ADM deliverable 3 by Diego Garcia-Olano

1 2010 US Bike Commuter data, clustering in python

We wish to study a Bike Commuters dataset for 374 cities and metropolitan areas of the United States from 2010, and enrich it with socio-economic and voting data we have for the US from prior deliverables. The datasets can be found at the League of American Bicyclists website, http://bikeleague.org/reports.

2 Merging Datasets

The initial 2010 commuters dataset is city based whereas the socio-economic voting US dataset we have from 2012 is county based so our first task is to merge the socio-economic data into the commuters dataset by doing the corresponding city-county mapping. This process was relatively straightforward for most areas as its a one to one mapping, but for about a fifth of the dataset, we find a city that lies across multiple counties. Those cases were done by hand and corresponded to either a "most reprsentative" county being selected for the city if the majority of it lay within a county or otherwise an average of the counties for a given city where used. By the end of this process, our dataset is as follows:

	City	State	Population	Total Workers	% Bicycle Commuters	Number of Commuters
1	Bloomington : 3	California: 97	Min. : 64299	Min. : 22928	Min. : 0.100	Min. : 32.0
2	Arlington : 2	Texas : 30	1st Qu.: 89938	1st Qu.: 40634	1st Qu.: 0.300	1st Qu.: 174.0
3	Aurora : 2	Florida : 26	Median : 124722	Median : 56907	Median : 0.600	Median : 389.5
4	Columbia : 2	Illinois : 14	Mean : 242273	Mean : 109332	Mean : 1.093	Mean : 1118.5
5	Columbus : 2	Colorado : 13	3rd Qu.: 210651	3rd Qu.: 97321	3rd Qu.: 1.200	3rd Qu.: 899.2
6	Fayetteville: 2	Arizona : 11	Max. :8184899	Max. :3615588	Max. :22.100	Max. :27917.0
7	(Other) :361	(Other) :183				

	%of female commuters	%of male commuters	county	dem	rep	isdem
1	Min. : 0.00	Min. : 0.00	Los Angeles: 22	Min. : 7114	Min. : 11647	no :105
2	1st Qu.: 0.00	1st Qu.: 65.00	Orange : 13	1st Qu.: 70152	1st Qu.: 56212	yes:269
3	Median : 19.00	Median : 81.00	Alameda : 8	Median : 149110	Median :109820	
4	Mean : 23.17	Mean : 76.83	Maricopa : 8	Mean : 300239	Mean :187712	
5	3rd Qu.: 35.00	3rd Qu.:100.00	Broward : 7	3rd Qu.: 387978	3rd Qu.:245034	
6	Max. :100.00	Max. :100.00	Clark : 7	Max. :1672164	Max. :699600	
7			(Other) :309			

	% with graduate degrees	median earnings	education index	income index
1	Min. : 4.20	Min. :16041	Min. :2.710	Min. :0.690
2	1st Qu.: 8.90	1st Qu.:27020	1st Qu.:4.603	1st Qu.:4.305
3	Median :10.35	Median :29531	Median :5.040	Median :4.920
4	Mean :11.65	Mean :30403	Mean :5.154	Mean :5.002
5	3rd Qu.:13.40	3rd Qu.:33068	3rd Qu.:5.718	3rd Qu.:5.710
6	Max. :36.70	Max. :56148	Max. :8.110	Max. :9.380

	white	african american	native american	asian american	other	latino
1	Min. : 7.80	Min. : 0.30	Min. : 0.1000	Min. : 0.600	Min. : 0.200	Min. : 1.20
2	1st Qu.:42.30	1st Qu.: 3.20	1st Qu.: 0.2000	1st Qu.: 2.300	1st Qu.: 2.000	1st Qu.: 7.30
3	Median :55.30	Median : 8.30	Median : 0.3000	Median : 4.100	Median : 2.400	Median :17.15
4	Mean :56.09	Mean :12.07	Mean : 0.5874	Mean : 6.653	Mean : 2.672	Mean :21.94
5	3rd Qu.:72.25	3rd Qu.:16.75	3rd Qu.: 0.5000	3rd Qu.: 8.600	3rd Qu.: 3.200	3rd Qu.:32.00
6	Max. :92.30	Max. :63.30	Max. :26.5000	Max. :43.000	Max. :27.700	Max. :90.60

3 Clustering in Python with Ipython, scipy, numpy, and pandas

```
For this phase, we load and examine the data in python.
We decide to use an IPython notebook to see how it compares against R and Julia.
To begin the session, from the commandline we run:  ipython notebook --gui wx --pylab
This opens a prompt in our web browser which looks familiar to the other interactive
statistics software we've seen in the past deliverables.

We read in our csv file and look at some sample statistics and commands.
> import pandas as pd
> bikeData = pd.read_csv('proj3/bike-election-data-final.csv')

> print(bikeData.columns)      # like names() in R
> bikeData.describe()   # like summary() in R
> bikeData.info()       # like str() in R, shows missing values in row 1
> bikeData.ix[1]['county']    #how to get a row by an index
> bikeData['State'].value_counts()
> commuters = bikeData.groupby('Percentage.of.Bicycle.Commuters','State')
> commuters = bikeData.groupby('State')

We see there are a few missing values, and find we need to remove the
first row which is an overall summary of the dataset and uninteresting.

> bikeData = bikeData[ bikeData.City != "United States"]

We now wish to perform a Principle Component Analysis on the dataset so that we can
move the data into a Factorial space which removes noise and helps for clustering.
We only take numeric columns, and standardize our data first.

> dataset = bikeData[bikeData.columns[[3,4,5,6,8,9,11,12,15,16,18,19,20,21,22,23,24,25,26,27]]]
> mu = dataset.mean()
> centered = dataset - mu
> data_s = centered / dataset.std()

PCA by diagonalizing correlation matrix
> nrow = data_s.shape[0]
> N = diag(ones(nrow) * 1/nrow)
> correlation = np.dot(np.dot(data_s.T,N), data_s)
> correlation.shape  #20x20
> eigvals, eigvecs = np.linalg.eig(correlation)
> eigvals.shape       #20,1
> eigvecs.shape       #20,20
> Psi = np.dot(data_s, eigvecs)
> Phi = np.dot(eigvecs,diag(sqrt(eigvals)))
> (eigvals[0] + eigvals[1]) / sum(eigvals)    #how much variance is describe in first plane

Plot variable space
> import matplotlib.pyplot as plt
> fig, ax = plt.subplots()
> plt.axis([-.5, 1.1, -1, 1])
> ax2 = plt.axes()
> c = 0
> for i in Phi[:,[0,1]]:
>    ax2.arrow(0, 0, i[0], i[1], head_width=.03, head_length=.03, fc='k', ec='k')
>    ax2.annotate(data_s.columns[c],(i[0],i[1]),fontsize=10)
>    c = c + 1
>
> ax2.plot([0, 0], [-10, 10], color='gray', linestyle='-', linewidth=1)
> ax2.plot([-10, 10], [0, 0], color='gray', linestyle='-', linewidth=1)
```

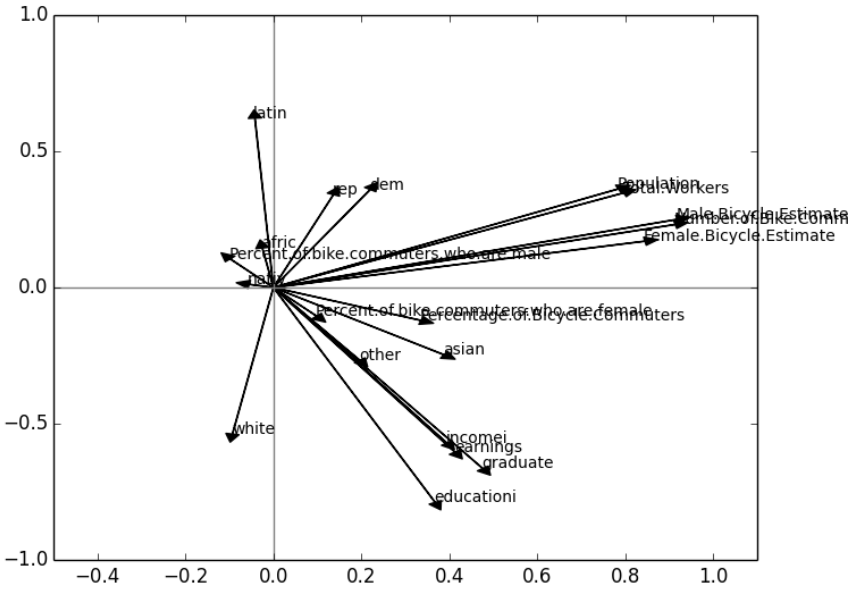
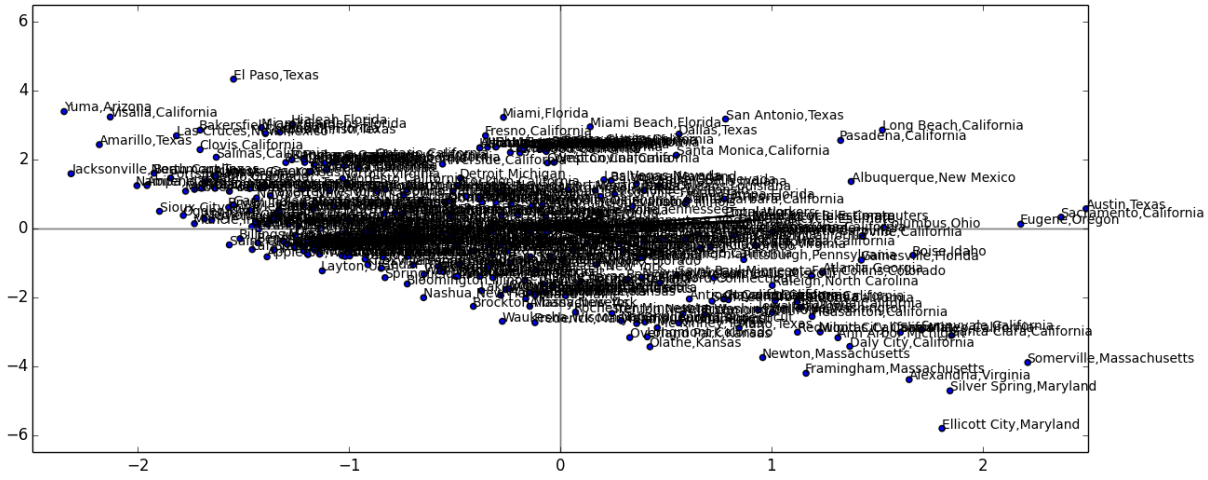
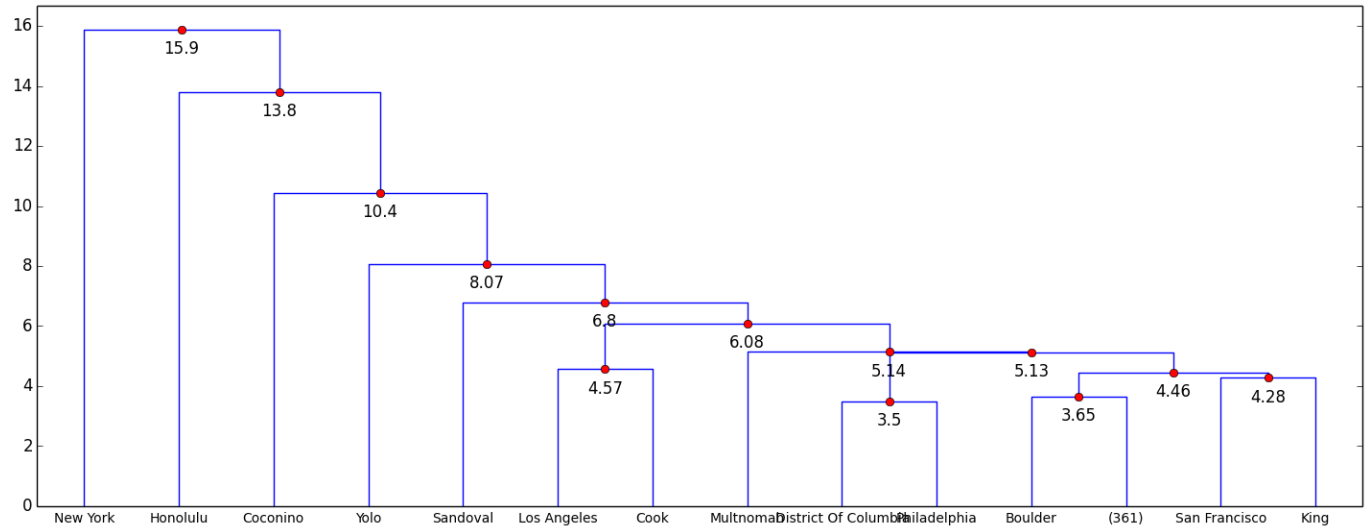


Figure 1: The first two planes describe 40.08% variance of the data:

```
Plot individual space
> plt.axis([-2.5, 2.5, -6.5, 6.5])
> ax3 = plt.axes()
> f = plt.gcf()
> f.set_size_inches(24, 24);
> c = 1
> for i in Psi[:,[0,1]]:
>    ax3.scatter(i[0],i[1])
>    lbl = bikeData.ix[c]['City'] +","+bikeData.ix[c]['State']
>    ax3.annotate(lbl,(i[0],i[1]),fontsize=10)
>    c = c + 1
>
> ax3.plot([0, 0], [-10, 10], color='gray', linestyle='-', linewidth=1)
> ax3.plot([-10, 10], [0, 0], color='gray', linestyle='-', linewidth=1)
```

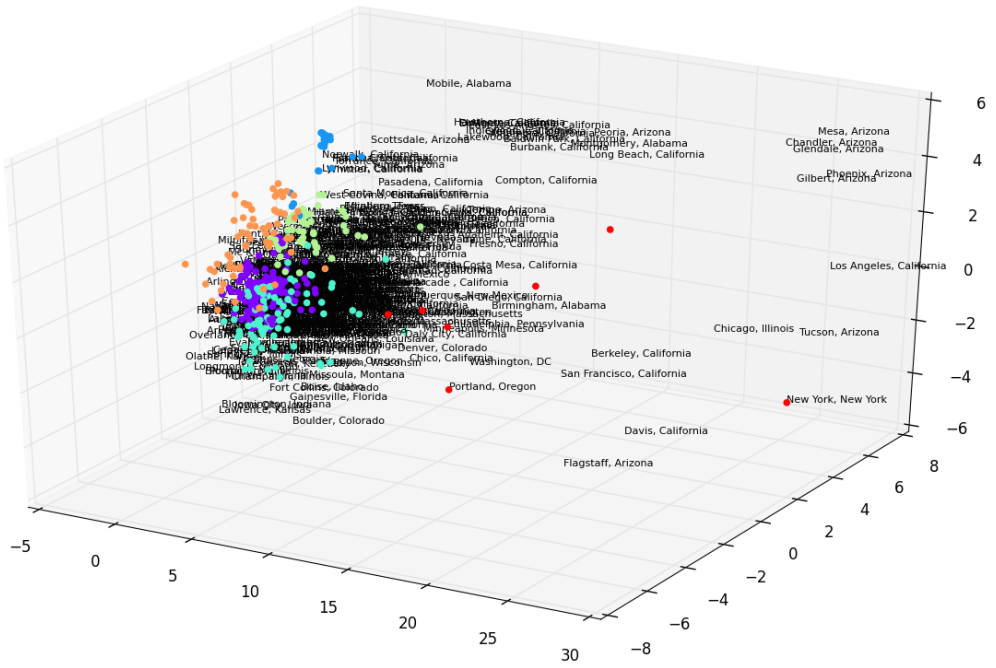


```
Perform a hierarchical clustering  on factorial space, Psi.
We do this to see how many clusters would be a good starting amount for k-means.
> from scipy.spatial.distance import pdist, squareform
> from scipy.cluster.hierarchy import linkage, dendrogram

#calculate a distance matrix
> distMatrix = pdist(Psi)
> linkageMatrix = linkage(distMatrix)
> actlabels = pd.Categorical.from_array(bikeData["county"])
> ddata = dendrogram(linkageMatrix, color_threshold=.95, leaf_font_size=10,
                   labels=actlabels, p=9, truncate_mode="level")

>for i, d in zip(ddata['icoord'], ddata['dcoord']):
>            x = 0.5 * sum(i[1:3])
>            y = d[1]
>            plt.plot(x, y, 'ro')
>            plt.annotate("%.3g" % y, (x, y), xytext=(0, -8),
                         textcoords='offset points',
                         va='top', ha='center')

> f = gcf()
> f.set_size_inches(20, 12);
```




```
We decide the number of classes present in our data and calculate the corresponding centroids.
In the above code, we set a cutoff (p=9) for sake of interpretability of the dendogram because
setting the value to a larger value (or unlimited), the visualization becomes very large
and there is little differentiation between splits after out cutoff.

From analysis, therefore it looks like there is a reasonable cutoff around 6 so we'll go with that.

Perform a k-means algorithm taking as seeds the centroids previously calculated

> from scipy.cluster.vq import kmeans,vq
> centroids,_ = kmeans(Psi,6 )       # computing K-Means with K = 6 (6 clusters)
> idx,_ = vq(Psi,centroids)          # assign each sample to a cluster
> colors = cm.rainbow(np.linspace(0, 1, 6))    # #plot original Psi now with cluster colors

> from mpl_toolkits.mplot3d import Axes3D
> from mpl_toolkits.mplot3d import proj3d
> ax3 = plt.axes(projection='3d')
```

```
> f = plt.gcf()
> f.set_size_inches(24, 24);

> c = 0
> for i in Psi[:,[0,1,2]]:
>     ax3.scatter(i[0],i[1],i[2],color=colors[idx[c]])
>     c = c + 1
>     lbl = bikeData.ix[c]['City'] + ", "+bikeData.ix[c]['State']
>     x2, y2, _ = proj3d.proj_transform(i[0],i[1],i[2], ax3.get_proj())
>     ax3.annotate(lbl,(x2,y2),fontsize=8)
```



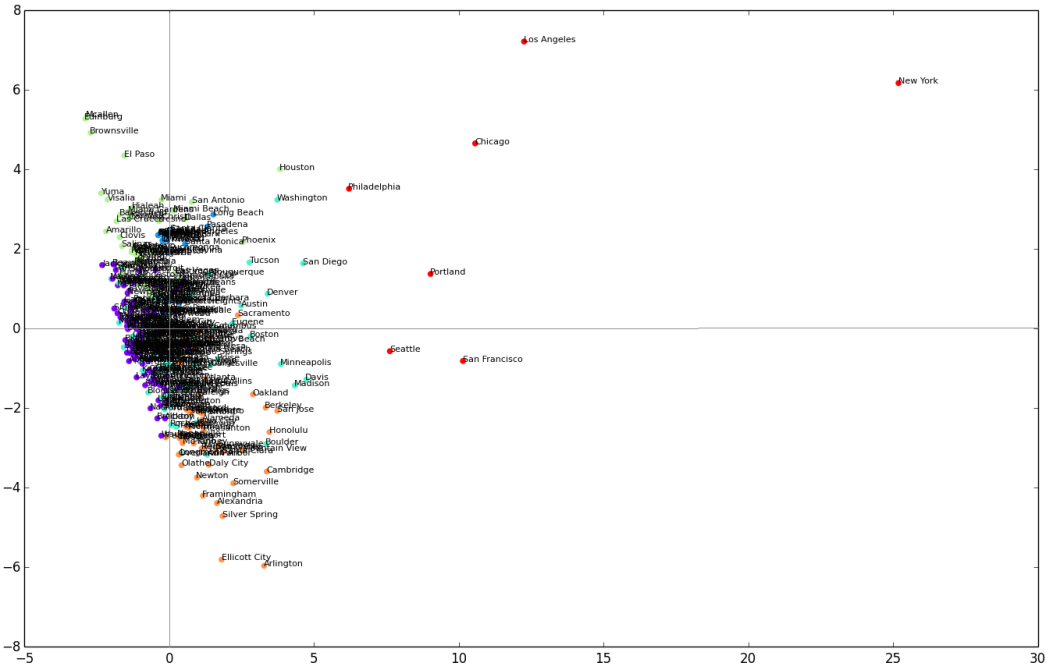
The 3d plot is actually a little hard to read so we go with a 2d representation.

```
> ax2 = plt.axes()
> f = plt.gcf(); f.set_size_inches(24, 24);
> c = 0
> for i in Psi[:,[0,1,2]]:
>     c = c + 1
>     ax2.scatter(i[0],i[1],color=colors[idx[c]])
>     lbl = bikeData.ix[c]['City']
>     ax2.annotate(lbl,(i[0],i[1]),fontsize=8)
```

Now we gather and summarize these clusters.

```
> bikeData['cluster'] = idx
> clustergroups = bikeData.groupby('cluster')
> clustergroups.describe()
```

Cluster 0 is the largest grouping with 127 out of 374, and is of places with a small number of overall riders, percentage wise more male than female (89%), and who have a higher percentage of african americans and lower asian americans versus the other clusters.
50 out of 127 of these counties voted republican, thus accounting for half of the republican areas in the study.



		Population	# Workers	% Bike	Bike Num	% female	% male	white	afric	latino	
cluster											
0	mean	171809.25	75931.39	0.65	454.81	10.55	**89.46**	66.50	**17.00**	10.65	
1	mean	120417.44	52698.84	0.98	560.80	20.32	79.68	30.38	10.88	43.91	
2	mean	226703.20	106108.38	2.07	1694.87	**50.83**	49.17	**66.97**	11.39	14.30	
3	mean	279362.17	123404.72	0.62	596.58	12.13	87.88	38.68	8.15	**45.74**	
4	mean	158906.04	74350.46	1.06	773.70	23.31	76.69	49.78	7.50	22.11	
5	mean	**2601540.29**	**1162422.57**	**2.56**	**16309.43**	30.14	69.86	47.91	15.00	20.61	

		dem	rep	graduate	earnings	educationi	incomei	asian	native	other	count
0	mean	126740.87	95753.53	10.36	29082.42	5.04	4.74	2.92	0.55	2.38	**127.00**
1	mean	**1634877.44**	**664336.64**	10.41	29375.76	4.76	4.88	12.32	0.18	2.34	25.00
2	mean	138366.38	104565.07	13.06	27809.58	5.51	4.42	3.76	**1.08**	2.51	84.00
3	mean	266726.45	241884.36	7.90	27966.23	4.06	4.46	4.84	0.51	2.11	64.00
4	mean	319595.21	230767.66	**15.97**	**38397.97**	**6.08**	**6.69**	**16.31**	0.29	**4.04**	67.00
5	mean	745006.71	244233.57	15.70	34888.29	5.50	5.95	13.01	0.33	3.10	7.00

Figure 2: Cluster statistics. black font is max and orange font is minimum per variable

Cluster 1 looks like it may have a data error regarding dem/rep high numbers so we look at the cluster more in depth.

```
> bikeData[bikeData["cluster"] == 1]
```

It turns out that the areas in this cluster are all within Los Angeles or Cook county (Chicago). Each area turns out to be neighborhoods within Los Angeles or Chicago, and the though the population size of those neighborhoods is around 100 thousand, the voting numbers reflect how the entire county which is much larger voted. Thus Cluster 1 reflect neighborhoods in Chicago and Los Angeles. It is 21 of the 97 Californian units studied.

Cluster 2, the 2nd largest cluster with 84 units, has the highest percentage of female riders(51%), is the most white (66%), is the poorest (median earnings of 27,809) has the largest native american population, and has the 2nd highest bike commuter percentage (2.07% of workers)

Cluster 3, represents the 64 areas with the lowest percentage of commuters (.62%) of all groups. It has the highest latino percentage (45.8%), and lowest education index (4.06) and graduate degree obtained percentage (7.9%). To see where these areas are located we run:

```
> bikeData[bikeData["cluster"] == 3].groupby('State').count()
California  23
```

```
Texas       16
Arizona     8
Nevada      6
Florida     5
Colorado    3
New Jersey 2
New Mexico 1
```

Cluster 4, are the cities/towns with the highest earnings (38,397), graduate degrees (15.97%), asian (16.31) and other percentages (4.04). 43 of the 97 individuals from California are within this grouping.

Cluster 5 is the small cluster (7 units) and contains the cities with the highest populations, workforce, percentage of commuters. It contains Los Angeles, San Francisco, Chicago, New York, Portland, Philadelphia, and Seattle.

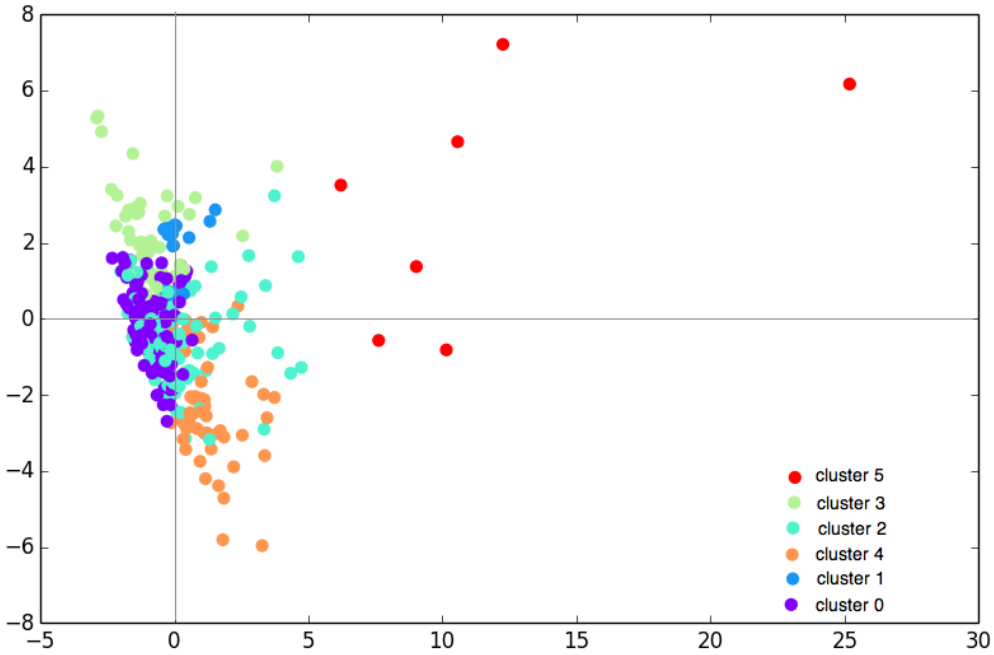


Figure 3: Clustering results

4 PCA as it applies to the Clusters

The variable cloud diagram presented earlier in the paper, showed that the first dimension (x axis) is highly correlated with the Total.Population, Number of Workers, percentage of female commuters, percentage of bicycle commuters and asian. The second dimension was associated with latino at the one end against white, higher income and higher education at the other.

Thus looking at our clusters, we see cluster 5 should be those with the highest populations, and that within the cluster the two points below the x-axis should have a higher percentage of female commuters and asian populations, whereas those higher above should have slightly higher latino populations. This is seen in our data with San Francisco and Seattle below the x-axis and New York, Los Angeles, and Chicago with higher y-values.

Cluster 4 according to the PCA, will be richer, have higher education and more white/asian which also

holds from the data. Within the cluster itself, points closer to the x-axis and more to the right, should be more asian (Oakland,Berkeley,SanJose), while those closer the yaxis and lower should be more white (Arlington,ElliotCity).

Cluster 3 should be more latino, less white, less afluent/educated and have a slightly african american and higher male commuter percentage, with those farther to the left being smaller in population to those farth to the right. Thus we see El Paso, Brownsville, Mcallen and Edinburg as being best representative of this cluster.

The remaining clusters 0, 1 and 2 would be more difficult to assess from the first two planes alone as they are relatively close to the origin, but its still possible to see cluster 0 would be smaller in population and commuter percentage, and cluster 1 would be more latino generally (which makes sense for Chicago/LA neighb This cluster 1 is overally influenced by the same socio-economic county data because these are all neighborhoo within the same two counties. Additionally, we could have plotted the 3rd and 4th principal components to see how the variables are spread there and then see how the clusters look like in those dimensions.

5 majority male vs female bike commuter population

```
boys = bikeData[bikeData["Percent.of.bike.commuters.female"] < bikeData["Percent.of.bike.commuters.male"]]
girls = bikeData[bikeData["Percent.of.bike.commuters.female"] > bikeData["Percent.of.bike.commuters.male"]]
df = pd.DataFrame()
df['male'] = boys.mean();   df['female']= girls.mean()
```

	male	female
Population	**252799.18**	159703.98
Total.Workers	**114192.14**	71418.54
Percentage.of.Bicycle.Commuters	**1.17**	0.55
Number.of.Bike.Commuters	**1228.32**	339.85
Percent.of.bike.commuters.who.are.	15.29	**78.65**
Percent.of.bike.commuters.who.are.	**84.72**	21.35
dem	**306664.02**	257041.80
rep	**191367.61**	162578.80
graduate	**11.68**	11.50
earnings	**30550.33**	29395.52
educationi	5.15	**5.17**
incomei	**5.04**	4.75
white	55.80	**57.83**
afric	**12.21**	10.93
nativ	0.56	**0.80**
asian	**6.74**	6.10
other	**2.68**	2.60
latin	**22.02**	21.75
count	327	47

Figure 4: breakdown between sexes

6 Remarks on IPython and pandas

Having been the first time I've used IPython, and particularly as a "notebook" with pandas, which allows for working and sharing of sessions via a web browser, I would say its not too disimilar from R, in the classic interactive sense. Trying to choose/manuever between using the combination of numpy, scipy, scikit-learn, and pandas was a little tricky, and required looking up some materials. Matplotlib was fairly easy to utlize as well. I found myself frustrated at times with the tab completion, and how the notebook feature works much better in Chrome than in Safari on Mac os x. I'm still more familiar with most of the functionality in R, I'd usually go with it unless the task required making it web accessabile / interactive which is where python and its stack would be of much greater advantage.