ENGR 1330-2022 Exam 3 - Laboratory Portion

Medrano, Giovanni

R11521018

ENGR 1330 Exam 3M - Demonstrate Laboratory/Programming Skills

---

If you are unable to download the file, create an empty notebook and copy paste the problems into Markdown cells and Code cells (problem-by-problem)

Problem 0 (5 pts) : *Profile your computer*

Execute the code cell below exactly as written. If you get an error just continue to the remaining problems.

In [5]:
```
# Preamble script block to identify host, user, and kernel
import sys
! hostname
! whoami
print(sys.executable)
print(sys.version)
print(sys.version_info)
```

```
DESKTOP-6HAS1BN
desktop-6has1bn\medra
C:\Users\medra\anaconda3\python.exe
3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
sys.version_info(major=3, minor=8, micro=5, releaselevel='final', serial=0)
```

Exercise 1 (5 pts) Download the datafile

The file is available on Blackboard and you will need to dowenload it and place it in the same folder/directory as your Exam ipynb file. It contains values of impact strength of packaging materials in foot-pounds of branded boxes.

Download the file and read it into a dataframe.

Download the necessary datafile

In [40]:
```
# download script goes here
localfile = open('boxes.csv','wb') # open connection to a local file same name as remot
localfile.write(rget.content) # extract from the remote the contents,insert into the lo
localfile.close() # close connection to the local file
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-40-1b51bbd38c54> in <module>
      1 # download script goes here
      2 localfile = open('boxes.csv','wb') # open connection to a local file same name
 as remote
```

```
----> 3 localfile.write(rget.content) # extract from the remote the contents,insert int
o the local file same name
      4 localfile.close() # close connection to the local file

NameError: name 'rget' is not defined
```

Store the datafile into a dataframe

In [1]:
```
# read dataframe script goes here
import pandas as pd
dfd = pd.read_csv('boxes.csv')
dfd
```

Out[1]:

	Amazon Branded Boxes	Walmart Branded Boxes	US Postal Service Branded Boxes
0	0.786840	1.085667	1.017169
1	1.278068	1.019425	1.225774
2	0.964830	0.798582	1.139398
3	0.968899	0.647075	1.478555
4	0.938127	0.856392	1.158053
...	...	...	...
249995	1.682455	0.267142	1.005418
249996	0.402352	0.632802	1.234473
249997	1.288016	1.201354	1.288564
249998	0.864747	0.713163	1.438149
249999	1.421109	0.729032	1.283414

250000 rows × 3 columns

Describe the dataframe, how many columns are in the dataframe? What are the column names?

In [5]:
```
# your script/answers go here
dfd.describe()
```

Out[5]:

	Amazon Branded Boxes	Walmart Branded Boxes	US Postal Service Branded Boxes
count	250000.000000	250000.000000	250000.000000
mean	1.210037	0.780308	1.199937
std	0.300282	0.200181	0.119895
min	-0.131195	-0.153808	0.663480
25%	1.007211	0.646009	1.118755
50%	1.211051	0.780008	1.200183
75%	1.411952	0.914654	1.280694
max	2.535326	1.655256	1.730012

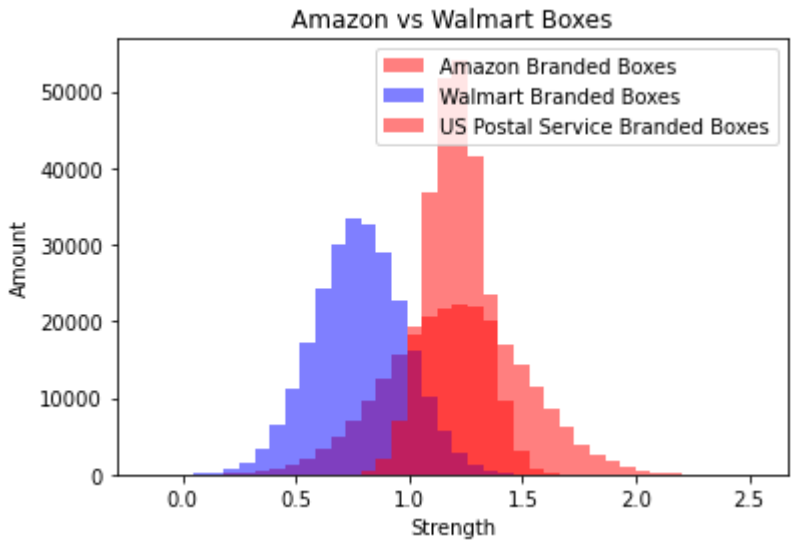
Exercise 2 (15 pts.) Produce a histogram of the Amazon series and the Walmart series on the same plot. Plot Amazon using red, and Walmart using blue.

> - Import suitable package to build histograms
> - Apply package with plotting call to prodice two histograms on same figure space
> - Label plot and axes with suitable annotation

Plot the histograms with proper formatting

In [21]:
```
# your script goes here
import matplotlib.pyplot as plt
from numpy.lib.histograms import histogram
#a = plt.subplots()
graph = dfd.plot.hist(density = False, title = "Amazon vs Walmart Boxes", color = ['red
graph.set_xlabel("Strength")
graph.set_ylabel("Amount")
```

Out[21]: Text(0, 0.5, 'Amount')



Comment on the histograms, do they overlap?

TYPE HERE: Your comments regarding the histograms here

Yes they overlap

Exercise 3 (5 pts.) Summary Statistics for the Amazon and Walmart Brands

> - Compute the mean strength and the standard deviation of the Amazon and Walmart brands
> - Identify which series has a greater mean value
> - Identify which series has the greater standard deviation

Compute the means and standard deviations

In [8]:
```
# your script goes here
import statistics
aMean = statistics.mean(dfd['Amazon Branded Boxes'])
wMean = statistics.mean(dfd['Walmart Branded Boxes'])

print(aMean, wMean)

print('============================')

aStand = statistics.stdev(dfd['Amazon Branded Boxes'])
wStand = statistics.stdev(dfd['Walmart Branded Boxes'])

print(aStand, wStand)

print('The Amazon branded boxes have a higher mean value')
print('The Amazon branded boxes have a higher standard deviation value')
```

```
1.210036656744606 0.7803082066047033
============================
0.3002821784033327 0.20018099285701327
The Amazon branded boxes have a higher mean value
The Amazon branded boxes have a higher standard deviation value
```

Identify which has the largest mean

TYPE HERE: Your comments regarding which has a greater mean

The Amazon branded boxes have a higher mean value

Identify which has the largest standard deviation

TYPE HERE: Your comments regarding which has a greater standard deviation

The Amazon branded boxes have a higher standard deviation value

Exercise 4 (5 pts.) Test the Amazon data for normality, interpret the results.

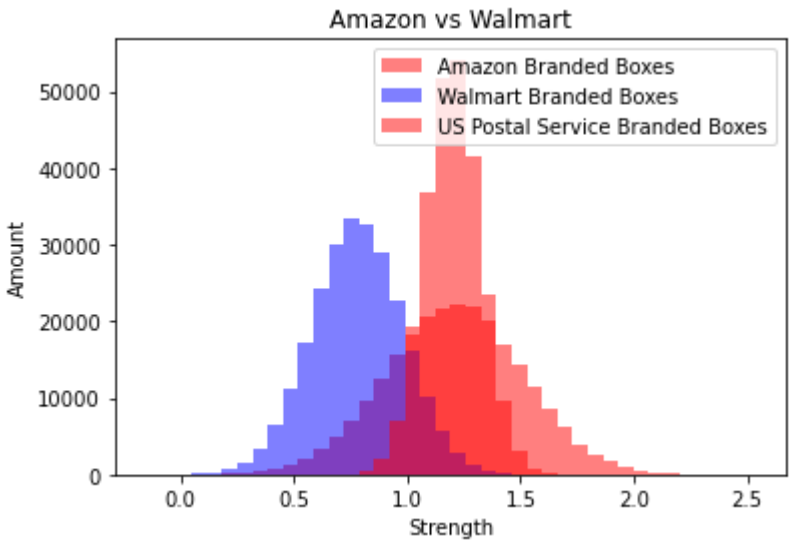
Build your test below

In [10]:
```
# your script here
from scipy.stats import shapiro

stat, k = shapiro(dfd['Amazon Branded Boxes'])
print('stat =', stat,'\nk is thus =', k)

if k > .05:
    print('Good chance its Gaussian')
else:
    print('Good chance its not Guaissian')
```

```
stat = 1.0000150203704834
k is thus = 1.0
Good chance its Gaussian
C:\Users\medra\anaconda3\lib\site-packages\scipy\stats\morestats.py:1681: UserWarning: p
-value may not be accurate for N > 5000.
  warnings.warn("p-value may not be accurate for N > 5000.")
```

In [12]:
```
# your script goes here
import matplotlib.pyplot as plt
from numpy.lib.histograms import histogram
#a = plt.subplots()
graph = dfd.plot.hist(density = False, title = "Amazon vs Walmart", color = ['red', 'bl
graph.set_xlabel("Strength")
graph.set_ylabel("Amount")
```

Out[12]:
```
Text(0, 0.5, 'Amount')
```



Interpret the results

The data is shown to be normal.

Exercise 5 (5 pts.) Test the Walmart data for normality, interpret the results.

Build your test below

In [14]:
```
# your script here
# your script here
from scipy.stats import shapiro

stat, k = shapiro(dfd['Walmart Branded Boxes'])
print('stat =', stat,'\nk is thus =', k)

if (k > .05):
    print('Good chance its Gaussian')
else:
    print('Good chance its not Guaissian')
```

```
stat = 1.0000152587890625
k is thus = 1.0
Good chance its Gaussian
```

Interpret the results

Again it is normal distribution.

Exercise 6 (10 pts.) Determine if there is evidence of a

difference in mean strength between the two brands.

Use an appropriate hypothesis test to support your assertion at a level of significance of $\alpha = 0.10$.

> - Choose a test and justify choice
> - Import suitable package to run the test
> - Apply the test and interpret the results
> - Report result with suitable annotation

Build your hypothesis test below

In [17]:
```
# your script here
from scipy.stats import mannwhitneyu as l

stat, k = l(dfd['Amazon Branded Boxes'],dfd['Walmart Branded Boxes'])
print('The statistic seems to be:',stat, '\nk-value at the rejection point is =', k)

if (k>0.05):
    print('The two seem to have the same distribution')
else:
    print('The two seem to have different distribution')
```

```
The statistic seems to be: 7305909682.0
k-value at the rejection point is = 0.0
The two seem to have different distribution
```

Interpret the results

According to the Mann-Whitneyu test it is apparant that the hypothesis is rejected and the two have different distributions

In [ ]:
```
```