DJANGO UNCHAINED

By

Siddarth Naik

Sai Balaguru Ganess

V. Jahnavi Reddy

Swetha Vemparala

# INTRODUCTION:

The aim of this document is to Design a python code (application) which will able to open up 2 Excel files (let's say **excel\_file\_new** and **excel\_file\_old**) and compare them. If the values of the data are equal we print zero and if they are unequal we print the difference between the values which indicates that there is a change in the data. If there is exact same set of data, it is normal for new data to exist in the excel\_file\_new, making that data table larger. This application will also tell what unique IDs exist in the new file and not the other (and vise-versa). In order to do this, we have used pandas and openpyxl.

## PANDAS:

Pandas is an open source library that allows to you perform data manipulation in Python. Pandas library is built on top of Numpy, meaning Pandas needs Numpy to operate. Pandas provide an easy way to create, manipulate and wrangle the data. Pandas is also an elegant solution for time series data.

Data scientists use Pandas for its following advantages:

* Easily handles missing data
* It uses Series for one-dimensional data structure and Data Frame for multi-dimensional data structure.
* It provides an efficient way to slice the data.
* It provides a flexible way to merge, concatenate or reshape the data.
* It includes a powerful time series tool to work with.

In a nutshell, Pandas is a useful library in data analysis. It can be used to perform data manipulation and analysis. Pandas provide powerful and easy-to-use data structures, as well as the means to quickly perform operations on these structures.

## OPENPYXL:

## Openpyxl is a Python library to read/write Excel 2010 xlsx/xlsm/xltx/xltm files. It was born from lack of existing library to read/write natively from Python the Office Open XML format. All kudos to the PHP Excel team as openpyxl was initially based on PHP Excel. By default openpyxl does not guard against quadratic blowup or billion laughs xml attacks. To guard against these attacks install defusedxml.

# Source Code (Pandas):

import pandas as pd

Here we are importing pandas to carry out operations with excel.

df1= pd.read\_excel('Excel\_old.xlsx')

df2= pd.read\_excel('Excel\_new.xlsx')

We are reading both the excel files to carry out the operations.

df1['id']=df1['Unique ID']+df1['Data Point']

df2['Id']=df1['Unique ID']+df1['Data Point']

Combining the first two columns of the DataFrames to create a new column containing the unique id for each row.

def create\_dict(cid,c,df):

dict\_temp = {}

df\_temp = df

colid=cid

col=c

for row in range(0,df\_temp.shape[0]):

dict\_temp[df\_temp.iloc[row,colid]]=df\_temp.iloc[row,col]

return dict\_temp

Here we are writing a function create a dictionary of the contents of the DataFrames.

list\_old = []

for i in range(2,len(df1.columns)-1):

list\_old.append(create\_dict(6,i,df1))

Here we are creating a list of dictionaries for old excel file.

list\_new = []

for i in range(2,len(df2.columns)-1):

list\_new.append(create\_dict(6,i,df2))

Here we are creating a list of dictionaries for new excel file.

list\_diff = []

Define a list of dictionaries which contains Key as ID & difference as Values.

def diff\_dict():

for i in range(len(list\_new)):

temp = {}

for key in list\_new[i].keys():

if key not in list\_old[i].keys():

temp[key] = list\_new[i][key]

else:

temp[key] = list\_new[i][key] - list\_old[i][key]

list\_diff.append(temp)

This is the list creation method

df\_name1=pd.DataFrame({'Unique ID':df2['Unique ID'],'Data Point':df2['Data Point']})

def create\_df():

col\_name = df2.columns[2:len(df2.columns)-1]

for i in range(len(list\_diff)):

df\_name1[str(col\_name[i])] = list\_diff[i].values()

Here we are creating new DataFrame for the result.

create\_df()

Here we are calling the function to calculate the difference and put in another DataFrame.

def color\_red(value):

color = 'red' if (value!=0 and type(value)!=str) else 'black'

return 'color: %s' %color

df=df\_name1.style.applymap(color\_red)

Here we are forming the new DataFrame to highlight the changed values in red.

df.to\_excel("Final.xlsx",index =False)

Exporting the final DataFrame to a new Excel File.

# Source Code (openpyxl):

Importing “Load\_workbook” & “Workbook” from “openpyxl” to operate on Excel sheets.

Importing “Fill, Font, Color, colors” from “openpyxl” to style the Excel sheet

from openpyxl import load\_workbook

from openpyxl import Workbook

from openpyxl.styles import Fill,Font,Color,colors

“wb\_final” is the object for the final Excel sheet. “wb\_old”, “wb\_new” are for the existing Excel sheets, on which operation has to be carried out.

wb\_final= Workbook()

wb\_old=load\_workbook('Excel\_old.xlsx')

wb\_new=load\_workbook('Excel\_new.xlsx')

Reading the sheets from the workbook.

sh1= wb\_old.active

sh2= wb\_new.active

sh3= wb\_final.active

Assigning the Column names for the final Excel Sheet – “Excel\_final.xlsx”

for i in range(1,sh2.max\_column+1):

sh3.cell(1,i).value=sh1.cell(1,i).value

wb\_final.save('Excel\_final.xlsx')

Differentiating the cell content between String & Numbers using a “if clause”

For String, values are assigned directly

for i in range(2, sh2.max\_row+1):

for j in range(1, sh2.max\_column+1):

if type(sh1.cell(row=i,column=j).value)==str and type(sh2.cell(row=i,column=j).value)==str :

sh3.cell(row=i, column=j).value = sh2.cell(row=i,column=j).value

For Numbers, Calculate the difference between the respective cells of two Excel Sheets

else:

sh3.cell(row=i,column=j).value = sh1.cell(row=i,column=j).value-sh2.cell(row=i,column=j).value

For non-zero values in “Excel\_final.xlsx”, style the value in Red color.

if sh3.cell(row=i,column=j).value!=0:

sh3.cell(row=i, column=j).font = Font(color=colors.RED)

wb\_final.save('Excel\_final.xlsx')

# Screenshots of Excel Sheets:

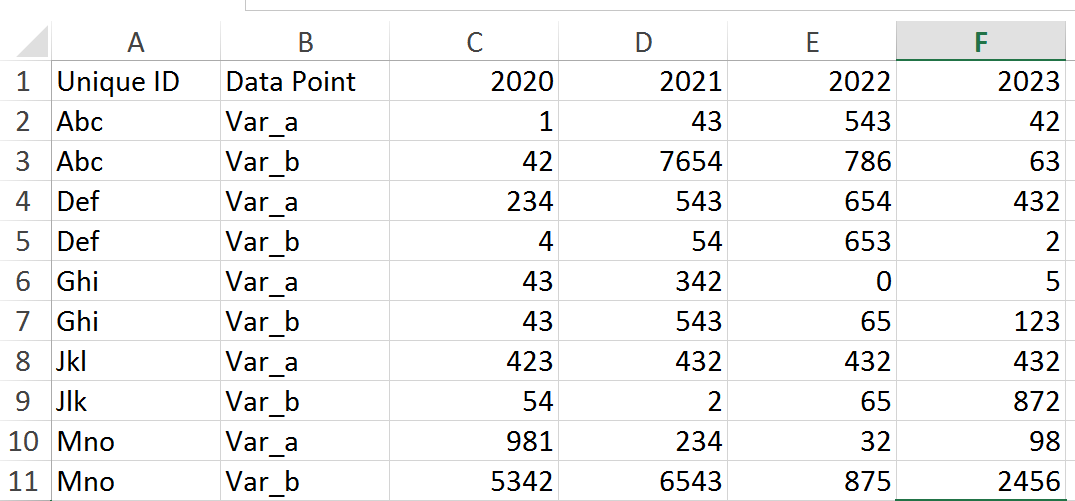


Table 1 : Excel\_old

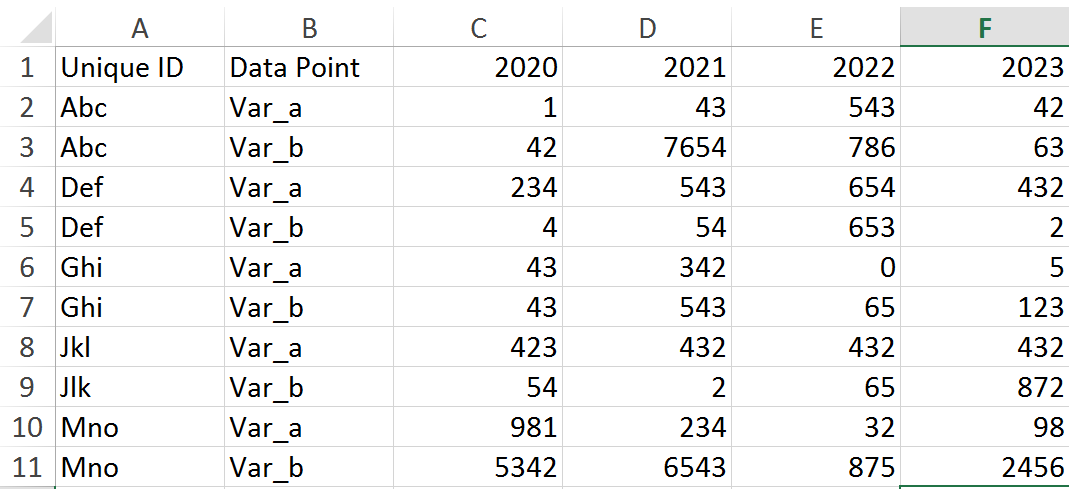
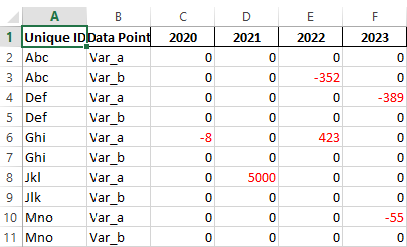


Table 2 : Excel\_new



# Conclusion:

Using pandas and openpyxl packages of python, the given task has been successfully executed.