NAME: NERELLA CHIRANJEEVI STUDENTID: 11442777
COURSE: 5502

ASSIGNMENT-4

<u>Output :</u>

Q1) Calculate the median income of male employees and the median income of female employee in the population. Consider the set of all employees in the datasets as the population.

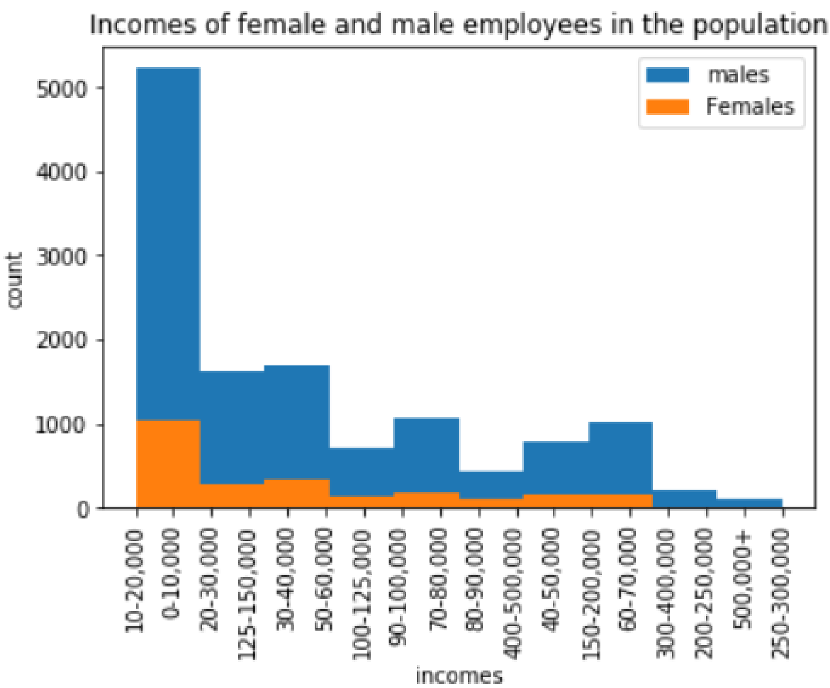
```python
#males=males.reset_index()
#Females=Females.reset_index()
#males=males.drop(columns="index")

males=p[p["Q1"]=="Male"]
Females=p[p["Q1"]=="Female"]
males_median=int(len(males["Q9"])/2)+1
Females_median=int(len(Females["Q9"])/2)+1
print("median of male salary",p.Q9[males_median])
print("median of Female salary",p.Q9[Females_median])
#print(males)
```

```
median of male salary 30-40,000
median of Female salary 0-10,000
```

Q2) Draw an overlaid graph to show the histograms of the incomes of female and male employees in the population.

```python
plt.hist(males["Q9"],label="males")
plt.hist(Females["Q9"],label="Females")
plt.xlabel("incomes")
plt.ylabel("count")
plt.title("Incomes of female and male employees in the population")
plt.xticks(rotation=90)
plt.legend()

```

```
]: <matplotlib.legend.Legend at 0x19648310388>
```



Q3) Select a sample from the population. Make sure your sample include 500 employees selected from the population, and consider how to ensure the sampling strategy is fair since the datasets include an overwhelming number of male employees compared to female employees

```python
males1.sample(n=250) #selecting 250 rows from males data

```

]:

	Q1	Q9
10453	Male	95000
10342	Male	45000
9999	Male	85000
5378	Male	5000
1353	Male	35000
...	...	...
11370	Male	25000
17049	Male	35000
12472	Male	195000
10	Male	25000
11156	Male	45000

250 rows × 2 columns

```
Females1.sample(n=250) #selecting 250 rows from females data

```

	Q1	Q9
11241	Female	120000
1407	Female	5000
3219	Female	85000
8836	Female	5000
18488	Female	120000
...	...	...
7632	Female	5000
6116	Female	25000
2223	Female	65000
22691	Female	25000
5217	Female	95000

250 rows × 2 columns

3(II) Define the test statistic, the null hypothesis and the alternative hypothesis

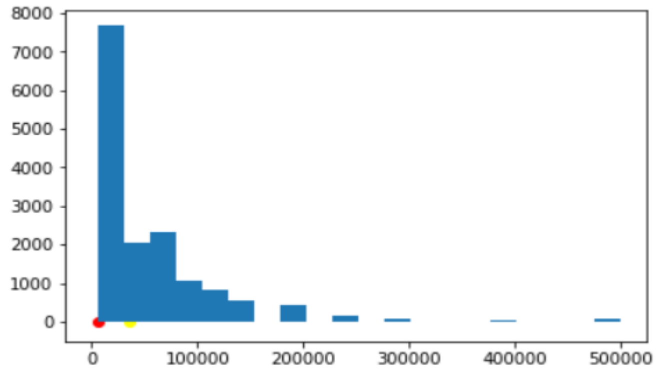
```
p["Q9"].mean()
from scipy.stats import ttest_1samp
salary_mean = p["Q9"].mean()

ttest,pval = ttest_1samp(p["Q9"],salary_mean)
ttest,pval
```

```
]: (0.0, 1.0)
```

3(III) Draw the income histogram for the sample; calculate the median income of the sample; and draw a red dot and a yellow dot for the female median income and male median income of the population respectively, in the histogram

```python
plt.hist(s["Q9"],bins=20)
plt.xticks()
plt.scatter(p.Q9[males_median],0,color="yellow")
plt.scatter(p.Q9[Females_median],0,color="red")
```

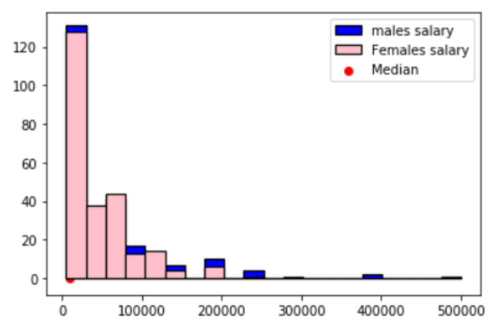
```
<matplotlib.collections.PathCollection at 0x196494f4b08>
```



3(iv) Draw the histogram of the test statistic of the sample, and draw a red dot to show the corresponding test statistic of the population (e.g. the difference of the median incomes between female and male employees) in the diagram

```python
plt.hist(sample1["Q9"],bins=20,alpha=1.0,edgecolor='black',color='blue', linewidth=1.2,label="males salary") # PLot hist
plt.hist(sample2["Q9"],bins=20,alpha=1.0,edgecolor='black',color='pink', linewidth=1.2,label="Females salary") # PLot hi

plt.scatter(diff_male_female_salary,0,color='red',label='Median') # Plot dot for Median differnce of male and female
plt.legend()
```

```
<matplotlib.legend.Legend at 0x196495d21c8>
```



3(v) Write a procedure to use bootstrap to produce at least 5000 samples

<u>Bootstrap sampling:</u>

- Drawn a sample from the original sample data with replacement with size n, and replicate 5000 times, each re-sampled sample is called a Bootstrap Sample, and there will totally 5000 Bootstrap Samples.

- Evaluated the mean/median for each Bootstrap Sample, and there will be totally 5000 estimates of re-sampled samples.
- Constructed a sampling distribution i.e. Mean/Median Difference with these 5000 Bootstrap statistics by storing it in array and use it to make further statistical inference like histograms etc.

```python
np.random.seed(500)
def bootstrap(population):
    sample=np.random.choice(population,len(population))
    v=sample.mean()
    return v


```

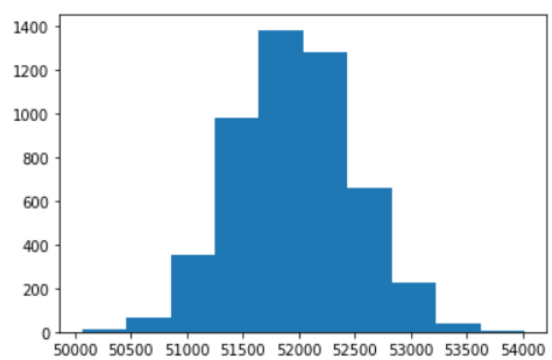
```python
s=[]
for i in range(5000):
    x=bootstrap(population["Q9"])
    s.append(x)
print(s)
```

```
[51889.34802571166, 51743.40810704447, 51467.27010363374, 51075.364029909484, 51298.37334382789, 52038.89544798636, 5238
6.52761380034, 52051.357733175915, 51571.55975337794, 51621.736849009576, 51613.53797717434, 51796.5367965368, 52583.9564
4759281, 51658.13983995802, 51823.42909615637, 52567.88665879575, 52557.72005772006, 52791.2239275875 6, 51356.4213564213
6, 51451.528269710085, 51748.32743014561, 52713.82657746294, 51426.93165420438, 50913.02636757182, 51361.01272464909, 523
68.16214088941, 52489.83339892431, 50639.839958021774, 52236.32428177883, 51127.50885478158, 54142.39800603437, 52272.399
317853866, 51936.57352748262, 51487.93126065853, 52133.34645152827, 52072.01889020071, 52658.07424898334, 51374.130919585
46, 52834.1860160042, 51863.43959071232, 52672.50426341336, 52030.040666404304, 51821.461366915915, 52578.05325987144, 52
687.59018759019, 50997.96667978486, 52383.24806506625, 52588.21986094713, 51495.14626787354, 52410.140364685816, 51671.58
598976781, 51691.26328217 2375, 52294.7002492457, 51515.15151515151, 51648.62914862915, 51742.752197297654, 51853.27298963
663, 51877.541650268926, 52236.98019152565, 52644.30014430014, 51483.3398924308, 51556.14587432769, 52155.97533779352, 51
675.52144824872, 51233.11032401941, 51172.110717565265, 51623.37662337662, 51788.33792470156, 52325.8559622196, 51615.505
7064148, 52388.823297914205, 52172.04512659058, 52762.69185360095, 52165.15807424898, 52287.813196904106, 51612.882067427
52, 51731.273776728325, 51661.41938869211, 51114.39065984521, 51589.92522628886, 51565.656565656565, 52167.12580348944, 5
1800.80020989112, 51892.955529319166, 52387.83943329398, 51496.13013249377, 52469.17224189951, 52372.753509117145, 52897.
48130657222, 51340.6795224977, 52155.97533779352, 52544.92981765709, 51727.338318247406, 51175.71822117276, 51879.8373343
8279, 52649.219467401286, 51803.7518037518, 51316.410861865406, 52465.2367834186, 52633.47763347763, 51537.45244654336, 5
1974.2883379247, 51499.73763610127, 51497.441951987406, 51894.59530368621, 51610.91433818707, 52389.15125278762, 52170.73
330709694, 51392.16843762298, 52231.40495867769, 52047.75022956841, 51626.0002623639, 53197.23206086842 4, 52402.925357470
81, 50915.650006559095, 51890.00393545848, 52413.74786829332, 53147.054965236784, 51506.62468844287, 51550.242686606325,
52127.4432638069, 51776.85950413223, 51369.867506231145, 51715.85989767808, 52370.785779876685, 51587.9574970484, 51476.7
```

3(vi) Draw the histogram of the test statistic of the bootstrap samples

```python
plt.hist(s)
```

```
(array([   9.,   67.,  356.,  980., 1384., 1279.,  658.,  228.,   36.,
          3.]),
 array([50063.29529057, 50458.67768595, 50854.06008133, 51249.44247672,
        51644.8248721 , 52040.20726748, 52435.58966286, 52830.97205824,
        53226.35445363, 53621.73684901, 54017.11924439]),
 <a list of 10 Patch objects>)
```



Q4) Data cleansing procedure:

- The given excel file has data with around 170 columns, drawing the desired columns like employee gender “Q1” and their salary range “Q9”.
- Removing the null values, blanks and zero values in both the columns using dropna() and drop(0) functions.
- Omitting the string values like “column I do not wish to disclose my approximate yearly compensation” from Q9 column for mean and median calculation.

Test statistic:

- A test statistic calculate the degree of agreement between a sample of data and the null hypothesis.
- A test statistic contains information about the data that is relevant for deciding whether to reject the null hypothesis. The sampling distribution of the test statistic under the null hypothesis is called the null distribution.

In the given scenario the test statistic is found by generating the random 500 rows sample of both male and females data by using below code.

```
Ttest , pval = ttest_1samp(population["Q9"],np.mean(s)) #for sample booststrap
ttest , pval
```

Random Samping:

A simple random sample is a subset of a statistical population in which each member of the subset has an equal probability of being chosen. A simple random sample is meant to be an unbiased representation of a group.

- Here the sample of 500 employees is drawn from the given excel by ensure the sampling strategy is fair.
- Sample data is taken from cleaned data

Confidence Interval:

- A confidence interval is a type of estimate computed from the statistics of the observed data. This gives a range of values for an unknown . The interval will generate a range that might contain the values. When we create the interval, we use a sample mean. Confidence interval can take number of probabilities, most common being a 95 to 99% confidence level.

P -value:

- In statistical hypothesis testing, the p-value or probability value is the probability of obtaining test results at least as extreme as the results actually observed during the test, assuming that the null hypothesis is correct(here p value is >0.05 so we accept the null hypothesis).