Assignment 2

For this assignment you'll be looking at 2017 data on immunizations from the CDC. Your datafile for this assignment is in assets/NISPUF17.csv (assets/NISPUF17.csv). A data users guide for this, which you'll need to map the variables in the data to the questions being asked, is available at assets/NIS-PUF17-DUG.pdf (assets/NIS-PUF17-DUG.pdf). **Note: you may have to go to your Jupyter tree (click on the Coursera image) and navigate to the assignment 2 assets folder to see this PDF file).**

Question 1

Write a function called `proportion_of_education` which returns the proportion of children in the dataset who had a mother with the education levels equal to less than high school (<12), high school (12), more than high school but not a college graduate (>12) and college degree.

This function should return a dictionary in the form of (use the correct numbers, do not round numbers):

```
{"less than high school":0.2,
"high school":0.4,
"more than high school but not college":0.2,
"college":0.2}
```

In [1]:

```
def proportion_of_education():
    # your code goes here
    import pandas as pd
    #Make unnamed column as serial number
    df = pd.read_csv('assets/NISPUF17.csv', index_col = 0)
    # Cleaning up the column headers to make them all lowercase
    cols = list(df.columns)
    cols = [x.lower().strip() for x in cols]
    df.columns = cols
    #Create a series of EDUC1
    edu_series = df['educ1']
    #This dictionary will first store count, before dividing by the length of da
ta in the end.
    result = {
        "less than high school": 0.0,
        "high school": 0.0,
        "more than high school but not college": 0.0,
        "college": 0.0
    }
    len_data = 0
    for item in edu_series:
        len_data+=1
        if item == 1:
            result['less than high school']+=1
        elif (item == 2):
            result['high school']+=1
        elif item == 3:
            result['more than high school but not college']+=1
        else:
            result['college']+=1
    result = {k:v / len_data for k, v in result.items()}
    return result
    # YOUR CODE HERE
    raise NotImplementedError()
test = proportion_of_education()
test
```

Out[1]:

```
{'less than high school': 0.10202002459160373,
 'high school': 0.172352011241876,
 'more than high school but not college': 0.24588090637625154,
 'college': 0.47974705779026877}
```

In [2]:

```
assert type(proportion_of_education())==type({}), "You must return a dictionar
y."
assert len(proportion_of_education()) == 4, "You have not returned a dictionary
 with four items in it."
assert "less than high school" in proportion_of_education().keys(), "You have no
t returned a dictionary with the correct keys."
assert "high school" in proportion_of_education().keys(), "You have not returned
a dictionary with the correct keys."
assert "more than high school but not college" in proportion_of_education().keys
(), "You have not returned a dictionary with the correct keys."
assert "college" in proportion_of_education().keys(), "You have not returned a d
ictionary with the correct keys."
```

Question 2

Let's explore the relationship between being fed breastmilk as a child and getting a seasonal influenza vaccine from a healthcare provider. Return a tuple of the average number of influenza vaccines for those children we know received breastmilk as a child and those who know did not.

This function should return a tuple in the form (use the correct numbers:

```
(2.5, 0.1)
```

In [3]:

```
def average_influenza_doses():
    import pandas as pd
    df = pd.read_csv('assets/NISPUF17.csv')
    # Cleaning up the column headers to make them all lowercase and remove any r
andom whitespace
    cols = list(df.columns)
    cols = [x.lower().strip() for x in cols]
    df.columns = cols
    # Obtaining columns cbf_01 and p numflu
    # Creating a boolean mask for kids who know whether they have taken breast m
ilk:
    boolean = (df['cbf_01'] == 1) | (df['cbf_01'] == 2)
    #Apply bm_mask to dataframe, drop all values with false or with 'NaN'
    new_df = df[boolean]
    #To this newdf, obtain the cbf_01 and p numflu columns.
    # Can I assume that those who have NaN values did not receive seasonal influ
enza vaccines?
    new_df = new_df[['cbf_01','p_numflu']].dropna()
    count1 = 0; row_count1 = 0
    count2 = 0; row_count2 = 0
    for index, row in new_df.iterrows():
        if(row['cbf_01']%2 == 1): # Received breast milk
            count1+=row['p_numflu']
            row_count1+=1
        else: # Did not receive breastmilk
            count2+=row['p_numflu']
            row_count2+=1
    total = count1+count2
    return (count1/row_count1, count2/row_count2)
    # YOUR CODE HERE
    raise NotImplementedError()
test = average_influenza_doses()
test
```

Out[3]:

```
(1.8799187420058687, 1.5963945918878317)
```

In [4]:

```
assert len(average_influenza_doses())==2, "Return two values in a tuple, the fir
st for yes and the second for no."
```

Question 3

It would be interesting to see if there is any evidence of a link between vaccine effectiveness and sex of the child. Calculate the ratio of the number of children who contracted chickenpox but were vaccinated against it (at least one varicella dose) versus those who were vaccinated but did not contract chicken pox. Return results by sex.

This function should return a dictionary in the form of (use the correct numbers):

```
{"male":0.2,
"female":0.4}
```

Note: To aid in verification, the `chickenpox_by_sex()['female']` value the autograder is looking for starts with the digits `0.0077`.

In [5]:

```
def chickenpox_by_sex():
    # YOUR CODE HERE
    import pandas as pd
    df = pd.read_csv('assets/NISPUF17.csv')
    cols = list(df.columns)
    cols = [x.lower().strip() for x in cols]
    df.columns = cols
    cpox_df = df[['sex','had_cpox','p_numvrc']].dropna(0)
    # Apply a boolean mask to only those who know they had cpox or not.
    cpox_df = cpox_df[(cpox_df['had_cpox']==1) | (cpox_df['had_cpox']==2)]
    #Apply a second mask to those who were vaccinated.
    cpox_df = cpox_df[cpox_df['p_numvrc']>0]
    #Data manipulation legend.
    #Male is 1, Female is 2
    #had cpox = 1, did not have = 2
    male_pos = male_neg = fm_pos = fm_neg = 0;
    for index, row in cpox_df.iterrows():
        if (row['sex'] %2 == 1): #Male
            if (row['had_cpox']%2 == 1): #have cpox
                male_pos+=1
            else: #no cpox
                male_neg+=1
        else: #Female
            if (row['had_cpox']%2 == 1): #have cpox
                fm_pos+=1
            else: #no cpox
                fm_neg+=1
    result = {'male': male_pos/male_neg, 'female':fm_pos/fm_neg}
    return result
    raise NotImplementedError()

test = chickenpox_by_sex()
test
```

Out[5]:

```
{'male': 0.009675583380762664, 'female': 0.0077918259335489565}
```

In [6]:

```
assert len(chickenpox_by_sex())==2, "Return a dictionary with two items, the fir
st for males and the second for females."
```

Question 4

A correlation is a statistical relationship between two variables. If we wanted to know if vaccines work, we might look at the correlation between the use of the vaccine and whether it results in prevention of the infection or disease [1]. In this question, you are to see if there is a correlation between having had the chicken pox and the number of chickenpox vaccine doses given (varicella).

Some notes on interpreting the answer. If the `had_chickenpox_column` is either `1` (for yes) or `2` for no, and that the `num_chickenpox_vaccine_column` is the number of doses a child has been given of the varicella vaccine, then a positive correlation (e.g. `corr > 0`) would mean that an increase in `had_chickenpox_column` (which means more no's) would mean an increase in the `num_chickenpox_vaccine_column` (which means more doses of vaccine). If `corr < 0` then there is a negative correlation, indicating that having had chickenpox is related to an increase in the number of vaccine doses. Also, `pval` refers to the probability the relationship observed is significant. In this case `pval` should be very very small (will end in `e-18` indicating a very small number), which means the result unlikely to be by chance.

[1] This isn't really the full picture, since we are not looking at when the dose was given. It's possible that children had chickenpox and then their parents went to get them the vaccine. Does this dataset have the data we would need to investigate the timing of the dose?

In [7]:

```
def corr_chickenpox():
    import scipy.stats as stats
    import numpy as np
    import pandas as pd

    df = pd.read_csv('assets/NISPUF17.csv')
    cols = list(df.columns)
    cols = [x.lower().strip() for x in cols]
    df.columns = cols
    #Keep columns that we want
    df = df[["p_numvrc","had_cpox"]].dropna()
    #Eliminate anomalies from dataset
    df= df[(df['had_cpox']==1)|(df['had_cpox']==2)]
    # here is some stub code to actually run the correlation
    corr, pval=stats.pearsonr(df["p_numvrc"],df["had_cpox"])
    print(pval)
    # just return the correlation
    return corr

    # YOUR CODE HERE
    raise NotImplementedError()
test = corr_chickenpox()
test
```

```
2.7780263182916748e-18
```

Out[7]:

```
0.07044873460147986
```

In [8]:

```
assert -1<=corr_chickenpox()<=1, "You must return a float number between -1.0 and 1.0."
```

```
2.7780263182916748e-18
```

In [ ]:

```

```