labassignment10bn

April 8, 2025

1 Lab Assignment 10: Exploratory Data Analysis, Part 1

1.1 DS 6001: Practice and Application of Data Science

1.1.1 Instructions

Please answer the following questions as completely as possible using text, code, and the results of code as needed. Format your answers in a Jupyter notebook. To receive full credit, make sure you address every part of the problem, and make sure your document is formatted in a clean and professional way.

In this lab, you will be working with the 2018 General Social Survey (GSS). The GSS is a sociological survey created and regularly collected since 1972 by the National Opinion Research Center at the University of Chicago. It is funded by the National Science Foundation. The GSS collects information and keeps a historical record of the concerns, experiences, attitudes, and practices of residents of the United States, and it is one of the most important data sources for the social sciences.

The data includes features that measure concepts that are notoriously difficult to ask about directly, such as religion, racism, and sexism. The data also include many different metrics of how successful a person is in his or her profession, including income, socioeconomic status, and occupational prestige. These occupational prestige scores are coded separately by the GSS. The full description of their methodology for measuring prestige is available here: http://gss.norc.org/Documents/reports/methodological-reports/MR122%20Occupational%20Prestige.pdf Here's a quote to give you an idea about how these scores are calculated:

> Respondents then were given small cards which each had a single occupational titles listed on it. Cards were in English or Spanish. They were given one card at a time in the preordained order. The interviewer then asked the respondent to "please put the card in the box at the top of the ladder if you think that occupation has the highest possible social standing. Put it in the box of the bottom of the ladder if you think it has the lowest possible social standing. If it belongs somewhere in between, just put it in the box that matches the social standing of the occupation."

The prestige scores are calculated from the aggregated rankings according to the method described above.

1.1.2 Problem 0

Import the following packages:

[20]:
```
import numpy as np
import pandas as pd
import sidetable
import weighted # this is a module of wquantiles, so type pip install␣
↪wquantiles or conda install wquantiles to get access to it
from scipy import stats
from sklearn import manifold
from sklearn import metrics
import prince
from ydata_profiling import ProfileReport
import matplotlib.pyplot as plt
pd.options.display.max_columns = None
```

Then load the GSS data with the following code:

[3]:
```
%%capture
gss = pd.read_csv("https://github.com/jkropko/DS-6001/raw/master/localdata/
↪gss2018.csv",
                 encoding='cp1252', na_values=['IAP','IAP,DK,NA,uncodeable',␣
↪'NOT SURE',
                                              'DK', 'IAP, DK, NA, uncodeable',␣
↪'.a', "CAN'T CHOOSE"])
```

1.1.3 Problem 1

Drop all columns except for the following: * `id` - a numeric unique ID for each person who responded to the survey * `wtss` - survey sample weights * `sex` - male or female * `educ` - years of formal education * `region` - region of the country where the respondent lives * `age` - age * `coninc` - the respondent's personal annual income * `prestg10` - the respondent's occupational prestige score, as measured by the GSS using the methodology described above * `mapres10` - the respondent's mother's occupational prestige score, as measured by the GSS using the methodology described above * `papres10` -the respondent's father's occupational prestige score, as measured by the GSS using the methodology described above * `sei10` - an index measuring the respondent's socioeconomic status * `satjob` - responses to "On the whole, how satisfied are you with the work you do?" * `fechld` - agree or disagree with: "A working mother can establish just as warm and secure a relationship with her children as a mother who does not work." * `fefam` - agree or disagree with: "It is much better for everyone involved if the man is the achiever outside the home and the woman takes care of the home and family." * `fepol` - agree or disagree with: "Most men are better suited emotionally for politics than are most women." * `fepresch` - agree or disagree with: "A preschool child is likely to suffer if his or her mother works." * `meovrwrk` - agree or disagree with: "Family life often suffers because men concentrate too much on their work."

Then rename any columns with names that are non-intuitive to you to more intuitive and descriptive ones. Finally, replace the "89 or older" values of `age` with 89, and convert `age` to a float data type. [1 point]

[4]:
```
```

```
gss = gss[['id', 'wtss', 'sex', 'educ', 'region', 'age', 'coninc', 'prestg10',␣
 ↪'mapres10', 'papres10', 'sei10', 'satjob', 'fechld', 'fefam', 'fepol',␣
 ↪'fepresch', 'meovrwrk']]
gss = gss.rename({'wtss': 'weights',
                  'coninc': 'income',
                  'prestg10': 'prestige',
                  'mapres10': 'mom_prestige',
                  'papres10': 'dad_prestige',
                  'sei10': 'social_econ_status'}, axis=1)
gss.age = gss.age.replace({'89 or older': 89})
gss.age = gss.age.astype(float)
gss
```

[4]:

	id	weights	sex	educ	region	age	income	prestige	\
0	1	2.357493	male	14.0	new england	43.0	NaN	47.0	
1	2	0.942997	female	10.0	new england	74.0	22782.5000	22.0	
2	3	0.942997	male	16.0	new england	42.0	112160.0000	61.0	
3	4	0.942997	female	16.0	new england	63.0	158201.8412	59.0	
4	5	0.942997	male	18.0	new england	71.0	158201.8412	53.0	
…	…	…	…	…	…	…	…	…	
2343	2344	0.471499	female	12.0	new england	37.0	NaN	47.0	
2344	2345	0.942997	female	12.0	new england	75.0	22782.5000	28.0	
2345	2346	0.942997	female	12.0	new england	67.0	70100.0000	40.0	
2346	2347	0.942997	male	16.0	new england	72.0	38555.0000	47.0	
2347	2348	0.471499	female	12.0	new england	79.0	NaN	33.0	

	mom_prestige	dad_prestige	social_econ_status	satjob	\
0	31.0	45.0	65.3	very satisfied	
1	32.0	39.0	14.8	NaN	
2	32.0	72.0	83.4	mod. satisfied	
3	NaN	39.0	69.3	very satisfied	
4	35.0	45.0	68.6	NaN	
…	…	…	…	…	
2343	31.0	72.0	38.8	mod. satisfied	
2344	NaN	27.0	21.6	very satisfied	
2345	45.0	53.0	41.8	NaN	
2346	53.0	50.0	62.7	NaN	
2347	NaN	46.0	13.6	very satisfied	

	fechld	fefam	fepol	fepresch	\
0	strongly agree	disagree	agree	strongly disagree	
1	NaN	NaN	NaN	NaN	
2	strongly agree	disagree	disagree	disagree	
3	agree	disagree	disagree	disagree	
4	NaN	NaN	NaN	NaN	
…	…	…	…	…	
2343	disagree	strongly disagree	disagree	strongly disagree	

```
2344     strongly agree           disagree  disagree            disagree
2345                NaN                NaN       NaN                 NaN
2346           disagree              agree  disagree      strongly agree
2347  strongly disagree     strongly agree  disagree      strongly agree

                       meovrwrk
0                         agree
1                           NaN
2                      disagree
3     neither agree nor disagree
4                           NaN
…                           …
2343                   disagree
2344                   disagree
2345                        NaN
2346                      agree
2347             strongly agree

[2348 rows x 17 columns]
```

1.1.4 Problem 2

Part a Use the `ProfileReport()` function to generate and embed an HTML formatted exploratory data analysis report in your notebook. Make sure that it includes a "Correlations" report along with "Overview" and "Variables". [1 point]

[5]:
```
profile = ProfileReport(gss,
                        title = "2018 General Social Survey Report",
                        html = {'style': {'full_width': True}},
                        minimal = False)
profile.to_notebook_iframe()
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

100%|      | 17/17 [00:00<00:00, 176.67it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

Part b Looking through the HTML report you displayed in part a, how many people in the data are from New England? [1 point]

124 people in the data are from New England.

Part c Looking through the HTML report you displayed in part a, which feature in the data has the highest number of missing values, and what percent of the values are missing for this feature? [1 point]

The variable fepol (attitude about women in politics) has the most missing values. About 36% of the values are missing.

Part d Looking through the HTML report you displayed in part a, which two distinct features in the data have the highest correlation? [1 point]

Prestige and Socio Economic Status have the highest correlation at .824.

1.1.5 Problem 3

On a primetime show on a 24-hour cable news network, two unpleasant-looking men in suits sit across a table from each other, scowling. One says "This economy is failing the middle-class. The average American today is making less than \$48,000 a year." The other screams "Fake news! The typical American makes more than $55,000 a year!" Explain, using words and code, how the data can support both of their arguments. Use the sample weights to calculate descriptive statistics that are more representative of the American adult population as a whole. [1 point]

[6]:
```
print(gss.income.median())
print(weighted.median(gss.income, gss.weights))
print(gss.income.mean())
gss_temp = gss.loc[~gss.income.isna()]
print(np.average(gss_temp.income, weights=gss_temp.weights))
```

```
38555.0
47317.5
49973.96077843866
55158.96280421564
```

In the code above I have found the median, weighted median, mean, and weighted mean, all which have different numbers which someone could use to explain how much the average American makes. The two numbers in question here are the weighted median and the weighted mean. Since the mean is greater than the median, this distribution is skewed to the right, which makes sense because there will be some individuals who make way more than the rest skewing the mean. However, both are measures of center which someone could use to bamboozle unsuspecting victims.

1.1.6 Problem 4

For each of the following parts, * generate a table that provides evidence about the relationship between the two features in the data that are relevant to each question, * interpret the table in words, * use a hypothesis test to assess the strength of the evidence in the table, * and provide a **specific and accurate** intepretation of the p-value associated with this hypothesis test beyond "significant or not".

Part a Is there a gender wage gap? That is, is there a difference between the average incomes of men and women? [2 points]

[7]:
```
gss.groupby('sex').agg({'income': 'mean'})
```

[7]:
```
              income
sex
```

```
female   47191.021452
male     53314.626187
```

[8]:
```
income_men = gss.query('sex == "male"').income.dropna()
income_women = gss.query('sex == "female"').income.dropna()
stats.ttest_ind(income_men, income_women, equal_var=False)
```

[8]:
```
TtestResult(statistic=np.float64(3.332824087618215),
pvalue=np.float64(0.0008749557881530089), df=np.float64(2053.1579577339658))
```

According to the data men make $53,314.63 and women make $47,191.02 on average.

This reslust is statistically significant with a p-value of 0.0009 That means assuming gender pay is equal, the probability we got the pay difference we did or more extreme is 0.0009. This means we reject the idea that pay is equal and conclude there is sufficient evidence for a gender pay gap. (This is how we teach it in AP Stats (or at least similar since our setups are different) so please let me know if this is not correct.)

Part b Are there different average values of occupational prestige for different levels of job satisfaction? [2 points]

[9]:
```
gss.groupby('satjob').agg({'prestige': 'mean'})
```

[9]:
```
                    prestige
satjob
a little dissat    40.946429
mod. satisfied     42.589984
very dissatisfied  43.000000
very satisfied     46.189320
```

[10]:
```
stats.f_oneway(gss.query('satjob == "very satisfied"').prestige.dropna(),
             gss.query('satjob == "mod. satisfied"').prestige.dropna(),
             gss.query('satjob == "a little dissat"').prestige.dropna(),
             gss.query('satjob == "very dissatisfied"').prestige.dropna())
```

[10]:
```
F_onewayResult(statistic=np.float64(12.205403153509735),
pvalue=np.float64(6.676686425029878e-08))
```

According to the data people who are the most satisfied with their job have the most job prestige. This is followed by people who are the least satisfied with thier job and then those who are moderately satisfied with their job. The people who are a little dissatisfied with their job have the least prestige onoaverage.

The diffence in result is statistically significant with a p-value of 0.00000007. This means that assuming there is no difference in prestige between job satisfaction levels, the probability we observed the difference we did or more extreme is 0.00000007. This means we reject the idea that prestige is the same over job satisfaction because we have sufficient evidence to suggest that prestige level is different across job satisfaction.

1.1.7 Problem 5

Report the Pearson's correlation between years of education, socioeconomic status, income, occupational prestige, and a person's mother's and father's occupational prestige? Then perform a hypothesis test for the correlation between years of education and socioeconomic status and provide a **specific and accurate** intepretation of the *p*-value associated with this hypothesis test beyond "significant or not". [2 points]

[11]:
```python
gss[['educ','social_econ_status', 'income', 'prestige', 'mom_prestige',␣
 ↪'dad_prestige']].corr()
```

[11]:
```
                        educ  social_econ_status    income  prestige  \
educ                1.000000            0.558169  0.389245  0.479933
social_econ_status  0.558169            1.000000  0.417210  0.835515
income              0.389245            0.417210  1.000000  0.340995
prestige            0.479933            0.835515  0.340995  1.000000
mom_prestige        0.269115            0.203486  0.164881  0.189262
dad_prestige        0.261417            0.210451  0.171048  0.192180

                    mom_prestige  dad_prestige
educ                    0.269115      0.261417
social_econ_status      0.203486      0.210451
income                  0.164881      0.171048
prestige                0.189262      0.192180
mom_prestige            1.000000      0.235750
dad_prestige            0.235750      1.000000
```

[12]:
```python
gss_corr = gss[['educ','social_econ_status',]].dropna()
stats.pearsonr(gss_corr['educ'], gss_corr['social_econ_status'])
```

[12]:
```
PearsonRResult(statistic=np.float64(0.5581686004626785),
pvalue=np.float64(3.7194488100285224e-184))
```

The correlation between education level and socioeconomic status is 0.56. With a p-value of practically 0, we conclude that a random sample could not have create a sample with a correlation as extreme as .56 so we reject the idea that the two values are uncorrelated and say we have convincing evidence that education level and socioeconomic status have a nonzero correlation.

1.1.8 Problem 6

Create a new categorical feature for age groups, with categories for 18-35, 36-49, 50-69, and 70 and older (see the module 8 notebook for an example of how to do this).

Then create a cross-tabulation in which the rows represent age groups and the columns represent responses to the statement that "It is much better for everyone involved if the man is the achiever outside the home and the woman takes care of the home and family." Rearrange the columns so that they are in the following order: strongly agree, agree, disagree, strongly disagree. Place row percents in the cells of this table.

Finally, use a hypothesis test that can tell use whether there is enough evidence to conclude that these two features have a relationship, and provide a specific and accurate intepretation of the *p*-value. [2 points]

[13]:
```
gss = gss.assign(age_group =
                 pd.cut(gss.age,
                        bins=[17,35,49,69,900],
                        labels=("18-35","36-49","50-69","70+")))
fam = ['strongly agree', 'agree', 'disagree', 'strongly disagree']
q6 = 100*pd.crosstab(gss.age_group, gss.fefam, normalize='columns').round(2)
q6[fam]
```

[13]:

fefam age_group	strongly agree	agree	disagree	strongly disagree
18-35	18.0	18.0	27.0	32.0
36-49	19.0	20.0	23.0	26.0
50-69	27.0	35.0	35.0	32.0
70+	35.0	27.0	14.0	10.0

[14]:
```
stats.chi2_contingency(q6.values)
```

[14]:
```
Chi2ContingencyResult(statistic=np.float64(27.433357868586086),
pvalue=np.float64(0.0011855043136876121), dof=9,
expected_freq=array([[23.86934673, 23.63065327, 23.63065327, 23.86934673],
       [22.11055276, 21.88944724, 21.88944724, 22.11055276],
       [32.4120603 , 32.0879397 , 32.0879397 , 32.4120603 ],
       [21.6080402 , 21.3919598 , 21.3919598 , 21.6080402 ]]))
```

Since our p-value is very low (0.001) we reject the null hypothesis. We have sufficient evidence age group and female staying home with the family are not independent. (This is closer to how we conclude in AP Stat…)

1.1.9 Problem 7

For this problem, you will conduct and interpret a correspondence analysis on the categorical features that ask respondents to state the extent to which they agree or disagree with the statements: * "A working mother can establish just as warm and secure a relationship with her children as a mother who does not work." * "It is much better for everyone involved if the man is the achiever outside the home and the woman takes care of the home and family." * "Most men are better suited emotionally for politics than are most women." * "A preschool child is likely to suffer if his or her mother works." * "Family life often suffers because men concentrate too much on their work."

Part a Conduct a correspondence analysis using the observed features listed above that measures two latent features. Plot the two latent categories for each category in each of the features used in the analysis. [2 points]

[15]:
```
q7 = gss[['fechld', 'fefam', 'fepol', 'fepresch', 'meovrwrk']].dropna()
mca = prince.MCA(n_components=2)
mca = mca.fit(q7)
```

[16]:
```
mca.row_coordinates(q7)
```

[16]:
```
             0         1
0    -0.202210  0.338292
2    -0.423360 -0.316910
3    -0.195576 -0.648698
5    -0.240091 -0.298100
8     0.341539  0.091194
…          …         …
2341  1.219021  0.567430
2343 -0.521778  0.384977
2344 -0.423360 -0.316910
2346  1.076899  0.642132
2347  1.440616  2.529636

[1454 rows x 2 columns]
```

Part b Display the latent features for every category in the observed features, sorted by the first latent feature. Describe in words what concept this feature is attempting to measure, and give the feature a name. [2 points]

[17]:
```
mca.column_coordinates(q7).sort_values(0)
```

[17]:
```
                                            0         1
fepresch__strongly disagree         -1.258061  0.886702
meovrwrk__strongly disagree         -1.135402  1.283824
fefam__strongly disagree            -0.922036  0.566817
fechld__strongly agree              -0.901117  0.472172
meovrwrk__neither agree nor disagree -0.480747 -0.163823
meovrwrk__disagree                  -0.228691 -0.242579
fepol__disagree                     -0.180399 -0.063738
fepresch__disagree                  -0.067885 -0.529264
fefam__disagree                      0.022158 -0.572465
fechld__agree                        0.080483 -0.586391
meovrwrk__agree                      0.358280 -0.187029
meovrwrk__strongly agree             0.536781  1.291999
fefam__agree                         0.878987 -0.076597
fechld__disagree                     0.918040 -0.010320
fepresch__agree                      0.919992 -0.036424
fepol__agree                         1.131104  0.399637
fechld__strongly disagree            1.218704  2.005412
fepresch__strongly agree             1.474177  2.233976
fefam__strongly agree                1.564731  2.002663
```

The first latent feature clearly has to do with level of agreement. The feature is sorted from strongly disagree to strongly agree (with a few exceptions).

Part c We can use the results of the MCA model to conduct some cool EDA. For one example, follow these steps:

1. Use the `.row_coordinates()` method to calculate values of the latent feature for every row in the data you passed to the MCA in part a. Extract the first column and store it in its own dataframe.

2. To join it with the full, cleaned GSS data based on row numbers (instead of on a primary key), use the `.join()` method. For example, if we named the cleaned GSS data `gss_clean` and if we named the dataframe in step 1 `latentfeature`, we can type

```
gss_clean = gss_clean.join(latentfeature, how="outer")
```

3. Create a cross-tabuation with age categories (that you constructed in problem 5) in the rows and sex in the columns. Instead of a frequency, place the mean value of the latent feature in the cells.

What does this table tell you about the relationship between sex, age, and the latent feature? [2 points]

[18]:
```python
mca_rows = mca.row_coordinates(q7)
row0 = mca_rows[0]
gss.join(row0, how="outer")
```

[18]:
```
        id   weights     sex  educ       region   age        income  prestige  \
0        1  2.357493    male  14.0  new england  43.0           NaN      47.0
1        2  0.942997  female  10.0  new england  74.0    22782.5000      22.0
2        3  0.942997    male  16.0  new england  42.0   112160.0000      61.0
3        4  0.942997  female  16.0  new england  63.0   158201.8412      59.0
4        5  0.942997    male  18.0  new england  71.0   158201.8412      53.0
…      …         …       …     …          …      …             …
2343  2344  0.471499  female  12.0  new england  37.0           NaN      47.0
2344  2345  0.942997  female  12.0  new england  75.0    22782.5000      28.0
2345  2346  0.942997  female  12.0  new england  67.0    70100.0000      40.0
2346  2347  0.942997    male  16.0  new england  72.0    38555.0000      47.0
2347  2348  0.471499  female  12.0  new england  79.0           NaN      33.0

      mom_prestige  dad_prestige  social_econ_status          satjob  \
0             31.0          45.0                65.3  very satisfied
1             32.0          39.0                14.8             NaN
2             32.0          72.0                83.4  mod. satisfied
3              NaN          39.0                69.3  very satisfied
4             35.0          45.0                68.6             NaN
…              …             …                   …               …
2343          31.0          72.0                38.8  mod. satisfied
2344           NaN          27.0                21.6  very satisfied
2345          45.0          53.0                41.8             NaN
2346          53.0          50.0                62.7             NaN
2347           NaN          46.0                13.6  very satisfied
```

```
                 fechld             fefam     fepol           fepresch  \
0        strongly agree          disagree     agree  strongly disagree
1                   NaN               NaN       NaN                NaN
2        strongly agree          disagree  disagree           disagree
3                 agree          disagree  disagree           disagree
4                   NaN               NaN       NaN                NaN
…                     …                 …         …                  …
2343           disagree  strongly disagree  disagree  strongly disagree
2344     strongly agree          disagree  disagree           disagree
2345                NaN               NaN       NaN                NaN
2346           disagree             agree  disagree     strongly agree
2347  strongly disagree    strongly agree  disagree     strongly agree

                        meovrwrk age_group         0
0                          agree     36-49 -0.202210
1                            NaN       70+       NaN
2                       disagree     36-49 -0.423360
3     neither agree nor disagree     50-69 -0.195576
4                            NaN       70+       NaN
…                              …         …         …
2343                    disagree     36-49 -0.521778
2344                    disagree       70+ -0.423360
2345                         NaN     50-69       NaN
2346                       agree       70+  1.076899
2347              strongly agree       70+  1.440616

[2348 rows x 19 columns]
```

[19]:

```
pd.crosstab(gss.age_group, gss.sex, values=row0, aggfunc='mean').round(2)
```

[19]:

```
sex        female  male
age_group
18-35       -0.24 -0.00
36-49       -0.14 -0.00
50-69       -0.13  0.22
70+          0.13  0.47
```

This tells me that as men men get older the more likely they are to agree, tho they sart being indifferent. Most women disagree until they get to be really old.