Data Bootcamp: Code Practice #3

Revised: November 3, 2017

Answer each of the questions below. We recommend code with comments. Please submit in hardcopy.

The data input parts of each question are included in this template:

https://raw.githubusercontent.com/NYUDataBootcamp/Materials/master/Code/Python/bootcamp_practice_3_template.py

Or just navigate from the GitHub page:
Code ⇒ Python ⇒ `bootcamp_practice_3_template.py` ⇒ click on Raw utton

1. Enter and run this code in Spyder to produce the dataframe `weo`:

```
import pandas as pd
data = {'BRA': [13.37, 13.30, 14.34, 15.07, 15.46, 15.98, 16.10],
        'JPN': [33.43, 31.83, 33.71, 34.29, 35.60, 36.79, 37.39],
        'USA': [48.30, 46.91, 48.31, 49.72, 51.41, 52.94, 54.60],
        'Year': [2008, 2009, 2010, 2011, 2012, 2013, 2014]}
weo  = pd.DataFrame(data)
```

   The numbers are GDP per person in thousands of US dollars, 2008 to 2014, variable PPPPC in the IMF's *World Economic Outlook* database.

   a. Explain the `import` statement.

   b. What type of object is `data`?

   c. Why does the last line have `pd` prior to the `DataFrame` function?

   d. What type of object is `weo`?

   e. How many rows does it have? Columns?

   f. What `dtypes` are the variables/columns? What does this mean?

   g. *Challenging.* What are each of these expressions? What type?
   ```
   weo['Year']
   weo[['Year']]
   weo[[3]]
   ```

   h. *Challenging.* Find and apply a method to convert `weo['Year']` to type `float`. *Hint:* The method begins with the letter `a`.

   i. Describe the result of the statement `t = weo.tail(3)`. What kind of object is `t`? What does it look like?

   j. How would you create a new dataframe that consists of the first 4 rows of `weo`?

   k. What type of object is `weo['BRA']`?

l. Create a new variable equal to the ratio of Brazil's GDP per capita to Japan's and add it to the ataFrame.

m. *Challenging.* Use the `drop()` method to eliminate this (new) variable from the dataframe.

n. What are `weo`'s row and column labels?

o. Set the index equal to the `Year` variable.

p. Change the names of the other variables to Brazil, Japan, and United States.

q. Export the dataframe to an Excel spreadsheet.

r. What method would you use to compute the mean for each country? What are the means?

s. *Challenging.* How would you compute means across countries for each year?

t. Plot the data by applying a `plot` method to `weo`.

u. *Challenging.* Change the colors of the lines to green (Brazil), red (Japan), and blue (US).

v. *Challenging.* Do the same plot with a log scale. *Hint:* Read the documentation for the plot method.

w. Plot Brazil on its own.

Solution:

a. The `import` statement adds the `pandas` data management package.

b. Dictionary — note the curly braces.

c. The `pd.` identifies `DataFrame` as a pandas function.

d. Dataframe.

e. `weo.shape` tells us that `weo` has 7 rows and 4 columns.

f. `weo.dtypes` generates the output

```
BRA     float64
JPN     float64
USA     float64
Year      int64
```

So the first three variables are floats, the last one is an integer.

g. Apply the type function. The first one is a series, the last two are dataframes. The difference here is the input. If the input is a string, it returns a variable. If the input is a list, it returns a dataframe. The content is similar, but they're (slightly) different objects. This isn't particularly important, other than to remind ourselves that the details matter here.

h. We use: `weo['Year'] = weo['Year'].astype(float)`. The right side changes the type to float, then we assign it back to `weo['Year']`.

i. `t = weo.tail(3)` is a dataframe consisting of the last three rows of `weo`.

j. `weo.head(4)`.

k. A series, a single variable (column).

l. Use the statement: `weo['ratio'] = weo['BRA']/weo['JPN']`.

m. Use: `weo = weo.drop(['ratio'], axis=1)`.

n. We get (and print) the row and column labels with

```
print('Row labels (index):', weo.index)
print('Column labels:', weo.columns)
```

o. `weo = weo.set_index(['Year'])`.

p.
```
weo = weo.rename(columns={'BRA': 'Brazil', 'JPN': 'Japan',
                          'USA': 'United States'})
```

q. `weo.to_excel('weo.xls')`

r. `print('Country means:', weo.mean(), sep='')`

s. `print('Year means:', weo.mean(axis=1), sep='')`

t. `weo.plot()` (The year labels are messed up here, we'll have to fix that/)

u. `weo.plot(color=['green', 'red', 'blue'])`

v. There's an ambiguity here over whether to use logs on the x axis, the y axis, or both. This does the y xis:
```
weo.plot(color=['green', 'red', 'blue'], logy=True)
```

w. `weo['Brazil'].plot()`

2. Use `read_csv()` to read the responses of a previous semesters entry poll from

https://raw.githubusercontent.com/NYUDataBootcamp/Materials/master/Data/entry_poll_fall16.csv

a. Read the file and assign it to the variable ep.

b. Describe its contents. What are the variables? The responses?

c. What data types are the variables?

d. Change the variable names to something shorter.

e. *Challenging.* Describe what this code does:
```
ep[list(ep)[1]].value_counts()
```
Suggestion: Break it into two or more statements and explain them one at a time.

Solution:

a. `ep = pd.read_csv(url)`

b. You can see them at the source. The file is a "dump" of questions (the variable names) and answers (the data). It's somewhat verbose, which makes it hard to print out and look at.

c. The dtypes are all objects: strings.

d. The names are so long it's easier to replace all of them with

```
ep.columns = ['time', 'program', 'career', 'code_exp', 'stats_exp',
              'media', 'other', 'major', 'data', 'why', 'topics']
```

e. This is practice with breaking complicated code into manageable pieces:

- `list(ep)` is a list of variable names, each of them strings
- `list(ep)[1]` is the second variable name,
  `'What program are you enrolled in?'`,
  or its replacement `program`.
- `ep[list(ep)[1]]` is the second variable — a series.
- `ep[list(ep)[1]].value_counts()` applies the `value_counts()` method, which lists the number of each response to the question. Here we get

```
MBA                          46
Undergraduate business       36
Other undergraduate          13
```

  and a collection of other responses with one or two each.

3. Consider the 538 college majors data at `url`:

```
url1 = 'https://raw.githubusercontent.com/fivethirtyeight/data/master/'
url2 = 'college-majors/recent-grads.csv'
url = url1 + url2
```

The variables are described at

https://github.com/fivethirtyeight/data/tree/master/college-majors

a. Create a dataframe `df538` from the csv file at `url` using `read_csv()`. What are its dimensions?

b. What argument/parameter would you use to read only the first ten lines of the file?

c. Extract the variables numbered `[2, 4, 15, 16, 17]`. What are the names of these variables? What do they represent?

d. Set the index equal to `Major`.

e. Use the `sort_values()` method to sort the data by `Total`.

f. What code would you use to extract the ten majors with the greatest number of people?

g. *Challenging*. Construct horizontal bar charts of the top ten majors sorted, first, by median salary and, second, by the salary of the 25th percentile. In each case plot just the variable you sorted on.

Solution:

a. Use the code: `df538 = pd.read_csv(url)`

b. `nrows=10`.

c. The variables are total (the total number for this major), median (median salary), P25th (25th percentile salary), and P75th (75th percentile salary). The last three describe the overall distribution of salaries: the 25th, 50th, and 75th percentiles.

d. `df = df.set_index(['Major'])`

e. `h = df.head(10)`

f. `h['Median'].sort_values().plot(kind='barh')`

g. `h['P25th'].sort_values().plot(kind='barh')`

4. Approximately how long did this assignment take you? Answer this by creating a .csv file and entering the time (in minutes) and posting it to your GitHub `my_first_repository` titled `time_for_practice3.csv`