Designing solution for Anomaly Detection

Import required libraries

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
```

Load Data

In [3]:

```
df = pd.read_csv('AnomalyData.csv')
```

In [4]:

```
df.columns
```

Out[4]:

```
Index(['State', 'state_code', 'data science', 'cluster analysis', 'college',
       'startup', 'entrepreneur', 'ceo', 'mortgage', 'nba', 'nfl', 'mlb',
       'fifa', 'modern dance', 'prius', 'escalade', 'subaru', 'jello', 'bbq',
       'royal family', 'obfuscation', 'unicorn', 'Extraversion',
       'Agreeableness', 'Conscientiousness', 'Neuroticism', 'Openness',
       'PsychRegions', 'region', 'division'],
      dtype='object')
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 30 columns):
State                    48 non-null object
state_code               48 non-null object
data science             48 non-null float64
cluster analysis         48 non-null float64
college                  48 non-null float64
startup                  48 non-null float64
entrepreneur             48 non-null float64
ceo                      48 non-null float64
mortgage                 48 non-null float64
nba                      48 non-null float64
nfl                      48 non-null float64
mlb                      48 non-null float64
fifa                     48 non-null float64
modern dance             48 non-null float64
prius                    48 non-null float64
escalade                 48 non-null float64
subaru                   48 non-null float64
jello                    48 non-null float64
bbq                      48 non-null float64
royal family             48 non-null float64
obfuscation              48 non-null float64
unicorn                  48 non-null float64
Extraversion             48 non-null float64
Agreeableness            48 non-null float64
Conscientiousness        48 non-null float64
Neuroticism              48 non-null float64
Openness                 48 non-null float64
PsychRegions             48 non-null int64
region                   48 non-null int64
```

```
division             48 non-null int64
dtypes: float64(25), int64(3), object(2)
memory usage: 11.4+ KB
```

Display all columns in the pandas dataset

In [7]:

```
pd.set_option('display.max_columns', None)
df.head()
```

Out[7]:

	State	state_code	data science	cluster analysis	college	startup	entrepreneur	ceo	mortgage	nba	r
0	Alabama	AL	-1.00	-0.13	1.10	-0.68	0.15	-0.73	1.53	-0.74	-1.
1	Arizona	AZ	-0.42	-0.73	-0.10	0.11	0.57	0.25	0.95	0.38	0.
2	Arkansas	AR	-0.66	-0.39	-0.64	-0.08	0.01	-0.66	-0.50	-0.71	-1.
3	California	CA	1.95	-0.62	-0.26	2.02	0.46	1.27	-0.97	1.46	-0.
4	Colorado	CO	0.34	0.00	-0.61	1.49	0.05	0.33	1.38	-0.80	1.

In [9]:

```
df.describe()
```

Out[9]:

	data science	cluster analysis	college	startup	entrepreneur	ceo	mortgage	nba
count	48.000000	48.000000	48.000000	48.000000	48.000000	48.000000	48.000000	48.000000
mean	-0.000833	-0.012500	0.060625	0.013542	0.031667	-0.030000	-0.026250	-0.025000
std	0.971397	0.972073	0.982906	1.023726	0.974069	0.910588	0.984956	0.998769
min	-1.270000	-1.700000	-1.960000	-1.830000	-1.940000	-1.380000	-2.400000	-1.720000
25%	-0.662500	-0.730000	-0.617500	-0.650000	-0.607500	-0.675000	-0.732500	-0.855000
50%	-0.235000	-0.135000	-0.050000	-0.055000	0.070000	-0.115000	-0.005000	-0.130000
75%	0.352500	0.412500	0.747500	0.332500	0.485000	0.420000	0.537500	0.612500
max	2.730000	2.910000	2.360000	2.630000	2.740000	2.460000	1.890000	2.120000

In [10]:

```
df.corr()
```

Out[10]:

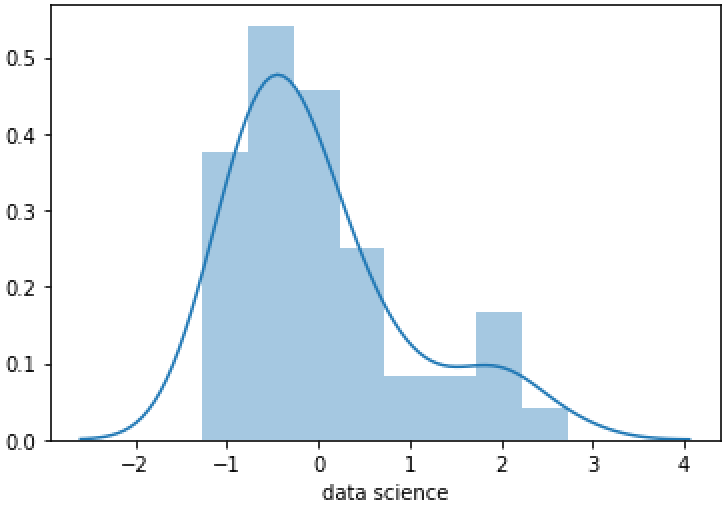
	data science	cluster analysis	college	startup	entrepreneur	ceo	mortgage	
data science	1.000000	0.515322	0.234620	0.571125	0.244883	0.827573	0.232255	0
cluster analysis	0.515322	1.000000	0.306682	0.417653	0.338971	0.423187	0.422482	-0
college	0.234620	0.306682	1.000000	0.034665	0.023155	0.265010	0.084482	-0
startup	0.571125	0.417653	0.034665	1.000000	0.144738	0.479108	0.408186	-0
entrepreneur	0.244883	0.338971	0.023155	0.144738	1.000000	0.466465	0.534047	0
ceo	0.827573	0.423187	0.265010	0.479108	0.466465	1.000000	0.383807	0

	data science	cluster analysis	college	startup	entrepreneur	ceo	mortgage	
mortgage	0.232255	0.422482	0.084482	0.408186	0.534047	0.383807	1.000000	0
nba	0.339015	-0.027146	-0.102293	-0.033708	0.489639	0.543882	0.073800	1
nfl	-0.010459	0.278204	-0.084217	0.123187	0.084266	-0.018916	0.213263	-0
mlb	0.456911	0.267177	0.306460	0.232927	0.020219	0.586652	0.069867	0
fifa	0.664236	0.126510	0.057498	0.105295	0.214568	0.638202	0.002059	0
modern dance	0.413652	0.370242	0.043303	0.496930	0.255552	0.340654	0.386186	0
prius	0.472205	0.069603	0.075744	0.607550	-0.094345	0.351721	0.140359	0
escalade	-0.391230	-0.350360	-0.187946	-0.424670	0.243247	-0.216864	-0.039584	0
subaru	0.109452	0.235362	0.152443	0.573378	-0.392856	0.021788	0.071206	-0
jello	-0.348133	0.039839	-0.017704	0.101297	-0.118478	-0.400209	-0.156475	-0
bbq	-0.149987	-0.421174	-0.095490	-0.195506	0.268836	-0.039982	0.012936	0
royal family	0.367900	0.521753	0.652759	0.276730	-0.032940	0.342438	0.312534	-0
obfuscation	0.573107	0.533919	0.118185	0.802191	0.042877	0.408758	0.459819	-0
unicorn	-0.006891	0.066124	-0.037327	0.622563	-0.250256	-0.116152	0.144367	-0
Extraversion	-0.150531	-0.142460	0.066038	-0.313433	0.431629	0.097022	0.060950	0
Agreeableness	-0.330719	-0.188341	-0.082737	-0.107036	0.024527	-0.395411	-0.116628	-0
Conscientiousness	-0.385545	-0.292934	-0.200468	-0.353204	0.263212	-0.289016	0.034910	0
Neuroticism	0.026570	0.158632	0.341316	-0.140783	-0.200065	0.077181	-0.125185	-0
Openness	0.526476	0.043046	-0.038550	0.212303	0.097779	0.499034	0.171571	0
PsychRegions	0.525460	0.355168	0.286159	0.374109	0.010911	0.485397	0.143261	0
region	-0.241637	-0.227701	-0.572893	-0.010380	0.049708	-0.364362	0.015506	-0
division	-0.237509	-0.305056	-0.570083	-0.001698	-0.010349	-0.371236	-0.052207	0

Do Exploratory Data Analysis:

```
In [17]: sns.distplot(df['data science'])
```

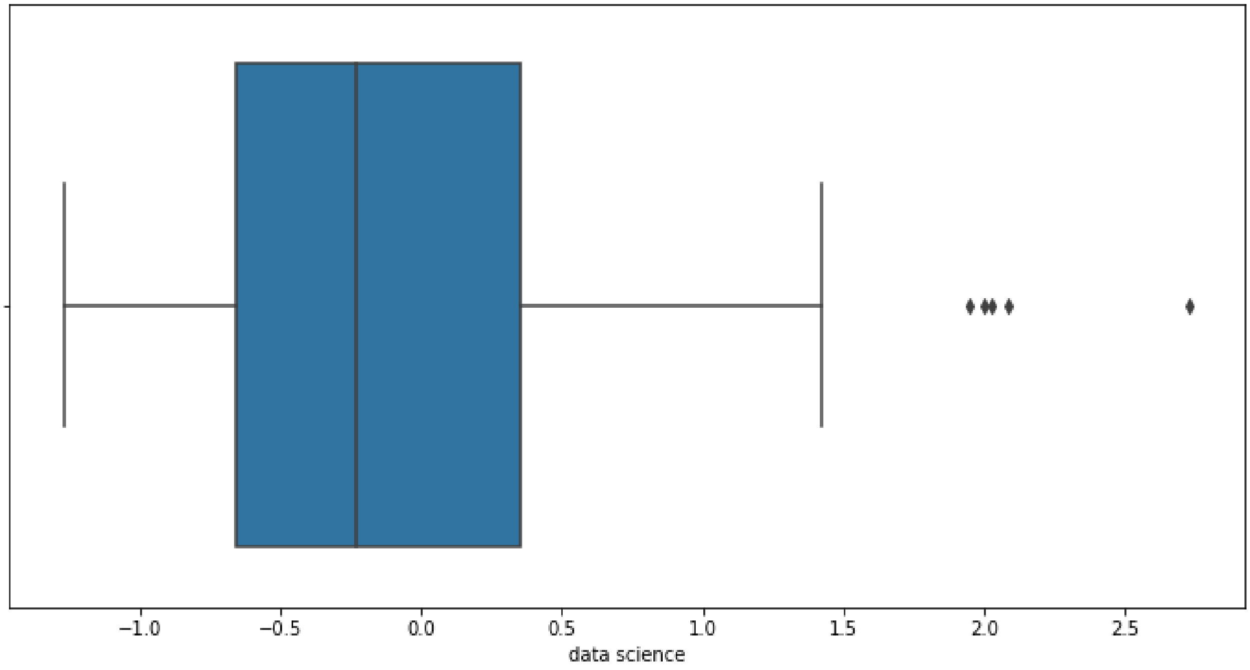
```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x259be18ef88>
```



Univariate outliers

In [24]:
```
plt.figure(figsize=(12,6))
sns.boxplot(df['data science'])
```

Out[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x259be33fb88>
```



As seen from above we can easily identify univariate outliers for data science based on the above plot

In [25]:
```
df.columns
```

Out[25]:
```
Index(['State', 'state_code', 'data science', 'cluster analysis', 'college',
       'startup', 'entrepreneur', 'ceo', 'mortgage', 'nba', 'nfl', 'mlb',
       'fifa', 'modern dance', 'prius', 'escalade', 'subaru', 'jello', 'bbq',
       'royal family', 'obfuscation', 'unicorn', 'Extraversion',
       'Agreeableness', 'Conscientiousness', 'Neuroticism', 'Openness',
       'PsychRegions', 'region', 'division'],
      dtype='object')
```

Creating custom boxplot for Outliers based on Quantile value & IQR:

In [26]:

```
variable = 'data science'
```

Using state code to label the outliers and then removing it to keep only the outlier variables

In [29]:

```
state_code = df['state_code']
data = df.loc[:, 'data science': 'Openness']
```

In [31]:

```
data.columns
```

Out[31]:

```
Index(['data science', 'cluster analysis', 'college', 'startup',
       'entrepreneur', 'ceo', 'mortgage', 'nba', 'nfl', 'mlb', 'fifa',
       'modern dance', 'prius', 'escalade', 'subaru', 'jello', 'bbq',
       'royal family', 'obfuscation', 'unicorn', 'Extraversion',
       'Agreeableness', 'Conscientiousness', 'Neuroticism', 'Openness'],
      dtype='object')
```

Getting Quantile values and Inter Quantile Range

In [32]:

```
QV1 = data[variable].quantile(0.25)
QV2 = data[variable].quantile(0.50)
QV3 = data[variable].quantile(0.75)

qv_limit = 1.5 * (QV3-QV1)
```

In [33]:

```
qv_limit
```

Out[33]:

```
1.5225
```

In [35]:

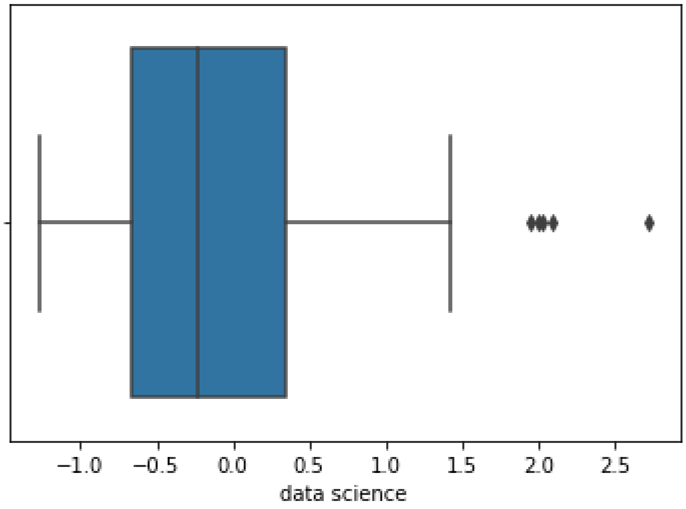
```
outlier_range = (data[variable]> QV3 +qv_limit) | (data[variable]<QV1-qv_limit)
outlier_data = data[variable][outlier_range]
outlier_name = state_code[outlier_range]
```

In [37]:

```
sns.boxplot(data[variable])
```

Out[37]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x259c145e3c8>
```



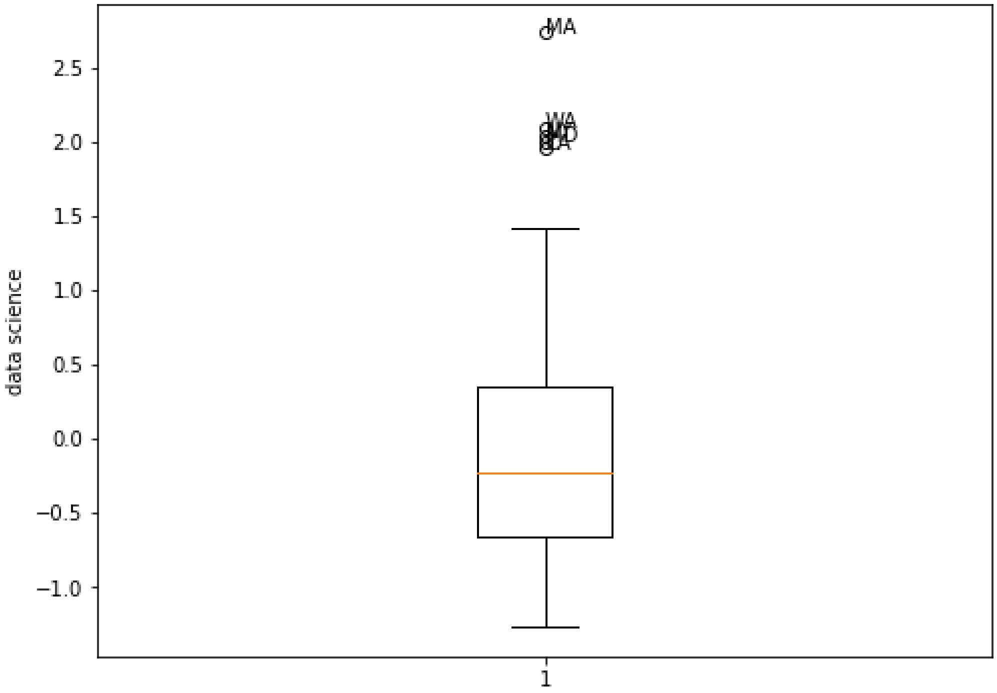
Plotting the data with states

In [39]:

```
import pylab
```

```
In [44]: fig = pylab.figure(figsize=(8,6))
         ax = fig.add_subplot(1, 1, 1)
         for name, y in zip(outlier_name, outlier_data):
             ax.text(1, y, name)
         ax.boxplot(data[variable])
         ax.set_ylabel(variable)
```

```
Out[44]: Text(0, 0.5, 'data science')
```



Multivariate Outlier detection using One Class Support Vector algorithm:

```
In [45]: from sklearn.svm import OneClassSVM
```

```
In [46]: ocsvm = OneClassSVM(nu=0.25, gamma=0.05)
```

List the names of outlier states based on One Class SVM algorithm

```
In [49]: ocsvm.fit(data)
```

```
Out[49]: OneClassSVM(cache_size=200, coef0=0.0, degree=3, gamma=0.05, kernel='rbf',
                     max_iter=-1, nu=0.25, random_state=None, shrinking=True, tol=0.001,
                     verbose=False)
```

```
In [50]: state_code[ocsvm.predict(data) ==-1]
```

```
Out[50]: 7     FL
         13    KS
         14    KY
         16    ME
         17    MD
         18    MA
         19    MI
         20    MN
         21    MS
         24    NE
         25    NV
         27    NJ
```

```
28    NM
30    NC
33    OK
39    TN
43    VA
Name: state_code, dtype: object
```

In [ ]:

```

```