Data Cleaning Process Code

Create new variable "SOC__CODE__major": used for creating "Occupational Classification" later

```r
library(ggplot2)
library(tidyverse)

data$SOC_CODE <- as.character(data$SOC_CODE)
data$SOC_CODE[which((data$SOC_CODE == "") & (data$JOB_TITLE == "JUNIOR ARCHITECT"))] <- "17-1011"
data$SOC_CODE[which((data$SOC_CODE == "") & (data$JOB_TITLE == "DRIVER"))] <- "53-0000"
#data <- data[which(data$SOC_CODE!=""),]   #if you want to drop NAs, uncomment this line

SOC_CODE_pattern <- "([1-5][0-9])[-]([0-9]{4})"
data$SOC_CODE <- ifelse(str_detect(data$SOC_NAME, SOC_CODE_pattern)==TRUE,str_match(data$SOC_NAME, SOC_C
data$SOC_CODE[which(data$SOC_CODE == "25/1071")] <- "25-1071"
data$SOC_CODE[which(data$SOC_CODE == "532011")] <- "53-2011"
data$SOC_CODE[which(data$SOC_CODE == "71-2141")] <- "17-2141"
data$SOC_CODE[which(data$SOC_CODE == "291069")] <- "29-1069"
data$SOC_CODE[which(data$SOC_CODE == "132011")] <- "13-2011"
data$SOC_CODE[which(data$SOC_CODE == "132051")] <- "13-2051"
data$SOC_CODE[which(data$SOC_CODE == "13-181.01")] <- "13-1081"
data$SOC_CODE[which(data$SOC_CODE == "151132")] <- "15-1132"
data$SOC_CODE[which(data$SOC_CODE == "29.1062")] <- "29-1062"
data$SOC_CODE[which(data$SOC_CODE == "43.9111.01")] <- "43-9111.01"
data$SOC_CODE[which(data$SOC_CODE == "15.1199.09")] <- "15-1199"

#CARETE NEW VARIABLE: SOC_CODE_major
data<-mutate(data, SOC_CODE_major = str_sub(data$SOC_CODE,1,2))
data$SOC_CODE_major <-as.factor(data$SOC_CODE_major)
data$SOC_CODE <- as.character(data$SOC_CODE)
data$SOC_CODE[which((data$SOC_CODE == "") & (data$JOB_TITLE == "JUNIOR ARCHITECT"))] <- "17-1011"
data$SOC_CODE[which((data$SOC_CODE == "") & (data$JOB_TITLE == "DRIVER"))] <- "53-0000"
#data <- data[which(data$SOC_CODE!=""),]   #if you want to drop NAs, uncomment this line

SOC_CODE_pattern <- "([1-5][0-9])[-]([0-9]{4})"
data$SOC_CODE <- ifelse(str_detect(data$SOC_NAME, SOC_CODE_pattern)==TRUE,str_match(data$SOC_NAME, SOC_C
data$SOC_CODE[which(data$SOC_CODE == "25/1071")] <- "25-1071"
data$SOC_CODE[which(data$SOC_CODE == "532011")] <- "53-2011"
data$SOC_CODE[which(data$SOC_CODE == "71-2141")] <- "17-2141"
data$SOC_CODE[which(data$SOC_CODE == "291069")] <- "29-1069"
data$SOC_CODE[which(data$SOC_CODE == "132011")] <- "13-2011"
data$SOC_CODE[which(data$SOC_CODE == "132051")] <- "13-2051"
data$SOC_CODE[which(data$SOC_CODE == "13-181.01")] <- "13-1081"
data$SOC_CODE[which(data$SOC_CODE == "151132")] <- "15-1132"
data$SOC_CODE[which(data$SOC_CODE == "29.1062")] <- "29-1062"
data$SOC_CODE[which(data$SOC_CODE == "43.9111.01")] <- "43-9111.01"
data$SOC_CODE[which(data$SOC_CODE == "15.1199.09")] <- "15-1199"

#CARETE NEW VARIABLE: SOC_CODE_major
data<-mutate(data, SOC_CODE_major = str_sub(data$SOC_CODE,1,2))
data$SOC_CODE_major <-as.factor(data$SOC_CODE_major)
```

Create new variable "sector" and "description": used for creating "Major Industry" later

```
library(tidyverse)
#extract several columns into dataframe NN
NN <- data.frame(data$CASE_NUMBER,data$CASE_STATUS,data$SOC_CODE,data$SOC_NAME,data$NAICS_CODE,data$WORK
#create a new column digits two store two-digit sector number
NN$sector <- 0
NN[is.na(NN)] <- 0
NN$description <- "0"
#subset NN according to data.NAICS_CODE
d2 <- subset(NN,data.NAICS_CODE <99 ,
select=c(data.CASE_NUMBER,data.NAICS_CODE,sector,description))
d2$sector<- d2$data.NAICS_CODE

d3 <- subset(NN,data.NAICS_CODE <999 & NN$data.NAICS_CODE >99,
select=c(data.CASE_NUMBER,data.NAICS_CODE,sector,description))
d3$sector<- d3$data.NAICS_CODE%/%10

d4 <- subset(NN,data.NAICS_CODE <9999 & NN$data.NAICS_CODE >999,
select=c(data.CASE_NUMBER,data.NAICS_CODE,sector,description))
d4$sector<- d4$data.NAICS_CODE%/%100

d5 <- subset(NN,data.NAICS_CODE <99999 & NN$data.NAICS_CODE >9999,
select=c(data.CASE_NUMBER,data.NAICS_CODE,sector,description))
d5$sector<- d5$data.NAICS_CODE%/%1000

d6 <- subset(NN,NN$data.NAICS_CODE >99999,
select=c(data.CASE_NUMBER,data.NAICS_CODE,sector,description))
d6$sector<- d6$data.NAICS_CODE%/%10000


#merge d2 d3 d4 d5 d6 into d
d <- rbind(d2,d3,d4,d5,d6)
d<- setNames(d,c("CASE_NUMBER", "NAICS_CODE","sector","description"))

d$description[d$sector== 11 ]= "Agriculture, Forestry, Fishing and Hunting
                                (not covered in economic census)"
d$description[d$sector== 21 ]= "Mining, Quarrying, and Oil and Gas Extraction"
d$description[d$sector== 22 ]= "Utilities"
d$description[d$sector== 23 ]="Construction"
d$description[d$sector== 31 ]="Manufacturing"
d$description[d$sector== 32 ]="Manufacturing"
d$description[d$sector== 33 ]="Manufacturing"
d$description[d$sector== 42 ]="Wholesale Trade"
d$description[d$sector== 44 ]="Retail Trade"
d$description[d$sector== 45 ]="Retail Trade"
d$description[d$sector== 48 ]="Transportation and Warehousing"
d$description[d$sector== 49 ]="Transportation and Warehousing"
d$description[d$sector== 51 ]="Information"
d$description[d$sector== 52 ]="Finance and Insurance"
d$description[d$sector== 53 ]="Real Estate and Rental and Leasing"
d$description[d$sector== 54 ]="Professional, Scientific, and Technical Services"
d$description[d$sector== 55 ]="Management of Companies and Enterprises"
d$description[d$sector== 56 ]="Administrative and Support and Waste Management
                                and Remediation Services"
```

```
d$description[d$sector== 61 ]="Educational Services"
d$description[d$sector== 62 ]="Health Care and Social Assistance"
d$description[d$sector== 71 ]="Arts, Entertainment, and Recreation"
d$description[d$sector== 72 ]="Accommodation and Food Services"
d$description[d$sector== 81 ]="Other Services (except Public Administration)"
d$description[d$sector== 92 ]="Public Administration (not covered in economic census)"

##merge data

newdata <- merge(x=data,y=d, by = "CASE_NUMBER", all = TRUE)
drops <- c("NAICS_CODE.y")
clean_data <- newdata[ , !(names(newdata) %in% drops)]

##save data to csv
write.csv(clean_data,'clean_data.csv')
```

Create new variable "annual wage": standardized wage

```
H1B <- read.csv('H-1B_Disclosure_Data_FY2018_Q4_EOY.csv')

#convert variable "WAGE_RATE_OF_PAY_FROM" from factor to numeric
class(H1B$WAGE_RATE_OF_PAY_FROM)
H1B$WAGE_RATE_OF_PAY_FROM <- as.numeric(gsub(",","",H1B$WAGE_RATE_OF_PAY_FROM))
H1B$WAGE_RATE_OF_PAY_FROM <- as.numeric(as.character(H1B$WAGE_RATE_OF_PAY_FROM))

#Standardize wage to annual wage
hour <- H1B %>% filter(WAGE_UNIT_OF_PAY == "Hour" & WAGE_RATE_OF_PAY_FROM<1000)%>%
  mutate(annual_wage = WAGE_RATE_OF_PAY_FROM*1780)
hour2 <- H1B %>% filter(WAGE_UNIT_OF_PAY == "Hour" & WAGE_RATE_OF_PAY_FROM>10000)%>%
  mutate(annual_wage = WAGE_RATE_OF_PAY_FROM)#entry mistake: enter annual wage as hour wage
hour3 <- H1B %>% filter(WAGE_UNIT_OF_PAY == "Hour" & WAGE_RATE_OF_PAY_FROM==5850.00)%>%
  mutate(annual_wage = WAGE_RATE_OF_PAY_FROM*12) #entry mistake: enter month wage as hour wage

week <- H1B %>% filter(WAGE_UNIT_OF_PAY == "Week"& WAGE_RATE_OF_PAY_FROM<70000) %>%
  mutate(annual_wage= WAGE_RATE_OF_PAY_FROM*51)
week2 <- H1B %>% filter(WAGE_UNIT_OF_PAY == "Week"& WAGE_RATE_OF_PAY_FROM>70000) %>%
  mutate(annual_wage= WAGE_RATE_OF_PAY_FROM) #entry mistake: enter annual wage as week wage

bi_weekly <- H1B %>%  filter(WAGE_UNIT_OF_PAY == "Bi-Weekly")%>% mutate(annual_wage= WAGE_RATE_OF_PAY_F
month <- H1B %>%  filter(WAGE_UNIT_OF_PAY == "Month") %>%
  mutate(annual_wage =WAGE_RATE_OF_PAY_FROM*12)
year <- H1B %>% filter(WAGE_UNIT_OF_PAY == "Year")%>%
  mutate(annual_wage = WAGE_RATE_OF_PAY_FROM)
na <-H1B %>% filter(WAGE_UNIT_OF_PAY == "")%>%
  mutate(annual_wage = WAGE_RATE_OF_PAY_FROM)

merge1 <- merge(hour,week,all=TRUE)
merge2 <- merge(merge1, bi_weekly, all = TRUE)
merge3 <- merge(merge2, month, all = TRUE)
merge4 <- merge(merge3, year, all = TRUE)
merge5 <- merge(merge4, na, all = TRUE)
merge6 <- merge(merge5, hour2, all = TRUE)
merge7 <- merge(merge6, hour3, all = TRUE)
merge8 <- merge(merge7, week2, all = TRUE)
```

```
#create new dummy variable "Entry_Mistake", representing entry mistake of WAGE_RATE_OF_PAY_FROM
H1B_clean <- merge8 %>% mutate(Entry_Mistake = case_when(
  WAGE_UNIT_OF_PAY == "Hour" & WAGE_RATE_OF_PAY_FROM>1000 ~ TRUE,
  WAGE_UNIT_OF_PAY == "Week"& WAGE_RATE_OF_PAY_FROM>70000 ~ TRUE,
  WAGE_UNIT_OF_PAY == "Year"& WAGE_RATE_OF_PAY_FROM>100000000 ~ TRUE,
  WAGE_RATE_OF_PAY_FROM == 0 ~TRUE))

H1B_clean$Entry_Mistake[is.na(H1B_clean$Entry_Mistakes)]=FALSE
H1B_clean$Entry_Mistake <- H1B_clean$Entry_Mistake %>% replace_na(FALSE)

#convert date variable from character to date class
H1B_clean$CASE_SUBMITTED <- as.Date(H1B_clean$CASE_SUBMITTED, format = "%m/%d/%y")
H1B_clean$DECISION_DATE <- as.Date(H1B_clean$DECISION_DATE, format = "%m/%d/%y")
H1B_clean$EMPLOYMENT_START_DATE <- as.Date(H1B_clean$EMPLOYMENT_START_DATE, format = "%m/%d/%y")
H1B_clean$EMPLOYMENT_END_DATE <- as.Date(H1B_clean$EMPLOYMENT_END_DATE, format = "%m/%d/%y")

#create new variable "Wait_Time", representing wait time between decision date and submitted date
H1B_clean$Wait_Time <- H1B_clean$DECISION_DATE-H1B_clean$CASE_SUBMITTED
H1B_clean$Wait_Time <-as.numeric(as.character(H1B_clean$Wait_Time))

#rename columns
colnames(H1B_clean)[colnames(H1B_clean)=="annual_wage"] <- "ANNUAL_WAGE"
colnames(H1B_clean)[colnames(H1B_clean)=="Wait_Time"] <- "WAIT_TIME"
colnames(H1B_clean)[colnames(H1B_clean)=="Entry_Mistake"] <- "ENTRY_MISTAKE"

write.csv(H1B_clean, "H1B_annualwage.csv")
```

Merge two cleaned data, create new variable "Occupational Classification" and "Major Industry"

```
library(tidyverse)
#merge data
df1 <- read.csv("H1B_annualwage.csv")
df2 <- read.csv('clean_data 2.csv')
mergedata <- merge(x = df1, y = df2, by = "CASE_NUMBER", all = TRUE)

selected_variable <- mergedata %>% select(CASE_NUMBER, CASE_STATUS.x, ANNUAL_WAGE, WAGE_RATE_OF_PAY_FRO

colnames(selected_variable)[colnames(selected_variable)=="SOC_CODE_major"] <- "MAJOR_SOC_CODE"
colnames(selected_variable)[colnames(selected_variable)=="description"] <- "MAJOR_INDUSTRY"
colnames(selected_variable)[colnames(selected_variable)=="sector"] <- "MAJOR_NAICS_CODE"

colnames(selected_variable)[colnames(selected_variable)=="CASE_STATUS.x"] <- "CASE_STATUS"
colnames(selected_variable)[colnames(selected_variable)=="WAGE_RATE_OF_PAY_FROM.x"] <- "WAGE_RATE_OF_PA
colnames(selected_variable)[colnames(selected_variable)=="WAGE_UNIT_OF_PAY.x"] <- "WAGE_UNIT_OF_PAY"
colnames(selected_variable)[colnames(selected_variable)=="JOB_TITLE.x"] <- "JOB_TITLE"
colnames(selected_variable)[colnames(selected_variable)=="CASE_SUBMITTED.x"] <- "CASE_SUBMITTED"
colnames(selected_variable)[colnames(selected_variable)=="DECISION_DATE.x"] <- "DECISION_DATE"
colnames(selected_variable)[colnames(selected_variable)=="EMPLOYER_CITY.x"] <- "EMPLOYER_CITY"
colnames(selected_variable)[colnames(selected_variable)=="EMPLOYER_STATE.x"] <- "EMPLOYER_STATE"
colnames(selected_variable)[colnames(selected_variable)=="EMPLOYER_POSTAL_CODE.x"] <- "EMPLOYER_POSTAL_
colnames(selected_variable)[colnames(selected_variable)=="EMPLOYER_COUNTRY.x"] <- "EMPLOYER_COUNTRY"
colnames(selected_variable)[colnames(selected_variable)=="SOC_CODE.x"] <- "SOC_CODE"
colnames(selected_variable)[colnames(selected_variable)=="SOC_NAME.x"] <- "SOC_NAME"
colnames(selected_variable)[colnames(selected_variable)=="PREVAILING_WAGE.x"] <- "PREVAILING_WAGE"
colnames(selected_variable)[colnames(selected_variable)=="PW_UNIT_OF_PAY.x"] <- "PW_UNIT_OF_PAY"
colnames(selected_variable)[colnames(selected_variable)=="NAICS_CODE.x"] <- "NAICS_CODE"
colnames(selected_variable)[colnames(selected_variable)=="EMPLOYER_NAME.x"] <- "EMPLOYER_NAME"
colnames(selected_variable)[colnames(selected_variable)=="H1B_DEPENDENT.x"] <- "H1B_DEPENDENT"

summary(factor(selected_variable$MAJOR_SOC_CODE))

#create new variable "OCCUPATIONAL_CLASSIFICATION" by variable "MAJOR_SOC_CODE"
selected_variable2 <- selected_variable %>%
  mutate(OCCUPATIONAL_CLASSIFICATION = case_when(
    MAJOR_SOC_CODE == "11" ~ "Management",
    MAJOR_SOC_CODE == "13" ~ "Business and Financial Operations",
    MAJOR_SOC_CODE == "15" ~ "Computer and Mathematical",
    MAJOR_SOC_CODE == "17" ~ "Architecture and Engineering",
    MAJOR_SOC_CODE == "19" ~ "Life, Physical, and Social Science",
    MAJOR_SOC_CODE == "21" ~ "Community and Social Service",
    MAJOR_SOC_CODE == "23" ~ "Legal Occupations",
    MAJOR_SOC_CODE == "25" ~ "Educational, Traning, and Library",
    MAJOR_SOC_CODE == "27" ~ "Arts, Design, Entertainment, Sports, and Media",
    MAJOR_SOC_CODE == "29" ~ "Healthcare Practitioners and Technical",
    MAJOR_SOC_CODE == "31" ~ "Healthcare Support",
    MAJOR_SOC_CODE == "33" ~ "Protective Service",
    MAJOR_SOC_CODE == "35" ~ "Food Preparation and Serving Related ",
    MAJOR_SOC_CODE == "37" ~ "Building and Grounds Cleaning and Maintenance",
    MAJOR_SOC_CODE == "39" ~ "Personal Care and Service",
    MAJOR_SOC_CODE == "40" ~ "Sales and Related",
    MAJOR_SOC_CODE == "41" ~ "Sales and Related",
    MAJOR_SOC_CODE == "43" ~ "Office and Administrative Support",
    MAJOR_SOC_CODE == "45" ~ "Farming, Fishing, and Forestry",
    MAJOR_SOC_CODE == "47" ~ "Construction and Extraction",
    MAJOR_SOC_CODE == "49" ~ "Installation, Maintenance, and Repair",
    MAJOR_SOC_CODE == "51" ~ "Production",
    MAJOR_SOC_CODE == "53" ~ "Transportation and Material Moving"
    ))

#check missing data pattern
H1B$EMPLOYER_STATE[H1B$EMPLOYER_STATE==""]=NA
H1B$H1B_DEPENDENT[H1B$H1B_DEPENDENT==""]=NA
H1B$MAJOR_INDUSTRY[H1B$MAJOR_INDUSTRY==0]=NA
H1B$MAJOR_NAICS_CODE[H1B$MAJOR_NAICS_CODE==0]=NA
H1B$EMPLOYER_COUNTRY[H1B$EMPLOYER_COUNTRY==""]=NA
H1B$WAGE_UNIT_OF_PAY[H1B$WAGE_UNIT_OF_PAY==""]=NA

H1B$MAJOR_INDUSTRY <- as.character(H1B$MAJOR_INDUSTRY)
H1B$MAJOR_INDUSTRY[H1B$MAJOR_INDUSTRY==
                     "Administrative and Support and Waste Management and Remediation Services"] <-
  "Administrative, Support, Waste Management, Remediation"
H1B$MAJOR_INDUSTRY[H1B$MAJOR_INDUSTRY==
                     "Agriculture, Forestry, Fishing and Hunting (not covered in economic census)"] <-
  "Agriculture, Forestry, Fishing and Hunting"
H1B$MAJOR_INDUSTRY[H1B$MAJOR_INDUSTRY==
                     "Public Administration (not covered in economic census)"] <-
  "Public Administration"

write.csv(selected_variable2, "H1B_26variable.csv")
```

Create Variable "Lower__Than__PW": True if the applicant's wage is smaller than prevailing wage

```
library(tidyverse)
library(ggpubr)

#load data
H1B <- read.csv('H1B_26variable.csv')
class(H1B$PW_UNIT_OF_PAY)
summary(as.factor(H1B$PW_UNIT_OF_PAY))

temp4 <- H1B %>% mutate(Lower_Than_PW = case_when(WAGE_RATE_OF_PAY_FROM >=PREVAILING_WAGE ~ FALSE,
                                                      WAGE_RATE_OF_PAY_FROM<PREVAILING_WAGE ~ TRUE))

yearpw <- temp4 %>% filter(WAGE_UNIT_OF_PAY =="Year" & PW_UNIT_OF_PAY=="Year")
hourpw <- temp4 %>% filter(WAGE_UNIT_OF_PAY =="Hour" & PW_UNIT_OF_PAY=="Hour")
weekpw <- temp4 %>% filter(WAGE_UNIT_OF_PAY =="Week" & PW_UNIT_OF_PAY=="Week")
biweekpw <- temp4 %>% filter(WAGE_UNIT_OF_PAY =="Bi-Weekly" & PW_UNIT_OF_PAY=="Bi-Weekly")
monthpw <- temp4 %>% filter(WAGE_UNIT_OF_PAY =="Month" & PW_UNIT_OF_PAY=="Month")

pwmerge1 <- merge(yearpw,hourpw,all=TRUE)
pwmerge2 <- merge(pwmerge1, weekpw, all = TRUE)
pwmerge3 <- merge(pwmerge2, biweekpw, all = TRUE)
pwmerge4 <- merge(pwmerge3, monthpw, all = TRUE)



temptemp <- pwmerge4 %>% filter(Lower_Than_PW==TRUE)
summary(as.factor(temptemp$CASE_STATUS))

write.csv(pwmerge4, "H1B_28variable.csv")
```

Drop outliers for annual wage

```
#load data
H1B <- read.csv('H1B_26variable.csv')

#Drop outliers of annual wages
upper <- summary(H1B$ANNUAL_WAGE)[5]+
  (summary(H1B$ANNUAL_WAGE)[5]-summary(H1B$ANNUAL_WAGE)[2])*1.5
H1B_clean <- H1B %>% filter(H1B$ANNUAL_WAGE<=upper & H1B$ANNUAL_WAGE>0)
```