```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt


#Read the data into the machine
friedman=pd.read_excel('/Users/jacobmcgraw/Downloads/data_friedman.xls',sheet_name='data')


#subset the data into only the observations from the year 2006
year_2006_friedman=friedman[friedman.year==2006]


#rename the columns in the dataset ot make calculations easier
year_2006_friedman=year_2006_friedman.rename(columns={'Inflation, Consumer
Prices':'inflation',\
                        'Number of Procedures to Start a Business':'noptsab'})


#Subset the data to only the observations that do not have empty inflation values
real_friedman_data=year_2006_friedman[~year_2006_friedman.inflation.isna()]


#A list of thje countries in our dataset
my_countries=real_friedman_data.contcode.values


#A series of lists that will serve as the columns for our final dataframe,
#with the averages of the mean tarrif, the mean years of school, mean democracy,
#mean life expectency, mean life expectency, mean buisness procedures,
#and mean infant mortality rate.
mean_tarrif=[]
mean_years_of_school=[]
mean_democracy=[]
mean_life_expectency=[]
mean_buisness_procedures=[]
mean_infant_mortality=[]


#adds the data to the lists
for i in real_friedman_data.contcode.values:

    mean_tarrif.append(np.nanmean(friedman\
                    [friedman.contcode==i].rename(columns={\
                    'Mean Tariff Rate':'mtr'}).mtr.values))
```

```
    mean_years_of_school.append(np.nanmean\
            (friedman[friedman.contcode==i].rename(columns={\
            'Years of Schooling':'years_of_school'}).years_of_school.values))

    mean_democracy.append(np.nanmean(friedman[friedman.contcode==i\
                                    ].Democracy.values))

    mean_life_expectency.append(np.nanmean(friedman[friedman.contcode==i\
                        ].rename(columns={'Life Expectancy':'life'}).life.values))


mean_buisness_procedures.append(np.nanmean(friedman[friedman.contcode==i].rename(colum
ns={\
                        'Number of Procedures to Start a Business':\
                        'buisness_procedures'}).buisness_procedures.values))


mean_infant_mortality.append(np.nanmean(friedman[friedman.contcode==i].rename(columns={
\
                        'Morality Rate of Infants (per 1000 births)':\
                        'buisness_procedures'}).buisness_procedures.values))


#a list of developed countries, according to the International Monetary Fund
developing=['Albania', 'Algeria', 'Angola','Argentina','Armenia','Bangladesh',\
          'Belarus','Benin','Bolivia','Botswana','Brazil','Bulgaria','Burkina Faso',\
           'Burundi','Cambodia','China','Colombia',"Cote d'Ivoire", 'Croatia',\
           'Dominican Republic','Ecuador','Egypt', 'El Salvador', 'Fiji',\
            'Ghana', 'Guatemala','Guinea-Bissau', 'Guyana', 'Haiti', 'Honduras', 'China',\
          'Hungary', 'India', 'Indonesia','Jamaica', 'Jordan', 'Kazakhstan', 'Kenya',\
           'Latvia', 'Lithuania', 'Madagascar', 'Malawi', 'Malaysia','Mauritania',\
           'Mauritius', 'Mexico', 'Moldova', 'Morocco','Namibia', 'Niger', 'Nigeria'\
          , 'Pakistan', 'Paraguay', 'Peru', 'Philippines', 'Poland', 'Romania', \
           'Russian Federation', 'Samoa', 'Senegal', 'South Africa', 'Sri Lanka',\
           'St. Lucia', 'Tanzania', 'Thailand', 'Togo', 'Tunisia', 'Turkey', 'Uganda',\
      'Ukraine','Uruguay','Venezuela','Egypt, Arab Rep.', 'Costa Rica', 'Mali', 'Korea, Rep.',\
          'Czech Republic', 'Estonia', 'Slovenia','Venezuela, RB']


#creates a list that will serve as our identifying column to tell wheather a
#country is developed or developing
my_developing_column=[]


#Adds the values to the list
```

```
for i in my_countries:
    if i in developing:
        my_developing_column.append('Developing')
    else:
        my_developing_column.append('Developed')


#Drops columns that are of no use to this particular analysis
my_real_friedman_data=real_friedman_data.drop(columns=['year','wbregion','Morality Rate of
Infants (per 1000 births)',\
                    'Life Expectancy', 'Years of Schooling','Democracy','Top Marginal Income Tax
Rate'\
                     ,'Mean Tariff Rate','Official vs. Black Market Exchange Rate '\
                     ,'Unnamed: 14','Unnamed: 15','Unnamed: 16'])


#Renames the columns to be readable to the user
my_real_friedman_data=my_real_friedman_data.rename(columns={'contcode':'Country','noptsa
b'\
                  :'Number of Procedures to Start a Buisness','inflation':'Inflation Rate'})


#Resets the index to normal
my_real_friedman_data=my_real_friedman_data.reset_index()


#Drops the index column
my_real_friedman_data=my_real_friedman_data.drop(columns=['index'])


#adds the average data values to the real friedman dataset
my_real_friedman_data['Average Highest Tax Rate']=mean_tarrif
my_real_friedman_data['Average Years Of School']=mean_years_of_school
my_real_friedman_data['Average Democracy']=mean_democracy
my_real_friedman_data['Average Life Expectency']=mean_life_expectency
my_real_friedman_data['Average Number of Procedures to Start a
Buisness']=mean_buisness_procedures
my_real_friedman_data['Average Infant Mortality Rate']=mean_infant_mortality
my_real_friedman_data['Developing Status']=my_developing_column


#Drops a non-needed line from the code
my_real_friedman_data=my_real_friedman_data.drop(columns=['Average Years Of School'])


#a list of countries that have missing data and therefore should be removed from the
```

```
#final table
bad_observations=["St. Lucia", 'Slovak Republic', 'Samoa',\
                  'Moldova','Mauritania','Macao, China',\
                  'Luxembourg','Lao PDR','Kyrgyz Republic',\
                  'Kazakhstan', 'Hong Kong, China', "Cote d'Ivoire",\
                  'Burkina Faso', 'Belarus', 'Angola', 'Iceland', 'Cambodia']


#This line of code Sets the index to 'Country' in order to
#delete the bad observations later
my_real_friedman_data=my_real_friedman_data.set_index('Country')


#This code drops the bad observartions and then resets the index
my_real_friedman_data=my_real_friedman_data.drop(index=bad_observations).reset_index()


#get rid of the comma as to avoid possible errors when reading the data into
#various sources (such as R, Python, JMP, Excel, or other such software)
my_real_countries=[i.replace(',','') for i in my_real_friedman_data.Country.values]


#This line of code Overwrites the Countries column with
#the corrected list
my_real_friedman_data['Country']=my_real_countries


#This line of code saves the txt file to the users machine, should they want to
#my_real_friedman_data.to_csv('real_friedman_data.txt', header=True, index=False, sep='\t',
mode='a')
```