Microsoft: DAT210x Programming with Python for Data Science

Bookmarks

Start Here

1. The Big Picture

2. Data And Features

Lecture: Features Premiere
Quiz

Lecture: Determining Features
Quiz

Lecture: Manipulating Data
Quiz

Lecture: Feature Representation
Quiz

Lecture: Wrangling Data
Quiz

Lab: Data and Features
Lab

Dive Deeper

Bookmark

Lab Assignment 4

Navigate over to ESPN's website for NHL Historic Player Points Statistics, for the years 2014-2015. The page has a table on it with a few stats we're interested in obtaining. But it's a bit messy! Clean it up for us, using the appropriate commands to:

1. Load up the table into a Pandas dataframe
2. Rename the columns so that they match the column definitions on the website
3. Get rid of any erroneous rows that has at least 4 NANs in them
4. Get rid of the **RK** column
5. Ensure there are no nan "holes" in your index
6. Check the dtypes of all columns, and ensure those that *should* be numeric are numeric

Lab Questions

(3/3 points)

Please enter a numeric value (e.g. 0, 1, 10.5, etc) which correctly answers the question(s) below:

After completing the 6 steps above, how many rows **remain** in this dataset? (Not to be confused with the index!)

- 3. Exploring Data
- 4. Transforming Data
- 5. Data Modeling

40 ✓ **Answer:** 40

How many unique **PCT** values exist in the table?

36 ✓ **Answer:** 36

What is the **value** you get by adding the **GP** values at *indices* 15 and 16 of this table?

164 ✓ **Answer:** 164

EXPLANATION

First load up your data using Pandas **.read_html()** method. This actually does not return a dataframe, but rather returns a list of dataframes, one per HTML table found on the page.

The next thing you're going to want to do is use **.columns = []** to assign names to all of the columns. It's important that at this step, you know the difference between using **.columns** and **names=** within one of your dataset load methods!

You're going to have to use a combination of **.drop()**, **.dropna()**, as well as some compound boolean selectors in order to clean up the table by getting rid of all the pesky rows. Review your previous assignments to see how this is done.

Don't forget to reset your index, and also pass **Drop=True** when you do that, otherwise a copy of the original index column will be retained.

Lastly, you need to check your table's data types. **.read_html()** always converts everything to a string, so you'll have to coerce your data to a numeric type using **pd.to_numeric()**. Read up on the documentation for that, and be sure to remember why that method is preferred to other methods, such as **astype()**.

You have used 2 of 2 submissions