In [1]:
```
# k-means clustering
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import KMeans
from matplotlib import pyplot
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import numpy as np
import numpy as np
from sklearn.cluster import KMeans
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
# the combined data
data_folder = './nhanes_input_data/'
# import the CSV as a pandas dataframe
df = pd.read_csv( data_folder + '0_dietaryIntakeDataForClassificationAndAnalysisData.cs
df.shape
```

Out[3]:
```
(193805, 87)
```

In [4]:
```
df.head(5)
```

Out[4]:

	RIDAGEYR_Age_in_years_at_screening	URDACT_Albumin_creatinine_ratio_mg_g	DataYear	SEQN - Respondent sequence number
0	53.0	3.0	2017-2018	95405.0
1	53.0	3.0	2017-2018	95405.0
2	53.0	3.0	2017-2018	95405.0
3	53.0	3.0	2017-2018	95405.0
4	53.0	3.0	2017-2018	95405.0

5 rows × 87 columns

In [5]:
```
# parameters to be used for KMeans clustring: centres
# X and/or kdf will have only features we want to create cluster around
kdf = df[

    [
        'RIDAGEYR_Age_in_years_at_screening'
        ,'URDACT_Albumin_creatinine_ratio_mg_g'
    ]
]
X = kdf
X[:5]
```

Out[5]:

	RIDAGEYR_Age_in_years_at_screening	URDACT_Albumin_creatinine_ratio_mg_g
0	53.0	3.0
1	53.0	3.0
2	53.0	3.0
3	53.0	3.0
4	53.0	3.0

In [6]:
```
# ref: internet (not my code, using as a library)
def clean_dataset(df):
    assert isinstance(df, pd.DataFrame), "df needs to be a pd.DataFrame"
    df.dropna(inplace=True)
    indices_to_keep = ~df.isin([np.nan, np.inf, -np.inf]).any(1)
    return df[indices_to_keep].astype(np.float64)
```

In [7]:
```
# X has the features to cluster around (centres: Age, ACR). df has the complete data
# after clustering is done using features in X, we find positions (index) for each data
# in a cluster then we use those index positions to cluster the data from df
X.shape, df.shape
```

Out[7]:
```
((193805, 2), (193805, 87))
```

In [8]:
```
X = clean_dataset(X)
```

In [9]:
```
# define the model
model = KMeans(n_clusters = 10) #,random_state=0, n_init="auto"

# fit the model
model.fit(X)
#model.labels_
```

Out[9]:
```
KMeans(n_clusters=10)
```

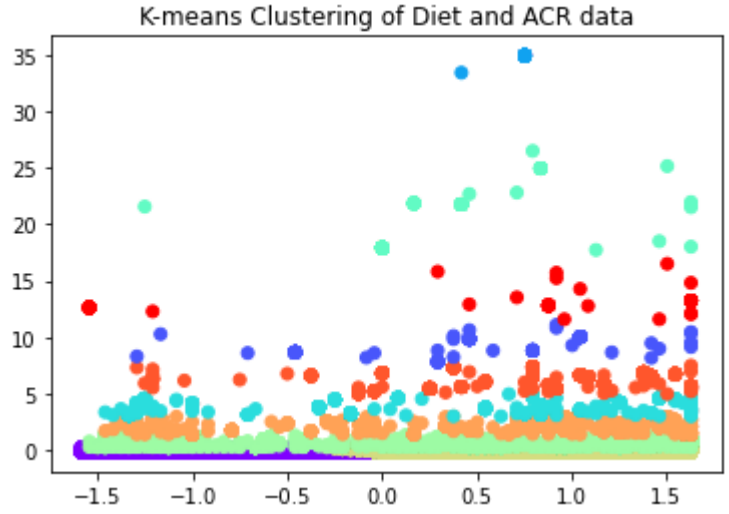
In [10]:
```
# Create csv files with the cluster daya
# One csv for one Cluster
```

In [11]:
```
howManyClusters = 10
for clusterId in range (howManyClusters):
    ind_list = np.where(model.labels_ == clusterId )[0]
    cluster = df.iloc[ind_list]
    cluster.to_csv('./nhanes_output_data/classifiedGroups/kmeanscluster/cluster-'
                   + str(clusterId) + '.csv');
```

In [12]:
```
model.cluster_centers_
```

Out[12]:
```
array([[  18.87596942,   12.04466503],
       [  51.67164179, 2496.77649254],
       [  58.87096774, 9398.2316129 ],
       [  50.85535466, 1061.83630042],
       [  49.14736842, 5597.96115789],
       [  49.44935831,  203.29598508],
       [  60.84809966,   15.0886079 ],
       [  57.20701513,  620.4038033 ],
       [  59.28975741, 1682.01998652],
       [  52.31707317, 3538.97792683]])
```

In [13]:
```
# Scatter plot to see each cluster points visually
std_data = StandardScaler().fit_transform(X)
plt.scatter(std_data[:,0], std_data[:,1], c = model.labels_, cmap = "rainbow")
plt.title("K-means Clustering of Diet and ACR data")
plt.show()
```



References:

print("Shape of cluster:", model.cluster*centers*.shape)

https://stackoverflow.com/questions/50297142/g
cluster-points-after-kmeans-in-a-list-format

https://machinelearningmastery.com/clustering-
algorithms-with-python/

https://stackoverflow.com/questions/50297142/g
cluster-points-after-kmeans-in-a-list-format

https://scikit-
learn.org/stable/modules/generated/sklearn.clus

https://datascience.stackexchange.com/question
k-means-clustering-over-multiple-columns

https://scikit-
learn.org/stable/modules/generated/sklearn.clus

> from sklearn.cluster import KMeans import numpy as np
> X = np.array([[1, 2], [1, 4], [1, 0], ... [10, 2], [10, 4], [10, 0]])
> kmeans = KMeans(n_clusters=2, random_state=0,
> n*init="auto").fit(X) kmeans.labels* array([1, 1, 1, 0, 0, 0],
> dtype=int32) kmeans.predict([[0, 0], [12, 3]]) array([1, 0],
> dtype=int32) kmeans.cluster*centers* array([[10., 2.], [1.,
> 2.]])

In [ ]: