Double-click (or enter) to edit

Project-2:Project for Statistical Learning

The Titan Insurance Company has just installed a new incentive payment scheme for its lift policy sales force. It wa success or failure of the new scheme. Indications are that the sales force is selling more policies, but sales always from month to month and it is not clear that the scheme has made a significant difference.

Life Insurance companies typically measure the monthly output of a salesperson as the total sum assured for the p the month. For example, suppose salesperson X has, in the month, sold seven policies for which the sums assured £10000, £35000. X's output for the month is the total of these sums assured, £61,500. Titan's new scheme is that th salaries but are paid large bonuses related to their output (i.e. to the total sum assured of policies sold by them). Th company, but they are looking for sales increases which more than compensate. The agreement with the sales forc least break even for the company, it will be abandoned after six months.

The scheme has now been in operation for four months. It has settled down after fluctuations in the first two month

To test the effectiveness of the scheme, Titan have taken a random sample of 30 salespeople measured their outpu to changeover and then measured it in the fourth month after the changeover (they have deliberately chosen month changeover). The outputs of the salespeople are shown in Table 1

SALESPERSON	Old Scheme (in thousands)	New Scheme (in thousands)
1	57	62
2	103	122
3	59	54
4	75	82
5	84	84
6	73	86
7	35	32
8	110	104
9	44	38
10	82	107
11	67	84
12	64	85
13	78	99
14	53	39

15	41	34
16	39	58
17	80	73
18	87	53
19	73	66
20	65	78
21	28	41
22	62	71
23	49	38
24	84	95
25	63	81
26	77	58
27	67	75
28	101	94
29	91	100
30	50	68

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
```

Double-click (or enter) to edit

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=9473189

Enter your authorization code:
··········
Mounted at /content/drive
```

```
data = [[1,57,62],
[2,103,122],
[3,59,54],
[4,75,82],
[5,84,84],
[6,73,86],
[7,35,32],
[8,110,104],
[9,44,38],
[10,82,107],
[11,67,84],
[12,64,85],
[13,78,99],
[14,53,39],
[15,41,34],
[16,39,58],
[17,80,73],
[18,87,53],
[19,73,66],
[20,65,78],
[21,28,41],
[22,62,71],
[23,49,38],
[24,84,95],
[25,63,81],
[26,77,58],
[27,67,75],
[28,101,94],
[29,91,100],
[30,50,68]]

df = pd.DataFrame(data)
df.columns = ['SALESPERSON','Old Scheme (in thousands)','New Scheme (in thousands)']
df.head()
```

	SALESPERSON	Old Scheme (in thousands)	New Scheme (in thousands)
0	1	57	62
1	2	103	122
2	3	59	54
3	4	75	82
4	5	84	84

Q1 Find the mean of old scheme and new scheme column. (5 points)

Preparing Data

```
# Mean, Standard Deviation and Variance of both the scheme
df['Old Scheme (in thousands)'] = df['Old Scheme (in thousands)'] * 1000
df['New Scheme (in thousands)'] = df['New Scheme (in thousands)'] * 1000
print('Mean of Old Scheme - ',df['Old Scheme (in thousands)'].mean())
print('Varaince in Old Scheme - ',df['Old Scheme (in thousands)'].var())
print('Standard Deviation of Old Scheme - ',df['Old Scheme (in thousands)'].std())
print("---------------------------------------------------------------------------------")
print('Mean of New Scheme - ',df['New Scheme (in thousands)'].mean())
print('Varaince in New Scheme - ',df['New Scheme (in thousands)'].var())
print('Standard Deviation of New Scheme - ',df['New Scheme (in thousands)'].std())
```

```
Mean of Old Scheme -  68033.33333333333
Varaince in Old Scheme -  418447126.4367815
Standard Deviation of Old Scheme -  20455.98021207445
-------------------------------------------------------------------------------
```

Q2 Use the five percent significance test over the data to determine the p value to check new scheme has significa

```
# Five Point Summary of old scheme

print("FIVE POINT SUMMARY of Old Scheme")
print ("Ist POINT: Smallest of Old Scheme - ",df['Old Scheme (in thousands)'].min())
print ("IInd POINT: Q1 quantile of Old Scheme - ",np.quantile(df['Old Scheme (in thousands)'], .25))
print ("IIIrd POINT: Median Old Scheme - ",df['Old Scheme (in thousands)'].median())
print ("IVth POINT: Q3 quantile of Old Scheme - ",np.quantile(df['Old Scheme (in thousands)'], .75))
print ("Vth POINT: Largest of Old Scheme - ",df['Old Scheme (in thousands)'].max())
```

```
FIVE POINT SUMMARY of Old Scheme
Ist POINT: Smallest of Old Scheme -  28000
IInd POINT: Q1 quantile of Old Scheme -  54000.0
IIIrd POINT: Median Old Scheme -  67000.0
IVth POINT: Q3 quantile of Old Scheme -  81500.0
Vth POINT: Largest of Old Scheme -  110000
```

```
# Five Point Summary of new scheme
print("FIVE POINT SUMMARY of New Scheme")
print ("Ist POINT: Smallest of New Scheme - ",df['New Scheme (in thousands)'].min())
print ("IInd POINT: Q1 quantile of New Scheme - ",np.quantile(df['New Scheme (in thousands)'], .25))
print ("IIIrd POINT: Median Old Scheme - ",df['New Scheme (in thousands)'].median())
print ("IVth POINT: Q3 quantile of New Scheme - ",np.quantile(df['New Scheme (in thousands)'], .75))
print ("Vth POINT: Largest of New Scheme - ",df['New Scheme (in thousands)'].max())
```

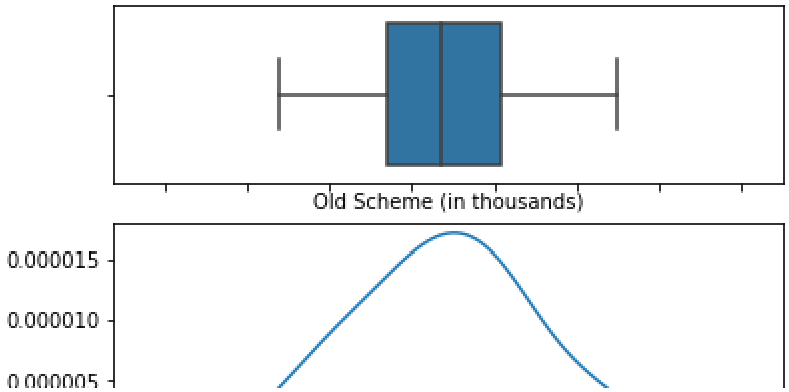
```
FIVE POINT SUMMARY of New Scheme
Ist POINT: Smallest of New Scheme -  32000
IInd POINT: Q1 quantile of New Scheme -  55000.0
IIIrd POINT: Median Old Scheme -  74000.0
IVth POINT: Q3 quantile of New Scheme -  85750.0
Vth POINT: Largest of New Scheme -  122000
```

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Cut the window in 2 parts
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.150, .185)})

# Add a graph in each part
sns.boxplot(df['Old Scheme (in thousands)'], ax=ax_box)
sns.distplot(df['Old Scheme (in thousands)'],hist = False, bins=(1000,1000), ax=ax_hist)
```

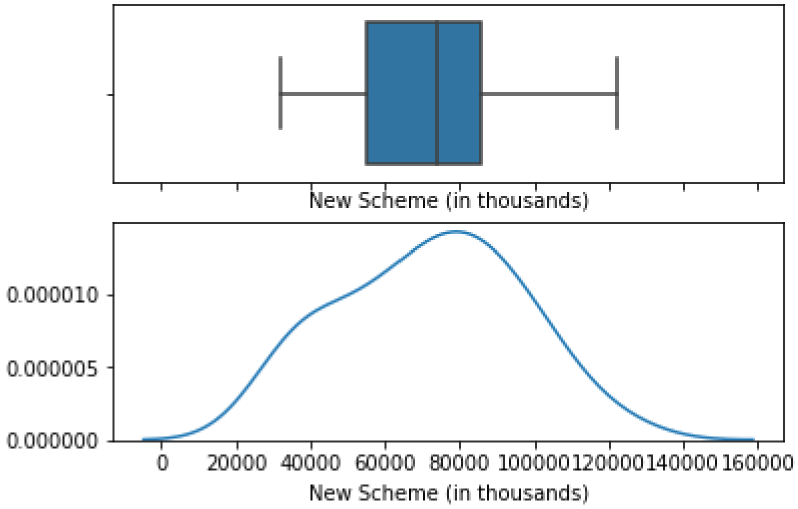
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcc4ac0bc18>
```



```
# Cut the window in 2 parts
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.150, .185)})

# Add a graph in each part
sns.boxplot(df['New Scheme (in thousands)'], ax=ax_box)
sns.distplot(df['New Scheme (in thousands)'],hist = False, bins=(1000,1000), ax=ax_hist)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcc4ab36ac8>
```



Inference: Old scheme data is more normally distributed, in comparission to New scheme data

Hypothesis

- Titan is looking, whether the new scheme has significantly raised the output, my guess we can find it by doinç

mu1 = Average sums assured by salesperson before changeover to new scheme. mu2 = Average sums assured by ε new scheme.

H0: mu1 = mu2 (mu2 - m1 = 0), i.e. No significant effect on sales after switching to new scheme.

HA: mu1 < mu2 (mu2-mu1 > 0), i.e. Significant effect on sales after switching to new scheme.

```
from scipy import stats
from scipy.stats import ttest_ind
from scipy.stats import norm

## one sample t-test
## Given Sample Size: n = 30
## Alpha : level of significance = 0.05
```

```
t_statistic, p_value = ttest_ind(df['New Scheme (in thousands)'],df['Old Scheme (in thousands)'])

print ("The t-statistic is: ", t_statistic)
print ("The p-value is: ", p_value)

# p_value > 0.05 => null hypothesis:
if (p_value > 0.05):
    print("Since p-value is higher than 0.05, we accept NULL hypothesis. The new scheme has NOT sign
else:
    print("Since p-value is lesser than 0.05, we accept Alternate hypothesis. The new scheme has sig
```

```
The t-statistic is:  0.6937067608923764
The p-value is:  0.49063515686248105
Since p-value is higher than 0.05, we accept NULL hypothesis. The new scheme has NOT sig
```

Q3 - What conclusion does the test (p-value) lead to? (2.5 points)

Here the p value is 0.49 which is greater than the 0.05. Hence accept the null hypothesis that the new scheme did nc

Q4 Suppose it has been calculated that in order for Titan to break even, the average output must increase by £500 old scheme. If this figure is alternative hypothesis, what is:

```
        a) The probability of a type 1 error? (2.5 points)

         b) What is the p- value of the hypothesis test if we test for a
                    difference of $5000? (10 points)

        c) Power of the test (5 points)
```

4.a) The probability of a type 1 error?

Probability of Type I error is significant level - i.e. 0.05 or 5%

Q 4. b) What is the p- value of the hypothesis test if we test for a difference of $5000? (10 points)

Let mu2 = Average sums assured by salesperson after changing to new scheme. mu1 = Average sums assured by s

MeanDeviation = Mu2 – Mu1

H0: MeanDeviation ≤ 5000 HA: MeanDeviation > 5000

```
t_statistic, p_value = ttest_ind(df['New Scheme (in thousands)'],df['Old Scheme (in thousands)'])

print ("The t-statistic is: ", t_statistic)
print ("The p-value is: ", p_value)

# p_value > 0.05 => null hypothesis:
if (p_value > 0.05):
    print("Since p-value is higher than 0.05, we accept NULL hypothesis. Mean Deviation is not great
else:
    print("Since p-value is lesser than 0.05, we accept Alternate hypothesis. Mean Deviation is grea

print ("Mean of Difference - ",(df['New Scheme (in thousands)'].mean())-(df['Old Scheme (in thousand
The t-statistic is:  0.6937067608923764
The p-value is:  0.49063515686248105
Since p-value is higher than 0.05, we accept NULL hypothesis. Mean Deviation is not grea
Mean of Difference -  4000.0
```

Q4.c) Power of the test (5 points)

Assuming mu2 = Average sums assured by salesperson after changing to new scheme.

mu1 = Average sums assured by salesperson before new scheme.

MeanDeviation = Mu2 – Mu1 = $4000.00

H0: MeanDeviation = 4000
HA: MeanDeviation > 0

Power = 1 - Type II Error

```python
from statsmodels.stats.power import ttest_power


import math

s1_pow = pow(df['Old Scheme (in thousands)'].std(),2)
s2_pow = pow(df['New Scheme (in thousands)'].std(),2)


effect_size = (df['New Scheme (in thousands)'].mean() - df['Old Scheme (in thousands)'].mean())


post_mean = df['New Scheme (in thousands)'].mean()
pre_mean = df['Old Scheme (in thousands)'].mean()

mean_diff = pre_mean - post_mean


denom = np.sqrt((29*df['Old Scheme (in thousands)'].var())+(29*df['New Scheme (in thousands)'].var()

effect_size = mean_diff/denom

print(ttest_power(effect_size = round(effect_size,2), nobs=30, alpha=0.05, alternative="two-sided"))
```

```
0.05516092434366573
```

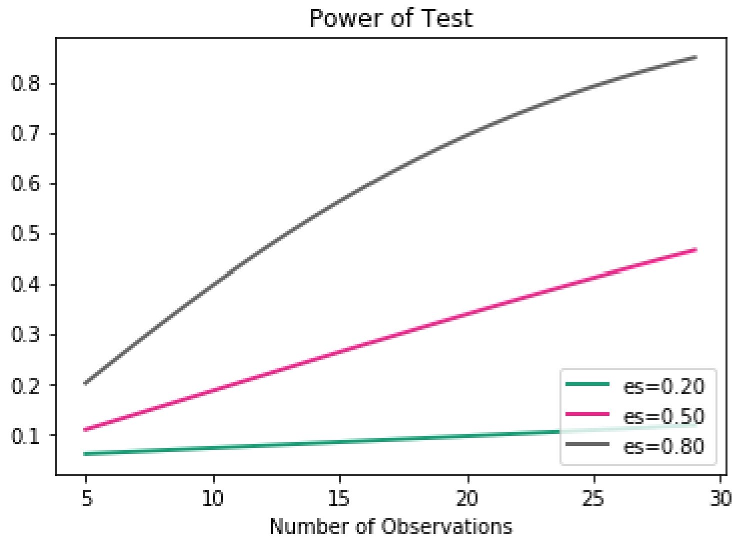
```python
from matplotlib import pyplot
from statsmodels.stats.power import TTestIndPower



analysis = TTestIndPower()
analysis.power(effect_size = round(effect_size,2), nobs1=30, alpha=0.05, alternative="two-sided")
```

```
0.052663806596821724
```

```python
effect_sizes = array([0.2, 0.5, 0.8])
sample_sizes = array(range(5, 30))
analysis.plot_power(dep_var='nobs', nobs=sample_sizes, effect_size=effect_sizes)
pyplot.show()
```


Power of Test
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
5
10
15
20
25
30
Number of Observations
es=0.20
es=0.50
es=0.80