```
# Please find below the Project for Statistical Learning course. This is an individual assignment. K
```

The Titan Insurance Company has just installed a new incentive payment scheme for its lift policy sales force. It wa failure of the new scheme. Indications are that the sales force is selling more policies, but sales always vary in an ur 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 example, suppose salesperson X has, in the month, sold seven policies for which the sums assured are £1000, £25( the month is the total of these sums assured, £61,500. Titan's new scheme is that the sales force receives low regul their output (i.e. to the total sum assured of policies sold by them). The scheme is expensive for the company, but th than compensate. The agreement with the sales force is that if the scheme does not at least break even for the com

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 and then measured it in the fourth month after the changeover (they have deliberately chosen months not too close salespeople are shown in Table 1

Questions

Find the mean of old scheme and new scheme column. (5 points) Use the five percent significance test over the dat has significantly raised outputs? (10 points) What conclusion does the test (p-value) lead to? (2.5 points) Suppose it break even, the average output must increase by £5000 in the scheme compared to the old scheme. If this figure is a 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?

c) Power of the test (5 points)

```
## https://www.datacamp.com/community/tutorials/web-scraping-using-python

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
% matplotlib inline
import seaborn as sns
%matplotlib inline
from scipy.stats import ttest_1samp,ttest_ind,wilcoxon
from statsmodels.stats.power import ttest_power
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Data.xlsx

- **Data.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 9158 bytes, last modifie

```
Saving Data.xlsx to Data (1).xlsx
```

```
import io
df = pd.read_excel(io.BytesIO(uploaded['Data.xlsx']))
```

```
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 |

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
SALESPERSON                  30 non-null int64
Old Scheme (in thousands)    30 non-null int64
New Scheme (in thousands)    30 non-null int64
dtypes: int64(3)
memory usage: 800.0 bytes
```

```
df.columns
```

```
Index(['SALESPERSON', 'Old Scheme (in thousands)',
       'New Scheme (in thousands)'],
      dtype='object')
```

```
df.SALESPERSON = pd.Categorical(df.SALESPERSON)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
SALESPERSON                  30 non-null category
Old Scheme (in thousands)    30 non-null int64
New Scheme (in thousands)    30 non-null int64
dtypes: category(1), int64(2)
memory usage: 2.1 KB
```

```
df.describe()
```

| | Old Scheme (in thousands) | New Scheme (in thousands) |
|---|---|---|
| **count** | 30.000000 | 30.000000 |
| **mean** | 68.033333 | 72.033333 |
| **std** | 20.455980 | 24.062395 |
| **min** | 28.000000 | 32.000000 |
| **25%** | 54.000000 | 55.000000 |
| **50%** | 67.000000 | 74.000000 |
| **max** | 110.000000 | 122.000000 |

```
df1=pd.DataFrame(df)
```

```
df1.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 |

```
df1['New']="NEW"
df1['Old']="OLD"
```

```
df1.head()
```

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

```
list1 = np.array(df1['Old Scheme (in thousands)'])
list1
```

```
array([ 57, 103,  59,  75,  84,  73,  35, 110,  44,  82,  67,  64,  78,
        53,  41,  39,  80,  87,  73,  65,  28,  62,  49,  84,  63,  77,
        67, 101,  91,  50])
```

```
list2 = np.array(df1['New Scheme (in thousands)'])
list2
```

```
array([ 62, 122,  54,  82,  84,  86,  32, 104,  38, 107,  84,  85,  99,
        39,  34,  58,  73,  53,  66,  78,  41,  71,  38,  95,  81,  58,
        75,  94, 100,  68])
```

```
scheme = np.append(list1,list2)
```

```
scheme
```

```
array([ 57, 103,  59,  75,  84,  73,  35, 110,  44,  82,  67,  64,  78,
        53,  41,  39,  80,  87,  73,  65,  28,  62,  49,  84,  63,  77,
        67, 101,  91,  50,  62, 122,  54,  82,  84,  86,  32, 104,  38,
       107,  84,  85,  99,  39,  34,  58,  73,  53,  66,  78,  41,  71,
        38,  95,  81,  58,  75,  94, 100,  68])
```

```
list3 = np.array(df1['Old'])
list4 = np.array(df1['New'])
scheme_type = np.append(list3,list4)
```

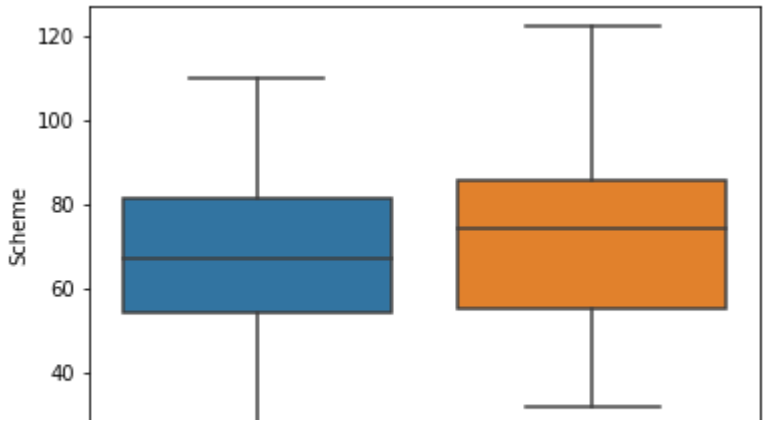
```
df_n = pd.DataFrame({'Scheme_type':scheme_type,'Scheme':scheme})
```

```
df_n.head()
```

| | Scheme_type | Scheme |
|---|---|---|
| **0** | OLD | 57 |
| **1** | OLD | 103 |
| **2** | OLD | 59 |
| **3** | OLD | 75 |
| **4** | OLD | 84 |

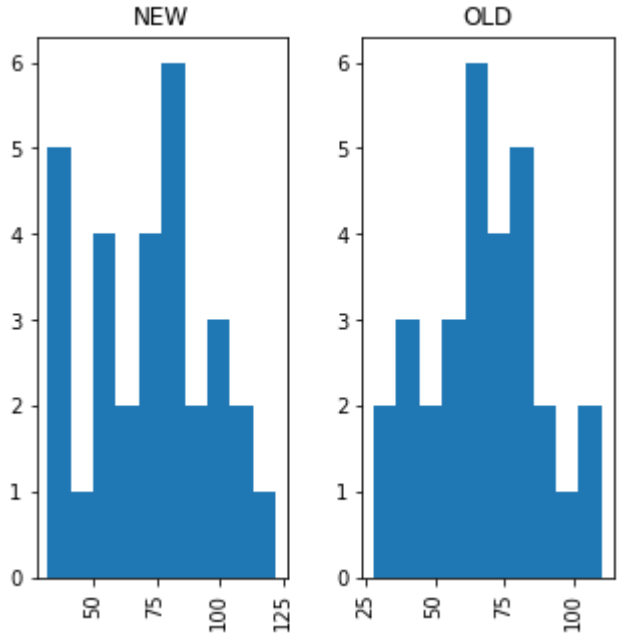
```
sns.boxplot(y='Scheme',x='Scheme_type',data=df_n)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff11917ec18>
```



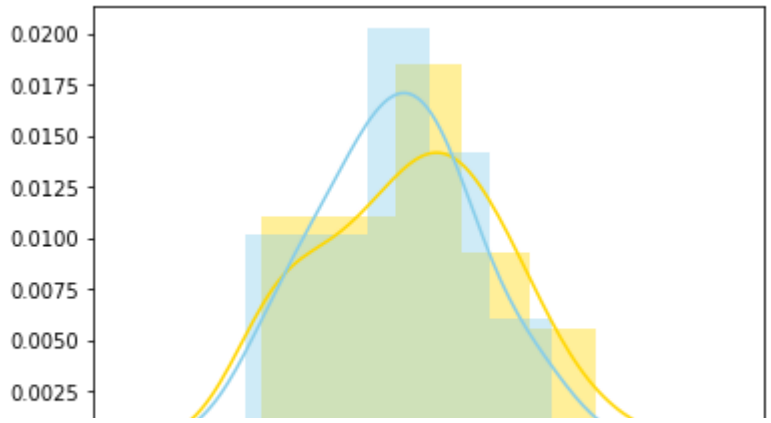

```
df_n.hist(by='Scheme_type',column='Scheme',figsize=(5,5))
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x7ff1184e49e8>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7ff1184337b8>],
      dtype=object)
```




```
sns.distplot(df_n[df_n['Scheme_type']=='NEW']['Scheme'],color='gold')
sns.distplot(df_n[df_n['Scheme_type']=='OLD']['Scheme'],color='skyblue')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff118583278>
```




```
df.columns
```

```
Index(['SALESPERSON', 'Old Scheme (in thousands)', 'New Scheme (in thousands)',
       'New', 'Old'],
      dtype='object')
```

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

old , new = df['Old Scheme (in thousands)'].mean() , df['New Scheme (in thousands)'].mean()

'Old scheme mean is = {} and new scheme mean is = {}'.format(old , new)
```

```
'Old scheme mean is = 68.03333333333333 and new scheme mean is = 72.03333333333333'
```

```
## Use the five percent significance test over the data to determine the p value to check new scheme

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


# H0 =  mean output is same for old and new scheme
# H1 =  mean output of new scheme is greater than old scheme


t_statistic, p_value = ttest_1samp(df['New Scheme (in thousands)']-df['Old Scheme (in thousands)'],0

print(t_statistic, p_value)  ## p value >> 0.05 so we accept Null hypothesis and says there is no si
```

```
1.5559143823544377 0.13057553961337662
```

```
z_statistic, p_value = wilcoxon(df['New Scheme (in thousands)']-df['Old Scheme (in thousands)'])

print(z_statistic, p_value)  ## p value > 0.05 so we accept Null hypothesis and says there is no sig
```

```
131.0 0.06116952762758769
```

```
Zstats = (np.mean(df['New Scheme (in thousands)']) - np.mean(df['Old Scheme (in thousands)']))/np.st
```

```
ttest_power(Zstats,nobs=len(np.array(df['New Scheme (in thousands)']))-1,alpha=0.05,alternative='lar
```

```
0.22474055598474652
```

```
## with ttest_power there is only 22% chance to reject null hypothesis , so there is no significance
```

```
##  Suppose it has been calculated that in order for Titan to break even, the average output must in
## If this figure is alternative hypothesis, what is: a) The probability of a type 1 error? (2.5 poi

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

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

```
t_statistic, p_value = ttest_1samp(df['New Scheme (in thousands)']-df['Old Scheme (in thousands)']-5
print(t_statistic, p_value)  ## p value >> 0.05 so we accept Null hypothesis and says there is no si
```

```
-0.3889785955886094 0.7001334912613286
```

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

## Ans - The probability of a type 1 error 70 %
```

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

```
dollar =5000
val = (dollar * 0.80)/1000
val
```

```
4.0
```

```
t_statistic, p_value = ttest_1samp(df['New Scheme (in thousands)']-df['Old Scheme (in thousands)']-v
print(t_statistic, p_value)  ## p- value of the hypothesis test if we test for a difference of $5000
```

```
0.0 1.0
```

```
## c ) power of test

Zstats = (5)/np.std(df['New Scheme (in thousands)'])
ttest_power(Zstats,nobs=len(np.array(df['New Scheme (in thousands)']))-1,alpha=0.05,alternative='lar
```

```
0.29660245254588913
```

```
## with ttest_power there is only 29% chance to reject null hypothesis , so there is no significance
```