Review_Basic_statistics

March 4, 2025

1 Data and Variables

Data can be viewed as the raw material from which information is obtained.
Variables add meaning to the data.

1.1 Tabular Data

- Before we process or analyze data, we have to capture and represent it using **Variables**
- **Variable** is a label (field, header or title) that we give to our data.
- Consider the following data in tabular form:

Faculty Name	Area of Specialization	Experience in yrs	International Faculty	# Assignments
Ahmad	Data Science	23	Yes	Medium
Ameen	Operations Research	10	No	High
Aqib	Simulation	13	Yes	High
Bashir	Data Science	20	No	Medium
Nadhir	Operations Research	18	Yes	Low

- The column headings *Faculty Name, Area of Specialization, Experience in yrs, International Faculty* and *# Assignments* are the **labels or fields**.

Note: There are many formats in which data can be represented, in this course we will only consider tabular format (i.e. the data will have row corresponding to records, and column corresponding to fields).

1.2 Field Types

Some of the common field types that we often come in tabular data are: 1. **Numerical Data:** these are numerical values of the filed, where the value has a meaning. For example, in the previous table, *Experience in yrs* is a numerical data. A value of 10 in the second row implies the corresponding faculty has 10 years of experience. 2. **Categorical Data:** there are non-numerical field values, where the value represents a category without any order. For example, in the previous table, *International Faculty* is a categorical data. A value of 'Yes' in the first row implies the corresponding faculty a foreign faculty. This column has two categories. Similarly, in the previous table, the column *Area of Specialization* is a categorical data, with 3 different categories. 1. **Nominal Data:** sometimes categorical data is represented using numbers for convenience. Such representation is called as nominal data. For example, in *International Faculty* column, one could replace 'Yes' with 1 and 'No' with 0, or vice-versa. These numbers do not have meaningful

mathematical or statistical insights. They are just representation for convenience. 2. **Ordinal Data:** these are non-numerical field values, where the value represents a category with order. For example, in the previous table, *# Assignments* is a ordinal data. A value of 'High' in the second row implies the corresponding faculty assigns too many homeworks.

Note: Nominal data should not be considered as numeric data. Generally, categorical and nominal data are synonyms, but in this course we restrict the definition of nominal data to numerical representation for convenience.

2   Field Segregation

Not all the fields in a given data table are same. They can be classified or segregated as: 1. **Independent Variables:** A variable that is thought to be controlled or not affected by other variables is called an independent variable. Typically, it is also known as **attribute, control, explanatory, regressor, input, predictor, observed, feature, field** variable. 2. **Dependent Variables:** A variable that depends on other variable (and maybe of interest in the analysis). It is also known as **predicted, explained, target, response, output/outcome, label** variable. 3. **Auxiliary Variables:** These are the variables that provide meta information about the data, and may not be helpful (or may not be meant) to be used in the analysis.

2.0.1   Example: Variables/Fields

Question-A: Consider the following table:

Stu ID	Score	Subj	GPA	Honor	Grade
2508	75	Math	2.97	0	B+
2679	81	Phys	3.25	0	B+
2416	95	Math	3.55	1	A
2720	85	Chem	3.12	0	B+
2575	81	Phys	3.09	0	B
2118	82	Math	3.33	0	B+
2060	97	Phys	3.78	1	A+

1. Identify all the variable/fields.
2. Identify the variable/field types.

1. Identify all the variable/fields.
   From the above table, we can see the following fields: *Stu ID, Score, Subj, GPA, Honor, and Grade.*

2. Identify the variable/field types. The following table gives the required information:

Field	Type
Stu ID	Nominal
Score	Numeric
Subj	Categorical
GPA	Numeric
Honor	Nominal

Field	Type
Grade	Ordinal

3 Frequency Distribution

A graph showing how many times each value occur in a variable.

3.1 Histogram

1. Plot values of observation on one of the axes (typically x-axis). The values can be ordered (if applicable)
2. Plot perpendicular bars to the above axis.
3. The height of the bar shows how many times each value occurred in the dataset.

3.1.1 Histograms of Non-numerical Data

3.1.2 Histograms of Numerical Data

1. When you have numeric values, it does not make sense to count the occurrences of each value.
2. Thus, we introduce the concept of buckets or bins.
3. The idea is to plot the bars for each bin/bucket, where the height of the bar indicates the number of values that belongs to the bin/bucket.

The following video shows the procedure.

3.2 Pie-Charts

1. Pie-charts are extends the visualization offered by histograms.
2. Typically, it is useful for the categorical data.
3. Unlike histogram, Pie-charts display the proportion/percentage of occurrence of each category.
4. The idea is to represent the proportion as the sectors of a circle, where the length of the arc represents the proportion.
5. When we depict all the categories, then the pie chart will be a full circle. Otherwise, it could be an incomplete circle.

Let us build the pie chart for the table given for non-numerical data. The process to draw pie-chart is shown in the following video.

3.2.1 Example: Hands on Histogram and Pie-Chart

Question-B: Consider the following table:

Stu ID	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
Score	75	81	95	85	81	82	97	89	76	100	77	79	87	87	79	75	100	82	83	98	86	87	71	78	81
Subj	M	P	M	P	P	P	P	P	M	M	P	P	P	M	P	M	M	C	C	M	C	P	P	P	M

1. Draw the histogram for *Subj*.

2. Draw the histogram for *Score* , with bins as 71-80, 81-90, 91-100.
3. For the above histogram, increase the bins to 30.
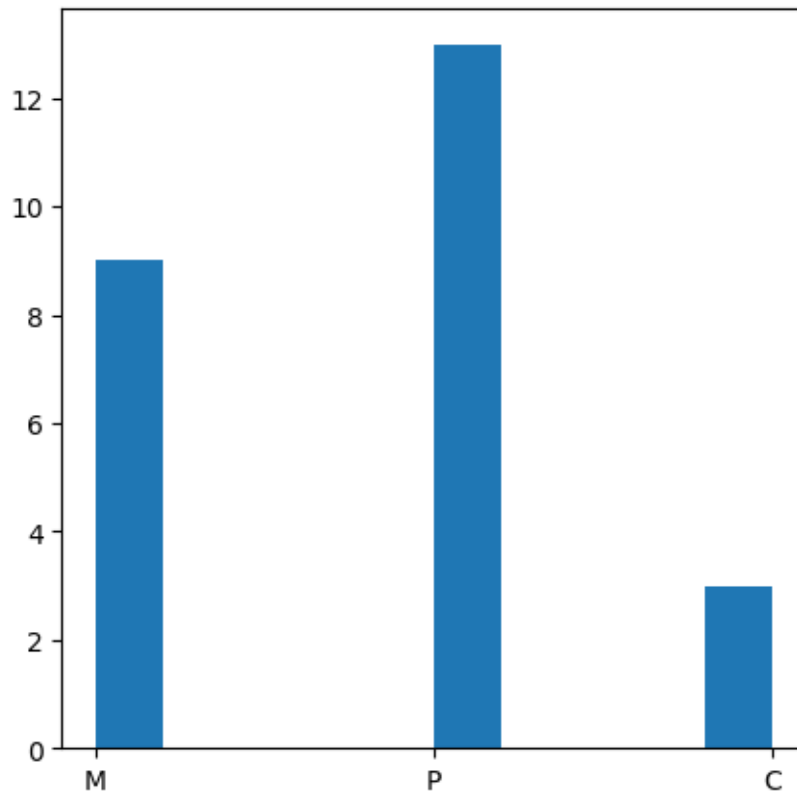4. Draw the pie-chart for *Subj*.

```python
[7]: %matplotlib inline
# %matplotlib notebook
# 1. Draw the histogram for *Subj*.

Subj = ['M', 'P', 'M', 'P', 'P', 'P', 'P', 'P', 'M', 'M', 'P', 'P',
        'P', 'M', 'P', 'M', 'M', 'C', 'C', 'M', 'C', 'P', 'P', 'P', 'M']

Score = [ 75,  81,  95,  85,  81,  82,  97,  89,  76, 100,  77,  79,  87,
          87,  79,  75, 100,  82,  83,  98, 86, 87, 71, 78, 81]

import matplotlib.pyplot as plt

plt.figure(figsize=(5,5))
plt.hist(Subj)
plt.show()
```

```
[8]:  %matplotlib inline
      # 2. Draw the histogram for *Score* , with bins as 71-80, 81-90, 91-100.

      import matplotlib.pyplot as plt
      plt.figure(figsize=(5,5))
      plt.hist(Score,bins=[71,81,91,100]) # Creates three bins as: 71-80, 81-90,␣
       ↪91-100
      plt.show()

      # Note: once you import a library in a Jupyter session,
      # you do not have to import it again in another cell or notebook.
      # Sometimes, we repeat the libraries in every cell for the sake of completeness.
```
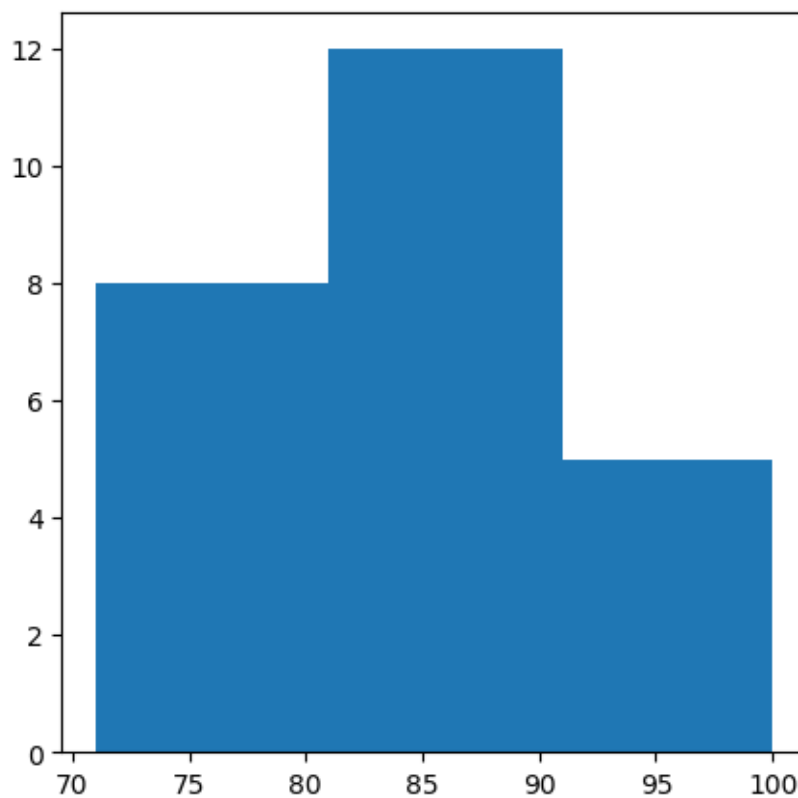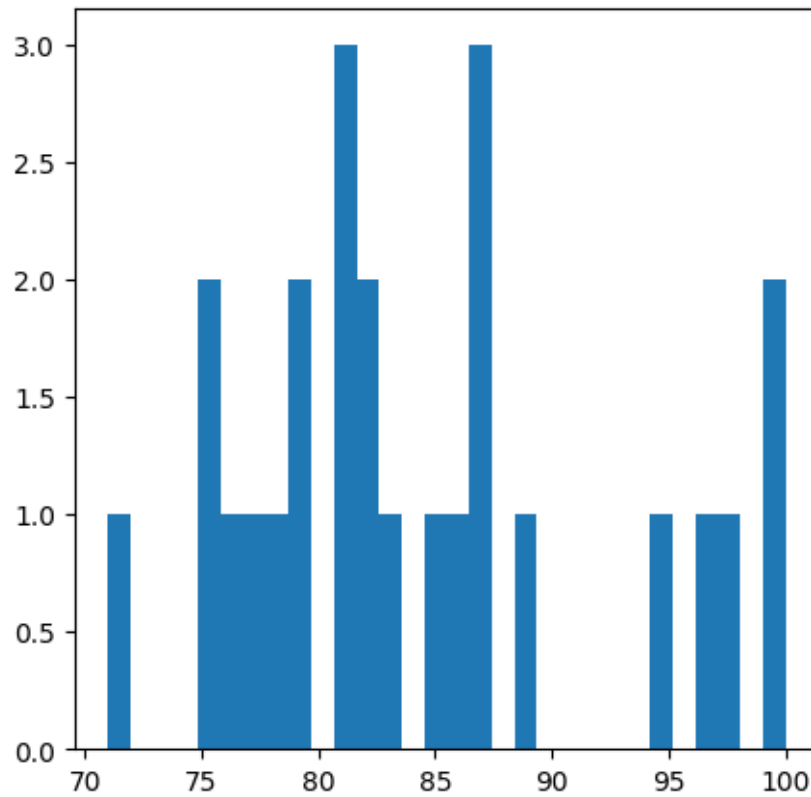
```
[9]:  # 3. For the above histogram, increase the bins to 30.

      import matplotlib.pyplot as plt
      plt.figure(figsize=(5,5))
      plt.hist(Score,bins=30)
      plt.show()
```

[10]:
```python
# 4. Draw the pie-chart for *Subj*.

import numpy as np
subj_arr = np.array(Subj)
subj_labels, subj_counts = np.unique(Subj,return_counts=True)

import matplotlib.pyplot as plt
plt.figure()
plt.pie(subj_counts, labels = subj_labels,autopct='%.2f%%') #autopct to format␣
 ↪labels
plt.show()

# # without using numpy to calculate subject labels
# cm,cp,cc=0,0,0
# for e in Subj:
#     if e=='M':
#         cm+=1
```

```
#      elif e=="P":
#         cp+=1
#      else:
#         cc+=1
# print(cm,cp,cc)

## To get counts
# print(Subj.count('M'))


# subj_counts = [(subj_arr==s).sum() for s in subj_labels]

# # Without List Comprehension
# sc=[]
# for e in subj_labels:
#     sc.append((subj_arr==e).sum())
# print(sc)

## To get unique values
# print(set(Subj))
```
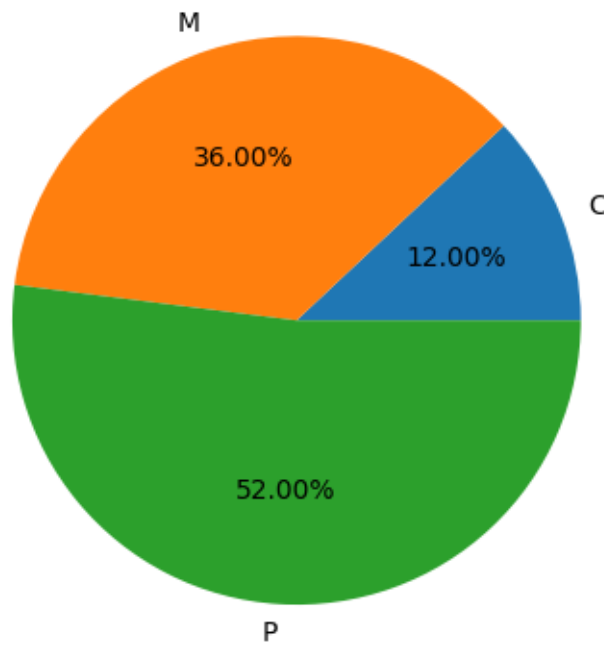
3.3 Normal Distribution

1. A histogram depicts the distribution of the variable.
2. A variable's distribution may not be easy to estimate.
3. However, it can be easy to check if it is closer to a particular distribution.
4. Among standard distributions, *Normal Distribution* is a prominent distribution.
5. Many statistical theories are based on the assumption of normal distribution.

So, let us see what normal distribution is, and how to know if a variable's distribution is closer to normal distribution.

Normal Distribution: 1. In an ideal world, data would be distributed symmetrically around the center of all the data. 2. If we draw a vertical line through the center of a distribution, both sides should look the same. 3. The so-called normal distribution is characterized by a bell-shaped curve.

Note: there actual process to measures Kurtosis and Skew will be skipped for now.

4 Measures of Centrality

How to estimate the center of distribution.

1. Often, one number can tell a lot about the distribution.
2. One of the basic measure is a number that points to the 'center' of the distribution.
3. However, the definition of 'center' is not straightforward. It depends upon the context.
4. In the following cells, we will look at the three common measures of 'center': **mean, median** and **mode**.

4.1 Mean

1. Mean is commonly referred as average, though they are not exactly synonyms.
2. Mean is often used to measure the central tendency of continuous data as well as discrete data.
3. If $x_1, x_2, ..., x_n$ are values, then the mean $\overline{x}$ is calculated as:

$$\overline{x} = \frac{x_1 + x_2 + ... + x_n}{n}$$

4. Mean is susceptible to the presence of outliers.
5. Mean is useful when the data distribution is normal distribution (or at least close to looking like a normal distribution).

4.2 Median

1. Median is the middle value of the data that has been sorted according to the values of the data.
2. When the data has even number of values, median is calculated as the average of the two middle values.
3. Typically, median is less susceptible to the presence of outliers (compared to mean).

4.3 Mode

1. Mode is the most frequently occurring value in a dataset.
2. Typically, mode is used for non-numerical data.
3. On a histogram, the category of the highest bar denotes the mode of the data.

4.3.1 Example: Calculate the mean, median and mode

Question-C: Consider the following table:

Stu ID	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
Score	75	81	95	85	81	82	97	89	76	100	77	79	87	87	79	75	100	82	83	98	86	87	71	78	81

1. Calculate the mean, median and mode of the data (by hand).
2. Calculate the above measures using Python.

1. Calculate the mean, median and mode of the data (by hand).

Mean:

$$\frac{75+81+95+85+81+82+97+89+76+100+77+79+87+87+79+75+100+82+83+98+86+87-}{25}$$

Median: (sorted in ascending order)

$$Score_{sorted} = [71, 75, 75, 76, 77, 78, 79, 79, 81, 81, 81, 82, \mathbf{82}, 83, 85, 86, 87, 87, 87, 89, 95, 97, 98, 100, 100]$$

Mode: (unique values and occurrence)

Value	71	75	76	77	78	79	81	82	83	85	86	87	89	95	97	98	100
Occurrence	2	1	1	1	2	3	2	1	1	1	3	1	1	1	1	2	

From the above table, we can see that the mode is not unique, and the values are 81 and 87.

```
[11]: %pip install scipy
```

```
Requirement already satisfied: scipy in
/home/motid/kfupm/assignment2-ICS/.venv/lib/python3.11/site-packages (1.15.2)
Requirement already satisfied: numpy<2.5,>=1.23.5 in
/home/motid/kfupm/assignment2-ICS/.venv/lib/python3.11/site-packages (from
scipy) (2.2.3)

[notice] A new release of pip is
available: 24.0 -> 25.0.1
[notice] To update, run:
pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

```
[12]: # 2. Calculate the above measures using Python.

      import numpy as np
      from scipy import stats

      Score = [75,  81,  95,  85,  81,  82,  97,  89,  76, 100,  77,  79,  87, 87, ␣
      ↪79,  75, 100,  82,  83,  98, 86, 87, 71, 78, 81]
      Score = np.array(Score)

      print(f'The mean is {Score.mean()}, the median is {np.median(Score)}, and the␣
      ↪mode is {stats.mode(Score)[0]}.')

      # stats.mode return single minimum value of the mode,
      # when multiple non-unique modes are present in the data.
```

The mean is 84.44, the median is 82.0, and the mode is 81.

```
[13]: # For python 3.8.
      # The following code displays all the modes.

      # from statistics import multimode
      # multimode(Score)
```

5   Dispersion of a Distribution

How to measure the spread of a distribution.

1. A measure of 'center' may not be enough in understanding the actual shape of a distribution.
2. The spread or dispersion of a distribution can be measured using the following typical measures: **Range, Interquartile Range, Variance** and **Standard Deviation**.

5.1   Range

1. It is the difference between the largest value and the smallest value.
2. It is susceptible to the outliers.

5.2   Interquartile Range

1. It is the range obtained after removing extreme values.
2. One convention is to cut-off the top and bottom one-quarter of the data and calculate the range of the remaining middle 50% of the scores.
3. The bottom one-quarter values of data are also known as *lower or first quartile* (25th percentile). It is denoted as Q_1.
4. The top one-quarter values of data are also known as *upper or third quartile* (75th percentile). It is denoted as Q_3.
5. The Interquartile Range(IQR) is now calculated as:

$$IQR = Q_3 - Q_1$$

5.3 Variance

1. It is a measure used to indicate how spread out the data points are.
2. If the individual observations vary greatly from the group mean, then the variance is big; and vice versa.
3. The variance of the population is defined by the following formula:

$$\sigma^2 = \frac{\sum (X_i - \overline{X})^2}{N}$$

   where σ^2 is the population variance, $\overline{X}$ is the population mean, X_i is the ith element from the population, and N is the number of elements in the population.
4. The variance of the sample is defined by the following formula:

$$s^2 = \frac{\sum (x_i - \overline{x})^2}{n - 1}$$

   where s^2 is the sample variance, $\overline{x}$ is the sample mean, x_i is the ith element from the sample, and n is the number of elements in the sample.
5. The above formula, the variance of the sample is an unbiased estimate of the variance of the population.

5.4 Standard Deviation

1. It is the square root of the variance.
2. It is computed as:

$$s = \sqrt{\frac{\sum (x_i - \overline{x})^2}{n - 1}}$$

where s is the sample standard deviation, $\overline{x}$ is the sample mean, x_i is the ith element from the sample, and n is the number of elements in the sample. 3. The advantage: the units of standard deviation is same as the units of the data. 4. The above is not true for variance.

5.4.1 Example: Measures of Dispersion

Question-D: Consider the following table:

Stu ID	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
Score	75	81	95	85	81	82	97	89	76	100	77	79	87	87	79	75	100	82	83	98	86	87	71	78	81

1. Calculate the Range, Variance, and Standard deviation of the data (by hand).
2. Calculate the above measures using Python.
3. Calculate the interquartile range.

1. Calculate the Range, Sample Variance, and corresponding Standard deviation of the data (by hand).

Range: The max value in the data is 100. The min value in the data is 71. Thus, the range is 100-71 $= 29$.

Variance: The following table illustrates the calculations:

	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
Score	75	81	95	85	81	82	97	89	76	100	77	79	87	87	79	75	100	82	83	98	86	87	71	78	81
$(x-\bar{x})$	- 9.44	10.56 3.44	0.56	- 2.44	12.56	4.56 8.44	15.56	6 7.44	- 5.44	2.56	2.56 5.44	- 9.44	15.56	- 2.44	1.44	-	13.56	6.56 13.44	2.56 4.44	- 3.44	-	-			
$(x-\bar{x})^2$	89.11	111.63	336.63	11.83	111.63	33.95	357.76	70.37	93.02	342.55	13.39	59.60	356.65	356.65	20.59	93.02	342.55	1.19	3.67	363.87	3.65	356.65	160.63	347.36	8336

The sum of the elements of the last row divided by 24 gives sample variance, which is equal to 67.34.

Standard deviation: It is the square root of variance, and for the given data it is equal to 8.21.

```
[14]:  import numpy as np

       Score = [ 75,  81,  95,  85,  81,  82,  97,  89,  76, 100,  77,  79,  87, 87, ↵
        →79,  75, 100,  82,  83,  98, 86, 87, 71, 78, 81]
       Score = np.array(Score)


       print(f"""The Range is {Score.max()-Score.min()},
       the sample variance is {np.var(Score,ddof=1)}, and
       the sample standard deviation is {np.std(Score,ddof=1)}.""")
```

```
The Range is 29,
the sample variance is 67.33999999999999, and
the sample standard deviation is 8.206095295571457.
```

```
[15]:  Q3, Q1 = np.percentile(Score, [75 ,25])

       ## Or indivudial
       # Q3 = np.percentile(Score, 75)
       # Q1 = np.percentile(Score, 25)

       IQR = Q3 - Q1
       print(f'The interquartile range is {IQR}')
       # print(Q3,Q1)
```

```
The interquartile range is 8.0
```

5.5   Box Plots

To build a boxplot, following things are required from the data.

1. The min and max values in the data.

2. The first and third quartile of the data.
3. The median of the data.

Now, using the above values, box plot is drawn as follows:

Example of box plot:

5.6   Comparing Box Plots

1. Compare position and length of the boxes.
2. Compare the position of medians.
3. Compare position and length of the whiskers.

5.6.1   Example: Box Plots

**Question-E:** Consider the following table:

Stu ID	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
Score	75	81	95	85	81	82	97	89	76	100	77	79	87	87	79	75	100	82	83	98	86	87	71	78	81

Draw the box plot of score using Python inbuilt function.

```
[16]:  # Draw the box plot of score.

       Score = [ 75,  81,  95,  85,  81,  82,  97,  89,  76, 100,  77,  79,  87, 87, ⌴
         ↪79,  75, 100,  82,  83,  98, 86, 87, 71, 78, 81]
       Score = np.array(Score)

       import matplotlib.pyplot as plt
       plt.figure(figsize=(3,8))
       plt.boxplot(Score)
       plt.title('Box Plot')
       plt.show()
```

Box Plot
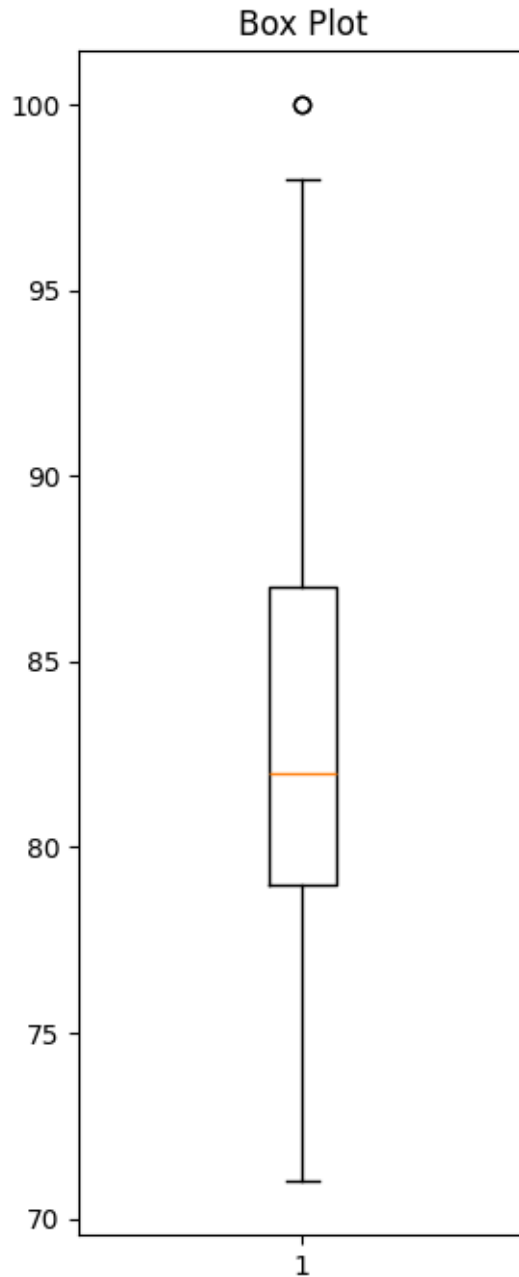
6   Comparing Distributions

Are the two distributions similar or dissimilar?

1. Often there is a need to compare different distributions to derive some important insights or make decisions.
2. Visual inspection via comparing histograms (or any graphical depiction of distribution) is not practical (and not easy).

3. Visual inspection via comparing box plots.
4. **Hypothesis testing** and **Confidence or probability value** are used for comparing distributions.

6.1 Hypothesis Testing

1. A hypothesis is a way to state our assumption or belief that could be tested.
2. The default knowledge or assumption could be stated as a *null hypothesis.*
3. The opposite of the default knowledge is stated as *alternative hypothesis.*
4. Example:

- Null hypothesis: no difference between the two distributions.
- Alternative hypothesis: there is a difference between the two distributions.

6.2 *P*-Value

1. p-value is the confidence or probability value of the null hypothesis.
2. It indicates how much we believe the two distributions are the same.
3. If p-value is very small, then we can reject the null hypothesis, and accept the alternative hypothesis.
4. If p-value is not very small, then we fail to reject the null hypothesis.
5. Typically, a p-value less than 0.05 or 5% is considered as very small.

6.3 Typical Tests

1. Shapiro-Wilk Test: Test if a data sample follows normal distribution.

- Null hypothesis: the sample has a normal distribution.

2. Student's t-test: Tests if the means of two independent samples are significantly different. Assume data follow normal distributions.

- Null hypothesis: the means of the samples are equal.

3. Mann-Whitney U Test: Tests if the means of two independent samples are significantly different.

- Null hypothesis: the means of the distributions are equal.

6.3.1 Hands on Hypothesis Testing

Question-F: Consider the following data:

Stu ID	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
Score_1	75	81	85	85	81	82	87	89	73	100	72	79	87	87	79	75	100	82	83	88	86	87	66	78	81
Score_2	70	99	70	100	78	66	76	79	65	77	100	85	71	99	66	84	87	66	66	74	81	83	85	75	66

1. Check if Score_1 and Score_2 follows normal distribution.
2. Draw the box plots of Score_1 and Score_2, and comment on their similarity/dissimilarity.
3. Check if the means of Score_1 and Score_2 are equal using Student's t-test.

4. Check if the means of Score_1 and Score_2 are equal using Mann-Whitney U Test.

```
[17]: import numpy as np

      Score_1 = [75, 81, 85, 85, 81, 82, 87, 89, 73, 100, 72, 79, 87, 87, 79, 75,␣
        ↪100, 82, 83, 88, 86, 87, 66, 78, 81]
      Score_2 = [70, 99, 70, 100, 78, 66, 76, 79, 65, 77, 100, 85, 71, 99, 66, 84,␣
        ↪87, 66, 66, 74, 81, 83, 85, 75, 66]
      # They can be of different lengths
```

```
[18]: # 1. Check if Score_1 and Score_2 follows normal distribution.
      from scipy.stats import shapiro
      xc=shapiro(Score_1)
      print(xc)

      _, pval = shapiro(Score_1)
      print('p-value',pval)

      if pval <0.05:
        print("We reject null hypothesis for Score_1")
      else:
        print("We fail to reject null hypothesis for Score_1")

      print("\n"*3)

      _, pval = shapiro(Score_2)
      print('p-value',pval)

      if pval <0.05:
        print("We reject null hypothesis for Score_2")
      else:
        print("We fail to reject null hypothesis for Score_2")
```

```
ShapiroResult(statistic=np.float64(0.952900951701251),
pvalue=np.float64(0.29115176848877733))
p-value 0.29115176848877733
We fail to reject null hypothesis for Score_1




p-value 0.012744178883043427
We reject null hypothesis for Score_2
```

```
[19]: # 2. Draw the box plots of Score_1 and Score_2, and comment on their similarity/
        ↪dissimilarity.
      import matplotlib.pyplot as plt
      fig =plt.figure(figsize=(6,8))
```

```
ax = fig.add_subplot(111)   #fig.add_subplot(ROW,COLUMN,POSITION)
plt.boxplot([Score_1,Score_2])
plt.title('Box Plot')
ax.set_xticklabels(['Score_1', 'Score_2'])
plt.show()



# # Looking at the box plots, we see the following:
# 1. The median for Score_1 is more than Score_2.
# 2. The IQRs are different.
# 3. For Score_1, the median is closer to Q1.
# 4. For Score_2, the median is almost at the center of Q1 and Q3.
# 5. They upper whisker lengths are different. Similarly, the lower whisker
  ↪lengths are different.

# There is a possibility of having differences in the distributions.
```

Box Plot
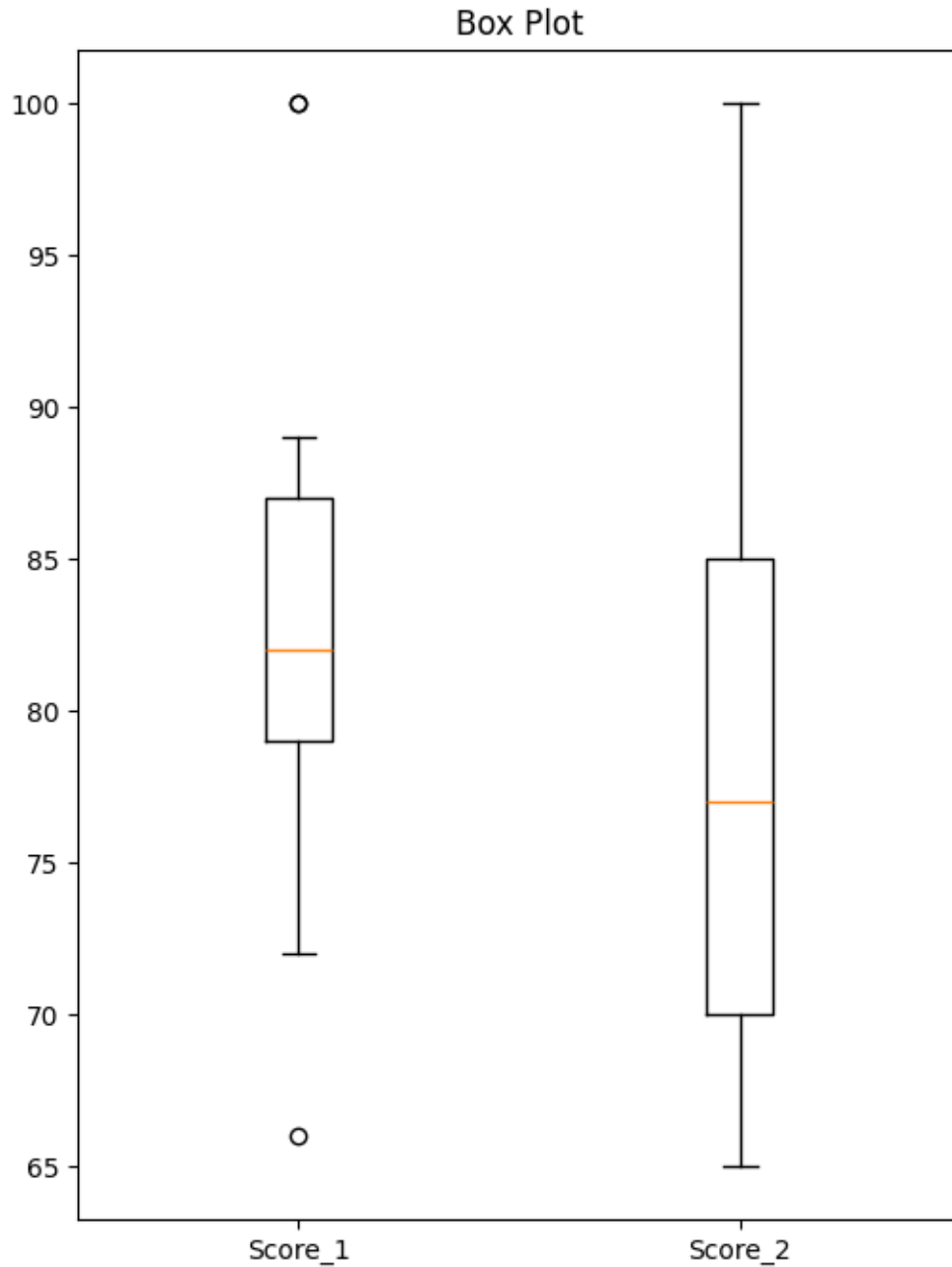
```
[20]: # 3. Check if the means of Score_1 and Score_2 are equal using Student's t-test.
      from scipy.stats import ttest_ind
      import numpy as np

      #Student's t-test is very good when the data follows normal distribution
      _,pval = ttest_ind(Score_1,Score_2,equal_var=False)
      print('p-value',pval)
```

```python
if pval <0.05:
  print("We reject null hypothesis")
else:
  print("We fail to reject null hypothesis")
```

```
p-value 0.15643810807436723
We fail to reject null hypothesis
```

[21]:
```python
# 4. Check if the means of Score_1 and Score_2 are equal using Mann-Whitney U␣
 ↪Test.
from scipy.stats import mannwhitneyu

#Mann-Whitney test is useful in general (no assumption of normal distribution)
_,pval = mannwhitneyu(Score_1,Score_2)
print('p-value',pval)

if pval <0.05:
  print("We reject null hypothesis")
else:
  print("We fail to reject null hypothesis")
```

```
p-value 0.05922484093227834
We fail to reject null hypothesis
```

7 References:

7.1 Theory:

1. Chirag Shah, "A Hands-On Introduction to Data Science," Cambridge University Press, 2020, **Section 3.3.1, 3.3.2, 3.3.3, 3.3.4**.

2. Plots: https://matplotlib.org/
3. Numpy: https://numpy.org/doc/stable/
4. Scipy: https://docs.python.org/3/library/statistics.html

[ ]: