basic_plot_1

February 2, 2023

[ ]:
```
#import libraries
%matplotlib notebook
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[ ]:
```
# read data file
df_can = pd.read_excel ('Canada.xlsx', sheet_name = 'Canada by Citizenship',␣
 ↪skiprows = range(20), skipfooter = 2)
```

[ ]:
```
# get the head of the dataframe
df_can.head()
```

[ ]:
```
# get the tail of the dataframe
df_can.tail()
```

[ ]:
```
# get info of the dataframe
df_can.info(verbose = False)
```

[ ]:
```
# get list of column headers
df_can.columns
```

[ ]:
```
# get list of indices
df_can.index
```

[ ]:
```
# print datatype of columns and indices
print(type(df_can.columns))
print(type(df_can.index))
```

[ ]:
```
# to get the columns as lists
df_can.columns.tolist()
```

[ ]:
```
# get the indices as list
df_can.index.tolist()
```

[ ]:
```
# size of dataframe (row, col)
df_can.shape
```

Datatypes in Pandas: float, int, bool, datetime64[ns], datetime64[ns, tz], timedelta[ns], category, and object (string). These datatypes have sizes, e.g. int64 and int32.

Let us clean the data set to remove a few unnecessary columns. We can use pandas drop() method as follows:

[ ]:
```
# in pandas axis=0 is rows and axis=1 is columns
df_can.drop(['AREA', 'REG', 'DEV', 'Type', 'Coverage'], axis = 1, inplace =␣
 ↪True)
df_can.head(3)
```

Let us rename the columns by using the rename() method and passing the 'oldname:newname' as a dictionary

[ ]:
```
df_can.rename(columns={'OdName':'Country', 'AreaName':'Continent', 'RegName':
 ↪'Region'}, inplace=True)
df_can.columns
```

[ ]:
```
# add a column at the end giving the total number of immigrants
df_can['Total'] = df_can.sum (axis = 1, numeric_only = True)
df_can.head(2)
```

[ ]:
```
# check numbr of null objects
df_can.isnull().sum()
```

[ ]:
```
# get summary of each column
df_can.describe()
```

Indexing and Slicing Select Columns * df.column_name * df.['column'] * df.['col1', 'col2', 'col3']

[ ]:
```
df_can['Country']
```

[ ]:
```
df_can[['Country', 1981, 1991, 2001]]
```

Select Rows * df.loc[label] * df.iloc[index]

[ ]:
```
# setindex to be the name of the Country
df_can.set_index('Country', inplace = True)
```

[ ]:
```
df_can.head(3)
```

[ ]:
```
df_can.loc['Japan']
```

[ ]:
```
df_can.iloc[87]
```

[ ]:
```
df_can.loc['Japan', 2013]
```

[ ]:
```
df_can.iloc[87, 36]
```

[ ]:
```
df_can.loc['Japan', [1981, 1991, 2001, 2011]]
```

[ ]:
```
# convert column names into strings
df_can.columns = list(map(str, df_can.columns))
```

Filtering Based on a Criteria

[ ]:
```
# create a condition
cond = (df_can['Continent'] == 'Asia')
print (cond)
```

[ ]:
```
# multiple conditions
# boolean operators: ~ (not), & (and), | (or)
df_can[(df_can['Continent'] == 'Asia') & (df_can['Region'] == 'Southern Asia')]
```

Plotting with MatPlotLib

[ ]:
```
# extract data for Haiti
years = list(map(str, range(1981, 2014)))
haiti = df_can.loc['Haiti', years]
haiti.head()
```

[ ]:
```
# change index value of Haiti to integer for plotting
haiti.index = haiti.index.map(int)

haiti.plot(kind = 'line')
plt.title ('Immigration from Haiti')
plt.xlabel ('Years')
plt.ylabel ('Number of Immigrants')
plt.text (2000, 6000, '2010 Earthquake')

plt.show()
```

[ ]:
```
# get the dataframe for China and India
df_ci = df_can.loc [['China', 'India'], years]
df_ci.head()
```

[ ]:
```
# plot the graph
df_ci.plot (kind = 'line')
```

[ ]:
```
df_ci = df_ci.transpose()
df_ci.head()
```

[ ]:
```
df_ci.index = df_ci.index.map(int)
df_ci.plot(kind = 'line')
plt.title ('Immigrants from China and India')
plt.xlabel ('Years')
plt.ylabel ('Number of Immigrants')
plt.show()
```

Compare the trend of top 5 countries that contributed the most to immigration to Canada

[ ]:
```
# sort the dataframe by Total in descending order
df_can.sort_values (by = 'Total', ascending = False, axis = 0, inplace = True)
```

[ ]:
```
# get the top 5 entries
df_top5 = df_can.head(5)
```

[ ]:
```
# transpose the dataframe
df_top5 = df_top5[years].transpose()
print (df_top5)
```

[ ]:
```
df_top5.index = df_top5.index.map(int)
df_top5.plot (kind = 'line', figsize = (14,8))
plt.title ('Immigration Trend of Top 5 Countries')
plt.xlabel ('Years')
plt.ylabel ('Number of Immigrants')
plt.show()
```