Learning Objectives

Data Analytics

Pandas

Working with CSV files

Data Analytics: An Analogy

You are An Apple Farmer

- You own an apple farm and you want to know the number of apples you grow but you are too busy with the farm so you hire someone to count them. You sell your apples too, and you get your apple counter to keep a record of the number of apples you have in the beginning and at the end of the day, every day. Many days and months pass and you put sheet after sheet of the apple count together and you discover patterns and trends in the purchasing behaviour of your customers.

- The trends and patterns help you realise that during the colder season, your output of apples are the same but people buy less than during the summer. You then set out to dig deeper into this trend, and find ways to keep the sales of apples consistent throughout the year, beating your competitors at the game and becoming an apple farm tycoon.

Data Analytics: An Analogy

Apples are your data, tracking them is important, *analysis is key*

Apples are your products/assets, knowing their movement is important. For starters, you will know if your supply of apples matches the market's demand, as well as the consistency of the ratio of demand to supply throughout the year.

Reducing the cost of apple production gives you your profit. When you have enough data, you will find trends and patterns in your production. These trends can help you understand your own organisation better, help you reduce inefficiency and therefore reduce your apple production costs.

Data Analytics

- In simple words, data analysis is the process of collecting and organizing data in order to draw helpful conclusions from it. The process of data analysis uses analytical and logical reasoning to gain information from the data.

- The main purpose of data analysis is to find meaning in data so that the derived knowledge can be used to make informed decisions.

Real-life Examples

1. **Economics**
   Analyzing data to form patterns and understanding trends about how the economy in various sectors is growing, is something very essential for economists. Therefore, a lot of economists have started using Python and Pandas to analyze huge datasets.

2. **Recommendation Systems**
   We all have used Spotify or Netflix and been appalled at the brilliant recommendations provided by these sites. The recommendation system is possible only by learning and handling huge masses of data.

3. **Stock Prediction**
   The stock market is extremely volatile. However, that doesn’t mean that it cannot be predicted. With the help of Pandas and a few other libraries like NumPy and matplotlib, we can easily make models which can predict how the stock markets turn out.

Why Pandas?

- In real-life, as a Data Scientist, we need to play with data and pandas essentially helps us to give a head start with it.

- Be it reading the data, analyzing /cleaning /changing / transforming. Pandas has a solution for all of it.

- It even helps in calculating basic statistics. If we go on there will be number of use-cases, we have summarized a few of them in the next slide.

- Pandas help in analyzing large volumes of data with ease.

Specific-Use Cases

- **Creating data:** You can create your own dataset using pandas.

- **Reading data:** Often we get data in various formats such as csv, excel, in sql database, json etc. Pandas have some simple functions to perform this task

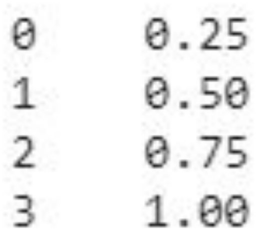
- **Analyzing data:** With the help of other libraries like matplotlib, it helps in statistical analysis and visualize data with plots such as histograms, scatter plots etc

- **Cleaning data and handling missingness:** Often Data Scientists get unclean data with a lot missing values and we need to have a solution to deal with it. Pandas facilitate in handling this issue.

Pandas Objects

- At the core of the pandas library there are two fundamental data structures/objects:
  - Series
  - Data Frames

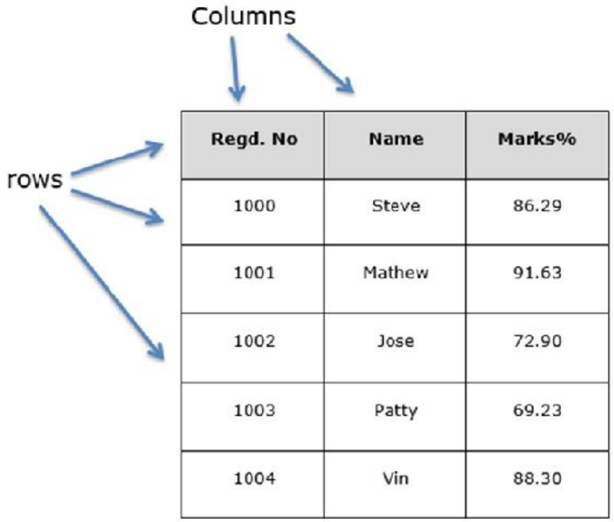
- **Series:** stores single column data along with an index. An index is just a way to "number" the Series object.



```
0    0.25
1    0.50
2    0.75
3    1.00
```

Pandas Objects

- **DataFrame:** is a two-dimensional tabular data structure with labeled axes (rows and columns). It is conceptually useful to think of a DataFrame object as a collection of Series objects.



Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

Pandas Operations

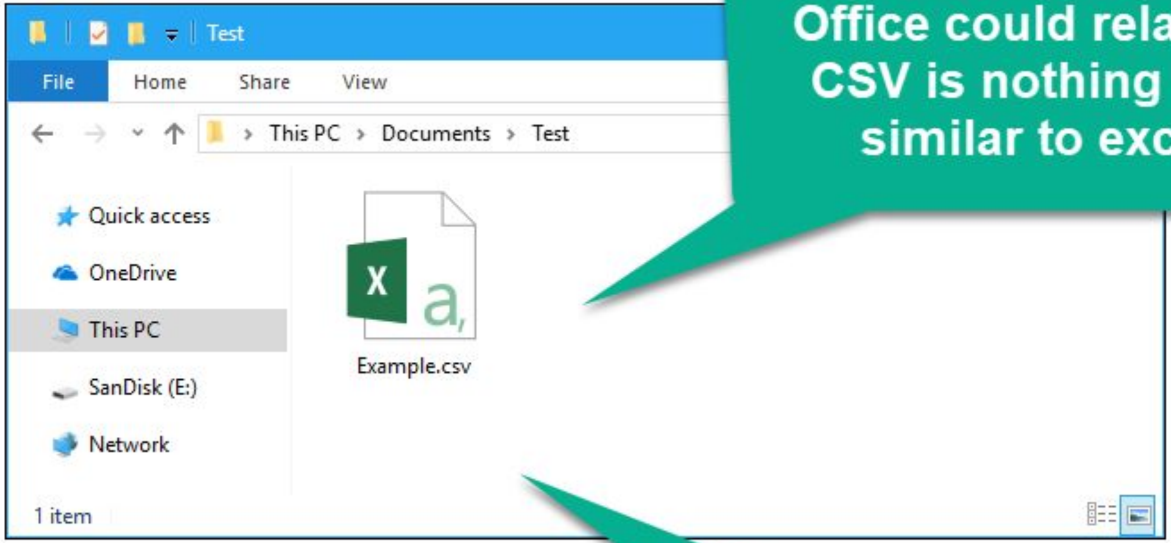
- Data indexing and selection
  - **Use case:** Helps fetch a data record when we are dealing with large volumes of data.
  - For ex: If you have a data with 10 million records, you can easily fetch information of a particular serial no/index no with pandas.
  - This operation may not be feasible with MS Excel while dealing with such large volumes

Pandas Operations

- Data Wrangling & Handling Missing Values - read the notebook.
  - **Use case:** Often Data Scientists get unclean data with a lot of missing values and we need to have a solution to deal with it. Pandas facilitate in handling this issue
- Pandas String Operations
  - **Use case:** Helps in Handling Missing Values

Working with CSVs

What is a CSV file?



Windows users using MS Office could relate to this. CSV is nothing but a file similar to excel files

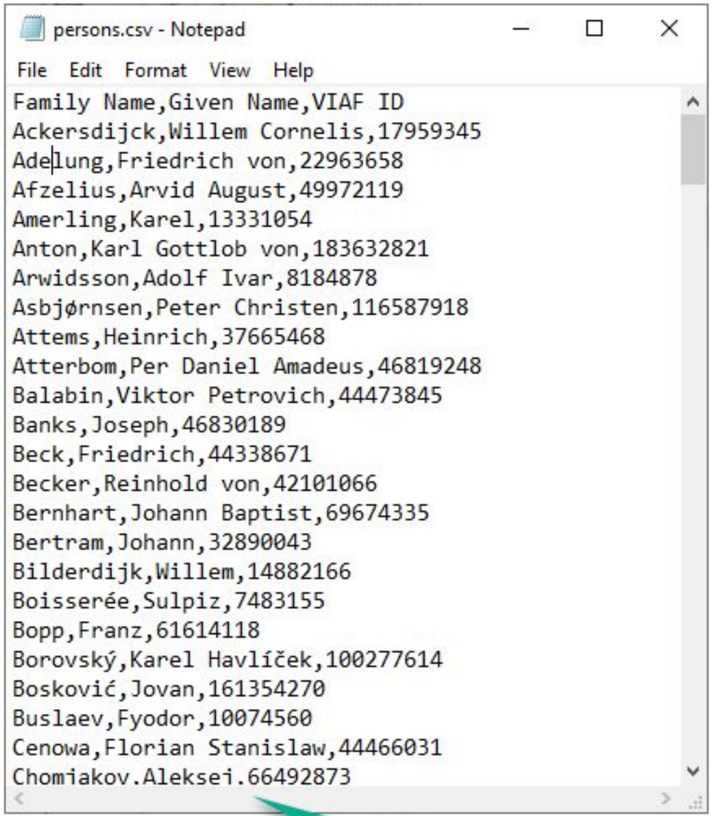
Only catch is, in CSV you store data with comma separated values

DPhi

What is a CSV ?

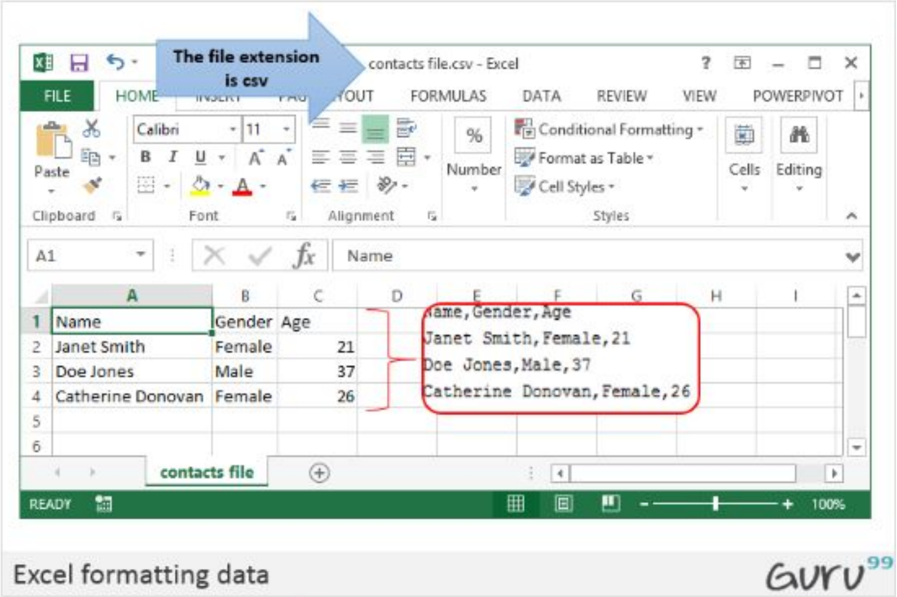
- CSV files are normally created by programs that handle large amounts of data. They are a convenient way to export data from spreadsheets and databases as well as import or use it in other programs.
- CSV (**Comma Separated Values**) is a simple file format used to store tabular data, such as a spreadsheet or database.
- A CSV file stores tabular data (numbers and text) in plain text.
- Each line of the file is a data record.
- Each record consists of one or more fields, separated by commas.
- The use of the comma as a field separator is the source of the name for this file format.

How does it look like?



```
Family Name,Given Name,VIAF ID
Ackersdijck,Willem Cornelis,17959345
Adelung,Friedrich von,22963658
Afzelius,Arvid August,49972119
Amerling,Karel,13331054
Anton,Karl Gottlob von,183632821
Arwidsson,Adolf Ivar,8184878
Asbjørnsen,Peter Christen,116587918
Attems,Heinrich,37665468
Atterbom,Per Daniel Amadeus,46819248
Balabin,Viktor Petrovich,44473845
Banks,Joseph,46830189
Beck,Friedrich,44338671
Becker,Reinhold von,42101066
Bernhart,Johann Baptist,69674335
Bertram,Johann,32890043
Bilderdijk,Willem,14882166
Boisserée,Sulpiz,7483155
Bopp,Franz,61614118
Borovský,Karel Havlíček,100277614
Bošković,Jovan,161354270
Buslaev,Fyodor,10074560
Cenowa,Florian Stanislaw,44466031
Chomiakov,Aleksei,66492873
```

Looks like this when opened on Notepad



The file extension is csv

	A	B	C
1	Name	Gender	Age
2	Janet Smith	Female	21
3	Doe Jones	Male	37
4	Catherine Donovan	Female	26

```
ame,Gender,Age
Janet Smith,Female,21
Doe Jones,Male,37
Catherine Donovan,Female,26
```

Excel formatting data

Looks like this when opened on MS Excel

Working with CSV files in Python

- For working CSV files in python, there is an inbuilt module named csv.

- However, a common method for working with CSV files is using Pandas. It makes importing and analyzing data much easier.

- One crucial feature of Pandas is its ability to write and read Excel, CSV, and many other types of files.

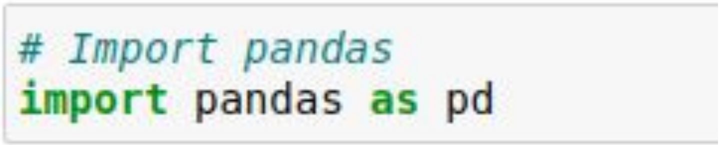
Pandas read_csv

- Functions like the Pandas read_csv() method enable you to work with files effectively.

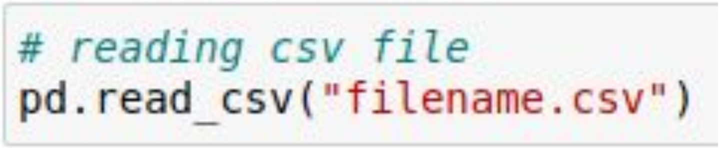
- The read_csv() function reads the CSV file into a DataFrame object.

- A CSV file is similar to a two-dimensional table and the DataFrame object represents two dimensional tabular view.

- The most basic way to read a csv file in Pandas:



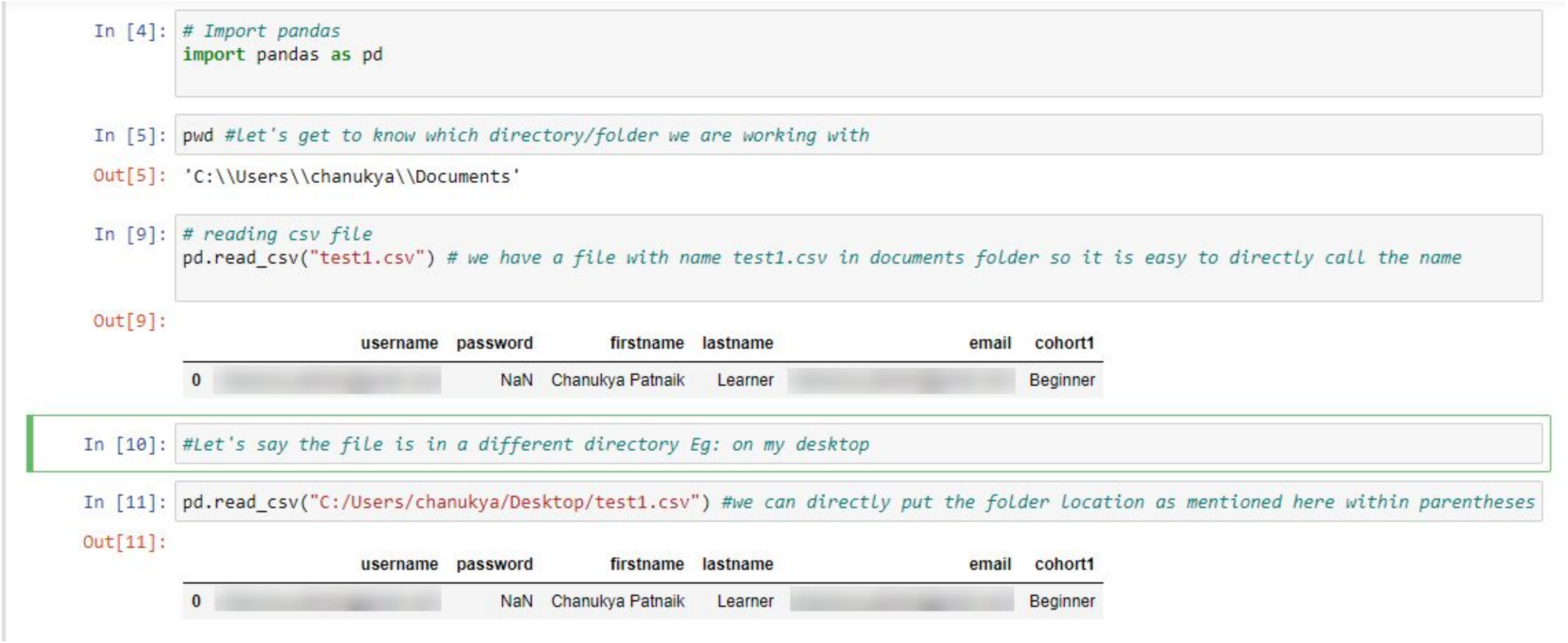
```python
# Import pandas
import pandas as pd
```



```python
# reading csv file
pd.read_csv("filename.csv")
```

- Read the next slide to understand how to provide filename

Pandas read_csv



In [4]:
```
# Import pandas
import pandas as pd
```

In [5]:
```
pwd #let's get to know which directory/folder we are working with
```

Out[5]:
```
'C:\\Users\\chanukya\\Documents'
```

In [9]:
```
# reading csv file
pd.read_csv("test1.csv") # we have a file with name test1.csv in documents folder so it is easy to directly call the name
```

Out[9]:

	username	password	firstname	lastname	email	cohort1
0	[redacted]	NaN	Chanukya Patnaik	Learner	[redacted]	Beginner

In [10]:
```
#Let's say the file is in a different directory Eg: on my desktop
```

In [11]:
```
pd.read_csv("C:/Users/chanukya/Desktop/test1.csv") #we can directly put the folder location as mentioned here within parentheses
```

Out[11]:

	username	password	firstname	lastname	email	cohort1
0	[redacted]	NaN	Chanukya Patnaik	Learner	[redacted]	Beginner

Pandas read_csv

- There are many other things one can do through this function only to change the returned object completely.

- For instance, one can read a csv file not only locally, but from a URL through read_csv or one can choose what columns needed to export so that we don’t have to edit the array later.

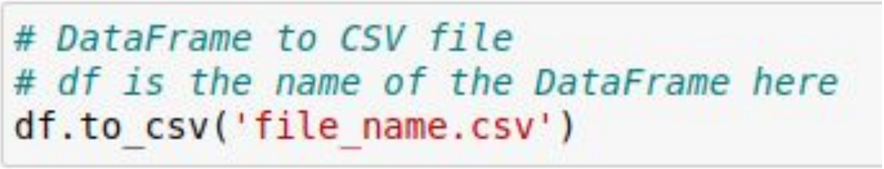
- These modifications can be done by the various arguments it takes.

- We don’t need to memorise all the arguments though, let’s have a look at few important ones in the next two slides.

Pandas to_csv

- The easiest way to write DataFrames to CSV files is using the Pandas to_csv function

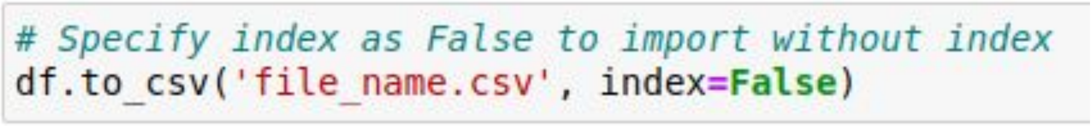
- Syntax:



```
# DataFrame to CSV file
# df is the name of the DataFrame here
df.to_csv('file_name.csv')
```

Where df is the name of your DataFrame

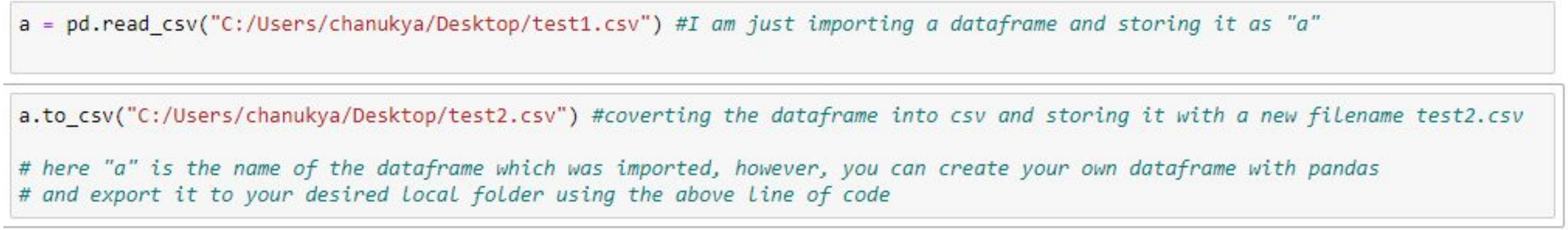
- If you want to export without the index, simply add index=False



```
# Specify index as False to import without index
df.to_csv('file_name.csv', index=False)
```

Pandas to_csv with example

Read the comments carefully



```
a = pd.read_csv("C:/Users/chanukya/Desktop/test1.csv") #I am just importing a dataframe and storing it as "a"
```

```
a.to_csv("C:/Users/chanukya/Desktop/test2.csv") #coverting the dataframe into csv and storing it with a new filename test2.csv

# here "a" is the name of the dataframe which was imported, however, you can create your own dataframe with pandas
# and export it to your desired local folder using the above line of code
```

Comprehensive Tutorial

- Must read-
  Here is an exhaustive tutorial on pandas with clear use-cases:

  https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/

DPhi

Notebook for Session

- We'll next have a look at a pre-recorded session on Pandas.
- Link to the Notebook used in the session: **https://dphi.tech/notebooks/852/manish_kc_06/day-1-notebook-introduction-to-pandas**
- **Make sure you attempt the exercises in the notebook.**

Slide Download Link

You can download these slides from the below link:

https://docs.google.com/presentation/d/1PcQQ9HCzc6y6nWCJ4UyGemQHKIlJVw6gv6YuchSxEdI/edit?usp=sharing

That’s it for this unit. Thank you!

Feel free to post any queries on Discuss.

DPhi