labassignment12

August 14, 2022

1 Lab Assignment 12: Interactive Visualizations

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.

1.2 Problem 0

Import the following libraries:

[1]:
```
import numpy as np
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
import dash
from jupyter_dash import JupyterDash
from dash import dcc
from dash import html
from dash.dependencies import Input, Output
external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
```

For this lab, we will be working with the 2019 General Social Survey one last time.

[2]:
```
%%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"])
import os
```

```
os.chdir('C:/Users/mk7kc/Documents/GitHub/DS6100-2022/Assignments/
 ↪labAssignment_12')
# gss.to_csv("gss.csv", sep=",",index=False)
gss = pd.read_csv('gss.csv')
```

Here is code that cleans the data and gets it ready to be used for data visualizations:

[3]:
```
mycols = ['id', 'wtss', 'sex', 'educ', 'region', 'age', 'coninc',
          'prestg10', 'mapres10', 'papres10', 'sei10', 'satjob',
          'fechld', 'fefam', 'fepol', 'fepresch', 'meovrwrk']
gss_clean = gss[mycols]
gss_clean = gss_clean.rename({'wtss':'weight',
                              'educ':'education',
                              'coninc':'income',
                              'prestg10':'job_prestige',
                              'mapres10':'mother_job_prestige',
                              'papres10':'father_job_prestige',
                              'sei10':'socioeconomic_index',
                              'fechld':'relationship',
                              'fefam':'male_breadwinner',
                              'fehire':'hire_women',
                              'fejobaff':'preference_hire_women',
                              'fepol':'men_bettersuited',
                              'fepresch':'child_suffer',
                              'meovrwrk':'men_overwork'},axis=1)
gss_clean.age = gss_clean.age.replace({'89 or older':'89'})
gss_clean.age = gss_clean.age.astype('float')
gss_clean = gss_clean.replace({'sex':{'female':'Female','male':'Male'}})
gss_clean
```

[3]:

	id	weight	sex	education	region	age	income	\
0	1	2.357493	Male	14.0	new england	43.0	NaN	
1	2	0.942997	Female	10.0	new england	74.0	22782.5000	
2	3	0.942997	Male	16.0	new england	42.0	112160.0000	
3	4	0.942997	Female	16.0	new england	63.0	158201.8412	
4	5	0.942997	Male	18.0	new england	71.0	158201.8412	
…	…	…	…	…	…	…	…	
2343	2344	0.471499	Female	12.0	new england	37.0	NaN	
2344	2345	0.942997	Female	12.0	new england	75.0	22782.5000	
2345	2346	0.942997	Female	12.0	new england	67.0	70100.0000	
2346	2347	0.942997	Male	16.0	new england	72.0	38555.0000	
2347	2348	0.471499	Female	12.0	new england	79.0	NaN	

	job_prestige	mother_job_prestige	father_job_prestige	\
0	47.0	31.0	45.0	
1	22.0	32.0	39.0	
2	61.0	32.0	72.0	

```
3            59.0                  NaN                 39.0
4            53.0                 35.0                 45.0
…             …                    …                    …
2343         47.0                 31.0                 72.0
2344         28.0                  NaN                 27.0
2345         40.0                 45.0                 53.0
2346         47.0                 53.0                 50.0
2347         33.0                  NaN                 46.0

      socioeconomic_index          satjob       relationship  \
0                    65.3  very satisfied     strongly agree
1                    14.8             NaN                NaN
2                    83.4  mod. satisfied     strongly agree
3                    69.3  very satisfied              agree
4                    68.6             NaN                NaN
…                     …              …                   …
2343                 38.8  mod. satisfied           disagree
2344                 21.6  very satisfied     strongly agree
2345                 41.8             NaN                NaN
2346                 62.7             NaN           disagree
2347                 13.6  very satisfied  strongly disagree

       male_breadwinner men_bettersuited       child_suffer  \
0              disagree            agree  strongly disagree
1                   NaN              NaN                NaN
2              disagree         disagree           disagree
3              disagree         disagree           disagree
4                   NaN              NaN                NaN
…                    …                …                  …
2343  strongly disagree         disagree  strongly disagree
2344           disagree         disagree           disagree
2345                NaN              NaN                NaN
2346              agree         disagree     strongly agree
2347     strongly agree         disagree     strongly agree

                    men_overwork
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]
```

The `gss_clean` dataframe now contains the following features:

- `id` - a numeric unique ID for each person who responded to the survey
- `weight` - survey sample weights
- `sex` - male or female
- `education` - years of formal education
- `region` - region of the country where the respondent lives
- `age` - age
- `income` - the respondent's personal annual income
- `job_prestige` - the respondent's occupational prestige score, as measured by the GSS using the methodology described above
- `mother_job_prestige` - the respondent's mother's occupational prestige score, as measured by the GSS using the methodology described above
- `father_job_prestige` -the respondent's father's occupational prestige score, as measured by the GSS using the methodology described above
- `socioeconomic_index` - an index measuring the respondent's socioeconomic status
- `satjob` - responses to "On the whole, how satisfied are you with the work you do?"
- `relationship` - 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."
- `male_breadwinner` - 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."
- `men_bettersuited` - agree or disagree with: "Most men are better suited emotionally for politics than are most women."
- `child_suffer` - agree or disagree with: "A preschool child is likely to suffer if his or her mother works."
- `men_overwork` - agree or disagree with: "Family life often suffers because men concentrate too much on their work."

1.3 Problem 1

Our goal in this lab is to build a dashboard that presents our findings from the GSS. A dashboard is meant to be shared with an audience, whether that audience is a manager, a client, a potential employer, or the general public. So we need to provide context for our results. One way to provide context is to write text using markdown code.

Find one or two websites that discuss the gender wage gap, and write a short paragraph in markdown code summarizing what these sources tell us. Include hyperlinks to these websites. Then write another short paragraph describing what the GSS is, what the data contain, how it was collected, and/or other information that you think your audience ought to know. A good starting point for information about the GSS is here: http://www.gss.norc.org/About-The-GSS

Then save the text as a Python string so that you can use the markdown code in your dashboard later.

It should go without saying, but no plagiarization! If you summarize a website, make sure you put the summary in your own words. Anything that is copied and pasted from the GSS webpage, Wikipedia, or another website without attribution will receive no credit.

(Don't spend too much time on this, and you might want to skip it during the Zoom session and return to it later so that you can focus on working on code with your classmates.) [1 point]

[4]:
```
markdown_text = '''
The most recent [2020 Census Bureau data](https://www.census.gov/data/tables/
 ↪time-series/demo/income-poverty/cps-pinc/pinc-05.html) showed that women␣
 ↪earned 83 cents for every $1 earned by men. This dashboard provides␣
 ↪descriptive statistics on sex differences using data from the 2018 General␣
 ↪Social Survey.

The [2018 General Social Survey](https://gss.norc.org/About-The-GSS) is a␣
 ↪personal-interview survey conducted by the National Opinion Research Center␣
 ↪at the University of Chicago. For the 2018 survey, 2,348 interviews were␣
 ↪conducted via full probability-sampling, and the median length of the␣
 ↪interviews was about one and a half hours. Demographic data and data related␣
 ↪to one's profession were collected. For the purpose of this dashboard a␣
 ↪subset of variables were selected, one of them being job prestige. Job␣
 ↪prestige was measured by having respondents rank different jobs in a␣
 ↪hierarchical ladder and aggregating the rankings. The full description of␣
 ↪their methodology can be found in their [report](http://gss.norc.org/
 ↪Documents/reports/methodological-reports/MR122%20Occupational%20Prestige.
 ↪pdf). Other variables visualized in this dashboard inclue annual income,␣
 ↪sex, years of education and socioeconomic index.
'''
```

1.4 Problem 2

Generate a table that shows the mean income, occupational prestige, socioeconomic index, and years of education for men and for women. Use a function from a `plotly` module to display a web-enabled version of this table. This table is for presentation purposes, so round every column to two decimal places and use more presentable column names. [3 points]

[5]:
```
table_draft = gss_clean.groupby('sex').agg({'job_prestige':'mean','income':
 ↪'mean','socioeconomic_index':'mean','education':'mean'}).reset_index()
table_draft = table_draft.rename({'job_prestige':'Average Job Prestige',
                                  'income':'Average Income',
                                  'socioeconomic_index':'Average Socioeconomic␣
 ↪Index',
                                  'education':"Average Education",
                                   'sex': "Sex"
                                  },axis=1)
table = ff.create_table(round(table_draft,2))
table.show()
```

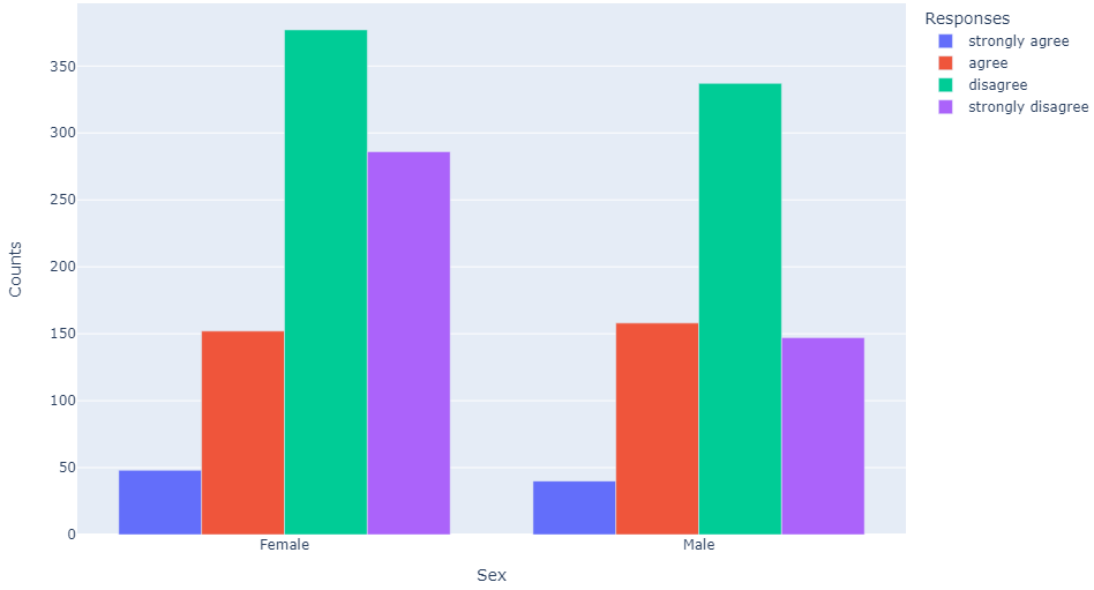
Sex	Average Job Prestige	Average Income	Average Socioeconomic Inde	Average Education
Female	44.67	47191.02	46.58	13.76
Male	44.7	53314.63	47.38	13.69

1.5 Problem 3

Create an interactive barplot that shows the number of men and women who respond with each level of agreement to `male_breadwinner`. Write presentable labels for the x and y-axes, but don't bother with a title because we will be using a subtitle on the dashboard for this graphic. [3 points]

[6]:
```
tab = pd.crosstab(gss_clean.sex,gss_clean.male_breadwinner).reset_index()
long = pd.melt(tab,id_vars=['sex'],
               value_vars=['agree','disagree','strongly agree','strongly␣
 ↪disagree']).rename({'male_breadwinner': 'Responses'},axis=1)
long['Responses'] =  long.Responses.astype('category').cat.
 ↪reorder_categories(['strongly agree',
                                                                        ␣
 ↪              'agree',
                                                                        ␣
 ↪              'disagree',
                                                                        ␣
 ↪              'strongly disagree'])
long = long.sort_values('Responses') # need to sort by Responses so that in the␣
 ↪graph they are listed in the order I want

fig = px.bar(long, x='sex',y='value',color='Responses',barmode='group',
       labels={'sex':'Sex','value':'Counts'},
             height=600, width = 800
      )
fig.show()
```



1.6 Problem 4

Create an interactive scatterplot with `job_prestige` on the x-axis and `income` on the y-axis. Color code the points by `sex` and make sure that the figure includes a legend for these colors. Also include two best-fit lines, one for men and one for women. Finally, include hover data that shows us the values of `education` and `socioeconomic_index` for any point the mouse hovers over. Write presentable labels for the x and y-axes, but don't bother with a title because we will be using a subtitle on the dashboard for this graphic. [3 points]

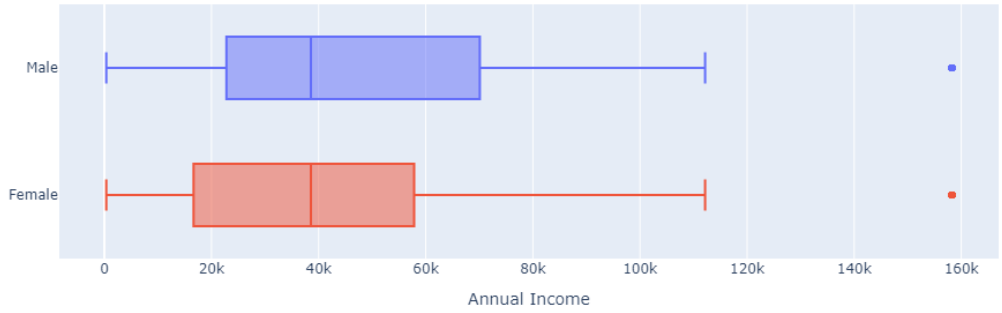
[7]:
```
fig2 = px.scatter(gss_clean, x='job_prestige', y='income',
                  color = 'sex',
                  trendline='ols',
                  height=600, width=700,
                  labels={'job_prestige':'Job Prestige',
                          'income':'Annual Income'},
                  hover_data=['education','socioeconomic_index'])
fig2.show()
```

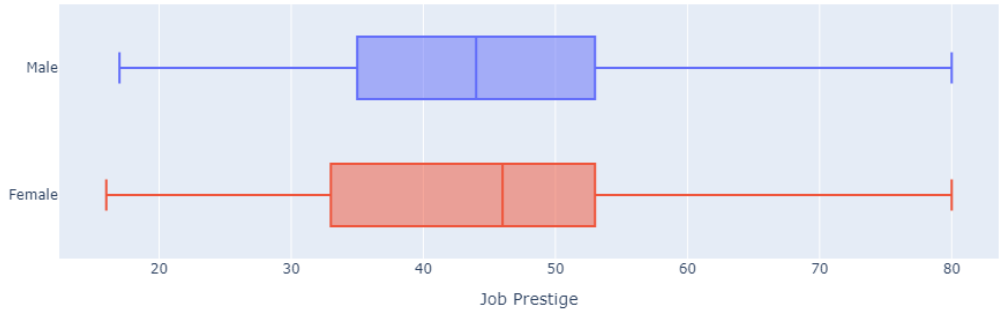
1.7 Problem 5

Create two interactive box plots: one that shows the distribution of `income` for men and for women, and one that shows the distribution of `job_prestige` for men and for women. Write presentable labels for the axis that contains `income` or `job_prestige` and remove the label for `sex`. Also, turn off the legend. Don't bother with titles because we will be using subtitles on the dashboard for these graphics. [3 points]

[8]:
```
fig3 = px.box(gss_clean, x='income', y = 'sex', color = 'sex',
                    labels={'income':'Annual Income', 'sex':''})
fig3.update_layout(showlegend=False)
fig3.show()

fig4 = px.box(gss_clean, x='job_prestige', y = 'sex', color = 'sex',
                    labels={'job_prestige':'Job Prestige', 'sex':''})
fig4.update_layout(showlegend=False)
fig4.show()
```





1.8 Problem 6

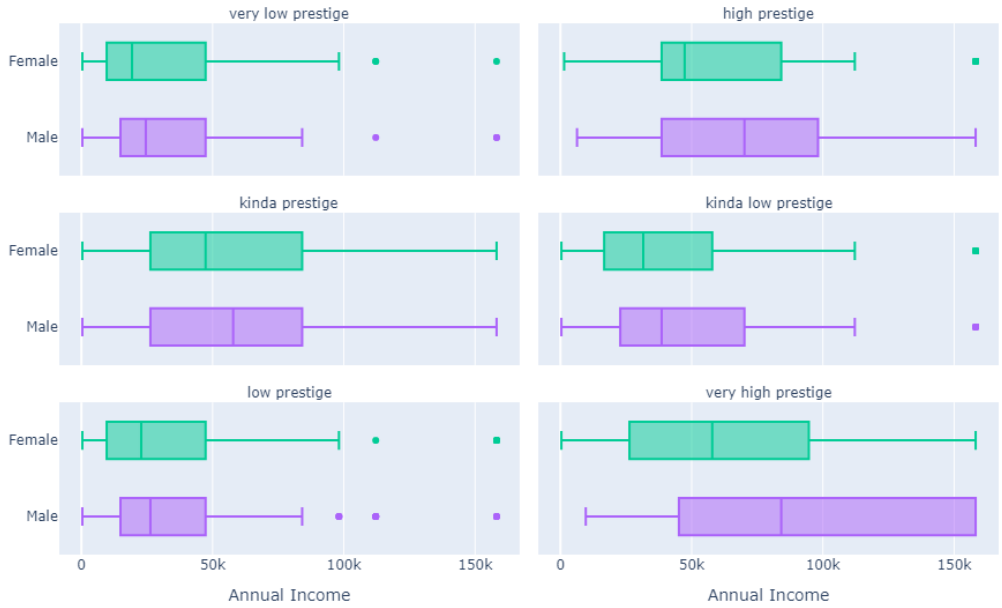
Create a new dataframe that contains only `income`, `sex`, and `job_prestige`. Then create a new feature in this dataframe that breaks `job_prestige` into six categories with equally sized ranges. Finally, drop all rows with any missing values in this dataframe.

Then create a facet grid with three rows and two columns in which each cell contains an interactive box plot comparing the income distributions of men and women for each of these new categories.

(If you want men to be represented by blue and women by red, you can include `color_discrete_map = {'male':'blue', 'female':'red'}` in your plotting function. Or use different colors if you want!) [3 points]

[9]:
```
new = gss_clean[['income','sex','job_prestige']]
new['prestige_cat'] = pd.cut(gss_clean.job_prestige,6,labels=('very low␣
 ↪prestige','low prestige','kinda low prestige','kinda prestige','high␣
 ↪prestige','very high prestige'))
new = new.dropna()
fig5 = px.box(new, x='income', y='sex', color='sex',
```

```
            facet_col='prestige_cat',facet_col_wrap=2,
            width=900,
            height=600,
            color_discrete_map={'female':'green','male':'purple'},
            labels={'income':'Annual Income', 'sex':''})
fig5.update_layout(showlegend=False)
fig5.for_each_annotation(lambda a: a.update(text=a.text.
 ↪replace("prestige_cat=", "")))
fig5.show()
```



1.9 Problem 7

Create a dashboard that displays the following elements:

- A descriptive title
- The markdown text you wrote in problem 1
- The table you made in problem 2
- The barplot you made in problem 3
- The scatterplot you made in problem 4
- The two boxplots you made in problem 5 side-by-side
- The faceted boxplots you made in problem 6
- Subtitles for all of the above elements

Use `JupyterDash` to display this dashboard directly in your Jupyter notebook.

Any working dashboard that displays all of the above elements will receive full credit. [4 points]

[10]:
```
external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
app = JupyterDash(__name__, assets_ignore='style.css',␣
 ↪external_stylesheets=external_stylesheets)

app.layout = html.Div(
    [
        html.H1("Exploring the 2018 General Social Survey"),

        dcc.Markdown(children = markdown_text),

        html.H2("Mean Income, Job Prestige, Socioeconomic Index, and Years of␣
 ↪Education by Sex"),

        dcc.Graph(figure=table),

        html.H2("Responses to '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?' by Sex"),

        dcc.Graph(figure=fig),

        html.H2("Relationship between Job Prestige and Income by Sex"),

        dcc.Graph(figure=fig2),

        html.Div([

            html.H2("Income Distribution by Sex"),

            dcc.Graph(figure=fig3)

        ], style = {'width':'48%', 'float':'left'}),
        html.Div([

            html.H2("Job Prestige by Sex"),

            dcc.Graph(figure=fig4)

        ], style = {'width':'48%', 'float':'right'}),

        html.H2("Income Distribution and Job Prestige by Sex"),

        dcc.Graph(figure=fig5)
```

```
    ]
)


if __name__ == '__main__':
    app.run_server(mode='inline', debug=True, port=8099)
```

```
<IPython.lib.display.IFrame at 0x24387d0aeb0>
```

1.10 Extra Credit (up to 10 bonus points)

Dashboards are all about good design, functionality, and accessability. For this extra credit problem, create another version of the dashboard you built for problem 7, but take extra steps to improve the appearance of the dashboard, add user-inputs, and host it on the internet with its own URL.

Challenge 1: Be creative and use a layout that significantly departs from the one used for the ANES data in the module 12 notebook. A good place to look for inspiration is the Dash gallery. We will award up to 3 bonus points for creativity, novelty, and style.

Challenge 2: Alter the barplot from problem 3 to include user inputs. Create two dropdown menus on the dashboard. The first one should allow a user to display bars for the categories of `satjob`, `relationship`, `male_breadwinner`, `men_bettersuited`, `child_suffer`, or `men_overwork`. The second one should allow a user to group the bars by `sex`, `region`, or `education`. After choosing a feature for the bars and one for the grouping, program the barplot to update automatically to display the user-inputted features. One bonus point will be awarded for a good effort, and 3 bonus points will be awarded for a working user-input barplot in the dashboard.

Challenge 3: Follow the steps listed in the module notebook to deploy your dashboard on Heroku. 1 bonus point will be awarded for a Heroku link to an app that isn't working. 4 bonus points will be awarded for a working Heroku link.

2 Heroku Link: https://gssdashboard.herokuapp.com/

[11]:
```
# Resources for styling/formatting
#import dash_bootstrap_components as dbc
#app = dash.Dash(external_stylesheets=[dbc.themes.CYBORG])
#https://dash-bootstrap-components.opensource.faculty.ai/docs/themes/explorer/
#https://hellodash.pythonanywhere.com/
```

[ ]:
```

```