Mattia Danese, Jianan Xu
CS 178 Visual Analytics
Lab 3

A. The null hypothesis is: cacti given fertilizer twice a week will not grow faster or slower (i.e. there will be no effect) than cacti given fertilizer once a week.
B. In order to answer RQ1, we will need to calculate and differentiate the growth of the "fertilizer twice a week" group and the "fertilizer once a week" group. The response variable is the growth rate of cacti and the groups are the "fertilizer twice a week" group and the "fertilizer once a week" group.
C. The amount of samples needed to answer RQ1 is $63.77 \approx 64$ samples

```python
# Part (c)
T = TTestIndPower()

alpha = 0.05
power = 0.80
effect_size = 0.50 # medium = 0.5
num_samples = T.solve_power(
    effect_size=effect_size,
    nobs1=None, alpha=alpha,
    power=power,
    ratio=1.0,
    alternative='two-sided')
print(num_samples)
```

```
~/Desktop/CS 178/Lab 3  base  system  19:21:51
python3 script.py
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/scipy/stats/_continuous_distns.py:6832: RuntimeWarning: invalid value encountered in _nct_sf
  return np.clip(_boost._nct_sf(x, df, nc), 0, 1)
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/scipy/stats/_continuous_distns.py:6826: RuntimeWarning: invalid value encountered in _nct_cdf
  return np.clip(_boost._nct_cdf(x, df, nc), 0, 1)
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/scipy/stats/_continuous_distns.py:6832: RuntimeWarning: overflow encountered in _nct_sf
  return np.clip(_boost._nct_sf(x, df, nc), 0, 1)
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/scipy/stats/_continuous_distns.py:6826: RuntimeWarning: overflow encountered in _nct_cdf
  return np.clip(_boost._nct_cdf(x, df, nc), 0, 1)
63.76561058785405
```

D. Below is how `exp_data.csv` was loaded

```python
# Part (d)
df = pd.read_csv("exp_data.csv")
```

E.

a.

```python
# Part (e)
print(df.head())
```

```
~/Desktop/CS 178/Lab 3
python3 script.py
   fert_grp  height_start  hours_sun  flowered  num_leaves_start  height_final  num_leaves_final
0         1          5.44          2         0                 6          7.91                 8
1         1          4.08          0         0                 6          7.11                 5
2         1          8.24          6         0                 6         10.99                 5
3         1          8.63          1         0                 6         11.31                 7
4         1          3.98          0         1                 7          6.81                 1
```

b.

```
print(df.describe())
```

```
~/Desktop/CS 178/Lab 3                                                                    base
python3 script.py
          fert_grp  height_start   hours_sun    flowered  num_leaves_start  height_final  num_leaves_final
count  130.000000    130.000000  130.000000  130.000000        130.000000    130.000000        130.000000
mean     1.500000      6.539769    4.330769    0.407692          6.076923      9.346538          4.153846
std      0.501934      2.227443    2.556004    0.493306          1.361801      2.253910          2.757545
min      1.000000     -1.840000    0.000000    0.000000          4.000000      0.480000          0.000000
25%      1.000000      5.002500    2.000000    0.000000          5.000000      7.762500          1.000000
50%      1.500000      6.580000    4.500000    0.000000          6.000000      9.410000          4.000000
75%      2.000000      8.207500    7.000000    1.000000          7.000000     10.877500          7.000000
max      2.000000     11.640000    8.000000    1.000000          8.000000     14.580000          8.000000
```

c. All the variables are typed as we would expect, except for some of the `height_start` variables as these are negative values (shown by negative minimum value for `height_start` in `describe()`) which are invalid because a cactus cannot have negative height.

d. Before the statistics included a negative minimum value for the `height_start` column, and after our fixes the minimum value for this column is 0.

```
print(df.describe())
df2 = df.copy()
df2["height_start"] = np.where(df["height_start"] < 0, 0, df["height_start"])
print(df2.describe())
```

```
~/Desktop/CS 178/Lab 3                                                                    base
python3 script.py
          fert_grp  height_start   hours_sun    flowered  num_leaves_start  height_final  num_leaves_final
count  130.000000    130.000000  130.000000  130.000000        130.000000    130.000000        130.000000
mean     1.500000      6.539769    4.330769    0.407692          6.076923      9.346538          4.153846
std      0.501934      2.227443    2.556004    0.493306          1.361801      2.253910          2.757545
min      1.000000     -1.840000    0.000000    0.000000          4.000000      0.480000          0.000000
25%      1.000000      5.002500    2.000000    0.000000          5.000000      7.762500          1.000000
50%      1.500000      6.580000    4.500000    0.000000          6.000000      9.410000          4.000000
75%      2.000000      8.207500    7.000000    1.000000          7.000000     10.877500          7.000000
max      2.000000     11.640000    8.000000    1.000000          8.000000     14.580000          8.000000
          fert_grp  height_start   hours_sun    flowered  num_leaves_start  height_final  num_leaves_final
count  130.000000    130.000000  130.000000  130.000000        130.000000    130.000000        130.000000
mean     1.500000      6.553923    4.330769    0.407692          6.076923      9.346538          4.153846
std      0.501934      2.179104    2.556004    0.493306          1.361801      2.253910          2.757545
min      1.000000      0.000000    0.000000    0.000000          4.000000      0.480000          0.000000
25%      1.000000      5.002500    2.000000    0.000000          5.000000      7.762500          1.000000
50%      1.500000      6.580000    4.500000    0.000000          6.000000      9.410000          4.000000
75%      2.000000      8.207500    7.000000    1.000000          7.000000     10.877500          7.000000
max      2.000000     11.640000    8.000000    1.000000          8.000000     14.580000          8.000000
```

F. The variables we need for our analysis are the change in height for every cactus (i.e. growth rate), the hours of sun per cactus, the flowered boolean per cactus, the start height, the final height, the number of leaves at the start and end.

```python
# Part (f)
change_in_heights = []
for i in range(len(df2["height_start"])):
    if df2["height_start"][i] == 0.0:
        temp = (df2["height_final"][i] - df2["height_start"][i])
    else:
        temp = (df2["height_final"][i] - df2["height_start"][i]) / df2["height_start"][i]
    change_in_heights.append(temp)
df2["change_in_height"] = change_in_heights
```

```
~/Desktop/CS 178/Lab 3                                                    base  system  19:14
python3 script.py
```

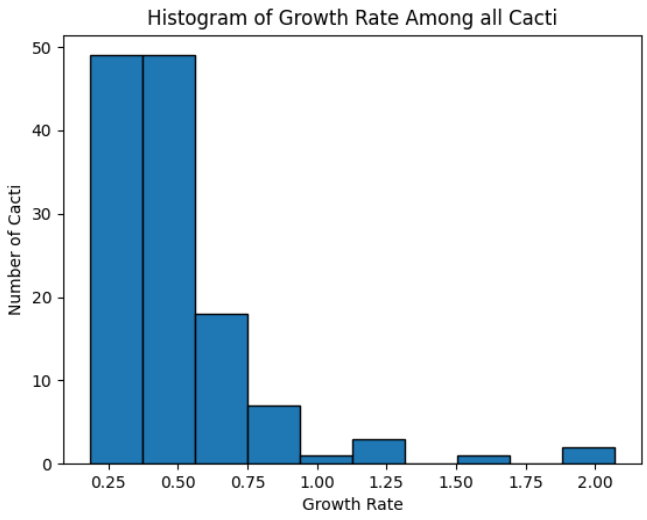
	fert_grp	height_start	hours_sun	flowered	num_leaves_start	height_final	num_leaves_final	change_in_height
count	130.000000	130.000000	130.000000	130.000000	130.000000	130.000000	130.000000	130.000000
mean	1.500000	6.553923	4.330769	0.407692	6.076923	9.346538	4.153846	0.497834
std	0.501934	2.179104	2.556004	0.493306	1.361801	2.253910	2.757545	0.295374
min	1.000000	0.000000	0.000000	0.000000	4.000000	0.480000	0.000000	0.182609
25%	1.000000	5.002500	2.000000	0.000000	5.000000	7.762500	1.000000	0.333775
50%	1.500000	6.580000	4.500000	0.000000	6.000000	9.410000	4.000000	0.436829
75%	2.000000	8.207500	7.000000	1.000000	7.000000	10.877500	7.000000	0.554536
max	2.000000	11.640000	8.000000	1.000000	8.000000	14.580000	8.000000	2.072222

G. It is clear that the data is not normally distributed as the histogram below does not have a bell-like curve.

```python
# Part (g)
fig,ax = plt.subplots(1,1)
ax.hist(df2["change_in_height"], edgecolor = "black")

ax.set_title("Histogram of Growth Rate Among all Cacti")
ax.set_xlabel('Growth Rate')
ax.set_ylabel('Number of Cacti')

plt.show()
```



H. The appropriate test to determine if the null hypothesis is rejected is the Mann-Whitney test because normality cannot be assumed and the data in the "fertilizer once a week" group was observed from different cacti than the data in the "fertilizer twice a week".
   a. Below are the assumptions of the Mann-Whitney test.[1]
      i. The first assumption of the Mann-Whitney test is that there is one dependent variable and it is measured at the continuous or ordinal level. This is met because the dependent variable, or response variable, is the growth rate of each cactus and this is a continuous value.
      ii. The second assumption of the Mann-Whitney test is that there is one independent variable that consists of two categorical, independent groups. This is met because the independent variable is the frequency of fertilization and this is a categorical, more specifically dichotomous, value (i.e. either once a week or twice a week).
      iii. The third assumption of the Mann-Whitney test is that there is independence of observation where there is no relationship between the observations in each group of the independent variable or between the groups themselves. This is met because there is no overlap whatsoever between the cacti in the "fertilizer once a week" group and the cacti in the "fertilizer twice a week".
      iv. The fourth assumption of the Mann-Whitney test is that it can be determined whether the distribution of scores, or response variables, for both groups of the independent variable have the same or different shape. This is met by the figures below. It is clear that the "fertilizer once a week" group and the "fertilizer twice a week" group have the same shape.

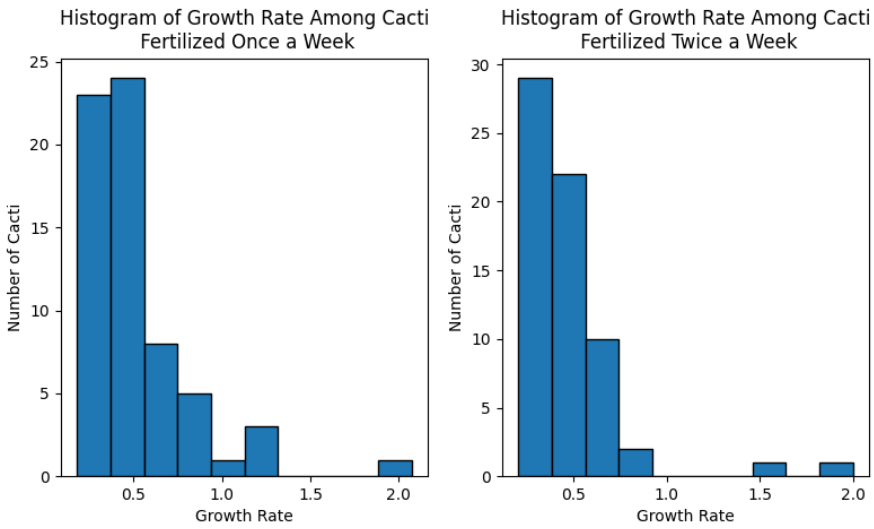
```
# Part (h)
fig,ax = plt.subplots(1,2)

ax[0].hist(df2[df2["fert_grp"] == 1]['change_in_height'], edgecolor = "black")
ax[0].set_title("Histogram of Growth Rate Among Cacti\n Fertilized Once a Week")
ax[0].set_xlabel('Growth Rate')
ax[0].set_ylabel('Number of Cacti')

ax[1].hist(df2[df2["fert_grp"] == 2]['change_in_height'], edgecolor = "black")
ax[1].set_title("Histogram of Growth Rate Among Cacti\n Fertilized Twice a Week")
ax[1].set_xlabel('Growth Rate')
ax[1].set_ylabel('Number of Cacti')

plt.show()
```

[1] https://statistics.laerd.com/statistical-guides/mann-whitney-u-test-assumptions.php



b. The results from running the Mann-Whitney test are: $U = 2492.0$ and $p = 0.07760740509$

```
x = df2[df2["fert_grp"] == 1]['change_in_height']
y = df2[df2["fert_grp"] == 2]['change_in_height']
results = mannwhitneyu(
    x,
    y,
    use_continuity=True,
    alternative='two-sided',
    axis=0,
    method='auto',
    nan_policy='propagate',
    keepdims=False)
```

```
~/Desktop/CS 178/Lab 3
python3 script.py
MannwhitneyuResult(statistic=2492.0, pvalue=0.07760740509432003)
```

I. In order to manipulate the data so that we can, in fact, reject the null hypothesis, we removed all the outliers in the "fertilizer twice a week" group while keeping the outliers in the "fertilizer once a week" group. P-hacking in this manner would not raise suspicions because it can be argued that removing the outliers was part of our data-processing and data-cleaning methodology. In other words, it is quite common to clean data from possible noise before analyzing it and running statistical tests. Additionally, the "fertilizer twice a week" group has more outliers than the "fertilizer once a week" group. Thus, it can be argued that the data of the "fertilizer twice a week" group was subject to more noise than the "fertilizer once a week" group, reason as to why the outliers were

only taken out of the “fertilizer twice a week” group and not the “fertilizer once a week” group. The results of the Mann-Whitney test after removing the outliers for the “fertilizer twice a week” group are below. We clearly get a p-value (0.0351) which is not only much smaller than our original p-value, but it is also under the 0.05 threshold.

```python
# part (i)
df3 = df2.copy()

# remove outliers in group 2
x = df3[df3["fert_grp"] == 1]
y = df3[df3["fert_grp"] == 2]

mean = y['change_in_height'].mean()
sd = y['change_in_height'].std()
y = y[(y['change_in_height'] <= mean+(3*sd))]

x = x['change_in_height']
y = y['change_in_height']

# get the new p-value
results = mannwhitneyu(
    x,
    y,
    use_continuity=True,
    alternative='two-sided',
    axis=0,
    method='auto',
    nan_policy='propagate',
    keepdims=False)
print(results)
```

```
~/Desktop/CS 178/Lab 3
python3 script.py
MannwhitneyuResult(statistic=2490.0, pvalue=0.03514847774208032)
```

J. We performed a Mann-Whitney test in order to determine if the null hypothesis may be rejected or not. The Mann-Whitney test revealed that the growth rates among the “fertilizer twice a week” group ($Md = 0.399$, $n = 65$) were significantly greater than growth rates in the “fertilizer once a week” group ($Md = 0.457$, $n = 65$), $U = 2490$, $p = 0.0351$, *two-sided*. Seeing as $p < 0.05$, it can be concluded that the null hypothesis is rejected and that cacti given fertilizer twice a week will grow faster or slower than cacti given fertilizer once a week.