main

June 2, 2023

[2]:
```python
import numpy as np
import pandas as pd
from scipy.cluster.hierarchy import dendrogram, linkage, cut_tree
import matplotlib.pyplot as plt


df = pd.read_csv('datafest2018-Updated-April12.csv')
```

[11]:
```python
df
```

[11]:
```
                date      companyId       jobId country stateProvince  \
0         2016-11-01   company00000  job0000000      CA            ON
1         2016-11-01   company00002  job0000002      US            AZ
2         2016-11-01   company00003  job0000003      US            GA
3         2016-11-01   company00005  job0000005      US            AR
4         2016-11-01   company00005  job0000006      US            AR
…                …              …           …     …             …
14586030  2017-11-30  company133804  job1041301      US            NV
14586031  2017-11-30   company36943  job1041302      US            TN
14586032  2017-11-30   company36943  job1041304      US            CA
14586033  2017-11-30  company221821  job1041309      US            CA
14586034  2017-11-30  company182722  job1041311      CA            AB

                            city  avgOverallRating  numReviews industry  \
0                      Cambridge               0.0         NaN      NaN
1                         Peoria               0.0         NaN      NaN
2                   Cartersville               3.7        71.0      NaN
3                        Malvern               5.0        46.0      NaN
4                        Augusta               5.0        46.0      NaN
…                            …                 …          …       …
14586030                    Reno               0.0         NaN      NaN
14586031                 Jackson               3.7        70.0      NaN
14586032           Santa Barbara               3.7        70.0      NaN
14586033  Rancho Santa Margarita               0.0         NaN      NaN
14586034                Edmonton               0.0         NaN      NaN

                                normTitle  … experienceRequired
```

```
0                                   driver  …                NaN  \
1          customer service representative  …                NaN
2                              host/hostess  …                NaN
3                          data entry clerk  …                NaN
4                          data entry clerk  …                NaN
…                                        …  …                  …
14586030  customer service representative  …                1.0
14586031               kitchen team member  …                NaN
14586032                restaurant manager  …                NaN
14586033               hospitality manager  …                2.0
14586034                guest service agent  …                1.0

          estimatedSalary  salaryCurrency  jobLanguage  supervisingJob
0                   40600             NaN           EN             0.0  \
1                   22800             NaN           EN             0.0
2                   22500             NaN           EN             0.0
3                   26100             NaN           EN             0.0
4                   26200             NaN           EN             0.0
…                       …               …            …               …
14586030            34300             NaN           EN             0.0
14586031            25900             NaN           EN             0.0
14586032            46500             NaN           EN             1.0
14586033            60800             NaN           EN             1.0
14586034            32300             NaN           EN             0.0

         licenseRequiredJob educationRequirements  jobAgeDays  clicks
0                       0.0                   NaN          99       4  \
1                       0.0           High School          99      12
2                       0.0                   NaN          99      15
3                       0.0           High School          99      25
4                       0.0           High School          99      33
…                         …                     …           …       …
14586030                0.0           High School           0      46
14586031                1.0           High School           0      17
14586032                0.0           High School           0      28
14586033                0.0      Higher Education           0      24
14586034                0.0           High School           0      26

         localClicks
0                  1
1                  2
2                  3
3                  8
4                  1
…                  …
14586030          12
14586031           3
```

```
14586032          16
14586033           1
14586034           3

[14586035 rows x 23 columns]
```

In the following chunk, I cleaned the data by selecting variables that I think is worth for following prediction, add a new column to generate the salary by making them into USD (the one who does not have any salary currency information will automatically update base on their country), and take out some na observations.

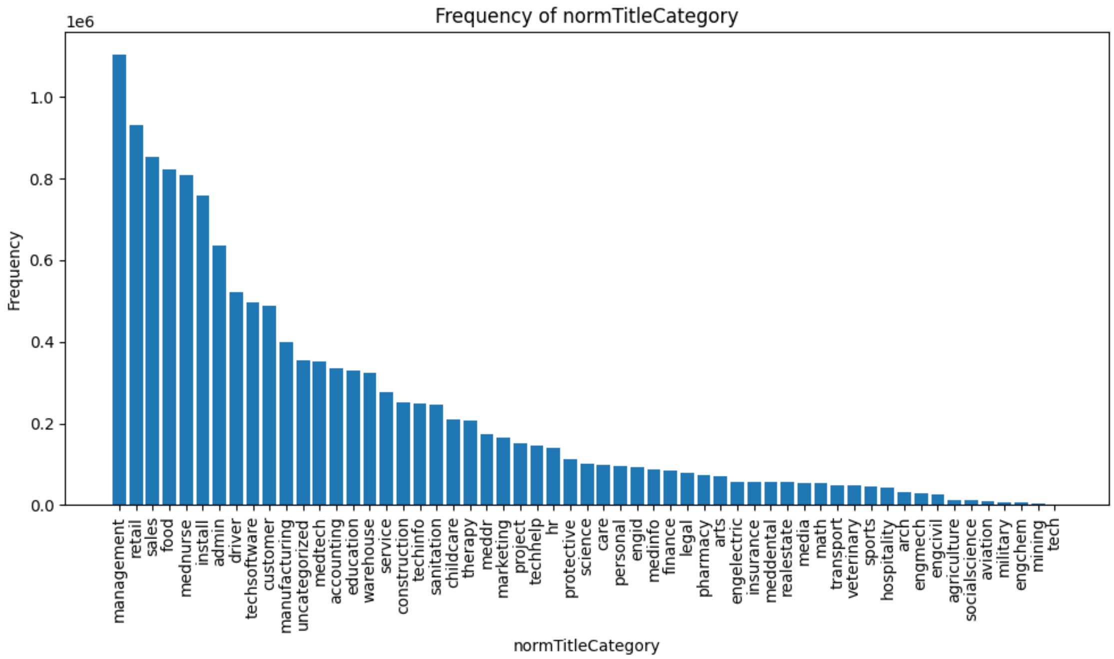
[3]:
```python
# Select columns and mutate 'experienceRequired' and 'count'
cleaned_df = df.drop(columns=['avgOverallRating', 'numReviews', 'descriptionCharacterLength', 'descriptionWordCount',
                              'jobLanguage', 'educationRequirements', 'supervisingJob', 'licenseRequiredJob',
                              'clicks', 'localClicks','industry'])
cleaned_df['experienceRequired'] = cleaned_df['experienceRequired'].fillna(0)
cleaned_df['count'] = np.arange(0, 14586035)

# Update 'salaryCurrency' based on country for US
ex_us = cleaned_df[(cleaned_df['salaryCurrency'].isnull()) & (cleaned_df['country'] == 'US')]['count']
cleaned_df.loc[ex_us, 'salaryCurrency'] = 'USD'

# Update 'salaryCurrency' based on country for CA
ex_ca = cleaned_df[(cleaned_df['salaryCurrency'].isnull()) & (cleaned_df['country'] == 'CA')]['count']
cleaned_df.loc[ex_ca, 'salaryCurrency'] = 'CAD'

# Update 'salaryCurrency' based on country for DE
ex_de = cleaned_df[(cleaned_df['salaryCurrency'].isnull()) & (cleaned_df['country'] == 'DE')]['count']
cleaned_df.loc[ex_de, 'salaryCurrency'] = 'DEM'

# Mutate 'salaryInUSD' based on 'salaryCurrency'
cleaned_df['salaryInUSD'] = np.where(cleaned_df['salaryCurrency'] == 'CAD',
                                     cleaned_df['estimatedSalary'] * 0.73,
                                     np.where(cleaned_df['salaryCurrency'] == 'DEM',
                                              cleaned_df['estimatedSalary'] * 0.58,
                                              cleaned_df['estimatedSalary']))

# Filter rows based on conditions
cleaned_df = cleaned_df[(cleaned_df['normTitle'] != '') &
                        (cleaned_df['stateProvince'] != '') &
```

```
                              (cleaned_df['city'] != '')]

cleaned_df.head()  # Print the resulting DataFrame
```

[3]:
```
         date     companyId       jobId country stateProvince          city  \
0  2016-11-01  company00000  job0000000      CA            ON     Cambridge
1  2016-11-01  company00002  job0000002      US            AZ        Peoria
2  2016-11-01  company00003  job0000003      US            GA  Cartersville
3  2016-11-01  company00005  job0000005      US            AR       Malvern
4  2016-11-01  company00005  job0000006      US            AR       Augusta

                        normTitle normTitleCategory  experienceRequired
0                          driver            driver                 0.0  \
1  customer service representative          customer                 0.0
2                     host/hostess              food                 0.0
3                 data entry clerk             admin                 0.0
4                 data entry clerk             admin                 0.0

   estimatedSalary salaryCurrency  jobAgeDays  count  salaryInUSD
0            40600            CAD          99      0      29638.0
1            22800            USD          99      1      22800.0
2            22500            USD          99      2      22500.0
3            26100            USD          99      3      26100.0
4            26200            USD          99      4      26200.0
```

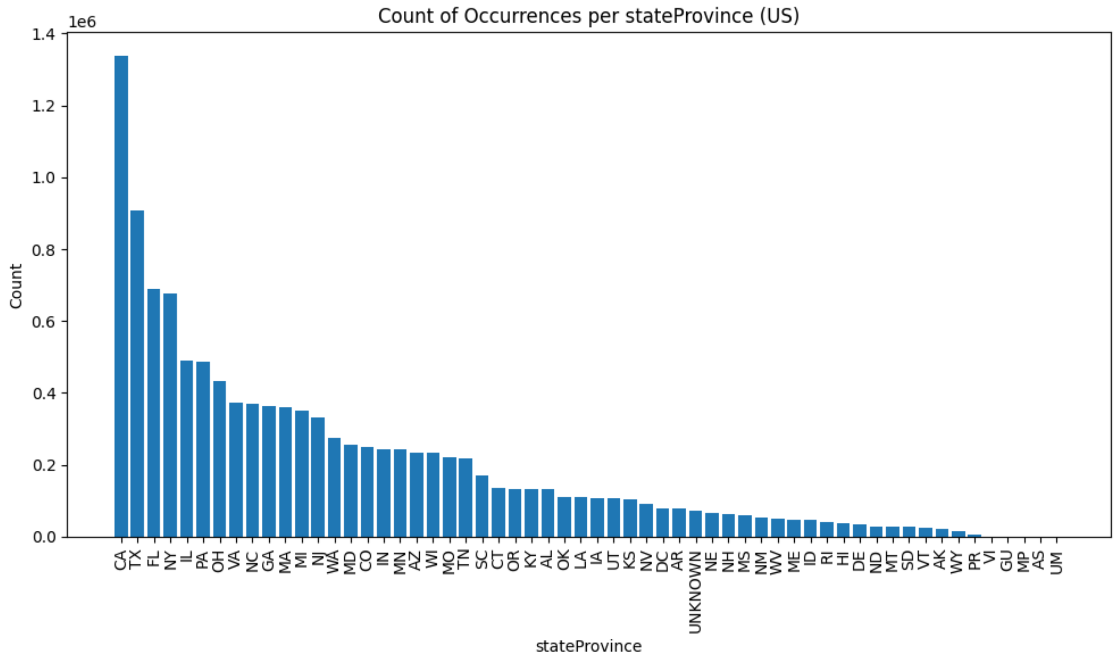
As we can see from the following plot that management is the largest category that has been offered from the dataset.

[4]:
```
selected_column = cleaned_df['normTitleCategory']


frequency_table = selected_column.value_counts().reset_index()


frequency_table.columns = ['normTitleCategory', 'Freq']


frequency_table = frequency_table.sort_values('Freq', ascending=False)

plt.figure(figsize=(10, 6))
plt.bar(frequency_table['normTitleCategory'], frequency_table['Freq'])
plt.xticks(rotation=90)
plt.xlabel('normTitleCategory')
plt.ylabel('Frequency')
plt.title('Frequency of normTitleCategory')
plt.tight_layout()
plt.show()
```



California and Texas has the most opportunity among all of the states in US.

[6]:
```
filtered_df = cleaned_df[cleaned_df['country'] == 'US']

state_province_counts = filtered_df['stateProvince'].value_counts().
↪reset_index()
state_province_counts.columns = ['stateProvince', 'Count']
state_province_counts = state_province_counts.sort_values('Count',␣
↪ascending=False)

plt.figure(figsize=(10, 6))
plt.bar(state_province_counts['stateProvince'], state_province_counts['Count'])
plt.xticks(rotation=90)
plt.xlabel('stateProvince')
plt.ylabel('Count')
plt.title('Count of Occurrences per stateProvince (US)')
plt.tight_layout()
plt.show()
```

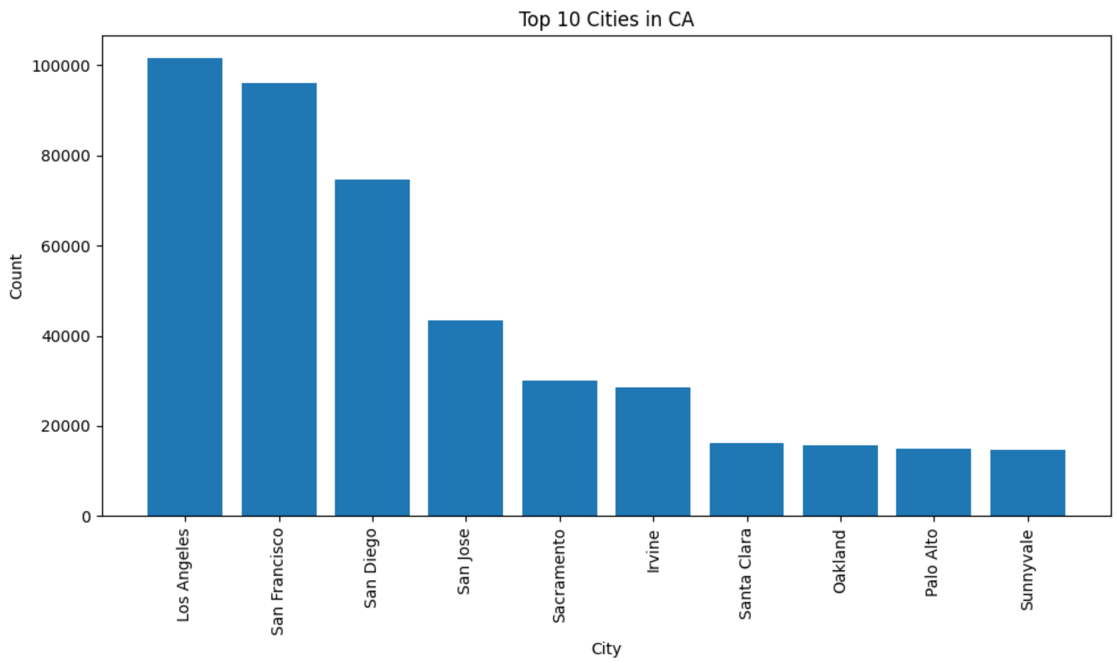


Los Angeles and San Francisco has the most opportunity among all of the city in California.

[7]:
```
CA_filtered = cleaned_df[cleaned_df['stateProvince'] == 'CA']

CA_city_counts = CA_filtered['city'].value_counts().reset_index()
CA_city_counts.columns = ['city', 'Count']
CA_city_counts = CA_city_counts.sort_values('Count', ascending=False).head(10)

print(CA_city_counts)

plt.figure(figsize=(10, 6))
plt.bar(CA_city_counts['city'], CA_city_counts['Count'])
plt.xticks(rotation=90)
plt.xlabel('City')
plt.ylabel('Count')
plt.title('Top 10 Cities in CA')
plt.tight_layout()
plt.show()
```

```
            city   Count
0    Los Angeles  101464
1  San Francisco   96072
2      San Diego   74685
3       San Jose   43410
4     Sacramento   29960
5         Irvine   28520
```

```
6    Santa Clara   16090
7        Oakland   15793
8      Palo Alto   14860
9      Sunnyvale   14753
```

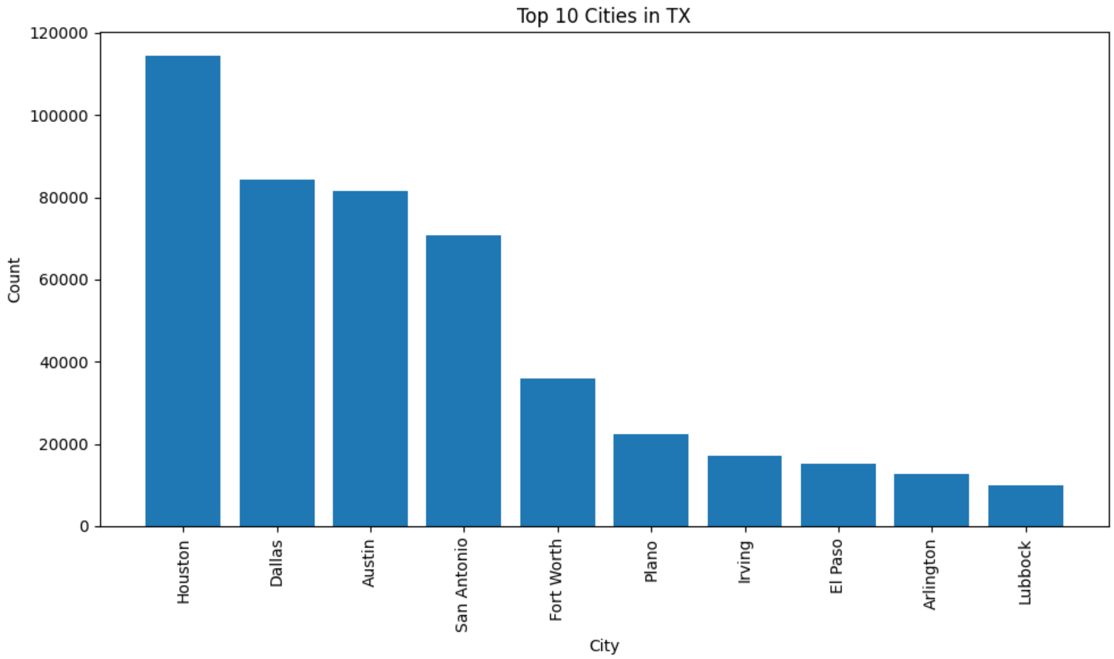


Houston and Dallas has the most opportunity among all of the city in Texas.

[9]:
```python
TX_filtered = cleaned_df[cleaned_df['stateProvince'] == 'TX']

TX_city_counts = TX_filtered['city'].value_counts().reset_index()
TX_city_counts.columns = ['city', 'Count']
TX_city_counts = TX_city_counts.sort_values('Count', ascending=False).head(10)

print(TX_city_counts)

plt.figure(figsize=(10, 6))
plt.bar(TX_city_counts['city'], TX_city_counts['Count'])
plt.xticks(rotation=90)
plt.xlabel('City')
plt.ylabel('Count')
plt.title('Top 10 Cities in TX')
plt.tight_layout()
plt.show()
```

```
          city   Count
0      Houston  114431
```

```
1          Dallas   84414
2          Austin   81542
3     San Antonio   70739
4      Fort Worth   36029
5           Plano   22457
6          Irving   17173
7         El Paso   15134
8       Arlington   12585
9         Lubbock   10045
```

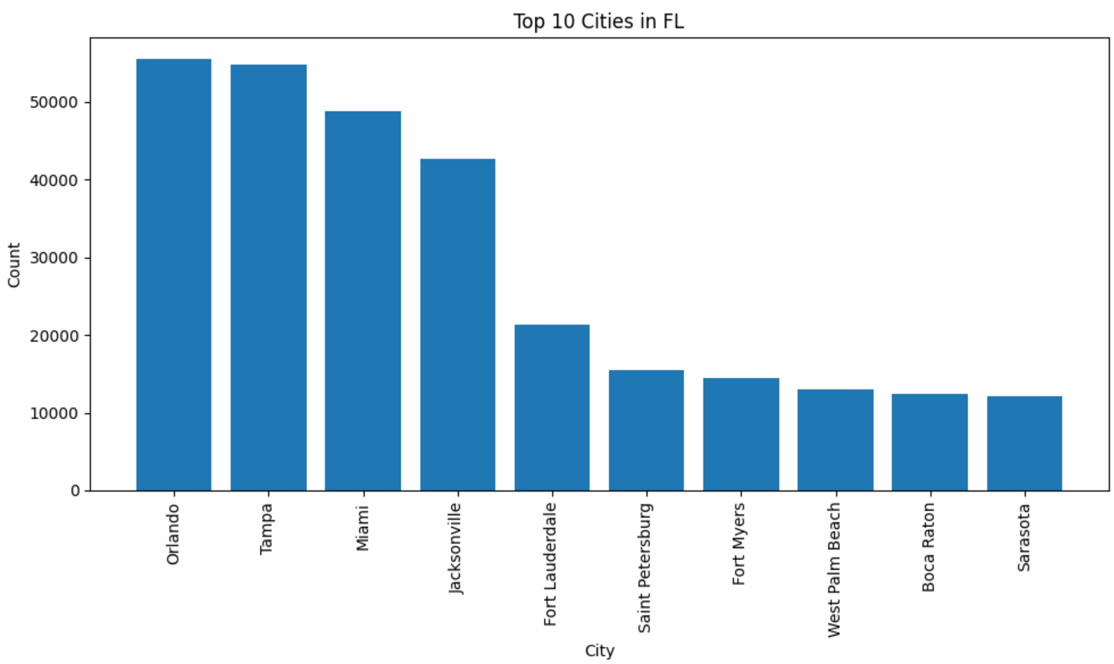


Orlando and Tampa has the most opportunity among all of the city in Florida.

[10]:
```
FL_filtered = cleaned_df[cleaned_df['stateProvince'] == 'FL']

FL_city_counts = FL_filtered['city'].value_counts().reset_index()
FL_city_counts.columns = ['city', 'Count']
FL_city_counts = FL_city_counts.sort_values('Count', ascending=False).head(10)

print(FL_city_counts)

plt.figure(figsize=(10, 6))
plt.bar(FL_city_counts['city'], FL_city_counts['Count'])
plt.xticks(rotation=90)
plt.xlabel('City')
plt.ylabel('Count')
plt.title('Top 10 Cities in FL')
```

```
plt.tight_layout()
plt.show()
```

```
                city  Count
0            Orlando  55449
1              Tampa  54771
2              Miami  48801
3       Jacksonville  42598
4    Fort Lauderdale  21316
5  Saint Petersburg  15422
6         Fort Myers  14402
7    West Palm Beach  12943
8         Boca Raton  12348
9           Sarasota  12161
```

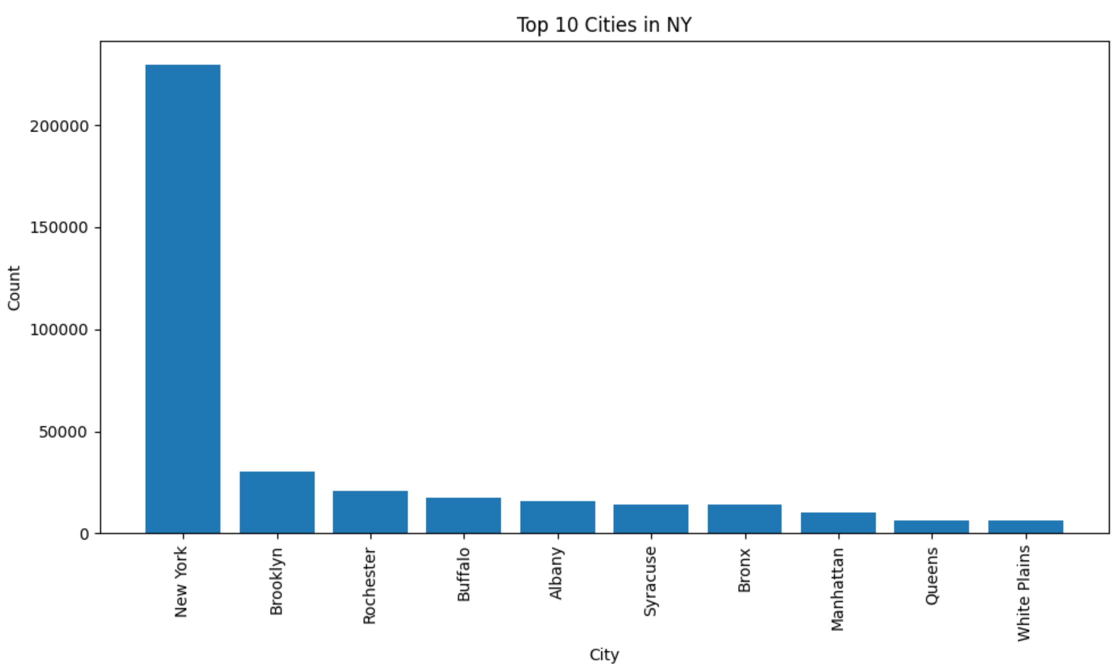


Most of the job has been offered in New York

[11]:
```
NY_filtered = cleaned_df[cleaned_df['stateProvince'] == 'NY']

NY_city_counts = NY_filtered['city'].value_counts().reset_index()
NY_city_counts.columns = ['city', 'Count']
NY_city_counts = NY_city_counts.sort_values('Count', ascending=False).head(10)

print(NY_city_counts)

plt.figure(figsize=(10, 6))
```

```
plt.bar(NY_city_counts['city'], NY_city_counts['Count'])
plt.xticks(rotation=90)
plt.xlabel('City')
plt.ylabel('Count')
plt.title('Top 10 Cities in NY')
plt.tight_layout()
plt.show()
```

```
           city   Count
0      New York  229557
1      Brooklyn   30439
2     Rochester   20981
3       Buffalo   17698
4        Albany   15535
5      Syracuse   14305
6         Bronx   14005
7     Manhattan   10152
8        Queens    6164
9  White Plains    6141
```

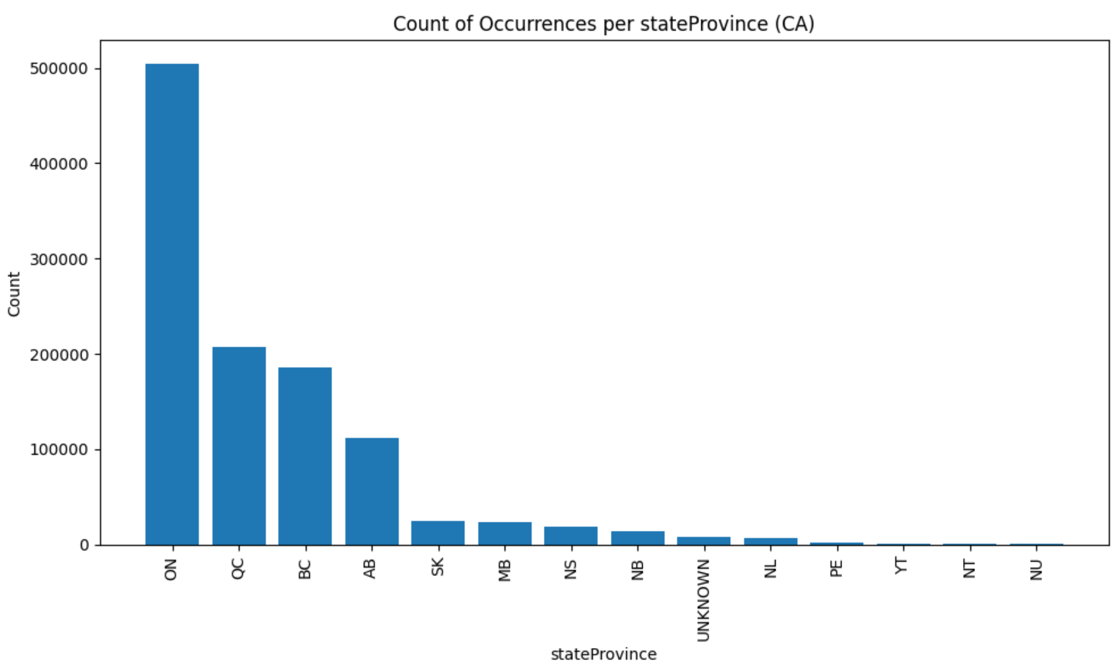


Ontario has the most opportunity among all of the states in Canada.

[12]:
```
CA_filtered = cleaned_df[cleaned_df['country'] == 'CA']

CA_state_counts = CA_filtered['stateProvince'].value_counts().reset_index()
CA_state_counts.columns = ['stateProvince', 'Count']
```

```
CA_state_counts = CA_state_counts.sort_values('Count', ascending=False)

print(CA_state_counts)

plt.figure(figsize=(10, 6))
plt.bar(CA_state_counts['stateProvince'], CA_state_counts['Count'])
plt.xticks(rotation=90)
plt.xlabel('stateProvince')
plt.ylabel('Count')
plt.title('Count of Occurrences per stateProvince (CA)')
plt.tight_layout()
plt.show()
```

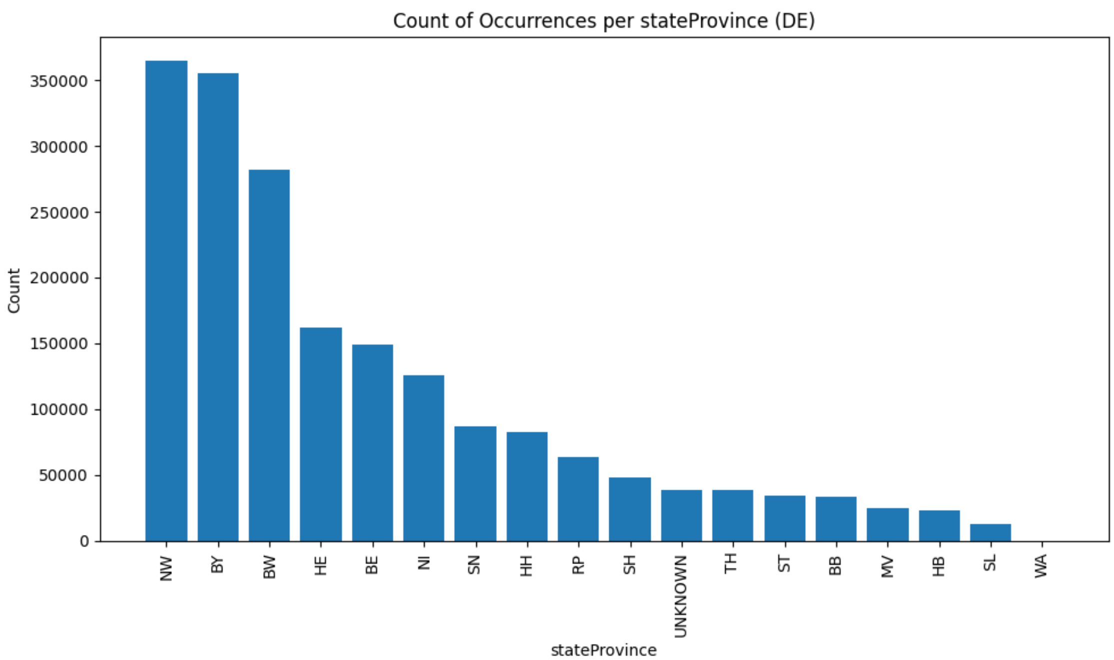
```
   stateProvince   Count
0             ON  503744
1             QC  207004
2             BC  185938
3             AB  112310
4             SK   24883
5             MB   23907
6             NS   19434
7             NB   13996
8        UNKNOWN    8719
9             NL    7205
10            PE    2316
11            YT    1234
12            NT     861
13            NU     785
```



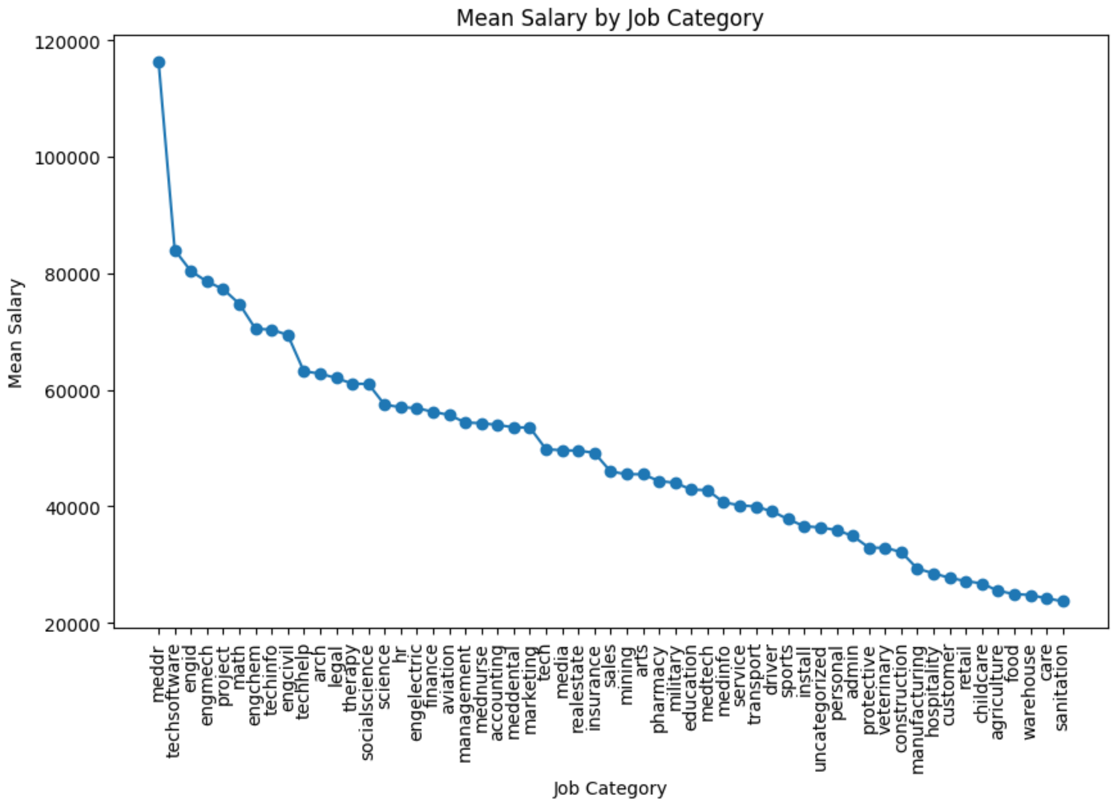
NW and BY has the most opportunity among all of the states in Germany.

[13]:
```
DE_filtered = cleaned_df[cleaned_df['country'] == 'DE']

DE_state_counts = DE_filtered['stateProvince'].value_counts().reset_index()
DE_state_counts.columns = ['stateProvince', 'Count']
DE_state_counts = DE_state_counts.sort_values('Count', ascending=False)

print(DE_state_counts)

plt.figure(figsize=(10, 6))
plt.bar(DE_state_counts['stateProvince'], DE_state_counts['Count'])
plt.xticks(rotation=90)
plt.xlabel('stateProvince')
plt.ylabel('Count')
plt.title('Count of Occurrences per stateProvince (DE)')
plt.tight_layout()
plt.show()
```

```
   stateProvince   Count
0             NW  364326
1             BY  354947
2             BW  282204
3             HE  162025
4             BE  148693
5             NI  125528
6             SN   87099
7             HH   82884
8             RP   63841
9             SH   47800
10       UNKNOWN   38795
11            TH   38577
12            ST   34330
13            BB   33735
14            MV   24935
15            HB   23301
16            SL   12774
17            WA       1
```



[24]:
```
salary_mean = cleaned_df.groupby('normTitleCategory')['estimatedSalary'].mean().
  ↪sort_values(ascending=False)

plt.figure(figsize=(10, 6))
plt.plot(salary_mean.index, salary_mean.values, marker='o')

plt.xlabel('Job Category')
plt.ylabel('Mean Salary')
plt.title('Mean Salary by Job Category')

plt.xticks(rotation=90)

plt.show()
```



[18]:

[18]:

```
                date      companyId       jobId country stateProvince
5479682   2017-01-05   company26274  job0222547      US            NC   \
5463205   2017-01-02   company30676  job0208315      US            MA
5463207   2017-01-04   company30676  job0208315      US            MA
5463206   2017-01-03   company30676  job0208315      US            MA
5463201   2016-12-29   company30676  job0208315      US            MA
…                …              …          …       …             …
12691273  2017-09-30  company214839  job0883165      US            IN
12691297  2017-10-24  company214839  job0883165      US            IN
12691298  2017-10-25  company214839  job0883165      US            IN
12691299  2017-10-26  company214839  job0883165      US            IN
12691287  2017-10-14  company214839  job0883165      US            IN

                city                normTitle normTitleCategory
5479682   Greenville           division chief        management   \
5463205       Woburn      director of finance        management
5463207       Woburn      director of finance        management
5463206       Woburn      director of finance        management
5463201       Woburn      director of finance        management
```

```
…             …                      …                 …
12691273        NaN  front office mitarbeiter       hospitality
12691297        NaN  front office mitarbeiter       hospitality
12691298        NaN  front office mitarbeiter       hospitality
12691299        NaN  front office mitarbeiter       hospitality
12691287        NaN  front office mitarbeiter       hospitality

          experienceRequired  estimatedSalary salaryCurrency  salaryInUSD
5479682                  0.0           208100            USD     208100.0  \
5463205                  0.0           202400            USD     202400.0
5463207                  0.0           202400            USD     202400.0
5463206                  0.0           202400            USD     202400.0
5463201                  0.0           202400            USD     202400.0
…                          …               …             …           …
12691273                 0.0                0            USD          0.0
12691297                 0.0                0            USD          0.0
12691298                 0.0                0            USD          0.0
12691299                 0.0                0            USD          0.0
12691287                 0.0                0            USD          0.0

          categoryScore  salaryScore  totalScore
5479682        0.502375     0.418039    0.920414
5463205        0.502375     0.406589    0.908964
5463207        0.502375     0.406589    0.908964
5463206        0.502375     0.406589    0.908964
5463201        0.502375     0.406589    0.908964
…                   …          …           …
12691273       0.018838     0.000000    0.018838
12691297       0.018838     0.000000    0.018838
12691298       0.018838     0.000000    0.018838
12691299       0.018838     0.000000    0.018838
12691287       0.018838     0.000000    0.018838

[10802021 rows x 15 columns]
```

[23]:

```

```

[23]:

```
                     count      mean       std       min       25%       50%
normTitleCategory
management        905199.0  0.616807  0.064711  0.502375  0.563243  0.596790  \
sales             626014.0  0.491613  0.046956  0.388470  0.454360  0.483488
mednurse          739435.0  0.481426  0.044234  0.382537  0.448427  0.482979
retail            826298.0  0.478212  0.025062  0.450181  0.463239  0.469868
food              695045.0  0.425312  0.019201  0.374695  0.413466  0.419492
techsoftware      376550.0  0.420960  0.043940  0.226485  0.394022  0.421744
install           606338.0  0.420568  0.025071  0.345269  0.404530  0.416583
admin             472995.0  0.363611  0.036258  0.289570  0.341398  0.351844
```

meddr	143801.0	0.345460	0.107898	0.114657	0.250454	0.352302
driver	450589.0	0.319646	0.039293	0.237652	0.292494	0.314189
customer	423174.0	0.278866	0.027457	0.222861	0.263439	0.272278
accounting	251419.0	0.273401	0.051411	0.153198	0.230136	0.264286
techinfo	195802.0	0.269230	0.046632	0.113350	0.239505	0.268834
medtech	327367.0	0.246835	0.043713	0.191153	0.220281	0.234142
manufacturing	208380.0	0.245576	0.028740	0.181183	0.227386	0.236426
uncategorized	285582.0	0.238386	0.052965	0.161371	0.202150	0.216011
education	301610.0	0.237353	0.032336	0.180776	0.214524	0.235818
project	131355.0	0.228485	0.045726	0.069051	0.199425	0.229959
therapy	192901.0	0.219549	0.034718	0.127418	0.196522	0.221030
engid	82528.0	0.214703	0.041493	0.042068	0.190722	0.213422
service	259455.0	0.207518	0.037866	0.157107	0.179204	0.198891
techhelp	121163.0	0.202794	0.051800	0.096768	0.156732	0.205446
warehouse	232490.0	0.197374	0.018013	0.147196	0.188176	0.193600
marketing	134863.0	0.190282	0.056381	0.075328	0.145436	0.178381
construction	121715.0	0.184454	0.029510	0.114702	0.166128	0.177980
engelectric	32400.0	0.181080	0.040822	0.068236	0.159839	0.183342
hr	120952.0	0.180774	0.048687	0.098271	0.143068	0.169986
math	48855.0	0.176619	0.056452	0.069921	0.133802	0.168555
legal	60778.0	0.176089	0.055904	0.072975	0.131031	0.168596
engmech	25369.0	0.174032	0.039045	0.057305	0.152725	0.175827
finance	54927.0	0.171237	0.067273	0.038722	0.110840	0.160860
science	86644.0	0.169273	0.061365	0.045928	0.120054	0.155209
engcivil	20408.0	0.166026	0.040832	0.050976	0.140972	0.169498
sanitation	204099.0	0.159816	0.014074	0.111678	0.149846	0.155672
arch	21036.0	0.158126	0.060323	0.055435	0.110276	0.142819
childcare	177678.0	0.150230	0.010531	0.095686	0.144100	0.148921
engchem	5140.0	0.149227	0.048209	0.048919	0.124050	0.153379
meddental	47671.0	0.140579	0.064298	0.066430	0.095156	0.117254
arts	44220.0	0.139579	0.047895	0.032450	0.098943	0.134098
realestate	46417.0	0.128849	0.043795	0.063340	0.092268	0.120994
socialscience	10315.0	0.127696	0.049726	0.043403	0.092620	0.118534
media	48539.0	0.127030	0.040220	0.059488	0.097656	0.122164
insurance	51891.0	0.126398	0.041968	0.066725	0.093040	0.116142
pharmacy	62697.0	0.123130	0.047040	0.067875	0.092985	0.102025
medinfo	83434.0	0.121549	0.036800	0.076114	0.099014	0.107451
aviation	8931.0	0.118518	0.050957	0.046812	0.076543	0.103260
protective	99464.0	0.117782	0.041032	0.082957	0.093202	0.102844
transport	36307.0	0.117201	0.055309	0.053413	0.077318	0.095196
personal	79647.0	0.116456	0.027303	0.078073	0.095148	0.111219
tech	1279.0	0.097793	0.029339	0.052713	0.071998	0.095502
sports	40627.0	0.096082	0.027038	0.049865	0.077587	0.090042
care	86256.0	0.094397	0.017195	0.071602	0.084861	0.090084
military	6747.0	0.091563	0.041177	0.038489	0.073041	0.077260
mining	1859.0	0.087643	0.051672	0.039889	0.053349	0.066205
veterinary	43246.0	0.086908	0.036477	0.051995	0.064650	0.074695

hospitality	29025.0	0.080042	0.033363	0.018838	0.060420	0.070063
agriculture	3095.0	0.078830	0.033653	0.035837	0.057332	0.070791

	75%	max
normTitleCategory		
management	0.659466	0.920414
sales	0.518442	0.772560
mednurse	0.507487	0.774060
retail	0.483528	0.776819
food	0.429737	0.647094
techsoftware	0.450068	0.592093
install	0.431449	0.742015
admin	0.373137	0.706404
meddr	0.451338	0.578899
driver	0.336487	0.676785
customer	0.282322	0.591884
accounting	0.310691	0.565612
techinfo	0.298364	0.499850
medtech	0.258650	0.585086
manufacturing	0.252898	0.440524
uncategorized	0.250363	0.533609
education	0.255103	0.496967
project	0.260494	0.452739
therapy	0.241520	0.484590
engid	0.242149	0.353639
service	0.226211	0.533563
techhelp	0.242811	0.408540
warehouse	0.200430	0.489703
marketing	0.223178	0.427477
construction	0.193247	0.508032
engelectric	0.207046	0.306484
hr	0.210565	0.394775
math	0.209736	0.418454
legal	0.209978	0.400818
engmech	0.198727	0.285911
finance	0.218112	0.404934
science	0.207840	0.445687
engcivil	0.192599	0.286613
sanitation	0.166118	0.294081
arch	0.199870	0.332654
childcare	0.154746	0.436787
engchem	0.181704	0.241567
meddental	0.166671	0.413758
arts	0.167093	0.317304
realestate	0.159162	0.322481
socialscience	0.159916	0.383901
media	0.146270	0.318629

insurance	0.146476	0.321646
pharmacy	0.140796	0.363375
medinfo	0.127540	0.386479
aviation	0.160110	0.258744
protective	0.123334	0.382274
transport	0.150038	0.326414
personal	0.135124	0.308487
tech	0.114787	0.161392
sports	0.108925	0.263405
care	0.098521	0.371924
military	0.091121	0.363519
mining	0.111002	0.264879
veterinary	0.092774	0.389681
hospitality	0.080710	0.300075
agriculture	0.088670	0.205785