

# **Data Science with R and Python**

Moinul Islam

9/15/2022

# Table of contents

<b>Preface</b>	<b>4</b>
<b>1 Basic Python</b>	<b>5</b>
1.1 Prerequisites . . . . .	5
1.2 Variables . . . . .	5
1.3 Strings . . . . .	5
1.3.1 Change case of string . . . . .	6
1.4 Numbers . . . . .	6
1.5 Comments . . . . .	8
1.6 Lists . . . . .	8
1.6.1 Neat outcome . . . . .	9
1.6.2 Print string . . . . .	9
1.7 Functions . . . . .	10
<b>2 Lecture 2</b>	<b>11</b>
<b>3 Lecture 3</b>	<b>12</b>
<b>4 Lecture 4</b>	<b>13</b>
<b>5 Lecture 5</b>	<b>14</b>
<b>6 Lecture 6</b>	<b>15</b>
<b>7 Lecture 7</b>	<b>16</b>
<b>8 Lecture 8</b>	<b>17</b>
<b>9 Lecture 9</b>	<b>18</b>
<b>10 Lecture 10</b>	<b>19</b>
<b>11 Lecture 11</b>	<b>20</b>
<b>12 Lecture 12</b>	<b>21</b>
<b>13 Lecture 13</b>	<b>22</b>

<b>14 Lecture 14</b>	<b>23</b>
<b>15 Lecture 15</b>	<b>24</b>
<b>References</b>	<b>25</b>

# Preface

This book contains the basic R and Python skills to learn data science. You can dive into the amazing data science world by using R and Python. Learn how to effectively analyze and visualize your data. No coding experience or skills needed. Data Science is about data gathering, analysis and decision-making. Data Science is used in many industries in the world today, e.g. banking, consultancy, healthcare, and manufacturing. Data science is not a one-step process such that will get to learned it in a short time. It's passes from many stages and every element is important. One should always follow the proper steps to reach the ladder. Every step has its value and it counts in your model.



# 1 Basic Python

## 1.1 Prerequisites

- There are three things we need to run the code in this lecture:
  - Python 3.10 (<https://www.python.org/downloads/>)
  - PyCharm community or professional (<https://www.jetbrains.com/pycharm/>)
  - A handful of other packages

## 1.2 Variables

- Let's start using a variable
- Variable names can contain only letters, numbers and underscores
- Space are not allowed in variable name
- Variables name should be short and descriptive

```
message = "Hello world!"  
print(message)
```

Hello world!

## 1.3 Strings

- A *string* is a series of characters
- In python, anything inside a quotes is considered a string
- We can use single or double quotes around the strings

```
message = "Python is a programming language"  
print(message)
```

Python is a programming language

```
message = 'I told my friend, "Python is a programming language"'
print(message)
```

I told my friend, “Python is a programming language”

### 1.3.1 Change case of string

- A lower case string can be changed to a title

```
name = "moinul islam"
print(name.title())
```

Moinul Islam

- We can also change the string to all upper or all lower

```
name = "moinul islam"
print(name.upper())
```

MOINUL ISLAM

## 1.4 Numbers

- Numbers are used quite often in python
  - Integers: We can add (+), multiply (\*), exponent (\*\*), and divide (/) integers in Python

```
print(2+3)
```

5

```
print(3-2)
```

1

```
print(2*3)
```

6

```
print(3**3)
```

27

```
print(3/2)
```

1.5

- Python supports the order of operation too.

```
print(2 + 3*4)
```

14

```
print((2+3)*4)
```

20

- Floats: Python calls any number with a decimal point a *float*

```
print(2*0.2)
```

0.4

- Integers and floats: When we divide two numbers, even if they are integers, you will always get a float

```
print(4/2)
```

2.0

- Underscore in numbers

- When you are writing long numbers, you can group digits using underscores

```
universe_age = 14_000_000_000  
print(universe_age)
```

14000000000

- Multiple assignments

- We can assign the value to more than one variables using just a single line

```
x, y, z = 12, 3, 5
print(x, y, z)
print(z)
```

12 3 5

5

- Constants

- A *constant* is a variable whose value stay the same throughout the life of a program.
- Python does not have built in command for constant
- However, a variable name with all capital letters treated as constant

```
ELON_MASK = 1_000_000_000_000
print(ELON_MASK)
```

1000000000000

## 1.5 Comments

- In Python, hash (#) indicates a comment.
- Anything following a # mark in your code is ignored by Python

```
# Say hello to your friends
print("Hello friends")
```

Hello friends

## 1.6 Lists

- List the elements of a variable



```
bicycles = ["trek", "cannondale", "redline", "specialized"]
print(bicycles)
```

```
['trek', 'cannondale', 'redline', 'specialized']
```

- Access an element from a list

```
bicycles = ["trek", "cannondale", "redline", "specialized"]
print(bicycles[0])
```

```
trek
```

### 1.6.1 Neat outcome

- You can format the element “trek” even more neatly by using title() method

```
bicycles = ["trek", "cannondale", "redline", "specialized"]
print(bicycles[0].title())
```

```
Trek
```

### 1.6.2 Print string

- You can print the strings that you are interested
- It starts counting from zero

```
bicycles = ["trek", "cannondale", "redline", "specialized"]
print(bicycles[1])
```

```
cannondale
```

- Python has special syntax to call the last item of a list

```
bicycles = ["trek", "cannondale", "redline", "specialized"]
print(bicycles[-1])
```

```
specialized
```

## 1.7 Functions

```
bicycles = ["trek", "cannondale", "redline", "specialized"]  
message = f"my first bicycle was a {bicycles[0].title()}"  
print(message)
```

my first bicycle was a Trek

- Modifying elements in a list

## 2 Lecture 2

## **3 Lecture 3**

## 4 Lecture 4

## 5 Lecture 5

## **6 Lecture 6**

## **7 Lecture 7**



## 8 Lecture 8

## 9 Lecture 9

## 10 Lecture 10

# 11 Lecture 11

## 12 Lecture 12

## **13 Lecture 13**

## 14 Lecture 14

## **15 Lecture 15**



## References