Statistical Learning Project

Anna Badalyan, Rebecca Di Francesco

21 06 2022

```
knitr::opts_chunk$set
```

```
## function (...)
## {
##     set2(resolve(...))
## }
## <bytecode: 0x0000000012e5b498>
## <environment: 0x0000000012e783e0>
```

Introduction

The project is focused on finding the determinants of income for the population of US households in the period 1999-2013. The Current Population Survey dataset[1], which contains labor force statistics in the US, was used. The dataset was pre-cleaned to contain only the observations for working adults aged 25 to 64.

The aim of the project is to identify the factors that influence the income and build a model able to predict the income based on census data. The Linear Regression was build to predict the value of income and the Logistic Regression was used to predict if the income exceeds 60,000 US dollars. We also investigated if there are differences in income between males and females and if these differences depend on the marital status and occupation.

The first part is dedicated to data wrangling which includes data cleaning and variable preparation. The second part covers exploratory data analysis, where we visualize the data and identify the relationships between the parameters. Then, by using statistical techniques such as ANOVA test, CHi-Squared test, Linear Regression and Logistic regression we answer out research questions. In the last part, the results are discussed.

Loading the libraries

```
library(ggplot2)
```

```
## Warning:     'ggplot2'          R     4.1.3
```

```
library(leaps)
```

```
## Warning:     'leaps'          R     4.1.3
```

```
library(forcats)
```

```
## Warning:     'forcats'          R     4.1.3
```

```
library(pROC)
```

```
## Warning:     'pROC'          R     4.1.3
```

```
library(MASS)
library(class)
library(car)
```

```
## Warning:     'car'           R     4.1.3
```

```
## Warning:     'carData'           R     4.1.3
```

```
library(glmnet)
```

```
## Warning:     'glmnet'           R     4.1.3
```

```
curPop <- read.csv("C:/Users/Anna/Desktop/CurrentPopulationSurvey.csv")
dim(curPop)
```

```
## [1] 344287    234
```

Data wrangling

Data cleaning

The dataset contains 344287 observations and 234 variables. The variables with prefix o_ are the original values provided by the US Census Bureau, while the respective variables without the prefix were cleaned by the providers of the dataset. In addition, the dataset providers generated additional variables based on the original ones.

For convenience, we compare the original and cleaned variables 5 at a time.

```
o_data <- curPop[,grep("o_", names(curPop), value=TRUE)] #dataframe of columns starting from "o_"
columns_containing_NAs <- names(which(sapply(o_data, function(x) any(is.na(x)))))

summary(curPop[, columns_containing_NAs][1:5])
```

```
##     o_county         o_farm            o_bpl          o_yrimmig
##   Min.   :    0   Min.   :1.00     Min.   : 9900   Min.   :   0
##   1st Qu.:    0   1st Qu.:1.00     1st Qu.: 9900   1st Qu.:   0
##   Median :    0   Median :1.00     Median : 9900   Median :   0
##   Mean   :10057   Mean   :1.01     Mean   :14168   Mean   : 358
##   3rd Qu.:13057   3rd Qu.:1.00     3rd Qu.: 9900   3rd Qu.:   0
##   Max.   :55139   Max.   :2.00     Max.   :96000   Max.   :2013
##   NA's   :87412   NA's   :256875   NA's   :87412   NA's   :87412
##    o_citizen
##   Min.   :0.00
##   1st Qu.:0.00
##   Median :0.00
##   Mean   :0.43
##   3rd Qu.:0.00
##   Max.   :3.00
##   NA's   :87412
```

```
new_variables <- gsub("o_",  "\\1",columns_containing_NAs[1:5])
summary(curPop[, new_variables])
```

```
##      county             farm             bpl           yrimmig
##   Min.   : 1003    Min.   :1.00      Min.   : 9900   Min.   :1949
##   1st Qu.: 8059    1st Qu.:1.00      1st Qu.: 9900   1st Qu.:1981
##   Median :22019    Median :1.00      Median : 9900   Median :1991
##  Mean   :23730    Mean   :1.01     Mean   :14082   Mean   :1990  
##  3rd Qu.:36103    3rd Qu.:1.00     3rd Qu.: 9900   3rd Qu.:1999  
##  Max.   :55139    Max.   :2.00     Max.   :72000   Max.   :2013  
##  NA's   :235427   NA's   :256875   NA's   :87681   NA's   :298083  
##     citizen     
##  Min.   :1.00   
##  1st Qu.:2.00   
##  Median :3.00   
##  Mean   :2.48   
##  3rd Qu.:3.00   
##  Max.   :3.00   
##  NA's   :299748
```

We can see that cleaned variables have more null values than in the original dataset null values were encoded as 0. For example, the variable *o_county* identifies the county code which is a numerical code that cannot be zero and the variable *county* encoded all the zeros as NA's. Thus, we will use the clean version of the variables.

However, this is not the case for the variable *o_yrimmig*.

```r
sum(curPop$o_yrimmig==0, na.rm=TRUE)/dim(curPop)[1]
```

```
## [1] 0.6119052
```

```r
sum(curPop$o_yrimmig==0, na.rm=TRUE)
```

```
## [1] 210671
```

We can see, that the total number of observations where the year of immigration is encoded as 0 is 210671 or 61%. This most probably represents the people who didn't immigrate to the US.

```r
length(unique(curPop$occ))
```

```
## [1] 1259
```

```r
length(unique(curPop$ind))
```

```
## [1] 699
```

The variables *occ* and *ind* contain 1259 and 699 unique values respectively. The models built using these variables will have to create 1259 and 699 dummy variables, which would be difficult to interpret. Thus, we will use the grouped occupation and industry related columns. They are already available as dummy variables and we will use them to reconstruct *industry* and *occupation*.

```r
curPop$industry[curPop$Agriculture == 1] <- 'Agriculture'
curPop$industry[curPop$miningconstruction == 1] <- 'MiningConstruction'
curPop$industry[curPop$durables == 1] <- 'Durables'
curPop$industry[curPop$nondurables == 1] <- 'Nondurables'
curPop$industry[curPop$Transport == 1] <- 'Transport'
curPop$industry[curPop$Utilities == 1] <- 'Utilities'
curPop$industry[curPop$Communications == 1] <- 'Communications'
curPop$industry[curPop$retailtrade == 1] <- 'RetailTrade'
curPop$industry[curPop$wholesaletrade == 1] <- 'WholesaleTrade'
curPop$industry[curPop$finance == 1] <- 'Finance'
curPop$industry[curPop$SocArtOther == 1] <- 'SocArtOther'
curPop$industry[curPop$hotelsrestaurants == 1] <- 'HotelsRestaurants'
curPop$industry[curPop$Medical == 1] <- 'Medical'
curPop$industry[curPop$Education == 1] <- 'Education'
curPop$industry[curPop$professional == 1] <- 'Professional'
```

```
curPop$industry[curPop$publicadmin == 1] <- 'Publicadmin'

curPop$occupation[curPop$manager == 1] <- 'manager'
curPop$occupation[curPop$business == 1] <- 'business'
curPop$occupation[curPop$financialop == 1] <- 'financialop'
curPop$occupation[curPop$computer == 1] <- 'computer'
curPop$occupation[curPop$architect == 1] <- 'architect'
curPop$occupation[curPop$scientist == 1] <- 'scientist'
curPop$occupation[curPop$socialworker == 1] <- 'socialworker'
curPop$occupation[curPop$postseceduc == 1] <- 'postseceduc'
curPop$occupation[curPop$legaleduc == 1] <- 'legaleduc'
curPop$occupation[curPop$artist == 1] <- 'artist'
curPop$occupation[curPop$lawyerphysician == 1] <- 'lawyerphysician'
curPop$occupation[curPop$healthcare == 1] <- 'healthcare'
curPop$occupation[curPop$healthsupport == 1] <- 'healthsupport'
curPop$occupation[curPop$protective == 1] <- 'protective'
curPop$occupation[curPop$foodcare == 1] <- 'foodcare'
curPop$occupation[curPop$building == 1] <- 'building'
curPop$occupation[curPop$sales == 1] <- 'sales'
curPop$occupation[curPop$officeadmin == 1] <- 'officeadmin'
curPop$occupation[curPop$farmer == 1] <- 'farmer'
curPop$occupation[curPop$constructextractinstall == 1] <- 'constructextractinstall'
curPop$occupation[curPop$production == 1] <- 'production'
curPop$occupation[curPop$transport == 1] <- 'transport'
```

Levels for industry:

```
unique(curPop$industry)
```

```
##  [1] "SocArtOther"        "HotelsRestaurants"  "Durables"
##  [4] "Professional"       "Publicadmin"        "Transport"
##  [7] "Medical"            "RetailTrade"        "WholesaleTrade"
## [10] "Education"          "Nondurables"        "MiningConstruction"
## [13] "Finance"            "Communications"     "Utilities"
## [16] "Agriculture"
```

Levels for occupation:

```
unique(curPop$occupation)
```

```
##  [1] "officeadmin"             "architect"
##  [3] "computer"                "manager"
##  [5] "protective"              "production"
##  [7] "sales"                   "transport"
##  [9] "constructextractinstall" "socialworker"
## [11] "postseceduc"             "healthcare"
## [13] "scientist"               "building"
## [15] "foodcare"                "financialop"
## [17] "legaleduc"               "lawyerphysician"
## [19] "business"                "farmer"
## [21] "healthsupport"           "artist"
```

We can see, then newly generated values don't contain any null values.

We can also notice that the variables gq, month, popstat, labforce, incbus, incfarm aren't informative as they contain the same value (their min, max and mean are the same), so we can drop them.

```
summary(curPop[c("gq", "month", "popstat", "labforce", "incbus", "incfarm")])
```

```
##        gq         month      popstat     labforce     incbus      incfarm
##  Min.   :1   Min.   :3   Min.   :1   Min.   :2   Min.   :0   Min.   :0
##  1st Qu.:1   1st Qu.:3   1st Qu.:1   1st Qu.:2   1st Qu.:0   1st Qu.:0
##  Median :1   Median :3   Median :1   Median :2   Median :0   Median :0
##  Mean   :1   Mean   :3   Mean   :1   Mean   :2   Mean   :0   Mean   :0
##  3rd Qu.:1   3rd Qu.:3   3rd Qu.:1   3rd Qu.:2   3rd Qu.:0   3rd Qu.:0
##  Max.   :1   Max.   :3   Max.   :1   Max.   :2   Max.   :0   Max.   :0
```

We can see that there are several variables for income, which seem identical, *incwage*, *niincwage*, *incwageman*. Let's check if they are identical:

```
sum(curPop$incwage == curPop$niincwage) == sum(curPop$incwage == curPop$incwageman)
```

```
## [1] TRUE
```

We verified, that the variables are identical, so we will use the *incwage* column.

In the dataset there are three variables starting with "tc" (i.e. topcoded) namely *tcoincwage*, *tcinclongj* and *tcincwage*. These variables were created to eliminate outliers from the corresponding original variables, i.e. *incwage*, *inclongj* and *incwageup*. We will not use the topcoded variables because the precious information could be lost and such outliers are peculiar to income distributions. Moreover, we will not use the variable *oincwage* as it corresponds to the earnings from other work including wage and salary, which is already present in *incwage*.

The variable *inclongj* describes the earnings from the longest job, thus it cannot be included as a predictor as it would coincide with *incwage*. Therefore, *incwage* is our final choice of income variable that we want to predict. We will not consider the column *hrwage* because it is the hourly wage that was calculated using *incwage* divided by the total hours worked.

Summary of *oincwage*

```
summary(curPop$oincwage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0       0       0    1010       0 1099999   42379
```

Summary of *tcoincwage*"

```
summary(curPop$tcoincwage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0     0.0     0.0   953.1     0.0 72500.0   42379
```

Summary of *incwage*

```
summary(curPop$incwage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      15   16700   30000   39762   50000 1259999
```

Summary of *tincwage*

```
summary(curPop$tcincwage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      15   16700   30000   39049   50000  435000
```

Summary of *inclongj*

```
summary(curPop$inclongj)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   19000   32000   42198   51000 1099999   42379
```

Summary of *tcinglongj*

```
summary(curPop$tcinclongj)
```

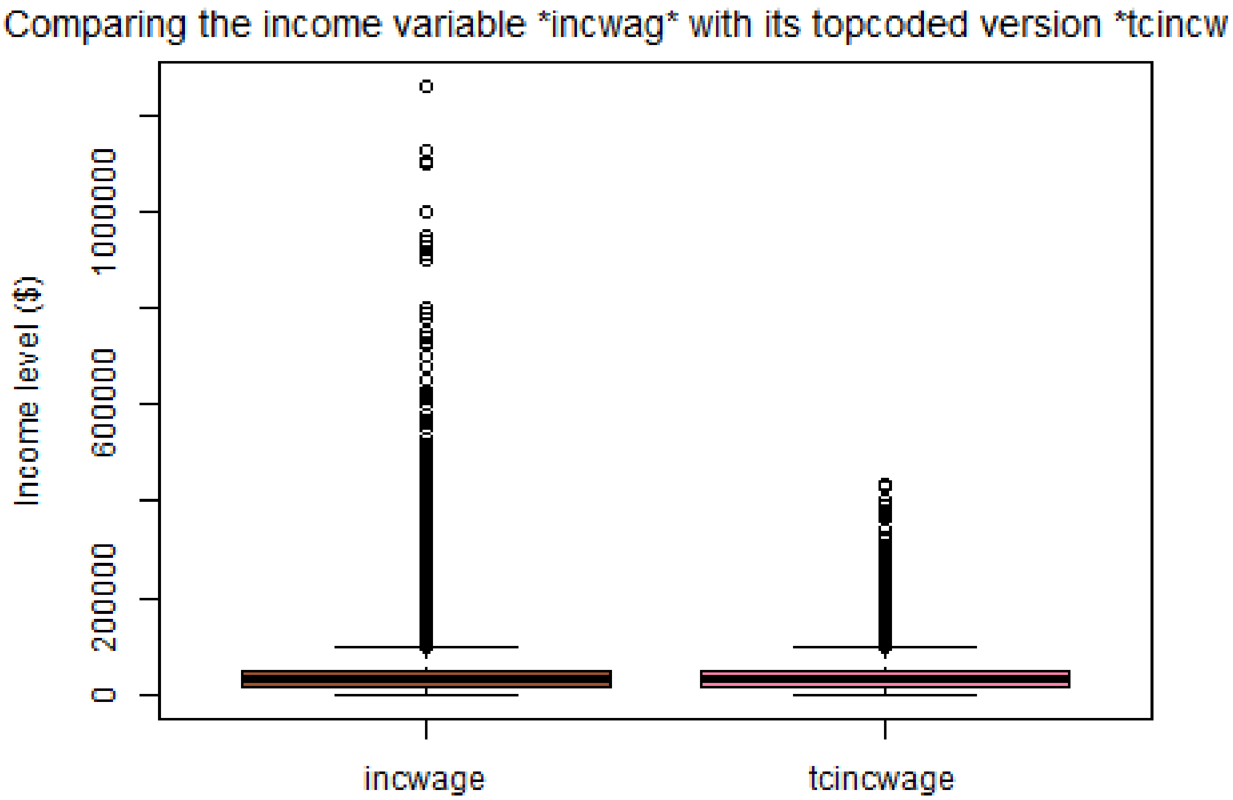
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   19000   32000   41393   51000  362500   42379
```

Percentage of entries of inclongj that are equal to incwage:

```
sum(curPop$incwage==curPop$inclongj, na.rm=TRUE)/dim(curPop)[1]
```

```
## [1] 0.7775257
```

```
par(mar=c(3.1, 4.7, 2.3, 2))
boxplot(curPop$incwage, curPop$tcincwage, names = c("incwage", "tcincwage"),col=c( "sienna", "paleviole
mtext(side=2, text="Income level ($)", line=3)
mtext(side=3, text="Comparing the income variable *incwag* with its topcoded version *tcincwage*", line=
```



There is a variable *srcearn*, which represents the source of earnings from the longest job and has two categories (1=wage and salary; 4=without pay). However, only 47 observations fall into the category 4 thus we can skip this feature as well. Summary of *srcearn*:

```
summary(as.factor(curPop$srcearn))
```

```
##      1      4   NA's
## 301861     47  42379
```

There are variables starting with “q”, which are data quality flags. Let’s consider the quality flags for the variables selected for analysis.

```
sum(curPop$quhrswor>0, na.rm=TRUE)
```

```
## [1] 1728
```

```
sum(curPop$qwkswork>0, na.rm=TRUE)
```

```
## [1] 946
```

```
sum(curPop$qincwage>0, na.rm=TRUE)
```

```
## [1] 0
```

There are some issues with the variables *uhrswor* and *wkswork*. However, the number of observations with issues is small compared to the total (344,287), so the columns were kept.

Regarding the education related variables, there are *sch*, *educ99* and *schlcoll*. The first two variables indicate educational attainment but the variable *educ99* only recorded responses from the year ’99 onwards, so *sch* is the complete version. While *schlcoll* can also be removed because it informs about school or college attendance for the year 2013 only.

We also do not consider the variabls *occly*, *indly* and *classwly* because they refer to previous year occupation, industry and class of worker and will be largely equal to the base year variables.

Then there are some variables related to the place of birth both of the respondents (*bpl*) and their parents (*mbpl*, *fbpl*). The birthplaces contain 169 unique values, so *nativity*, which is a birthplace with only 5 unique values, is a better choice. Number of levels of *bpl*, birthplace:

```
length(unique(as.factor(curPop$bpl)))-1
```

```
## [1] 169
```

Number of levels of *o_nativity*:

```
length(unique(as.factor(curPop$o_nativity)))-1
```

```
## [1] 6
```

There are also two variables that are closely related: *o_yrimmig* and *o_citizen*. The former is the year of immigration and the latter is the citizenship status.

```
summary(as.factor(curPop$o_yrimmig))
```

```
##      0   1949   1959   1964   1969   1974   1979   1981   1983   1985   1987 
## 210671     81    851   1052   1600   2677   3692   2504   1720   2283   2196 
##   1989   1991   1993   1995   1997   1999   2001   2003   2005   2007   2009 
##   2830   2863   2306   2656   2362   3340   3469   2216   1701   1710    593 
##   2010   2011   2013   NA's 
##    442    538    522  87412
```

```
summary(as.factor(curPop$o_citizen))
```

```
##      0      1      2      3   NA's 
## 212336   2480  17979  24080  87412
```

The variable *o_yrimmig* contains many zeros that are probably related to people that never immigrated to the US. So we decided to encode this variable differently:

```
diff <- curPop$year- curPop$o_yrimmig

diff[diff<=5] <- 1
diff[diff>5&diff<=10] <-2
diff[diff>10&diff<=20]<-3
diff[diff>20&diff<1999]<-4
diff[diff>=1999] <- 0
curPop$immig_year <- as.factor(diff)
summary(curPop$immig_year)
```

```
##      0      1      2      3      4   NA's
## 210671   5057   9244  14096  17807  87412
```

0 = never immigrated,1=less than 5y ago, 2 = less than 10y & more than 5 year ago, 3 = less than 20y ago $ more than 10, 4 = immigrated more than 20yago

```
data <- curPop[c("year", "numprec", "region", "statefip", "metro", "metarea", "county","relate", "age",
```

Finally, we consider the null values.

```
colSums(is.na(data))
```

```
##       year    numprec     region   statefip      metro    metarea     county
##          0          0          0          0       9759     103939     235427
##     relate        age        sex       race      marst immig_year  o_citizen
##          0          0          0          0          0      87412      87412
##   nativity        sch    empstat occupation   industry    classwkr   wkswork1
##      87824          0          0          0          0          0          0
##    hrswork   uhrswork      union      ftype    inflate    incwage
##      10555          0      42379          0          0          0
```

We can see that the columns *metarea* and *county* are missing 103939 and 235427 observations respectively, so they won't be used for further analysis.

```
data$metarea <- NULL
data$county <- NULL
```

Columns *o_nativity*, *immig_year* and *o_citizen* interestingly contain around 87,000 NA's. Thus we check if these missing values are in the same rows, if so there may be another reason of the missing data which is not simply random. As we can see, all the NAs are in the same rows:

```
sum(is.na(curPop$immig_year)==is.na(curPop$o_citizen))
```

```
## [1] 344287
```

```
sum(is.na(curPop$immig_year)==is.na(curPop$o_nativity))
```

```
## [1] 344287
```

```
sum(is.na(curPop$o_citizen)==is.na(curPop$o_nativity))
```

```
## [1] 344287
```

We discover that all the NAs are for the year 1990 and 1981.

```
unique(curPop[is.na(curPop$o_citizen),"year"])
```

```
## [1] 1990 1981
```

```
unique(curPop[is.na(curPop$immig_year),"year"])
```

```
## [1] 1990 1981
```

```
unique(curPop[is.na(curPop$o_nativity),"year"])
```

```
## [1] 1990 1981
```

Given this new information, the null values were removed.

```
data <- na.omit(data)
```

Most of the variables in the dataset are categorical, but R reads them as numbers. We will need to represent them as factors for further modeling.

```
col.list <-c("region", "statefip", "metro","relate", "sex", "race", "marst", "immig_year", "o_citizen",

for (col in col.list) {
  data[[col]] <- as.factor(data[[col]])
}

summary(data)
```

```
##       year         numprec           region         statefip      metro
##  Min.   :1999   Min.   : 1.000   31     :43522   6      : 22873   1:46464
##  1st Qu.:2007   1st Qu.: 2.000   42     :37132   48     : 14693   2:60425
##  Median :2009   Median : 3.000   21     :29267   36     : 10338   3:97104
##  Mean   :2008   Mean   : 3.228   22     :29146   12     :  9967   4:43309
##  3rd Qu.:2011   3rd Qu.: 4.000   41     :26084   17     :  7888
##  Max.   :2013   Max.   :16.000   11     :25024   42     :  7758
##                                  (Other):57127   (Other):173785
##      relate            age          sex         race       marst      immig_year
##  101    :138495   Min.   :25.00   1:124972   1:165262   1:161203   0:202718
##  201    : 79118   1st Qu.:34.00   2:122330   2: 24735   2:  3491   1:  4875
##  301    :  8505   Median :42.00              3: 39329   3:  6094   2:  8929
##  1114   :  7822   Mean   :42.35              4: 17976   4: 28776   3: 13636
##  1115   :  3484   3rd Qu.:50.00                         5:  3466   4: 17144
##  501    :  2912   Max.   :64.00                         6: 44272
##  (Other):  6966
##  o_citizen  nativity        sch        empstat
##  0:204345   1:186331   12     :71338   10:247302
##  1:  2360   2:  4242   16     :56001
##  2: 17307   3:  4315   13     :43089
##  3: 23290   4:  7819   18     :30736
##             5: 44595   14     :27423
##                        11     : 4130
##                        (Other):14585
##                   occupation               industry      classwkr
##  officeadmin            : 34728   Medical     : 30053   21:199985
##  manager                : 26462   Education   : 27239   25:  8954
##  sales                  : 21926   Professional: 24122   27: 14280
##  constructextractinstall: 21600   RetailTrade : 23974   28: 24023
##  production             : 17680   Durables    : 20083   29:    60
##  legaleduc              : 16776   SocArtOther : 18477
##  (Other)                :108130   (Other)     :103354
##     wkswork1        hrswork         uhrswork     union      ftype
##  Min.   : 1.00   Min.   : 1.00   Min.   : 1.00   0:204051   1:195353
##  1st Qu.:52.00   1st Qu.:38.00   1st Qu.:40.00   1: 36257   2: 33290
##  Median :52.00   Median :40.00   Median :40.00   2:  6371   3:  5403
##  Mean   :49.53   Mean   :40.09   Mean   :40.72   3:   623   4:   962
##  3rd Qu.:52.00   3rd Qu.:45.00   3rd Qu.:42.00              5: 12294
##  Max.   :52.00   Max.   :99.00   Max.   :99.00
##
##     inflate          incwage
##  Min.   :0.9589   Min.   :     25
##  1st Qu.:1.0000   1st Qu.:  22000
##  Median :1.0159   Median :  36000
##  Mean   :1.0502   Mean   :  46704
##  3rd Qu.:1.0731   3rd Qu.:  57000
##  Max.   :1.2717   Max.   :1259999
##
```

After cleaning, the variable *empstat* has only the observation of the category “At work”, thus we can remove this variable:

```
data$empstat <- NULL
```

Recoding variables

Let’s plot the *sch* column for education.

```
summary(data$sch)
```

```
##     0   2.5   5.5   7.5     9    10    11    12    13    14    16    18
##   481  1475  3526  2743  3100  3260  4130 71338 43089 27423 56001 30736
```

We can see that there are very few values for people that didn’t finish school, so we can group them to ‘nosc’ class. We tried grouping the variables by the school levels (elementary, middle, high), but the linear regression analysis showed that there was no significant difference in income between those groups and people who didn’t attend school at all. Thus, these levels were merged.

```
levels(data$sch) <- c(levels(data$sch),"nosc", "fsch", "scol", "asoc", "bach", "advd")
data$sch[data$sch == 0] <- 'nosc'
data$sch[data$sch == 1] <- 'nosc'
data$sch[data$sch == 2] <- 'nosc'
data$sch[data$sch == 2.5] <- 'nosc'
data$sch[data$sch == 3] <- 'nosc'
data$sch[data$sch == 4] <- 'nosc'
data$sch[data$sch == 5] <- 'nosc'
data$sch[data$sch == 5.5] <- 'nosc'
data$sch[data$sch == 6] <- 'nosc'
data$sch[data$sch == 7] <- 'nosc'
data$sch[data$sch == 7.5] <- 'nosc'
data$sch[data$sch == 8] <- 'nosc'
data$sch[data$sch == 9] <- 'nosc'
data$sch[data$sch == 10] <- 'nosc'
data$sch[data$sch == 11] <- 'nosc'
data$sch[data$sch == 12] <- 'fsch'
data$sch[data$sch == 13] <- 'scol'
data$sch[data$sch == 14] <- 'asoc'
data$sch[data$sch == 16] <- 'bach'
data$sch[data$sch == 18] <- 'advd'
data$sch <- droplevels(data$sch)
summary(data$sch)
```

```
##  nosc  fsch  scol  asoc  bach  advd
## 18715 71338 43089 27423 56001 30736
```

The class of workers variable is organised into 7 levels:(Self-empl=10, private sector=21, government=24, Federal govt employee=25, State govt employee=27, Local govt employee=28, Unpaid family worker=29). The majority of observation is in the category of private sector and then we have some observation for the category 25, 27, 28 that we grouped together into “Public sector”. Since unpaid family worker are only a small amount of units compared to all the rest we can combine them inside “Private sector” category as well.

```
summary(as.factor(data$classwkr))
```

```
##     21     25     27     28     29
## 199985   8954  14280  24023     60
```

```
data$classwkr <- gsub('25', 'Public sector',data$classwkr)
data$classwkr <- gsub('27', 'Public sector', data$classwkr)
data$classwkr <- gsub('28', 'Public sector', data$classwkr)

data$classwkr <- gsub('21', 'Private sector',data$classwkr)
data$classwkr <- gsub('29', 'Private sector', data$classwkr)

data$classwkr <- as.factor(data$classwkr)

summary(data$classwkr)
```

```
## Private sector  Public sector
##         200045          47257
```

The summary of *relate* variable shows that the class 1242 has 18 observations which correspond to foster child category. However, it is impossible that a person is a foster child at the age 25 or higher, which means that there was an error with this variable.

```
summary(data$relate)
```

```
##    101    201    301    501    701    901   1001   1114   1115   1241   1242
## 138495  79118   8505   2912   2179    209   2628   7822   3484    664     18
##   1260
##   1268
```

Thus, the observations with relate == 1242 were removed.

```
data <- subset(data, subset=relate != 1242)
data$relate <- droplevels(data$relate)
summary(data$relate)
```

```
##    101    201    301    501    701    901   1001   1114   1115   1241   1260
## 138495  79118   8505   2912   2179    209   2628   7822   3484    664   1268
```

Before visualizing the data, we need to calculate the real income by multiplying the inflation rate by *incwage*.

```
data$realincwage <- data$incwage*data$inflate
```

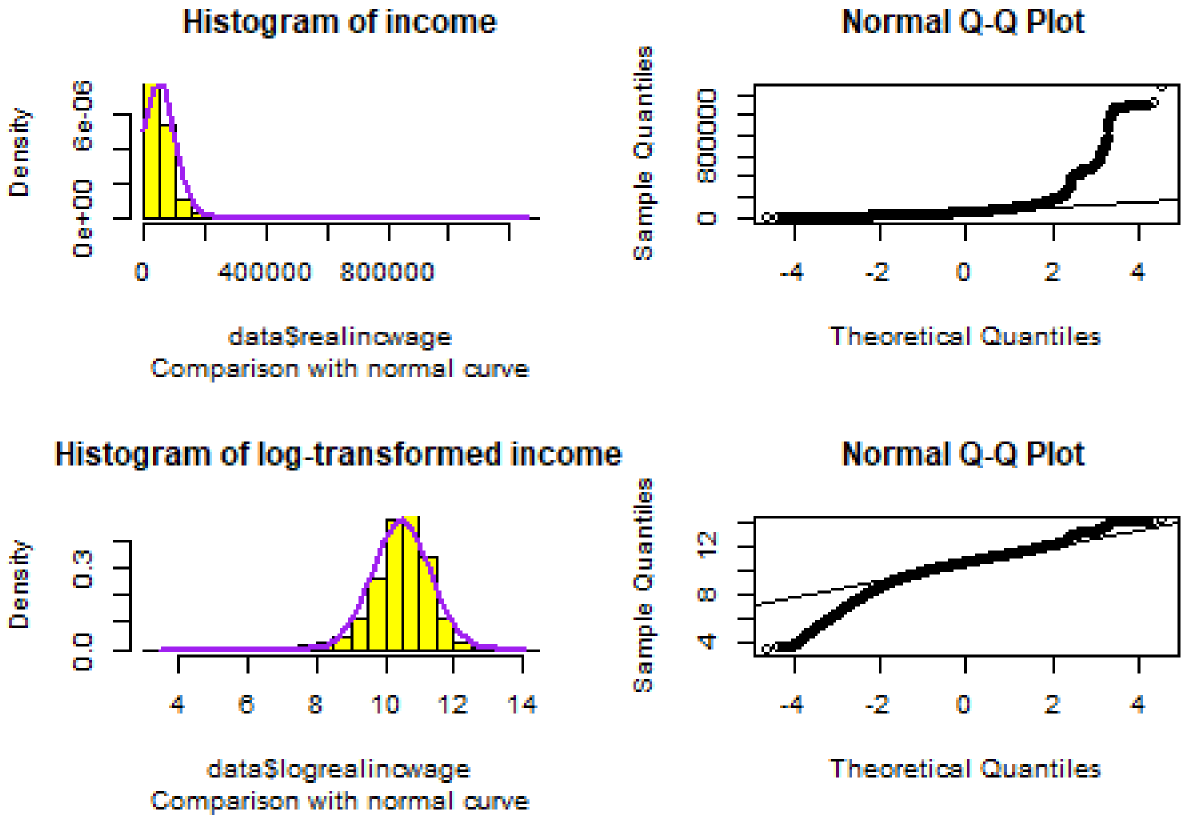
Let’s also create a binary income variable with 60,000 dollars as threshold, this variable will be used in order to perform a logistic regression.

```
data$binaryincome <- as.factor(ifelse(data$realincwage >=60000, 1, 0))
summary(data$binaryincome)
```

```
##      0      1
## 186250  61034
```

Exploratory Data Analysis

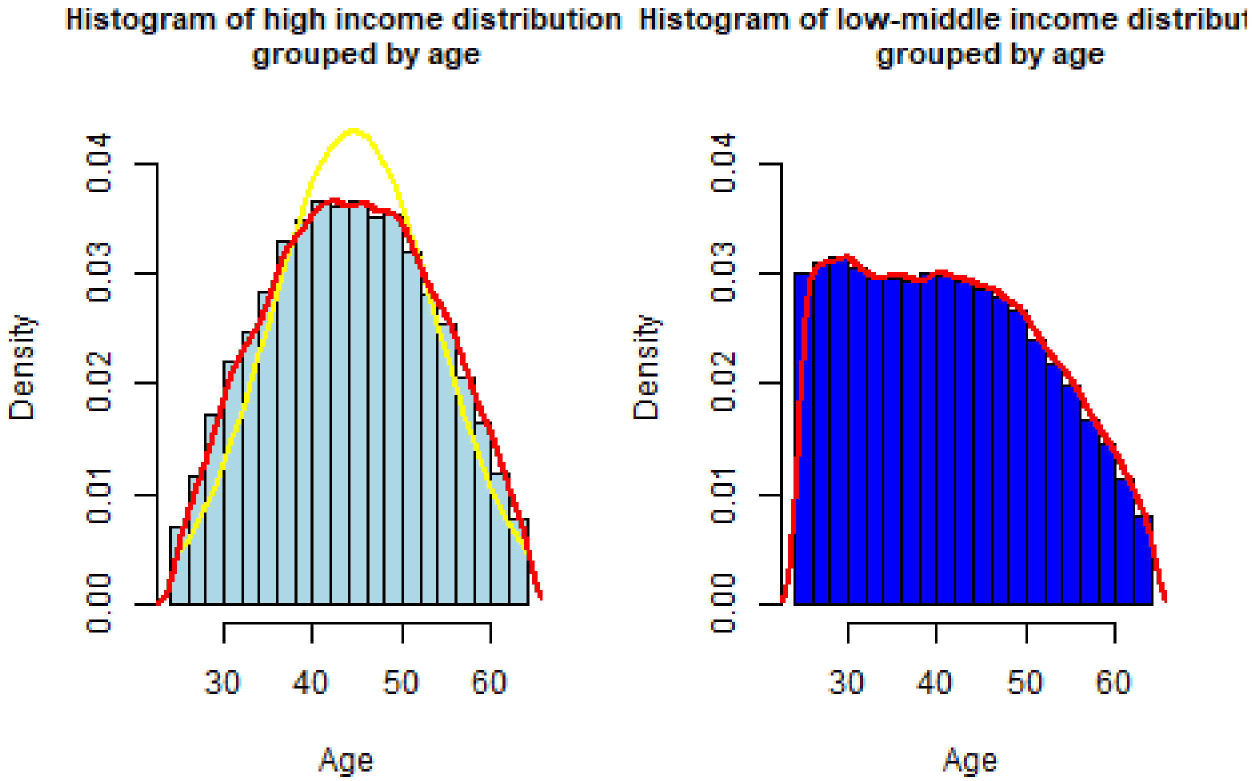
We can see that the distribution of *realincwage* is highly left skewed. Thus, we will apply the log transform.



The log transform of income is almost normally distributed apart from the long left tail, which is also visible in the Normal Q-Q Plot.

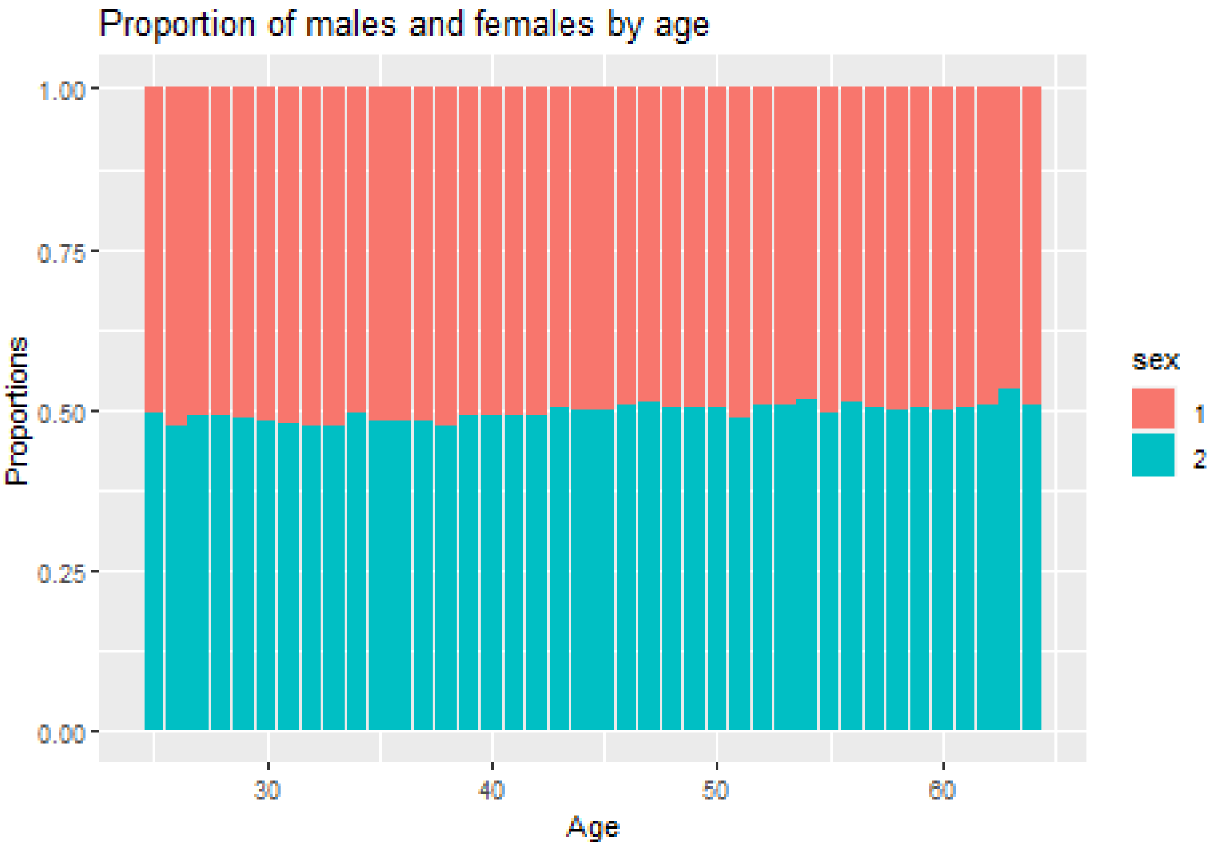
We have plotted the distributions of *year*, *region*, *statefip*, *metro*, *marst*, *nativity*, *union*, *wkswork1*, *uhrswork* variables and didn't notice any problems.

In the density histogram below we can see the different distributions of income (using *binaryincome*) across different age categories. On the left, we can see that the high income (over 60k) for different ages is distributed almost like a normal distribution. While the low-middle income distribution has a descending shape.



Now, let's check if the age variable grouped by sex is balanced. We want to avoid imbalance because as we noticed above young people tend to have lower income as opposite of older people. Thus if we had more younger males than younger females or viceversa this would bias our analysis. Luckily, it seems that we have a balanced number of males and females for each year of age.

```
# Stacked + percent
ggplot(data, aes(fill=sex, y=age, x=age)) +
    geom_bar(position="fill", stat="identity") +
    labs(title  = "Proportion of males and females by age", x ="Age", y="Proportions")
```

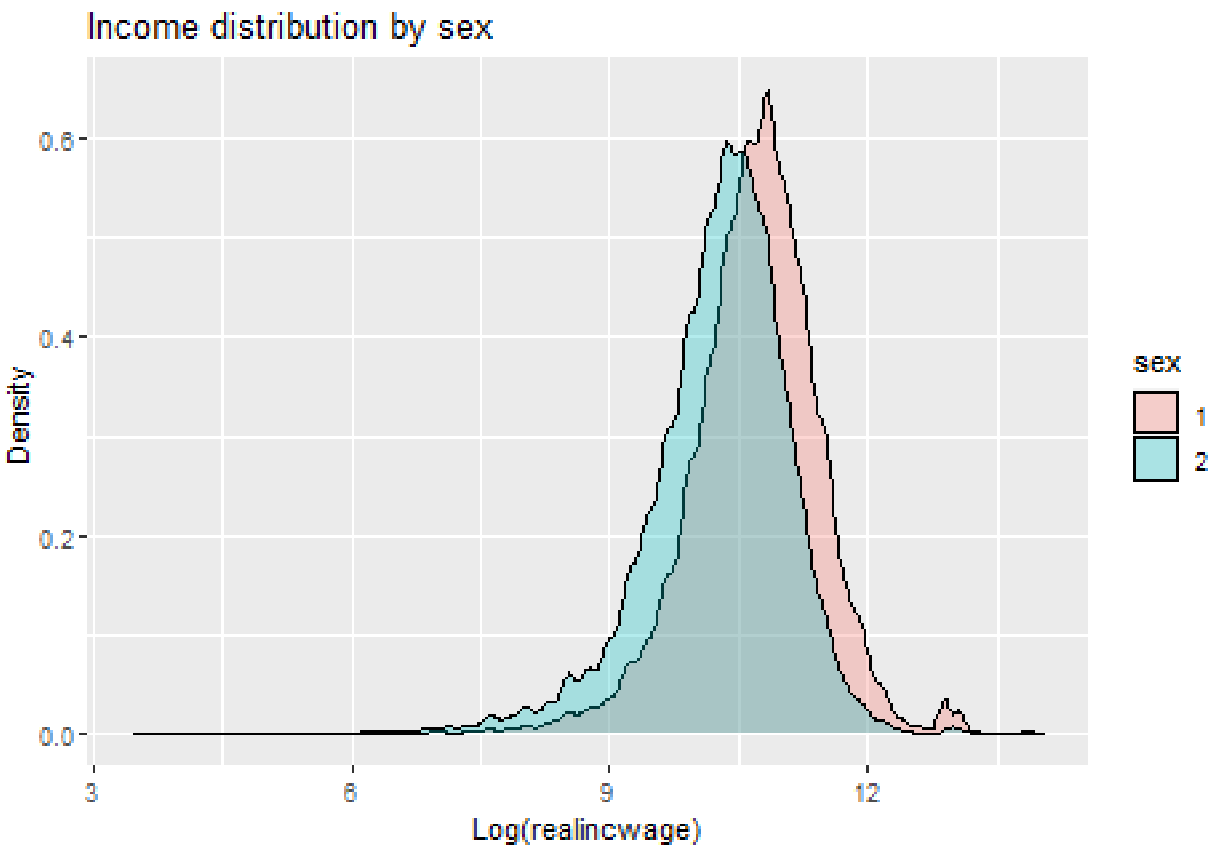


Let's compare the median and mean income for males and females to see if there is a difference. In this case the median is more meaningful because outliers can skew the average. As we see, the median income is around 10,000 dollars higher for males, while when considering the mean, the difference in income between the two group is even larger than 10,000 dollars.

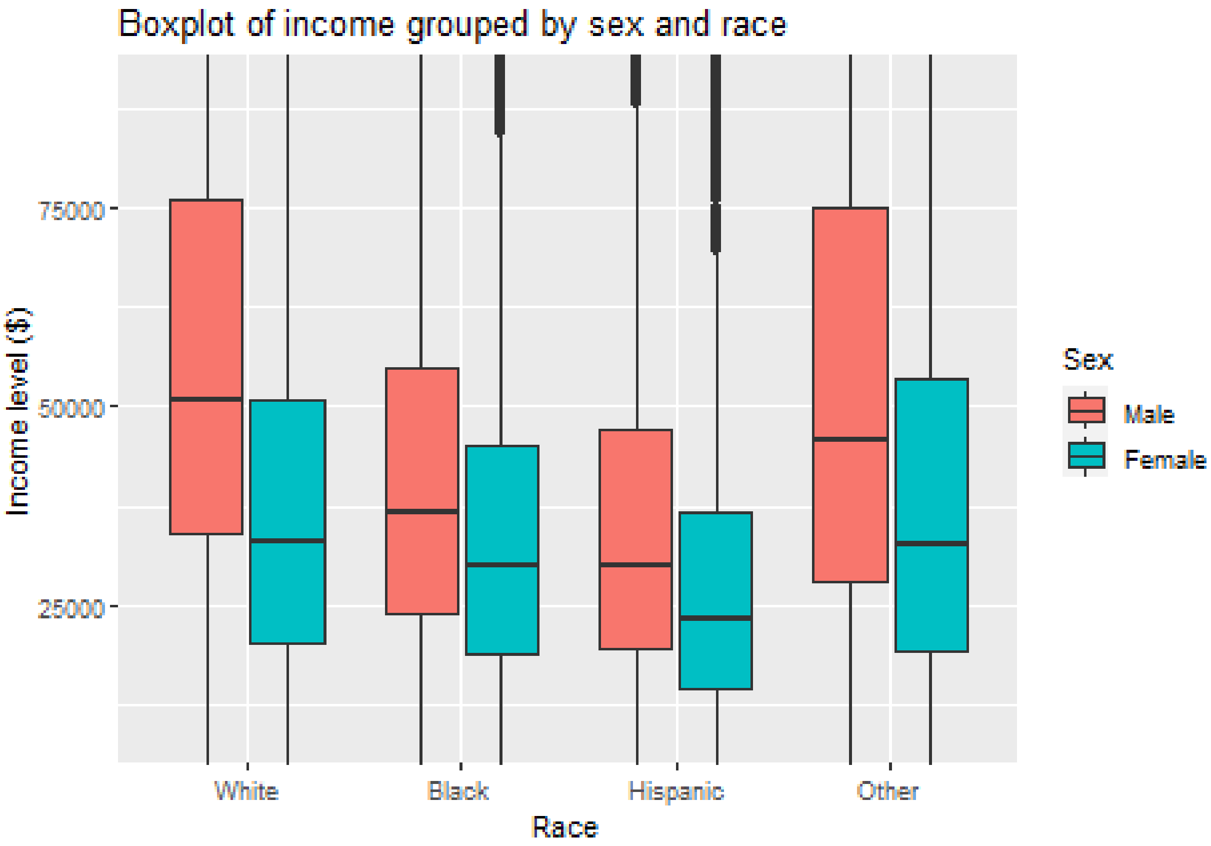
```
group_median = aggregate(data$realincwage, list(data$sex), FUN=median)
colnames(group_median) <- c("Sex", "Median income ($)")
levels(group_median$Sex) <- c("Male","Female")

group_mean = aggregate(data$realincwage, list(data$sex), FUN=mean)
colnames(group_mean) <- c("Sex", "Average income ($)")
group_mean$Sex <- NULL
cbind(group_median, group_mean)
```

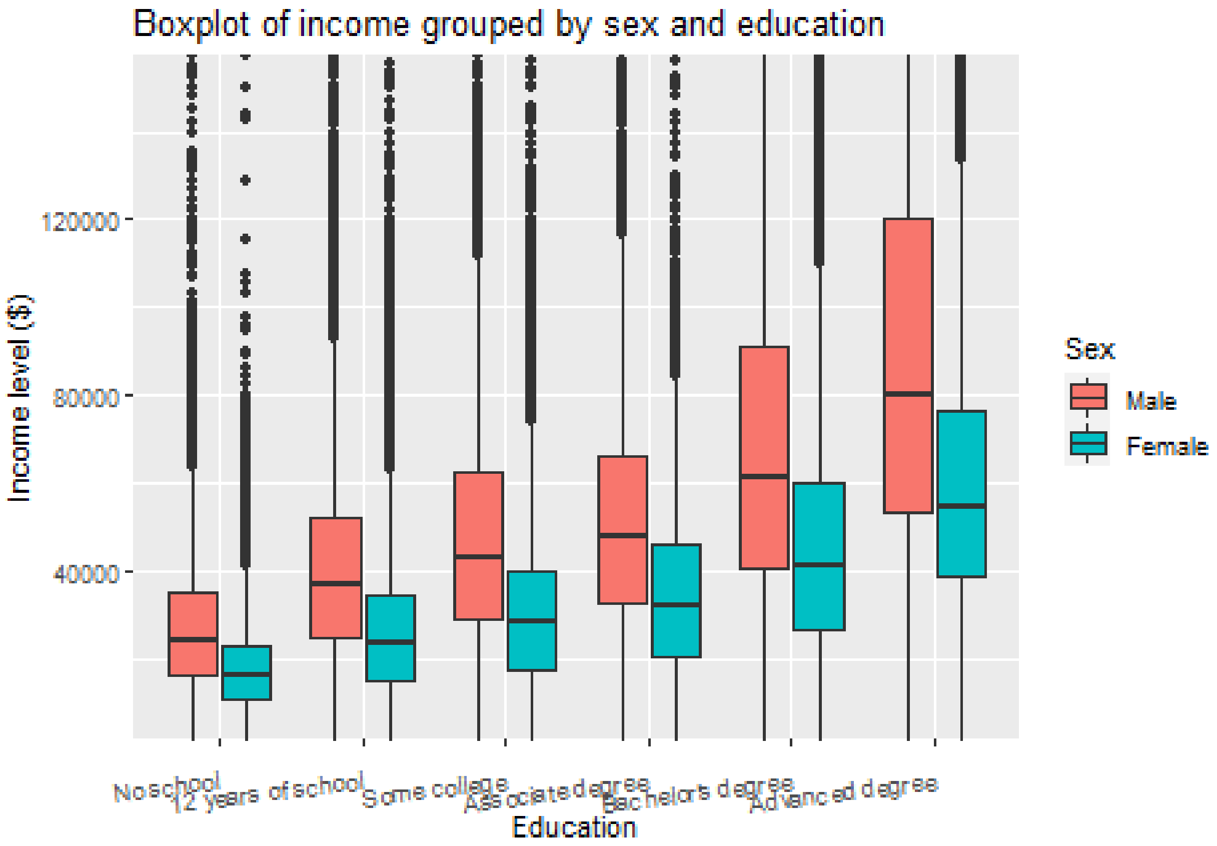
```
##      Sex Median income ($) Average income ($)
## 1   Male          45714.68           58415.13
## 2 Female          31120.32           38404.24
```

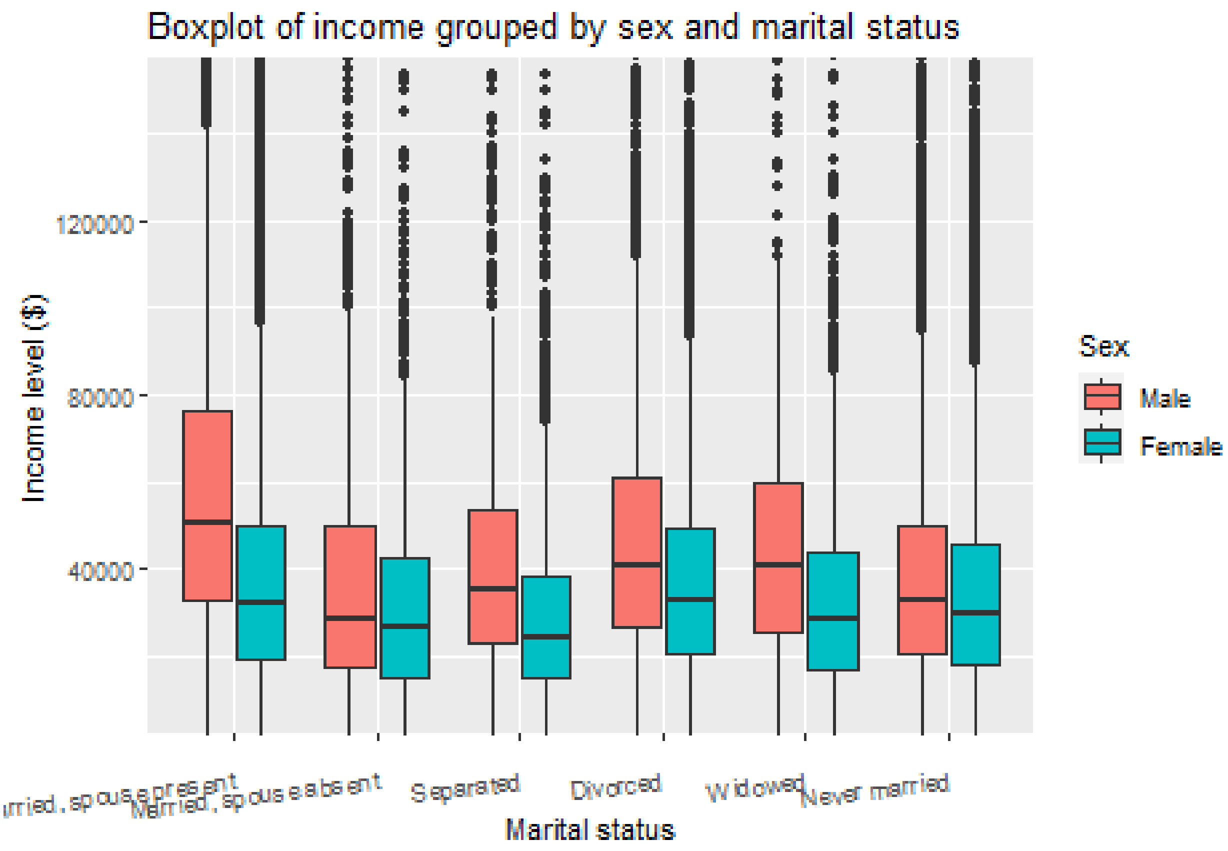
```
ggplot(data, aes(x=log(realincwage), fill=sex)) +
    geom_density(alpha=.3) +
    labs(title  = "Income distribution by sex", x ="Log(realincwage)", y="Density")
```

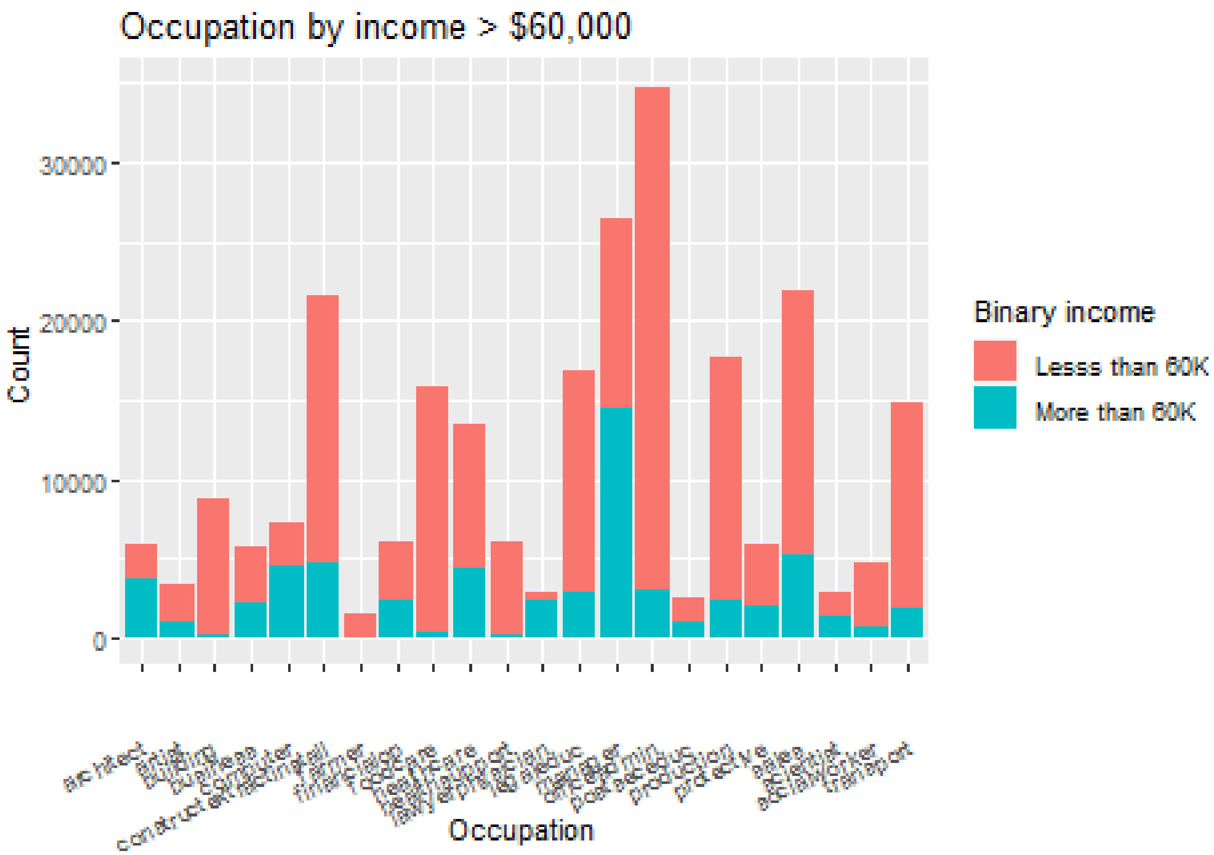


```
##       Race Average male income ($) Average female income ($)
## 1    Black                44250.53                  36104.23
## 2 Hispanic                39176.28                  29308.66
## 3    Other                58930.52                  42098.49
## 4    White                65180.22                  40343.54
##   Abs difference in income
## 1                 8146.299
## 2                 9867.619
## 3                16832.026
## 4                24836.680
```

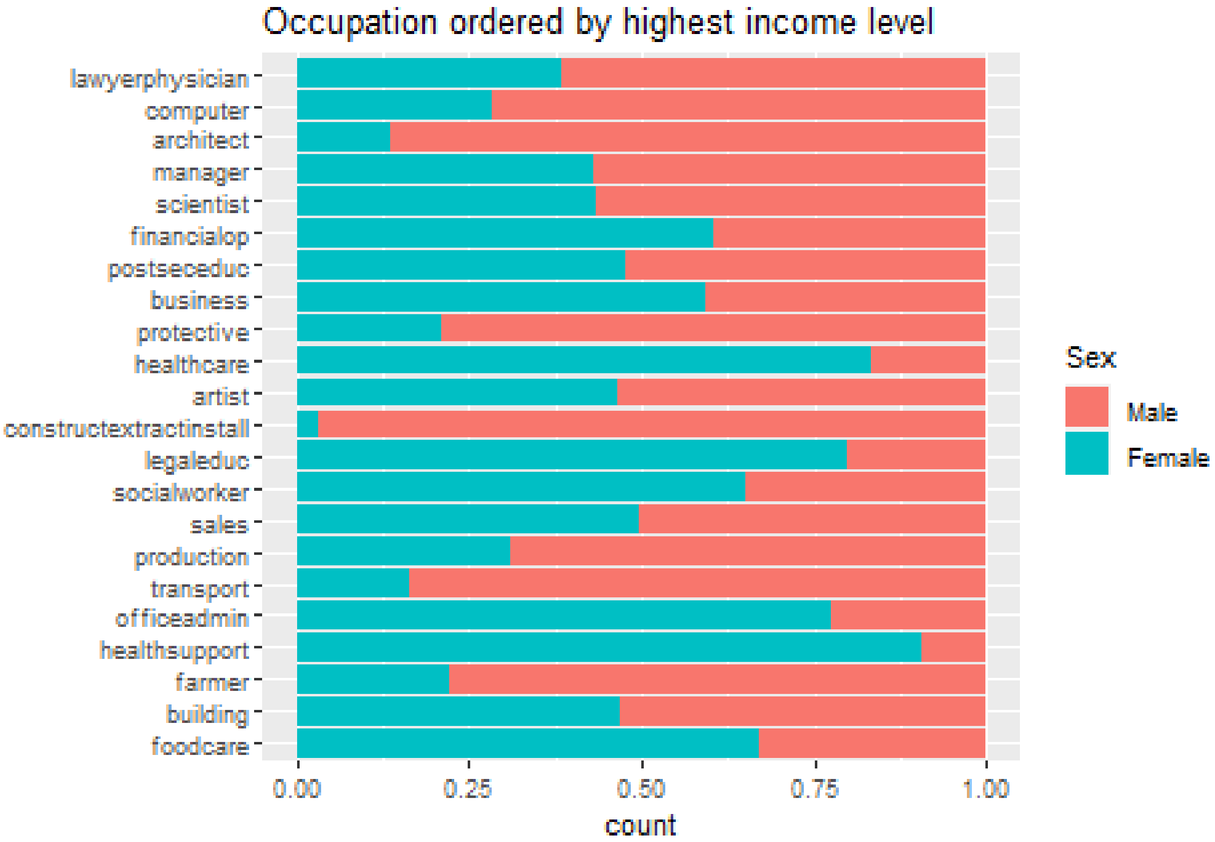

Boxplot of income grouped by sex and race
Income level ($)
75000
50000
25000
White
Black
Hispanic
Other
Race
Sex
Male
Female


Boxplot of income grouped by sex and education
Income level ($)
120000
80000
40000
No school
12 years of school
Some college
Associate degree
Bachelors degree
Advanced degree
Education
Sex
Male
Female

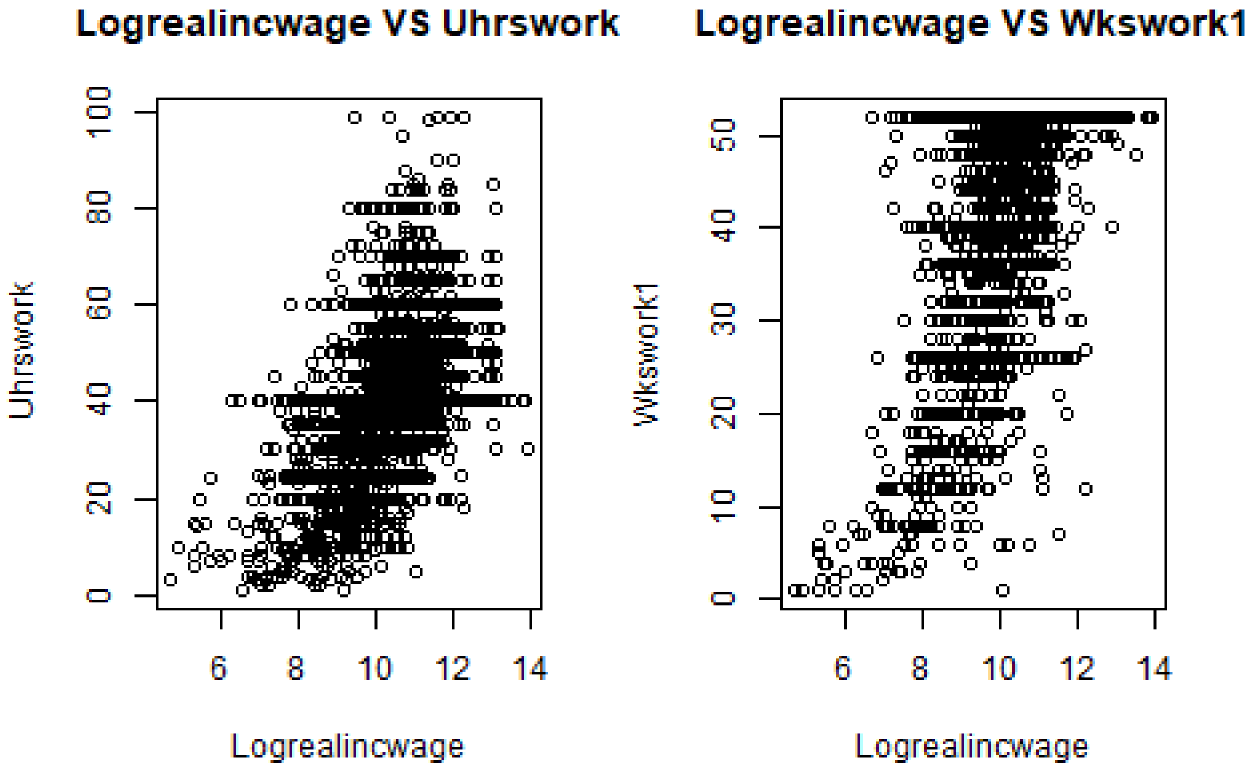

Boxplot of income grouped by sex and marital status
Income level ($)
120000
80000
40000
Marital status
arried, spouse present
Married, spouse absent
Separated
Divorced
Widowed
Never married
Sex
Male
Female



We can see that some occupations clearly have more observations with income higher than 60,000 USD. For example, while most of the office admins and farmers earn less than 60,000 USD, most of lawers, physicians and computer specialists earn more.



It is clear that sex distribution between occupations is different. While occupations like transport, construction and architect are mostly observed among males, office admin, foodcare and healthcare related jobs are mostly observed among females. We can also notice that higher paying jobs are male dominated.



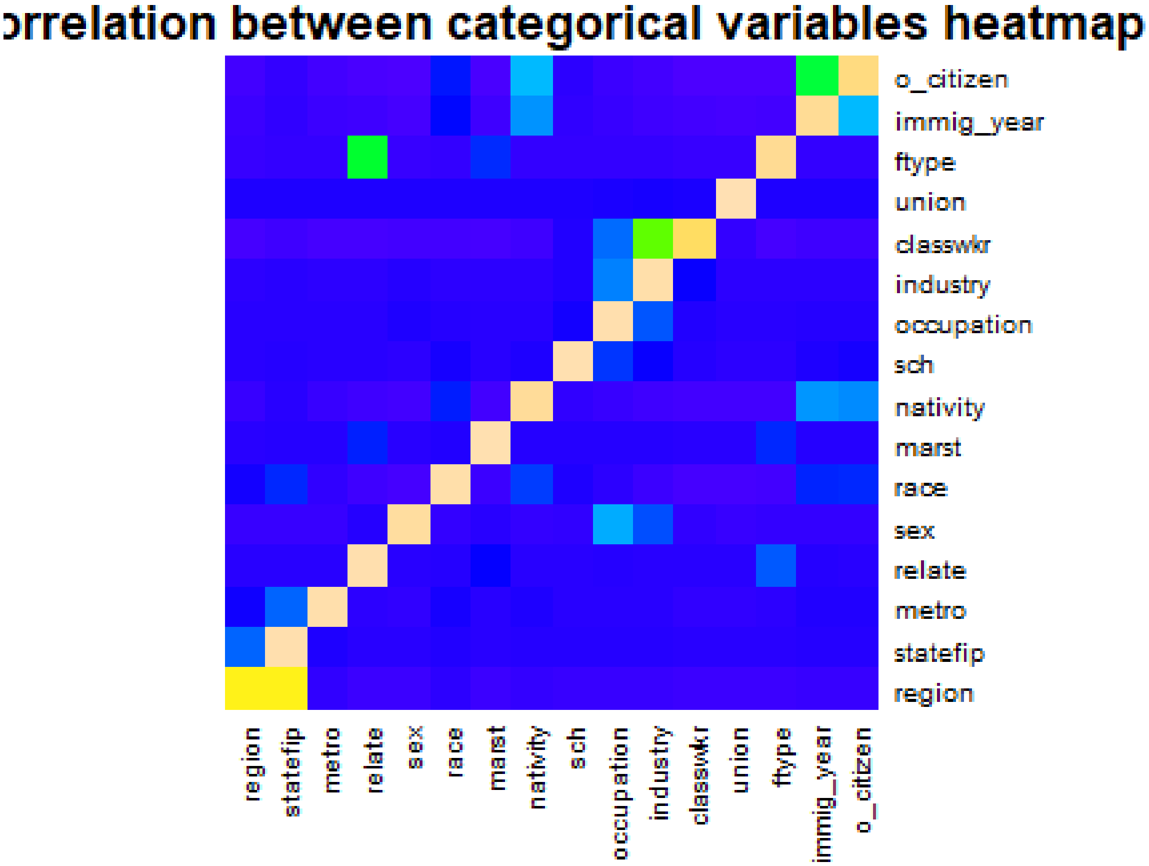
We plotted the sample of 10,000 observations of *logrealincwage* with *uhrswork* and *wkswork1* which show a clear trend of increase in realincome with the increase in time dedicated to work. To identify the extend of this trend, we will build a Linear Regression model.

Statistical analysis

Chi-Square Test

We will use the Chi-Square Test to perform a correlation analysis between categorical variables.

```
data.cat <- data[c("region", "statefip", "metro", "relate", "sex", "race", "marst", "nativity", "sch",

chisq.matrix <- function(x) {
  names <- colnames(x);
  ndim <- length(names)
  pvals <- matrix(nrow=ndim, ncol=ndim, dimnames = list(names, names))
  stats <- matrix(nrow=ndim, ncol=ndim, dimnames = list(names, names))
  for (i in 1:ndim) {
    for (j in i:ndim) {
      test <- chisq.test(x[,i],x[,j], simulate.p.value = TRUE)
      pvals[i,j] = test$p.value
      pvals[j,i] = pvals[i,j]
      stats[i,j] = test$statistic
      stats[j,i] = stats[i,j]
    }
  }
  return (list("p.values"=pvals, "statistics"=stats))
}

mat <- chisq.matrix(data.cat)
#mat$p.values
heatmap(mat$statistics, col = topo.colors(256), Colv = NA, Rowv = NA, main="Correlation between categor
```



As the p-value for all pairs of variables is 0.0005, there is a correlation between all variables. The value of test statistics shows that the correlation is particularly high for the following pairs: *region* and*statefip*, *relate* and *ftype*. Thus, we use only 1 of the variables in each pair: *region* and *relate*. Then, *o_citizen* is correlated with both *nativity* and *immig_year*. As the variables prepresent similar information related to immigration, we will use only *nativity*. Interestingly, *occupation* variable is correlated with *sex*, which proves our observations about the gender disproportions for some occupations. As the meaning of the variables is different, we will keep both of them. Similarly, *indusry* and *classwkr* are correlated, but we will keep both variables.

ANOVA

Let now see if there is a statistically significant difference between the mean income for males and females (H1). In this case the continuous income variable *realincwage* is the dependent variable and *sex* is the independent variable. The assumption of sample independence can be considered true. It remains to check the normality of residuals and the variance equality assumption.

From the boxplot of before we could already saw visually that the variance was slightly higher for the males group given that the the interquartile range for males was larger than the one for females.

To test this, we run a Bartlett's Test to determine whether or not the income variances between males and females are different. The p-value is smaller than that 0.05 significance level, so we have evidence that the samples do not have equal variances.

```
bartlett.test(data$realincwage ~ data$sex)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  data$realincwage by data$sex
## Bartlett's K-squared = 26241, df = 1, p-value < 2.2e-16
```

In general, ANOVA's are considered to be fairly robust against violations of the equal variances assumption as long as each group has the same sample size, which is the case:

```
summary(as.factor(data$sex))
```

```
##      1      2
## 124963 122321
```

```
res.aov <- aov(log(data$realincwage) ~ data$sex, data = data)
summary(res.aov)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## data$sex         1  11550   11550   17548 <2e-16 ***
## Residuals   247282 162761       1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

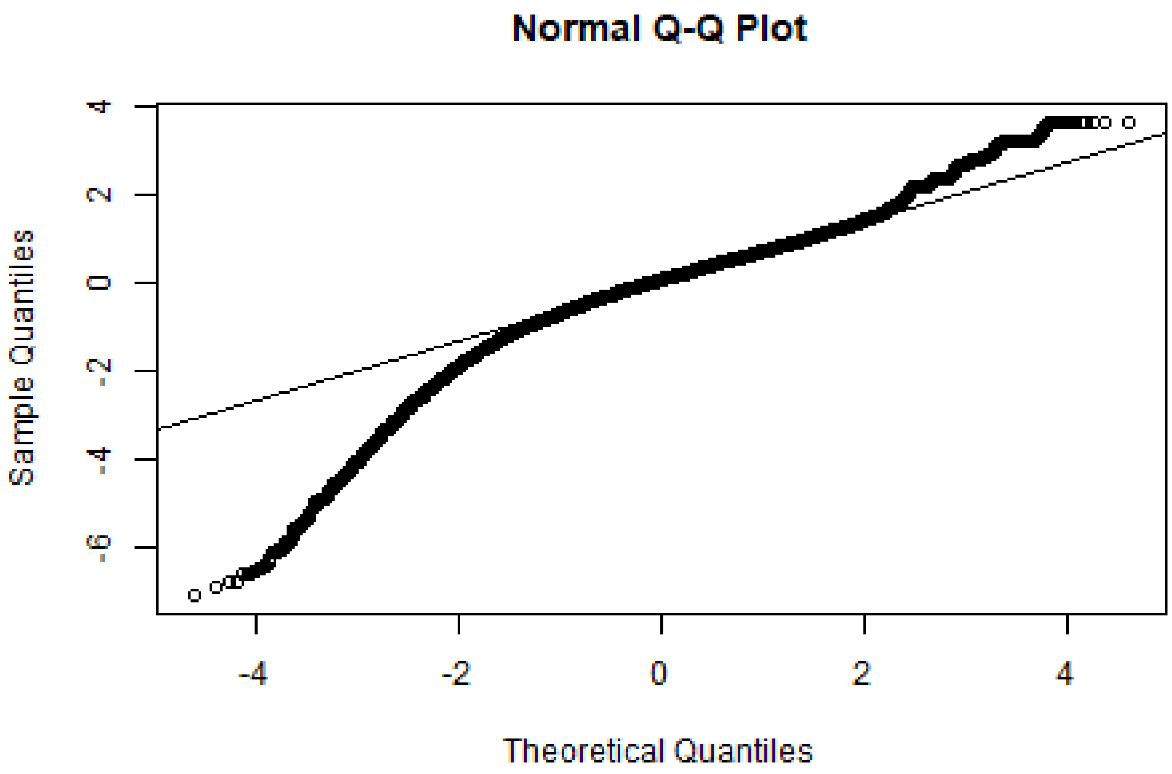
As the p-value is less than the significance level 0.05, we can conclude that there are significant differences in average income between the males and females. Still, running the ANOVA test with the assumption of equality of variances that is violated can cause more frequent type I error. Thus let's try with Welch's ANOVA. For normal, different-variance, and balanced data (i.e. same-size samples), Welch's has the most power and the lowest type I error rate. By looking at the result of this test we can draw the same conclusion as with the ANOVA test.

```
oneway.test(log(data$realincwage) ~ data$sex, data = data,
            var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  log(data$realincwage) and data$sex
## F = 17516, num df = 1, denom df = 244505, p-value < 2.2e-16
```

Clearly from the Q-Q plot below the residuals are not normally distributed however the one-way is considered a robust test against the normality assumption.

```
qqnorm(res.aov$residuals)
qqline(res.aov$residuals)
```



Regression

Linear Regression

Let’s define the variables we will use in the regression.

```
data.reg <- data[c("year", "numprec", "region", "metro", "relate", "age", "sex", "race", "marst", "nati

data.reg$year <- scale(data.reg$year)
data.reg$numprec <- scale(data.reg$numprec)
data.reg$age <- scale(data.reg$age)
data.reg$wkswork1 <- scale(data.reg$wkswork1)
data.reg$uhrswork <- scale(data.reg$uhrswork)

set.seed(1)

train <- sample(1:nrow(data.reg), nrow(data.reg)*0.75)
test <- (-train)
y <- log(data.reg$realincwage)
y.test <- y[test]
```

We will build the linear regression using the *realincwage* as a response.

```
reg.out <- lm(log(realincwage) ~ . , data = data.reg[train,])
summary(reg.out)
```

```
##
## Call:
```

```
## lm(formula = log(realincwage) ~ ., data = data.reg[train, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2381 -0.2842  0.0122  0.2957  5.5169
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       1.034e+01  2.067e-02 500.195  < 2e-16 ***
## year                              2.077e-02  1.203e-03  17.257  < 2e-16 ***
## numprec                           9.705e-03  1.396e-03   6.951 3.65e-12 ***
## region12                          2.058e-03  5.388e-03   0.382 0.702509
## region21                         -7.440e-02  5.115e-03 -14.546  < 2e-16 ***
## region22                         -1.111e-01  5.093e-03 -21.815  < 2e-16 ***
## region31                         -5.346e-02  4.772e-03 -11.202  < 2e-16 ***
## region32                         -1.507e-01  6.768e-03 -22.270  < 2e-16 ***
## region33                         -1.227e-01  5.593e-03 -21.944  < 2e-16 ***
## region41                         -6.647e-02  5.275e-03 -12.601  < 2e-16 ***
## region42                          2.993e-02  5.013e-03   5.970 2.37e-09 ***
## metro2                            1.386e-01  3.974e-03  34.864  < 2e-16 ***
## metro3                            1.625e-01  3.504e-03  46.393  < 2e-16 ***
## metro4                            7.530e-02  3.973e-03  18.954  < 2e-16 ***
## relate201                        -1.619e-02  3.007e-03  -5.386 7.22e-08 ***
## relate301                        -1.598e-01  7.031e-03 -22.728  < 2e-16 ***
## relate501                        -6.413e-02  1.106e-02  -5.797 6.77e-09 ***
## relate701                        -1.157e-01  1.290e-02  -8.967  < 2e-16 ***
## relate901                        -2.100e-01  3.935e-02  -5.336 9.49e-08 ***
## relate1001                       -1.444e-01  1.186e-02 -12.177  < 2e-16 ***
## relate1114                       -2.517e-02  7.175e-03  -3.508 0.000452 ***
## relate1115                       -9.253e-02  1.029e-02  -8.994  < 2e-16 ***
## relate1241                       -1.364e-01  2.267e-02  -6.015 1.80e-09 ***
## relate1260                       -1.130e-01  1.666e-02  -6.783 1.18e-11 ***
## age                               5.575e-02  1.323e-03  42.142  < 2e-16 ***
## sex2                             -1.973e-01  2.931e-03 -67.311  < 2e-16 ***
## race2                            -7.028e-02  4.322e-03 -16.263  < 2e-16 ***
## race3                            -8.739e-02  4.509e-03 -19.384  < 2e-16 ***
## race4                            -3.937e-02  5.335e-03  -7.379 1.60e-13 ***
## marst2                           -5.957e-02  1.028e-02  -5.797 6.78e-09 ***
## marst3                           -7.039e-02  8.020e-03  -8.777  < 2e-16 ***
## marst4                           -3.049e-02  4.394e-03  -6.939 3.98e-12 ***
## marst5                           -7.769e-02  1.040e-02  -7.467 8.23e-14 ***
## marst6                           -8.318e-02  4.301e-03 -19.337  < 2e-16 ***
## nativity2                         6.403e-05  9.302e-03   0.007 0.994508
## nativity3                         2.549e-02  9.102e-03   2.800 0.005110 **
## nativity4                         4.219e-02  7.256e-03   5.814 6.10e-09 ***
## nativity5                        -8.629e-02  4.225e-03 -20.426  < 2e-16 ***
## schfsch                           1.576e-01  5.203e-03  30.298  < 2e-16 ***
## schscol                           2.289e-01  5.696e-03  40.178  < 2e-16 ***
## schasoc                           2.763e-01  6.197e-03  44.587  < 2e-16 ***
## schbach                           4.488e-01  5.899e-03  76.078  < 2e-16 ***
## schadvd                           6.474e-01  6.773e-03  95.583  < 2e-16 ***
## occupationartist                 -1.783e-01  1.279e-02 -13.943  < 2e-16 ***
## occupationbuilding               -4.969e-01  1.053e-02 -47.206  < 2e-16 ***
## occupationbusiness               -6.016e-02  1.113e-02  -5.404 6.54e-08 ***
## occupationcomputer                 8.870e-02  1.043e-02   8.503  < 2e-16 ***
## occupationconstructextractinstall -1.815e-01  9.317e-03 -19.477  < 2e-16 ***
## occupationfarmer                  -4.400e-01  2.254e-02 -19.525  < 2e-16 ***
## occupationfinancialop             -5.124e-02  1.107e-02  -4.627 3.71e-06 ***
## occupationfoodcare                -4.411e-01  1.012e-02 -43.588  < 2e-16 ***
## occupationhealthcare               5.385e-02  1.030e-02   5.227 1.72e-07 ***
## occupationhealthsupport           -4.106e-01  1.198e-02 -34.274  < 2e-16 ***
## occupationlawyerphysician          2.724e-01  1.388e-02  19.631  < 2e-16 ***
## occupationlegaleduc               -2.689e-01  1.015e-02 -26.508  < 2e-16 ***
## occupationmanager                  2.606e-02  8.690e-03   2.998 0.002715 **
## occupationofficeadmin             -3.000e-01  8.785e-03 -34.150  < 2e-16 ***
## occupationpostseceduc             -1.539e-01  1.501e-02 -10.254  < 2e-16 ***
## occupationproduction              -3.219e-01  9.278e-03 -34.696  < 2e-16 ***
## occupationprotective              -2.298e-01  1.154e-02 -19.911  < 2e-16 ***
## occupationsales                   -2.048e-01  9.320e-03 -21.979  < 2e-16 ***
## occupationscientist               -7.739e-02  1.352e-02  -5.723 1.05e-08 ***
## occupationsocialworker            -3.197e-01  1.203e-02 -26.583  < 2e-16 ***
## occupationtransport               -3.770e-01  9.707e-03 -38.840  < 2e-16 ***
## industryCommunications             2.362e-01  1.963e-02  12.032  < 2e-16 ***
## industryDurables                   2.584e-01  1.859e-02  13.897  < 2e-16 ***
## industryEducation                  4.073e-02  1.898e-02   2.146 0.031879 *
## industryFinance                    2.494e-01  1.867e-02  13.359  < 2e-16 ***
## industryHotelsRestaurants         -3.156e-02  1.909e-02  -1.653 0.098267 .
## industryMedical                    1.488e-01  1.868e-02   7.966 1.65e-15 ***
## industryMiningConstruction         2.211e-01  1.883e-02  11.739  < 2e-16 ***
## industryNondurables                2.384e-01  1.878e-02  12.694  < 2e-16 ***
## industryProfessional               1.982e-01  1.846e-02  10.737  < 2e-16 ***
## industryPublicadmin                2.114e-01  1.920e-02  11.009  < 2e-16 ***
## industryRetailTrade               -1.343e-03  1.855e-02  -0.072 0.942299
## industrySocArtOther               -4.084e-03  1.865e-02  -0.219 0.826628
## industryTransport                  2.506e-01  1.898e-02  13.207  < 2e-16 ***
## industryUtilities                  3.805e-01  2.129e-02  17.871  < 2e-16 ***
## industryWholesaleTrade             2.242e-01  1.912e-02  11.724  < 2e-16 ***
## classwkrPublic sector              4.023e-02  4.897e-03   8.216  < 2e-16 ***
## union1                            -3.178e-02  3.367e-03  -9.437  < 2e-16 ***
## union2                             1.410e-01  7.632e-03  18.472  < 2e-16 ***
## union3                             4.044e-02  2.332e-02   1.734 0.082856 .
## wkswork1                           3.001e-01  1.223e-03 245.402  < 2e-16 ***
## uhrswork                           2.767e-01  1.291e-03 214.375  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5082 on 185378 degrees of freedom
## Multiple R-squared:  0.6338, Adjusted R-squared:  0.6337
## F-statistic:  3820 on 84 and 185378 DF,  p-value: < 2.2e-16
```

The summary shows that the residuals are symmetrically distributed with the median equal to almost 0 (0.0106). We will have a closer look at the residuals later. We can see from the summary, that the full Linear regression model explains 63.32% of variance associated with the response variable. The p-value of the the F-statistic is nearly 0, which means that at least one variable is associated with the response and we reject that null hypothesis that all coefficients are equal to zero. P-values associated with some of the coefficients are larger that 0.05, which means that there's no evidence of significance. The coefficients related to continuous variables are all significant, while the insignificant ones are the dummy variables.

First, we perform Variance Inflation Factor analysis to check for multicollinearity.

```
vif(reg.out)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## year         1.038691  1        1.019162
## numprec      1.395668  1        1.181384
## region       1.448826  8        1.023443
## metro        1.313151  3        1.046451
## relate       2.049415 10        1.036529
## age          1.257563  1        1.121411
## sex          1.542013  1        1.241778
## race         2.544578  3        1.168430
## marst        2.420928  5        1.092441
## nativity     2.040228  4        1.093226
## sch          2.523549  5        1.096986
## occupation  99.624194 21        1.115784
## industry   104.167262 15        1.167502
## classwkr     2.660145  1        1.630995
## union        1.058182  3        1.009470
## wkswork1     1.072985  1        1.035850
## uhrswork     1.197321  1        1.094222
```

The adjusted GVIF^(1/(2*Df)) shows that there's no evidence of substantial multicollinearity among the variables. Thus, we can perform the avona test to see if there are differences in groups.

```
anova(reg.out)
```

```
## Analysis of Variance Table
##
## Response: log(realincwage)
##                Df Sum Sq Mean Sq  F value    Pr(>F)
## year            1    149   149.0   577.17 < 2.2e-16 ***
## numprec         1    406   406.4  1573.85 < 2.2e-16 ***
## region          8    757    94.7   366.66 < 2.2e-16 ***
## metro           3   1724   574.8  2225.85 < 2.2e-16 ***
## relate         10   3059   305.9  1184.69 < 2.2e-16 ***
## age             1    857   856.9  3318.47 < 2.2e-16 ***
## sex             1   9076  9076.2 35148.71 < 2.2e-16 ***
## race            3   3982  1327.4  5140.55 < 2.2e-16 ***
## marst           5    612   122.5   474.31 < 2.2e-16 ***
## nativity        4    754   188.4   729.56 < 2.2e-16 ***
## sch             5  14310  2861.9 11083.22 < 2.2e-16 ***
## occupation     21  10590   504.3  1952.85 < 2.2e-16 ***
## industry       15   2592   172.8   669.31 < 2.2e-16 ***
## classwkr        1     29    29.1   112.59 < 2.2e-16 ***
## union           3    190    63.4   245.46 < 2.2e-16 ***
## wkswork1        1  21908 21908.2 84842.45 < 2.2e-16 ***
## uhrswork        1  11867 11867.1 45956.80 < 2.2e-16 ***
## Residuals  185378  47869     0.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

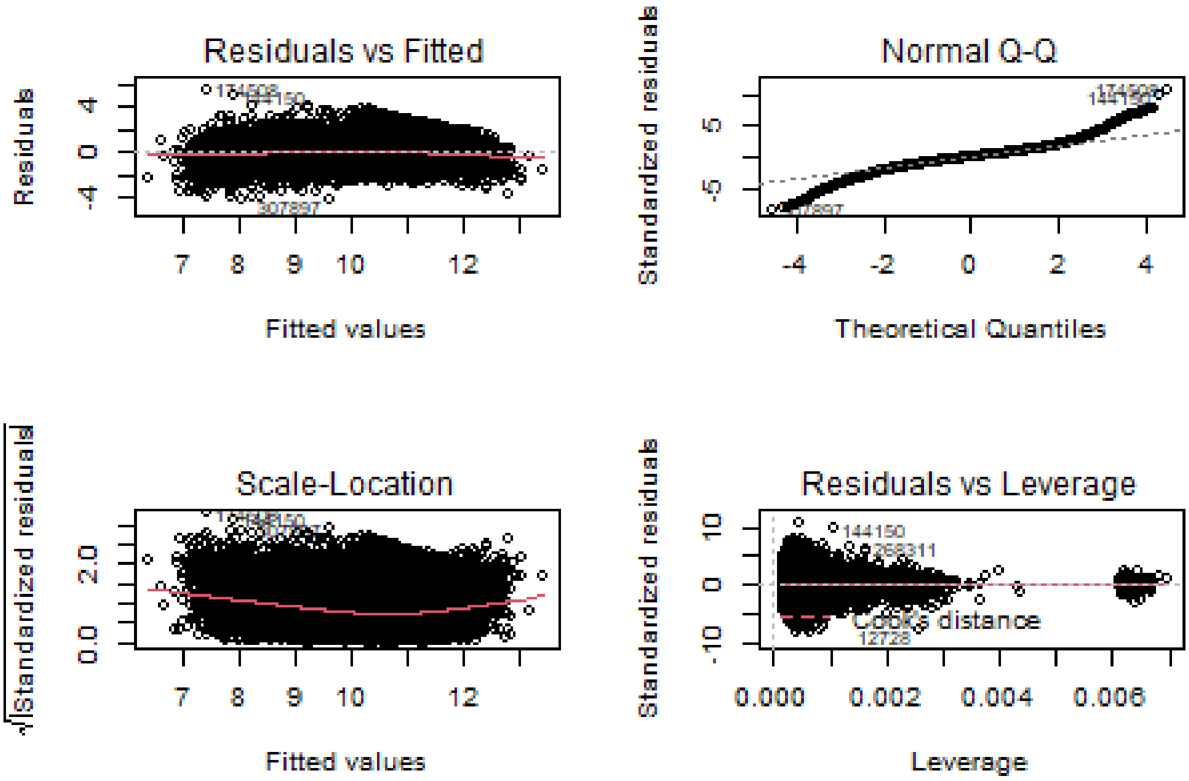
The result of the anova test show that for each variable there are at least 2 groups with the significant difference in means. To see the difference within some of the variables in more details, we perform a TukeyHSD test.

```r
a <- aov(log(realincwage) ~ race +sch +marst +nativity +union, data = data.reg[train,])
TukeyHSD(a)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log(realincwage) ~ race + sch + marst + nativity + union, data = data.reg[train, 
##
## $race
##            diff         lwr         upr p adj
## 2-1 -0.24518565 -0.26064572 -0.22972557     0
## 3-1 -0.39631512 -0.40903758 -0.38359266     0
## 4-1 -0.05241954 -0.07019702 -0.03464205     0
## 3-2 -0.15112947 -0.16953451 -0.13272443     0
## 4-2  0.19276611  0.17056423  0.21496799     0
## 4-3  0.34389558  0.32350513  0.36428603     0
##
## $sch
##                 diff        lwr        upr p adj
## fsch-nosc 0.1961848 0.17554560 0.21682395     0
## scol-nosc 0.3149262 0.29291511 0.33693720     0
## asoc-nosc 0.3928731 0.36903575 0.41671037     0
## bach-nosc 0.6703254 0.64911076 0.69153996     0
## advd-nosc 0.9780250 0.95469798 1.00135199     0
## scol-fsch 0.1187414 0.10339874 0.13408402     0
## asoc-fsch 0.1966883 0.17882413 0.21455243     0
## bach-fsch 0.4741406 0.45996420 0.48831697     0
## advd-fsch 0.7818402 0.76466291 0.79901751     0
## asoc-scol 0.0779469 0.05851399 0.09737982     0
## bach-scol 0.3553992 0.33929081 0.37150760     0
## advd-scol 0.6630988 0.64429537 0.68190229     0
## bach-asoc 0.2774523 0.25892633 0.29597828     0
## advd-asoc 0.5851519 0.56424018 0.60606368     0
## advd-bach 0.3076996 0.28983504 0.32556421     0
##
## $marst
##             diff         lwr          upr     p adj
## 2-1 -0.155099601 -0.19768416 -0.112515044 0.0000000
## 3-1 -0.187714784 -0.22061391 -0.154815656 0.0000000
## 4-1 -0.067500494 -0.08359117 -0.051409818 0.0000000
## 5-1 -0.195476078 -0.23865014 -0.152302011 0.0000000
## 6-1 -0.217339912 -0.23082163 -0.203858197 0.0000000
## 3-2 -0.032615183 -0.08569399  0.020463623 0.4977987
## 4-2  0.087599107  0.04294581  0.132252404 0.0000003
## 5-2 -0.040376477 -0.10036820  0.019615248 0.3909794
## 6-2 -0.062240311 -0.10602117 -0.018459451 0.0007241
## 4-3  0.120214290  0.08467804  0.155750538 0.0000000
## 5-3 -0.007761294 -0.06131421  0.045791625 0.9984688
## 6-3 -0.029625128 -0.06405871  0.004808456 0.1387800
## 5-4 -0.127975584 -0.17319143 -0.082759741 0.0000000
## 6-4 -0.149839418 -0.16887173 -0.130807103 0.0000000
## 6-5 -0.021863834 -0.06621831  0.022490638 0.7242465
##
## $nativity
##            diff          lwr         upr     p adj
## 2-1  0.03997249  0.002223120  0.07772187 0.0316742
## 3-1  0.07398687  0.036945475  0.11102826 0.0000005
## 4-1  0.11614107  0.088517498  0.14376465 0.0000000
## 5-1 -0.04505776 -0.057736864 -0.03237865 0.0000000
## 3-2  0.03401437 -0.018282032  0.08631078 0.3887138
## 4-2  0.07616858  0.030060974  0.12227618 0.0000647
## 5-2 -0.08503025 -0.124063686 -0.04599681 0.0000000
## 4-3  0.04215420 -0.003375574  0.08768398 0.0850078
## 5-3 -0.11904462 -0.157393791 -0.08069546 0.0000000
## 5-4 -0.16119883 -0.190552802 -0.13184486 0.0000000
##
## $union
##            diff         lwr         upr     p adj
## 1-0 -0.03563952 -0.04853697 -0.02274206 0.0000000
## 2-0  0.16985837  0.14090616  0.19881058 0.0000000
## 3-0  0.02565716 -0.06424590  0.11556021 0.8837834
## 2-1  0.20549789  0.17460747  0.23638831 0.0000000
## 3-1  0.06129667 -0.02924915  0.15184250 0.3033232
## 3-2 -0.14420121 -0.23838419 -0.05001824 0.0004860
```

The results show that there is significant difference in income between all races and educational levels. Regarding the marital status, there's a significant difference between category 1 (married people with a present spouse), category 4(divorced) and all other categories. There's no evidence of difference between category 6 (never married) and 5 (widowed) or 5 and 3 (separated). Regarding nativity, there's a significant difference between group 1 (native born people) and all others. There's no evidence of difference between group 2 (father foreign, mother native) and 3 (father native, mother foreign), and group 3 and 4 (both parents foreign born). The p-value for other groups is lower that 0.05, so the difference is statistically significant. Regarding the union variable, there's no significant difference between categories 3 and 0, and 3 and 1.

```
par(mfrow=c(2,2))
plot(reg.out)
```



```r
par(mfrow=c(1,1))
```

The residuals vs fitted values behave well and we don't see any systematic behaviors. The Q-Q plot shows that the observations don't follow the normal distribution and have fat tails. The scale location plot shows that there is some systematic behavior as the red line goes down a bit in the middle.

Calculating the MSE on the train and test set:

```r
lm.pred.new <- predict(reg.out, newdata = data.reg[test, ])
lm.pred <- predict(reg.out)
```

MSE on train set:

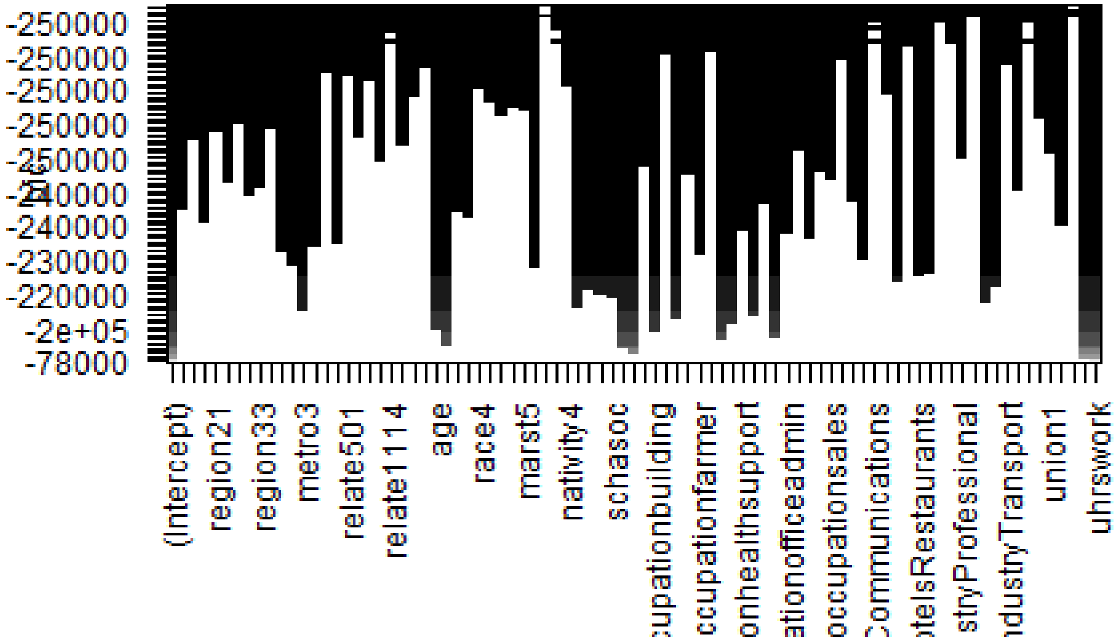
```r
mean((lm.pred - y[train])^2)
```

```
## [1] 0.2581037
```

MSE on test set:

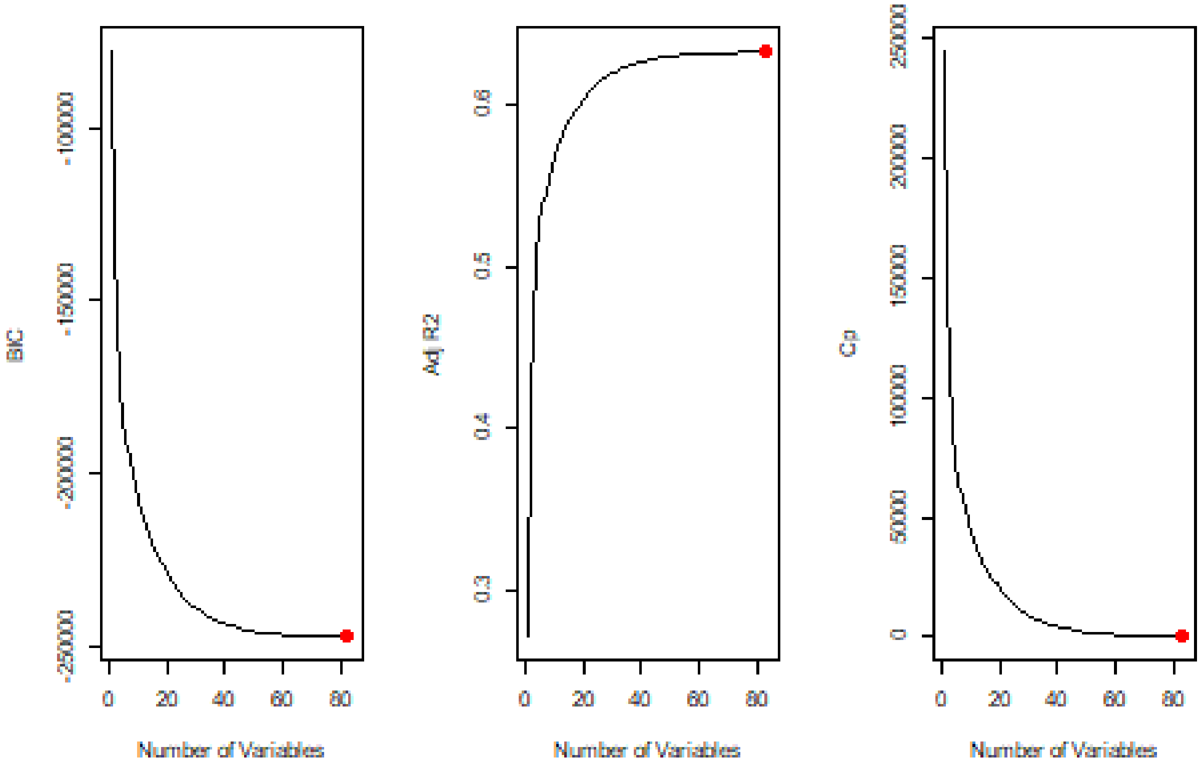
```r
mean((lm.pred.new - y[test])^2)
```

```
## [1] 0.2591257
```



Forward selection

```
par(mfrow=c(1,3))
plot(fwd.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')
n.fwd <- which.min(fwd.summary$bic)
points(n.fwd,fwd.summary$bic[n.fwd],col="red",cex=2,pch=20)

plot(fwd.summary$adjr2,xlab="Number of Variables",ylab="Adj R2",type='l')
n.fwd <- which.max(fwd.summary$adjr2)
points(n.fwd,fwd.summary$adjr2[n.fwd],col="red",cex=2,pch=20)

plot(fwd.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
n.fwd <- which.min(fwd.summary$cp)
points(n.fwd,fwd.summary$cp[n.fwd],col="red",cex=2,pch=20)
```



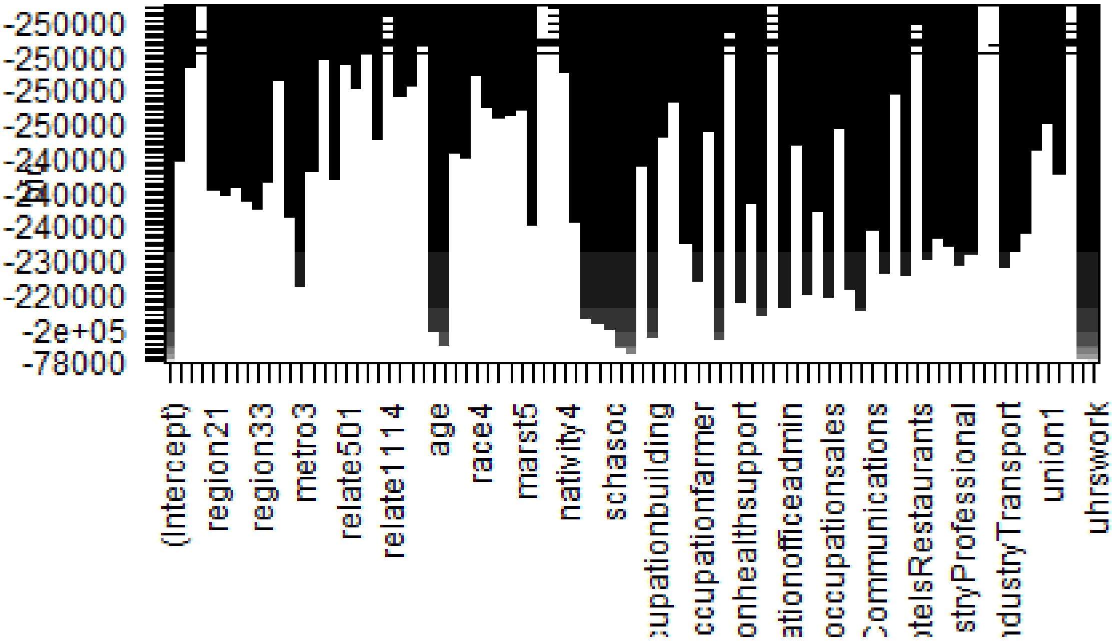
```
par(mfrow=c(1,1))
```

We can see that using BIC, Adj R^2 and AIC as a selection parameter, the subset which includes almost all of the variables is the best. In fact, only some of dummy variables are excluded from the model, which is impossible to implement in practice. Thus, we will use the full model.

```
coef(regfit.fwd, 5)
```

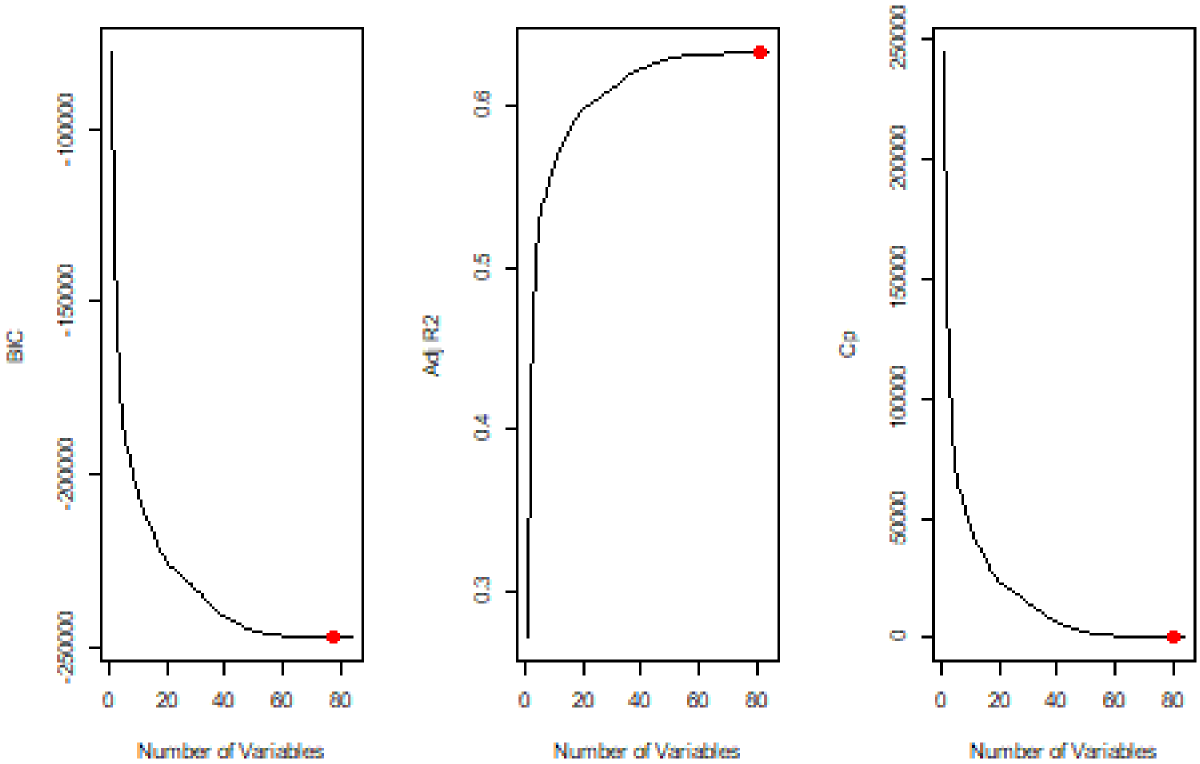
```
## (Intercept)        sex2     schbach     schadvd    wkswork1    uhrswork
##  10.4044325  -0.2339676   0.4282102   0.6811420   0.3250044   0.3077703
```

The most important variables chosen by the forward elimination procedure are *wkswork1*, *urswork*, *schbach*, *schadvd* and *sex*.



Backward selection

```
par(mfrow=c(1,3))
plot(bwd.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')
n.bwd.bic <- which.min(bwd.summary$bic)
points(n.bwd.bic,bwd.summary$bic[n.bwd.bic],col="red",cex=2,pch=20)

plot(bwd.summary$adjr2,xlab="Number of Variables",ylab="Adj R2",type='l')
n.bwd <- which.max(bwd.summary$adjr2)
points(n.bwd,bwd.summary$adjr2[n.bwd],col="red",cex=2,pch=20)

plot(bwd.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
n.bwd <- which.min(bwd.summary$cp)
points(n.bwd,bwd.summary$cp[n.bwd],col="red",cex=2,pch=20)
```



```
par(mfrow=c(1,1))
```

The results of the backward selection method are similar to forward selection with dummy variables removed from the model. As we cannot technically remove only some of the dummy variables, we will continue using the full model and to apply shrinkage techniques.

```
coef(regfit.bwd, 5)
```

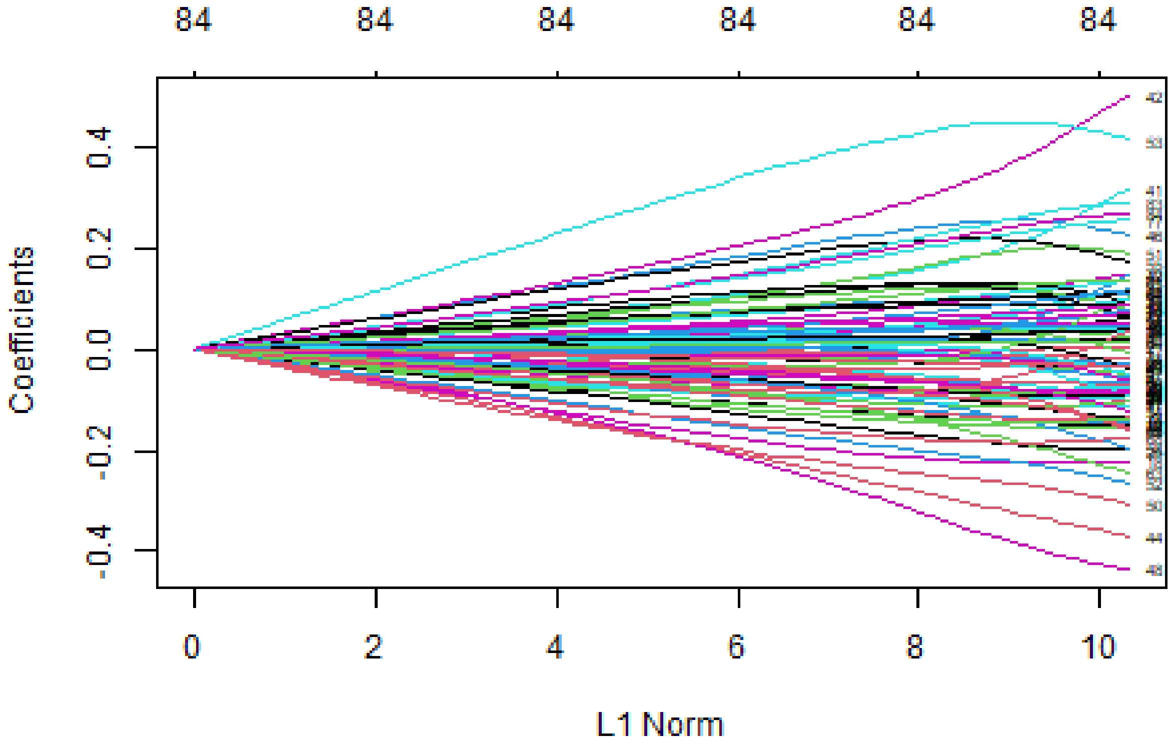
```
## (Intercept)        sex2     schbach     schadvd    wkswork1    uhrswork
##  10.4044325  -0.2339676   0.4282102   0.6811420   0.3250044   0.3077703
```

Interestingly, backward selection method selected the same 5 most important variables as the forward selection method.

Ridge regression

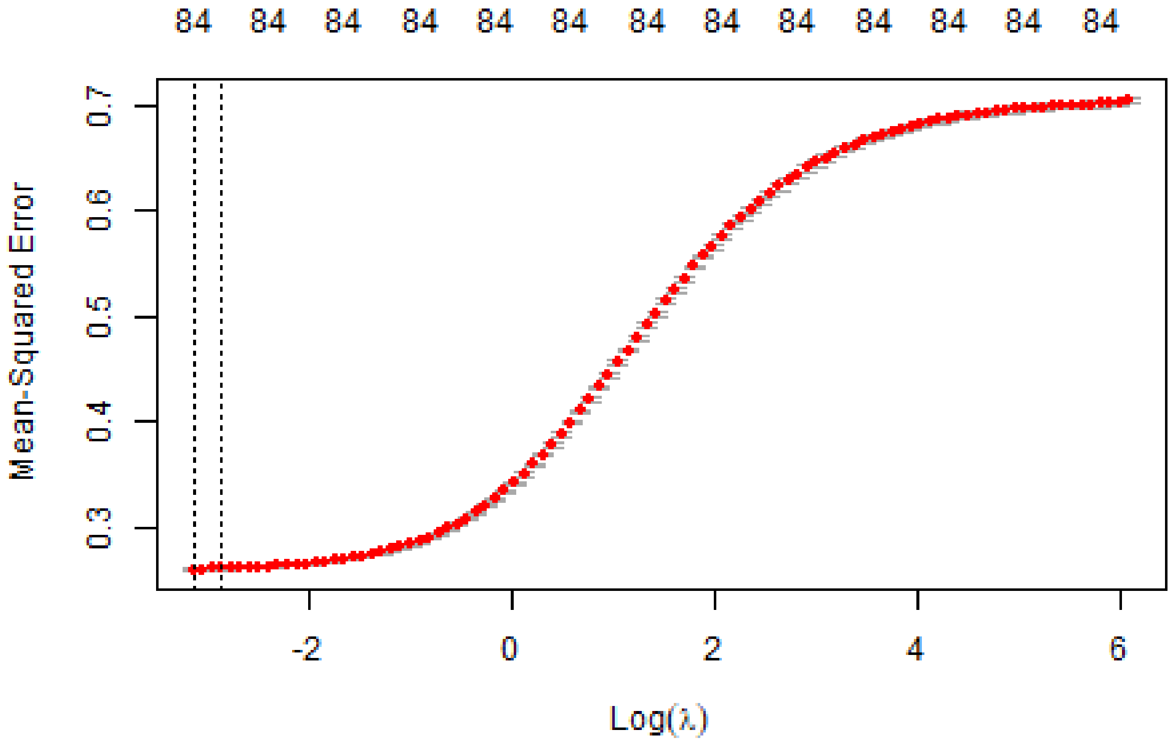
The Ridge regression was built with the grid of lambdas ranging from 1000 to 0.0001.

```
X <- model.matrix(log(realincwage) ~ . , data = data.reg)
X <- X[,-1]
y.test <- y[test]
grid <- 10^seq(3, -4, length=100)

ridge.mod <- glmnet(X, y, alpha=0, standardize = TRUE)
plot(ridge.mod, label=TRUE)
```



Then we perform cross validation to find the best value of lambda.

```
cv.out <- cv.glmnet(X[train, ], y[train], alpha = 0, nfold=10, type.measure = "mse")
plot(cv.out)
```



The graph shows that the lower the value of lambda, the lower the MSE.

```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.04371554
```

The value of best lambda is equal to 0.04371554

MSE with the best lambda:

```
ridge.pred <- predict(ridge.mod, s = bestlam, newx = X[train, ])
ridge.pred.new <- predict(ridge.mod, s = bestlam, newx = X[test, ])
mean((ridge.pred - y[train])^2) # train set
```

```
## [1] 0.2601077
```

```
mean((ridge.pred.new - y.test)^2) # test set
```

```
## [1] 0.2610079
```

MSE with lambda = 0:

```
ridge.pred2 <- predict(ridge.mod, s = 0, newx = X[train, ])
ridge.pred2.new <- predict(ridge.mod, s = 0, newx = X[test, ])
mean((ridge.pred2 - y[train])^2)
```

```
## [1] 0.2601056
```

```
mean((ridge.pred2.new - y.test)^2)
```

```
## [1] 0.2610058
```

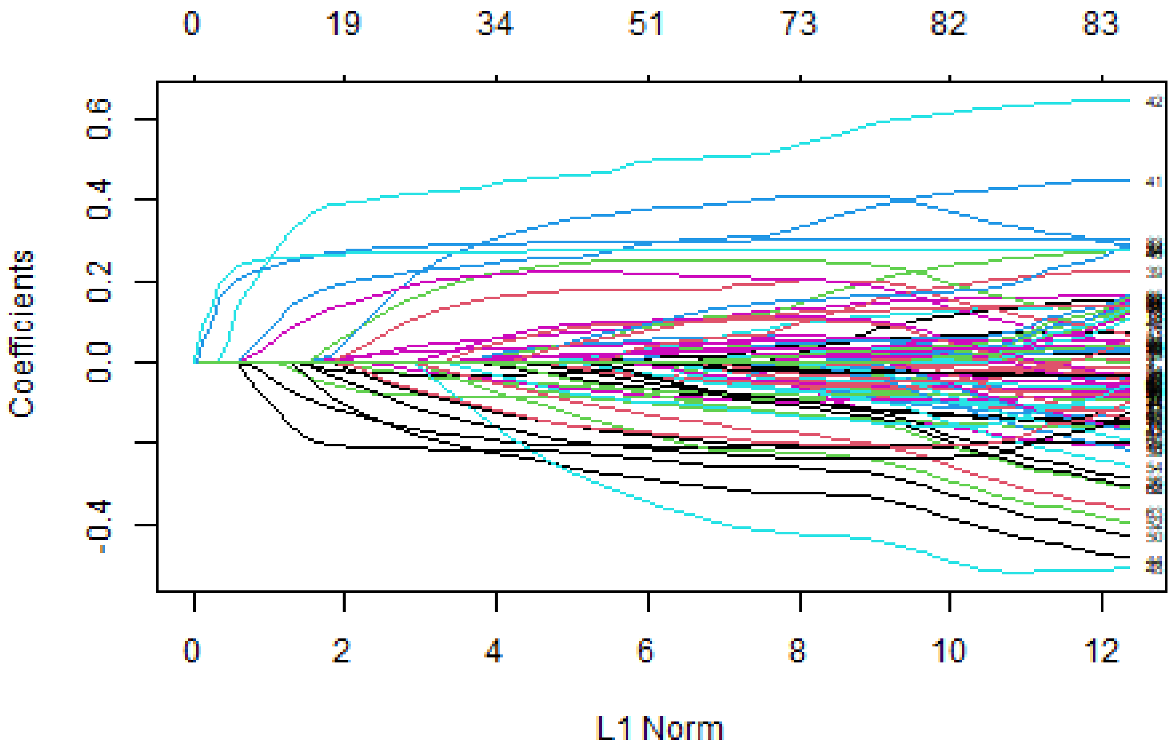
We can see that the value of MSE with lambda = 0 is slightly lower than the bet lambda. Thus, we can conclude that the model without the L2 regularization term has better predicting capabilities.

Lasso Regression

The Lasso Regression was built using the same grid as the Ridge regression.

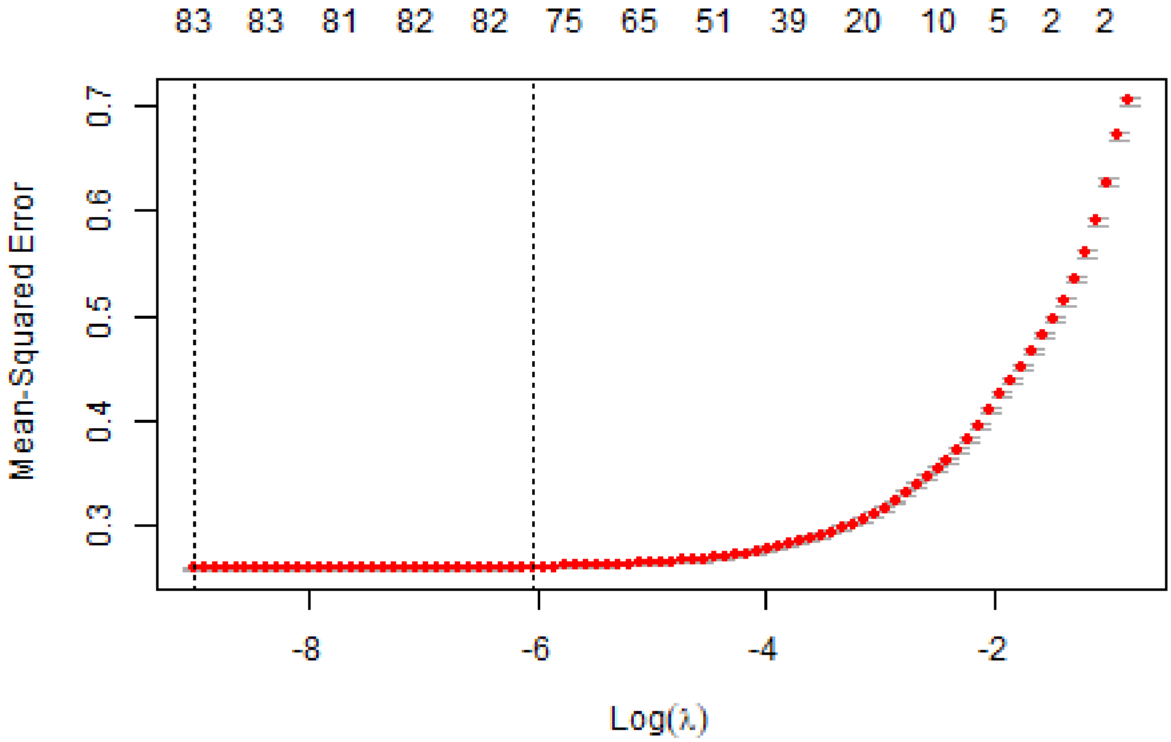
```
lasso.mod <- glmnet(X[train,],y[train],alpha=1,lambda=grid)
plot(lasso.mod, label=TRUE)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



To select the best value of lambda the cross validation technique was used.

```
set.seed(1)
cv.out <- cv.glmnet(X[train,], y[train], alpha=1, nfold=10, type.measure = "mse")
plot(cv.out)
```



Similarly to the ridge regression, the value of MSE was lower with the lower lambda.

Best lambda:

```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.0001216411
```

MSE with the best lambda:

```
lasso.pred <- predict(lasso.mod, s=bestlam, newx=X[train,])
lasso.pred.new <- predict(lasso.mod, s=bestlam, newx=X[test,])
mean((lasso.pred-y[train])^2)
```

```
## [1] 0.2581568
```

```
mean((lasso.pred.new-y.test)^2)
```

```
## [1] 0.2591973
```

MSE with lambda = 0:

```
lasso.pred <- predict(lasso.mod, s=0, newx=X[train,])
lasso.pred.new <- predict(lasso.mod, s=0, newx=X[test,])
mean((lasso.pred-y[train])^2)
```

```
## [1] 0.258149
```

```
mean((lasso.pred.new-y.test)^2)
```

```
## [1] 0.2591872
```

The best lambda is almost 0 and we can see the MSE with lambda = 0 is slightly lower than the MSE with the best lambda. This means that the L1 norm regularization also didn't bring any improvement to the model.

Let's check the variables eliminated by the model with the best lambda:

```
i <-99
lasso.mod$lambda[i]
```

```
## [1] 0.0001176812
```

```
beta.L <- coef(lasso.mod)[,i]
beta.L[beta.L == 0]
```

```
## nativity2
##         0
```

The Lasso Regression with the best lambda eliminates only the variable nativity2, which was equal to 6.403e-05 even with the full model. The result is consistent with the one obtained using forward and backward elimination techniques.

```
i <-50
lasso.mod$lambda[i]
```

```
## [1] 0.3430469
```

```
beta.L <- coef(lasso.mod)[,i]
beta.L[beta.L != 0]
```

```
## (Intercept)    wkswork1    uhrswork
## 10.47055346  0.04707381  0.08388096
```

By choosing 50th lambda, which is equal to 0.343, we can see that except the intercept, 2 most important variables are *wkswork1* and *uhrswork*. Thus, we will fit a polynomial regression using these variables.

Polynomial regression

```
pol.out <- lm(log(realincwage) ~ . -classwkr +I(uhrswork^2) + I(wkswork1^2), data = data.reg[train,])
summary(pol.out)$r.sq
```

```
## [1] 0.6627389
```

We can see that adding 2 square terms increased the R-squared from 0.63 to 0.66.

In the data visualization stage we noticed that the income difference between males and females was less for never married people compared to married. To take into account this effect, we will add interaction between marital status and sex.

```
pol.out2 <- lm(log(realincwage) ~ . -classwkr +I(uhrswork^2) + I(wkswork1^2) +sex:marst, data = data.reg
summary(pol.out2)$r.sq
```

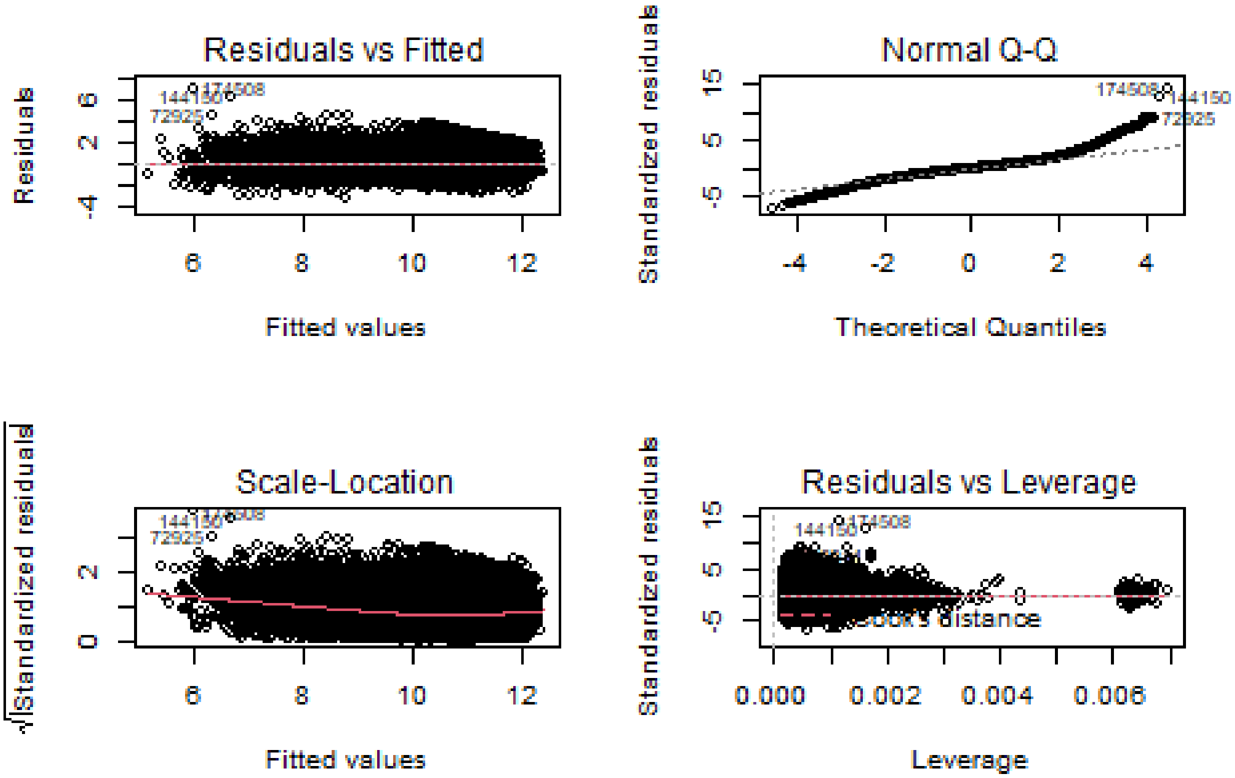
```
## [1] 0.6639231
```

All added dummy variables are significant. The adjusted R-squared increased slightly from 0.6627 to 0.6639. We will carry out the ANOVA test to compare the 2 models.

```
anova(pol.out, pol.out2)
```

```
## Analysis of Variance Table
##
## Model 1: log(realincwage) ~ (year + numprec + region + metro + relate +
##     age + sex + race + marst + nativity + sch + occupation +
##     industry + classwkr + union + wkswork1 + uhrswork) - classwkr +
##     I(uhrswork^2) + I(wkswork1^2)
## Model 2: log(realincwage) ~ (year + numprec + region + metro + relate +
##     age + sex + race + marst + nativity + sch + occupation +
##     industry + classwkr + union + wkswork1 + uhrswork) - classwkr +
##     I(uhrswork^2) + I(wkswork1^2) + sex:marst
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1 185377 44091
## 2 185372 43936  5    154.81 130.63 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value associated with the F-statistic is almost 0, thus we can conclude that the model with the interaction effect provides better fit to the data.



The residuals behave normally as the line on Residuals Vs Fitted plot is almost straight. The Normal Q-Q plot still shows that the income distribution has fat tails and our model is not able to correctly capture this data.

```
pol.pred <- predict(pol.out2)
pol.pred.new <- predict(pol.out2, newdata = data.reg[test, ])
mean((pol.pred-y[train])^2)
```

```
## [1] 0.2368999
```

```
mean((pol.pred.new-y.test)^2)
```

```
## [1] 0.2369433
```

The MSE on the test set reduced from 0.259 to 0.2369 compared to the multiple regression model without the squared and interaction terms.

Results

Finally, we interpret the results achieved by the best model.

```
summary(pol.out2)
```

```
##
## Call:
## lm(formula = log(realincwage) ~ . - classwkr + I(uhrswork^2) +
##     I(wkswork1^2) + sex:marst, data = data.reg[train, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4074 -0.2828  0.0024  0.2786  6.9179
##
## Coefficients:
##                                    Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                      10.4793821  0.0198332  528.375  < 2e-16 ***
## year                              0.0217263  0.0011527   18.847  < 2e-16 ***
## numprec                           0.0099931  0.0013443    7.434 1.06e-13 ***
## region12                          0.0030985  0.0051621    0.600  0.54835
## region21                         -0.0747323  0.0048998  -15.252  < 2e-16 ***
## region22                         -0.1129350  0.0048785  -23.150  < 2e-16 ***
## region31                         -0.0574295  0.0045717  -12.562  < 2e-16 ***
## region32                         -0.1551274  0.0064840  -23.925  < 2e-16 ***
## region33                         -0.1210277  0.0053580  -22.588  < 2e-16 ***
## region41                         -0.0681649  0.0050516  -13.494  < 2e-16 ***
## region42                          0.0303963  0.0048008    6.331 2.43e-10 ***
## metro2                            0.1322279  0.0038034   34.766  < 2e-16 ***
## metro3                            0.1583508  0.0033528   47.229  < 2e-16 ***
## metro4                            0.0712937  0.0038052   18.736  < 2e-16 ***
## relate201                        -0.0055949  0.0029135   -1.920  0.05482 .
## relate301                        -0.1507776  0.0067432  -22.360  < 2e-16 ***
## relate501                        -0.0712739  0.0106027   -6.722 1.79e-11 ***
## relate701                        -0.1120416  0.0123720   -9.056  < 2e-16 ***
## relate901                        -0.2058285  0.0376998   -5.460 4.78e-08 ***
## relate1001                       -0.1536909  0.0113709  -13.516  < 2e-16 ***
## relate1114                       -0.0282192  0.0068803   -4.101 4.11e-05 ***
## relate1115                       -0.0844717  0.0098787   -8.551  < 2e-16 ***
## relate1241                       -0.1188903  0.0217422   -5.468 4.55e-08 ***
## relate1260                       -0.0987726  0.0159717   -6.184 6.25e-10 ***
## age                               0.0590428  0.0012669   46.604  < 2e-16 ***
## sex2                             -0.2359243  0.0033765  -69.873  < 2e-16 ***
## race2                            -0.0841922  0.0041451  -20.311  < 2e-16 ***
## race3                            -0.0994951  0.0043207  -23.027  < 2e-16 ***
## race4                            -0.0434441  0.0051100   -8.502  < 2e-16 ***
## marst2                           -0.1025209  0.0134407   -7.628 2.40e-14 ***
## marst3                           -0.1085914  0.0122402   -8.872  < 2e-16 ***
## marst4                           -0.0727668  0.0060564  -12.015  < 2e-16 ***
## marst5                           -0.1172408  0.0209568   -5.594 2.22e-08 ***
## marst6                           -0.1621284  0.0050806  -31.911  < 2e-16 ***
## nativity2                         0.0079668  0.0089125    0.894  0.37138
## nativity3                          0.0286567  0.0087205    3.286  0.00102 **
## nativity4                          0.0446680  0.0069519    6.425 1.32e-10 ***
## nativity5                         -0.0881275  0.0040466  -21.778  < 2e-16 ***
## schfsch                            0.1599876  0.0049855   32.091  < 2e-16 ***
## schscol                            0.2358567  0.0054581   43.212  < 2e-16 ***
## schasoc                            0.2805314  0.0059375   47.247  < 2e-16 ***
## schbach                            0.4593953  0.0056525   81.273  < 2e-16 ***
## schadvd                            0.6579806  0.0064887  101.405  < 2e-16 ***
## occupationartist                  -0.1650380  0.0122544  -13.468  < 2e-16 ***
## occupationbuilding                -0.4671952  0.0100893  -46.306  < 2e-16 ***
## occupationbusiness                -0.0579550  0.0106672   -5.433 5.55e-08 ***
## occupationcomputer                 0.0830492  0.0099926    8.311  < 2e-16 ***
## occupationconstructextractinstall -0.1952505  0.0089270  -21.872  < 2e-16 ***
## occupationfarmer                  -0.4436144  0.0215886  -20.549  < 2e-16 ***
## occupationfinancialop             -0.0484549  0.0106096   -4.567 4.95e-06 ***
## occupationfoodcare                -0.4036182  0.0097064  -41.583  < 2e-16 ***
## occupationhealthcare               0.0679644  0.0098741    6.883 5.87e-12 ***
## occupationhealthsupport           -0.3872478  0.0114811  -33.729  < 2e-16 ***
## occupationlawyerphysician          0.3400662  0.0133143   25.541  < 2e-16 ***
## occupationlegaleduc               -0.2710609  0.0097193  -27.889  < 2e-16 ***
## occupationmanager                  0.0392808  0.0083254    4.718 2.38e-06 ***
## occupationofficeadmin             -0.2913070  0.0084197  -34.598  < 2e-16 ***
## occupationpostseceduc             -0.1037370  0.0143819   -7.213 5.49e-13 ***
## occupationproduction              -0.3261310  0.0088895  -36.687  < 2e-16 ***
## occupationprotective              -0.2062335  0.0110616  -18.644  < 2e-16 ***
## occupationsales                   -0.1898741  0.0089297  -21.263  < 2e-16 ***
## occupationscientist               -0.0762701  0.0129549   -5.887 3.93e-09 ***
## occupationsocialworker            -0.3179691  0.0115205  -27.600  < 2e-16 ***
## occupationtransport               -0.3639019  0.0093001  -39.129  < 2e-16 ***
## industryCommunications             0.2083362  0.0188083   11.077  < 2e-16 ***
## industryDurables                   0.2177439  0.0178124   12.224  < 2e-16 ***
## industryEducation                  0.0258270  0.0179936    1.435  0.15119
## industryFinance                    0.2114350  0.0178878   11.820  < 2e-16 ***
## industryHotelsRestaurants         -0.0565786  0.0182811   -3.095  0.00197 **
## industryMedical                    0.1274450  0.0178984    7.120 1.08e-12 ***
## industryMiningConstruction         0.1944182  0.0180428   10.775  < 2e-16 ***
## industryNondurables                0.2009705  0.0179899   11.171  < 2e-16 ***
## industryProfessional               0.1621866  0.0176867    9.170  < 2e-16 ***
## industryPublicadmin                0.2118327  0.0179665   11.790  < 2e-16 ***
## industryRetailTrade               -0.0201246  0.0177675   -1.133  0.25736
## industrySocArtOther               -0.0115495  0.0178634   -0.647  0.51793
## industryTransport                  0.2413150  0.0181737   13.278  < 2e-16 ***
## industryUtilities                  0.3482010  0.0203934   17.074  < 2e-16 ***
## industryWholesaleTrade             0.1866543  0.0183172   10.190  < 2e-16 ***
## union1                            -0.0311768  0.0032244   -9.669  < 2e-16 ***
## union2                             0.1280588  0.0072987   17.545  < 2e-16 ***
## union3                             0.0243705  0.0223355    1.091  0.27522
## wkswork1                           0.0952923  0.0030348   31.400  < 2e-16 ***
## uhrswork                           0.2933571  0.0012515  234.395  < 2e-16 ***
## I(uhrswork^2)                     -0.0497824  0.0004595 -108.343  < 2e-16 ***
## I(wkswork1^2)                     -0.0484898  0.0007436  -65.205  < 2e-16 ***
## sex2:marst2                        0.0903450  0.0191695    4.713 2.44e-06 ***
## sex2:marst3                        0.0681648  0.0152885    4.459 8.26e-06 ***
## sex2:marst4                        0.0703482  0.0074495    9.443  < 2e-16 ***
## sex2:marst5                          0.0676145  0.0235747    2.868  0.00413 **
## sex2:marst6                          0.1538160  0.0061666   24.943  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4868 on 185372 degrees of freedom
## Multiple R-squared:  0.6639, Adjusted R-squared:  0.6638
## F-statistic:  4069 on 90 and 185372 DF,  p-value: < 2.2e-16
```

The year variable has a positive coefficient, which means that on average, people earn more every year. The p-value associated with region12 is larger than 0.05, which means that there's no evidence of difference between the average income in the aforementioned region and the base region. The negative coefficients for all the *relate* levels shows that on average people earn less than the head of their household, which was used as a base class. The coefficient of the sex2 variable shows the same result obtained with anova analysis, there's a significant difference between the earnings of males and females with the later earning less than the former. Regarding race, white people tend to earn more than others with the Hispanic having lower earnings on average. Interestingly, married individuals with a present spouse tend to earn more on average than other people. When taking into account the interaction effect, never married females earn on average more than married ones. The coefficient for *nativity2* is more than 0.05, which means that there's no evidence of difference in earnings of native-born people and people whose fathers are foreign, while mothers are native. However, the people whose mothers were foreign and fathers native (category 3) and people with both parents foreign born earn on average more than natives. Foreign born people themselves, however, earn less than natives on average. The positive education coefficients indicate that each subsequent level of education achieved leads to increase in average earnings. We previously fitted the model with different categories for grades finished at school and there wasn't any evidence of difference between them. Regarding occupation, architect was used as a base class and we can see that on average, only managers, healthcare workers, computer workers, lawyers and physician earn more than architects. On average, lawyers and physicians have the highest earnings, while people doing building related jobs have the lowest wage earning. Regarding industry, the p-value associated with Hotels and Restaurants, Retail Trade, Social work, arts and other services is larger than 0.05 threshold, which means that there's no evidence of difference between the mentioned classes and the base class (agriculture). Public sector workers tend to earn more than private sector workers. The positive coefficients for *uhrswork* and *wkswork1* indicate that a 1 hour increase in the usual hours worked per week increases the log of income by 0.296., while an increase in the a number of weeks worked per year increases the log of income by 0.096. The negative coefficient for *uhrswork^2* and *wkswork1^2* indicates that at some point, there's no additional income caused by working more.

Classification

Logistic Regression

Let now try to predict whether the income will be higher than 60,000 dollars given some selected covariates. In order to perform classification, linear regression cannot be applied because its predictions would range between -infinite and + infinite possibly but we are interested in the probability of "success" (i.e., either income is higher than 60k or not) which has to be in the range 0-1. For this task, thus given the Bernoulli distribution of the response variable it is necessary to apply a non-linear function that is the Logistic function.

```
data.log <- data[c("year","numprec","region", "metro", "age","sex", "race","marst","nativity","sch", "oc

data.log$year <- scale(data.log$year)
data.log$numprec <- scale(data.log$numprec)
data.log$age <- scale(data.log$age)
data.log$wkswork1 <- scale(data.log$wkswork1)
data.log$uhrswork <- scale(data.log$uhrswork)
```

```r
set.seed(1)

train <- sample(1:nrow(data.log), nrow(data.log)*0.75)
test <- (-train)

y.train <- data[train, "binaryincome"]
y.test <- data$binaryincome[test]

data.log.train <-data.log[train, ]
data.log.test <- data.log[test,]
```

We will first run a model with all the possible predictors. Then for every variable we have the estimated coefficients and their estimated st.errors. Considering that maximum likelihood estimates are asymptotically normally distributed and asymptotically unbiased, z-scores can be computed by dividing the coefficient with the estimated std.error. Then the associated P-values under 0.05 indicate that the predictors have a statistically significant relationship with the response variable in the model. In this model, we have that all predictors are highly significant except for *class_wkr*, which indicate whether a person works in the public sector or private sector. However since *class_wkr* was highly correlated with *industry*, its effect may be cancelled by *industry*. Then some levels of categorical variables are also not significant however these have to be interpreted together. R automatically created for every categorical variable n-1 dummy variables. So, for categorical variables we cannot decide to drop only levels that are non significant because their interpretation is dependent upon the other levels. Also, a small p-value alone is not so indicative, it is also important to have large effect sizes in the estimated coefficient. This is the case for the coefficient corresponding to the dummy variable of advanced degree *schadvd* extracted from the categorical variable *sch*. This tell us that having an advanced degree when compared to not having finished school changes the log odds of income greater than 60k by a multiplicative factor of exp(2.98), keeping all other predictors fixed. The interpretation of continuous variable coefficients is slightly different, let us consider as an example *uhrswork* that is the usual number of hours worked in a week. The estimated coefficient of 0.65 means that for every unit change in *uhrswork*, so for every extra hour worked, the log odds of income higher than 60k increase by a multiplicative factor of exp(0.65) given that all the other predictors are fixed. Generally, we can say that negative coefficients lead to a decrease in the probability of income higher than 60k since the odds are multiplied by a number smaller than one while if coefficients are positive, an increase of the x variable associated to the coefficient will lead to an increase in the probability of income greater than 60k. Below the table of coefficients there is the null and residual deviance. Then we have AIC, the Akaike Information Criterion which in this context is just the Residual deviance adjusted for the number of parameters in the model. AIC can be used to compare models, lower AIC scores are better. Then, the number of Fisher Scoring iterations, 9 in this case, is an indicator of how quickly the glm() function converged on the maximum likelihood estimates for the coefficients.

By examining the coefficients, we can confirm some expected results. Being a woman decrease the log(odds) of income compared to being a man and being white tends to increase them with respect to other races. Regarding education, the higher the education level the greater the log(odds) when compared to people that did not finish school. The base level for occupation is architect and since architect had one of the highest median income between occupation, we can see that also here most coefficient are negative when compared with architect occupation. The largest effect is for health support category which is an occupation that would decrease the log(odds) of high income by a multiplicative facor of exp(2.70), also for farmers and people working in the building industry the coefficient are negative and large. For the number of hours worked in a week and the number of weeks worked in a year the coefficient are respectively 0.58 and 0.65 thus as expected working more hours increases the log(odds) of greater income even if the effect size seems mild. The variable *relate* indicates the relationship to household head and in this case the base level is the household head, the coefficient are all negative and thus not being the household head decrease log(odds) of income. More interestingly, belonging to one of the following categories: married but spouse absent, separated, divorced, widowed or never married, also decreases the log(odds) of income when compared to married people with spouse present. Then, the age variable matters but its effect is smaller than for the number of hours worked or number of weeks worked, probably also because this dataset only contain information about people older than 25 years old.

```
logm1 <- glm(binaryincome ~ . , data = data.log.train, family = binomial)
summary(logm1)
```

```
## 
## Call:
## glm(formula = binaryincome ~ ., family = binomial, data = data.log.train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.2229  -0.5667  -0.2598  -0.0160   4.0886  
## 
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                       -2.717650   0.141687 -19.181  < 2e-16 ***
## year                               0.101220   0.007229  14.003  < 2e-16 ***
## numprec                            0.108927   0.008566  12.716  < 2e-16 ***
## region12                           0.113680   0.030268   3.756 0.000173 ***
## region21                          -0.261912   0.029211  -8.966  < 2e-16 ***
## region22                          -0.531736   0.029521 -18.012  < 2e-16 ***
## region31                          -0.192172   0.027007  -7.116 1.11e-12 ***
## region32                          -0.593552   0.041676 -14.242  < 2e-16 ***
## region33                          -0.464353   0.033157 -14.005  < 2e-16 ***
## region41                          -0.241575   0.030447  -7.934 2.12e-15 ***
## region42                           0.183935   0.028374   6.482 9.03e-11 ***
## metro2                             0.693162   0.024707  28.055  < 2e-16 ***
## metro3                             0.787559   0.021680  36.327  < 2e-16 ***
## metro4                             0.350574   0.024694  14.196  < 2e-16 ***
## age                                0.349010   0.007846  44.485  < 2e-16 ***
## sex2                              -0.739346   0.016918 -43.702  < 2e-16 ***
## race2                             -0.328278   0.026914 -12.197  < 2e-16 ***
## race3                             -0.417714   0.028304 -14.758  < 2e-16 ***
## race4                             -0.153139   0.031045  -4.933 8.10e-07 ***
## marst2                            -0.116425   0.069331  -1.679 0.093103 .  
## marst3                            -0.280562   0.056188  -4.993 5.94e-07 ***
## marst4                            -0.142128   0.025949  -5.477 4.32e-08 ***
## marst5                            -0.333718   0.068994  -4.837 1.32e-06 ***
## marst6                            -0.349696   0.026573 -13.160  < 2e-16 ***
## nativity2                          0.036620   0.052945   0.692 0.489151    
## nativity3                          0.103292   0.050574   2.042 0.041113 *  
## nativity4                          0.254004   0.041554   6.113 9.80e-10 ***
## nativity5                         -0.285576   0.025555 -11.175  < 2e-16 ***
## schfsch                            0.848308   0.051594  16.442  < 2e-16 ***
## schscol                            1.228954   0.052879  23.241  < 2e-16 ***
## schasoc                            1.416133   0.054256  26.101  < 2e-16 ***
## schbach                            2.159558   0.052722  40.961  < 2e-16 ***
## schadvd                            2.984996   0.055536  53.749  < 2e-16 ***
## occupationartist                  -0.878502   0.062767 -13.996  < 2e-16 ***
## occupationbuilding                -2.486380   0.090840 -27.371  < 2e-16 ***
## occupationbusiness                -0.562123   0.052940 -10.618  < 2e-16 ***
## occupationcomputer                 0.299692   0.049234   6.087 1.15e-09 ***
## occupationconstructextractinstall -0.906049   0.045295 -20.003  < 2e-16 ***
## occupationfarmer                  -2.060258   0.202573 -10.170  < 2e-16 ***
## occupationfinancialop             -0.565660   0.052745 -10.724  < 2e-16 ***
## occupationfoodcare                -1.983400   0.077065 -25.737  < 2e-16 ***
## occupationhealthcare              -0.077129   0.051980  -1.484 0.137854    
## occupationhealthsupport           -2.700880   0.123869 -21.804  < 2e-16 ***
## occupationlawyerphysician          0.052500   0.074178   0.708 0.479095    
## occupationlegaleduc               -1.311072   0.054166 -24.205  < 2e-16 ***
## occupationmanager                 -0.021222   0.041646  -0.510 0.610351    
## occupationofficeadmin             -1.777833   0.045611 -38.978  < 2e-16 ***
## occupationpostseceduc             -0.415105   0.075318  -5.511 3.56e-08 ***
## occupationproduction              -1.467142   0.047980 -30.578  < 2e-16 ***
## occupationprotective              -0.782126   0.056255 -13.903  < 2e-16 ***
## occupationsales                   -0.741175   0.046357 -15.989  < 2e-16 ***
## occupationscientist               -0.601402   0.063510  -9.469  < 2e-16 ***
## occupationsocialworker            -1.900735   0.068070 -27.923  < 2e-16 ***
## occupationtransport               -1.642998   0.052687 -31.184  < 2e-16 ***
## industryCommunications             1.182364   0.131717   8.977  < 2e-16 ***
## industryDurables                   1.163605   0.128149   9.080  < 2e-16 ***
## industryEducation                  0.140110   0.130372   1.075 0.282510    
## industryFinance                    1.020324   0.128291   7.953 1.82e-15 ***
## industryHotelsRestaurants         -0.331360   0.137782  -2.405 0.016175 *  
## industryMedical                    0.589736   0.129220   4.564 5.02e-06 ***
## industryMiningConstruction         1.213961   0.128864   9.420  < 2e-16 ***
## industryNondurables                1.211913   0.129505   9.358  < 2e-16 ***
## industryProfessional               0.987700   0.127582   7.742 9.81e-15 ***
## industryPublicadmin                1.117551   0.130046   8.593  < 2e-16 ***
## industryRetailTrade                0.274752   0.129042   2.129 0.033241 *  
## industrySocArtOther                0.062965   0.130197   0.484 0.628663    
## industryTransport                  1.255618   0.130153   9.647  < 2e-16 ***
## industryUtilities                  1.947533   0.136864  14.230  < 2e-16 ***
## industryWholesaleTrade             0.956355   0.130970   7.302 2.83e-13 ***
## classwkrPublic sector              0.016173   0.028179   0.574 0.566006    
## wkswork1                           0.586096   0.014869  39.416  < 2e-16 ***
## uhrswork                           0.657408   0.008365  78.594  < 2e-16 ***
## union1                            -0.090460   0.020023  -4.518 6.25e-06 ***
## union2                             0.442180   0.038859  11.379  < 2e-16 ***
## union3                             0.025612   0.129483   0.198 0.843201    
## relate201                         -0.061299   0.016818  -3.645 0.000268 ***
## relate301                         -0.994348   0.059950 -16.586  < 2e-16 ***
## relate501                         -0.429802   0.076810  -5.596 2.20e-08 ***
## relate701                         -0.720166   0.112367  -6.409 1.46e-10 ***
## relate901                         -1.186949   0.479506  -2.475 0.013310 *  
## relate1001                        -0.951965   0.103306  -9.215  < 2e-16 ***
## relate1114                        -0.202625   0.047931  -4.227 2.36e-05 ***
## relate1115                        -0.554162   0.073019  -7.589 3.22e-14 ***
## relate1241                        -0.916512   0.178196  -5.143 2.70e-07 ***
## relate1260                        -0.770325   0.134585  -5.724 1.04e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 207430  on 185462  degrees of freedom
## Residual deviance: 132045  on 185378  degrees of freedom
## AIC: 132215
##
```

```
## Number of Fisher Scoring iterations: 6
```

Let's try now to fit a second model without considering the variable *classwkr* that had non significant coefficient. By removing this covariate, all the variables have significant coefficient thus the data supports the fact that all the regressors included now are relevant for having income greater than 60,000 dollars.

```
logm2 <- glm(binaryincome ~.-classwkr , data = data.log.train, family = binomial)
summary(logm2)
```

```
##
## Call:
## glm(formula = binaryincome ~ . - classwkr, family = binomial,
##     data = data.log.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2232  -0.5666  -0.2599  -0.0160   4.0885
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -2.714230   0.141559 -19.174  < 2e-16 ***
## year                              0.101144   0.007227  13.995  < 2e-16 ***
## numprec                           0.108931   0.008566  12.716  < 2e-16 ***
## region12                          0.113907   0.030266   3.764 0.000168 ***
## region21                         -0.261775   0.029211  -8.962  < 2e-16 ***
## region22                         -0.531341   0.029513 -18.004  < 2e-16 ***
## region31                         -0.191794   0.026999  -7.104 1.21e-12 ***
## region32                         -0.593169   0.041670 -14.235  < 2e-16 ***
## region33                         -0.464025   0.033152 -13.997  < 2e-16 ***
## region41                         -0.241026   0.030432  -7.920 2.37e-15 ***
## region42                          0.184429   0.028361   6.503 7.88e-11 ***
## metro2                            0.692495   0.024680  28.059  < 2e-16 ***
## metro3                            0.786986   0.021657  36.339  < 2e-16 ***
## metro4                            0.350251   0.024688  14.187  < 2e-16 ***
## age                               0.349258   0.007834  44.584  < 2e-16 ***
## sex2                             -0.739382   0.016918 -43.704  < 2e-16 ***
## race2                            -0.327772   0.026899 -12.185  < 2e-16 ***
## race3                            -0.417447   0.028300 -14.751  < 2e-16 ***
## race4                            -0.152715   0.031036  -4.921 8.63e-07 ***
## marst2                           -0.116397   0.069330  -1.679 0.093173 .
## marst3                           -0.280544   0.056188  -4.993 5.95e-07 ***
## marst4                           -0.142129   0.025949  -5.477 4.32e-08 ***
## marst5                           -0.333889   0.068994  -4.839 1.30e-06 ***
## marst6                           -0.349723   0.026573 -13.161  < 2e-16 ***
## nativity2                         0.036549   0.052944   0.690 0.489985
## nativity3                         0.103330   0.050572   2.043 0.041031 *
## nativity4                         0.253875   0.041553   6.110 9.99e-10 ***
## nativity5                        -0.285952   0.025547 -11.193  < 2e-16 ***
## schfsch                           0.848663   0.051590  16.450  < 2e-16 ***
## schscol                           1.229332   0.052876  23.249  < 2e-16 ***
## schasoc                           1.416426   0.054254  26.107  < 2e-16 ***
## schbach                           2.159972   0.052718  40.972  < 2e-16 ***
## schadvd                           2.985777   0.055520  53.778  < 2e-16 ***
## occupationartist                 -0.879348   0.062751 -14.013  < 2e-16 ***
## occupationbuilding               -2.485890   0.090835 -27.367  < 2e-16 ***
## occupationbusiness                -0.562311   0.052939 -10.622  < 2e-16 ***
## occupationcomputer                 0.299135   0.049225   6.077 1.23e-09 ***
## occupationconstructextractinstall -0.906706   0.045281 -20.024  < 2e-16 ***
## occupationfarmer                  -2.062387   0.202556 -10.182  < 2e-16 ***
## occupationfinancialop             -0.566214   0.052736 -10.737  < 2e-16 ***
## occupationfoodcare                -1.983761   0.077062 -25.742  < 2e-16 ***
## occupationhealthcare              -0.077441   0.051977  -1.490 0.136246    
## occupationhealthsupport           -2.701318   0.123869 -21.808  < 2e-16 ***
## occupationlawyerphysician          0.052158   0.074176   0.703 0.481953    
## occupationlegaleduc               -1.309130   0.054057 -24.218  < 2e-16 ***
## occupationmanager                 -0.021853   0.041632  -0.525 0.599647    
## occupationofficeadmin             -1.777474   0.045605 -38.975  < 2e-16 ***
## occupationpostseceduc             -0.416629   0.075275  -5.535 3.12e-08 ***
## occupationproduction              -1.467591   0.047974 -30.591  < 2e-16 ***
## occupationprotective              -0.782010   0.056255 -13.901  < 2e-16 ***
## occupationsales                   -0.741835   0.046342 -16.008  < 2e-16 ***
## occupationscientist               -0.600933   0.063505  -9.463  < 2e-16 ***
## occupationsocialworker            -1.900332   0.068070 -27.917  < 2e-16 ***
## occupationtransport               -1.644653   0.052615 -31.258  < 2e-16 ***
## industryCommunications             1.179791   0.131639   8.962  < 2e-16 ***
## industryDurables                   1.160553   0.128037   9.064  < 2e-16 ***
## industryEducation                  0.147263   0.129773   1.135 0.256472    
## industryFinance                    1.017582   0.128200   7.937 2.06e-15 ***
## industryHotelsRestaurants         -0.334173   0.137693  -2.427 0.015226 *  
## industryMedical                    0.587811   0.129176   4.550 5.35e-06 ***
## industryMiningConstruction         1.212152   0.128824   9.409  < 2e-16 ***
## industryNondurables                1.208749   0.129386   9.342  < 2e-16 ***
## industryProfessional               0.985079   0.127499   7.726 1.11e-14 ***
## industryPublicadmin                1.129881   0.128262   8.809  < 2e-16 ***
## industryRetailTrade                0.271794   0.128937   2.108 0.035035 *  
## industrySocArtOther                0.060924   0.130147   0.468 0.639703    
## industryTransport                  1.256286   0.130148   9.653  < 2e-16 ***
## industryUtilities                  1.948443   0.136855  14.237  < 2e-16 ***
## industryWholesaleTrade             0.953449   0.130870   7.285 3.21e-13 ***
## wkswork1                           0.586164   0.014869  39.423  < 2e-16 ***
## uhrswork                           0.657313   0.008363  78.600  < 2e-16 ***
## union1                            -0.090830   0.020013  -4.539 5.66e-06 ***
## union2                             0.443620   0.038778  11.440  < 2e-16 ***
## union3                             0.027359   0.129454   0.211 0.832623    
## relate201                         -0.061370   0.016818  -3.649 0.000263 ***
## relate301                         -0.994377   0.059949 -16.587  < 2e-16 ***
## relate501                         -0.429727   0.076809  -5.595 2.21e-08 ***
## relate701                         -0.720431   0.112369  -6.411 1.44e-10 ***
## relate901                         -1.186747   0.479525  -2.475 0.013330 *  
## relate1001                        -0.952129   0.103309  -9.216  < 2e-16 ***
## relate1114                        -0.202477   0.047930  -4.224 2.40e-05 ***
## relate1115                        -0.554248   0.073017  -7.591 3.18e-14 ***
## relate1241                        -0.916714   0.178200  -5.144 2.69e-07 ***
## relate1260                        -0.770639   0.134588  -5.726 1.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
```

```
##     Null deviance: 207430  on 185462  degrees of freedom
## Residual deviance: 132045  on 185379  degrees of freedom
## AIC: 132213
##
## Number of Fisher Scoring iterations: 6
```

The AIC is slightly lower, from 132215.2 to 132213.5, indicating the second model as better. This is because it requires less number of predictors but reaches almost the same level of precision.

```
logm1$aic
```

```
## [1] 132215.2
```

```
logm2$aic
```

```
## [1] 132213.5
```

Since the second model is a reduced version of the full model, we can compare these nested models using the anova function:

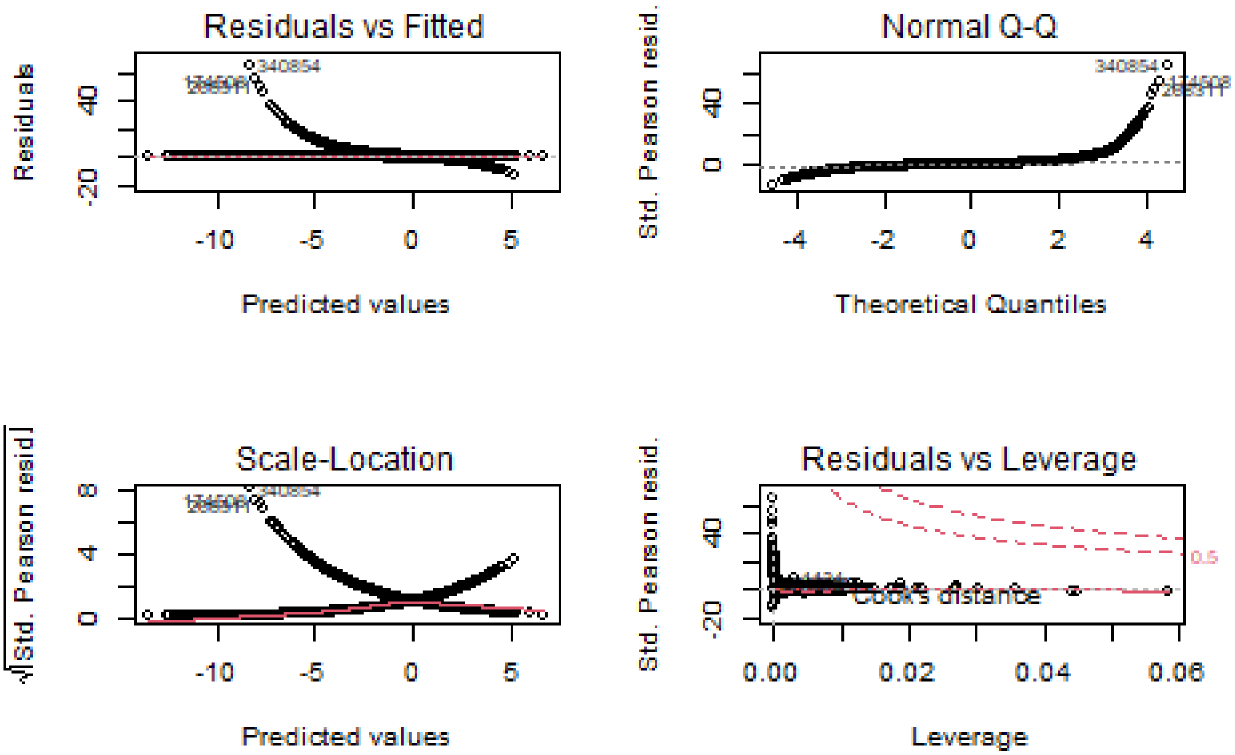
```
anova(logm1, logm2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: binaryincome ~ year + numprec + region + metro + age + sex +
##     race + marst + nativity + sch + occupation + industry + classwkr +
##     wkswork1 + uhrswork + union + relate
## Model 2: binaryincome ~ (year + numprec + region + metro + age + sex +
##     race + marst + nativity + sch + occupation + industry + classwkr +
##     wkswork1 + uhrswork + union + relate) - classwkr
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1    185378     132045
## 2    185379     132045 -1 -0.32926   0.5661
```

This p-value is the same of the p-value obtained in the full model for the *classwkr* variable because the sample size is very large. However we checked the result of ANOVA function because it is a more reliable approximation. The null hypothesis of this test is that the coefficient of *classwkr* is zero and given the large p-value it is not possible to reject this hypothesis. In fact, we also noticed that the coefficients of *industry* that is the variable that has the highest correlation with *classwkr* remains unchanged after removing this variable.

Let's now plot the residuals of the full model:

```
par(mfrow=c(2,2))
plot(logm1)
```



Let's have a look at the confusion matrix of the first model with all possible predictors: '

```
logistic.prob <- predict(logm1, type="response") #link is for logit, response for prob
logistic.pred.train <- rep(0, dim(data.log.train)[1])
logistic.pred.train[logistic.prob>0.5] <- 1
table(logistic.pred.train, y.train)
```

```
##                    y.train
## logistic.pred.train      0      1
##                   0 129591  20072
##                   1  10028  25772
```

Let's have a look at the confusion matrix with the second reduced model:

```
logistic.prob2 <- predict(logm2,  type="response") #link is for logit, response for prob
logistic.pred.train2 <- rep(0, dim(data.log.train)[1])
logistic.pred.train2[logistic.prob2>0.5] <- 1
table(logistic.pred.train2, y.train)
```

```
##                     y.train
## logistic.pred.train2      0      1
##                    0 129595  20066
##                    1  10024  25778
```

As we can see from these two table, the second model without the variable *classwkr* produce even more correct classifications. However the difference is barely noticeable: the training error rate is for both 16%. For both models, 2/3 of these misclassifications is linked to observations that were classified as high income and the model wrongly predicted them as lower than 60,000 dollars. Given the imbalanced dataset, that contain more data for income lower than 60k, this type of behavior in favor of false negative classifications was expected. However of course these predictions are quite optimistic, because we are making predictions of the response variable on same data used to train the model and because of the data imbalance this type of metric may be misleading. In fact a trivial classifier that always predict zero as response variable would have a training error of 24% which would generally be considered small error rate.

```
# overall (training) error rate
(20072+10028)/dim(data.log.train)[1]
```

```
## [1] 0.1622965
```

```
(20066+10024)/dim(data.log.train)[1]
```

```
## [1] 0.1622426
```

In our case, our purpose is simply to build a reliable classifier that predicts whether the income is higher than 60,000. In some other problems given the nature of the classification either minimizing the number of false positive or minimizing the number of false positive could be preferred. In this statistical analysis, we want to have a balanced number of false positives and false negatives. False negative rate for full and reduced model:

```
20072/(20072+25772)
```

```
## [1] 0.4378326
```

```
20066/(20066+25778)
```

```
## [1] 0.4377018
```

Let imagine a use case for a classifier that predicts high income, perhaps it could be used by banks or investment funds to identify potential wealthy customers. With a false negative rate of almost 43% for both models, this classifier is not anymore reliable for the purpose of this task. A solution to this problem is to change the threshold, i.e. changing the classification rule that was previously 0.5 to a lower threshold of 0.3. By printing the error rates for the first model we see that now the False negative rate decreased while the general training error increased slightly. Clearly 0.5 is the best threshold to minimize the overall training error rate.

```
logistic.prob1 <- predict(logm1,  type="response") #link is for logit, response for prob
logistic.pred.train1 <- rep(0, dim(data.log.train)[1])
logistic.pred.train1[logistic.prob1>0.3] <- 1
table(logistic.pred.train1, y.train)
```

```
##                     y.train
## logistic.pred.train1      0      1
##                    0 116187  10970
##                    1  23432  34874
```

FNR:

```
10970/(10970+34874)
```

```
## [1] 0.2392898
```

Overall training error rate:

```
(10970+ 23432)/dim(data.log.train)[1]
```

```
## [1] 0.1854925
```

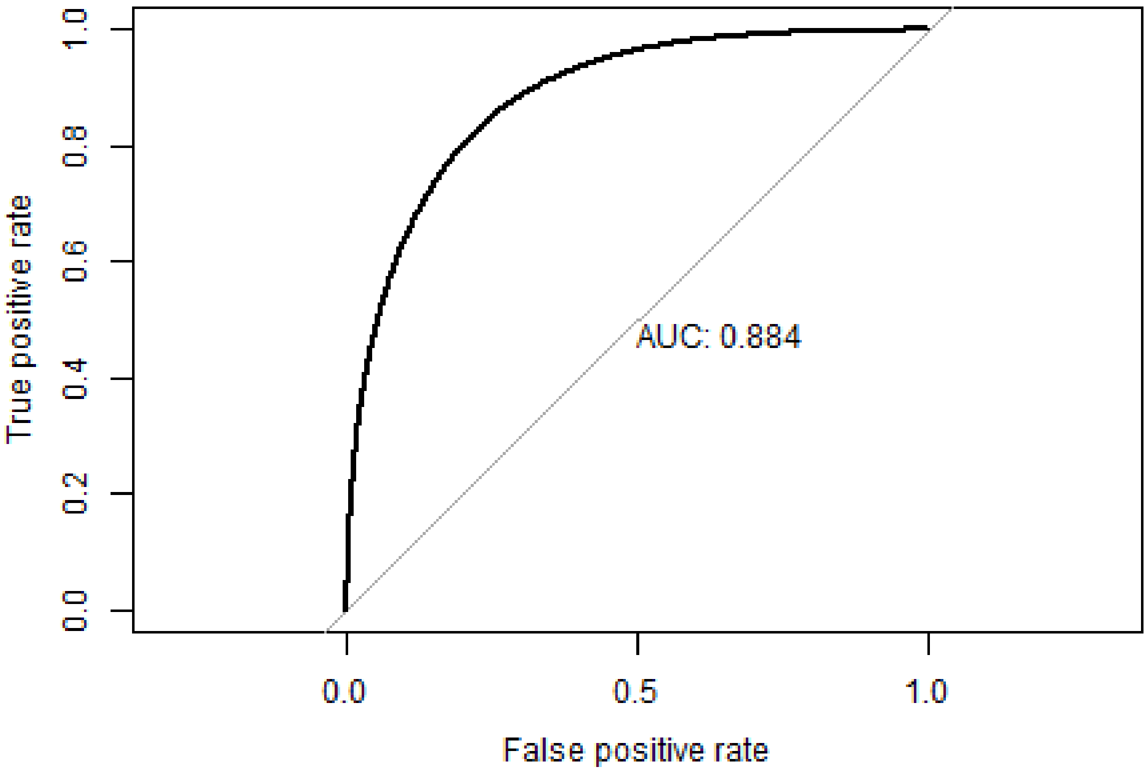
Since a model with less predictors and the same predictive power should be preferred in order to reduce variability, we are going to continue the analysis on the model without *classwkr*.

Let's look at the ROC curve which shows the general behavior of a classifier without the need of specifying a threshold. The area under the curve is 88%, definitively better than a random classifier.

```
roc.out <- roc(y.train, logistic.prob2, levels=c('0', '1'))
```

```
## Setting direction: controls < cases
```

```
plot(roc.out, print.auc=TRUE, legacy.axes=TRUE, xlab="False positive rate", ylab="True positive rate")
```



```
coords(roc.out, "best")
```

```
##   threshold specificity sensitivity
## 1 0.2370178   0.7798795   0.8227685
```

By using a threshold that maximizes the sum of specificity and sensitivity we can reach more balance between TPR and FNR:

```
logistic.prob2 <- predict(logm2,  type="response") #link is for logit, response for prob
logistic.pred.train2 <- rep(0, dim(data.log.train)[1])
logistic.pred.train2[logistic.prob1>0.2370178   ] <- 1
table(logistic.pred.train2, y.train)
```

```
##                       y.train
## logistic.pred.train2      0      1
##                     0 108872   8137
##                     1  30747  37707
```

```
tp = 37707
tn = 108872
fp = 30747
```

```
fn = 8137
```

FPR:

```
fp/(fp+tn) #FP/FP+TN
```

```
## [1] 0.2202207
```

FNR:

```
fn/(fn+tp) #FN/FN+TP
```

```
## [1] 0.1774932
```

However the ROC curve is also biased in this case because it considers specificity that is always going to be high in this case given the imbalanced dataset towards the 0 class. A better metrics for imbalanced data is F1-Measure which is the harmonic mean of precision and recall.

```
prec = tp/(tp+fp)
recall = tp/(tp+fn)
f1 = 2*(prec*recall)/(prec+recall)
f1
```

```
## [1] 0.6598016
```

Now, let's check the predictions on the hold-out set:

```
logistic.prob2 <- predict(logm2, data.log.test, type="response")
logistic.pred.test2 <- rep(0, dim(data.log.test)[1])
logistic.pred.test2[logistic.prob2>0.2370178] <- 1
table(logistic.pred.test2, y.test)
```

```
##                     y.test
## logistic.pred.test2     0     1
##                   0 36295  2703
##                   1 10336 12487
```

The F1 score is basically the same on the test set, which is a good sign since it means there is no overfitting.

```
tp =12487
fp = 10336
tn = 36295
fn = 2703
prec = tp/(tp+fp)
recall = tp/(tp+fn)
f1 = 2*(prec*recall)/(prec+recall)
f1
```

```
## [1] 0.6569858
```

Linear discriminant analysis

Let now try with a different supervised classification technique, Linear Discriminant analysis (LDA). This classifier uses the Bayes theorem to make classifications.

```
lda.fit <- lda(binaryincome ~ ., data = data.log, subset=train)
lda.fit
```

```
## Call:
## lda(binaryincome ~ ., data = data.log, subset = train)
##
```

```
## Prior probabilities of groups:
##         0         1 
## 0.7528132 0.2471868 
## 
## Group means:
##          year     numprec   region12  region21   region22  region31   region32
## 0 -0.02390973  0.00395408 0.09049628 0.1187159 0.12507610 0.1723118 0.04940588
## 1  0.07480830 -0.01841130 0.11454062 0.1154786 0.09735189 0.1856295 0.03154175
##     region33   region41  region42    metro2    metro3    metro4        age
## 0 0.09603994 0.10965556 0.1437269 0.2437992 0.3634391 0.1802477 -0.0666405
## 1 0.07071809 0.09514877 0.1678082 0.2452884 0.4816116 0.1592793  0.2077156
##        sex2      race2      race3      race4      marst2     marst3    marst4
## 0 0.5543157 0.11071559 0.18574120 0.06902356 0.016000688 0.02824114 0.1236221
## 1 0.3127563 0.06681354 0.07662944 0.08459122 0.009532327 0.01308786 0.0940363
##       marst5    marst6  nativity2  nativity3  nativity4 nativity5   schfsch
## 0 0.01578582 0.2010615 0.01607947 0.01585028 0.03093419 0.1945939 0.3399824
## 1 0.00857255 0.1128828 0.01897740 0.02235843 0.03518454 0.1377498 0.1333653
##     schscol    schasoc   schbach    schadvd occupationartist occupationbuilding
## 0 0.1901604 0.11596559 0.1854117 0.07155187       0.01297101        0.046096878
## 1 0.1242911 0.09493063 0.3539613 0.28239246       0.01766862        0.003686415
##   occupationbusiness occupationcomputer occupationconstructextractinstall
## 0         0.01913063         0.01432470                        0.09048195
## 1         0.03477009         0.07484076                        0.07843993
##   occupationfarmer occupationfinancialop occupationfoodcare
## 0      0.008107779            0.01967497        0.082202279
## 1      0.000828898            0.03956897        0.006587558
##   occupationhealthcare occupationhealthsupport occupationlawyerphysician
## 0            0.0492125             0.031800829               0.003337655
## 1            0.0699110             0.001854114               0.038870954
##   occupationlegaleduc occupationmanager occupationofficeadmin
## 0          0.07491101        0.06496967            0.17135920
## 1          0.04626560        0.23582148            0.04881773
##   occupationpostseceduc occupationproduction occupationprotective
## 0           0.008208052           0.08271797           0.02098568
## 1           0.017319606           0.03778030           0.03394119
##   occupationsales occupationscientist occupationsocialworker
## 0      0.08950071         0.008007506             0.02186665
## 1      0.08441672         0.022991013             0.01147369
##   occupationtransport industryCommunications industryDurables industryEducation
## 0          0.06893045             0.01988268       0.07328515        0.11559315
## 1          0.03173807             0.03884914       0.10557543        0.09305471
##   industryFinance industryHotelsRestaurants industryMedical
## 0      0.06064361                0.06342976       0.1254127
## 1      0.09305471                0.01081930       0.1075386
##   industryMiningConstruction industryNondurables industryProfessional
## 0                 0.06370909          0.04780152            0.0835488
## 1                 0.06853678          0.05102085            0.1416543
##   industryPublicadmin industryRetailTrade industrySocArtOther industryTransport
## 0          0.05340964          0.11176846          0.08819000        0.04507982
## 1          0.09981677          0.05270046          0.03357037        0.04336445
##   industryUtilities industryWholesaleTrade classwkrPublic sector    wkswork1
## 0       0.007556278             0.03017498             0.1842299 -0.08165613
## 1       0.021616787             0.03605706             0.2111290  0.25367425
##     uhrswork    union1     union2      union3 relate201  relate301   relate501
## 0 -0.1677860 0.1485829 0.02331345 0.002535472 0.3156590 0.04283085 0.013780359
## 1  0.5104912 0.1426141 0.03232702 0.002704825 0.3336969 0.00927057 0.006565745
##     relate701    relate901  relate1001 relate1114  relate1115  relate1241
## 0 0.010944069 0.0011674629 0.013028313 0.03561120 0.016638137 0.003316168
## 1 0.002508507 0.0001090655 0.003141087 0.01851933 0.006892941 0.001047029
##    relate1260
## 0 0.006288542
## 1 0.001810488
##
## Coefficients of linear discriminants:
##                                             LD1
## year                               5.490282e-02
## numprec                            8.701838e-02
## region12                           6.200249e-02
## region21                          -1.442596e-01
## region22                          -2.977018e-01
## region31                          -1.109248e-01
## region32                          -3.123048e-01
## region33                          -2.327130e-01
## region41                          -1.400091e-01
## region42                           1.033626e-01
## metro2                             3.434942e-01
## metro3                             4.216372e-01
## metro4                             1.602466e-01
## age                                2.080392e-01
## sex2                              -4.666083e-01
## race2                             -1.930574e-01
## race3                             -2.351256e-01
## race4                             -1.345322e-01
## marst2                            -6.304484e-02
## marst3                            -9.432547e-02
## marst4                            -1.095437e-01
## marst5                            -1.864805e-01
## marst6                            -1.847482e-01
## nativity2                          3.467358e-02
## nativity3                          8.380395e-02
## nativity4                          1.113183e-01
## nativity5                         -1.687328e-01
## schfsch                            1.277456e-01
## schscol                            3.136499e-01
## schasoc                            4.014489e-01
## schbach                            9.966779e-01
## schadvd                            1.703385e+00
## occupationartist                  -1.036137e+00
## occupationbuilding                -1.405087e+00
## occupationbusiness                -7.639873e-01
## occupationcomputer                 1.616245e-01
## occupationconstructextractinstall -1.193851e+00
## occupationfarmer                  -1.340624e+00
## occupationfinancialop             -7.501387e-01
## occupationfoodcare                -1.124088e+00
## occupationhealthcare              -5.731780e-01
## occupationhealthsupport           -1.332484e+00
## occupationlawyerphysician          1.432701e-02
## occupationlegaleduc               -1.424833e+00
## occupationmanager                 -1.913094e-01
## occupationofficeadmin             -1.495229e+00
## occupationpostseceduc             -7.447026e-01
## occupationproduction              -1.504138e+00
## occupationprotective              -8.849685e-01
## occupationsales                   -9.232283e-01
## occupationscientist               -6.997632e-01
## occupationsocialworker            -1.792539e+00
## occupationtransport               -1.514537e+00
## industryCommunications             7.016017e-01
## industryDurables                   6.547752e-01
## industryEducation                  8.334386e-05
## industryFinance                    5.412228e-01
## industryHotelsRestaurants          5.183267e-03
## industryMedical                    2.689030e-01
## industryMiningConstruction         6.130375e-01
## industryNondurables                6.803916e-01
## industryProfessional               5.030238e-01
## industryPublicadmin                6.433410e-01
## industryRetailTrade                1.399724e-01
## industrySocArtOther                8.155449e-02
## industryTransport                  6.209428e-01
## industryUtilities                  1.300569e+00
## industryWholesaleTrade             5.185936e-01
## classwkrPublic sector             -1.490326e-02
## wkswork1                           1.043309e-01
## uhrswork                           3.437992e-01
## union1                            -4.745031e-02
## union2                             2.779307e-01
## union3                            -2.451946e-02
## relate201                         -4.750774e-02
## relate301                         -3.393021e-01
## relate501                         -2.495192e-01
## relate701                         -2.770576e-01
## relate901                         -2.448525e-01
## relate1001                        -4.029447e-01
## relate1114                        -1.108935e-01
## relate1115                        -3.144017e-01
## relate1241                        -4.118518e-01
## relate1260                        -3.257041e-01
```

The LDA output indicates prior probabilities of $0 = 0.75$ and $1 = 0.25$; in other words, 75% is the proportion of people that had income lower than 60,000 in the training data and this is used to estimate the probability of sampling a respondent that belongs to this class before collecting the data. Then LDA provide group mean estimates for every predictor. Most of the means are positive, except for a few cases. Thus for the variable *year*, if the respondent's income is classified as 0, it means that the year corresponding to when the information was collected was some year before than for a respondent's income classified as 1. Then, for *numprec* which identifies the number of people in a household unit, the mean is negative for the group 1 which means that if the observation is classified as higher income, the number of people in that unit is typically lower than the number of people in a household unit for an observation classified as low income. Then for both *wkswork1*, *uhrswork*, and *age* the means are clearly negative for group 0 since in fact we expected that working less time and being younger can influence the predictions in favor of the lower class income.

Let's see the prediction on the test set:

```
lda.pred <- predict(lda.fit, data.log.test)
lda.class <- lda.pred$class
table(lda.class,y.test)
```

```
##          y.test
## lda.class     0     1
##         0 43144  6824
##         1  3487  8366
```

The overall test error rate is the same of logistic regression, while F1 Score is lower.

Overall test error rate:

```
mean(lda.class!=y.test)
```

```
## [1] 0.166788
```

False negative rate:

```
fn/(fn+tp)
```

```
## [1] 0.4492429
```

False positive rate:

```
fp/(fp+tn)
```

```
## [1] 0.07477858
```

Sensitivity:

```
tp/(tp+fn)
```

```
## [1] 0.5507571
```

Specificity:

```
tn/(tn+fp)
```

```
## [1] 0.9252214
```

F1 score:

```
prec = tp/(tp+fp)
recall = tp/(tp+fn)
f1 = 2*(prec*recall)/(prec+recall)
f1
```

```
## [1] 0.6187183
```

KNN

Let's now try binary classification with KNN, a non-parametric approach.

```
var.knn <- colnames(data.log)
var.knn
```

```
##  [1] "year"          "numprec"      "region"       "metro"        "age"
##  [6] "sex"           "race"         "marst"        "nativity"     "sch"
## [11] "occupation"    "industry"     "classwkr"     "wkswork1"     "uhrswork"
## [16] "union"         "relate"       "binaryincome"
```

```
data.knn <- data[var.knn]

for (var in var.knn) {
  data.knn[[var]] <- as.numeric(data[[var]])
}


set.seed(1)

train <- sample(1:nrow(data.knn), nrow(data.knn)*0.75)
test <- (-train)
y <- data$binaryincome
y.test <- y[test]
```

We tried different values for k which indicates the number of the nearest neighbors.

```
knn.pred <- knn(data.knn[train,], data.knn[test,], y[train], k=3)
table(knn.pred,y.test)
```

```
##         y.test
## knn.pred     0     1
##        0 42304  6417
##        1  4327  8773
```

```
mean(knn.pred==y.test)
```

```
## [1] 0.8262079
```

F1 score:

```
f1
```

```
## [1] 0.6202192
```

```
knn.pred <- knn(data.knn[train,], data.knn[test,], y[train], k=5)
table(knn.pred,y.test)
```

```
##         y.test
## knn.pred     0     1
##        0 42895  6527
##        1  3736  8663
```

```
mean(knn.pred==y.test)
```

```
## [1] 0.8339885
```

F1 score:

```
f1
```

```
## [1] 0.6280039
```

We found that the higher the k the more observations are classified as the majority class. The model with k = 5 was the best, with a F1 score of 0.62.

Conclusions

The best model for predicting income is a Multiple Regression model with squared and interaction terms. The best MSE and adjusted R-squared achieved are 0.237 and 0.6639 respectively. Based on the adjusted

R-squared, the model explains only 66.39% of variance in income, which is mainly due to the fat tails in the income distribution. Therefore, the model cannot be used to provide accurate predictions of the income. However, the linear regression model can be used to understand the factors that determine the difference in income. In our analysis, we found out that the most significant predictors of income were the variables related to the total time worked, education and sex.

To sum up, between the different classification models used (Logistic regression, LDA and KNN) the model that performed better on test data was Logistic regression first, then KNN and lastly LDA. The main metric that we used to compare these model is F1-score that is more suitable for imbalanced datasets. The F1-scores were: 0.65, 0.62 and 0.61 respectively. By using logistic regression to understand which covariates are more determinant in order to predict income we got similar results as Linear regression.

Bibliography

[1] fedesoriano. (January 2022). Gender Pay Gap Dataset. Retrieved [01/05/22] from https://www.kaggle.com/fedesoriano/gender-pay-gap-dataset.