



Introduction to Machine Learning

Numpy, Pandas, Matplotlib: Basic Exercises

Jeson junzhen@42.us.org
Jesonleejunzhen.com

Summary: Learn about the essential Python libraries for machine learning

Contents

I	Intro to NumPy	4
II	Numpy	6
III	Exercise 00: Creating a numpy array using arange	7
IV	Exercise 01: Extract elements from an array	8
V	Exercise 02: Converting a one-dimensional array to two-dimensional	9
VI	Exercise 03: Inspecting a numpy array	10
VII	Exercise 04: Stack them (arrays) up!	11
VIII	Exercise 05: Replacing missing value	12
IX	Exercise 06: Drop all missing values from a numpy array	13
X	Exercise 07: Find duplicate records in a numpy array	14
XI	Pandas	15
XII	Exercise 00: Series	16
XIII	Exercise 01: DataFrame	17
XIV	Exercise 02: Display summary	18
XV	Exercise 03: Displaying rows	19
XVI	Exercise 04: Retrieving data	20
XVII	Exercise 05: Adding and dropping data	21
XVIII	Exercise 06: Calculate	22
XIX	Exercise 07: Sorting	23
XX	Matplotlib	24
XXI	Exercise 00: Linear graph	25
XXII	Exercise 01: Linear circle graph	26
XXIII	Exercise 02: Bar graph	27

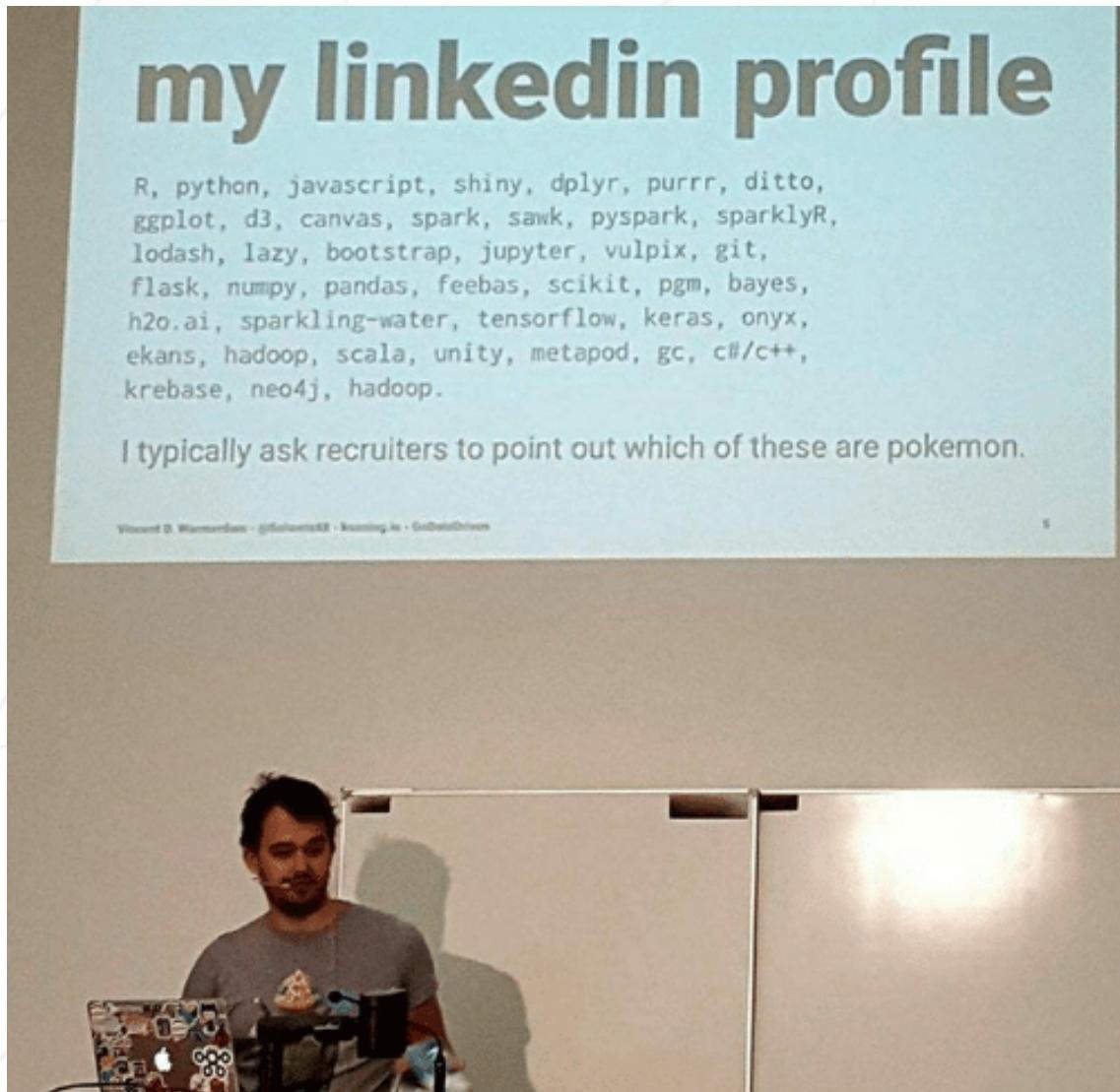
XXIV Exercise 03: Scatter Plot**29****XXV Exercise 04: Scatter plot with groups****31**



Eat, Sleep, Code, Repeat.

Chapter I

Intro to NumPy



This guy wins at LinkedIn..

Before we dive into learning about ‘machine learning’, let’s take a look at some of the essential programming skills needed to build a machine learning project using Python.

Some of the most commonly used python libraries are Numpy, Pandas, and Matplotlib.

These powerful libraries can be used to pre-process, analyze, and visualize data before we even begin building a machine learning model.

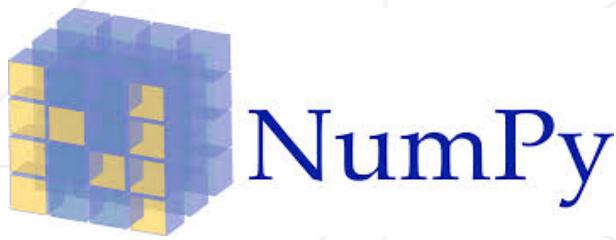
So let's not waste time and dive straight into it! Let's start off with downloading the libraries by doing this:

```
pip3 install numpy  
pip3 install pandas  
pip3 install matplotlib
```

You may need to add the ‘–user’ flag in order to add the libraries to your home directory, which won’t require any special privileges.

Chapter II

Numpy



Numpy is a math library for python. It enables us to do computation efficiently and effectively. Numpy has many amazing capabilities such as the ability to manipulate large arrays and matrices of numeric data.

To use the NumPy module, we need to import it using:

```
import numpy as np
```

Take a look at [this Numpy cheat sheet by DataCamp](#) to get an overall understanding.

Chapter III

Exercise 00: Creating a numpy array using arange

	Exercise
Creating a numpy array using arange	
Topics to study : Numpy Array	
Files to turn in : 00_numpy_array.py	
Forbidden functions : None	
Notes : n/a	

Create a Numpy array using arange. A NumPy array is a grid of values. They are similar to lists, except that every element of an array must be the same type. Write a program that will print the following output.

Desired output:

```
#> [0 1 2 3 4 5 6 7 8 9]
```

Okay, but why do I have to use an np array rather than a regular array?

Numpy arrays are better in terms of faster computation and ease of manipulation.

More information: <https://stackoverflow.com/questions/993984/what-are-the-advantages-of-numpy-over-regular-python-lists/994010994010>

Chapter IV

Exercise 01: Extract elements from an array

	Exercise
	Extract elements from an array
Topics to study :	Operator
Files to turn in :	01_operator.py
Forbidden functions :	None
Notes :	n/a

Declare and initialize an array as done below:

```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Then, extract all odd numbers from arr to achieve the desired output.
Desired output:

```
#> [1 3 5 7 9]
```

Chapter V

Exercise 02: Converting a one-dimensional array to two-dimensional

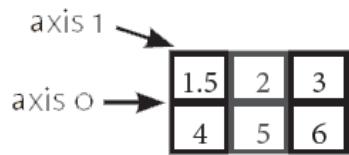
	Exercise
	Converting a one-dimensional array to two-dimensional
Topics to study :	Arrange, Reshape
Files to turn in :	02_dimensions.py
Forbidden functions :	None
Notes :	n/a

Convert a 1D array to a 2D array with 2 rows using arange

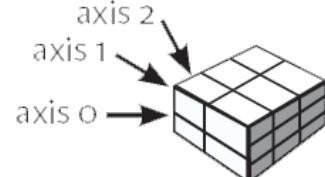
1D array



2D array



3D array



Declare and initialize an array as done below

```
np.arange(10)
```

Try recreating this output:

```
#> [[0 1 2 3 4]
   [5 6 7 8 9]]
```

Chapter VI

Exercise 03: Inspecting a numpy array

	Exercise
Inspecting a numpy array	
Topics to study : Python Objects	
Files to turn in : 03_dimensions.py	
Forbidden functions : None	
Notes : n/a	

Create a 2d array with 3 rows and 4 columns

```
myList = [[1, 2, 3, 4], [3, 4, 5, 6], [5, 6, 7, 8]]
```

Inspect its shape, datatype, size, number dimensions

Chapter VII

Exercise 04: Stack them (arrays) up!

	Exercise
	Stack them (arrays) up!
Topics to study :	2D array
Files to turn in :	04_stack.py
Forbidden functions :	None
Notes :	n/a

Stack the arrays a and b vertically

```
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)
```

Try creating this output:

```
#> [[0 1 2 3 4]
 [5 6 7 8 9]
 [1 1 1 1 1]
 [1 1 1 1 1]]
```

Chapter VIII

Exercise 05: Replacing missing value

	Exercise
	Replacing missing value
Topics to study :	<code>isnan, isninf</code>
Files to turn in :	<code>05_replace.py</code>
Forbidden functions :	None
Notes :	n/a

```
myList = [[1, 2, 3, 4], [3, 4, 5, 6], [5, 6, 7, 8]]
arr = np.array(myList, dtype='float')
arr[1,1] = np.nan # not a number
arr[1,2] = np.inf # infinite
print(arr)
```

Output:

```
#> [[ 1.  2.  3.  4.]
#>  [ 3. -1. -1.  6.]
#>  [ 5.  6.  7.  8.]]
```

Chapter IX

Exercise 06: Drop all missing values from a numpy array

	Exercise
Drop all missing values from a numpy array	
Topics to study : <code>isnan</code>	
Files to turn in : <code>06_drop.py</code>	
Forbidden functions : None	
Notes : n/a	

```
np.array([1,2,3,np.nan,5,6,7,np.nan])
```

Output:

```
[ 1.  2.  3.  5.  6.  7.]
```

Chapter X

Exercise 07: Find duplicate records in a numpy array

	Exercise
Find duplicate records in a numpy array	
Topics to study : <code>np.unique</code>	
Files to turn in : <code>07_duplicate.py</code>	
Forbidden functions : None	
Notes : n/a	

Stack the arrays a and b vertically

```
np.random.seed(100)
a = np.random.randint(0, 5, 10)
print('Array: ', a)
```

Mark the unique position as false

Output:

```
#> [False  True False  True False False  True  True  True  True]
```

Chapter XI

Pandas



Pandas is one of the most popular Python libraries for Data Science and Analytics. I like to say it's the "SQL of Python." Why? Because pandas helps you to manage two-dimensional data tables in Python. Of course, it has many more features, let's explore the power of pandas in this tutorial.

To use pandas, we need to import it using:

```
import pandas as pd
```

Take a look at [this pandas cheat sheet by DataCamp](#) to get an overall understanding.

Chapter XII

Exercise 00: Series

	Exercise
	Series
Topics to study :	Series
Files to turn in :	00_series.py
Forbidden functions :	None
Notes :	n/a

Create a one-dimensional labeled array capable of holding any data type using a Series.

```
#> A    3  
#> B    5  
#> C    7  
#> D    4
```

Chapter XIII

Exercise 01: DataFrame

	Exercise
	DataFrame
Topics to study :	DataFrame
Files to turn in :	01_data_frame.py
Forbidden functions :	None
Notes :	n/a

Create a two-dimensional labeled data structure with columns of potentially different types using a DataFrame.

Example:

```
data = {'Country': ['Belgium', 'India', 'Brazil'],
'Capital': ['Brussels', 'New Delhi', 'Brasilia'],
'Population': [11190846, 1303171035, 207847528]}
```

	Country	Capital	Population
1	Belgium	Brussels	11190846
2	India	New Delhi	1303171035
3	Brazil	Brasilia	207847

Now take this data:

```
data = {'animal': ['cat', 'cat', 'snake', 'dog', 'dog', 'cat', 'snake', 'cat', 'dog', 'dog'],
'age': [2.5, 3, 0.5, np.nan, 5, 2, 4.5, np.nan, 7, 3],
'veisits': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Using the dictionary above, answer the next 6 questions.

Chapter XIV

Exercise 02: Display summary

	Exercise
	Display summary
Topics to study :	<code>info()</code> , <code>describe()</code>
Files to turn in :	<code>02_display.py</code>
Forbidden functions :	<code>None</code>
Notes :	<code>n/a</code>

Display a summary of the basic information about this DataFrame and its data.

Chapter XV

Exercise 03: Displaying rows

	Exercise
	Displaying rows
Topics to study :	iloc
Files to turn in :	03_rows.py
Forbidden functions :	None
Notes :	n/a

Return the first 3 rows of the DataFrame df.

Chapter XVI

Exercise 04: Retrieving data

	Exercise
	Retrieving data
Topics to study :	.between()
Files to turn in :	04_retrieve.py
Forbidden functions :	None
Notes :	n/a

Select the rows the age is between 2 and 4 (inclusive).

Chapter XVII

Exercise 05: Adding and dropping data

	Exercise
	Adding and dropping data
Topics to study :	drop()
Files to turn in :	05_adding_dropping.py
Forbidden functions :	None
Notes :	n/a

Append a new row 'k' to df with your choice of values for each column. Then delete that row to return the original DataFrame.

Chapter XVIII

Exercise 06: Calculate

	Exercise
	Calculate
Topics to study :	group_by()
Files to turn in :	06_calculate.py
Forbidden functions :	None
Notes :	n/a

Calculate the mean age for each different animal in df.

Chapter XIX

Exercise 07: Sorting

	Exercise
	Sorting
Topics to study :	<code>sort_values()</code>
Files to turn in :	<code>07_sort.py</code>
Forbidden functions :	None
Notes :	n/a

Sort df first by the values in the 'age' in descending order, then by the values in the 'visit' column in ascending order.

Chapter XX

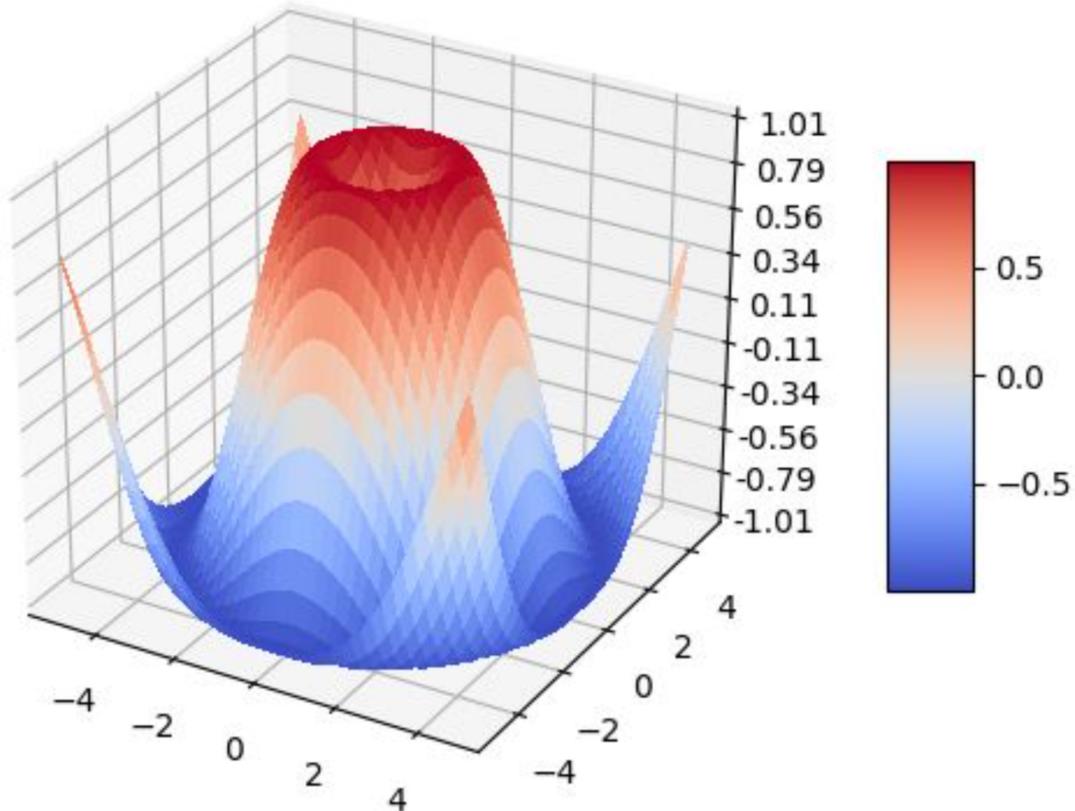
Matplotlib

Matplotlib is a python library used to create 2D graphs and plots. It has a module named pyplot which makes things easy for plotting by providing features to control line styles, font properties, formatting axes etc.

To use Matplotlib, we need to import it using:

```
import matplotlib.pyplot as plt
```

Take a look at [this matplotlib cheat sheet by DataCamp](#) to get an overall understanding.

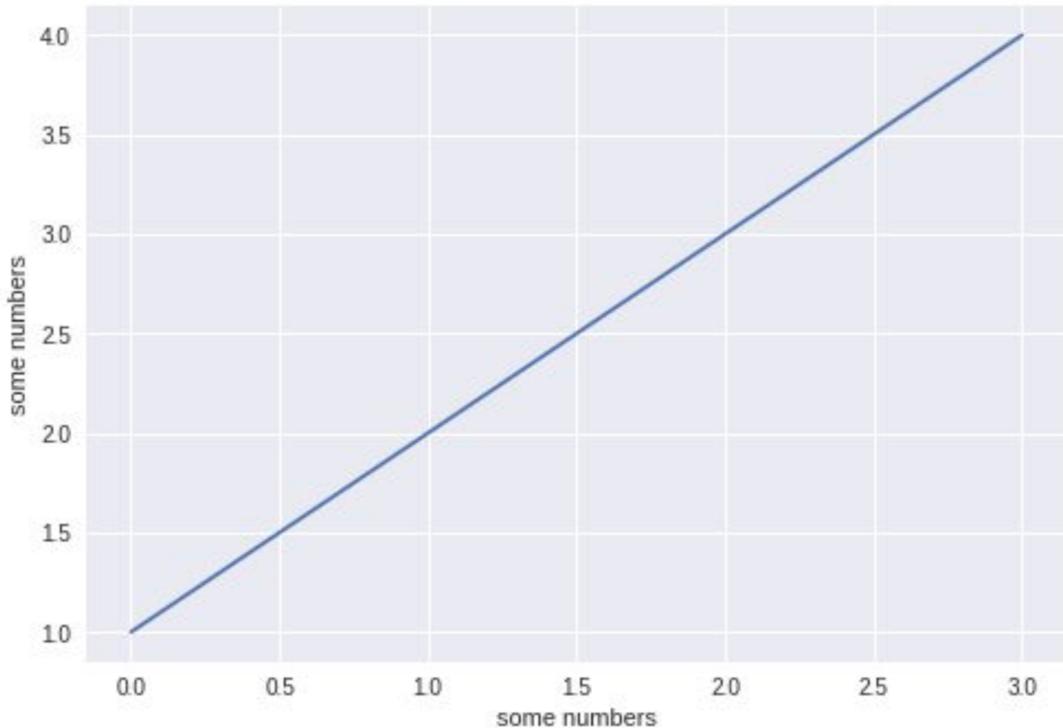


Chapter XXI

Exercise 00: Linear graph

	Exercise
	Linear graph
Topics to study :	<code>plotting</code>
Files to turn in :	<code>00_linear_graph.py</code>
Forbidden functions :	None
Notes :	n/a

Plot this linear graph using matplotlib.



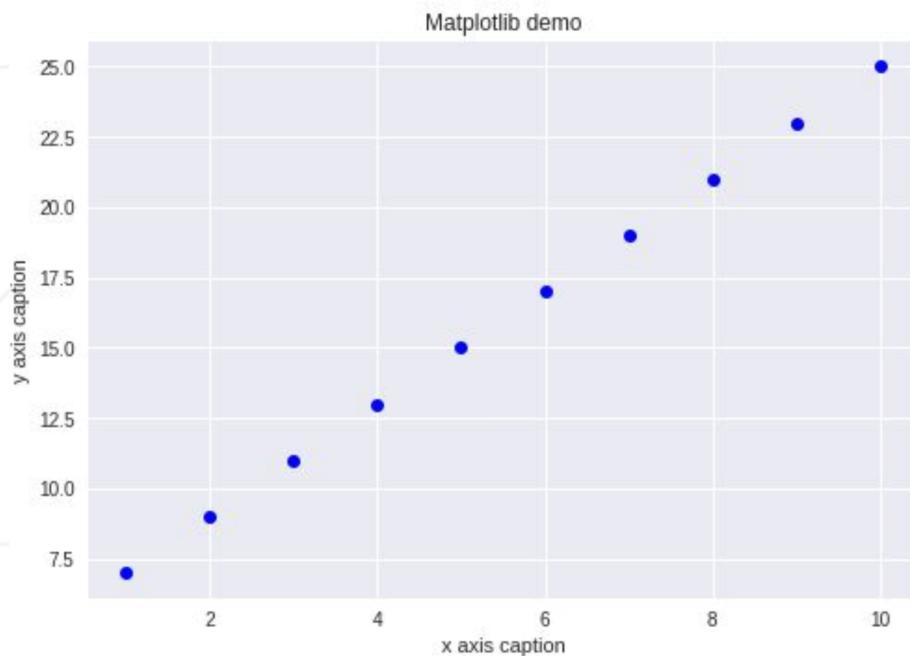
Chapter XXII

Exercise 01: Linear circle graph

	Exercise
	Linear circle graph
Topics to study :	plotting
Files to turn in :	01_linear_circle.py
Forbidden functions :	None
Notes :	n/a

Plot this linear graph using matplotlib with this data:

```
import numpy as np
x = np.arange(1,11)
y = 2 * x + 5
```



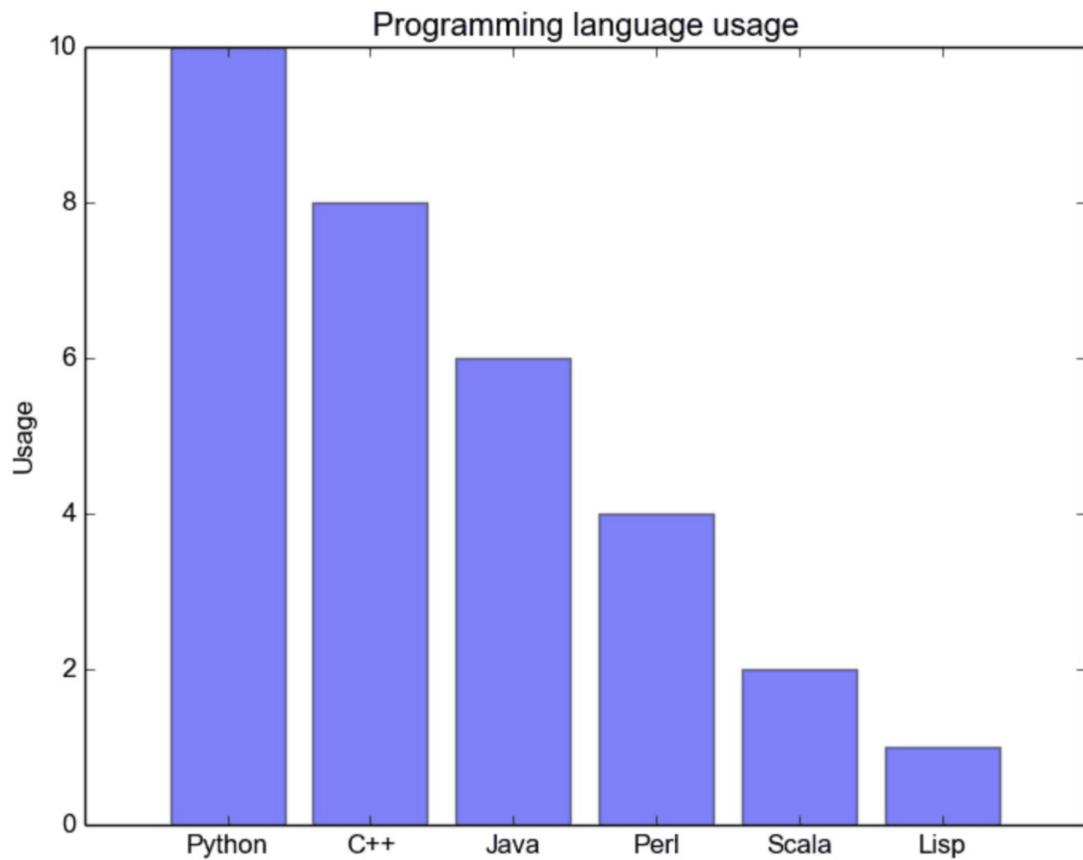
Chapter XXIII

Exercise 02: Bar graph

	Exercise
	Bar graph
Topics to study :	plotting
Files to turn in :	02_bar.py
Forbidden functions :	None
Notes :	n/a

Plot a bar graph with this data:

```
import matplotlib.pyplot as plt; plt.rcParams()  
import numpy as np  
import matplotlib.pyplot as plt  
  
objects = ('Python', 'C++', 'Java', 'Perl', 'Scala', 'Lisp')  
y_pos = np.arange(len(objects))  
performance = [10,8,6,4,2,1]
```



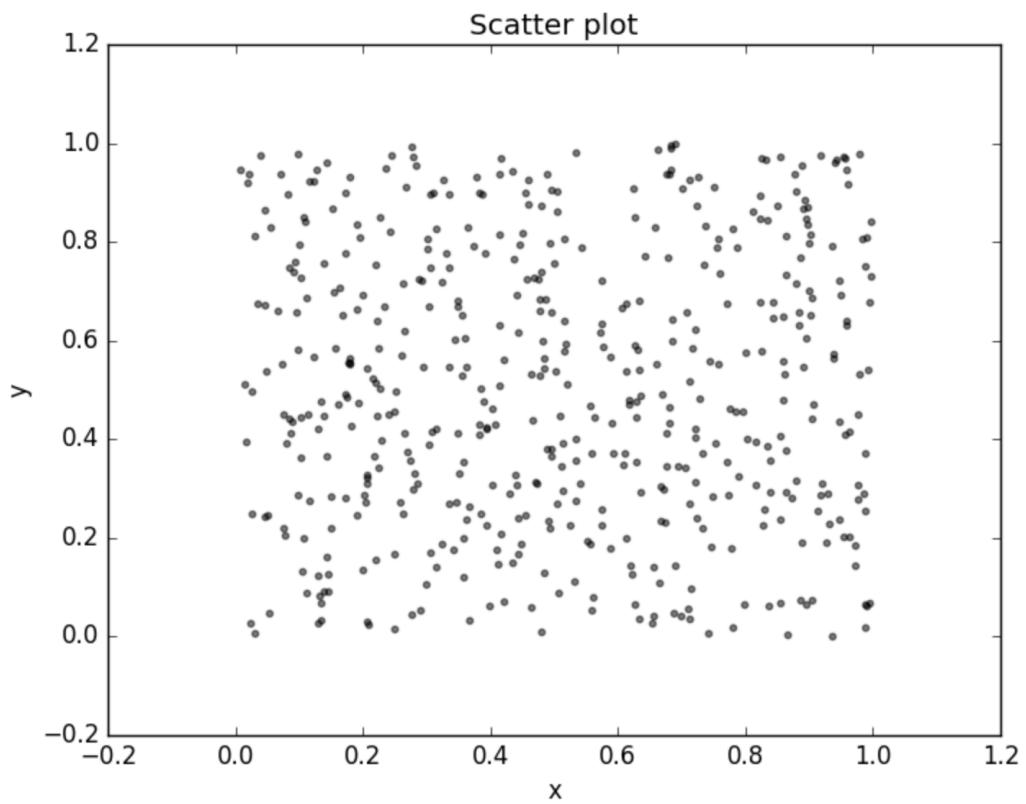
Chapter XXIV

Exercise 03: Scatter Plot

	Exercise
	Scatter Plot
Topics to study :	<i>plotting</i>
Files to turn in :	03_scatterplot.py
Forbidden functions :	None
Notes :	n/a

Plot a scatter plot with this data:

```
N = 500
x = np.random.rand(N)
y = np.random.rand(N)
colors = (0,0,0)
area = np.pi*3
```



Chapter XXV

Exercise 04: Scatter plot with groups

	Exercise
	Scatter plot with groups
Topics to study :	plotting
Files to turn in :	04_group_scatterplot.py
Forbidden functions :	None
Notes :	n/a

Plot a scatter plot with this data:

```
# Create data
N = 60
g1 = (0.6 + 0.6 * np.random.rand(N), np.random.rand(N))
g2 = (0.4+0.3 * np.random.rand(N), 0.5*np.random.rand(N))
g3 = (0.3*np.random.rand(N),0.3*np.random.rand(N))

data = (g1, g2, g3)
colors = ("red", "green", "blue")
groups = ("coffee", "tea", "water")
```

