Big Data Assignment

In [1]:

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

In [2]:

```
#Practice with Basic function
no = 10 + 5 * 3 / 4.0

print(no)
```

```
13.75
```

In [3]:

```
#Single % sign use to get division. dividend % divisor = remainder.
remainder = 15 % 9
print(remainder)
```

```
6
```

In [4]:

```
#using strings

word= ('Hello everyone' +' '+ 'My name is Bhagyashree')
print(word)
```

```
Hello everyone My name is Bhagyashree
```

In [5]:

```
#creating while loop
count = 1
while (count <= 15):
   print ('The count is:', count+0.25)
   count = count + 1

print ("While Loop Practice")
```

```
The count is: 1.25
The count is: 2.25
The count is: 3.25
The count is: 4.25
The count is: 5.25
The count is: 6.25
The count is: 7.25
The count is: 8.25
The count is: 9.25
The count is: 10.25
The count is: 11.25
The count is: 12.25
The count is: 13.25
The count is: 14.25
The count is: 15.25
While Loop Practice
```

In [6]:

```
#Example
for letter in 'Assignment':
   print ('Letter :', letter)

Books = ['Python','Java','C++']

#Example
for Book in Books:
   print ('Book name :', Book)

print ("For Loop Practice")
```

```
Letter : A
Letter : s
Letter : s
Letter : i
Letter : g
Letter : n
Letter : m
Letter : e
Letter : n
Letter : t
Book name : Python
Book name : Java
Book name : C++
For Loop Practice
```

In [7]:

```
# creating if condition

name = "Pallavi"
age = 25
if name == "Pallavi" and age == 25:
    print("Your name is Pallavi, and you are also 25 years old.")

if name == "Pallavi" or name == "Bhagyashree":
    print("Your name is either Pallavi or Bhagyashree.")
```

```
Your name is Pallavi, and you are also 25 years old.
Your name is either Pallavi or Bhagyashree.
```

In [9]:

```
# creating if and else condition

x = 10
if x == 10:
    print("x equals ten")
else:
    print("x does not equal to ten.")

y =  20

if y==10:
    print('y equals to tewnty')
else:
    print('y does not to equal to twenty')
```

```
x equals ten
y does not to equal to twenty
```

In [19]:

```
# Create 2 new lists height and weight
x = [87,  17, 82, 91, 15, 51]
y = [65, 92, 25, 98, 18, 45]

# Create 2 numpy arrays from height and weight
x1 = np.array(x)
y1 = np.array(y)
print(type(x))

c = x1/y1 **2
print(c)
```

```
<class 'list'>
[0.02059172 0.00200851 0.1312     0.00947522 0.0462963  0.02518519]
```

In [25]:

```
# reading file from system
#reading sample dataset to perform basic operations

import pandas as pd

df = pd.read_csv('Downloads/2019-plu-total-hab-data.csv')

df.head()
```

Out[25]:

	Geography	Timeframe	Current Year Week Ending	Type	ASP Current Year	Total Bulk and Bags Units	4046 Units	4225 Units	4770 Units	TotalBagged Units	SmlBagged Units
0	Total U.S.	Weekly	2019-01-07 00:00:00	Conventional	1.02	44749707.48	14377053.08	11890232.90	893721.10	17588700.40	12829493.40
1	Albany	Weekly	2019-01-07 00:00:00	Conventional	1.07	129222.29	3789.30	112635.18	158.00	12639.81	8877.95
2	Atlanta	Weekly	2019-01-07 00:00:00	Conventional	0.92	828971.15	388574.98	38902.85	3482.04	398011.28	299475.26
3	Baltimore/Washington	Weekly	2019-01-07 00:00:00	Conventional	1.31	925391.38	102652.85	530128.43	8212.94	284397.16	263150.78
4	Boise	Weekly	2019-01-07 00:00:00	Conventional	1.23	108261.98	43723.19	7085.86	14435.46	43017.47	23932.54

In [26]:

```
# This is how we read columns data
df.head(10)
```

Out[26]:

	Geography	Timeframe	Current Year Week Ending	Type	ASP Current Year	Total Bulk and Bags Units	4046 Units	4225 Units	4770 Units	TotalBagged Units	SmlBagged Units
0	Total U.S.	Weekly	2019-01-07 00:00:00	Conventional	1.02	44749707.48	14377053.08	11890232.90	893721.10	17588700.40	12829493.40
1	Albany	Weekly	2019-01-07 00:00:00	Conventional	1.07	129222.29	3789.30	112635.18	158.00	12639.81	8877.95
2	Atlanta	Weekly	2019-01-07 00:00:00	Conventional	0.92	828971.15	388574.98	38902.85	3482.04	398011.28	299475.26
3	Baltimore/Washington	Weekly	2019-01-07 00:00:00	Conventional	1.31	925391.38	102652.85	530128.43	8212.94	284397.16	263150.78
4	Boise	Weekly	2019-01-07 00:00:00	Conventional	1.23	108261.98	43723.19	7085.86	14435.46	43017.47	23932.54
5	Boston	Weekly	2019-01-07 00:00:00	Conventional	1.34	767731.38	11483.09	597937.87	2510.57	155799.85	109108.51
6	Buffalo/Rochester	Weekly	2019-01-07 00:00:00	Conventional	1.19	197137.26	9963.81	92129.51	281.96	94761.98	53069.02
7	California	Weekly	2019-01-07 00:00:00	Conventional	1.06	7195166.45	1989731.98	2089603.56	118964.72	2996866.19	2851060.95
8	Charlotte	Weekly	2019-01-07 00:00:00	Conventional	1.10	323640.60	122312.03	63971.60	9550.23	127806.74	108828.50
9	Chicago	Weekly	2019-01-07 00:00:00	Conventional	1.31	739020.76	186828.35	371930.88	46176.93	134084.60	111203.24

In [27]:

```
#creating tail of dataset
df.tail(5)
```

Out[27]:

	Geography	Timeframe	Current Year Week Ending	Type	ASP Current Year	Total Bulk and Bags Units	4046 Units	4225 Units	4770 Units	TotalBagged Units	SmlBagged Units	LrgBagged Units	X-LrgBagged Units
5179	West	Weekly	2019-11-10 00:00:00	Organic	1.91	235865.0	16366.0	19561.0	454.0	199484.0	135025.0	64370.0	89.0
5180	West Tex/New Mexico	Weekly	2019-12-01 00:00:00	Organic	1.52	17646.0	1162.0	457.0	1114.0	14914.0	14358.0	556.0	0.0
5181	West Tex/New Mexico	Weekly	2019-11-24 00:00:00	Organic	1.62	20587.0	1065.0	444.0	1635.0	17443.0	16791.0	652.0	0.0
5182	West Tex/New Mexico	Weekly	2019-11-17 00:00:00	Organic	1.68	17278.0	1088.0	340.0	1230.0	14620.0	14618.0	2.0	0.0
5183	West Tex/New Mexico	Weekly	2019-11-10 00:00:00	Organic	1.69	20175.0	871.0	383.0	1507.0	17414.0	17101.0	312.0	0.0

In [28]:

```
#creating sample
df.sample()
```

Out[28]:

	Geography	Timeframe	Current Year Week Ending	Type	ASP Current Year	Total Bulk and Bags Units	4046 Units	4225 Units	4770 Units	TotalBagged Units	SmlBagged Units	LrgBagged Units	X-LrgBagged Units
3306	Denver	Weekly	2019-08-04 00:00:00	Organic	2.21	26289.46	5631.94	1340.69	2.33	19314.5	19166.36	148.14	0.0

In [29]:

```
#creating information of dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5184 entries, 0 to 5183
Data columns (total 13 columns):
Geography                    5184 non-null object
Timeframe                    5184 non-null object
Current Year Week Ending     5184 non-null object
Type                         5184 non-null object
ASP Current Year             5184 non-null float64
Total Bulk and Bags Units    5184 non-null float64
4046 Units                   5184 non-null float64
4225 Units                   5184 non-null float64
4770 Units                   5184 non-null float64
TotalBagged Units            5184 non-null float64
SmlBagged Units              5184 non-null float64
LrgBagged Units              5184 non-null float64
X-LrgBagged Units            5184 non-null float64
dtypes: float64(9), object(4)
memory usage: 526.6+ KB
```

In [30]:

```
#creating decribing in numerical format
df.describe()
```

Out[30]:

	ASP Current Year	Total Bulk and Bags Units	4046 Units	4225 Units	4770 Units	TotalBagged Units	SmlBagged Units	LrgBagged Units	X-LrgBagged Units
count	5184.000000	5.184000e+03	5.184000e+03	5.184000e+03	5.184000e+03	5.184000e+03	5.184000e+03	5.184000e+03	5184.000000
mean	1.412741	1.044700e+06	2.884655e+05	2.659674e+05	2.205838e+04	4.682060e+05	2.952141e+05	1.613626e+05	11629.342990
std	0.371065	4.168012e+06	1.239021e+06	1.097338e+06	9.815348e+04	1.814242e+06	1.125809e+06	6.643936e+05	56767.333535
min	0.540000	2.534500e+02	0.000000e+00	0.000000e+00	0.000000e+00	4.555000e+01	4.555000e+01	0.000000e+00	0.000000
25%	1.147500	1.802272e+04	5.576625e+02	2.402883e+03	0.000000e+00	1.337082e+04	1.016043e+04	1.378890e+03	0.000000
50%	1.370000	1.556058e+05	1.285466e+04	1.773226e+04	2.669300e+02	7.898010e+04	5.354757e+04	1.520447e+04	3.330000
75%	1.630000	5.441795e+05	1.173928e+05	1.237169e+05	5.422078e+03	2.383269e+05	1.463408e+05	7.039264e+04	1865.455000
max	2.780000	6.245151e+07	1.949892e+07	1.788781e+07	1.591800e+06	2.347299e+07	1.526452e+07	8.378356e+06	844929.830000

In [42]:

```
# checking count from dataset

#missing count values range


count= df.isnull().sum()

count

total_cells = np.product(df)

total_missing = count.sum()

print('Total Missing Values:',total_missing)
```

```
Total Missing Values: 0
```

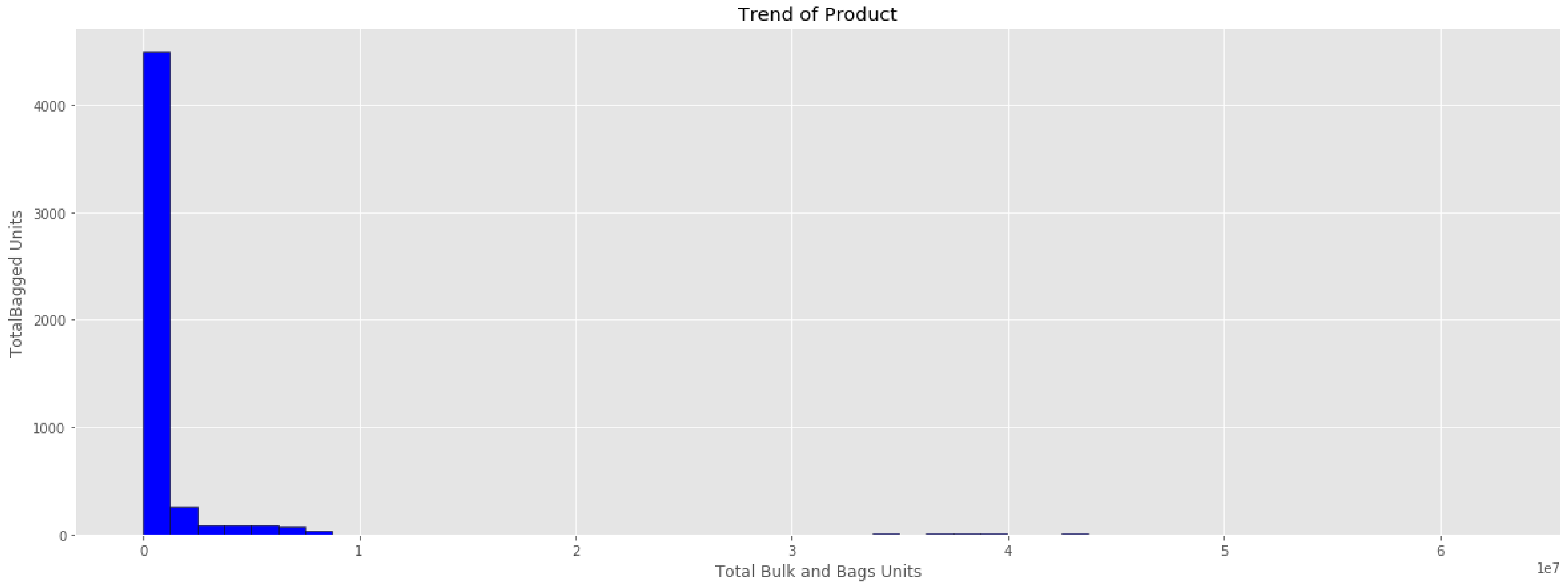
In [47]:
```
#importing the following library

import matplotlib.pyplot as plt
import seaborn as sns

#creating histogram

plt.figure()
plt.figure(figsize=(20,7))
plt.hist(df['Total Bulk and Bags Units'],color = 'blue', edgecolor = 'black', bins= 50)
plt.style.use('ggplot')
plt.title('Trend of Product')
plt.xlabel('Total Bulk and Bags Units')
plt.ylabel('TotalBagged Units')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



In [ ]: