

# Resume Analysis System

2011102144 강석윤

2012104110 이상록

2013104108 이환희

# CONTENTS

- Previous Topic
- Topic in progress
- Extract Data
- Transform
- Load
- Visualization(outcomes)

# Previous Topic



데이터 엔지니어(Hadoop/HBase/Druid)  
Kakao Corp • 경희대학교  
대한민국 • 351명

1촌 맷기

RDBMS로 다루기 어려운 양의 방대한 데이터를 분석하기 위한 기술스택을 보유하고 있습니다. 빅데이터 엔지니어링에 필수적인 각종 오픈소스 프로젝트를 엔터프라이즈 환경에 도입하고 개선한 경험이 있습니다. - Java...

더 보기 ▾

## 2 프로젝트

### Apache Tajo

Contributor

- Report bug
- Apply bugfix
- Code migration
- Improve tsql(tajo-cli)

[프로젝트 보기](#)

### Initialroot Analyzer

Overall Architect Design  
Develop log Analyzing process with Apache Hadoop, Apache Mahout, mongoDB.

다른 리더(명)

### 업계 지식

### 소프트웨어 개발

### 도구 & 테크놀로지

<b>Java</b>	<b>자바</b>
<b>C++</b>	<b>AWS</b>
기타 기술	
<b>HBase</b>	<b>Tajo</b>
<b>Solr</b>	<b>OLAP</b>
<b>Mahout</b>	<b>Oracle Cloud</b>
<b>Lucene</b>	

# Previous Topic

- [cjswotl6274@khu.ac.kr](mailto:cjswotl6274@khu.ac.kr)
- [cjswotl6274@naver.com](mailto:cjswotl6274@naver.com)
- [1500sheep@gmail.com](mailto:1500sheep@gmail.com)
- [iin35kkk@naver.com](mailto:iin35kkk@naver.com)
- [desppencile1@gmail.com](mailto:desppencile1@gmail.com)
- [desppencile2@gmail.com](mailto:desppencile2@gmail.com)
- [desppencile3@gmail.com](mailto:desppencile3@gmail.com)
- [desppencile4@gmail.com](mailto:desppencile4@gmail.com)
- [desppencile5@gmail.com](mailto:desppencile5@gmail.com)
- [desppencile6@gmail.com](mailto:desppencile6@gmail.com)
- [desppencile7@gmail.com](mailto:desppencile7@gmail.com)
- [desppencile8@gmail.com](mailto:desppencile8@gmail.com)
- [itsforxls@gmail.com](mailto:itsforxls@gmail.com)
- [temporary317@naver.com](mailto:temporary317@naver.com)
- [tgwlsr6@gmail.com](mailto:tgwlsr6@gmail.com)
- [fhrslaaj@naver.com](mailto:fhrslaaj@naver.com)
- [fhrslaaj@daum.net](mailto:fhrslaaj@daum.net)
- [2012104110@khu.ac.kr](mailto:2012104110@khu.ac.kr)
- [...](#)

The screenshot shows a LinkedIn profile page with a dark header bar. On the right side of the header are '로그인' (Login) and '회원 가입' (Sign Up) buttons. Below the header, there's a large central message area with a blue exclamation mark icon and the text '계정이 차단되었습니다.' (Your account has been blocked). A blue button labeled '본인 인증' (Self-validation) is positioned below the message. To the left of the main message, there are two sections: '왜 그런가요?' (Why was it blocked?) and '작오가 있다고 생각하세요?' (Do you think it's a worm?). Both sections contain explanatory text and links. At the bottom of the page, there's a horizontal footer bar with various LinkedIn navigation links.

LinkedIn

로그인 회원 가입

계정이 차단되었습니다.

본인 인증

왜 그런가요?

회원님이 계정에서 LinkedIn 서비스 약관에 어긋나는 활동이 발견되어 계정이 차단되었습니다.

작오가 있다고 생각하세요?

작오로 인해 계정이 차단되었다고 생각하시면 먼저 본인 인증을 양료해 주시기 바랍니다.

정부에서 발행한 신분증으로 본인 인증을 마치시면 작오가 있었는지 확인해 드립니다. 실제로 작오가 있었을 경우 계정 차단을 해제한 후 연락을 드리겠습니다.

고객센터 | 소개 | 채용 | 광고 | 채용 솔루션 | 세일즈 솔루션 | 소규모 사업체 | 모바일 | 언어 | 온라인 클래스 | 프로파일  
채용 광고 검색 | 전체 목록 회원 | Pulse | 회사 | 대학  
LinkedIn © 2018 | 사용약관 | 개인정보 취급방침 | 커뮤니티정책 | 구매정책 | 저작권정책 | 메일 받지 않기

only crawl about 1,200 users << 512,000 users

# Topic in progress

## Analysis Resume in Indeed

The screenshot shows the Indeed search interface. The search bar has 'data engineer' entered. Below the search bar, there's a tip about using a zip code in the 'Where' field. A banner for 'Indeed Prime' offers great tech companies. The search results show various job listings, including one for 'Data Engineer - Experienced Associate' at PwC. This specific listing includes a company overview, a 'PwC LOS Overview' section, and a 'Job Description' section detailing the role's responsibilities.

- by Job

- by City

- by State

- by Company

# Extract Data - code

Hadoop Overview Datanodes Snapshot Startup Progress Utilities ▾

## Browse Directory

/project\_lastest/code

Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	project	supergroup	7.99 KB	2018. 6. 13. 오후 8:20:45	2	128 MB	Analyze_word_v2.py
-rw-r--r--	project	supergroup	3.32 KB	2018. 6. 13. 오후 8:20:45	2	128 MB	Indeed.py
-rw-r--r--	project	supergroup	1.95 KB	2018. 6. 13. 오후 8:20:45	2	128 MB	column_count.py
-rw-r--r--	project	supergroup	4.32 KB	2018. 6. 13. 오후 8:23:24	2	128 MB	find_word_v2.py
-rw-r--r--	project	supergroup	11.17 KB	2018. 6. 13. 오후 8:20:45	2	128 MB	indeedcrawling.py
-rw-r--r--	project	supergroup	4.94 KB	2018. 6. 13. 오후 8:20:45	2	128 MB	indeedcrawling_last.py

# Extract Data - crawling URL

crawling url using selenium

```
mysql> select * from jobs;
+-----+-----+
| id | name
+-----+
| 1 | data engineer
| 2 | big data engineer
| 3 | machine learning engineer
| 4 | deep learning engineer
| 5 | data warehouse engineer
| 6 | data analyst
| 7 | data architect
| 8 | database administrator
| 9 | database developer
| 10 | software engineer
| 11 | software embedded engineer
| 12 | software developer
| 13 | application developer
| 14 | ios developer
| 15 | android developer
| 16 | application engineer
| 17 | web developer
| 18 | full stack developer
| 19 | web front end developer
| 20 | ui developer
| 21 | ux developer
| 22 | web back end developer
| 23 | system administrator
| 24 | system analyst
| 25 | server developer
| 26 | server administrator
| 27 | windows administrator
| 28 | linux administrator
| 29 | unix administrator
| 30 | wireless network engineer
| 31 | network engineer
| 32 | network administrator
| 33 | network architect
| 34 | security analyst
| 35 | security engineer
| 36 | cloud architect
| 37 | scrum master
| 38 | information technology manager
| 39 | it project manager
+-----+
39 rows in set (0.00 sec)
```



<Jobs - 39>

	Vancouver	Washington	167405
146	Cape Coral	Florida	165831
147	Sioux Falls	South Dakota	164676
148	Edmonton	Alberta	164323
149	Pearl City	Arizona	162592
150	Pembroke Pines	Florida	162329
151	Elk Grove	California	161807
152	Long Beach	Texas	160314
153	Lancaster	California	159523
154	Corona	California	159503
155	Eugene	Oregon	159198
156	Palm Desert	California	157162
157	Bethesda	California	156562
158	Springfield	Massachusetts	153703
159	Pasadena	Texas	152735
160	Fort Collins	Colorado	152061
161	Hayward	Texas	151774
162	Orlando	California	151348
163	Cary	North Carolina	151088
164	Rockford	Illinois	150251
165	Alexandria	Virginia	148892
166	Edmonton	Alberta	148698
167	McKinney	Texas	148559
168	Kansas City	Kansas	148483
169	Joliet	Illinois	147866
170	Albuquerque	New Mexico	147539
171	Torrance	California	147578
172	Bridgewater	Connecticut	147216
173	Lakewood	Colorado	147214
174	Hollywood	Florida	146526
175	Bethlehem	New Jersey	146498
176	Naperville	Illinois	146864
177	Syracuse	New York	144669
178	Mesquite	Texas	143484
179	Dayton	Ohio	143483
180	Lawrenceville	Georgia	142772
181	Clarksville	Tennessee	142357
182	Orange	California	139969
183	Pasadena	California	139751
184	Rancho Cucamonga	California	139301
185	Killeen	Texas	137147
186	Frisco	Texas	136791
187	Hampton	Virginia	136699
188	McAllen	Texas	136509
189	Lansing	Michigan	134873
190	Bellevue	Washington	133992
191	West Valley City	Utah	133579
192	Columbia	South Carolina	133351
193	Lawrence	Kansas	133345
194	Sterling Heights	Michigan	132244
195	New Haven	Connecticut	130666
196	Hannover	Florida	130288
197	Waco	Texas	129909
198	Thousand Oaks	California	127831
199	Cedar Rapids	Iowa	128429
200	Charleston	South Carolina	127999

<Cities and States - 39>

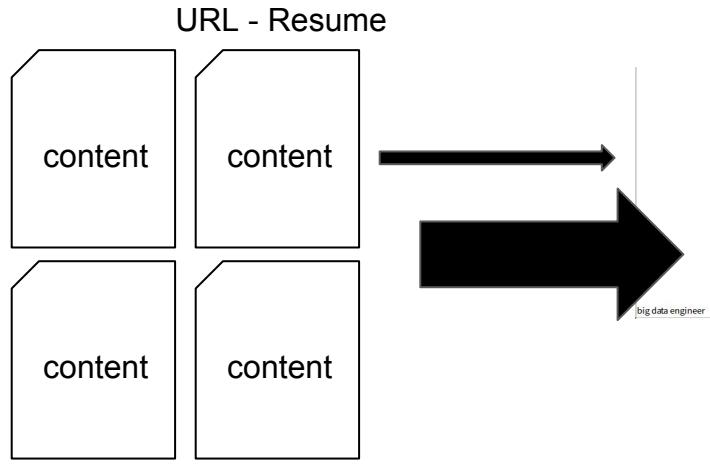
```
| id=e857d2bd71c27d4e&vjs=3
| 832610 | application developer | Phoenix | Arizona | https://www.indeed.com/rc/clk?jk=2132e9ce03835210&fcc
| id=2e25f3358ccba1cf&vjs=3
| 832611 | application developer | Phoenix | Arizona | https://www.indeed.com/rc/clk?jk=f556aaa37088c7142&fcc
| id=2e25f3358ccba1cf&vjs=3
| 832612 | application developer | Phoenix | Arizona | https://www.indeed.com/rc/clk?jk=cc88a5862cab9bd7&fcc
| id=a9bc06f57177e2&vjs=3
| 832613 | application developer | Phoenix | Arizona | https://www.indeed.com/rc/clk?jk=98bd2e680373ef8f&fcc
| id=2e25f3358ccba1cf&vjs=3
| 832614 | application developer | Phoenix | Arizona | https://www.indeed.com/rc/clk?jk=f55724bd453d28ac&fcc
| id=2e25f3358ccba1cf&vjs=3
| 832615 | application developer | Phoenix | Arizona | https://www.indeed.com/rc/clk?jk=374a9edc53b9289&fcc
| id=2e25f3358ccba1cf&vjs=3
+-----+
1261 rows in set (0.01 sec)

mysql> select count(*) from url_crawling;
+-----+
| count(*) |
+-----+
| 828215 |
+-----+
1 row in set (0.16 sec)
```

<URL output - 828,215>

# Extract Data - crawling Resume

crawling resume using BeautifulSoup



Browse Directory

/project\_latest/output

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	project	supergroup	95.12 KB	2018. 6. 13. 오후 8:30:48	2	128 MB	ability_words_city.csv
-rw-r--r--	project	supergroup	313.96 KB	2018. 6. 13. 오후 8:30:48	2	128 MB	city_result.csv
-rw-r--r--	project	supergroup	2.1 MB	2018. 6. 13. 오후 8:30:48	2	128 MB	company_result.csv
-rw-r--r--	project	supergroup	866.93 MB	2018. 6. 13. 오후 8:30:55	2	128 MB	result_all.csv
-rw-r--r--	project	supergroup	11.55 MB	2018. 6. 13. 오후 8:30:55	2	128 MB	result_all_no_content.csv
-rw-r--r--	project	supergroup	193.41 MB	2018. 6. 13. 오후 8:30:56	2	128 MB	result_content_dataanalyst.csv
-rw-r--r--	project	supergroup	104 MB	2018. 6. 13. 오후 8:30:57	2	128 MB	result_content_dataarchitect.csv
-rw-r--r--	project	supergroup	80.29 MB	2018. 6. 13. 오후 8:30:57	2	128 MB	result_content_databaseadministrator_databasedeveloper.csv
-rw-r--r--	project	supergroup	237.47 MB	2018. 6. 13. 오후 8:31:03	2	128 MB	result_content_dataengineer_bigdataengineer.csv
-rw-r--r--	project	supergroup	70.44 MB	2018. 6. 13. 오후 8:31:04	2	128 MB	result_content_machinelearning_deeplearning_datawarehouse.csv
-rw-r--r--	project	supergroup	63.58 MB	2018. 6. 13. 오후 8:31:04	2	128 MB	result_content_softwareembedded_softwaredeveloper.csv
-rw-r--r--	project	supergroup	117.74 MB	2018. 6. 13. 오후 8:31:05	2	128 MB	result_content_softwareengineer.csv
-rw-r--r--	project	supergroup	14.01 MB	2018. 6. 13. 오후 8:31:05	2	128 MB	select_all.csv
-rw-r--r--	project	supergroup	117.5 KB	2018. 6. 13. 오후 8:31:05	2	128 MB	state_result.csv

Hadoop, 2017.

# Transform - wordcount(job,city,state)

```
project@master:~ => ./spark/bin/spark-submit --master
```

```
basic_folder ="/project_lastest/output/"
basic_folder_spark ="hdfs://master:9000/project_lastest/"
input_name= basic_folder + "result_all_no_content.csv"
output_name_company = basic_folder_spark + "company_result"
output_name_city = basic_folder_spark + "city_result"
output_name_state = basic_folder_spark + "state_result"

# rdd_city = rdd.map(lambda x=>(x[2],x[0])).groupByKey()
# rdd_state = rdd.map(lambda x=>(x[3],x[0])).groupByKey()
# rdd_company = rdd.map(lambda x=>(x[0],x[0])).groupByKey()

column_frequency = ["frequency"]
column_job = ["job"]
column_city = ["city"]
column_state = ["state"]

rdd = sc.textFile(input_name).map(lambda line:line.replace('\"','')).map(lambda line:line.split(","))
# In[3]:
```

```
rdd_company = rdd.map(lambda x:([x[0].strip(),x[1]]))
wordcount_company = rdd_company.map(lambda x: (x,1)).reduceByKey(lambda x,y:x+y).map(lambda x:[x[0][0],x[0][1],x[1]])
df_company = sqlc.createDataFrame(wordcount_company)

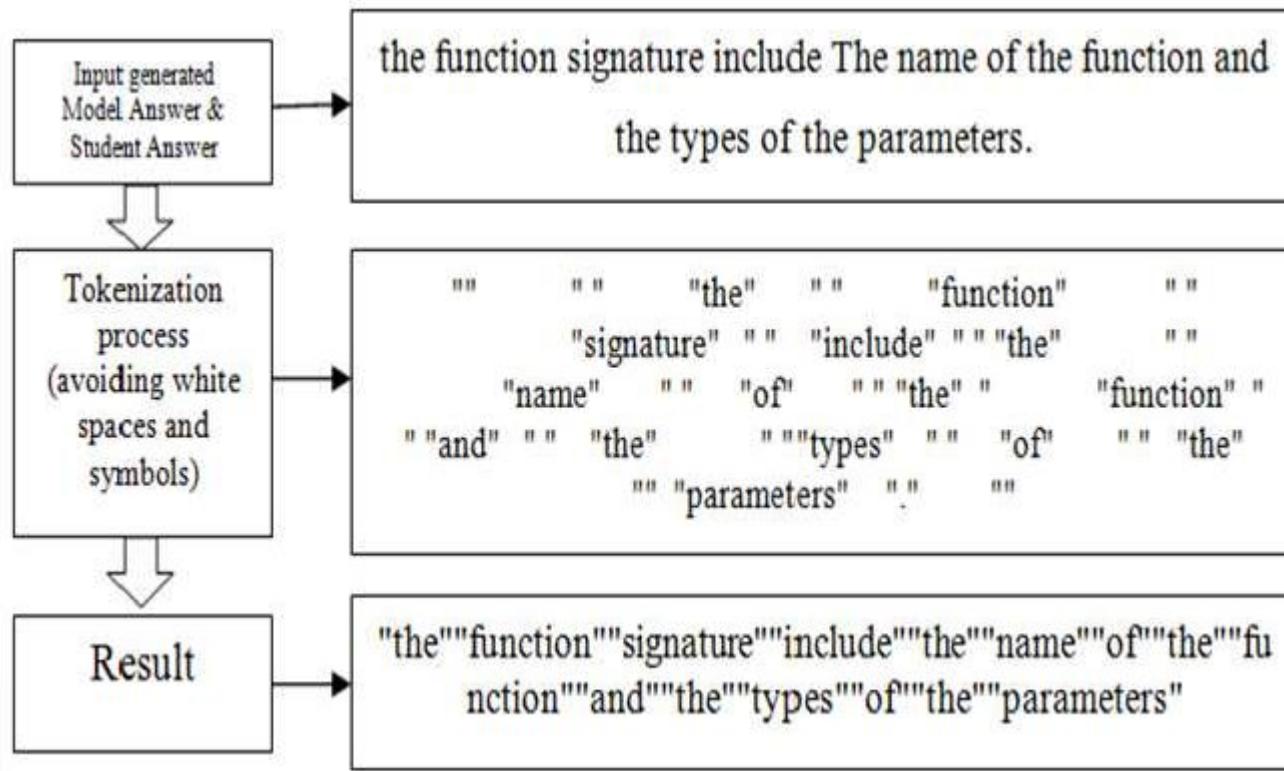
rdd_city = rdd.map(lambda x:(x[3].strip(),x[1]))
wordcount_city = rdd_city.map(lambda x: (x,1)).reduceByKey(lambda x,y:x+y).map(lambda x:[x[0][0],x[0][1],x[1]])
df_city = sqlc.createDataFrame(wordcount_city)

rdd_state = rdd.map(lambda x:([x[0].strip(),x[1]]))
wordcount_state = rdd_state.map(lambda x: (x,1)).reduceByKey(lambda x,y:x+y).map(lambda x:[x[0][0],x[0][1],x[1]])
df_state = sqlc.createDataFrame(wordcount_state)

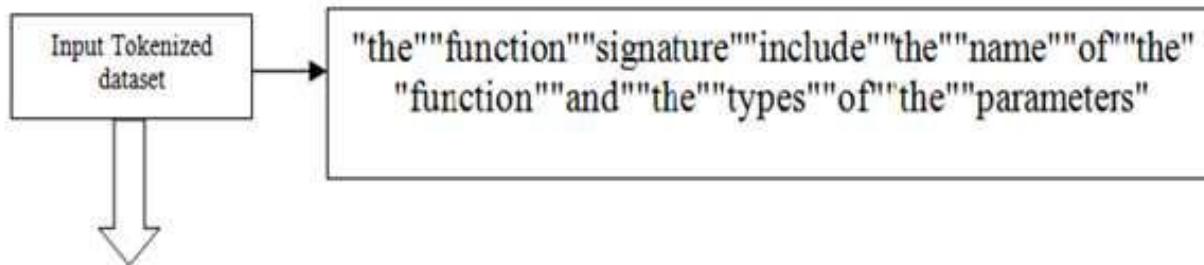
df_company.write.format("com.databricks.spark.csv").save(output_name_company)
df_city.write.format("com.databricks.spark.csv").save(output_name_city)
df_state.write.format("com.databricks.spark.csv").save(output_name_state)
Hadoop,2017.
```

company,job,frequency  
GS5,data analyst,3  
Ricciome Resources,machine learning engineer,2  
IFA North America,data engineer,2  
CVirtual,database administrator,1  
**Hadoop**,database administrator,3  
Orbis,database administrator,3  
Indiana Farm Bureau Insurance,software developer,2  
TEKREQS,software embedded engineer,1  
Mount Carmel Health,data analyst,1  
Evonik,data engineer,3  
Tk-Chain,software engineer,1  
Project\_isitsol,database administrator,2  
VyStar Credit Union,big data engineer,1  
Synectic Solutions,data engineer,3  
Sharonview Federal Credit Union,data engineer,1  
ReviewTrackers,software engineer,4  
PLEXSYS Interface Products,software engineer,2  
Merit Medical Systems Inc.,data engineer,9  
ICES,big data engineer,1  
IPG Photonics Corporation,software embedded engineer,2  
NFI Industries,software developer,1  
SENTEL Corporation,software engineer,1  
Williams-Sonoma,software developer,3  
"company\_result.csv" 60180L, 2204720C

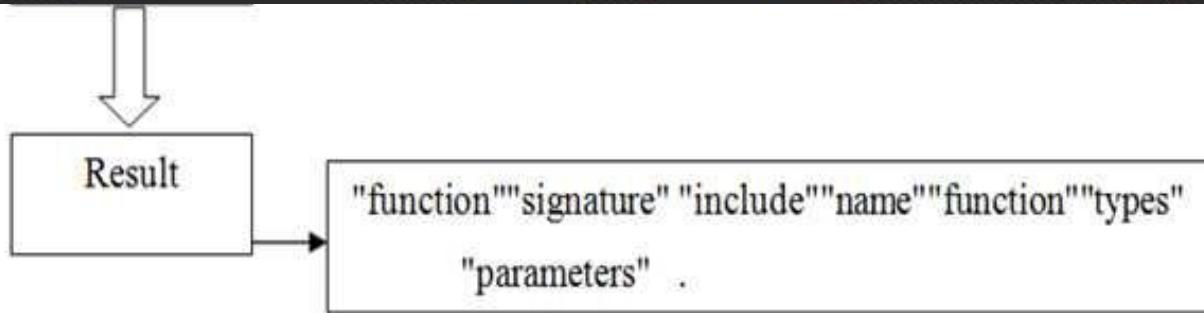
## Transform - NLTK\_wordtokenize



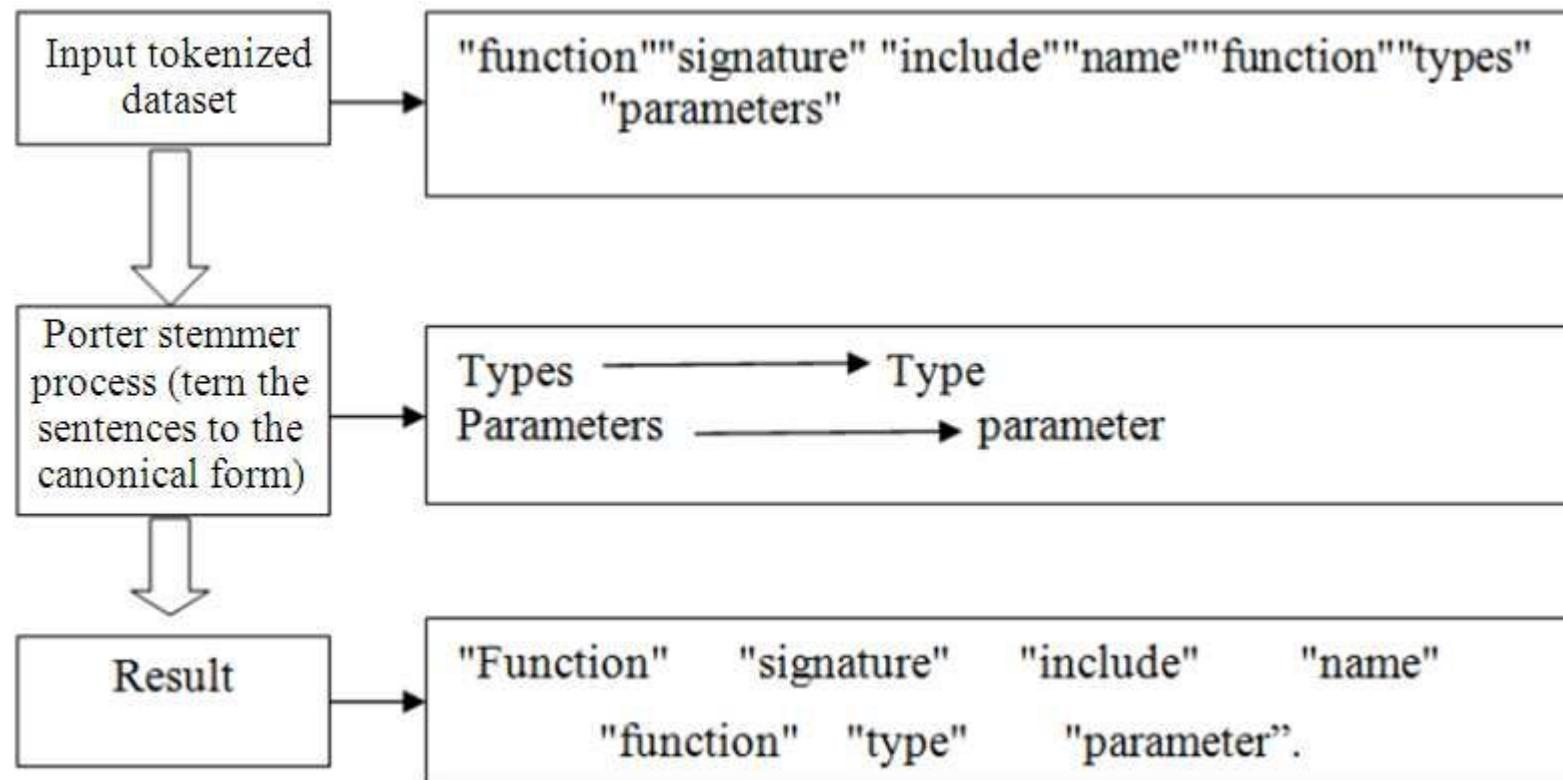
## Transform - NLTK\_stopword



```
english_stops = set(stopwords.words('english'))
english_stops.update(( '*', '&', '“', '”', '‘', '’', '‘‘', '’’, '?’', '‘,’', '’,’'))
words = [word for word in words if word not in english_stops]
```



## Transform - NLTK\_stemming



## Transform - NLTK\_sentiword

```
#####
<bad.a.01: PosScore=0.0 NegScore=0.625>
<bad.s.02: PosScore=0.25 NegScore=0.25>
<bad.s.03: PosScore=0.0 NegScore=0.75>
<bad.s.04: PosScore=0.0 NegScore=0.75>
<regretful.a.01: PosScore=0.0 NegScore=0.625>
<bad.s.06: PosScore=0.0 NegScore=0.75>
<bad.s.07: PosScore=0.0 NegScore=0.625>
<bad.s.08: PosScore=0.0 NegScore=0.5>
<bad.s.09: PosScore=0.0 NegScore=0.75>
<bad.s.10: PosScore=0.0 NegScore=1.0>
<bad.s.11: PosScore=0.0 NegScore=0.375>
<bad.s.12: PosScore=0.0 NegScore=0.75>
<bad.s.13: PosScore=0.0 NegScore=0.75>
<bad.s.14: PosScore=0.0 NegScore=0.75>
-0.6428571428571429
#####
```

```
    if count_N != 0:
        return (posscore - negscore) / count_N
    return 0
#####
```

```
#####
<good.a.01: PosScore=0.75 NegScore=0.0>
<full.s.06: PosScore=0.0 NegScore=0.0>
<good.a.03: PosScore=1.0 NegScore=0.0>
<estimable.s.02: PosScore=1.0 NegScore=0.0>
<beneficial.s.01: PosScore=0.625 NegScore=0.0>
<good.s.06: PosScore=1.0 NegScore=0.0>
<good.s.07: PosScore=0.75 NegScore=0.0>
<adept.s.01: PosScore=0.625 NegScore=0.0>
<good.s.09: PosScore=0.625 NegScore=0.0>
<dear.s.02: PosScore=0.5 NegScore=0.0>
<dependable.s.04: PosScore=0.5 NegScore=0.0>
<good.s.12: PosScore=0.375 NegScore=0.0>
<good.s.13: PosScore=0.625 NegScore=0.0>
<effective.s.04: PosScore=0.0 NegScore=0.0>
<good.s.15: PosScore=0.625 NegScore=0.0>
<good.s.16: PosScore=0.75 NegScore=0.0>
<good.s.17: PosScore=0.75 NegScore=0.0>
<good.s.18: PosScore=0.875 NegScore=0.0>
<good.s.19: PosScore=0.5 NegScore=0.0>
<good.s.20: PosScore=0.375 NegScore=0.125>
<good.s.21: PosScore=0.75 NegScore=0.0>
0.6130952380952381
#####
```

## Transform - set adj'sentiscore to target word

We wish you have experience in python

0.1875

- ▶ tokenize by sentence
  - ▷ tokenize by word
  - ▷ get senti score each word
  - ▷ add all senti score
  - ▷ set senti score in current sentence's word

# Transform - all(job, company, city, state, word, score, result)

ll.csv - LibreOffice Calc

	A	B	C	D	E	F	G	H	I
1	job	company		city	state	word	frequency	score	
2	58040	data architect	JP Morgan Chase	New York	NY	data	39404	0	9829.3900105895
3	194692	software embedded engin	Google	New York	NY	develop	74047	0	7722.9425828238
4	226592	machine learning engine	Amazon.com	New York	NY	data	40334	0	7634.8332751108
5	231093	machine learning engine	Capital One	New York	NY	model	8346	0.625	7268.2089553035
6	57931	data architect	JP Morgan Chase	New York	NY	work	39422	0	7130.7907099629
7	198813	software embedded engin	Intertek	New York	NY	statu	8211	0	7120.7913130645
8	198858	software embedded engin	Intertek	New York	NY	must	8922	0.625	6965.4841607627
9	57939	data architect	JP Morgan Chase	New York	NY	develop	45838	0	6916.5258924721
10	69388	data architect	Rackspace	Jersey City	NJ	must	8248	0.625	6907.4729126317
11	57996	data architect	JP Morgan Chase	New York	NY	model	7234	0.625	6718.2393312032
12	168814	software engineer	Studio Entertainment	New York	NY	must	8239	0.625	6612.7054590939
13	58229	data architect	JP Morgan Chase	New York	NY	experi	56502	0	6524.9201762475
14	198977	software embedded engin	Intertek	New York	NY	applic	29878	0	6388.9336592192
15	198936	software embedded engin	Intertek	New York	NY	work	46728	0	6377.188719256
16	8632	data analyst	Comcast	New York	NY	model	6658	0.625	6353.6783344899
17	198825	software embedded engin	Intertek	New York	NY	softwar	53856	0	6133.7366411122
18	163761	software engineer	Pixify	New York	NY	statu	7343	0	6023.0564603321
19	226431	machine learning engine	Amazon.com	New York	NY	develop	37459	0	5914.6603462914
20	226562	machine learning engine	Amazon.com	New York	NY	work	37219	0	5818.7996698256
21	58253	data architect	JP Morgan Chase	New York	NY	busi	32169	0	5756.6968933747
22	57955	data architect	JP Morgan Chase	New York	NY	applic	24403	0	5750.0985595376
23	163742	software engineer	Pixify	New York	NY	work	39974	0	5648.4351224678

# Transform - words\_‘col\_index’(col\_index, word, frequency, score, result)

lib.csv - LibreOffice Calc

	A	B	C	D	E	F	G	H	I
1	job	company	city	state	word	frequency	score	result	
2	18405 data architect	JP Morgan Chase	New York	NY	data	39404	0	9623.890105895	
3	19495 software embedded engineer	JP Morgan Chase	New York	NY	developer	46017	0	7722.34298280238	
4	22655 machine learning engineer	Amazon.com	New York	NY	data	40334	0	7634.8332751108	
5	23105 machine learning engineer	Capital One	New York	NY	model	8346	0.625	7268.2089553035	
6	57931 data architect	JP Morgan Chase	New York	NY	work	39422	0	7130.790799629	
7	18855 data embedded engineer	Amazon.com	New York	NY	data	42311	0	7130.790799629	
8	19885 software embedded engineer	Intertek	New York	NY	must	8922	0.625	6995.4941607627	
9	57930 data architect	JP Morgan Chase	New York	NY	developer	45838	0	6916.5258924721	
10	67961 data architect	JP Morgan Chase	Jersey City	NY	must	42310	0	6718.2393113032	
11	57930 data architect	JP Morgan Chase	New York	NY	model	7234	0.625	6718.2393113032	
12	16881 software engineer	Studio Entertainment	New York	NY	must	8239	0.625	6612.7054590939	
13	58223 data architect	JP Morgan Chase	New York	NY	expert	56502	0	6524.9201762475	
14	18855 data embedded engineer	Amazon.com	New York	NY	model	28976	0.625	6524.9201762475	
15	19893 software embedded engineer	Intertek	New York	NY	work	46728	0	6377.188719256	
16	8632 data analyst	Comcast	New York	NY	model	6658	0.625	6353.6783344899	
17	18825 software embedded engineer	Intertek	New York	NY	software	53856	0	6133.736611122	
18	18825 software developer	Intertek	New York	NY	data	7343	0	6028.736611122	
19	22643 machine learning engineer	Amazon.com	New York	NY	developer	37459	0	5914.6603462014	
20	22656 machine learning engineer	Amazon.com	New York	NY	work	37219	0	5818.7996698256	
21	37955 data architect	JP Morgan Chase	New York	NY	expert	24403	0	5750.9895553376	
22	163742 software engineer	Poole	New York	NY	work	39974	0	5648.4351224678	



PycharmProjects Normalization word

	Programming.csv		Resume_words.csv		ability_words.csv
--	-----------------	--	------------------	--	-------------------

ming\_job.csv - LibreOffice Calc

	A	B	C	D	E
1	col_index	word	frequency	score	result
2	machine learning engineer	python	126	0	40.6453712562
3	data architect	application	67	0	33.2552126679
4	software embedded engineer	embedded	154	0	32.6879548409
5	software embedded engineer	java	102	0	31.9209722222
6	machine learning engineer	java	88	0	31.8343553114
7	software developer	java	94	0	30.1060363248
8	machine learning engineer	modeling	45	0	27.6715537723
9	data analyst	modeling	43	0	25.709029433
10	data architect	database	41	0	22.4450542419
11	data architect	sql	93	0	20.7148489011
12	software engineer	experience	141	0	20.3358601391
13	data architect	scala	19	0	19.9213733211
14	software embedded engineer	matlab	27	0	18.6150575397
15	software engineer	application	58	0	17.9254090354
16	machine learning engineer	sql	58	0	17.5772097311
17	data architect	deep	20	0	16.9976254579
18	machine learning engineer	deep	50	0	15.3043711844
19	software embedded engineer	python	42	0	14.8931071429
20	machine learning engineer	spark	27	0	13.5772550366
21	data analyst	python	16	0	12.8740842491
22	software embedded engineer	application	54	0	12.7509559746
23	machine learning engineer	application	30	0	11.8482174908
24	software embedded engineer	c++	85	0	11.5014166908
25	data analyst	scala	7	0	11.1212359944
26	software engineer	java	56	0	9.6563095238
27	software developer	database	20	0	9.4198333333
28	machine learning engineer	matlab	12	0	8.3214285714

► Make csv as 4 way

# Load - cluster mode

## Summary

Security is off.  
Safemode is off.  
411 files and directories, 378 blocks = 789 total filesystem object(s).

Heap Memory used 28.31 MB of 208.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 47.34 MB of 48.35 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	94.52 GB
DFS Used:	3.47 GB (3.68%)
Non DFS Used:	17.34 GB
DFS Remaining:	69.47 GB (73.5%)
Block Pool Used:	3.47 GB (3.68%)
DataNodes usages% (Min/Median/Max/stdDev):	3.68% / 3.68% / 3.68% / 0.00%
Live Nodes	2 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	2018. 6. 13. 오후 8:19:07

## NameNode Journal Status

Current transaction ID: 3407

Journal Manager State

## Browse Directory

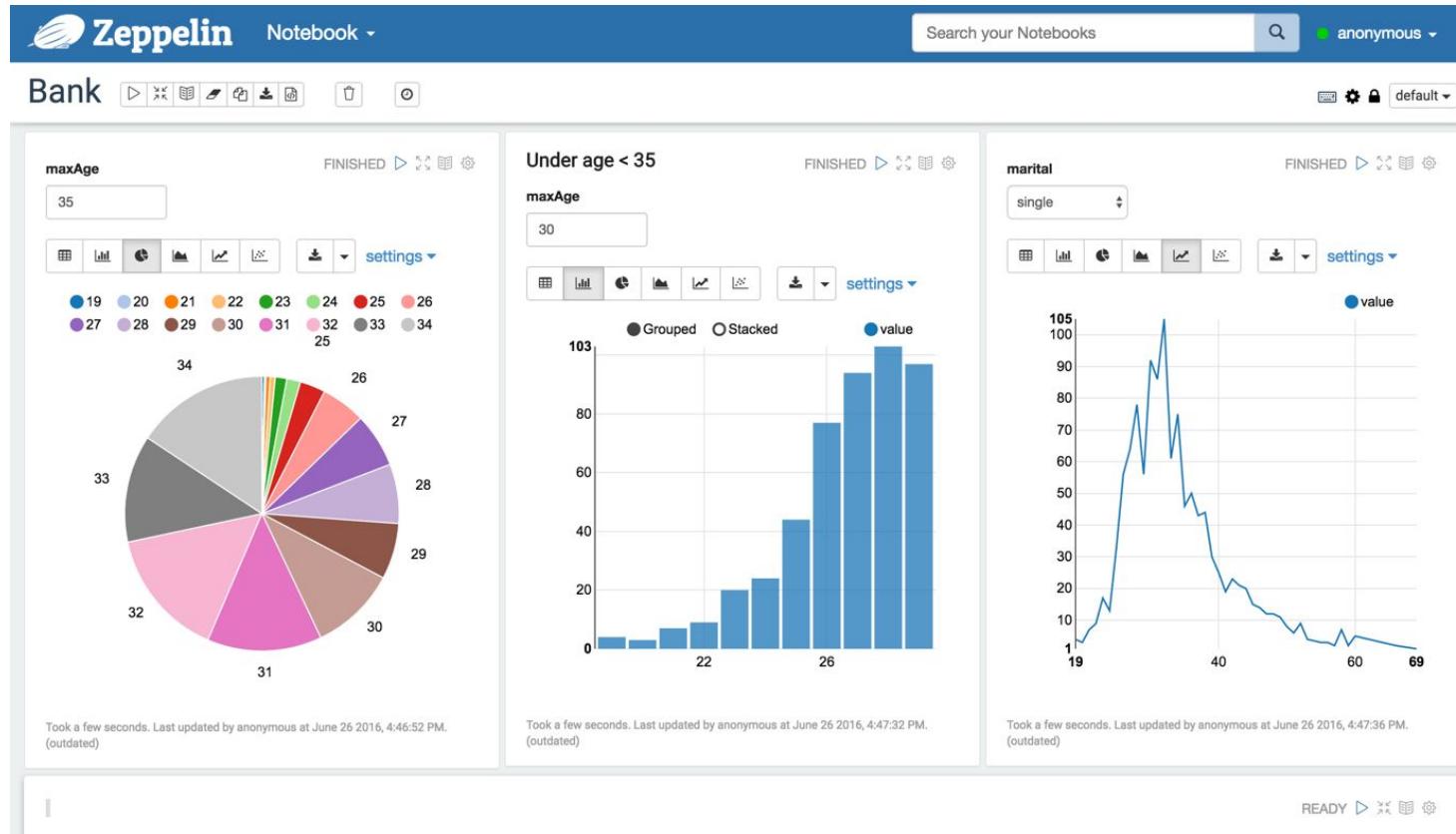
/project\_lastest

Go!

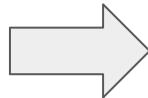
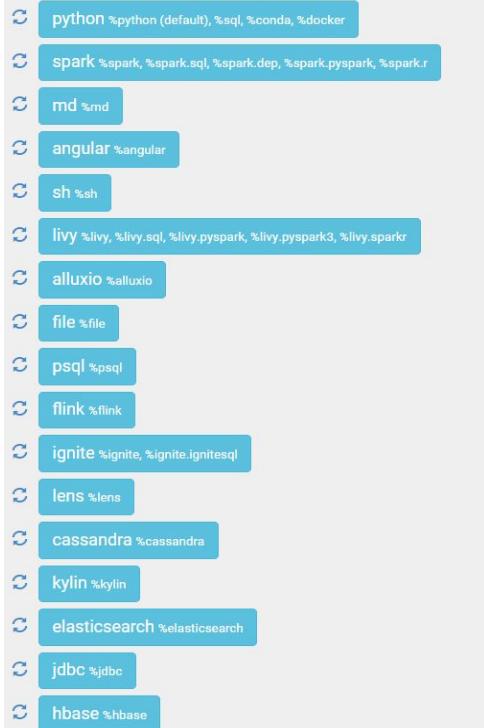
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	project	supergroup	0 B	2018. 6. 14. 오후 1:25:19	0	0 B	city_result
drwxr-xr-x	project	supergroup	0 B	2018. 6. 13. 오후 8:23:24	0	0 B	code
drwxr-xr-x	project	supergroup	0 B	2018. 6. 14. 오후 1:25:19	0	0 B	company_result
drwxr-xr-x	project	supergroup	0 B	2018. 6. 13. 오후 8:29:09	0	0 B	join_data
drwxr-xr-x	project	supergroup	0 B	2018. 6. 13. 오후 8:31:05	0	0 B	output
drwxr-xr-x	project	supergroup	0 B	2018. 6. 14. 오후 1:25:19	0	0 B	state_result

Hadoop, 2017.

# Visualization(with Zeppelin)



# Visualization(with Zeppelin)



python %python, %python.sql, %python.conda, %python.docker ●

## Option

The interpreter will be instantiated  in  process.

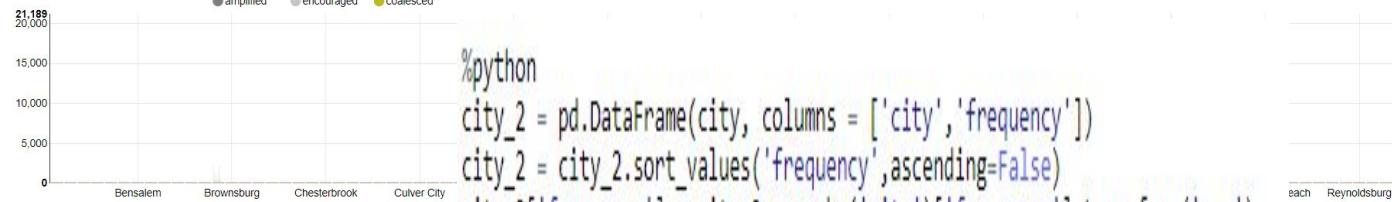
Connect to existing process

Set permission

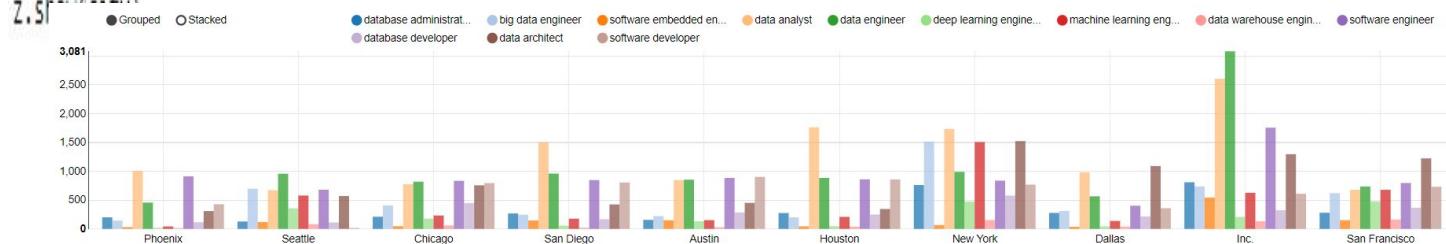
## Properties

name	value
zeppelin.interpreter.localRepo	C:\zeppelin-0.7.3-bin-all\local-repo\2DFBT2WRD
zeppelin.interpreter.output.limit	102400
zeppelin.python	python
zeppelin.python.maxResult	250000

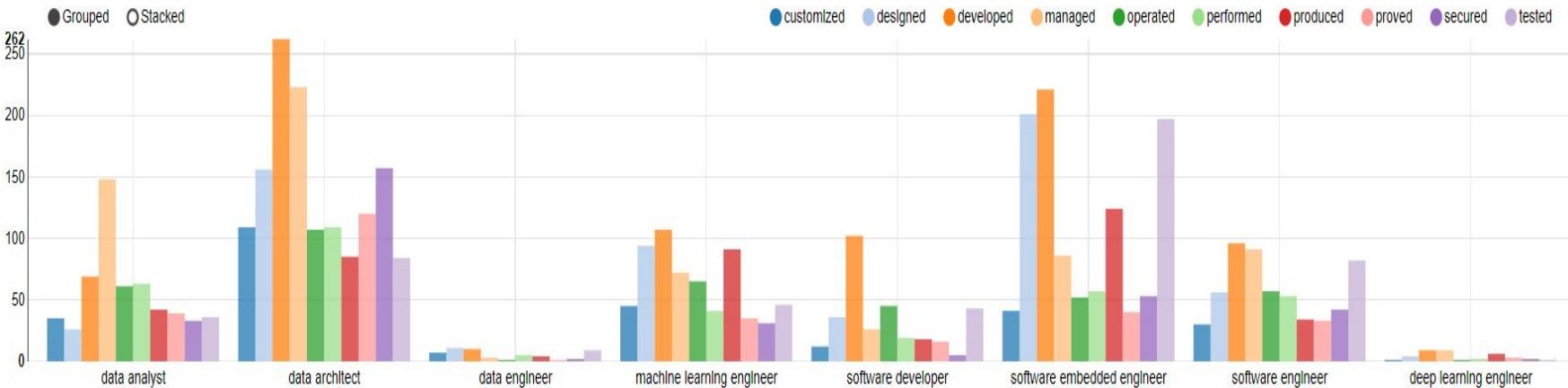
# Visualization(data preparation)



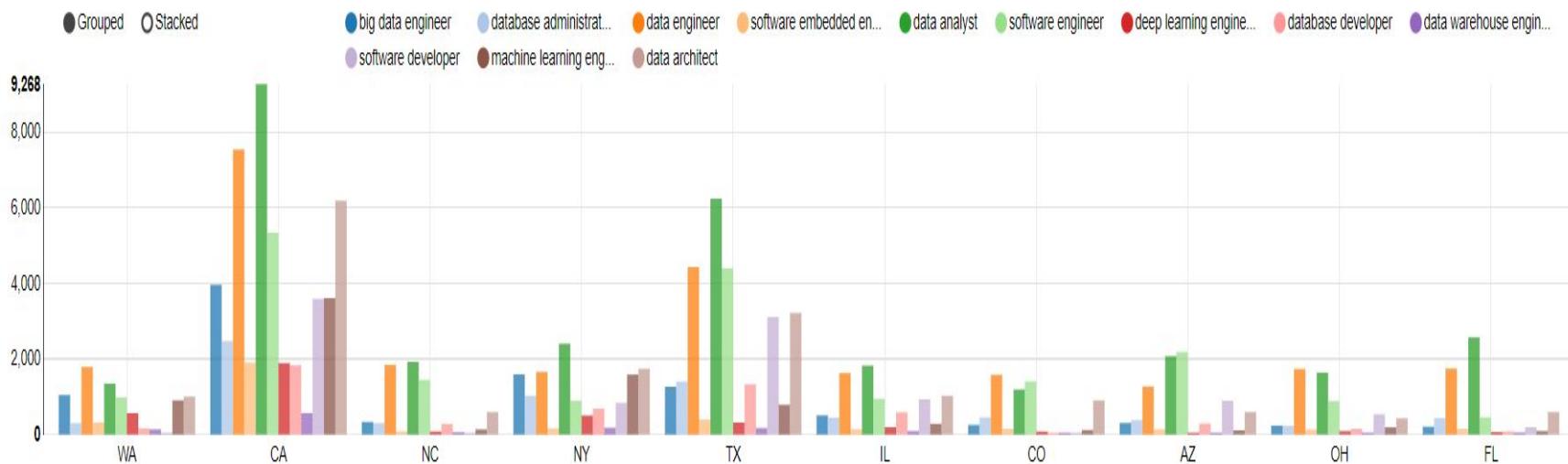
```
city_2 = pd.DataFrame(city, columns = ['city','frequency'])
city_2 = city_2.sort_values('frequency',ascending=False)
city_2['frequency'] = city_2.groupby('city')['frequency'].transform('sum')
city_2 = city_2.drop_duplicates(['city'], keep='first')
city_h = city_2.head(10)
city_h = pd.DataFrame(city_h, columns = ['city'])
city = pd.merge(city,city_h)
```



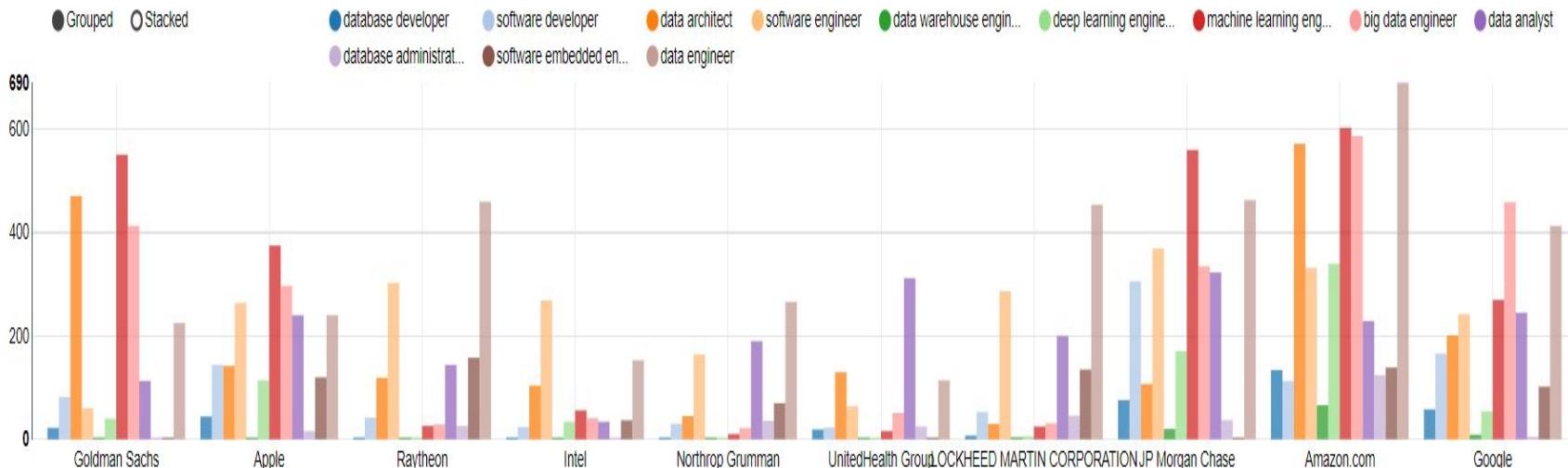
# Visualization



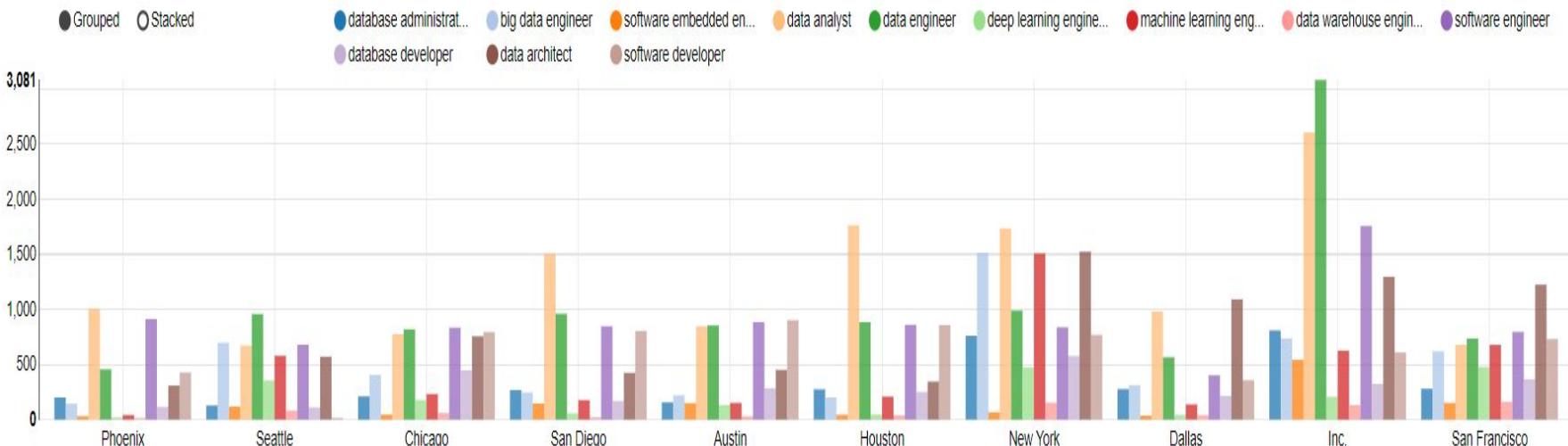
# Visualization



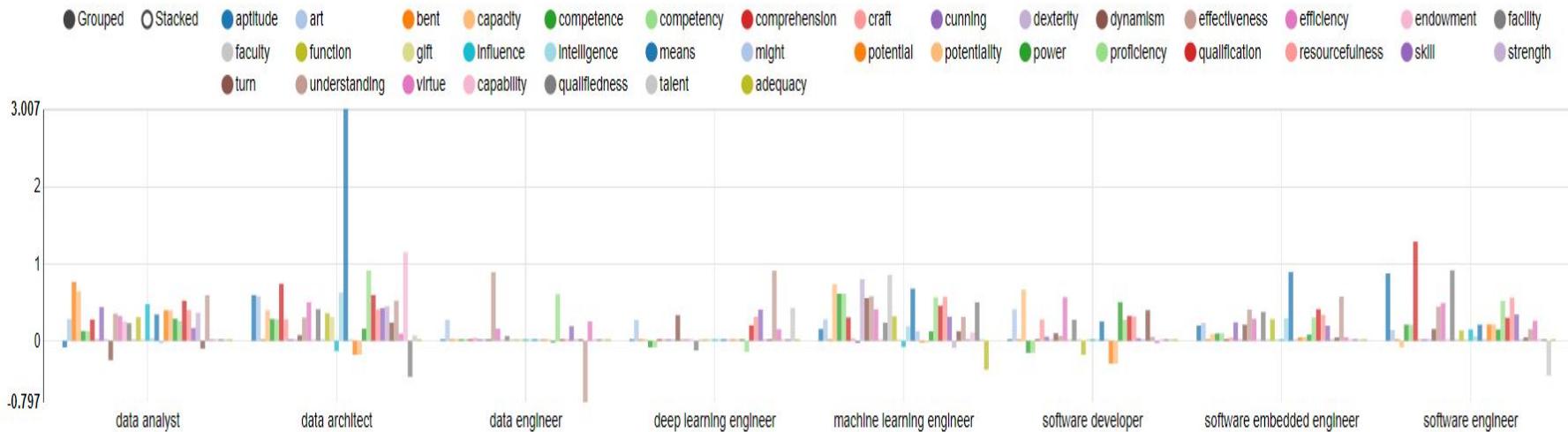
# Visualization



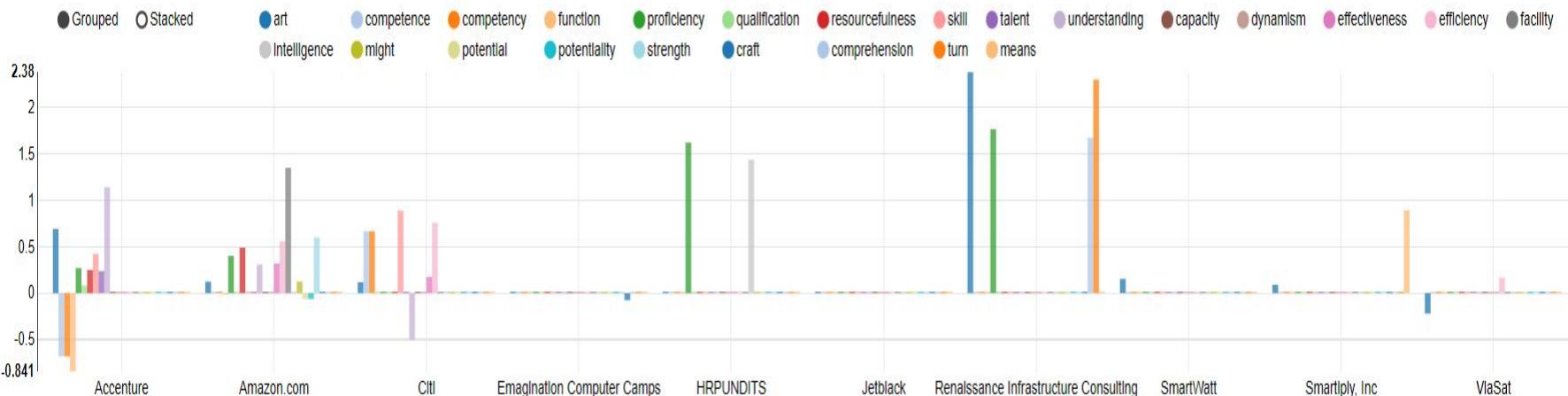
# Visualization



# Visualization



# Visualization



# Visualization

