H1-B Visa Petitions for Data Science Positions in 2015

Sharan Naribole

Contributed by Sharan Naribole. He is currently undertaking the part-time online bootcamp organized by NYC Data Science Academy (Dec 2016- April 2017). This blog is based on his bootcamp project - R Exploratory Data Analysis

Abstract

The H1-B is an employment-based, non-immigrant visa category for temporary foreign workers in the United States. Every year, the US immigration department accepts over 200,000 petitions and selects 85,000 applications through a random process. The application data is available for public access to perform in-depth longitudinal research and analysis. This data provides key insights into the prevailing wages for job titles being sponsored by US employers under H1-B visa category. In particular, I utilize the 2015 H1-B petition disclosure data to analyze the Salary distribution across different industries, states and seniority levels for Data Science positions.

Let's begin by loading R packages.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(zipcode)
library(readxl)
```

H1-B Visa Data Introduction

The H1-B is an employment-based, non-immigrant visa category for temporary foreign workers in the United States. For a foreign national to apply for H1-B visa, an US employer must offer a job and petition for H1-B visa with the US immigration department. This is the most common visa status applied for and held by international students once they complete college/ higher education (Masters, PhD) and work in a full-time position.

The Office of Foreign Labor Certification (OFLC) generates program data that is useful information about the immigration programs including the H1-B visa. The disclosure data updated annually is available at https://www.foreignlaborcert.doleta.gov/performancedata.cfm

Loading H1-B Visa data for 2015 downloaded from

```
h1b_data = read_excel("data/H-1B_Disclosure_Data_FY15_Q4.xlsx")
```

Let's explore the columns in our data

```
colnames(h1b_data)
```

```
##  [1] "CASE_NUMBER"           "CASE_STATUS"
##  [3] "CASE_SUBMITTED"        "DECISION_DATE"
##  [5] "VISA_CLASS"            "EMPLOYMENT_START_DATE"
```

```
##  [7] "EMPLOYMENT_END_DATE"   "EMPLOYER_NAME"
##  [9] "EMPLOYER_ADDRESS1"     "EMPLOYER_ADDRESS2"
## [11] "EMPLOYER_CITY"         "EMPLOYER_STATE"
## [13] "EMPLOYER_POSTAL_CODE"  "EMPLOYER_COUNTRY"
## [15] "EMPLOYER_PROVINCE"     "EMPLOYER_PHONE"
## [17] "EMPLOYER_PHONE_EXT"    "AGENT_ATTORNEY_NAME"
## [19] "AGENT_ATTORNEY_CITY"   "AGENT_ATTORNEY_STATE"
## [21] "JOB_TITLE"             "SOC_CODE"
## [23] "SOC_NAME"              "NAIC_CODE"
## [25] "TOTAL WORKERS"         "FULL_TIME_POSITION"
## [27] "PREVAILING_WAGE"       "PW_UNIT_OF_PAY"
## [29] "PW_WAGE_LEVEL"         "PW_WAGE_SOURCE"
## [31] "PW_WAGE_SOURCE_YEAR"   "PW_WAGE_SOURCE_OTHER"
## [33] "WAGE_RATE_OF_PAY"      "WAGE_UNIT_OF_PAY"
## [35] "H-1B_DEPENDENT"        "WILLFUL VIOLATOR"
## [37] "WORKSITE_CITY"         "WORKSITE_COUNTY"
## [39] "WORKSITE_STATE"        "WORKSITE_POSTAL_CODE"
```

The useful columns for our data analysis include:

1) EMPLOYER_NAME : Name of employer submitting the H1-B application.
2) JOB_TITLE : Title of the job using which we can filter the Data Science positions and the Seniority Level
3) SOC_NAME The broad area/industry associated with a job as classified by the Standard Occupational (SOC) System. This gives us insight into the fields in which Data Scientist positions are being offered.
4) PREVAILING_WAGE The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer's minimum requirements for the position. (Source: https://www.usavisanow.com/h-1b-visa/h1b-visa-resources/prevailing-wage/). This column will be our key metric in the data analysis.
5) WORKSITE_CITY, WORKSITE_STATE The foreign worker's intended area of employment. We will explore the relationship between prevailing wage for Data Scientist position across different locations.

I focus on the annual prevailing wage for full-time positions. Consequently, the rows confirming full_time positions are filtered.

```
h1b_df = h1b_data %>% filter(FULL_TIME_POSITION == 'Y')
```

The column PW_UNIT_OF_PAY indicates the frequency of payment. As shown below, the PW_UNIT_OF_PAY varies from hour to Year.

```
h1b_df %>% select(PW_UNIT_OF_PAY ) %>% sapply(function(x) unique(x))
```

```
##      PW_UNIT_OF_PAY
## [1,] "Year"
## [2,] "Hour"
## [3,] "Month"
## [4,] "Week"
## [5,] NA
## [6,] "Bi-Weekly"
```

First, the rows with missing values for PW_UNIT_OF_PAY need to be removed. Then, we convert wage of remaining rows to annual scale.

```
h1b_df = h1b_df %>% filter(!is.na(PW_UNIT_OF_PAY))
h1b_df = h1b_df %>% mutate(PREVAILING_WAGE = as.numeric(PREVAILING_WAGE))
```

```
#Function to transform wage to annual scale
pw_unit_to_yearly = function(prevailing_wage, pw_unit_of_pay) {
  return(ifelse(pw_unit_of_pay == "Year", prevailing_wage, ifelse(pw_unit_of_pay == "Hour", 2080*prevail
}

h1b_df = h1b_df %>% mutate(PREVAILING_WAGE = pw_unit_to_yearly(PREVAILING_WAGE, PW_UNIT_OF_PAY))
h1b_df = h1b_df %>% mutate(PW_UNIT_OF_PAY = "Year")
```

Next step is to classify Jobs based on the seniority of the position. This is because seniority results in a higher wage. For example, a Senior/ Lead Data Scientist is expected to earn more than a regular Data Scientist.

```
h1b_df = h1b_df %>% mutate(lead = ifelse(regexpr('lead', tolower(JOB_TITLE)) != -1 | regexpr('senior', 

h1b_df = h1b_df %>% mutate(manager = ifelse(regexpr('manager', tolower(JOB_TITLE)) != -1 | regexpr('dir

h1b_df = h1b_df %>% mutate(data_scientist = ifelse(regexpr('data scientist', tolower(JOB_TITLE)) != -1,

h1b_df = h1b_df %>% mutate(data_analyst = ifelse(regexpr('data analyst', tolower(JOB_TITLE)) != -1, 1, 

h1b_df = h1b_df %>% mutate(data_engineer = ifelse(regexpr('data engineer', tolower(JOB_TITLE)) != -1, 1

h1b_df = h1b_df %>% mutate(machine_learning = ifelse(regexpr('machine learning', tolower(JOB_TITLE)) !=

h1b_df = h1b_df %>% mutate(job_class = ifelse(lead == 1, 2, ifelse(manager == 1, 3, 1)))
```

Last part of the data transformation is retaining only the useful columns.

```
h1b_df = h1b_df %>%  select(EMPLOYER_NAME, JOB_TITLE, SOC_NAME, PREVAILING_WAGE, WORKSITE_CITY, WORKSIT
```

Exploratory Data Analysis

Next, I analyze the prevailing wages for Data Scientist positions.

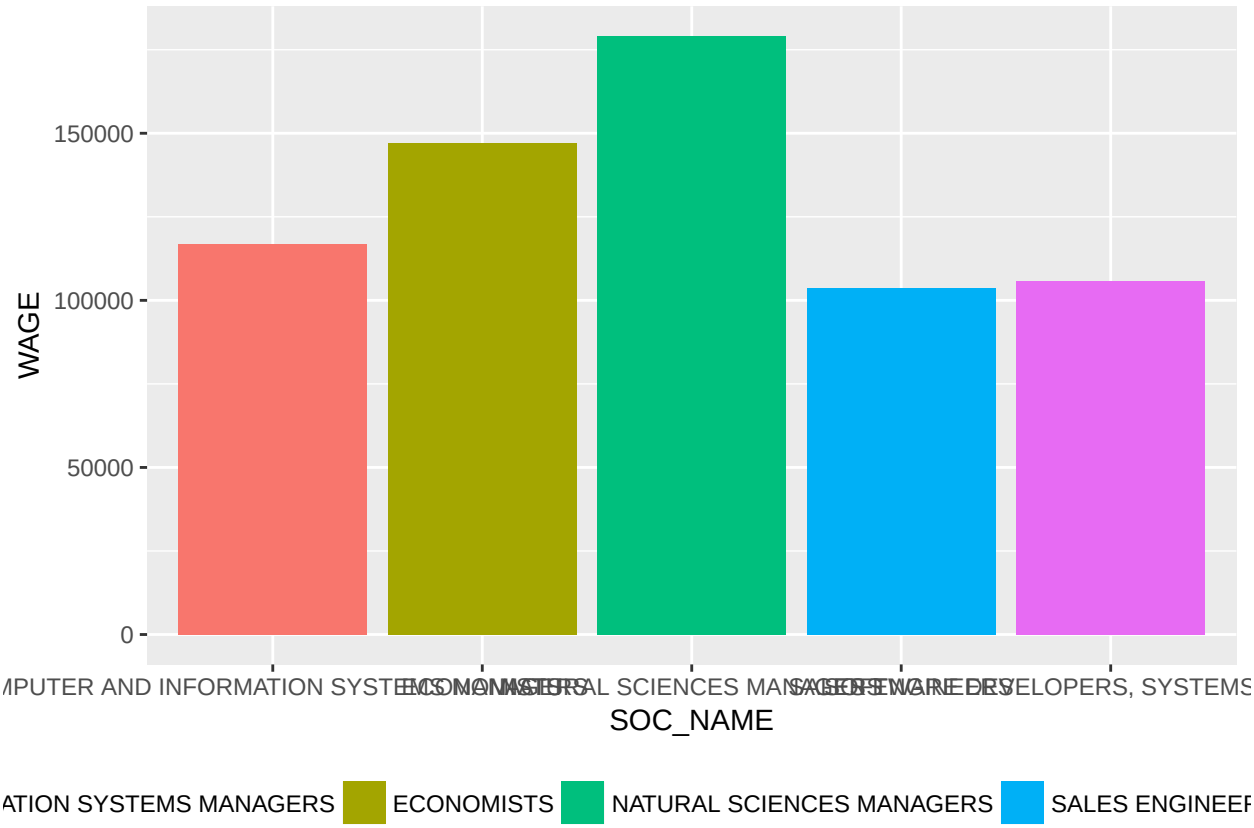
Seniority Analysis

```
h1b_df = h1b_df %>% mutate(JOB_CLASS = as.factor(JOB_CLASS))
seniority = h1b_df %>% filter(data_scientist == 1) %>% group_by(JOB_CLASS)

ggplot(data = seniority,aes(x = JOB_CLASS,y=PREVAILING_WAGE)) + geom_violin(aes(fill=JOB_CLASS)) + scal
```


```
soc_names = h1b_df %>% filter(data_scientist == 1) %>% group_by(SOC_NAME) %>% summarise(WAGE = mean(PRE

soc_names
```

```
## # A tibble: 35 × 2
##                                       SOC_NAME      WAGE
##                                          <chr>     <dbl>
## 1                    NATURAL SCIENCES MANAGERS 179046.00
## 2                                   ECONOMISTS 146976.00
## 3     COMPUTER AND INFORMATION SYSTEMS MANAGERS 116708.60
## 4         SOFTWARE DEVELOPERS, SYSTEMS SOFTWARE 105736.29
## 5                               SALES ENGINEERS 103688.00
## 6  COMPUTER AND INFORMATION RESEARCH SCIENTISTS 102789.70
## 7             SOFTWARE DEVELOPERS, APPLICATIONS 100969.03
## 8              COMPUTER OCCUPATIONS, ALL OTHER*  96034.00
## 9                           FINANCIAL ANALYSTS  95545.00
## 10                              MATHEMATICIANS  92144.61
## # ... with 25 more rows
```

```
ggplot(data = soc_names[1:5,],aes(x = SOC_NAME,y=WAGE)) + geom_bar(stat="identity",aes(fill=SOC_NAME)) +
```


150000
100000
50000
0
WAGE
SOC_NAME
ATION SYSTEMS MANAGERS
ECONOMISTS
NATURAL SCIENCES MANAGERS
SALES ENGINEER