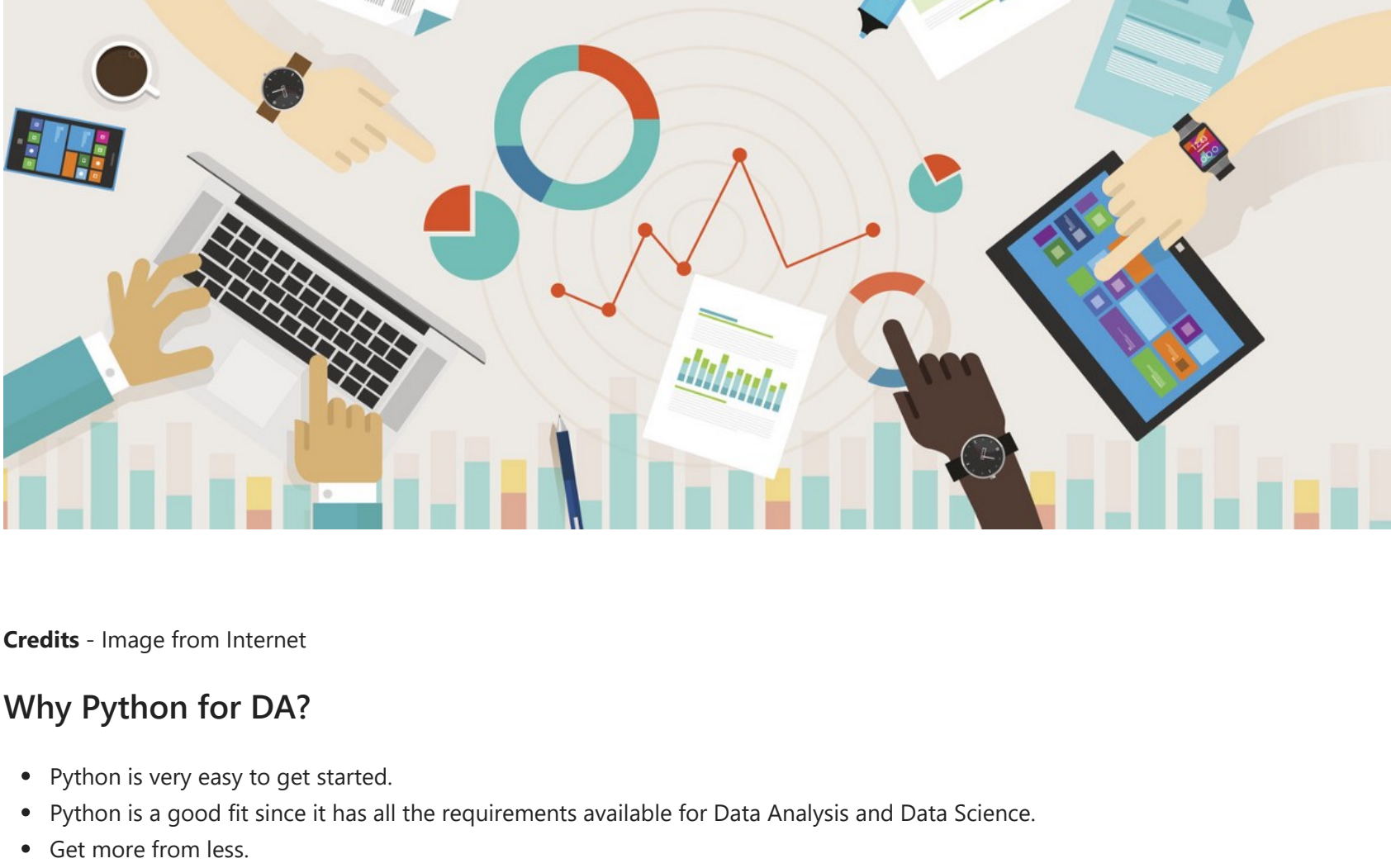


Basic Understanding of Data Using Python



Credits - Image from Internet

Why Python for DA?

- Python is very easy to get started.
- Python is a good fit since it has all the requirements available for Data Analysis and Data Science.
- Get more from less.
- The community of Python is so large that you can find anything easily.

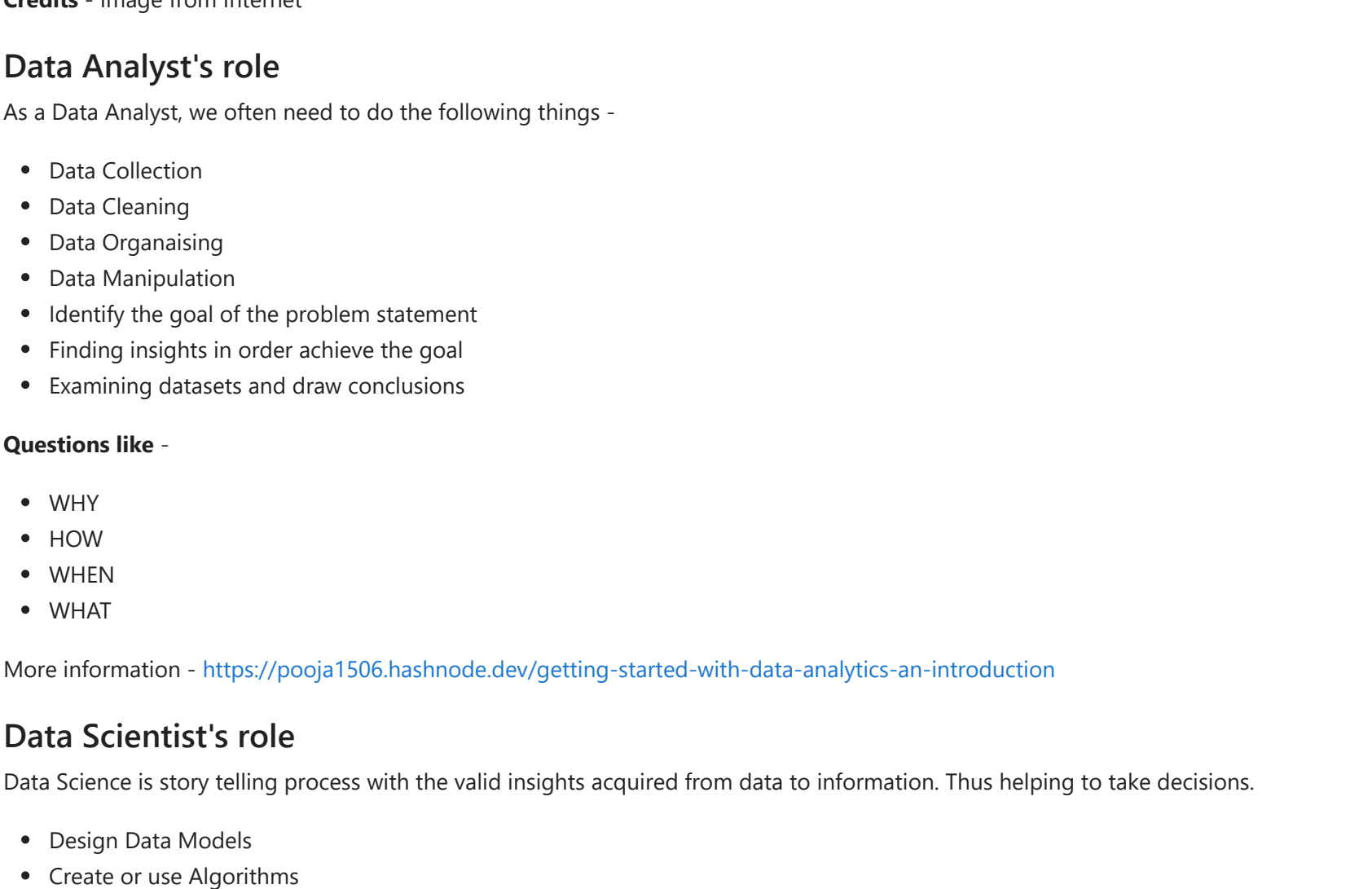
Note - Check google trends for comparison.

- Before clicking the below link, please sign-in your google account on web.
- Link → https://trends.google.com/trends/explore?cat=1299&date=today%205-y&q=%2Fm%2F05z1_%2Fm%2F0212jm,%2Fm%2F03djj7,%2Fm%2F052tr,%2Fm%2F07sbkfb

But when it comes to speed and compatibility, Python is slower...

- Link → <https://juliacomputing.com/blog/2020/06/fast-csv/>

Data Analysis vs Data Science



Credits - Image from Internet

Data Analyst's role

As a Data Analyst, we often need to do the following things -

- Data Collection
- Data Cleaning
- Data Organising
- Data Manipulation
- Identify the goal of the problem statement
- Finding insights in order to achieve the goal
- Examining datasets and draw conclusions

Questions like -

- WHY
- HOW
- WHEN
- WHAT

More information - <https://pooja1506.hashnode.dev/getting-started-with-data-analytics-an-introduction>

Data Scientist's role

Data Science is a story telling process with the valid insights acquired from data to information. Thus helping to take decisions.

- Design Data Models
- Create or use Algorithms
- Predict the future outcomes with accuracy
- Make decisions from the insights

More information

- <https://www.northeastern.edu/graduate/blog/what-does-a-data-scientist-do/>
- <https://www.northeastern.edu/graduate/blog/data-analytics-vs-data-science/>

Note - Data Scientist with analytical skills is a Blessing upon the blessed.

Practise question

1. Collect data from online using Pandas.
2. Check if data cleaning is necessary.

- **yes** → Clean the data

- **no** → Proceed
- 3. Identify the relationship between data variables.
- Apply Correlation (Bi-variate analysis)
- Plot the relationship

- **Data Source** → http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_Dinov_020108_HeightsWeights

1. Collect data from online

1. Pandas is python library mainly used for data analysis.
2. It is similar to doing analysis on Excel.
3. It is one of the best open source libraries available for doing data manipulation and data wrangling.

More information → <https://pandas.pydata.org/>

```
In [1]: # import statement
import pandas as pd

py -m pip install pandas --user

read_html() extracts all the tables from the html page.

In [2]: # data_source
data_source = 'http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_Dinov_020108_HeightsWeights'

In [3]: # data storing
data = pd.read_html(data_source)

In [4]: # check type
type(data)

Out[4]: list

In [5]: # check len
len(data)

Out[5]: 3

In [6]: # display table 0
data[0]

Out[6]: 0 Contents 1 SOCR Data - 25,000 Records of Human...

In [7]: # display table 1
data[1]

Out[7]:
   Index  Height(Inches)  Weight(Pounds)
0      1      65.78      112.99
1      2      71.52      136.49
2      3      69.40      153.03
3      4      68.22      142.34
4      5      67.79      144.30
...    ...            ...
195    196      65.80      120.84
196    197      66.11      115.78
197    198      68.24      128.30
198    199      68.02      127.47
199    200      71.39      127.88

200 rows x 3 columns

In [8]: # display table 2
data[2]

Out[8]:
   0      1      2      3      4      5      6      7      8      9     10     11
0 (default) Deutsch Español Français Italiano Português 日本語  България العربية المتحدة الإمارات العربية المتحدة Suomi  हिंदी भारत में Norge
1 한국어 中文 繁体中文 Русский Nederlands Ελληνικά Hrvatska Česká republika Danmark Polska România Sverige

In [9]: # df
df = data[1]

In [10]: # display the head
df.head()

Out[10]:
   Index  Height(Inches)  Weight(Pounds)
0      1      65.78      112.99
1      2      71.52      136.49
2      3      69.40      153.03
3      4      68.22      142.34
4      5      67.79      144.30

In [11]: # display the tail
df.tail()

Out[11]:
   Index  Height(Inches)  Weight(Pounds)
195    196      65.80      120.84
196    197      66.11      115.78
197    198      68.24      128.30
198    199      68.02      127.47
199    200      71.39      127.88

In [12]: # display the columns
df.columns

Out[12]: Index(['Index', 'Height(Inches)', 'Weight(Pounds)'], dtype='object')

In [13]: # select any column
df['Height(Inches)'].head()

Out[13]:
0    65.78
1    71.52
2    69.40
3    68.22
4    67.79
Name: Height(Inches), dtype: float64
```

2. Check if data cleaning is necessary

Data Cleaning is one of the important aspects in both Data Analysis and Data Science roles.

- It is one of the procedural steps where a data analyst or data scientist spends most of their time.

More information → https://en.wikipedia.org/wiki/Data_cleaning

a. Check for any NaN values → Missing values

Is my df null? If yes, show me the columns if any.

```
In [14]: # isnull
df.isnull()

Out[14]:
Index      False
Height(Inches)  False
Weight(Pounds)  False
dtype: bool

In [15]: df.isnull().sum()

Out[15]:
Index      0
Height(Inches)  0
Weight(Pounds)  0
dtype: int64

In [16]: # isna
df.isna()

Out[16]:
Index      False
Height(Inches)  False
Weight(Pounds)  False
dtype: bool
```

- Since the dataset is sort of big, we cannot see all the values. Infact we cannot comprehend the actual missing values from the `isna()` dataset.
- In order to get the actual values (indices), the below function can be used.

Above result is clear, every column has `non-nan` values. Hence we can proceed with further steps.

b. Check for the datatypes for each column

```
In [17]: # dtypes
df.dtypes

Out[17]:
Index      int64
Height(Inches)  float64
Weight(Pounds)  float64
dtype: object

In [18]: # info
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   Index      200 non-null    int64
 1   Height(Inches)  200 non-null    float64
 2   Weight(Pounds)  200 non-null    float64
dtypes: float64(2), int64(1)
memory usage: 4.8 KB
```

Seems like every column has a unique data type. If at all there is then it is required to purify the data - make sure all the values are of same type.

c. Overall description of the data frame

```
In [19]: # describe
df.describe()

Out[19]:
   Index  Height(Inches)  Weight(Pounds)
count  200.000000      200.000000      200.000000
mean    100.500000      67.949800      127.221950
std     57.879185      1.940363      11.960959
min      0.000000      63.430000      97.900000
25%     50.750000      66.522500      119.895000
50%     100.500000      67.935000      127.875000
75%     150.250000      69.202500      136.097500
max     200.000000      73.900000      158.960000
```

d. Some visualization to explore more about the data

- We can use pandas plotting functions like `plot()` to explore about the data visually.
- `plot()` can show the following plots -
 - **line** → line plot (default)
 - **bar** → vertical bar plot
 - **barh** → horizontal bar plot
 - **hist** → histogram
 - **box** → boxplot
 - **kde** → Kernel Density Estimation plot
 - **density** → same as 'kde'
 - **area** → area plot
 - **pie** → pie plot
 - **scatter** → scatter plot
 - **hexbin** → hexbin plot

Ugly plot example

```
In [20]: # plot
df.plot()

Out[20]: <AxesSubplot:~>


```

The above is the plot of all the data variables. If at all there is then it is required to purify the data - make sure all the values are of same type.

Plotting without unimportant data variables - excluded Index

The above is the plot of both `Heights` and `Weights` from the data frame `df`.

```
In [21]: # selecting multiple columns and displaying head - heights and weights
df[['Height(Inches)', 'Weight(Pounds)']].head()

Out[21]:
   Height(Inches)  Weight(Pounds)
0      65.78      112.99
1      71.52      136.49
2      69.40      153.03
3      68.22      142.34
4      67.79      144.30

In [22]: # line plot with heights and weights
df[['Height(Inches)', 'Weight(Pounds)']].plot()

Out[22]: <AxesSubplot:~>


```

3. Relationship between data variables

Correlation - one of the statistical measurements applied to find out if any two variables are linearly related.

- If one variable is increasing, then other variable also increases. Vice versa.
- For example
 - If income of an employee increases then the household expenses increase.
 - If income of an employee decreases then the household expenses decrease.
- Scatter plot is really helpful to find the relationship between two variables. With this, it can be easily noticed the linear trend as well.

Correlation plots based on the correlation value obtained. → https://en.wikipedia.org/wiki/Correlation_and_dependence#/media/File:Correlation_examples2.svg

b. Plot the relationship

```
In [23]: # df.plot

In [24]: # scatter plot
# x - heights
# y - weights
df.plot(x='Height(Inches)', y='Weight(Pounds)', kind='scatter')

Out[24]: <AxesSubplot:~>


```

From the above plot, we can see that when `Heights` increase, then `Weights` also increased.

What if we interchange the values?

```
In [25]: # scatter plot
# x - weights
# y - heights
df.plot(x='Weight(Pounds)', y='Height(Inches)', kind='scatter')

Out[25]: <AxesSubplot:~>


```

a. Find the Correlation

Correlation value ranges from `-1` to `1`.

- If the calculated correlation value is -
 - `-1`, then it is perfectly **negative correlation**
 - `1`, then it is perfectly **positive correlation**
 - `< -1`, then it means that **error** in the correlation measurement
 - `> 1`, then it means that **error** in the correlation measurement

More information → <https://www.investopedia.com/terms/c/correlationcoefficient.asp>

```
In [26]: # correlation
df.corr()

Out[26]:
   Index  Height(Inches)  Weight(Pounds)
Index      1.000000      -0.094260      -0.128882
Height(Inches) -0.094260      1.000000      0.556865
Weight(Pounds) -0.128882      0.556865      1.000000

In [27]: # correlation table with colormap
relation = df.corr()
relation.style.background_gradient(cmap='Greens')

Out[27]:
   Index  Height(Inches)  Weight(Pounds)
Index      1.000000      -0.094260      -0.128882
Height(Inches) -0.094260      1.000000      0.556865
Weight(Pounds) -0.128882      0.556865      1.000000
```

Case Study → Activity

1. Select any one of these or you can find your own topic of interest not specifically from below.

- Study to analyse peoples' habits on YouTube platform
- Study to analyse the changes occurred in peoples' life due to Demonitization
- Study to analyse the students' overall development due to online education

2. Create a google form where you can have a set of questions and answer options.

- Have atleast 8 to 10 questions
- 3. Collect the data from your friends, families etc. (by sharing the link).
- The data will be stored in your drive (in a spreadsheet)

4. Once the data is collected -
 - Create your own data variables from the questions
 - Try to basic analysis like processing and visualization

Note - To learn how to create google forms (Questionnaires) and collect the data,

- Please watch this video → <https://www.youtube.com/watch?v=vQw2jDlyIDU>

End

Loading MathJax...
Loading MathJax...
Loading MathJax...
Loading MathJax...