Job Recommendation System

(Machine Learning for Data Science - Final project)

By

GunjanBatra

Anita Chimnani

SreevarshaPutcha

INDEX

1.INTRODUCTION

PROBLEM STATEMENT:

Develop a **Job Recommendation System** to predict what jobs users will apply to base on their Previous applications, demographic information, and work history.

DATASET DESCRIPTION:

We will use the data provided by CareerBuilder.com for Job Recommendation Challenge at www.kaggle.com**.** This data contains various file containing the timing of each window, the users, their corresponding Window ID and demographic and professional information and user's work history. Also, we are given information about job postings and applications made by users to jobs. We have data for users, job postings, and job applications that users have made to job postings. In total, the applications span 13 weeks. The given data has split the applications into 7 groups, and therefore each group represents a 13-day window. In each window, users have been split into two groups, Test and Train. The Test users are those who made 5 or more applications in the 4-day test period, and the Train users are those who did not.
In each window, we are given all the job applications that users in that window made to jobs in that window during the 9-day training period. This data is in apps.tsv.
In this project, we limited the scope to predicting which jobs in a given window the users should apply to.

MODELS:

Recommendation Algorithm - We have used **Content Based Filtering** (CBF) in which features of items are abstract and compared with a profile of the user's preference. This algorithm tries to recommend items that are similar to those that a user liked in the past. It is widely applied in Information retrieval (IR).

SIMILARITY CALCULATION METHODS:

To calculate the similarity between the job applications have used **Cosine similarity.**

Tools/Coding Languages used – R

2. DATA PREPROCESSING:

Data preprocessing is a technique that involves transforming raw data into an understandable format. Our given data is incomplete, inconsistent, and lacks in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

As part of data preprocessing we started off by making sure that all our data is complete and consistent.

Completeness Check: We checked and removed all the blank spaces and NAs a few predefined functions in R along with a few regular expressions to identify the spaces which were not identifiable upfront by the functions in R.

Consistency Check: at the prima facie we removed all the punctuation marks, unintelligible characters and unnecessary numeric data in categorical information and vice versa from our data by using regular expressions and the *gsub*() function in R.

Since our data consists of information in which the term "NA" could actually mean something, it was not an option for us to substitute all our "to be removed" data with "NA", instead we used regular expressions to replace these characters by a space and then removed all the white spaces.

Having done all these we reduced our user's data and job's data from an initial total row count of 2,901,222 rows over 36 rows, And after our preliminary processing it came down to 2,568,371 rows over 19 columns for users data and 151 rows for jobs data. The reduction in jobs data is much greater because we only considered the first 1000 rows in window 6 to be able to process our computations in the limited amount of time.

After the primary processing we started exploring the data using visualization techniques in R and eventually came across details which could be potential outliers and incorrect information which might influence our models in an incorrect manner.

- So we removed the users who graduated before 1977 and those who have a work experience of more than 40 years, because these people were clearly the outliers in our context.
- We removed all the state names which are longer than 30 characters, because so far the longest state name is 26 characters.
- We also removed all the country names and state names which were wrongly entered into their respective columns, we took the list of states and countries in existence and compared them against our data and did the data cleaning using the sql functions in R, to make sure our analysis was correct.

We created dummy variables for columns which said yes/ no or had minimum categories to be able to find the correlation between them and other important variables.

3. DATA EXPLORATION AND ANALYSIS

As part of exploring the data we started creating visual representations of the data in the form of graphs and charts.

Our data is primarily split into two parts:

1. Data related to users: users data + user history + apps (i.e. applications sent by users)

2. Data related to the jobs posted: we used the first 1000 rows of jobs data 6.

USERS DATA

Summary Tables:

We created summary variables which would give a tabular summary of details like*:

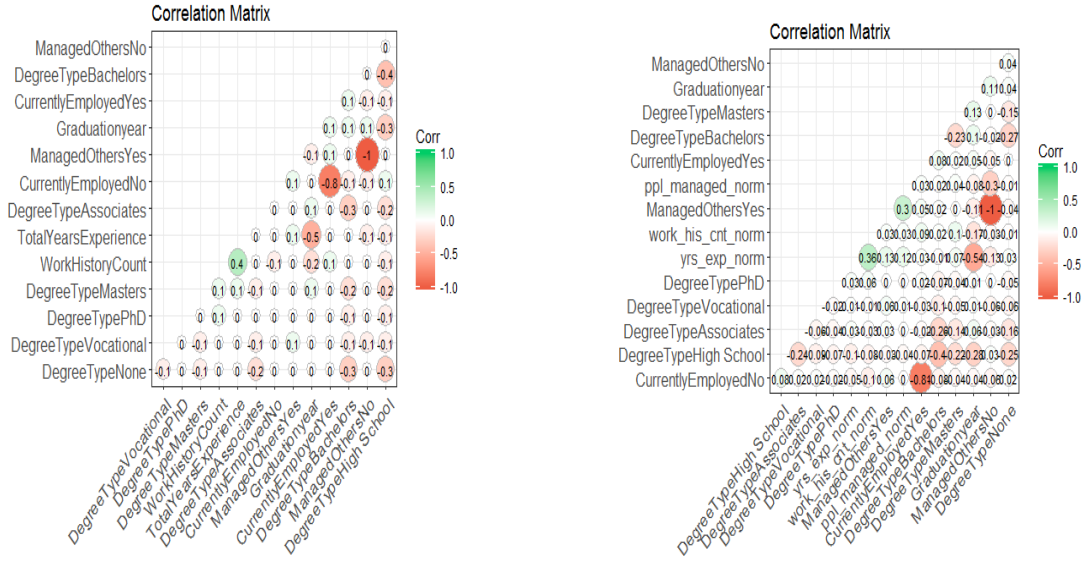
- Number of applicants grouped by Zip code
- Number of applicants grouped by City
- Number of applicants grouped by state
- Number of applicants grouped by country
- Number of applicants grouped by total years of work experience
- Number of applicants grouped by managers or not
- Number of applicants grouped by their work history
- Number of applicants grouped by Degree type
- Number of applicants grouped by the year they graduated in
- Number of applicants grouped by same months in which they applied for jobs.
- Number of applicants grouped by currently employed or not
- Number of applicants grouped by number of people they have managed
- Number of applicants grouped by the top 30 jobs(by Id and title) they applied to.

Summary using SQL fucntions: Then we used functions from sqldf package to identify categories of people in cominations as mentioned below:

- Applicants who are currently emplyed grouped by city.
- Applicants who are currently emplyed , have managed others and grouped by city, state and coutnry.
- Applicants who are currently not emplyed , have managed others and grouped by city, state and coutnry.
- Applicants who are currently not emplyed grouped by city.
- Applicants who are currently not emplyed , have not managed others and grouped by city, state and coutnry.
- Applicants who are neither currently emplyed nor have managed others and grouped by city, state and coutnry.

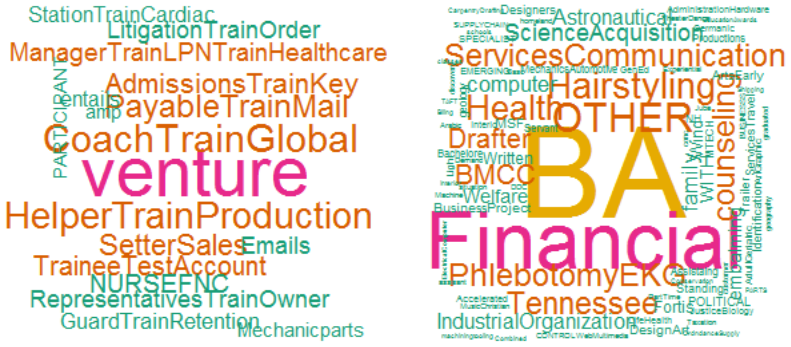
Correlation Matrix: we created dummy variables for columns which said yes/ no or had minimum categories like whether the applicant is employed or not, has managed people before or not, the type of their graduation degree, to be able to find the correlation between them and

other important variables like years of experience, work history count and graduation year. To see if any of these effect their currently employed status.



Since our data is not influenced by any imbalance in the observations it can be seen that after normalization also the strengths of correlation between the variables has not improoved as compared to before.

Word Cloud : we creatred the word cloud to get a priliminary estimate of which words are most dominantly used in the job titles which have been applied by our users and the majors done by them.



JOBS DATA

We again created summary variables which would give a tabular summary of details like:

- Number of days the application for job applications are open for.
- Number of jobs grouped by their starting and ending month for application.

Then we used SQL functions to identify jobs in cominations as mentioned below:

- Jobs which grouped by state and country
- Jobs which are grouped by cities in USA

We creatred the word cloud to get a priliminary estimate of which words are most dominantly used in the job titles, Descrition and requirements as specified in the job applications.

Since we did not perform any text cleaning operations on this data, it is observed that our word cloud is reflecting terms which do not make much sence or are stop words like “of”. These were processed later where after text mining out puts have made more sence, as shown below:

OBSERVATIONS:

1. Most of the jobs posted started in either May or June of 2012 and had an application time of 29-30 days.
2. maximum number of graduates are from bachelors degree and these are the maximum number of people who are currently employed and have/have not managed others, on the other hand the highest un employed people are the ones who have completed their high school only and these people have no experience of being managers.
3. USA obviously is the country which is offering maximum number of jobs with Florida, California and Texas being the states - As expected the job applicats are also highest in number from the same states aswell.
4. When it comes to citites, the maximum number of applicants are from Chicago, while the maximum jobs are posted in a Fort Lauderdale.
5. Most of the applicants are the ones who have graduated in the year 2012 or have a work experience of 7 years or have changed nearly 4 employers in the past.
6. The ratio of people who are currently employed vs not is about 60:40 which is much higher as compared to those who have been managers vs not which is 25:75 .
7. The maximum number of people who are currently employed (or not employed) and/ or who have or have managed (or not managed) others are from Charlotte by City and Florida by State.
8. Maximum number of applications are for jobs which are in lines of being “customer service representatives”

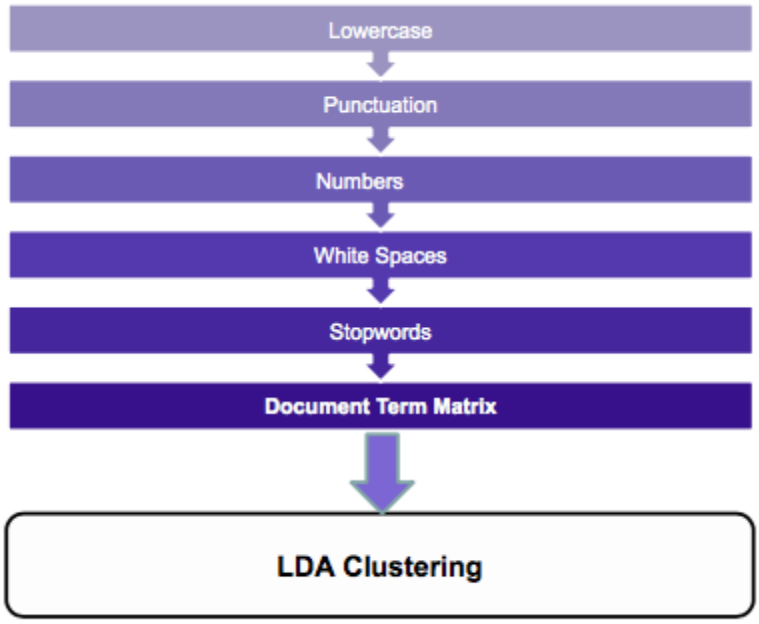
4. TEXT MINING

TEXT PREPROCESSING

Preprocessing tries to extract meaningful information from natural language text. It may be characterized as the process of analyzing text to extract information that is useful for a specific purpose. Preprocessing involves a series of activities to be performed in order to efficiently mine the information. These activities are: (Also refer diagram1)

- Converting all the documents in one case to avoid any ambiguity.
- Removing all the punctuations numbers and whitespaces
- Stop Words Elimination - Stop words are a division of natural language.

The motive that stop-words should be removed from a text is that they make the text look heavier and less important for analysis. Removing stop words reduces the dimensionality of term space. The most common words in text documents are articles, prepositions, and pronouns, etc. that does not give the meaning of the documents. These words are treated as stop words. Example for stop words: the, in, a, an, with, etc. Stop words are removed from documents because those words are not measured as keywords in text mining applications.



DOCUMENT TERM MATRIX

The corpus is encoded as a document-term matrix X with n rows and p columns, where p represents the number of words in the lexicon (full or pre-processed lexicon), and n is the number of documents. Encoding the corpus as a matrix allows one to utilize the power of linear algebra to analyze the document collection.

JOB OFFERS CLUSTERING

Job clustering was required to approach the assignment task in terms of classification. Each candidate will be classified for one or more classes of jobs according to the model learnt from the matching information. We have used Topic Models using Latent Dirichlet Allocation (LDA). LDA is a probabilistic model for topics in a corpus. It assumes a very specific graphical model structure on a corpus of documents, and infers the latent variables of the model. The latent variables (roughly speaking) specify what topics each document is associated with, and the word-distribution of each topic. These topics are inferred without supervision, so they do not have `names' (e.g., sports, politics), but rather are defined by the distribution they induce over the vocabulary words. LDA can be very useful as a preprocessing step in such a pipeline to extract information or features from your corpus. It will be especially useful if one has reason to believe that the corpus one is modeling matches the model assumptions, i.e. documents come from a larger collection, and each one was generated with a subset of the topics of the corpus in mind, and even more so if you feel that these topics may be related to the task you are interested in. Thus, by looking at the word-distributions, a person can deduce what the topic is related to. In our case we will look how jobs are clustered, and what topics they are related to. We also used Gibbs Sampling while implementation of LDA. In Topic Models, each document is represented as the probabilities of belonging to a number of topics. That is, each job offer is assigned a probability for each of a number of possible topics. The number of topics is set a priori. To be able to define clusters of job offers, a threshold has been experimentally set. All jobs that exceed this threshold are considered to belong to that Topic cluster.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
"business"	"relationships"	"sales"	"care"	"sales"	"clients"
"development"	"sales"	"customers"	"nursing"	"retail"	"office"
"accounting"	"representative"	"account"	"patients"	"client"	"insurance"
"management"	"professional"	"services"	"rnÃ"	"products"	"time"
"support"	"technology"	"products"	"staff"	"store"	"nbspspanprp"
"design"	"marketsource"	"data"	"health"	"services"	"restaurant"
"ensure"	"responsibilities"	"level"	"include"	"serviceslir"	"aligncenter"
"information"	"customers"	"existing"	"medical"	"liproviding"	"apply"
"related"	"specific"	"within"	"process"	"lirulrbr"	"financial"
"technical"	"environment"	"assigned"	"registered"	"tor"	"career"

After all job offers have been assigned to a cluster in the previous step, each candidate (from training data) is assigned to a job cluster along the matching information. This is well described in diagram below.

```
> head(app_Clust_table, n=15)
       ldaOut.topics
UserID  1 2 3 4 5 6
  6104  0 0 0 0 0 1
  10992 0 0 0 0 1 0
  24965 0 0 0 0 0 1
  25781 0 0 0 0 0 1
  32896 0 0 0 0 0 1
  38665 0 0 0 1 0 0
  44187 0 0 0 0 0 1
  44355 0 0 0 0 0 1
  47191 0 0 0 0 0 1
  52879 0 0 0 0 0 1
  53867 1 0 0 0 0 0
  56941 1 0 0 0 0 0
  59350 0 0 0 1 0 0
  73730 0 0 0 0 0 1
  80164 0 0 0 0 0 1
```

Since users may belong to one or more clusters, which makes sense, as one person may be interested to apply for jobs in two fields. This is the reason we have used LDA and not K-means in our analysis.

RESULTS

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster6	Total
Jobs	33	25	16	26	17	34	151
Users	46	25	11	5	5	152	244

After Removing 105 Stop words and converting 151 Documents we predicted 6 jobs topics. We observed that 46 users got allocated to Cluster 1, 25 to Cluster 2 and so on. An important observation here is that Cluster 1 has 33 jobs, Cluster 2 has 25 job, and so on. So we need the system to rank all the job offers in each of the clusters with the aim of highlighting the jobs that are more appropriate for a particular user.

5. COSINE SIMILARITY RANKING

The cosine similarity is used as the ranking method for deciding top 5 jobs for every user. Given a candidate, its cosine similarity is calculated with respect to each job using the attributes – Location (City, Zip code, Country) and Job Title.

Similarity between User City and Job Opening City

Similarity $\text{Matrix}_{\text{City}}$ - We obtained the matrix, M_{City}, measuring the Cosine Similarity measures between City of User and City of Job Opening. It is shown in the following figure.

$M_{City}[i,j] =$	Cosine Distance between City of UserID[j] and City of Job[j]

```
> dist_matrix_city
          [,1] [,2]      [,3] [,4]      [,5]      [,6]      [,7]      [,8] [,9]      [,10]
[1,] 1.0000000    1 1.0000000    0 1.0000000 1.0000000 1.0000000 1.0000000    1 1.0000000
[2,] 1.0000000    1 1.0000000    1 0.8933996 0.8933996 0.8093075 0.7303201    1 0.8443002
[3,] 1.0000000    1 0.7327388    1 1.0000000 1.0000000 1.0000000 1.0000000    1 1.0000000
[4,] 1.0000000    1 1.0000000    1 1.0000000 1.0000000 1.0000000 1.0000000    1 1.0000000
[5,] 0.8639172    1 1.0000000    1 1.0000000 1.0000000 0.7418011 0.8174258    1 0.8945907
          [,11] [,12] [,13]     [,14]     [,15] [,16] [,17]     [,18]     [,19]     [,20]
[1,] 1.0000000     1     1 1.0000000 1.0000000     1     1 1.0000000 0.8418861 0.8418861
[2,] 0.7538170     1     1 0.8443002 1.0000000     1     1 1.0000000 1.0000000 1.0000000
[3,] 1.0000000     1     1 1.0000000 1.0000000     1     1 1.0000000 1.0000000 1.0000000
[4,] 0.8333333     1     1 0.8945907 1.0000000     1     1 0.8174258 1.0000000 1.0000000
[5,] 0.8333333     1     1 0.8945907 0.8908911     1     1 0.8174258 1.0000000 1.0000000
     [,21] [,22]     [,23]     [,24] [,25] [,26] [,27]     [,28]
[1,]     1     1 0.8881966 0.5285955     1     1     1 1.0000000
[2,]     1     1 1.0000000 1.0000000     1     1     1 0.7989924
[3,]     1     1 1.0000000 1.0000000     1     1     1 1.0000000
[4,]     1     1 1.0000000 1.0000000     1     1     1 1.0000000
[5,]     1     1 1.0000000 1.0000000     1     1     1 0.8639172
```

Similarity between User Country and Job Opening Country

Similarity $\text{Matrix}_{\text{Country}}$ - Similarly, we obtained the matrix, M_{Country}, measuring the Cosine Similarity measures between Country of User and Country of Job Opening.

$M_{\text{Country}}[i,j] =$	Cosine Distance between Country of UserID[j] and City of Job[j]

Similarity between User Zip Code and Job Opening Zip Code

Similarity $\text{Matrix}_{\text{ZipCode}}$ - Similarly, we obtained the matrix, M_{ZipCode}, measuring the Cosine Similarity measures between ZipCode of User and ZipCode of Job Opening.

$M_{ZipCode}[i,j] =$	Cosine Distance between ZipCode of UserID[j] and Job Title of JobID[j]

Similarity between Job Titles of User Applications and Job Title of Job Openings

Similarity $\text{Matrix}_{\text{Job Title}}$ - Similarly, we obtained the matrix, $M_{\text{Job Title}}$, measuring the Cosine Similarity measures between all previous Job Titles of a User and Job Titles of Job Opening. To obtain the User Job Title we aggregated the Job Titles of Users's previous applications.

$M_{Job\ Title}[i,j] =$	Cosine Similarity between Job Title of UserID[j] and Job Title of JobID[j]

Final Similarity Measure

Finally, we added the Similarity Matrices to Obtain a Cumulative Distance Matrix for each Cluster :

Cumulative Distance Matrix , Mk=	$\lambda_{\text{Job Title}}(M_{k,\text{Job Title}})+\lambda_{\text{City}}(M_{k,\ \text{City}}) + \lambda_{\text{State}}(M_{k,\text{State}}) + \lambda_{\text{ZipCode}}(M_{k,\text{ZipCode}})$

Here k is the cluster ID. For simplicity, we assumed the weights, $\lambda_{\text{Job Title}}= \lambda_{\text{City}}=\lambda_{\text{State}}=\lambda_{\text{ZipCode}}= 1$. We finally obtained the Cumulative Similarity Matrix, Mkfor all the clusters.

JOB RANKING

We selected the largest 5 Distances for every UserID and their corresponding jobs were ranked as Top 5 Recommended Jobs for every User in a cluster.

6. CONCLUSION AND FUTURE WORK

We have presented a system for clustering the jobs based on their job description using LDA Clustering. Further, we predicted jobs for Users based on the cluster they belonged to and also developed a Job Ranking Algorithm for the Users using Cosine similarity measure based on Location and Job Title attributes of Users and Jobs. This way we ranked Top-5 Jobs for every User.

In future, we would like to extend our work as follows:

- Include the case when a candidate is assigned to more than one job cluster.
- Increase the size of our data set and incorporate more Natural Language Processing techniques like Entity Recognition.
- Use collaborative filtering, for User-User and Job-Job as a method for Job Recommendation.

7. CHALLENGES

As part of Data preprocessing and exploration our major challenge was to find all the cleaning that needed to be done in the data, given nearly 2,901,000 rows it was not an easy task to find all possible characters and dirty entries in the observations.

We, multiple times, came across scenarios where our visual representations showed labels which were unintelligible, or did not belong to the context there, that was when we had to go back clean that part of the data and get back to analyzing all of it after the changes were made.

This was a vicious cycle which took up quite some time and finally we could clean and condensed it to 2,568,000, which was then taken ahead for text mining.

The second major challenge was the LDA clustering. The topics that were generated in the clustering were not making sense initially. It is after lot of text mining process, that we got sensible clusters from the description column of the Jobs data. Also the Jobs data was humongous due to which we just went ahead with one jobs file(jobs6.tsv) and thus we took users corresponding to the window 6 only, for furthers assignment of users to clusters.

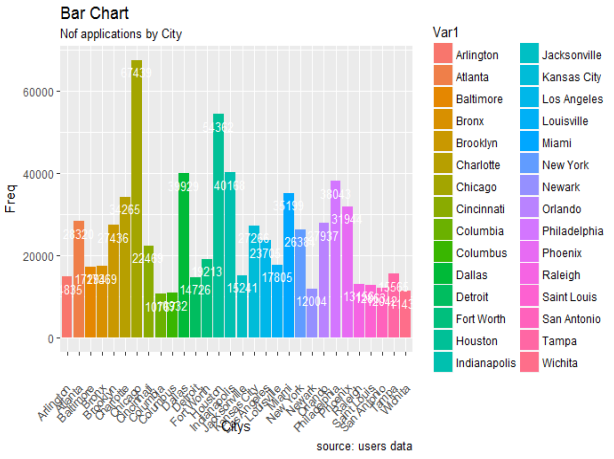
8. REFERENCES

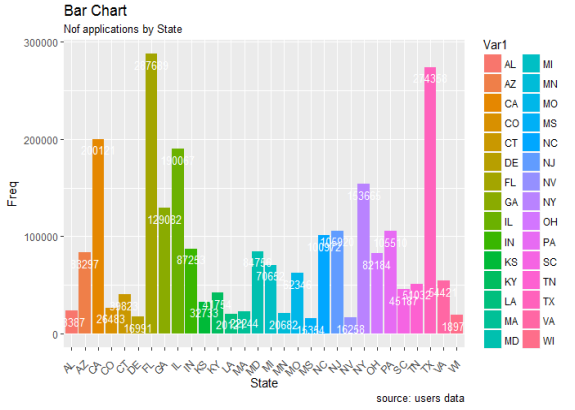
M.Poch, N. Bel, S. Espeja, F. Navio; "Ranking Job Offers for Candidates: learning hidden knowledge from Big Data"

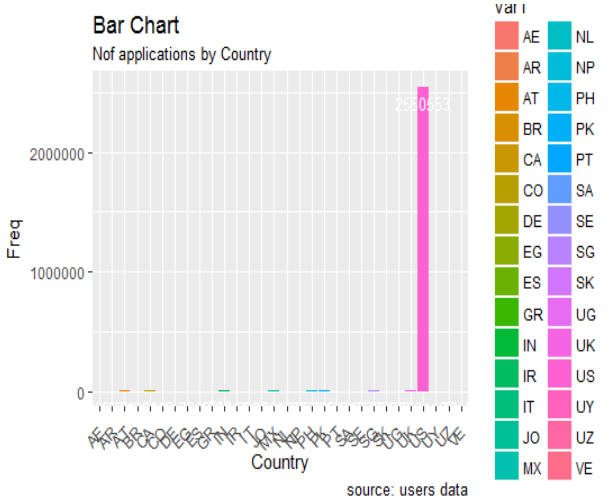
S. Mishra, M. Reddy; "A Bottom-Up Approach to Job Recommendation System"

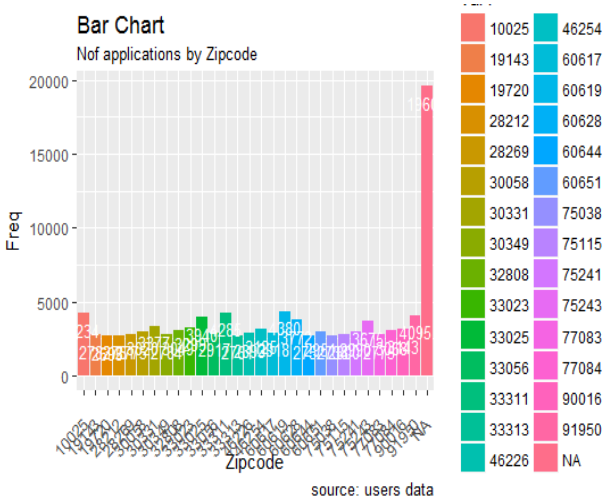
APPENDIX

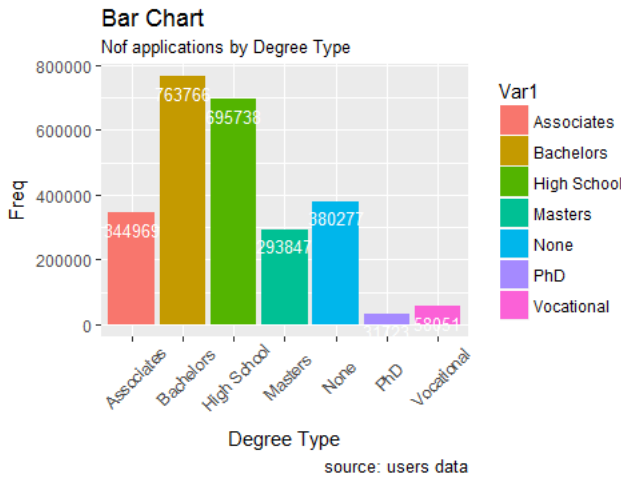
USERS DATA

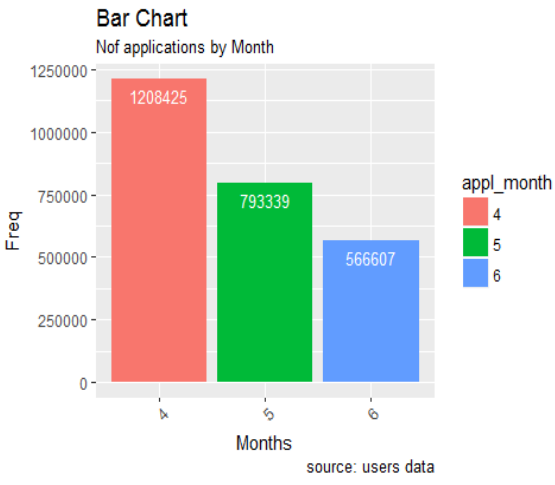


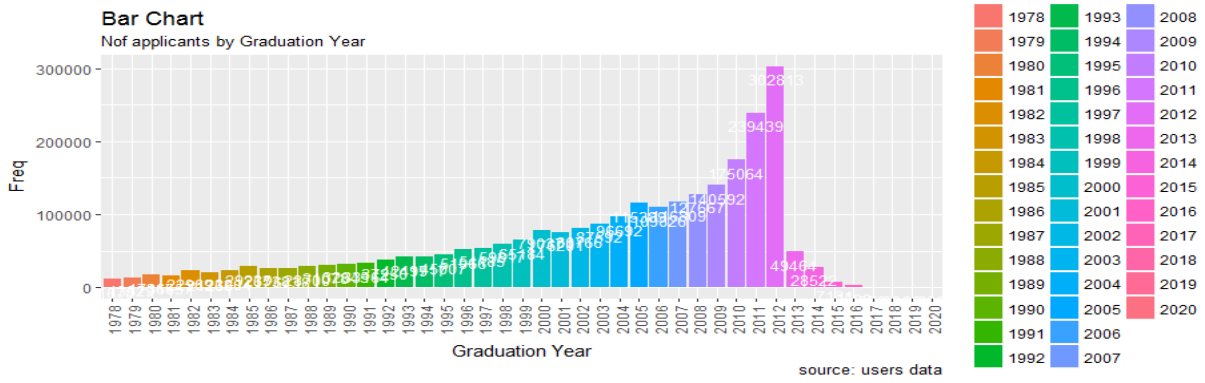


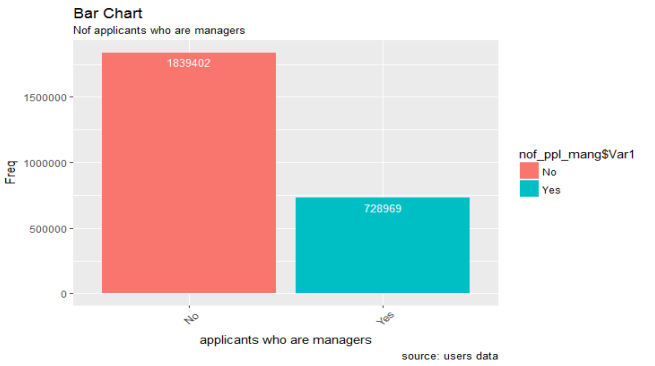


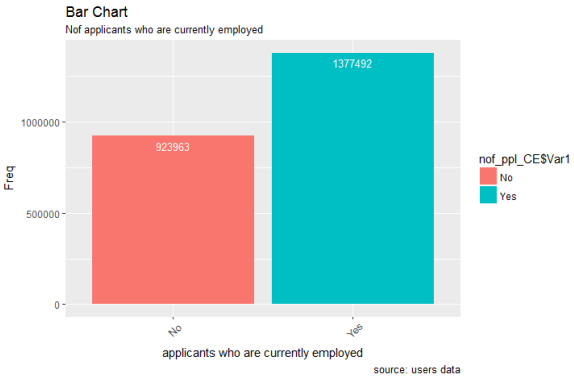


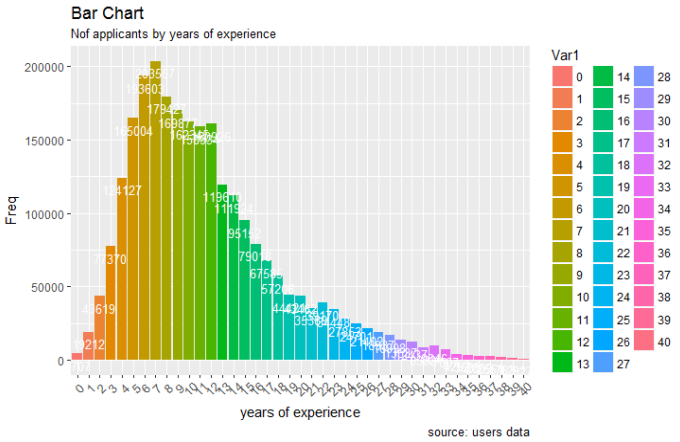


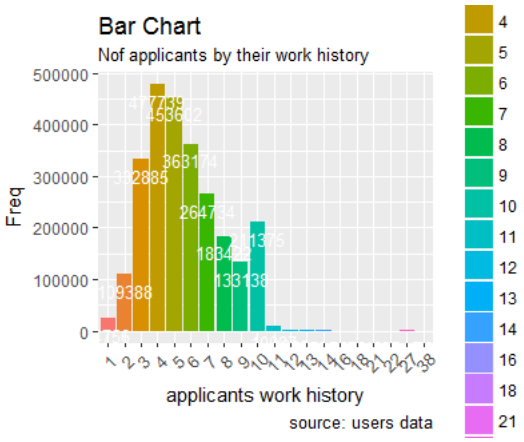


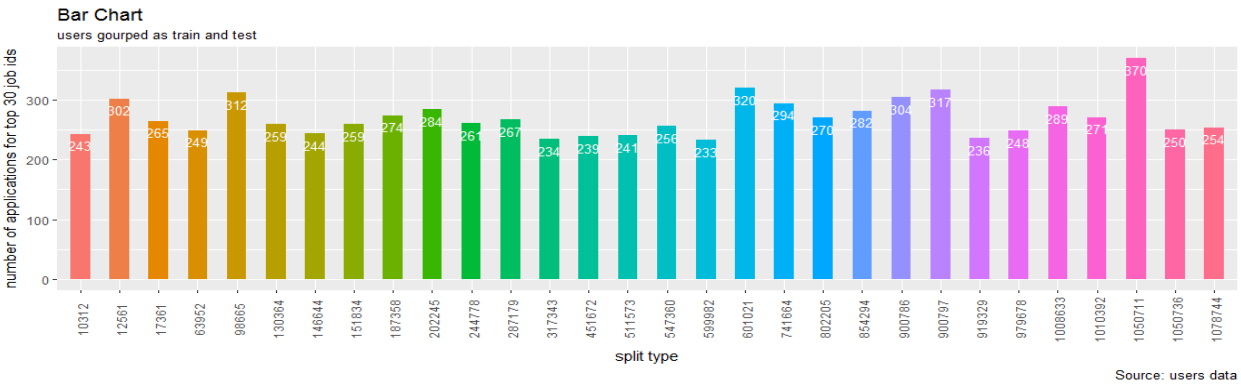

Bar Chart
Nof applicants by Graduation Year
Freq
Graduation Year
source: users data

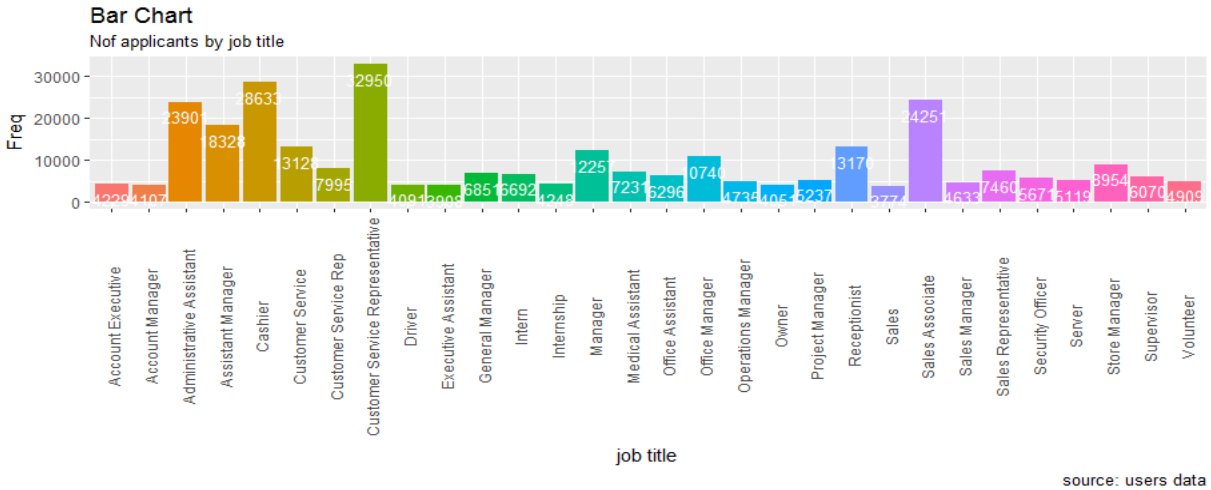

Bar Chart
Nof applicants who are managers
1839402
728969
Freq
applicants who are managers
source: users data
nof_ppl_mang$Var1
No
Yes

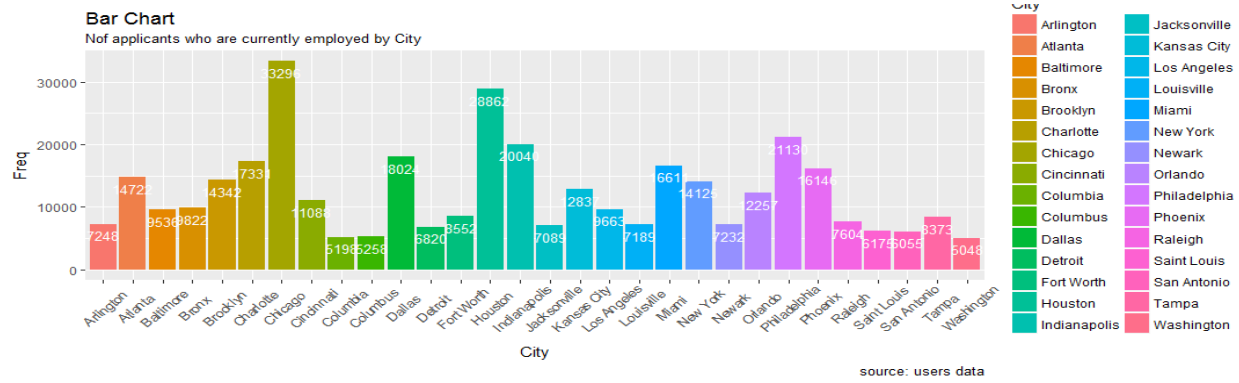

Bar Chart
Nof applicants who are currently employed
923963
1377492
Freq
applicants who are currently employed
source: users data
nof_ppl_CE$Var1
No
Yes

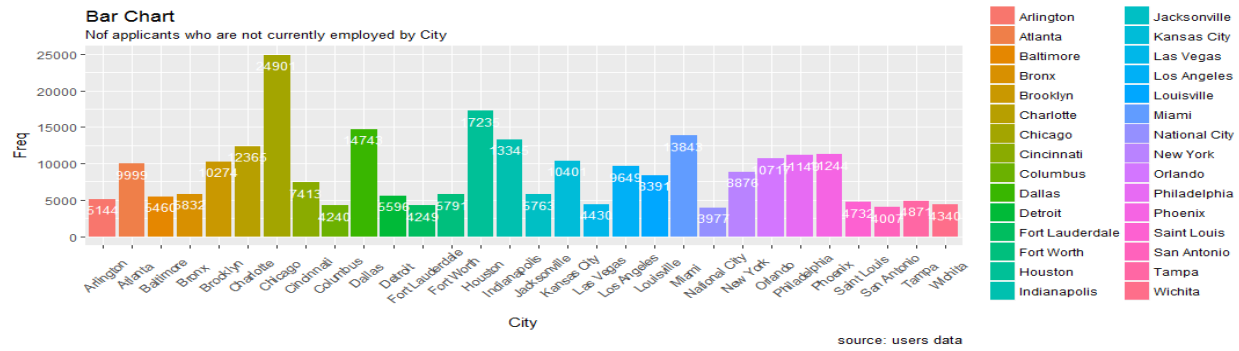

Bar Chart
Nof applicants by years of experience
Freq
years of experience
source: users data
Var1

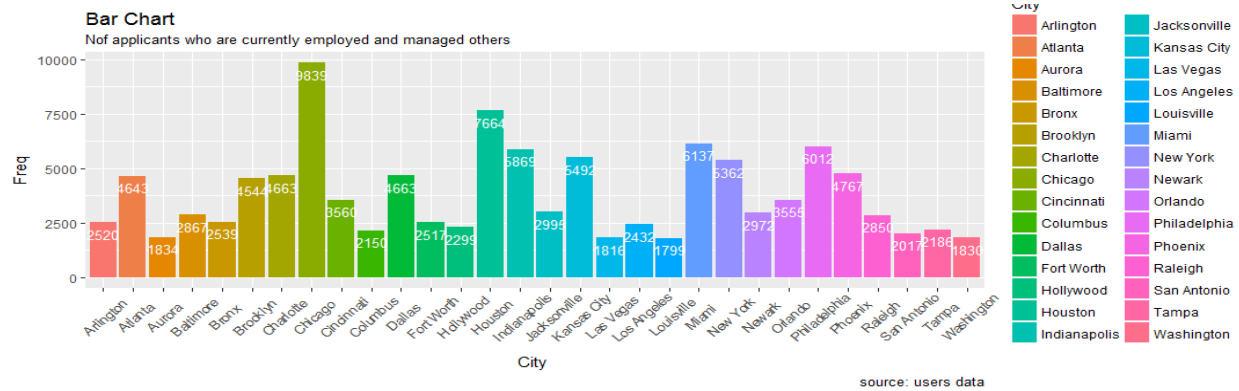

Bar Chart
Nof applicants by their work history
Freq
applicants work history
source: users data

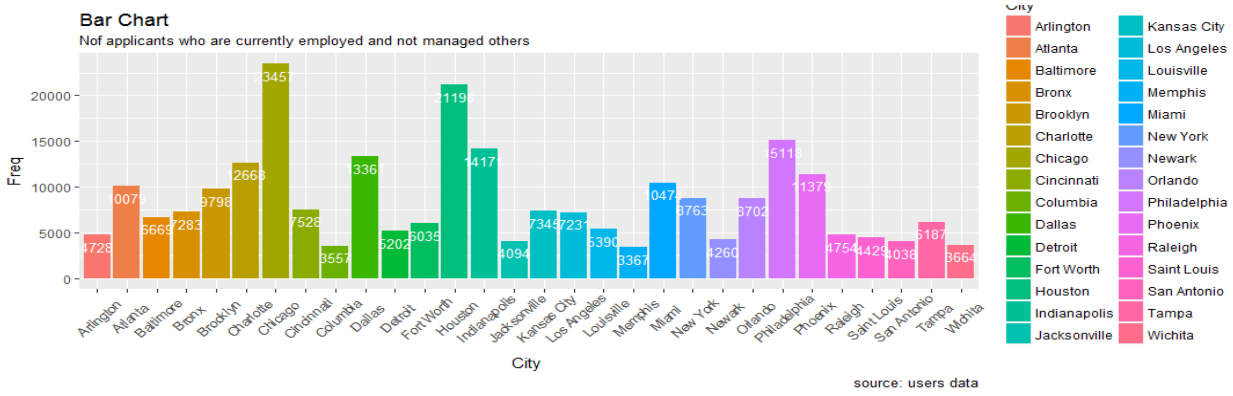

Bar Chart
users gourped as train and test
number of applications for top 30 job ids
split type
Source: users data

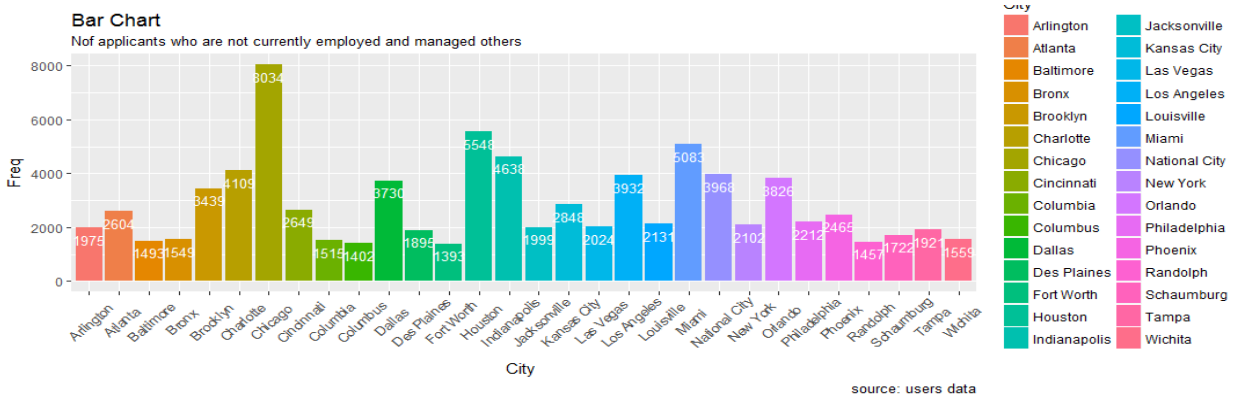

Bar Chart
Nof applicants by job title
Freq
job title
Account Executive
Account Manager
Administrative Assistant
Assistant Manager
Cashier
Customer Service
Customer Service Rep
Customer Service Representative
Driver
Executive Assistant
General Manager
Intern
Internship
Manager
Medical Assistant
Office Assistant
Office Manager
Operations Manager
Owner
Project Manager
Receptionist
Sales
Sales Associate
Sales Manager
Sales Representative
Security Officer
Server
Store Manager
Supervisor
Volunteer
source: users data

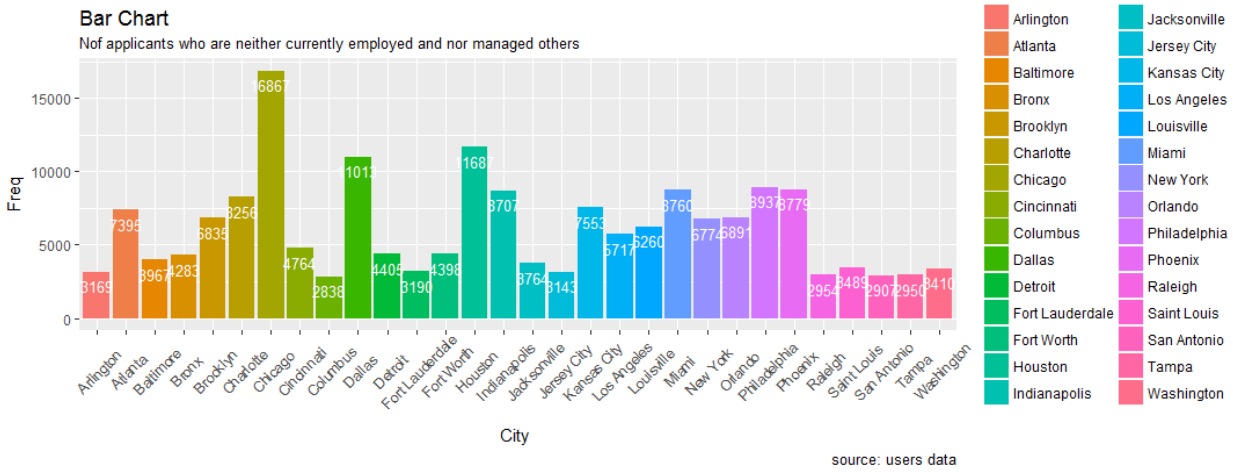

Bar Chart
Nof applicants who are currently employed by City
Freq
City
Arlington
Atlanta
Baltimore
Bronx
Brooklyn
Charlotte
Chicago
Cincinnati
Columbia
Columbus
Dallas
Detroit
Fort Worth
Houston
Indianapolis
Jacksonville
Kansas City
Los Angeles
Louisville
Miami
New York
Newark
Orlando
Philadelphia
Phoenix
Raleigh
Saint Louis
San Antonio
Tampa
Washington
source: users data

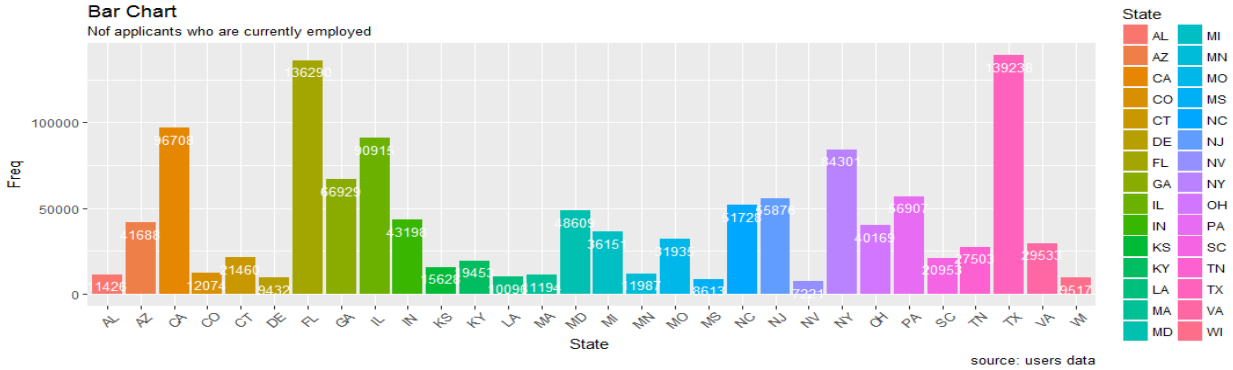

Bar Chart
Nof applicants who are not currently employed by City
Freq
City
Arlington
Atlanta
Baltimore
Bronx
Brooklyn
Charlotte
Chicago
Cincinnati
Columbus
Dallas
Detroit
Fort Lauderdale
Fort Worth
Houston
Indianapolis
Jacksonville
Kansas City
Las Vegas
Los Angeles
Louisville
Miami
National City
New York
Orlando
Philadelphia
Phoenix
Saint Louis
San Antonio
Tampa
Wichita
source: users data

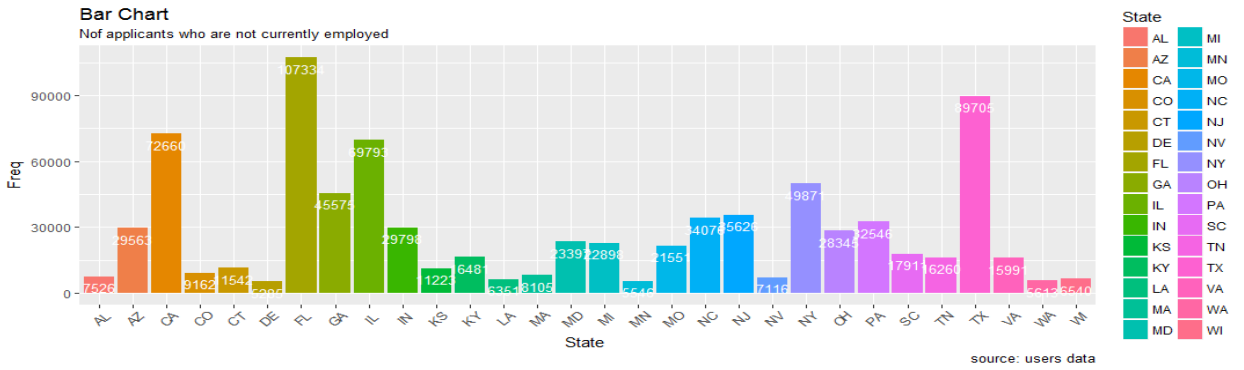

Bar Chart
Nof applicants who are currently employed and managed others
Freq
City
Arlington
Atlanta
Aurora
Baltimore
Bronx
Brooklyn
Charlotte
Chicago
Cincinnati
Columbus
Dallas
Fort Worth
Hollywood
Houston
Indianapolis
Jacksonville
Kansas City
Las Vegas
Los Angeles
Louisville
Miami
New York
Newark
Orlando
Philadelphia
Phoenix
Raleigh
San Antonio
Tampa
Washington
source: users data

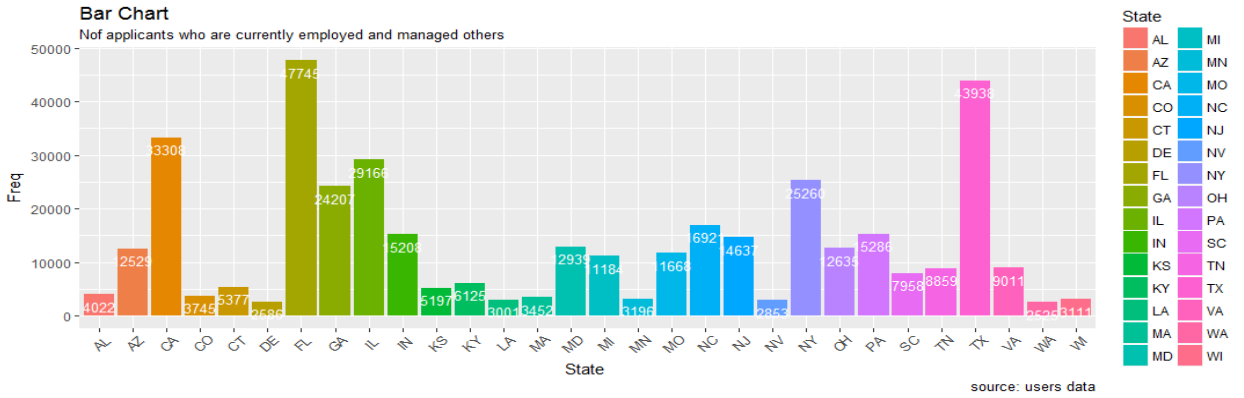

Bar Chart
Nof applicants who are currently employed and not managed others
Freq
City
source: users data
Arlington
Atlanta
Baltimore
Bronx
Brooklyn
Charlotte
Chicago
Cincinnati
Columbia
Dallas
Detroit
Fort Worth
Houston
Indianapolis
Jacksonville
Kansas City
Los Angeles
Louisville
Memphis
Miami
New York
Newark
Orlando
Philadelphia
Phoenix
Raleigh
Saint Louis
San Antonio
Tampa
Wichita

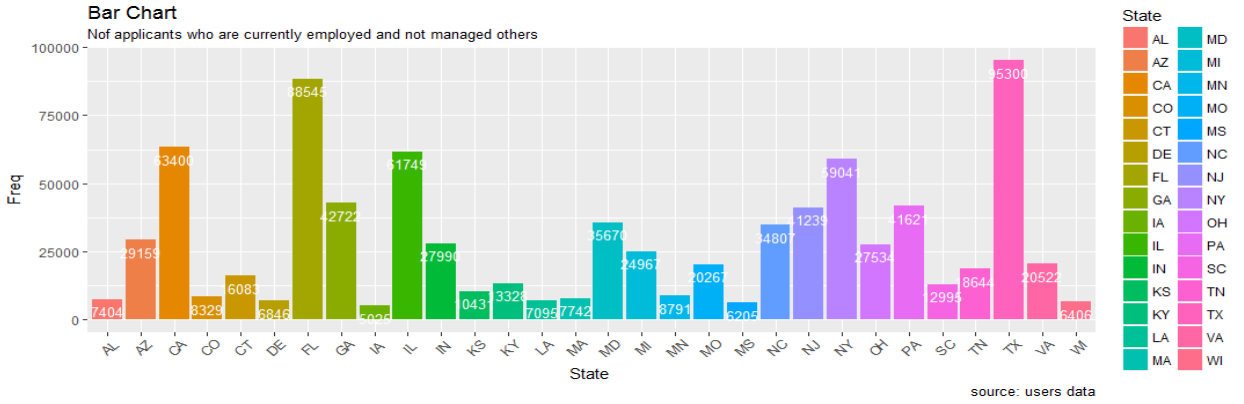

Bar Chart
Nof applicants who are not currently employed and managed others
Freq
City
source: users data
Arlington
Atlanta
Baltimore
Bronx
Brooklyn
Charlotte
Chicago
Cincinnati
Columbia
Columbus
Dallas
Des Plaines
Fort Worth
Houston
Indianapolis
Jacksonville
Kansas City
Las Vegas
Los Angeles
Louisville
Miami
National City
New York
Orlando
Philadelphia
Phoenix
Randolph
Schaumburg
Tampa
Wichita

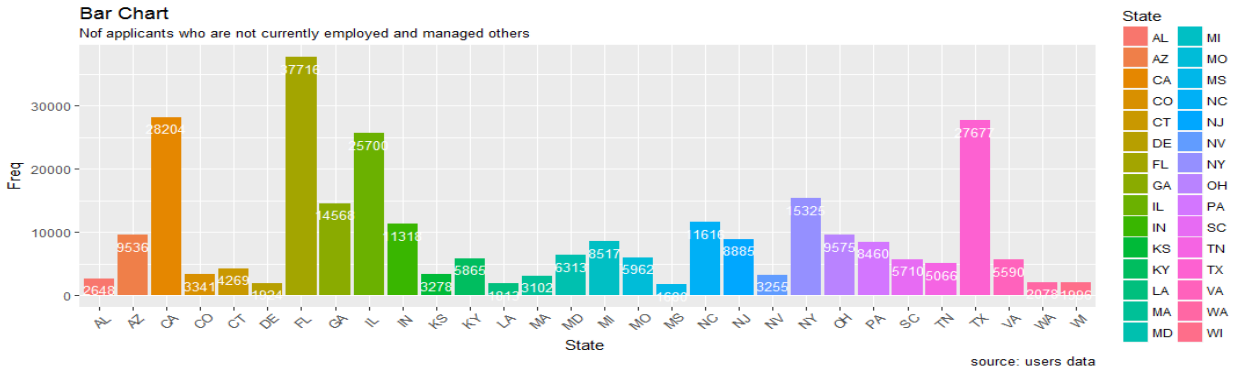

Bar Chart
Nof applicants who are neither currently employed and nor managed others
Freq
City
source: users data
Arlington
Atlanta
Baltimore
Bronx
Brooklyn
Charlotte
Chicago
Cincinnati
Columbus
Dallas
Detroit
Fort Lauderdale
Fort Worth
Houston
Indianapolis
Jacksonville
Jersey City
Kansas City
Los Angeles
Louisville
Miami
New York
Orlando
Philadelphia
Phoenix
Raleigh
Saint Louis
San Antonio
Tampa
Washington

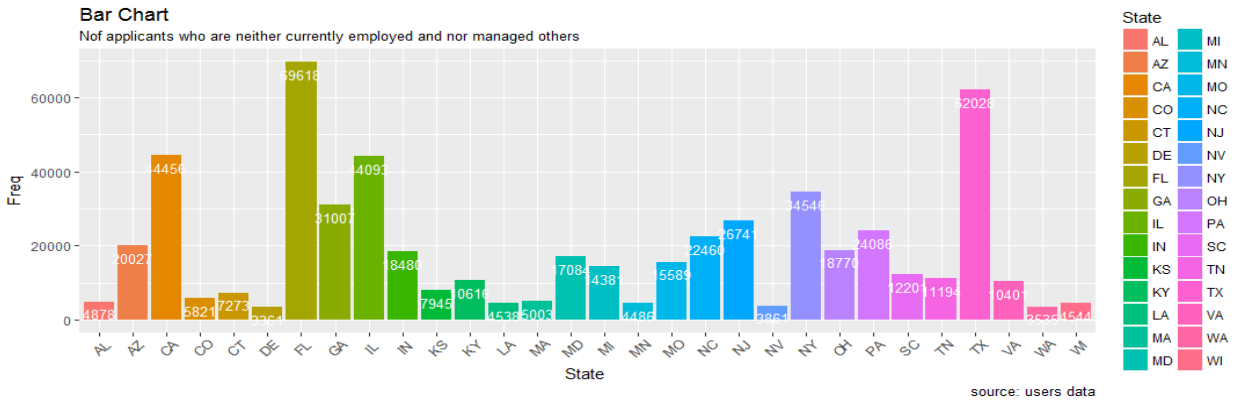

Bar Chart
Nof applicants who are currently employed
Freq
State
source: users data
AL
AZ
CA
CO
CT
DE
FL
GA
IL
IN
KS
KY
LA
MA
MD
MI
MN
MO
MS
NC
NJ
NV
NY
OH
PA
SC
TN
TX
VA
WI

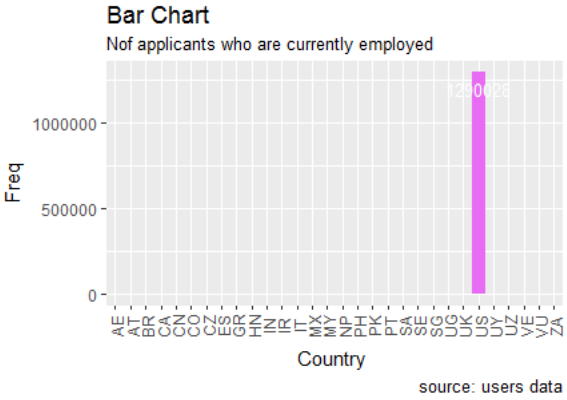

Bar Chart
Nof applicants who are not currently employed
Freq
State
source: users data
State
AL AZ CA CO CT DE FL GA IL IN KS KY LA MA MD MI MN MO NC NJ NV NY OH PA SC TN TX VA WA WI
MI MN MO NC NJ NV NY OH PA SC TN TX VA WA WI

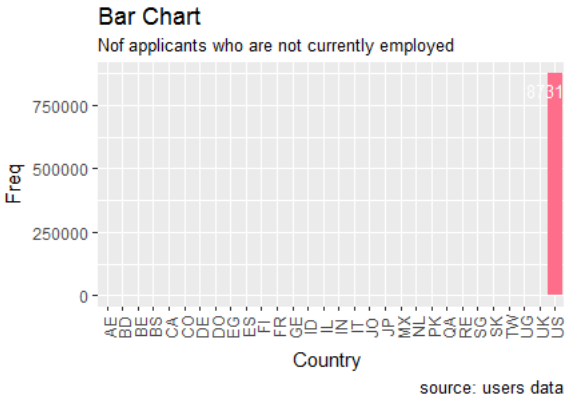

Bar Chart
Nof applicants who are currently employed and managed others
Freq
State
source: users data
State
AL AZ CA CO CT DE FL GA IL IN KS KY LA MA MD MI MN MO NC NJ NV NY OH PA SC TN TX VA WA WI
MI MN MO NC NJ NV NY OH PA SC TN TX VA WA WI

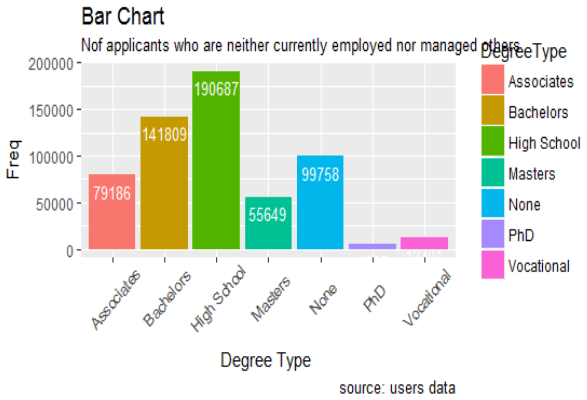

Bar Chart
Nof applicants who are currently employed and not managed others
Freq
State
source: users data
State
AL AZ CA CO CT DE FL GA IA IL IN KS KY LA MA MD MI MN MO MS NC NJ NV NY OH PA SC TN TX VA WI
MD MI MN MO NC NJ NY OH PA SC TN TX VA WI

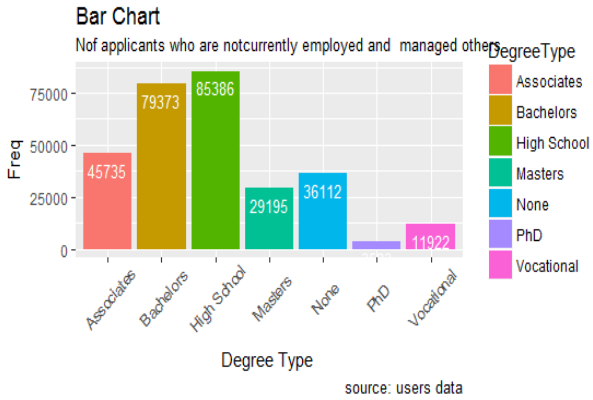

Bar Chart
Nof applicants who are not currently employed and managed others
Freq
State
source: users data
State
AL AZ CA CO CT DE FL GA IL IN KS KY LA MA MD MI MN MO MS NC NJ NV NY OH PA SC TN TX VA WA WI
MI MO MS NC NJ NV NY OH PA SC TN TX VA WA WI

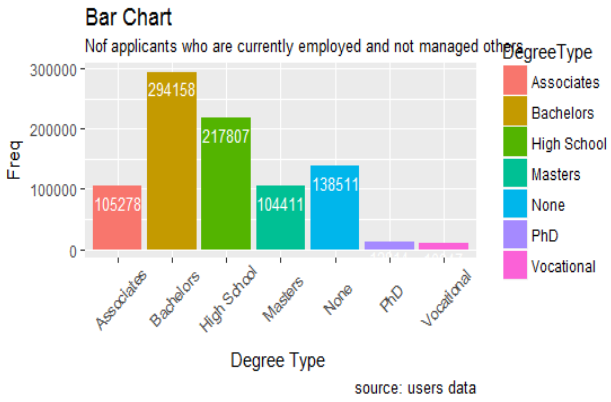

Bar Chart
Nof applicants who are neither currently employed and nor managed others
Freq
State
source: users data
State
AL AZ CA CO CT DE FL GA IL IN KS KY LA MA MD MI MN MO NC NJ NV NY OH PA SC TN TX VA WA WI
MI MN MO NC NJ NV NY OH PA SC TN TX VA WA WI

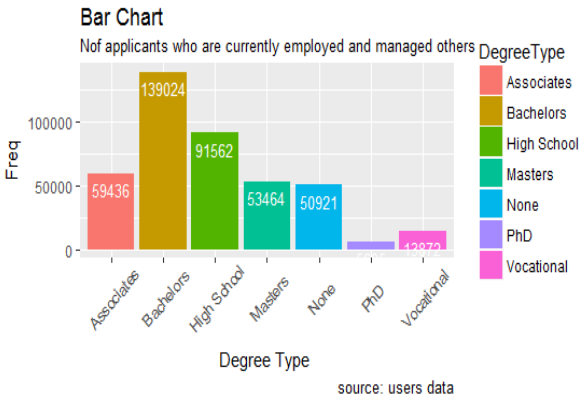

Bar Chart
Nof applicants who are currently employed
Freq
Country
source: users data


Bar Chart
Nof applicants who are not currently employed
Freq
Country
source: users data

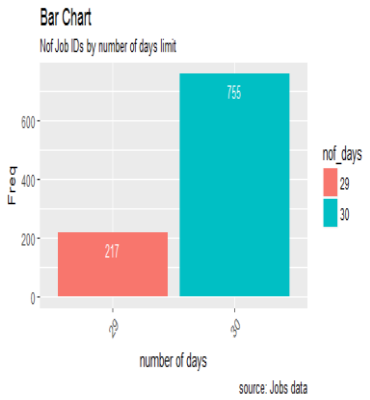

Bar Chart
Nof applicants who are neither currently employed nor managed others
Freq
Degree Type
source: users data
DegreeType
Associates
Bachelors
High School
Masters
None
PhD
Vocational

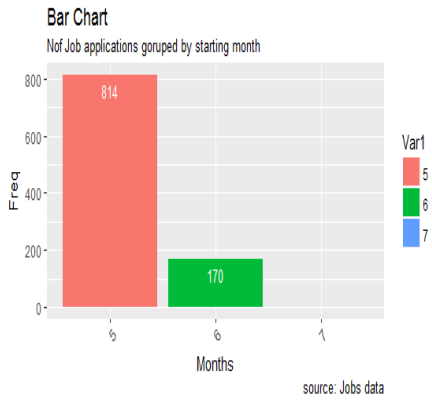

Bar Chart
Nof applicants who are notcurrently employed and managed others
Freq
Degree Type
source: users data
DegreeType
Associates
Bachelors
High School
Masters
None
PhD
Vocational

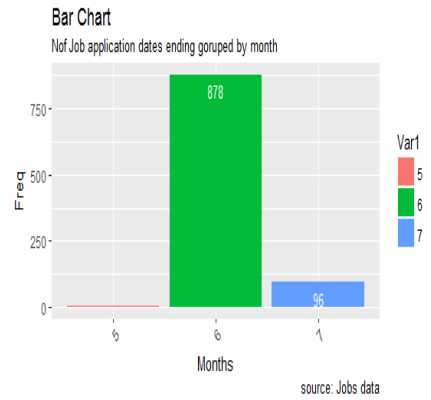

Bar Chart
Nof applicants who are currently employed and not managed others
Freq
Degree Type
source: users data
DegreeType
Associates
Bachelors
High School
Masters
None
PhD
Vocational

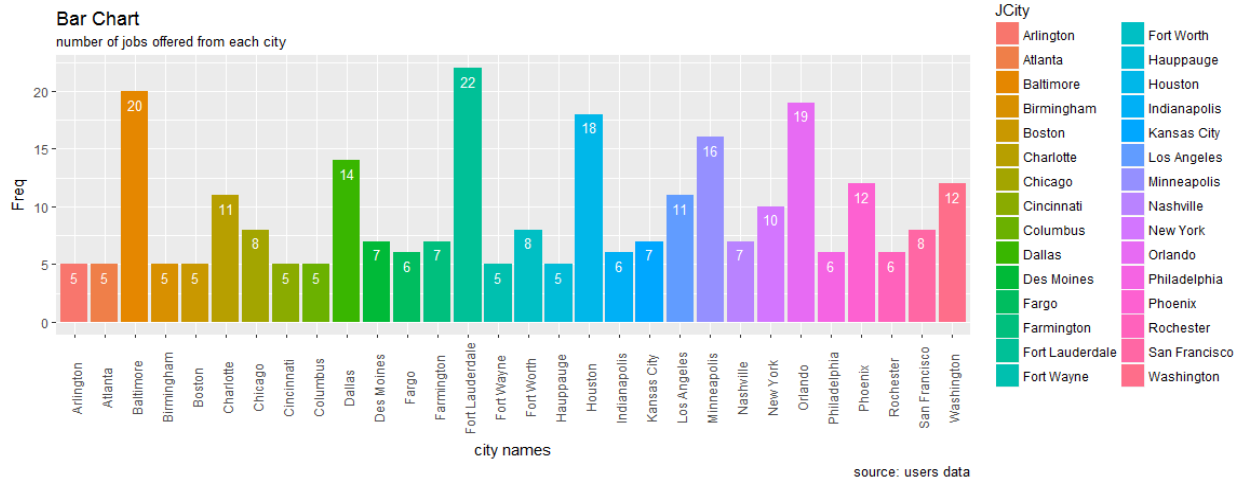

Bar Chart
Nof applicants who are currently employed and managed others
Freq
Degree Type
source: users data
DegreeType
Associates
Bachelors
High School
Masters
None
PhD
Vocational

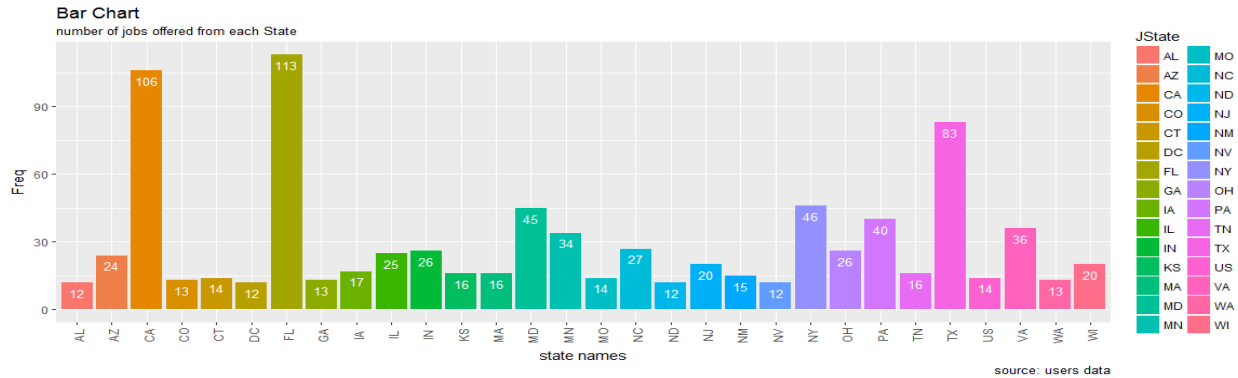
JOBS DATA:

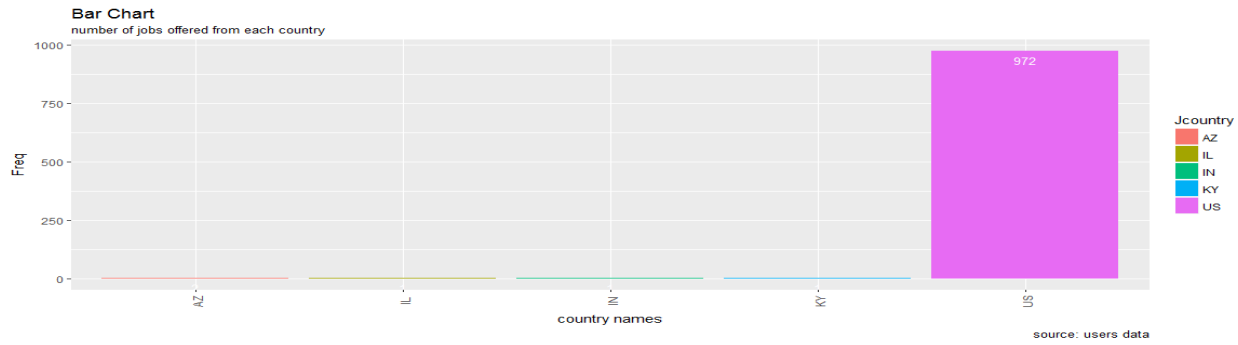

Bar Chart
Nof Job IDs by number of days limit
Freq
number of days
nof_days
29
30
755
217
source: Jobs data


Bar Chart
Nof Job applications goruped by starting month
Freq
Months
Var1
5
6
7
814
170
source: Jobs data


Bar Chart
Nof Job application dates ending goruped by month
Freq
Months
Var1
5
6
7
878
96
source: Jobs data


Bar Chart
number of jobs offered from each city
Freq
city names
JCity
Arlington
Atlanta
Baltimore
Birmingham
Boston
Charlotte
Chicago
Cincinnati
Columbus
Dallas
Des Moines
Fargo
Farmington
Fort Lauderdale
Fort Wayne
Fort Worth
Hauppauge
Houston
Indianapolis
Kansas City
Los Angeles
Minneapolis
Nashville
New York
Orlando
Philadelphia
Phoenix
Rochester
San Francisco
Washington
source: users data


Bar Chart
number of jobs offered from each State
Freq
state names
JState
source: users data


Bar Chart
number of jobs offered from each country
Freq
country names
Jcountry
AZ
IL
IN
KY
US
source: users data