WHOLESALE CUSTOMER ANALYSIS

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
import os
os.chdir('C:\\Users\\WELCOME\\Downloads\\PYTHON FILES\\SMDM\\project')
```

In [3]:
```
df= pd.read_csv('Wholesale Customer.csv')
df.head()
```

Out[3]:

	Buyer/Spender	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatesse
0	1	Retail	Other	12669	9656	7561	214	2674	133
1	2	Retail	Other	7057	9810	9568	1762	3293	17
2	3	Retail	Other	6353	8808	7684	2405	3516	784
3	4	Hotel	Other	13265	1196	4221	6404	507	178
4	5	Retail	Other	22615	5410	7198	3915	1777	518

In [4]:
```
df.shape
```

Out[4]:
```
(440, 9)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 9 columns):
```

```
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Buyer/Spender     440 non-null    int64
 1   Channel           440 non-null    object
 2   Region            440 non-null    object
 3   Fresh             440 non-null    int64
 4   Milk              440 non-null    int64
 5   Grocery           440 non-null    int64
 6   Frozen            440 non-null    int64
 7   Detergents_Paper  440 non-null    int64
 8   Delicatessen      440 non-null    int64
dtypes: int64(7), object(2)
memory usage: 31.1+ KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Buyer/Spender       0
Channel             0
Region              0
Fresh               0
Milk                0
Grocery             0
Frozen              0
Detergents_Paper    0
Delicatessen        0
dtype: int64
```

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 [7]:
```
df.describe(include = 'all').T
```

Out[7]:

	count	unique	top	freq	mean	std	min	25%	50%	75%
Buyer/Spender	440	NaN	NaN	NaN	220.5	127.161	1	110.75	220.5	330.25
Channel	440	2	Hotel	298	NaN	NaN	NaN	NaN	NaN	NaN

	count	unique	top	freq	mean	std	min	25%	50%	75%
Region	440	3	Other	316	NaN	NaN	NaN	NaN	NaN	NaN
Fresh	440	NaN	NaN	NaN	12000.3	12647.3	3	3127.75	8504	16933.8
Milk	440	NaN	NaN	NaN	5796.27	7380.38	55	1533	3627	7190.25
Grocery	440	NaN	NaN	NaN	7951.28	9503.16	3	2153	4755.5	10655.8
Frozen	440	NaN	NaN	NaN	3071.93	4854.67	25	742.25	1526	3554.25
Detergents_Paper	440	NaN	NaN	NaN	2881.49	4767.85	3	256.75	816.5	3922
Delicatessen	440	NaN	NaN	NaN	1524.87	2820.11	3	408.25	965.5	1820.25

In [8]:
```
df.groupby(['Region', 'Channel']).size()
```

Out[8]:
```
Region  Channel
Lisbon  Hotel       59
        Retail      18
Oporto  Hotel       28
        Retail      19
Other   Hotel      211
        Retail     105
dtype: int64
```

In [9]:
```
dfnew = df.drop(['Buyer/Spender'], axis= 1)
dfnew.head()
```

Out[9]:

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

In [10]:
```
dfnew['Total'] = dfnew[['Fresh', 'Milk', 'Grocery', 'Frozen', 'Detergen
ts_Paper', 'Delicatessen']].sum(axis=1)
dfnew.head()
```

Out[10]:

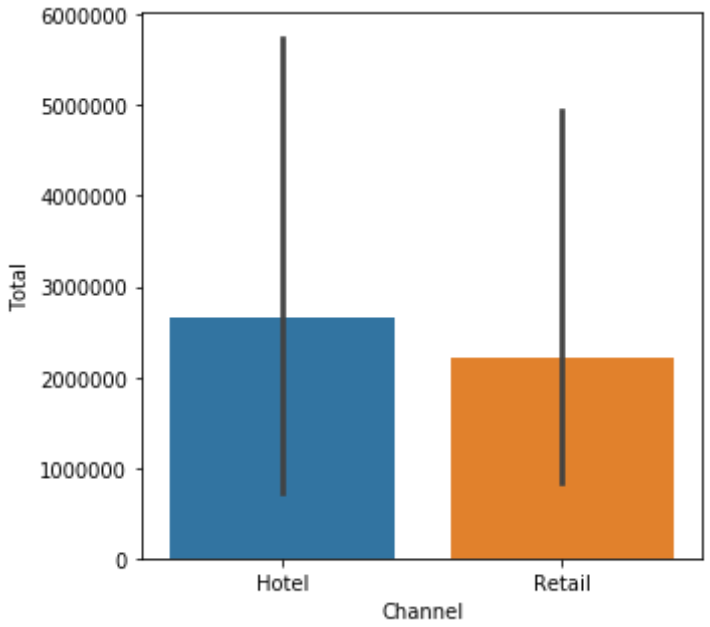
	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen	Total
0	Retail	Other	12669	9656	7561	214	2674	1338	34112
1	Retail	Other	7057	9810	9568	1762	3293	1776	33266
2	Retail	Other	6353	8808	7684	2405	3516	7844	36610
3	Hotel	Other	13265	1196	4221	6404	507	1788	27381
4	Retail	Other	22615	5410	7198	3915	1777	5185	46100

In [11]:
```
results1 = dfnew.groupby(['Region','Channel']).sum().reset_index()
results1
```

Out[11]:

	Region	Channel	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen	Tota
0	Lisbon	Hotel	761233	228342	237542	184512	56081	70632	153834
1	Lisbon	Retail	93600	194112	332495	46514	148055	33695	84847
2	Oporto	Hotel	326215	64519	123074	160861	13516	30965	71915
3	Oporto	Retail	138506	174625	310200	29271	159795	23541	83593
4	Other	Hotel	2928269	735753	820101	771606	165990	320358	574207
5	Other	Retail	1032308	1153006	1675150	158886	724420	191752	493552

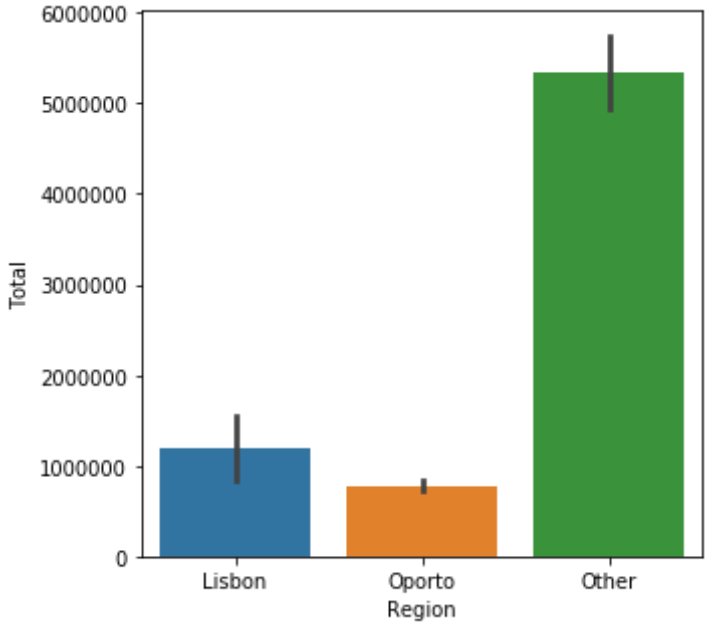
In [12]:
```
plt.figure(figsize=(5,5))
sns.barplot(x=results1['Channel'], y=results1['Total'], data =results1
);
plt.show()
dfnew.groupby('Channel',).sum().reset_index()
```



Out[12]:

	Channel	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen	Total
0	Hotel	4015717	1028614	1180717	1116979	235587	421955	7999569
1	Retail	1264414	1521743	2317845	234671	1032270	248988	6619931

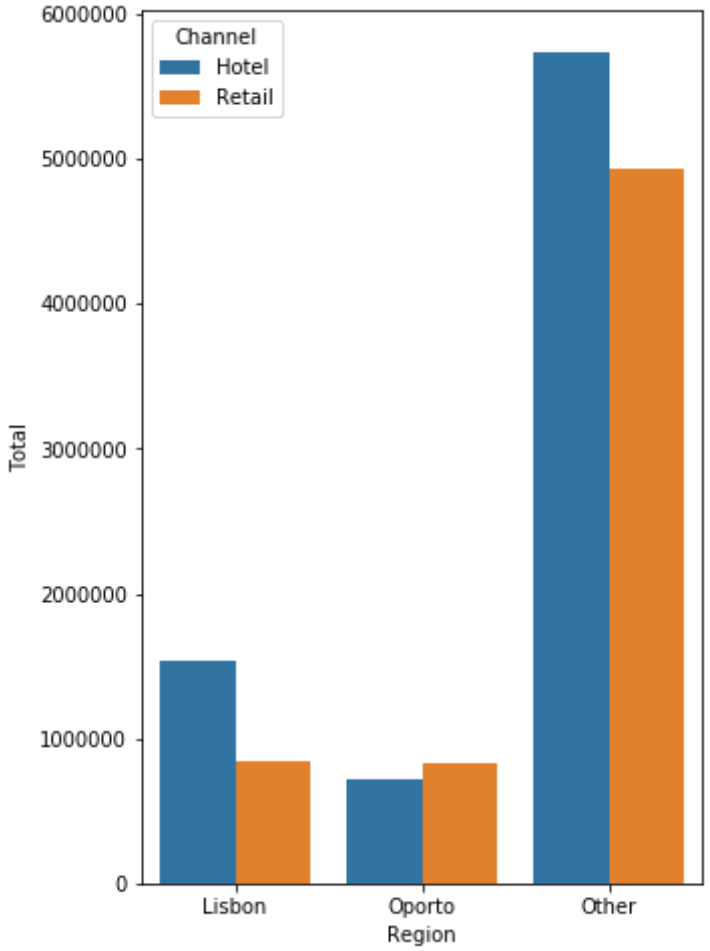
In [13]:
```
plt.figure(figsize=(5,5))
sns.barplot(x=results1['Region'], y=results1['Total'], data =results1);
plt.show()
dfnew.groupby('Region',).sum().reset_index()
```



Out[13]:

	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen	Total
0	Lisbon	854833	422454	570037	231026	204136	104327	2386813
1	Oporto	464721	239144	433274	190132	173311	54506	1555088
2	Other	3960577	1888759	2495251	930492	890410	512110	10677599

In [14]:
```
plt.figure(figsize=(5,8))
sns.barplot(x=results1['Region'], y=results1['Total'], hue='Channel', d
ata =results1);
plt.show()
```



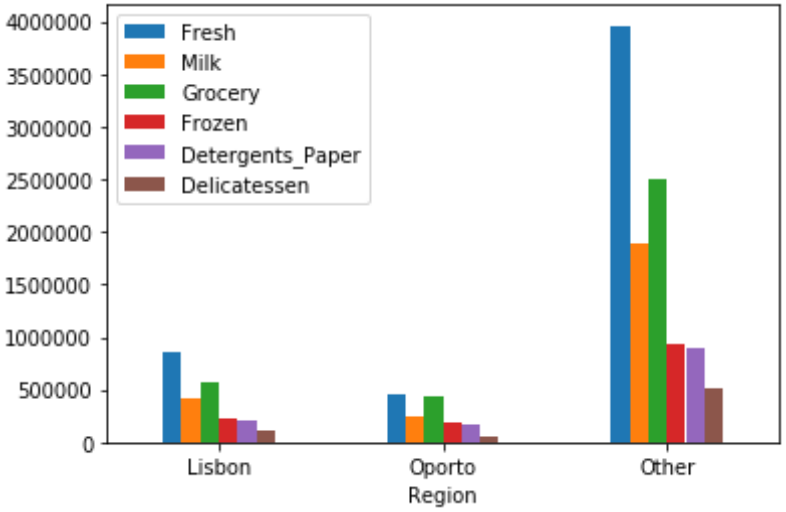
1.2 There are 6 different varieties of items are considered. Do all varieties show similar behaviour across Region and Channel?

In [15]:
```
results = dfnew.groupby('Region').sum()
resultsnew= results.drop(columns='Total')
resultsnew
```

Out[15]:

Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
Lisbon	854833	422454	570037	231026	204136	104327
Oporto	464721	239144	433274	190132	173311	54506
Other	3960577	1888759	2495251	930492	890410	512110

In [16]:
```
resultsnew.plot.bar(rot=0);
```

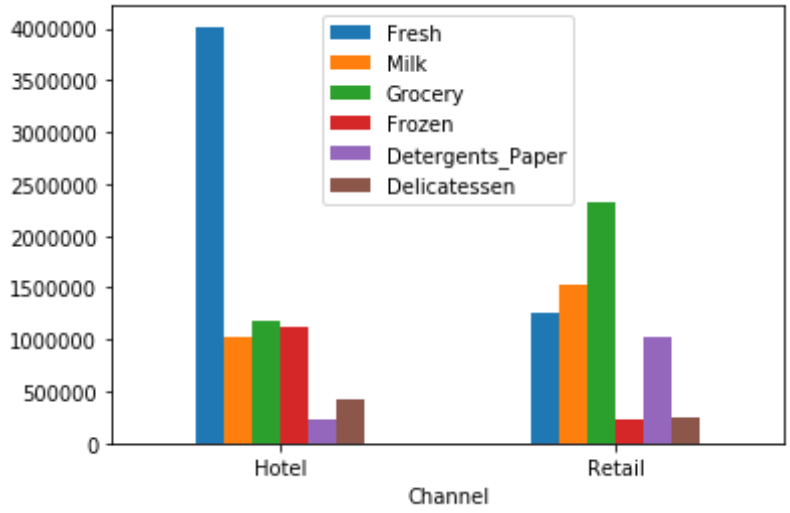


In [17]:
```
results = dfnew.groupby('Channel').sum()
results2 =results.drop(columns='Total')
results2
```

Out[17]:

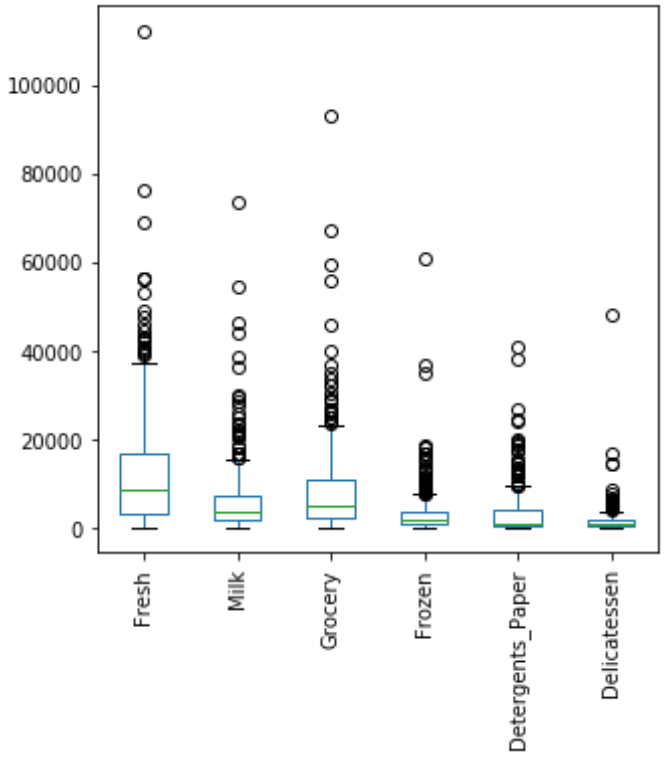
Channel	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
Hotel	4015717	1028614	1180717	1116979	235587	421955
Retail	1264414	1521743	2317845	234671	1032270	248988

In [18]:
```
results2.plot.bar(rot=0);
```



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

In [27]:
```
plt.figure (figsize=(5,5))
dfnewout.boxplot(fontsize=None,
    rot=90,
    grid=False);
plt.show()
```



In [23]:
```
Q1 =dfnewout.quantile(0.25)
Q3 = dfnewout.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

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

In [26]:
```
dfnewout.std()
```

Out[26]:
```
Fresh               12647.328865
```

```
Milk                 7380.377175
Grocery              9503.162829
Frozen               4854.673333
Detergents_Paper     4767.854448
Delicatessen         2820.105937
dtype: float64
```

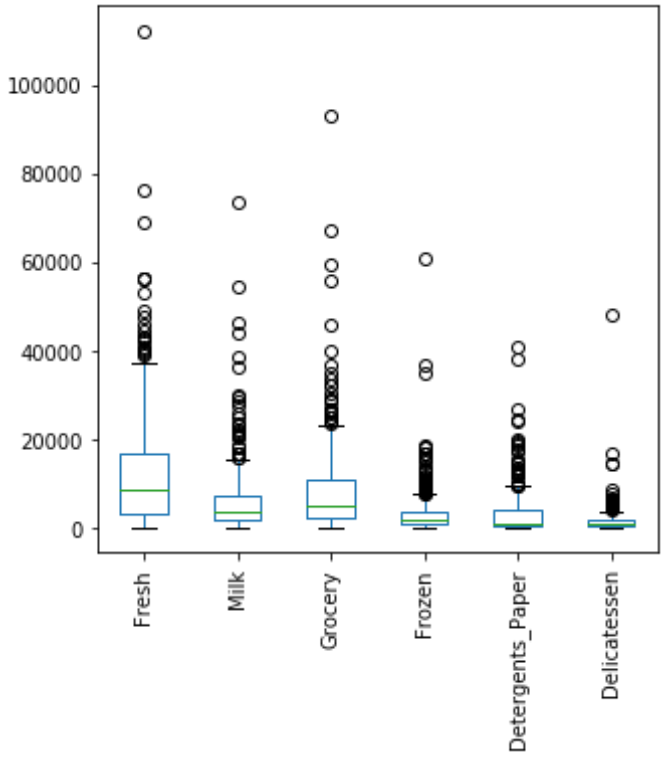
1.4 Are there any outliers in the data?

In [21]:
```
dfnewout = df.drop(['Buyer/Spender'], axis= 1)
dfnewout.head()
```

Out[21]:

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

In [25]:
```
plt.figure (figsize=(5,5))
dfnewout.boxplot(fontsize=None,
    rot=90,
    grid=False);
plt.show()
```



In [24]:
```
dfo = (dfnewout < (Q1 - 1.5 * IQR)) |(dfnewout > (Q3 + 1.5 * IQR))
dfo.head()
```

Out[24]:

	Channel	Delicatessen	Detergents_Paper	Fresh	Frozen	Grocery	Milk	Region
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	True	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	True	False	False	False	False	False	False

In [ ]:
```

```

In [47]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from scipy import stats
```

In [2]:
```
import os
os.chdir('C:\\Users\\WELCOME\\Downloads\\PYTHON FILES\\SMDM\\project')
```

In [64]:
```
df= pd.read_csv('Survey-1.csv')
df.head()
```

Out[64]:

	ID	Gender	Age	Class	Major	Grad Intention	GPA	Employment	Salary	Social Networking	Sat
0	1	Female	20	Junior	Other	Yes	2.9	Full-Time	50.0	1	
1	2	Male	23	Senior	Management	Yes	3.6	Part-Time	25.0	1	
2	3	Male	21	Junior	Other	Yes	2.5	Part-Time	45.0	2	
3	4	Male	21	Junior	CIS	Yes	2.5	Full-Time	40.0	4	
4	5	Male	23	Senior	Other	Undecided	2.8	Unemployed	40.0	2	

In [66]:
```
df.shape
```

Out[66]:
```
(62, 14)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 14 columns):
```

```
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   ID                 62 non-null     int64  
 1   Gender             62 non-null     object 
 2   Age                62 non-null     int64  
 3   Class              62 non-null     object 
 4   Major              62 non-null     object 
 5   Grad Intention     62 non-null     object 
 6   GPA                62 non-null     float64
 7   Employment         62 non-null     object 
 8   Salary             62 non-null     float64
 9   Social Networking  62 non-null     int64  
 10  Satisfaction       62 non-null     int64  
 11  Spending           62 non-null     int64  
 12  Computer           62 non-null     object 
 13  Text Messages      62 non-null     int64  
dtypes: float64(2), int64(6), object(6)
memory usage: 6.9+ KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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 [7]:
```
df.describe(include = 'all').T
```

Out[7]:

	count	unique	top	freq	mean	std	min	25%	50%	75%
ID	62	NaN	NaN	NaN	31.5	18.0416	1	16.25	31.5	46.75
Gender	62	2	Female	33	NaN	NaN	NaN	NaN	NaN	NaN
Age	62	NaN	NaN	NaN	21.129	1.43131	18	20	21	22
Class	62	3	Senior	31	NaN	NaN	NaN	NaN	NaN	NaN
Major	62	8	Retailing/Marketing	14	NaN	NaN	NaN	NaN	NaN	NaN
Grad Intention	62	3	Yes	28	NaN	NaN	NaN	NaN	NaN	NaN
GPA	62	NaN	NaN	NaN	3.12903	0.377388	2.3	2.9	3.15	3.4
Employment	62	3	Part-Time	43	NaN	NaN	NaN	NaN	NaN	NaN
Salary	62	NaN	NaN	NaN	48.5484	12.0809	25	40	50	55
Social Networking	62	NaN	NaN	NaN	1.51613	0.844305	0	1	1	2
Satisfaction	62	NaN	NaN	NaN	3.74194	1.21379	1	3	4	4
Spending	62	NaN	NaN	NaN	482.016	221.954	100	312.5	500	600
Computer	62	3	Laptop	55	NaN	NaN	NaN	NaN	NaN	NaN
Text Messages	62	NaN	NaN	NaN	246.21	214.466	0	100	200	300

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

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

In [8]:
```
pd.crosstab(df.Gender, df.Major)
```

Out[8]:

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

In [9]:
```
pd.crosstab(df.Gender, df['Grad Intention'])
```

Out[9]:

Grad Intention	No	Undecided	Yes
Gender			
Female	9	13	11
Male	3	9	17

In [10]:
```
pd.crosstab(df.Gender, df['Employment'])
```

Out[10]:

Employment	Full-Time	Part-Time	Unemployed
Gender			
Female	3	24	6
Male	7	19	3

In [11]:
```
pd.crosstab(df.Gender, df['Computer'])
```

Out[11]:

Computer	Desktop	Laptop	Tablet
Gender			
Female	2	29	2
Male	3	26	0

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

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

2.2.2. What is the probability that a randomly selected CMSU student will be female?

In [12]:
```
No_of_male = (df['Gender'] =='Male').sum()
print(No_of_male)
No_of_female = (df['Gender'] =='Female').sum()
print(No_of_female)
Total_value =df['Gender'].value_counts().sum()
print(Total_value)
```

```
29
33
62
```

In [13]:
```
p_male = No_of_male/Total_value

print('The probability that a randomly selected CMSU student will be ma
le',(p_male)*100)
```

```
The probability that a randomly selected CMSU student will be male 46.7
74193548387096
```

In [14]:
```
p_female= No_of_female/Total_value

print('The probability that a randomly selected CMSU student will be ma
le',(p_female)*100)
```

```
The probability that a randomly selected CMSU student will be male 53.2
258064516129
```

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.

2.3.2 Find the conditional probability of different majors among the female students of CMSU.

In [48]:
```
pd.crosstab(df['Gender'], df['Major'],margins = True,normalize='index')
```

Out[48]:

Major	Accounting	CIS	Economics/Finance	International Business	Management	Other	Retailing
Gender							
Female	0.090909	0.090909	0.212121	0.121212	0.121212	0.090909	
Male	0.137931	0.034483	0.137931	0.068966	0.206897	0.137931	
All	0.112903	0.064516	0.177419	0.096774	0.161290	0.112903	

In [ ]:
```

```

In [ ]:
```

```

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

2.4.1. Find the probability That a randomly chosen student is a male and intends to graduate.

2.4.2 Find the probability that a randomly selected student is a female and does NOT have a laptop.

In [16]:
```
df['Grad Intention'].value_counts()
```

Out[16]:
```
Yes          28
Undecided    22
No           12
Name: Grad Intention, dtype: int64
```

In [17]:
```
df.groupby(['Gender', 'Grad Intention']).size()
```

Out[17]:
```
Gender  Grad Intention
Female  No                 9
        Undecided         13
        Yes               11
Male    No                 3
        Undecided          9
        Yes               17
dtype: int64
```

In [50]:
```
pd.crosstab(df.Gender, df['Grad Intention'],margins = True,normalize='index')
```

Out[50]:

Grad Intention	No	Undecided	Yes
Gender			
Female	0.272727	0.393939	0.333333
Male	0.103448	0.310345	0.586207
All	0.193548	0.354839	0.451613

In [19]:
```
df['Computer'].value_counts()
```

Out[19]:
```
Laptop     55
Desktop     5
Tablet      2
Name: Computer, dtype: int64
```

In [20]:
```
df.groupby(['Gender', 'Computer']).size()
```

Out[20]:
```
Gender  Computer
Female  Desktop      2
        Laptop      29
        Tablet       2
Male    Desktop      3
        Laptop      26
dtype: int64
```

In [21]:
```
pd.crosstab(df.Gender, df['Computer'],margins = True,normalize='index')
```

Out[21]:

Computer	Desktop	Laptop	Tablet
Gender			
Female	0.060606	0.878788	0.060606
Male	0.103448	0.896552	0.000000
All	0.080645	0.887097	0.032258

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

2.5.1. Find the probability that a randomly chosen student is either a male or has full-time employment?

In [22]:
```
df['Employment'].value_counts()
```

Out[22]:
```
Part-Time     43
Full-Time     10
Unemployed     9
Name: Employment, dtype: int64
```

In [23]:
```
No_of_fulltime_emp = (df['Employment'] =='Full-Time').sum()
No_of_fulltime_emp
```

Out[23]:
```
10
```

In [24]:
```
No_male_fulltime_emp = ((df['Employment'] =='Full-Time') & (df['Gender'
] =='Male')).sum()
No_male_fulltime_emp
```

Out[24]:
```
7
```

In [25]:
```
df.groupby(['Gender', 'Employment']).size()
```

Out[25]:
```
Gender  Employment
Female  Full-Time      3
        Part-Time     24
        Unemployed     6
Male    Full-Time      7
        Part-Time     19
        Unemployed     3
dtype: int64
```

In [26]:
```
p_of_male_stu =No_of_male/Total_value
print(round((p_of_male_stu), 4)*100)
p_of_fulltime_emp =No_of_fulltime_emp/Total_value
print(round((p_of_fulltime_emp), 4)*100)
p_of_male_fulltime_emp = No_male_fulltime_emp/Total_value
print(round((p_of_male_fulltime_emp), 4)*100)
```

```
46.77
16.13
11.29
```

In [27]:
```
p = p_of_male_stu+p_of_fulltime_emp-p_of_male_fulltime_emp
print(' The probability that a randomly chosen student is either a male
 or has full-time employment', p*100 ,'%')
```

```
 The probability that a randomly chosen student is either a male or has
full-time employment 51.61290322580645 %
```

2.5.2. Find the conditional probability that given a female student is randomly chosen, she is majoring in international business or management.

In [53]:
```
(df['Major'].value_counts())
(df.groupby(['Gender', 'Major']).size())
val_1 = (df['Gender']=='Female').sum()
val_2 = ((df['Gender']=='Female') & (df['Major'] == 'International Busi
ness')).sum()
val_3 = ((df['Gender']=='Female') & (df['Major'] == 'Management')).sum
()
```

```
val_4 = val_2 + val_3
val_4

print('Probability that given a female student is randomly chosen, she
 is majoring in international business or management',round((val_4/val_
1)*100, 2), "%")
```

```
Probability that given a female student is randomly chosen, she is majo
ring in international business or management 24.24 %
```

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 the graduate intention and being female are independent events?

In [29]:
```
df[df['Grad Intention'] == 'Undecided'].index
df_cont = df.drop(df[df['Grad Intention'] == 'Undecided'].index)
df_cont.head()
```

Out[29]:

	ID	Gender	Age	Class	Major	Grad Intention	GPA	Employment	Salary	Social Networking	Satis
0	1	Female	20	Junior	Other	Yes	2.9	Full-Time	50.0	1	
1	2	Male	23	Senior	Management	Yes	3.6	Part-Time	25.0	1	
2	3	Male	21	Junior	Other	Yes	2.5	Part-Time	45.0	2	
3	4	Male	21	Junior	CIS	Yes	2.5	Full-Time	40.0	4	
8	9	Female	20	Junior	Management	Yes	3.6	Unemployed	30.0	0	

In [55]:
```
pd.crosstab(df['Gender'] , df_cont['Grad Intention'] ,margins =True)
```

Out[55]:

Grad Intention	No	Yes	All
Gender			
Female	9	11	20

Grad Intention	No	Yes	All
Gender			
Male	3	17	20
All	12	28	40

In [61]:
```
pd.crosstab(df['Gender'] , df_cont['Grad Intention'] ,margins =True,normalize='index')
```

Out[61]:

Grad Intention	No	Yes
Gender		
Female	0.45	0.55
Male	0.15	0.85
All	0.30	0.70

In [58]:
```
 print('The probability that a randomly selected Student is Female',(20/40)*100)
```

```
The probability that a randomly selected Student is Female 50.0
```

In [59]:
```
val2 = ((df['Gender']=='Female') & (df_cont['Grad Intention'] == 'Yes')).sum()
print('The probability that a randomly selected student is female and intends to graduate',(val2/20)*100, '%' )
print('They are not independent events')
```

```
The probability that a randomly selected student is female and intends
to graduate 55.00000000000001 %
They are not independent events
```

In [ ]:
```

```

2.7. Note that there are four numerical (continuous) variables in the data set, GPA, Salary,

Spending, and Text Messages.

2.6.1. If a student is chosen randomly, what is the probability that his/her GPA is less than 3?

In [34]:
```
No_of_stud_less = (df['GPA'] < 3).sum()
print(No_of_stud_less)
print(Total_value)
```

```
17
62
```

In [35]:
```
No_of_stud_less = (df['GPA'] < 3).sum()
p_of_stud_less =No_of_stud_less/Total_value
print('The probability that his/her GPA is less than 3 is', (p_of_stud_
less)*100, '%')
```

```
The probability that his/her GPA is less than 3 is 27.419354838709676 %
```

In [ ]:
```

```

Find the conditional probability that a randomly selected male earns 50 or more. Find the conditional probability that a randomly selected female earns 50 or more.

In [36]:
```
((df['Gender']=='Male') & (df['Salary'] >= 50)).sum()
```

Out[36]:
```
14
```

In [37]:
```
pd.crosstab((df['Gender']=='Male'), (df['Salary'] >= 50), normalize='in
dex')
```

Out[37]:

Salary	False	True
Gender		
False	0.454545	0.545455
True	0.517241	0.482759

In [38]:
```
((df['Gender']=='Female') & (df['Salary'] >= 50)).sum()
```

Out[38]: 18

In [63]:
```
pd.crosstab((df['Gender']=='Female'), (df['Salary'] >= 50), normalize=
'index')
```

Out[63]:

Salary	False	True
Gender		
False	0.517241	0.482759
True	0.454545	0.545455

2.8. 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. Write a note summarizing your conclusions.

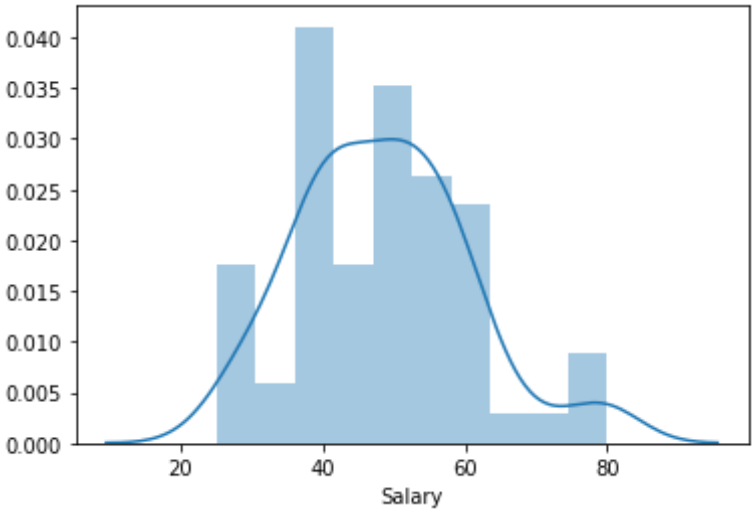
In [ ]:

In [ ]:

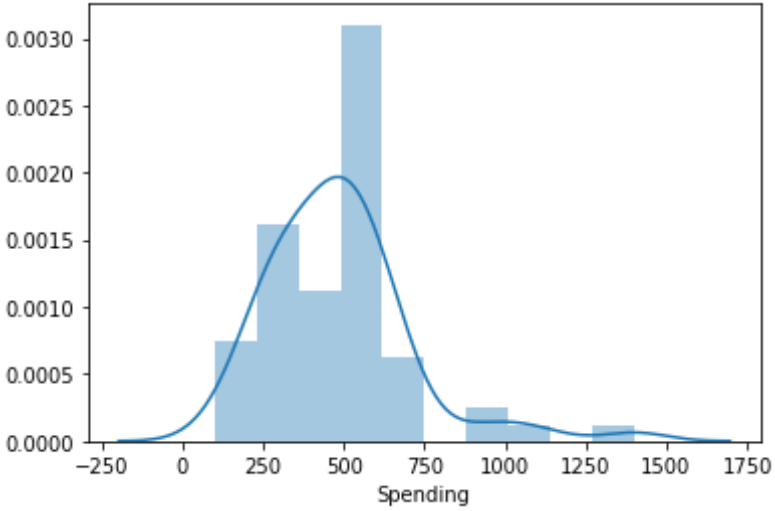
In [ ]:

In [ ]:

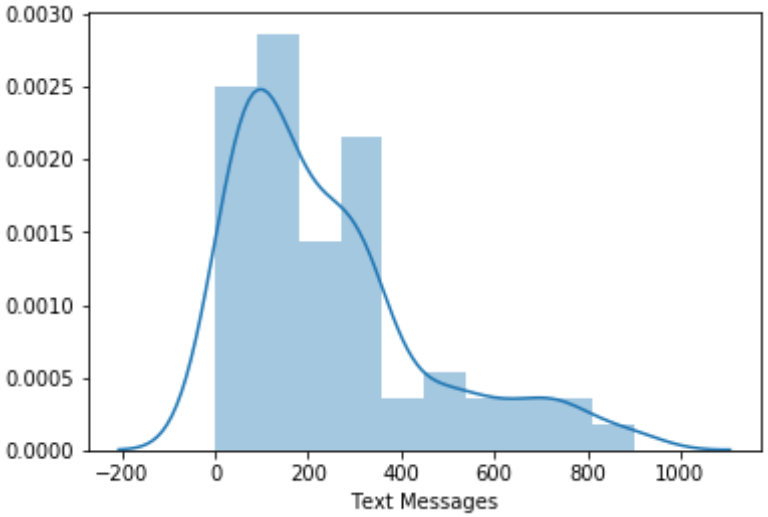
In [42]:
```
sns.distplot(df['Salary'], bins = 10, kde=True, rug= False);
```



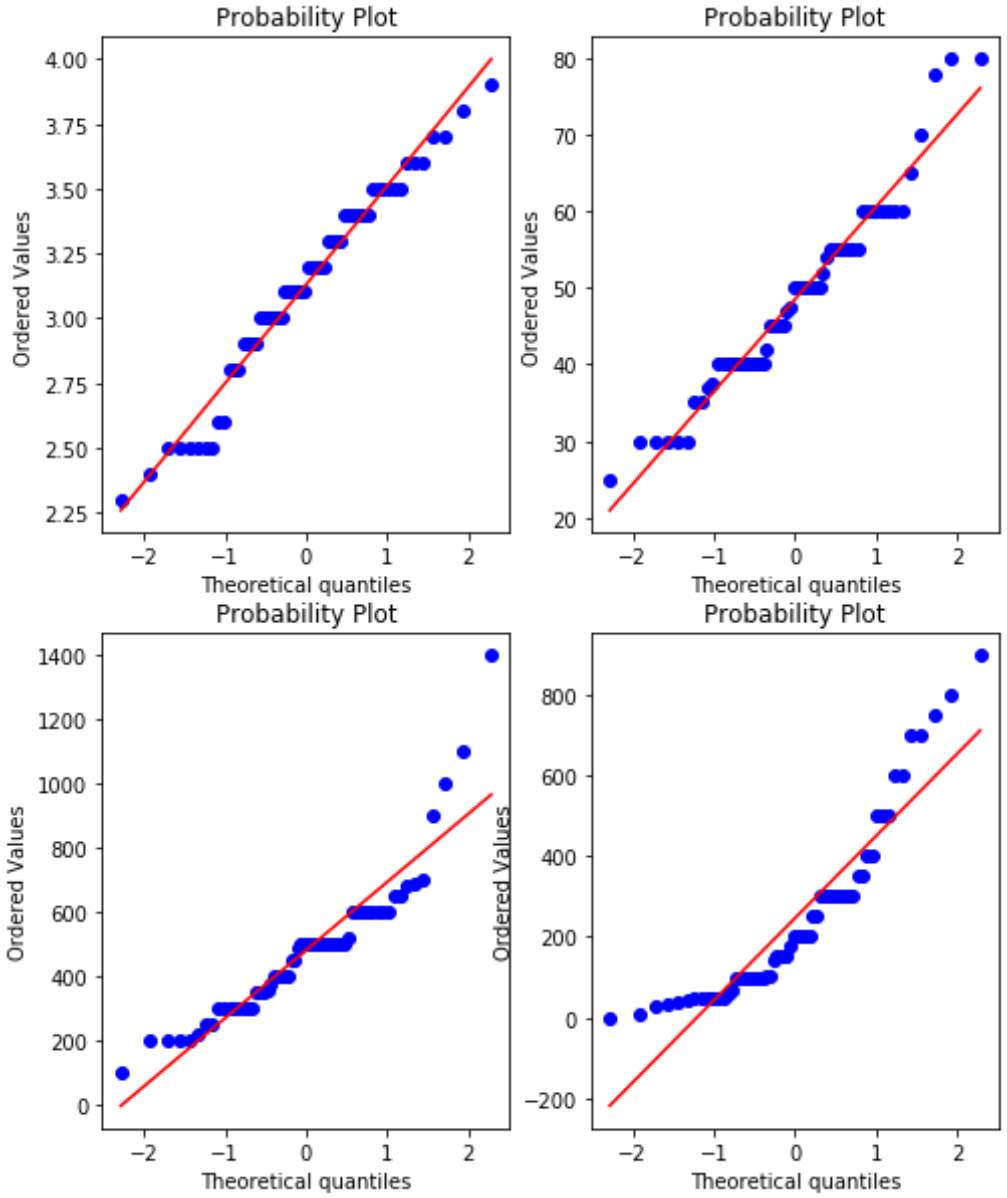
In [43]:
```
sns.distplot(df['Spending'], bins = 10, kde=True, rug= False);
```



In [44]:
```
sns.distplot(df['Text Messages'], bins = 10, kde=True, rug= False);
```



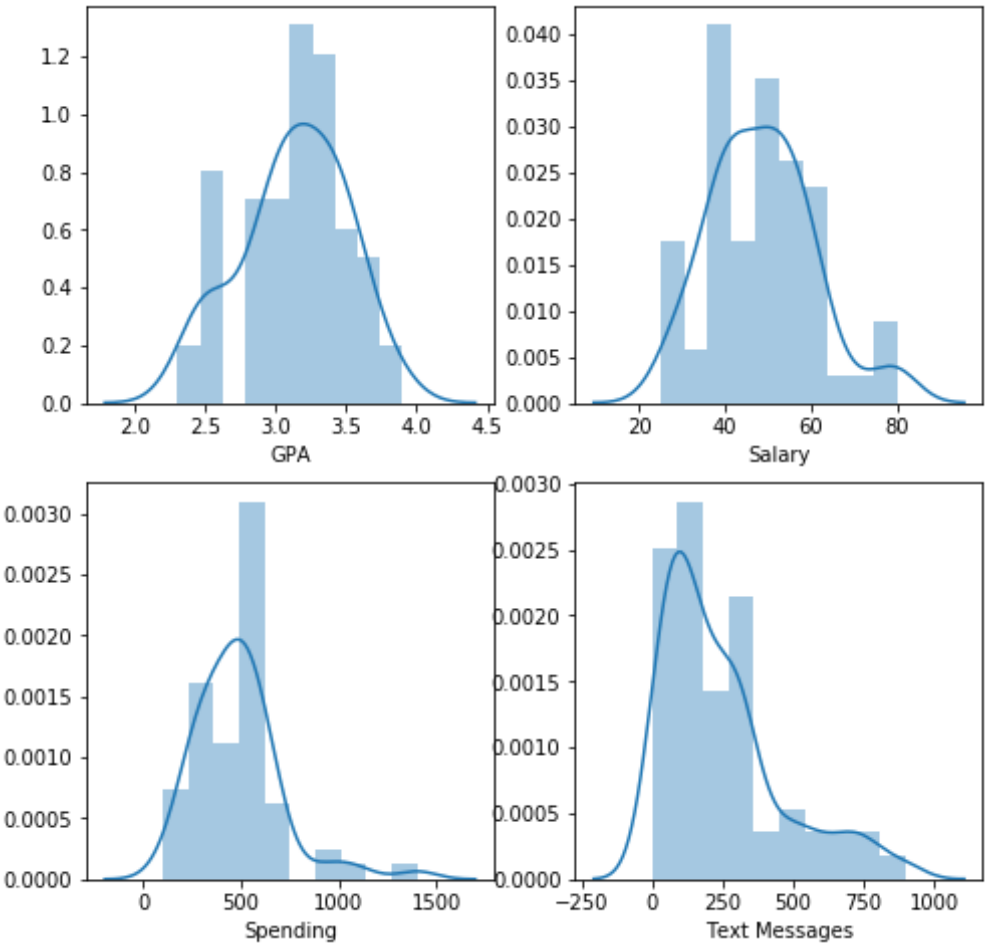
In [45]:
```
plt.figure(figsize=[8, 10])

plt.subplot(2, 2, 1)
stats.probplot(df['GPA'], plot=plt);


plt.subplot(2, 2, 2)
stats.probplot(df['Salary'], plot=plt);



plt.subplot(2, 2, 3)
stats.probplot(df['Spending'], plot=plt);



plt.subplot(2, 2, 4)
stats.probplot(df['Text Messages'], plot=plt);

plt.show()
```



In [46]:
```
plt.figure(figsize=[8, 8])
```

```
plt.subplot(2, 2, 1)
sns.distplot(df['GPA'], bins = 10, kde=True, rug= False);
print('skew value of GPA is', df['GPA'].skew())

plt.subplot(2, 2, 2)
sns.distplot(df['Salary'], bins = 10, kde=True, rug= False);
print('skew value of Salary is', df['Salary'].skew())


plt.subplot(2, 2, 3)
sns.distplot(df['Spending'], bins = 10, kde=True, rug= False);
print('skew value of Spending is', df['Spending'].skew())


plt.subplot(2, 2, 4)
sns.distplot(df['Text Messages'], bins = 10, kde=True, rug= False);
print('skew value of Text Message is', df['Text Messages'].skew())

plt.show()
```

```
skew value of GPA is -0.3146000894506981
skew value of Salary is 0.5347008436225946
skew value of Spending is 1.5859147414045331
skew value of Text Message is 1.2958079731054333
```



In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import scipy.stats as stats
from   scipy.stats               import ttest_1samp, ttest_ind
import statsmodels.stats.api as sm
```

In [2]:
```
import os
os.chdir('C:\\Users\\WELCOME\\Downloads\\PYTHON FILES\\SMDM\\project')
```

In [3]:
```
df= pd.read_csv('A & B shingles-1.csv')
df.head()
```

Out[3]:

	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 [4]:
```
df.isnull().sum()
```

Out[4]:
```
A    0
B    5
dtype: int64
```

In [32]:
```
df.shape
```

Out[32]:
```
(36, 2)
```

In [33]:
```
df.describe().T
```

Out[33]:

	count	mean	std	min	25%	50%	75%	max
A	36.0	0.316667	0.135731	0.13	0.2075	0.29	0.3925	0.72
B	31.0	0.273548	0.137296	0.10	0.1600	0.23	0.4000	0.58

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 is 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.

3.1 Do you think there is evidence that means moisture contents in both types of shingles are within the permissible limits? State your conclusions clearly showing all steps.

Define Null and alternate hypothesis for sample A

step 1:

Testing whether the moisture content is less the permissible limit

The null hypothesis states that the moisture content of sample A is greater or than equal to the

permissible limit, $\mu \geq 0.35$

The alternative hypothesis states that the moisture content of sample A is less than permissible limit, $\mu < 0.35$

$H0 : \mu \geq 0.35$

$HA : \mu < 0.35$

Step 2: Decide the significance level

Here we select $\alpha = 0.05$ as given in the question.

Step 3: Identify the test statistic¶

We have two samples (A and B) and we do not know the population standard deviation. Sample sizes for both samples are not the same. The sample size is , n > 30. So we use the t distribution and the $tSTAT$ test statistic for one sample test for A sample. One tail test for sample A

Step 4: Calculate the p - value and test statistic

In [30]:
```
t_statistic, p_value = ttest_1samp(df['A'],0.35, nan_policy='omit')
print('tstat',t_statistic)
print('P Value',p_value/2)
```

```
tstat -1.4735046253382782
P Value 0.07477633144907513
```

Out[30]:
```
False
```

Step 5: Decide to reject or accept null hypothesis

In [27]:
```
print ("one-sample t-test p-value=", p_value/2)

alpha_level = 0.05

if (p_value/2) < alpha_level:
    print('We have enough evidence to reject the null hypothesis in fav
our of alternative hypothesis')

else:
    print('We do not have enough evidence to reject the null hypothesis
 in favour of alternative hypothesis')
    print('We conclude that the moisture content is greater than permis
sible limit in sample A.')
```

```
one-sample t-test p-value= 0.07477633144907513
We do not have enough evidence to reject the null hypothesis in favour
of alternative hypothesis
We conclude that the moisture content is greater than permissible limit
in sample A.
```

Define Null and alternate hypothesis for sample B

step 1:

Testing whether the moisture content is less the permissible limit

The null hypothesis states that the moisture content of sample B is greater or than equal to the permissible limit, $\mu \geq 0.35$

The alternative hypothesis states that the moisture content of sample B is less than permissible limit, $\mu < 0.35$

$H0 : \mu \geq 0.35$

$HA : \mu < 0.35$

Step 2: Decide the significance level

Here we select α = 0.05 as given in the question.

Step 3: Identify the test statistic¶

We have two samples (A and B) and we do not know the population standard deviation. Sample sizes for both samples are not the same. The sample size is , n > 30. So we use the t distribution and the $tSTAT$ test statistic for one sample test for B sample. one tail test for Sample B

Step 4: Calculate the p - value and test statistic

In [28]:
```
t_statistic, p_value = ttest_1samp(df['B'],0.35, nan_policy='omit')
print('tstat',t_statistic)
print('P Value',p_value/2)
```

```
tstat -3.1003313069986995
P Value 0.0020904774003191826
```

Step 5: Decide to reject or accept null hypothesis

In [29]:
```
print ("one-sample t-test p-value=", p_value/2)

alpha_level = 0.05

if (p_value/2) < alpha_level:
    print('We have enough evidence to reject the null hypothesis in fav
our of alternative hypothesis')
    print('We conclude that the moisture content is less than permissib
```

```
le limit in sample B.')

else:
    print('We do not have enough evidence to reject the null hypothesis
 in favour of alternative hypothesis')
```

```
one-sample t-test p-value= 0.0020904774003191826
We have enough evidence to reject the null hypothesis in favour of alte
rnative hypothesis
We conclude that the moisture content is less than permissible limit in
sample B.
```

In [ ]:

In [ ]:

In [ ]:

3.2 Do you think that the population mean 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?

step 1:

Define Null and alternate hypothesis

In testing whether the mean for shingles A and Shingles B are the same, the null hypothesis states that the mean of shingle A to mean of shingle B are the same, μA equals μB. The alternative hypothesis states that the mean are different, μA is not equal to μB

- H_0: μA - $\mu B \neq 0$ i.e $\mu A \neq \mu B$

- H_A: μA - μB = 0 i.e μA = μB

Step 2: Decide the significance level

Here we select α = 0.05 and the population standard deviation is not known.

Step 3: Identify the test statistic

We have two samples and we do not know the population standard deviation. Sample sizes for both samples are not the same. The sample size is , n > 30. So we use the t distribution and the $tSTAT$ test statistic for two sample test.

Step 4: Calculate the p - value and test statistic

In [7]:
```
t_statistic, p_value  = ttest_ind(df['A'],df['B'],nan_policy='omit')
print('tstat',t_statistic)
print('P Value',p_value)
```

```
tstat 1.2896282719661123
P Value 0.2017496571835306
```

Step 5: Decide to reject or accept null hypothesis

In [8]:
```
print ("two-sample t-test p-value=", p_value)

alpha_level = 0.05

if p_value < alpha_level:
    print('We have enough evidence to reject the null hypothesis in fav
our of alternative hypothesis')
```

```
else:
    print('We do not have enough evidence to reject the null hypothesis
 in favour of alternative hypothesis')
    print('We conclude that mean for shingles A and singles B are not t
he same')
```

```
two-sample t-test p-value= 0.2017496571835306
We do not have enough evidence to reject the null hypothesis in favour
of alternative hypothesis
We conclude that mean for shingles A and singles B are not the same
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: