Chapter 1

Exercise 1.12

For polynomial of degree 3, the std of the error for the training set and the validation set are 31,988.52 and 35,587.52, respectively. As For polynomial of degree 4, the std of the error for the training set and the validation set are 21,824.15 and 37,426.63, respectively. As polynomial increases beyond 2, the higher the accuracy for the training set but errors for the validation set increase. Therefore, the above models do not generalize well from the training set to the validation set. see uploaded excel for result

Exercise 1.13

P(Spam) = 0.25, P(Word|Spam) = 0.4, and P(Word) = 0.125 and using Bayes' theorem P(Spam|Word)=P(Word|Spam)P(Spam)}/{P(Word)} = 0.4 * 0.25 /0.125 = 0.8 There is an 80% chance that an email containing the word is spam.

Set up for CH2 question

In [1]:

```
#2.13
# loading packages

import os

import pandas as pd
import numpy as np

# plotting packages
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as clrs

# Kmeans algorithm from scikit-learn
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
```

Load raw data

In [2]:

```
# load raw data
DATA_FOLDER = './'
raw = pd.read_csv(os.path.join(DATA_FOLDER, 'country risk 2019 data.csv'))

# check the raw data
print("Size of the dataset (row, col): ", raw.shape)
print("\nFirst 5 rows\n", raw.head(n=5))
```

```
Size of the dataset (row, col):  (121, 6)

First 5 rows
      Country Abbrev  Corruption  Peace  Legal  GDP Growth
0     Albania     AL          35  1.821  4.546       2.983
1     Algeria     DZ          35  2.219  4.435       2.553
2   Argentina     AR          45  1.989  5.087      -3.061
3     Armenia     AM          42  2.294  4.812       6.000
4   Australia     AU          77  1.419  8.363       1.713
```

Simple exploratory analysis

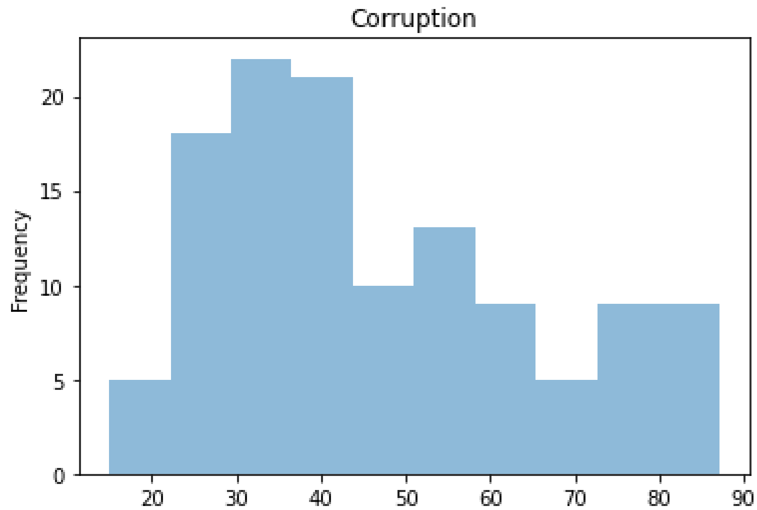
Print summary statistics

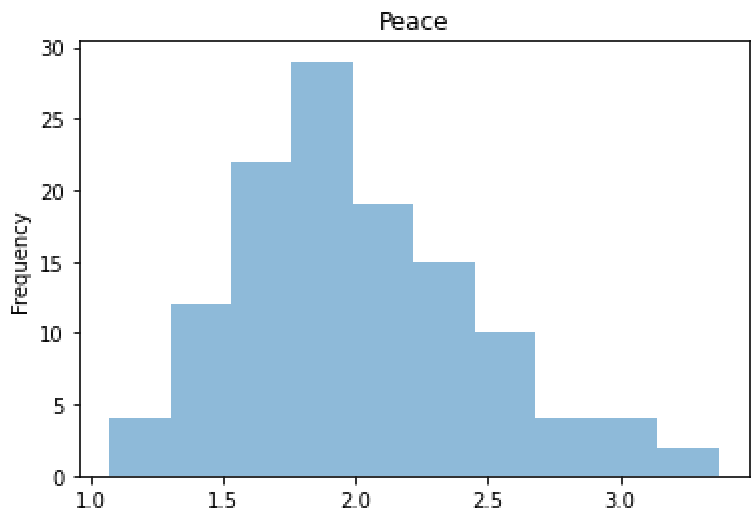
In [3]:
```
# print summary statistics
print("\nSummary statistics\n", raw.describe())
print("\nCorrelation matrix\n", raw.corr())
```

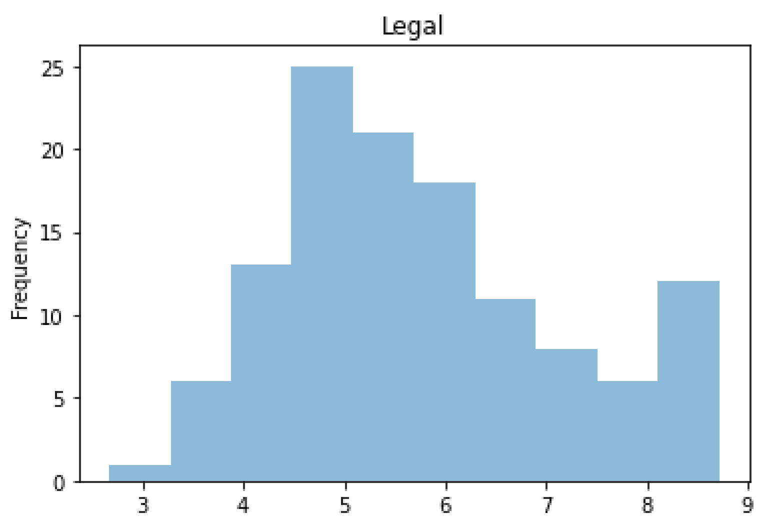
```
Summary statistics
        Corruption       Peace       Legal  GDP Growth
count  121.000000  121.000000  121.000000  121.000000
mean    46.842975    2.001017    5.752529    2.657529
std     18.702499    0.461485    1.373932    2.563741
min     15.000000    1.072000    2.671000   -9.459000
25%     33.000000    1.699000    4.785000    1.249000
50%     41.000000    1.939000    5.455000    2.600000
75%     60.000000    2.294000    6.488000    4.000000
max     87.000000    3.369000    8.712000    7.800000

Correlation matrix
              Corruption     Peace     Legal  GDP Growth
Corruption    1.000000 -0.705002  0.938512   -0.123545
Peace        -0.705002  1.000000 -0.662233   -0.004428
Legal         0.938512 -0.662233  1.000000   -0.150369
GDP Growth   -0.123545 -0.004428 -0.150369    1.000000
```

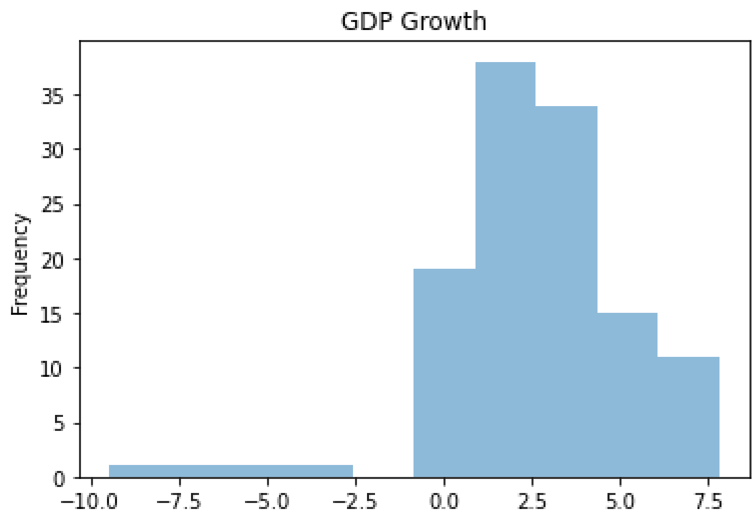
Plot histogram

In [4]:
```
# plot histograms
plt.figure(1)
raw['Corruption'].plot(kind = 'hist', title = 'Corruption', alpha = 0.5)

plt.figure(2)
raw['Peace'].plot(kind = 'hist', title = 'Peace', alpha = 0.5)

plt.figure(3)
raw['Legal'].plot(kind = 'hist', title = 'Legal', alpha = 0.5)

plt.figure(4)
raw['GDP Growth'].plot(kind = 'hist', title = 'GDP Growth', alpha = 0.5)

plt.show()
```


Corruption
Frequency
0
5
10
15
20
20
30
40
50
60
70
80
90


Peace
Frequency
0
5
10
15
20
25
30
1.0
1.5
2.0
2.5
3.0


Legal
Frequency
0
5
10
15
20
25
3
4
5
6
7
8
9



K means cluster

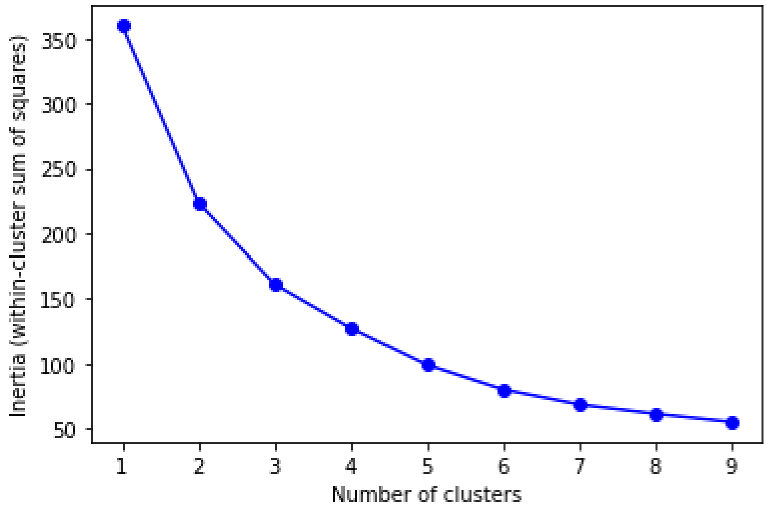
In [5]:
```
#3 feature
X_3 = raw[['Peace', 'Legal', 'GDP Growth']]
X_3 = (X_3 - X_3.mean()) / X_3.std()
print(X_3.head(5))
```

```
      Peace     Legal  GDP Growth
0 -0.390081 -0.878158    0.126952
1  0.472352 -0.958948   -0.040772
2 -0.026039 -0.484397   -2.230541
3  0.634871 -0.684553    1.303747
4 -1.261182  1.900001   -0.368418
```

Perform elbow method

In [6]:
```
# https://stackoverflow.com/questions/41540751/sklearn-kmeans-equivalent-of-elbow-meth

Ks = range(1, 10)
inertia = [KMeans(i).fit(X_3).inertia_ for i in Ks]

fig = plt.figure()
plt.plot(Ks, inertia, '-bo')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia (within-cluster sum of squares)')
plt.show()
```

```
C:\Users\Junho\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: UserWarni
ng: KMeans is known to have a memory leak on Windows with MKL, when there are less ch
unks than available threads. You can avoid it by setting the environment variable OMP
_NUM_THREADS=1.
  warnings.warn(
```



Exercise 2.13 testing n_init

In [7]:
```
k = 3

#2.13 testing n_init with different number

#n_init=2
kmeans2 = KMeans(n_clusters=k,n_init=2, random_state=1)
kmeans2.fit(X_3)
y2=kmeans2.labels_
print("\n inertia for n_init=2 :", kmeans2.inertia_)
print(" cluster centers: \n", kmeans2.cluster_centers_)


#n_init=10
kmeans = KMeans(n_clusters=k, random_state=1)
kmeans.fit(X_3)
y = kmeans.labels_
print("\n inertia for n_init=10:", kmeans.inertia_)
print("cluster centers: ", kmeans.cluster_centers_)
print("cluster labels 10: \n", y)


#n_init=20
kmeans20 = KMeans(n_clusters=k,n_init=20, random_state=1)
kmeans20.fit(X_3)
y20=kmeans20.labels_
print("\n inertia for n_init=20 ", kmeans20.inertia_)
print("cluster centers: \n", kmeans20.cluster_centers_)


#n_init=30
kmeans30 = KMeans(n_clusters=k,n_init=30, random_state=1)
```

```
kmeans30.fit(X_3)
y30=kmeans30.labels_
print("\n inertia for n_init=30:", kmeans30.inertia_)
print("cluster centers: \n", kmeans30.cluster_centers_)


#n_init=50
kmeans50 = KMeans(n_clusters=k,n_init=50, random_state=1)
kmeans50.fit(X_3)
y50=kmeans50.labels_
print("\n inertia for n_init=50:", kmeans50.inertia_)
print(" cluster centers: \n", kmeans50.cluster_centers_)
```

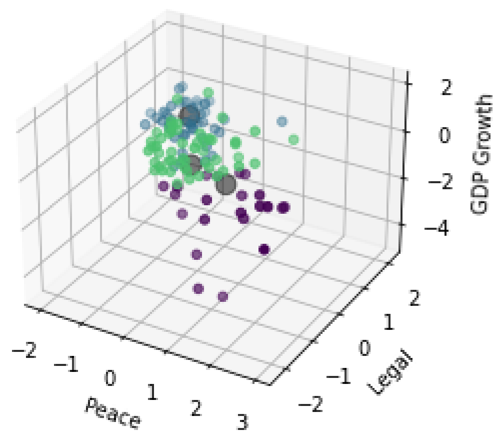
```
 inertia for n_init=2 : 169.24242908631018
 cluster centers:
 [[ 0.53110654 -0.61456608  0.34774502]
 [-0.85103491  0.99692377 -0.22524313]
 [ 0.70529573 -0.95894794 -3.43893096]]

 inertia for n_init=10: 161.1333871005255
cluster centers:  [[ 1.22506036 -0.83385901 -1.07842464]
 [-0.85097477  1.02149992 -0.23897931]
 [ 0.23006626 -0.54045468  0.65506397]]
cluster labels 10:
 [2 2 0 2 1 1 2 2 2 1 2 2 2 1 0 2 0 2 1 0 1 2 2 1 2 1 1 0 1 2 0 2 2 1 2 1 1
 2 2 1 2 2 2 2 1 1 2 2 0 1 2 1 1 1 1 2 2 1 1 1 0 0 1 2 2 1 2 2 1 0 2 2 2 2
 2 1 1 0 0 1 1 0 2 0 2 2 1 1 1 1 0 2 0 2 2 2 1 1 1 0 1 2 1 1 1 2 2 2 0 2 0
 2 0 1 1 1 1 2 0 2 0]

 inertia for n_init=20  161.1333871005255
cluster centers:
 [[ 1.22506036 -0.83385901 -1.07842464]
 [-0.85097477  1.02149992 -0.23897931]
 [ 0.23006626 -0.54045468  0.65506397]]

 inertia for n_init=30: 161.1333871005255
cluster centers:
 [[ 1.22506036 -0.83385901 -1.07842464]
 [-0.85097477  1.02149992 -0.23897931]
 [ 0.23006626 -0.54045468  0.65506397]]

 inertia for n_init=50: 161.1333871005255
 cluster centers:
 [[ 1.22506036 -0.83385901 -1.07842464]
 [-0.85097477  1.02149992 -0.23897931]
 [ 0.23006626 -0.54045468  0.65506397]]
```

Visualize the result (3D plot)

In [8]:

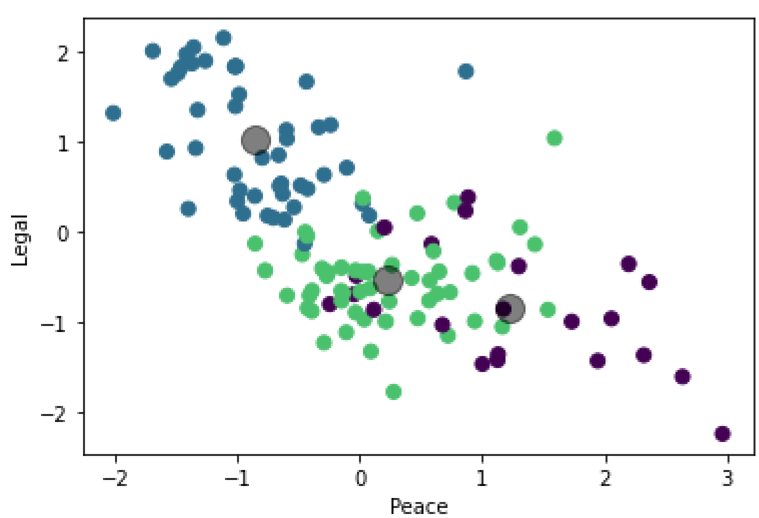
```
# set up the color
norm = clrs.Normalize(vmin=0.,vmax=y.max() + 0.8)
cmap = cm.viridis

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(X_3.iloc[:,0], X_3.iloc[:,1], X_3.iloc[:,2], c=cmap(norm(y)), marker='o')

centers = kmeans.cluster_centers_
```

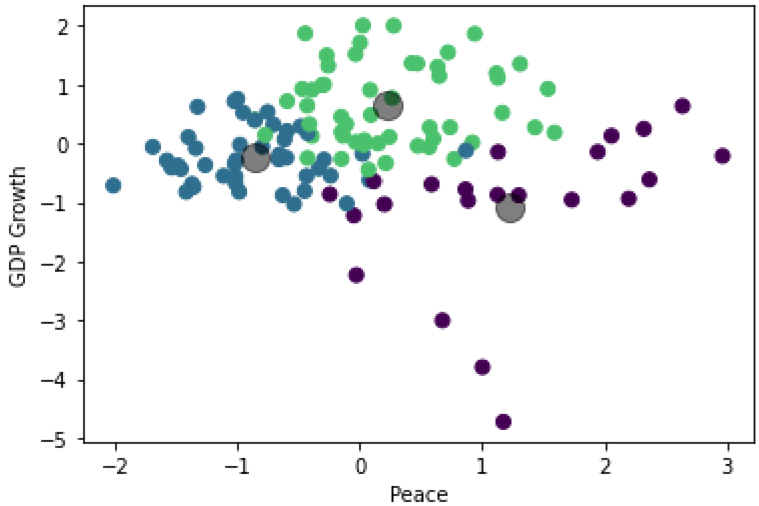
```
ax.scatter(centers[:, 0], centers[:, 1], c='black', s=100, alpha=0.5)

ax.set_xlabel('Peace')
ax.set_ylabel('Legal')
ax.set_zlabel('GDP Growth')

plt.show()
```

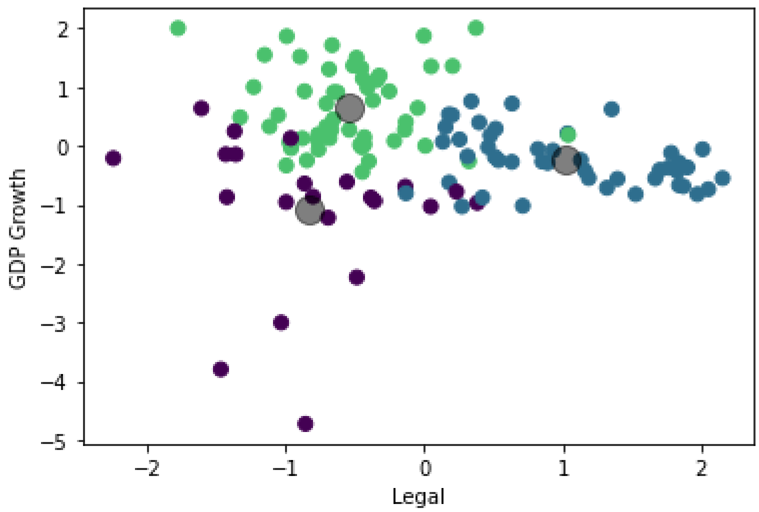


Visualize the result (3 2D plots)

In [9]:
```
%matplotlib inline
import matplotlib.pyplot as plt

figs = [(0, 1), (0, 2), (1, 2)]
labels = ['Peace', 'Legal', 'GDP Growth']

for i in range(3):
    fig = plt.figure(i)
    plt.scatter(X_3.iloc[:,figs[i][0]], X_3.iloc[:,figs[i][1]], c=cmap(norm(y)), s=50)
    plt.scatter(centers[:, figs[i][0]], centers[:, figs[i][1]], c='black', s=200, alph
    plt.xlabel(labels[figs[i][0]])
    plt.ylabel(labels[figs[i][1]])

plt.show()
```



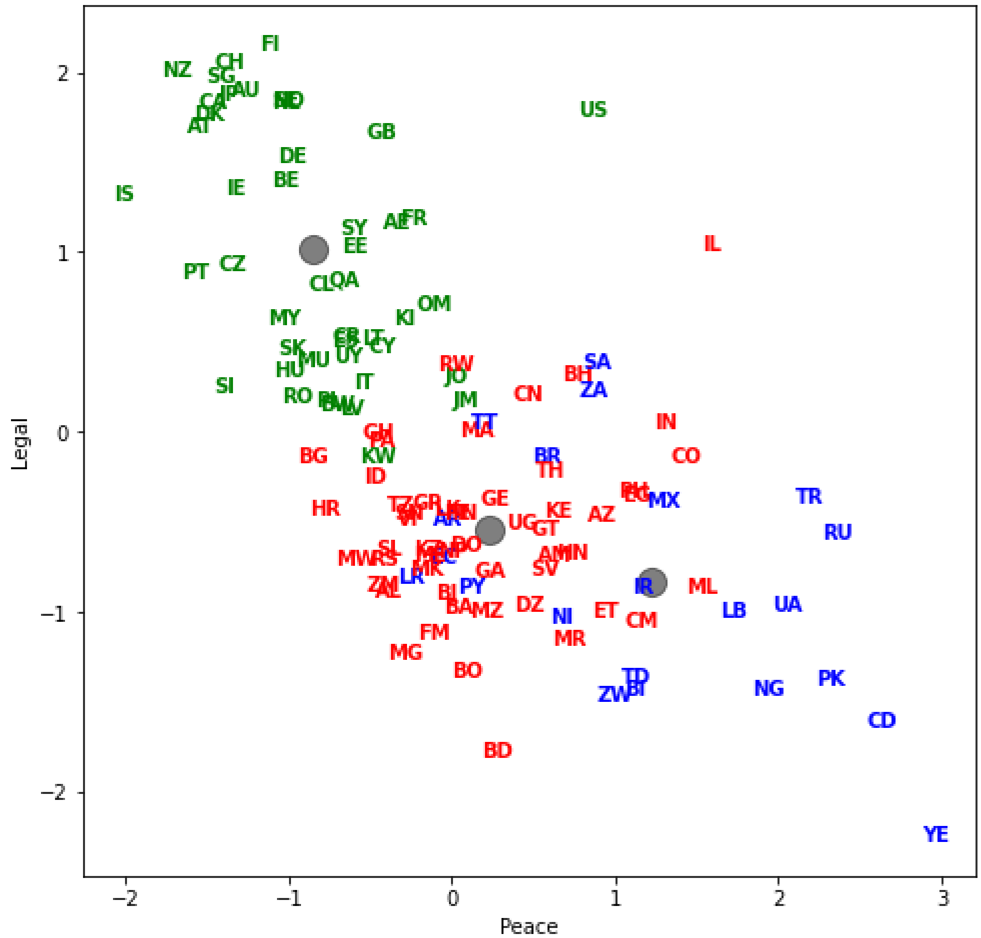


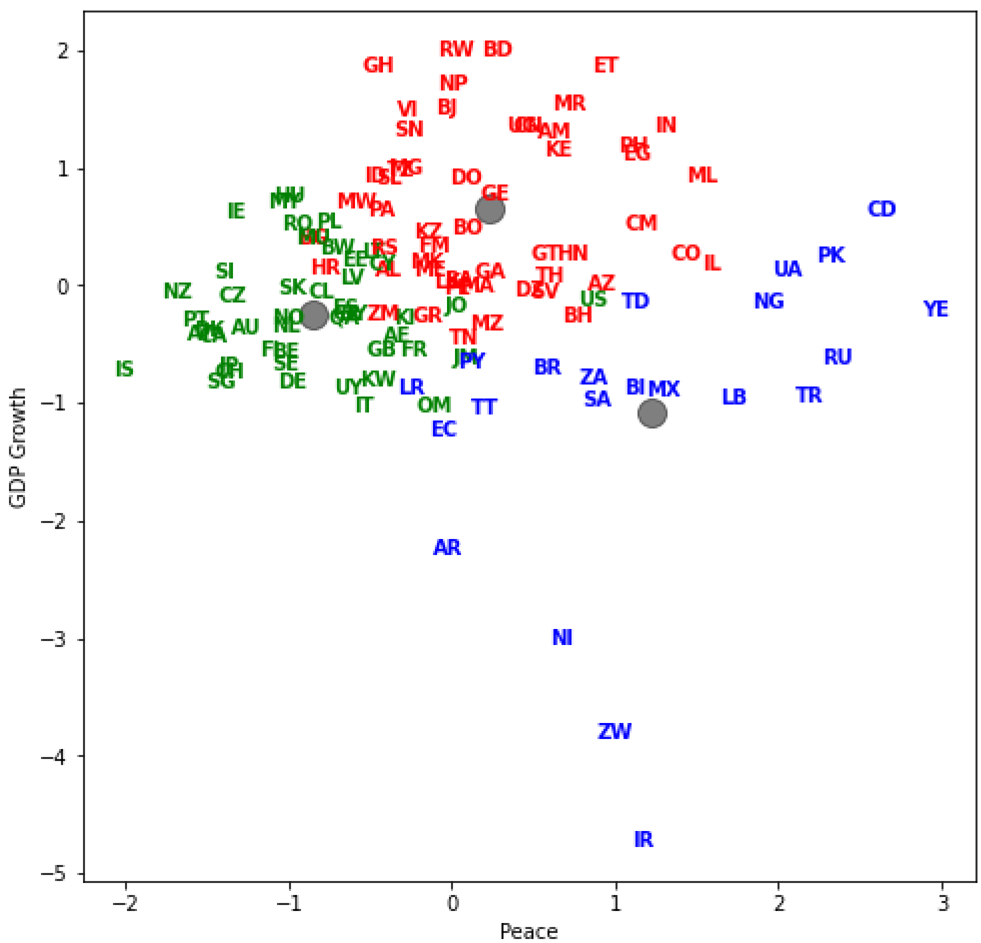


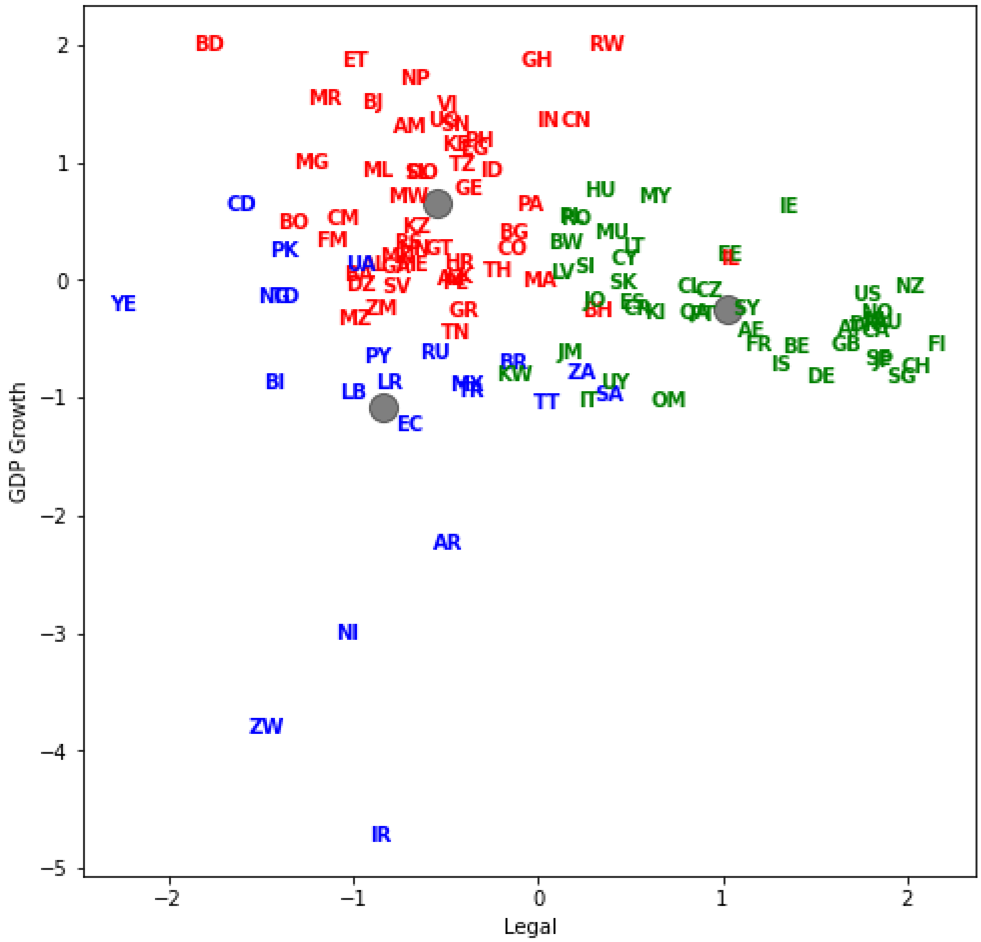
Visualize the result (3 2D plots)

In [10]:

```
%matplotlib inline
import matplotlib.pyplot as plt

figs = [(0, 1), (0, 2), (1, 2)]
labels = ['Peace', 'Legal', 'GDP Growth']
colors = ['blue','green', 'red']

for i in range(3):
    fig = plt.figure(i, figsize=(8, 8))
    x_1 = figs[i][0]
    x_2 = figs[i][1]
    plt.scatter(X_3.iloc[:, x_1], X_3.iloc[:, x_2], c=y, s=0, alpha=0)
    plt.scatter(centers[:, x_1], centers[:, x_2], c='black', s=200, alpha=0.5)
    for j in range(X_3.shape[0]):
        plt.text(X_3.iloc[j, x_1], X_3.iloc[j, x_2], raw['Abbrev'].iloc[j],
                 color=colors[y[j]], weight='semibold', horizontalalignment = 'center'
    plt.xlabel(labels[x_1])
    plt.ylabel(labels[x_2])

plt.show()
```


Legal
Peace
2
1
0
−1
−2
−2
−1
0
1
2
3


GDP Growth
Peace
2
1
0
−1
−2
−3
−4
−5
−2
−1
0
1
2
3



List the result

In [11]:
```
print('3 features')
result = pd.DataFrame({'Country':raw['Country'], 'Abbrev':raw['Abbrev'], 'Label':y})
risk = []
for label in result['Label']:
    if label == 0: risk.append('High')
    if label == 1: risk.append('Low')
    if label == 2: risk.append('Moderate')
result.insert(3, 'risk for 3 Features', risk)
with pd.option_context('display.max_rows', None, 'display.max_columns', 4):
    print(result.sort_values('Label'))
result_3Features = result
```

```
3 features
                         Country Abbrev  Label risk for 3 Features
60                       Lebanon     LB      0               High
30                       Ecuador     EC      0               High
48                          Iran     IR      0               High
61                       Liberia     LR      0               High
69                        Mexico     MX      0               High
77                     Nicaragua     NI      0               High
78                       Nigeria     NG      0               High
81                      Pakistan     PK      0               High
83                      Paraguay     PY      0               High
90                        Russia     RU      0               High
92                  Saudi Arabia     SA      0               High
99                  South Africa     ZA      0               High
108          Trinidad and Tobago     TT      0               High
110                       Turkey     TR      0               High
112                      Ukraine     UA      0               High
118                        Yemen     YE      0               High
27   Democratic Republic of Congo     CD      0               High
19                          Chad     TD      0               High
120                     Zimbabwe     ZW      0               High
14                        Brazil     BR      0               High
16                       Burundi     BI      0               High
2                      Argentina     AR      0               High
51                         Italy     IT      1                Low
103                  Switzerland     CH      1                Low
49                       Ireland     IE      1                Low
102                       Sweden     SE      1                Low
86                        Poland     PL      1                Low
52                       Jamaica     JM      1                Low
53                         Japan     JP      1                Low
54                        Jordan     JO      1                Low
100                        Spain     ES      1                Low
57                 Korea (South)     KI      1                Low
58                        Kuwait     KW      1                Low
59                        Latvia     LV      1                Low
5                        Austria     AT      1                Low
13                      Botswana     BW      1                Low
62                     Lithuania     LT      1                Low
98                      Slovenia     SI      1                Low
97                      Slovakia     SK      1                Low
65                      Malaysia     MY      1                Low
96                     Singapore     SG      1                Low
68                     Mauritius     MU      1                Low
104                       Taiwan     SY      1                Low
45                       Iceland     IS      1                Low
44                       Hungary     HU      1                Low
75                   Netherlands     NL      1                Low
20                         Chile     CL      1                Low
87                      Portugal     PT      1                Low
9                        Belgium     BE      1                Low
23                    Costa Rica     CR      1                Low
116                      Uruguay     UY      1                Low
25                        Cyprus     CY      1                Low
26                Czech Republic     CZ      1                Low
28                       Denmark     DK      1                Low
115                United States     US      1                Low
114               United Kingdom     GB      1                Low
18                        Canada     CA      1                Low
113         United Arab Emirates     AE      1                Low
```

80	Oman	OM	1	Low
35	Finland	FI	1	Low
36	France	FR	1	Low
79	Norway	NO	1	Low
88	Qatar	QA	1	Low
39	Germany	DE	1	Low
4	Australia	AU	1	Low
89	Romania	RO	1	Low
76	New Zealand	NZ	1	Low
33	Estonia	EE	1	Low
7	Bahrain	BH	2	Moderate
91	Rwanda	RW	2	Moderate
6	Azerbaijan	AZ	2	Moderate
17	Cameroon	CM	2	Moderate
94	Serbia	RS	2	Moderate
95	Sierra Leone	SL	2	Moderate
101	Sri Lanka	LK	2	Moderate
105	Tanzania	TZ	2	Moderate
106	Thailand	TH	2	Moderate
107	The FYR of Macedonia	MK	2	Moderate
109	Tunisia	TN	2	Moderate
3	Armenia	AM	2	Moderate
111	Uganda	UG	2	Moderate
117	Vietnam	VI	2	Moderate
1	Algeria	DZ	2	Moderate
93	Senegal	SN	2	Moderate
85	Philippines	PH	2	Moderate
12	Bosnia and Herzegovina	BA	2	Moderate
8	Bangladesh	BD	2	Moderate
43	Honduras	HN	2	Moderate
42	Guatemala	GT	2	Moderate
41	Greece	GR	2	Moderate
40	Ghana	GH	2	Moderate
38	Georgia	GE	2	Moderate
37	Gabon	GA	2	Moderate
46	India	IN	2	Moderate
34	Ethiopia	ET	2	Moderate
31	Egypt	EG	2	Moderate
15	Bulgaria	BG	2	Moderate
29	Dominican Republic	DO	2	Moderate
24	Croatia	HR	2	Moderate
22	Colombia	CO	2	Moderate
21	China	CN	2	Moderate
32	El Salvador	SV	2	Moderate
84	Peru	PE	2	Moderate
47	Indonesia	ID	2	Moderate
55	Kazakhstan	KZ	2	Moderate
82	Panama	PA	2	Moderate
10	Benin	BJ	2	Moderate
11	Bolivia	BO	2	Moderate
74	Nepal	NP	2	Moderate
73	Mozambique	MZ	2	Moderate
72	Morocco	MA	2	Moderate
50	Israel	IL	2	Moderate
71	Montenegro	ME	2	Moderate
67	Mauritania	MR	2	Moderate
66	Mali	ML	2	Moderate
64	Malawi	MW	2	Moderate
63	Madagascar	MG	2	Moderate
119	Zambia	ZM	2	Moderate

```
56                    Kenya    KE      2           Moderate
70                  Moldova    FM      2           Moderate
0                   Albania    AL      2           Moderate
```

In [12]:

```
#countries
high = 0
moderate = 0
low = 0
for label in result['Label']:
    if label == 0:
        high +=1
    if label == 1:
        low +=1
    if label == 2:
        moderate +=1
print('\nnumber of high risk countries when n_ini =10:',high)
print('number of moderate risk countries when n_ini =10:',moderate)
print('number of low risk countries when n_ini =10:',low)


result2 = pd.DataFrame({'Country':raw['Country'], 'Abbrev':raw['Abbrev'], 'Label':y2})
#countries
high2 = 0
moderate2 = 0
low2 = 0
for label in result2['Label']:
    if label == 0:
        high2 +=1
    if label == 1:
        low2 +=1
    if label == 2:
        moderate2 +=1
print('\nnumber of high risk countries when n_ini = 2:',high2)
print('number of moderate risk countries when n_ini = 2:',moderate2)
print('number of low risk countries when n_ini = 2:',low2)


result20 = pd.DataFrame({'Country':raw['Country'], 'Abbrev':raw['Abbrev'], 'Label':y2
#countries
high20 = 0
moderate20 = 0
low20 = 0
for label in result20['Label']:
    if label == 0:
        high20 +=1
    if label == 1:
        low20 +=1
    if label == 2:
        moderate20 +=1
print('\nnumber of high risk countries when n_ini = 20:',high20)
print('number of moderate risk countries when n_ini = 20:',moderate20)
print('number of low risk countries when n_ini = 20:',low20)
```

```
number of high risk countries when n_ini =10: 22
number of moderate risk countries when n_ini =10: 53
number of low risk countries when n_ini =10: 46

number of high risk countries when n_ini = 2: 70
number of moderate risk countries when n_ini = 2: 4
number of low risk countries when n_ini = 2: 47

number of high risk countries when n_ini = 20: 22
number of moderate risk countries when n_ini = 20: 53
number of low risk countries when n_ini = 20: 46
```

In [13]:

```
# Silhouette Analysis
range_n_clusters=[2,3,4,5,6,7,8,9,10]
for n_clusters in range_n_clusters:
    clusterer=KMeans(n_clusters=n_clusters, random_state=1)
    cluster_labels=clusterer.fit_predict(X_3)
    silhouette_avg=silhouette_score(X_3,cluster_labels)
    print("For n_clusters=", n_clusters,
          "The average silhouette_score is :", silhouette_avg)
```

```
For n_clusters= 2 The average silhouette_score is : 0.3509139523852161
For n_clusters= 3 The average silhouette_score is : 0.3558522334350506
For n_clusters= 4 The average silhouette_score is : 0.3372449209416129
For n_clusters= 5 The average silhouette_score is : 0.34438420977393375
For n_clusters= 6 The average silhouette_score is : 0.34875382122984605
For n_clusters= 7 The average silhouette_score is : 0.3603542108728006
For n_clusters= 8 The average silhouette_score is : 0.3394917368960437
For n_clusters= 9 The average silhouette_score is : 0.3152647236003266
For n_clusters= 10 The average silhouette_score is : 0.3090796538425007
```

Answer for Exercise 2.13

With the default n_init= 10, inertia is 161.13 with the cluster centers located at [[1.22 -0.83 -1.07],[-0.85 1.02 -0.23],[0.23 -0.54 0.65]] , with 70 countries in cluster 0(high), 4 countries in cluster 1(low), and 47 countries in cluster 2(moderate) respectively When we lower the n-init =2, the inertia increased to 169.24, which meant the sum of the square increased, worsening. with the cluster centers located at [[0.53 -0.61 0.34],[-0.85 0.99 -0.22],[0.70 -0.95 -3.43]] with 22 countries in cluster 0(high), 53 countries in cluster 1(low), and 46 countries in cluster 2(moderate) respectively If we increase the n_init to 20,30,50 or even higher the inertia and the locations of the cluster center will remain the same as the n_init = 10.

Exercise 2.14 A) 4 features

In [14]:

```
#4 feature
X_4 = raw[['Corruption','Peace', 'Legal', 'GDP Growth']]
```

```
X_4 = (X_4 - X_4.mean()) / X_4.std()
print(X_4.head(5))
print(' ')
```

```
   Corruption     Peace     Legal  GDP Growth
0   -0.633230 -0.390081 -0.878158    0.126952
1   -0.633230  0.472352 -0.958948   -0.040772
2   -0.098542 -0.026039 -0.484397   -2.230541
3   -0.258948  0.634871 -0.684553    1.303747
4    1.612460 -1.261182  1.900001   -0.368418
```

In [15]:

```
#4 feature
kmeans_4 = KMeans(n_clusters=k,n_init=10, random_state=1)
kmeans_4.fit(X_4)
y_4 = kmeans_4.labels_
print("inertia for 4 features ", kmeans_4.inertia_)

print("cluster centers: ", kmeans_4.cluster_centers_)
print("cluster labels: ", y_4)
```

```
inertia for 4 features  194.4046655009297
cluster centers:  [[ 1.17949284 -0.89877793  1.12417837 -0.26007806]
 [-0.49863571  0.17066495 -0.47838646  0.5929059 ]
 [-0.88356071  1.22506036 -0.83385901 -1.07842464]]
cluster labels:  [1 1 2 1 0 0 1 1 1 0 1 1 1 0 2 1 2 1 0 2 0 1 1 0 1 0 0 2 0 1 2 1 1 0
1 0 0
 1 1 0 1 1 1 1 1 0 1 1 2 0 1 0 1 0 1 1 1 0 1 0 2 2 0 1 1 0 1 1 0 2 1 1 1 1
 1 0 0 2 2 0 0 2 1 2 1 1 0 0 0 1 2 1 2 1 1 1 0 0 0 2 0 1 0 0 0 1 1 1 2 1 2
 1 2 0 0 0 0 1 2 1 2]
```

In [16]:

```
print('4 features')
result = pd.DataFrame({'Country':raw['Country'], 'Abbrev':raw['Abbrev'], 'Label':y_4})

#countries
high4 = 0
moderate4 = 0
low4 = 0
for label in result['Label']:
    if label == 0:
        low4 +=1
    if label == 1:
        moderate4 +=1
    if label == 2:
        high4 +=1
print('number of high risk countries:',high4)
print('number of moderate risk countries:',moderate4)
print('number of low risk countries:',low4)
print('\n')

risk = []
for label in result['Label']:
    if label == 0: risk.append('Low')
    if label == 1: risk.append('Moderate')
    if label == 2: risk.append('High')
result.insert(3, 'risk for 4 Features', risk)
with pd.option_context('display.max_rows', None, 'display.max_columns', 4):
    print(result.sort_values('Label'))
result_4Features = result
```

```
4 features
number of high risk countries: 22
number of moderate risk countries: 58
number of low risk countries: 41
```

	Country	Abbrev	Label	risk for 4 Features
51	Italy	IT	0	Low
33	Estonia	EE	0	Low
35	Finland	FI	0	Low
36	France	FR	0	Low
39	Germany	DE	0	Low
45	Iceland	IS	0	Low
49	Ireland	IE	0	Low
53	Japan	JP	0	Low
57	Korea (South)	KI	0	Low
59	Latvia	LV	0	Low
62	Lithuania	LT	0	Low
65	Malaysia	MY	0	Low
68	Mauritius	MU	0	Low
75	Netherlands	NL	0	Low
76	New Zealand	NZ	0	Low
79	Norway	NO	0	Low
80	Oman	OM	0	Low
86	Poland	PL	0	Low
87	Portugal	PT	0	Low
88	Qatar	QA	0	Low
104	Taiwan	SY	0	Low
96	Singapore	SG	0	Low
97	Slovakia	SK	0	Low
98	Slovenia	SI	0	Low
103	Switzerland	CH	0	Low
100	Spain	ES	0	Low
113	United Arab Emirates	AE	0	Low
28	Denmark	DK	0	Low
102	Sweden	SE	0	Low
26	Czech Republic	CZ	0	Low
114	United Kingdom	GB	0	Low
115	United States	US	0	Low
25	Cyprus	CY	0	Low
4	Australia	AU	0	Low
5	Austria	AT	0	Low
23	Costa Rica	CR	0	Low
116	Uruguay	UY	0	Low
20	Chile	CL	0	Low
9	Belgium	BE	0	Low
13	Botswana	BW	0	Low
18	Canada	CA	0	Low
29	Dominican Republic	DO	1	Moderate
10	Benin	BJ	1	Moderate
11	Bolivia	BO	1	Moderate
73	Mozambique	MZ	1	Moderate
12	Bosnia and Herzegovina	BA	1	Moderate
74	Nepal	NP	1	Moderate
72	Morocco	MA	1	Moderate
71	Montenegro	ME	1	Moderate
109	Tunisia	TN	1	Moderate
107	The FYR of Macedonia	MK	1	Moderate
84	Peru	PE	1	Moderate
106	Thailand	TH	1	Moderate

70	Moldova	FM	1	Moderate
85	Philippines	PH	1	Moderate
8	Bangladesh	BD	1	Moderate
7	Bahrain	BH	1	Moderate
89	Romania	RO	1	Moderate
105	Tanzania	TZ	1	Moderate
91	Rwanda	RW	1	Moderate
6	Azerbaijan	AZ	1	Moderate
93	Senegal	SN	1	Moderate
94	Serbia	RS	1	Moderate
95	Sierra Leone	SL	1	Moderate
3	Armenia	AM	1	Moderate
1	Algeria	DZ	1	Moderate
82	Panama	PA	1	Moderate
117	Vietnam	VI	1	Moderate
15	Bulgaria	BG	1	Moderate
47	Indonesia	ID	1	Moderate
31	Egypt	EG	1	Moderate
32	El Salvador	SV	1	Moderate
34	Ethiopia	ET	1	Moderate
24	Croatia	HR	1	Moderate
37	Gabon	GA	1	Moderate
38	Georgia	GE	1	Moderate
22	Colombia	CO	1	Moderate
40	Ghana	GH	1	Moderate
41	Greece	GR	1	Moderate
42	Guatemala	GT	1	Moderate
43	Honduras	HN	1	Moderate
44	Hungary	HU	1	Moderate
21	China	CN	1	Moderate
46	India	IN	1	Moderate
67	Mauritania	MR	1	Moderate
0	Albania	AL	1	Moderate
50	Israel	IL	1	Moderate
66	Mali	ML	1	Moderate
64	Malawi	MW	1	Moderate
63	Madagascar	MG	1	Moderate
111	Uganda	UG	1	Moderate
119	Zambia	ZM	1	Moderate
58	Kuwait	KW	1	Moderate
17	Cameroon	CM	1	Moderate
55	Kazakhstan	KZ	1	Moderate
54	Jordan	JO	1	Moderate
52	Jamaica	JM	1	Moderate
101	Sri Lanka	LK	1	Moderate
56	Kenya	KE	1	Moderate
112	Ukraine	UA	2	High
108	Trinidad and Tobago	TT	2	High
118	Yemen	YE	2	High
110	Turkey	TR	2	High
60	Lebanon	LB	2	High
92	Saudi Arabia	SA	2	High
2	Argentina	AR	2	High
14	Brazil	BR	2	High
16	Burundi	BI	2	High
19	Chad	TD	2	High
27	Democratic Republic of Congo	CD	2	High
30	Ecuador	EC	2	High
99	South Africa	ZA	2	High
48	Iran	IR	2	High

```
69                    Mexico    MX    2            High
77                 Nicaragua    NI    2            High
78                   Nigeria    NG    2            High
81                  Pakistan    PK    2            High
83                  Paraguay    PY    2            High
90                    Russia    RU    2            High
61                   Liberia    LR    2            High
120                 Zimbabwe    ZW    2            High
```

In [17]:
```
for i in range(len(result_3Features)):
    if result_3Features.iloc[i]['risk for 3 Features'] != result_4Features.iloc[i]['ri
        print(result_4Features.iloc[i]['Country'],'risk changed from ',result_3Feature
```

```
Hungary risk changed from  Low  to  Moderate
Jamaica risk changed from  Low  to  Moderate
Jordan risk changed from  Low  to  Moderate
Kuwait risk changed from  Low  to  Moderate
Romania risk changed from  Low  to  Moderate
```

Answer for 2.14 A)

when we compare the high-risk cluster between the four features and three features, we can see that there is no difference; all the high-risk countries remain the same (22 countries); this is because the Corruption and Legal risk index are highly correlated(0.938512) according to the correlation matrix. Hence adding corruption did not affect the number or members of the high-risk countries. But it does have some effects on the lower-risk countries. I have observed five additional countries that have moved from low to moderated risk: Hungary, Jamaica, Kuwait and Romania. As a result, moderate-risk countries have increased from 53 to 58, while low-risk countries have decreased from 46 to 41.

Exercise 2.14 B)

In [18]:
```
from sklearn.cluster import AgglomerativeClustering
data=X_3

# ward:minimizes the variance of the distances of all the observations of the two sets
model_ward = AgglomerativeClustering(n_clusters=3, linkage='ward')
clusters_ward = model_ward.fit_predict(data)
cluster_centers_ward = []
for i in range(model_ward.n_clusters):
    cluster_points_ward = data[clusters_ward==i]
    cluster_centers_ward.append(np.mean(cluster_points_ward, axis=0))
agg_clustering_ward = AgglomerativeClustering(n_clusters=3, linkage='ward')
agg_clustering_ward.fit(X_3)
cluster_labels_ward = agg_clustering_ward.labels_
print('Ward Method:\n','cluster centers: \n',cluster_centers_ward)
```

```
print('cluster labels:')
print(cluster_labels_ward,'\n')



# complete: maximum distance between all the observations of the two sets.
model_complete = AgglomerativeClustering(n_clusters=3, linkage='complete')
clusters_complete = model_complete.fit_predict(data)
cluster_centers_complete = []
for i in range(model_complete.n_clusters):
    cluster_points_complete = data[clusters_complete==i]
    cluster_centers_complete.append(np.mean(cluster_points_complete, axis=0))
agg_clustering_complete = AgglomerativeClustering(n_clusters=3, linkage='complete')
agg_clustering_complete.fit(X_3)
cluster_labels_complete = agg_clustering_complete.labels_
print('\n Complete Method:\n','cluster centers: \n',cluster_centers_complete)
print('cluster labels:')
print(cluster_labels_complete,'\n')


# average: average distance between all the observations of the two sets.
model_average = AgglomerativeClustering(n_clusters=3, linkage='average')
clusters_average = model_average.fit_predict(data)
cluster_centers_average = []
for i in range(model_average.n_clusters):
    cluster_points_average = data[clusters_average==i]
    cluster_centers_average.append(np.mean(cluster_points_average, axis=0))
agg_clustering_average = AgglomerativeClustering(n_clusters=3, linkage='average')
agg_clustering_average.fit(X_3)
cluster_labels_average = agg_clustering_average.labels_
print('\n Average Method:\n','cluster centers: \n',cluster_centers_average)
print('cluster labels:')
print(cluster_labels_average,'\n')


# single: minimum distance between all the observations of the two sets.
model_single = AgglomerativeClustering(n_clusters=3, linkage='single')
clusters_single = model_single.fit_predict(data)
cluster_centers_single = []
for i in range(model_single.n_clusters):
    cluster_points_single = data[clusters_single==i]
    cluster_centers_single.append(np.mean(cluster_points_single, axis=0))
agg_clustering_single = AgglomerativeClustering(n_clusters=3, linkage='single')
agg_clustering_single.fit(X_3)
cluster_labels_single = agg_clustering_single.labels_
print('\n Single Method:\n','cluster centers: \n',cluster_centers_single)
print('cluster labels:')
```

```
Ward Method:
 cluster centers:
 [Peace         0.135708
Legal        -0.361024
GDP Growth    0.321242
dtype: float64, Peace        -1.015364
Legal         1.246616
GDP Growth   -0.240331
dtype: float64, Peace         1.758263
Legal        -1.145015
GDP Growth   -1.091385
dtype: float64]
cluster labels:
[0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 2 0 1 2 1 0 0 1 0 1 1 2 1 0 0 0 0 1 0 1 1
 0 0 1 0 0 0 0 1 1 0 0 2 1 0 0 0 1 0 0 0 1 0 0 2 0 1 0 0 1 0 0 0 2 0 0 0 0
 0 1 1 2 2 1 0 2 0 0 0 0 0 1 1 0 2 0 0 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0 0 2
 0 2 1 1 0 0 0 2 0 2]


 Complete Method:
 cluster centers:
 [Peace        -0.478015
Legal         0.408516
GDP Growth   -0.134457
dtype: float64, Peace         0.949074
Legal        -1.117131
GDP Growth   -3.841728
dtype: float64, Peace         0.969513
Legal        -0.803741
GDP Growth    0.605844
dtype: float64]
cluster labels:
[0 0 0 2 0 0 2 0 2 0 2 2 0 0 0 0 2 2 0 2 0 2 2 0 0 0 0 2 0 2 0 2 0 0 2 0 0
 0 2 0 0 0 2 2 0 0 2 0 1 0 0 0 0 0 0 0 2 0 0 0 2 0 0 2 0 0 2 2 0 2 2 0 0 0
 2 0 0 1 2 0 0 2 0 0 0 2 0 0 0 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2
 2 2 0 0 0 0 2 2 0 1]


 Average Method:
 cluster centers:
 [Peace        -0.231978
Legal         0.155798
GDP Growth    0.165188
dtype: float64, Peace         0.705296
Legal        -0.958948
GDP Growth   -3.438931
dtype: float64, Peace         1.978951
Legal        -1.152619
GDP Growth   -0.341292
dtype: float64]
cluster labels:
[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 2 0 0 0 0
 0 0 0 1 2 0 0 2 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2
 0 2 0 0 0 0 0 2 0 1]


 Single Method:
 cluster centers:
 [Peace         0.949074
```

```
Legal        -1.117131
GDP Growth   -3.841728
dtype: float64, Peace         1.228606
Legal         1.411257
GDP Growth    0.032558
dtype: float64, Peace        -0.045728
Legal         0.004559
GDP Growth    0.098794
dtype: float64]
cluster labels:
```

In [19]:

```
# high risk cluster for ward = 2
print('comparison between ward and kmean')
for i in range(len(result_3Features)):
    raw_label = result_3Features.iloc[i]['Label']
    ward_label = agg_clustering_ward.labels_[i]

    if raw_label == 0 and ward_label == 0:
        print('raw 0 ward 0 country:' + result_3Features.iloc[i]['Country'])
    elif raw_label == 0 and ward_label == 1:
        print('raw 0 ward 2 country:' + result_3Features.iloc[i]['Country'])

# high risk cluster for complete = 1
print('\ncomparison between complete and kmean')
for i in range(len(result_3Features)):
    raw_label = result_3Features.iloc[i]['Label']
    complete_label = agg_clustering_complete.labels_[i]

    if raw_label == 0 and complete_label == 0:
        print('raw 0 ward 0 country:' + result_3Features.iloc[i]['Country'])
    elif raw_label == 0 and complete_label == 2:
        print('raw 0 ward 2 country:' + result_3Features.iloc[i]['Country'])

# high risk cluster for average = 2
print('\ncomparison between average and kmean')
for i in range(len(result_3Features)):
    raw_label = result_3Features.iloc[i]['Label']
    average_label = agg_clustering_average.labels_[i]

    if raw_label == 0 and average_label == 0:
        print('raw 0 ward 0 country:' + result_3Features.iloc[i]['Country'])
    elif raw_label == 0 and average_label == 1:
        print('raw 0 ward 2 country:' + result_3Features.iloc[i]['Country'])

# high risk cluster for single = 0
print('\ncomparison between single and kmean')
for i in range(len(result_3Features)):
    raw_label = result_3Features.iloc[i]['Label']
    single_label = agg_clustering_single.labels_[i]

    if raw_label == 0 and single_label == 1:
        print('raw 0 ward 0 country:' + result_3Features.iloc[i]['Country'])
    elif raw_label == 0 and single_label == 2:
        print('raw 0 ward 2 country:' + result_3Features.iloc[i]['Country'])
```

```
comparison between ward and kmean
raw 0 ward 0 country:Argentina
raw 0 ward 0 country:Brazil
raw 0 ward 0 country:Ecuador
raw 0 ward 0 country:Liberia
raw 0 ward 0 country:Paraguay
raw 0 ward 0 country:Saudi Arabia
raw 0 ward 0 country:South Africa
raw 0 ward 0 country:Trinidad and Tobago

comparison between complete and kmean
raw 0 ward 0 country:Argentina
raw 0 ward 0 country:Brazil
raw 0 ward 2 country:Burundi
raw 0 ward 2 country:Chad
raw 0 ward 2 country:Democratic Republic of Congo
raw 0 ward 0 country:Ecuador
raw 0 ward 2 country:Lebanon
raw 0 ward 0 country:Liberia
raw 0 ward 2 country:Mexico
raw 0 ward 2 country:Nigeria
raw 0 ward 2 country:Pakistan
raw 0 ward 0 country:Paraguay
raw 0 ward 2 country:Russia
raw 0 ward 0 country:Saudi Arabia
raw 0 ward 0 country:South Africa
raw 0 ward 0 country:Trinidad and Tobago
raw 0 ward 2 country:Turkey
raw 0 ward 2 country:Ukraine
raw 0 ward 2 country:Yemen

comparison between average and kmean
raw 0 ward 2 country:Argentina
raw 0 ward 0 country:Brazil
raw 0 ward 0 country:Ecuador
raw 0 ward 2 country:Iran
raw 0 ward 0 country:Liberia
raw 0 ward 2 country:Nicaragua
raw 0 ward 0 country:Paraguay
raw 0 ward 0 country:Saudi Arabia
raw 0 ward 0 country:South Africa
raw 0 ward 0 country:Trinidad and Tobago
raw 0 ward 2 country:Zimbabwe

comparison between single and kmean
raw 0 ward 2 country:Argentina
raw 0 ward 2 country:Brazil
raw 0 ward 2 country:Burundi
raw 0 ward 2 country:Chad
raw 0 ward 2 country:Democratic Republic of Congo
raw 0 ward 2 country:Ecuador
raw 0 ward 2 country:Lebanon
raw 0 ward 2 country:Liberia
raw 0 ward 2 country:Mexico
raw 0 ward 2 country:Nigeria
raw 0 ward 2 country:Pakistan
raw 0 ward 2 country:Paraguay
raw 0 ward 2 country:Russia
raw 0 ward 2 country:Saudi Arabia
raw 0 ward 2 country:South Africa
```

```
raw 0 ward 2 country:Trinidad and Tobago
raw 0 ward 2 country:Turkey
raw 0 ward 2 country:Ukraine
raw 0 ward 2 country:Yemen
```

Answer for 2.14 B)

there are difference between the hierarchical clustering package and the scikit-learn package kmean and the diffidence within high risk cluster are demonstrated on the above cell

---

Exercise 2.15

In [20]:
```
raw_data = {'Country': ['Venezuela'],'Abbrev': ['VN'],'Corruption': [16],'Peace': [2.6
df = pd.DataFrame(raw_data)
raw_VN = pd.concat([raw,df],ignore_index = True)
X_VN = raw_VN
raw_VN
```

Out[20]:

	Country	Abbrev	Corruption	Peace	Legal	GDP Growth
0	Albania	AL	35	1.821	4.546	2.983
1	Algeria	DZ	35	2.219	4.435	2.553
2	Argentina	AR	45	1.989	5.087	-3.061
3	Armenia	AM	42	2.294	4.812	6.000
4	Australia	AU	77	1.419	8.363	1.713
...	...	...	...	...	...	...
117	Vietnam	VI	37	1.877	5.084	6.500
118	Yemen	YE	15	3.369	2.671	2.113
119	Zambia	ZM	34	1.805	4.592	2.021
120	Zimbabwe	ZW	24	2.463	3.738	-7.077
121	Venezuela	VN	16	2.600	2.895	-35.000

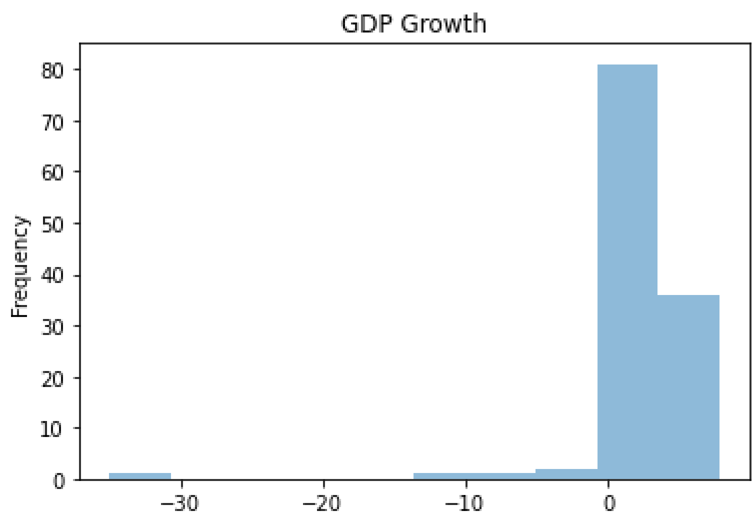
122 rows × 6 columns

In [21]:
```
X_VN = raw_VN[['Peace', 'Legal', 'GDP Growth']]
X_VN = (X_VN - X_VN.mean()) / X_VN.std()
print(X_VN.tail(5))
```

```
        Peace     Legal  GDP Growth
117 -0.278601 -0.463277    0.974593
118  2.945514 -2.196149   -0.055375
119 -0.434189 -0.816602   -0.076974
120  0.987707 -1.429893   -2.212977
121  1.283755 -2.035285   -8.768659
```

In [22]:
```
plt.figure(4)
raw_VN['GDP Growth'].plot(kind = 'hist', title = 'GDP Growth', alpha = 0.5)
```

Out[22]:
```
<AxesSubplot:title={'center':'GDP Growth'}, ylabel='Frequency'>
```



In [23]:
```
# print summary statistics
print("\nSummary statistics after adding Venezuela\n", raw_VN.describe())
print("\nCorrelation matrix after adding Venezuela\n", raw_VN.corr())
```

```
Summary statistics after adding Venezuela
        Corruption       Peace       Legal  GDP Growth
count  122.000000  122.000000  122.000000  122.000000
mean    46.590164    2.005926    5.729107    2.348861
std     18.833219    0.462763    1.392486    4.259358
min     15.000000    1.072000    2.671000  -35.000000
25%     32.250000    1.700750    4.726500    1.245250
50%     41.000000    1.945000    5.430000    2.597500
75%     59.750000    2.298500    6.481000    3.998750
max     87.000000    3.369000    8.712000    7.800000

Correlation matrix after adding Venezuela
             Corruption     Peace     Legal  GDP Growth
Corruption    1.000000 -0.709781  0.939526    0.045444
Peace        -0.709781  1.000000 -0.667992   -0.096436
Legal         0.939526 -0.667992  1.000000    0.060148
GDP Growth    0.045444 -0.096436  0.060148    1.000000
```

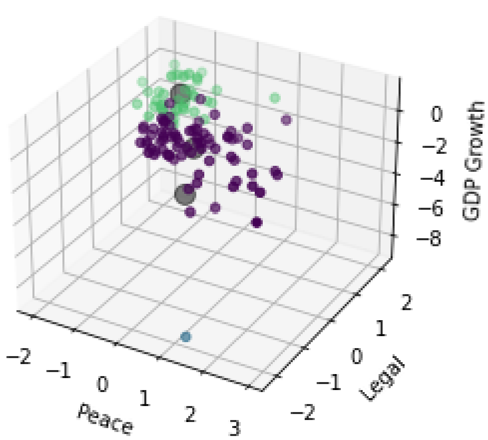
In [24]:
```
k = 3
kmeans_VN = KMeans(n_clusters=k, random_state=1)
kmeans_VN.fit(X_VN)
y_VN = kmeans_VN.labels_
print("inertia after adding Venezuela is", kmeans_VN.inertia_)
print(' ')
print("cluster centers after adding Venezuela is: ", kmeans_VN.cluster_centers_)
print(' ')
```

```
print("cluster labels after adding Venezuela is: ", y_VN)
print(' ')
```

```
inertia after adding Venezuela is 147.57635157028682

cluster centers after adding Venezuela is:  [[ 0.50295568 -0.57899064  0.14262782]
 [ 1.28375483 -2.03528534 -8.76865913]
 [-0.90934868  1.05949011 -0.05031098]]

cluster labels after adding Venezuela is:  [0 0 0 0 2 2 0 0 0 2 0 0 0 2 0 2 0 0 2 0 2
0 0 2 0 2 2 0 2 0 0 0 0 2 0 2 2
 0 0 2 0 0 0 0 2 2 0 0 0 2 0 2 0 2 0 0 0 2 0 2 0 0 2 0 0 2 0 0 2 0 0 0 0 0
 0 2 2 0 0 2 2 0 0 0 0 0 2 2 2 2 0 0 0 0 0 0 2 2 2 0 2 0 2 2 2 0 0 0 0 0 0
 0 0 2 2 2 2 0 0 0 0 1]
```

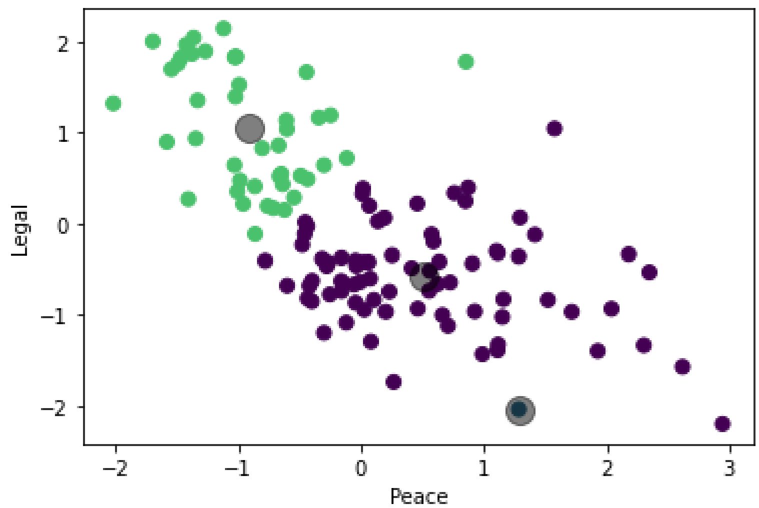
In [25]:

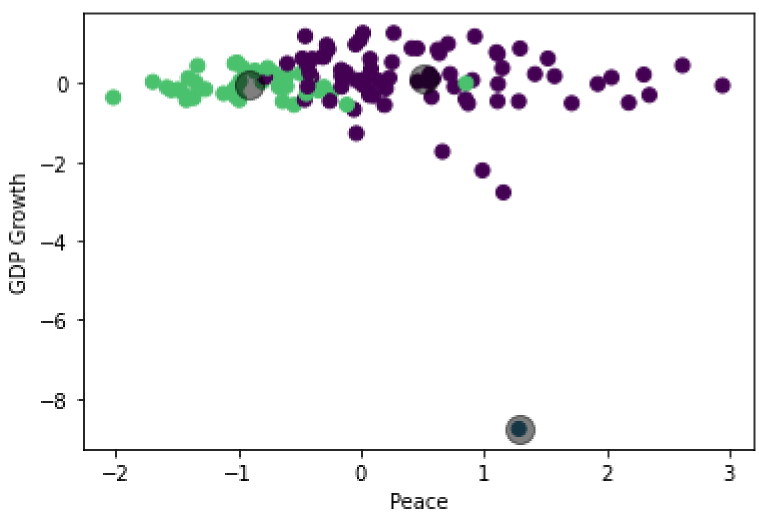
```
norm = clrs.Normalize(vmin=0.,vmax=y_VN.max() + 0.8)
cmap = cm.viridis
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_VN.iloc[:,0], X_VN.iloc[:,1], X_VN.iloc[:,2], c=cmap(norm(y_VN)), marker=
centers = kmeans_VN.cluster_centers_
ax.scatter(centers[:, 0], centers[:, 1], c='black', s=100, alpha=0.5)
ax.set_xlabel('Peace')
ax.set_ylabel('Legal')
ax.set_zlabel('GDP Growth')
plt.show()
```

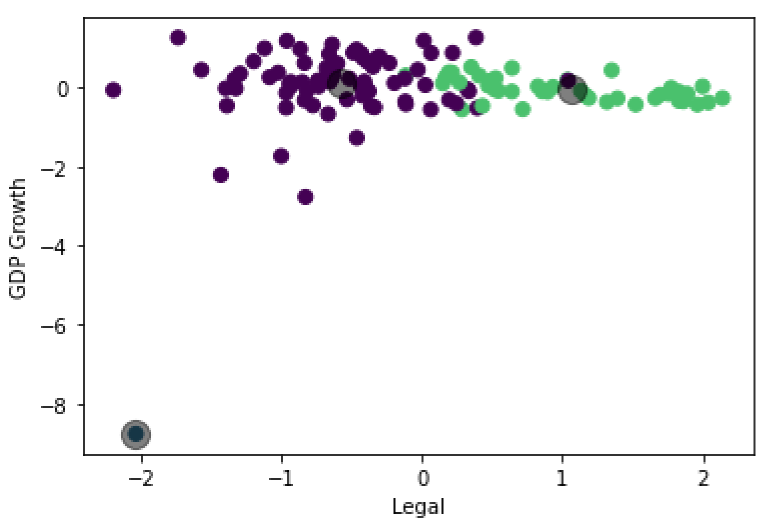


In [26]:

```
%matplotlib inline
import matplotlib.pyplot as plt

figs = [(0, 1), (0, 2), (1, 2)]
labels = ['Peace', 'Legal', 'GDP Growth']

for i in range(3):
    fig = plt.figure(i)
    plt.scatter(X_VN.iloc[:,figs[i][0]], X_VN.iloc[:,figs[i][1]], c=cmap(norm(y_VN)),
    plt.scatter(centers[:, figs[i][0]], centers[:, figs[i][1]], c='black', s=200, alph
    plt.xlabel(labels[figs[i][0]])
    plt.ylabel(labels[figs[i][1]])

plt.show()
```



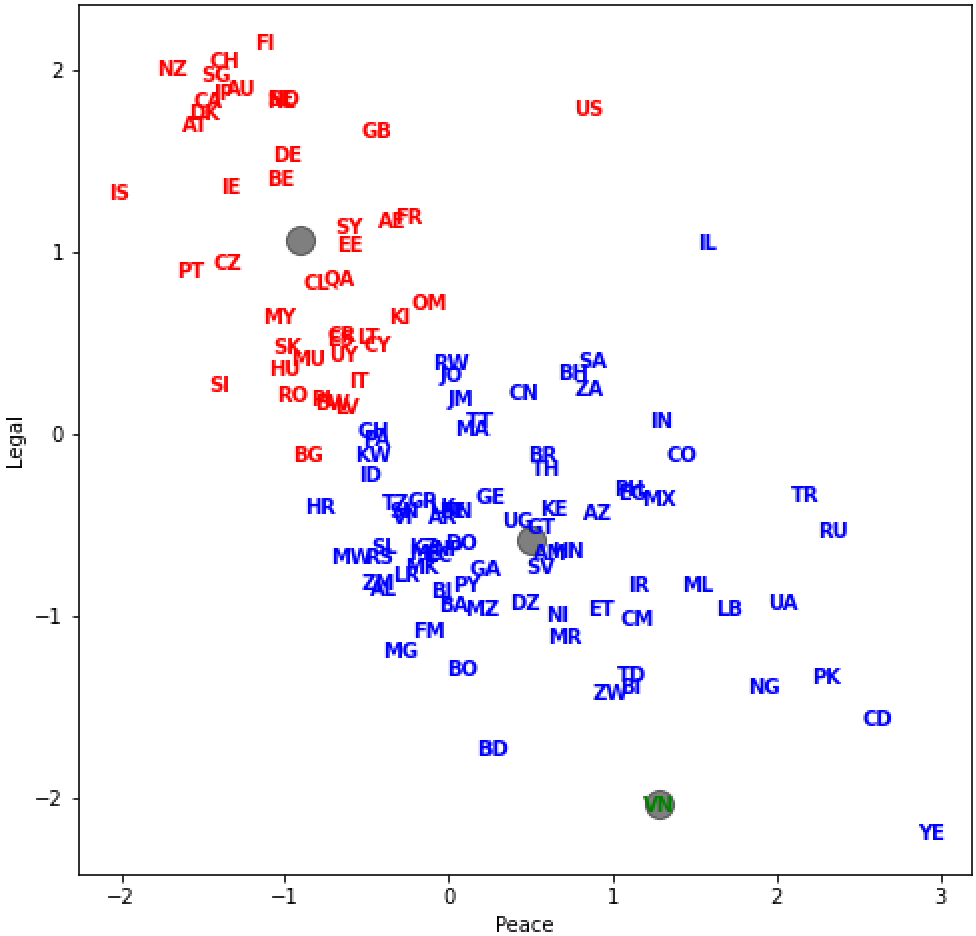


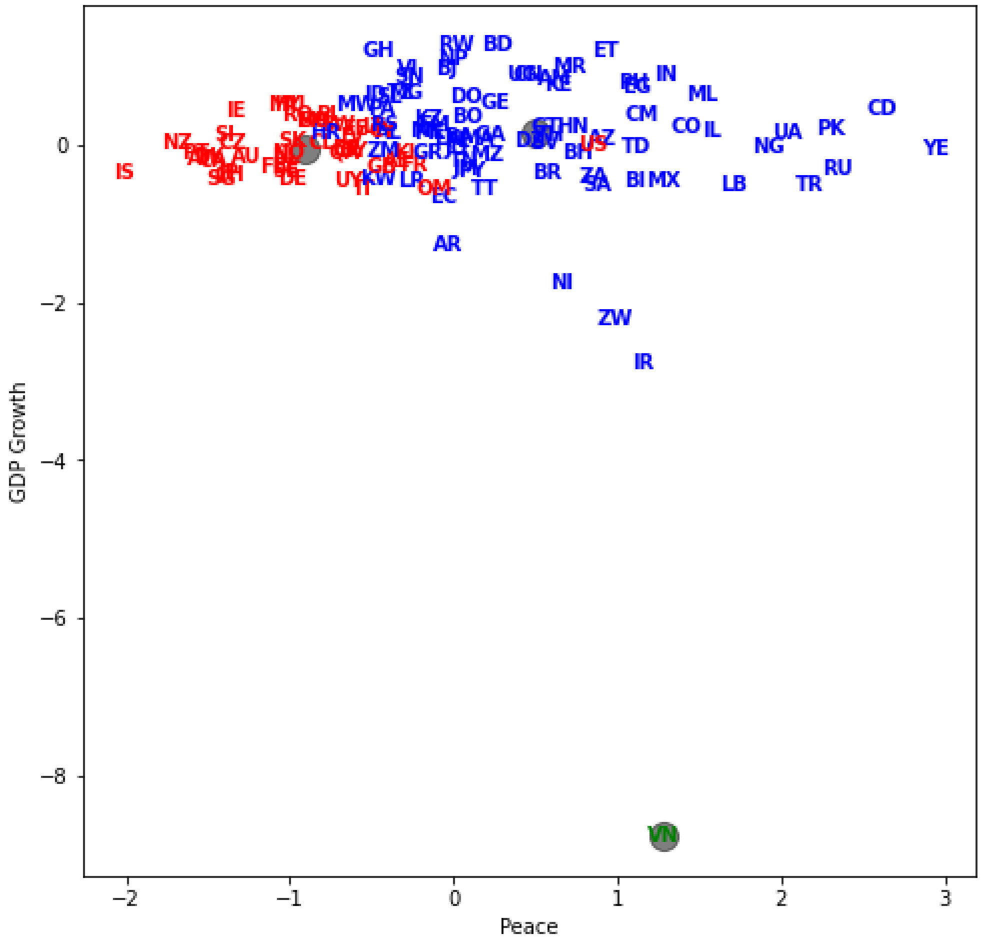


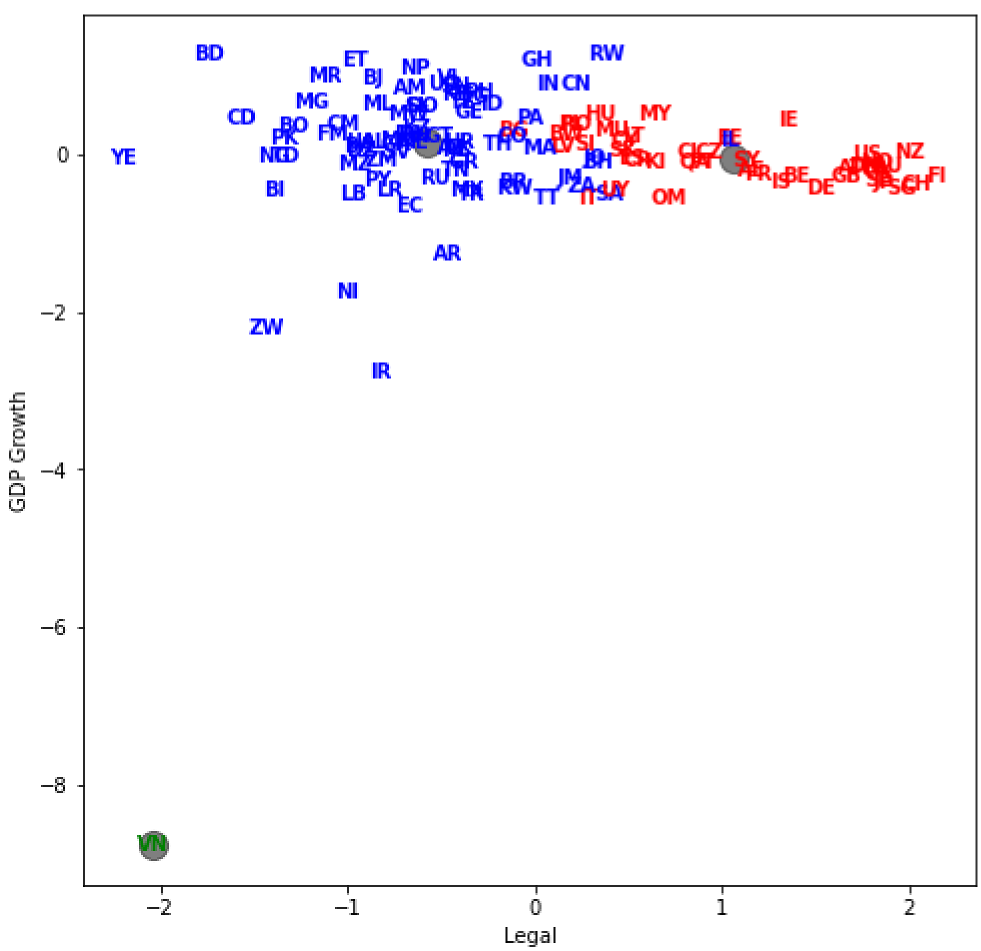
In [27]:

```
%matplotlib inline
import matplotlib.pyplot as plt

figs = [(0, 1), (0, 2), (1, 2)]
labels = ['Peace', 'Legal', 'GDP Growth']
colors = ['blue','green', 'red']

for i in range(3):
    fig = plt.figure(i, figsize=(8, 8))
    x_1 = figs[i][0]
```

```
    x_2 = figs[i][1]
    plt.scatter(X_VN.iloc[:, x_1], X_VN.iloc[:, x_2], c=y_VN, s=0, alpha=0)
    plt.scatter(centers[:, x_1], centers[:, x_2], c='black', s=200, alpha=0.5)
    for j in range(X_VN.shape[0]):
        plt.text(X_VN.iloc[j, x_1], X_VN.iloc[j, x_2], raw_VN['Abbrev'].iloc[j],
                 color=colors[y_VN[j]], weight='semibold', horizontalalignment = 'cent
    plt.xlabel(labels[x_1])
    plt.ylabel(labels[x_2])

plt.show()
```




GDP Growth
Peace
0
−2
−4
−6
−8
−2
−1
0
1
2
3


GDP Growth
Legal
0
−2
−4
−6
−8
−2
−1
0
1
2

In [28]:
```
print('after adding VN')
result = pd.DataFrame({'Country':raw_VN['Country'], 'Abbrev':raw_VN['Abbrev'], 'Label'

#countries
highVN = 0
moderateVN = 0
lowVN = 0
for label in result['Label']:
    if label == 0:
        lowVN +=1
    if label == 1:
        moderateVN +=1
    if label == 2:
        highVN +=1
print('number of high risk countries:',highVN)
print('number of moderate risk countries:',moderateVN)
print('number of low risk countries:',lowVN)
print('\n ')

risk = []
for label in result['Label']:
    if label == 0: risk.append('Low')
    if label == 1: risk.append('Moderate')
    if label == 2: risk.append('High')
result.insert(3, 'risk after adding VN', risk)
with pd.option_context('display.max_rows', None, 'display.max_columns', 4):
    print(result.sort_values('Label'))
result_VN = result
```

```
after adding VN
number of high risk countries: 44
number of moderate risk countries: 1
number of low risk countries: 77
```

	Country	Abbrev	Label	risk after adding VN
0	Albania	AL	0	Low
78	Nigeria	NG	0	Low
77	Nicaragua	NI	0	Low
74	Nepal	NP	0	Low
73	Mozambique	MZ	0	Low
72	Morocco	MA	0	Low
71	Montenegro	ME	0	Low
70	Moldova	FM	0	Low
69	Mexico	MX	0	Low
67	Mauritania	MR	0	Low
81	Pakistan	PK	0	Low
66	Mali	ML	0	Low
63	Madagascar	MG	0	Low
61	Liberia	LR	0	Low
120	Zimbabwe	ZW	0	Low
58	Kuwait	KW	0	Low
56	Kenya	KE	0	Low
55	Kazakhstan	KZ	0	Low
54	Jordan	JO	0	Low
52	Jamaica	JM	0	Low
50	Israel	IL	0	Low
64	Malawi	MW	0	Low
48	Iran	IR	0	Low
82	Panama	PA	0	Low
84	Peru	PE	0	Low
119	Zambia	ZM	0	Low
118	Yemen	YE	0	Low
117	Vietnam	VI	0	Low
112	Ukraine	UA	0	Low
111	Uganda	UG	0	Low
110	Turkey	TR	0	Low
109	Tunisia	TN	0	Low
108	Trinidad and Tobago	TT	0	Low
107	The FYR of Macedonia	MK	0	Low
83	Paraguay	PY	0	Low
106	Thailand	TH	0	Low
101	Sri Lanka	LK	0	Low
99	South Africa	ZA	0	Low
95	Sierra Leone	SL	0	Low
94	Serbia	RS	0	Low
93	Senegal	SN	0	Low
92	Saudi Arabia	SA	0	Low
91	Rwanda	RW	0	Low
90	Russia	RU	0	Low
85	Philippines	PH	0	Low
105	Tanzania	TZ	0	Low
47	Indonesia	ID	0	Low
60	Lebanon	LB	0	Low
21	China	CN	0	Low
27	Democratic Republic of Congo	CD	0	Low
46	India	IN	0	Low
29	Dominican Republic	DO	0	Low
30	Ecuador	EC	0	Low

31	Egypt	EG	0	Low
32	El Salvador	SV	0	Low
19	Chad	TD	0	Low
17	Cameroon	CM	0	Low
34	Ethiopia	ET	0	Low
16	Burundi	BI	0	Low
14	Brazil	BR	0	Low
12	Bosnia and Herzegovina	BA	0	Low
22	Colombia	CO	0	Low
11	Bolivia	BO	0	Low
38	Georgia	GE	0	Low
10	Benin	BJ	0	Low
8	Bangladesh	BD	0	Low
7	Bahrain	BH	0	Low
40	Ghana	GH	0	Low
41	Greece	GR	0	Low
42	Guatemala	GT	0	Low
6	Azerbaijan	AZ	0	Low
43	Honduras	HN	0	Low
3	Armenia	AM	0	Low
2	Argentina	AR	0	Low
1	Algeria	DZ	0	Low
37	Gabon	GA	0	Low
24	Croatia	HR	0	Low
121	Venezuela	VN	1	Moderate
25	Cyprus	CY	2	High
102	Sweden	SE	2	High
103	Switzerland	CH	2	High
104	Taiwan	SY	2	High
62	Lithuania	LT	2	High
59	Latvia	LV	2	High
9	Belgium	BE	2	High
39	Germany	DE	2	High
57	Korea (South)	KI	2	High
36	France	FR	2	High
53	Japan	JP	2	High
4	Australia	AU	2	High
113	United Arab Emirates	AE	2	High
114	United Kingdom	GB	2	High
115	United States	US	2	High
116	Uruguay	UY	2	High
51	Italy	IT	2	High
44	Hungary	HU	2	High
49	Ireland	IE	2	High
5	Austria	AT	2	High
100	Spain	ES	2	High
13	Botswana	BW	2	High
98	Slovenia	SI	2	High
79	Norway	NO	2	High
80	Oman	OM	2	High
76	New Zealand	NZ	2	High
23	Costa Rica	CR	2	High
75	Netherlands	NL	2	High
28	Denmark	DK	2	High
20	Chile	CL	2	High
86	Poland	PL	2	High
87	Portugal	PT	2	High
88	Qatar	QA	2	High
89	Romania	RO	2	High
68	Mauritius	MU	2	High

```
18          Canada   CA   2    High
33         Estonia   EE   2    High
65        Malaysia   MY   2    High
15        Bulgaria   BG   2    High
35         Finland   FI   2    High
96       Singapore   SG   2    High
97        Slovakia   SK   2    High
26  Czech Republic   CZ   2    High
45         Iceland   IS   2    High
```

In [29]:

```
 #countries
high_VN = 0
moderate_VN = 0
low_VN = 0
for label in result_VN['Label']:
    if label == 0:
        low_VN +=1
    if label == 1:
        moderate_VN +=1
    if label == 2:
        high_VN +=1
print('after adding Venezuela')
print('number of high risk countries:',high_VN)
print('number of moderate risk countries:',moderate_VN)
print('number of low risk countries:',low_VN)
```

```
after adding Venezuela
number of high risk countries: 44
number of moderate risk countries: 1
number of low risk countries: 77
```

In [30]:

```
# Silhouette Analysis
range_n_clusters=[2,3,4,5,6,7,8,9,10]
for n_clusters in range_n_clusters:
    clusterer=KMeans(n_clusters=n_clusters, random_state=1)
    cluster_labels=clusterer.fit_predict(X_VN)
    silhouette_avg=silhouette_score(X_VN,cluster_labels)
    print("For n_clusters=", n_clusters,
          "The average silhouette_score is :", silhouette_avg)
```

```
For n_clusters= 2 The average silhouette_score is : 0.3860706397862347
For n_clusters= 3 The average silhouette_score is : 0.4060901042248943
For n_clusters= 4 The average silhouette_score is : 0.3808812129650175
For n_clusters= 5 The average silhouette_score is : 0.330518549682561
For n_clusters= 6 The average silhouette_score is : 0.3458206705941977
For n_clusters= 7 The average silhouette_score is : 0.35975995710106656
For n_clusters= 8 The average silhouette_score is : 0.34113116069185917
For n_clusters= 9 The average silhouette_score is : 0.3286221418241615
For n_clusters= 10 The average silhouette_score is : 0.32181439409640594
```

Answer for 2.15

Venezuela's features are relatively extreme compared to the rest of the data set. After adding Venezuela to the data set, the standard deviation and mean of the new data set have shifted by a fair amount across the four features, almost doubling the standard deviation for GDP. As the result, the new data point(Venezuela) becomes a cluster center itself and others were relocated to higher or lower risk sector, and as the result of that it actually improve the inertia from 161.13

to 147.57. Given the above observations, I have concluded that the K-mean is highly sensitive to outliers.

In [ ]: