Assignment 4

September 24, 2017

You are currently looking at ***version 1.1*** *of this notebook. To download notebooks and datafiles, as well as get help on Jupyter notebooks in the Coursera platform, visit the Jupyter Notebook FAQ course resource.*

```
In [1]: import pandas as pd
        import numpy as np
        from scipy.stats import ttest_ind
```

1 Assignment 4 - Hypothesis Testing

This assignment requires more individual learning than previous assignments - you are encouraged to check out the pandas documentation to find functions or methods you might not have used yet, or ask questions on Stack Overflow and tag them as pandas and python related. And of course, the discussion forums are open for interaction with your peers and the course staff.

Definitions: * A *quarter* is a specific three month period, Q1 is January through March, Q2 is April through June, Q3 is July through September, Q4 is October through December. * A *recession* is defined as starting with two consecutive quarters of GDP decline, and ending with two consecutive quarters of GDP growth. * A *recession bottom* is the quarter within a recession which had the lowest GDP. * A *university town* is a city which has a high percentage of university students compared to the total population of the city.

Hypothesis: University towns have their mean housing prices less effected by recessions. Run a t-test to compare the ratio of the mean price of houses in university towns the quarter before the recession starts compared to the recession bottom. (`price_ratio=quarter_before_recession/recession_bottom`)

The following data files are available for this assignment: * From the Zillow research data site there is housing data for the United States. In particular the datafile for all homes at a city level, `City_Zhvi_AllHomes.csv`, has median home sale prices at a fine grained level. * From the Wikipedia page on college towns is a list of university towns in the United States which has been copy and pasted into the file `university_towns.txt`. * From Bureau of Economic Analysis, US Department of Commerce, the GDP over time of the United States in current dollars (use the chained value in 2009 dollars), in quarterly intervals, in the file `gdplev.xls`. For this assignment, only look at GDP data from the first quarter of 2000 onward.

Each function in this assignment below is worth 10%, with the exception of `run_ttest()`, which is worth 50%.

```
In [ ]: # Use this dictionary to map state names to two letter acronyms
        states = {'OH': 'Ohio', 'KY': 'Kentucky', 'AS': 'American Samoa',
                  'NV': 'Nevada', 'WY': 'Wyoming', 'NA': 'National',
                  'AL': 'Alabama', 'MD': 'Maryland', 'AK': 'Alaska',
                  'UT': 'Utah', 'OR': 'Oregon', 'MT': 'Montana',
                  'IL': 'Illinois', 'TN': 'Tennessee',
                  'DC': 'District of Columbia', 'VT': 'Vermont',
                  'ID': 'Idaho', 'AR': 'Arkansas', 'ME': 'Maine',
                  'WA': 'Washington', 'HI': 'Hawaii', 'WI': 'Wisconsin',
                  'MI': 'Michigan', 'IN': 'Indiana', 'NJ': 'New Jersey',
                  'AZ': 'Arizona', 'GU': 'Guam', 'MS': 'Mississippi',
                  'PR': 'Puerto Rico', 'NC': 'North Carolina',
                  'TX': 'Texas', 'SD': 'South Dakota',
                  'MP': 'Northern Mariana Islands', 'IA': 'Iowa',
                  'MO': 'Missouri', 'CT': 'Connecticut',
                  'WV': 'West Virginia', 'SC': 'South Carolina',
                  'LA': 'Louisiana', 'KS': 'Kansas', 'NY': 'New York',
                  'NE': 'Nebraska', 'OK': 'Oklahoma', 'FL': 'Florida',
                  'CA': 'California', 'CO': 'Colorado', 'PA': 'Pennsylvania',
                  'DE': 'Delaware', 'NM': 'New Mexico', 'RI': 'Rhode Island',
                  'MN': 'Minnesota', 'VI': 'Virgin Islands',
                  'NH': 'New Hampshire', 'MA': 'Massachusetts', 'GA': 'Georgia',
                  'ND': 'North Dakota', 'VA': 'Virginia'}
```

```
In [ ]: def get_list_of_university_towns():
            '''Returns a DataFrame of towns and the states they are in from the
            university_towns.txt list. The format of the DataFrame should be:
            DataFrame( [ ["Michigan", "Ann Arbor"], ["Michigan", "Yipsilanti"] ],
            columns=["State", "RegionName"]  )

            The following cleaning needs to be done:

            1. For "State", removing characters from "[" to the end.
            2. For "RegionName", when applicable, removing every character from
               " (" to the end.
            3. Depending on how you read the data, you may need to remove
                newline character '\n'. '''

            return "ANSWER"
```

```
In [ ]: def get_recession_start():
            '''Returns the year and quarter of the recession start time as a
            string value in a format such as 2005q3'''

            return "ANSWER"
```

```
In [ ]: def get_recession_end():
            '''Returns the year and quarter of the recession end time as a
            string value in a format such as 2005q3'''

            return "ANSWER"
```

```
In [ ]: def get_recession_bottom():
            '''Returns the year and quarter of the recession bottom time as a
            string value in a format such as 2005q3'''

            return "ANSWER"
```

```
In [ ]: def convert_housing_data_to_quarters():
            '''Converts the housing data to quarters and returns it as mean
            values in a dataframe. This dataframe should be a dataframe with
            columns for 2000q1 through 2016q3, and should have a multi-index
            in the shape of ["State","RegionName"].

            Note: Quarters are defined in the assignment description, they are
            not arbitrary three month periods.

            The resulting dataframe should have 67 columns, and 10,730 rows.
            '''

            return "ANSWER"
```

```
In [ ]: def run_ttest():
            '''First creates new data showing the decline or growth of housing
            prices between the recession start and the recession bottom. Then
            runs a ttest comparing the university town values to the
            non-university towns values, return whether the alternative
            hypothesis (that the two groups are the same) is true or not as
            well as the p-value of the confidence.

            Return the tuple (different, p, better) where different=True if
            the t-test is True at a p<0.01 (we reject the null hypothesis), or
            different=False if otherwise (we cannot reject the null hypothesis).
            The variable p should be equal to the exact p value returned from
            scipy.stats.ttest_ind(). The value for better should be either
            "university town" or "non-university town" depending on which has a
            lower mean price ratio (which is equivilent to a reduced market loss).
            '''

            return "ANSWER"
```