ASSIGNMENT 3

CB.EN.U4CSE20238

Dataset consists of 3 Attributes → Ethnicity, Eligible, Panels:

Ethnicity describes about the ethnic categories of people in Alameda.

Eligible describes about the fraction of people eligible grouped by ethnicity.

Panels describes about the proportion of people currently chosen for the panel.

We have taken

Null Hypothesis: -panels were selected at random from the population of eligible jurors.

Alternate Hypothesis: -panels were not selected at random

```python
#add a column for difference in eligible and panel
Al_df_1['jury_with_diffs'] = Al_df_1['Panels']-Al_df_1['Eligible']
Al_df_1
```

Ethnicity	Eligible	Panels	jury_with_diffs
Asian	0.15	0.26	0.11
Black	0.18	0.08	-0.10
Latino	0.12	0.08	-0.04
White	0.54	0.54	0.00
Other	0.01	0.04	0.03

```python
Al_df_1 = Al_df.set_index('Ethnicity')
Al_df_1
```

Ethnicity	Eligible	Panels
Asian	0.15	0.26
Black	0.18	0.08
Latino	0.12	0.08
White	0.54	0.54
Other	0.01	0.04

Take a look at the column Difference and notice that the sum of its entries is 0: the positive entries add up to 0.14, exactly canceling the total of the negative entries which is -0.14.

This is numerical evidence of the fact that in the bar chart, the gold bars exceed the blue bars by exactly as much as the blue bars exceed the gold. The proportions in each of the two columns Panels and Eligible add up to 1, and so the give-and-take between their entries must add up to 0.

To avoid the cancellation, we drop the negative signs and then add all the entries. But this gives us two times the total of the positive entries (equivalently, two times the total of the negative entries, with the sign removed). So we divide the sum by 2.

```
#finding ab
Al_df_1['Abs.Difference']=abs(Al_df_1['jury_with_diffs'])
Al_df_1
```

Ethnicity	Eligible	Panels	jury_with_diffs	Abs.Difference
Asian	0.15	0.26	0.11	0.11
Black	0.18	0.08	-0.10	0.10
Latino	0.12	0.08	-0.04	0.04
White	0.54	0.54	0.00	0.00
Other	0.01	0.04	0.03	0.03

```
[11] test_statistic = Al_df_1['Abs.Difference'].sum()/2
     test_statistic
```

```
0.14
```

Therefore, it would only make sense logically if the number of members selected at random that are in excess are same as the one in deficit. 14%, -14% in our case.

```
[13] def table_tvd(table, label, other):
         return total_variation_distance(table[label], table[other])

     observed_stat =table_tvd(Al_df, 'Eligible', 'Panels')
     print(observed_stat)
```

```
0.14
```

```
panel_size = 1453
import numpy.random as npr
np.random.multinomial(1453,[0.15,0.18,0.12,0.54,0.01])
```

```
array([195, 275, 178, 790,  15])
```

This quantity 0.14 is the total variation distance (TVD) between the distribution of ethnicities in the eligible juror population and the distribution in the panels.

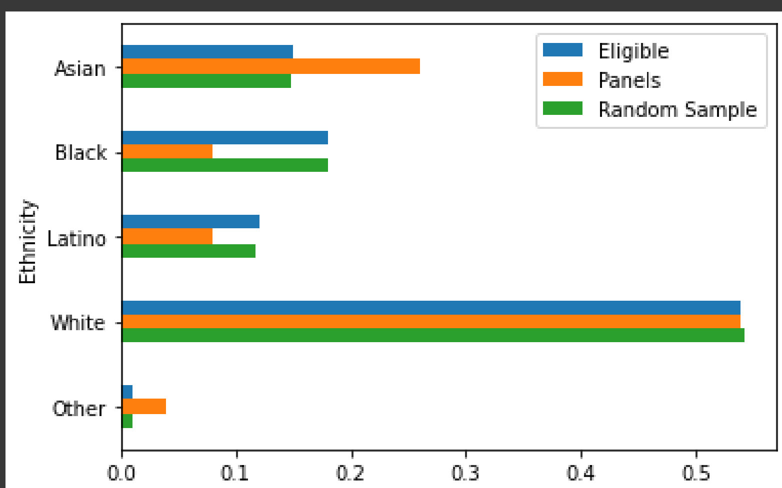
We could have obtained the same result by just adding the positive differences. But our method of including all the absolute differences eliminates the need to keep track of which differences are positive and which are not.

```python
Al_df_2 = pd.DataFrame(Al_df_1,columns = ['Eligible','Panels'])
Al_df_2['Random Sample'] = np.random.multinomial(1453,[0.15,0.18,0.12,0.54,0.01])/1453
Al_df_2
```

Ethnicity	Eligible	Panels	Random Sample
Asian	0.15	0.26	0.148658
Black	0.18	0.08	0.181005
Latino	0.12	0.08	0.117688
White	0.54	0.54	0.542326
Other	0.01	0.04	0.010323

the distribution of the random sample is close to the distribution of the eligible population and is different from the distribution of the panels.

```python
Al_df_2.plot.barh()
plt.ylabel('Ethnicity')
plt.gca().invert_yaxis()
```



The green bar are closer in size to the blue bars than the orange bars are. The randomsample resembles the eligible population, but the panels don't.

```
#Difference between eligible and Random sample
TVD = (abs(Al_df_2['Eligible']-Al_df_2['Random Sample'])).sum()/2
TVD
```

```
0.003654507914659287
```

```
[19] simulations = 5000
     tvd_list=[]
     for i in np.arange(simulations):
         Al_df_2["Random Sample"]=(npr.multinomial(1453,[0.15, 0.18, 0.12, 0.54, 0.01]))/panel_size
         tvd_list.append(table_tvd(Al_df_2, 'Eligible', 'Random Sample'))

     tvd_list
```

```
0.01203716448726768,
0.014115622849277348,
0.009298004129387479,
0.013695801789401205,
0.014714384033035056,
0.015760495526496066,
0.019848589125946305,
0.011810048176187206,
0.010963523743977993,
```

```
[20] tvd_final_df=pd.DataFrame(tvd_list)
     tvd_final_df.rename(columns={0:"TVD"},inplace=True) # renaming column
     tvd_final_df.head()
```

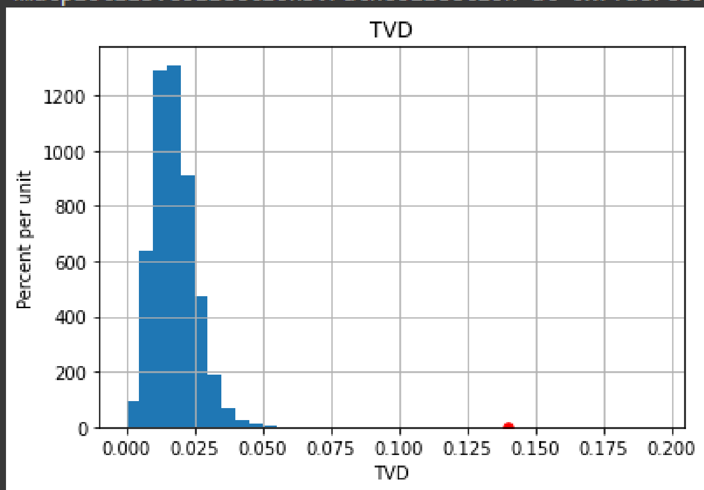
	TVD
0	0.015719
1	0.007928
2	0.010337
3	0.018830
4	0.030172

Now, we find out the Difference between Random value and the Actual Eligible value, 0.01209222 in our case.

Repeat this task 5000 more times, and find this difference each time and store it in a dataframe.

```
tvd_final_df.hist(bins=np.arange(0,0.2,0.005))
plt.ylabel('Percent per unit')
plt.xlabel('TVD')
plt.scatter(observed_stat, 0, color='red', s=30)
```

```
<matplotlib.collections.PathCollection at 0x7fde7c8916d0>
```



From this histogram we can see that the values what we got are far away from the scatter plot point. By this we can say that this is because we have sufficient proof to prove that the Alternative hypothesis turns out to be True, meaning, there was a clear bias.

Plot a histogram to visualize such a huge data easily and also use scatter plot to plot a point of the observed difference, and hence to Reject our Null Hypothesis.