In [4]:
```python
# Run this cell to set up the notebook, but please don't change it.

# These lines import the Numpy and Datascience modules.
import numpy as np
from datascience import *
from tkinter import *

# These lines do some fancy plotting magic.
import matplotlib
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import warnings
warnings.simplefilter('ignore', FutureWarning)
```

In [5]:
```python
IBM= 'Ability to look at things differently, debug, troubleshoot, design and implement solutions to complex tec
Reddit = 'Required Qualifications: Actively working towards a Bachelor's, Master's, or PhD degree in Computer S
Amazon = 'Basic qualifications Currently enrolled in or will receive a Bachelor's in Computer Science, Computer
Tata='Computer Engineering (CE), Computer Science (CS) or Software Engineering (SE), Information Technology, In
Two_sigma='You will take on the following responsibilities: Independently research and develop hypotheses based
Infosys='Qualifications Basic At least or In pursuit of a 4- year Bachelor's degree required from an accredited
Microsoft='Minimum of a bachelor's degree with 2+ years of relevant experience in investment banking, strategy,
Linkedin='BA/BS degree in engineering, math, statistics, physics, computer science, business or related discipl
Apple='Experience in designed data structures and models for a data lake and cloud data warehouses supporting d
Tesla= 'Currently pursuing an undergrad or master's degree in engineering, Computer Science, Business, and/or o
```

In [6]:
```python
type(IBM)
```

Out[6]:
```
str
```

In [7]:
```python
#Converting List to string
def listToString(s):
    str1 = ""
    for ele in s:
        str1 += ele
    return str1
```

In [8]:
```python
#Changing the Table into array
def converter(sample):
    array=make_array()
    array=sample.split(',')
    listToString(array)
    array=sample.split('.')
    listToString(array)
    array=sample.split(' ')
    listToString(array)
    return array
```

In [9]:
```python
def removing_process(a, column_of_the_table):
    a = a.where(column_of_the_table, are.not_equal_to('the'))
    a = a.where(column_of_the_table, are.not_equal_to('a'))
    a = a.where(column_of_the_table, are.not_equal_to('that'))
    a= a.where(column_of_the_table, are.not_equal_to('and'))
    a= a.where(column_of_the_table, are.not_equal_to('or'))
    a = a.where(column_of_the_table, are.not_equal_to('in'))
    a= a.where(column_of_the_table, are.not_equal_to('of'))
    a= a.where(column_of_the_table, are.not_equal_to('as'))
    a= a.where(column_of_the_table, are.not_equal_to('with'))
    a= a.where(column_of_the_table, are.not_equal_to('such'))
    a= a.where(column_of_the_table, are.not_equal_to('as'))
    a= a.where(column_of_the_table, are.not_equal_to('to'))
    a= a.where(column_of_the_table, are.not_equal_to('and'))
    a= a.where(column_of_the_table, are.not_equal_to('at'))
    a= a.where(column_of_the_table, are.not_equal_to('2022'))
    a= a.where(column_of_the_table, are.not_equal_to('are'))
    a= a.where(column_of_the_table, are.not_equal_to('within'))
    a= a.where(column_of_the_table, are.not_equal_to('is'))
    a= a.where(column_of_the_table, are.not_equal_to('be'))
    a= a.where(column_of_the_table, are.not_equal_to('from'))
    a= a.where(column_of_the_table, are.not_equal_to('-'))
    a= a.where(column_of_the_table, are.not_equal_to('Are'))
    a= a.where(column_of_the_table, are.not_equal_to('on'))
    a= a.where(column_of_the_table, are.not_equal_to('our'))
    a= a.where(column_of_the_table, are.not_equal_to('it'))
    a= a.where(column_of_the_table, are.not_equal_to('You'))
    a= a.where(column_of_the_table, are.not_equal_to('an'))
    a= a.where(column_of_the_table, are.not_equal_to('can'))
    a= a.where(column_of_the_table, are.not_equal_to('(e.g.,'))
    a= a.where(column_of_the_table, are.not_equal_to('you'))
    a= a.where(column_of_the_table, are.not_equal_to('your'))
    a= a.where(column_of_the_table, are.not_equal_to('all'))
    a= a.where(column_of_the_table, are.not_equal_to('All'))
    a= a.where(column_of_the_table, are.not_equal_to('over'))
    a= a.where(column_of_the_table, are.not_equal_to('per'))
    a= a.where(column_of_the_table, are.not_equal_to("Master's"))
    a= a.where(column_of_the_table, are.not_equal_to("As"))
    a= a.where(column_of_the_table, are.not_equal_to("Qualifications:"))
    a= a.where(column_of_the_table, are.not_equal_to("A"))
    a= a.where(column_of_the_table, are.not_equal_to("Bachelor's"))
    a= a.where(column_of_the_table, are.not_equal_to("You're"))
    a= a.where(column_of_the_table, are.not_equal_to("will"))
    return a
```

In [10]:
```python
#ALL IN ONE
main=0
def change_table(company_name,main):
    main =Table().with_column(company_name, converter(main))
    main = main.group(company_name).sort('count',descending=True)
    main = removing_process(main, company_name)
    main = main.take(np.arange(0,40))
    return main
```

In [11]:
```python
a= change_table('Amazon',Amazon).take(np.arange(0,40)).column('Amazon')
b= change_table('IBM',IBM).take(np.arange(0,40)).column('IBM')
c= change_table('reddit',Reddit).take(np.arange(0,40)).column('reddit')
d= change_table('Tata',Tata).take(np.arange(0,40)).column('Tata')
e= change_table('Two_sigma',Two_sigma).take(np.arange(0,40)).column('Two_sigma')
f= change_table('Infosys',Infosys).take(np.arange(0,40)).column('Infosys')
g= change_table('Microsoft',Microsoft).take(np.arange(0,40)).column('Microsoft')
h= change_table('Linkedin',Linkedin).take(np.arange(0,40)).column('Linkedin')
i= change_table('Apple',Apple).take(np.arange(0,40)).column('Apple')
j= change_table('Tesla',Tesla).take(np.arange(0,40)).column('Tesla')
#a.with_column('IBM',b).with_column('reddit', c).with_column('Tata', d).with_column('Two_sigma',e).with_column(
da= np.append(a,b)
da= np.append(da,c)
da= np.append(da,d)
da= np.append(da,e)
da= np.append(da,f)
da= np.append(da,g)
da= np.append(da,h)
Table().with_columns('group',np.append(da,i)).group('group').sort('count',descending=True).show()
```

group	count
Preferred	5
SQL	5
Computer	4
Experience	4
Knowledge	4
Strong	4
business	4
data	4
experience	4
Basic	3
Data	3
ETL	3
Python	3
Science,	3
computer	3
degree	3
2+	2
Ability	2
BI	2
Engineering	2
Excellent	2
Exposure	2
Information	2
Interest	2
Management	2
Mathematics,	2
Physics,	2
Python,	2
Qualifications	2
Software	2
Statistics,	2
Systems,	2
Technology,	2
Willingness	2
able	2
analysis,	2
analytical	2
bachelor's	2
banking,	2
based	2
basic	2
technical	2
work	2
&	1
(AWS,	1
(CE),	1
(CS)	1
(Functional,	1
(SE),	1
(Subject	1
(e.g.	1
(preferable	1
(strong	1
/	1
100%	1
18	1
2023	1
2024	1
3.0	1
4-	1
AWS	1
Actively	1
Agile	1
ApacheSpark),	1
Apple's	1
Applied	1
Artificial	1
At	1
Attend	1
August	1
Azure,	1
BA/BS	1
Bachelor's,	1
Background	1
Build	1
Business	1
Can	1
Candidate	1
Capital,	1
City	1
Cloud	1
Cognitive	1
Concise	1
Conduct	1
Critical	1
Cumulative	1
Curious	1
Currently	1
Degree	1
Demonstrated	1
Description	1
Design	1
Development	1
Development/	1
Domain	1
Engineering,Information	1
Enrolled	1
Extremely	1
Fall	1
Familiar	1
FileMaker	1
Filemaker	1
Finance,	1
Francisco	1
GPA	1
Good	1
Graduating	1
HQ	1
Hadoop	1
High-energy,	1
Highly	1
If	1
In	1
Independently	1
Integration,	1
Intelligence,	1
It's	1
JSON/NoSQL,	1
Java	1
Join	1
KPI	1
KornShell)	1
Lead	1
MATLAB.	1
Management,	1
Master's	1
Master's,	1
Math,	1
Mental	1
Methodologies.	1
Minimum	1
New	1
Operations	1
Partner	1
Perform	1
Performed	1
PhD	1
PhD)	1
Planning	1
Positions	1
Prep,	1
Previous	1
Pro	1
Proactive	1
Produce	1
Production	1
Proficient	1
Programming	1
Python)	1
Python.	1
QuickSight	1
R,	1
R.	1
Reddit	1
Required	1
Research,	1
Respond	1
SAS,	1
SLAs.	1
SPSS,	1
SQL,	1
San	1
Science	1
September	1
Sigma's	1
Start-ups,	1
States	1
Statistics	1
Summer	1
Support	1
System,	1
Systems.	1
Tableau	1
Tableau,	1
Telecom,	1
Testing	1
The	1
Thinking	1
This	1
Two	1
U.S.	1
US.	1
USA.	1
United	1
Venture	1
Visualization,	1
Work	1
Working	1
York	1
abilities	1
ability	1
academic	1
accuracy.	1
achieving	1
across	1
actionable	1
activities	1
adapt	1
advanced	1
advocating	1
aggregate	1
algorithms	1
analysis	1
analysis;	1
analytical,	1
analytical:	1
analytics	1
analyze	1
application.	1
applications	1
apply	1
approach	1
approximately	1
architecture	1
articulately	1
aspects	1
attitude:	1
audiences,	1
audiences/cultures	1
background	1
between	1
big	1
building	1
building/using	1
but	1
can:	1
candidate	1
capable	1
challenging	1
chemistry,	1
circles	1
clearly	1
collaboration	1
collaboratively	1
collect	1
collegial	1
come	1
communicate	1
communications:	1
community.	1
company	1
competitive	1
complex	1
concepts	1
conceptual	1
consulting,	1
corralling	1
costs,	1
cross-functional	1
datasets	1
debate	1
decomposing	1
delivering	1
develop	1
discipline	1
following	1
for	1
good	1
including	1
insights	1
internal	1
investment	1
issues	1
language	1
manufacturing	1
market	1
multiple	1
other	1
pipelines	1
plans	1
problems,	1
pursuit	1
qualifications	1
quantitative	1
real-world	1
related	1
relocate	1
research	1
role	1
science,	1
skills	1
skills.	1
solving	1
strategy	1
towards	1
under	1
understanding	1
using	1
warehouse	1
working	1

In [12]:
```python
change_table('Apple',Apple).show()
```

Apple	count
data	10
for	3
manufacturing	3
&	2
Experience	2
The	2
across	2
analysis,	2
costs,	2
experience	2
insights	2
internal	2
other	2
Apple's	1
BI	1
Build	1
Description	1
ETL	1
FileMaker	1
Filemaker	1
KPI	1
Lead	1
Prep,	1
Pro	1
Produce	1
Python	1
R.	1
SQL,	1
Strong	1
Tableau	1
Tableau,	1
accuracy.	1
actionable	1
advocating	1
aggregate	1
applications	1
architecture	1
based	1
business	1
candidate	1

In [ ]: