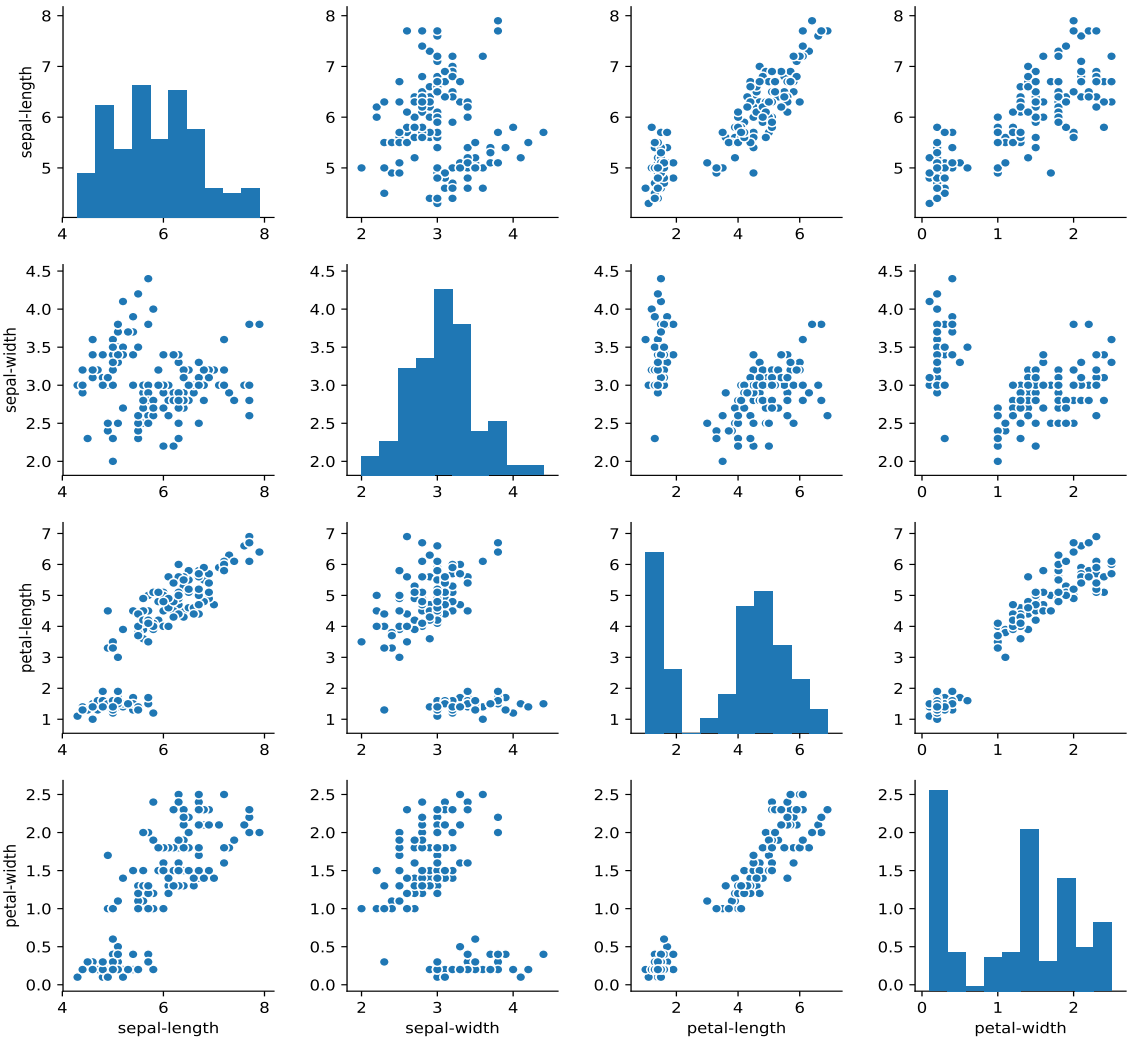
Violin Plots



Pairwise Relationships

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

features = ["sepal-length", "sepal-width",
             "petal-length", "petal-width"]

pair_plot = sns.pairplot(data[features])
plt.show()
```

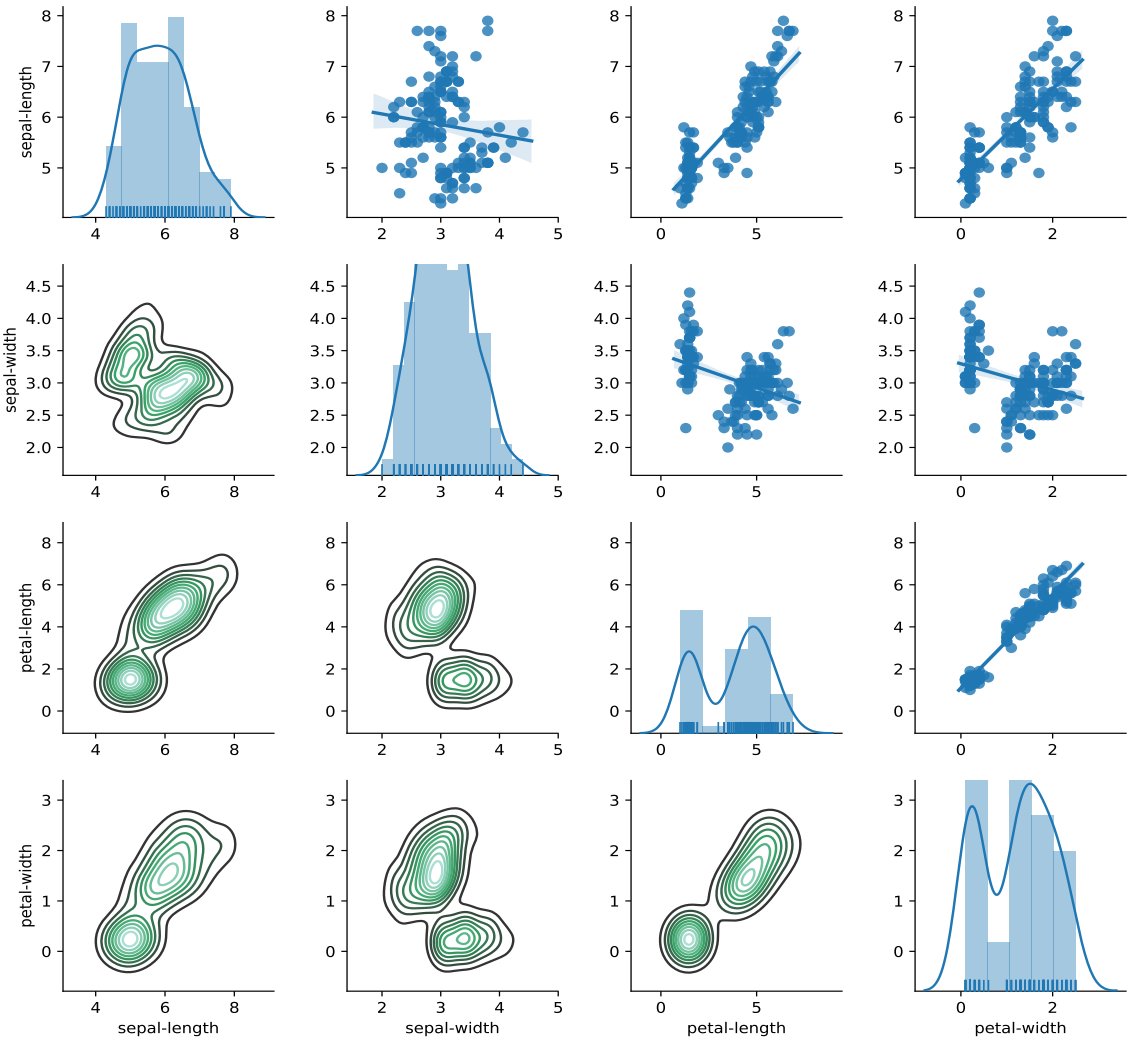
Pairwise Relationships



Specific Pairwise Relationships

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

features = ["sepal-length", "sepal-width",
            "petal-length", "petal-width"]

pair_grid = sns.PairGrid(data[features])
pair_grid = pair_grid.map_upper(sns.regplot)
pair_grid = pair_grid.map_lower(sns.kdeplot)
pair_grid = pair_grid.map_diag(sns.distplot,
                               rug=True)
plt.show()
```

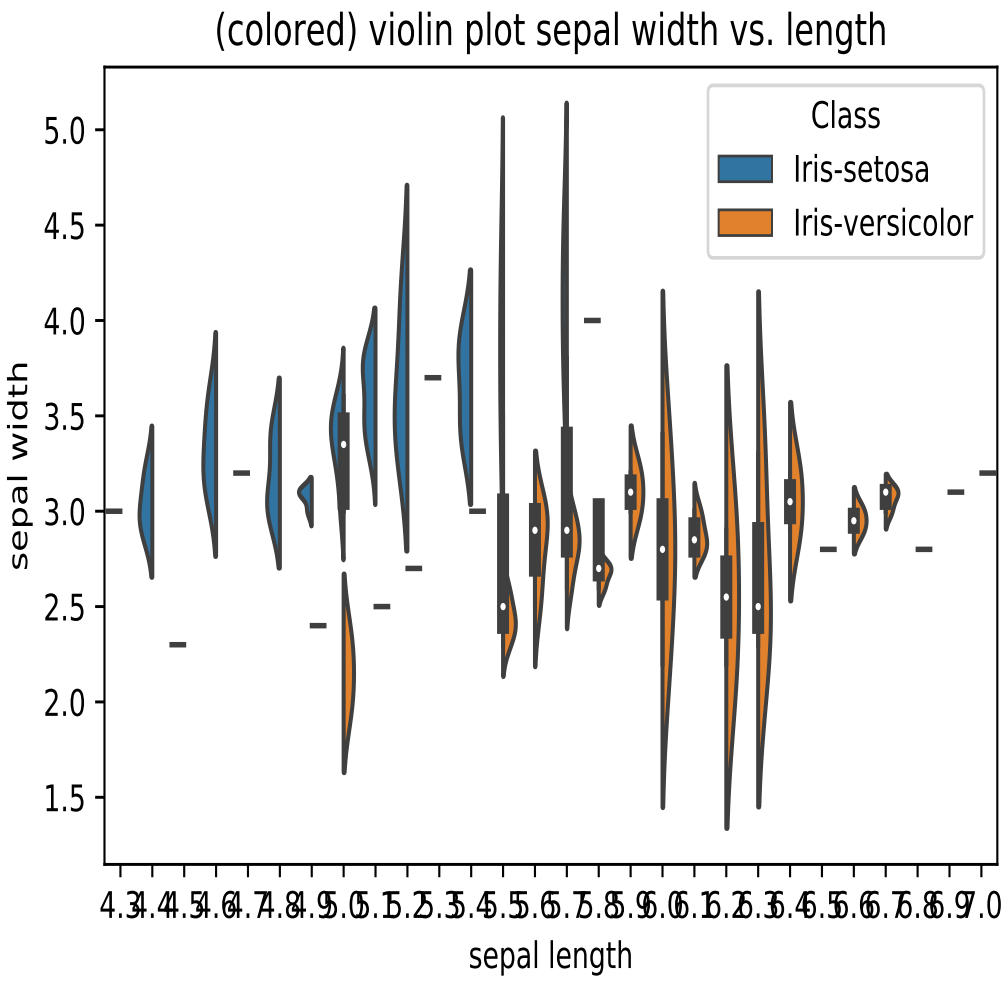
Specific Relationships



Colored Violin Plot

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
       "sepal-length", "sepal-width",
       "petal-length", "petal-width",
           "Class"])

colored_violin, ax = plt.subplots()
new_data =data[data["Class"].isin([
         "Iris-setosa","Iris-versicolor"])]
ax = sns.violinplot(x = "sepal-length",
          y = "sepal-width", hue="Class",
          data = new_data, split = True)
ax.set_title("(colored) violin plot sepal \
                  width vs. length")
ax.set_xlabel("sepal length")
ax.set_ylabel("sepal width")
plt.show()
```

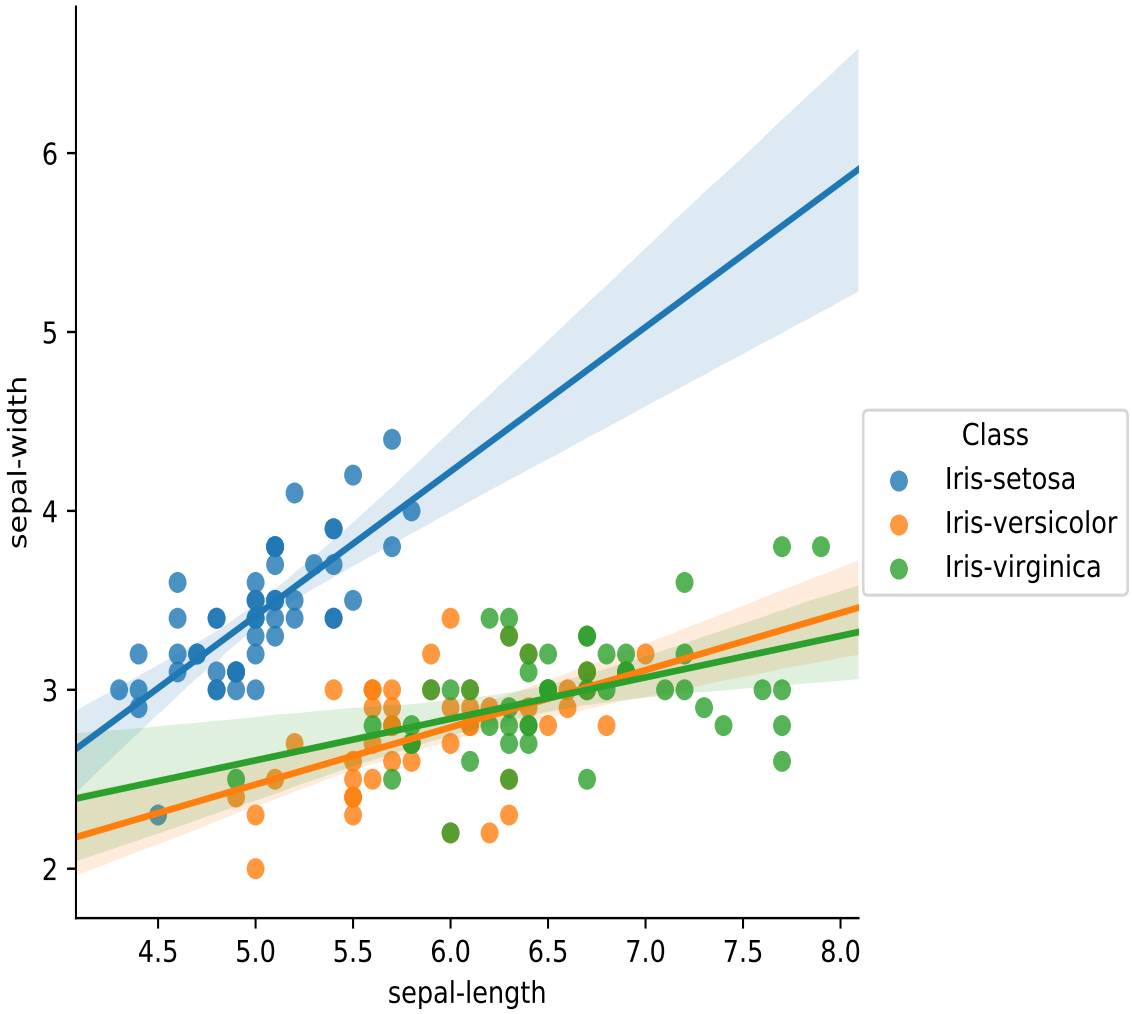
Colored Violin Plot



Regression Plot by Class

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

fig = sns.lmplot(x = "epal-length",
            y = "sepal-width", hue = "Class",
            data = data, fit_reg=True)

plt.show()
```

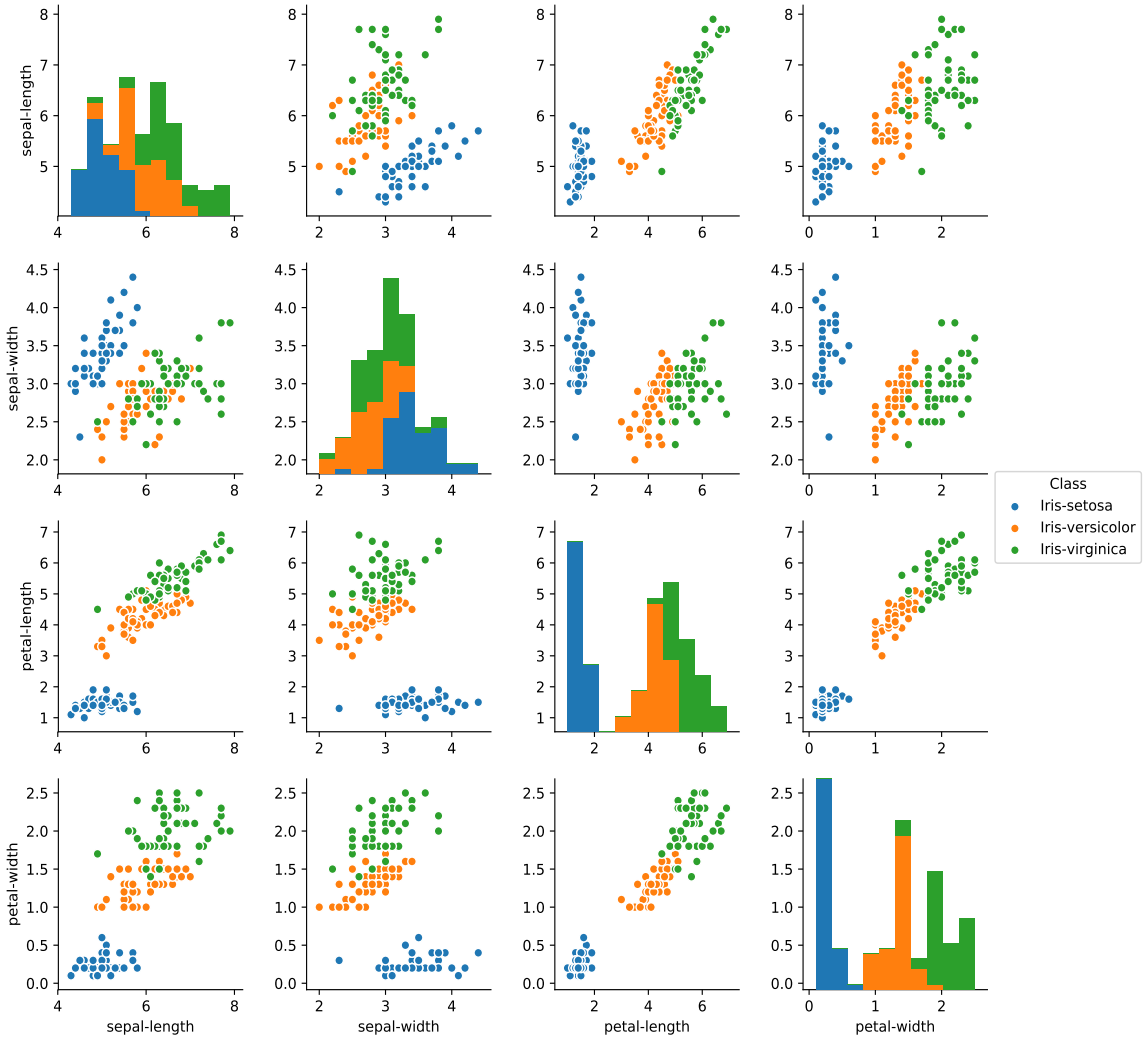
Regression Plot By Class Illustration



Colored Pair Plots

```
import numpy as np
import pandas as pdf

url = r"https://archive.ics.uci.edu/ml/" \
 + r"machine-learning-databases/iris/iris.data"

data = pd.read_csv(url, names=[
        "sepal-length", "sepal-width",
        "petal-length", "petal-width",
            "Class"])

features = ["sepal-length", "sepal-width",
            "petal-length", "petal-width",
            "Class"]

fig=sns.pairplot(data[features], hue="Class")
plt.show()
```

Colored Pair Plot



Concepts Check:

(a) iris labels and features
(b) statistics
(c) histograms
(d) scatterplots and counts
(e) counting
(f) bar and violin plots
(g) pair plots