INFO370 Problem Set 1: Python, data manipulations (100 pt)

Your name:

Deadline: Wed, Jan 22th 10:30am

Instructions

This is the first problem set. These 100 points will give you up to 10 points of the final grade.
You have two somewhat different tasks:

- create and compute variables, and do tricks with lists and dicts and functions. The good background reading is the Lubanovic (2014) book, chapters 2 (data types), 3 (lists, dicts, sets), 4 (code structures). The basics is also explained in python notes https://otoomet.bitbucket.io/machinelearning-py.html#2_Python.
- Analyze and manipulate a dataset using filtering, merging, and related functions. This requires you to use numpy and pandas libraries. The background is provided by McKinney (2018), chapters 4, 5 (numpy and pandas), 7 (data cleaning). The basics is also explained in python notes https://otoomet.bitbucket.io/machinelearning-py.html#3_Numpy_and_Pandas.

General requirements:

- All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment. In particular, note that Stack Overflow is licensed as Creative Commons (CC-BY-SA). This means you have to attribute any code you pick from SO (a link to the question/answer webpage will normally do).
- As the final submission, you should submit a) code; b) output; and c) explanations. If you are working with jupyter notebooks, all this will be included automatically but you still have to submit both your original file (so you grader can actually run the code), and an html version of it (which is much faster to check).
- Working together is fun and useful but you have to submit your own work. Discussing the solutions and problems with your classmates is all right but do not copy-paste their solution! First understand it, and thereafter create your own solution. Please list all your collaborators!
- Partial credit will be awarded for questions for which a serious attempt at finding an answer has been shown. Attempt each question and to document your reasoning process even if you cannot quite get there!

1 Base python (30pt)

Normally we ask you to write textual answers and explain the results. However, in this basic programming task this may not be necessary.

1.1 Computing (4pt)

Compute the following numbers, assign those to suitably named variables, and print:

1. How many seconds are there in year?
2. How many seconds is a typical human lifetime? Use the previously computed seconds in year.
3. What is your age in seconds? Compute this based on your age.
4. Create a new variable
5. Create a logical variable that is true if you are more than 700M seconds old. Use logical operators, not if/else!

1.2 Lists (8pt)

1. Create a list 'movies' that contains the names of at least six movies you like
2. Using indexing and **slicing**, Create a list of three first movies in the list
3. Use slicing and list concatenation to create a list of the first two and the last two movies

1.3 Loops (8pt)

1. Create a list 'numbers' that is the numbers 70 through 79: in the following manner: create an empty list and add numbers to it in a loop (do not use list comprehension or 'list' function here!)
2. Use a loop to compute the mean value of the list: add the values to their sum in a loop, and at the end divide the sum by the number of the values
3. Use loop to compute sum of all numbers 1..100
4. Compute product of all numbers 1..100 (denoted by 100!)

1.4 Dicts (10pt)

1. Create a word frequency table using dicts that take a looong string (feel free to copy-paste some text here), splits it into individual words, and counts the number of occurrencies of each word (and prints the result). Let's ignore punctuation. Your code should run over individual words and increase the counter for that word, stored in a list. It has to check if a word exists in the dictionary, and if not, take an appropriate action.

   Hint: use the split() method: `"I have no clue".split()` -> `['I', 'have', 'no', 'clue']`

   Hint 2: convert everything to lower case

   Hint 3: use triple quotes `"""` to create long multi-line strings

   Hint 4:

1.5 Functions

1. Write a function that takes in time in the form of HHMM (hours-minutes), and returns it in the form of HH.HH (hours + fractions of hours). For instance, 1015 → 10.25 (10 hrs 15 mins → 10.25 hrs). Demonstrate it works using values 1015 and 345.

   Hint: use modulo operator % and integer division operator //. Modulo of 100 gives you minutes and integer division by 100 gives you hours

2 Work with NYC flights data

The second task is to analyze flights out of NYC airports in 2013.

Here explanations are a major part of the solution:

- Your results will only count if accompanied with sufficiently and clear explanatory text. Just plain output, with no explanation, will not count.
- Be sure that each visualization (graph or table) adds value to your written explanation; avoid redundancy – you do no need four different visualizations of the same pattern.
- Don't output irrelevant, or too much of relevant information. A few figures is helpful. A few thousand figures is useless.

2.1 Setup (5pt)

In this problem set you will work with NYC flights data. The data is copied from the corresponding R package, you can read the documentation at e.g. RDocumentation.

1. Load the data
2. Ensure you know the variables in the data. Keep the documentation nearby.
3. Make sure you have read the background readings about pandas (see above).

2.2 Explore the data (15pt)

First, let's do some data exploration. Answer the following questions: show the code, the computation results, and comment the results in the accompanying text.

1. How many flights out of NYC where there in 2013?
2. How many NYC airports are included in this data? Which airports are these?
3. Into how many airports did the airlines fly from NYC in 2013?
4. How many flights were there from NYC to Seattle (airport code *SEA*)?
5. Were the any flights from NYC to Spokane (GEG)?
6. What is the typical delay of the flights in this data?
7. Did you remember to check how good is the delay variable? Are there missings? Are there any implausible or invalid entries? Go and check this if you haven't done it already.
8. Where was there room for interpretation with these questions (and answers)? How would that affect decisions based on this data?

2.3 Let's fly to Denver! (20pt)

Now let's see how is it to fly from NYC to Denver (airport code *DEN*).

1. How many flights were there from NYC airports to Denver in 2013?
2. How many airlines fly from NYC to Denver?
3. Which are these airlines (find the 2-letter abbreviations)? How many times did each of these go to Denver?
4. What was the average flight duration to Denver? Use arrival and departure time, and compute the flight time yourself! Do not use the airtime variable!

   Compare your results with those you find in flight schedules. What do you find?
5. What percentage of flights to Denver were delayed at arrival by more than 15 minutes?

2.4 What are these planes? (20pt)

Your final data analysis task is to analyze the planes that flew to Denver. You need to load the *planes.csv* dataset and merge with the flights data.

1. Load the planes data. What are the variables? How many planes do we have?
2. What would be the *merge key*, the variable that can connect a flight in the flights data with a plane in the planes data?
3. Merge the two datasets. Do you want to do inner, full, left, right merge?
4. How many flights to Denver do we have where we don't have the data about number seats in the plane?
5. What was the largest plane in terms of seats that flew from NYUC to Denver? (or what were the largest planes if there were several equally large planes) Tell us it's number of seats, manufacturer, model, and year the plane was built.

2.5 Think about all this (10pt)

Finally, think about the questions and the analysis.

1. What are your main concerns related to the analysis above? Were questions, answers, methodology your were able to use, and data good enough?
2. Can you envision any business-relevant analysis you might do using this data? Hint: it does not have to be freight business, e.g. journalism is business too.

3 How much time did you spend?

And finally-finally, tell us how much time (how many hours) did you spend on this PS!

References

Lubanovic, B. (2014) *Introducing Python: Modern Computing in Simple Packages*, O'Reilly Media.

McKinney, W. (2018) *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, O'Reilly Media, 2nd edn.