assignment2

March 14, 2021

1 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`. 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`. **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).**

1.1 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}
```

[5]:

```python
import pandas as pd
def proportion_of_education():
    data = pd.read_csv("assets/NISPUF17.csv")
    total = len(data)
    edu_data = data['EDUC1'].value_counts().to_dict()
    for key in sorted(edu_data):
        # calculate percentage
        edu_data[key] = edu_data[key] / total
    #     print(key, edu_data[key])
    education_level = {
            "less than high school": edu_data[1],
            "high school":edu_data[2],
            "more than high school but not college":edu_data[3],
            "college":edu_data[4]
    }
    return education_level
```

[6]:

```python
print(proportion_of_education())
```

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

[ ]:

```
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 items in it."
assert "less than high school" in proportion_of_education().keys(), "You have␣
  ↪not 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␣
  ↪dictionary with the correct keys."
```

1.2 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)
```

[ ]:

```
import pandas as pd
import numpy as np
def average_influenza_doses():
    df = pd.read_csv("assets/NISPUF17.csv")
    cbf_flu_df = df.loc[:,['CBF_01','P_NUMFLU']] # retrieve rows w/ these 2␣
  ↪cols
    cbf_flu_df_breastfed = cbf_flu_df[cbf_flu_df['CBF_01'] == 1].dropna() #␣
  ↪breastfed
    cbf_flu_df_not_breastfed = cbf_flu_df[cbf_flu_df['CBF_01'] == 2].dropna() #␣
  ↪not breastfed
    cbf_flu_df_breastfed_flu =cbf_flu_df_breastfed['P_NUMFLU'].values.copy()
    cbf_flu_df_breastfed_flu[np.isnan(cbf_flu_df_breastfed_flu)] = 0
    val1=np.sum(cbf_flu_df_breastfed_flu)/len(cbf_flu_df_breastfed_flu)
    cbf_flu_df_not_breastfed_not_flu=cbf_flu_df_not_breastfed['P_NUMFLU'].
  ↪values.copy()
    cbf_flu_df_not_breastfed_not_flu[np.
  ↪isnan(cbf_flu_df_not_breastfed_not_flu)] = 0
    val2=np.sum(cbf_flu_df_not_breastfed_not_flu)/
  ↪len(cbf_flu_df_not_breastfed_not_flu)
    avg_flu_doses =(val1,val2)
    return avg_flu_doses
```

[ ]:
```
print(average_influenza_doses())
```

[ ]:
```
assert len(average_influenza_doses())==2, "Return two values in a tuple, the␣
↪first for yes and the second for no."
```

1.3 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`.

[ ]:
```
def chickenpox_by_sex():
    import pandas as pd
    import numpy as np
    df = pd.read_csv("assets/NISPUF17.csv", index_col=0)
    # had_cpox: yes == 1, no == 2
    # P_NUMVRC: 0, 1, 2, 3, nan
    # sex: male == 1, female == 2
    vaxed_cpox_sex=df[df['P_NUMVRC'].gt(0) & df['HAD_CPOX'].lt(3)].loc[:
↪,['HAD_CPOX','SEX']]
    vaxed_cpox_sex
    # obtain values for calculation
    # ppl who had cpox
    vaxed_had_cpox_male = len(vaxed_cpox_sex[(vaxed_cpox_sex['HAD_CPOX']==1) &␣
↪(vaxed_cpox_sex['SEX']==1)])
    vaxed_had_cpox_female = len(vaxed_cpox_sex[(vaxed_cpox_sex['HAD_CPOX']==1)␣
↪& (vaxed_cpox_sex['SEX']==2)])
    # ppl who did not have cpox
    vaxed_no_cpox_male = len(vaxed_cpox_sex[(vaxed_cpox_sex['HAD_CPOX']==2) &␣
↪(vaxed_cpox_sex['SEX']==1)])
    vaxed_no_cpox_female = len(vaxed_cpox_sex[(vaxed_cpox_sex['HAD_CPOX']==2) &␣
↪(vaxed_cpox_sex['SEX']==2)])
    cpox_sex_dict = {}
    cpox_sex_dict['male'] = vaxed_had_cpox_male/vaxed_no_cpox_male
    cpox_sex_dict['female'] = vaxed_had_cpox_female/vaxed_no_cpox_female
    return cpox_sex_dict
```

[ ]:
```
assert len(chickenpox_by_sex())==2, "Return a dictionary with two items, the␣
↪first for males and the second for females."
```

1.4 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. 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 `pval` means that the observed correlation is highly unlikely to occur by chance. In this case, `pval` should be very small (will end 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?

[ ]:

```
def corr_chickenpox():
    import scipy.stats as stats
    import numpy as np
    import pandas as pd
    # EXAMPLE - START
    # 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))})
    # 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
    # EXAMPLE - END

    # ACTUAL IMPLEMENTATION
    df = pd.read_csv("assets/NISPUF17.csv", index_col=0)
    # extract cpox rows with yes & no, extract all rows, extract only 2␣
 ↪columns, drop all nan values
    df=df[df['HAD_CPOX'].lt(3)].loc[:,['HAD_CPOX','P_NUMVRC']].dropna()
    df.columns=['had_chickenpox_column','num_chickenpox_vaccine_column']
    corr, pval=stats.
 ↪pearsonr(df["had_chickenpox_column"],df["num_chickenpox_vaccine_column"])
    return corr
```

[ ]:

```
print(corr_chickenpox())
```

[ ]:

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

[ ]:

```

```