In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
```

Question 1

alternative; *A/B test*; *blocking*; *causal*; *controllable*; *design*; *experimental*; *explanatory*; *factor*; *factorial*; *F-test*; *levels*; *metric of interest*; *nuisance*; *null*; *observational*; *power*; *randomization*; *replication*; *response*; *sample*; *sample size*; *significance level*; *statistical*; *t-test*; *Type I*; *Type II*; *uncontrollable*; *unit*; *valid*; *Z-test*;

(a) [2 points] The Facebook experiment can be colloquially referred to as an A/B test. More formally, we think of this as an experiment with one factor that has two levels.

(b) [1 point] A power analysis was used to determine that each condition required 500 users. Such an analysis is used to control Type II error.

(c) [1 points] The fact that the numbers 513 and 487 are greater than 1 reflect the experimental design principle called replication.

(d) [1 point] Each of the Facebook users in this experiment is considered an experimental unit.

(e) [1 point] The manner in which the users were selected for inclusion in the experiment is an example of the experimental design principle called randomization.

(f) [1 point] The session duration measured for each user may be referred to as the response variable.

(g) [1 point] The *average* session duration is the metric of interest.

(h) [1 point] The hypothesis test that is most appropriate for addressing the primary question in this experiment is a t-test.

(i) [2 points] Suppose the test in (i) is carried out and a p-value is calculated. In order to draw a formal conclusion this p-value must be compared to the significance level whose value is chosen to control Type I error.

(j) [2 points] The benefit of such an experiment, relative to an observational study, is that it more easily facilitates causal inference.

(k) [2 points] 'Device-type' and 'operating system' are examples of nuisance factors

here, and the manner in which they are dealt with is an example of blocking.

Question 2

(a) In each of the following questions, calculate the appropriate test statistic given the null hypothesis and the relevant data summaries. You do not need to calculate the p-value here. You may use Python for this question, but make sure to show your work.

i. [1 point] H0 : μ1 = μ2 (assuming σ1 and σ2 are unknown but equal)

* n1 = 750,ˆ μ1 = y1 = 15,ˆ σ1 = s1 = 2
* n2 = 750,ˆ μ2 = y2 = 10,ˆ σ2 = s2 = 3

ii. [1 point] H0 : μ1 ≥ μ2 (assuming σ1 and σ2 are unknown and unequal)

* n1 = 500,ˆ μ1 = y1 = 100,ˆ σ1 = s1 = 10
* n2 = 500,ˆ μ2 = y2 = 110,ˆ σ2 = s2 = 11

iii. [1 point] H0 : σ1^2 = σ2^2

* n1 = 500,ˆ μ1 = y1 = 100,ˆ σ1 = s1 = 10
* n2 = 500,ˆ μ2 = y2 = 110,ˆ σ2 = s2 = 11

In [ ]:
```
# i. H0 : μ1 = μ2 (assuming σ1 and σ2 are unknown but equal)
n1 = 750
n2 = 750
var = ((n1-1)*(2**2) + (n2-1)*(3**2))/(n1+n2-2)
t = (15 - 10)/(var*pow((1/n1+1/n2), 0.5))
t
```

Out[ ]: 14.89608979310545

In [ ]:
```
# ii. [1 point] H0 : μ1 ≥ μ2 (assuming σ1 and σ2 are unknown and unequal)
n1 = 500
n2 = 500
var = (10**2/n1) + (11**2/n2)
t = (100 - 110)/pow(var, 0.5)
```

```
t
```

Out[ ]: -15.04142093990467

In [ ]:
```
# iii. H0 : σ1^2 = σ2^2
t = 10**2/11**2
t
```

Out[ ]: 0.8264462809917356

(b) Suppose we perform an experiment with two conditions containing $n_1 = 418$ and $n_2 = 405$ units, respectively. For each of the hypotheses and test statistics below, state the null distribution and calculate the appropriate p-value. Note that in the case of Welch's t-test you may use the approximate degrees of freedom $\min(n_1, n_2) - 1$. You may use Python for this question, but make sure to show your work.

i. [2 points] H0 : μ1 = μ2 vs. HA : μ1 != μ2 (assuming σ1 and σ2 are unknown and unequal)

* t = −0.34

ii. [2 points] H0 : μ1 ≤ μ2 vs. HA : μ1 > μ2 (assuming σ1 and σ2 are unknown but equal)

* t = 1.76

iii. [2 points] H0 : σ1^2 = σ2^2 H_A: H0 : σ1^2 != σ2^2

* t = 1.02

In [ ]:
```
# i. H0 : μ1 = μ2 vs. HA : μ1/ = μ2 (assuming σ1 and σ2 are unknown and u
t = -0.34
pv = stats.t.cdf(t, 404) + 1 - stats.t.cdf(-t, 404)
pv
```

Out[ ]: 0.7340332278172301

In [ ]:
```
# ii. H0 : μ1 ≤ μ2 vs. HA : μ1 > μ2 (assuming σ1 and σ2 are unknown but e
t = 1.76
pv = stats.t.sf(t, 418+405-2)  ## p(T>=t)
pv
```

Out[ ]: 0.03939008297075401

In [ ]:
```
# iii. H0 : σ1^2 = σ2^2 H_A: H0 : σ1^2 != σ2^2
t = 1.02
```

```
pv = 1 - stats.f.cdf(t, dfn=417, dfd=404) + stats.f.cdf(1/t, dfn=417, dfd
pv
```

Out[ ]: 0.8411211647544122

Question 3

Punchh is a loyalty and engagement platform used by over 275 brick-and-mortar retailers such as Dairy Queen, Quiznos, Pizza Hut, and Denny's. Punchh provides these retailers with a machine learning-based software that may be embedded in their apps to manage personalized promotions that encourage customer loyalty. IHOP, for example, can use Punchh's machine learning algorithms to determine how and when to target their app's users with loyalty promotions (like a "Kids Eat Free" promotion). Suppose that Punchh is experimenting with the algorithm underlying their campaign delivery recommendations. In particular, suppose that the Product and UX Research team has developed a new version of the algorithm and they want to determine whether users who receive promotions from the new algorithm are more engaged than users who receive promotions from the existing version. Engagement is measured by the revenue generated by users exposed to each version of the algorithm.

Suppose that the machine learning scientists on the *Product and UX Research* team run an experiment in partnership with IHOP where $n = 500$ users of the IHOP app receive promotions recommended by the *existing* version of Punchh's algorithm and $n = 500$ users of the IHOP app receive promotions recommended by the *new* version of Punchh's algorithm. Each user is sent promotions by their respective algorithms over the course of 2 weeks and the amount of money they spend on the app (in dollars) in the following month is recorded. Interest lies in determining whether the average dollars spent under the new algorithm is significantly more than with the existing algorithm.

(a) [2 points] What is the metric of interest and what is the corresponding response variable?

```
MOI: Average dollars spent. Response variable: The
measurement of money that each user spent.
```

(b) [2 points] What is the design factor and what are its levels?

```
Factor: algorithm. Levels: {existing algorithm, new
algorithm}
```

(c) [1 points] What constitutes an experimental unit in this experiment?

```
Users that are using IHOP app
```

(d) [2 points] State the null and alternative hypotheses for this experiment. Be sure to define any notation you use.

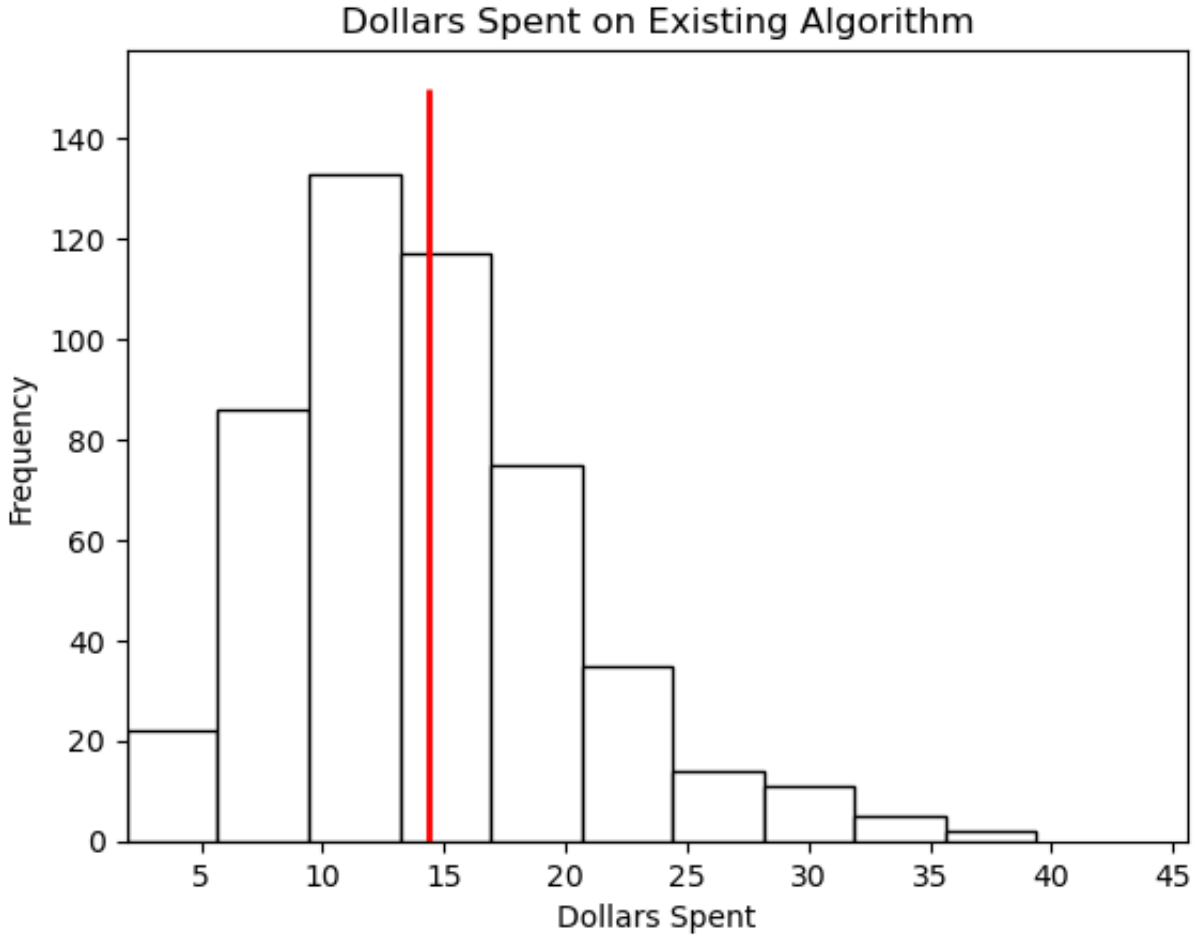
μ1 = average money spent in existing algorithm
μ2 = average money spent in new algorithm
H0: μ1 >= μ2 vs. HA = μ1 < μ2

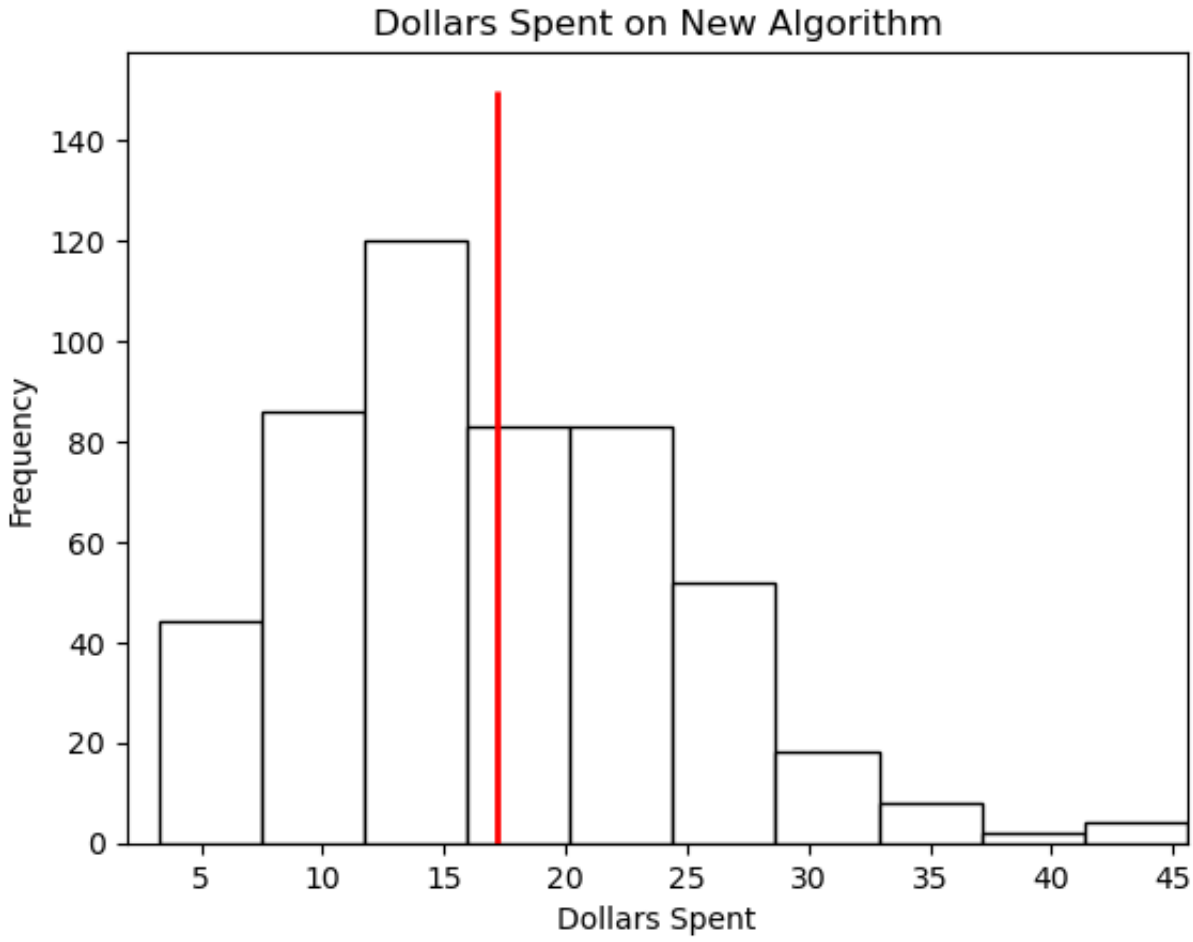
(e) [2 points] The file `punchh.csv` contains observations on the dollars spent for each of the 1000 IHOP app users. Plot two histograms of "dollars spent", one for each condition. On each histogram indicate the mean of the "dollars spent" distributions with red lines. Be sure to add a title and axis labels. Comment on which condition appears to maximize average dollars spent.

In [ ]:
```
df = pd.read_csv('./data/punchh.csv')
cond1 = df[df['Alg.Version']=='Existing']["Dollars.Spent"]
cond2 = df[df['Alg.Version']=='New']["Dollars.Spent"]
df1 = len(cond1)
df2 = len(cond2)
xmin = min(min(cond1), min(cond2))
xmax = max(max(cond1), max(cond2))
plt.figure()
plt.hist(cond1, color = "white", edgecolor = "black")
plt.xlim(xmin,xmax)
plt.xlabel("Dollars Spent")
plt.ylabel("Frequency")
plt.title("Dollars Spent on Existing Algorithm")
plt.vlines(x = np.mean(cond1), ymin = 0, ymax = 150, color = "red", linew

plt.figure()
plt.hist(cond2, color = "white", edgecolor = "black")
plt.xlim(xmin,xmax)
plt.xlabel("Dollars Spent")
plt.ylabel("Frequency")
plt.title("Dollars Spent on New Algorithm")
plt.vlines(x = np.mean(cond2), ymin = 0, ymax = 150, color = "red", linew
```

Out[ ]:
```
<matplotlib.collections.LineCollection at 0x144b8cbd0>
```


Dollars Spent on Existing Algorithm
Frequency
Dollars Spent
0
20
40
60
80
100
120
140
5
10
15
20
25
30
35
40
45


Dollars Spent on New Algorithm
Frequency
Dollars Spent
0
20
40
60
80
100
120
140
5
10
15
20
25
30
35
40
45

(f) [4 points] Using the observed data, test the hypothesis in (d) at a 5% significance level. Clearly state your conclusion in the context of the problem. Explain whether this conclusion is surprising, given what you see in the histograms from part (e). You may use Python for this question, but make sure to show your work.

In [ ]:
```
t = np.var(cond1, ddof = 1)/np.var(cond2, ddof = 1)
print("t =", t)
```

```
t = 0.6545396509678213
```

In [ ]:
```
pv = stats.f.cdf(t, dfn=df1, dfd=df2) + 1 - stats.f.cdf(1/t, dfn=df1, dfd
print("p-value =", pv)
```

```
p-value = 2.3749666006045445e-06
```

This p-value is smaller than 5% that we reject $H_0 : \sigma_1^2 = \sigma_2^2$, next step using Welch's t-test to test:

$$H_0 : \mu_1 \geq \mu_2 \text{ vs. } H_A : \mu_1 < \mu_2$$

In [ ]:
```
t, pv = stats.ttest_ind(cond1, cond2, equal_var = False, alternative = 'l
print("t =", t)
print("p-value =", pv)
```

```
t = -6.4463225406763405
p-value = 9.074779103707131e-11
```

This is an extremely small p-value, providing evidence against the null hypothesis above, suggesting that average dollar spent in the new algorithm could be greater than that of the existing algorithm. Doesn't surprise me from what I see in historgrams.

(g) [1 point] Comment on the suitability of the t-test for this problem and these data.

```
I think it is appropriate. t-test is commonly used to
compare the means of two independent groups. Therefore, a
two-sample t-test is a suitable statistical test for this
problem. The two data are from two individual sets,
existing and new algorithms. But we don't know how the
users are selected, is it randomly or not, that is one
concern.
```