Notebook

March 27, 2019

1 Simpson's Paradox Homework Example

Here is a combination of some of the code Sam kindly showed us in class, plus the visualizations I showed you, for our simpson's paradox example on 3/25/19.

Here are a few additional FYIs:

1. The source of the underlying dataset is an article entitled "Simpson's Paradox: A Data Set and Discrimination Case Study" in the Journal of Statistics Education, Volume 22, Number 1 (2014) by Stanley A. Taylor and Amy E. Mickel
2. Taylor and Mickel talk about pivot tables as a good solution for looking at these data in their article. That's a slightly more powerful version of some of the groupby code Sam showed us. For a nice explanation of how to do pivot tables in Pandas, see this web page

```
In [1]: import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        import statsmodels.formula.api as sm
        import numpy as np
        %matplotlib inline

        SECRET_PASSWORD = "INSERT PASSWORD FOR CLASS SERVER HERE"
        endpoint = "https://gobbledygook.herokuapp.com/data?file={}&password={}".format("mickel.
        df = pd.read_csv(endpoint)
```

```
/Users/pauliglot/.local/share/virtualenvs/library_gobbledygook-NjdFyGcL/lib/python3.6/site-packa
examples.directory is deprecated; in the future, examples will be found relative to the 'datapat
  "found relative to the 'datapath' directory.".format(key))
```

```
In [2]: df.head()
```

```
Out[2]:       ID  Age Cohort  Age  Gender  Expenditures           Ethnicity
        0  10210       13-17   17  Female          2113  White not Hispanic
        1  10409       22-50   37    Male         41924  White not Hispanic
        2  10486       0 - 5    3    Male          1454            Hispanic
        3  10538       18-21   19  Female          6400            Hispanic
        4  10568       13-17   13    Male          4412  White not Hispanic
```

Let's look at the (good) choices that Sam made for poking around in these data.

In [3]:
```python
# begin sam's code
df.Ethnicity.unique()
```

Out[3]:
```
array(['White not Hispanic', 'Hispanic', 'Black', 'Multi Race', 'Asian',
       'American Indian', 'Other', 'Native Hawaiian'], dtype=object)
```

In [4]:
```python
df.groupby("Ethnicity")["Expenditures"].mean()
```

Out[4]:
```
Ethnicity
American Indian       36438.250000
Asian                 18392.372093
Black                 20884.593220
Hispanic              11065.569149
Multi Race             4456.730769
Native Hawaiian       42782.333333
Other                  3316.500000
White not Hispanic    24697.548628
Name: Expenditures, dtype: float64
```

In [5]:
```python
df.groupby("Age Cohort")["Expenditures"].mean()
```

Out[5]:
```
Age Cohort
 0 - 5     1415.280488
 51 +     53521.896226
13-17      3922.613208
18-21      9888.537688
22-50     40209.283186
6-12       2226.862857
Name: Expenditures, dtype: float64
```

In [6]:
```python
df.groupby("Gender")["Expenditures"].mean()
```

Out[6]:
```
Gender
Female    18129.606362
Male      18001.195171
Name: Expenditures, dtype: float64
```

In [7]:
```python
df.groupby(["Age Cohort", "Ethnicity"])["Gender"].count()
```

Out[7]:
```
Age Cohort  Ethnicity
 0 - 5      Asian                  8
            Black                  3
            Hispanic              44
            Multi Race             7
            White not Hispanic    20
 51 +       American Indian        2
            Asian                 13
```

```
                 Black                  7
                 Hispanic              17
                 Native Hawaiian        1
                 White not Hispanic    66
13-17            American Indian        1
                 Asian                 20
                 Black                 12
                 Hispanic             103
                 Multi Race             7
                 Other                  2
                 White not Hispanic    67
18-21            Asian                 41
                 Black                  9
                 Hispanic              78
                 Multi Race             2
                 White not Hispanic    69
22-50            American Indian        1
                 Asian                 29
                 Black                 17
                 Hispanic              43
                 Multi Race             1
                 Native Hawaiian        2
                 White not Hispanic   133
6-12             Asian                 18
                 Black                 11
                 Hispanic              91
                 Multi Race             9
                 White not Hispanic    46
Name: Gender, dtype: int64
```

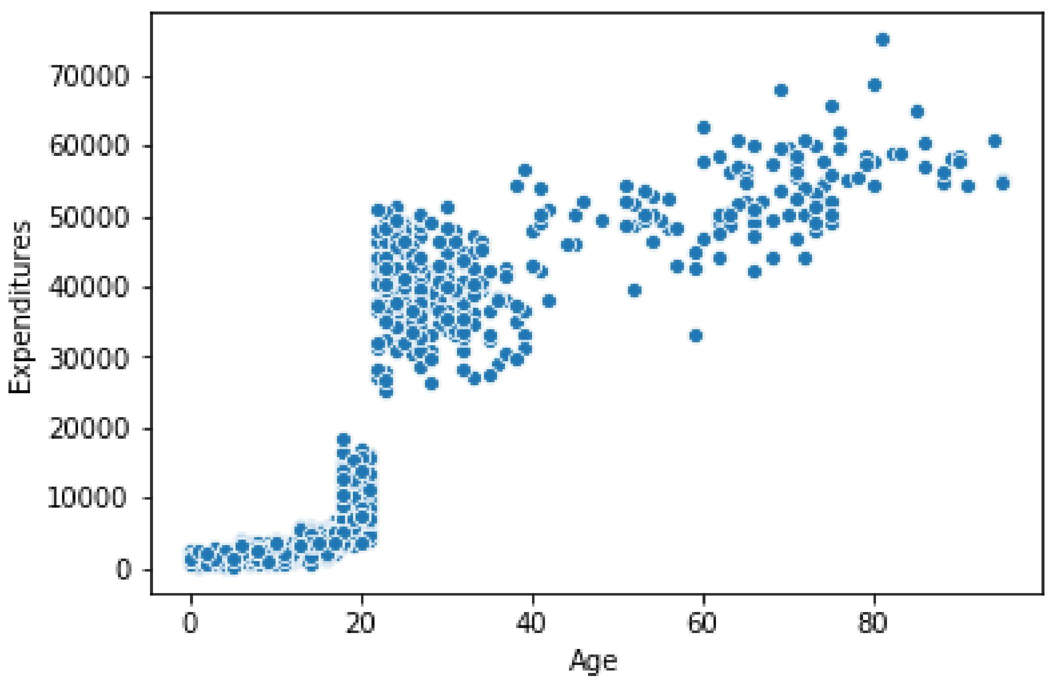
In [8]:
```
age_buckets = df.groupby(["Age Cohort"])["Gender"].count()
df.groupby(["Age Cohort", "Ethnicity"])["Gender"].count() / age_buckets * 100
```

Out[8]:
```
Age Cohort  Ethnicity
 0 - 5      Asian                  9.756098
            Black                  3.658537
            Hispanic              53.658537
            Multi Race             8.536585
            White not Hispanic    24.390244
 51 +       American Indian        1.886792
            Asian                 12.264151
            Black                  6.603774
            Hispanic              16.037736
            Native Hawaiian        0.943396
            White not Hispanic    62.264151
13-17       American Indian        0.471698
            Asian                  9.433962
            Black                  5.660377
```

```
                Hispanic               48.584906
                Multi Race              3.301887
                Other                   0.943396
                White not Hispanic     31.603774
18-21           Asian                  20.603015
                Black                   4.522613
                Hispanic               39.195980
                Multi Race              1.005025
                White not Hispanic     34.673367
22-50           American Indian         0.442478
                Asian                  12.831858
                Black                   7.522124
                Hispanic               19.026549
                Multi Race              0.442478
                Native Hawaiian         0.884956
                White not Hispanic     58.849558
6-12            Asian                  10.285714
                Black                   6.285714
                Hispanic               52.000000
                Multi Race              5.142857
                White not Hispanic     26.285714
Name: Gender, dtype: float64
```

In [9]:
```
sns.scatterplot(df["Age"], df["Expenditures"])
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x114f49898>



```
In [10]: mod = sm.ols(formula="Expenditures ~ Age", data=df)

         res = mod.fit()

         print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Expenditures   R-squared:                       0.711
Model:                            OLS   Adj. R-squared:                  0.711
Method:                 Least Squares   F-statistic:                     2456.
Date:                Wed, 27 Mar 2019   Prob (F-statistic):          2.64e-271
Time:                        16:48:03   Log-Likelihood:                -10678.
No. Observations:                1000   AIC:                         2.136e+04
Df Residuals:                     998   BIC:                         2.137e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept  -2285.6473    528.305     -4.326      0.000   -3322.364   -1248.931
Age          892.6067     18.011     49.558      0.000     857.262     927.951
==============================================================================
Omnibus:                      165.825   Durbin-Watson:                   1.980
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              251.596
Skew:                           1.165   Prob(JB):                     2.33e-55
Kurtosis:                       3.780   Cond. No.                         46.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
In [11]: mod = sm.ols(formula="Expenditures ~ Ethnicity", data=df)

         res = mod.fit()

         print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Expenditures   R-squared:                       0.118
Model:                            OLS   Adj. R-squared:                  0.112
Method:                 Least Squares   F-statistic:                     18.94
Date:                Wed, 27 Mar 2019   Prob (F-statistic):           7.89e-24
Time:                        16:48:03   Log-Likelihood:                -11236.
No. Observations:                1000   AIC:                         2.249e+04
Df Residuals:                     992   BIC:                         2.253e+04
```

```
Df Model:                           7
Covariance Type:            nonrobust
==================================================================================================
                                     coef    std err          t      P>|t|      [0.025      0.9
--------------------------------------------------------------------------------------------------
Intercept                       3.644e+04   9209.742      3.956      0.000    1.84e+04    5.45e
Ethnicity[T.Asian]             -1.805e+04   9351.439     -1.930      0.054   -3.64e+04     304.
Ethnicity[T.Black]             -1.555e+04   9516.817     -1.634      0.103   -3.42e+04    3121.
Ethnicity[T.Hispanic]          -2.537e+04   9258.600     -2.740      0.006   -4.35e+04   -7203.
Ethnicity[T.Multi Race]        -3.198e+04   9892.850     -3.233      0.001   -5.14e+04   -1.26e
Ethnicity[T.Native Hawaiian]    6344.0833   1.41e+04      0.451      0.652   -2.13e+04     3.4e
Ethnicity[T.Other]             -3.312e+04     1.6e+04     -2.076      0.038   -6.44e+04   -1818.
Ethnicity[T.White not Hispanic] -1.174e+04   9255.562     -1.269      0.205   -2.99e+04    6422.
==============================================================================
Omnibus:                       95.947   Durbin-Watson:                   2.015
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              104.142
Skew:                           0.747   Prob(JB):                     2.43e-23
Kurtosis:                       2.485   Cond. No.                         53.4
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [12]:
```python
mod = sm.ols(formula="Expenditures ~ Ethnicity + Age + Gender", data=df)

res = mod.fit()

print(res.summary())
```

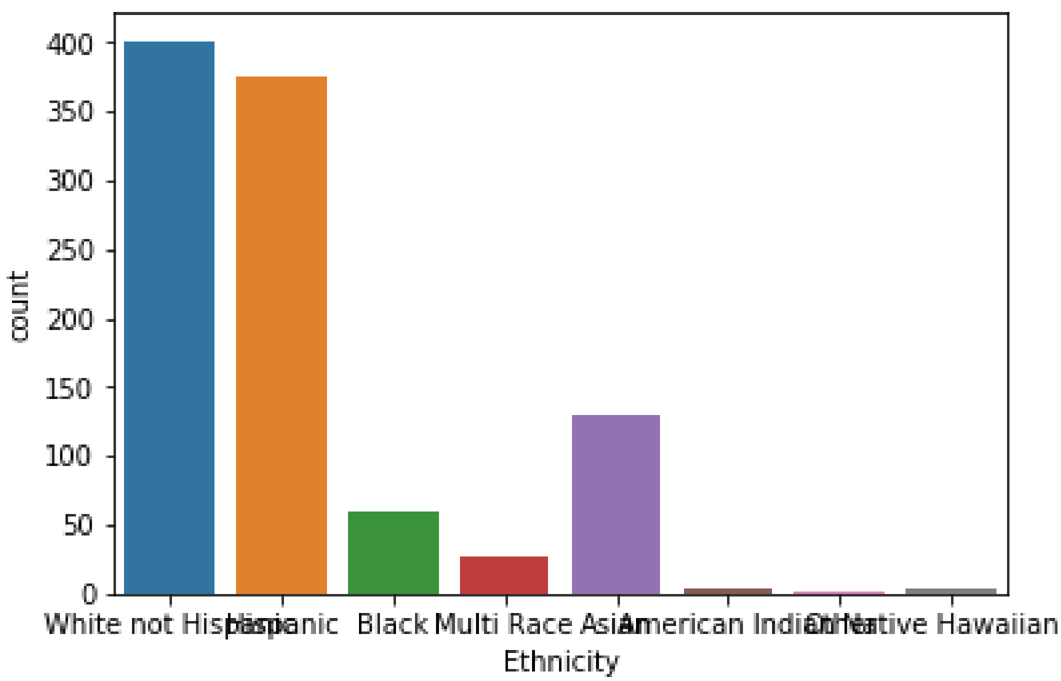
```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Expenditures   R-squared:                       0.724
Model:                            OLS   Adj. R-squared:                  0.721
Method:                 Least Squares   F-statistic:                     288.3
Date:                Wed, 27 Mar 2019   Prob (F-statistic):          1.82e-269
Time:                        16:48:03   Log-Likelihood:                -10655.
No. Observations:                1000   AIC:                         2.133e+04
Df Residuals:                     990   BIC:                         2.138e+04
Df Model:                           9
Covariance Type:            nonrobust
==================================================================================================
                                     coef    std err          t      P>|t|      [0.025      0.9
--------------------------------------------------------------------------------------------------
Intercept                      -9478.0155   5254.102     -1.804      0.072   -1.98e+04     832.
Ethnicity[T.Asian]              8083.2889   5270.650      1.534      0.125   -2259.641    1.84e
Ethnicity[T.Black]              9224.2697   5360.504      1.721      0.086   -1294.986    1.97e
Ethnicity[T.Hispanic]           5664.1767   5230.543      1.083      0.279   -4600.049    1.59e
```

```
Ethnicity[T.Multi Race]          5193.5583   5600.352      0.927      0.354   -5796.366    1.62e
Ethnicity[T.Native Hawaiian]     2.155e+04   7886.344      2.732      0.006    6071.684     3.7e
Ethnicity[T.Other]               -894.9578   8962.598     -0.100      0.920   -1.85e+04    1.67e
Ethnicity[T.White not Hispanic]  1.014e+04   5207.487      1.948      0.052     -75.692    2.04e
Gender[T.Male]                   -251.7477    653.446     -0.385      0.700   -1534.046    1030.
Age                               863.4592     18.526     46.607      0.000     827.104     899.
==============================================================================
Omnibus:                      158.613   Durbin-Watson:                   1.983
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              236.989
Skew:                           1.121   Prob(JB):                     3.45e-52
Kurtosis:                       3.811   Cond. No.                     1.36e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.36e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [13]:
```python
# gowder code (visualizations) begins here
sns.countplot(df["Ethnicity"])
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x115094b38>



In [14]:
```python
def bin_ethnicity(eth):
    if eth == "White not Hispanic":
```

```
            return "white"
        elif eth == "Hispanic":
            return "hispanic"
        return "other"



# there is doubtless a better way to do this involving the apply function in pandas or
# But I'm rusty with my Pandas data tranformations.

df["binned_eth"] =np.array([bin_ethnicity(x) for x in list(df["Ethnicity"])])
```

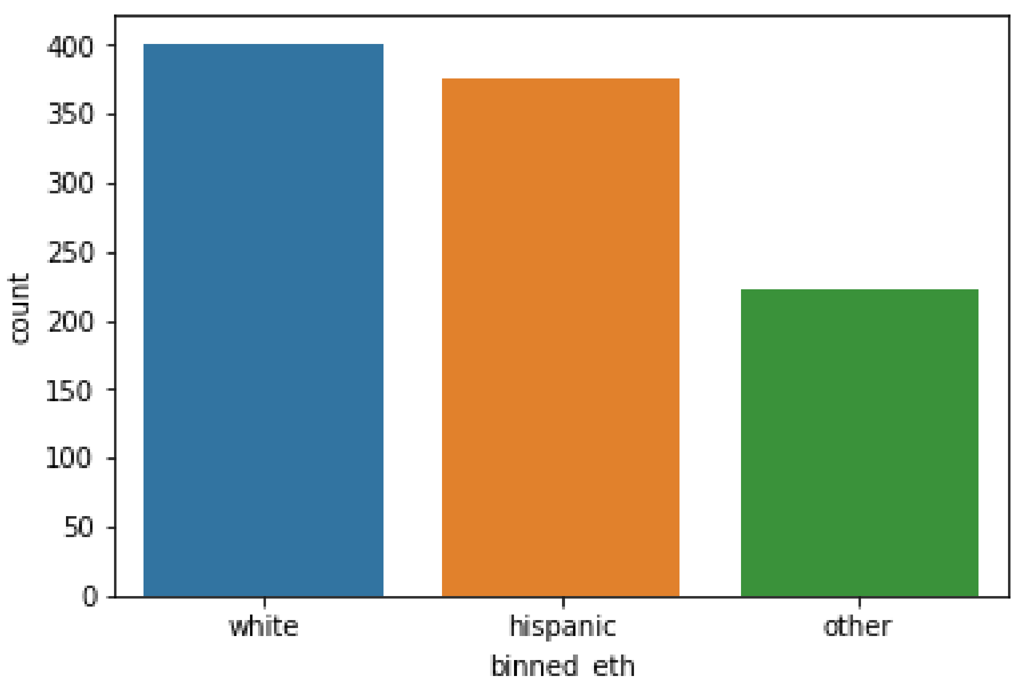
In [15]:
```
df.head()
```

Out[15]:

	ID	Age Cohort	Age	Gender	Expenditures	Ethnicity	binned_eth
0	10210	13-17	17	Female	2113	White not Hispanic	white
1	10409	22-50	37	Male	41924	White not Hispanic	white
2	10486	0 - 5	3	Male	1454	Hispanic	hispanic
3	10538	18-21	19	Female	6400	Hispanic	hispanic
4	10568	13-17	13	Male	4412	White not Hispanic	white

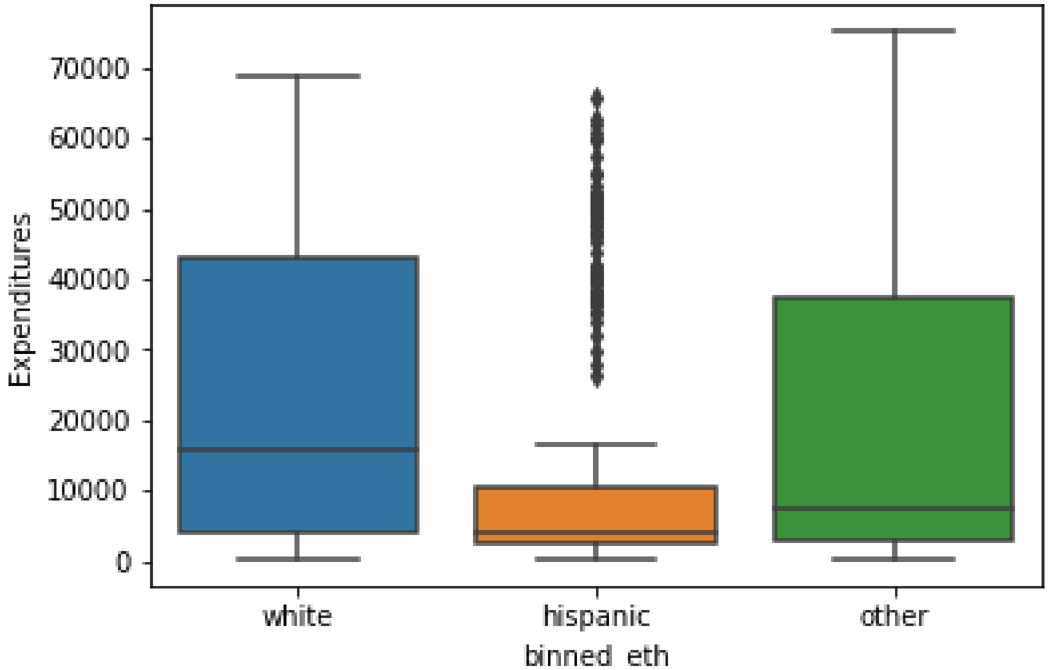
In [16]:
```
sns.countplot(df["binned_eth"])
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x115091160>



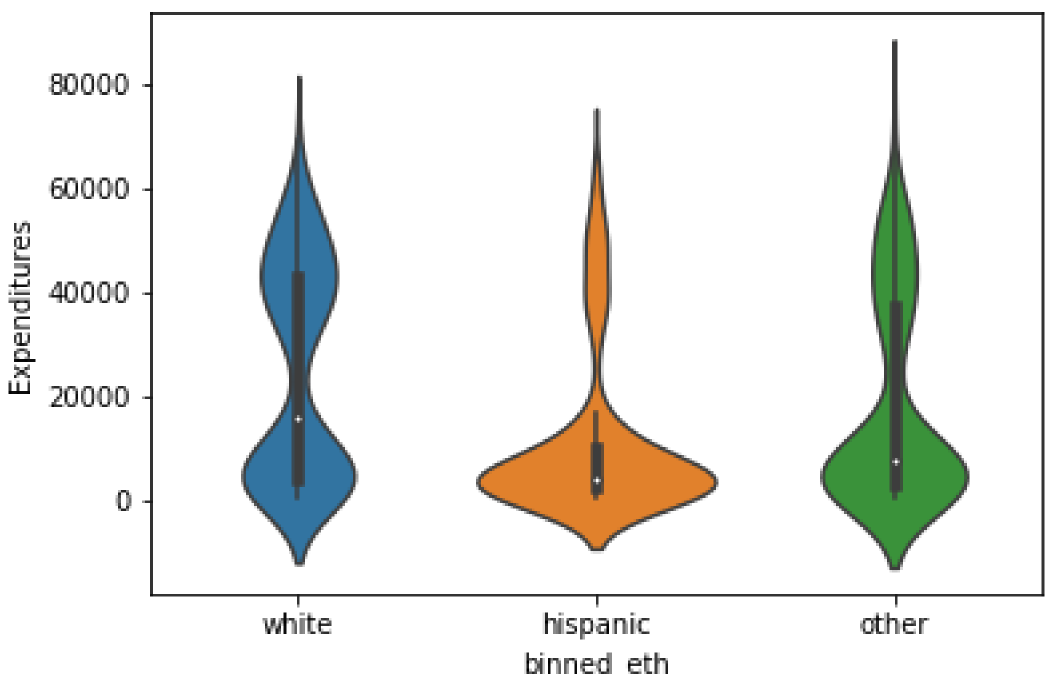
In [17]:
```
sns.boxplot(x=df["binned_eth"], y=df["Expenditures"])
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x11514ccf8>




```
In [18]: sns.violinplot(x=df["binned_eth"], y=df["Expenditures"])
```

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x115142470>



```
In [19]: cohorts = sorted(df["Age Cohort"].unique())  # just sorting this now like a sensible pe
```

```
In [20]: cohorts
```

```
Out[20]: [' 0 - 5', ' 51 +', '13-17', '18-21', '22-50', '6-12']
```

I'm going to make a couple of changes from the code I showed in class here.

First, I'm going to sort our pandas dataframe by the value of binned ethnicity in order to try to get our columns in the violin plots to come out right.

Second, I'm going to sort the list of cohorts so that it's easy to generate plots in order.

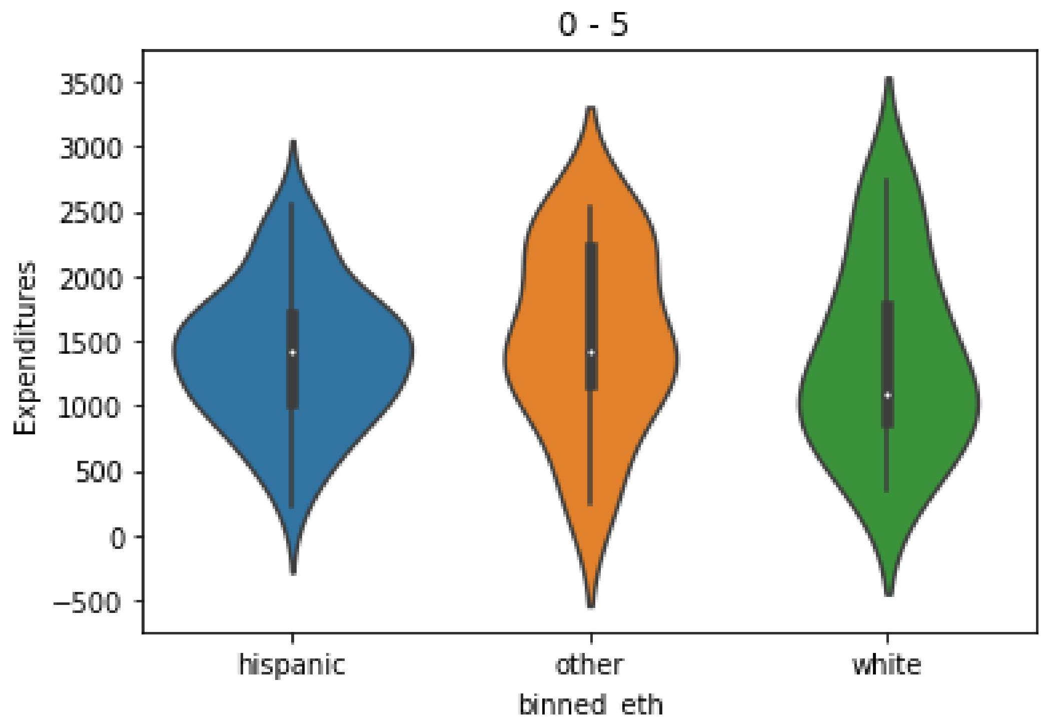
Third, I'm going to change the function that generates the violin plot to let me loop over and show a plot for each cohort.

```
In [21]: df.sort_values("binned_eth", inplace=True)
```

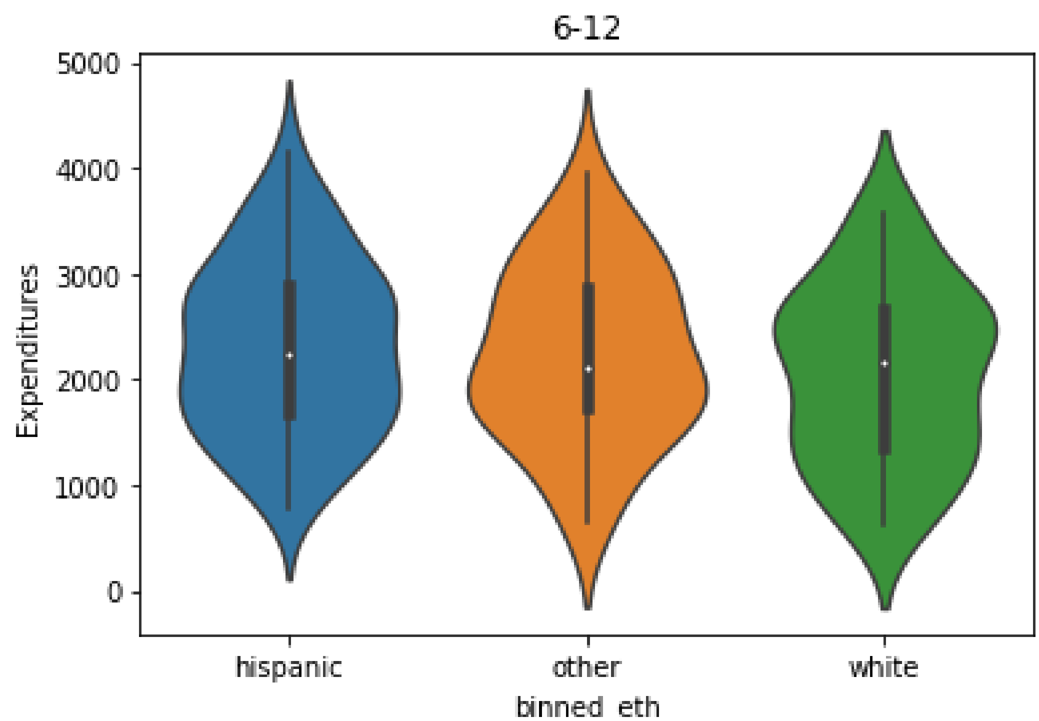
```
In [22]: import re
         def sorting_function(elem):
             e = elem.strip()
             e = re.split(r"[-\+\s]", e)
             return int(e[0])
         cohorts = sorted(cohorts, key=sorting_function)
```

```
In [23]: import matplotlib.pyplot as plt # this is a change from my code in class to make it wor
         def subsetted_violin(cohort):
             temp_df = df[df["Age Cohort"] == cohort]
             plt.figure()
             sns.violinplot(x=temp_df["binned_eth"], y=temp_df["Expenditures"])
             plt.title(cohort)
```

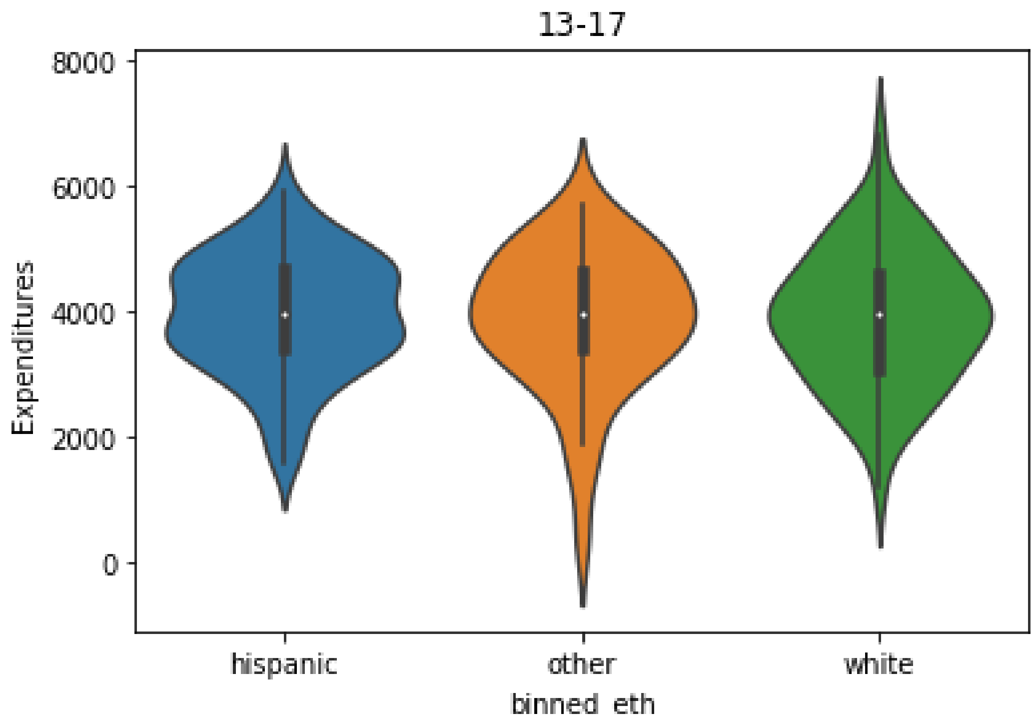
```
In [24]: for cohort in cohorts:
             subsetted_violin(cohort)
```

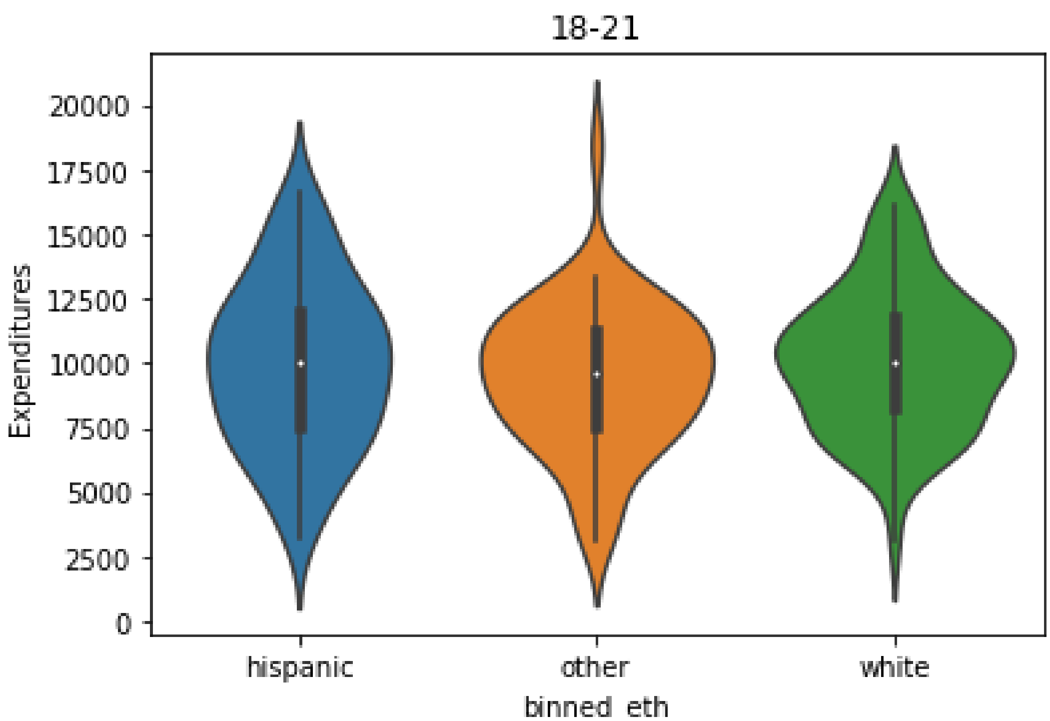


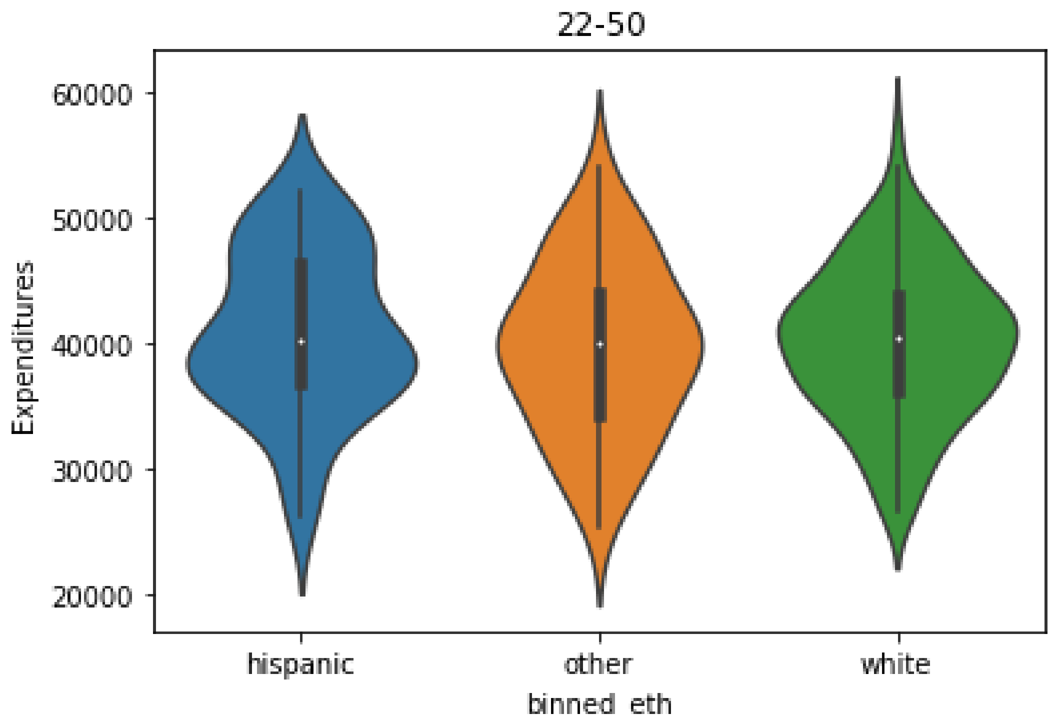
0 - 5
3500
3000
2500
2000
1500
1000
500
0
−500
Expenditures
hispanic
other
white
binned_eth

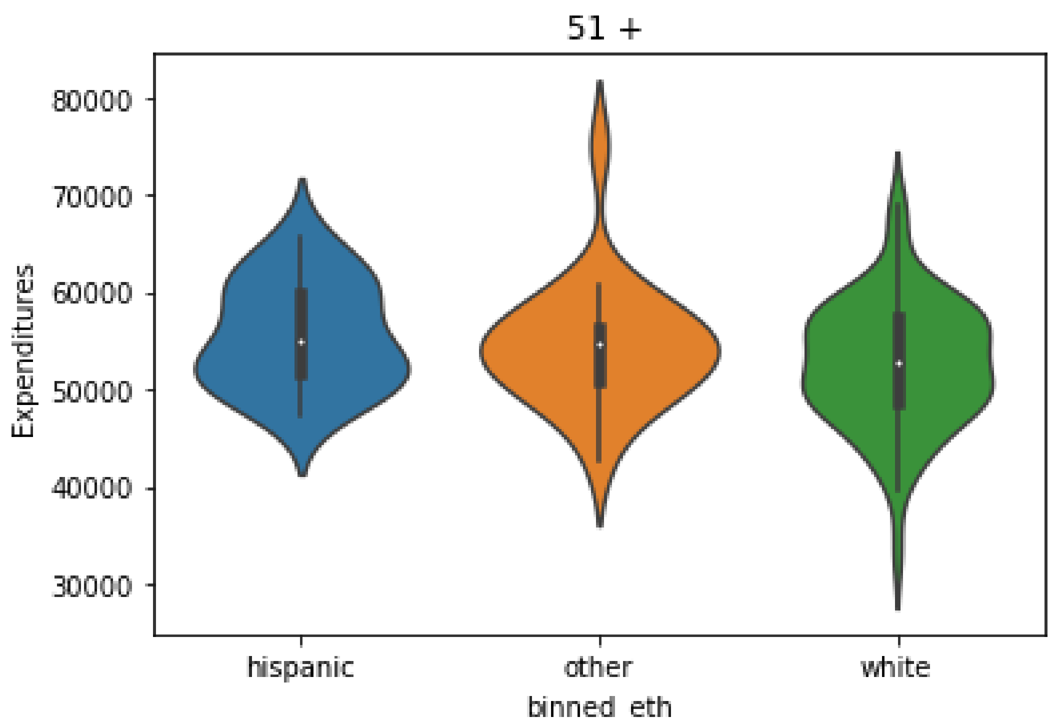


6-12
5000
4000
3000
2000
1000
0
Expenditures
hispanic
other
white
binned_eth


13-17
Expenditures
8000
6000
4000
2000
0
hispanic
other
white
binned_eth

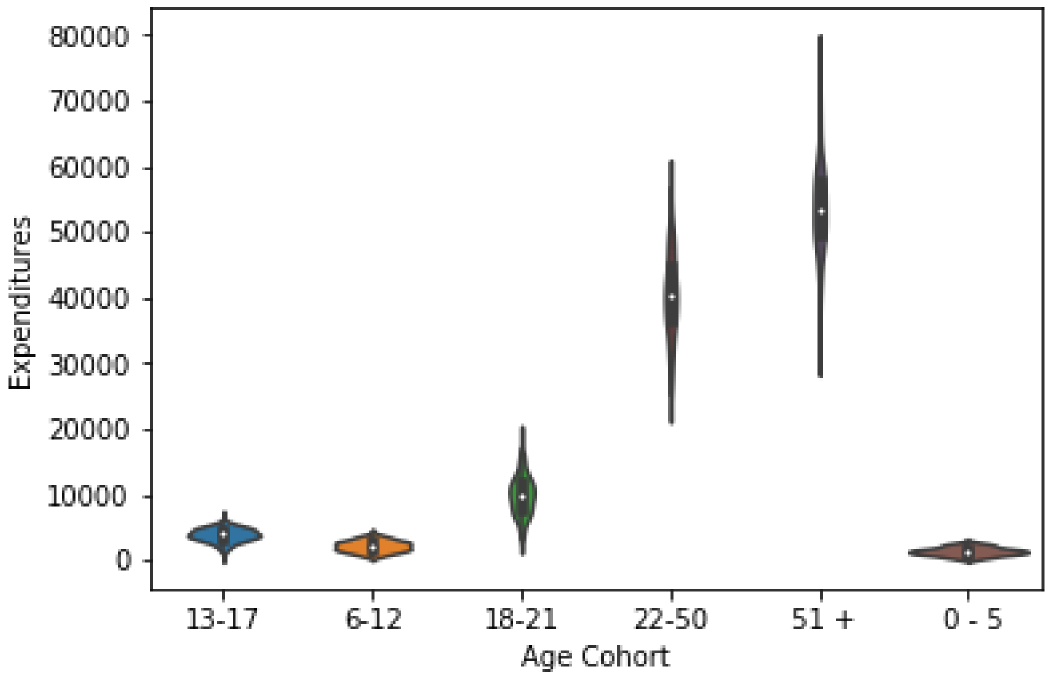

18-21
Expenditures
20000
17500
15000
12500
10000
7500
5000
2500
0
hispanic
other
white
binned_eth


22-50
60000
50000
40000
30000
20000
Expenditures
hispanic
other
white
binned_eth


51 +
80000
70000
60000
50000
40000
30000
Expenditures
hispanic
other
white
binned_eth

```
In [25]: sns.violinplot(x=df["Age Cohort"], y=df["Expenditures"])
```

```
Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x115877ac8>
```

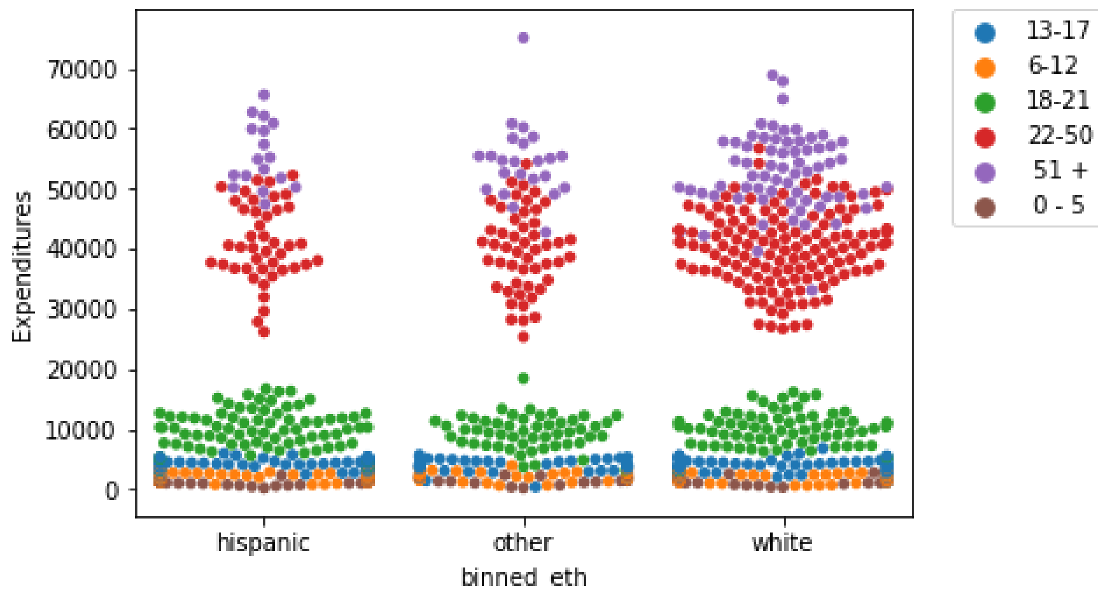


Anna showed us another very useful plot—the swarm plot, which is sort of like a violin plot, but with dots for individual items as well as the capacity to see an extra dimension by colorizing those dots—which can be very useful for seeing the relationship between ethnicity, age cohort, and expenditures in these data.

(Seaborn legends can be obnoxious; this second line of code uses matplotlib to take the existing legend and shove it over to the right so it doesn't end up on top of the figure.)

```
In [26]: sns.swarmplot(df["binned_eth"], df["Expenditures"], df["Age Cohort"], size=5)
         plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

```
Out[26]: <matplotlib.legend.Legend at 0x11554e208>
```



If we want to, we can even use a catplot to divide up our swarmplots by some other dimension, like gender, so we can eyeball whether there are any differences in there too. See the seaborn docs for more cool things we can do with these swarm and swarm-adjacent plots.

```
In [27]: sns.catplot(x="binned_eth", y="Expenditures", hue="Age Cohort", col="Gender", data=df)
```

```
Out[27]: <seaborn.axisgrid.FacetGrid at 0x10cdc8f60>
```