Data Wrangling in python: Walkthrough with a practical example

# Data Wrangling definition

Wrangling is just another fancy word for data preparation. It is the processing of data to make further exploration and analysis straightforward and less error prone. Sometimes it is as simple as opening an Excel formatted file in MS Excel, and you are well on your way. Most times, as we will surely experience, it is not that simple. Most data in the world are inherently messy or ‘untidy’, if you will. Sometimes, the data is arranged in rows, but you want them in the form of columns to make the next step of your analysis possible. Most times, this is just one step of the metaphorical staircase that is data cleaning.

Here we will follow a typical data wrangling framework for cleaning some part of a sample dataset. This is a real-world dataset, as you will find out soon, that will help us in getting our hands dirty with the first step of the complete data analysis process.

You can download the complete code base and the datasets used in this example from the github repo using this link: <https://github.com/rashokanand/wrangling-armenian-job-postings-kaggle>

Here you will find the Jupyter notebook along with the zip folder used in the analysis of this dataset.

# Background of dataset used

The dataset that we are using can be found on Kaggle. It contains over 19000 job posting in the period between 2004 to 2015. The data was mined using text mining techniques from an Armenian HR portal. Though most of the heavy lifting of data wrangling is already been done, it still needs a little work to make it useable. It should be noted that our objective is not to completely and exhaustively clean the dataset, but to get a first hand taste of what data cleaning is and how it can be done in the real world. The objective is to use a consistent and reproducible framework deliberately designed to aid in systematizing data wrangling.

# Framework for data wrangling

Wrangling literally means ‘to round up, herd or take charge of (livestock)’. At least that’s what the Oxford dictionary definition says. It is probably safe to say that, it’s true in the case of data wrangling as well. Here we need to get the data in order to make the next steps of data analysis possible.

We here will follow a three step process to complete the data wrangling process: Gather, Assess and Clean.

# Gather

This consists of getting the data into the programming environment or the spreadsheet application that we are using presently. This may include the actual data download, extract and then import. For the purpose of this article, we will download the files manually, then proceed doing the rest of the process manually.

Using the link in the references, go ahead and download the zip file. We may need to register an account in Kaggle for this. The downloaded file’s name is ‘armenian-online-job-postings.zip’. Do not unzip it just yet. We will do it directly using python in a jupyter notebook.

Create a folder which will be your current working directory. Move the zip into this folder. Start up the Jupyter notebook in this directory. You may need to change your directory in the terminal first using ‘cd’. Then through the command ‘Jupyter notebook’ launch the notebook in your browser. If you do not understand what just happened, you need to first understand what are jupyter notebooks and how to install and run them. The best resource for this purpose (that I found to be most helpful) is jupyter.org. It contains tutorials and examples for installing and running the same successfully. Comment if you are stuck anywhere in this process.

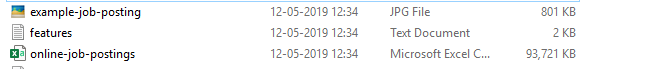
Alright then, let’s go ahead. Using the zipfile module to extract the files in the archive is simple using the ‘extractall’ method

# Extract all contents from zip file

with zipfile.ZipFile('armenian-online-job-postings.zip', 'r') as myzip:

myzip.extractall()

This code works exactly like how when we right click on an archive and click on ‘extract all files here’. Reading up on the documentation of zipfile, or for that matter any module that you would like help on is a very valuable and underrated skill.



After the files are extracted we are now looking at a collection of 3 files. Our interest is with the ‘online-job-postings.csv’ file.

Using pandas we can read in the dataset like so.

# df = pd.read\_csv('online-job-postings.csv')

The above code will work only when the csv is in the same directory as your jupyter notebook. Or we need to pass the complete path as the argument to the read\_csv method of pandas. It then stores the output dataframe into the variable ‘df’.

Congratulations, you are done with the ‘Gather’ step of the data wrangling process. Two more to go.

# Assess

# Clean

## Define

## Code

## Test

# References

<https://www.kaggle.com/udacity/armenian-online-job-postings>