# All Codes

```
'''
File: data_concatenation.py
Project: Data concatenation
File Created: Wednesday, 16th November 2022 3:15:53 am
Author: Raymond Yan Jin (yanjinn@connect.hku.hk)
-----
Last Modified: Wednesday, 16th November 2022 6:25:57 am
Modified By: Raymond Yan Jin (yanjinn@connect.hku.hk>)
-----
Copyright 2022 - 2022 Business School, The University of Hong Kong
'''

import numpy as np
import pandas as pd

# Read the data
df1 = pd.read_csv('data/2014.csv')
df2 = pd.read_csv('data/2015.csv')

# data exploration
df1.head()
df2.head()
df1.shape   #(54492, 58)
df2.shape   #(39706, 58)

# check the columns
if df1.columns.tolist() == df2.columns.tolist():
    print('The columns are the same')

# concatenate the data
df = pd.concat([df1, df2], axis=0, ignore_index=True)

# check the shape
df.head()
df.shape   #(94198, 58)

# store the data into the csv file
df.to_csv('data/output.csv', index=False)
```

```
'''
File: transformation.ipynb
Project: Data transformation
File Created: Wednesday, 26th October 2022 2:43:08 pm (UTC+8:00)
Author: Raymond Yan Jin (yanjinn@connect.hku.hk)
-----
Last Modified: Monday, 21st November 2022 10:06:35 am (UTC+8:00)
Modified By: Raymond Yan Jin (yanjinn@connect.hku.hk>)
-----
Copyright 2022 - 2022 Business School, The University of Hong Kong
```

```
'''

import numpy as np
import pandas as pd
import os

# Read the data
deal_data = pd.read_csv('data/deal_level_data.csv')
quarter_data = pd.read_csv('data/quarter_level_data.csv')

# Data exploration
deal_data.head()
quarter_data.head()
deal_data.shape # (3005,1467)
quarter_data.shape  # (75125,42)
deal_data.columns.to_list()
quarter_data.columns.to_list()
# 75125/3005 = 25 so the logic is to divide the deal data according to the
quarters

# too slow, takes 20mins on intel i5-1250OH
# even though I could improve the efficiency by using multiprocessing, I don't
think it's technically meaningful.
for i in range(quarter_data.shape[0]):
    deal = quarter_data.iloc[i,:]
    acquirer = deal['Acquirer_CUSIP']
    target = deal['Target_CUSIP']
    year = deal['Year_Announced']
    quarter_to_event = deal['quarter_to_the_event_date']

    if quarter_to_event < 0:
        for j in range(16,42):
            quarter_data.iloc[i,j] = deal_data.loc[(deal_data['Acquirer_CUSIP']
== acquirer)\
                                                   & (deal_data['Target_CUSIP'] ==
target)\
                                                   & (deal_data['Year_Announced']
== year),\

quarter_data.columns[j]+'__'+str(-quarter_to_event)].values[0]
    elif quarter_to_event > 0:
        for j in range(16,42):
            quarter_data.iloc[i,j] = deal_data.loc[(deal_data['Acquirer_CUSIP']
== acquirer)\
                                                   & (deal_data['Target_CUSIP'] ==
target)\
                                                   & (deal_data['Year_Announced']
== year),\

quarter_data.columns[j]+'_'+str(quarter_to_event)].values[0]
    else:
        for j in range(16,42):
            quarter_data.iloc[i,j] = deal_data.loc[(deal_data['Acquirer_CUSIP']
== acquirer)\
```

```
                                                    & (deal_data['Target_CUSIP'] ==
target)\
                                                    & (deal_data['Year_Announced']
== year),\

quarter_data.columns[j]].values[0]

quarter_data.to_csv('data/quarter_level_data_output.csv',index=False)
```

```
'''
File: datetime.ipynb
Project: Datetime
File Created: Wednesday, 16th November 2022 3:15:53 am
Author: Raymond Yan Jin (yanjinn@connect.hku.hk)
-----
Last Modified: Wednesday, 16th November 2022 6:25:19 am
Modified By: Raymond Yan Jin (yanjinn@connect.hku.hk>)
-----
Copyright 2022 - 2022 Business School, The University of Hong Kong
'''


import numpy as np
import pandas as pd
import datetime as dt

# read the data
data = pd.read_csv(r'./data/ukpound_exchange.csv')

# data exploration
data.shape # (8429, 4)
data.head()

# select the row of date as the end of the month
data['Date'] = pd.to_datetime(data['Date'])
output = data[data['Date'].dt.is_month_end]

# check the output data
output.head()
output.shape # (276, 4)

# write the output data to csv
output.to_csv(r'./data/output.csv', index = False)
```

```
'''
File: multiprocessing.ipynb
Project: Fuzzy and multiprocessing
File Created: Wednesday, 16th November 2022 3:15:53 am
Author: Raymond Yan Jin (yanjinn@connect.hku.hk)
-----
Last Modified: Wednesday, 16th November 2022 6:24:17 am
Modified By: Raymond Yan Jin (yanjinn@connect.hku.hk>)
-----
```

```
Copyright 2022 - 2022 Business School, The University of Hong Kong
'''

import pandas as pd
from multiprocessing import Pool
#from multiprocessing import Manager
from fuzzywuzzy import fuzz
import numpy as np
import csv

# read the data
acquirer_data = pd.read_excel("./data/acquirers.xlsx")
bank_data = pd.read_csv("./data/bank_names.csv")
output = pd.DataFrame(columns=["acquirer", "match1", "match2", "match3",
"match4", "match5"])
output["acquirer"] = acquirer_data["Acquirer Name"]

bank_list = bank_data['bank_names'].tolist()


# fuzzy match
def fuzzy_match(index):

    match_list = []
    firm_name = acquirer_data.loc[index, 'Acquirer Name']
    for bank_name in bank_list:
        similarity = fuzz.ratio(firm_name, bank_name) + \
                     fuzz.partial_ratio(firm_name, bank_name) + \
                     fuzz.token_sort_ratio(firm_name, bank_name) + \
                     fuzz.token_set_ratio(firm_name, bank_name)
        match_list.append((similarity, bank_name))

    match_list.sort(key=lambda x: x[0], reverse=True)
    match = [element[1] for element in match_list[:5]]
    # append the result to a csv file
    with open("./data/output.csv", "a", newline="") as f:
        writer = csv.writer(f)
        writer.writerow([index] + [firm_name] + match)
        f.close()

# parallelize the fuzzy match
if __name__ == '__main__':
    global output
    pool = Pool(4)
    pool.map(fuzzy_match, range(acquirer_data.shape[0]))

# sort the result
output = pd.read_csv("./data/output.csv", header=None)
output.columns = ["index", "acquirer", "match1", "match2", "match3", "match4",
"match5"]
output = output.sort_values(by="index")
```

```
'''
```

```
File: geolocation.ipynb
Project: Google API geolocation
File Created: Wednesday, 16th November 2022 3:15:53 am
Author: Raymond Yan Jin (yanjinn@connect.hku.hk)
-----
Last Modified: Wednesday, 16th November 2022 6:23:37 am
Modified By: Raymond Yan Jin (yanjinn@connect.hku.hk>)
-----
Copyright 2022 - 2022 Business School, The University of Hong Kong
'''

from vincenty import vincenty
import googlemaps
from datetime import datetime
import pandas as pd
import time
import numpy as np

comp_data = pd.read_excel("./data/coname_addresses.xlsx")
gmaps = googlemaps.Client(key='AIzaSyBP9a_wlCkhsKDNFtdtBNOjplpLLgJAK88')
# geocode_result = gmaps.geocode('1600 Amphitheatre Parkway, Mountain View, CA')
WhiteHouse = (38.8976763, -77.0365298)

output = pd.DataFrame()
# initialize the output dataframe
output['index'] = comp_data.index
output['CONAME'] = comp_data['CONAME']
output['address'] = comp_data['address']
output['distance'] = 0
output['lat'] = 0
output['lng'] = 0


for i in range(comp_data.shape[0]):
    #sleep for 1 sec
    if output['distance'][i] == 0:

        data_comp = comp_data.iloc[i]
        coname = data_comp['CONAME']
        address = data_comp['address']
        geocode_result = gmaps.geocode(address)
        try:
            lat = geocode_result[0]['geometry']['location']['lat']
            lng = geocode_result[0]['geometry']['location']['lng']
            distance = vincenty((lat, lng), WhiteHouse)

            output['distance'][i] = distance
            output['lat'][i] = lat
            output['lng'][i] = lng
        except:
            print("Error: ", i, coname, address)
            pass
        output.to_excel("temp.xlsx", index=False)
        time.sleep(np.random.uniform(2, 4))
```

```
# count the number of zero distance
geocode_result = gmaps.geocode(comp_data.iloc[593]['address'])

# export to excel
output.to_excel("output.xlsx", index=False)
```

```
'''
File: scholar.ipynb
Project: webscrap
File Created: Monday, 21st November 2022 3:02:42 pm (UTC+8:00)
Author: Raymond Yan Jin (yanjinn@connect.hku.hk)
-----
Last Modified: Monday, 21st November 2022 7:42:11 pm (UTC+8:00)
Modified By: Raymond Yan Jin (yanjinn@connect.hku.hk>)
-----
Copyright 2022 - 2022 Business School, The University of Hong Kong
'''


import pandas as pd
import numpy as np
import selenium
from selenium import webdriver
import time


browser = webdriver.Chrome(r"C:\Program Files\Development\chromedriver.exe")

vol = input("Enter the volume number(can be list seperated by comma, e.g
140;141): ")
issue = input("Enter the issue number(can be list seperated by comma, e.g 1;2):
")

vol = vol.split(";")
issue = issue.split(";")


for v in vol:
    for i in issue:
        url = "https://www.sciencedirect.com/journal/journal-of-financial-
economics/vol/{}/issue/{}".format(v,i)
        browser.get(url)
        #time.sleep(5)
        num = len(browser.find_elements("class name", "js-article-title"))
        articles = []
        articles_url = []
        authors = []
        for j in range(num):
            articles.append(browser.find_elements("class name", "js-article-
title")[j].text)
            # get the link of each article
            articles_url.append(browser.find_elements("class name", "article-
content-title")[j].get_attribute("href"))
            # get the author of each article
```

```
            authors.append(browser.find_elements("class name", "js-article-
author-list")[j].text)
            #drop the empty url
            articles_url = [x for x in articles_url if x != '']
        scholar = pd.DataFrame({"Article":articles, "Author":authors,
"URL":articles_url})

scholar.to_csv("scholar.csv", index = False)
```

```
'''
File: SFC.ipynb
Project: webscrap
File Created: Thursday, 24th November 2022 6:23:06 pm (UTC+8:00)
Author: Raymond Yan Jin (yanjinn@connect.hku.hk)
-----
Last Modified: Friday, 25th November 2022 5:01:44 pm (UTC+8:00)
Modified By: Raymond Yan Jin (yanjinn@connect.hku.hk>)
-----
Copyright 2022 - 2022 Business School, The University of Hong Kong
'''



import pandas as pd
import numpy as np
import selenium
from selenium import webdriver
import time
import math

browser = webdriver.Chrome(r"C:\Program Files\Development\chromedriver.exe")

url = "https://apps.sfc.hk/productlistWeb/searchProduct/UTMF.do"
browser.get(url)

fund_list = pd.DataFrame(columns = ['Product_Name', 'Sub_Fund_Name', 'Issuer',
'Auth_Date', 'Doc', 'Deriv_Fund'])

for i in range(2,2087):
    Product_Name = browser.find_elements("xpath",
"/html/body/div[3]/div/div[2]/form[2]/div[3]/div/table/tbody/tr[{}]/td[1]".forma
t(i))[0].text
    Sub_Fund_Name = browser.find_elements("xpath",
"/html/body/div[3]/div/div[2]/form[2]/div[3]/div/table/tbody/tr[{}]/td[2]".forma
t(i))[0].text
    Issuer = browser.find_elements("xpath",
"/html/body/div[3]/div/div[2]/form[2]/div[3]/div/table/tbody/tr[{}]/td[3]".forma
t(i))[0].text
    Auth_Date = browser.find_elements("xpath",
"/html/body/div[3]/div/div[2]/form[2]/div[3]/div/table/tbody/tr[{}]/td[4]".forma
t(i))[0].text
    Doc = browser.find_elements("xpath",
"/html/body/div[3]/div/div[2]/form[2]/div[3]/div/table/tbody/tr[{}]/td[5]".forma
t(i))[0].text
```

```
    Deriv_Fund = browser.find_elements("xpath",
"/html/body/div[3]/div/div[2]/form[2]/div[3]/div/table/tbody/tr[{}]/td[6]".forma
t(i))[0].text
    #fund_list = fund_list.append({'Product_Name': Product_Name, 'Sub_Fund_Name':
Sub_Fund_Name, 'Issuer': Issuer, 'Auth_Date': Auth_Date, 'Doc': Doc,
'Deriv_Fund': Deriv_Fund}, ignore_index=True)
    fund_list = pd.concat([fund_list, pd.DataFrame({'Product_Name':
Product_Name, 'Sub_Fund_Name': Sub_Fund_Name, 'Issuer': Issuer, 'Auth_Date':
Auth_Date, 'Doc': Doc, 'Deriv_Fund': Deriv_Fund}, index=[0])],
ignore_index=True)

fund_list.to_csv("data/temp_fund_list.csv", index=False)

fund_list = pd.read_csv('data/temp_fund_list.csv')
fund_list['Product_ID'] = fund_list['Product_Name'].apply(lambda x: x.split(' ')
[-1].strip('()'))
fund_list['Sub_Fund_ID'] = fund_list['Sub_Fund_Name'].apply(lambda x: x.split('
')[-1].strip('()') if type(x) == str else np.nan)
#fund_list['doc_url'] =
"https://apps.sfc.hk/productlistWeb/searchProduct/getDocListNoDate.do?
lang=EN&ceref={}&docType=OD".format(fund_list['Sub_Fund_ID'])
for i in range(fund_list.shape[0]):
    if fund_list.loc[i, 'Doc'] != ' ':
        fund_list.loc[i, 'doc_url'] =
"https://apps.sfc.hk/productlistWeb/searchProduct/getDocListNoDate.do?
lang=EN&ceref={}&docType=OD".format(fund_list.loc[i, 'Sub_Fund_ID'])
fund_list.to_csv('data/fund_list.csv', index=False)
# you need to manually change the download dir! Sorry for the inconvenience!
options = webdriver.ChromeOptions()
prefs = {"profile.default_content_settings.popups": 0,
        "download.default_directory": r"C:\Projects\FundsData\data\Key_Stats\\",
# IMPORTANT - ENDING SLASH V IMPORTANT
        "directory_upgrade": True}
options.add_experimental_option("prefs",prefs)
browser = webdriver.Chrome(executable_path=r"C:\Program
Files\Development\chromedriver.exe", options=options)
fund_list = pd.read_csv('data/fund_list.csv')
# in my project this should be range(2086), for your time and convenience, I have
set it to 10
for i in range(15):
    if type(fund_list.iloc[i]['doc_url']) != float:
        url = fund_list.iloc[i]['doc_url']
        browser.get(url)
        try:

 browser.find_element_by_xpath('/html/body/div/div/div[3]/div/div/div/div[2]/tab
le/tbody/tr[3]/td/a').click()
        except:
            pass
```