Code in R studio:

**Data Preparation**

# Load all of the packages used in the following analysis

library(dplyr)

library(ggplot2)

library(readxl)

library(hashmap)

library(ggmap)

library(tidyr)

library(RColorBrewer)

library(knitr)

library(gridExtra)

#Empty data frame

h1b\_df = data.frame()

new\_df = data.frame()

# Load each year's data

for(year in seq(2017,2012)) {

print(paste0("Year=", as.character(year)))

raw\_data\_path = paste0("~/Downloads/capstone project/DATA/",year,"rawdata.xlsx")

raw\_data\_path

new\_df = read\_excel(raw\_data\_path)

print(paste0( as.character(year), " Raw data size: ", as.character(dim(new\_df))))

# Changing column names of data before 2015

if(year != 2015 & year != 2016 & year != 2017){

new\_df = new\_df %>%

mutate(CASE\_NUMBER = LCA\_CASE\_NUMBER,

CASE\_STATUS = STATUS,

EMPLOYER\_NAME = LCA\_CASE\_EMPLOYER\_NAME,

SOC\_NAME = LCA\_CASE\_SOC\_NAME,

SOC\_CODE = LCA\_CASE\_SOC\_CODE,

JOB\_TITLE = LCA\_CASE\_JOB\_TITLE,

FULL\_TIME\_POSITION = FULL\_TIME\_POS,

PREVAILING\_WAGE = PW\_1,

PW\_UNIT\_OF\_PAY = PW\_UNIT\_1,

WORKSITE\_CITY = LCA\_CASE\_WORKLOC1\_CITY,

WORKSITE\_STATE = LCA\_CASE\_WORKLOC1\_STATE)

}

#write.csv(new\_df,"~/Downloads/capstone project/DATA/2011rawdatadeal.csv")

# Adding Year column to dataframe

new\_df = new\_df %>%

mutate(YEAR = as.character(year))

print(paste0( as.character(year), " Mutated data size: ", as.character(dim(new\_df))))

# Selecting only the relevant columns

new\_df = new\_df %>%

select(CASE\_NUMBER,

CASE\_STATUS,

EMPLOYER\_NAME,

SOC\_NAME,

SOC\_CODE,

JOB\_TITLE,

FULL\_TIME\_POSITION,

PREVAILING\_WAGE,

PW\_UNIT\_OF\_PAY,

WORKSITE\_CITY,

WORKSITE\_STATE,

YEAR)

# Merging data with already transformed data

h1b\_df = rbind(h1b\_df, new\_df)

print(paste0( as.character(year), " Merged data size: ",as.character(dim(h1b\_df))))

}

write.csv(h1b\_df,"~/Downloads/capstone project/DATA/h1bdata.csv")

# Saving read data frame

saveRDS(h1b\_df,"h1b\_df\_no\_transform.rds")

# h1b\_df\_tx will undergo all transformations

h1b\_df\_tx <- h1b\_df

colnames(h1b\_df\_tx)

glimps(h1b\_df\_tx)

**H-1B Petition Case Status**

# Convert all the lowercase characters to uppercase

h1b <- data.frame(lapply(h1b\_df\_tx, function(v) {

if (is.character(v)) return(toupper(v))

else return(v)

}))

h1b <- tbl\_df(h1b)

glimpse(h1b)

# Keep only "CERTIFIED" H1B cases

certified\_h1b <- h1b %>%

filter(CASE\_STATUS == "CERTIFIED")

glimpse(certified\_h1b)

# Count H1B petitions filed in each year

case\_quantity\_per\_year <- certified\_h1b %>%

group\_by(YEAR) %>%

summarise(Certified = n())

# Bar plot showing the H1B quantities in each year

ggplot(case\_quantity\_per\_year, aes(y = Certified, x = YEAR, fill = YEAR)) +

geom\_bar(stat = "identity", alpha = 0.7, width = 0.5) +

scale\_y\_continuous(limits = c(0, 570000),

breaks = seq(0, 570000, 100000),

labels = scales::comma) +

ggtitle("Certified H-1B Petitios in Each Year")+

theme(

plot.title = element\_text(size = rel(1.3)),

panel.background = element\_rect(fill = '#f0f0f0'),

legend.position = "none"

)

**Top 10 occupations that have the most H1B applicants**

# Function to return top N occupations that have the most H1B applicants

top\_N\_SOC <- function(num) {

certified\_h1b %>%

filter(!is.na(certified\_h1b$SOC\_NAME)) %>%

group\_by(SOC\_NAME) %>%

summarise(num\_apps = n()) %>%

arrange(desc(num\_apps)) %>%

slice(1:num)

}

# Bar plot to show the top 10 H1B occupations

ggplot(top\_N\_SOC(10),

aes(x = reorder(SOC\_NAME, num\_apps), y = num\_apps)) +

geom\_bar(stat = "identity", alpha = 0.9, fill = "orange", width = 0.7) +

coord\_flip() +

scale\_y\_continuous() +

geom\_text(aes(label = num\_apps), hjust = -0.2, size = 2) +

ggtitle("Top 10 Occupations with Most H1B Petitions in 2012-2017 Period") +

theme(plot.title = element\_text(size = rel(1)),

axis.text.y = element\_text(size = rel(0.8))) +

labs(x = "SOC Name", y = "No. of Applications")

**Top 10 Employers that have the most H1B applicants**

# Function to return the top N employers that have the most H1B workers

top\_N\_employers <- function(num\_emp) {

certified\_h1b %>%

group\_by(EMPLOYER\_NAME) %>%

summarise(num\_apps = n()) %>%

arrange(desc(num\_apps)) %>%

slice(1:num\_emp)

}

# Bar plot to show the top 10 employers who filed the most h1b visa applications

ggplot(top\_N\_employers(10),

aes(x = reorder(EMPLOYER\_NAME, num\_apps), y = num\_apps)) +

geom\_bar(stat = "identity", alpha = 0.9, fill = "sky blue", width = 0.7) +

coord\_flip() +

scale\_y\_continuous(limits = c(0, 160000), breaks = seq(0, 140000, 20000)) +

geom\_text(aes(label = num\_apps), hjust = -0.2, size = 2) +

ggtitle("Top 10 Employers with most applications") +

theme(plot.title = element\_text(size = rel(1)),

axis.text.y = element\_text(size = rel(0.8))) +

labs(x = "Employer Name", y = "No. of Applications")

**H1B petitions by states**

# Count H1B petitions filed by each state

petition\_by\_state <- certified\_h1b %>%

filter(WORKSITE\_STATE != "NA") %>%

group\_by(region = tolower(WORKSITE\_STATE)) %>%

summarise(no\_petitions = n()) %>%

arrange(desc(no\_petitions))

g <- ggplot(petition\_by\_state[1:10,], aes(x=reorder(region,no\_petitions),y=no\_petitions)) +

geom\_bar(stat = "identity", fill = "blue") + coord\_flip() +

ggtitle("H1B petitions by state in 2012-2017 Period")+

get\_theme() +

labs(x = "Number of petitions", y = "State") + scale\_color\_discrete()

g

**Count H1B petitions filed in CA, NY and TX in each year**

# Count H1B petitions filed in CA, NY and TX in each year

cnt\_case\_per\_year <- certified\_h1b %>%

filter(WORKSITE\_STATE %in% c("CA", "NY", "TX")) %>%

group\_by(YEAR, WORKSITE\_STATE) %>%

summarise(count = n()) %>%

arrange(YEAR, WORKSITE\_STATE)

# Bar plot showing H1B quantities of each state in each year

ggplot(cnt\_case\_per\_year, aes(x = YEAR, y = count, fill = WORKSITE\_STATE)) +

geom\_bar(stat = "identity", position = position\_dodge(), alpha = 0.8,

color = "grey") +

ggtitle("Quantity of H1B cases in California, New York and Texas from 2012 to 2017") +

theme(legend.position = "bottom",

plot.title = element\_text(size = rel(1.3)))

**Prevailing wages of top 10 employers**

# Top 10 employers who filed the most H1B petitions

top\_10\_employers <- certified\_h1b %>%

group\_by(EMPLOYER\_NAME) %>%

summarise(num\_apps = n()) %>%

arrange(desc(num\_apps)) %>%

slice(1:10) %>%

select(EMPLOYER\_NAME)

employers\_boxplot\_df <- certified\_h1b %>%

filter(EMPLOYER\_NAME %in% top\_10\_employers$EMPLOYER\_NAME)

# Boxplot showing the wage distribution of each employer

ggplot(employers\_boxplot\_df, aes(y = PREVAILING\_WAGE, x = EMPLOYER\_NAME,

fill = EMPLOYER\_NAME, notch = TRUE, notchwidth = .3)) +

geom\_boxplot(notch = TRUE) +

#scale\_y\_continuous(limits = c(0, 150000),

#breaks = seq(0, 150000, 10000)) +

ggtitle("Wages for H1B cases in top 10 companies")+

theme(

plot.title = element\_text(size = rel(1.3)),

panel.background = element\_rect(fill = '#f0f0f0'),

axis.text.x=element\_blank(),

legend.position = "bottom",

legend.title = element\_text(size = rel(0.7)),

legend.text = element\_text(size = rel(0.4)),

panel.grid.major = element\_line(colour = '#f0f0f0'),

panel.grid.major.x = element\_line(linetype = 'blank'),

panel.grid.minor = element\_line(linetype = 'blank')

)

**Top 10 occupations with highest median prevailing wages**

# Top 10 occupations with the highest wages

top\_10\_soc\_with\_highest\_wage <- certified\_h1b %>%

group\_by(SOC\_NAME) %>%

summarise(median\_wage = median(PREVAILING\_WAGE)) %>%

arrange(desc(median\_wage)) %>%

slice(1:10) %>%

select(SOC\_NAME, median\_wage)

# Bar plot showing median wages for each occupation

ggplot(top\_10\_soc\_with\_highest\_wage, aes(y = median\_wage, x = reorder(SOC\_NAME, median\_wage))) +

geom\_bar(stat = "identity", fill = "blue", alpha = 0.7, width = 0.7) +

#scale\_y\_continuous(limits = c(0, 150000),

#breaks = seq(0, 150000, 5000)) +

ggtitle("Top 10 occupations with highest median prevailing wages") +

coord\_flip() +

theme(

plot.title = element\_text(size = rel(1)),

axis.text.x=element\_text(size = rel(0.8)),

legend.position = "bottom"

) +

labs(x = "Occupational Name")

knitr::kable(top\_10\_soc\_with\_highest\_wage)

**Trend of wages from 2012 to 2017**

set.seed(123)

# Filter out the top 1% outliers in the prevailing\_wage variable

wage\_year\_sample <- subset(certified\_h1b[sample(1:nrow(certified\_h1b), 300000),

c(8, 12)],

!is.na(PREVAILING\_WAGE) &

PREVAILING\_WAGE <= quantile(certified\_h1b$PREVAILING\_WAGE,

0.99))

wage\_year\_sample <- wage\_year\_sample %>%

group\_by(YEAR) %>%

mutate(mean\_wage = mean(PREVAILING\_WAGE),

median\_wage = median(PREVAILING\_WAGE),

'10th\_percentile' = quantile(PREVAILING\_WAGE, 0.1),

'90th\_percentile' = quantile(PREVAILING\_WAGE, 0.9))

wage\_year\_stats <- wage\_year\_sample %>%

distinct(mean\_wage, median\_wage, `10th\_percentile`, `90th\_percentile`)

wage\_year\_stats <- wage\_year\_stats[order(wage\_year\_stats$YEAR),

c(ncol(wage\_year\_stats),

1:(ncol(wage\_year\_stats) - 1))]

# From wide to long format

wage\_year\_stats\_long <- gather(wage\_year\_stats, statistics, values,

mean\_wage, median\_wage,`10th\_percentile`, `90th\_percentile`,

factor\_key = TRUE)

# Trend of median, mean, 10th percentile and 90th percentile of wages

wage\_trend <-

ggplot(wage\_year\_stats\_long ,

aes(x = YEAR, y = values, group = statistics)) +

geom\_line(aes(color = statistics), lineend = "round", size = 1) +

expand\_limits(y = 0) +

scale\_y\_continuous(breaks = seq(0, 120000, 10000), labels = scales::comma) +

ggtitle("Wage Trend from 2011 to 2016: Line Chart") +

labs(y = "wage / $") +

theme(plot.title = element\_text(size = rel(1.3)),

legend.position = "bottom")

wage\_trend

**Changes in the proportion of computer and mathematical occupations**

# Computer and Mathematical Occupations

cmo <- c("COMPUTER AND INFORMATION RESEARCH SCIENTISTS",

"COMPUTER SYSTEMS ANALYSTS", "INFORMATION SECURITY ANALYSTS",

"COMPUTER PROGRAMMERS", "SOFTWARE DEVELOPERS, APPLICATIONS",

"SOFTWARE DEVELOPERS, SYSTEMS SOFTWARE", "WEB DEVELOPERS",

"DATABASE ADMINISTRATORS",

"NETWORK AND COMPUTER SYSTEMS ADMINISTRATORS",

"COMPUTER NETWORK ARCHITECTS",

"COMPUTER USER SUPPORT SPECIALISTS", "COMPUTER NETWORK SUPPORT SPECIALISTS",

"COMPUTER OCCUPATIONS, ALL OTHER", "ACTURIES", "MATHEMATICIANS",

"OPERATIONS RESEARCH ANALYSTS",

"STATISTICIANS", "MATHEMATICAL TECHNICIANS",

"MATHEMATICAL SCIENCE OCCUPATIONS, ALL OTHER")

# Generate a dataset containing H1B records with computer and mathematical occupations

cmo\_quantity\_per\_year <- certified\_h1b %>%

filter(SOC\_NAME %in% cmo & !is.na(PREVAILING\_WAGE)) %>%

#certified\_h1b$PREVAILING\_WAGE.numbers <- as.numeric(as.vector(certified\_h1b$PREVAILING\_WAGE))

group\_by(YEAR) %>%

summarise(cm\_occupations = n())

#cm\_median\_wage = median(PREVAILING\_WAGE))

# Generate a dataset containing H1B records with computer and mathematical occupations

case\_quantity\_per\_year <- certified\_h1b %>%

#filter(SOC\_NAME %in% cmo & !is.na(PREVAILING\_WAGE)) %>%

#certified\_h1b$PREVAILING\_WAGE.numbers <- as.numeric(as.vector(certified\_h1b$PREVAILING\_WAGE))

group\_by(YEAR) %>%

summarise(all\_occupations = n())

#cm\_median\_wage = median(PR

# Join two datasets together

multi\_df1 <- merge(x = cmo\_quantity\_per\_year,

y = case\_quantity\_per\_year,

by = "YEAR",

all = TRUE)

# Calculate the percentage of computer and mathematical occupations

multi\_df1 <- multi\_df1 %>%

mutate(other\_occupations = all\_occupations - cm\_occupations,

cm\_percent = cm\_occupations / all\_occupations) %>%

select(YEAR, cm\_occupations, other\_occupations)

multi\_df1 <- gather(multi\_df1, occupation, count, cm\_occupations:other\_occupations)

# Stacked bar plot showing the proportions of computer and

# mathematical occupations over years

ggplot(multi\_df1,

aes(x = YEAR, y = count, fill = occupation)) +

geom\_bar(stat = "identity") +

scale\_y\_continuous(breaks = seq(0, 600000, 100000),

limits = c(0, 600000), labels = scales::comma) +

labs(x = "Year", y = "No. of H1B Cases",

title = "Change in the proportion of computer and mathematical occupations")