```
import numpy as np
import pandas as pd
```

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
data = pd.read_csv('/content/drive/My Drive/4aug/titan.csv')
```

```
data
```

Saved successfully!

	SALESPERSON	Old Scheme	New Scheme
0	1	57	62
1	2	103	122
2	3	59	54
3	4	75	82
4	5	84	84
5	6	73	86
6	7	35	32
7	8	110	104
8	9	44	38
9	10	82	107
10	11	67	84
11	12	64	85
12	13	78	99
13	14	53	39
14	15	41	34
15	16	39	58
16	17	80	73
17	18	87	53
18	19	73	66
19	20	65	78
20	21	28	41
21	22	62	71
22	23	49	38
23	24	84	95
			81
25	26	77	58
26	27	67	75

Saved successfully!

```
from scipy.stats import zscore
from sklearn.preprocessing import Imputer
from sklearn.metrics import accuracy_score
import seaborn as sns
import os
%matplotlib inline


from scipy.stats import ttest_1samp, wilcoxon
from statsmodels.stats.power import ttest_power
import matplotlib.pyplot as plt
```

```
#1. Find the mean of old scheme and new scheme column.
```

```
data.loc[:,"Old Scheme"].mean()
```

```
68.03333333333333
```

```
data.loc[:,"New Scheme"].mean()
```

```
72.03333333333333
```

```
data.describe()
```

	SALESPERSON	Old Scheme	New Scheme
count	30.000000	30.000000	30.000000
mean	15.500000	68.033333	72.033333
std	8.803408	20.455980	24.062395
min	1.000000	28.000000	32.000000
25%	8.250000	54.000000	55.000000
50%	15.500000	67.000000	74.000000
75%	22.750000	81.500000	85.750000
max	30.000000	110.000000	122.000000

```
from scipy.stats import *
# Use the five percent significance test over the data to determine the p value to check new sche
```

```
t_statistic,p_value= stats.ttest_ind(data['Old Scheme'] , data['New Scheme'])
```

```
print(t_statistic,p_value)
```

```
-0.6937067608923764 0.49063515686248105
```

```
# here in this case we p value is 0.13 which is gretaer than 0.05
                                    - that new schemen does not raise the output significantly
```

Saved successfully!

```
#The probability of a type 1 error?
#prob  of type 1 error = significant level = 0.05 or 5%
```

```
zstats=(5)/np.std((data['New Scheme']))
```

```
zstats
```

```
0.2113453913629282
```

```
ttest_power(zstats,nobs=29,alpha=0.05, alternative='larger')
```

```
#power of test
```

```
0.29660245254588913
```

Saved successfully!