**Can Python supersede Ms Excel?**

**– A Data Scientist Perspective.**



I remember those days when I started my career with learning software like lotus 1-2-3. It had limited features but was good enough as compared to the time. Time moved on so the evolution on technologies and now we have Microsoft Excel, which has become an indispensable part of our day to day official life.

Ease of use and flexibility of different regular and analytic functions have made it so robust that we cannot think beyond it. In contrast, Python’s reputation is that it is more difficult to use whereas in reality it is the opposite and once we have mastered it perhaps we can do everything we can do in excel apart from its additional software development and analytical beauties. For example, we might have the boring task of copying certain data from one spreadsheet and pasting it into another one or we might have to go through thousands of rows and pick out just a handful of them to make small edits based on some criteria or we might have to look through hundreds of spreadsheets of department budgets, searching for any that are in the red. These are exactly the sort of boring, mindless spreadsheet tasks that Python can do for us.

Being human, we are not so open to changes and need very strong reasons, demonstrations to make one believe that the suggested change is really worthy. Hope this article creates that belief in you.

**There are several reasons to use Python over Excel:**

1. **We can perform nearly all regular tasks of Excels** (day to day mathematical functions, vlookup, conditional statements, pivot table etc.).

* **Here is vlookup using python (in 4 simple lines):**

df = pd.read\_excel(filename)

results = df.merge(filename, on='Name')

print (results)

* **A simple Pivot table showing us the sum of SepalWidth in values, SepalLength in Row Column and Name in Column Labels:**

pd.pivot\_table(df, values='SepalWidth', index = 'SepalLength', column = 'Name', aggfunc - np.sum)

* **Groupby (Subtotals in Excel):**

df[['Name','SepalLength']].groupby('name').sum()

GroupBy = df.groupby('Name').sum()

Group\_By.append(pd.DataFrame(df[['SepalLength', 'SepalWidth','PetalLength','PetalWidth']].sum()).T)

1. **Data cleanup is easier in Python:** Cleaning up data is way easier and time saving in Python as compared to Excel. For example, with a few lines of code we can detect and replace missing values in a data set irrespective of its size:

import pandas as pd

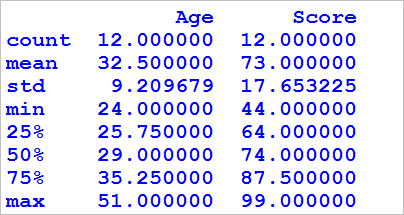
nba = pd.read\_csv("nba.csv")

nba["College"].fillna("No College", inplace = True)

1. **Descriptive Statistics:** Getting insights and generating descriptive statistics that summarize the central tendency, dispersion and shape of a dataset’s distribution, excluding NaN values of a dataset at the beginning is very important for a Data scientist. This can be done both in Excel and Python. In Python it is easier than it is in Excel. It can be done with a single line of code:

df.describe()

**Output:**



1. **Python code is easier to reproduce.** Commenting code in Python is very easy and thus code can be used to repurpose easily.Data manipulation and data analysis code can be saved as scripts and be reused many times with better version control. We can be sure that our code has executed and the output is correct and consistent. In this sense it is easier for other users to repeat our code.

# importing pandas module

import pandas as pd

# making data frame from csv file

test = pd.read\_csv("test.csv")

1. **Python is faster doing difficult calculations, generating and scheduling reports:** For difficult and analytical calculations, Python has been found to be working faster as compared to excel. For example, creating a multi-page report consisting of tables and charts is very time consuming in Excel where as with a little Python scripting we can have automatically pulled all the data from the various sources, compiled everything into pandas DataFrames, written the whole report out to Excel, and emailed it automatically to intended recipients. The best part, this could have been scheduled to run at a specific time and saved countless hours of work.
2. **Python is easier than vba/macros in Excel:** Syntaxes in Python are in colloquial English and thus are easier to be written and used.Also**,** Python is user-friendly, and both beginners and experienced analysts can use the language with ease. Excel uses [VBA](https://corporatefinanceinstitute.com/resources/excel/study/vba-dictionary/) language, which is a personalized platform that uses macros to automate tasks for data analysis. VBAs are complex to operate, and they make Excel difficult to work with when dealing with multiple operations during data analysis. The use of macros to automate tasks is more complex than the automation of tasks in the Python environment.
3. **Python works well with huge data**: It has been found that Excel tends to react slow while processing huge data and thus I recommend Python for such cases specially or I/O, data pipelines, automation, faster at calculating complex equations/algorithms. Python handles big data much more efficiently than Excel. If a business requires to crunch vast amounts of data in near real time, and if we are increasingly faced with semi-structured and unstructured data, we are better served with Python.
4. **Python is open source and has access to an enormous number of libraries:** Python has a huge number of libraries associated with it thus can perform nearly anything irrespective of its complexity. I can't think of a data related problem that I would not be able to solve in Python in conjunction with one of its libraries. Whether it's extracting data from Excel, cleansing data, performing calculations, visualizing data or utilizing various APIs - there is certainly a package out there.

In conclusion, we can say that while Python may be popular with the programming community, Excel is much more prevalent in the wider world. Most officer managers, sales people, marketers use Excel. It is indeed a great tool if we know how of it, and it has turned many non-technical people into expert analysts.



The answer to whether we should use Python or Excel is not an easy one to answer. But in the end, there is no either/or: Instead, we can use them together.

Excel is great for viewing data, performing basic analysis, and drawing simple graphs, but it really isn’t suitable for cleaning up data. If we have a large file (more than 500 Mb) with missing data, dates in different formats, no headers, it will take us ages to clean it by hand. The same can be said if our data is spread across a dozen CSV files, which is fairly common. Doing all these cleanup is trivial with Python and [Pandas](http://pandas.pydata.org/), a Python library for data analysis. Built on top of [Numpy](http://www.numpy.org/), Pandas makes high-level tasks easy, and we can write our results back to an Excel file, so we can continue to share the results of our analysis with non-programmers.

To be clear, I do not advocate leaving Excel behind – it is a powerful tool with many uses! But as an Excel user, being able to also leverage the power of Python can save us hours of time and open up career advancement opportunities.

It is worth remembering that these two tools can work well in tandem, and we may find that some tasks are best left in Excel, while others would benefit from the power, flexibility, and transparency that are offered by Python.

So, while Excel isn’t going away, Python is a great tool if we want clean data and perform higher-level data analysis.



Sr. Project Manager – Digital Publishing

Aspiring Data Scientist - INSAID