Instructions for Submitting the Project:

You have to submit 2 files:
Answer Report: In this you need to submit all the answers to all the questions in a sequential manner. It should include the detailed explanation of approach used, insights, inferences, all outputs of codes like graphs, tables etc. Your report should not be filled with codes. You will be evaluated based on the business report. Jupyter Notebook file: This is a must and will be used for reference while evaluating Any assignment found copied/ plagiarized with another person will not be graded and marked as zero. Please ensure timely submission as a post-deadline assignment will not be accepted.

Problem 1 Statement:

A wholesale distributor operating in different regions of Portugal has information on annual spending of several items in their stores across different regions and channels. The data consists of 440 large retailers' annual spending on 6 different varieties of products in 3 different regions (Lisbon, Oporto, Other) and across different sales channel (Hotel, Retail).

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set (color_codes=True)
import os
```

executed in 1m 51.3s, finished 22:36:44 2020-12-12

In [2]:

```
p1= pd.read_csv(r'E:\Great Learning\Projects\Wholesale+Customers+Data.csv')
```

executed in 572ms, finished 14:18:48 2020-12-10

In [3]:

```
p1.head()
```

executed in 934ms, finished 14:19:31 2020-12-10

Out[3]:

	Buyer/Spender	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicates
0	1	Retail	Other	12669	9656	7561	214	2674	1
1	2	Retail	Other	7057	9810	9568	1762	3293	1
2	3	Retail	Other	6353	8808	7684	2405	3516	7
3	4	Hotel	Other	13265	1196	4221	6404	507	1
4	5	Retail	Other	22615	5410	7198	3915	1777	5

In [7]:

```
p1.shape
```

executed in 14ms, finished 16:18:18 2020-12-07

Out[7]:

```
(440, 9)
```

In [8]:

```
p1.size
```

executed in 17ms, finished 16:18:27 2020-12-07

Out[8]:

```
3960
```

In [10]:

```
p1.info()
```

executed in 31ms, finished 16:18:40 2020-12-07

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 9 columns):
Buyer/Spender       440 non-null int64
Channel             440 non-null object
Region              440 non-null object
Fresh               440 non-null int64
Milk                440 non-null int64
Grocery             440 non-null int64
Frozen              440 non-null int64
Detergents_Paper    440 non-null int64
Delicatessen        440 non-null int64
dtypes: int64(7), object(2)
memory usage: 31.1+ KB
```

In [11]:

```
p1.isna().values.any()
```

executed in 23ms, finished 16:20:04 2020-12-07

Out[11]:

```
False
```

In [13]:

```
p1.isnull().sum()
```

executed in 14ms, finished 16:20:45 2020-12-07

Out[13]:

```
Buyer/Spender       0
Channel             0
Region              0
Fresh               0
Milk                0
Grocery             0
Frozen              0
Detergents_Paper    0
Delicatessen        0
dtype: int64
```

1.1 Use methods of descriptive statistics to summarize data. Which Region and which Channel seems to spend more? Which Region and which Channel seems to spend less?

In [15]:

```
p1.describe()
```

executed in 304ms, finished 16:38:35 2020-12-07

Out[15]:

	Buyer/Spender	Fresh	Milk	Grocery	Frozen	Detergents_Pap
count	440.000000	440.000000	440.000000	440.000000	440.000000	440.0000
mean	220.500000	12000.297727	5796.265909	7951.277273	3071.931818	2881.4931
std	127.161315	12647.328865	7380.377175	9503.162829	4854.673333	4767.8544
min	1.000000	3.000000	55.000000	3.000000	25.000000	3.0000
25%	110.750000	3127.750000	1533.000000	2153.000000	742.250000	256.7500
50%	220.500000	8504.000000	3627.000000	4755.500000	1526.000000	816.5000
75%	330.250000	16933.750000	7190.250000	10655.750000	3554.250000	3922.0000
max	440.000000	112151.000000	73498.000000	92780.000000	60869.000000	40827.0000

In [17]:

```
p1['Channel'].value_counts()
```

executed in 54ms, finished 16:39:32 2020-12-07

Out[17]:

```
Hotel     298
Retail    142
Name: Channel, dtype: int64
```

In [18]:

```
p1['Region'].value_counts()
```

executed in 20ms, finished 16:39:59 2020-12-07

Out[18]:

```
Other     316
Lisbon     77
Oporto     47
Name: Region, dtype: int64
```

In [21]:

```
p1[['Channel']].groupby(p1['Buyer/Spender']).max()
```

executed in 538ms, finished 16:51:48 2020-12-07

Out[21]:

	Channel
Buyer/Spender	
1	Retail
2	Retail
3	Retail
4	Hotel
5	Retail
...	...
436	Hotel
437	Hotel
438	Retail
439	Hotel
440	Hotel

440 rows × 1 columns

In [23]:

```
p1[['Buyer/Spender']].groupby(p1['Channel']).max()
```

executed in 109ms, finished 16:52:25 2020-12-07

Out[23]:

	Buyer/Spender
Channel	
Hotel	440
Retail	438

In [24]:

```
p1['Buyer/Spender'].groupby(p1['Region']).max()
```

executed in 40ms, finished 16:53:15 2020-12-07

Out[24]:

```
Region
Lisbon     273
Oporto     340
Other      440
Name: Buyer/Spender, dtype: int64
```

In [27]:

```
p1['Buyer/Spender'].groupby(p1['Channel']).min()
```

executed in 22ms, finished 17:22:39 2020-12-07

Out[27]:

```
Channel
Hotel     4
Retail    1
Name: Buyer/Spender, dtype: int64
```

1.1.2 Which Region and which Channel seems to spend more?

Region which is spending more is :

Other spending of 440.

Channel Seems to spend more is :

Hotel spending of 440

1.1.3 Which Region and which Channel seems to spend less?

Region which is spending less is :

Lisbon spending of 273.

Channel Seems to spend less is

Retail spending of 438.

Even though there is only difference of “2” between (Channel) Hotel and Retail so we can’t give the final decision who is spending more as difference of 2 will not make any difference.

Hence both channel’s Hotel and Retail are contributing the same to the Business.

1.2 There are 6 different varieties of items are considered. Do all varieties show similar behaviour across Region and Channel? Provide justification for your answer

In [28]:

```
### Lets pull the Data again.

p1.head()
```

executed in 43ms, finished 17:23:24 2020-12-07

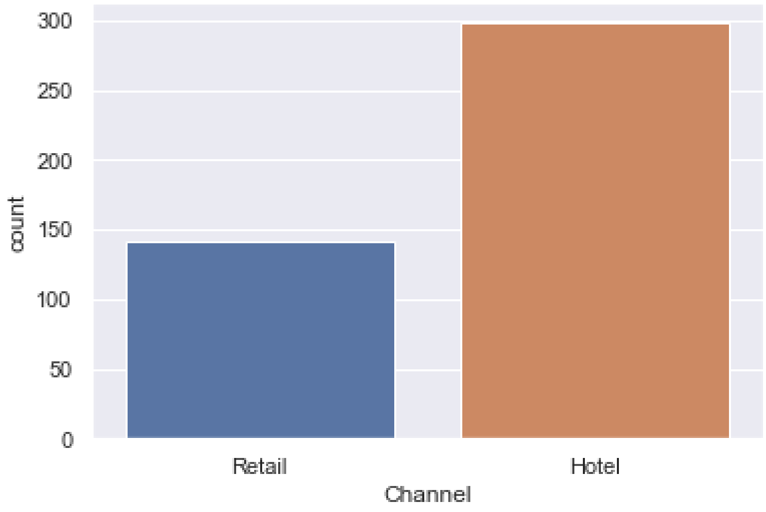
Out[28]:

	Buyer/Spender	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicates
0	1	Retail	Other	12669	9656	7561	214	2674	1
1	2	Retail	Other	7057	9810	9568	1762	3293	1
2	3	Retail	Other	6353	8808	7684	2405	3516	7
3	4	Hotel	Other	13265	1196	4221	6404	507	1
4	5	Retail	Other	22615	5410	7198	3915	1777	5

In [36]:

```
sns.countplot(p1['Channel']);
```

executed in 1.27s, finished 17:40:21 2020-12-07



In [44]:

```
p1['Fresh'].groupby(p1['Channel']).mean().round()
```

executed in 24ms, finished 17:47:48 2020-12-07

Out[44]:

```
Channel
Hotel     13476.0
Retail     8904.0
Name: Fresh, dtype: float64
```

In [45]:

```
p1[['Fresh']].groupby(p1['Region']).mean().round()
```

executed in 26ms, finished 17:49:27 2020-12-07

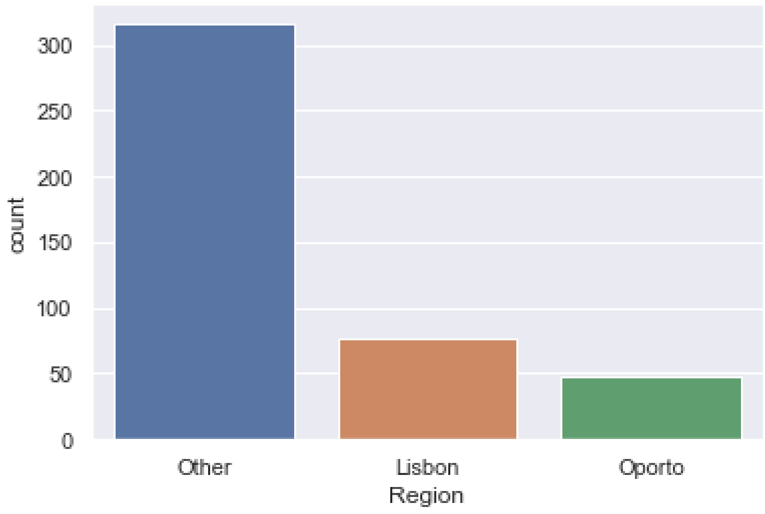
Out[45]:

	Fresh
Region	
Lisbon	11102.0
Oporto	9888.0
Other	12533.0

In [46]:

```
sns.countplot(p1['Region']);
```

executed in 225ms, finished 17:50:12 2020-12-07



In [50]:

```
p1.describe()
```

executed in 73ms, finished 18:02:38 2020-12-07

Out[50]:

	Buyer/Spender	Fresh	Milk	Grocery	Frozen	Detergents_Pap
count	440.000000	440.000000	440.000000	440.000000	440.000000	440.0000
mean	220.500000	12000.297727	5796.265909	7951.277273	3071.931818	2881.4931
std	127.161315	12647.328865	7380.377175	9503.162829	4854.673333	4767.8544
min	1.000000	3.000000	55.000000	3.000000	25.000000	3.0000
25%	110.750000	3127.750000	1533.000000	2153.000000	742.250000	256.7500
50%	220.500000	8504.000000	3627.000000	4755.500000	1526.000000	816.5000
75%	330.250000	16933.750000	7190.250000	10655.750000	3554.250000	3922.0000
max	440.000000	112151.000000	73498.000000	92780.000000	60869.000000	40827.0000

1.3 On the basis of a descriptive measure of variability, which item shows the most inconsistent behaviour? Which items show the least inconsistent behaviour?

In [212]:

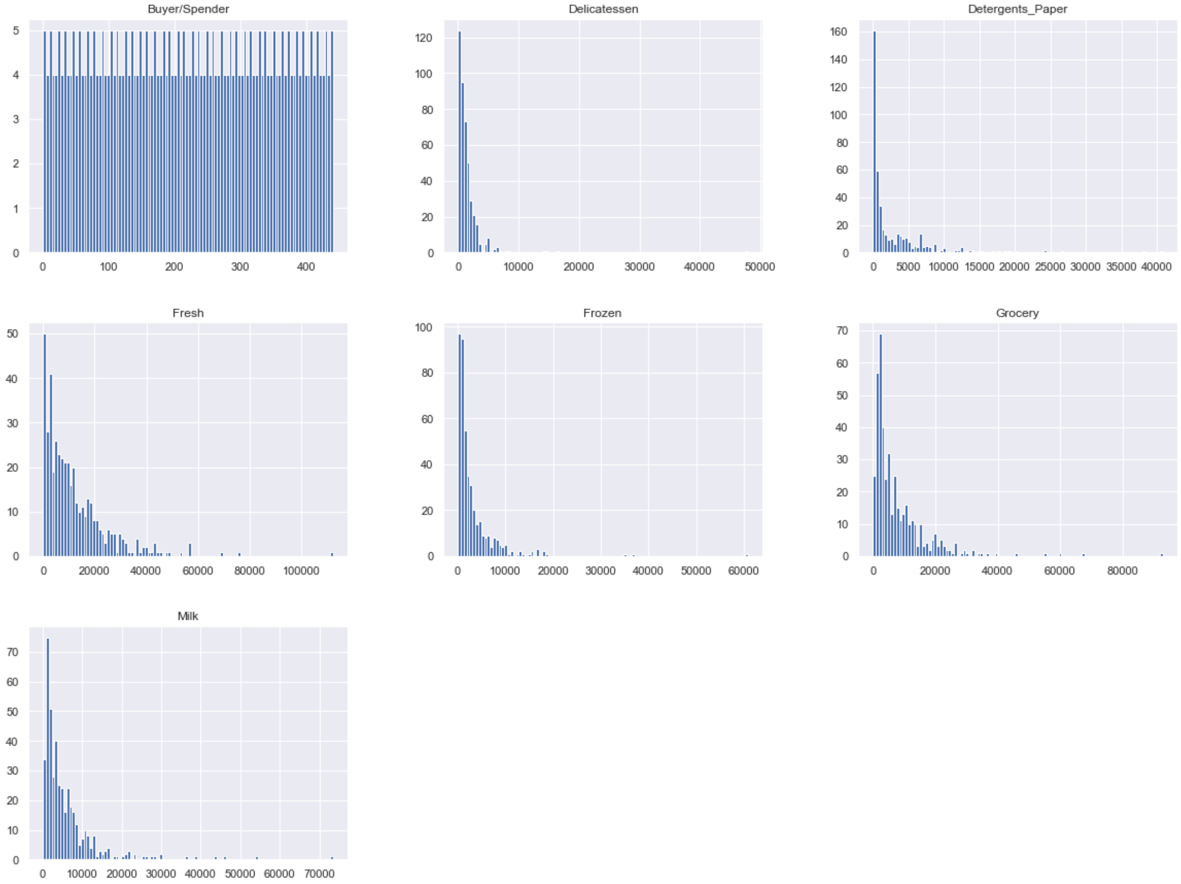
```
# Lets try to visualise the Data by plotting Histogram.
```

executed in 9ms, finished 21:14:44 2020-12-12

In [67]:

```
p1.hist(bins=100,figsize=(20,15))
plt.show()
```

executed in 4.65s, finished 19:04:56 2020-12-07



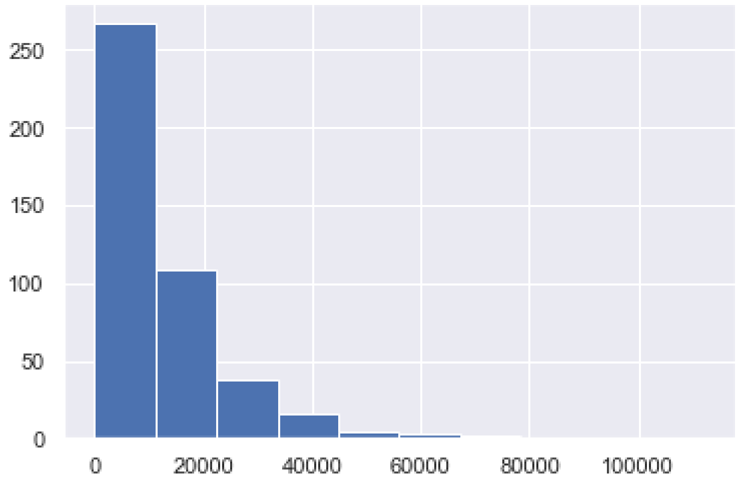
In [68]:

```
plt.hist(p1['Fresh']) #### Most inconsistent behavioural variable.
```

executed in 416ms, finished 19:07:27 2020-12-07

Out[68]:

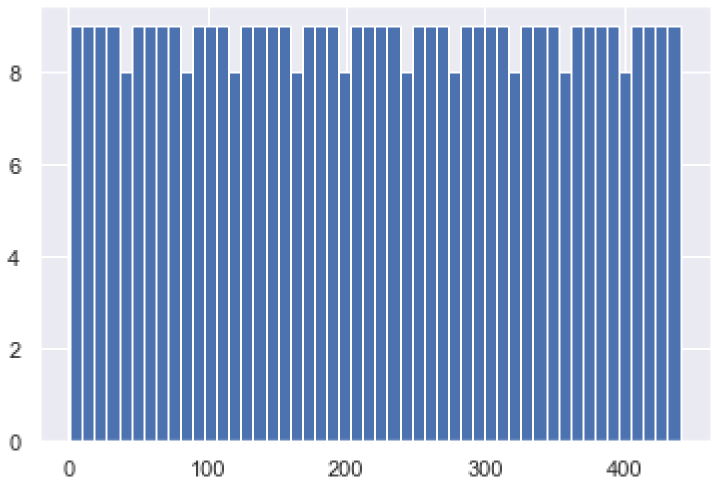
```
(array([267., 109.,  38.,  16.,   4.,   3.,   2.,   0.,   0.,   1.]),
 array([3.000000e+00, 1.121780e+04, 2.243260e+04, 3.364740e+04,
        4.486220e+04, 5.607700e+04, 6.729180e+04, 7.850660e+04,
        8.972140e+04, 1.009362e+05, 1.121510e+05]),
 <a list of 10 Patch objects>)
```



In [72]:

```
plt.hist(p1['Buyer/Spender'],bins=50); ### Most consistent behavioural variable.
```

executed in 581ms, finished 19:11:38 2020-12-07



1.4 Are there any outliers in the data?

#We can simply answer this question by checking the description of all the variables itself and YES there are outliers in the given Data.

Lets visualise how many and how far are they.

Outliers are those items/values in Data which can impact the End result hence its most important to figure out about the Outliers.

Only Numerical variable will be able to have the Outliers as it deals only with numbers.

In [83]:

```
### Lets create a new data frame which will only have a Numerical Variable and will store i

num_p1=p1[p1.dtypes[p1.dtypes !='object'].index]
```

executed in 14ms, finished 23:28:16 2020-12-07

In [91]:

```
# Lets check whether we successfully create a new df.

num_p1.head()
```

executed in 29ms, finished 23:38:13 2020-12-07

Out[91]:

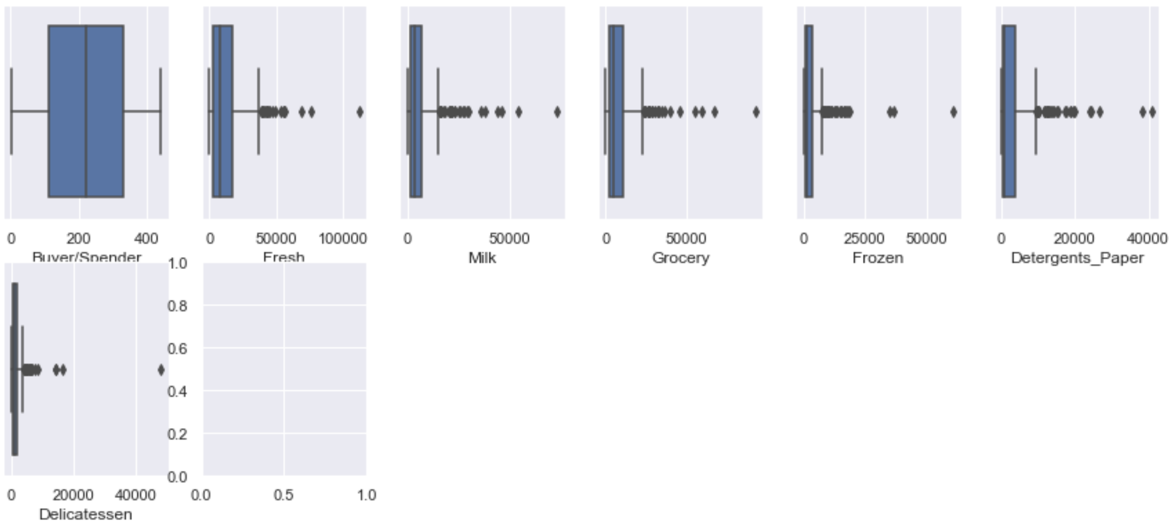
	Buyer/Spender	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
0	1	12669	9656	7561	214	2674	1338
1	2	7057	9810	9568	1762	3293	1776
2	3	6353	8808	7684	2405	3516	7844
3	4	13265	1196	4221	6404	507	1788
4	5	22615	5410	7198	3915	1777	5185

In [102]:

```
# So now there is no Categrical Variable so we can check the outliers using Loop.


fig=plt.figure(figsize=(15,20))

for i in range (0,len(num_p1)):
    ax=fig.add_subplot(6,6,i+1)
    sns.boxplot(num_p1[num_p1.columns[i]])
```

executed in 2.56s, finished 23:41:55 2020-12-07

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-102-50003ac4a6be> in <module>
      6 for i in range (0,len(num_p1)):
      7     ax=fig.add_subplot(6,6,i+1)
----> 8     sns.boxplot(num_p1[num_p1.columns[i]])

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in __
getitem__(self, key)
   4278         if is_scalar(key):
   4279             key = com.cast_scalar_indexer(key)
-> 4280             return getitem(key)
   4281 
   4282         if isinstance(key, slice):

IndexError: index 7 is out of bounds for axis 0 with size 7
```



In [103]:

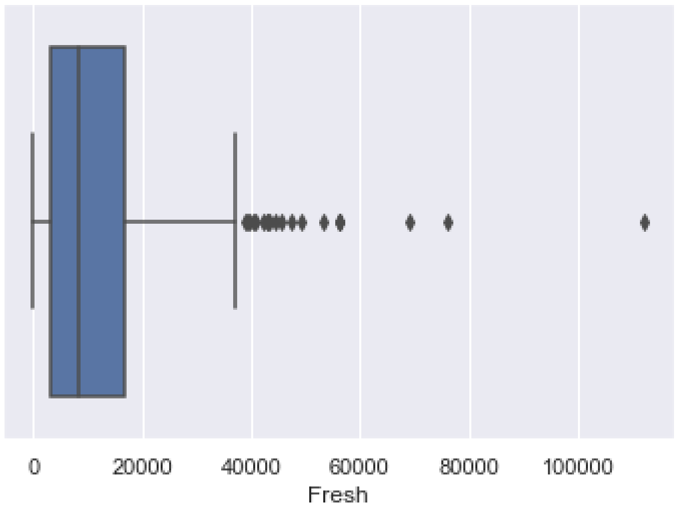
```
### Lets find out the IQR to check where the maximum data lies for every variable and also
```

executed in 11ms, finished 23:48:36 2020-12-07

In [171]:

```
sns.boxplot(p1['Fresh']);
```

executed in 3.06s, finished 14:32:45 2020-12-12



In [104]:

```
q1 = p1.quantile(0.25)
q3 = p1.quantile(0.75 , numeric_only=True,axis=0)
```

executed in 11ms, finished 23:50:01 2020-12-07

In [106]:

```
IQR = q3 - q1
print(IQR)
```

executed in 534ms, finished 23:50:27 2020-12-07

```
Buyer/Spender         219.50
Fresh               13806.00
Milk                 5657.25
Grocery              8502.75
Frozen               2812.00
Detergents_Paper     3665.25
Delicatessen         1412.00
dtype: float64
```

In [107]:

```
IQR * 1.5
```

executed in 10ms, finished 23:50:58 2020-12-07

Out[107]:

```
Buyer/Spender         329.250
Fresh               20709.000
Milk                 8485.875
Grocery             12754.125
Frozen               4218.000
Detergents_Paper     5497.875
Delicatessen         2118.000
dtype: float64
```

In [108]:

```
p1.describe()
```

executed in 102ms, finished 23:51:13 2020-12-07

Out[108]:

	Buyer/Spender	Fresh	Milk	Grocery	Frozen	Detergents_Pap
count	440.000000	440.000000	440.000000	440.000000	440.000000	440.0000
mean	220.500000	12000.297727	5796.265909	7951.277273	3071.931818	2881.4931
std	127.161315	12647.328865	7380.377175	9503.162829	4854.673333	4767.8544
min	1.000000	3.000000	55.000000	3.000000	25.000000	3.0000
25%	110.750000	3127.750000	1533.000000	2153.000000	742.250000	256.7500
50%	220.500000	8504.000000	3627.000000	4755.500000	1526.000000	816.5000
75%	330.250000	16933.750000	7190.250000	10655.750000	3554.250000	3922.0000
max	440.000000	112151.000000	73498.000000	92780.000000	60869.000000	40827.0000

It clearly shows how many Outliers are present in every single Numerical Variable when its compare to their Max value and IQR (Max Data region) for every Numerical variable.

So if one of my variable's data is having IQR (which means 50%) of my data and the max value is too far anf above max value all will

consider as an outliers because they are not contributing for meaning the all Data.

Problem 2

The Student News Service at Clear Mountain State University (CMSU) has decided to gather data about the undergraduate students that attend CMSU. CMSU creates and distributes a survey of 14 questions and receives responses from 62 undergraduates (stored in the Survey data set).

In [2]:

```
p2=pd.read_csv(r'E:\Great Learning\Projects\Survey-1.csv')
```

executed in 563ms, finished 22:37:37 2020-12-12

In [114]:

```
# Lets check few basic understanding of Data.

p2.head(8)
```

executed in 251ms, finished 16:59:52 2020-12-08

Out[114]:

	ID	Gender	Age	Class	Major	Grad Intention	GPA	Employment	Salary	Soc Networki
0	1	Female	20	Junior	Other	Yes	2.9	Full-Time	50.0	
1	2	Male	23	Senior	Management	Yes	3.6	Part-Time	25.0	
2	3	Male	21	Junior	Other	Yes	2.5	Part-Time	45.0	
3	4	Male	21	Junior	CIS	Yes	2.5	Full-Time	40.0	
4	5	Male	23	Senior	Other	Undecided	2.8	Unemployed	40.0	
5	6	Female	22	Senior	Economics/Finance	Undecided	2.3	Unemployed	78.0	
6	7	Female	21	Junior	Other	Undecided	3.0	Part-Time	50.0	
7	8	Female	22	Senior	Other	Undecided	3.1	Full-Time	80.0	

In [115]:

```
p2.info()
```

executed in 348ms, finished 17:00:14 2020-12-08

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 14 columns):
ID                   62 non-null int64
Gender               62 non-null object
Age                  62 non-null int64
Class                62 non-null object
Major                62 non-null object
Grad Intention       62 non-null object
GPA                  62 non-null float64
Employment           62 non-null object
Salary               62 non-null float64
Social Networking    62 non-null int64
Satisfaction         62 non-null int64
Spending             62 non-null int64
Computer             62 non-null object
Text Messages        62 non-null int64
dtypes: float64(2), int64(6), object(6)
memory usage: 6.9+ KB
```

In [116]:

```
p2.size
```

executed in 113ms, finished 17:00:23 2020-12-08

Out[116]:

```
868
```

In [118]:

```
p2.shape
```

executed in 23ms, finished 17:00:33 2020-12-08

Out[118]:

```
(62, 14)
```

In [120]:

```
p2.columns
```

executed in 11ms, finished 17:00:50 2020-12-08

Out[120]:

```
Index(['ID', 'Gender', 'Age', 'Class', 'Major', 'Grad Intention', 'GPA',
       'Employment', 'Salary', 'Social Networking', 'Satisfaction', 'Spendin
g',
       'Computer', 'Text Messages'],
      dtype='object')
```

In [121]:

```
p2.isnull().sum()
```

executed in 67ms, finished 17:16:14 2020-12-08

Out[121]:

```
ID                   0
Gender               0
Age                  0
Class                0
Major                0
Grad Intention       0
GPA                  0
Employment           0
Salary               0
Social Networking    0
Satisfaction         0
Spending             0
Computer             0
Text Messages        0
dtype: int64
```

In [122]:

```
# So null or missing value in the Data set which means it is balanced.
```

executed in 13ms, finished 17:26:28 2020-12-08

2.1. For this data, construct the following contingency tables (Keep Gender as row variable)

In [123]:

```
### 2.1.1. Gender and Major

### 2.1.2. Gender and Grad Intention

###2.1.3. Gender and Employment

### 2.1.4. Gender and Computer
```

executed in 5ms, finished 17:32:30 2020-12-08

In [7]:

```
### 2.1.1. Gender and Major

d = pd.crosstab (p2['Gender'],p2['Major'],margins=True)
```

executed in 5.76s, finished 14:28:49 2020-12-10

In [8]:

```
d
```

executed in 31ms, finished 14:28:51 2020-12-10

Out[8]:

Major	Accounting	CIS	Economics/Finance	International Business	Management	Other	Retailing/Mark
Gender							
Female	3	3	7	4	4	3	
Male	4	1	4	2	6	4	
All	7	4	11	6	10	7	

In [14]:

```
### 2.1.2. Gender and Grad Intention

r = pd.crosstab(p2['Gender'],p2['Grad Intention'],margins=True)
```

executed in 86ms, finished 14:36:18 2020-12-10

In [15]:

```
r
```

executed in 18ms, finished 14:36:20 2020-12-10

Out[15]:

Grad Intention	No	Undecided	Yes	All
Gender				
Female	9	13	11	33
Male	3	9	17	29
All	12	22	28	62

In [22]:

```
### 2.1.4. Gender and Empployment
p = pd.crosstab(p2['Gender'],p2['Employment'],margins=True)
```

executed in 92ms, finished 14:49:01 2020-12-10

In [23]:

```
p
```

executed in 19ms, finished 14:49:03 2020-12-10

Out[23]:

Employment	Full-Time	Part-Time	Unemployed	All
Gender				
Female	3	24	6	33
Male	7	19	3	29
All	10	43	9	62

In [18]:

```
#### 2.1.4. Gender and Computer

t = pd.crosstab(p2['Gender'],p2['Computer'],margins=True)
```

executed in 82ms, finished 14:43:54 2020-12-10

In [19]:

```
t
```

executed in 13ms, finished 14:43:56 2020-12-10

Out[19]:

Computer	Desktop	Laptop	Tablet	All
Gender				
Female	2	29	2	33
Male	3	26	0	29
All	5	55	2	62

2.2. Assume that the sample is representative of the population of CMSU. Based on the data, answer the following question:

In [147]:

```
### 2.2.1. What is the probability that a randomly selected CMSU student will be male?

p2['Gender'].value_counts()
```

executed in 127ms, finished 18:54:31 2020-12-08

Out[147]:

```
Female    33
Male      29
Name: Gender, dtype: int64
```

In [6]:

```
Total_Students = 62
Male_Students = 29
#29 / 62
prob_rand_male = Male_Students / Total_Students

print('Probability of randomly selected CMSU student will be male is',prob_rand_male)
```

executed in 7ms, finished 22:38:52 2020-12-12

```
Probability of randomly selected CMSU student will be male is 0.467741935483
87094
```

In [7]:

```
# 2.2.2. What is the probability that a randomly selected CMSU student will be female?
Female_Students = 33

prob_random_female = Female_Students / Total_Students
print('Probability of randomly selected CMSU student will be Female is',prob_random_female)
```

executed in 19ms, finished 22:39:56 2020-12-12

```
Probability of randomly selected CMSU student will be Female is 0.5322580645
16129
```

2.3. Assume that the sample is representative of the population of CMSU. Based on the data, answer the following question:

2.3.1. Find the conditional probability of different majors among the male students in CMSU.

Contional probability is to find the probability of one event when other event is already done.

Example : To find conditional probability of event A when Event B already occur.

It is represented by : P(A/B)= P(A and B) / P(B).

P(A and B) = common elements in both the events.

P(B) = Sample space of complete event.

so in the above question we need to find out the P(Majors/Male students).

Lets check the data.

In [10]:

```
# There are total 62  Majors and out of it 29 are Male.
Male_Students = 29
total_majors = 62
Conditional_prob = Male_Students / total_majors
print('The conditional probability of different majors among the male students in CMSU is',
```

executed in 14ms, finished 22:42:37 2020-12-12

```
The conditional probability of different majors among the male students in C
MSU is 0.46774193548387094
```

In [11]:

```
# 2.3.2 Find the conditional probability of different majors among the female students of C
# same for the female students will be :
total_majors = 62
Female_Students = 33
Condi_prob_female = Female_Students / total_majors
print('The conditional probability of different majors among the female students of CMSU is
```

executed in 10ms, finished 22:43:38 2020-12-12

```
The conditional probability of different majors among the female students of
CMSU is 0.532258064516129
```

2.4. Assume that the sample is a representative of the population of CMSU. Based on the data, answer the following question:

In [16]:

```
# 2.4.1 Find the probability That a randomly chosen student is a male and intends to gradua

r
```

executed in 13ms, finished 14:36:24 2020-12-10

Out[16]:

Grad Intention	No	Undecided	Yes	All
Gender				
Female	9	13	11	33
Male	3	9	17	29
All	12	22	28	62

In [13]:

```
# So total  male grad intention are 17 and total male are 29.
grad_intention_male = 17
Male_Students = 29
Male_grad_int = grad_intention_male/ Male_Students
print ('The probability That a randomly chosen student is a male and intends to graduate is
```

executed in 10ms, finished 22:45:09 2020-12-12

```
The probability That a randomly chosen student is a male and intends to grad
uate is 0.5862068965517241
```

In [20]:

```
# 2.4.2 Find the probability that a randomly selected student is a female and does NOT have
t
```

executed in 24ms, finished 14:44:01 2020-12-10

Out[20]:

Computer	Desktop	Laptop	Tablet	All
Gender				
Female	2	29	2	33
Male	3	26	0	29
All	5	55	2	62

In [17]:

```
# There are total 33 female students and only 4 not have laptop.
no_lap = 4
Female_Students = 33
female_not_laptop = no_lap / Female_Students
print('The probability that a randomly selected student is a female and does NOT have a lap
```

executed in 6ms, finished 22:46:26 2020-12-12

```
The probability that a randomly selected student is a female and does NOT ha
ve a laptop is 0.12121212121212122
```

2.5. Assume that the sample is a representative of the population of CMSU. Based on the data, answer the following question:

In [24]:

```
# 2.5.1 Find the probability that a randomly chosen student is either a male or has a full-
p
```

executed in 12ms, finished 14:49:09 2020-12-10

Out[24]:

Employment	Full-Time	Part-Time	Unemployed	All
Gender				
Female	3	24	6	33
Male	7	19	3	29
All	10	43	9	62

In [23]:

```
# Total full time employement male are 7.
Male_Students = 29
total_full_male_employement = 7
prob_stu_male_full = total_full_male_employement / Male_Students



print('the probability that a randomly chosen student is male and have full time employment
```

executed in 7ms, finished 22:50:24 2020-12-12

```
the probability that a randomly chosen student is male and have full time em
ployment is 0.2413793103448276
```

In [24]:

```
# 2.5.2 Find the conditional probability that given a female student is randomly chosen, sh

# Total females who are major in International business and management are 8
Tot_fe_in_inter_busines = 8
Female_Students = 33
prob_fem_maj_int_bus= Tot_fe_in_inter_busines / Female_Students
print ('The conditional probability that given a female student is randomly chosen, she is
```

executed in 9ms, finished 22:52:11 2020-12-12

```
The conditional probability that given a female student is randomly chosen,
she is majoring in international business or management 0.24242424242424243
```

2.6 Construct a contingency table of Gender and Intent to Graduate at 2 levels (Yes/No). The Undecided students are not considered now and the table is a 2x2 table. Do you think graduate intention and being female are independent events?

In [28]:

```
r
```

executed in 23ms, finished 15:32:11 2020-12-10

Out[28]:

Grad Intention	No	Undecided	Yes	All
Gender				
Female	9	13	11	33
Male	3	9	17	29
All	12	22	28	62

There are Total 33 Females with Grad Intention No and Yes - No - 9 (27 %) and Yes - 11 (33%)

So it clearly shows that Females are not dependent on Grad Intention and this both are Independent Events.

Some Female students are not Grad and Some are Grad intention which directly reflect

that the both events are not connected and hence both are Independent Events.

2.7 Note that there are four numerical (continuous) variables in the data set, GPA, Salary, Spending and Text Messages. Answer the following questions based on the data

In [34]:

```
### for easy understanding lets drop all the discontinous columns from the existing data se

# first check all the columns
p2.columns
```

executed in 14ms, finished 15:36:27 2020-12-10

Out[34]:

```
Index(['ID', 'Gender', 'Age', 'Class', 'Major', 'Grad Intention', 'GPA',
       'Employment', 'Salary', 'Social Networking', 'Satisfaction', 'Spendin
g',
       'Computer', 'Text Messages'],
      dtype='object')
```

In [35]:

```
df = p2.drop(columns=['Age','Class','Major','Grad Intention','Employment','Social Networkir
```

executed in 11ms, finished 15:39:43 2020-12-10

In [36]:

```
df.head()
```

executed in 37ms, finished 15:39:47 2020-12-10

Out[36]:

	ID	Gender	GPA	Salary	Spending	Text Messages
0	1	Female	2.9	50.0	350	200
1	2	Male	3.6	25.0	360	50
2	3	Male	2.5	45.0	600	200
3	4	Male	2.5	40.0	600	250
4	5	Male	2.8	40.0	500	100

In [41]:

```
### 2.7.1 If a student is chosen randomly, what is the probability that his/her GPA is less

o = df['GPA'] < 3
```

executed in 16ms, finished 15:41:58 2020-12-10

In [38]:

```
o
```

executed in 10ms, finished 15:40:46 2020-12-10

Out[38]:

```
0      True
1     False
2      True
3      True
4      True
      ...  
57     True
58     True
59     True
60    False
61    False
Name: GPA, Length: 62, dtype: bool
```

In [25]:

```
### There are 24 Students have less than 3 GPA.

stu_less_gpa = 24
Total_Students = 62
prob_gpa_less_3 = stu_less_gpa/ Total_Students

print('If a student is chosen randomly, then the probability that his/her GPA is less than
```

executed in 19ms, finished 22:55:21 2020-12-12

```
If a student is chosen randomly, then the probability that his/her GPA is le
ss than 3 is 0.3870967741935484
```

In [43]:

```
# 2.7.2 Find conditional probability that a randomly selected male earns 50 or more.
# Find conditional probability that a randomly selected female earns 50 or more.

s = pd.crosstab(df['Gender'],df['Salary'],margins=True)
```

executed in 335ms, finished 15:46:34 2020-12-10

In [44]:

```
s
```

executed in 29ms, finished 15:46:36 2020-12-10

Out[44]:

Salary	25.0	30.0	35.0	37.0	37.5	40.0	42.0	45.0	47.0	47.5	50.0	52.0	54.0	55.0	60.0
Gender															
Female	0	5	1	0	1	5	1	1	0	1	5	0	0	5	5
Male	1	0	1	1	0	7	0	4	1	0	4	1	1	3	3
All	1	5	2	1	1	12	1	5	1	1	9	1	1	8	8

In [26]:

```
Male_Students = 29
Female_Students = 33
Male_earns_more_50 = 14
Female_earns_more_50 = 18
con_prob_male_earns = Male_earns_more_50 / Male_Students
con_prob_female_earns = Female_earns_more_50 / Female_Students
print('conditional probability that a randomly selected male earns 50 or more is',con_prob_
print('conditional probability that a randomly selected female earns 50 or more is',con_pro
```

executed in 14ms, finished 22:58:30 2020-12-12

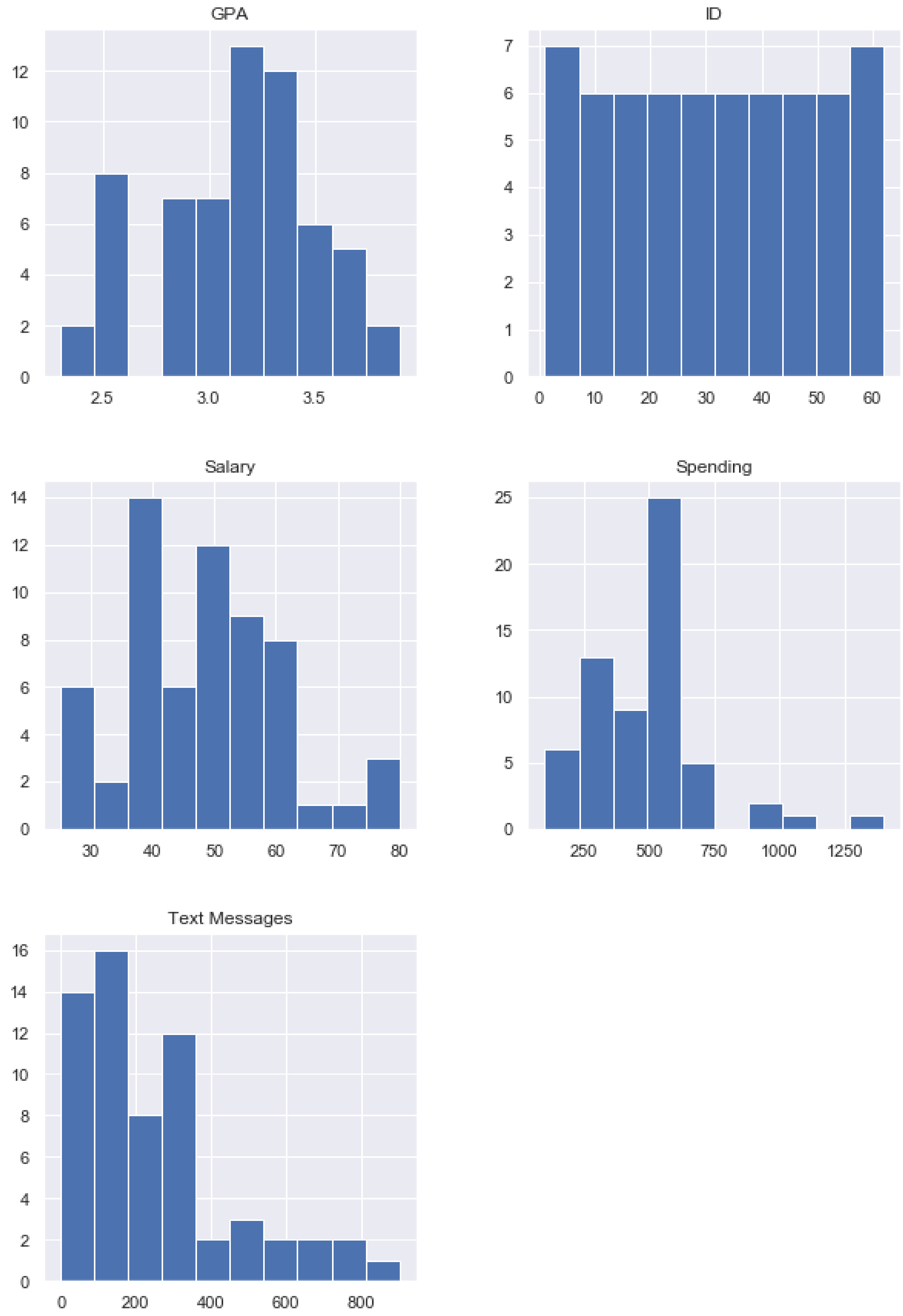
```
conditional probability that a randomly selected male earns 50 or more is 0.
4827586206896552
conditional probability that a randomly selected female earns 50 or more is
0.5454545454545454
```

2.8.1 Note that there are four numerical (continuous) variables in the data set, GPA, Salary, Spending and Text Messages. For each of them comment whether they follow a normal distribution.

In [82]:

```
df.hist(bins=10,figsize=(10,15));
plt.show()
```

executed in 1.76s, finished 16:34:12 2020-12-10



In [85]:

```
df.describe()
```

executed in 191ms, finished 16:40:16 2020-12-10

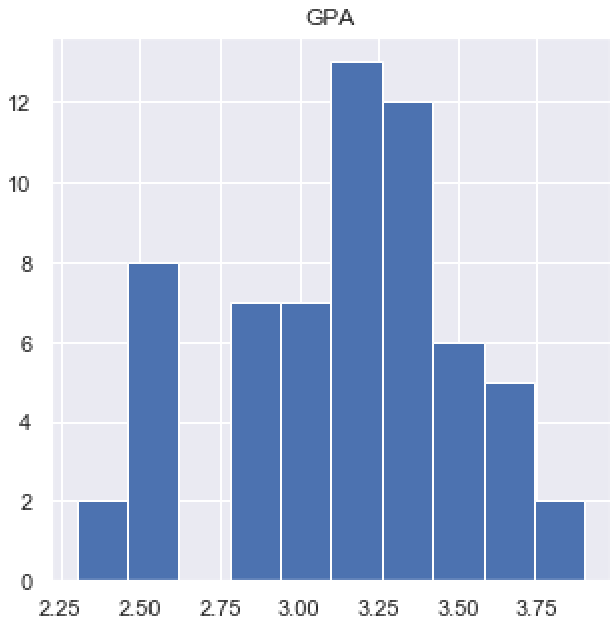
Out[85]:

	ID	GPA	Salary	Spending	Text Messages
count	62.000000	62.000000	62.000000	62.000000	62.000000
mean	31.500000	3.129032	48.548387	482.016129	246.209677
std	18.041619	0.377388	12.080912	221.953805	214.465950
min	1.000000	2.300000	25.000000	100.000000	0.000000
25%	16.250000	2.900000	40.000000	312.500000	100.000000
50%	31.500000	3.150000	50.000000	500.000000	200.000000
75%	46.750000	3.400000	55.000000	600.000000	300.000000
max	62.000000	3.900000	80.000000	1400.000000	900.000000

In [109]:

```
df.hist(column='GPA',figsize=(5,5));
plt.show()
```

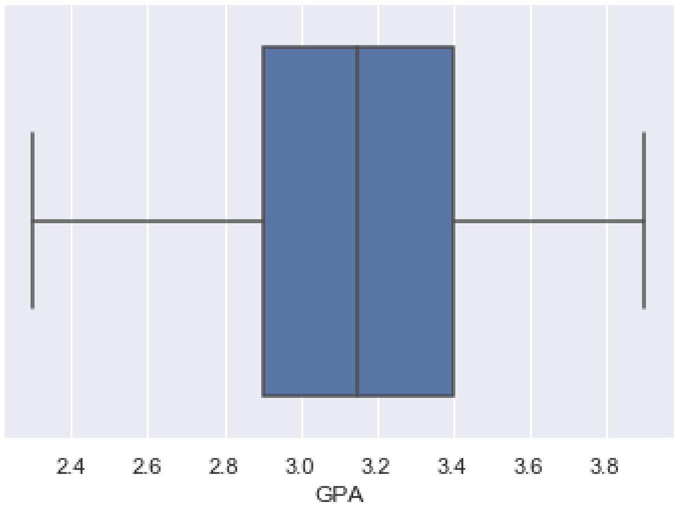
executed in 2.69s, finished 18:00:37 2020-12-10



In [89]:

```
sns.boxplot(df['GPA']);
```

executed in 392ms, finished 16:42:31 2020-12-10



In [96]:

```
t.head()
```

executed in 24ms, finished 16:47:26 2020-12-10

Out[96]:

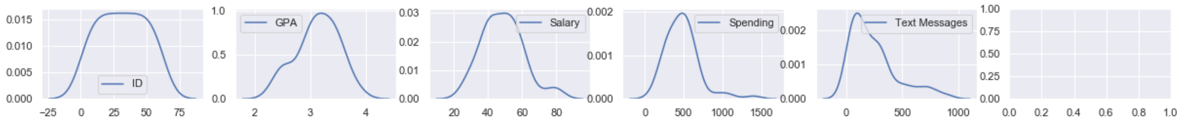
	ID	GPA	Salary	Spending	Text Messages
0	1	2.9	50.0	350	200
1	2	3.6	25.0	360	50
2	3	2.5	45.0	600	200
3	4	2.5	40.0	600	250
4	5	2.8	40.0	500	100

In [103]:

```
fig=plt.figure(figsize=(35,20))

for i in range (0,len(t)):
    ax=fig.add_subplot(10,10,i+1)
    sns.kdeplot(t[t.columns[i]])
```

executed in 2.90s, finished 16:50:28 2020-12-10

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-103-0ded3621f6d2> in <module>
      3 for i in range (0,len(t)):
      4     ax=fig.add_subplot(10,10,i+1)
----> 5     sns.kdeplot(t[t.columns[i]])

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in __
getitem__(self, key)
   4278         if is_scalar(key):
   4279             key = com.cast_scalar_indexer(key)
-> 4280             return getitem(key)
   4281 
   4282         if isinstance(key, slice):

IndexError: index 5 is out of bounds for axis 0 with size 5
```



2.8.2 Write a note summarizing your conclusions.

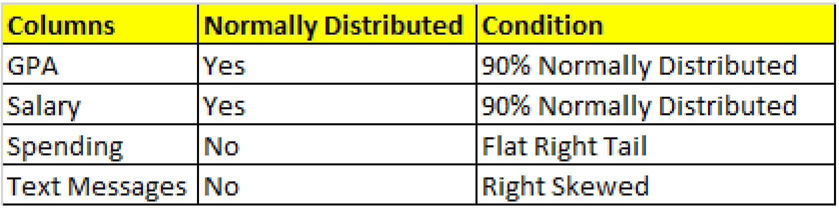
1 :GPA : As per the above images distribution of GPA is not exactly the Normal distribution as its somehow skewed at the beginning but later its distributed properly so we can say that it has a “NORMAL DISTRIBUTIED and also with the measure of central tendencies we can rectify that it is Normally Distributed.

2 :SALARY : As we can see at both the images it’s a little bit right skewed but not as a flat tail and when we checked the Central measures it shows that the Data is almost normally distributed so YES we can say that this is also Normally distributed.

3 : SPENDING : In the above images if Data allows us to cap the value at the certain level say around 800 then it will become a Normal Distribution but with the existing Data set it is not Normally distributed.

4 : Above distribution is not Normally Distributed as it has a Flat tail starting from the middle to the End.It is Right skewed and central tendency which we know for this Column is Mean 246.2 and Std. dev is 214.5 which are somehow close to each other and the concept of Normally Distributed is totally fail in this Column, hence this is not Normally Distributed.

Salary Yes 90% Normally Distributed Spending No Flat Right Tail Text Messages No Right Skewed



Columns	Normally Distributed	Condition
GPA	Yes	90% Normally Distributed
Salary	Yes	90% Normally Distributed
Spending	No	Flat Right Tail
Text Messages	No	Right Skewed

Problem 3

An important quality characteristic used by the manufacturers of ABC asphalt shingles is the amount of moisture the shingles contain when they are packaged. Customers may feel that they have purchased a product lacking in quality if they find moisture and wet shingles inside the packaging. In some cases, excessive moisture can cause the granules attached to the shingles for texture and colouring purposes to fall off the shingles resulting in appearance problems. To monitor the amount of moisture present, the company conducts moisture tests. A shingle is weighed and then dried. The shingle is then reweighed, and based on the amount of moisture taken out of the product, the pounds of moisture per 100 square feet are calculated. The company would like to show that the mean moisture content is less than 0.35 pound per 100 square feet.The file (A & B shingles.csv) includes 36 measurements (in pounds per 100 square feet) for A shingles and 31 for B shingles.

In [110]:

```
p3=pd.read_csv(r'E:\Great Learning\Projects\A & B shingles-1.csv')
```

executed in 63ms, finished 18:22:42 2020-12-10

In [111]:

```
p3.head()
```

executed in 22ms, finished 18:22:47 2020-12-10

Out[111]:

```
      A     B
0  0.44  0.14
1  0.61  0.15
2  0.47  0.31
3  0.30  0.16
4  0.15  0.37
```

In [112]:

```
p3.columns
```

executed in 11ms, finished 18:22:57 2020-12-10

Out[112]:

```
Index(['A', 'B'], dtype='object')
```

In [113]:

```
p3.size
```

executed in 6ms, finished 18:26:42 2020-12-10

Out[113]:

```
72
```

In [115]:

```
p3.shape
```

executed in 20ms, finished 18:26:55 2020-12-10

Out[115]:

```
(36, 2)
```

So There are 2 Numeric Columns and Data is Continuous as per the above.No Variable columns available.It is having 36 Rows and 2 Columns.Total size of 72.All the above information we calculated by running the codes in the Python.

In [118]:

```
### 3.1 Do you think there is evidence that mean moisture contents in both types of shingle

### Let us check the mean of both the Shingles.

p3['A'].mean()
```

executed in 16ms, finished 23:15:18 2020-12-10

Out[118]:

```
0.3166666666666666
```

In [123]:

```
p3['B'].mean()
```

executed in 8ms, finished 23:16:27 2020-12-10

Out[123]:

```
0.2735483870967742
```

In [124]:

```
p3.describe()
```

executed in 93ms, finished 23:17:15 2020-12-10

Out[124]:

```
               A          B
count  36.000000  31.000000
 mean   0.316667   0.273548
  std   0.135731   0.137296
  min   0.130000   0.100000
  25%   0.207500   0.160000
  50%   0.290000   0.230000
  75%   0.392500   0.400000
  max   0.720000   0.580000
```

Based on the objective we need to show/prove that the Mean Moisture in both (A and B) shingles are less than 0.35 pound per 100 Square Feet.

Its not mention that the given DATA is based on Population Data so we will use data as a Sample.

Steps

1 :Lets calculate the MEAN of Each column as we have given (A and B)

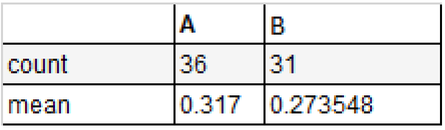
2: A is having 36 Measurements.B is having 31 Measurements.

3:Mean of A (measurement) is which can be calculated by using (p3['A']. mean()) = 0.31666

4:Mean of B (measurement) is which can be calculated by using (p3['A']. mean()) = 0.273548

5 : A B

count 36 31 mean 0.316667 0.273548



	A	B
count	36	31
mean	0.317	0.273548

6 : So above table clearly shows that both the Means are within the Permissible limit as max limit is less than 0.35 pound per 100 Sqr feet.

Hence Objective of the company is clearly achieved.

In [125]:

```
### 3.2 Do you think that the population means for shingles A and B are equal?
#Form the hypothesis and conduct the test of the hypothesis.
#What assumption do you need to check before the test for equality of means is performed?
```

executed in 6ms, finished 23:30:45 2020-12-10

Population Means for Shingles A and B are not equal:

As of now we have total of 36 counts for Shingles A and 31 counts for Shingles B and their mean are different.

We know the given Data is a Sample Data and the mean of given Data is Different , if :

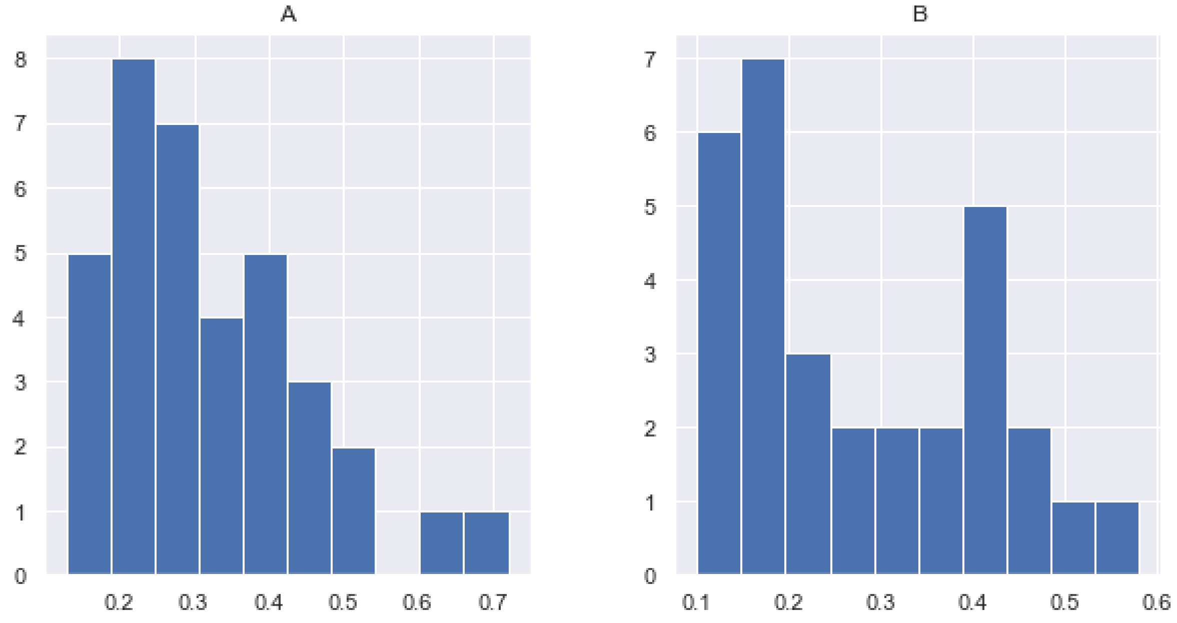
We consider that Population Mean for Shingles A and B are equal then concept of Sample Mean will also be Equal hence we can conclude

that Sample mean of Shingles A (0.317) and B (0.2735) are different and which shows that their population mean is also different or not equal.

In [130]:

```
# lets check by drawing histogram of both the Shingles to get an idea of their distribution

p3.hist(bins=10,figsize=(10,5))
plt.show()
```

executed in 1.07s, finished 17:20:25 2020-12-11



As per the above,Data is not Normally distibuted and also it clearly shows that their Population Mean will also not Equal.

In [131]:

```
#Form the hypothesis and conduct the test of the hypothesis.
#What assumption do you need to check before the test for equality of means is performed?

# Hypothesis Testing - We conduct the test based on our observations to prove our assumptic
```

executed in 18ms, finished 17:33:34 2020-12-11

Step : 1 :

We will define the Null Hypothesis (H0) and Alternative Hypothesis (H1)

Null Hypothesis (H0) = Mean Moisture for Shingles A and Shingles B are Same/Equal.

Alternative Hypothesis (H1) = Mean Moisture for Shingles A and Shingles B are not Same/Equal.

Step : 2 :

Value of Significance (alpha) is given 0.35 (as define value is 0.05 but in the problem statement its not mentioned but Manufacturers want to show that the value is less than 0.35 so we will take this under consideration for significance.

ALPHA = 0.35.

Step : 3 :

There are two sets of Data given which are independent hence we will be doing 2 Sample independent T Test.

```
                    ##### t = x1(bar) - x2(bar)
                              ---------------
                           Sqrt (s1^2/n1 + s2^2/n2)

                      x1 (bar) = Mean of Shingles A
                      x2 (bar) = Mean of Shinges B
                      s1       = Standard Deviation of Shingles A
                      s2       = Standard Deviation of Shingles B
                      n1       = Number of Elements in Shingles A
                      n2       = Number of Elements in Shingles B.

                       X1 bar=0.316667
                       X2 bar=0.273548
                       S1= 0.135731
                       S2=0.137296
                       N1=36
                       N2=31
                       D=65 (N1+N2-2)
```

In [177]:

```
X1_bar = 0.316667
X2_bar = 0.273548
Mean = X1_bar - X2_bar
Mean
```

executed in 61ms, finished 15:38:29 2020-12-12

Out[177]:

```
0.043118999999999996
```

In [185]:

```
S1 = 0.135731
S2 = 0.137296

Std_1 = S1 * S1
Std_2 = S2 * S2
```

executed in 51ms, finished 15:42:16 2020-12-12

In [187]:

```
Std_1
```

executed in 12ms, finished 15:42:37 2020-12-12

Out[187]:

```
0.018422904360999998
```

In [188]:

```
Std_2
```

executed in 10ms, finished 15:42:47 2020-12-12

Out[188]:

```
0.018850191616
```

In [191]:

```
Std_1 / 36
```

executed in 15ms, finished 15:43:24 2020-12-12

Out[191]:

```
0.0005117473433611111
```

In [192]:

```
Std_2 / 31
```

executed in 19ms, finished 15:43:34 2020-12-12

Out[192]:

```
0.0006080706972903225
```

In [193]:

```
0.000511747343361111 + 0.0006080706972903225
```

executed in 21ms, finished 15:43:49 2020-12-12

Out[193]:

```
0.0011198180406514335
```

In [194]:

```
np.sqrt (0.0011198180406514335)
```

executed in 18ms, finished 15:44:05 2020-12-12

Out[194]:

```
0.033463682413198845
```

In [195]:

```
0.043118999999999996 / 0.033463682413198845
```

executed in 20ms, finished 15:44:32 2020-12-12

Out[195]:

```
1.288531234177409
```

In [200]:

```
t = 1.288531234177409
```

executed in 17ms, finished 16:58:22 2020-12-12

In [168]:

```
import scipy.stats as stats
from scipy.stats import ttest_1samp, ttest_ind
from statsmodels.stats.power import ttest_power
from scipy.stats import ttest_ind, ttest_ind_from_stats
from scipy.special import stdtr
```

executed in 61ms, finished 23:51:41 2020-12-11

In [201]:

```
p = 1-stats.t.cdf(t , 65, 2)
```

executed in 38ms, finished 16:58:29 2020-12-12

In [202]:

```
p
```

executed in 16ms, finished 16:58:31 2020-12-12

Out[202]:

```
0.760330298519559
```

So here our P value (0.760330298) is greater than 0.35 so We Fail to Reject the Null Hypothesis.

Which means that Shingles A and Shingles B MEAN are Same/Equal which shows even the calculation is showing that there is difference in Mean's for both the Shingles but after doing t-test it shows that

their means are SAME.Which also give us an idea even the number of counts were not exactly same but that does not impact the the MEAN.

We know that Standard Deviation for both the Shingles which are nearly the same and all the central tendencies are much saturated within the limits which shows that Data is smaller and belongs to the same group as the t-score is very less.t score is less then it belongs to same group and t score is higher than it belongs to the different group.

The conclusion is:

Number of counts for Shingles B are less than Shingles A which may give the reason for accepting that the MEANS are Same.We can increase the number of measurements for both the Shingles so that we can be able to test more for equality of MEANS also while increasing the measurements we will able to find out if there are any differences, but as per the given Data we know that the values are saturated or close to each other and t score is less so we know that they are from the same groups so ideally we can proceed that even while increasing the measurements we can say that their MEANS will be equal.

Hence objective of ABC Manufacturers for conducting Moisture Test is successfully Achieved where they will be able to show that their moisture content is less and now the Problem of excessive moisture which was causing the granules attached to the shingles for texture and colouring purposes to fall off the shingles resulting in appearance problems will not Repeat and customers will get the QUALITY product as per their expectation's.

.

In [ ]: