Tikrit University
Computer Science Dept.

Master Degree
Lecture -3-

Associate Professor

Dr. Eng. Zaidoon.T.AL-Qaysi

College of Computer Science and Mathematics

(2023-2024)

- **Data exploration,** also known as exploratory data analysis, provides a set of tools to obtain fundamental understanding of a dataset. The results of data exploration can be extremely powerful in grasping the structure of the data, the distribution of the values, and the presence of extreme values and the interrelationships between the attributes in the dataset. Data exploration also provides guidance on applying the right kind of further statistical and data science treatment. Exploration can be broadly classified into two types' **descriptive statistics** and **data visualization.**

- **Descriptive statistics** is the process of condensing key characteristics of the dataset into simple numeric metrics. Some of the common quantitative metrics used are mean, standard deviation, and correlation.

- **Visualization** is the process of projecting the data, or parts of it, into multi-dimensional space or abstract images. All the useful (and adorable) charts fall under this category. Data exploration in the context of data science uses both descriptive statistics and visualization techniques.

- In the data science process, **data exploration** is leveraged in many different steps including preprocessing or data preparation, modeling, and interpretation of the modeling results.

- **Data Sample:** A subset of observations from a group. ˆ

- **Data Population:** All possible observations from a group.

- The **Iris dataset** is used for learning data science mainly because it is simple to understand, explore, and can be used to illustrate how different data science algorithms approach the problem on the same standard dataset.

- In general, descriptive analysis covers the following characteristics of the sample or population dataset. Descriptive statistics can be broadly classified into **univariate** and **multivariate** exploration depending on the number of attributes under analysis.

- A variable that is thought to be controlled or not affected by other variables is called an **independent variable.** A variable that depends on other variables (most often other independent variables) is called a **dependent variable.** In the case of a prediction problem, an independent variable is also called a **predictor variable** and a dependent variable is called an **outcome variable.**

Table 1: Dataset Descriptive Statistics

Characteristics of the Dataset	Measurement Technique
Center of the dataset	Mean, median, and mode
Spread of the dataset	Range, variance, and standard deviation
Shape of the distribution of the dataset	Symmetry, skewness, and kurtosis

- **Univariate data exploration** denotes analysis of one attribute at a time.

- **Univariate Visualization** investigating one attribute at a time using univariate charts. This techniques give an idea of how the attribute values are distributed and the shape of the distribution. An example Iris dataset for one species, I. setosa, has 50 observations and 4 attributes, as shown in Table 2. Here some of the descriptive statistics for sepal length attribute are explored.

Table 2: Iris Dataset and Descriptive Statistics

Observation	Sepal Length	Sepal Width	Petal Length	Petal Width
1	5.1	3.5	1.4	0.2
2	4.9	3.1	1.5	0.1
...	...	...	...	...
49	5	3.4	1.5	0.2
50	4.4	2.9	1.4	0.2
Statistics	**Sepal Length**	**Sepal Width**	**Petal Length**	**Petal Width**
Mean	5.006	3.418	1.464	0.244
Median	5.000	3.400	1.500	0.200
Mode	5.100	3.400	1.500	0.200
Range	1.500	2.100	0.900	0.500
Standard deviation	0.352	0.381	0.174	0.107
Variance	0.124	0.145	0.030	0.011

- A **histogram** is one of the most basic visualization techniques to understand the frequency of the occurrence of values. It shows the distribution of the data by plotting the frequency of occurrence in a range. In a histogram, the attribute under inquiry is shown on the horizontal axis and the frequency of occurrence is on the vertical axis. For a continuous numeric data type, the range or binning value to group a range of values need to be specified. For example, in the case of human height in centimeters, all the occurrences between 152.00 and 152.99 are grouped under 152. There is no optimal number of bins or bin width that works for all the distributions. If the bin width is too small, the distribution becomes more precise but reveals the noise due to sampling. A general rule of thumb is to have a number of bins equal to the square root or cube root of the number of data points.

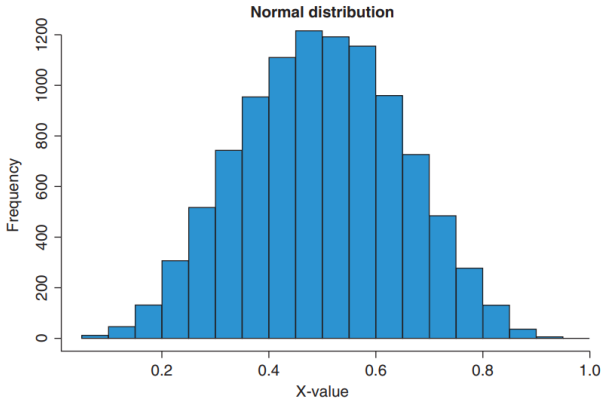


Figure 1: Example of a normal distribution.

- **Normal Distribution:** In an ideal world, data would be distributed symmetrically around the center of all scores. Thus, if we drew a vertical line through the center of a distribution, both sides should look the same. This so-called normal distribution is characterized by a bell-shaped curve, an example of which is shown in **Figure 1**. There are two ways in which a distribution can deviate from normal:

  1. Lack of symmetry (called skew).
  2. Pointiness (called kurtosis).

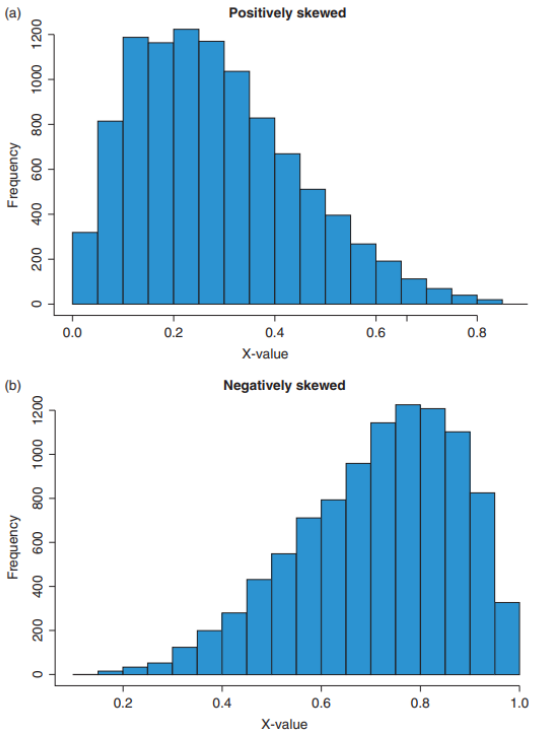


Figure 2: Examples of skewed distributions.

Table 3 represents Productivity measured in terms of output for a group of data science professionals. Some of them went through extensive statistics training (represented as "Y" in the Training column) while others did not (N). The dataset also contains the work experience (denoted as Experience) of each professional in terms of number of working hours. A histogram can be created from the numbers in the Productivity column, as shown in **Figure 3.**

Table 3: Iris Dataset and Descriptive Statistics

Productivity	Experience	Training
5	1	Y
2	0	N
10	10	Y
4	5	Y
6	5	Y
12	15	Y
5	10	Y
6	2	Y
4	4	Y
3	5	N
9	5	Y
8	10	Y
11	15	Y
13	19	Y
4	5	N
5	7	N
7	12	Y
8	15	N
12	20	Y
3	5	N
15	20	Y

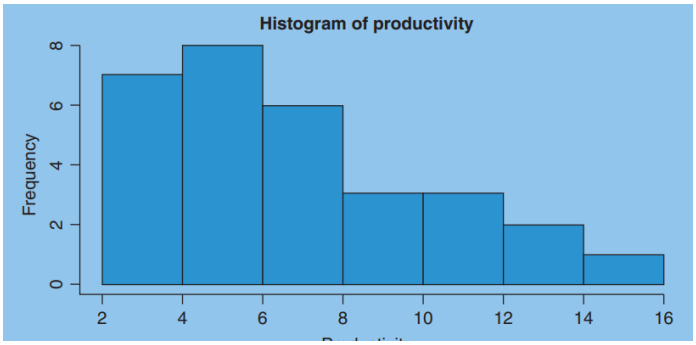


Figure 3: Histogram of productivity data

Histograms are used to find the central location, range, and shape of distribution. In the case of the petal length attribute in the Iris dataset, the data is multimodal (**Figure 4**), where the distribution does not follow the bell curve pattern. Instead, there are two peaks in the distribution. This is due to the fact that there are 150 observations of three different species (hence, distributions) in the dataset. A histogram can be stratified to include different classes in order to gain more insight. The enhanced histogram with class labels shows the dataset is made of three different distributions (**Figure 5**).

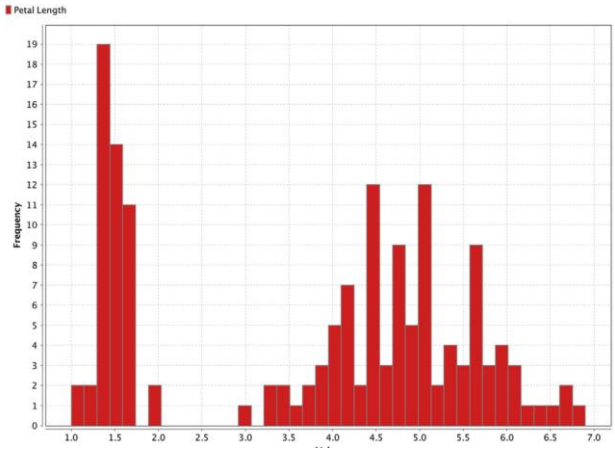


Figure 4: Histogram of petal length in Iris dataset.

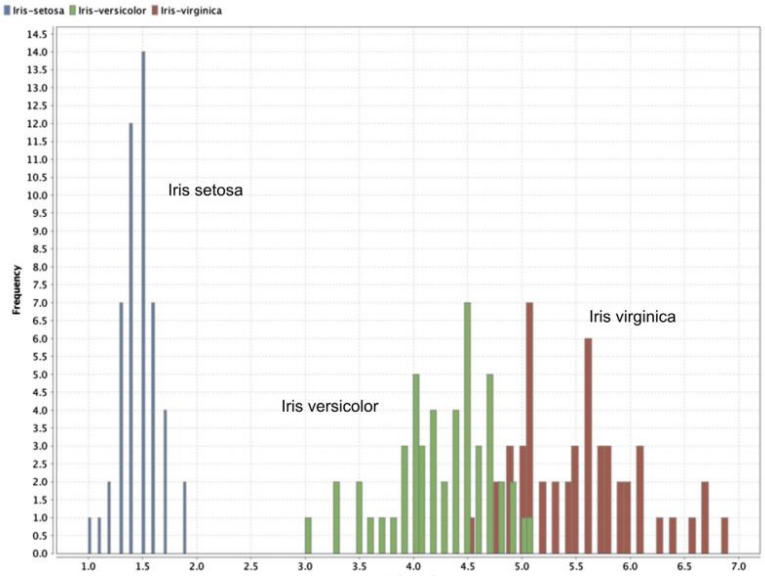


Figure 5: Class-stratified histogram of petal length in Iris dataset.

- A **histogram** worked fine for numerical data, but what about categorical data? In other words, how do we visualize the data when it's distributed in a few finite categories? We have such data in the third column called "Training." For that, we can create a pie chart, as shown in **Figure 6**.

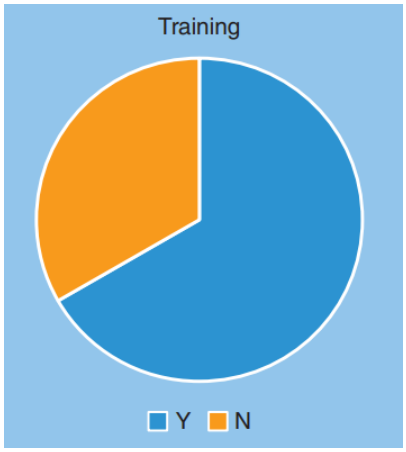


Figure 6: Pie chart showing the distribution of "Training" in the Productivity data.

- **Measure of Central Tendency**: The objective of finding the central location of an attribute is to quantify the dataset with one central or most common number.

1. **Mean:** The mean is the arithmetic average of all observations in the dataset. It is calculated by summing all the data points and dividing by the number of data points. The mean for sepal length in centimeters is 5.0060.

2. **Median:** The median is the value of the central point in the distribution. The median is calculated by sorting all the observations from small to large and selecting the mid-point observation in the sorted list. If the number of data points is even, then the average of the middle two data points is used as the median. The median for sepal length is in centimeters is 5.0000.

3. **Mode:** The mode is the most frequently occurring observation. In the dataset, data points may be repetitive, and the most repetitive data point is the mode of the dataset. In this example, the mode in centimeters is 5.1000.

- In an attribute, the mean, median, and mode may be different numbers, and this indicates the shape of the distribution. If the dataset has outliers, the mean will get affected while in most cases the median will not. The mode of the distribution can be different from the mean or median, if the underlying dataset has more than one natural normal distribution.

- **Measure of Spread:** In desert regions, it is common for the temperature to cross above 110 F during the day and drop below 30F during the night while the average temperature for a 24-hour period is around 70F. Obviously, the experience of living in the desert is not the same as living in a tropical region with the same average daily temperature around 70 F, where the temperature within the day is between 60 F and 80F. What matters here is not just the central location of the temperature, but the spread of the temperature. There are two common metrics to quantify spread.

    1. **Range:** The range is the difference between the maximum value and the minimum value of the attribute. The range is simple to calculate and articulate but has shortcomings as it is severely impacted by the presence of outliers and fails to consider the distribution of all other data points in the attributes. In the example, the range for the temperature in the desert is 80F and the range for the tropics is 20F. The desert region experiences larger temperature swings as indicated by the range.

2. **Deviation:** The variance and standard deviation measures the spread, by considering all the values of the attribute. Deviation is simply measured as the difference between any given value (xi) and the mean of the sample (μ). The variance is the sum of the squared deviations of all data points divided by the number of data points.

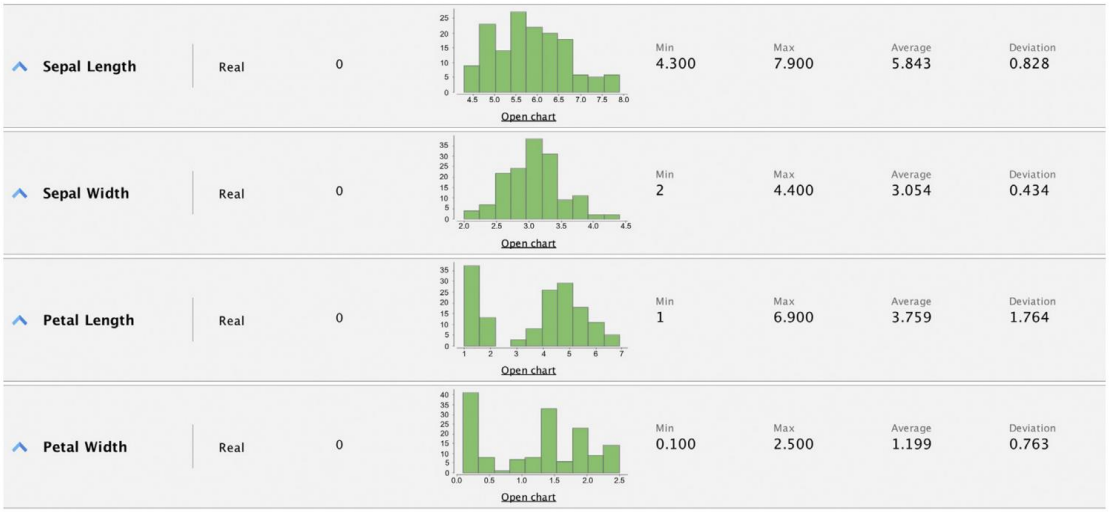


Figure 7: Univariate summary of the Iris dataset

Quartile: A box whisker plot is a simple visual way of showing the distribution of a continuous variable with information such as quartiles, median, and outliers, overlaid by mean and standard deviation. The main attraction of box whisker or quartile charts is that distributions of multiple attributes can be compared side by side and the overlap between them can be deduced. The quartiles are denoted by Q1, Q2, and Q3 points, which indicate the data points with a 25% bin size. In a distribution, 25% of the data points will be below Q1, 50% will be below Q2, and 75% will be below Q3. The Q1 and Q3 points in a box whisker plot are denoted by the edges of the box as shown in **Figure 8**. The Q2 point, the median of the distribution, is indicated by a cross line within the box. The outliers are denoted by circles at the end of the whisker line. In some cases, the mean point is denoted by a solid dot overlay followed by standard deviation as a line overlay. **Figure 9** shows that the quartile charts for all four attributes of the Iris dataset are plotted side by side. Petal length can be observed as having the broadest range and the sepal width has a narrow range, out of all of the four attributes. **Figure 10** shows boxplots for the "Productivity" and "Experience" columns.

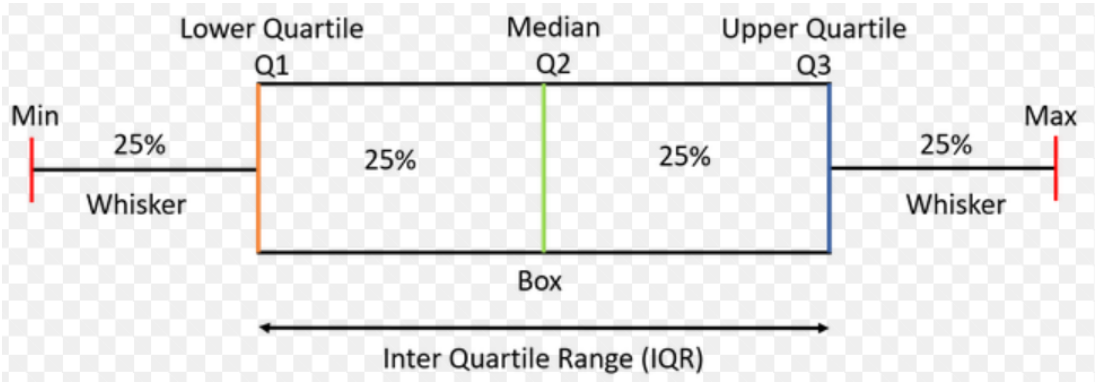


Figure 8: Quartile plot of Iris dataset.

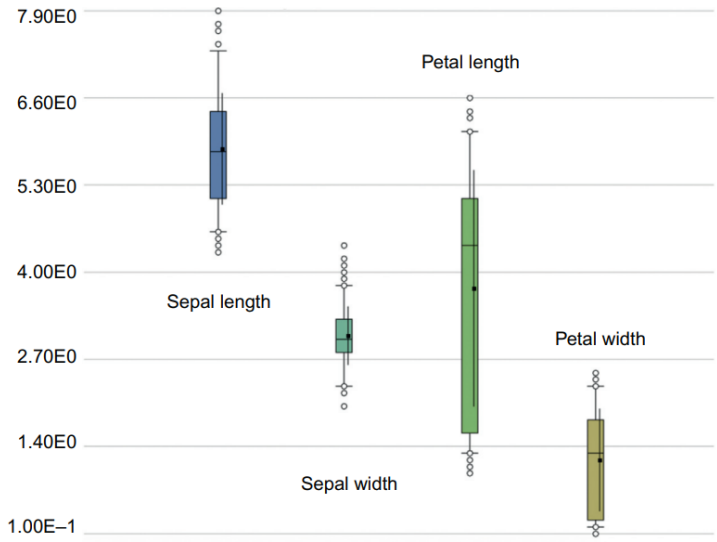


Figure 9: Quartile plot of Iris dataset.

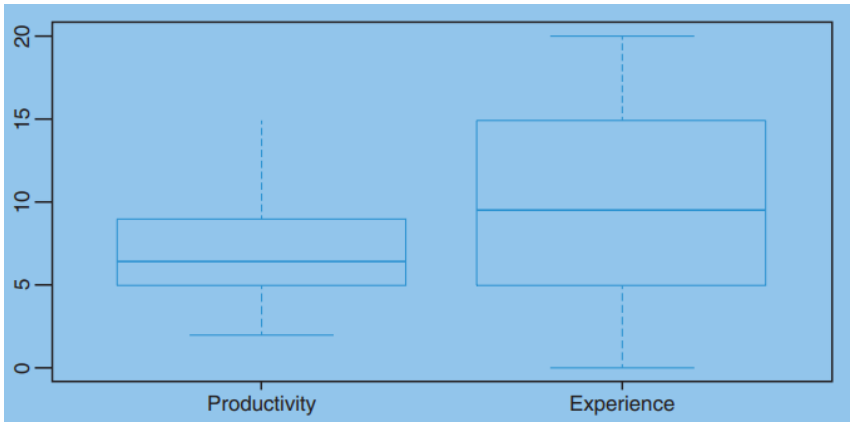


Figure 10: Boxplot for the "Productivity" and "Experience" columns of the Productivity dataset

- For continuous numeric attributes like petal length, instead of visualizing the actual data in the sample, its normal distribution function can be visualized instead. The normal distribution function of a continuous random variable is given by the formula:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{(x-\mu)^2/2\sigma^2}$$

where μ is the mean of the distribution and σ is the standard deviation of the distribution. Here an inherent assumption is being made that the measurements of petal length (or any continuous variable) follow the normal distribution, and hence, its distribution can be visualized instead of the actual values. The normal distribution is also called the Gaussian distribution or "bell curve" due to its bell shape. The normal distribution function shows the probability of occurrence of a data point within a range of values. If a dataset exhibits normal distribution, then 68.2% of data points will fall within one standard deviation from the mean; 95.4% of the points will fall within 2σ and 99.7% within 3σ of the mean. When the normal distribution curves are stratified by class type, more insight into the data can be gained. **Figure 11** shows the normal distribution curves for petal length measurement for each Iris species type. From the distribution chart, it can be inferred that the petal length for the I. setosa sample is more distinct and cohesive than I. versicolor and I. virginica.

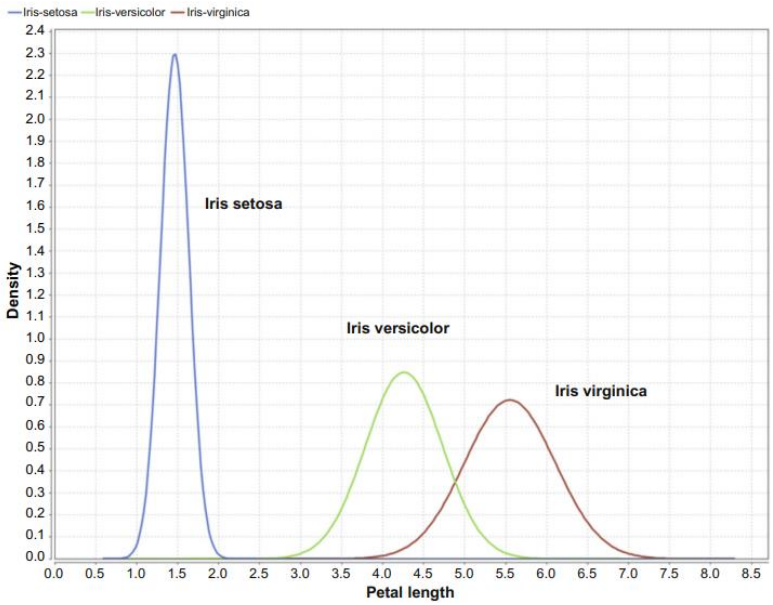


Figure 11: Distribution of petal length in Iris dataset

- **Multivariate Visualization**: The multivariate visual exploration considers more than one attribute in the same visual. The techniques discussed in this section focus on the relationship of one attribute with another attribute. These visualizations examine two to four attributes simultaneously.

- A **scatterplot** is one of the most powerful yet simple visual plots available. In a scatterplot, the data points are marked in Cartesian space with attributes of the dataset aligned with the coordinates. The attributes are usually of continuous data type. One of the key observations that can be concluded from a scatterplot is the existence of a relationship between two attributes under inquiry. If the attributes are linearly correlated, then the data points align closer to an imaginary straight line; if they are not correlated, the data points are scattered. Apart from basic correlation, scatterplots can also indicate the existence of patterns or groups of clusters in the data and identify outliers in the data. This is particularly useful for low-dimensional datasets. **Figure 12** shows the scatterplot between petal length (x-axis) and petal width (y-axis). These two attributes are slightly correlated, because this is a measurement of the same part of the flower.

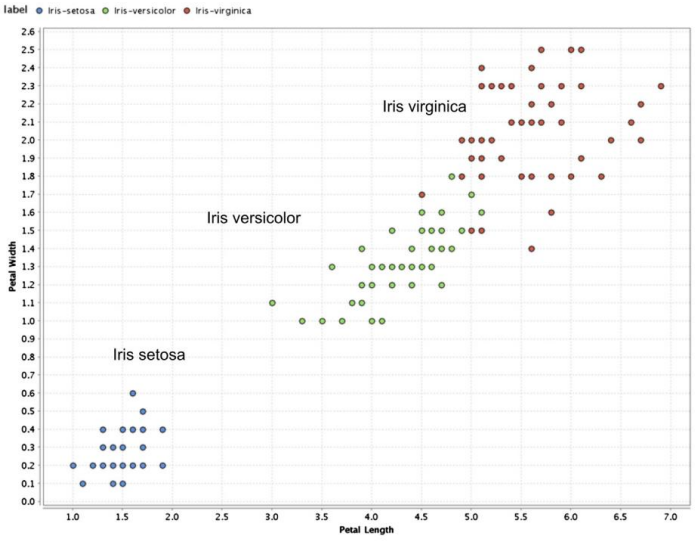


Figure 12: Scatterplot of Iris dataset.

When the data markers are colored to indicate different species using class labels, more patterns can be observed. There is a cluster of data points, all belonging to species I. setosa, on the lower left side of the plot. I. setosa has much smaller petals. This feature can be used as a rule to predict the species of unlabeled observations. One of the limitations of scatterplots is that only two attributes can be used at a time, with an additional attribute possibly shown in the color of the data marker. However, the colors are usually reserved for class labels.

- **Scatter Matrix** If the dataset has more than two attributes, it is important to look at combinations of all the attributes through a scatterplot. A scatter matrix solves this need by comparing all combinations of attributes with individual scatterplots and arranging these plots in a matrix.

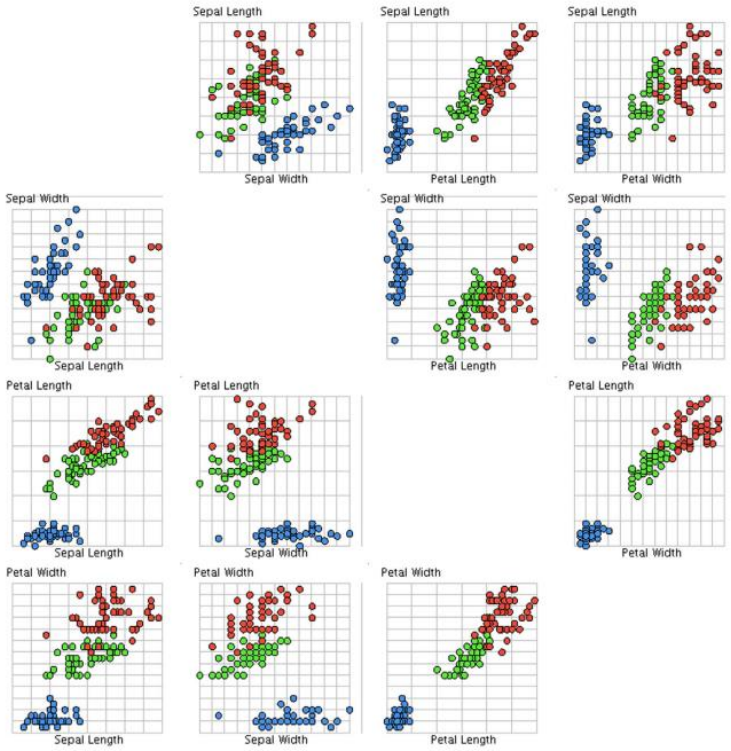


Figure 13: Scatter matrix plot of Iris dataset

- A scatter matrix for all four attributes in the Iris dataset is shown in **Figure 13**. The color of the data point is used to indicate the species of the flower. Since there are four attributes, there are four rows and four columns, for a total of 16 scatter charts.

- A **bubble chart** is a variation of a simple scatterplot with the addition of one more attribute, which is used to determine the size of the data point. In the Iris dataset, petal length and petal width are used for x and y-axis, respectively and sepal width is used for the size of the data point. The color of the data point represents a species class label (**Figure 14**).

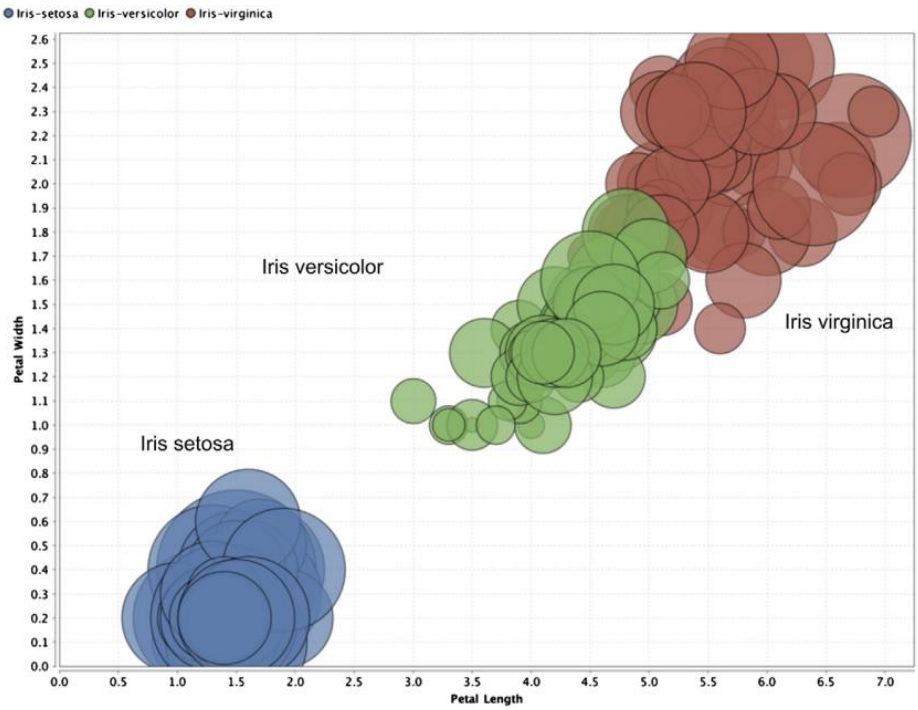
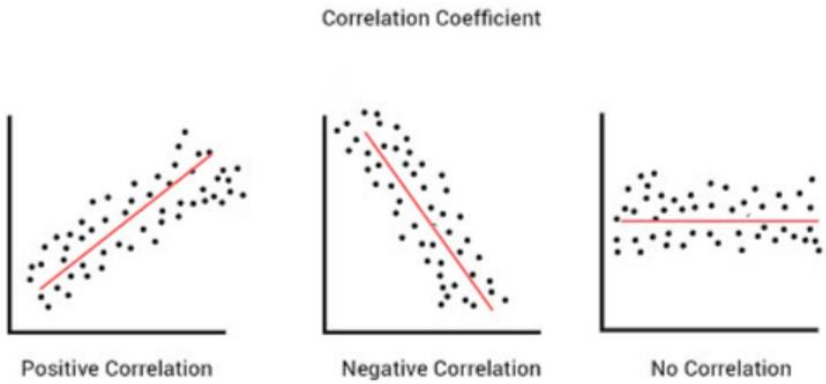


Figure 14: Bubble chart of Iris dataset.

- **Correlation** is a statistical analysis that is used to measure and describe the strength and direction of the relationship between two variables. Strength indicates how closely two variables are related to each other, and direction indicates how one variable would change its value as the value of the other variable changes. Correlation is a simple statistical measure that examines how two variables change together over time.

  Take, for example, "umbrella" and "rain." If someone who grew up in a place where it never rained saw rain for the first time, this person would observe that, whenever it rains, people use umbrellas. They may also notice that, on dry days, folks do not carry umbrellas. By definition, "rain" and "umbrella" are said to be correlated! More specifically, this relationship is strong and positive.



An important statistic, the Pearson's r correlation, is widely used to measure the degree of the relationship between linear related variables. When examining the stock market, for example, the Pearson's r correlation can measure the degree to which two commodities are related.

The following formula is used to calculate the Pearson's r correlation:

$$r = \frac{N\sum xy - \sum x \sum y}{\sqrt{\left[N\sum x^2 - \left(\sum x\right)^2\right]\left[N\sum y^2 - \left(\sum y\right)^2\right]}},$$

where

r = Pearson's r correlation coefficient,
N = number of values in each dataset,
$\sum xy$ = sum of the products of paired scores,
$\sum x$ = sum of x scores,
$\sum y$ = sum of y scores,

$\sum x^2$ = sum of squared x scores, and
$\sum y^2$ = sum of squared y scores.[6]

Hands-On Example: Correlation

Let us use the formula in Equation and calculate Pearson's r correlation coefficient for the height–weight pair with the data provided in Table 4

First we will calculate various quantities needed for solving Pearson's r correlation formula:

N = number of values in each dataset = 10
$\sum xy$ = sum of the products of paired scores = 98,335.30
$\sum x$ = sum of x scores = 670.70
$\sum y$ = sum of y scores = 1463
$\sum x^2$ = sum of squared x scores = 45,058.21
$\sum y^2$ = sum of squared y scores = 218,015

Plugging these into the Pearson's r correlation formula gives us 0.39 (approximated to two decimal places) as the correlation coefficient. This indicates two things: (1) "height" and "weight" are positively related, which means that, as one goes up, so does the other; and (2) the strength of their relation is medium.

Table 4 Height–weight data.

Height	Weight
64.5	118
73.3	143
68.8	172
65	147
69	146
64.5	138
66	175
66.3	134
68.8	172
64.5	118

- **Roadmap for data exploration** if there is a new dataset that has not been investigated before, having a structured way to explore and analyze the data will be helpful. Here is a roadmap to inquire a new dataset. Not all steps may be relevant for every dataset and the order may need to be adjusted for some sets, so this roadmap is intended as a guideline.

1. **Organize the dataset:** Structure the dataset with standard rows and columns. Organizing the dataset to have objects or instances in rows and dimensions or attributes in columns will be helpful for many data analysis tools. Identify the target or "class label" attribute, if applicable.

2. **Find the central point** for each attribute: **Calculate mean**, **median**, and **mode** for each attribute and the class label. If all three values are very different, it may indicate the presence of an outlier, or a multimodal or non-normal distribution for an attribute.

3. **Understand the spread of each attribute**: Calculate the standard deviation and range for an attribute. Compare the standard deviation with the mean to understand the spread of the data, along with the max and min data points.

4. **Visualize the distribution of each attribute**: Develop the histogram and distribution plots for each attribute. Repeat the same for class-stratified histograms and distribution plots, where the plots are either repeated or color-coded for each class.

5. **Pivot the data:** Sometimes called dimensional slicing, a pivot is helpful to comprehend different values of the attributes. This technique can stratify by class and drill down to the details of any of the attributes. Microsoft Excel and Business Intelligence tools popularized this technique of data analysis for a wider audience.

6. **Watch out for outliers:** Use a scatterplot or quartiles to find outliers. The presence of outliers skews some measures like mean, variance, and range. Exclude outliers and rerun the analysis. Notice if the results change.

7. **Understand the relationship between attributes:** Measure the correlation between attributes and develop a correlation matrix. Notice what attributes are dependent on each other and investigate why they are dependent.

8. **Visualize the relationship between attributes:** Plot a quick scatter matrix to discover the relationship between multiple attributes at once. Zoom in on the attribute pairs with simple two-dimensional scatterplots stratified by class.

9. **Visualize high-dimensional datasets:** Create parallel charts and Andrews curves to observe the class differences exhibited by each attribute. Deviation charts provide a quick assessment of the spread of each class for each attribute

- ## Practical Examples:

```
# calculate a 5-number summary
from numpy import percentile
from numpy.random import seed
from numpy.random import rand
# seed random number generator
seed(1)
# generate data sample
data = rand(1000)
# calculate quartiles
quartiles = percentile(data, [25, 50, 75])
# calculate min/max
data_min, data_max = data.min(), data.max()
# display 5-number summary
print('Min: %.3f' % data_min)
print('Q1: %.3f' % quartiles[0])
print('Median: %.3f' % quartiles[1])
print('Q3: %.3f' % quartiles[2])
print('Max: %.3f' % data_max)
```

```
Min: 0.000
Q1: 0.252
Median: 0.508
Q3: 0.751
Max: 0.997
```

Example **1**: calculating a 5 number summary of a data sample.

```
# example of a bar chart
from random import seed
from random import randint
from matplotlib import pyplot
# seed the random number generator
seed(1)
# names for categories
x = ['red', 'green', 'blue']
# quantities for each category
y = [randint(0, 100), randint(0, 100), randint(0, 100)]
# create bar chart
pyplot.bar(x, y)
# show line plot
pyplot.show()
```

Example **2**: Creating a bar chart from data.

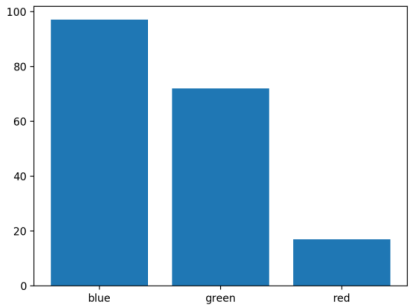


Figure 15: Output of Example 2

```
# example of a histogram plot
from numpy.random import seed
from numpy.random import randn
from matplotlib import pyplot
# seed the random number generator
seed(1)
# random numbers drawn from a Gaussian distribution
x = randn(1000)
# create histogram plot
pyplot.hist(x)
# show line plot
pyplot.show()
```

Example 3: Creating a histogram plot from data

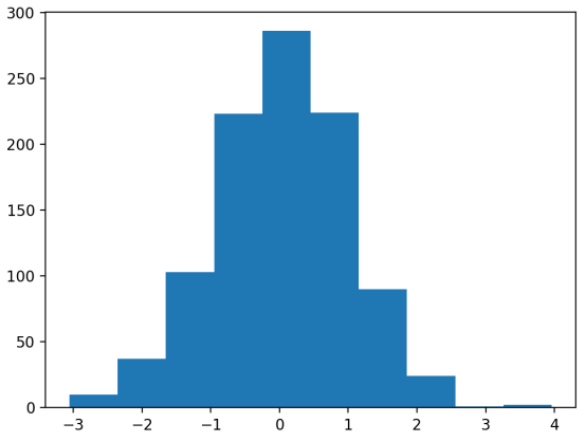


Figure 16: Output of Example 3

```
# example of a box and whisker plot
from numpy.random import seed
from numpy.random import randn
from matplotlib import pyplot
# seed the random number generator
seed(1)
# random numbers drawn from a Gaussian distribution
x = [randn(1000), 5 * randn(1000), 10 * randn(1000)]
# create box and whisker plot
pyplot.boxplot(x)
# show line plot
pyplot.show()
```

Example 4: Creating a box and whisker plot from data.

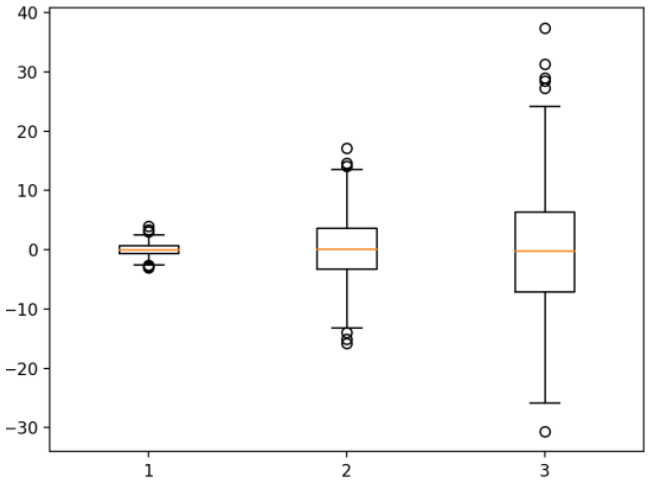


Figure 17: Output of Example 4

```
# example of a scatter plot
from numpy.random import seed
from numpy.random import randn
from matplotlib import pyplot
# seed the random number generator
seed(1)
# first variable
x = 20 * randn(1000) + 100
# second variable
y = x + (10 * randn(1000) + 50)
# create scatter plot
pyplot.scatter(x, y)
# show line plot
pyplot.show()
```

Example 5: Creating a scatter plot from data.

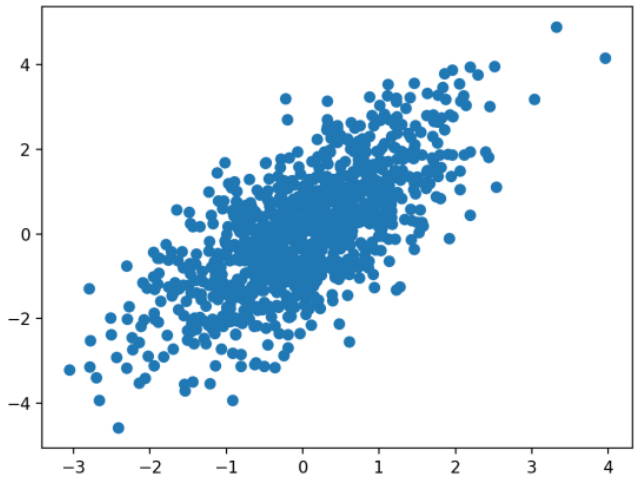


Figure 18: Output of Example 5

```
# View first 20 rows
from pandas import read_csv
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
peek = data.head(20)
print(peek)
```

```
    preg  plas  pres  skin  test  mass   pedi  age  class
0      6   148    72    35     0  33.6  0.627   50      1
1      1    85    66    29     0  26.6  0.351   31      0
2      8   183    64     0     0  23.3  0.672   32      1
3      1    89    66    23    94  28.1  0.167   21      0
4      0   137    40    35   168  43.1  2.288   33      1
5      5   116    74     0     0  25.6  0.201   30      0
6      3    78    50    32    88  31.0  0.248   26      1
7     10   115     0     0     0  35.3  0.134   29      0
8      2   197    70    45   543  30.5  0.158   53      1
9      8   125    96     0     0   0.0  0.232   54      1
10     4   110    92     0     0  37.6  0.191   30      0
11    10   168    74     0     0  38.0  0.537   34      1
12    10   139    80     0     0  27.1  1.441   57      0
13     1   189    60    23   846  30.1  0.398   59      1
14     5   166    72    19   175  25.8  0.587   51      1
15     7   100     0     0     0  30.0  0.484   32      1
16     0   118    84    47   230  45.8  0.551   31      1
17     7   107    74     0     0  29.6  0.254   31      1
18     1   103    30    38    83  43.3  0.183   33      0
19     1   115    70    30    96  34.6  0.529   32      1
```

Example 6: Reviewing the first few rows of data.

```
# Dimensions of your data
from pandas import read_csv
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
shape = data.shape
print(shape)
```

```
(768, 9)
```

Example 7: Reviewing the shape of the data.

```
# Data Types for Each Attribute
from pandas import read_csv
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
types = data.dtypes
print(types)
```

```
preg      int64
plas      int64
pres      int64
skin      int64
test      int64
mass    float64
pedi    float64
age       int64
class     int64
dtype: object
```

Example 8: Reviewing the data types of the data

```
# Statistical Summary
from pandas import read_csv
from pandas import set_option
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
set_option('display.width', 100)
set_option('precision', 3)
description = data.describe()
print(description)
```

```
          preg    plas    pres    skin    test    mass    pedi     age   class
count  768.000 768.000 768.000 768.000 768.000 768.000 768.000 768.000 768.000
mean     3.845 120.895  69.105  20.536  79.799  31.993   0.472  33.241   0.349
std      3.370  31.973  19.356  15.952 115.244   7.884   0.331  11.760   0.477
min      0.000   0.000   0.000   0.000   0.000   0.000   0.078  21.000   0.000
25%      1.000  99.000  62.000   0.000   0.000  27.300   0.244  24.000   0.000
50%      3.000 117.000  72.000  23.000  30.500  32.000   0.372  29.000   0.000
75%      6.000 140.250  80.000  32.000 127.250  36.600   0.626  41.000   1.000
max     17.000 199.000 122.000  99.000 846.000  67.100   2.420  81.000   1.000
```

Example 9: Reviewing a statistical summary of the data.

```
# Class Distribution
from pandas import read_csv
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
class_counts = data.groupby('class').size()
print(class_counts)
```

```
class
0    500
1    268
```

Example 10: Reviewing a class breakdown of the data.

```
# Pairwise Pearson correlations
from pandas import read_csv
from pandas import set_option
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
set_option('display.width', 100)
set_option('precision', 3)
correlations = data.corr(method='pearson')
print(correlations)
```

```
       preg   plas   pres   skin   test   mass   pedi    age  class
preg  1.000  0.129  0.141 -0.082 -0.074  0.018 -0.034  0.544  0.222
plas  0.129  1.000  0.153  0.057  0.331  0.221  0.137  0.264  0.467
pres  0.141  0.153  1.000  0.207  0.089  0.282  0.041  0.240  0.065
skin -0.082  0.057  0.207  1.000  0.437  0.393  0.184 -0.114  0.075
test -0.074  0.331  0.089  0.437  1.000  0.198  0.185 -0.042  0.131
mass  0.018  0.221  0.282  0.393  0.198  1.000  0.141  0.036  0.293
pedi -0.034  0.137  0.041  0.184  0.185  0.141  1.000  0.034  0.174
age   0.544  0.264  0.240 -0.114 -0.042  0.036  0.034  1.000  0.238
class 0.222  0.467  0.065  0.075  0.131  0.293  0.174  0.238  1.000
```

Example 11: Reviewing correlations of attributes in the data.

```
# Skew for each attribute
from pandas import read_csv
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
skew = data.skew()
print(skew)
```

```
preg     0.901674
plas     0.173754
pres    -1.843608
skin     0.109372
test     2.272251
mass    -0.428982
pedi     1.919911
age      1.129597
class    0.635017
```

Example 12: Reviewing skew of attribute distributions in the data

```
# Univariate Histograms
from matplotlib import pyplot
from pandas import read_csv
filename = 'pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
data.hist()
pyplot.show()
```

Example 13: Example of creating histogram plots

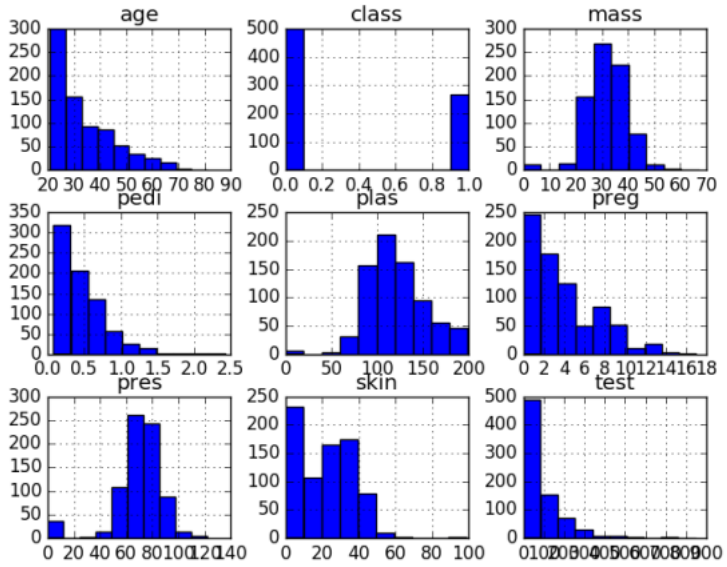


Figure 19: Output of Example 13

```
# Univariate Density Plots
from matplotlib import pyplot
from pandas import read_csv
filename = 'pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
data.plot(kind='density', subplots=True, layout=(3,3), sharex=False)
pyplot.show()
```

Example 14: Creating density plots.

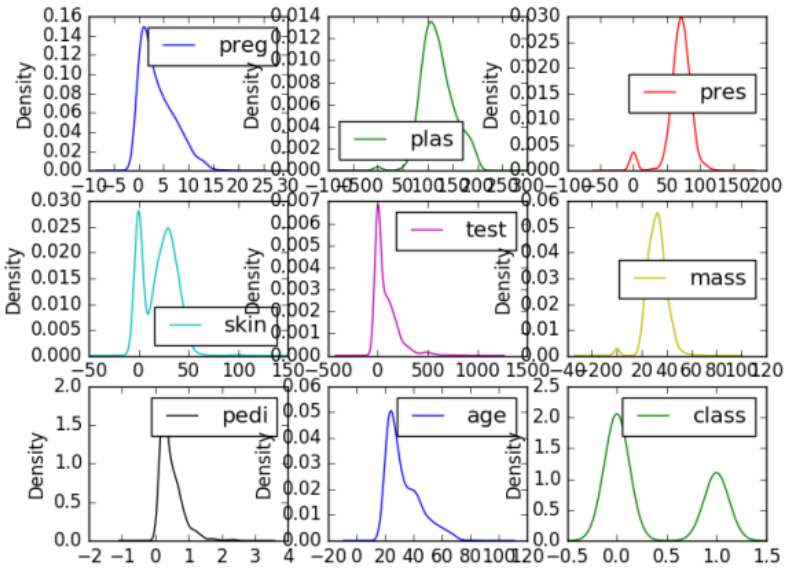


Figure 20: Output of Example 14

```
# Box and Whisker Plots
from matplotlib import pyplot
from pandas import read_csv
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
data.plot(kind='box', subplots=True, layout=(3,3), sharex=False, sharey=False)
pyplot.show()
```

Example 15: Creating box and whisker plots.

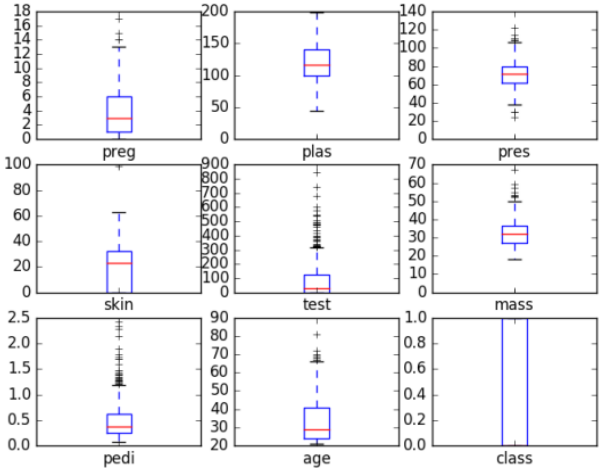


Figure 21: Output of Example 15

```
# Correlation Matrix Plot
from matplotlib import pyplot
from pandas import read_csv
import numpy
filename = 'pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
correlations = data.corr()
# plot correlation matrix
fig = pyplot.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(correlations, vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = numpy.arange(0,9,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(names)
ax.set_yticklabels(names)
pyplot.show()
```

Example 16: Creating a correlation matrix plot

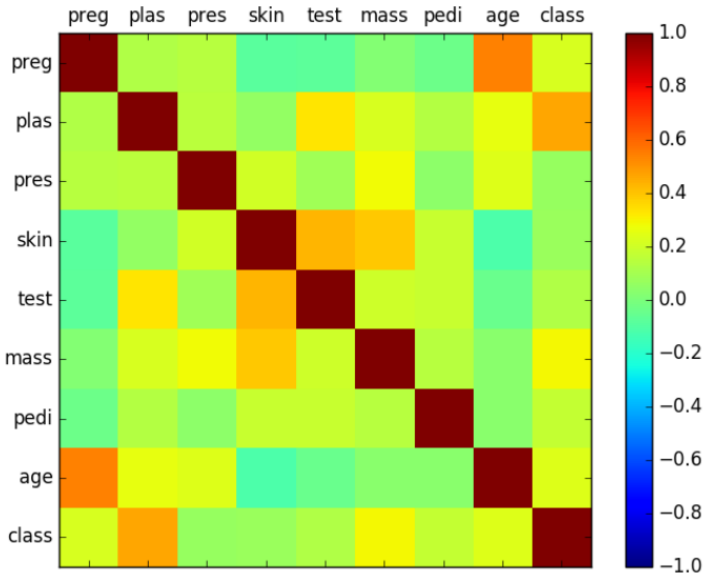


Figure 22: Output of Example 16

```
# Scatterplot Matrix
from matplotlib import pyplot
from pandas import read_csv
from pandas.plotting import scatter_matrix
filename = "pima-indians-diabetes.data.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(filename, names=names)
scatter_matrix(data)
pyplot.show()
```

Example 17: Creating a scatter plot matrix

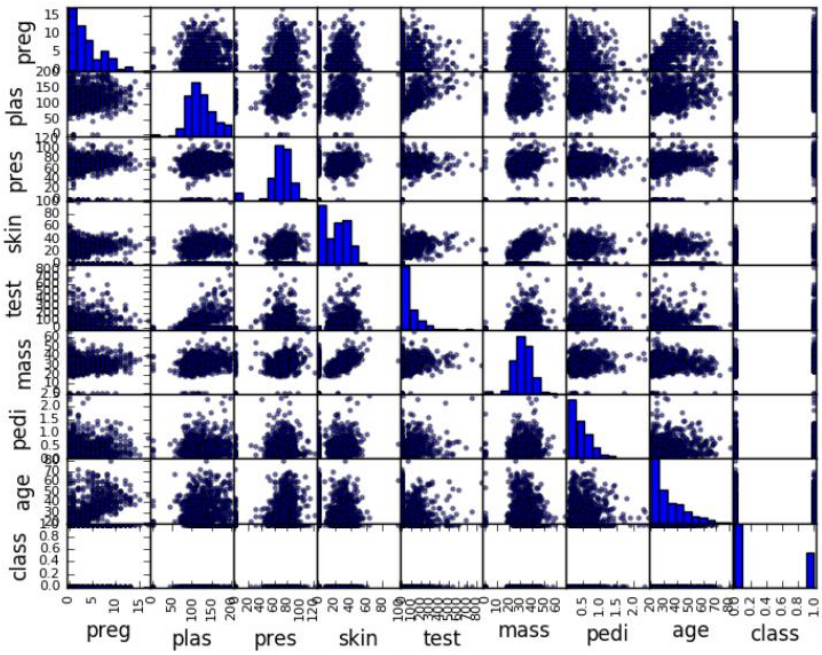


Figure 23: Output of Example 17