In [11]:
```
# import libraries
import numpy as np
import pandas as pd
import scipy.stats as stats
from ipykernel import kernelapp as app
import matplotlib.pyplot as plt
```

Problem 1

In [12]:
```
# Create DataFrame from the given Data
lst_qualification = ['High School','Bachelors','Masters','PHD']
lst_female = [60,54,46,41]
lst_male = [40,44,53,57]
df=pd.DataFrame({'Qualification':lst_qualification,'Count_F': lst_female ,'Count_M': l
```

Out[12]:

	Qualification	Count_F	Count_M
0	High School	60	40
1	Bachelors	54	44
2	Masters	46	53
3	PHD	41	57

In [13]:
```
##USING Z SCORE AND p VALUE

# Add column in the Dataframe for Mean, Standard Deviation, Z Score
# and P Values for Female(F) and Male (M)

df['Mean_F']=df['Count_F'].mean()
df['Mean_M']=df['Count_M'].mean()

df['Std_Dev_F']=df['Count_F'].std()
df['Std_Dev_M']=df['Count_M'].std()

df['Z_F']=stats.zscore(df['Count_F'])
df['Z_M']=stats.zscore(df['Count_M'])

df['p_F']=[stats.norm.cdf(pval) for pval in stats.zscore(df['Count_F'])]
df['p_M']=[stats.norm.cdf(pval) for pval in stats.zscore(df['Count_M'])]
df.head()
```

Out[13]:

	Qualification	Count_F	Count_M	Mean_F	Mean_M	Std_Dev_F	Std_Dev_M	Z_F	Z_M	p_F	
0	High School	60	40	50.25	48.5	8.421203	7.852813	1.336903	-1.249865	0.909373	0.
1	Bachelors	54	44	50.25	48.5	8.421203	7.852813	0.514193	-0.661693	0.696442	0.
2	Masters	46	53	50.25	48.5	8.421203	7.852813	-0.582752	0.661693	0.280030	0.
3	PHD	41	57	50.25	48.5	8.421203	7.852813	-1.268344	1.249865	0.102338	0.

In [14]:
```
print('Conclutions from the above table: \npvalue of Male and Female (more than 5%, th

print('Female populations is more at High School and Bachelors')
print('Female populations is less at Masters and PHD\n')

print('Male populations is less at High School and Bachelors')
print('Male populations is more at Masters and PHD')
```

```
Conclutions from the above table:
pvalue of Male and Female (more than 5%, there is a relationship
between the gender of an individual and the level of education that they have obta
ined.

Female populations is more at High School and Bachelors
Female populations is less at Masters and PHD

Male populations is less at High School and Bachelors
Male populations is more at Masters and PHD
```

In [15]:
```
##Using Chi-square test

# redefine the dataset
df=df[['Qualification','Count_F','Count_M']]

N = 395          # Sample Size
df['Count_Total']=df.Count_F+df.Count_M

# Expected frequency = ((row total×column)/total sample size
df['ef_F']=(df.Count_F.sum()*df.Count_Total)/N
df['ef_M']=df.Count_Total-df.ef_F

# Chi Sqaure value χ2=∑(Observe freq-Expected Freq)2/Expected Freq
df['chi_F']=[(math.pow((df.Count_F.values[i]-df.ef_F.values[i]),2))/df.ef_F.values[i]
df['chi_M']=[(math.pow((df.Count_M.values[i]-df.ef_M.values[i]),2))/df.ef_M.values[i]
df
```

```
C:\Users\HP\Anaconda3\lib\site-packages\ipykernel_launcher.py:7: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy)
  import sys
C:\Users\HP\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy)
  # Remove the CWD from sys.path while we load stuff.
C:\Users\HP\Anaconda3\lib\site-packages\ipykernel_launcher.py:11: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy)
  # This is added back by InteractiveShellApp.init_path()
C:\Users\HP\Anaconda3\lib\site-packages\ipykernel_launcher.py:14: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy)

C:\Users\HP\Anaconda3\lib\site-packages\ipykernel_launcher.py:15: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy)
  from ipykernel import kernelapp as app
```

In [17]:
```
chi_sq_stat =df.chi_F.sum() + df.chi_M.sum()
print("Chi-Square Test Statstic value:\t", chi_sq_stat)
dof = 3       # Degree of Freedom - here dof =3

# Calculate P value from chi_square_stat and degree of freedom using cdf function
p_val = 1 - stats.chi2.cdf(chi_sq_stat,dof)
print("Chi-Square P value\t\t", p_val)

α =0.05  # significance level, confidence level 95%

#Calculate chi-square crtical value
chi_critical= stats.chi2.ppf(0.95,dof)
print("Chi-Square Test Critical value:\t", chi_critical)

print('\nAs Chi-Square Test Statstic value (8.006) greater than Chi-Square Test Critic
```

```
Chi-Square Test Statstic value:  8.006066246262538
Chi-Square P value               0.04588650089174717
Chi-Square Test Critical value:  7.814727903251179

As Chi-Square Test Statstic value (8.006) greater than Chi-Square Test Critical va
lue (7.815)
by Null hypothesis, it can be concluded Education level depends on gender (at 5% s
ignificance level)
```

In [20]:
```
# Create DataFrame from the given Data
lst_group1 = [51, 45, 33, 45, 67]
lst_group2 = [23, 43, 23, 43, 45]
lst_group3 = [56, 76, 74, 87, 56]
df=pd.DataFrame({'Group1':lst_group1,'Group2': lst_group2 ,'Group3': lst_group3})
```

Out[20]:

	Group1	Group2	Group3
0	51	23	56
1	45	43	76
2	33	23	74
3	45	43	87
4	67	45	56

In [22]:
```
p_Val=stats.f_oneway(df['Group1'],df['Group2'],df['Group3']).pvalue
F_Val=stats.f_oneway(df['Group1'],df['Group2'],df['Group3']).statistic

α = 0.05                    # Significance level, confidence level 95%

print('Null Hypothesis: \t Group1=Group2=Group3')

print('\nHypothesis testing with 5% significance')

print('\nHere p Value greater than α , so Null Hypothesis(Group1=Group2=Group3) can be

print('\nWriting up the results in APA format:')

print('\t Significance level:\t', round(α,4))
print('\t F Value:\t\t', round(F_Val,4))
print('\t p Value:\t\t', round(p_Val,4), ' <', round(α,4) , '(Significance level)' )
```

```
Null Hypothesis:         Group1=Group2=Group3

Hypothesis testing with 5% significance

Here p Value greater than α , so Null Hypothesis(Group1=Group2=Group3) can be Acce
pted.

Writing up the results in APA format:
         Significance level:     0.05
         F Value:                9.7472
         p Value:                0.0031  < 0.05 (Significance level)
         So, Accept Null Hypothesis:     Group1=Group2=Group3
```

Problem 3

In [24]:
```
# Create DataFrame from the given Data
lst_group1 = [10,20,30,40,50]
lst_group2 = [5,10,15, 20, 25]

df=pd.DataFrame({'Group1':lst_group1,'Group2': lst_group2})
```

Out[24]:

	Group1	Group2
0	10	5
1	20	10
2	30	15
3	40	20
4	50	25

In [26]:
```
# Add column in the Dataframe for Mean, Standard Deviation and Variance

df['Mean_Group1']=df['Group1'].mean()
df['Mean_Group2']=df['Group2'].mean()

df['Std_Dev_Group1']=df['Group1'].std()
df['Std_Dev_Group2']=df['Group2'].std()

df['Var_Group1']=df['Group1'].var()
df['Var_Group2']=df['Group2'].var()
```

Out[26]:

	Group1	Group2	Mean_Group1	Mean_Group2	Std_Dev_Group1	Std_Dev_Group2	Var_Group1	Var_Group2
0	10	5	30.0	15.0	15.811388	7.905694	250.0	62.5
1	20	10	30.0	15.0	15.811388	7.905694	250.0	62.5
2	30	15	30.0	15.0	15.811388	7.905694	250.0	62.5
3	40	20	30.0	15.0	15.811388	7.905694	250.0	62.5
4	50	25	30.0	15.0	15.811388	7.905694	250.0	62.5

In [27]:
```
# Calculate the P Values
# Hypothesis Test
print('Null Hypothesis Group1 = Group2')

α =0.05  # significance level, confidence level 95%
print('\nSignificance level:\t', round(α,4))

# F test
# F-Test Formula:\t (Varience of Group 1)/(Varience of Group 1)
F_Val=df['Group1'].var()/df['Group2'].var()
print('F Test Results:\t\t',F_Val)

p_Val = stats.f.cdf(F_Val, len(df['Group1'])-1,len(df['Group1'])-1)

print('p Values is:\t\t',p_Val)

print('\nHere:\t p Value:\t', round(p_Val,4), ' >', round(α,4) , '(Significance level)')
print('\t So, Reject Null Hypothesis: \t Group1=Group2' )
```

```
Null Hypothesis Group1 = Group2

Significance level:      0.05
F Test Results:          4.0
p Values is:             0.896

Here:    p Value:        0.896  > 0.05 (Significance level)
         So, Reject Null Hypothesis:     Group1=Group2
```