Statistical Computing with Python

Author: Laercio Serra

This is a quick tutorial and here I'll show you “how-to do” some statistical programming tasks using python. For that, is necessary to have some basic knowledge with python and be familiar with statistical programming in a language like R, Stata, SAS, SPSS or Matlab.

Part 2

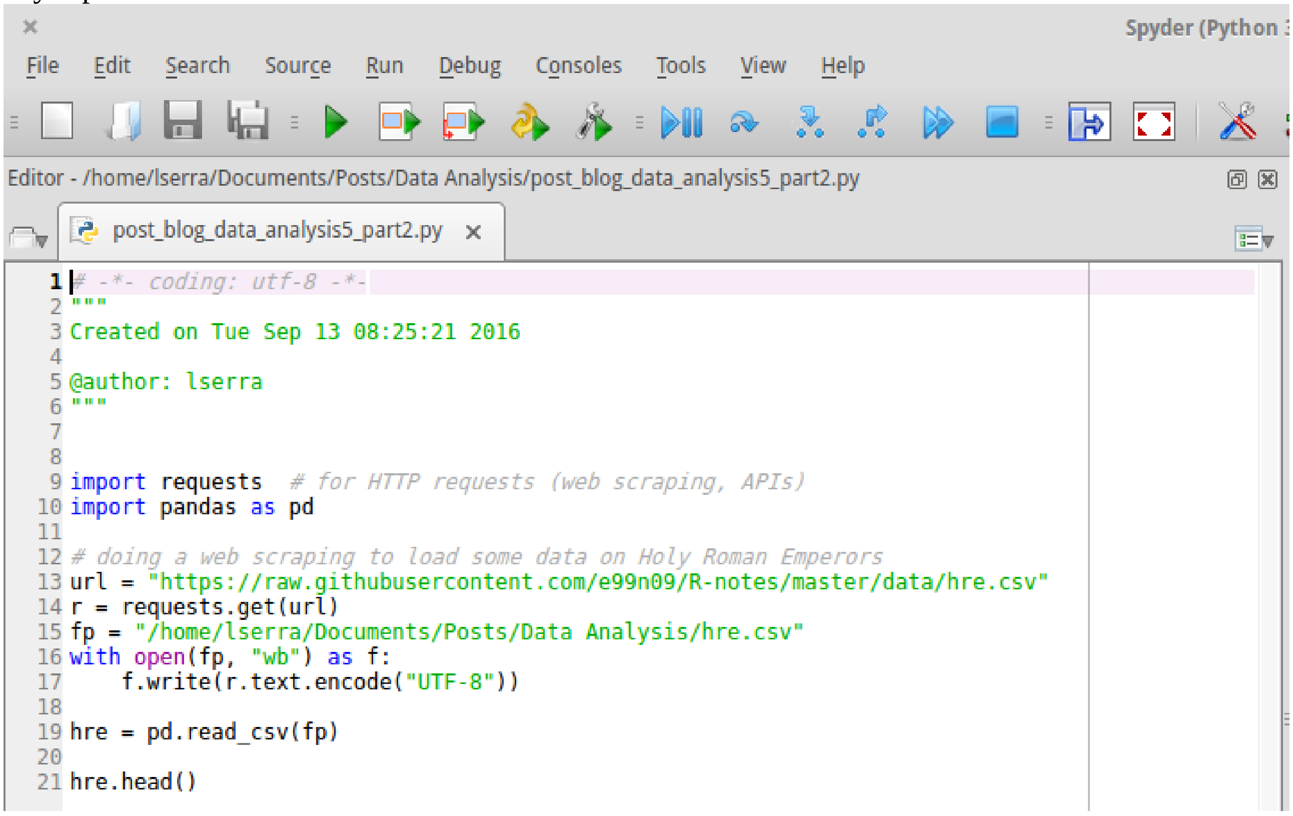
This part is about the data cleaning and exploratory analysis tasks. Here's a more complicated example that demonstrates a basic data cleaning workflow leading to the creation of some exploratory plots and the running of a linear regression. The dataset was transcribed from Wikipedia by hand. It contains all the Holy Roman Emperors and the important milestones in their lives (birth, death, coronation, etc.). The goal of the analysis will be to explore whether a relationship exists between emperor birth year and emperor lifespan.

Data Source: https://en.wikipedia.org/wiki/Holy_Roman_Emperor

1. **Simple data cleaning and exploratory analysis**

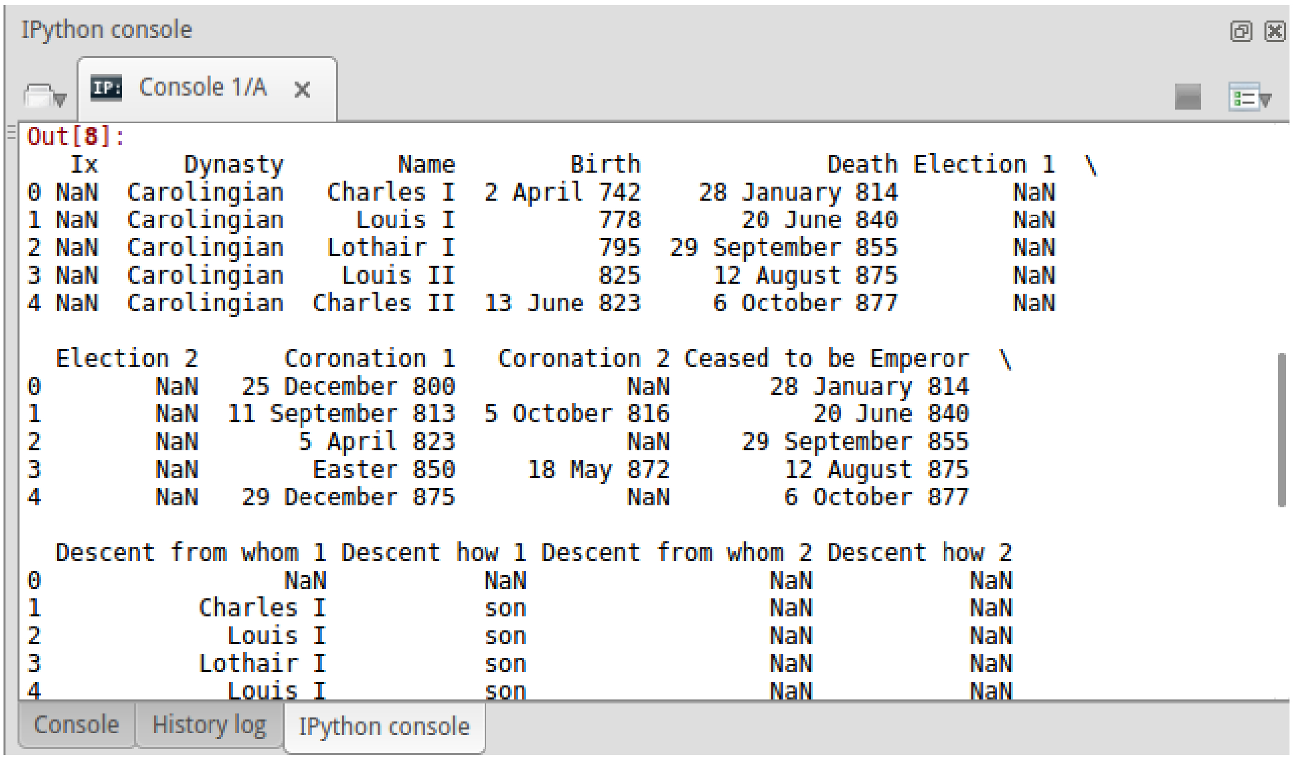
One reason people choose Python over R is that they intend to interact a lot with the web, either by scraping pages directly or requesting data through an API. You can do those things in R, but in the context of a project already using Python, there's a benefit to sticking with one language.

Try reproduce the code bellow:


Spyder (Python 3
File
Edit
Search
Source
Run
Debug
Consoles
Tools
View
Help
Editor - /home/lserra/Documents/Posts/Data Analysis/post_blog_data_analysis5_part2.py
post_blog_data_analysis5_part2.py

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Sep 13 08:25:21 2016

@author: lserra
"""


import requests  # for HTTP requests (web scraping, APIs)
import pandas as pd

# doing a web scraping to load some data on Holy Roman Emperors
url = "https://raw.githubusercontent.com/e99n09/R-notes/master/data/hre.csv"
r = requests.get(url)
fp = "/home/lserra/Documents/Posts/Data Analysis/hre.csv"
with open(fp, "wb") as f:
    f.write(r.text.encode("UTF-8"))

hre = pd.read_csv(fp)

hre.head()
```

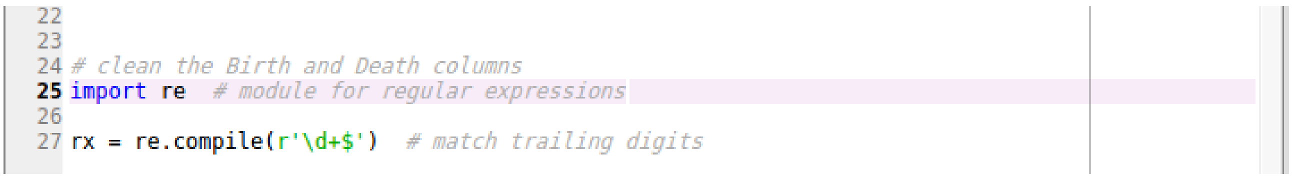
Now press the button F5 (Run File) and the results are:



```
Out[8]:
   Ix      Dynasty        Name         Birth              Death Election 1  \
0 NaN  Carolingian   Charles I  2 April 742    28 January 814        NaN
1 NaN  Carolingian     Louis I          778       20 June 840        NaN
2 NaN  Carolingian   Lothair I          795  29 September 855        NaN
3 NaN  Carolingian    Louis II          825    12 August 875        NaN
4 NaN  Carolingian  Charles II  13 June 823    6 October 877        NaN

  Election 2       Coronation 1  Coronation 2 Ceased to be Emperor  \
0        NaN   25 December 800           NaN       28 January 814
1        NaN  11 September 813  5 October 816          20 June 840
2        NaN       5 April 823           NaN     29 September 855
3        NaN        Easter 850     18 May 872        12 August 875
4        NaN   29 December 875           NaN        6 October 877

  Descent from whom 1 Descent how 1 Descent from whom 2 Descent how 2
0                 NaN           NaN                 NaN           NaN
1           Charles I           son                 NaN           NaN
2             Louis I           son                 NaN           NaN
3           Lothair I           son                 NaN           NaN
4             Louis I           son                 NaN           NaN
```

If you can see this same result. Congratulations...You did a great job!

Now as we can see in the figure above we need to do some cleaning data tasks. Here let's need to use a new library. This new library is a module for regular expressions.



```python


# clean the Birth and Death columns
import re  # module for regular expressions

rx = re.compile(r'\d+$')  # match trailing digits
```

This function applies the regular expression to an input column (Birth and Death), flattens the resulting list, converts it to a Series object, and finally converts the type of the Series object from string to integer. For more information into what different parts of the code do, see:

- https://docs.python.org/2/howto/regex.html
- http://stackoverflow.com/questions/11860476/how-to-unlist-a-python-list
- http://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.html

For more details or information about the use of this library, you can access the documentation on http://matplotlib.org/api/pyplot_api.html