Lab 04 – pandas II

Submission: `lab04.ipynb`, pushed to your Git repo on `master`.
Points: 10
Due: Monday, January 28, 11:59pm

Objectives

- `pandas`

Introduction

Create a **`lab04.ipynb`** file. Create your header cell, then your first cell to set up your imports:

```
import sys
import numpy as np
import pandas as pd
```

As mentioned in the previous lab, a large portion of this lab is taken from snippets scattered throughout the enormous documentation and tutorials from the pandas website at http://pandas.pydata.org/pandas-docs/stable/index.html .

I also recommend the lynda.com (now Linkedin Learning) courses, which are available for free. Simply go there and log in through Bucknell, and have fun! Search for any skill you are after. There are no lack of material for searching `"Python data science"`.

Finally, please take note of the two online O'Reilly books I put on Moodle that are often cited as excellent places to learn numpy and pandas, and to use as a reference later in your work. One book in particular, Python for Data Analysis, is actually written by the developer of pandas, Wes McKinney.

Exercises

To start, let's recreate the quiz scores dataset. Place the import statements at the top of your notebook file, create your first cell, and copy the following into it:

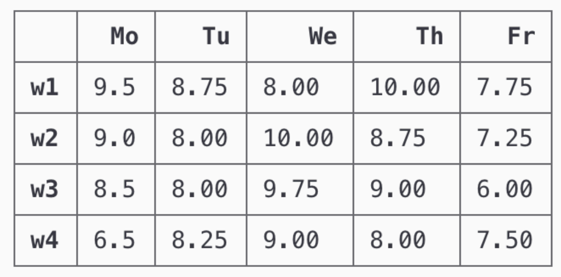
```
days = ["Mon","Tue","Wed","Thur","Fri"]
scores = pd.DataFrame([pd.Series([9.5, 8.75, 8, 10, 7.75], index=days, name="week_1"),
                       pd.Series([9, 8, 10, 8.75, 7.25], index=days, name="week_2"),
                       pd.Series([8.5, 8, 9.75, 9, 6], index=days, name="week_3"),
                       pd.Series([6.5, 8.25, 9, 8, 7.5], index=days, name="week_4")])
scores
```

1) [P] Report the number of dimensions, the shape, the size of the data, and the data types in the scores.

2) [P] How many observations? How many variables (i.e. columns)?

3) [P] Rename the `'Thur'` column header to be `'Thu'` Show the new scores data frame

4) [P] Rename the index to be `'w1'`, `'w2'`, `'w3'`, and `'w4'`. Rename the variables to be `'Mo'`, `'Tu'`, `'We'`, `'Th'`, `'Fr'`. Your new updates scores set should be as follows:



	Mo	Tu	We	Th	Fr
w1	9.5	8.75	8.00	10.00	7.75
w2	9.0	8.00	10.00	8.75	7.25
w3	8.5	8.00	9.75	9.00	6.00
w4	6.5	8.25	9.00	8.00	7.50

5) [M] Compare the type of the expression `scores['Mo']` vs. `scores[['Mo']]`. What is the difference? Explain.

6) [M] What does the `keys()` method of data frames do? Demonstrate it on the `scores` data frame.

7) [M] Read about the `describe()` method of data frames. (FYI - This is essentially the equivalent to the `summary()` function in R.)

8) [P] Demonstrate the `describe()` method on `scores`. What type does it return? (This makes it quite usable!) Look at the output. Which day had the largest standard deviation in quiz scores?

9) [P] Write code to directly determine which day had the largest standard deviation in quiz scores. Your result should simply output the abbreviated name from the columns.

10) [P] Show the result of `describe()` on the *transpose* of the `scores` data frame. Which week had the best average score?

11) [M] What does the `.values` attribute do?

12) [P] What is the mean of the entire dataset? (HINT: `.values` may make this easy)

13) [P] What is the mean for each day? Do not use `describe()`

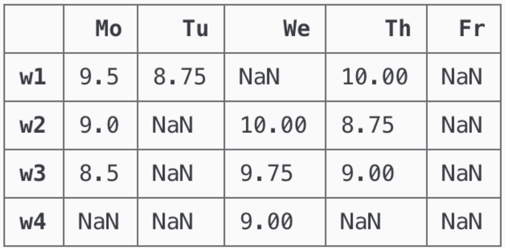
14) [P] Select the data frame for only Monday – Thursday using `.loc`, and again using `.iloc`

15) [P] Show the mean for each week after your minimum score for each week was dropped.

16) [P] Select the weeks where Friday's score was greater than 7.25, using the `[ ]` operator

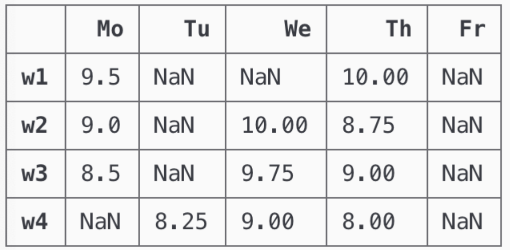
17) [P] Use the `where()` method to repeat the previous exercise.

18) Select the data where each value is greater than the mean for *all* of the data. Your result should look like:



	Mo	Tu	We	Th	Fr
w1	9.5	8.75	NaN	10.00	NaN
w2	9.0	NaN	10.00	8.75	NaN
w3	8.5	NaN	9.75	9.00	NaN
w4	NaN	NaN	9.00	NaN	NaN

19) [P] Select the scores where each value is greater than the week's mean value. Your result should look slightly different than the previous exercise:



	Mo	Tu	We	Th	Fr
w1	9.5	NaN	NaN	10.00	NaN
w2	9.0	NaN	10.00	8.75	NaN
w3	8.5	NaN	9.75	9.00	NaN
w4	NaN	8.25	9.00	8.00	NaN

For the next several exercises, you will work with your first real dataset. Include the following code in a cell:

```
url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv'
chip_df = pd.read_csv(url, sep = '\t')
```

The data was originally available for public use on Kaggle (https://www.kaggle.com/) Get to know Kaggle! If you have not yet set up an account on Kaggle, you should. It's amazing.

The data represents a single day of transactions at a busy Chipotle store. This was placed on the Kaggle website, and is available from many repositories (including the one above.) The 'item_name' variable is a nominal / categorical variable. 'choice_description' contains a list of items that are part of the main item ordered (as listed in 'item_name'.) (For example, a Chicken Bowl may contain Rice, Pinto Beans, Fresh Tomato Salsa, etc.) The rest of the columns are self explanatory.

The rest of this exercise will have you doing some basic EDA on your data. Many of these questions are based on an enormous set of pandas exercises available online at https://github.com/guipsamora/pandas_exercises . You should NOT simply copy and paste those answers! Needless to say, you're not going to learn much that way.

20) [P] How many columns in chip_df?

21) [P] How many observations are there in chip_df?

22) [P] Show the columns of chip_df

23) [P] Show the first 10 observations of chip_df

24) [P] How many unique item_name entries are there?

25) How many distinct orders are there?

26) [P] What were the top 5 ordered items? (Be sure to consider the quantity of each item order!) How many of each were ordered?

27) [P] How many orders were for "Steak Burrito"?

28) [P] What is the most frequent item ordered for "Steak Burrito" orders? (You need to explore the choice_description variable!)

29) [P] Run the describe() method on chip_df. Why are variables missing in the output? (Read the help for describe()!)

30) [P] Can you readily compute the mean of the "`item_price`" variable? If not, why not?

31) [P] Convert the `item_price` field to a floating point number. Confirm by showing the output of `info()` NOTE – This is a classic example of the type of cleaning and preprocessing you need to always deal with when working with data. We'll be doing a LOT of this very soon!

32) [P] Compute the mean price for an order (NOTE – This is NOT just a mean of the `item_price` column!

33) What was total revenue for the data? (NOTE - Make sure you consider the `quantity` variable!)

34) [P] Using Python and markdown, discuss the distribution of `quantity` – how many orders had quantity of 1? How many had 2? What is the average quantity? Range of values? Etc.

Deliverables

Commit and push `lab04.ipynb`. Be sure you have every cell run, and output generated. Verify that your file is pushed properly on Gitlab.