Nanditha.V
160119733133

Data Science and Big Data Analytics

Experiment 3: Mean, Median, Mode, Variance, Standard Deviation, Hypothesis Testing

AIM: To calculate the mean, median, mode, variance, standard deviation and perform Hypothesis Testing.

DESCRIPTION:

Hypothesis testing is a key concept in statistics, analytics, and data science. There are four steps to perform Hypothesis Testing:

- Set the Hypothesis
- Set the Significance Level, Criteria for a decision
- Compute the test statistics
- Make a decision

z -tests are a statistical way of testing a hypothesis when either:

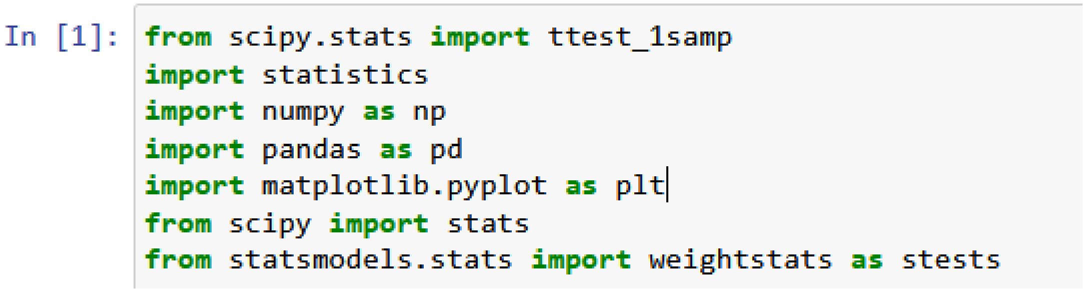
- We know the population variance, or
- We do not know the population variance but our sample size is large n ≥ 30

t-tests are a statistical way of testing a hypothesis when:

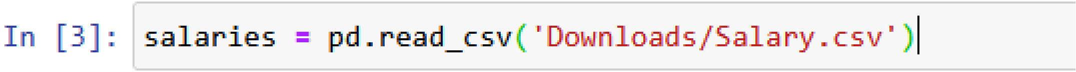
- We do not know the population variance
- Our sample size is small, n < 30

CODE AND ANALYSIS:

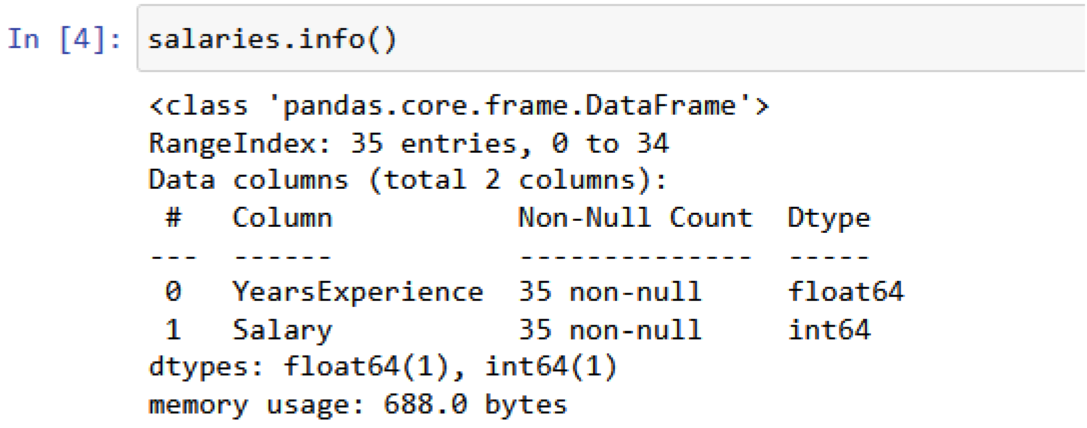
1. Import and get to know the data



```
In [1]: from scipy.stats import ttest_1samp
        import statistics
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from scipy import stats
        from statsmodels.stats import weightstats as stests
```



```
In [3]: salaries = pd.read_csv('Downloads/Salary.csv')
```

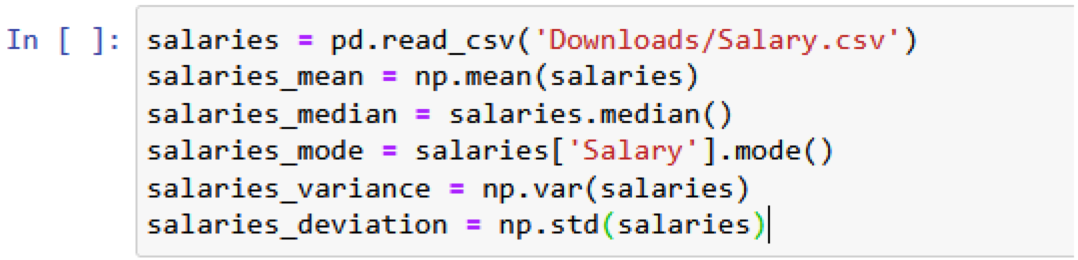


```
In [4]: salaries.info()
```

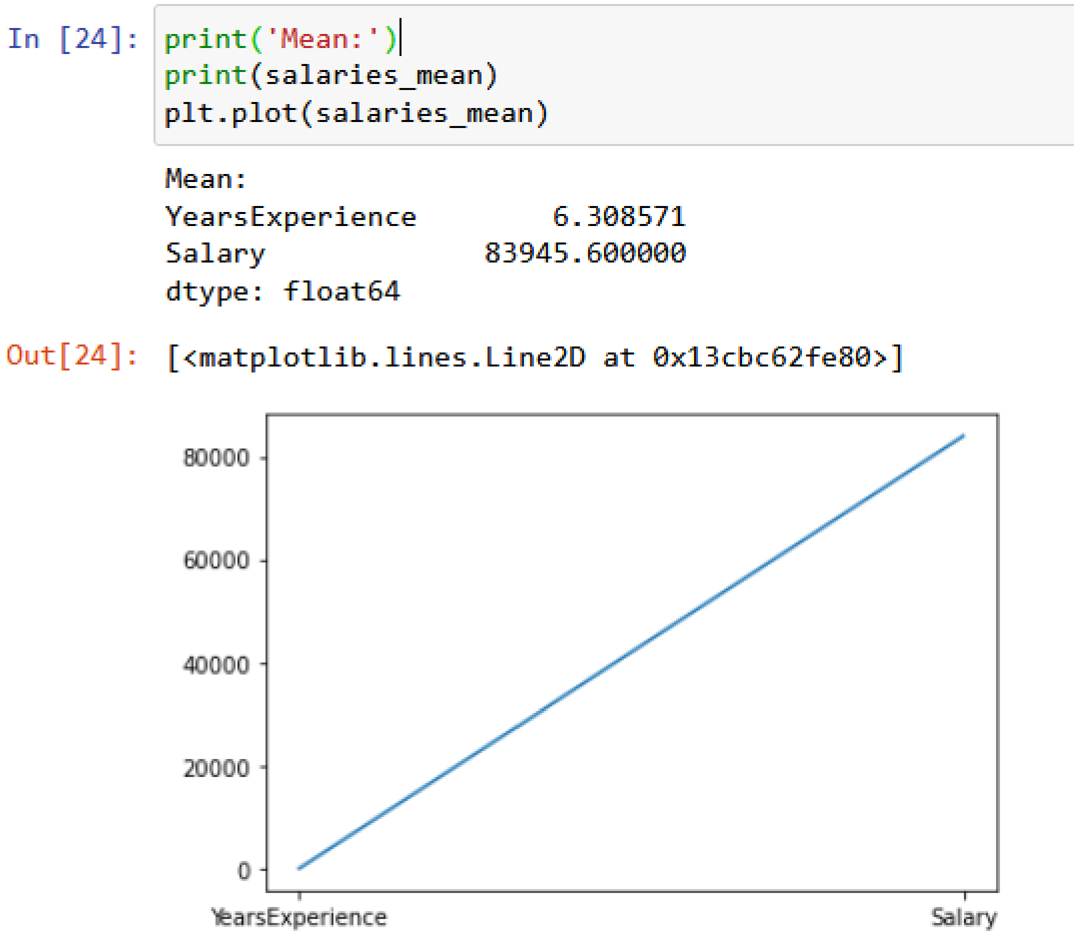
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  35 non-null     float64
 1   Salary           35 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 688.0 bytes
```

OUTPUT ANALYSIS: There are no null values in any of the fields and the data types are also correctly set, therefore code cleaning can be skipped.

2. Calculate the mean, median, mode, variance, standard deviation and plot the graphs



```
In [ ]: salaries = pd.read_csv('Downloads/Salary.csv')
        salaries_mean = np.mean(salaries)
        salaries_median = salaries.median()
        salaries_mode = salaries['Salary'].mode()
        salaries_variance = np.var(salaries)
        salaries_deviation = np.std(salaries)
```

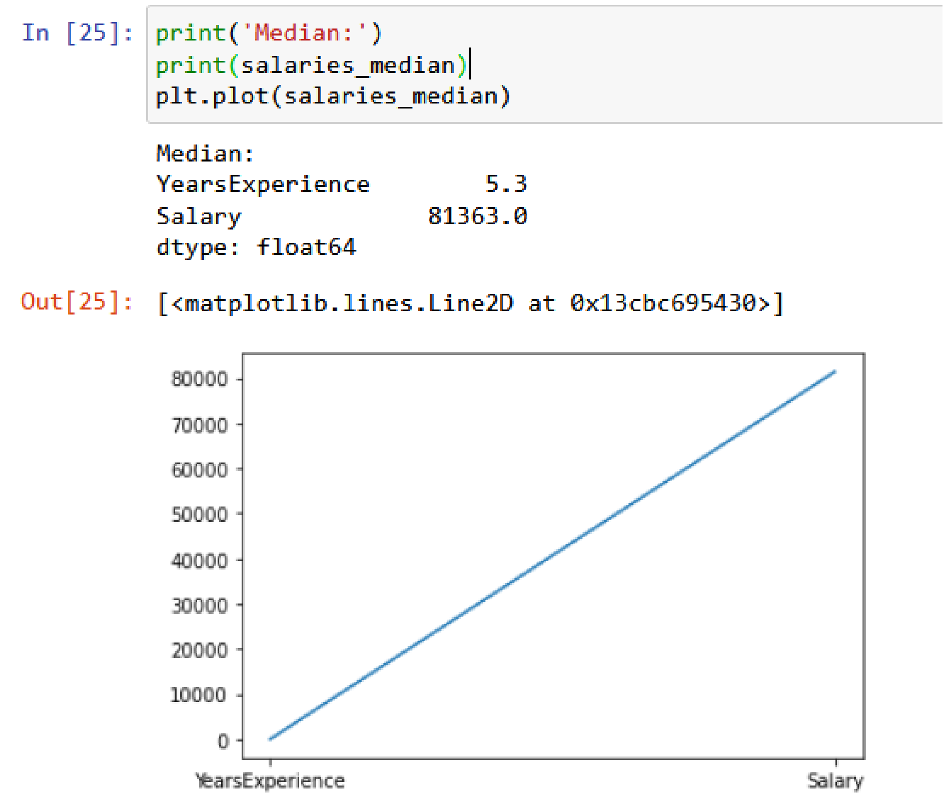


```
In [24]: print('Mean:')
         print(salaries_mean)
         plt.plot(salaries_mean)
```

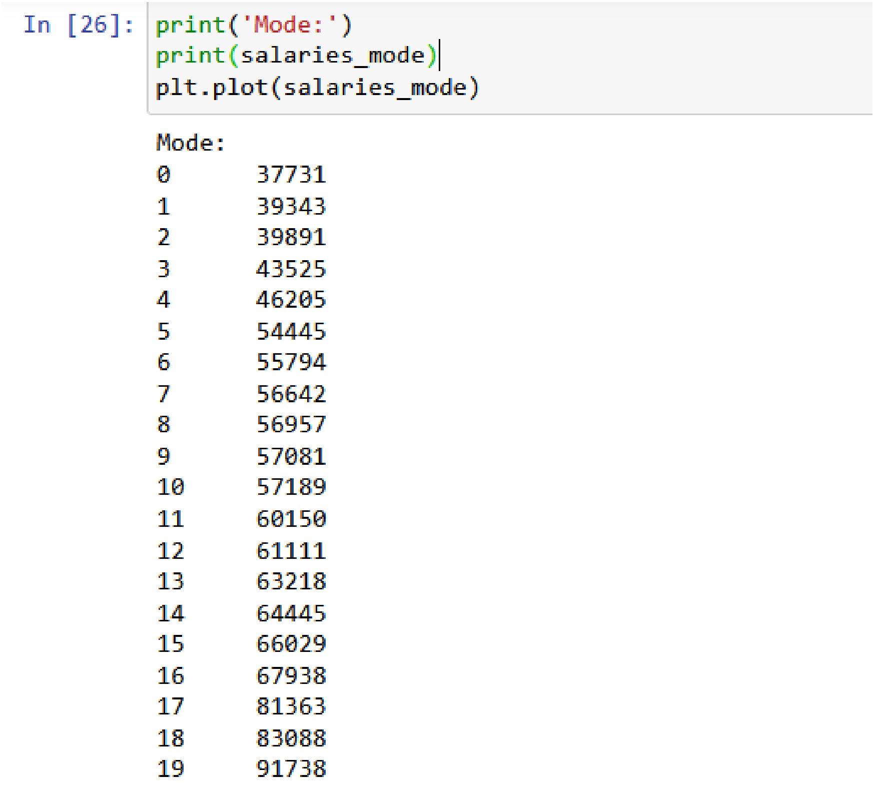
```
Mean:
YearsExperience        6.308571
Salary             83945.600000
dtype: float64
```

```
Out[24]: [<matplotlib.lines.Line2D at 0x13cbc62fe80>]
```

Nanditha.V
160119733133

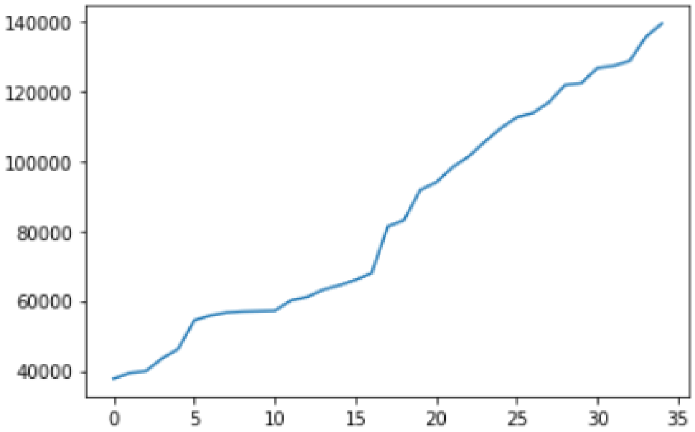


```
In [25]: print('Median:')
         print(salaries_median)
         plt.plot(salaries_median)

         Median:
         YearsExperience        5.3
         Salary             81363.0
         dtype: float64

Out[25]: [<matplotlib.lines.Line2D at 0x13cbc695430>]
```

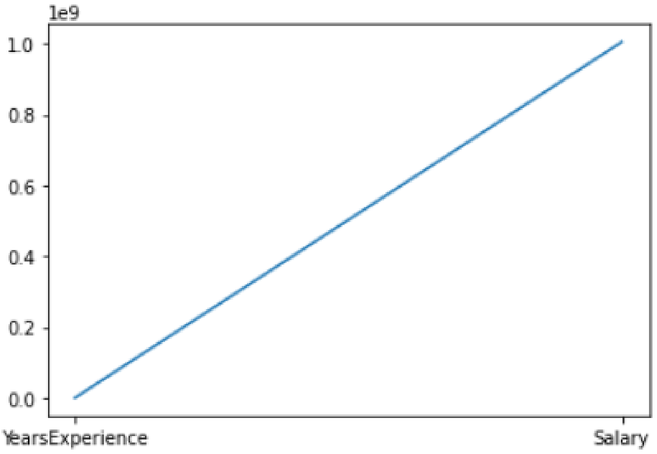


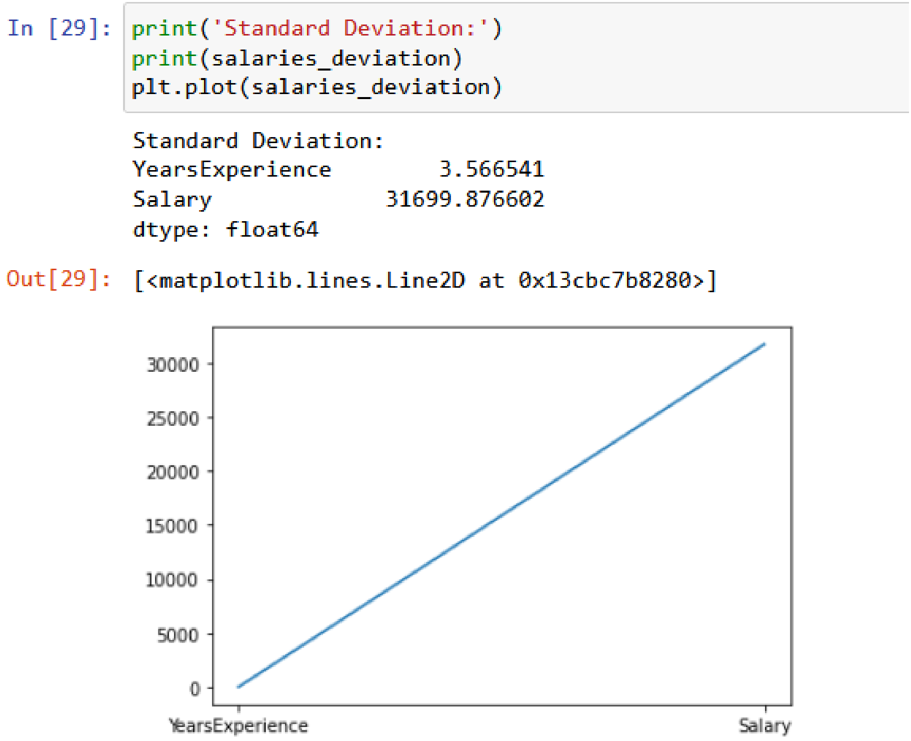
```
In [26]: print('Mode:')
         print(salaries_mode)
         plt.plot(salaries_mode)

         Mode:
         0       37731
         1       39343
         2       39891
         3       43525
         4       46205
         5       54445
         6       55794
         7       56642
         8       56957
         9       57081
         10      57189
         11      60150
         12      61111
         13      63218
         14      64445
         15      66029
         16      67938
         17      81363
         18      83088
         19      91738
```

```
Out[26]: [<matplotlib.lines.Line2D at 0x13cbc6ed9a0>]
```



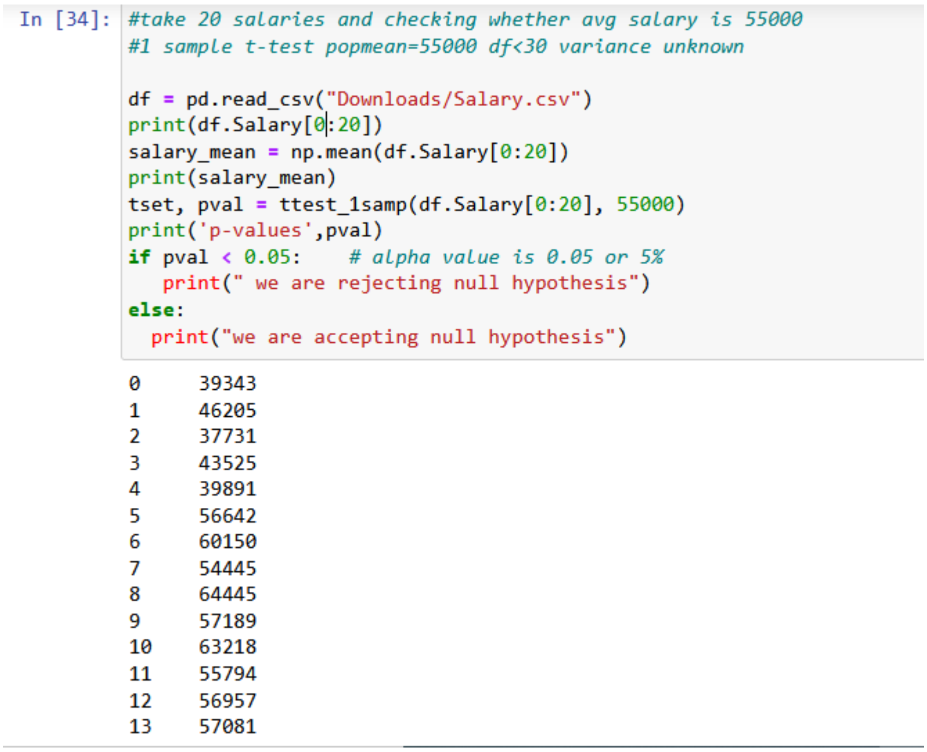
```
In [28]: print('Variance:')
         print(salaries_variance)
         plt.plot(salaries_variance)

         Variance:
         YearsExperience    1.272021e+01
         Salary             1.004882e+09
         dtype: float64

Out[28]: [<matplotlib.lines.Line2D at 0x13cba51b5e0>]
```



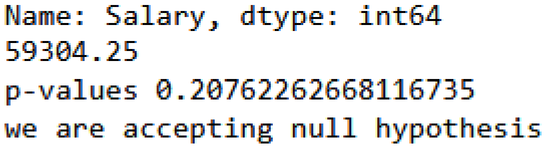


```
In [29]: print('Standard Deviation:')
         print(salaries_deviation)
         plt.plot(salaries_deviation)

         Standard Deviation:
         YearsExperience        3.566541
         Salary             31699.876602
         dtype: float64

Out[29]: [<matplotlib.lines.Line2D at 0x13cbc7b8280>]
```

3. Perform Hypothesis Testing – one sample t-test, two independent sample t-test, z-test

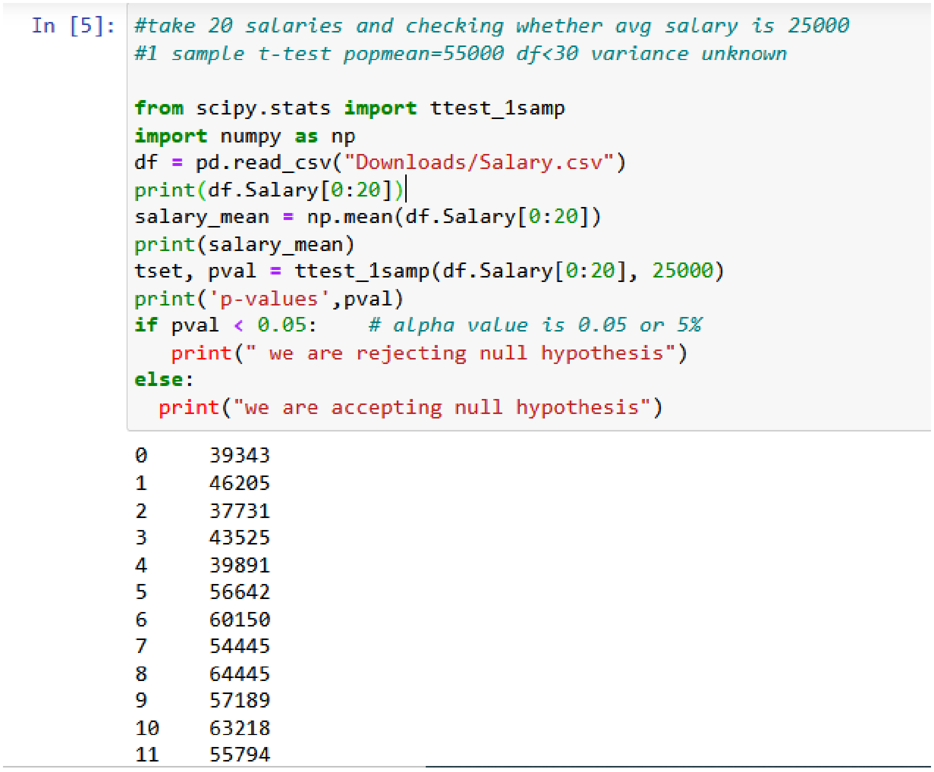


```
In [34]: #take 20 salaries and checking whether avg salary is 55000
         #1 sample t-test popmean=55000 df<30 variance unknown

         df = pd.read_csv("Downloads/Salary.csv")
         print(df.Salary[0:20])
         salary_mean = np.mean(df.Salary[0:20])
         print(salary_mean)
         tset, pval = ttest_1samp(df.Salary[0:20], 55000)
         print('p-values',pval)
         if pval < 0.05:    # alpha value is 0.05 or 5%
            print(" we are rejecting null hypothesis")
         else:
           print("we are accepting null hypothesis")

         0     39343
         1     46205
         2     37731
         3     43525
         4     39891
         5     56642
         6     60150
         7     54445
         8     64445
         9     57189
         10    63218
         11    55794
         12    56957
         13    57081
```



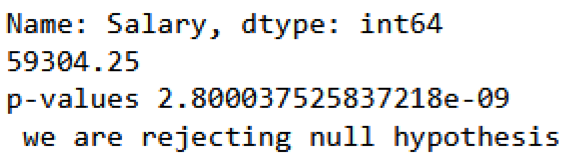
```
Name: Salary, dtype: int64
59304.25
p-values 0.20762262668116735
we are accepting null hypothesis
```

OUTPUT ANALYSIS: Since the sample size is less than 30 and we do not know the population variance, t-test is suitable for hypothesis testing. For the 20 salaries, the average is indeed less than 55000, hence we are accepting the null hypothesis.



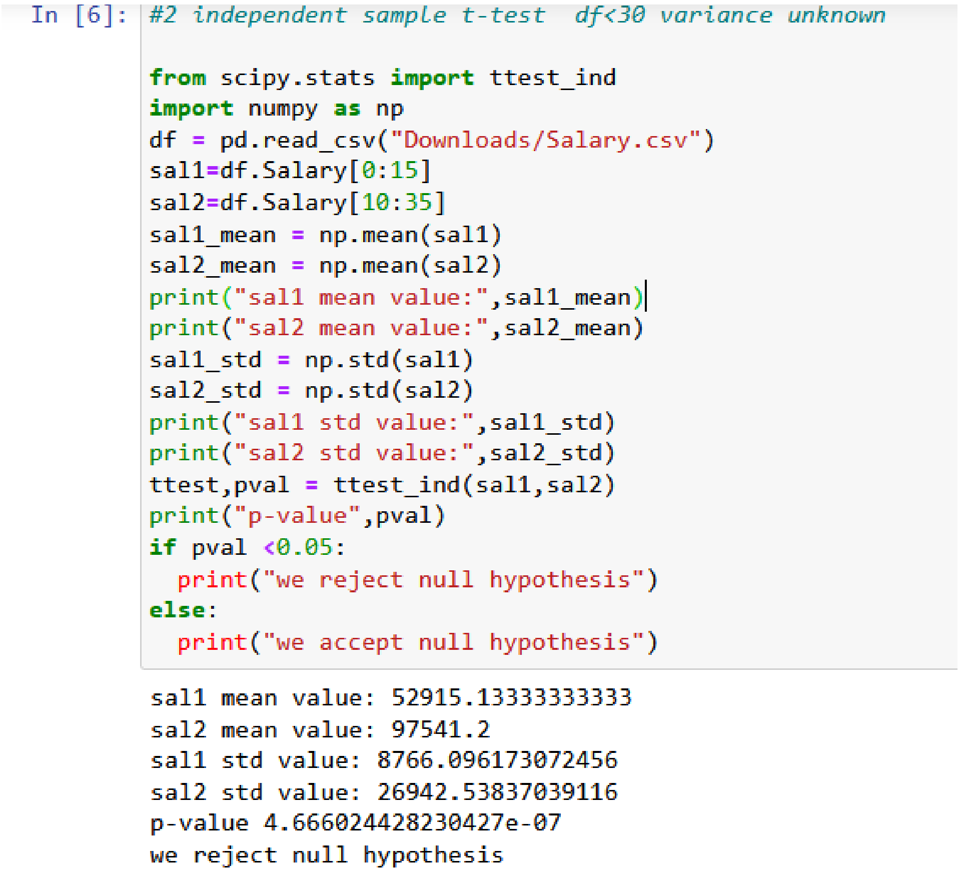
```
In [5]: #take 20 salaries and checking whether avg salary is 25000
        #1 sample t-test popmean=55000 df<30 variance unknown

        from scipy.stats import ttest_1samp
        import numpy as np
        df = pd.read_csv("Downloads/Salary.csv")
        print(df.Salary[0:20])
        salary_mean = np.mean(df.Salary[0:20])
        print(salary_mean)
        tset, pval = ttest_1samp(df.Salary[0:20], 25000)
        print('p-values',pval)
        if pval < 0.05:    # alpha value is 0.05 or 5%
           print(" we are rejecting null hypothesis")
        else:
          print("we are accepting null hypothesis")
```

```
0     39343
1     46205
2     37731
3     43525
4     39891
5     56642
6     60150
7     54445
8     64445
9     57189
10    63218
11    55794
```



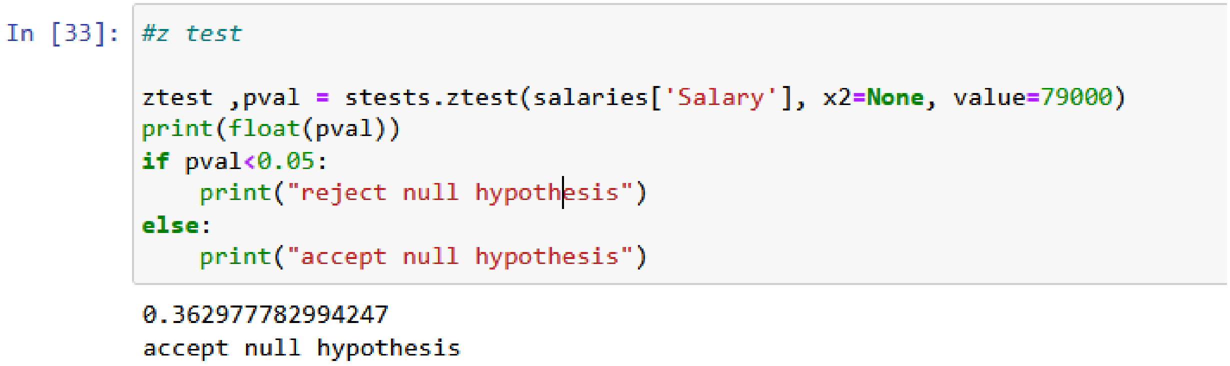
```
Name: Salary, dtype: int64
59304.25
p-values 2.800037525837218e-09
 we are rejecting null hypothesis
```

OUTPUT ANALYSIS: Since the sample size is less than 30 and we do not know the population variance, t-test is suitable for hypothesis testing. For the 20 salaries, the average is not less than 25000, hence we are rejecting the null hypothesis.



```
In [6]: #2 independent sample t-test  df<30 variance unknown

        from scipy.stats import ttest_ind
        import numpy as np
        df = pd.read_csv("Downloads/Salary.csv")
        sal1=df.Salary[0:15]
        sal2=df.Salary[10:35]
        sal1_mean = np.mean(sal1)
        sal2_mean = np.mean(sal2)
        print("sal1 mean value:",sal1_mean)
        print("sal2 mean value:",sal2_mean)
        sal1_std = np.std(sal1)
        sal2_std = np.std(sal2)
        print("sal1 std value:",sal1_std)
        print("sal2 std value:",sal2_std)
        ttest,pval = ttest_ind(sal1,sal2)
        print("p-value",pval)
        if pval <0.05:
          print("we reject null hypothesis")
        else:
          print("we accept null hypothesis")

        sal1 mean value: 52915.13333333333
        sal2 mean value: 97541.2
        sal1 std value: 8766.096173072456
        sal2 std value: 26942.53837039116
        p-value 4.666024428230427e-07
        we reject null hypothesis
```

OUTPUT ANALYSIS: Since the sample size is less than 30 and we do not know the population variance, t-test is suitable for hypothesis testing.



```
In [33]: #z test

         ztest ,pval = stests.ztest(salaries['Salary'], x2=None, value=79000)
         print(float(pval))
         if pval<0.05:
             print("reject null hypothesis")
         else:
             print("accept null hypothesis")

         0.362977782994247
         accept null hypothesis
```

OUTPUT ANALYSIS: Since the sample size is greater than 30, z-test is suitable for hypothesis testing. For the given salaries, the average is less than 79000, hence we are accepting the null hypothesis.