STAT3612 Lecture 2

Data Exploration

Dr. Aijun Zhang

8 September 2020

Department of 統計及精算學系
Statistics & Actuarial Science

Table of Contents

John Tukey

John Tukey (1915–2000)
Wikipedia

- Proposed "Exploratory Data Analysis"
- Coined terms: Boxplot, Stem-and-Leaf plot, ANOVA (Analysis of Variance)
- Also coined terms "Bit" and "Software"
- Co-Developed famous methods:
  Fast Fourier Transform, Projection Pursuit, Jackknife Estimation
- Famous quote:
  "The best thing about being a statistician is that you get to play in everyone's backyard."

John Tukey: The Future of Data Analysis

Excerpt from Donoho (2015) "50 years of Data Science"

3 *The Future of Data Analysis*, 1962

This paper was prepared for the John Tukey centennial. More than 50 years ago, John prophecied that something like today's Data Science moment would be coming. In "The Future of Data Analysis" [42], John deeply shocked his readers (academic statisticians) with the following introductory paragraphs:[16]

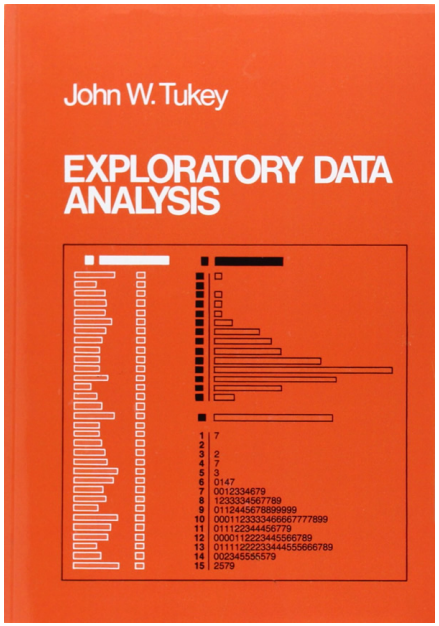
> *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt. ... All in all I have come to feel that my central interest is in data analysis, which I take to include, among other things: procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data*

This paper was published in 1962 in "The Annals of Mathematical Statistics", the central venue for mathematically-advanced statistical research of the day. Other articles appearing in that journal

[16] One questions why the journal even allowed this to be published! ...

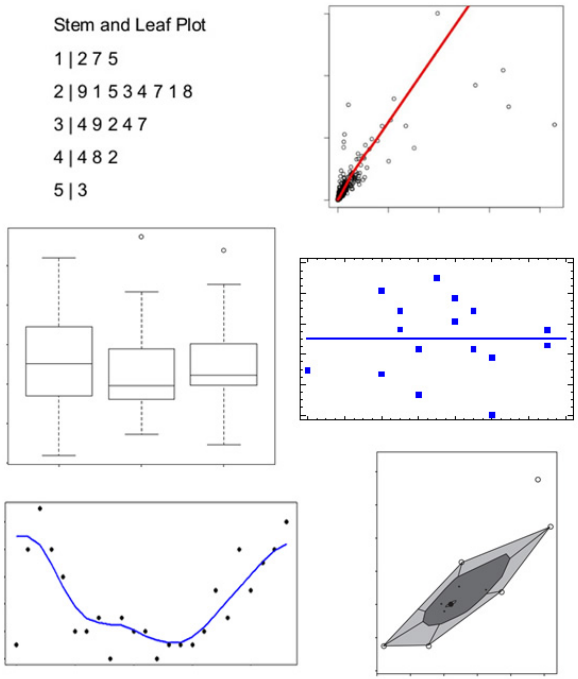
Reference: Donoho, David (2017). 50 Years of Data Science. *Journal of Computational and Graphical Statistics*, 26(4), 745-766.

John Tukey: Exploratory Data Analysis



John Tukey (1977)

- Stem-and-Leaf plot
- Scatter plot
- Box-plot, Outliers
- Residual plot
- Smoother
- Bag plot
- Five-number summary



"The greatest value of a picture is when it forces us to notice what we never expected to see." (John Tukey, 1977)

Exploratory Data Analysis (EDA)

The EDA is a statistical approach to make sense of data by using a variety of techniques (mostly graphical). It may help

- Assess assumption about the variable distribution
- Identify relationship between variables
- Extract important variables
- Suggest use of appropriate models
- Detect problems of the collected data (e.g. outliers, missing data, measurement errors)

Table of Contents

Iris Dataset

In [1]:
```
from sklearn import datasets
iris = datasets.load_iris()
print(iris['DESCR'])
```

```
Iris Plants Database
====================

Notes
-----
Data Set Characteristics:
    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
```

iris setosa

iris versicolor

iris virginica

In [2]:
```
import pandas as pd
X = pd.DataFrame(iris.data, columns=iris['feature_names'])
X.head()
```

Out[2]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [3]:
```
X.shape
```

Out[3]:
```
(150, 4)
```

In [4]:
```
X.describe()
```

Out[4]:

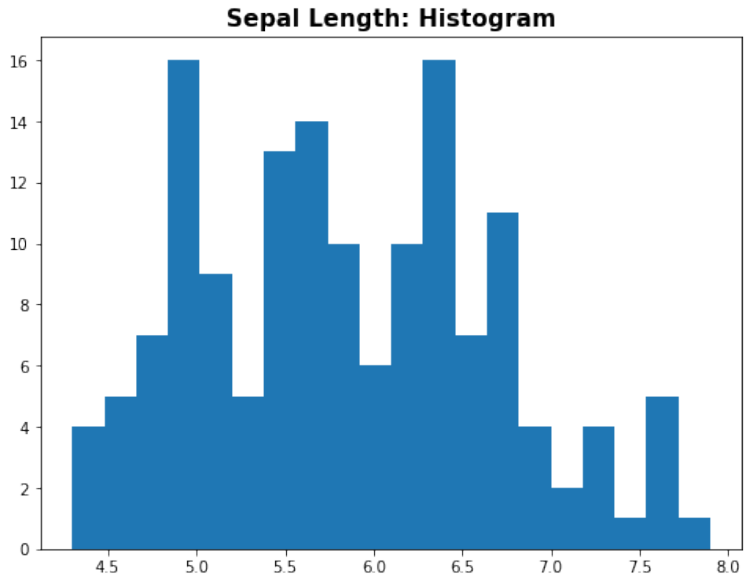
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000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

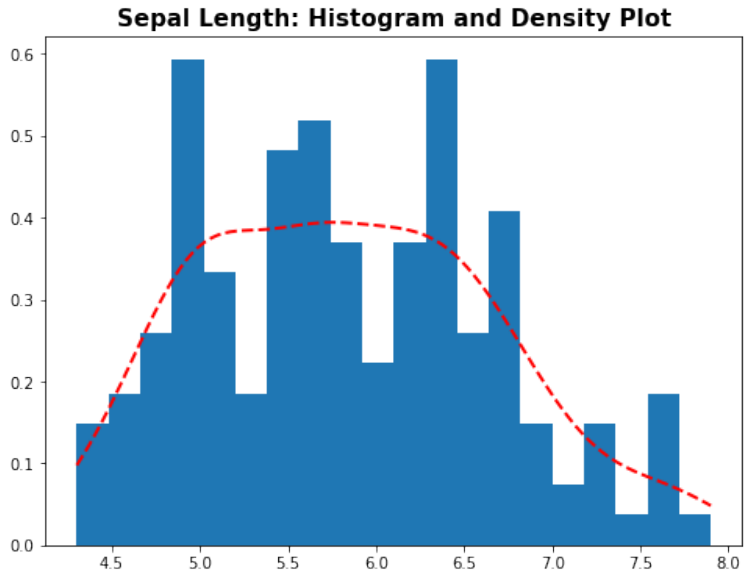
Basic Plots with Matplotlib

- Histogram and Density Plots
- Boxplot
- Bar Chart
- Scatter Plot

Histogram and Density Plot

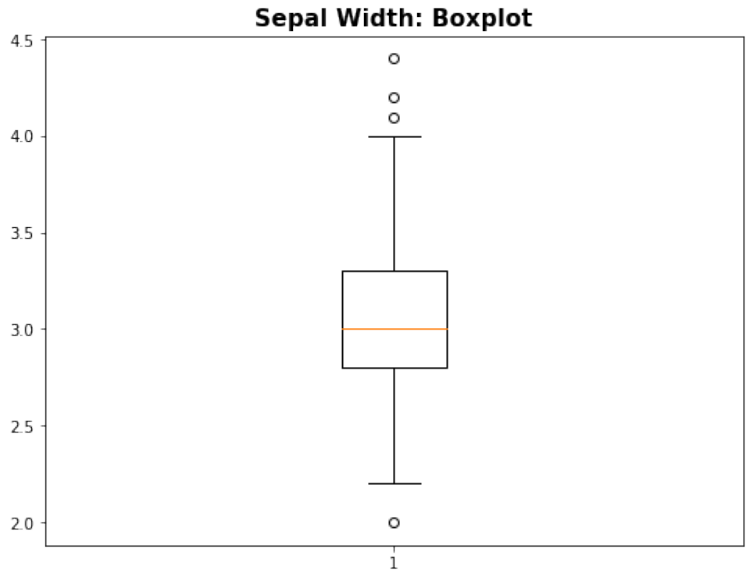
Use histogram and density plot to check distribution of a numerical variable:

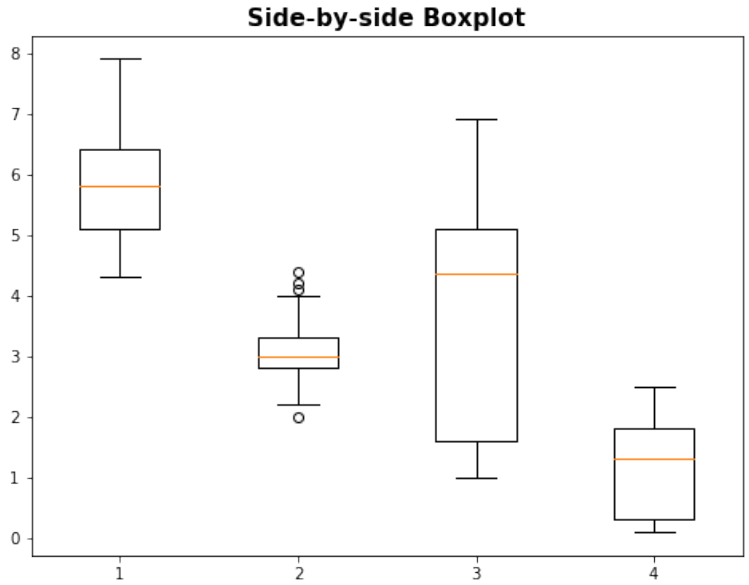




Boxplot

Use boxplot (univariate and side-by-side) to check quantiles and outliers:

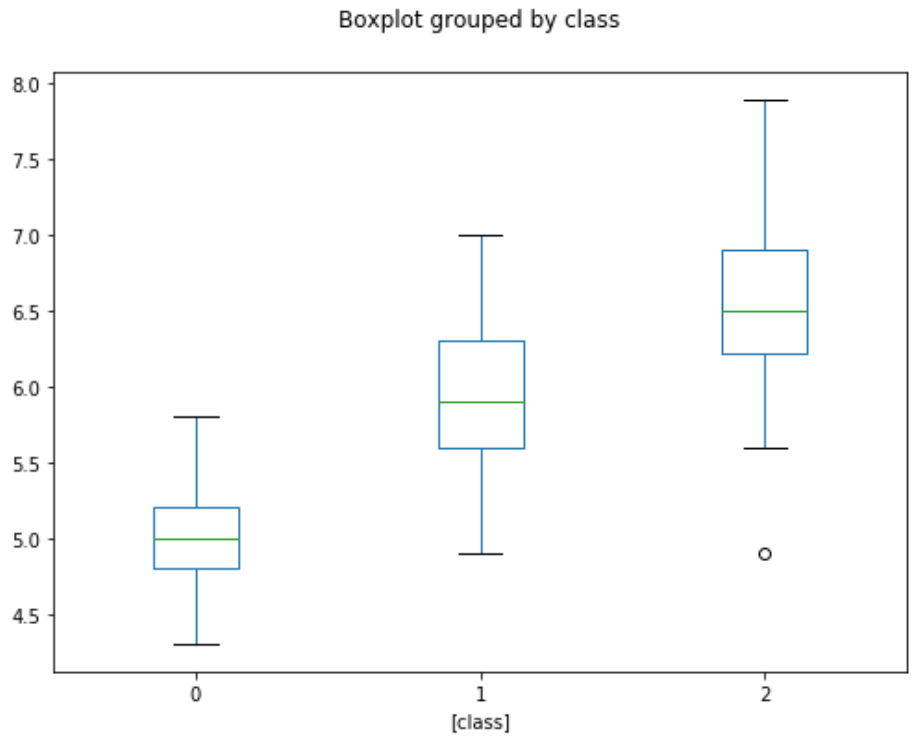




Boxplot

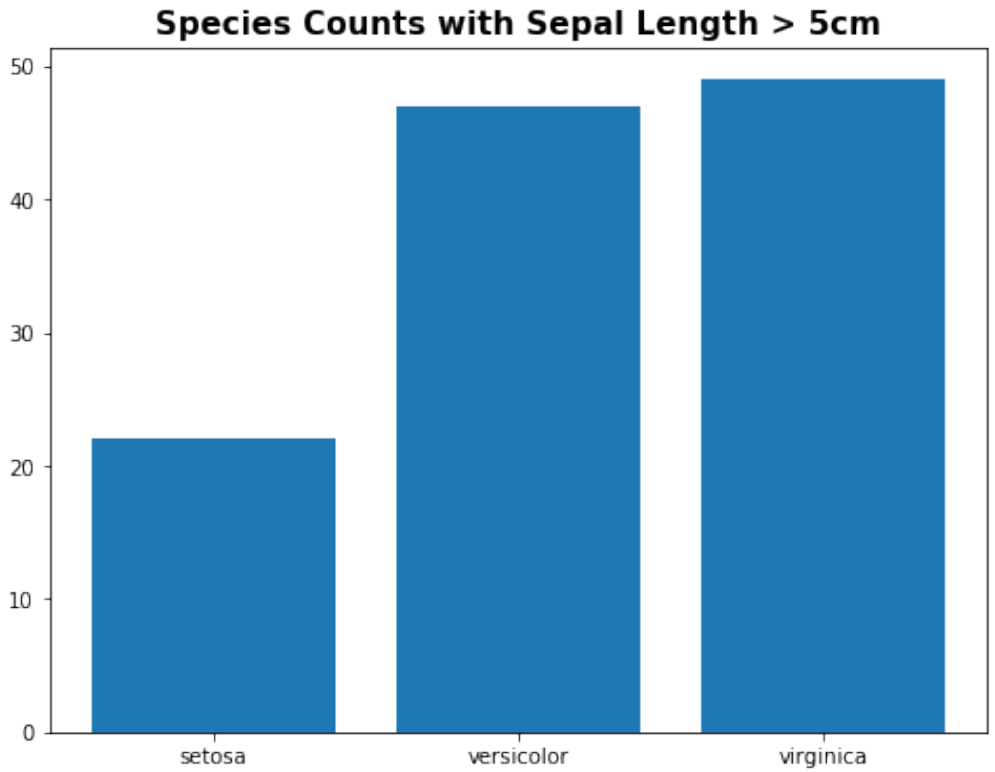
Compare to the boxplot with grouping variable:

```
Data[['sepal length (cm)', 'class']].boxplot(by="class",figsize=(8,6))
plt.title('');
plt.grid('off')
```



Bar Chart for Categorical Variables

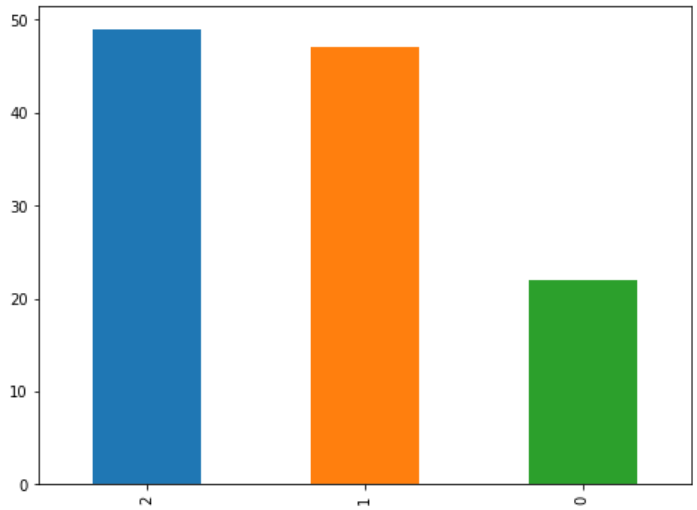
Use bar chart to check the distribution of a categorical variable:



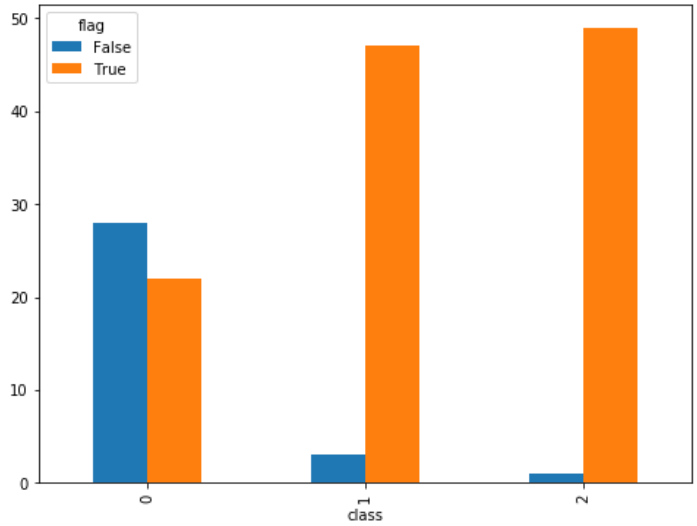
Bar Chart for Categorical Variables

Use clustered bar chart to plot the two-way contingency table:

```
pd.value_counts(Data[Data['flag']]['class']).plot.bar(figsize=(8,6))
plt.show()
```

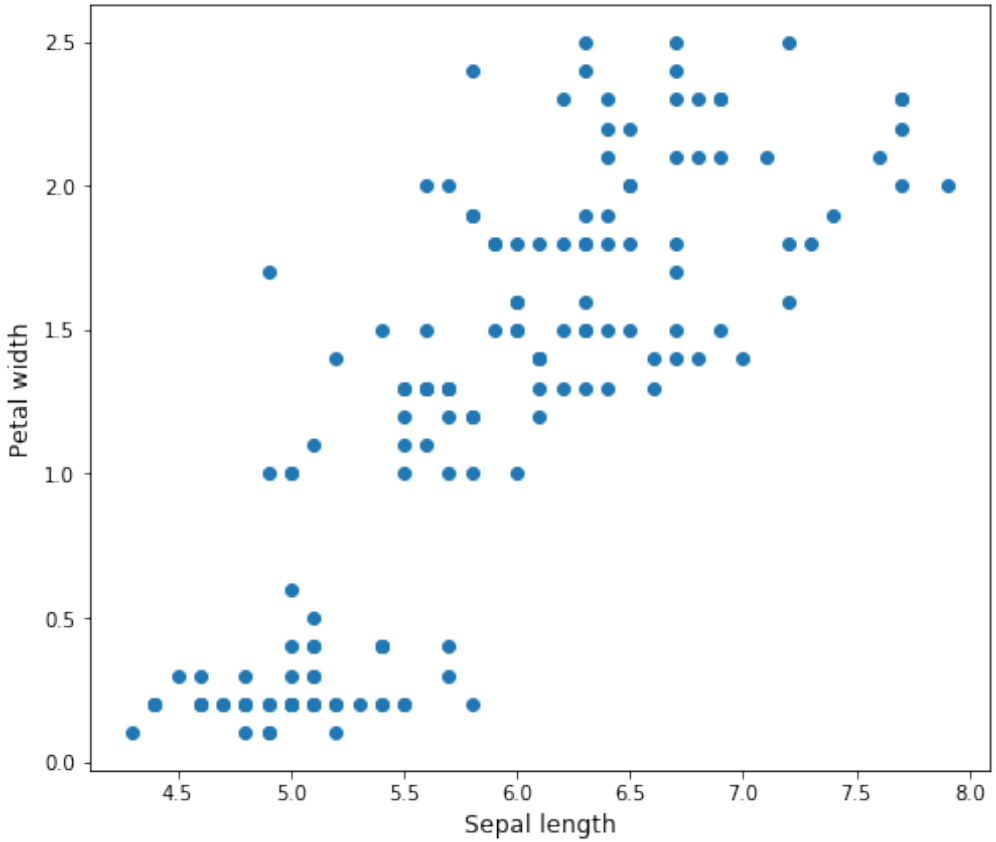



```
pd.crosstab(Data['class'], Data['flag']).plot.bar(figsize=(8,6))
plt.show()
```



Scatter Plot

Use scatter plot (also called xyplot) to check relationship between variables:



Scatter Plot

More sophisticated: adding colors and linear regression fit

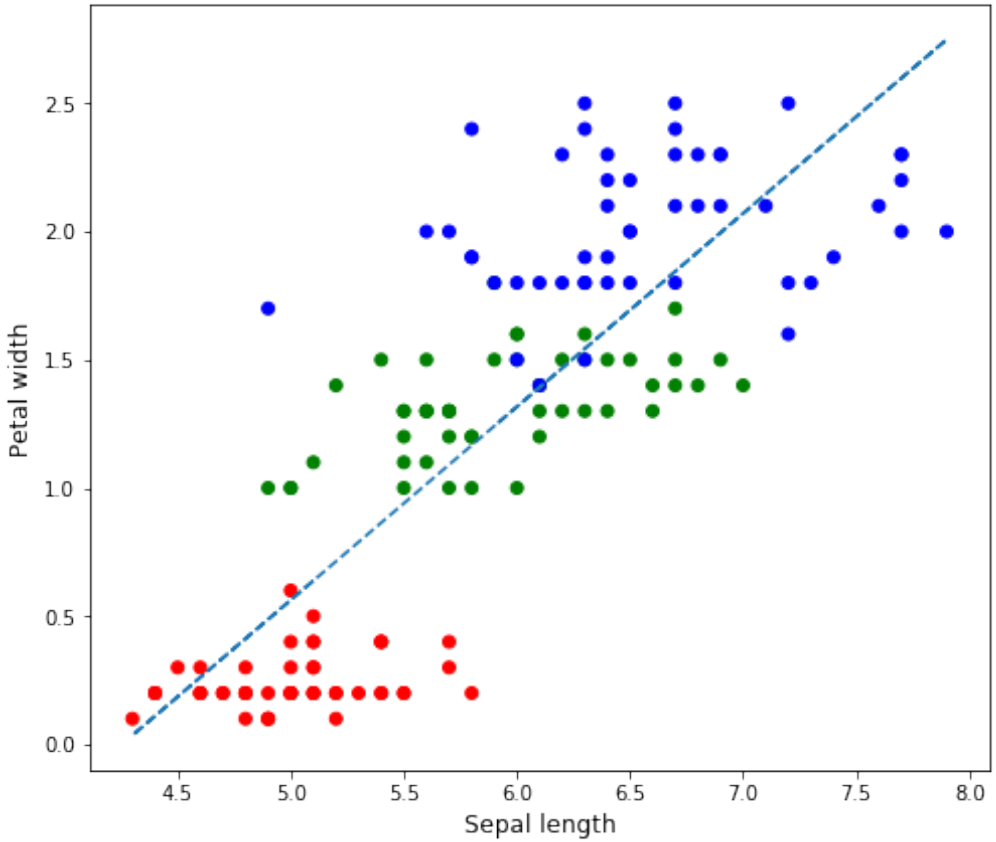


Table of Contents

Pandas for Data Exploration

Data Manipulation:

- Data merge/join
- Data subset/filter, sort/arrange
- New variable creation/mutation
- Group_by Summary

Data Visualization:

- Mixture Histograms
- Side-by-side Boxplots
- Stacked Bar Charts
- Pairwise Scatter Plot

Data Merge/Join

```
Data = pd.merge(X, y, left_index = True, right_index=True)
Data.head(10)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5.0	3.4	1.5	0.2	0
8	4.4	2.9	1.4	0.2	0
9	4.9	3.1	1.5	0.1	0

Data Subset/Filter

```
tmp = Data[(Data['class']==1) & (Data['sepal length (cm)']> 6.6)]
tmp
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
50	7.0	3.2	4.7	1.4	1
52	6.9	3.1	4.9	1.5	1
65	6.7	3.1	4.4	1.4	1
76	6.8	2.8	4.8	1.4	1
77	6.7	3.0	5.0	1.7	1
86	6.7	3.1	4.7	1.5	1

Data Sort/Arrange

```
tmp.sort_values(by = ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)'],
                ascending = [True, True, False])
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
77	6.7	3.0	5.0	1.7	1
86	6.7	3.1	4.7	1.5	1
65	6.7	3.1	4.4	1.4	1
76	6.8	2.8	4.8	1.4	1
52	6.9	3.1	4.9	1.5	1
50	7.0	3.2	4.7	1.4	1

New Variable Create/Mutate

```
Data['sepal.size'] = Data['sepal length (cm)']*Data['sepal width (cm)']
Data['petal.size'] = Data['petal length (cm)']*Data['petal width (cm)']
Data.head(8)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class	sepal.size	petal.size
0	5.1	3.5	1.4	0.2	0	17.85	0.28
1	4.9	3.0	1.4	0.2	0	14.70	0.28
2	4.7	3.2	1.3	0.2	0	15.04	0.26
3	4.6	3.1	1.5	0.2	0	14.26	0.30
4	5.0	3.6	1.4	0.2	0	18.00	0.28
5	5.4	3.9	1.7	0.4	0	21.06	0.68
6	4.6	3.4	1.4	0.3	0	15.64	0.42
7	5.0	3.4	1.5	0.2	0	17.00	0.30

Group_by Summary

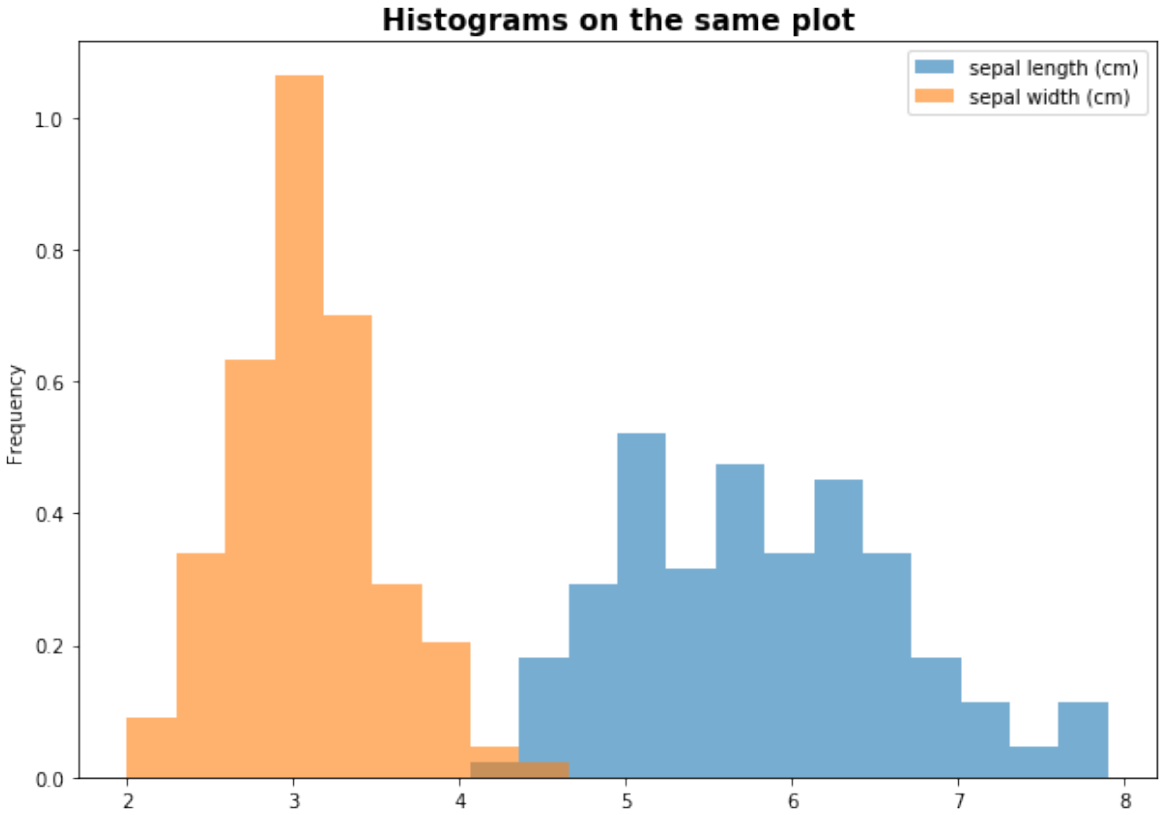
```
Data.groupby('class').mean()
```

class	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	sepal.size	petal.size
0	5.006	3.428	1.462	0.246	17.2578	0.3656
1	5.936	2.770	4.260	1.326	16.5262	5.7204
2	6.588	2.974	5.552	2.026	19.6846	11.2962

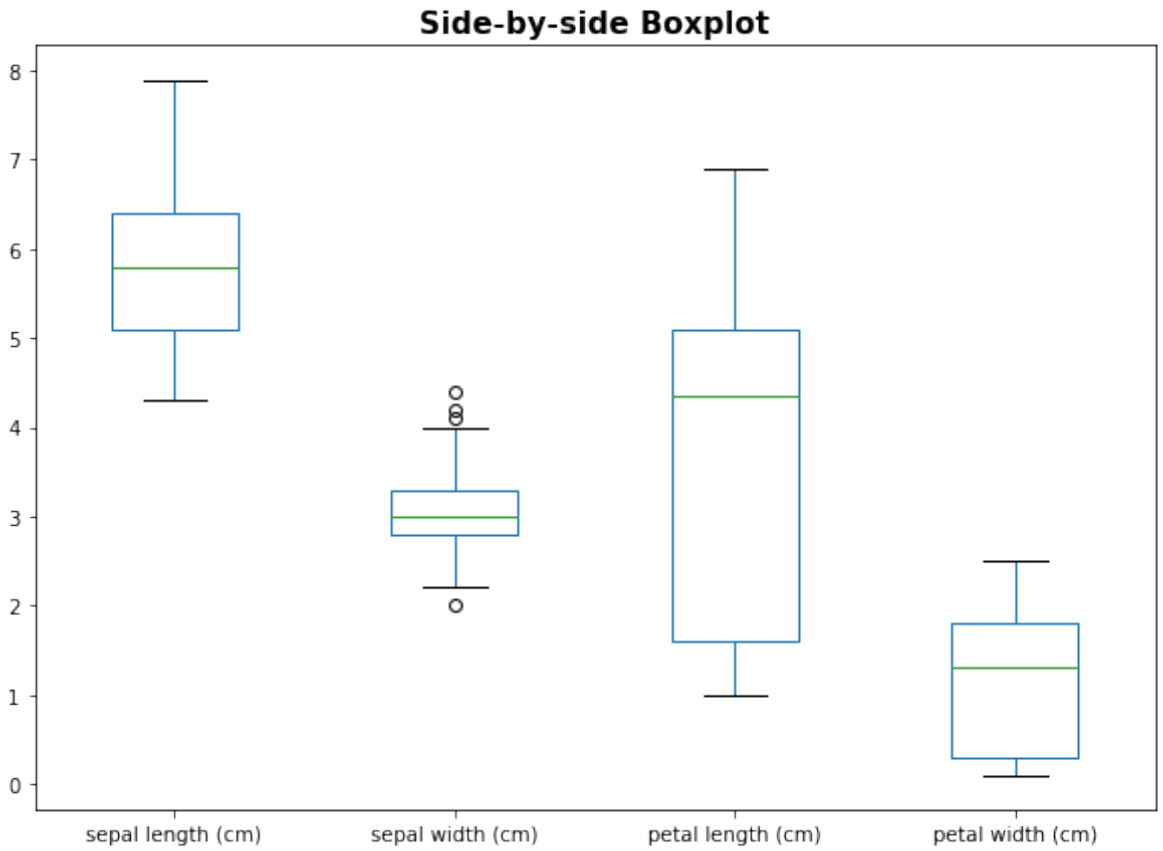
```
Data.groupby('class').var()
```

class	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	sepal.size	petal.size
0	0.124249	0.143690	0.030159	0.011106	8.607034	0.032817
1	0.266433	0.098469	0.220816	0.039106	8.219012	1.872526
2	0.404343	0.104004	0.304588	0.075433	11.963180	4.654428

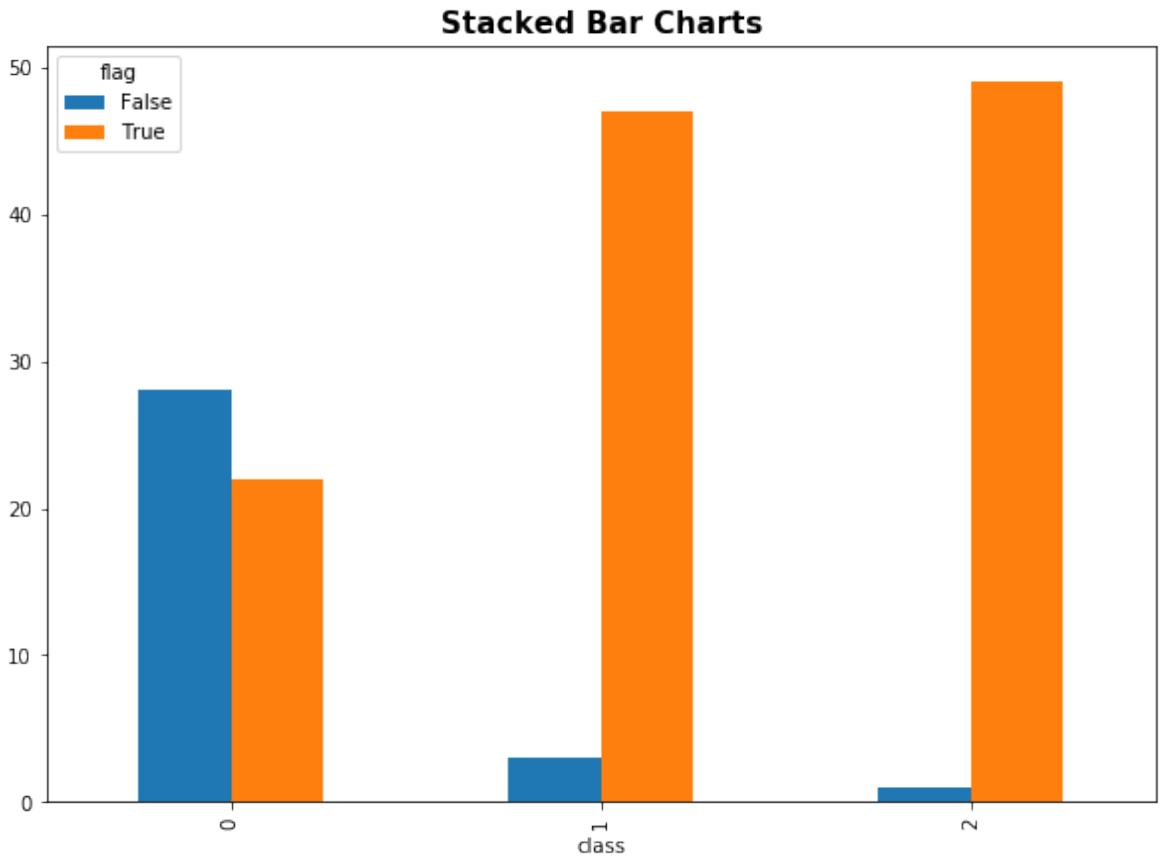
Mixture Histograms



Side-by-side Boxplots



Stacked Bar Charts



Pairwise Scatter Plot

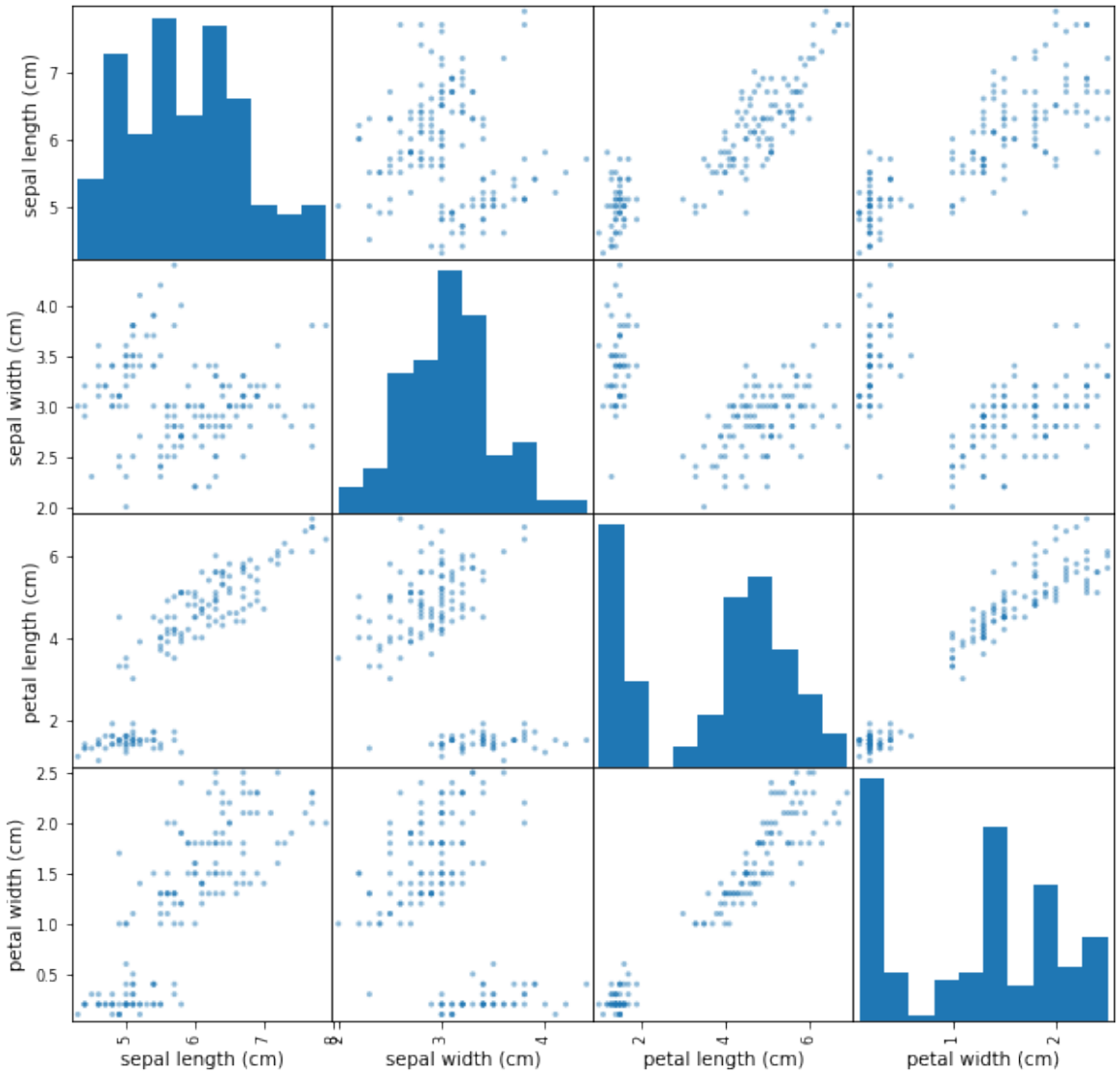


Table of Contents

Next Level of Data Visualizatio in Pythonn

- Python Seaborn
- Python ggplot2
- Python Plotly
- Alternatively, R:ggplot2/dplyr tools in Stat3622 lecture notes.

Thank You!

Q&A or Email ajzhang@umich.edu