QUERY 1-

6.1 Create a table/data frame with the closing prices of 30 different stocks, with 10 from each of the caps

In [1]:

```python
%matplotlib inline
from math import sqrt
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
symbol_dict={
    'BAJFINANCE': 'Bajaj Finance',
    'TCS': 'TCS',
    'AXISBANK': 'Axis Bank',
    'HDFC':'HDFC Bank',
    'CIPLA':'cipla Pharma',
    'GAIL':'GAIL',
    'HINDALCO':'Hindalco',
    'INFY':'Infosys',
    'ADANIPOWER':'Adani Power',
    'DHFL':'DHFL',
    'AMARAJABAT':'Amara Raja Batteries',
    'APOLLOTYRE':'Apollo Tyres',
    'IDBI':'IDBI Bank',
    'EXIDEIND':'Exide Industries',
    'PNB':'Punjab National Bank',
    'VOLTAS':'Voltas',
    'SUNPHARMA':'SunPharma',
    'WELSPUNIND':'WelSpun India',
    'FORTIS':'Fortis',
    'VENKEYS':'Venkeys',
    'IDFC':'IDFC',
    'PVR':'PVR Cinemas',
    'NCC':'NCC',
    'RCOM':'Reliance Communications',
    'BAJAJELEC':'Baj Electric',
    'MARUTI':'Maruti',
    'M&M':'Mahindra',
    'HEROMOTOCO':'Hero Motors',
    'AJANTPHARM':'Ajanata Pharma',
    'JETAIRWAYS':'Jet',}
symbols, names =np.array(sorted(symbol_dict.items())).T
quotes=[]
for symbol in symbols:
    quotes.append(pd.read_csv(symbol+'.csv'))
for i,q in enumerate(quotes):
    quotes[i]=quotes[i][quotes[i].Series=='EQ']
Close_prices = np.vstack(q['Close Price'] for q in quotes)
```

In [2]:
```
Close_prices
```

Out[2]:
```
array([[  30.25,   32.85,   33.1 , ...,   40.95,   41.45,   38.45],
       [1633.5 , 1634.25, 1654.35, ..., 1064.4 , 1068.35, 1043.8 ],
       [ 933.4 ,  924.7 ,  937.7 , ...,  635.55,  634.9 ,  625.85],
       ...,
       [1169.7 , 1177.  , 1188.  , ..., 1930.1 , 1820.65, 1706.75],
       [ 431.85,  432.45,  430.2 , ...,  574.1 ,  580.05,  572.2 ],
       [  90.25,   90.65,   88.85, ...,   51.5 ,   52.6 ,   51.5 ]])
```

In [3]:
```
prices =pd.DataFrame(Close_prices)
```

In [4]:
```
prices_df=prices.T
prices_df.columns=symbols
```

In [5]:
```
prices_df.head()
```

Out[5]:

	ADANIPOWER	AJANTPHARM	AMARAJABAT	APOLLOTYRE	AXISBANK	BAJAJELEC	BAJFINANCE	CIPLA	DHFL	EXIDEIND	...	MARUTI	NC
0	30.25	1633.50	933.40	231.90	500.1	341.15	1332.95	569.00	431.40	245.80	...	6823.90	97.3
1	32.85	1634.25	924.70	234.40	501.5	347.00	1347.75	565.60	424.45	244.70	...	6953.95	100.4
2	33.10	1654.35	937.70	237.35	502.8	349.85	1324.80	562.35	429.00	243.20	...	6958.20	101.4
3	31.90	1633.40	912.10	232.65	492.0	334.10	1314.55	560.10	417.95	239.85	...	6831.05	97.0
4	32.40	1670.25	895.75	234.65	501.7	336.20	1289.15	564.95	404.20	238.15	...	6790.55	95.4

5 rows × 30 columns

QUERY 2-

6.2 Calculate average annual percentage return and volatility of all 30 stocks over a theoretical one year period

In [6]:
```
returns= prices_df.pct_change().mean()*252
returns=pd.DataFrame(returns)
returns.columns=['Returns']
returns['Volatility']=prices_df.pct_change().std()*sqrt(252)
returns
```

Out[6]:

	Returns	Volatility
ADANIPOWER	0.305273	0.610532
AJANTPHARM	-0.173891	0.332171
AMARAJABAT	-0.168654	0.267425
APOLLOTYRE	-0.064303	0.310445
AXISBANK	0.233194	0.277917
BAJAJELEC	0.316053	0.402414
BAJFINANCE	0.454332	0.321058
CIPLA	0.011817	0.254457
DHFL	-0.449581	0.662880
EXIDEIND	-0.051484	0.262985
FORTIS	-0.145989	0.421250
GAIL	-0.052551	0.338425
HDFC	0.137945	0.214557
HEROMOTOCO	-0.143545	0.239141
HINDALCO	0.040263	0.336753
IDBI	-0.296008	0.453787
IDFC	-0.245479	0.359303
INFY	-0.023617	0.418994
JETAIRWAYS	-0.481019	0.624854
M&M	-0.272401	0.440570
MARUTI	0.003937	0.225459
NCC	0.059373	0.445222
PNB	-0.239440	0.545390
PVR	0.111090	0.310198
RCOM	-0.881729	0.985366

	Returns	Volatility
SUNPHARMA	-0.192912	0.348418
TCS	0.075027	0.431617
VENKEYS	0.381711	0.627383
VOLTAS	0.191030	0.308346
WELSPUNIND	-0.208058	0.399071

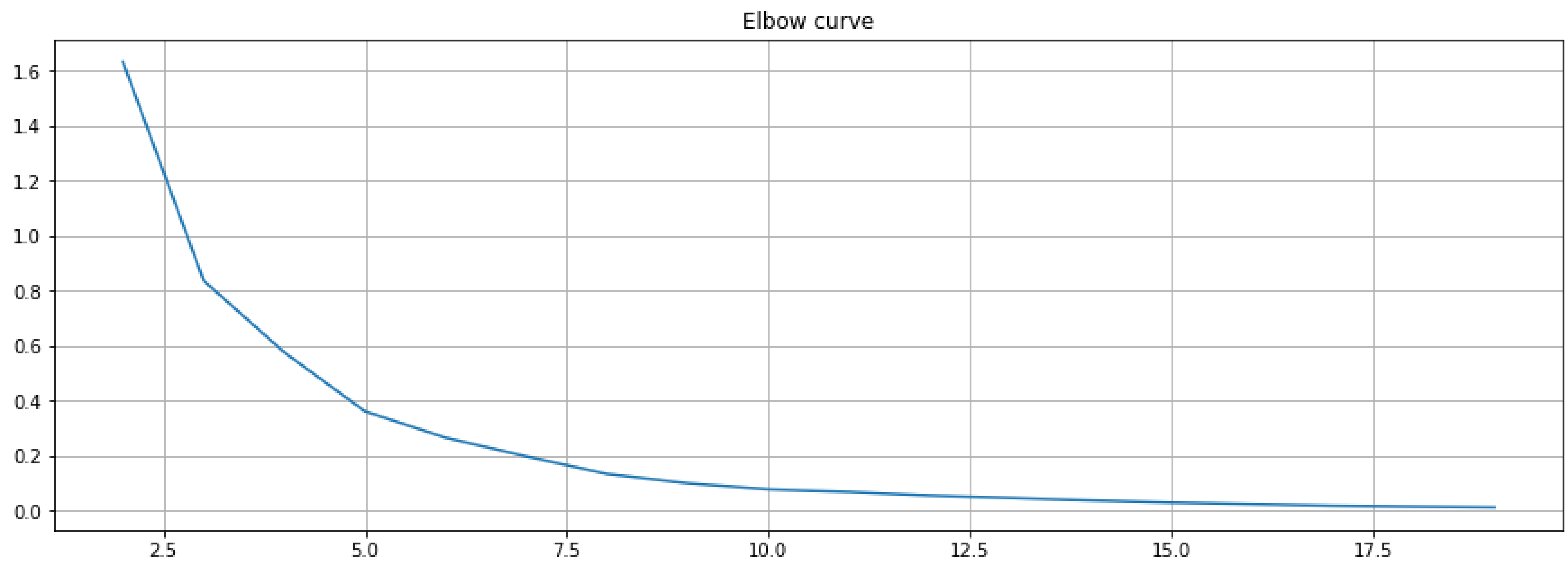
QUERY 3-

6.3 Cluster the 30 stocks according to their mean annual Volatilities and Returns using K-means clustering. Identify the optimum number of clusters using the Elbow curve method

In [7]:
```
data=np.asarray([np.asarray(returns['Returns']),np.asarray(returns['Volatility'])]).T
cleaned_data=np.where(np.isnan(data),0, data)
cleaned_data
```

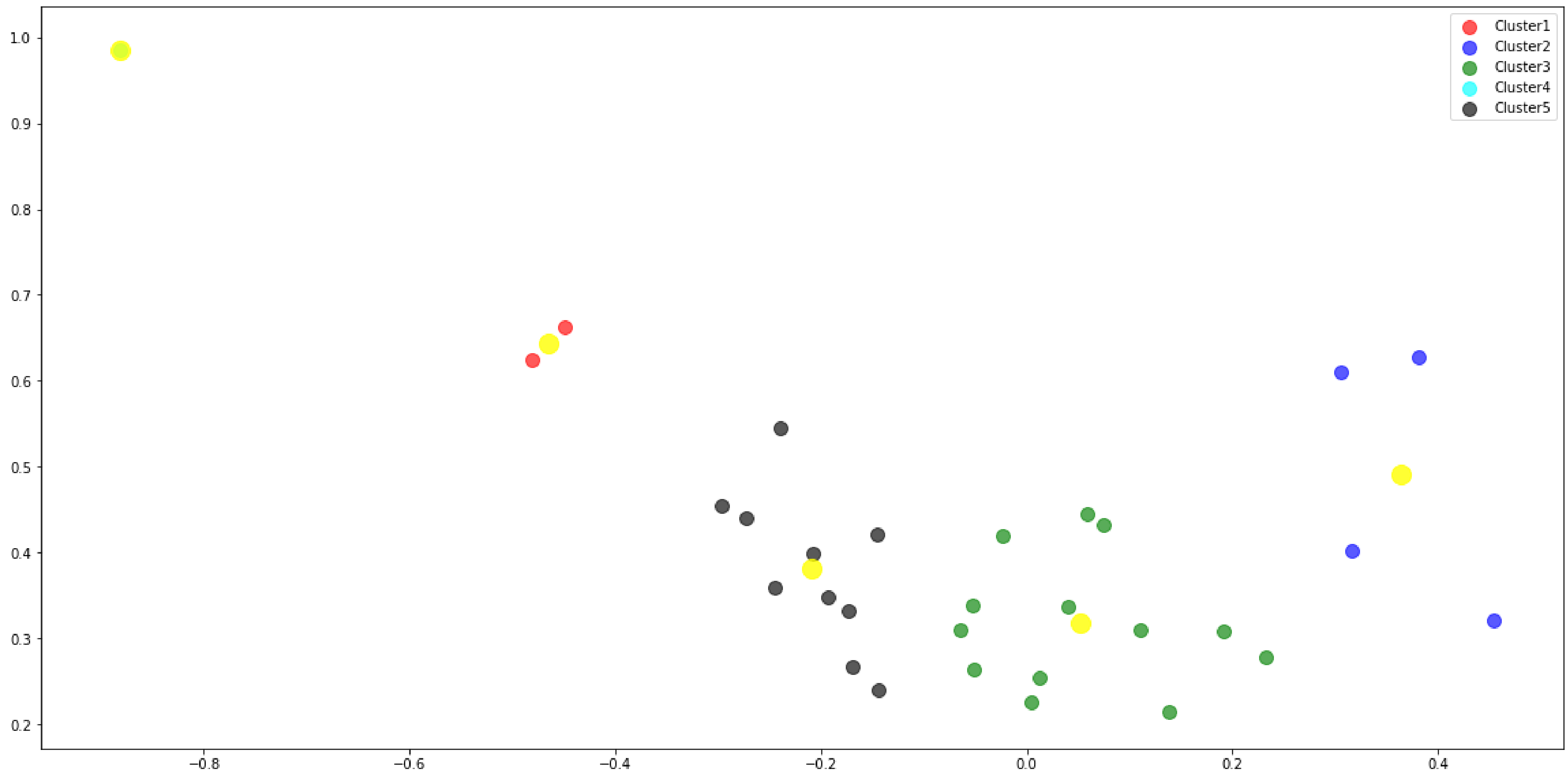
Out[7]:
```
array([[ 0.30527262,  0.6105322 ],
       [-0.17389139,  0.3321713 ],
       [-0.16865437,  0.26742539],
       [-0.06430261,  0.31044457],
       [ 0.23319431,  0.27791735],
       [ 0.31605318,  0.40241355],
       [ 0.4543322 ,  0.32105847],
       [ 0.01181652,  0.25445666],
       [-0.44958148,  0.66288009],
       [-0.05148401,  0.26298513],
       [-0.14598944,  0.42125042],
       [-0.0525509 ,  0.33842468],
       [ 0.13794545,  0.21455739],
       [-0.14354459,  0.23914078],
       [ 0.04026343,  0.3367533 ],
       [-0.29600826,  0.45378729],
       [-0.24547917,  0.35930325],
       [-0.02361697,  0.41899354],
       [-0.4810188 ,  0.62485368],
       [-0.27240099,  0.44056952],
       [ 0.00393675,  0.22545935],
       [ 0.05937295,  0.44522202],
       [-0.23944009,  0.54538954],
       [ 0.11108968,  0.31019811],
       [-0.88172863,  0.98536613],
       [-0.19291169,  0.34841844],
       [ 0.07502698,  0.43161712],
       [ 0.38171065,  0.62738334],
       [ 0.19103016,  0.30834649],
       [-0.20805802,  0.39907054]])
```

In [8]:
```
from sklearn.cluster import KMeans
X=cleaned_data
wcss=[]
for k in range(2,20):
    k_means=KMeans(n_clusters=k)
    k_means.fit(X)
    wcss.append(k_means.inertia_)
fig=plt.figure(figsize=(15,5))
plt.plot(range(2,20),wcss)
plt.grid(True)
plt.title('Elbow curve')
plt.show()
```



In [9]:
```
from scipy.cluster.vq import kmeans,vq
centroids,_=kmeans(cleaned_data,5)
idx,_=vq(cleaned_data,centroids)
data=cleaned_data
```

In [10]:
```
from matplotlib.pyplot import figure
figure(figsize=(20,10))
plt.scatter(X[idx==0,0],X[idx==0,1],s=100,c='red',label='Cluster1',alpha=0.65)
plt.scatter(X[idx==1,0],X[idx==1,1],s=100,c='blue',label='Cluster2',alpha=0.65)
plt.scatter(X[idx==2,0],X[idx==2,1],s=100,c='green',label='Cluster3',alpha=0.65)
plt.scatter(X[idx==3,0],X[idx==3,1],s=100,c='cyan',label='Cluster4',alpha=0.65)
plt.scatter(X[idx==4,0],X[idx==4,1],s=100,c='black',label='Cluster5',alpha=0.65)
plt.scatter(centroids[:,0],centroids[:,1],s=200,c='yellow',alpha=0.8)
plt.legend()
plt.show()
```



QUERY 4-

6.4 Prepare a separate Data frame to show which stocks belong to the same cluster

In [11]:
```
details=[(name, cluster) for name, cluster in zip(returns.index,idx)]
```

In [12]:
```
labels=['Stock Symbol','Cluster']
df=pd.DataFrame.from_records(details, columns=labels)
df.Cluster=df.Cluster+1
df.sort_values('Cluster')
```

Out[12]:

	Stock Symbol	Cluster
8	DHFL	1
18	JETAIRWAYS	1
0	ADANIPOWER	2
27	VENKEYS	2
5	BAJAJELEC	2
6	BAJFINANCE	2
26	TCS	3
23	PVR	3
21	NCC	3
20	MARUTI	3
17	INFY	3
28	VOLTAS	3
14	HINDALCO	3
4	AXISBANK	3
11	GAIL	3
9	EXIDEIND	3
7	CIPLA	3
12	HDFC	3
3	APOLLOTYRE	3
24	RCOM	4
1	AJANTPHARM	5
2	AMARAJABAT	5
25	SUNPHARMA	5
13	HEROMOTOCO	5
19	M&M	5
10	FORTIS	5

	Stock Symbol	Cluster
16	IDFC	5
15	IDBI	5
22	PNB	5
29	WELSPUNIND	5