Annexure 2

Applied Computational Statistics

Project Report

Faculty name Sonal Saurabh

Student name J Vishwanath

Roll No.: 20csu296

Semester: DSA-1

Group:

Department of Computer Science and Engineering

The NorthCap University, Gurugram- 122001, India

Session 2021-22

Table of Contents

S.No		Page No.
1.	Project Description	
2.	Problem Statement	
3.	Analysis 3.1 Hardware Requirements 3.2 Software Requirements	
4.	Design 4.1 Data/Input Output Description: 4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps	
5.	Implementation and Testing (stage/module wise)	
6.	Output (Screenshots)	
7.	Conclusion and Future Scope	

Sawyam
Smirage (45)
~~Nikhil~~
Mridul
Vian
Yuti

CASE STUDY 5

Confidence interval for difference of two means, dependent samples

Weight loss example, kg

Background	Somebody has developed a diet and an exercise program for losing weight. It seems that it works like a charm. However, you are interested in how much weight are you likely to lose. You have a sample of 10 people who have already completed the 12-week program.
Task 1	Calculate the mean and standard deviation of the dataset
Task 2	Determine the appropriate statistic to use
Task 3	Calculate the 95% confidence interval
Task 4	Interpret the result and see if the diet plan is effective or not
Optional	You can try to calculate the 90% and 99% confidence intervals to see the difference. There is no solution provided for these cases.

Subject	Weight before (kg)	Weight after (kg)	Difference
1	103.68	92.87	-10.81
2	110.68	101.58	-9.10
3	119.05	105.66	-13.39
4	101.75	96.18	-5.57
5	91.69	86.97	-4.72
6	112.03	105.90	-6.13
7	88.84	80.56	-8.28
8	105.18	97.00	-8.18
9	110.37	99.27	-11.10
10	120.99	107.44	-13.55

Weight Loss, A Case Study

Background	Somebody has developed a diet and an exercise program for losing weight. It seems that it works like a charm. However, you are interested in how much weight are you likely to lose. You have a sample of 10 people who have already completed the 12-week program.
Task 1	Calculate the mean and standard deviation of the dataset
Task 2	Determine the appropriate statistic to use
Task 3	Calculate the 95% confidence interval
Task 4	Interpret the result and see if the diet plan is effective or not
Optional	You can try to calculate the 90% and 99% confidence intervals to see the difference. There is no solution provided for these cases.

Weights Before and after

The DataSets

```
#Before Weight
xi = [103.68,110.68,119.05,101.75,91.69,112.03,88.84,105.18,110.37,120.99]
x1 = pd.Series(xi)
```

[4] ✓ 0.1s

```
#After Weight
xj = [92.87,101.58,105.66,96.18,86.97,105.90,80.56,97.00,99.27,107.44]
x2 = pd.Series(xj)
```

[5] ✓ 0.9s

DataSet Overview

```
print(x1.describe() , "\n",x2.describe())
```

[16] ✓ 0.1s

```
count     10.000000
mean     106.426000
std       10.508764
min       88.840000
25%      102.232500
50%      107.775000
75%      111.692500
max      120.990000
dtype: float64
 count     10.00000
mean      97.34300
std        8.67151
min       80.56000
25%       93.69750
50%       98.13500
75%      104.64000
max      107.44000
dtype: float64
```

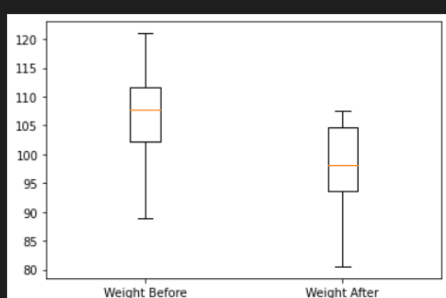
Step 1: Assumptions (Conditions):

- A quantitative variable for two independent groups.
  - Quantitaive variable is the weights of Individuals
  - Grouping variable is the Weight before and after diet.
- Size of sample > 30 or < 30 for both groups.
  - n1 and n2 = 10
- Is the population/sample approximately distributed.?
  - Using sample to estimate the population distribution, so as to see if it is
    - free of Outliers
    - Symmetric
    - Unimodal
  - Box Plot : None have outliers.

```python
# Distribution overiew of the given sample
my_dict = {'Weight Before': xi, 'Weight After':xj}

fig, ax = plt.subplots()
ax.boxplot(my_dict.values())
ax.set_xticklabels(my_dict.keys())
```

[8] ✓ 0.5s

```
[Text(1, 0, 'Weight Before'), Text(2, 0, 'Weight After')]
```



Step 2: Calculating the Interval

1. We could begin by computing the sample sizes (n1 and n2), means, and standard deviations (s1 and s2) in each sample.
2. The parameter of interest is the difference in population means, μ1 - μ2. The point estimate for the difference in population means is the difference in sample means:

$$\bar{x}_1 - \bar{x}_2$$

3. Calculating Pool Standard Deviation
   Since we are taking datasets from same area, it will be pooled and independent.

$$S_p = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$$

```
standarddev = math.sqrt(((10 - 1)*s1 * s1 + (10-1)*s2 * s2) / (10 + 10-2))
standarddev
```

[64] ✓ 0.8s

```
9.634033567399367
```

4. If n1 < 30 or n2 < 30, use the t-table:

$$(\bar{x}_1 - \bar{x}_2) \pm t \; S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Use the t-table with degrees of freedom = n_1+n_2-2

5. For 95% interval and df = 18

TABLE D

t distribution critical values

	Upper-tail probability *p*											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	0.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	0.765	0.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	0.741	0.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	0.689	0.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	0.687	0.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	0.686	0.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	0.686	0.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	0.685	0.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768

Appropriate Statistic to use i.e T-Statistic

```python
# To find the T critical value
t = stats.t.ppf(q=1-.05/2,df=18)
t
```

[68] ✓ 0.8s

```
2.10092204024096
```

So, the 95% confidence interval for the difference is (18.127 ,0.038)

```python

n1 = len(xi)
n2 = len(xj)

print ((x1.mean() - x2.mean()) + 2.10*(9.63)*math.sqrt((1/n1) + (1/n2)))
print ((x1.mean() - x2.mean()) - t*(9.63)*math.sqrt((1/n1) + (1/n2)))

```

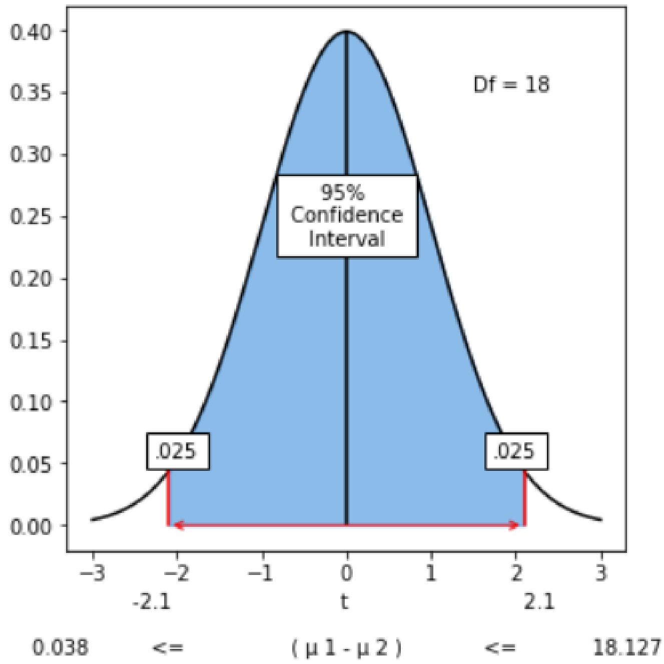
[75] ✓ 0.7s

```
18.12700054179565
0.03502853799539629
9.04400054179565
```

6. Plotting Appropriately

```python
x_min = -3
x_max = 3
plt.figure(figsize=(5,5))
mean = 0
std = 1

x = np.linspace(x_min, x_max, 1000)

y = scipy.stats.norm.pdf(x,mean,std)

plt.plot(x,y, color='black')

pt1 = -2.1
plt.plot([pt1 ,pt1 ],[0.0,scipy.stats.norm.pdf(pt1 ,mean, std)], color='red')

pt2 = 2.1
plt.plot([pt2 ,pt2 ],[0.0,scipy.stats.norm.pdf(pt2 ,mean, std)], color='red')
plt.xlabel("-2.1                                     t                                       2.1 \n \n0.038          <=                    ( \u03BC 1 - \u03BC 2 )
plt.vlines(x = 0, ymin = 0, ymax = 0.40,
           colors = 'black',
           label = 'vline_multiple - full height')
plt.text(0, 0.25, '95% \n Confidence \n Interval ', ha='center', va='center',rotation='horizontal', bbox={'facecolor':'white'})
plt.text(-2, 0.06, '.025', ha='center', va='center',rotation='horizontal', bbox={'facecolor':'white'})
plt.text(2, 0.06, '.025', ha='center', va='center',rotation='horizontal', bbox={'facecolor':'white'})

ptx = np.linspace(pt1, pt2, 1000)
pty = scipy.stats.norm.pdf(ptx,mean,std)
plt.annotate('', xy=(-2.1, 0.0), xytext=(2.1, 0.0),
             arrowprops=dict(arrowstyle='<->', color='red'))


plt.fill_between(ptx, pty, color='#187ad4', alpha=0.5)
plt.text(1.5,0.35, "Df = 18")

plt.show()

```

[70] ✓ 0.9s Python



Interval:
0.038 < $\mu 1 - \mu 2$ *< 18.127*

Step 3: Inferences And Interpretation

The researcher is 95% confident that the difference in population average of **weights before and weights after is between 0.038 and 18.127.**

The point estimate for the difference in **population means is 9.08** with the **error of 9.044.**

Hence we are 95% confident that the population mean for weights before is more than the population mean test score for weights after by between 0.038 and 18.127. Therefore, we can say that the plan is indeed *EFFECTIVE.*