

Data Scientist : Skill Up!!

Data Scientist, the sexiest job of the 21st century, says [Harvard Business Review \(https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century\)](https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century).

Python and **R** are the primary programming languages used by academia and industry. Most wanna be data scientists spend hours trying to hone their skills in R and Python. But is there more to it?

Q1 : Are there any other technical skill(s) that is being sought by various employers?

Dice.com a job portal, makes available an API, using which jobs matching the 'data scientist' key words were retrieved. It returned about 30K jobs. The site was scrapped using Python and the data analyzed.

Q2 : Can jobs out there be clustered? Is there an inherent pattern to these jobs?

Finally, with 30K jobs out there,

Q3 : Can one come up with a simple approach to find most relevant job given the skills one has.

Most of the job portal's out there lets you search by a job title, but data scientist jobs involves diverse range of skills. How can one find a job which more or less perfectly aligns with ones skills?

The data analysis to follows intends to address each of the above questions.

Tasks performed in the analysis,

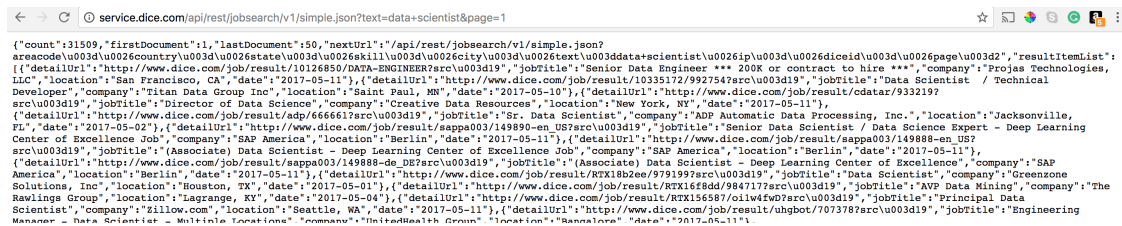
1. [Acquiring data - Scraped data on Dice.com API using Python/BS4.](#)
2. [Data Cleansing and Preprocessing](#)
3. [Data Exploration](#)
4. [Clustering](#)
 - [Visualizing Clusters](#)
 - [Cluster Interpretation](#)
5. [Salary Regression](#)
6. [Best Suited Job using Nearest Neighbor](#)
7. [Conclusion](#)

Q1 : Are there any other technical skill(s) that is being sought by various employers?

1. Acquiring data - Scraped data on Dice.com API using Python/BS4

To help answer the above question,

1) Data was scrapped from the Dice.com API. The api URL is as follows



```
{
  "count": 31509,
  "firstDocument": 1,
  "lastDocument": 50,
  "nextUrl": "/api/rest/jobsearch/v1/simple.json?areacode\u003d\u0026country\u003d\u0026state\u003d\u0026skill\u003d\u0026city\u003d\u0026text\u003d\u0026data+scientist\u0026ip\u003d\u0026diceid\u003d\u0026page\u003d2",
  "resultItemList": [
    {
      "detailUrl": "http://www.dice.com/job/result/10126850/DATA-ENGINEER?src\u003d19",
      "jobTitle": "Senior Data Engineer *** 200K or contract to hire ***",
      "company": "Projes Technologies, LLC",
      "location": "San Francisco, CA",
      "date": "2017-05-11",
      "detailUrl": "http://www.dice.com/job/result/10335172/992754?src\u003d19",
      "jobTitle": "Data Scientist / Technical Developer",
      "company": "Titan Data Group Inc",
      "location": "Saint Paul, MN",
      "date": "2017-05-10",
      "detailUrl": "http://www.dice.com/job/result/cdata/933219?src\u003d19",
      "jobTitle": "Director of Data Science",
      "company": "Creative Data Resources",
      "location": "New York, NY",
      "date": "2017-05-11",
      "detailUrl": "http://www.dice.com/job/result/asp/66661?src\u003d19",
      "jobTitle": "Sr. Data Scientist",
      "company": "ADP Automatic Data Processing, Inc.",
      "location": "Jacksonville, FL",
      "date": "2017-05-02",
      "detailUrl": "http://www.dice.com/job/result/sappa003/149890-en-US?src\u003d19",
      "jobTitle": "Senior Data Scientist / Data Science Expert - Deep Learning Center of Excellence Job",
      "company": "SAP America",
      "location": "Berlin",
      "date": "2017-05-11",
      "detailUrl": "http://www.dice.com/job/result/sappa003/149888-en-US?src\u003d19",
      "jobTitle": "(Associate) Data Scientist - Deep Learning Center of Excellence Job",
      "company": "SAP America",
      "location": "Berlin",
      "date": "2017-05-11",
      "detailUrl": "http://www.dice.com/job/result/sappa003/149888-de-DE?src\u003d19",
      "jobTitle": "(Associate) Data Scientist - Deep Learning Center of Excellence",
      "company": "SAP America",
      "location": "Berlin",
      "date": "2017-05-11",
      "detailUrl": "http://www.dice.com/job/result/RTX18b2ee/979199?src\u003d19",
      "jobTitle": "Data Scientist",
      "company": "Greenzone Solutions, Inc",
      "location": "Houston, TX",
      "date": "2017-05-01",
      "detailUrl": "http://www.dice.com/job/result/RTX16f8dd/984717?src\u003d19",
      "jobTitle": "AVP Data Mining",
      "company": "The Rawlings Group",
      "location": "Lagrange, KY",
      "date": "2017-05-04",
      "detailUrl": "http://www.dice.com/job/result/RTX16587/011w4fw?src\u003d19",
      "jobTitle": "Principal Data Scientist",
      "company": "Willow.com",
      "location": "Seattle, WA",
      "date": "2017-05-11",
      "detailUrl": "http://www.dice.com/job/result/uhgbot/707378?src\u003d19",
      "jobTitle": "Engineering Manager - Data Scientist - Multiple Locations",
      "company": "UnitedHealth Group",
      "location": "Bannockburn",
      "date": "2017-05-11"
    }
  ]
}
```

2) Scrapping data of the DICE.com API was time consuming. The API basically give out URL's of job's matching the data scientist keyword. There were ~30,000 jobs, therefore URL's. Once the URL's were downloaded, each of the URL were to be accessed and from the job description - skills, salar info (if avaiable), location information, company name etc was scrapped. The process took nearly a day to run. The data was saved onto csv files for further analysis.

Scraping Notebook : [Data Scientist Skills - Data Scraping.ipynb \(Data Scientist Skills - Data Scraping.ipynb\)](#)

2. Data Cleansing and Preprocessing

The high level cleansing activities are as follows,

1. Split the location column into two separate columns - city and state. This will allow for easy mapping of the data on map of US.
2. Travel requirement of the job ranged from no telecommuting to 100% WFH. This was grouped into 'Office', 'WFH', 'Office with 100% travel', 'Office with some travel', 'WFH with some travel', 'WFH with 100% travel'
3. Jobs were also grouped into 'Full time' and 'contract'.
4. Each job had skills were comma separated, this was converted to multiple rows, with one skill per row for a job. The dataset was further cleaned to remove skills which were more than two words to handle bad data. Due to large number of skills about 22K unique skills, only skills that show up on at least 50 job listings was retained. This reduced the number of skills to ~400
5. Skills were pivoted from rows to columns, this will ensure one row per job listing and will also aid the sklearn clustering algorithm.

Data Prep Notebook : [Data Scientist Skills - Data Prep.ipynb \(Data Scientist Skills - Data Prep.ipynb\)](#)

3. Data Exploration

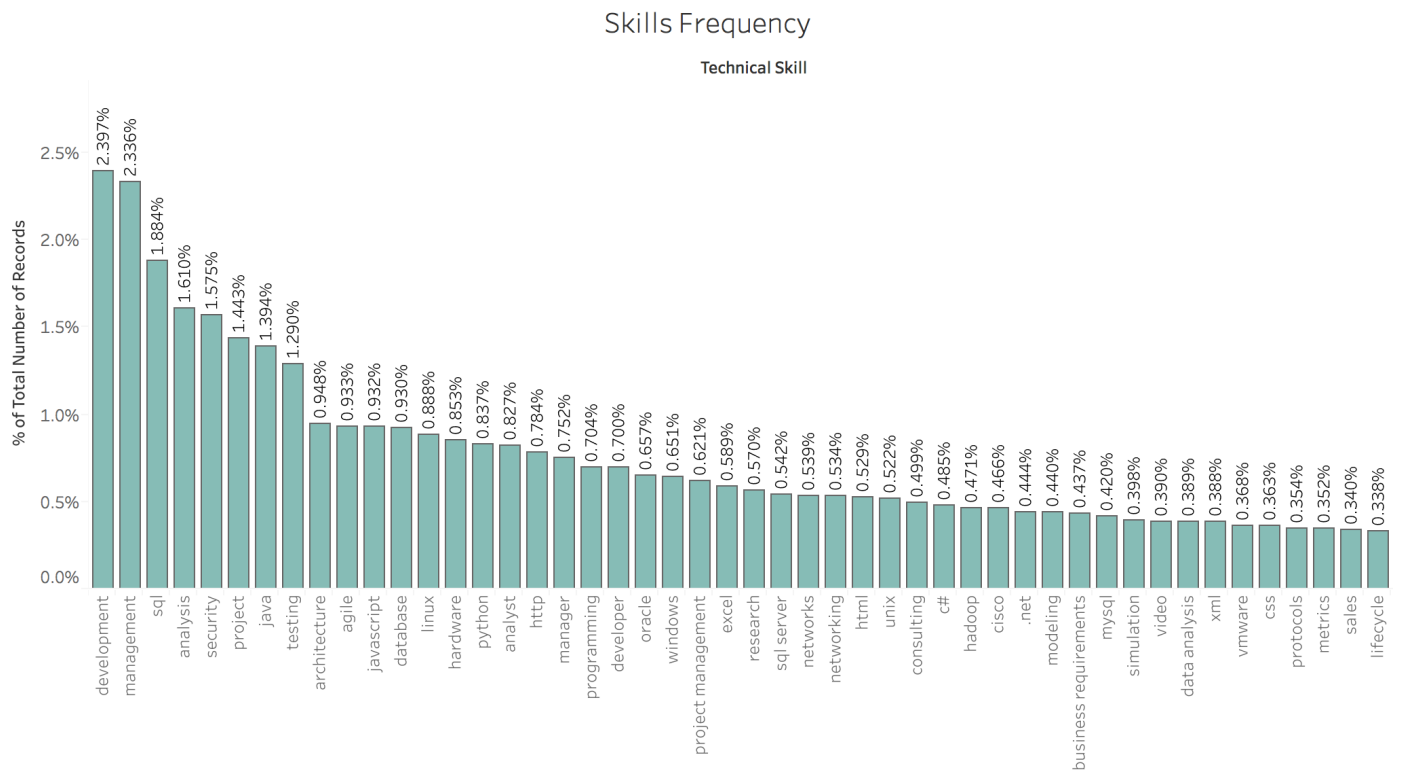
Now that the data was extracted, the next logical step was to explore it by visualizing the data. **Tableau** was used to quickly visualize interactions, patterns in the data.

The word cloud of the various skills generic skill buckets like 'development','management','analysis' stands out. 'Python' as one of the top skills. But it is interesting to see 'Java' , 'SQL' and databases such as 'Oracle' stands out. 'Excel' is still not past its glory days.

Skills - Word Cloud

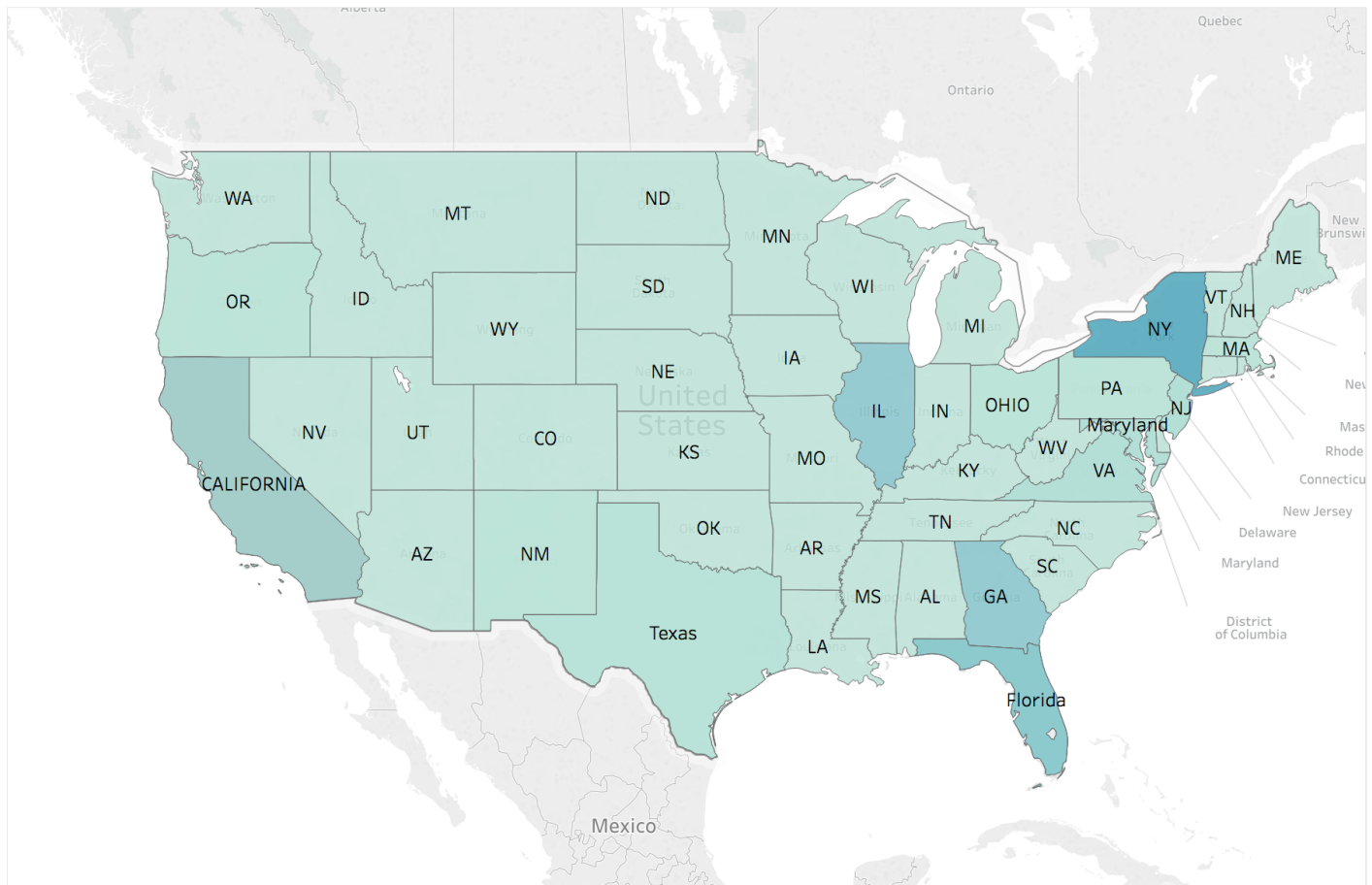


The previous view - world cloud although is prettier of the two. The traditional bar plot is the more insightful as it maintains the skills sorted by magnitude

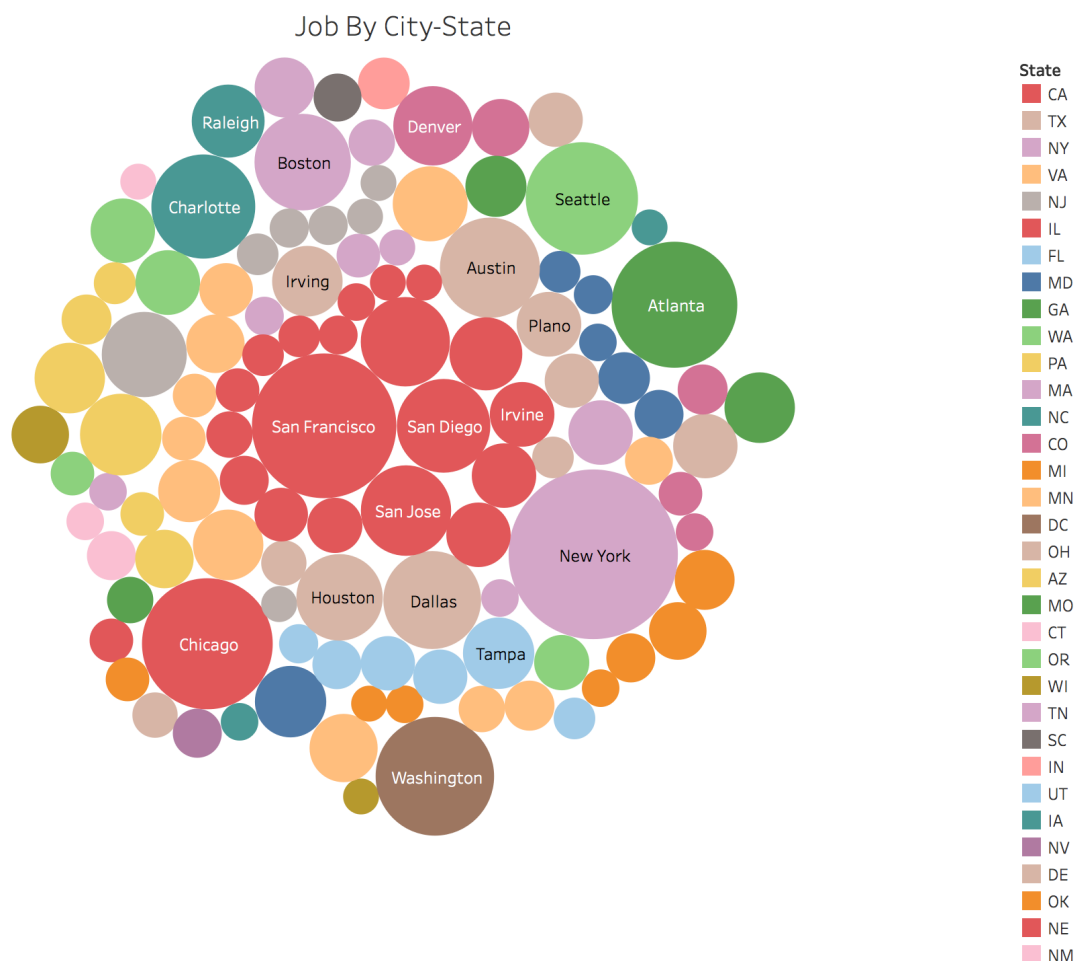


As the data had the location information, plotting it on a map was the most intuitive thing to do. As one would expect large concentration of Data Science jobs are in New York and California. But one can also see Florida and Georgia are right there up with the big boys.

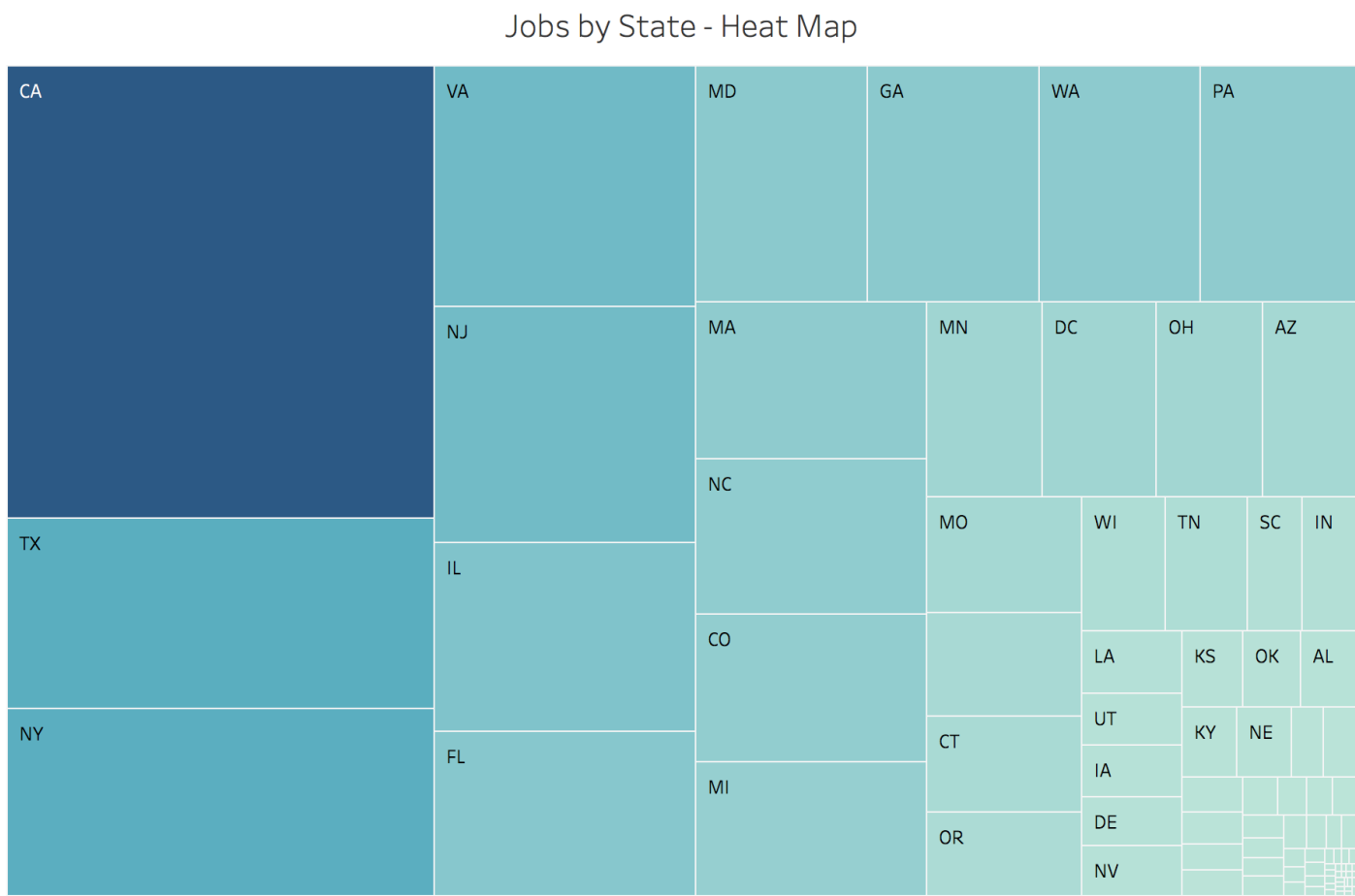
Jobs by State



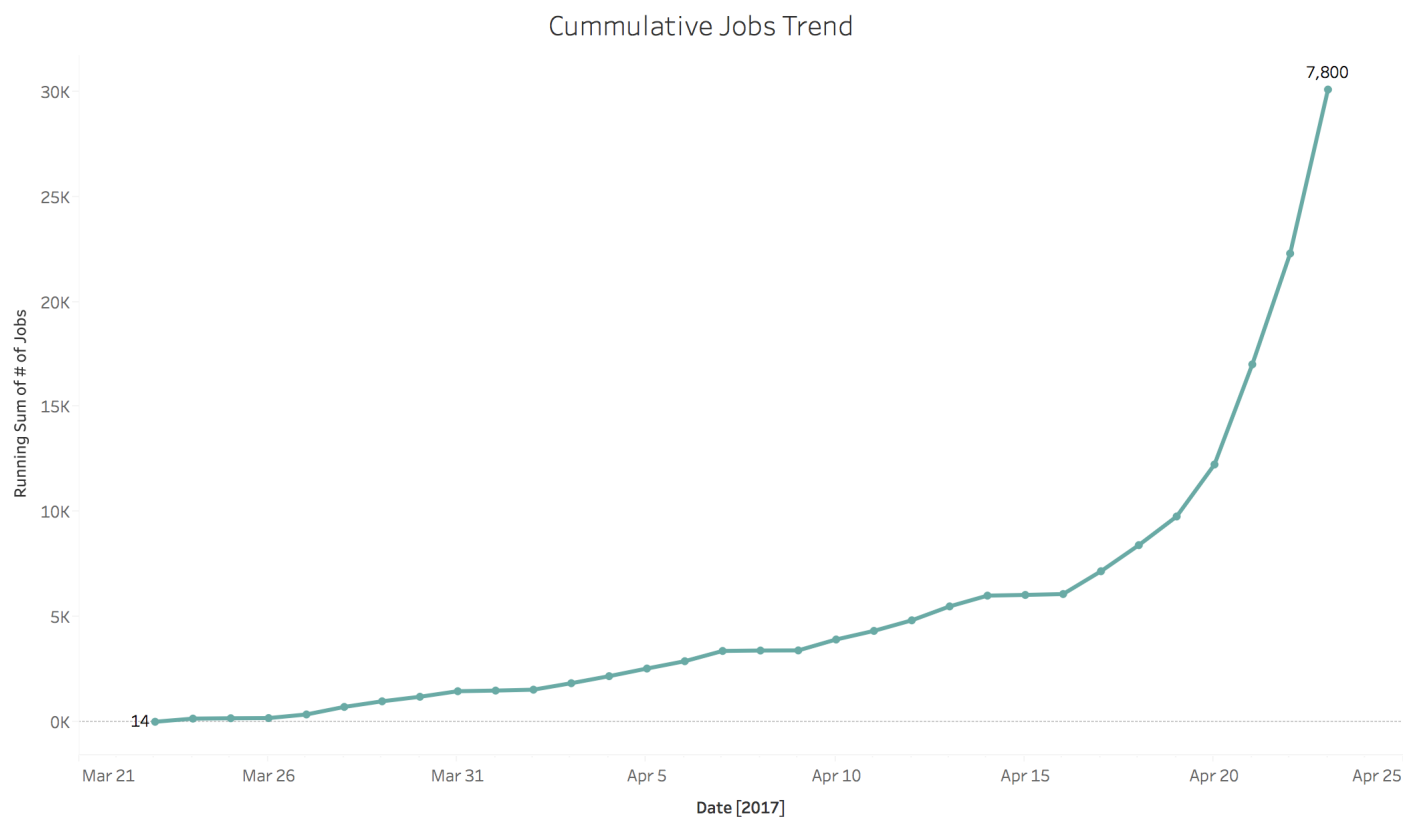
A city view of the data shows bigger cities such as NYC, Austin, Chicago, Seattle, Boston have large concentration of Data Science jobs.



Another view of the data by state, a heat map sorted in order # of jobs. Again the usual suspects right up there.

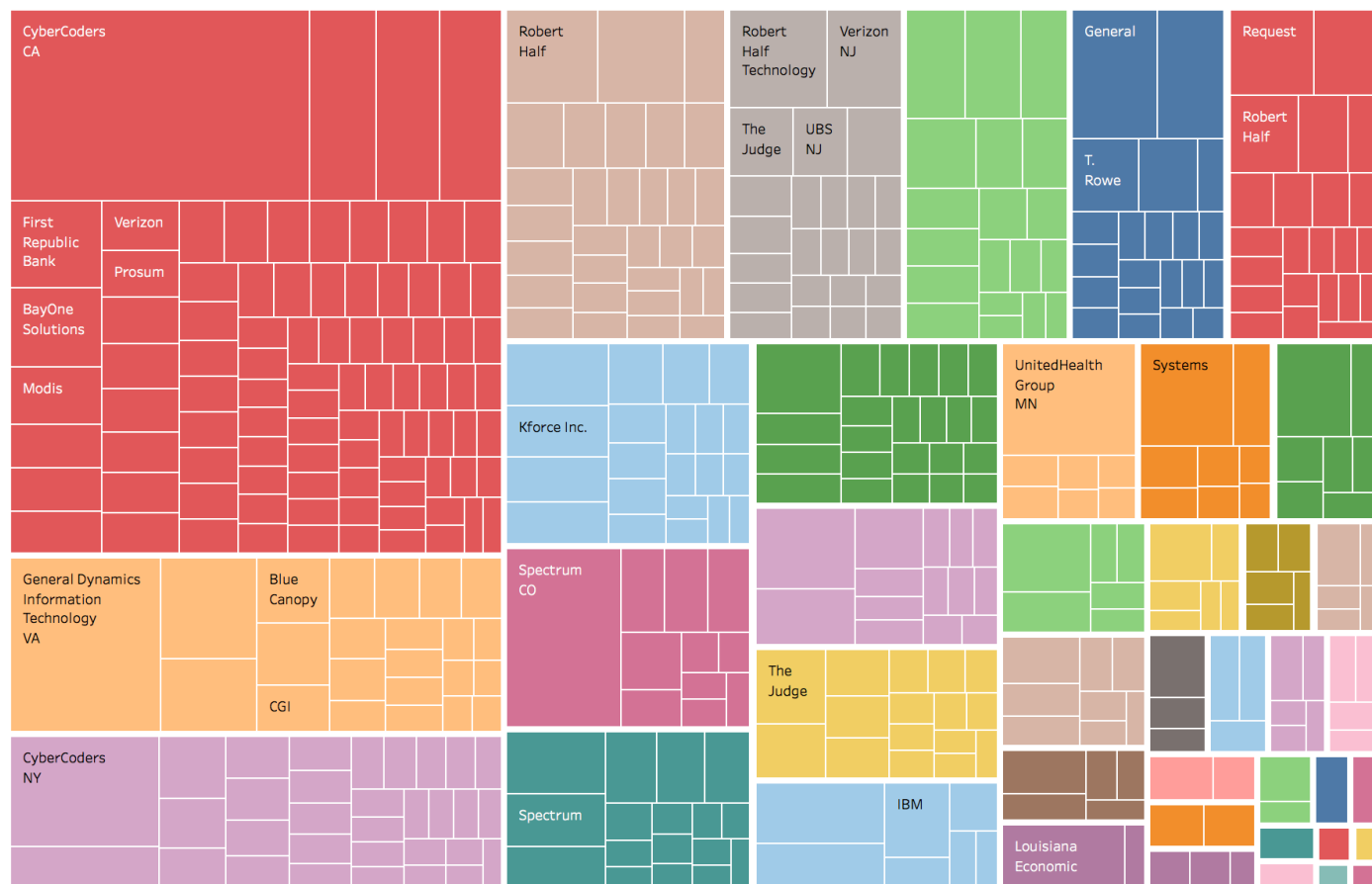


The data was pulled mid of April 2017. There are job positings open for over a month. Also nearly 8k jobs get added on a daily basis.



A view of top employers for every state - Cyber Coder in CA and NY, General Dynamics in VA and FL, Spectrum in CO, Robert Half in TX

Jobs by State/Employers



4. Clustering

There were 30,000 jobs scrapped off the Dice site. With so many unique skills is there inherent cluster to these jobs. The objective was to cluster the the jobs on skills they required to help identify groups of jobs. This can potentially help a prospective data scientist to hone specific set of skills to land the jobs of interest.

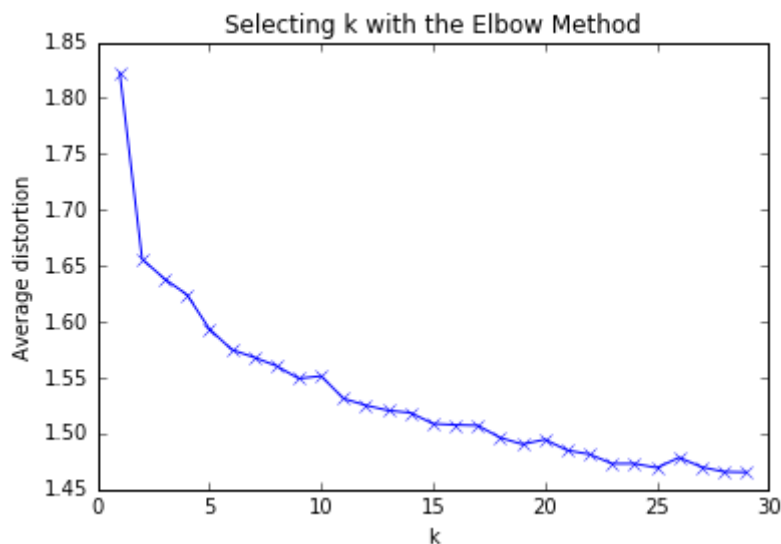
KMeans Clustering

Kmeans was used to cluster on the jobs pivoted data. Now one had to choose 'K'. The elbow method was used to choose 'K', basically looking for the value of 'K' when the distortion flattens out. I picked 6 for the value of 'K'.

```
In [84]: #https://www.packtpub.com/books/content/clustering-k-means
import numpy as np
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
import matplotlib.pyplot as plt

K = range(1, 30)
meandistortions = []
for k in K:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(job_detail_cluster_df)
    meandistortions.append(sum(np.min(cdist(job_detail_cluster_df, kmeans.cluster_centers_, 'euclidean'), axis=1)) /
                             job_detail_cluster_df.shape[0])
```

```
In [85]: #elbow method
plt.plot(K, meandistortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Average distortion')
plt.title('Selecting k with the Elbow Method')
plt.show()
```



```
In [133]: #fitting data for 6 clusters
kmeans = KMeans(n_clusters=6)
kmeans.fit(job_detail_cluster_df)
```

```
Out[133]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
                  n_clusters=6, n_init=10, n_jobs=1, precompute_distances='auto',
                  random_state=None, tol=0.0001, verbose=0)
```

```
In [134]: #adding the cluster column
job_detail_cluster_df['cluster'] = kmeans.labels_
```

```
In [135]: job_detail_cluster_df['job_listing_no'] = job_detail_df_comb['job_listing_no']
```

```
In [136]: job_detail_cluster_df.head()
```

```
Out[136]:
```

	3d	access	accounting	active directory	adobe	aerospace	agile	aix	ajax	alcatel	...	webk
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0

5 rows × 376 columns

```
In [137]: #saving a copy of the clustered data
job_detail_cluster_df.to_csv('data/clustered_job.csv',index=False)
```

```
In [138]: job_detail_cluster_df=job_detail_cluster_df.drop('job_listing_no',axis=1)
```

```
In [139]: #cluster summary
job_detail_cluster_df.groupby('cluster')['cluster'].count()
```

```
Out[139]: cluster
0      19553
1       1226
2       2851
3       1934
4       1453
5       3083
Name: cluster, dtype: int64
```

Visualizing Clusters

The dataset is high dimensional to help visualize in 2d PCA and LDA methods were used. About 60% of data is assigned to cluster 0 and the rest is distributed among rest of the clusters.

```
In [140]: import matplotlib.pyplot as plt
import pandas as pd

from sklearn.decomposition import PCA as sklearnPCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.datasets.samples_generator import make_blobs

from pandas.tools.plotting import parallel_coordinates
```

```
In [141]: pca_df = job_detail_cluster_df.copy();
```

```
In [142]: pca_df=pca_df.drop('cluster',axis=1)
```

```
In [143]: pca = sklearnPCA(n_components=2) #2-dimensional PCA  
transformed = pd.DataFrame(pca.fit_transform(pca_df))
```

```
In [144]: transformed.head()
```

```
Out[144]:
```

	0	1
0	-0.502763	-0.097960
1	-0.502763	-0.097960
2	0.808102	0.173707
3	0.344094	0.339268
4	-0.455611	0.076819

```
In [145]: pca_vis_df = pd.concat([transformed,job_detail_cluster_df['cluster']],axis=1)
```

```
In [146]: pca_vis_df.columns = ['x','y','cluster']
```

```
In [147]: pca_vis_df.head()
```

```
Out[147]:
```

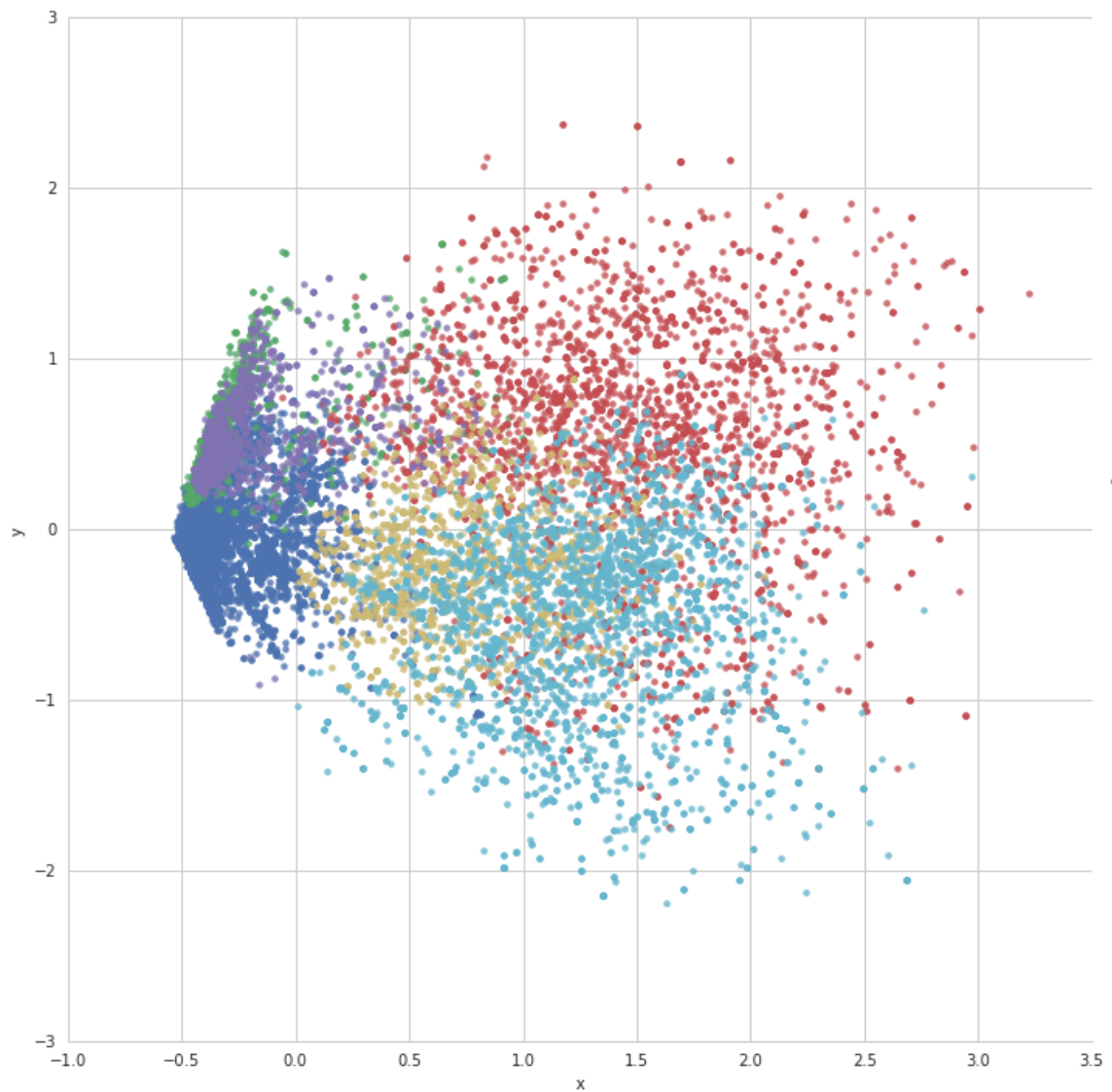
	x	y	cluster
0	-0.502763	-0.097960	0
1	-0.502763	-0.097960	0
2	0.808102	0.173707	2
3	0.344094	0.339268	4
4	-0.455611	0.076819	0

```
In [148]: import seaborn as sns
```

One can see that there is a reasonable separation among the clusters

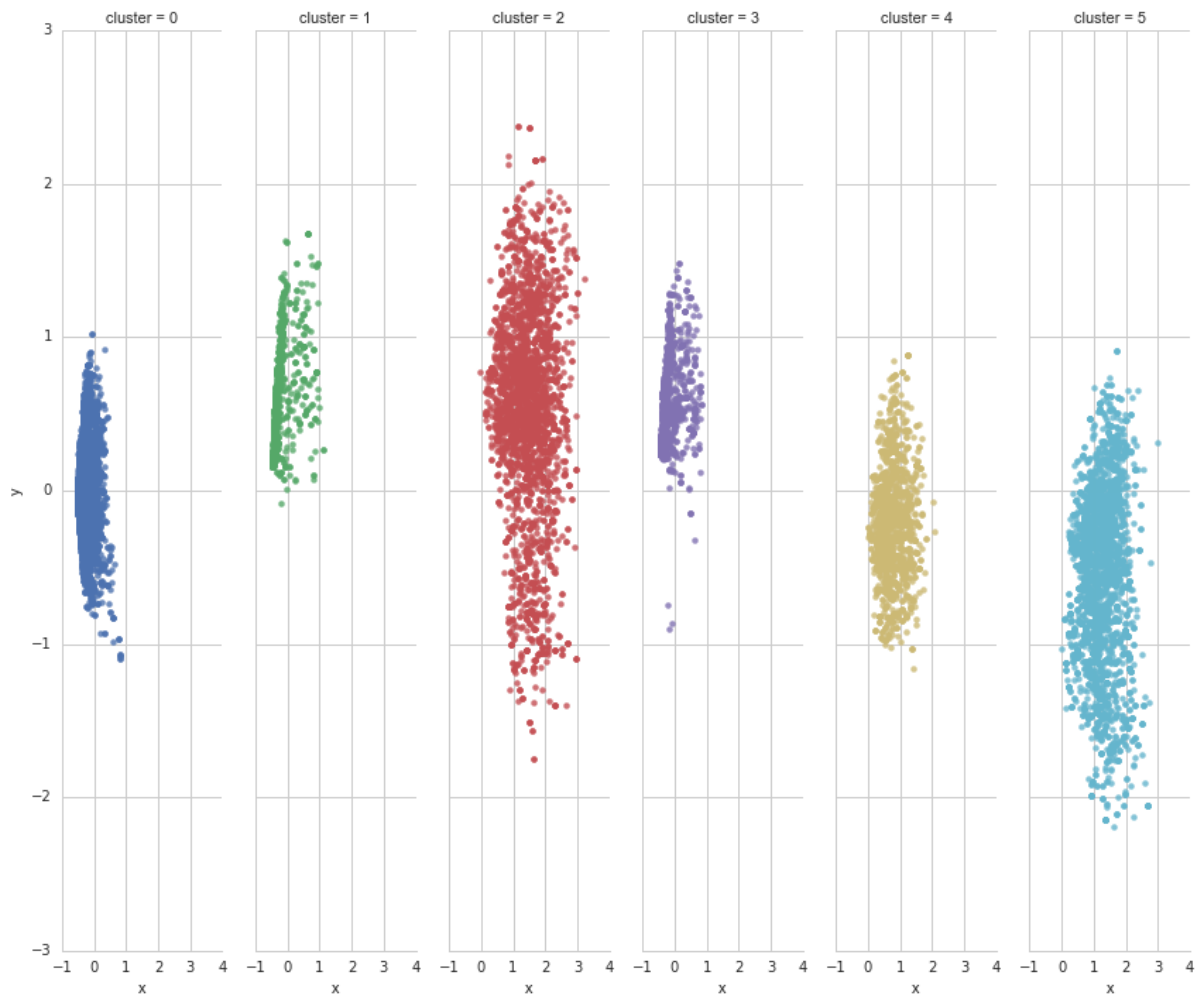
```
In [149]: sns.lmplot('x', 'y', data=pca_vis_df, hue='cluster', fit_reg=False,  
size=10)
```

```
Out[149]: <seaborn.axisgrid.FacetGrid at 0x14e54c748>
```



```
In [150]: sns.set_style("whitegrid")
sns.lmplot('x', 'y', data=pca_vis_df, hue='cluster', col='cluster', fit_
reg=False, size=10, aspect=.2)
```

Out[150]: <seaborn.axisgrid.FacetGrid at 0x140469c88>



LDA Visualization

```
In [151]: lda = LDA(n_components=2) #2-dimensional LDA
lda_transformed = pd.DataFrame(lda.fit_transform(pca_df, job_detail_cluster_df['cluster']))
```

```
In [152]: lda_transformed.head()
```

```
Out[152]:
```

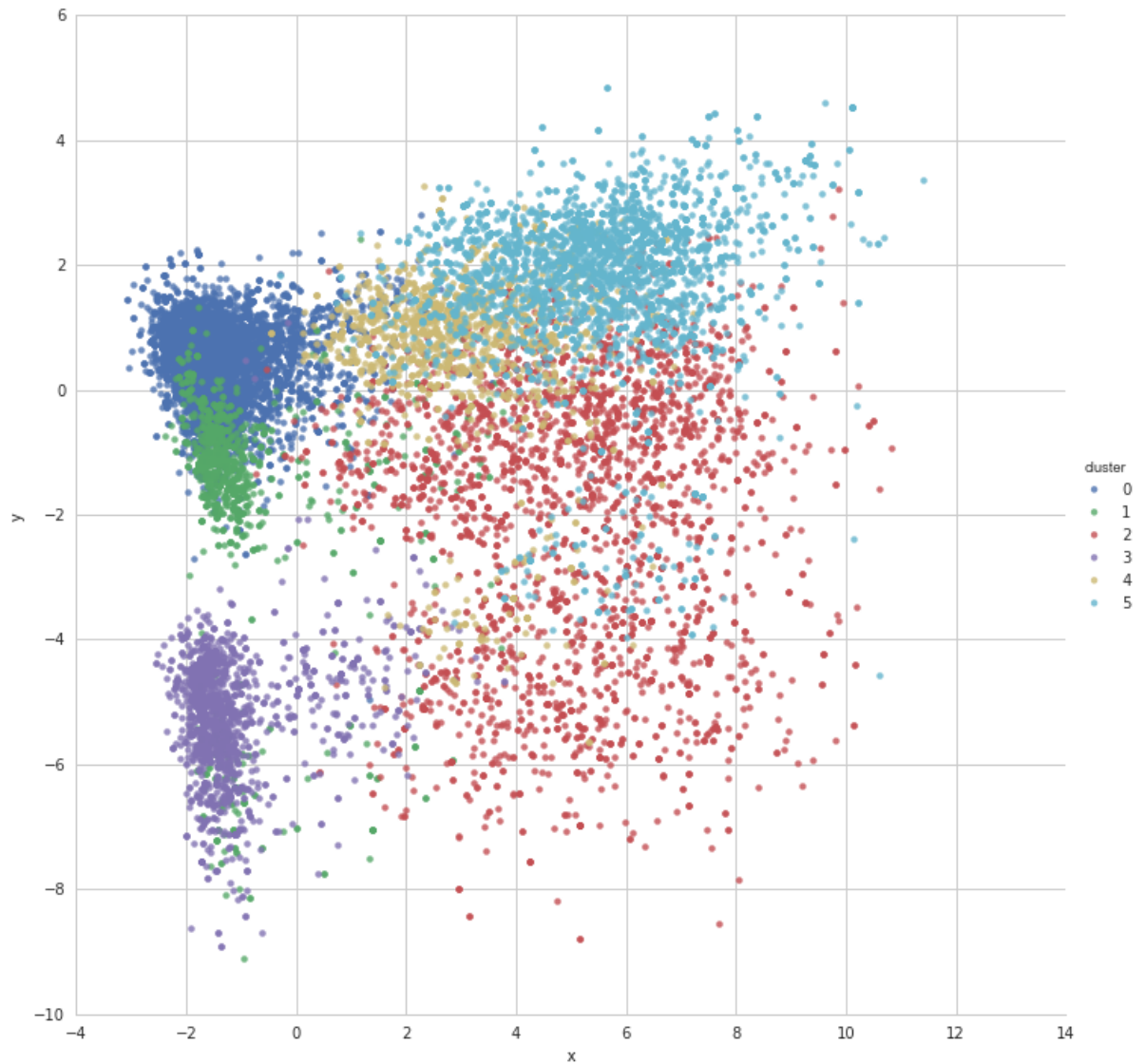
	0	1
0	-1.811530	0.661668
1	-1.811530	0.661668
2	2.954020	-0.343657
3	1.369423	0.069220
4	-1.865512	0.441628

```
In [153]: lda_vis_df =  
pd.concat([lda_transformed,job_detail_cluster_df['cluster']],axis=1)
```

```
In [154]: lda_vis_df.columns = ['x','y','cluster']
```

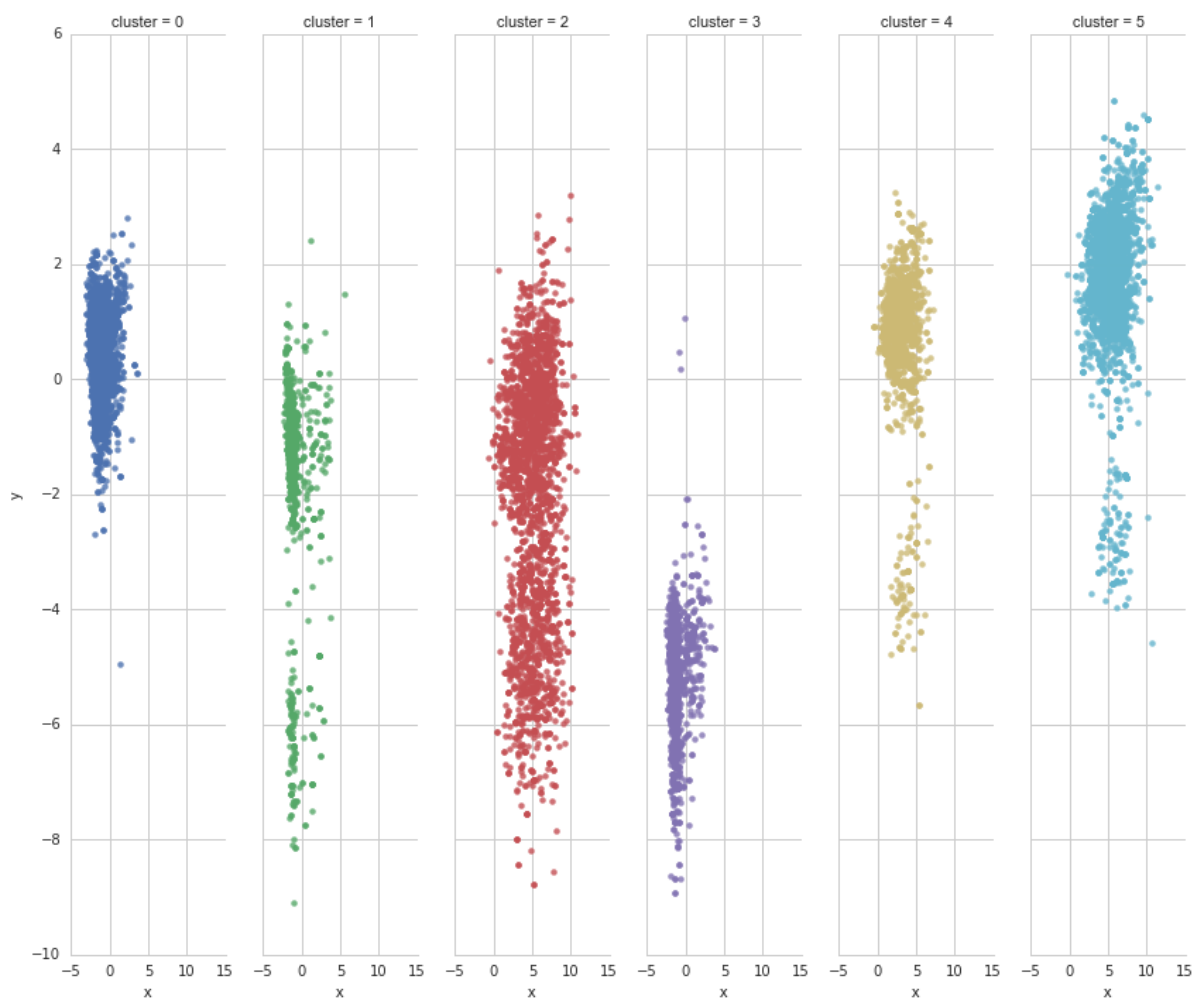
```
In [155]: sns.lmplot('x', 'y', data=lda_vis_df, hue='cluster', fit_reg=False,  
size=10)
```

```
Out[155]: <seaborn.axisgrid.FacetGrid at 0x14e54cc50>
```




```
In [156]: sns.lmplot('x', 'y', data=lda_vis_df, hue='cluster', col='cluster', fit_
reg=False, size=10, aspect=.2)
```

```
Out[156]: <seaborn.axisgrid.FacetGrid at 0x14c3314e0>
```



Interpreting Clusters

Word Cloud by Cluster

Cluster 0 with 60% of the jobs is basically the **programmer**, the **data engineer** with Python being the most important skill.

Cluster 1 like the remaining clusters has about 10% of the jobs, is the **visualization engineer** with Javascript being the most prominent skill.

Cluster 2, again 10% of the jobs is the **Database Engineer** with oracle, sql server, java standing out.

Cluster 3 is the **Big Data Engineer** with technologies like hadoop, spark forming the cluster.

Cluster 4 and 5 is the **Manager** of Data Science Teams.

Skills Cluster Analysis



Salary Regression

Some of the jobs made the salary offered public, but it was only 10% of the jobs. The salary field needed some cleaning. After wading out outliers, about 2k jobs were left. The data being high dimensional, 2k records is not enough.

```
In [199]: salary_df.shape
```

```
Out[199]: (3901, 9)
```

```
In [169]: salary_df = job_detail_df.loc[['-' in str(i) for i in  
list(job_detail_df.salary_info.values)],:]
```

```
In [200]: salary_list = [re.sub('^[0-9-]+', '', salary) for salary in salary_df.sa  
lary_info]
```

```
In [205]: salary_list= [str(sal).split('-')[1] for sal in salary_list]
```

```
In [ ]: for sal in salary_list:  
        if sal == '':
```

```
In [209]: def sal_check(sal):  
        if sal == '':  
            return 0  
        else:  
            return int(sal)
```

```
In [212]: salary_list=[sal_check(sal) for sal in salary_list]
```

```
In [213]: salary_df.loc[:, 'salary_c'] = salary_list
```

/Users/ajaykliyara/anaconda_py3/anaconda/lib/python3.5/site-packages/pa
ndas/core/indexing.py:465: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>
self.obj[item] = s

```
In [214]: salary_df.head()
```

Out[214]:

	job_listing_no	job_type	page	salary_info	technical_skills	travel_info
0	1	Full Time, Contract Corp-To-Corp, Contract Ind...	1	150-200K+RSU+BONUS	R, Python, JavaScript, C, Java, SQL, Data Sci...	Telecommuting not available\nTravel not required
1	2	Full Time	1	180-220k	5+ years experience as a Data Scientist. 3+ ye...	Telecommuting not available\nTravel not required
20	21	Full Time	1	150000.00–180000.00 per annum	Manager, Data Scientist, Hadoop, Cassandra, Sp...	Telecommuting not available\nTravel not required
27	28	Full Time	1	-	Required Qualifications ? * Bachelor degree in...	Telecommuting not available\nTravel not required
36	37	NaN	1	130000.00 - 130000.00	(See Job Description)	Telecommuting not available\nTravel not required

```
In [228]: #removing outliers
salary_df=salary_df.loc[(salary_df['salary_c']>50000)&(salary_df['salary_c']<500000),:]
```

```
In [229]: salary_df.shape
```

```
Out[229]: (1926, 9)
```

```
In [234]: int_df = pd.concat([job_header_df,job_detail_cluster_df],axis=1)
```

```
In [235]: int_df.shape
```

```
Out[235]: (30100, 384)
```

```
In [240]: salary_df=
salary_df.join(int_df,on='job_listing_no',how='inner',lsuffix='sal_')
```

```
In [255]: salary_df=salary_df.astype(str)
```

```
In [257]: salary_df.head()
```

```
Out[257]:
```

	job_listing_nosal_	job_type	pagesal_	salary_info	technical_skills	travel_inf
72	73	Full Time	2	100,000-130,000	SQL, ETL, BI, Python	Telecomm not available\ not require
105	106	Full Time, Full-time, Employee	3	100,000–150,000	Machine Learning, Data Mining, Python, Hadoop,...	Telecomm not available\ not require
109	110	Full Time	3	105000 - 120000	Data Science, machine learning	Telecomm not available\ not require
116	117	Full Time, Full-time, Employee	3	160,000–225,000	Data Scientist, Machine Learning, Personalizat...	Telecomm not available\ not require
118	119	Full Time	3	150,000–250,000	Data Science, Python	Telecomm not available\ not require

5 rows × 393 columns

```
In [259]: import csv
```

```
In [263]: salary_df.columns
```

```
Out[263]: Index(['job_listing_nosal_', 'job_type', 'pagesal_', 'salary_info',
                'technical_skills', 'travel_info', 'job_travel_info', 'job_type_
                info',
                'salary_c', 'company',
                ...,
                'web services', 'weblogic', 'websphere', 'windows', 'windows ser
                ver',
                'wireless', 'writer', 'xml', 'xslt', 'cluster'],
                dtype='object', length=393)
```

```
In [267]: salary_reg = salary_df.drop(['job_listing_nosal_', 'job_type', 'pagesal_
                _', 'salary_info',
                'technical_skills',
                'travel_info', 'date', 'detailUrl', 'jobTitle', 'job_listing_no', 'page', 'loc
                ation', 'State'], axis=1)
```

```
In [268]: salary_reg.to_csv('data/salary.csv',index=False,quoting=csv.QUOTE_ALL)
```

```
In [252]: sal_only = salary_df.loc[:,['job_listing_no','salary_c']]
```

```
In [253]: sal_only.to_csv('data/salary_only.csv',index=False)
```

```
In [244]: salary_df.head()
```

```
Out[244]:
```

	job_listing_nosal_	job_type	pagesal_	salary_info	technical_skills	travel_inf
72	73	Full Time	2	100,000-130,000	SQL, ETL, BI, Python	Telecomm not available\ not requir
105	106	Full Time, Full-time, Employee	3	100,000–150,000	Machine Learning, Data Mining, Python, Hadoop,...	Telecomm not available\ not requir
109	110	Full Time	3	105000 - 120000	Data Science, machine learning	Telecomm not available\ not requir
116	117	Full Time, Full-time, Employee	3	160,000–225,000	Data Scientist, Machine Learning, Personalizat...	Telecomm not available\ not requir
118	119	Full Time	3	150,000–250,000	Data Science, Python	Telecomm not available\ not requir





5 rows × 393 columns

The key features contributing to the salary was identified using 'Azure ML-Boosted Decision Trees' feature importance module. On the limited set of Data R and Python were one of the top contributors.

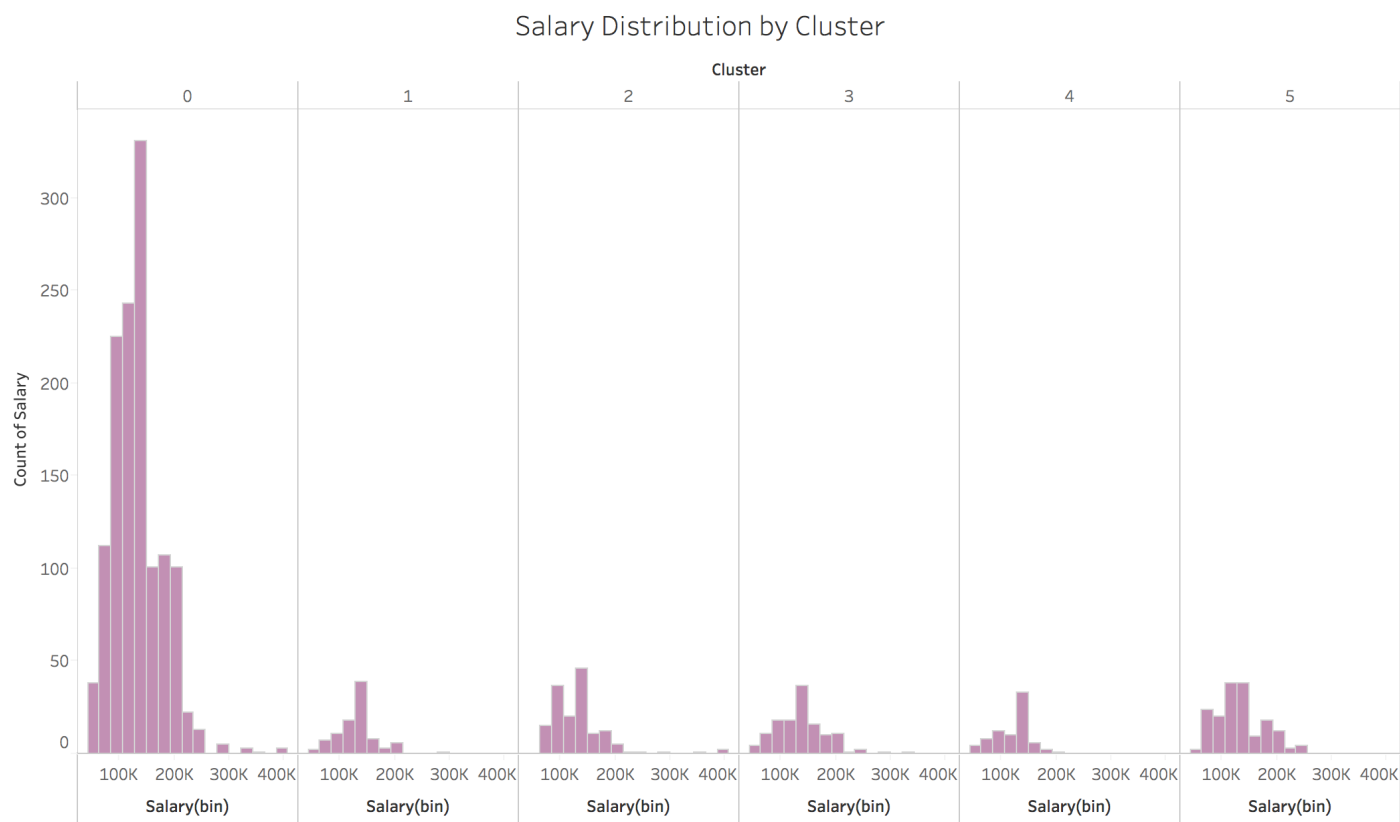
Data Scientist - Salary Regression > Permutation Feature Importance > Feature importance

rows
100

columns
2

	Feature	Score
view as  		
	r	296.649797
	jenkins	166.546655
	soap	160.84493
	python	120.986502
	data warehousing	96.945243
	oracle	63.646468
	sales	46.059119
	git	42.453172
	excel	30.748514
	salesforce	19.301068

All clusters seems to have a mean salary around 120k. But please note only <10% of jobs made salary available.



Nearest Neighbors

Data Science is a vast field with diverse set of skills. The existing job portals allow searching by job title or specific skills, what if one had to find the best match based on all of one's skills. The below analysis tests out this concept - train the Nearest Neighbor Algorithm on jobs data. On the trained model, input a job seekers skill set and find the nearest lying jobs.

The algorithm is quick to train and gives more customized results.

```
In [269]: from sklearn.neighbors import NearestNeighbors
```

```
In [270]: neigh = NearestNeighbors(5, 0.5)
```

```
In [272]: nn_job_skill_df = job_detail_cluster_df.copy()
```

```
In [274]: nn_job_skill_df=nn_job_skill_df.drop('cluster',axis=1)
```



```
In [275]: #fitting the data
neigh.fit(nn_job_skill_df)
```

```
Out[275]: NearestNeighbors(algorithm='auto', leaf_size=30, metric='minkowski',
metric_params=None, n_jobs=1, n_neighbors=5, p=2, radius=0.5)
```

Testing the nearest neighbor algorithm, by passed on the records from the dataset itself. Record with row index 10 was passed to the model and it returned the nearest 5 neighbors, it is evident from the results which shows row index 10 as the nearest record.

```
In [279]: test_job = nn_job_skill_df.iloc[10,:]
```

```
In [280]: neigh.kneighbors(test_job,5,return_distance=True)
```

```
/Users/ajaykliyara/anaconda_py3/anaconda/lib/python3.5/site-packages/sk
learn/utils/validation.py:395: DeprecationWarning: Passing 1d arrays as
data is deprecated in 0.17 and will raise ValueError in 0.19. Reshape y
our data either using X.reshape(-1, 1) if your data has a single featur
e or X.reshape(1, -1) if it contains a single sample.
DeprecationWarning)
```

```
Out[280]: (array([[ 0.          ,  1.          ,  1.          ,  3.31662479,  3.3166247
9]]),
array([[ 10,  15,   9, 2477, 191]]))
```

Conclusion

Answering the questions initially raised,**Q1 : Are there any other technical skill(s) that is being sought by various employers?**

The word cloud of the various skills generic skill buckets like 'development', 'management', 'analysis' stands out. 'Python' as expected is one of the top skills. But it is interesting to see '**Java**', '**SQL**' and databases such as 'Oracle' stands out. '**Excel**' is still not past its glory days. As one would expect large concentration of Data Science jobs are in New York and California. But one can also see Florida and Georgia are right there up with the big boys. A city view of the data shows bigger cities such as NYC, Austin, Chicago, Seattle, Boston have large concentration of Data Science jobs. A view of top employers for every state - Cyber Coder in CA and NY, General Dynamics in VA and FL, Spectrum in CO, Robert Half in TX

Q2 : Can jobs out there be clustered? Is there an inherent pattern to these jobs?

Kmeans clustering of K=6 was trained.

Cluster 0 with 60% of the jobs is basically the **programmer**, the **data engineer** with Python being the most important skill.

Cluster 1 like the remaining clusters has about 10% of the jobs, is the **visualization engineer** with Javascript being the most prominent skill.

Cluster 2, again 10% of the jobs is the **Database Engineer** with oracle, sql server, java standing out.

Cluster 3 is the **Big Data Engineer** with technologies like hadoop, spark forming the cluster.

Cluster 4 and 5 is the **Manager** of Data Science Teams.

Some of the jobs made the salary offered public, but it was only 10% of the jobs. The salary field needed some cleaning. After wading out outliers, about 2k jobs were left. The data being high dimensional, 2k records is not enough. The key features contributing to the salary was identified using 'Azure ML-Boosted Decision Trees' feature importance module. On the limited set of Data R and Python were one of the top contributors.

Q3 : Can one come up with a simple approach to find most relevant job given the skills one has.

Data Science is a vast field with diverse set of skills. The existing job portals allow searching by job title or specific skills, what if one had to find the best match based on all of one's skills. The analysis tests out the concept that by training the **Nearest Neighbor Algorithm** on jobs data. Then On the trained model, input a job seekers skill set and find the nearest lying jobs. The algorithm proved to work efficiently and accurately. The algorithm is quick to train and gives more customized results.

Next Steps.....

Use the Nearest Neighbor algorithm to build an application on top of the Dice.com jobs data, to allow for job seekers to identify better suiting jobs.

Reference

Jobs Data - www.Dice.com

Kmeans - <https://www.packtpub.com/books/content/clustering-k-means>
(<https://www.packtpub.com/books/content/clustering-k-means>)

Multi-Dim Visualization - <http://www.apnorton.com/blog/2016/12/19/Visualizing-Multidimensional-Data-in-Python/> (<http://www.apnorton.com/blog/2016/12/19/Visualizing-Multidimensional-Data-in-Python/>)