Statistical Thinking in Python Part 1

1). Graphical exploratory data analysis

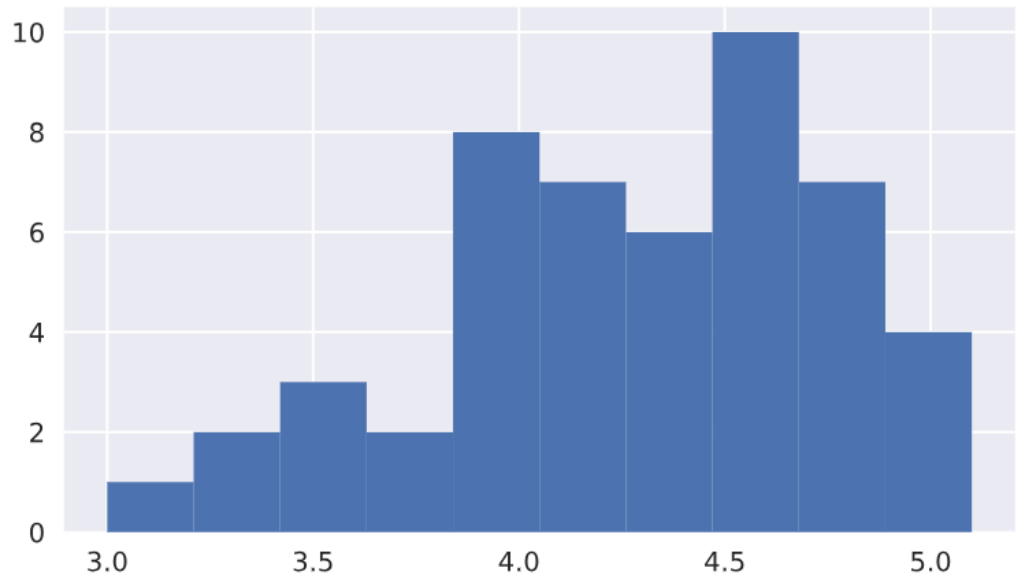
a). Plotting a histogram of Iris data

```
# Import plotting modules
import matplotlib.pyplot as plt
import seaborn as sns

# Set default Seaborn style
sns.set()

# Plot histogram of versicolor petal lengths
plt.hist(versicolor_petal_length)

# Show histogram
plt.show()
```

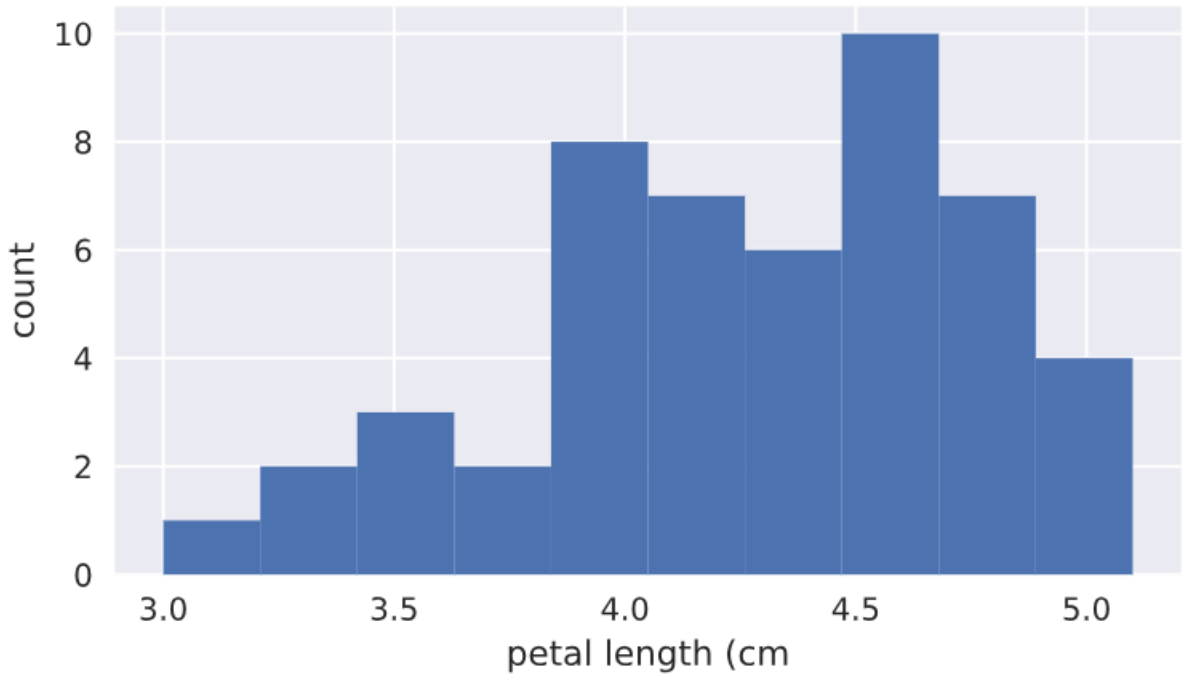


b). Labeling axes

```
# Plot histogram of versicolor petal lengths
_ = plt.hist(versicolor_petal_length)

# Label axes
plt.xlabel('petal length (cm')
plt.ylabel('count')

# Show histogram
plt.show()
```



c). Adjusting no of bins in histogram

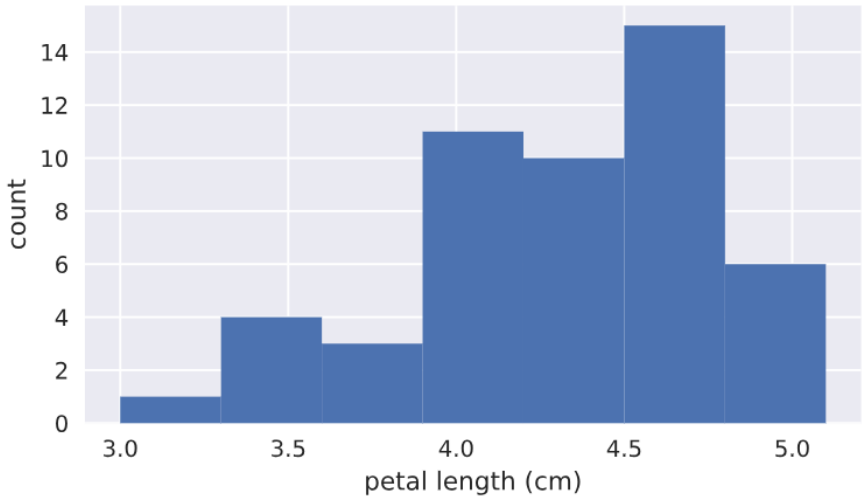
```
# Import numpy
import numpy as np
# Compute number of data points: n_data
n_data=len(versicolor_petal_length)

# Number of bins is the square root of number of data points: n_bins
n_bins=np.sqrt(n_data)

# Convert number of bins to integer: n_bins
n_bins=int(n_bins)

# Plot the histogram
plt.hist(versicolor_petal_length, bins=n_bins)

# Label axes
_ = plt.xlabel('petal length (cm)')
_ = plt.ylabel('count')
# Show histogram
plt.show()
```

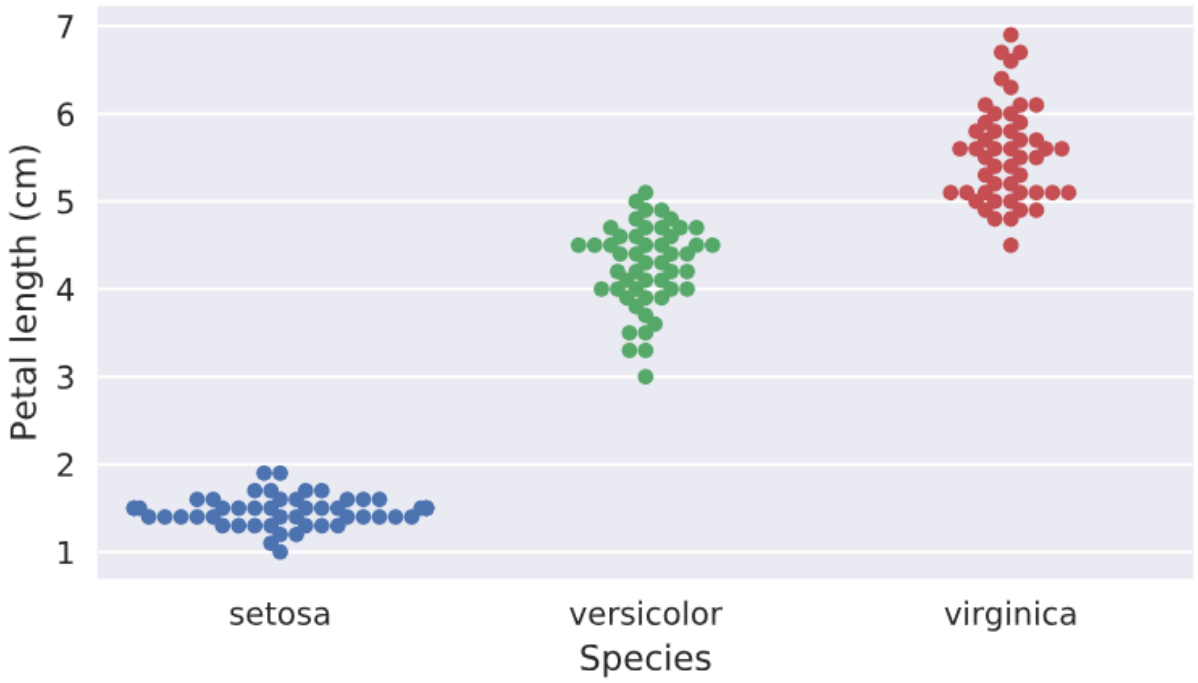


d). Bee swarm plot

```
# Create bee swarm plot with Seaborn's default settings
_=sns.swarmplot(x='species',y='petal length (cm)', data=df)

# Label the axes
_=plt.xlabel('Species')
_=plt.ylabel('Petal length (cm)')

# Show the plot
plt.show()
```



e). Computing the ECDF

```
def ecdf(data):
    """Compute ECDF for a one-dimensional array of measurements."""
    # Number of data points: n
    n = len(data)

    # x-data for the ECDF: x
    x = np.sort(data)

    # y-data for the ECDF: y
    y = np.arange(1, n+1) / n

    return x, y
```

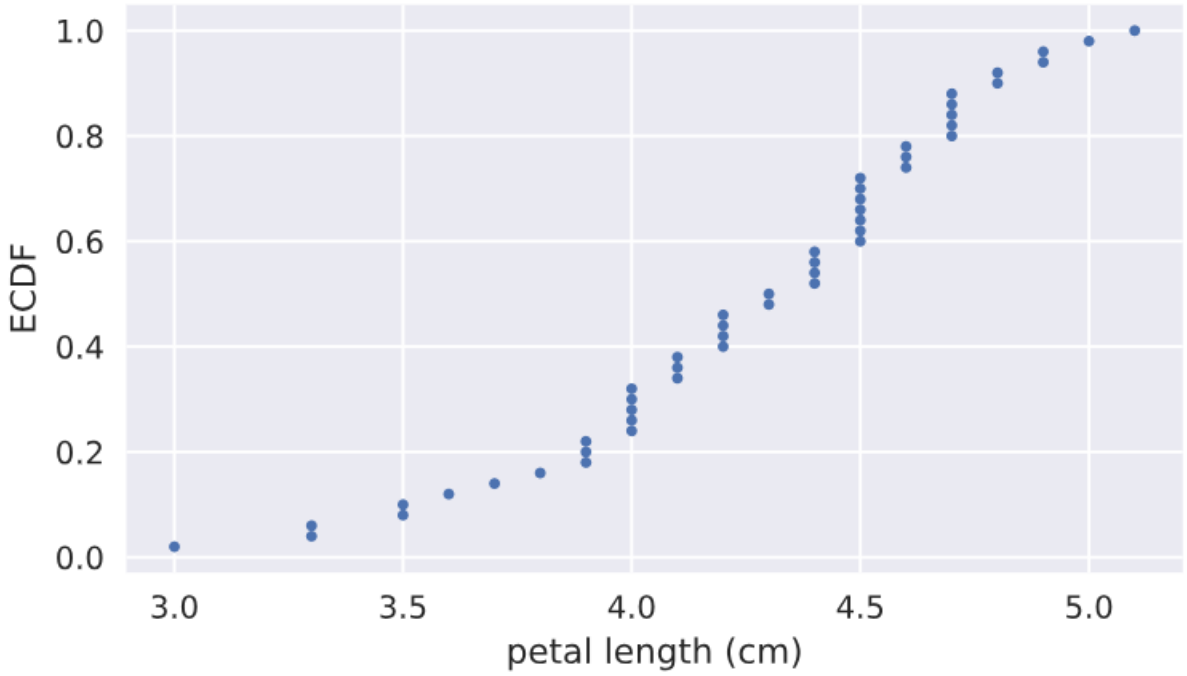
f). Plotting the ECDF

```
# Compute ECDF for versicolor data: x_vers, y_vers
x_vers, y_vers = ecdf(versicolor_petal_length)

# Generate plot
plt.plot(x_vers, y_vers, marker='.', linestyle='none')

# Label the axes
plt.xlabel('petal length (cm)')
plt.ylabel('ECDF')

# Display the plot
plt.show()
```



g). <u>Comparison of ECDF</u>

```
# Compute ECDFs
x_set, y_set=ecdf(setosa_petal_length)
x_vers, y_vers=ecdf(versicolor_petal_length)
x_virg, y_virg=ecdf(virginica_petal_length)

# Plot all ECDFs on the same plot
plt.plot(x_set,y_set,marker='.',linestyle='none')
plt.plot(x_vers,y_vers,marker='.',linestyle='none')
plt.plot(x_virg,y_virg,marker='.',linestyle='none')

# Annotate the plot
plt.legend(('setosa', 'versicolor', 'virginica'), loc='lower right')
_ = plt.xlabel('petal length (cm)')
_ = plt.ylabel('ECDF')
# Display the plot
plt.show()
```