Introduction to Data Science

Importing Recoding and Visualizing Data.

Joanna Bieri DATA101

Important Information

- Email: joanna_bieri@redlands.edu
- Office Hours: Duke 209 Click Here for Joanna's Schedule

Announcements

In about TWO WEEKS - Data Ethics We will start looking for resources.

Data Science Visualization - from start to finish

Survey of religious traditions and income.

Source: pewforum.org/religious-landscape-study/income-distribution, Retrieved 14 April, 2020

The Data

	Religious tradition	Less than $30,000	$30,000-$49,999
0	Buddhist	0.36	0.18
1	Catholic	0.36	0.19
2	Evangelical Protestant	0.35	0.22
3	Hindu	0.17	0.13
4	Historically Black Protestant	0.53	0.22
5	Jehovah's Witness	0.48	0.25
6	Jewish	0.16	0.15
7	Mainline Protestant	0.29	0.20
8	Mormon	0.27	0.20
9	Muslim	0.34	0.17
10	Orthodox Christian	0.18	0.17
11	Unaffiliated (religious "nones")	0.33	0.20

Renaming Columns

We can use the **.rename()** function to do this:

```
DF.rename( columns={ 'old name 1':'new name 1' ,
                     'old name 2':'new name 2' ,
                     'old name 3':'new name 3', ... },
                     inplace=True )
```

- Here we use curly brackets $\{\}$ to surround the names.
- You list the old name first then a colon then the new name.
- The flag inplace=True tells Python the change the data frame directly

Pivots- Rearranging Data

Often the data we are given is in an order that is hard to use. To rearrange the data in our data frame we have two options

- The **.pivot()** reshapes your data so that one of the columns can become the row labels. To use the pivot method in Pandas, you need to specify three parameters:
  - **index**: Which column should be used to identify and order your rows vertically
  - **columns**: Which column should be used to create the new columns in our reshaped DataFrame.
  - **values**: Which column(s) should be used to fill the values in the cells of our DataFrame.

Melts - Rearranging Data

- The **pd.melt()** can take column labels and melt them into your table values. To use the pd.melt() method in Pandas you need to specify three parameters:
  - The DataFrame that you are melting
  - **id_vars** the list of columns that should remain in your new data frame - used to index the unique rows.
  - **var_name** the name of the new column that will hold the names of the new rows.
  - **value_name** the name of the new column that will hold the values of the new rows

Pivot Example

	name	exam	grade
0	Alice	one	92
1	Bob	one	95
2	Eve	one	70
3	Eve	two	86
4	Alice	two	90
5	Bob	two	80

Pivot Example

Here is the plan:

- use index=‘name’
- use columns=‘exam’
- use values=‘grade’
- then add the average column

```
df_new = df.pivot(index='name',
                  columns='exam',
                  values='grade')
```

Pivot Example - Pivot

exam name	one	two
Alice	92	90
Bob	95	80
Eve	70	86

Pivot Example - Add Average

exam name	one	two	average
Alice	92	90	91.0
Bob	95	80	87.5
Eve	70	86	78.0

Melt example

	Name	Math_2022	Math_2023	Science_2022	Science_2023
0	Alice	85	92	70	75
1	Bob	90	88	82	85
2	Eve	78	95	75	80

Melt example

- send the data frame **df** into **pd.melt()**
- use id_vars = [‘name’]
- use var_name = ‘Subject_Year’
- use value_name=‘Score’

```
df_new = pd.melt(df,
                 id_vars=['Name'],
                 var_name='Subject_Year',
                 value_name='Score')
```

Melt example

	Name	Subject_Year	Score
0	Alice	Math_2022	85
1	Bob	Math_2022	90
2	Eve	Math_2022	78
3	Alice	Math_2023	92
4	Bob	Math_2023	88
5	Eve	Math_2023	95
6	Alice	Science_2022	70
7	Bob	Science_2022	82
8	Eve	Science_2022	75
9	Alice	Science_2023	75
10	Bob	Science_2023	85
11	Eve	Science_2023	80

Religions data

	religion	Less than $30,000	$30,000-$49,999
0	Buddhist	0.36	0.18
1	Catholic	0.36	0.19
2	Evangelical Protestant	0.35	0.22
3	Hindu	0.17	0.13
4	Historically Black Protestant	0.53	0.22
5	Jehovah's Witness	0.48	0.25
6	Jewish	0.16	0.15
7	Mainline Protestant	0.29	0.20
8	Mormon	0.27	0.20
9	Muslim	0.34	0.17
10	Orthodox Christian	0.18	0.17
11	Unaffiliated (religious "nones")	0.33	0.20

Get rid of those dollar signs

What does this code do?

```
DF_new['income']=
   DF_new['income'].apply(lambda x: str(x).replace('$',''))
```

The data - no dollar signs

	religion	n	income	proportion
0	Buddhist	233	Less than 30,000	0.36
24	Buddhist	233	50,000-99,999	0.32
36	Buddhist	233	100,000 or more	0.13
12	Buddhist	233	30,000-49,999	0.18
1	Catholic	6137	Less than 30,000	0.36
25	Catholic	6137	50,000-99,999	0.26
13	Catholic	6137	30,000-49,999	0.19
37	Catholic	6137	100,000 or more	0.19
26	Evangelical Protestant	7462	50,000-99,999	0.28
38	Evangelical Protestant	7462	100,000 or more	0.14
14	Evangelical Protestant	7462	30,000-49,999	0.22
2	Evangelical Protestant	7462	Less than 30,000	0.35
3	Hindu	172	Less than 30,000	0.17
27	Hindu	172	50,000-99,999	0.34
15	Hindu	172	30,000-49,999	0.13

Reset the index.

Resetting the index makes the numbers on the left hand side in order from 0 to 47, instead of them being based on the original data frame that was not sorted alphabetically.

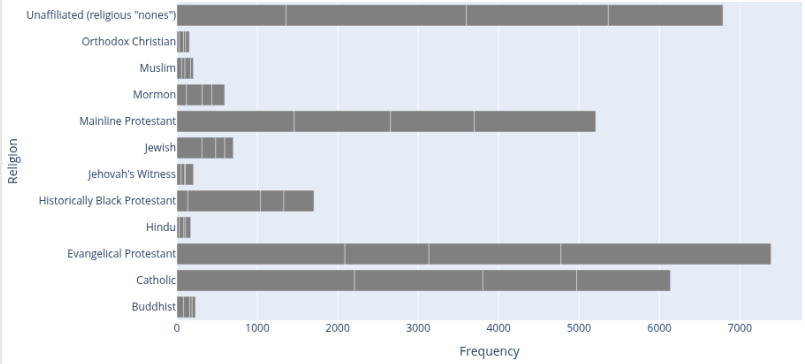
```
DF_new.reset_index(inplace=True)
```

The data - reset index

	index	religion	n	income	prop
0	0	Buddhist	233	Less than 30,000	0.36
1	24	Buddhist	233	50,000-99,999	0.32
2	36	Buddhist	233	100,000 or more	0.13
3	12	Buddhist	233	30,000-49,999	0.18
4	1	Catholic	6137	Less than 30,000	0.36
5	25	Catholic	6137	50,000-99,999	0.26
6	13	Catholic	6137	30,000-49,999	0.19
7	37	Catholic	6137	100,000 or more	0.19
8	26	Evangelical Protestant	7462	50,000-99,999	0.28
9	38	Evangelical Protestant	7462	100,000 or more	0.14
10	14	Evangelical Protestant	7462	30,000-49,999	0.22
11	2	Evangelical Protestant	7462	Less than 30,000	0.35
12	3	Hindu	172	Less than 30,000	0.17
13	27	Hindu	172	50,000-99,999	0.34
14	15	Hindu	172	30,000-49,999	0.13

Make a Bar Plot

Put the religions on the y-axis

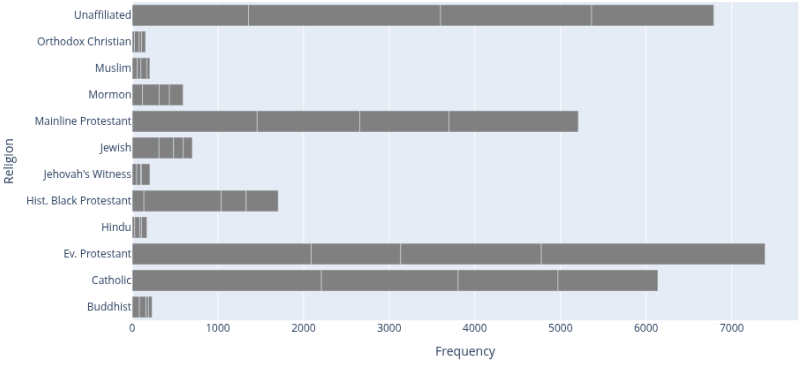


Rename some of the categories

Some of the religion names are really long, making them a bit harder to read. We can do this using the **.replace()** method.

```
DF.replace('old word','new word')
```

New bar plot

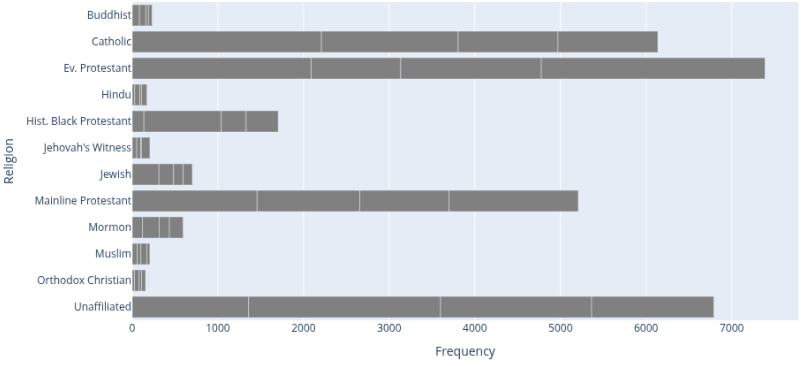

Unaffiliated
Orthodox Christian
Muslim
Mormon
Mainline Protestant
Jewish
Jehovah's Witness
Hist. Black Protestant
Hindu
Ev. Protestant
Catholic
Buddhist
Religion
0
1000
2000
3000
4000
5000
6000
7000
Frequency

Reverse the order of the categories

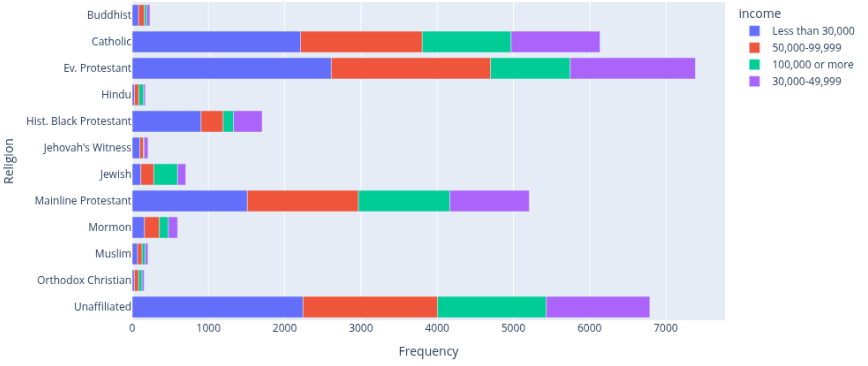
Here is the bar plot code:

```
fig = px.bar(DF_new,y='religion',
             x='frequency',
             color_discrete_sequence=['gray'])

fig.update_layout(yaxis={'categoryorder':'category descending'
                  xaxis_title="Frequency",
                  yaxis_title="Religion",
                  autosize=False,
                  width=800,
                  height=500)

fig.show()
```

New bar plot


Buddhist
Catholic
Ev. Protestant
Hindu
Hist. Black Protestant
Jehovah's Witness
Jewish
Mainline Protestant
Mormon
Muslim
Orthodox Christian
Unaffiliated
Religion
0
1000
2000
3000
4000
5000
6000
7000
Frequency

Add the income information in color

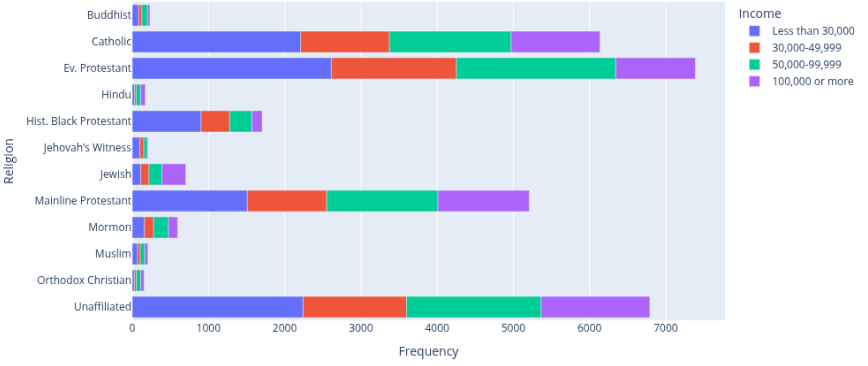


Let's fix those category orders

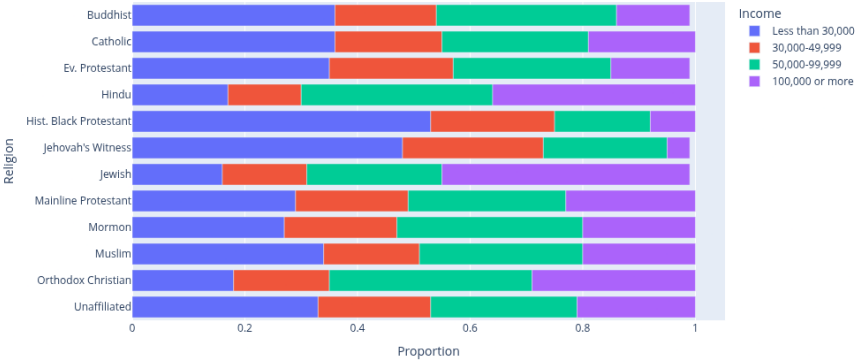
Plot code:

```
fig = px.bar(DF_new,
             y='religion',
             x='frequency',
             color='income',
             category_orders={'income' :
                              ['Less than 30,000',
                              '30,000-49,999',
                              '50,000-99,999',
                              '100,000 or more']})

... the rest is the same
```

New bar plot


Buddhist
Catholic
Ev. Protestant
Hindu
Hist. Black Protestant
Jehovah's Witness
Jewish
Mainline Protestant
Mormon
Muslim
Orthodox Christian
Unaffiliated
Religion
0
1000
2000
3000
4000
5000
6000
7000
Frequency
Income
Less than 30,000
30,000-49,999
50,000-99,999
100,000 or more

Consider proportions instead of frequencies



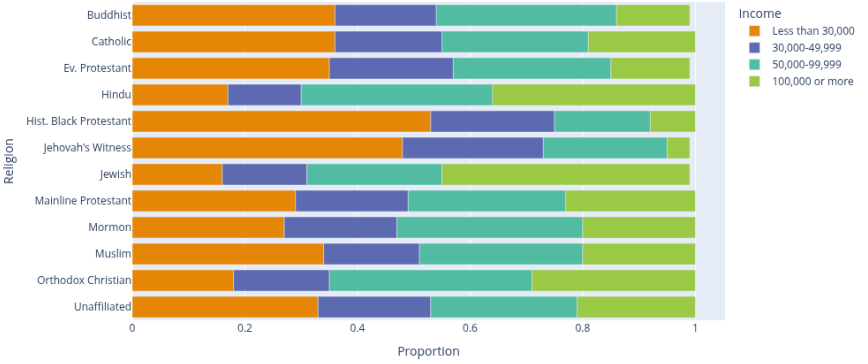
Try some new colors

We can add a built in color sequence using the command:

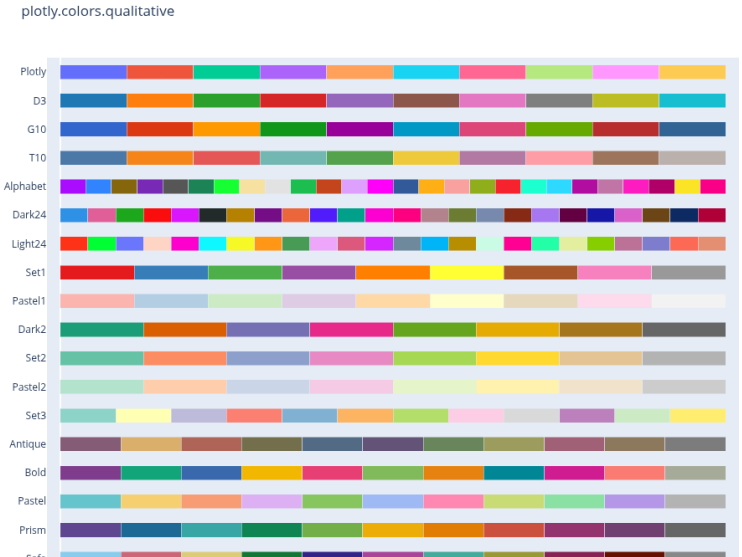
```
color_discrete_sequence=px.colors.qualitative.Vivid
```

In this example we picked the 'Vivid' color sequence. See below for more choices!

New bar plot


Buddhist
Catholic
Ev. Protestant
Hindu
Hist. Black Protestant
Jehovah's Witness
Jewish
Mainline Protestant
Mormon
Muslim
Orthodox Christian
Unaffiliated
Religion
0
0.2
0.4
0.6
0.8
1
Proportion
Income
Less than 30,000
30,000-49,999
50,000-99,999
100,000 or more

Choices for color_discrete_sequence=px.colors.qualitative.



Apply Templates and move the legend to the bottom

WE WILL LOOK AT THIS CODE IN PIECES!

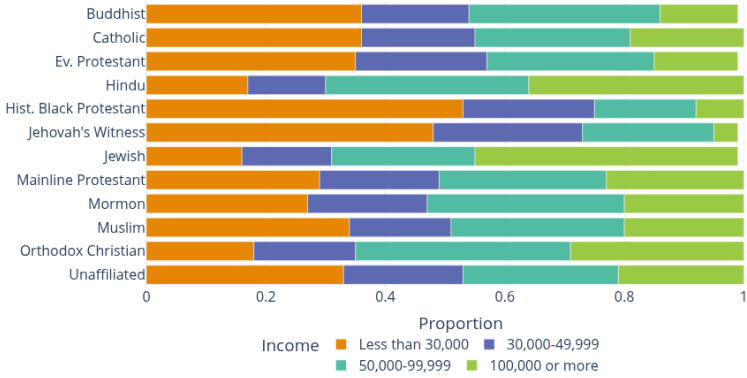
Create the figure

```
fig = px.bar(DF_new,
             y='religion',
             x='proportion',
             color='income',
             color_discrete_sequence=
                    px.colors.qualitative.Vivid,
             category_orders={'income' :
                              ['Less than 30,000',
                              '30,000-49,999',
                              '50,000-99,999',
                              '100,000 or more']})

...
```

Update the layout

```
fig.update_layout(template="plotly_white",
                  yaxis={'categoryorder':'category descending'
                  xaxis_title="Proportion",
                  yaxis_title="",
                  legend_title='Income',
                  legend={'orientation':"h",
                          'yanchor':"bottom",
                          'y':-0.33,
                          'xanchor':"right",
                          'x':0.55},
                  font={'size':16},
                  autosize=False,
                  width=800,
                  height=500)
fig.show()
```

New bar plot


Buddhist
Catholic
Ev. Protestant
Hindu
Hist. Black Protestant
Jehovah's Witness
Jewish
Mainline Protestant
Mormon
Muslim
Orthodox Christian
Unaffiliated
0
0.2
0.4
0.6
0.8
1
Proportion
Income
Less than 30,000
30,000-49,999
50,000-99,999
100,000 or more