Computational Statistics Final

Liam Dillingham

April 29, 2019

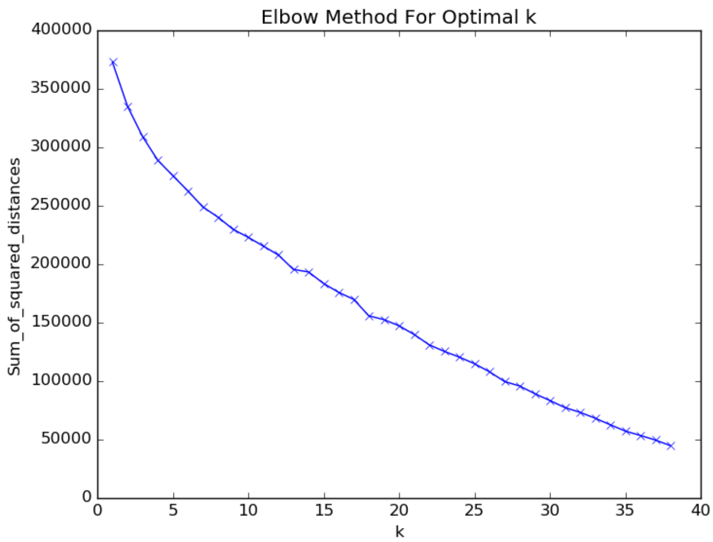
1 R Code for Data Preprocessing

```
# Biplot great for visualization
library(devtools)
install_github("vqv/ggbiplot")
library(ggbiplot)

library(ISLR)
nci.labs=NCI60$labs   ## NCI data
nci.data=NCI60$data ## Labels
dim(nci.data)

# Scale data to have mean=0 and sd=1 (each gene on same scale)
sd.data = scale(nci.data) #

nci.pca = prcomp(nci.data, center = T, scale. = T)
summary(nci.pca)

write.csv(nci.pca$x, file = "nci60.csv", row.names = F)
```

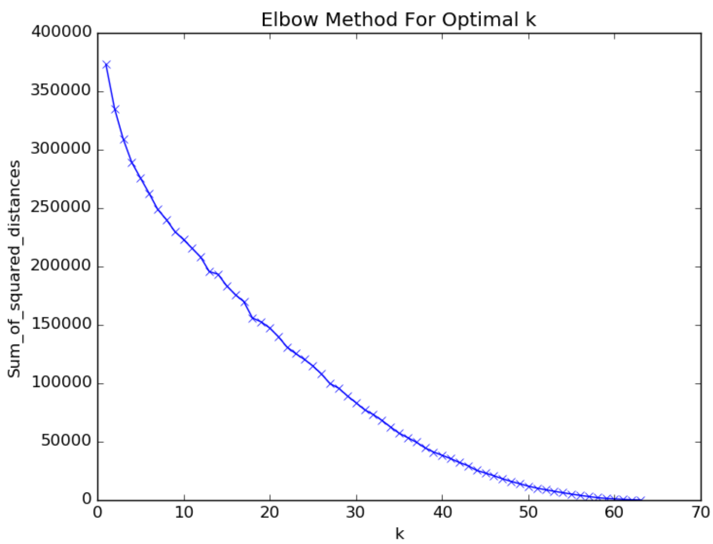
2 Python Code

```
import pandas as pd
import numpy as np

import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import SpectralClustering, AgglomerativeClustering, DBSCAN, KMe
from sklearn.manifold import TSNE, MDS

plt.style.use('classic')

# cancer_data = pd.read_csv('./nci60.csv') # data is already standardized
cancer_data = pd.read_csv('./nci60.csv')
```

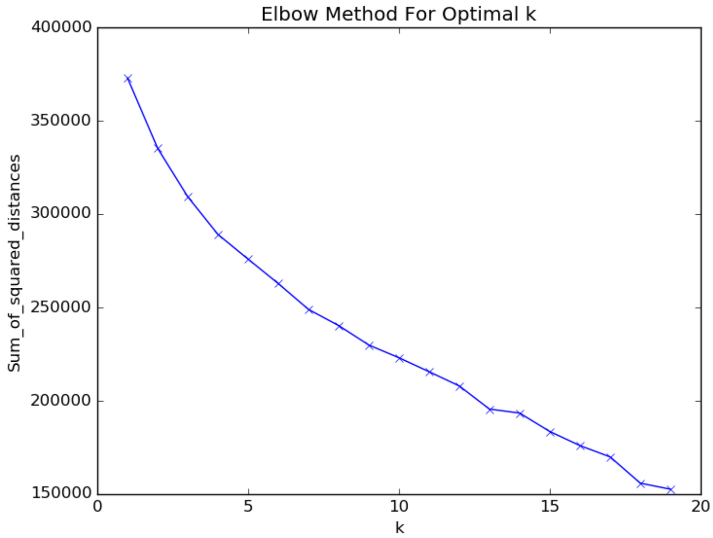
```
# We determined that we need 39 PCs to get >85% of the variance
pcs = []
for i in range(39):
    pcs.append('PC{}'.format(i+1))
cancer_subset = cancer_data[pcs]

sum_of_squared_distances = []
K = range(1,30)
for k in K:
    km = KMeans(n_clusters=k, n_init = 50, random_state = 0)
    km = km.fit(cancer_subset)
    sum_of_squared_distances.append(km.inertia_)

plt.plot(K, sum_of_squared_distances, 'bx-')
plt.xlabel('k')
plt.ylabel('Sum of squared distances')
plt.title('Elbow Method For Optimal k')
plt.show()
```



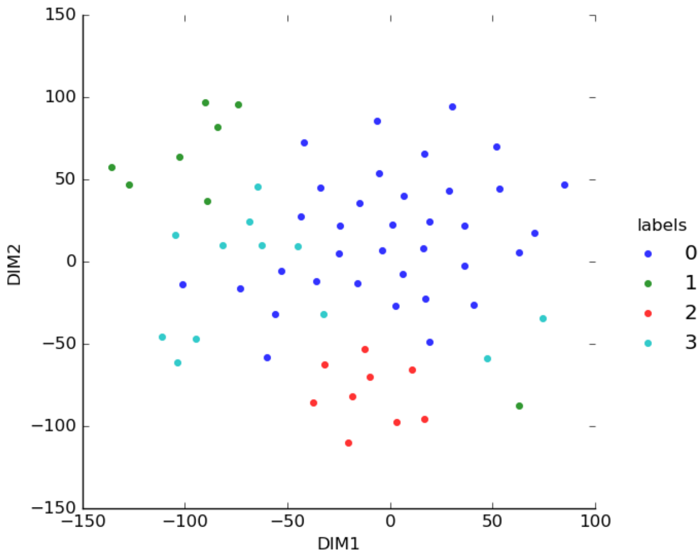
(a) Optimal k for maximum $k = 50$



(b) Optimal k for maximum $k = n$



(a) Optimal k for maximum $k = 20$



(a) Optimal k for maximum $k = 20$