An Analysis of PERM Labor Certification and Labor Condition Applications from the United States Department of Labor

Arunkumar Ranganathan
Brian Detweiler
Jacques Anthony

October 19, 2016

Abstract

Foreign born workers make up 17% of the United States workforce. In 2014, nearly one million foreign nationals became lawful permanent residents in the United States. Of those one million, 140,000 came through visas which are allocated to employment based residency. Where are these workers, and what do the demographics look like? How does each company's compensation measure up? Here, we use statistical analysis and business analytics to examine visa application data from the U.S. Department of Labor from 2008 to 2016. We intend to create an interactive data product that will make this publicly available information more accessible to the students who are entering the workforce, as well as to US citizens and permanent residents. This will empower them to competitively position themselves in the job market by making more informed decisions.

1 Introduction

The U.S. Department of Labor provides data for Labor Condition Applications and PERM Labor Certifications dating back to 2008. This data contains a wealth of job market information including prevailing wage, and the wages offered by particular companies to individuals with particular qualifications. In this paper we will analyze two types of visa workers in the US: H1-B and permanent visa workers.To protect foreign workers to get abused by their employers, businesses are required to pay the visa holder the higher of the prevailing wage in the company for the position or the prevailing wage for the occupation in the area of employment.The goal of this paper is to find disparities in wages for domestic and foreign workers if any exist. The H1-B visa program allows businesses to seek help from professionnals with at least a bachelor's degree in areas such as science, technology, engineering, math,and computer related occupations to name a few.From 2000 to 2005 over 50% of the applicants were from China and India. Each year there's a cap of 65,000 applications for the H-1B visa program and additional 20,000 applications for Applicants with U.S Masters degree. Applicants whose petitons for permanent residency did not get granted must leave the US for a year but are eligible to reapply later on. The maximum length for the H1-B visa is six years. Permanent residency in the US may be granted based on different factors. First, an individual may be granted residency through the H1-B program where the employer petitions for permanent residency on his behalf. Second, people with remarkable ability in science, education, arts, business, and education such as professors, researchers, or business executives and managers are eligible to apply for permanent residency without a labor crefication. Lastly,business investors who invest over half a million dollars in the US, employees of US foreign service posts, retired employees from international organizations, and other class of aliens may also petition for permanent residency with no labor certification. The cap for the number of applications available each year is approximately 140,000.

1.1 Document reproducibility

The entirety of this project is reproducible using R (version 3.1 and above) with Knitr package for R. Anyone with R version 3.1 with Knitr can unweave the document and reproduce all the text code and embedded R charts.

2 About the data

The Office of Foreign Labor Certification, under the Department of Labor provides data for PERM Labor Certification (LC) applications and Labor Condition Applications (LCA) via XLSX files. Data is available from 2008 onward. The iCERT system was implemented in 2009, so there are two files for 2009. Each file is structured similar to the others, but there are differences which must be addressed.

2.1 H-1B Data preparation

The H-1B data is about 75% larger than the PERM data, and spreadsheet programs do not handle these well, so the first task was to export these to CSV formatted files so that they could be handled with better tools. When exporting, they were also given more uniform file names. Once in CSV, we needed to identify common columns across all spreadsheets. The difficulty here, is that the columns do not have the same names across spreadsheets, even though they may be holding the same data.

Using the UNIX tool `head -n 10 *.csv > headers.txt`, we took the first ten rows of each file and put them into a separate file. Each of the CSVs first ten rows were then copied and pasted into another spreadsheet, and we undertook a manual effort to match columns of the same identity. We also discarded some excess information that we deemed to be unnecessary for our purposes.

It is also important to note that there was not always a match for the columns we had selected. For instance, we found some interesting information regarding the attorney used by the employer to file the H-1B application. This was only introduced in the 2015 and 2016 datasets, however, so prior years would have no data for this.

After determining the standard columns, we wrote an import script in R that made use of the `data.table::fread` function. This allowed us to not only quickly read in the file, but also select only the columns of interest, and rename them to the standard naming convention upon read.

Once the data was read into individual data frames, additional cleaning rules were applied. In some years, wage data contained invalid numeric characters such as dollar signs or a range of wages in a single column. To get around this, dollar signs were removed before converting to numeric, and ranges were split into a *from* and a *to* column. Ultimately, all wages were transformed into ranges. If there was no range for the wage, then the wage itself was used as the range.

Another normalization task was the wage unit. Some wages were represented as yearly salary, some as hourly, and others as monthly, weekly, and bi-weekly. There can be subtle differences in each type of pay, but these were normalized according to a yearly salary. Hourly wage was multiplied by 40 hours a week and 52 weeks a year. Monthly wage was multiplied by 12, weekly by 52, and bi-weekly by 26. This allows all wages to be treated roughly on the same scale.

2.2 Perm Data preparation

Perm data is available from year 2008 to current. Year 2015 and 2016 have over 120 attributes while 2008-2014 have 27-30 attributes, we plan on using 27 attributes common from 2008-16, while additional 14 attributes only on 2015-16 is useful for limited analysis. We can find information such as employer details, employee details, job function, salary, university, job city, number of years of experience, country of citizenship, and industry.

Steps: Download PERMFY2008.xlsx,PERMFY2009.xlsx, PERMFY2010.xlsx,PERMFY2011.xlsx, PERMFY2012Q4.xlsx, PERMFY14Q4.xlsx, $\text{PERM}_{D}isclosureDataFY15Q4.xlsx, PERMDisclosureDataFY16.xlsxfromdepartmentoflaborwebs$

Perm data downloaded from department of foreign labor is in .xlsx form. R-Program that loads this data uses two libraries 1) xlsx and 2) dplyr. We have loaded this data in R - cleaned them, gathered only required columns and created .rds file. In this process of creating final file, we went through all the files and got only columns that are essential for our data analysis. Multiple .xlsx files are initially filtered with select 41 columns and merged them all back again into one large file with 622,637 records. In short, two-step process to get a clean perm data, first step is to download data from DOL, then, run code to parse the file create a final output - with no intermediate manual steps required. We will discuss what and how we cleaned the data next. In total there were 9 files downloaded, with each files ranging from 40,000 to 90,000 records. Most column names from 2008-2014 were same with minor differences, while 2015-16 columns were similar.

Column names are standardized to 2016 for posterity - hoping next years' data would be in the same format as 2016. To standardize all column, we have removed any space between columns to underscore and created all column names in upper case. First step in preparing data, we selected all the relevant columns and filled in not available columns as NA's (such as employer address 2 in 2008 year data). Subsetting these columns within a function is created in order to call the function to create final subset for each year. Once we have all the data available for each year, final final is created with raw data without standardizing or further cleaning. This dataset can be used by anyone looking for perm data and can standardize based on their need.

Data cleaning specific to this project is also required not just for standardizing data but also to have the data in a specific format.

During the process of creating final file, most columns are read as text and few columns such as 'decision$_date'$ are $read$ as $date, while$ $salary$ $information$ are $read$ as $numeric. Decision$ $date$ is $first$ $transformed$ to $YYYY-MM-DD$ $format$ for $consistency$ $within$ our $database.$

Transforming all character variables to upper case deemed as necessary to avoid duplicity or incorrectness when performing data manipulation.

Wage information is marked in unit as weekly, monthly, bi-weekly, and annually. This wage units weren't consistent in the file - so we have to make all the weekly to 'WK', monthly to' MTH' and bi-weekly to 'BIWK' and yearly to 'YR' across file

Wage information was normalized to create a new column just with annual salary - i.e. weekly, hourly, monthly, and bi-weekly salaries are transformed to yearly. Range of salary of salary is also normalized in this step.

Employer zip codes are trimmed to 5 digits to keep all the zip codes at the same level of depth

Employment state data is both at abbreviated and expanded level. We need to standardize either as a two digit code or expanded for consistency - for this exercise state names are created at expanded level for all 50 US states as well as US territories

Next we need to create a unique id to tie all the information back together as well as to create index in the future - row number is created as unique id

This cleaned data is saved as an .rds file before next step.

Another .rds file (PermEmpMapsdat.rds) - one is with summarized employer name address, city, state and zip codes order by number or perm applications processed and by their mean salary. This dataset would give us which employer sponsors most employees for permanent full time employment as well as who pays more. Another use of this dataset is to use it to look for geocodes.

In order to get geocodes for employer address we decided to go with a distinct employer addresses in the order of most common perm application employer who also pays the most. As there can only be 2,500 address latitude and longitude can be pulled from google maps every day, we decided to create a program that keeps running sleeps, wakes up every hour and hit the google server to look for the addresses. This program will collect all the addresses and store them in a temporary file and updated them back again based on their index in a main file. This program is designed to run at multiple locations for different index ranges to collect as much address as possible.

2.3 Shortcomings

As mentioned in the previous section, because the data is not homogeneous, there are bound to be disparities. Missing data - columns which are not found across all spreadsheets - is the biggest issue. We can make assumptions when there is sparse data, but it would not be prudent to make assumptions where there is no data. For this reason, we fully disclose the absence data where necessary.

All of the data has been entered by humans at some point, so there are likely many human-generated errors. Some of these can be seen as outliers. Particularly in the 2008 and pre-iCERT 2009 data, the wage unit is most certainly incorrect in some spots. For example, some wages are listed at $500 per hour, but the intended unit may have been per week. It is not possible to fix this programmatically though, because there are, in fact, some jobs that pay $500 per hour (CEOs, for instance). This data must be dealt with in one of two ways. They can either be corrected by hand inspection of outliers, or outliers can be removed completely. This results in a slight loss of fidelity. Extremely high paying jobs, such as CEO or physician may not be displayed.

Another issue is the switch from U.S. Citizenship and Immigration Services Dictionary of Occupational Titles (DOT) codes in 2008 and pre-iCERT 2009 data, to the North American Industry Classification System (NAICS) codes. The DOT codes are three digits and fairly high-level, where as the NAICS codes are hierarchical, with the first two being the industry, and the specification of the job title narrowing with up to six digits. For this reason, it is difficult to get consistent job titles across years.

Perm data is comprehensive for 2015 and 2016, however the data for each year have inconsistencies both in terms of data dictionary. We have to make some educated judgements about some categories and ignore many cases with NA values. Year 2008-2014 have 25 attributes with most of them do not give us substantial information other than salary and employer name. Nature of this data made it hard to graph plots based of these categorical variables. Here is a gist of categories within this data

53,011 - Job title based on Perm data 46,410 - Job Title based on Work 40,431 - Study Major 135,955 - Employers 213 - Citizen's from countries

3 Methods

The numeric data provided by the H-1B and PERM datasets are mostly in the form of wages, both the wage that the employer is offering and the prevailing wage.[1] Also of interest, are the number of workers an applicant is filing for, and the implicit number of applications faceted by status and year.

3.1 Data product

The data is provided as an interactive `Shiny` application, that allows the user to filter wages by various criteria.

The plots consist of distributions and and heat maps of wages across the United States.Our objective is to have data that can be sliced by users into different factors i.e. by state, city, employer, job major, function, salary. Data products we produce from this data will help our target audience make informed decision about kinds of employment held by high skilled immigrants, who are hiring, what kind of skills employers are looking. First we will look the number of applications processed every year since 2008. Looking data every year will help see if there is an increasing or decreasing trend in number of applications processed - although there are cap on number of employment based immigration every year, still an increasing trend or maximum use of employment based immigration will give us insights. Next, we can look for immigration based on geography - where hiring is growing and which geography lacks in hiring. Not just in terms of absolute numbers rather than of normalized hiring data would give us more information. We can look at time series data on hiring by job function, job majors and university of education. Box plots, histograms, bar charts and heat maps are some of those plots we will use to draw inferences.

4 Results

4.1 Overview

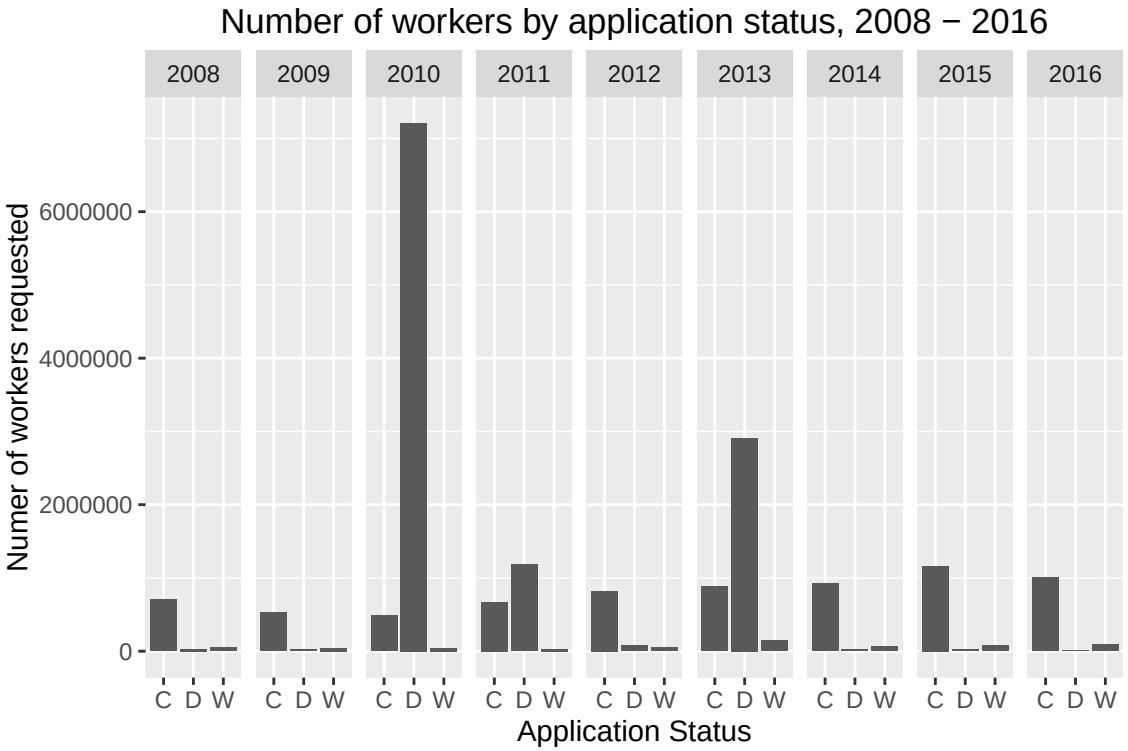
4.2 Prevailing Wage

It would be interesting to see how many workers are getting approved for H-1B visas over the years. We can see this by plotting the number of workers within each visa status category (Certified, Denied, and Withdrawn).

[1] Prevailing wage is defined as the hourly wage, usual benefits and overtime, paid to the majority of workers, laborers, and mechanics within a particular geographic area.

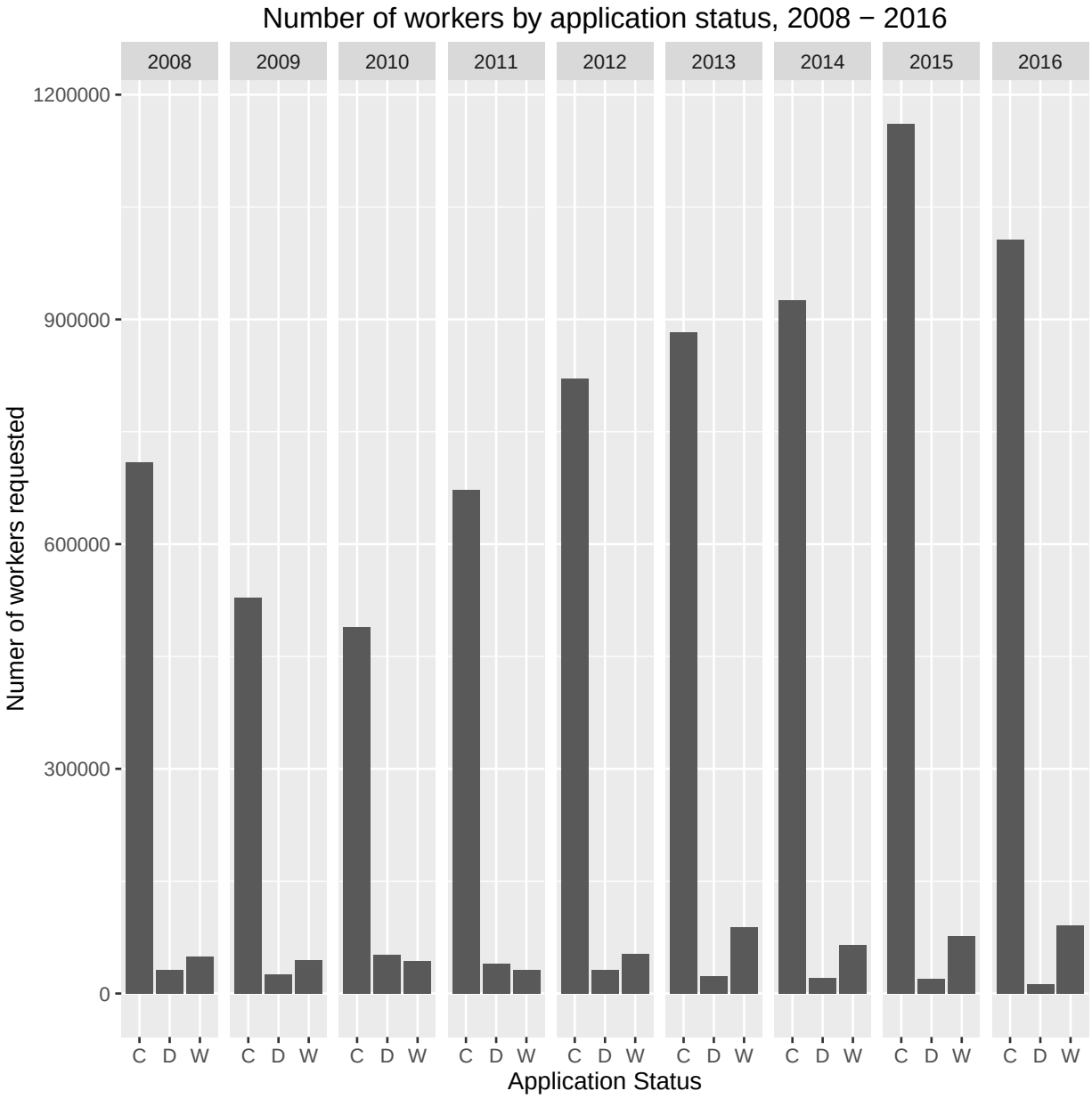
Table 1: Applications requesting over 1000 workers

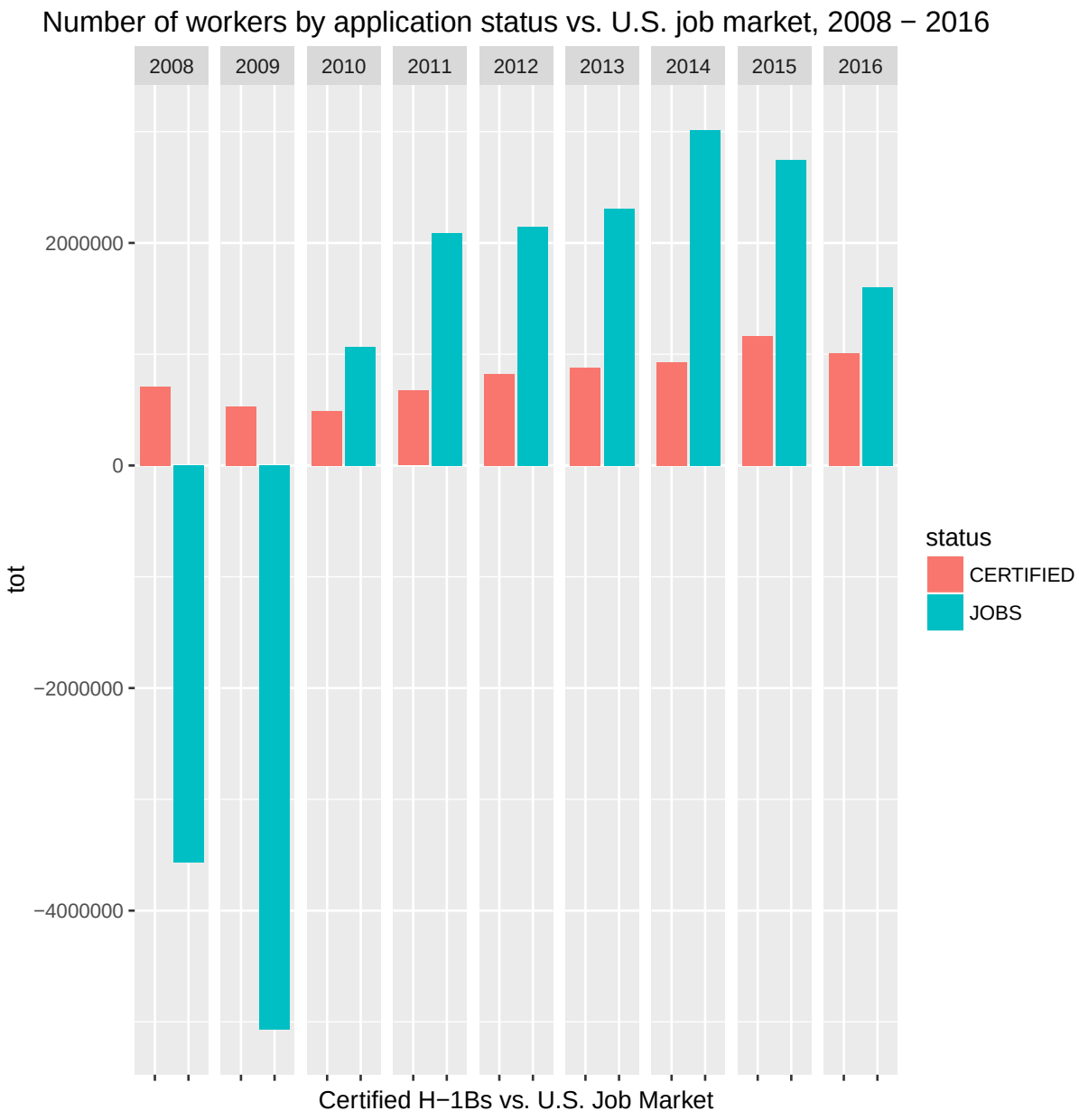
fy	total_workers	status
2011	2010	DENIED
2012	2011	DENIED
2013	2012	DENIED
2013	2012	DENIED
2014	2013	DENIED
2011	10000	DENIED
2012	43555	DENIED
2010	52000	DENIED
2010	53000	DENIED
2010	53000	DENIED
2013	58500	DENIED
2013	60000	WITHDRAWN
2013	793172	DENIED
2011	1132014	DENIED
2013	2031308	DENIED
2010	7000000	DENIED

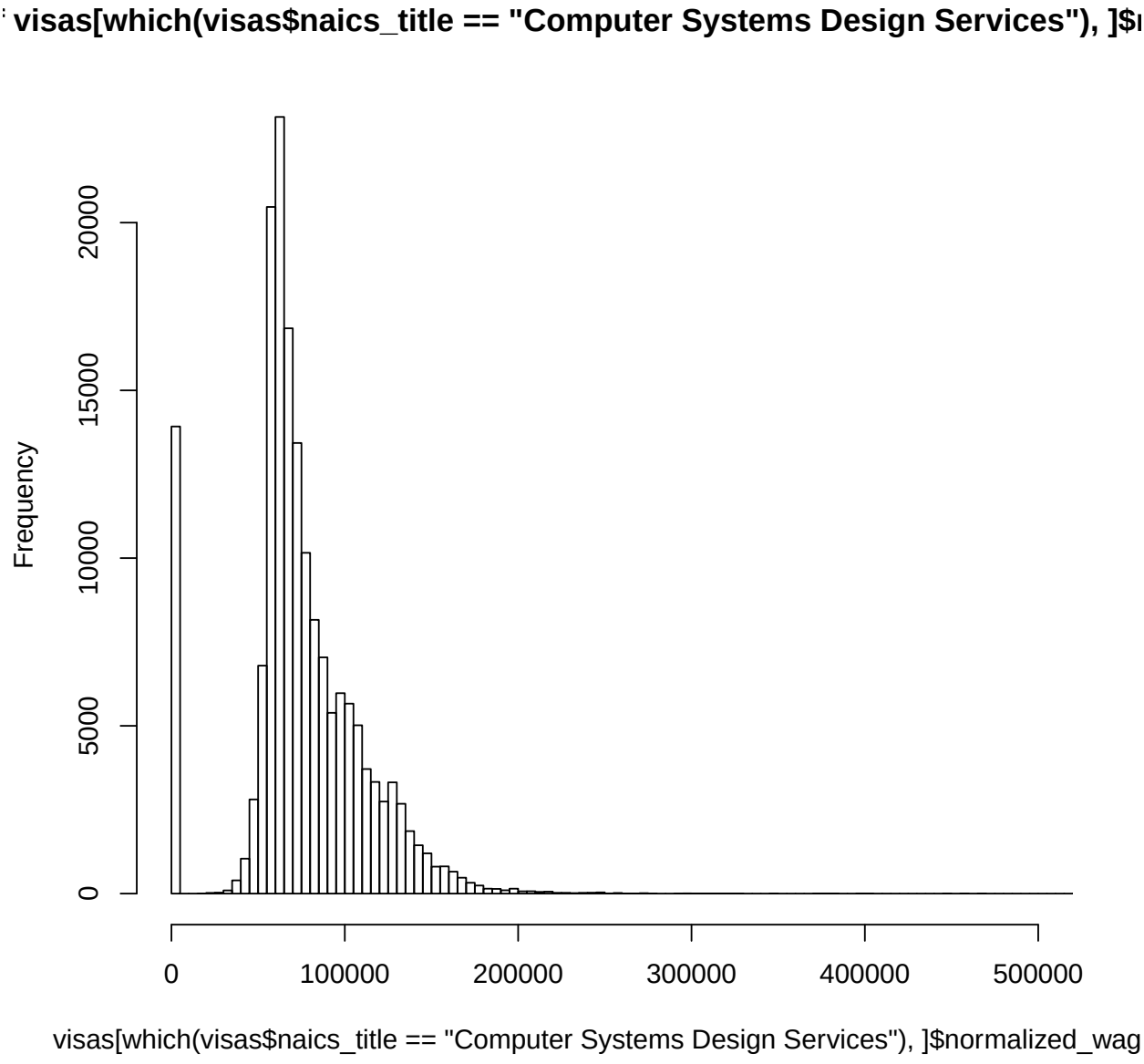


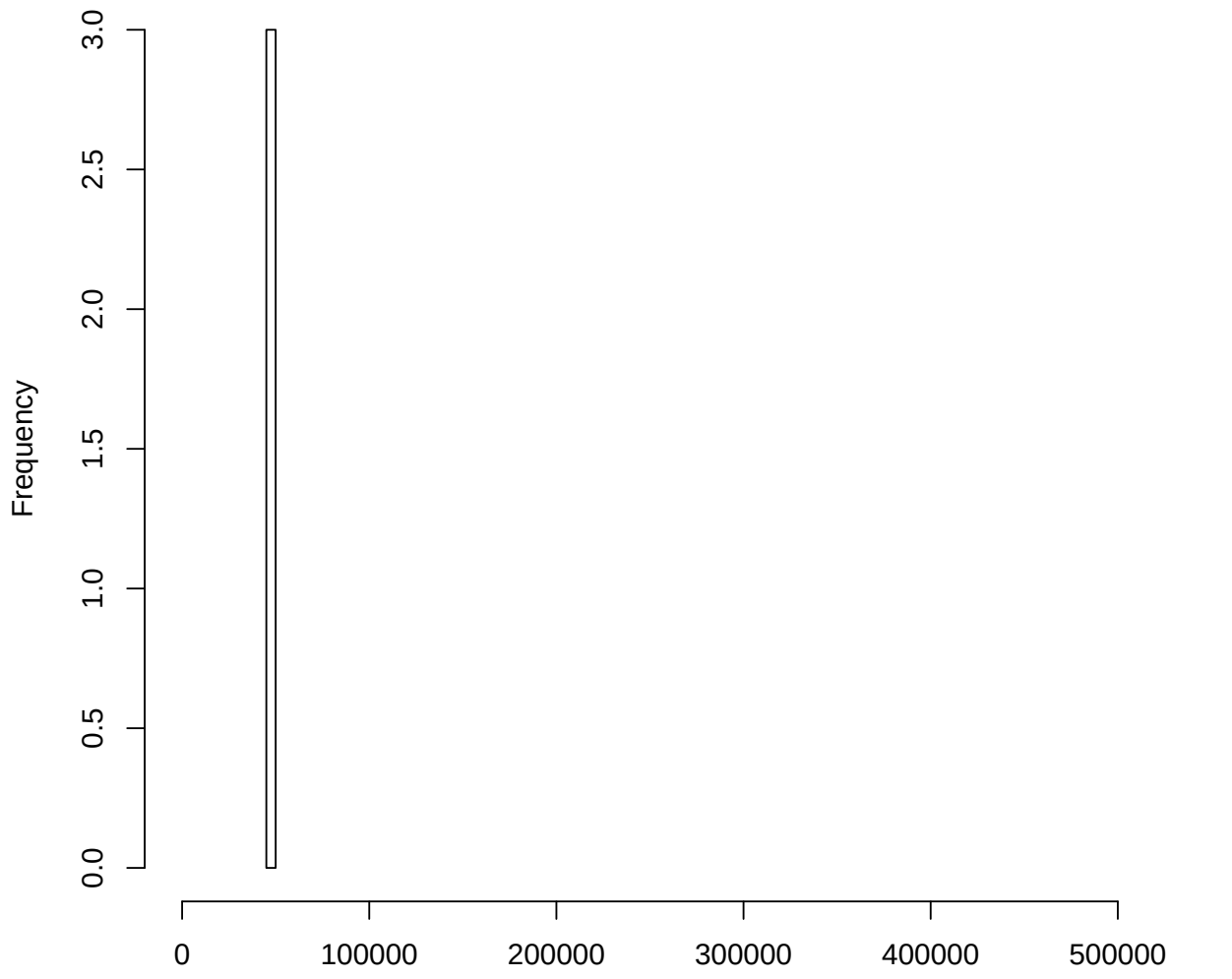
Clearly, we can see huge outliers in 2010 and 2013 that don't seem to fit the data. These could be explained as outliers. Upon investigation, it is safe to say that all H-1B applications requesting over 1,000 total workers are either denied or withdrawn.

Once we remove all requests for more than 1,000 workers, we can start to see a pattern. The number of denied and withdrawn applications remains fairly constant, but the number of certified workers shows a steady rise after 2010, most likely due to a strengthening economy and returning jobs.


Number of workers by application status, 2008 − 2016
2008
2009
2010
2011
2012
2013
2014
2015
2016
1200000
900000
600000
300000
0
Numer of workers requested
C D W
Application Status

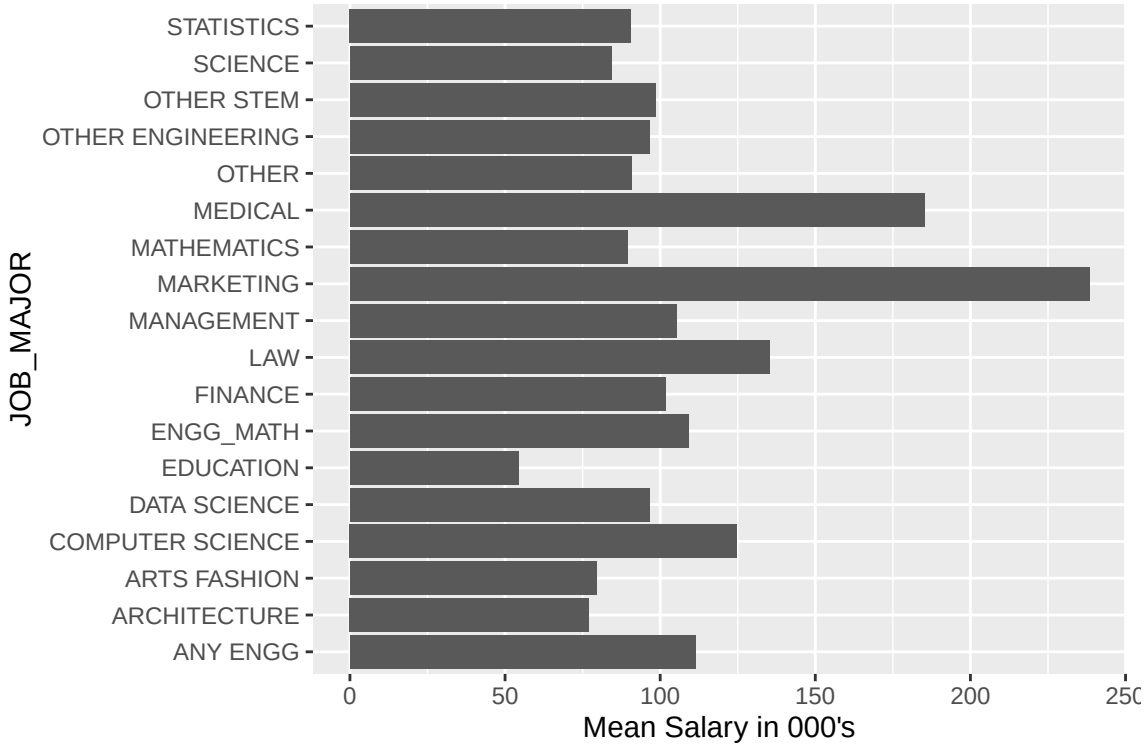

Number of workers by application status vs. U.S. job market, 2008 − 2016
2008
2009
2010
2011
2012
2013
2014
2015
2016
2000000
0
−2000000
−4000000
tot
status
CERTIFIED
JOBS
Certified H−1Bs vs. U.S. Job Market

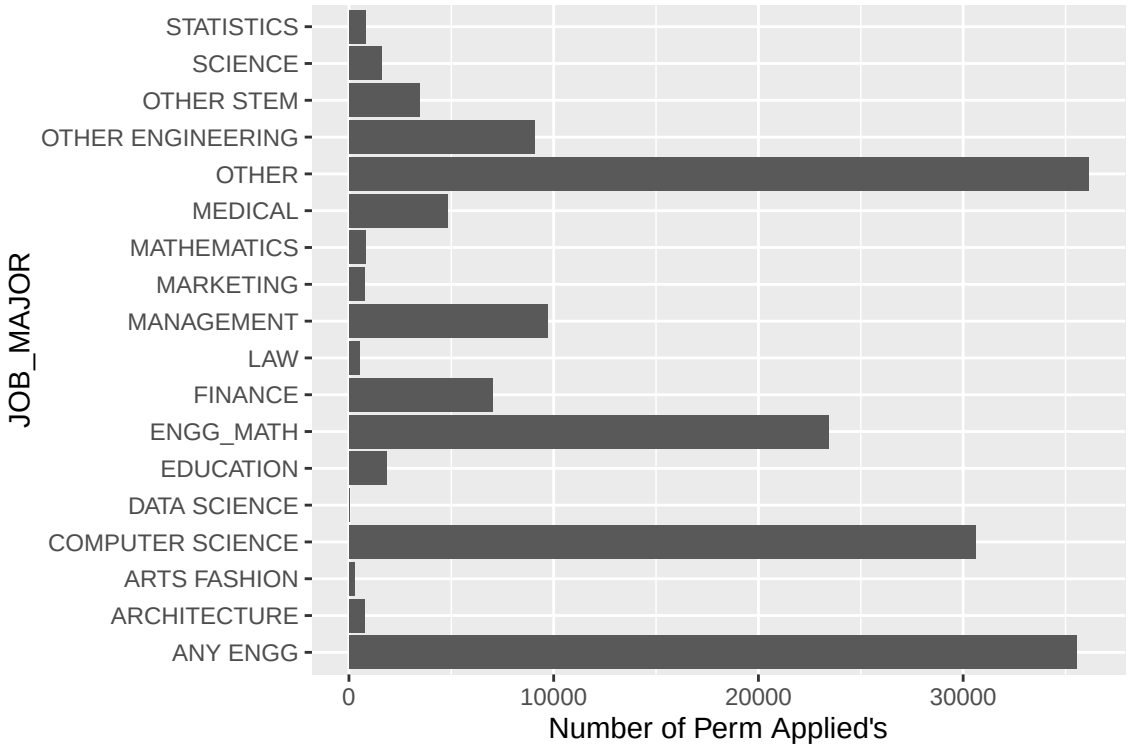

visas[which(visas$naics_title == "Computer Systems Design Services"),]$
Frequency
0
5000
10000
15000
20000
0
100000
200000
300000
400000
500000
visas[which(visas$naics_title == "Computer Systems Design Services"),]$normalized_wag

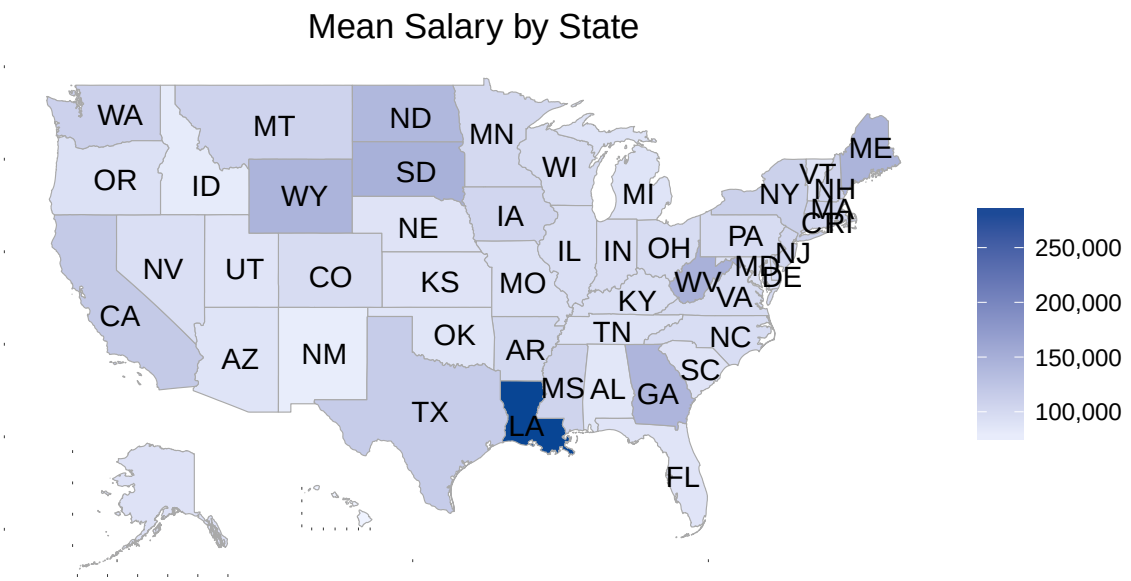


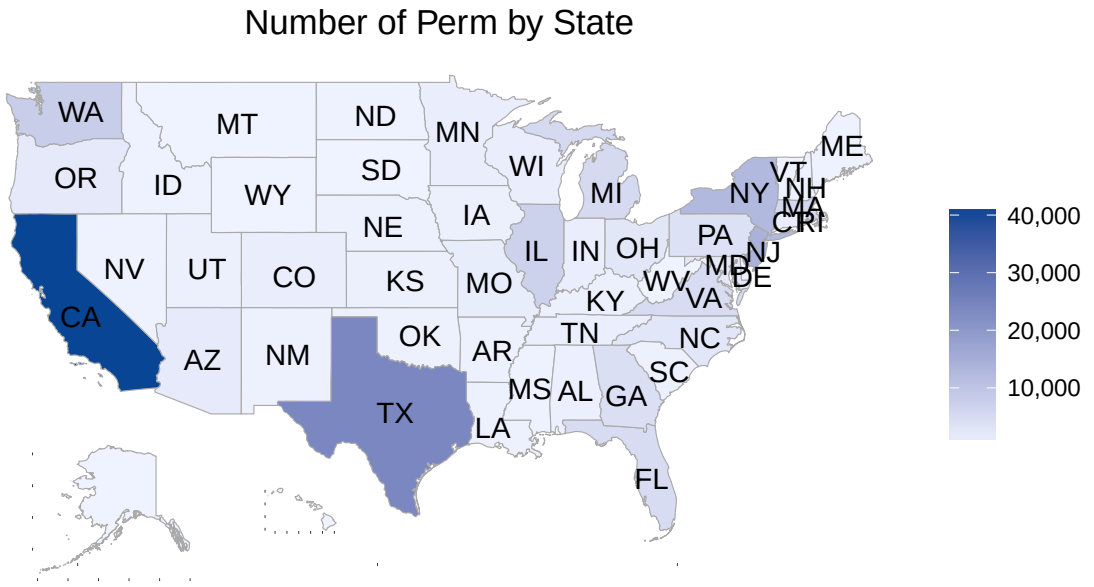

```
## Loading required package:  gsubfn
## Loading required package:  proto
## Loading required package:  RSQLite
## Loading required package:  DBI
## Loading required package:  acs
## Loading required package:  stringr
## Loading required package:  plyr
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package:  'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##    arrange, count, desc, failwith, id, mutate, rename, summarise,
##    summarize
## Loading required package:  XML
##
## Attaching package:  'acs'
## The following object is masked from 'package:dplyr':
##
##    combine
## The following object is masked from 'package:base':
##
##    apply
##
## Attaching package:  'DescTools'
## The following object is masked from 'package:data.table':
##
##    %like%
## Loading required package:  tcltk
```

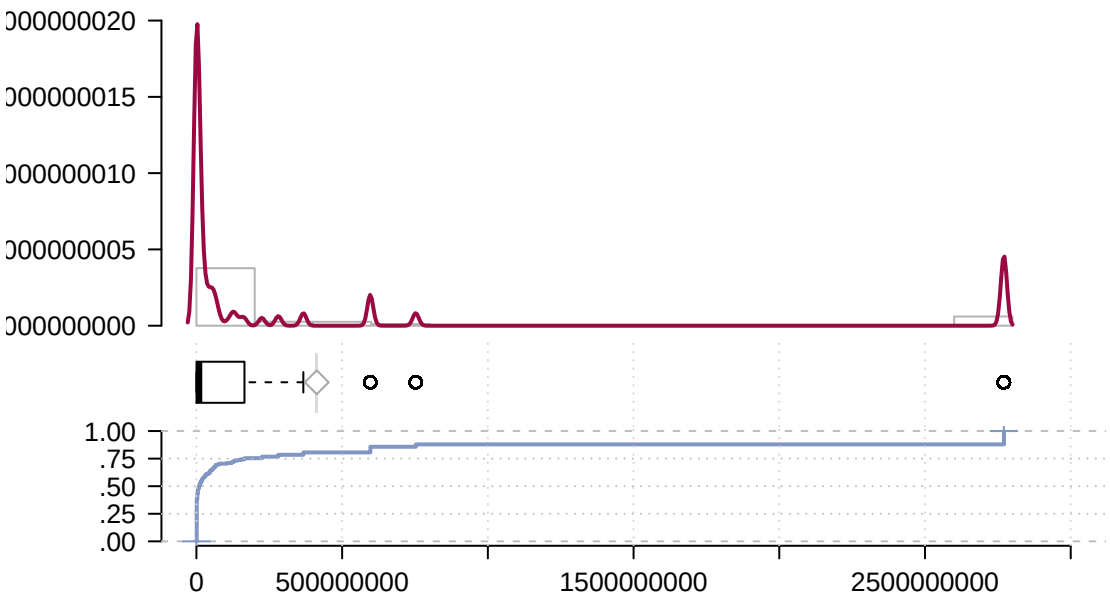



JOB_MAJOR
STATISTICS
SCIENCE
OTHER STEM
OTHER ENGINEERING
OTHER
MEDICAL
MATHEMATICS
MARKETING
MANAGEMENT
LAW
FINANCE
ENGG_MATH
EDUCATION
DATA SCIENCE
COMPUTER SCIENCE
ARTS FASHION
ARCHITECTURE
ANY ENGG
0
10000
20000
30000
Number of Perm Applied's

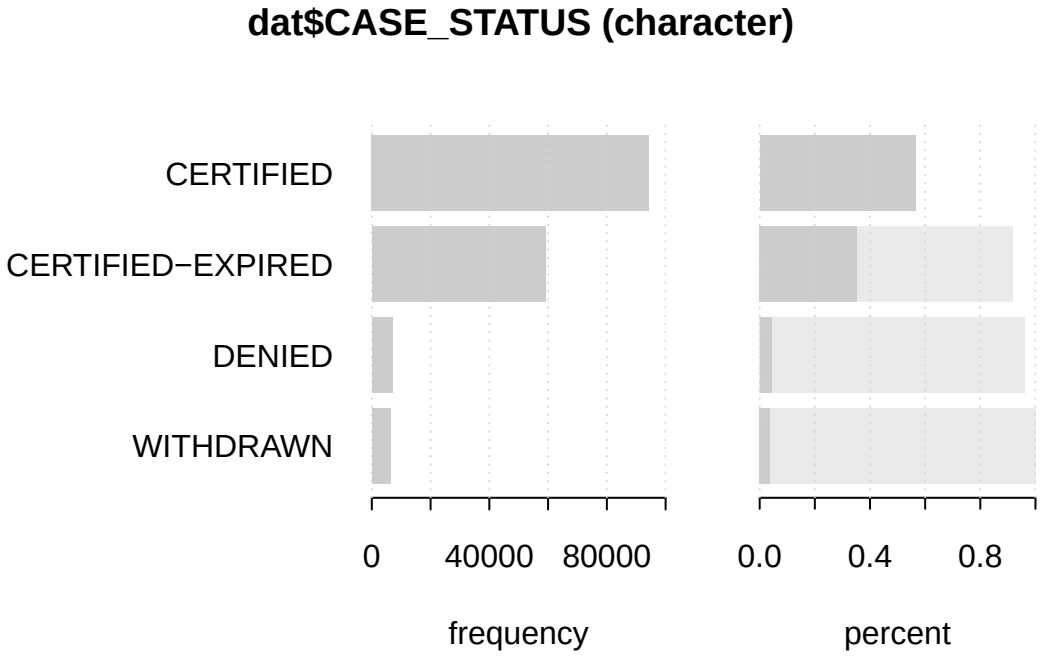

Mean Salary by State
WA
MT
ND
MN
OR
ID
SD
WI
MI
WY
NY
VT
NH
ME
MA
RI
CT
NE
IA
PA
NJ
IL
IN
OH
NV
UT
CO
KS
MO
MD
DE
WV
VA
KY
CA
TN
NC
AZ
NM
OK
AR
SC
MS
AL
GA
TX
LA
FL
250,000
200,000
150,000
100,000

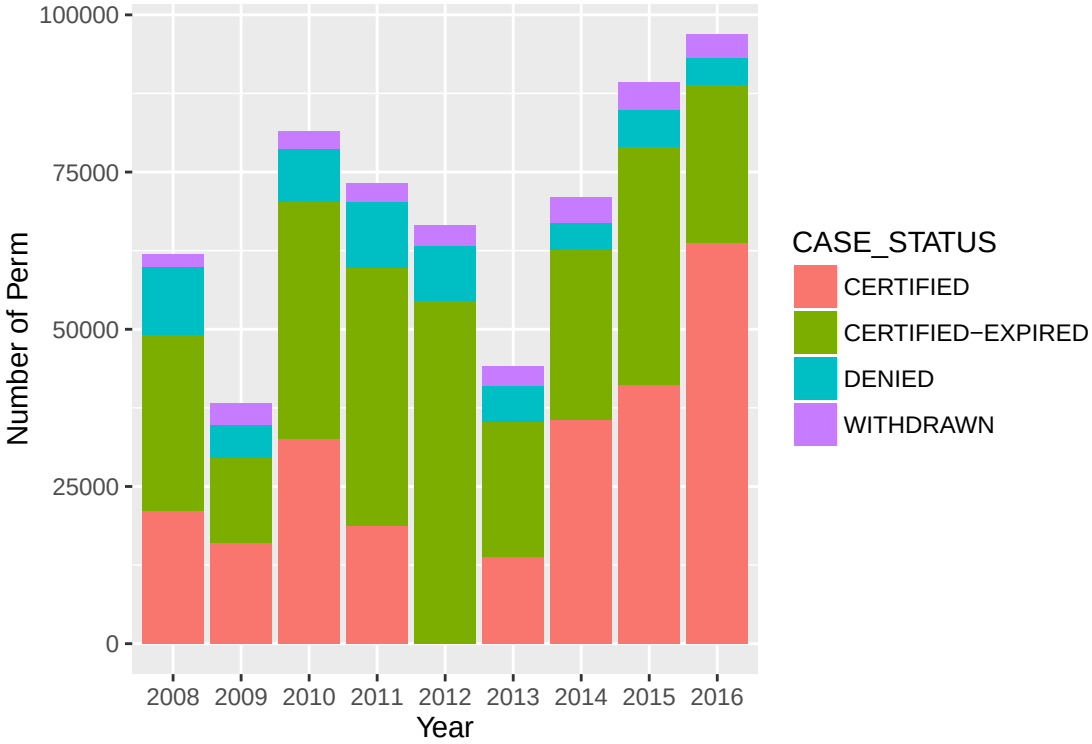


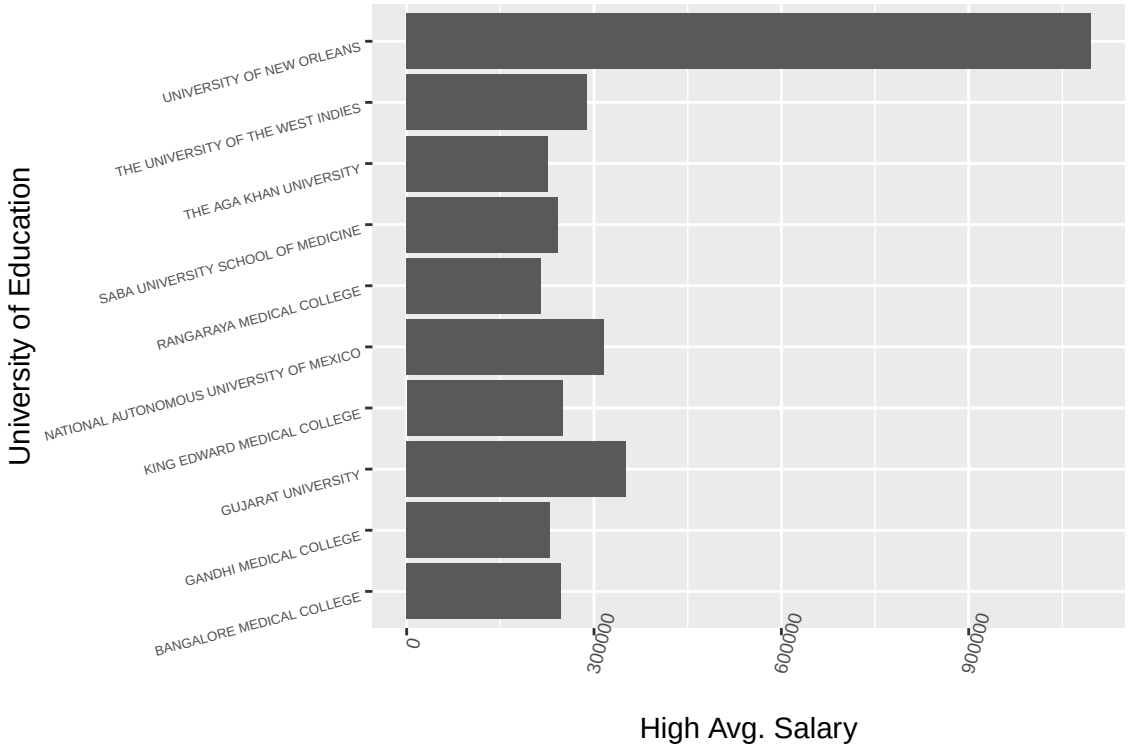

```
## ------------------------------------------------------------------------
## Salary distribution
##
##          length               n             NAs          unique              0s            mean          m
##         166'940         166'940               0         16'078              71    412896497.46     218223
##
##             .05             .10             .25          median             .75             .90
##        79122.85        98700.00      280683.75      9716042.00   164784658.39  2770620713.41  27706207
##
##           range              sd           vcoef             mad             IQR            skew
##   2770620713.41   891625878.16            2.16    14283430.67    164503974.64            2.17
##
## lowest : 0.0 (71), 2200.6, 15080.0 (5), 16800.0, 16869.0
## highest: 283180159.0 (9), 366904996.76 (3'577), 596542911.0 (8'550), 752476139.02 (3'646), 277062071
```




```
## ------------------------------------------------------------------------
## dat$CASE_STATUS (character)
##
##    length       n     NAs  unique  levels   dupes
##   166'940 166'940       0       4       4       y
##
##                level    freq   perc  cumfreq  cumperc
## 1          CERTIFIED  94'359  56.5%   94'359    56.5%
## 2  CERTIFIED-EXPIRED  59'049  35.4%  153'408    91.9%
## 3             DENIED   7'141   4.3%  160'549    96.2%
## 4          WITHDRAWN   6'391   3.8%  166'940   100.0%
```


dat$CASE_STATUS (character)
CERTIFIED
CERTIFIED−EXPIRED
DENIED
WITHDRAWN
0
40000
80000
frequency
0.0
0.4
0.8
percent


100000
75000
50000
25000
0
Number of Perm
2008
2009
2010
2011
2012
2013
2014
2015
2016
Year
CASE_STATUS
CERTIFIED
CERTIFIED−EXPIRED
DENIED
WITHDRAWN



4.3 Offered Wage

4.4 H-1B vs. PERM

5 Conclusion

```
#visas <- readRDS('H1BVisas.rds')
#perm <- readRDS('PermData.rds')
#perm.map <- readRDS('PermEmpMapsdat.rds')

#hist(visas[which(visas$normalized_wage < 250000),]$normalized_wage, breaks=500)

#findmode <- function(x, na.rm = TRUE) {

  #if(na.rm){
    #x = x[!is.na(x)]
  #}

  #ux <- unique(x)
  #return(ux[which.max(tabulate(match(x, ux)))])
#}

#wage.mode <- findmode(visas$normalized_wage)
#wage.mode
#abline(v=wage.mode + 500, col="red")

#hist(visas[which(visas$normalized_prevailing_wage < 250000),]$normalized_prevailing_wage, breaks=500)
#pw.mode <- findmode(visas$normalized_prevailing_wage)
#pw.mode
#abline(v=wage.mode + 500, col="red")
```

References

[1] U.S. Department of Labor, Office of Foreign Labor Certification Disclosure Data, https://www.foreignlaborcert.doleta.gov/performancedata.cfm