HarvardX Data Science Final Project: A Classification Task Using Machine Learning Techniques

Gretta Digbeu

July 28th, 2019

A.Introduction

This machine learning exercise is the final component in the capstone course of the Harvard edX Data Science online professional certificate program. Students pursuing the verified track are asked to apply predictive machine learning techniques using a publicly available dataset to solve a problem of their choice.

We chose to explore the 1996 *Adult* dataset, also known as the *Census Income* dataset, available on the UCI's Machine Learning Repository. The data was extracted by Silicon Valley researchers (Ronny Kohavi and Barry Becker) from the 1994 Current Population Survey (CPS) data held in the Census database. It is already partitioned into training and testing sets containing 32,561 and 16,281 observations respectively (30,162 and 15,060 after removing unknowns), and includes 14 attributes. The variable of interest, gross income, was discretized into two ranges with a threshold of $50,000. The data is therefore ideal for a multivariate classification exercise with the goal of predicting this binary outcome variable.

Before exploring the data for trends and patterns, let us present a breakdown of the attributes contained therein:

Features and Their Types

attribute	type	description
age	continuous	Age in years
workclass	categorical	Class of worker
final_weight	continuous	Demographic weight
education_level	categorical	Highest level of education achieved
education_num	continuous	Total years of education
marital_status	categorical	Marital status
occupation	categorical	Type of occupation
relationship	categorical	Relationship of survey responder to the head of household
race	categorical	Ethnic origin
sex	binary	Biological sex
capital_gain	continuous	Profits made from the sale of real estate, investments and personal property
capital_loss	continuous	Losses incurred when capital assets decrease in value
hours_per_week	continuous	Average number of hours worked per week
native_country	categorical	Country of origin
income_category	categorical	Income in relation to 50k

We note that there are 14 attributes, 6 of which are continuous and 8 are categorical. This includes the *weight* feature assigned to each observation by the researchers at the Population Division of the Census Bureau. These weights are controlled to independent estimates of the civilian non-institutional population of the US, using 3 sets of controls: a single cell estimate of the population 16+ for each state; controls for Hispanic Origin by age and sex; and controls by Race, age and sex. People with similar demographic characteristics should therefore have similar weights, with one important caveat: because the CPS sample is a collection of 51 separate state samples, each with its own probability of selection, such similarities only apply to observations from the same state.

B.Exploratory Data Analysis

In order to identify trends and patterns in the census data, we must conduct some exploratory data analysis. Our findings will guide the methods employed in this exercise, and help us determine whether any transformations must be made to the data. These include but are not limited to: removing attributes highly correlated with other features, applying log transformations or scaling/standardizing the values of continuous attributes, and removing attributes with very few unique values or close to zero variation in our outcome variable.

I.Data at a Glance

After importing the training and testing sets separately and removing the unknowns, we assign the column names listed above. Note that for ease of understanding, these variable names differ slightly from the column names listed on the UCI Machine Learning Repository. Next, we examine the structure of each subset to ensure that they are structurally identical because we want to explore these trends over the entire sample, as opposed to examining the training and testing sets separately. The results are as follows:

Training Set

variable	class	levels
age	integer	NA
workclass	Factor w/ 8 levels	“Federal-gov”, “Local-gov”, “Never-worked”, “Private”, …
final_weight	integer	NA
education_level	Factor w/ 16 levels	“10th”, “11th”, “12th”, “1st-4th”, …
education_num	integer	NA
marital_status	Factor w/ 7 levels	“Divorced”, “Married-AF-spouse”, “Married-civ-spouse”, …
occupation	Factor w/ 14 levels	“Adm-clerical”, “Armed-Forces”, “Craft-repair”, …
relationship	Factor w/ 6 levels	“Husband”, “Not-in-family”, “Other-relative”, “Own-child”, …
race	Factor w/ 5 levels	“Amer-Indian-Eskimo”, “Asian-Pac-Islander”, “Black”, “Other”, …
sex	Factor w/ 2 levels	“Female”, “Male”
capital_gain	integer	NA
capital_loss	integer	NA
hours_per_week	integer	NA
native_country	Factor w/ 41 levels	“Cambodia”, “Canada”, “China”, “Columbia”, …
income_category	Factor w/ 2 levels	“<=50K”, “>50K”

Testing Set

variable	class	levels
age	integer	NA
workclass	Factor w/ 8 levels	“Federal-gov”, “Local-gov”, “Never-worked”, “Private”, …
final_weight	integer	NA
education_level	Factor w/ 16 levels	“10th”, “11th”, “12th”, “1st-4th”, …
education_num	integer	NA
marital_status	Factor w/ 7 levels	“Divorced”, “Married-AF-spouse”, “Married-civ-spouse”, …
occupation	Factor w/ 14 levels	“Adm-clerical”, “Armed-Forces”, “Craft-repair”, …
relationship	Factor w/ 6 levels	“Husband”, “Not-in-family”, “Other-relative”, “Own-child”, …
race	Factor w/ 5 levels	“Amer-Indian-Eskimo”, “Asian-Pac-Islander”, “Black”, “Other”, …

variable	class	levels
sex	Factor w/ 2 levels	“Female”, “Male”
capital_gain	integer	NA
capital_loss	integer	NA
hours_per_week	integer	NA
native_country	Factor w/ 40 levels	“Cambodia”, “Canada”, “China”, “Columbia”, …
income_category	Factor w/ 2 levels	“<=50K.”, “>50K.”

Looking at the structure of each subset reveals that the levels of our variable of interest (income_category), are syntactically different. The levels in the testing set have an unnecessary period at the end of each expression.

We therefore recode the levels in the testing data so they match the training set:

```
levels(testing$income_category)[1] <- "<=50K"
levels(testing$income_category)[2] <- ">50K"
```

Moreover, a quick look at the relationship between the **education_level** and **education_num** variables reveals that the latter does not correspond to the total number of years of education. Rather, it is simply a numeric representation of the **education_level** attribute:

Relationship between education_level and education_num features: Training Set

education_level	education_num
Preschool	1
1st-4th	2
5th-6th	3
7th-8th	4
9th	5
10th	6
11th	7
12th	8
HS-grad	9
Some-college	10
Assoc-voc	11
Assoc-acdm	12
Bachelors	13
Masters	14
Prof-school	15
Doctorate	16

Relationship between education_level and education_num features: Training Set

education_level	education_num
Preschool	1
1st-4th	2

education_level	education_num
5th-6th	3
7th-8th	4
9th	5
10th	6
11th	7
12th	8
HS-grad	9
Some-college	10
Assoc-voc	11
Assoc-acdm	12
Bachelors	13
Masters	14
Prof-school	15
Doctorate	16

We can therefore remove the **education_num** feature, as it is a redundant variable that provides no additional information about our population. We remove it from both the testing and training data frames.

We can now proceed to bind the rows and explore the full dataset. Excluding the unknowns, our census sample has **45,222** observations.

```
## 'data.frame':    45222 obs. of  14 variables:
##  $ age            : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass      : Factor w/ 8 levels "Federal-gov",..: 7 6 4 4 4 4 4 6 4 4 ...
##  $ final_weight   : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education_level: Factor w/ 16 levels "10th","11th",..: 10 10 12 2 10 13 7 12 13 10 ...
##  $ marital_status : Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ occupation     : Factor w/ 14 levels "Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4 ...
##  $ relationship   : Factor w/ 6 levels "Husband","Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
##  $ race           : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
##  $ sex            : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 1 2 1 2 ...
##  $ capital_gain   : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital_loss   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week : int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native_country : Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 5 39 23 39 39 39 ...
##  $ income_category: Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 2 2 ...
##  - attr(*, "na.action")= 'omit' Named int  15 28 39 52 62 70 78 94 107 129 ...
##   ..- attr(*, "names")= chr  "15" "28" "39" "52" ...
```

First, we look at the prevalence of each class of our outcome variable. What is relative proportion of people making more than $50,000 a year in our sample?

income_category	n	pct	percentage
<=50K	34014	0.752156	75.2%
>50K	11208	0.247844	24.8%

We note that at 75.4%, the prevalence of people with incomes below or equal to $50,000/year is much higher than that of people making more than $50,000/year. The implication of the uneven prevalence is that when it comes time to assessing our algorithm, it will be more useful to look at a precision/recall curve as opposed to a ROC curve. This curve plots the true positive rate (sensitivity) against specificity, so that we can gauge how for each class of our income variable(<= 50K and >50K), the probability of a **true positive** given a **predicted positive** changes in relation to the true positive rate. The high prevalence of people with incomes equal to or below the $50,000 threshold also means that: instead of taking the balanced accuracy (F1 score) generated from the confusion matrix at face value, we should compute

a weighted score using the f_meas() function, adjusting the *Beta* accordingly.

II.Deep Dive

II.1 Categorical Attributes

We now proceed to take a deeper dive into a selection of features one by one, starting with the categorical variables. These are **workclass, education_level, marital_status, occupation, relationship, race, sex, and native_country**. These are all factor variables so we take a look at the levels contained in each.

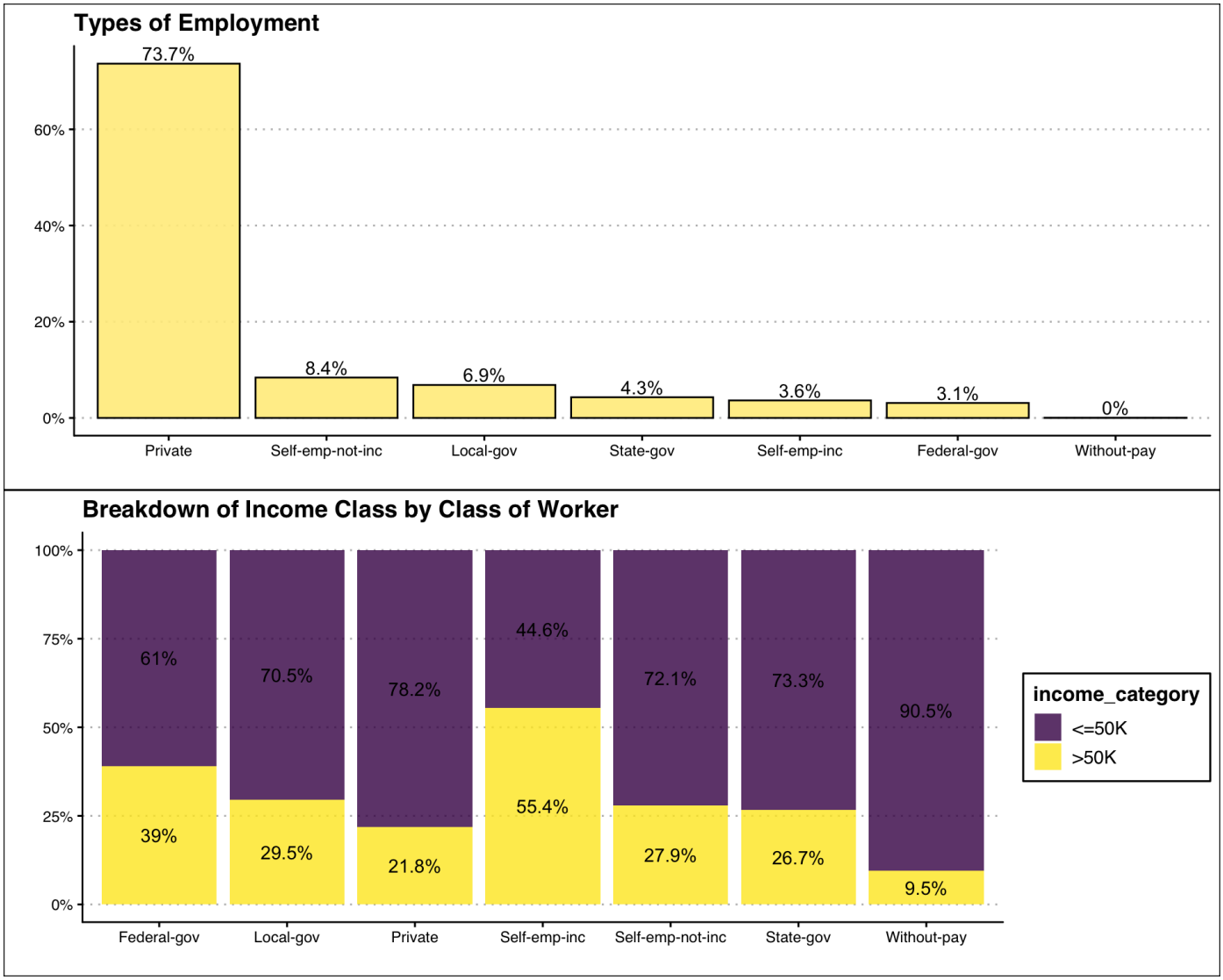
We notice that the variable **work_class** (class of worker) has an unused level ("never-worked"), so we drop it from the data.

```
summary(data_all$workclass)
```

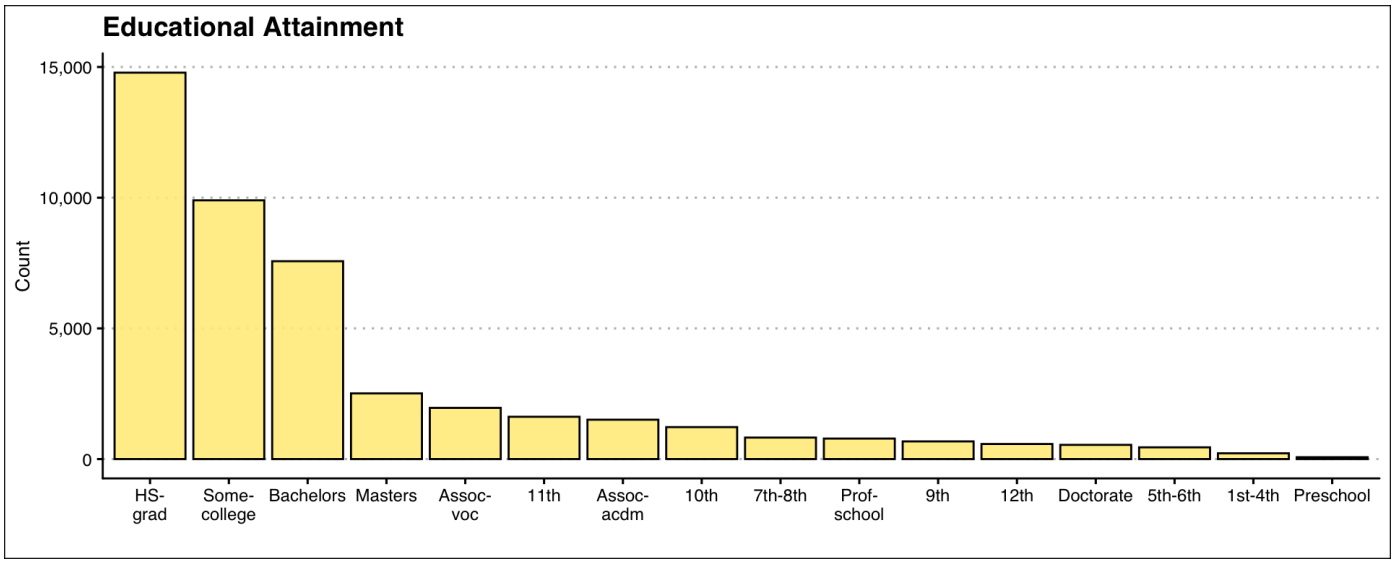
```
##      Federal-gov        Local-gov     Never-worked          Private
##             1406             3100                0            33307
##     Self-emp-inc Self-emp-not-inc        State-gov      Without-pay
##             1646             3796             1946               21
```

```
#Remove unused level "never_worked"
  data_all$workclass <- droplevels(data_all$workclass)
```

Upon summarizing the data along this feature, we note that most survey respondents (73.7%) are private sector workers, and that the highest concentration of individuals earning more than 50K/year is found among the incorporated self-employed workers (self-emp-inc).



Next, we look at the **education_level** variable. We see that there is an unnecessary level of detail.



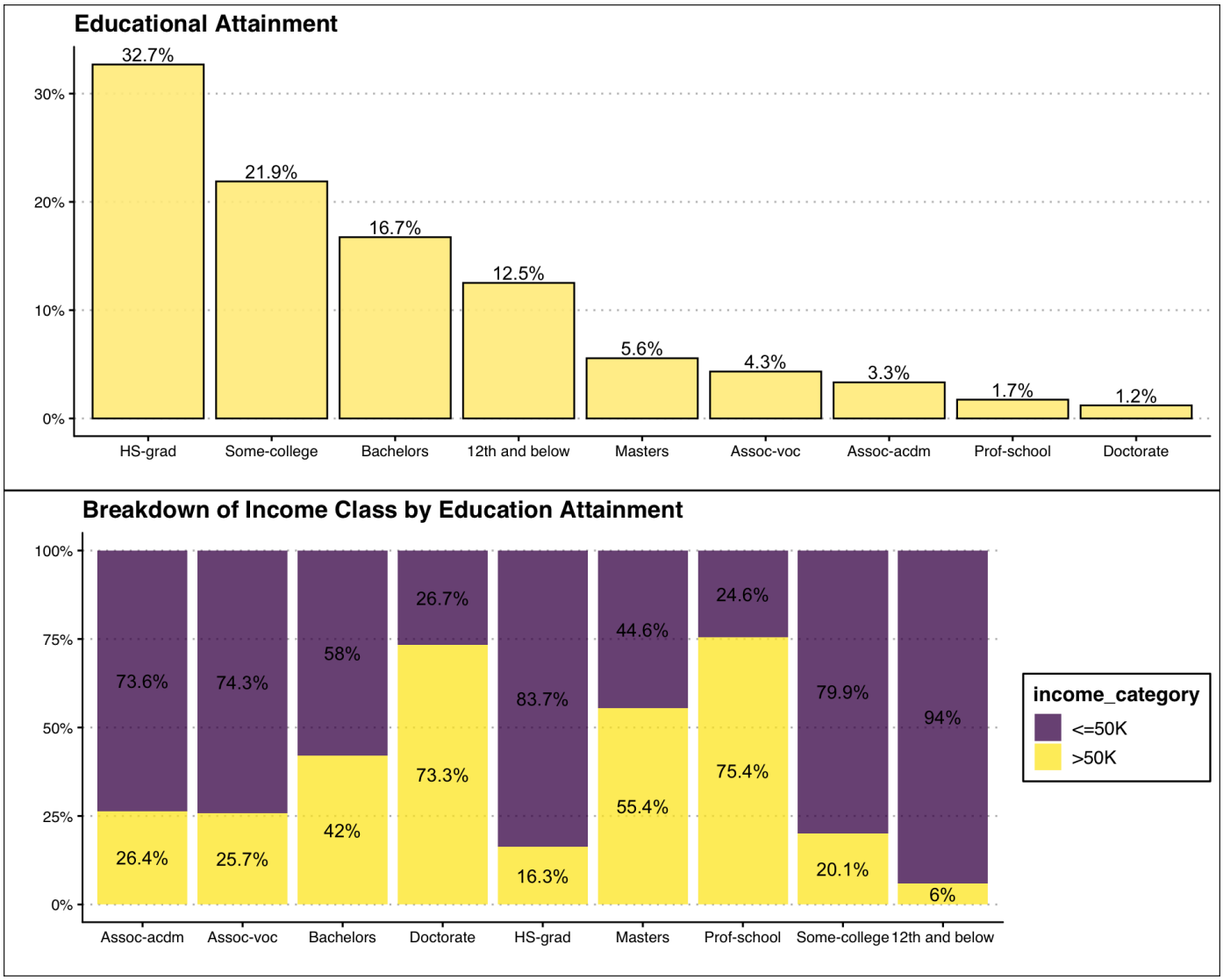
Distinguishing between people who completed 7th-8th grade, versus 10th, versus 12th grade, and so on, would create additional degrees of freedom that are unlikely to add value to our predictive model. We therefore recode the variable so that all levels below *High School Graduate* are combined into a single class which we call *12th and below*.

```
data_all$education_level <- combineLevels(data_all$education_level, c('Preschool', '1st-4th', '5th-6th', '7th-8th', '9th', '10th', '11th', '12th'), newLabel = "12th and below")
```

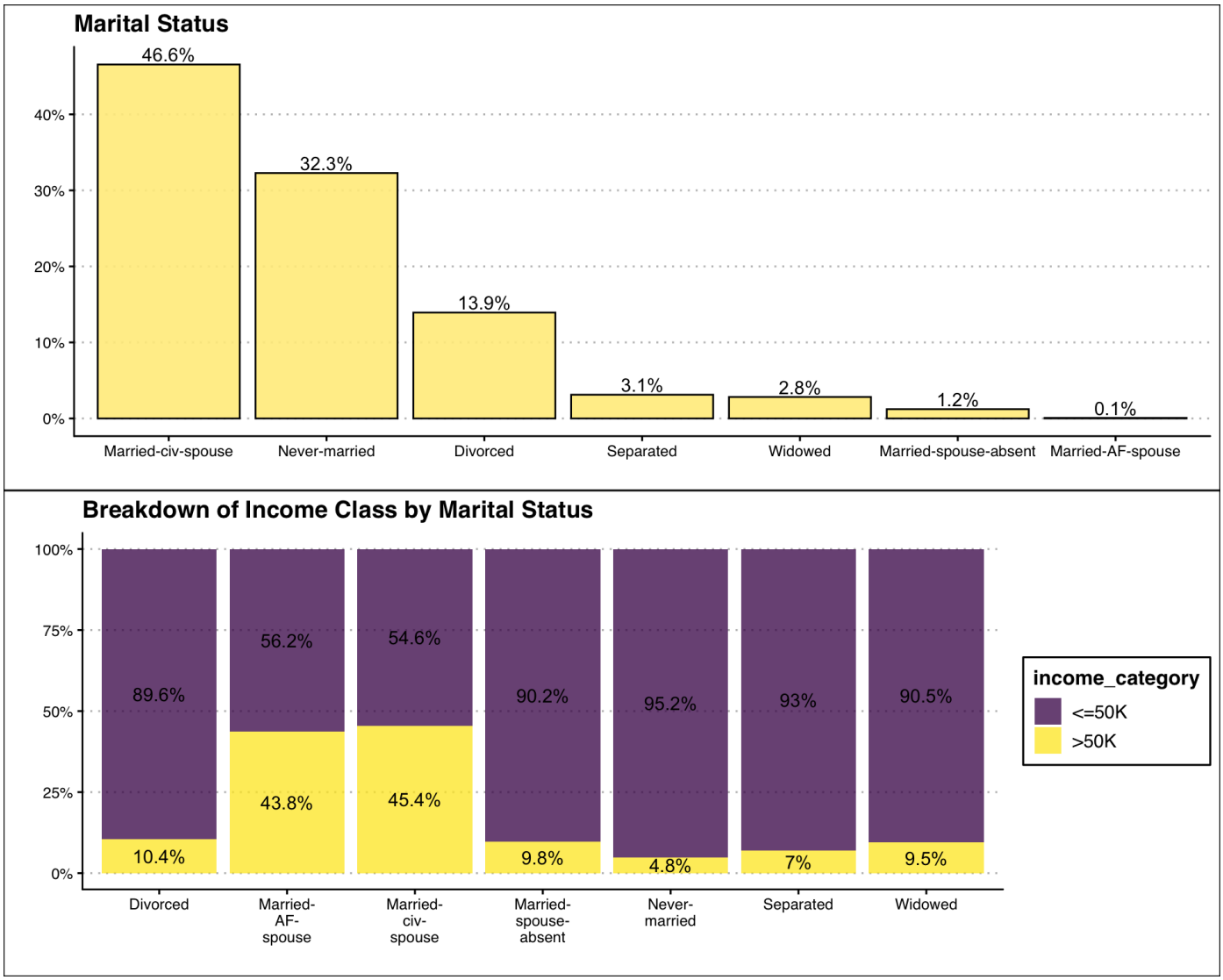
```
levels(data_all$education_level)
```

```
## [1] "Assoc-acdm"     "Assoc-voc"      "Bachelors"      "Doctorate"
## [5] "HS-grad"        "Masters"        "Prof-school"    "Some-college"
## [9] "12th and below"
```

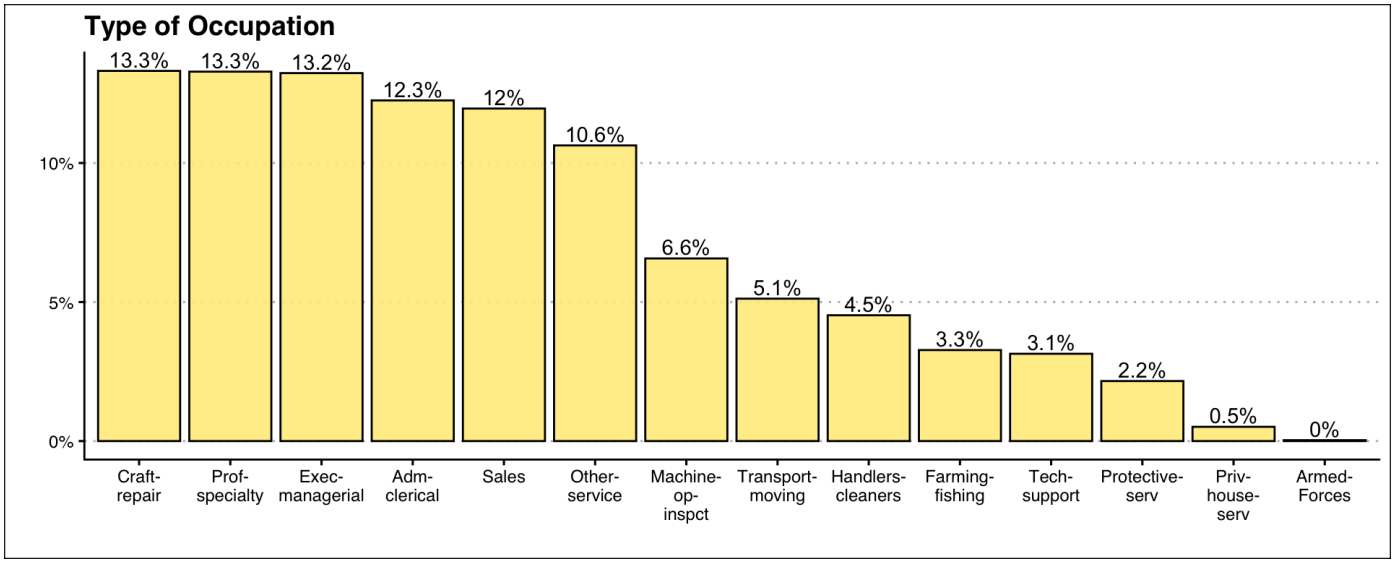
The resulting summary shows that high school graduates are the most common type of worker (32.7%), followed by people who completed some college (21.9%) and those who obtained a Bachelor's degree. We also see that the smallest proportion of people making less than 50K/year is found among those with educational attainment of 12th grade and below, while the highest proportion (75.4%) is found among those with professional degrees beyond a Bachelor's degree (*Prof-school*).

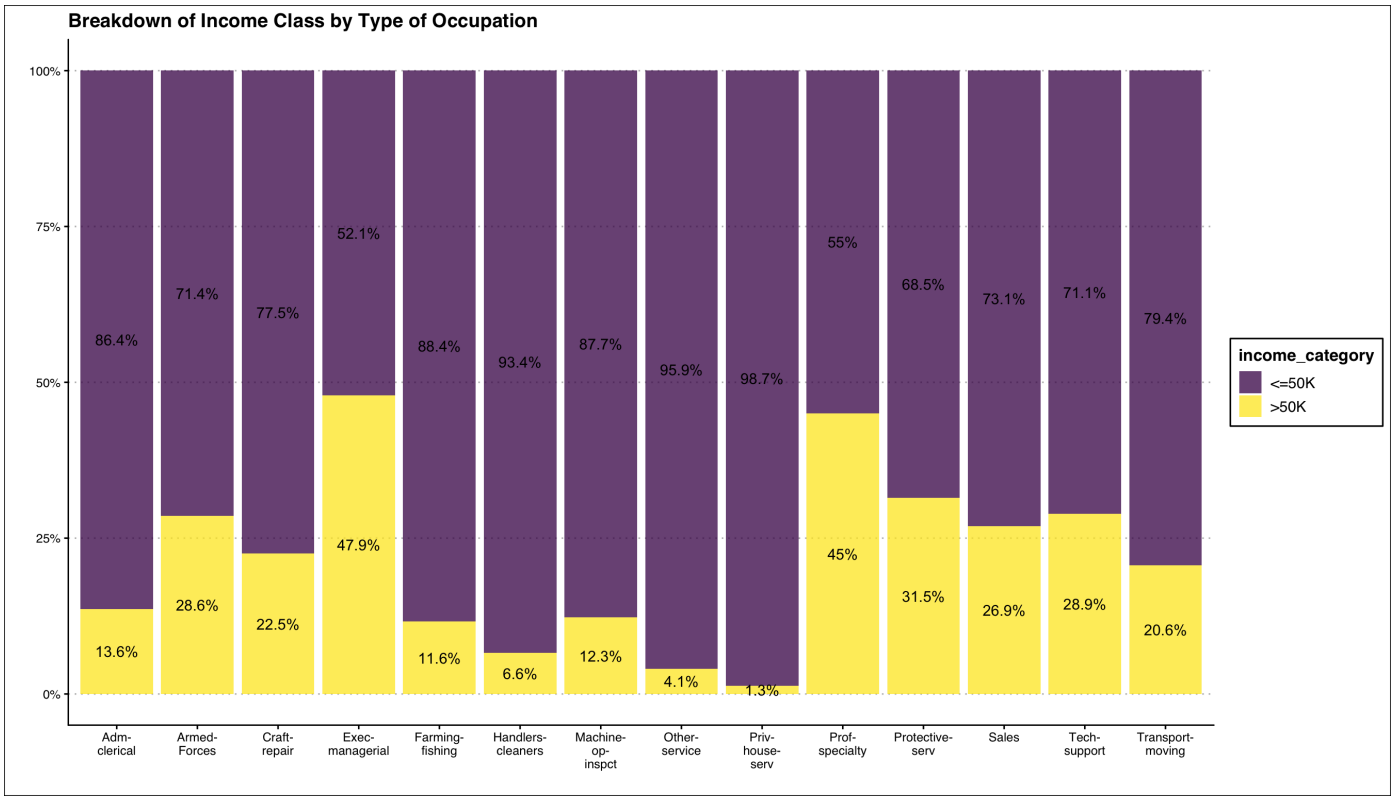


Next, we examine the **marital_status** variable. We find that civilian married individuals make up the greatest proportion of the data (46.6%), and that these same people are more likely to be earning more than 50K/year (45.4%).

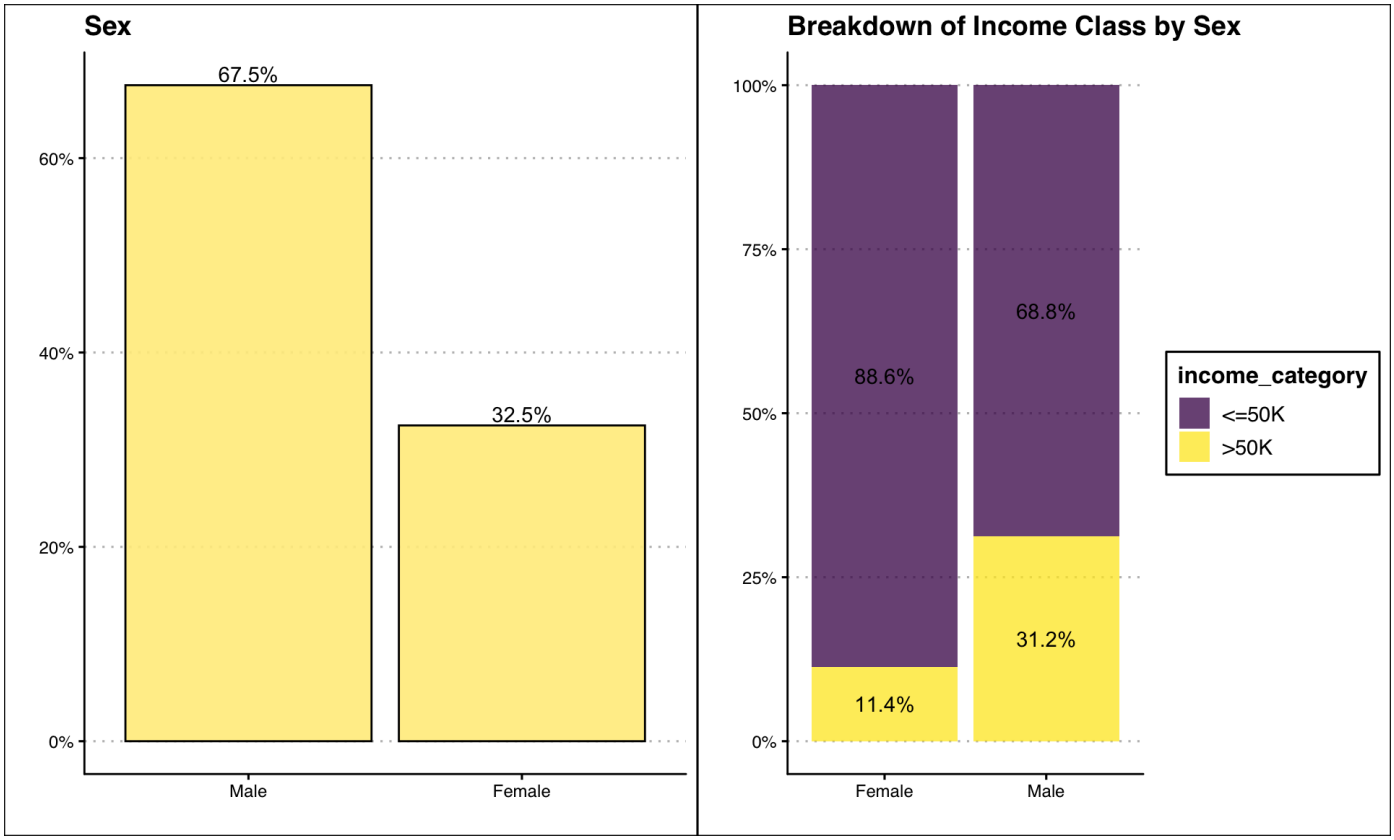


Looking at the **occupation** variable, we note that occupations are fairly evenly distributed in the sample, and that, unsurprisingly, individuals in managerial roles (*Exec-managerial*) and those with a white-collar specialty (*Prof-specialty*) are most likely to make above 50K/year, with 47.9% and 45% of respondents in that income category respectively.





The **sex** variable reveals that the majority of survey responders (67.5%) are male, and that men are more likely than women to earn more than 50K/year (32.1% versus 11.4%).



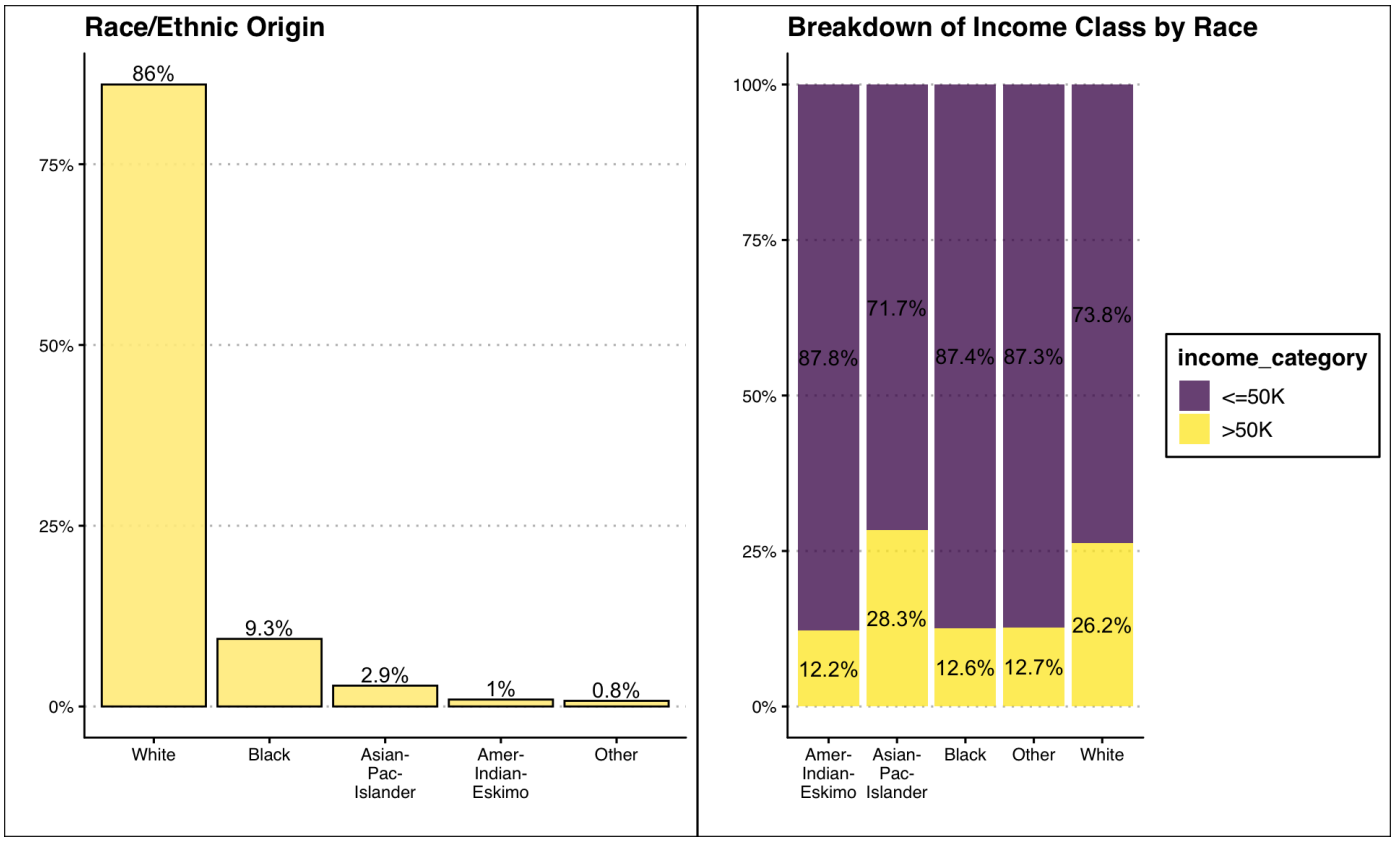
We skip over the **relationship** variable because the metadata does not provide enough information on what it means. While census bureau documentation explains that the **relationship** question denotes each household member's relationship to the primary householder (the person who owns or rents the housing unit), the variable in our dataset does not contain a class which is the equivalent of "self". The levels and their frequencies are as follows:

```
##        Husband  Not-in-family Other-relative      Own-child      Unmarried
##          18666          11702           1349           6626           4788
##           Wife
##           2091
```

The absence of a class equivalent to "self" suggests that the UCI sample excludes primary householders altogether, yet the fact that most

survey respondents are male (67.5%) and “husband” is the most common class of the **relationship** variable suggests otherwise. Another possibility is that **relationship** actually denotes the role of the individual in the household. This would explain the high prevalence of males and “husband” as the relationship type; however we do not have enough information to ascertain our hypothesis.

We move on to examine variability along the **race** variable, and note that an overwhelming majority (86%) of the survey respondents are white, and that the rate of people earning more than 50K/year is highest among the Asian/Pacific Islander ethnic group (28.3%), followed by caucassians at 26.2%.



Finally, we examine the structure of the **native_country** variable. A summary of this categorical feature shows that it contains 41 different levels, most of which have very few observations.

```
##  [1] "Cambodia"                  "Canada"
##  [3] "China"                     "Columbia"
##  [5] "Cuba"                      "Dominican-Republic"
##  [7] "Ecuador"                   "El-Salvador"
##  [9] "England"                   "France"
## [11] "Germany"                   "Greece"
## [13] "Guatemala"                 "Haiti"
## [15] "Holand-Netherlands"        "Honduras"
## [17] "Hong"                      "Hungary"
## [19] "India"                     "Iran"
## [21] "Ireland"                   "Italy"
## [23] "Jamaica"                   "Japan"
## [25] "Laos"                      "Mexico"
## [27] "Nicaragua"                 "Outlying-US(Guam-USVI-etc)"
## [29] "Peru"                      "Philippines"
## [31] "Poland"                    "Portugal"
## [33] "Puerto-Rico"               "Scotland"
## [35] "South"                     "Taiwan"
## [37] "Thailand"                  "Trinadad&Tobago"
## [39] "United-States"             "Vietnam"
## [41] "Yugoslavia"
```

Because keeping so many levels as an input is likely to add unnecessary degrees of freedom and bring down our model’s performance, we collapse countries of origin into regional categories. We assign these categories to a new variable, **native_region**:

```
Asia <- c("Cambodia", "India", "Laos", "Thailand", "Vietnam", "Hong", "Iran", "China", "Japan", "Philippine
s", "Taiwan")

    Europe <- c("France", "Italy", "Poland", "Scotland", "Germany", "Portugal", "Yugoslavia", "England", "Gr
eece", "Holand-Netherlands", "Hungary", "Ireland")

    North_America <- c("Outlying-US(Guam-USVI-etc)", "Canada", "United-States", "Puerto-Rico")

    Latin_America_Carrib <- c("Columbia", "Ecuador", "Guatemala", "Honduras", "Cuba", "El-Salvador", "Haiti"
, "Jamaica", "Mexico", "Peru", "Trinadad&Tobago", "Dominican-Republic", "Nicaragua")

    Unknown <- c("South")

    data_all <- data_all %>% mutate(native_region = case_when(native_country %in% Asia ~ "Asia",
                                    native_country %in% Europe ~ "Europe",
                                    native_country %in% North_America ~ "North_America",
                                    native_country %in% Latin_America_Carrib ~ "Latin_America_Carrib",
                                    native_country %in% Unknown ~ "Unknown"))
```

Our **native_region** feature will replace the **native_country** variable. It has the following values:

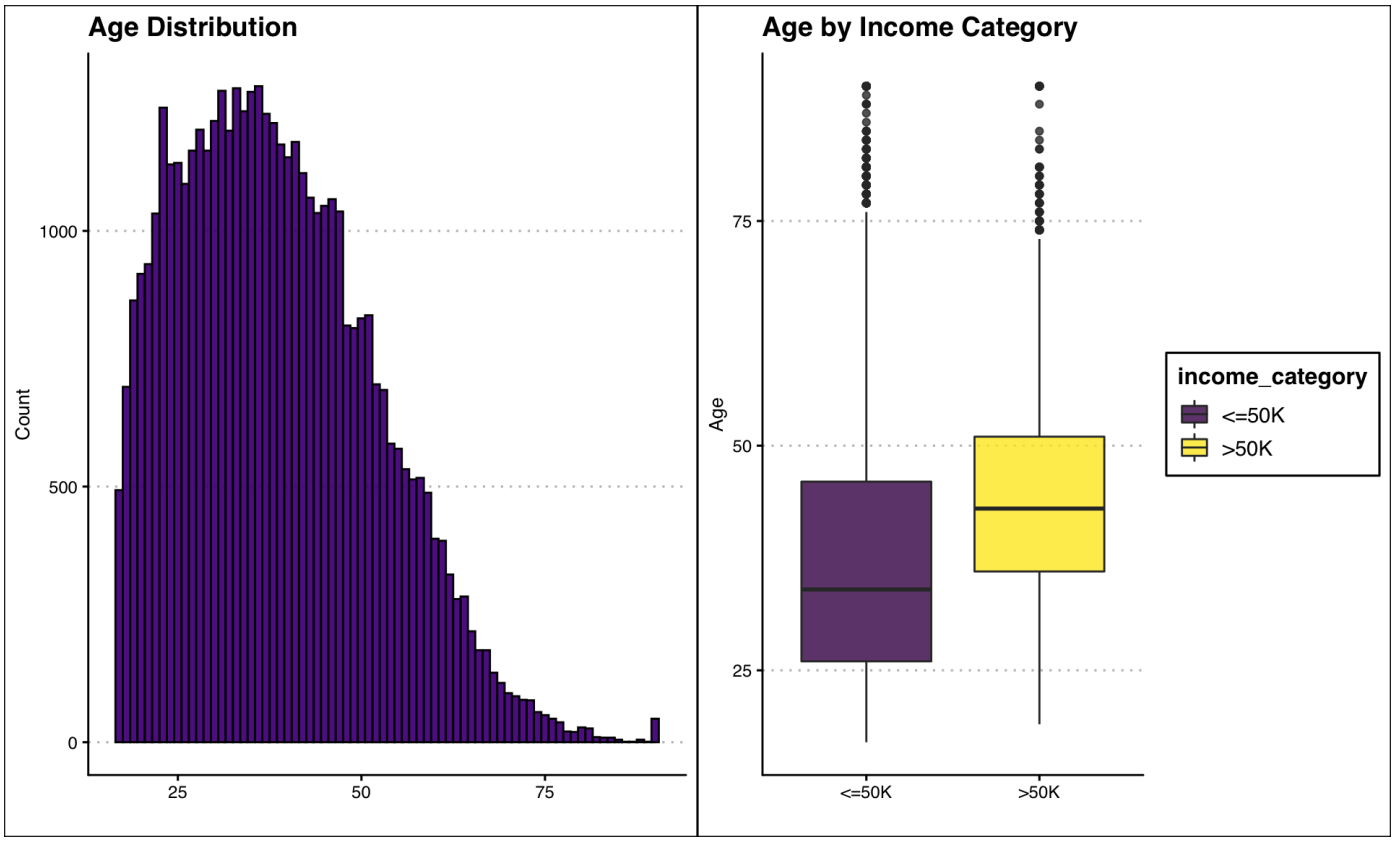
```
## [1] "Asia"                 "Europe"               "Latin_America_Carrib"
## [4] "North_America"        "Unknown"
```

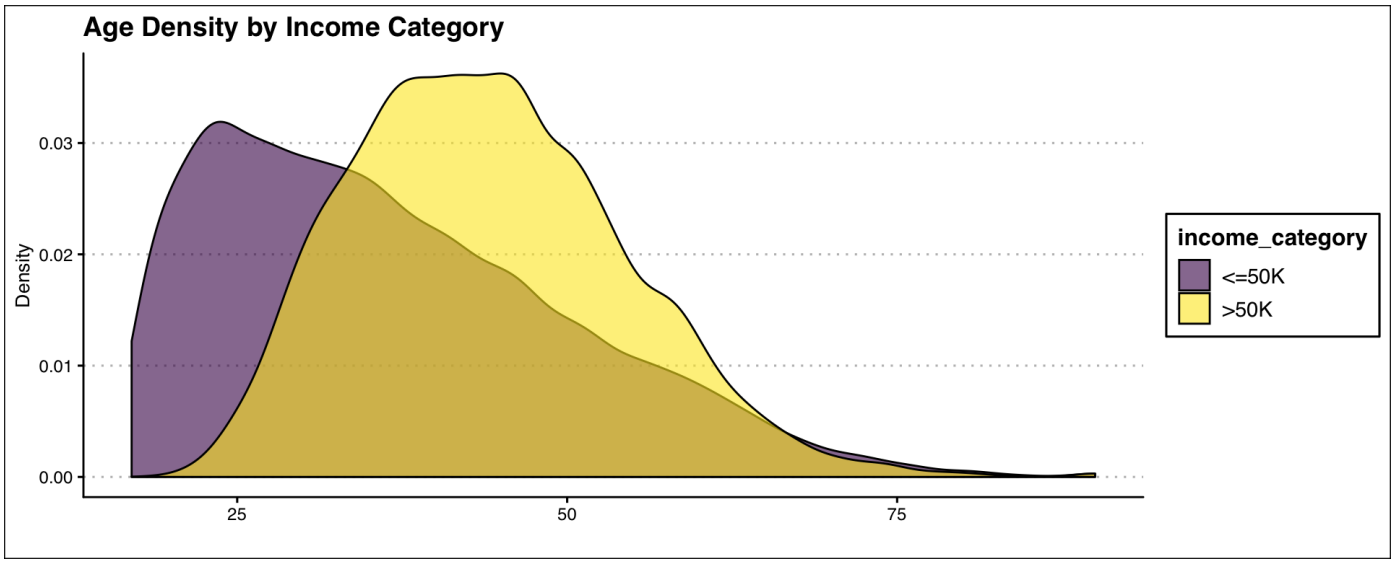
Here, “Unknown” corresponds to the entries with the ambiguous value of “South”.

II.2 Continuous Attributes

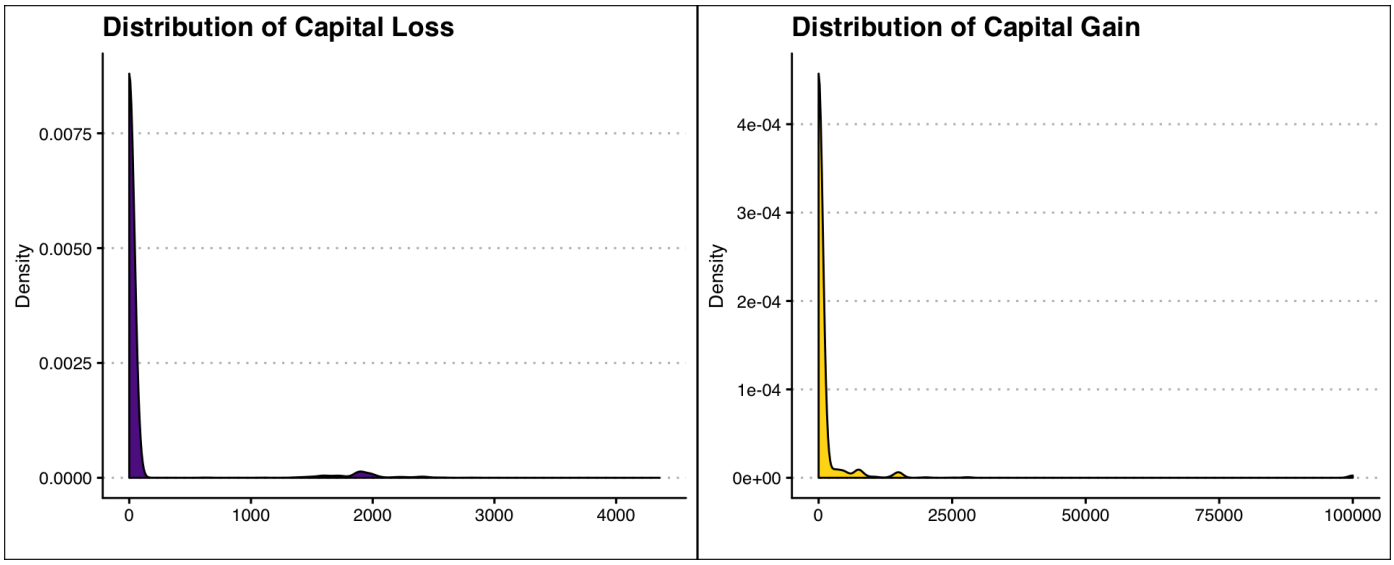
We move on to examine the distribution of values along our continuous predictors. We see that 50% of survey respondents are between the ages of 28 and 47, and and that the age variable has a right-skewed distribution, as there is a higher concentration of individuals below the median age than above it. This what we would expect from sampling a population of working-age individuals. We also see from looking at boxplots and density plots of age along our two income categories that the age distribution of people earning more than 50K/year is older and and has less variability than that of people earning below that threshold.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   28.00   37.00   38.55   47.00   90.00
```





Finally, we examine the distribution of values contained in the **capital_loss** and the **capital_gain** variables. Density plots of these attributes suggest that the values are heavily skewed towards zero.



We perform a summary computation to ascertain this and see that there are indeed very few instances of values that differ from zero. 91.6% of the survey respondents have no capital gain, and and 95.3% have no capital loss.

```
data_all %>% filter(capital_gain == 0) %>% summarise(value = 0, prevalence = n()/45222*100)
```

```
    data_all %>% filter(capital_loss == 0) %>% summarise(value = 0, prevalence = n()/45222*100)
```

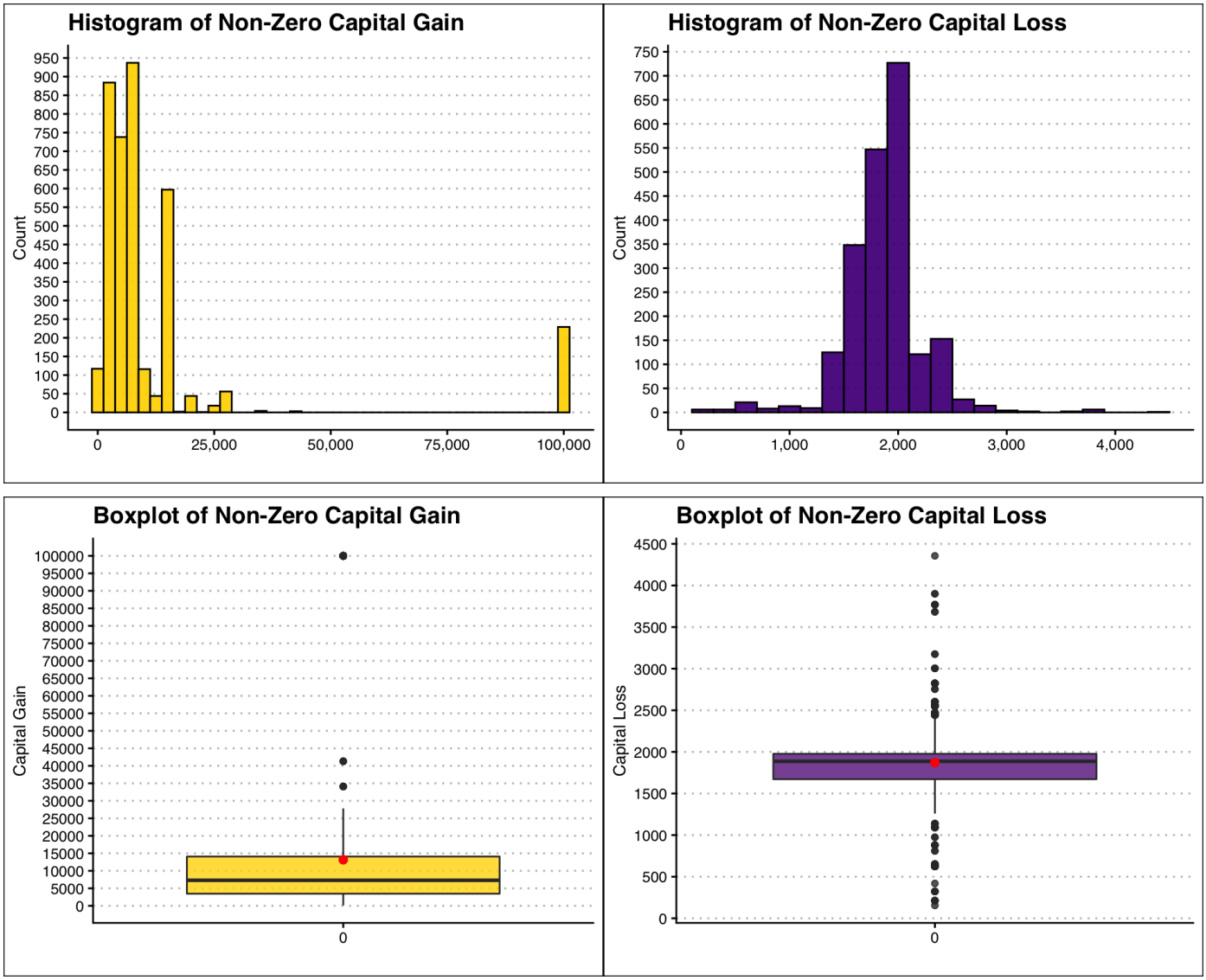
Such a high prevalence of zeros in a continuous variable is likely to distort our predictions. We must therefore create bins so that we can treat both **capital_gain** and **capital_loss** as categorical features in our predictive model. In order to compute these bins, we first explore the distribution of the non-zero values for each variable. We compute the quantiles and get the below results:

Non-Zero Capital Gain and Capital Loss Quantiles

	Capital_Gain	Capital_Loss
0%	114	155
25%	3464	1672
50%	7298	1887
75%	14084	1977
100%	99999	4356

We note that the interquartile range for **capital_gain** (10,620) is significantly larger than the IQR for **capital_loss** (305). Then, we create histograms and boxplots of the non-zeros values of **capital_gain** and **capital_loss** to further investigate the distribution of these attributes

before creating cutoff values for our bins.



We see that the distribution of non-zero values of **capital_gain** is left-skewed, with most pepple having a capital gain between $2,500 and $27,500 - the most common value being $7,500 - and that there is a handful of outliers with a capital gain of $100,000. The distribution for **capital_loss** on the other hand, is more symmetric, with most values falling between $1,400 and $2,400 - the most common value being $2,000 - and a large number of outliers above and below this window.

We then create the following levels for **capital_gain** as a factor variable:

- Values equal to or below the first quartile of non-zero values ($3,464): Low
- Values between the first and third quartile of non-zero values ($3,464 to $14,084): Medium
- Values above the third quartile of non-zero values ($14,084): High

We do the same for **capital_loss** and create the following levels:

- Values equal to or below the first quartile of non-zero values ($1,672): Low
- Values between the first and third quartile of non-zero values ($1,672 to $1,977): Medium
- Values above the third quartile of non-zero values ($1,977): High

The code looks like this:

```
data_all <- data_all %>%
      mutate(cap_gain = ifelse(capital_gain <= 3464, " Low",
                               ifelse(capital_gain > 3464 & capital_gain < 14084, " Medium", " High")))


    data_all$cap_gain <- factor(data_all$cap_gain,
                                ordered = TRUE,
                                levels = c(" Low", " Medium", " High"))

    data_all <- data_all %>%
      mutate(cap_loss = ifelse(capital_loss <= 1672, " Low",
                               ifelse(capital_gain > 1672 & capital_gain < 1977, " Medium", " High")))


    data_all$cap_loss <- factor(data_all$cap_loss,
                                ordered = TRUE,
                                levels = c(" Low", " Medium", " High"))
```

After making these modifications to our features, we proceed to conduct a correlation analysis in order to assess how interrelated they are. This is important because in order to build an efficient predictive model, it is best to only include variables that **uniquely** explain some amount of variance in the outcome. Because correlations are most easily computed on numeric variables, we convert all the values of our categorical variables to numeric levels after subsetting our data to exclude the outcome variable (**income_category**) and the demographic **weight** variable. Simple integer encoding is not enough however, since our categorical variables are nominal and have a multiplicity of levels (some are binary while others have many levels). We therefore one-hot encode all the categorical variables so each only has two values: *Yes (1) or No (0)*.