1. Test cell (Score: 1.0 / 1.0)
2. Test cell (Score: 1.0 / 1.0)
3. Test cell (Score: 1.0 / 1.0)
4. Test cell (Score: 1.0 / 1.0)

Assignment 2¶

For this assignment you'll be looking at 2017 data on immunizations from the CDC. Your datafile for this assignment is 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 folde 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]:

Student's answer

```python
def proportion_of_education():
    # your code goes here
    # YOUR CODE HERE
    # raise NotImplementedError()
    import pandas as pd
    import numpy as np
    df = pd.read_csv("assets/NISPUF17.csv", index_col=0)
    EDUS=df['EDUC1']
    edus=np.sort(EDUS.values)
    poe={"less than high school":0,
        "high school":0,
        "more than high school but not college":0,
        "college":0}
    n=len(edus)
    poe["less than high school"]=np.sum(edus==1)/n
    poe["high school"]=np.sum(edus==2)/n
    poe["more than high school but not college"]=np.sum(edus==3)/n
    poe["college"]=np.sum(edus==4)/n
    return poe
proportion_of_education()
```

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]:

Grade cell: `cell-c0eeef201366f51c`   Score: 1.0 / 1.0

```python
assert type(proportion_of_education())==type({}), "You must return a dictionary."
assert len(proportion_of_education()) == 4, "You have not returned a dictionary with four i
assert "less than high school" in proportion_of_education().keys(), "You have not returned
assert "high school" in proportion_of_education().keys(), "You have not returned a dictiona
assert "more than high school but not college" in proportion_of_education().keys(), "You ha
assert "college" in proportion_of_education().keys(), "You have not returned a dictionary w
```

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]:

Student's answer

```python
def average_influenza_doses():
    # YOUR CODE HERE
    # raise NotImplementedError()
    import pandas as pd
    import numpy as np
    df = pd.read_csv("assets/NISPUF17.csv", index_col=0)

    cbf_flu=df.loc[:,['CBF_01','P_NUMFLU']]


    cbf_flu1=cbf_flu[cbf_flu['CBF_01'] ==1].dropna()
    cbf_flu2=cbf_flu[cbf_flu['CBF_01'] ==2].dropna()

    flu1=cbf_flu1['P_NUMFLU'].values.copy()
    flu1[np.isnan(flu1)] = 0
    f1=np.sum(flu1)/len(flu1)

    flu2=cbf_flu2['P_NUMFLU'].values.copy()
    flu2[np.isnan(flu2)] = 0
    f2=np.sum(flu2)/len(flu2)

    aid =(f1,f2)
    return aid
average_influenza_doses()
```

Out[3]:

```
(1.8799187420058687, 1.5963945918878317)
```

In [4]:

Grade cell: `cell-54a3ba6cff31caa7`   Score: 1.0 / 1.0

```python
assert len(average_influenza_doses())==2, "Return two values in a tuple, the first for yes
```

Question 3¶

It would be interesting to see if there is any evidence of a link between vaccine effectiveness and sex of the child. Calcu the ratio of the number of children who contracted chickenpox but were vaccinated against it (at least one varicella dos 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 d `0.0077`.

In [5]:

Student's answer

```python
def chickenpox_by_sex():
    # YOUR CODE HERE
    # raise NotImplementedError()
    import pandas as pd
    import numpy as np
    df = pd.read_csv("assets/NISPUF17.csv", index_col=0)

    cpo_sex=df[df['P_NUMVRC'].gt(0) & df['HAD_CPOX'].lt(3)].loc[:,['HAD_CPOX','SEX']]
    #Male 1 Female 2
    cpo1_sex1=len(cpo_sex[(cpo_sex['HAD_CPOX']==1) & (cpo_sex['SEX']==1)])
    cpo1_sex2=len(cpo_sex[(cpo_sex['HAD_CPOX']==1) & (cpo_sex['SEX']==2)])
    cpo2_sex1=len(cpo_sex[(cpo_sex['HAD_CPOX']==2) & (cpo_sex['SEX']==1)])
    cpo2_sex2=len(cpo_sex[(cpo_sex['HAD_CPOX']==2) & (cpo_sex['SEX']==2)])

    cbs={"male":0,
        "female":0}
    cbs['male']=cpo1_sex1/cpo2_sex1
    cbs['female']=cpo1_sex2/cpo2_sex2
    return cbs
chickenpox_by_sex()
```

Out[5]:

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

In [6]:

Grade cell: cell-c4f1714db100c865   Score: 1.0 / 1.0

```python
assert len(chickenpox_by_sex())==2, "Return a dictionary with two items, the first for male
```

Question 4¶

A correlation is a statistical relationship between two variables. If we wanted to know if vaccines work, we might look a 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. The `had_chickenpox_column` is either `1` (for yes) or `2` (for no), and the `num_chickenpox_vaccine_column` is the number of doses a child has been given of the varicella vaccine. A positive correlation (e.g., `corr > 0`) means that an increase in `had_chickenpox_column` (which means more no's) would also increase the values of `num_chickenpox_vaccine_column` (which means more doses of vaccine). If there is a negative correlation (e.g., `corr < 0`), it indicates that having had chickenpox is related to an increase in the number of vaccine doses.

Also, `pval` is the probability that we observe a correlation between `had_chickenpox_column` and `num_chickenpox_vaccine_column` which is greater than or equal to a particular value occurred by chance. A small `pv` means that the observed correlation is highly unlikely to occur by chance. In this case, `pval` should be very small (will in `e-18` indicating a very small number).

[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]:

Student's answer

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

    # this is just an example dataframe
    # df=pd.DataFrame({"had_chickenpox_column":np.random.randint(1,3,size=(100)),
    #             "num_chickenpox_vaccine_column":np.random.randint(0,6,size=(100))})
    df = pd.read_csv("assets/NISPUF17.csv", index_col=0)

    df=df[df['HAD_CPOX'].lt(3)].loc[:,['HAD_CPOX','P_NUMVRC']].dropna()
    df.columns=['had_chickenpox_column','num_chickenpox_vaccine_column']
    # here is some stub code to actually run the correlation
    corr, pval=stats.pearsonr(df["had_chickenpox_column"],df["num_chickenpox_vaccine_column"

    # just return the correlation
    return corr
corr_chickenpox()
```

Out[7]:

```
0.07044873460147989
```

In [8]:

Grade cell: cell-73408733533a29a5   Score: 1.0 / 1.0

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

This assignment was graded by mooc_adswpy:e5e20d3b91dd, v1.46.070623