Predicting Canadian Annual Income Through Census Data

2025-03-12

First Version: 2023-04-19
Latest Version: 2025-03-12

```
version
```

```
##                _
## platform       aarch64-apple-darwin20
## arch           aarch64
## os             darwin20
## system         aarch64, darwin20
## status
## major          4
## minor          3.2
## year           2023
## month          10
## day            31
## svn rev        85441
## language       R
## version.string R version 4.3.2 (2023-10-31)
## nickname       Eye Holes
```

```
library(haven)
library(ggplot2)
library(randomForest)
library(caret)
library(cvms)
library(ggimage)
library(rsvg)
library(ROCR)
library(pROC)
library(tibble)
library(scales)
library(dplyr)
```

Dataset and Data Preparation

2016 Census Public Use Microdata File (PUMF)

Released by Statistics Canada, the 2016 Census Public Use Microdata File (PUMF) features 123 variables on various individual and family-level socioeconomic characteristics surveyed during the 2015-2016 Census. It includes a total of 930,421 observations representing 2.7% of the entire population living within Canadian territory at the time of the census. An estimated weight is also attached to each observation to represent the total number of people in the population that share the respondent's characteristics.

```r
cdata16_origin <- read_sav('./Census_2016_Individual_PUMF.sav')
cdata16 <- cdata16_origin
# total records
nrow(cdata16)
```

```
## [1] 930421
```

Dependent Variables

There are 32 features under the income category in the 2016 census data that represent different measurements of the variable. The variable **TOTINC** (Total Income) has been chosen as the dependent variable since it is both a numerical attribute and represents income information at an individual level. The variable captures all sources of income, such as employment, pension, government transfer, and capital gains.

```r
table(cdata16$TotInc)
```

```
##
##   -50000   -49000   -48000   -47000   -44000   -42000   -41000   -40000
##       40        1        1        2        3        1        2        3
##   -38000   -37000   -36000   -35000   -34000   -33000   -32000   -31000
##        1        1        4        5        3        3        4        2
##   -30000   -29000   -28000   -27000   -26000   -25000   -24000   -23000
##       53        8        7        6       10        7        9       12
##   -22000   -21000   -20000   -19000   -18000   -17000   -16000   -15000
##       17        7       14        6       12       16       18       23
##   -14000   -13000   -12000   -11000   -10000    -9000    -8000    -7000
##       25       30       25       26       47       61       56       66
##    -6000    -5000    -4000    -3000    -2000    -1000       -1        1
##       80      100      131      139      206      290      179    10254
##     1000     2000     3000     4000     5000     6000     7000     8000
##    15527     9446     8447     7978     7890     8132     8714     9780
##     9000    10000    11000    12000    13000    14000    15000    16000
##    10211    10807    11850    12582    12866    12632    12068    11220
##    17000    18000    19000    20000    21000    22000    23000    24000
##    12767    14167    13440    13145    12036    11356    10855    10357
##    25000    26000    27000    28000    29000    30000    31000    32000
##    10111     9524     9337     9080     9149     8892     8766     8831
##    33000    34000    35000    36000    37000    38000    39000    40000
##     8579     8666     8432     8494     8464     8476     8340     8473
##    41000    42000    43000    44000    45000    46000    47000    48000
##     8164     8087     7849     7858     7669     7288     7104     6984
##    49000    50000    51000    52000    53000    54000    55000    56000
##     6888     6735     6531     6207     5971     5949     5874     5592
##    57000    58000    59000    60000    61000    62000    63000    64000
##     5623     5385     5084     5123     4889     4693     4586     4563
##    65000    66000    67000    68000    69000    70000    71000    72000
##     4365     4323     4189     3968     3935     3782     3755     3703
##    73000    74000    75000    76000    77000    78000    79000    80000
##     3445     3368     3406     3406     3168     3102     3022     3065
##    81000    82000    83000    84000    85000    86000    87000    88000
##     2865     2829     2770     2550     2534     2494     2408     2402
##    89000    90000    91000    92000    93000    94000    95000    96000
```

```
##     2254     2173     2248     2119     2087     2043     2072     2039
##    97000    98000    99000    1e+05   101000   110000   120000   130000
##     1928     1837     1760     7850       16    11521     8009     5675
##   140000   150000   155893   159785   160000   170000   173041   173208
##     4188     3015       68      103     2199     1654       42      430
##   180000   185953   190000   198184    2e+05   200865   207174   209851
##     1394       19     1041       66      754      115      483      261
##   210000   210834   214592   215096   216809   220000   222338   222385
##      631       76       81       59      140      451       40       55
##   223755   224526   230000   230563   234544   240000   242446   243904
##       56       81      399       70       56      331       87       77
##   250000   252126   252140   260000   261934   269662   270000   270375
##      259       54       13      172       73       72      187      138
##   271968   277619   278138   278417   280000   290000   299821    3e+05
##       82      108      686      129      144       88      205      104
##   310000   319657   320000   320168   320322   321385   330000   330960
##       77       39       60      482       20      418       57      203
##   332712   340000   345553   348311   350000   355053   360000   370000
##       70       50       41       66       52       82       32       30
##   371103   374742   380000   380112   380747   381470   390000   394608
##       11      240       23      224      114       50       25      441
##    4e+05   400988   410000   418101   420000   430000   435139   440000
##       11       84       12     1006        8        7       85        8
##   445021   450000   460000   470000   480000   484028   487798   490000
##      152        8       14        6        7       60       34        7
##   490486    5e+05   500719   510000   520000   521667   524443   527416
##       89        5       82        2       10       79      181       76
##   530000   532891   540000   568304   571371   607851   614858   623566
##        3       48        3      162       66       87       63      123
##   638816   652888   681433   897244  1056323  1081553  1586814 88888888
##      111      642      401      223      838      242      202     5542
## 99999999
##   188605
```

The range of TOTINC is from -$50,000 to $1,586,814, and there are 5,542 unavailable records (with value 88,888,888) and 188,605 records which are not applicable (with value 99,999,999).

The reference period for information gathered on TOTINC is set in 2015, and that income information for respondents below the age of 15 were not gathered. Therefore, only the income of **respondents who are at or above 15 years old and worked in 2015** are examined as the target population in this analysis.

To separate eligible observations that fit this criteria from the rest of the data set, the variable **WKSWRK** (Weeks worked in 2015), which illustrates the number of weeks an individual worked in 2015, is used to identify those who performed work in 2015.

```
target <- subset(cdata16, TotInc!=88888888 & TotInc!=99999999 & WKSWRK!=0 & WKSWRK!=9)

# remaining records
nrow(target)
```

```
## [1] 514025
```

Regarding the variable WKSWRK, 16,035 records represent individuals who worked in 2016 only, while 391,418 records are marked as not applicable. These non-applicable records pertain to individuals who worked before 2015, never worked, or are under 15 years of age.

After cleaning not applicable and not available data following our criteria, 416,396 observations were removed from the original total of 930,421 observations, leaving 514,025 records remaining as the target data set.

Based on theoretical correlation with personal income and the percentage of usable data, **27 features** were chosen as predictor variables. **Due to the presence of pre-estimated weights, imputation is not implemented for missing values.**

```
# delete all unusable data
df2<-subset(target, AGEGRP!=88 & AGEGRP!=1 & AGEGRP!=2 & AGEGRP!=3 & AGEGRP!=4 & AGEGRP!=5
            & GENSTAT!=8 & IMMCAT5!=88 & CIP2011!=88 & CIP2011!=99 & HDGREE!=88
            & HDGREE!=99 & TotInc!=88888888 & TotInc!=99999999 & MOB1!=8 & MOB1!=9
            & KOL!=8 & COW!=8 & COW!=9 & FPTWK!=8 & FPTWK!=9 & LSTWRK!=9 & WKSWRK!=9
            & WKSWRK!=0 & WKSWRK!=9 & BedRm!=8 & CONDO!=8 & DTYPE!=8 & HCORENEED_IND!=888
            & NOS!=8 & REPAIR!=8 & ROOMS!=88 & DPGRSUM!=88 & NAICS!=88 & NOCS!=88
            & CfSize!=8 & HHSIZE!=8)

# records for modelling
nrow(df2)
```

```
## [1] 434778
```

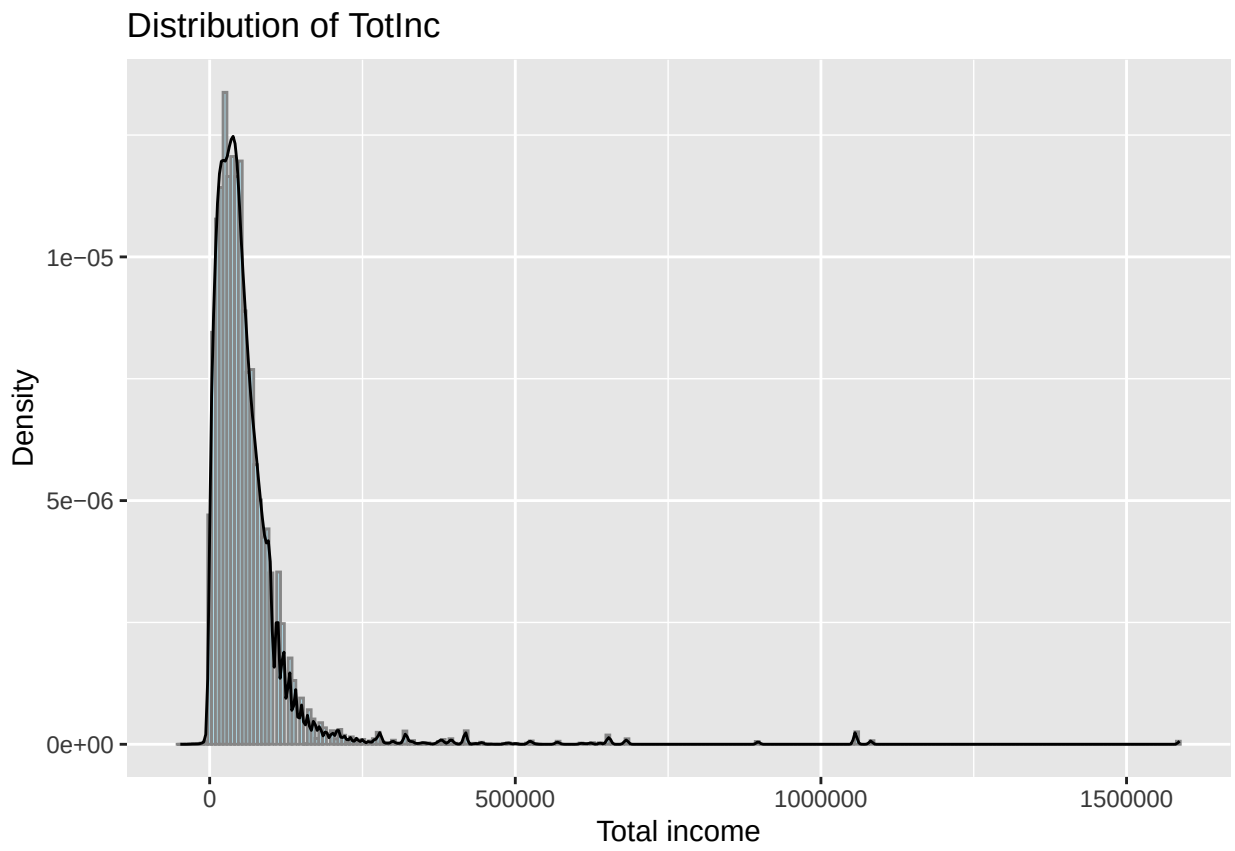
```
# percentage representing target dataset based on weights
percent_target=(sum(df2$WEIGHT)/sum(target$WEIGHT))*100; percent_target
```

```
## [1] 84.58311
```

After cleaning all the missing data among the predictor variables, 79,247 further observations were removed from the data set before the eventual analysis. The remaining sample size, **434,778, represents 84.58% of the target population** of our study, which once again constitutes all respondents who were at or above fifteen years of age during the time of the census and have worked in 2015.

```
# data for analysis
df3 <- df2[,c("WEIGHT", "AGEGRP", "MarStH", "Sex", "GENSTAT", "IMMCAT5", "CIP2011",
              "HDGREE", "TotInc", "MOB1", "KOL", "Citizen", "COW", "FPTWK", "LSTWRK",
              "WKSWRK", "BedRm", "CONDO", "DTYPE", "HCORENEED_IND", "NOS", "REPAIR",
              "ROOMS", "DPGRSUM", "NAICS", "NOCS", "CfSize", "HHSIZE", "DETH123")]

# keep a dataset for coding in python
df4 <- df3
```

```
# density curve with the histogram
ggplot(df3, aes(x=TotInc, y=after_stat(density))) +
  geom_histogram(fill="lightblue", color="grey60", bins = 262) +
  geom_density() + ggtitle("Distribution of TotInc") +
  xlab("Total income") + ylab("Density")
```



Variable Type Conversion

```
# check class
sapply(df3, class)
```

```
## $WEIGHT
## [1] "numeric"
##
## $AGEGRP
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $MarStH
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $Sex
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $GENSTAT
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $IMMCAT5
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $CIP2011
```

```
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $HDGREE
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $TotInc
## [1] "numeric"
##
## $MOB1
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $KOL
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $Citizen
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $COW
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $FPTWK
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $LSTWRK
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $WKSWRK
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $BedRm
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $CONDO
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $DTYPE
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $HCORENEED_IND
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $NOS
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $REPAIR
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $ROOMS
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $DPGRSUM
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $NAICS
```

```
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $NOCS
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $CfSize
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $HHSIZE
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $DETH123
## [1] "haven_labelled" "vctrs_vctr"     "double"
```

```
# convert to factor
cols <- c("AGEGRP", "MarStH", "Sex", "GENSTAT", "IMMCAT5", "CIP2011", "HDGREE", "MOB1",
          "KOL", "Citizen", "COW", "FPTWK", "LSTWRK", "WKSWRK", "BedRm", "CONDO",
          "DTYPE", "HCORENEED_IND", "NOS", "REPAIR", "ROOMS", "DPGRSUM", "NAICS",
          "NOCS", "CfSize", "HHSIZE", "DETH123")

df3[cols] <- lapply(df3[cols], as.factor)

# confirm class
sapply(df3, class)
```

```
##        WEIGHT        AGEGRP        MarStH           Sex       GENSTAT
##     "numeric"      "factor"      "factor"      "factor"      "factor"
##       IMMCAT5       CIP2011        HDGREE        TotInc          MOB1
##      "factor"      "factor"      "factor"     "numeric"      "factor"
##           KOL       Citizen           COW         FPTWK        LSTWRK
##      "factor"      "factor"      "factor"      "factor"      "factor"
##        WKSWRK         BedRm         CONDO         DTYPE HCORENEED_IND
##      "factor"      "factor"      "factor"      "factor"      "factor"
##           NOS        REPAIR         ROOMS       DPGRSUM         NAICS
##      "factor"      "factor"      "factor"      "factor"      "factor"
##          NOCS        CfSize        HHSIZE       DETH123
##      "factor"      "factor"      "factor"      "factor"
```

All explanatory variables are categorical ones.

```
str(df3)
```

```
## tibble [434,778 x 29] (S3: tbl_df/tbl/data.frame)
##  $ WEIGHT       : num [1:434778] 37 37.1 37 37 37 ...
##   ..- attr(*, "label")= chr "Individuals weighting factor"
##   ..- attr(*, "format.spss")= chr "F17.13"
##   ..- attr(*, "display_width")= int 18
##  $ AGEGRP       : Factor w/ 16 levels "6","7","8","9",..: 6 10 9 3 6 6 6 7 3 6 ...
##  $ MarStH       : Factor w/ 6 levels "1","2","3","4",..: 2 3 2 1 3 2 2 2 1 2 ...
##  $ Sex          : Factor w/ 2 levels "1","2": 2 1 2 2 2 1 1 1 2 2 ...
##  $ GENSTAT      : Factor w/ 4 levels "1","2","3","4": 4 4 4 4 3 2 4 4 4 1 ...
##  $ IMMCAT5      : Factor w/ 6 levels "1","2","3","21",..: 1 1 1 1 1 1 1 1 1 4 ...
##  $ CIP2011      : Factor w/ 12 levels "1","2","3","4",..: 8 5 11 12 12 4 12 4 12 7 ...
```

```
##  $ HDGREE       : Factor w/ 13 levels "1","2","3","4",..: 4 6 3 2 2 9 2 7 2 9 ...
##  $ TotInc       : num [1:434778] 97000 30000 53000 27000 43000 72000 58000 6000 24000 120000 ...
##   ..- attr(*, "label")= chr "Income: Total income"
##   ..- attr(*, "format.spss")= chr "F8.0"
##   ..- attr(*, "display_width")= int 10
##  $ MOB1         : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 2 1 1 1 1 ...
##  $ KOL          : Factor w/ 4 levels "1","2","3","4": 1 3 2 1 1 1 1 1 1 1 ...
##  $ Citizen      : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 2 ...
##  $ COW          : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ FPTWK        : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ LSTWRK       : Factor w/ 2 levels "2","3": 2 2 2 2 2 2 2 2 2 2 ...
##  $ WKSWRK       : Factor w/ 6 levels "1","2","3","4",..: 6 5 6 4 6 6 6 6 6 6 ...
##  $ BedRm        : Factor w/ 6 levels "0","1","2","3",..: 6 3 3 5 5 4 4 4 4 4 ...
##  $ CONDO        : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ...
##  $ DTYPE        : Factor w/ 3 levels "1","2","3": 1 1 2 1 1 1 2 1 1 3 ...
##  $ HCORENEED_IND: Factor w/ 2 levels "0","100": 1 1 1 1 1 1 1 1 1 1 ...
##  $ NOS          : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ REPAIR       : Factor w/ 3 levels "1","2","3": 1 3 1 1 2 1 1 1 2 1 ...
##  $ ROOMS        : Factor w/ 11 levels "1","2","3","4",..: 11 5 4 10 10 7 7 10 8 8 ...
##  $ DPGRSUM      : Factor w/ 15 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 13 ...
##  $ NAICS        : Factor w/ 19 levels "1","2","3","4",..: 12 14 18 12 19 10 19 15 7 12 ...
##  $ NOCS         : Factor w/ 10 levels "1","2","3","4",..: 8 2 7 3 8 5 2 5 7 3 ...
##  $ CfSize       : Factor w/ 7 levels "1","2","3","4",..: 4 2 2 4 4 3 4 4 4 4 ...
##  $ HHSIZE       : Factor w/ 7 levels "1","2","3","4",..: 4 2 2 4 4 3 4 4 4 4 ...
##  $ DETH123      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 1 2 2 ...
```

Training and Testing indicator

```
# set seed for reproducibility
set.seed(123)

# leave 20% of dataset for testing
test <- sample(c(FALSE, TRUE), nrow(df3), replace=TRUE, prob=c(0.8,0.2))

# size of test set
sum(test)
```

```
## [1] 87123
```

```
# size of training set
train <- !test
sum(train)
```

```
## [1] 347655
```

An **80-20 split** was applied to the dataset, resulting in 87,123 records in the test set and 347,655 records in the training set.

Models

Individual weights associated with each observation are passed into all three kinds of models as the **weight parameter** to ensure that the results are generalisable to the entire Canadian population.

Regeression Model

Multiple Regression

```
# fit the weighted multiple regression model
MF <- lm(TotInc ~ AGEGRP + MarStH + Sex + GENSTAT + IMMCAT5 + CIP2011 + HDGREE + MOB1
         + KOL + Citizen + COW + FPTWK + LSTWRK + WKSWRK + BedRm + CONDO + DTYPE
         + HCORENEED_IND + NOS+ REPAIR + ROOMS + DPGRSUM + NAICS + NOCS + CfSize
         + HHSIZE + DETH123, weights = WEIGHT, data = df3, subset = train)

# view the results
summary(MF)
```

```
##
## Call:
## lm(formula = TotInc ~ AGEGRP + MarStH + Sex + GENSTAT + IMMCAT5 +
##     CIP2011 + HDGREE + MOB1 + KOL + Citizen + COW + FPTWK + LSTWRK +
##     WKSWRK + BedRm + CONDO + DTYPE + HCORENEED_IND + NOS + REPAIR +
##     ROOMS + DPGRSUM + NAICS + NOCS + CfSize + HHSIZE + DETH123,
##     data = df3, subset = train, weights = WEIGHT)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -1460532  -149185   -37483    73131  9258342
##
## Coefficients: (2 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      107329.5     4309.1  24.908  < 2e-16 ***
## AGEGRP7             313.0     1182.9   0.265 0.791335
## AGEGRP8           -2975.9     1067.7  -2.787 0.005316 **
## AGEGRP9           -2677.6     1122.1  -2.386 0.017022 *
## AGEGRP10           4412.5     1144.5   3.855 0.000116 ***
## AGEGRP11          10312.2     1152.6   8.947  < 2e-16 ***
## AGEGRP12          15588.9     1157.1  13.472  < 2e-16 ***
## AGEGRP13          18471.8     1158.9  15.939  < 2e-16 ***
## AGEGRP14          19725.6     1156.4  17.058  < 2e-16 ***
## AGEGRP15          21080.6     1168.8  18.037  < 2e-16 ***
## AGEGRP16          22050.9     1207.1  18.268  < 2e-16 ***
## AGEGRP17          24379.3     1302.4  18.719  < 2e-16 ***
## AGEGRP18          28404.9     1588.2  17.885  < 2e-16 ***
## AGEGRP19          32842.1     2287.5  14.357  < 2e-16 ***
## AGEGRP20          34595.5     3768.7   9.180  < 2e-16 ***
## AGEGRP21          37352.2     7200.6   5.187 2.13e-07 ***
## MarStH2           10569.0      489.9  21.573  < 2e-16 ***
## MarStH3            6719.9      527.3  12.744  < 2e-16 ***
## MarStH4            8818.2      943.7   9.344  < 2e-16 ***
```

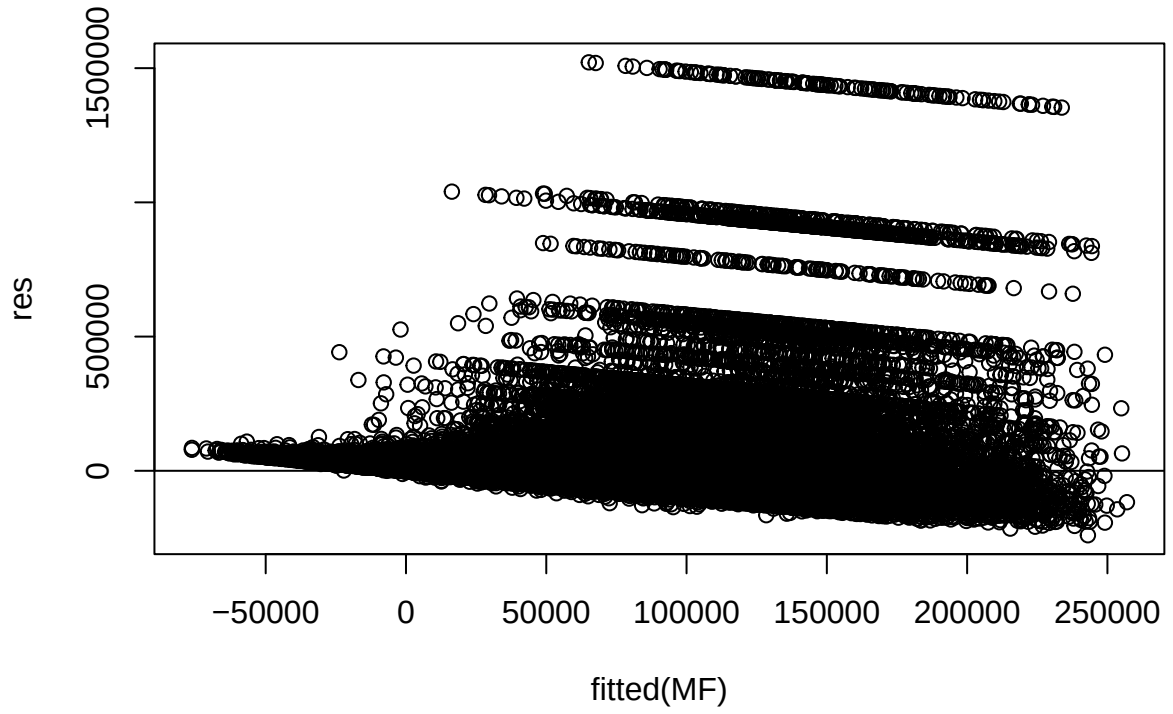
```
## MarStH5           6039.7      673.1   8.974  < 2e-16 ***
## MarStH6           8394.5     1260.9   6.658 2.79e-11 ***
## Sex2             17614.4      303.2  58.096  < 2e-16 ***
## GENSTAT2          1757.8     2217.5   0.793 0.427958
## GENSTAT3         -1216.4     2219.2  -0.548 0.583616
## GENSTAT4         -2416.3     2178.7  -1.109 0.267408
## IMMCAT52          2012.4     2366.5   0.850 0.395105
## IMMCAT53         -9052.6     2559.4  -3.537 0.000405 ***
## IMMCAT521       -14998.6     2290.0  -6.550 5.77e-11 ***
## IMMCAT522        -9044.8     2322.3  -3.895 9.83e-05 ***
## IMMCAT523        -9814.4     2426.2  -4.045 5.23e-05 ***
## CIP20112          -265.4     1129.7  -0.235 0.814289
## CIP20113         -5007.6     1002.6  -4.994 5.90e-07 ***
## CIP20114          7784.2      858.2   9.070  < 2e-16 ***
## CIP20115         12737.5      835.7  15.242  < 2e-16 ***
## CIP20116          2051.7     1099.0   1.867 0.061922 .
## CIP20117          5888.4     1113.6   5.288 1.24e-07 ***
## CIP20118         14009.6      902.8  15.518  < 2e-16 ***
## CIP20119           364.4     1433.6   0.254 0.799351
## CIP201110         9430.2      953.1   9.894  < 2e-16 ***
## CIP201111        11685.7     1045.9  11.173  < 2e-16 ***
## CIP201113       -71753.2     1731.2 -41.446  < 2e-16 ***
## HDGREE2           3944.6      479.2   8.232  < 2e-16 ***
## HDGREE3         -78545.2     1569.1 -50.058  < 2e-16 ***
## HDGREE4         -70875.2     1604.3 -44.177  < 2e-16 ***
## HDGREE5         -77792.6     1612.4 -48.247  < 2e-16 ***
## HDGREE6         -74115.0     1499.0 -49.444  < 2e-16 ***
## HDGREE7         -69137.6     1510.1 -45.783  < 2e-16 ***
## HDGREE8         -65206.8     1626.1 -40.099  < 2e-16 ***
## HDGREE9         -49684.5     1455.2 -34.142  < 2e-16 ***
## HDGREE10        -40771.9     1715.9 -23.761  < 2e-16 ***
## HDGREE11         38437.7     2148.1  17.894  < 2e-16 ***
## HDGREE12        -32846.3     1513.6 -21.701  < 2e-16 ***
## HDGREE13              NA         NA      NA       NA
## MOB12             1100.0      476.0   2.311 0.020855 *
## MOB13             1590.2     1167.6   1.362 0.173213
## MOB14            -1711.7      818.4  -2.091 0.036493 *
## MOB15             1280.5     1575.4   0.813 0.416324
## MOB16           -15160.7     2254.0  -6.726 1.74e-11 ***
## KOL2             -9105.7      468.3 -19.446  < 2e-16 ***
## KOL3             -2905.9      330.5  -8.791  < 2e-16 ***
## KOL4             -4311.8     1648.2  -2.616 0.008894 **
## Citizen2           696.6      677.4   1.028 0.303814
## Citizen3              NA         NA      NA       NA
## COW2            -14840.5     2931.7  -5.062 4.15e-07 ***
## COW3            -11764.1      976.2 -12.051  < 2e-16 ***
## COW4             15311.3      862.4  17.755  < 2e-16 ***
## COW5            -21359.7      613.2 -34.834  < 2e-16 ***
## COW6              -721.9      985.7  -0.732 0.463983
## FPTWK2          -14618.1      358.5 -40.781  < 2e-16 ***
## LSTWRK3           3049.6      481.2   6.338 2.33e-10 ***
## WKSWRK2          -2489.0      785.1  -3.170 0.001522 **
## WKSWRK3           2059.6      764.1   2.696 0.007027 **
## WKSWRK4           3696.7      787.1   4.697 2.64e-06 ***
```

```
## WKSWRK5           11920.0      705.7  16.891  < 2e-16 ***
## WKSWRK6           10909.7      676.3  16.131  < 2e-16 ***
## BedRm1             2593.5     4726.8   0.549 0.583235
## BedRm2             7664.1     4765.2   1.608 0.107759
## BedRm3            10286.3     4782.8   2.151 0.031502 *
## BedRm4            15287.5     4794.7   3.188 0.001431 **
## BedRm5            18710.8     4813.9   3.887 0.000102 ***
## CONDO1             5694.6      453.3  12.564  < 2e-16 ***
## DTYPE2            -1434.8      412.7  -3.477 0.000508 ***
## DTYPE3            -2058.9      407.3  -5.055 4.31e-07 ***
## HCORENEED_IND100 -17889.9      546.7 -32.723  < 2e-16 ***
## NOS1              -6663.4      622.0 -10.713  < 2e-16 ***
## REPAIR2           -4796.6      282.1 -17.006  < 2e-16 ***
## REPAIR3           -3983.9      546.6  -7.289 3.13e-13 ***
## ROOMS2             1950.9     5454.4   0.358 0.720588
## ROOMS3             2020.8     5482.8   0.369 0.712451
## ROOMS4             2531.5     5508.7   0.460 0.645844
## ROOMS5             3031.8     5518.7   0.549 0.582758
## ROOMS6             3642.9     5526.4   0.659 0.509782
## ROOMS7             4088.8     5530.6   0.739 0.459722
## ROOMS8             6246.8     5534.2   1.129 0.258996
## ROOMS9             7843.3     5541.8   1.415 0.156977
## ROOMS10           12966.4     5546.4   2.338 0.019399 *
## ROOMS11           21297.2     5551.9   3.836 0.000125 ***
## DPGRSUM2         -13343.1      715.1 -18.660  < 2e-16 ***
## DPGRSUM3         -10648.9      765.6 -13.909  < 2e-16 ***
## DPGRSUM4          -5284.1      880.5  -6.001 1.96e-09 ***
## DPGRSUM5         -10492.5      941.7 -11.142  < 2e-16 ***
## DPGRSUM6          -5590.1     1220.5  -4.580 4.65e-06 ***
## DPGRSUM7          -9757.9     1314.9  -7.421 1.16e-13 ***
## DPGRSUM8          -5548.2     1577.2  -3.518 0.000435 ***
## DPGRSUM9         -10600.4     1728.1  -6.134 8.57e-10 ***
## DPGRSUM10        -11567.4     2029.7  -5.699 1.21e-08 ***
## DPGRSUM11         -2625.5     3558.7  -0.738 0.460647
## DPGRSUM12         -8708.4     2258.1  -3.856 0.000115 ***
## DPGRSUM13         -7778.9     2093.2  -3.716 0.000202 ***
## DPGRSUM14         -2603.6     1414.5  -1.841 0.065673 .
## DPGRSUM15         -1075.0      727.9  -1.477 0.139726
## NAICS2            78566.8     1473.8  53.308  < 2e-16 ***
## NAICS3            45509.3     1895.7  24.007  < 2e-16 ***
## NAICS4            11493.2     1225.1   9.382  < 2e-16 ***
## NAICS5            12215.3     1217.8  10.031  < 2e-16 ***
## NAICS6            15385.4     1296.1  11.871  < 2e-16 ***
## NAICS7             -183.2     1190.1  -0.154 0.877667
## NAICS8             9163.5     1268.7   7.223 5.10e-13 ***
## NAICS9            16456.3     1418.4  11.602  < 2e-16 ***
## NAICS10           34316.0     1282.6  26.756  < 2e-16 ***
## NAICS11           14716.7     1476.1   9.970  < 2e-16 ***
## NAICS12           19479.5     1229.9  15.838  < 2e-16 ***
## NAICS13            1691.4     1237.8   1.366 0.171785
## NAICS14           -3862.3     1280.9  -3.015 0.002567 **
## NAICS15            -712.1     1245.6  -0.572 0.567515
## NAICS16             790.8     1423.2   0.556 0.578457
## NAICS17             116.2     1245.2   0.093 0.925651
```

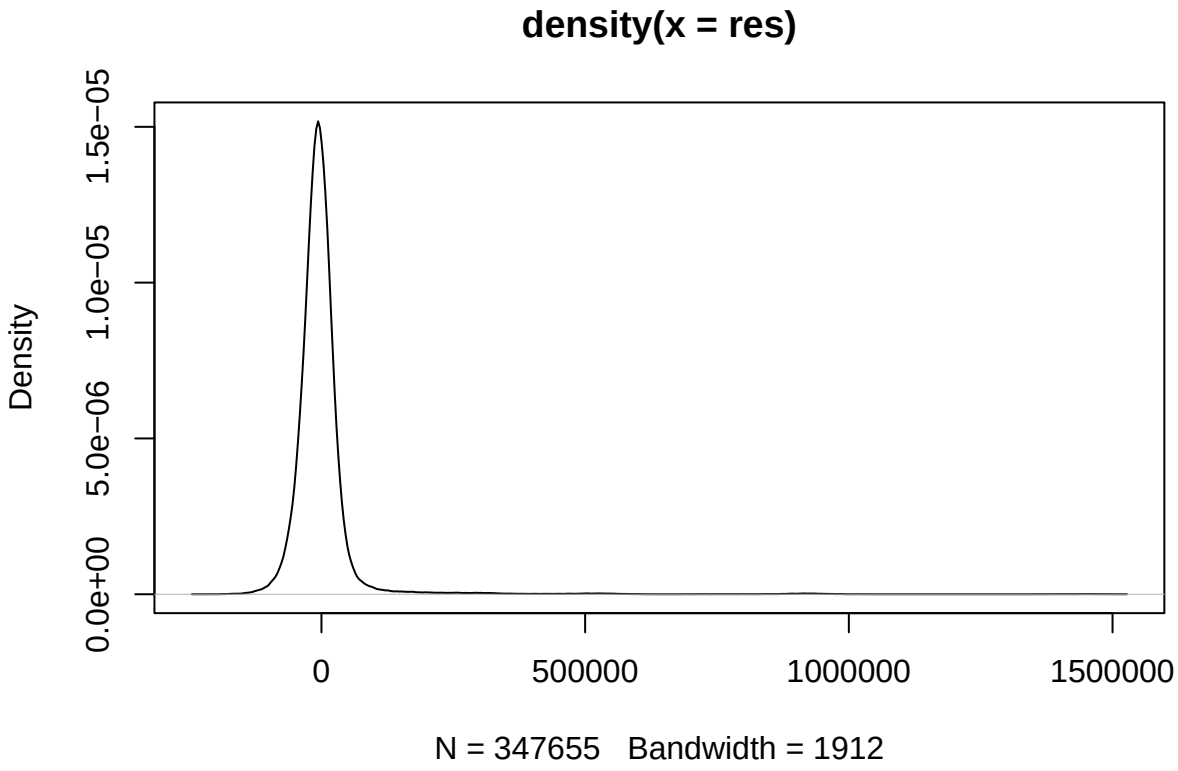
```
## NAICS18           -448.8     1274.7  -0.352 0.724752
## NAICS19           8993.7     1238.4   7.262 3.82e-13 ***
## NOCS2           -29508.8      523.1 -56.407  < 2e-16 ***
## NOCS3           -25858.2      670.7 -38.554  < 2e-16 ***
## NOCS4           -13015.7      849.7 -15.318  < 2e-16 ***
## NOCS5           -20899.3      634.4 -32.944  < 2e-16 ***
## NOCS6           -31973.3      898.1 -35.600  < 2e-16 ***
## NOCS7           -30508.1      513.1 -59.462  < 2e-16 ***
## NOCS8           -36804.5      584.5 -62.963  < 2e-16 ***
## NOCS9           -34535.1     1086.8 -31.776  < 2e-16 ***
## NOCS10          -36338.7      801.1 -45.364  < 2e-16 ***
## CfSize2           -469.2      673.7  -0.696 0.486153
## CfSize3           3227.1      770.9   4.186 2.84e-05 ***
## CfSize4           9208.3      871.2  10.569  < 2e-16 ***
## CfSize5          11354.3     1027.8  11.047  < 2e-16 ***
## CfSize6          13229.1     1495.7   8.845  < 2e-16 ***
## CfSize7          12949.0     2188.4   5.917 3.28e-09 ***
## HHSIZE2          -9232.4      722.2 -12.784  < 2e-16 ***
## HHSIZE3         -14495.4      841.0 -17.236  < 2e-16 ***
## HHSIZE4         -18347.4      943.2 -19.452  < 2e-16 ***
## HHSIZE5         -21066.4     1069.1 -19.705  < 2e-16 ***
## HHSIZE6         -25636.9     1327.8 -19.308  < 2e-16 ***
## HHSIZE7         -29787.5     1500.3 -19.854  < 2e-16 ***
## DETH1232          1252.6      280.5   4.466 7.96e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 442700 on 347506 degrees of freedom
## Multiple R-squared:  0.227,  Adjusted R-squared:  0.2267
## F-statistic: 689.5 on 148 and 347506 DF,  p-value: < 2.2e-16
```

Multiple regression model returned a relatively low **R-squared score of 0.227**, signaling a poor fit that is capable of explaining only 22.7% of the variation in individual income. It thus be concluded that multiple regression is largely ineffective at producing accurate numerical predictions of income here.

```
# residual plots
res <- resid(MF)
plot(fitted(MF), res)
abline(0,0)
```




```
plot(density(res))
```



Classification Model

Values of the income variable are first divided into **five bins** based on the **2015 federal income tax brackets** to render the dataset suitable for a classification approach. Another binning method that separates income into only **two categories** based on the data's **median value of 44,000** is also implemented to serve as an additional classification test.

```
# decide intervals for 5 bins
table(df3$TotInc)
```

```
##
##  -48000  -47000  -44000  -40000  -37000  -35000  -34000  -33000  -30000  -29000
##       1       1       1       1       1       1       1       1       8       2
##  -28000  -27000  -26000  -25000  -24000  -23000  -22000  -21000  -20000  -19000
##       1       3       1       5       4       3       2       3       6       1
##  -18000  -17000  -16000  -15000  -14000  -13000  -12000  -11000  -10000   -9000
##       4       5       4       5      11      11       7      10      22      16
##   -8000   -7000   -6000   -5000   -4000   -3000   -2000   -1000      -1       1
##      20      26      29      36      55      67      91     117      76    2016
##    1000    2000    3000    4000    5000    6000    7000    8000    9000   10000
##    3758    3254    3437    3493    3635    3703    3840    4032    4318    4531
##   11000   12000   13000   14000   15000   16000   17000   18000   19000   20000
##    4699    4852    5022    5101    5143    5141    5149    5224    5116    5317
##   21000   22000   23000   24000   25000   26000   27000   28000   29000   30000
##    5164    5141    5245    5210    5306    5088    5197    5241    5341    5208
```

```
##    31000    32000    33000    34000    35000    36000    37000    38000    39000    40000
##     5233     5273     5291     5382     5290     5421     5514     5482     5485     5650
##    41000    42000    43000    44000    45000    46000    47000    48000    49000    50000
##     5389     5426     5291     5315     5212     5085     4905     4905     4776     4789
##    51000    52000    53000    54000    55000    56000    57000    58000    59000    60000
##     4649     4350     4218     4276     4227     4011     4097     3919     3684     3702
##    61000    62000    63000    64000    65000    66000    67000    68000    69000    70000
##     3570     3424     3436     3375     3282     3244     3119     2987     3003     2895
##    71000    72000    73000    74000    75000    76000    77000    78000    79000    80000
##     2847     2846     2698     2634     2658     2683     2513     2436     2381     2435
##    81000    82000    83000    84000    85000    86000    87000    88000    89000    90000
##     2280     2283     2213     2061     2037     2038     1947     1969     1835     1762
##    91000    92000    93000    94000    95000    96000    97000    98000    99000    1e+05
##     1825     1719     1739     1705     1744     1703     1602     1547     1448     6570
##   101000   110000   120000   130000   140000   150000   155893   159785   160000   170000
##        9     9626     6743     4823     3569     2584       33       54     1877     1416
##   173041   173208   180000   185953   190000   198184    2e+05   200865   207174   209851
##       22      307     1207        9      913       42      669       48      309      154
##   210000   210834   214592   215096   216809   220000   222338   222385   223755   224526
##      559       40       39       38       93      399       13       28       26       46
##   230000   230563   234544   240000   242446   243904   250000   252126   252140   260000
##      350       40       25      300       52       46      242       27        3      149
##   261934   269662   270000   270375   271968   277619   278138   278417   280000   290000
##       41       42      166       77       30       72      527       72      137       85
##   299821    3e+05   310000   319657   320000   320168   320322   321385   330000   330960
##      136       92       71       26       58      370       11      277       56      111
##   332712   340000   345553   348311   350000   355053   360000   370000   371103   374742
##       40       47       30       46       47       45       32       28        3      157
##   380000   380112   380747   381470   390000   394608    4e+05   400988   410000   418101
##       21      150       67       30       24      312       10       46       11      750
##   420000   430000   435139   440000   445021   450000   460000   470000   480000   484028
##        6        7       48        8      115        8       13        4        7       43
##   487798   490000   490486    5e+05   500719   510000   520000   521667   524443   527416
##       19        7       55        5       45        2       10       57      129       57
##   530000   532891   540000   568304   571371   607851   614858   623566   638816   652888
##        2       21        3      122       49       68       35       93       82      518
##   681433   897244  1056323  1081553  1586814
##      308      158      698      201      172
```

The range of TOTINC spans from -$48,000 to $1,586,814, with a median of $44,000.

```
# calculate the median value of TotInc
median(df3$TotInc)
```

```
## [1] 44000
```

Random Forest

Five Bins Income levels and ranges for each bin is as follows:

- Low: $0 or less
- Low-Medium: $1 to $44,701

- Medium: $44,702 to $89,401
- Medium-High: $89,402 to $138,586
- High: Over $138,586

```
# create breaks for the cut function
breaks5 <- c(-60000, 10, 44701, 89401, 138586, 1600000)

# minimum -48000/maximum 1586814 -> set -60000/1600000 as the first/last value in breaks
# based on 2015 Federal Tax Brackets
# [44,701(included) 89,401(included) 138,586(included) over]

# Values that would have been rounded to zero have been replaced by 1 or -1.
# numbers of records: -1(76) 1(2016)
# next value: 1000 -> set 10 as the second value in breaks -> low: under zero
# values near 44,701: 44,000 and 45,000 -> set 44,701 -> Low-Medium
# values near 89,401: 89,000 and 90,000 -> set 89,401 -> Medium
# values near 138,586: 130,000 and 140,000 -> set 138,586 -> Medium-High
```

```
# create labels for the new factor variable
labels5 <- c("Low", "Low-Medium", "Medium", "Medium-High", "High")
```

```
# categorize the values in TotInc and create a new variable
df3$TotInc5 <- cut(df3$TotInc, breaks = breaks5, labels = labels5)

summary(df3$TotInc5)
```

```
##          Low  Low-Medium      Medium Medium-High        High
##         2675      216348      147934       44565       23256
```

It should be noted that the number of observations in each bracket is **not evenly divided**. A total of 216,348 observations fall into the low-medium category and another 147,934 belong to the medium category.

```
train_weights <- df3$WEIGHT[train]

set.seed(123)

RF5 <- randomForest(TotInc5 ~ AGEGRP + MarStH + Sex + GENSTAT + IMMCAT5 + CIP2011 + HDGREE
                    + MOB1 + KOL + Citizen + COW + FPTWK + LSTWRK + WKSWRK + BedRm + CONDO
                    + DTYPE + HCORENEED_IND + NOS+ REPAIR+ ROOMS + DPGRSUM + NAICS + NOCS
                    + CfSize + HHSIZE + DETH123,
                    data = df3, subset = train, weights = train_weights, ntree=10)

# result
print(RF5)
```

```
##
## Call:
##  randomForest(formula = TotInc5 ~ AGEGRP + MarStH + Sex + GENSTAT +      IMMCAT5 + CIP2011 + HDGREE +
##                Type of random forest: classification
##                      Number of trees: 10
## No. of variables tried at each split: 5
```

```
##
##         OOB estimate of  error rate: 39.16%
## Confusion matrix:
##              Low Low-Medium Medium Medium-High High class.error
## Low           69       1752    222          46   32   0.9674682
## Low-Medium   875     133557  32118        3301 1492   0.2205284
## Medium       173      38787  63217       11200 3809   0.4605414
## Medium-High   42       5763  17789        8290 3303   0.7644016
## High          32       2871   6972        4217 4269   0.7674963
```

```
# predicted class
RF5.pred <- predict(RF5, newdata=subset(df3,test), type="class")

# confusion matrix
test.ct5 <- table(Predicted=RF5.pred, Actual=df3$TotInc5[test]); test.ct5
```

```
##              Actual
## Predicted       Low Low-Medium Medium Medium-High  High
##   Low             3         16      0           1     0
##   Low-Medium    465      35251   9062        1154   606
##   Medium         55       7518  18095        5204  1969
##   Medium-High     5        337   1888        1935   963
##   High            2        138    551         735  1170
```

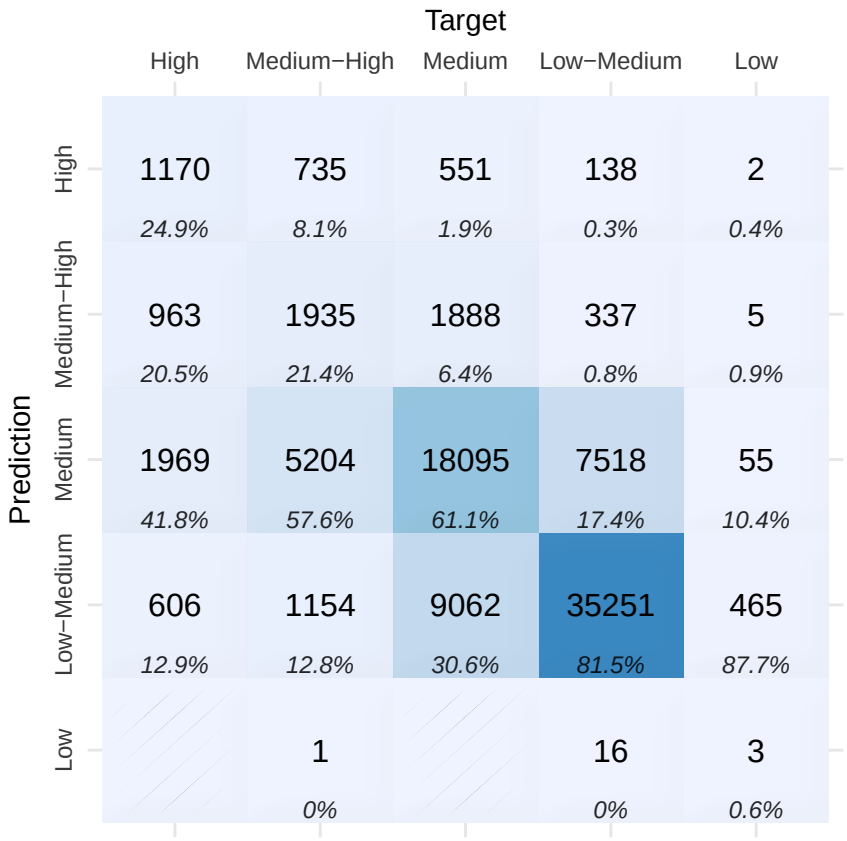
```
# Accuracy
sum(diag(test.ct5)) / sum(test.ct5)
```

```
## [1] 0.6479804
```

```
# library(ggimage)
# library(rsvg)
# library(cvms)
cm5_se <- confusion_matrix(targets = df3$TotInc5[test], predictions = RF5.pred)
```

```
## Warning in n_correct * n_samples: NAs produced by integer overflow
```

```
plot_confusion_matrix(cm5_se$`Confusion Matrix`[[1]],
                      class_order = c("Low","Low-Medium","Medium","Medium-High","High"),
                      add_row_percentages = FALSE,
                      add_normalized = FALSE,
                      font_col_percentages = font(size = 3),
                      add_arrows = FALSE)
```

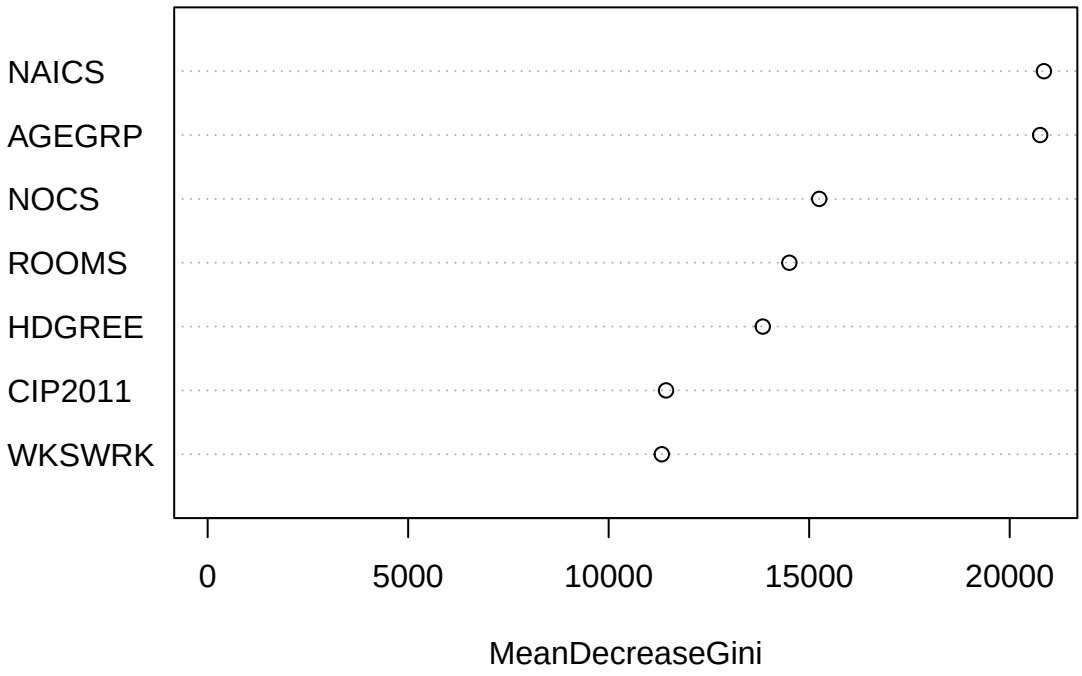



```
# get the variable importance
var_imp5 <- importance(RF5); print(var_imp5)
```

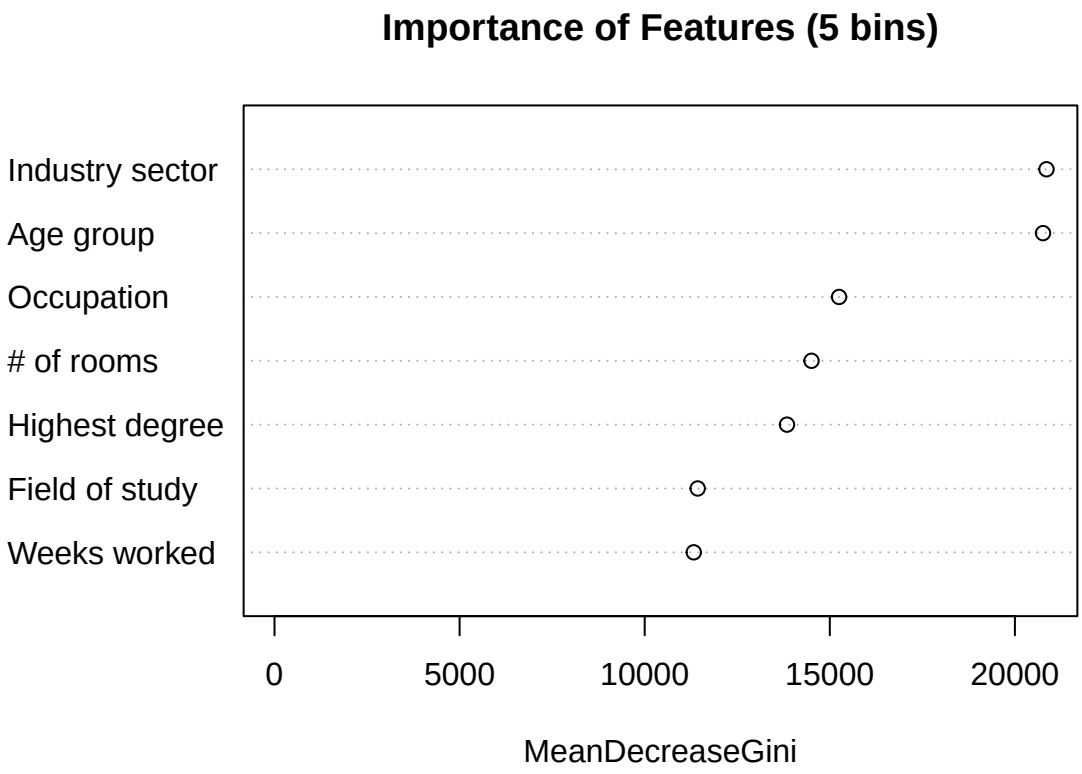
```
##                MeanDecreaseGini
## AGEGRP              20759.3855
## MarStH               8051.5339
## Sex                  3167.9019
## GENSTAT              4793.9717
## IMMCAT5              3261.9021
## CIP2011             11430.6900
## HDGREE              13843.4258
## MOB1                 3301.7154
## KOL                  4822.7214
## Citizen              1887.4742
## COW                  4224.4965
## FPTWK                9089.0502
## LSTWRK               1388.4461
## WKSWRK              11325.0897
## BedRm                7833.3688
## CONDO                1689.7937
## DTYPE                4438.6529
## HCORENEED_IND        4478.6187
## NOS                   784.1332
## REPAIR               5081.8078
## ROOMS               14505.3539
## DPGRSUM              5699.9736
```

```
## NAICS               20853.0293
## NOCS                15250.3355
## CfSize               7420.3939
## HHSIZE               8198.7458
## DETH123              3798.5517
```

```r
# get variables first
varImpPlot(RF5, sort = TRUE, n.var = 7, main="")
```




```r
varImpPlot(RF5, sort = T, n.var = 7, main = "Importance of Features (5 bins)",
           labels=c("Weeks worked", "Field of study", "Highest degree", "# of rooms",
                    "Occupation", "Age group","Industry sector"))
```




```
# calculate precision for each class
precision5 <- diag(test.ct5) / colSums(test.ct5); precision5
```

```
##         Low  Low-Medium      Medium Medium-High        High
## 0.005660377 0.814863615 0.611400189 0.214309447 0.248513169
```

```
# create breaks for the cut function
break2 <- c(-60000, 44701, 1600000)

# create labels for the new factor variable
label2 <- c("<= $44K", "Over $44K")

# categorize the values in TotInc
df3$TotInc2 <- cut(df3$TotInc, breaks = break2, labels = label2)

summary(df3$TotInc2)
```

Two Bins

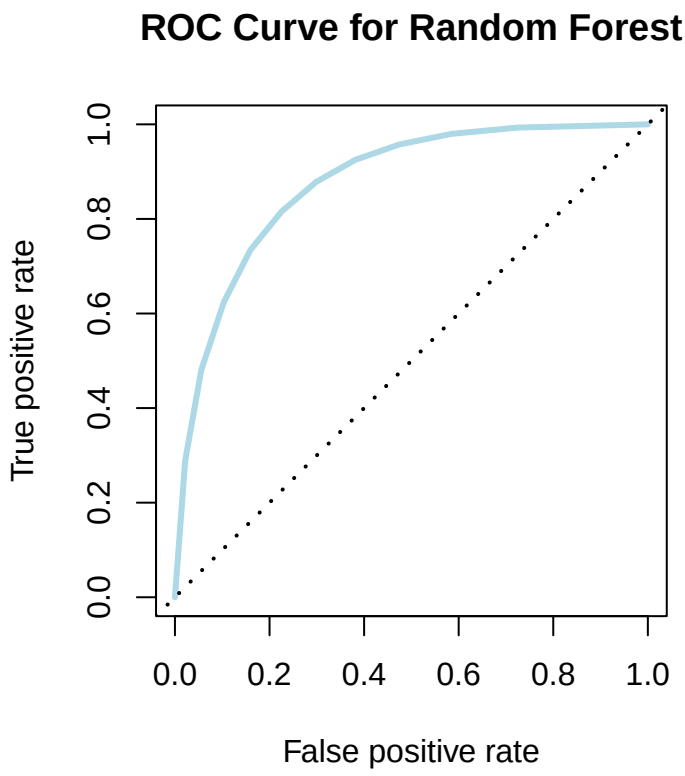
```
##   <= $44K Over $44K
##    219023    215755
```

Income observations are split into two categories, with 219,023 having a total income lower or equal to the median value of $44,000 and 215,755 having an income above it.

```
train_weights <- df3$WEIGHT[train]

set.seed(2023)
RF2 <- randomForest(TotInc2 ~ AGEGRP + MarStH + Sex + GENSTAT + IMMCAT5 + CIP2011 + HDGREE
                    + MOB1 + KOL + Citizen + COW + FPTWK + LSTWRK + WKSWRK + BedRm + CONDO
                    + DTYPE + HCORENEED_IND + NOS + REPAIR + ROOMS + DPGRSUM + NAICS
                    + NOCS + CfSize + HHSIZE + DETH123,
                    data = df3, subset = train, weights = train_weights, ntree=10)

print(RF2)
```

```
##
## Call:
##  randomForest(formula = TotInc2 ~ AGEGRP + MarStH + Sex + GENSTAT +      IMMCAT5 + CIP2011 + HDGREE +
##                Type of random forest: classification
##                      Number of trees: 10
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 23.34%
## Confusion matrix:
##           <= $44K Over $44K class.error
## <= $44K    127844     45528   0.2626030
## Over $44K   34757    135855   0.2037196
```

```
RF2.pred <- predict(RF2, newdata=subset(df3,test), type="class")
RF2.prob <- predict(RF2, newdata=subset(df3,test), type="prob")

# library(ROCR)
par(pty = "s")
plot(performance(prediction(RF2.prob[,2], df3$TotInc2[test]), "tpr", "fpr"),
     main = "ROC Curve for Random Forest", lwd= 3, col="lightblue")
abline(a=0, b= 1, lty=3, lwd = 2)
```




```
test.ct2 <- table(Predicted=RF2.pred, Actual=df3$TotInc2[test]); test.ct2
```

```
##            Actual
## Predicted   <= $44K Over $44K
##   <= $44K     32850      7155
##   Over $44K   10940     36178
```

```
# accuracy
sum(diag(test.ct2)) / sum(test.ct2)
```

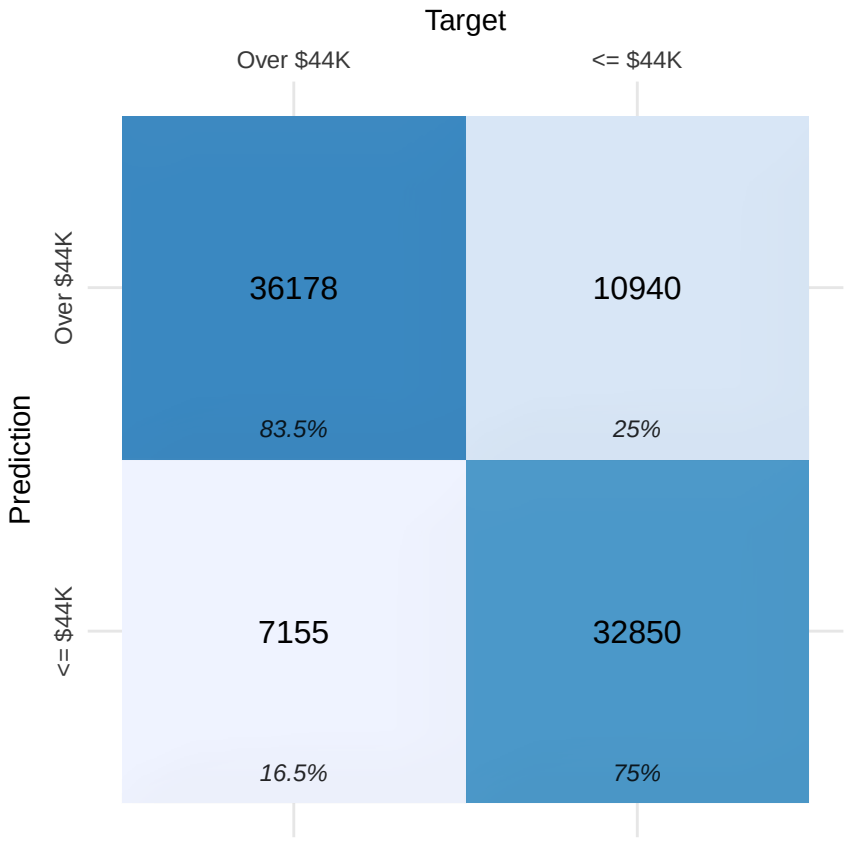
```
## [1] 0.7923051
```

```
# calculate precision for each class
precision <- diag(test.ct2) / rowSums(test.ct2); precision
```

```
##   <= $44K Over $44K
## 0.8211474 0.7678170
```

```
# library(ggimage)
# library(rsvg)
# library(cvms)
cm2_se <- confusion_matrix(targets = df3$TotInc2[test], predictions = RF2.pred)
```

```
plot_confusion_matrix(cm2_se$`Confusion Matrix`[[1]],
                      class_order = c("<= $44K", "Over $44K"),
                      add_row_percentages = FALSE, add_normalized = FALSE,
                      font_col_percentages = font(size = 3), add_arrows = FALSE)
```

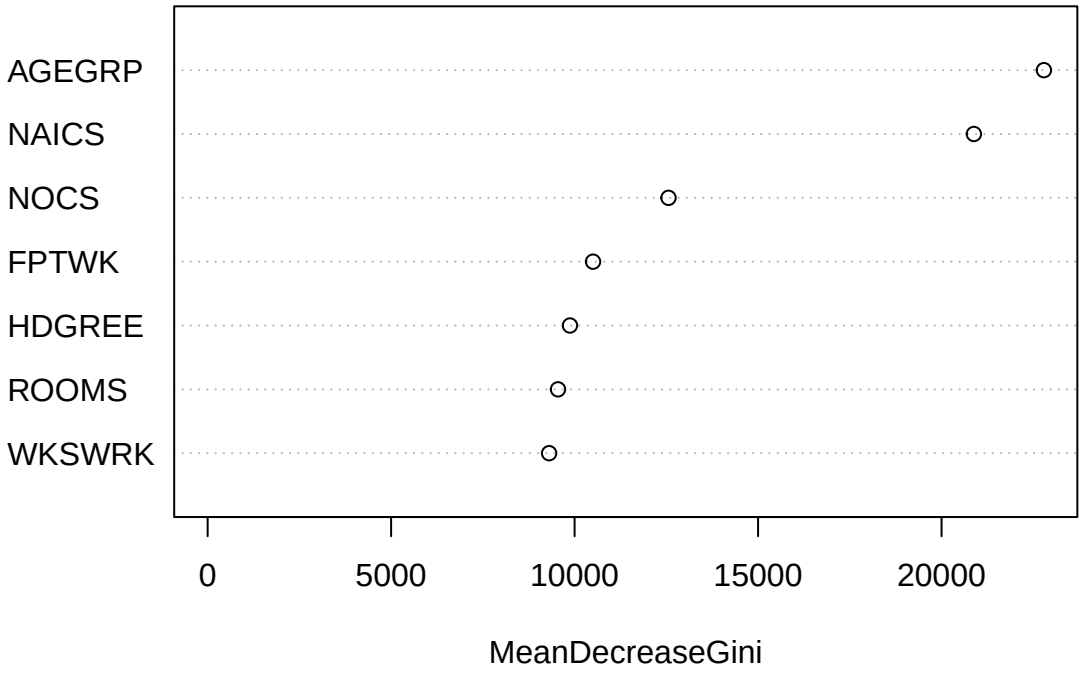



```
# get the variable importance
var_imp2 <- importance(RF2); print(var_imp2)
```

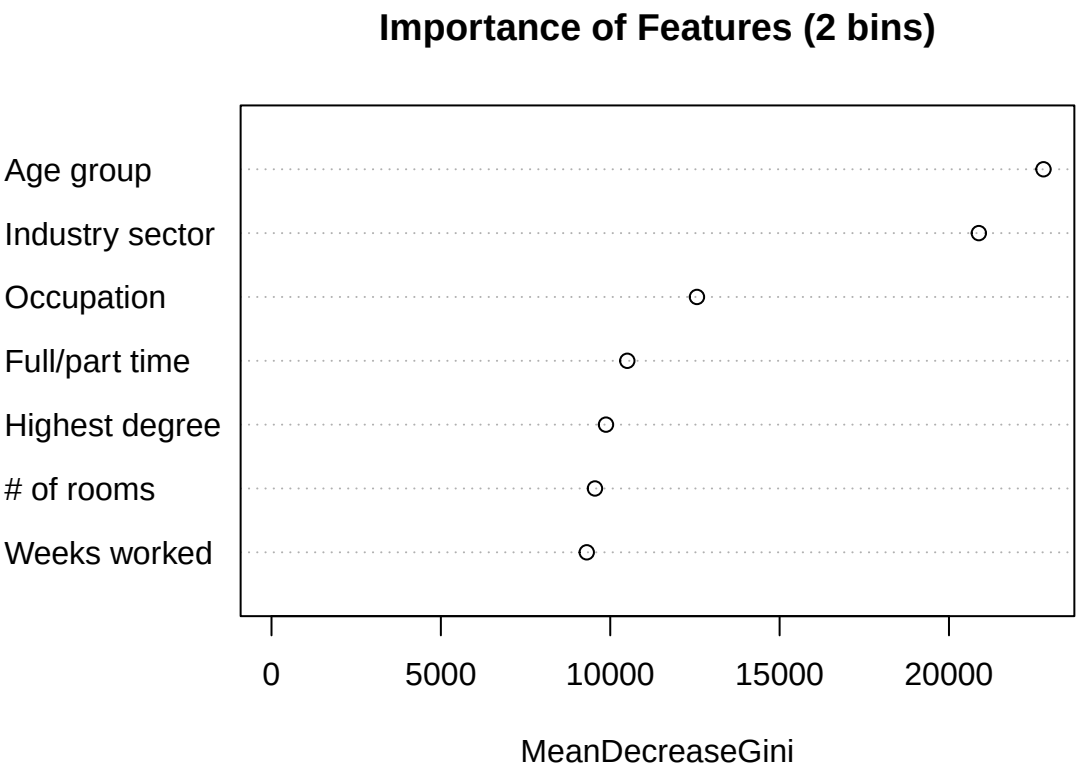
```
##               MeanDecreaseGini
## AGEGRP             22789.7004
## MarStH              6478.3637
## Sex                 3062.2666
## GENSTAT             2747.9364
## IMMCAT5             2452.9595
## CIP2011             9025.1992
## HDGREE              9874.0049
## MOB1                2248.7743
## KOL                 3344.6886
## Citizen             1237.5312
## COW                 3976.7982
## FPTWK              10502.8517
## LSTWRK              1246.7044
## WKSWRK              9306.1886
## BedRm               4855.7923
## CONDO               1143.5400
```

```
## DTYPE               2857.0682
## HCORENEED_IND       6468.2581
## NOS                  789.9489
## REPAIR              3058.9090
## ROOMS               9548.0913
## DPGRSUM             4014.5124
## NAICS              20881.1111
## NOCS               12559.3181
## CfSize              4483.3740
## HHSIZE              5319.9055
## DETH123             2109.0505
```

```r
varImpPlot(RF2, sort = TRUE, n.var = 7, main="")
```




```r
varImpPlot(RF2, sort = T, n.var = 7, main = "Importance of Features (2 bins)",
           labels=c("Weeks worked","# of rooms", "Highest degree", "Full/part time",
                    "Occupation", "Industry sector", "Age group"))
```




```
# precision of 0
p2_0 = test.ct2[1, 1] / sum(test.ct2[1, ]); p2_0
```

```
## [1] 0.8211474
```

```
# recall of 0
r2_0 = test.ct2[1, 1] / sum(test.ct2[,1]); r2_0
```

```
## [1] 0.7501713
```

```
# F-1 score of 0
f2_0 = 2 * p2_0  * r2_0 / (p2_0 + r2_0); f2_0
```

```
## [1] 0.7840563
```

```
# precision of 1
p2_1 = test.ct2[2, 2] / sum(test.ct2[2, ]); p2_1
```

```
## [1] 0.767817
```

```
# recall of 1
r2_1 = test.ct2[2, 2] / sum(test.ct2[, 2]); r2_1
```

```
## [1] 0.8348833
```

```
# F-1 score of 1
f2_1 = 2 * p2_1 * r2_1 / (p2_1 + r2_1); f2_1
```

```
## [1] 0.7999469
```

Naive Bayes

Since R does not have a suitable package for handling weights in Naive Bayes modeling, Python was used instead, utilizing the Scikit-learn library. The objective here is to create an appropriate dataset while **ensuring that the training and test sets remain consistent** for the modeling process in Python.

```
sapply(df4, class)
```

Create Suitable Variables

```
## $WEIGHT
## [1] "numeric"
##
## $AGEGRP
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $MarStH
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $Sex
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $GENSTAT
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $IMMCAT5
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $CIP2011
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $HDGREE
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $TotInc
## [1] "numeric"
##
## $MOB1
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $KOL
## [1] "haven_labelled" "vctrs_vctr"     "double"
```

```
##
## $Citizen
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $COW
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $FPTWK
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $LSTWRK
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $WKSWRK
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $BedRm
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $CONDO
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $DTYPE
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $HCORENEED_IND
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $NOS
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $REPAIR
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $ROOMS
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $DPGRSUM
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $NAICS
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $NOCS
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $CfSize
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $HHSIZE
## [1] "haven_labelled" "vctrs_vctr"     "double"
##
## $DETH123
## [1] "haven_labelled" "vctrs_vctr"     "double"
```

```r
#change labels
levels(df3$AGEGRP)
```

```
##  [1] "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15" "16" "17" "18" "19" "20"
## [16] "21"
```

```r
df4$AGEGRP <- factor(df4$AGEGRP,
                     levels=c("6" , "7" , "8" , "9" , "10", "11", "12", "13" ,"14",
                              "15" ,"16" ,"17" ,"18", "19", "20" ,"21"),
                     labels=c("0","1","2","3","4","5","6" , "7" , "8" , "9" , "10",
                              "11", "12", "13" ,"14", "15"))

table(df4$AGEGRP)
```

```
##
##     0     1     2     3     4     5     6     7     8     9    10    11    12
##  8437 13113 40495 43313 45120 44095 43558 45345 51413 46657 30677 14979  5249
##    13    14    15
##  1686   508   133
```

```r
levels(df3$MarStH)
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```r
df4$MarStH <- factor(df4$MarStH,
                     levels = c("1", "2", "3", "4", "5", "6"),
                     labels = c("0", "1", "2", "3", "4", "5"))
```

```r
levels(df3$Sex)
```

```
## [1] "1" "2"
```

```r
df4$Sex <- factor(df4$Sex,
                  levels = c("1", "2"),
                  labels = c("0", "1"))
```

```r
levels(df3$GENSTAT)
```

```
## [1] "1" "2" "3" "4"
```

```r
df4$GENSTAT <- factor(df4$GENSTAT,
                      levels = c("1", "2", "3", "4"),
                      labels = c("0", "1", "2", "3"))
```

```r
levels(df3$IMMCAT5)
```

```
## [1] "1"  "2"  "3"  "21" "22" "23"
```

```
df4$IMMCAT5 <- factor(df4$IMMCAT5,
                     levels = c("1",  "2",  "3",  "21", "22", "23"),
                     labels = c("0", "1",  "2",  "3", "4", "5"))
```

```
levels(df3$CIP2011)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "13"
```

```
df4$CIP2011 <- factor(df4$CIP2011,
                     levels = c("1", "2",  "3",  "4",  "5",  "6",  "7",  "8",  "9",
                                "10", "11", "13"),
                     labels = c("0", "1",  "2",  "3",  "4",  "5",  "6",  "7",  "8",
                                "9",  "10", "11"))
```

```
levels(df3$HDGREE)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13"
```

```
df4$HDGREE <- factor(df4$HDGREE,
                    levels = c("1", "2",  "3",  "4",  "5",  "6",  "7",  "8",  "9",
                               "10", "11", "12", "13"),
                    labels = c("0", "1",  "2",  "3",  "4",  "5",  "6",  "7",  "8",
                               "9",  "10", "11", "12"))
```

```
levels(df3$MOB1)
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```
df4$MOB1 <- factor(df4$MOB1,
                   levels = c("1", "2",  "3",  "4",  "5",  "6"),
                   labels = c("0", "1",  "2",  "3",  "4",  "5"))
```

```
levels(df3$KOL)
```

```
## [1] "1" "2" "3" "4"
```

```
df4$KOL <- factor(df4$KOL,
                  levels = c("1", "2",  "3",  "4"),
                  labels = c("0", "1",  "2",  "3"))
```

```
levels(df3$Citizen)
```

```
## [1] "1" "2" "3"
```

```
df4$Citizen <- factor(df4$Citizen,
                     levels = c("1", "2",  "3"),
                     labels = c("0", "1",  "2"))
```

```r
levels(df3$COW)
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```r
df4$COW <- factor(df4$COW,
                  levels = c("1", "2",  "3",  "4",  "5",  "6"),
                  labels = c("0", "1",  "2",  "3",  "4",  "5"))
```

```r
levels(df3$FPTWK)
```

```
## [1] "1" "2"
```

```r
df4$FPTWK <- factor(df4$FPTWK,
                    levels = c("1", "2"),
                    labels = c("0", "1"))
```

```r
levels(df3$LSTWRK)
```

```
## [1] "2" "3"
```

```r
df4$LSTWRK <- factor(df4$LSTWRK,
                     levels = c("2", "3"),
                     labels = c("0", "1"))
```

```r
levels(df3$WKSWRK)
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```r
df4$WKSWRK <- factor(df4$WKSWRK,
                     levels = c("1", "2",  "3",  "4",  "5",  "6"),
                     labels = c("0", "1",  "2",  "3",  "4",  "5"))
```

```r
levels(df3$BedRm)
```

```
## [1] "0" "1" "2" "3" "4" "5"
```

```r
df4$BedRm <- factor(df4$BedRm)
```

```r
levels(df3$CONDO)
```

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

```r
df4$CONDO <- factor(df4$CONDO)
```

```r
levels(df3$DTYPE)
```

```
## [1] "1" "2" "3"
```

```
df4$DTYPE <- factor(df4$DTYPE,
                    levels = c("1", "2",  "3"),
                    labels = c("0", "1",  "2"))
```

```
levels(df3$HCORENEED_IND)
```

```
## [1] "0"   "100"
```

```
df4$HCORENEED_IND <- factor(df4$HCORENEED_IND,
                            levels = c("0", "100"),
                            labels = c("0", "1"))
```

```
levels(df3$NOS)
```

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

```
df4$NOS <- factor(df4$NOS)
```

```
levels(df3$REPAIR)
```

```
## [1] "1" "2" "3"
```

```
df4$REPAIR <- factor(df4$REPAIR,
                     levels = c("1", "2",  "3"),
                     labels = c("0", "1",  "2"))
```

```
levels(df3$ROOMS)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11"
```

```
df4$ROOMS <- factor(df4$ROOMS,
                    levels = c("1", "2",  "3",  "4",  "5",  "6",  "7",  "8",  "9",
                               "10", "11"),
                    labels = c("0", "1",  "2",  "3",  "4",  "5",  "6",  "7",  "8",
                               "9",  "10"))
```

```
levels(df3$DPGRSUM)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
```

```
df4$DPGRSUM <- factor(df4$DPGRSUM,
                      levels = c("1", "2",  "3",  "4",  "5",  "6",  "7",  "8",  "9",
                                 "10", "11", "12", "13", "14", "15"),
                      labels = c("0", "1",  "2",  "3",  "4",  "5",  "6",  "7",  "8",
                                 "9",  "10", "11", "12", "13", "14"))
```

```r
levels(df3$NAICS)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19"
```

```r
df4$NAICS <- factor(df4$NAICS,
                    levels = c("1", "2",  "3",  "4",  "5",  "6",  "7",  "8",  "9",
                               "10", "11", "12", "13", "14", "15", "16","17","18","19"),
                    labels = c("0", "1",  "2",  "3",  "4",  "5",  "6",  "7",  "8",
                               "9",  "10", "11", "12", "13", "14", "15","16","17","18"))
```

```r
levels(df3$NOCS)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```r
df4$NOCS <- factor(df4$NOCS,
                   levels = c("1", "2",  "3",  "4",  "5",  "6",  "7",  "8",  "9", "10"),
                   labels = c("0", "1",  "2",  "3",  "4",  "5",  "6",  "7",  "8", "9"))
```

```r
levels(df3$CfSize)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7"
```

```r
df4$CfSize <- factor(df4$CfSize,
                     levels = c("1", "2",  "3",  "4",  "5",  "6",  "7"),
                     labels = c("0", "1",  "2",  "3",  "4",  "5",  "6"))
```

```r
levels(df3$HHSIZE)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7"
```

```r
df4$HHSIZE <- factor(df4$HHSIZE,
                     levels = c("1", "2",  "3",  "4",  "5",  "6",  "7"),
                     labels = c("0", "1",  "2",  "3",  "4",  "5",  "6"))
```

```r
levels(df3$DETH123)
```

```
## [1] "1" "2"
```

```r
df4$DETH123 <- factor(df4$DETH123,
                      levels = c("1", "2"),
                      labels = c("0", "1"))
```

```r
str(df4)
```

```
## tibble [434,778 x 29] (S3: tbl_df/tbl/data.frame)
##  $ WEIGHT       : num [1:434778] 37 37.1 37 37 37 ...
##   ..- attr(*, "label")= chr "Individuals weighting factor"
```

```
##   ..- attr(*, "format.spss")= chr "F17.13"
##   ..- attr(*, "display_width")= int 18
##  $ AGEGRP       : Factor w/ 16 levels "0","1","2","3",..: 6 10 9 3 6 6 6 7 3 6 ...
##  $ MarStH       : Factor w/ 6 levels "0","1","2","3",..: 2 3 2 1 3 2 2 2 1 2 ...
##  $ Sex          : Factor w/ 2 levels "0","1": 2 1 2 2 2 1 1 1 2 2 ...
##  $ GENSTAT      : Factor w/ 4 levels "0","1","2","3": 4 4 4 4 3 2 4 4 4 1 ...
##  $ IMMCAT5      : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 4 ...
##  $ CIP2011      : Factor w/ 12 levels "0","1","2","3",..: 8 5 11 12 12 4 12 4 12 7 ...
##  $ HDGREE       : Factor w/ 13 levels "0","1","2","3",..: 4 6 3 2 2 9 2 7 2 9 ...
##  $ TotInc       : num [1:434778] 97000 30000 53000 27000 43000 72000 58000 6000 24000 120000 ...
##   ..- attr(*, "label")= chr "Income: Total income"
##   ..- attr(*, "format.spss")= chr "F8.0"
##   ..- attr(*, "display_width")= int 10
##  $ MOB1         : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 2 1 1 1 1 ...
##  $ KOL          : Factor w/ 4 levels "0","1","2","3": 1 3 2 1 1 1 1 1 1 1 ...
##  $ Citizen      : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 2 ...
##  $ COW          : Factor w/ 6 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ FPTWK        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ LSTWRK       : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ WKSWRK       : Factor w/ 6 levels "0","1","2","3",..: 6 5 6 4 6 6 6 6 6 6 ...
##  $ BedRm        : Factor w/ 6 levels "0","1","2","3",..: 6 3 3 5 5 4 4 4 4 4 ...
##  $ CONDO        : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ...
##  $ DTYPE        : Factor w/ 3 levels "0","1","2": 1 1 2 1 1 1 2 1 1 3 ...
##  $ HCORENEED_IND: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ NOS          : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ REPAIR       : Factor w/ 3 levels "0","1","2": 1 3 1 1 2 1 1 1 2 1 ...
##  $ ROOMS        : Factor w/ 11 levels "0","1","2","3",..: 11 5 4 10 10 7 7 10 8 8 ...
##  $ DPGRSUM      : Factor w/ 15 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1 13 ...
##  $ NAICS        : Factor w/ 19 levels "0","1","2","3",..: 12 14 18 12 19 10 19 15 7 12 ...
##  $ NOCS         : Factor w/ 10 levels "0","1","2","3",..: 8 2 7 3 8 5 2 5 7 3 ...
##  $ CfSize       : Factor w/ 7 levels "0","1","2","3",..: 4 2 2 4 4 3 4 4 4 4 ...
##  $ HHSIZE       : Factor w/ 7 levels "0","1","2","3",..: 4 2 2 4 4 3 4 4 4 4 ...
##  $ DETH123      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 2 2 ...
```

```
# categorize the values in TotInc and create a new variable (5 bins)
df4$TotInc5 <- cut(df4$TotInc, breaks = breaks5, labels = labels5)

# categorize the values in TotInc and create a new variable (2 bins)
df4$TotInc2 <- cut(df4$TotInc, breaks = break2, labels = label2)
```

```
# obtain and save training data
table(train)
```

Create and save training and test sets

```
## train
##  FALSE   TRUE
##  87123 347655
```

```r
df4.train <- subset(df4, train)
write.csv(df4.train, "df4-train.csv", row.names = FALSE)

# obtain and save testing data
df4.test <- subset(df4, test)
write.csv(df4.test, "df4-test.csv", row.names = FALSE)
```

Modelling Process Check **NaiveBayes.ipynb** for details.

Income Distribution Across Important Features (5 bins)

As an extension of findings on variable importance, the analysis here examines income distribution by features that possessed the greatest mean decrease in Gini. The following bar plots thus represent how income is distributed across the categories of each feature for the Canadian population as a whole based on the distribution of available observations in our target data set.

Industry Sector and Income Distribution

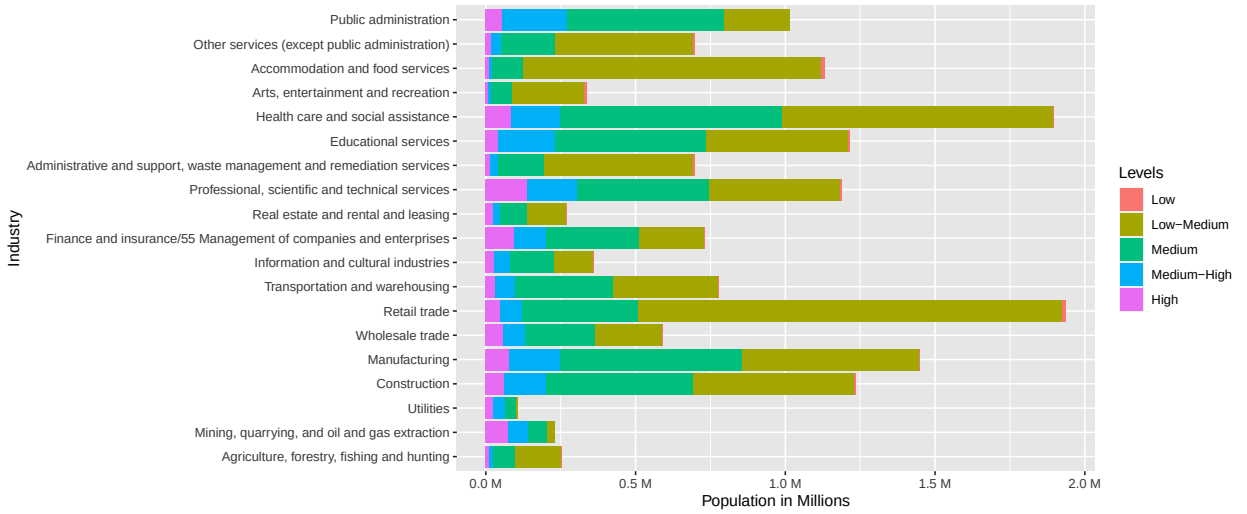
```r
# number of sample
table(df3$NAICS,df3$TotInc5)
```

```
##
##        Low Low-Medium Medium Medium-High  High
##   1     80       4096   2027         391   211
##   2     10        709   1682        1815  1960
##   3      2        237   1018        1043   598
##   4    247      14462  13291        3738  1612
##   5    125      15947  16359        4586  2075
##   6     44       6092   6326        1967  1484
##   7    430      38168  10460        2014  1232
##   8     84       9426   8817        1820   808
##   9     65       3524   3949        1469   682
##   10    50       5877   8378        2916  2475
##   11    90       3521   2443         654   599
##   12   210      11828  11836        4567  3638
##   13   175      13437   4134         676   352
##   14   146      12841  13610        5092  1071
##   15   126      24458  19949        4453  2230
##   16   202       6534   1852         297   165
##   17   369      26903   2736         314   219
##   18   165      12392   4886         931   407
##   19    55       5896  14181        5822  1438
```

```r
sum(table(df3$NAICS,df3$TotInc5))
```

```
## [1] 434778
```

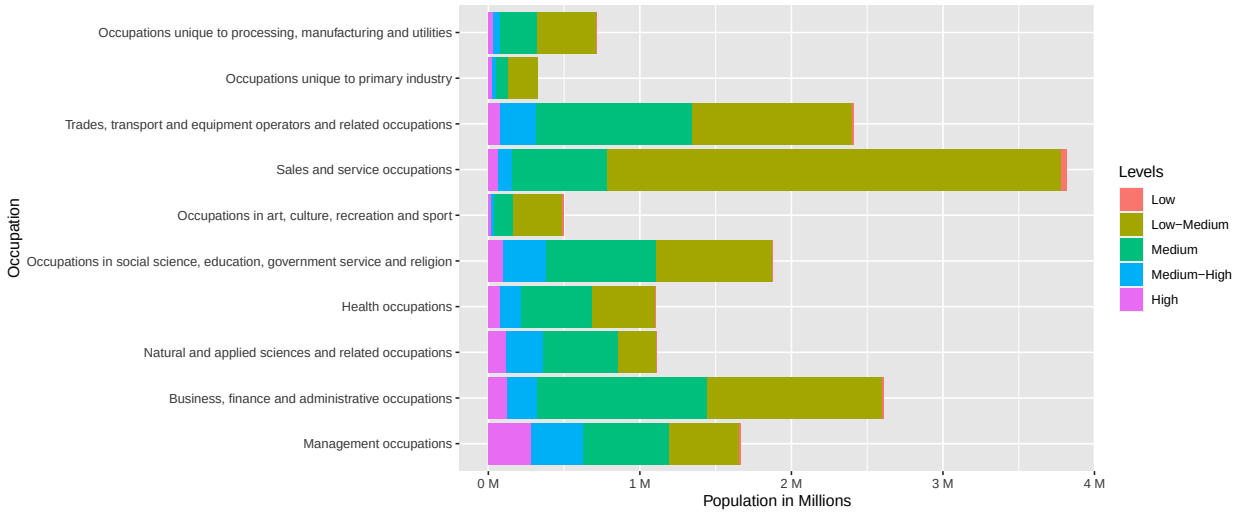
```r
indu<-df3[,c("WEIGHT","TotInc5","NAICS")]

dfff <- indu %>%
  mutate(Industry = recode(NAICS, "1" = "Agriculture, forestry, fishing and hunting",
           "2" = "Mining, quarrying, and oil and gas extraction",
           "3" = "Utilities",
           "4" = "Construction",
           "5" = "Manufacturing",
           "6" = "Wholesale trade",
           "7" = "Retail trade",
           "8" = "Transportation and warehousing",
           "9" = "Information and cultural industries",
           "10" = "Finance and insurance/55 Management of companies and enterprises",
           "11" = "Real estate and rental and leasing",
           "12" = "Professional, scientific and technical services",
           "13" = "Administrative and support, waste management and remediation services",
           "14" = "Educational services",
           "15" = "Health care and social assistance",
           "16" = "Arts, entertainment and recreation",
           "17" = "Accommodation and food services",
           "18" = "Other services (except public administration)",
           "19"= "Public administration"))
colnames(dfff)[2]<-"Levels"

ggplot(data=dfff, aes(x=Industry, y=WEIGHT, fill=Levels)) +
  geom_bar(stat="identity")  +
  ylab("Population in Millions") +
  scale_y_continuous(labels = unit_format(unit = "M", scale = 1e-6))+
  coord_flip()
```



It can be seen that those working in the **professional, scientific, and technical services sector** as well as the **public administration sectors** tend to have the highest rate of belonging to the medium-high and high income categories, while **retail trade and accommodation and food services** are most heavily populated by those in the low-medium income bracket or below.

Occupation and Income Distribution

```
ococ<-df3[,c("WEIGHT","TotInc5","NOCS")]

occupation <- ococ %>%
  mutate(Occupation = recode(NOCS, "1" = "Management occupations",
         "2" = "Business, finance and administrative occupations",
         "3" = "Natural and applied sciences and related occupations",
         "4" = "Health occupations",
         "5" = "Occupations in social science, education, government service and religion",
         "6" = "Occupations in art, culture, recreation and sport",
         "7" = "Sales and service occupations",
         "8" = "Trades, transport and equipment operators and related occupations",
         "9" = "Occupations unique to primary industry",
         "10" = "Occupations unique to processing, manufacturing and utilities"))

colnames(occupation)[2]<-"Levels"

ggplot(data=occupation, aes(x=Occupation, y=WEIGHT, fill=Levels)) +
  geom_bar(stat="identity")  +
  ylab("Population in Millions") +
  scale_y_continuous(labels = unit_format(unit = "M", scale = 1e-6))+
  coord_flip()
```

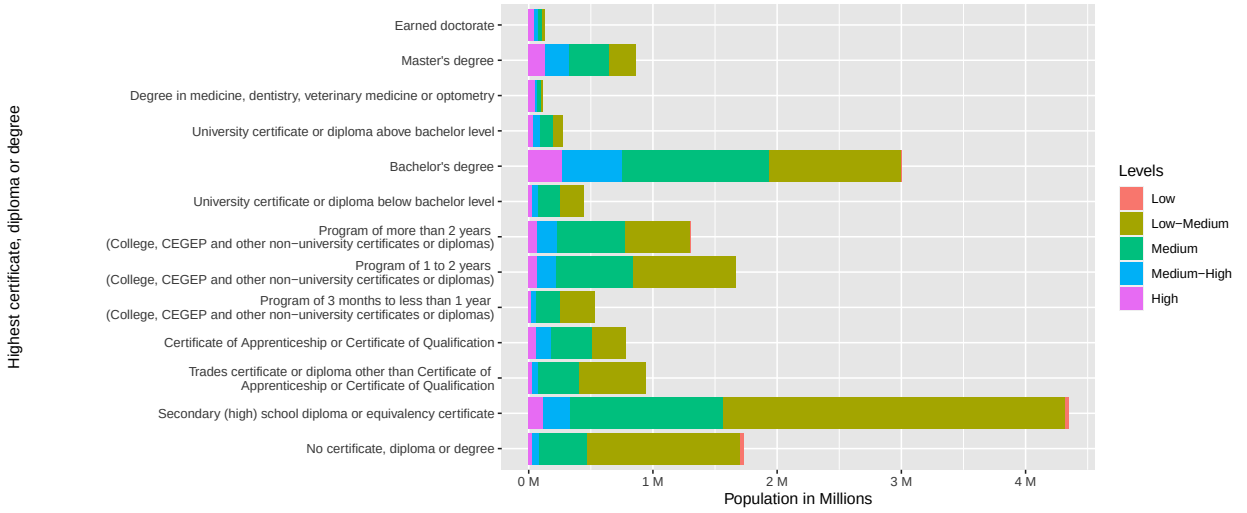


Canadians with jobs in **management, business, finance, and administrative sections** tend to have the highest income, whereas those with the lowest income are more likely to be **occupying jobs in the sales and services sector**.

Highest Degree Obtained and Income Distribution

```
deg<-df3[,c("WEIGHT","TotInc5","HDGREE")]

degree <- deg %>%
  mutate(Degree = recode(HDGREE, "1" = "No certificate, diploma or degree",
```

```
                        "2" = "Secondary (high) school diploma or equivalency certificate",
                        "3" = "Trades certificate or diploma other than Certificate of
                        Apprenticeship or Certificate of Qualification",
                        "4" = "Certificate of Apprenticeship or Certificate of Qualification",
                        "5" = "Program of 3 months to less than 1 year
                        (College, CEGEP and other non-university certificates or diplomas)",
                        "6" = "Program of 1 to 2 years
                        (College, CEGEP and other non-university certificates or diplomas)",
                        "7" = "Program of more than 2 years
                        (College, CEGEP and other non-university certificates or diplomas)",
                        "8" = "University certificate or diploma below bachelor level",
                        "9" = "Bachelor's degree",
                        "10" = "University certificate or diploma above bachelor level",
                        "11" = "Degree in medicine, dentistry, veterinary medicine or optometry",
                        "12" = "Master's degree",
                        "13"="Earned doctorate"))

colnames(degree)[2]<-"Levels"

ggplot(data=degree, aes(x=Degree, y=WEIGHT, fill=Levels)) +
  geom_bar(stat="identity")  + xlab("Highest certificate, diploma or degree")+
  ylab("Population in Millions") +
  scale_y_continuous(labels = unit_format(unit = "M", scale = 1e-6))+
  coord_flip()
```

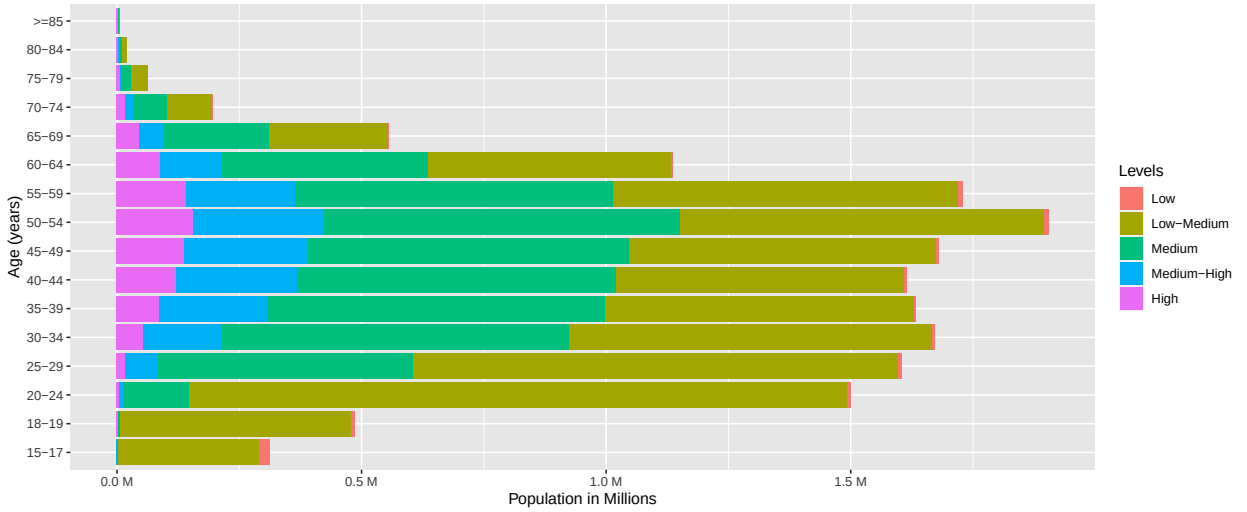


Those with **secondary education or below** are most likely to fall into low-medium and low income brackets, while **masters, doctorates, and degrees in the medical industry** have proportionately higher income. It is also worth noting that **earning a Bachelor's degree** marks a somewhat significant increase in one's probability of being in the medium-high bracket or above compared to holding a lower degree.

Age and Income Distribution

```
ageage<-df3[,c("WEIGHT","TotInc5","AGEGRP")]
```

```r
age <- ageage %>%
  mutate(Age = recode(AGEGRP,
                      "1" = "0-4",
                      "2" = "5-6",
                      "3" = "7-9",
                      "4" = "10-11",
                      "5" = "12-14",
                      "6" = "15-17",
                      "7" = "18-19",
                      "8" = "20-24",
                      "9" = "25-29",
                      "10" = "30-34",
                      "11" = "35-39",
                      "12" = "40-44",
                      "13" = "45-49",
                      "14" = "50-54",
                      "15" = "55-59",
                      "16" = "60-64",
                      "17" = "65-69",
                      "18" = "70-74",
                      "19" = "75-79",
                      "20" = "80-84",
                      "21" = ">=85"))

colnames(age)[2]<-"Levels"

ggplot(data=age, aes(x=Age, y=WEIGHT, fill=Levels)) +
  geom_bar(stat="identity")  + xlab("Age (years)")+
  ylab("Population in Millions") +
  scale_y_continuous(labels = unit_format(unit = "M", scale = 1e-6))+
  coord_flip()
```



Income tends to rise the fastest when one enters their **mid-twenties** and begins to drop after **55 years of age**.