Census Income Data

Adult

Author Name

29 April, 2022

Contents

```
#Please know that you can use a html output but you need to keep the sectioning.

#Please Reference your figures and tables so that it is readable

#Each update is important to keep for grading
```

1 Update 5

- Please put a bulleted list of things you have accomplished since the last update
  - Include things that didn't work but you tried
  - Things you are planning on doing
  - Questions that you might have on your project.
- Reference the sections and figures you are discussing here

2 Update 1

2.0.0.0.1 The goal is to train a binary classifier to predict the income which has

2.0.0.0.2 two possible values '>50K' and '<50K'.

```
library(dplyr)

library(ggplot2)

library(plyr)
```

2.0.0.0.3 Importing required libraries.

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(gmodels)

library(grid)
```

```r
library(vcd)

library(scales)

library(ggthemes)
```

```r
df = read.csv('adult.csv',header=T,na.strings =c("?","NA"))
```

2.0.0.0.4 Importing the dataset adult.csv.

2.1 The missing values in the dataset are indicated by "?".

2.2 Let's get more information about the training data.

```r
summary(df)
```

```
##       age         workclass             fnlwgt         education
##  Min.   :17.00   Length:32561       Min.   :  12285   Length:32561
##  1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character
##  Median :37.00   Mode  :character   Median : 178356   Mode  :character
##  Mean   :38.58                      Mean   : 189778
##  3rd Qu.:48.00                      3rd Qu.: 237051
##  Max.   :90.00                      Max.   :1484705
##  education.num   marital.status      occupation         relationship
##  Min.   : 1.00   Length:32561       Length:32561       Length:32561
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.08
##  3rd Qu.:12.00
##  Max.   :16.00
##      race               sex             capital.gain    capital.loss
##  Length:32561       Length:32561       Min.   :    0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0
##                                        Mean   : 1078   Mean   :  87.3
##                                        3rd Qu.:    0   3rd Qu.:   0.0
##                                        Max.   :99999   Max.   :4356.0
##  hours.per.week  native.country        income
##  Min.   : 1.00   Length:32561       Length:32561
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.44
##  3rd Qu.:45.00
##  Max.   :99.00
```

```r
str(df)
```

```
## 'data.frame':    32561 obs. of  15 variables:
```

```
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : chr  NA "Private" NA "Private" ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 .
##  $ education     : chr  "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: chr  "Widowed" "Widowed" "Widowed" "Divorced" ...
##  $ occupation    : chr  NA "Exec-managerial" NA "Machine-op-inspct" ...
##  $ relationship  : chr  "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ...
##  $ race          : chr  "White" "White" "Black" "White" ...
##  $ sex           : chr  "Female" "Female" "Female" "Female" ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

2.3 There are 32561 samples in the training dataset.

2.4 There are both categorical and numerical columns in the dataset.

2.5 The columns workClass, occupation, native-country have missing values.

2.6 Let's look the numerical and the categorical data with the help of some visualizations.

2.7 Handling Numerical Columns.

2.8 Select the numerical columns using the sapply function.

```r
num_attributes <- which(sapply(df,is.numeric))

print(num_attributes)
```

```
##            age         fnlwgt  education.num   capital.gain   capital.loss
##              1              3              5             11             12
## hours.per.week
##             13
```

2.9 [‘age’, ‘fnlwgt’, ‘education-num’, ‘capital-gain’, ‘capital-loss’, ‘hours-per-week’] are

2.10 numerical columns.

2.11 The variables “age”, “hours-per-week” are self-explanatory.

2.12 The variable “fnlwgt” is sampling weight.

2.13 The variable “education-num” is number of years of education in total.

2.14 The variable “capital-gain/capital-loss” is the income from investment sources other than

2.15 salary/wages.

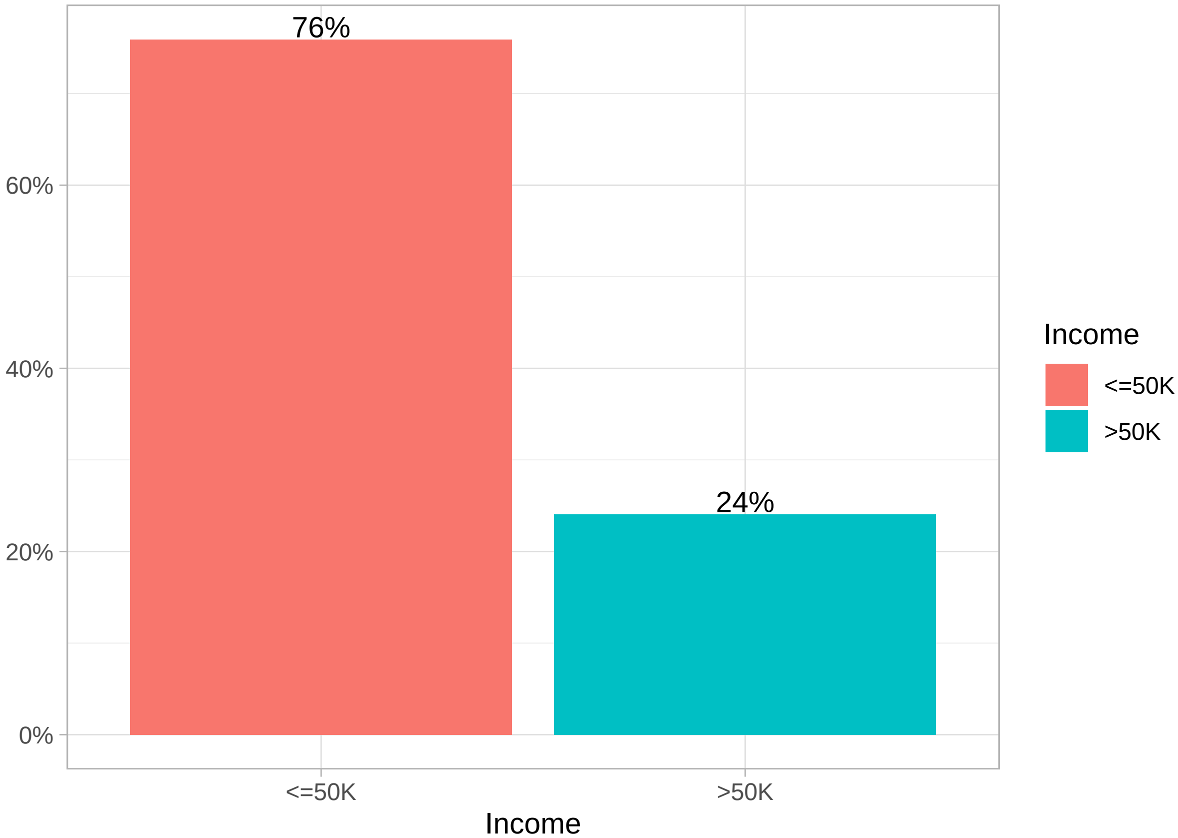
2.16 “fnlwgt” is not related to the target variable income and will be removed before building the

2.17 model

DATA VISUALIZATION

```
library(ggplot2)

ggplot(data = df, mapping = aes(x = df$income, fill = df$income)) + geom_bar(mapping = aes(y =
```

```
## Warning: Use of `df$income` is discouraged. Use `income` instead.
## Use of `df$income` is discouraged. Use `income` instead.
## Use of `df$income` is discouraged. Use `income` instead.
## Use of `df$income` is discouraged. Use `income` instead.
```



2.17.1 The graph obtained shows us the percentage of people earning less than 50K a year and more

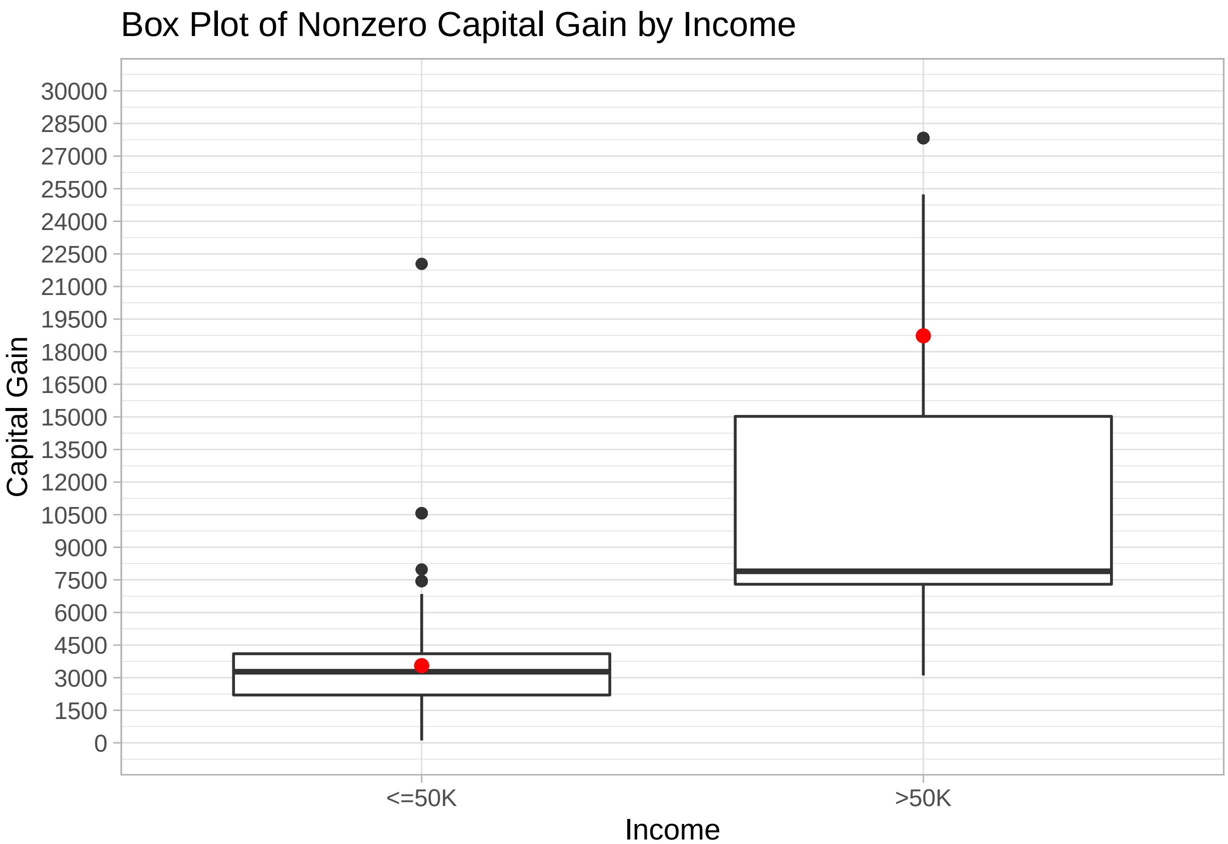
2.18 than 50K. We see that 76% of the participants in the study are paid less than 50K and 24% are

2.18.1 paid more than 50K.

```
ggplot(mapping = aes(x = income, y = capital.gain), data = subset(df, df$capital.gain > 0)) + g
```

2.18.1.0.1 CAPITAL GAIN and CAPITAL LOSS

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



2.18.1.0.2 box plots of capital gain grouped by income. The mean value is depicted with a filled red

###dot and the black horizontal line inside the boxes is the median. We can see that for people

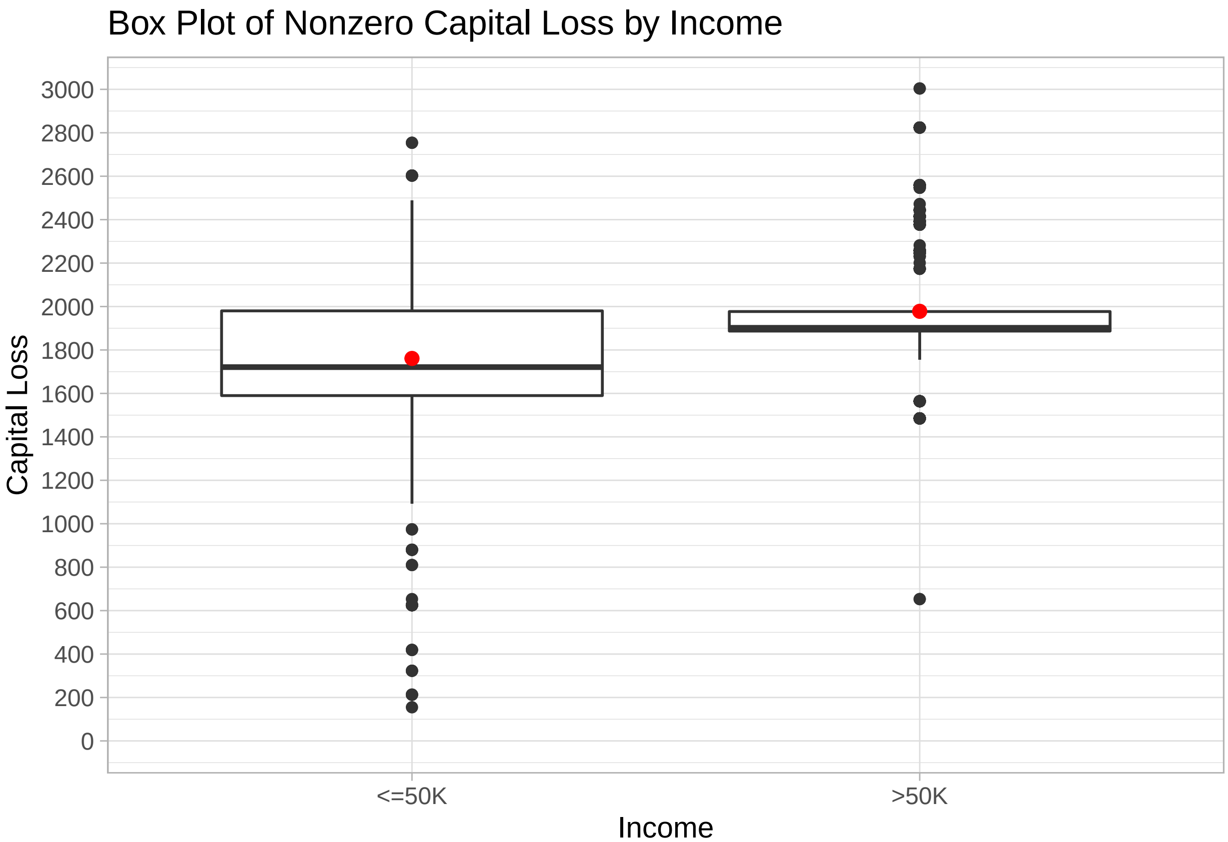
2.18.2 earning more than 50K a year, the bulk of the values (50% of the data points) as well as the

2.18.3 median, and the mean value of the capital gain are significantly greater than these of people

2.18.4 earning less than 50K

```
ggplot(mapping = aes(x = income, y = capital.loss), data = subset(df, df$capital.loss > 0)) + g
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



2.18.4.0.1 As a conclusion, we can say that there is evidence for strong relationship between the

2.18.5 nonzero values of "capital.gain" and "capital.loss", and "income". However, we will not

2.18.6 include these variables in the predictive model because of the extremely high number of zeros

2.18.7 among their values.

```
summary(df$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   28.00   37.00   38.58   48.00   90.00
```

```
IQR(df$age)
```

```
## [1] 20
```

2.18.7.0.1 The median age is 37 years and the mean age is 38 years. The summary shows that at least

2.18.8 50% of the people in the study are between 28 and 48 years old, which makes sense since the

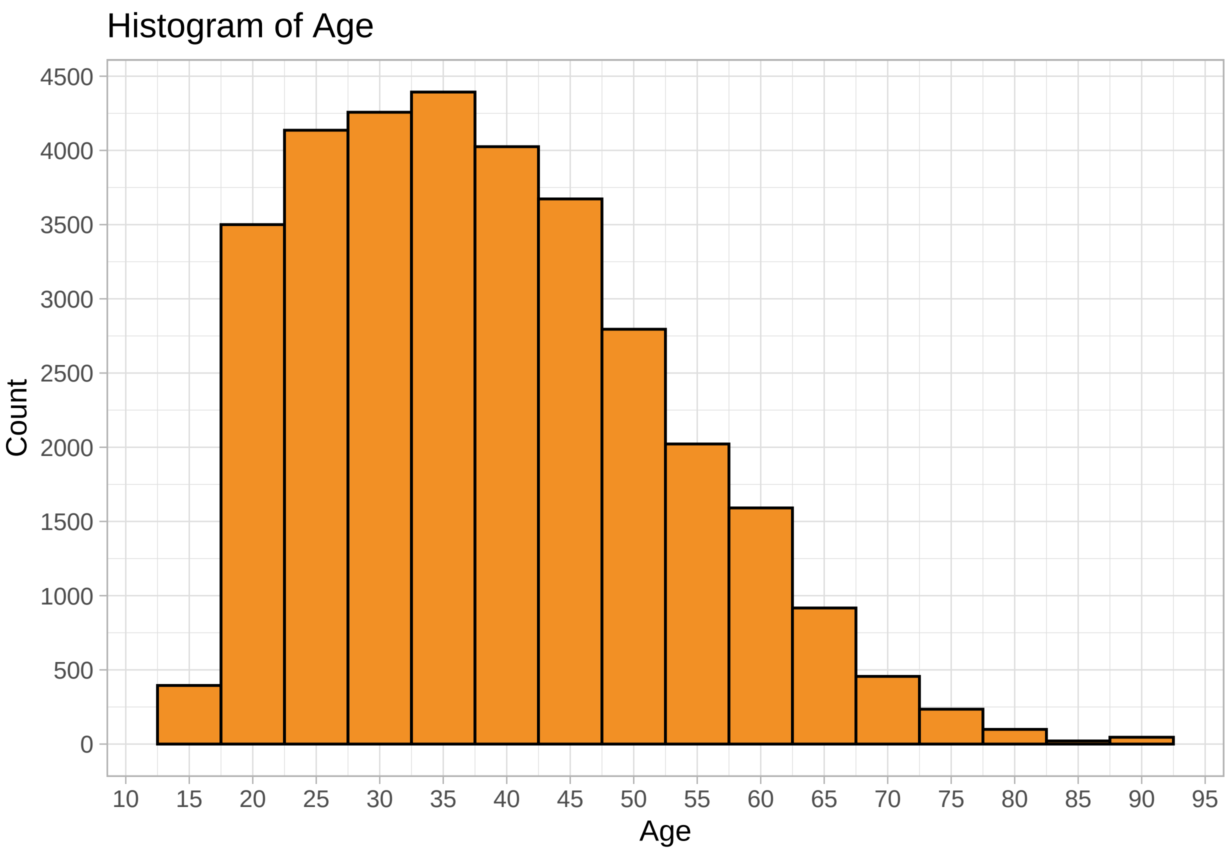
2.18.9 participants in the survey should be of working age. Of course, there are some outliers, such

2.18.10 as individuals being between 75 and 90 years old. To visualize the summary statistic we also

2.18.11 show a box plot of the variable "age":

```
qplot(x = df$age, data = df, binwidth = 5, color = I('black'), fill = I('#F29025'), xlab = "Age
```

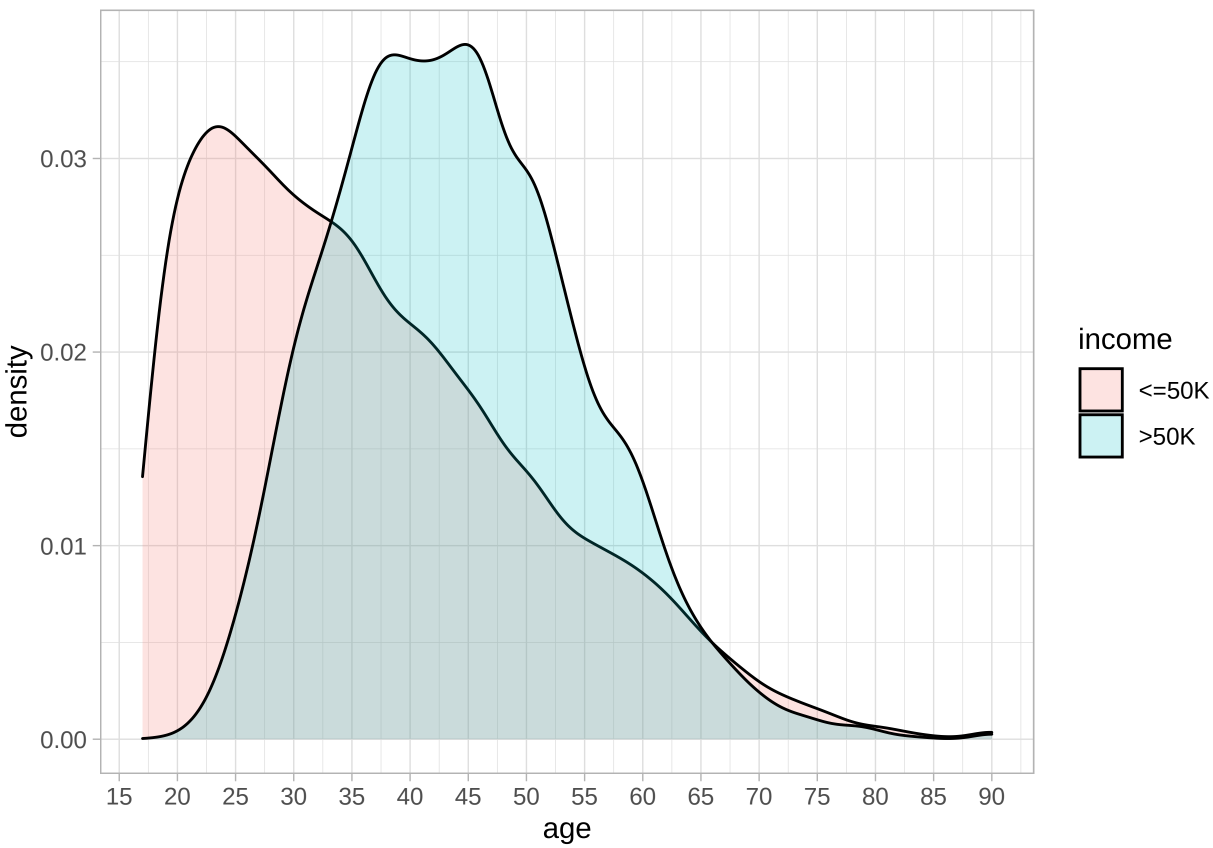
```
## Warning: Use of `df$age` is discouraged. Use `age` instead.
```



2.18.11.0.1 From the histogram of "age" we can see that the bulk of individuals are between 20 and 50

2.18.12 years old

```
ggplot(data = df, aes(age, fill = income)) + geom_density(alpha = 0.2) + scale_x_continuous(bre
```



2.18.12.0.1 The density plot clearly shows that age and income are correlated – people of greater age

2.18.13 have higher income.

```
summary(df$education)
```

```
##    Length     Class      Mode
##     32561 character character
```

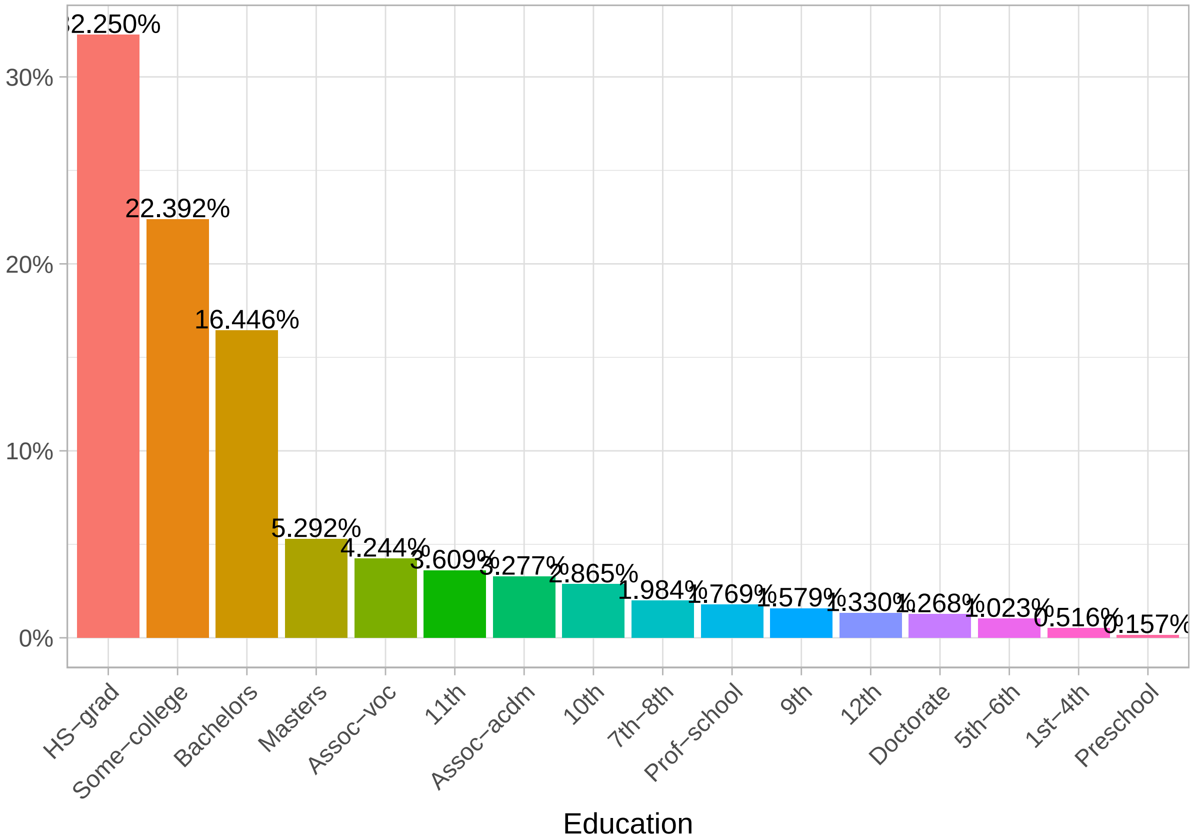
2.18.13.0.1 The majority of people have a high school degree - 10501, college degreee - 7291 and

2.18.14 bachelor degree - 5355. The bar plot below shows the percentage of people belonging to each

2.18.15 category of "education"

```
df$education <- factor(df$education, levels = names(sort(table(df$education), decreasing =TRUE)
ggplot(df, aes(x = df$education, fill = df$education)) + geom_bar(aes(y = (..count..)/sum(..cou
```

```
## Warning: Use of `df$education` is discouraged. Use `education` instead.
## Use of `df$education` is discouraged. Use `education` instead.
## Use of `df$education` is discouraged. Use `education` instead.
## Use of `df$education` is discouraged. Use `education` instead.
```



2.18.15.0.1 Above are the few visualization plots for few variables, to understand the patterns of the

2.18.16 columns along with correlation of those columns with the target variable (Income).

3 Update 2

Since there are no people with education "Preschool" who earn more than 50K a year, as we can see below,

```
nrow(subset(df, df$education == " Preschool" & df$income == " >50K" ))
```

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

We will remove the factor level "Preschool" before we continue further with the analysis.

In order to do that we create a character vector "modified.edu" with elements equal to the factor levels of "education", and then we alter the vector by removing the element " Preschool":

```
modified.edu <- levels(df$education)

modified.edu
```

```
##  [1] "HS-grad"      "Some-college" "Bachelors"    "Masters"      "Assoc-voc"
##  [6] "11th"         "Assoc-acdm"   "10th"         "7th-8th"      "Prof-school"
## [11] "9th"          "12th"         "Doctorate"    "5th-6th"      "1st-4th"
## [16] "Preschool"
```

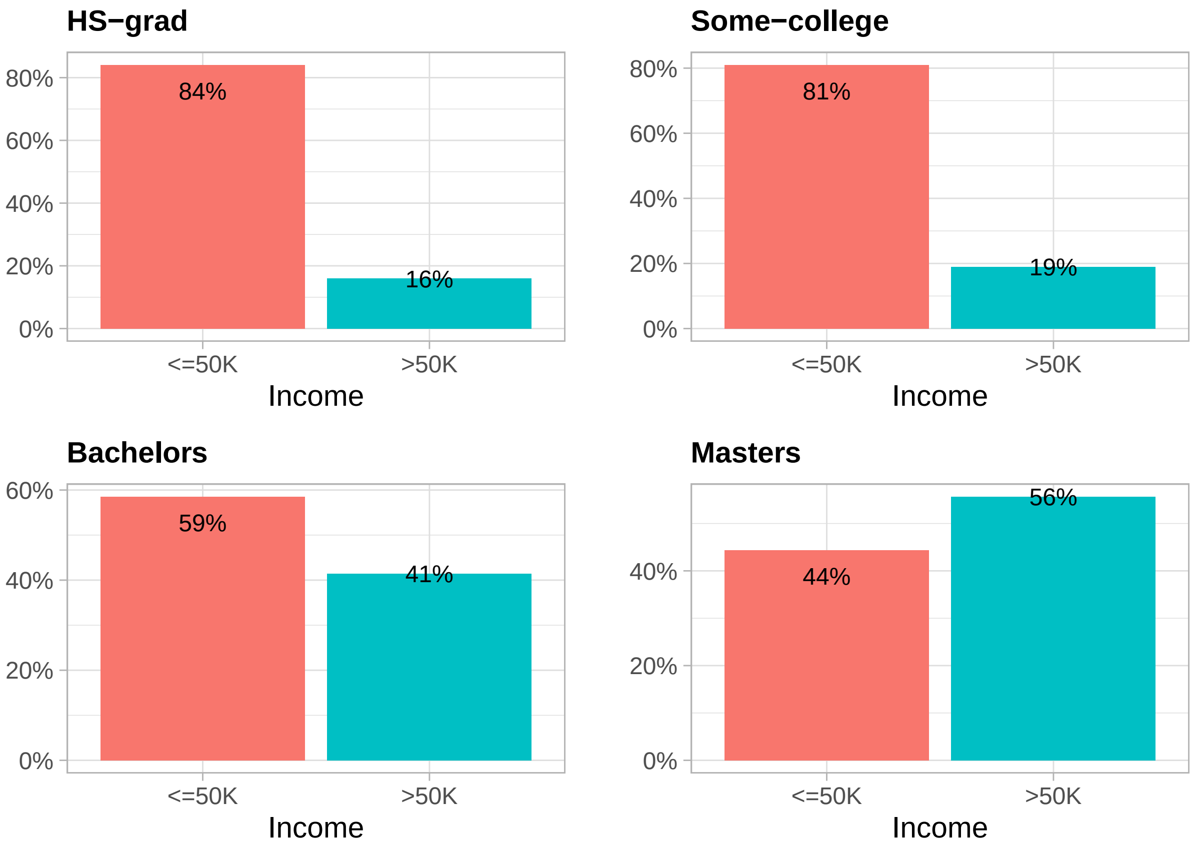
```
modified.edu <- modified.edu[!is.element(modified.edu, "Preschool")]
```

```r
modified.edu
```

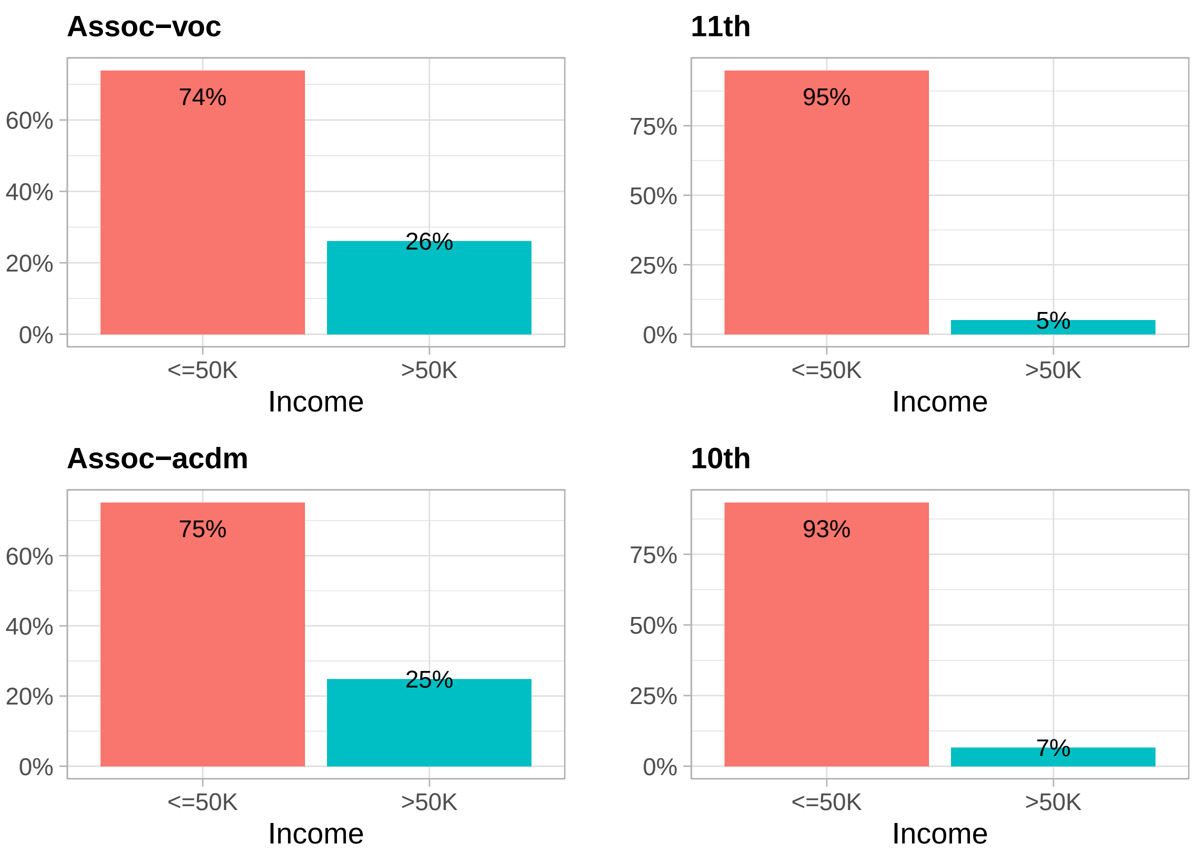
```
##  [1] "HS-grad"      "Some-college" "Bachelors"    "Masters"      "Assoc-voc"
##  [6] "11th"         "Assoc-acdm"   "10th"         "7th-8th"      "Prof-school"
## [11] "9th"          "12th"         "Doctorate"    "5th-6th"      "1st-4th"
```

After that, we display the bar plot of each education category grouped by income:

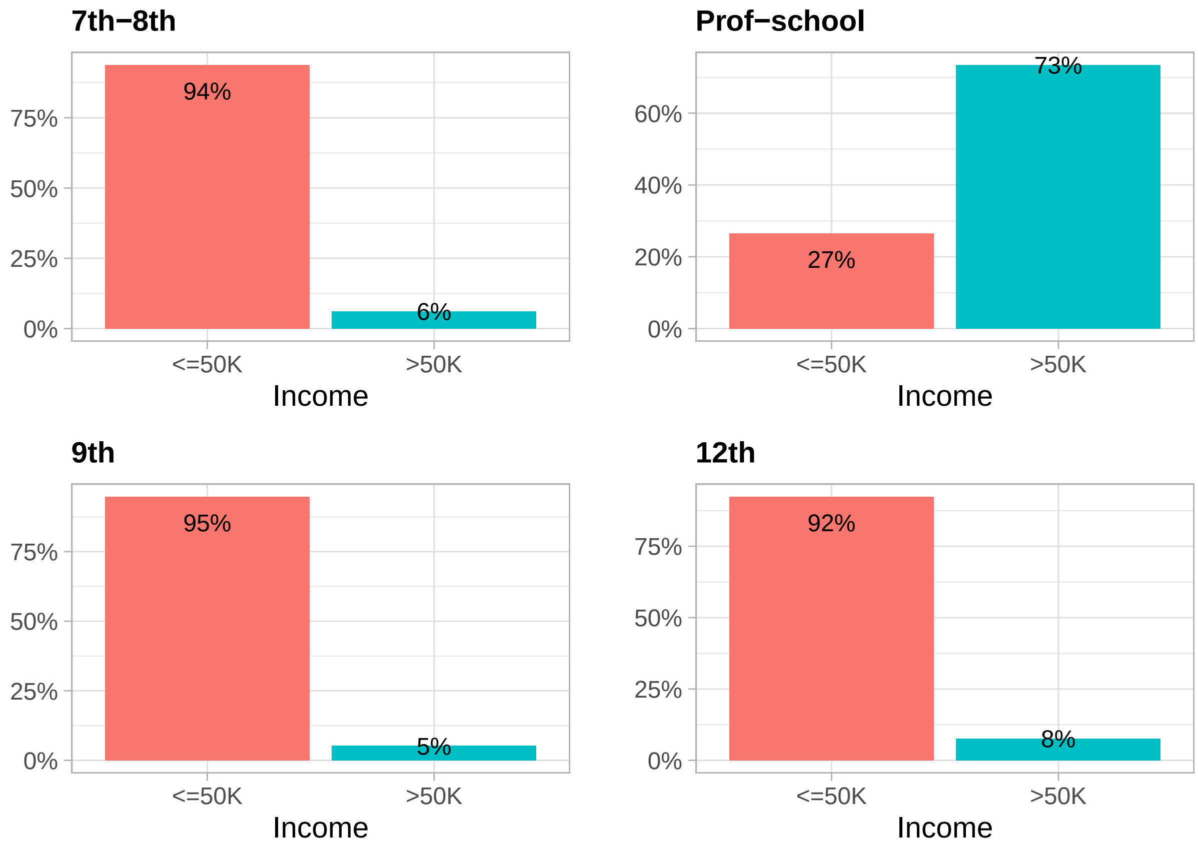
```r
lg.mod.edu <- lapply(modified.edu, function(v){

  ggplot(data = subset(df, df$education == v),
         aes(x = subset(df, df$education == v)$income,
             fill = subset(df, df$education == v)$income)) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                  y = (..count..)/sum(..count..) ),
             stat = "count",
             vjust =  c(2, 0.5),
             size = 3) +
    labs(x = "Income",
         y = "",
         fill = "Income") +
    ggtitle(v) +
    theme(legend.position = 'none',
          plot.title = element_text(size = 11, face = "bold")) +
    scale_y_continuous(labels = percent) })


grid.arrange(grobs = lg.mod.edu[1:4], ncol = 2)
```



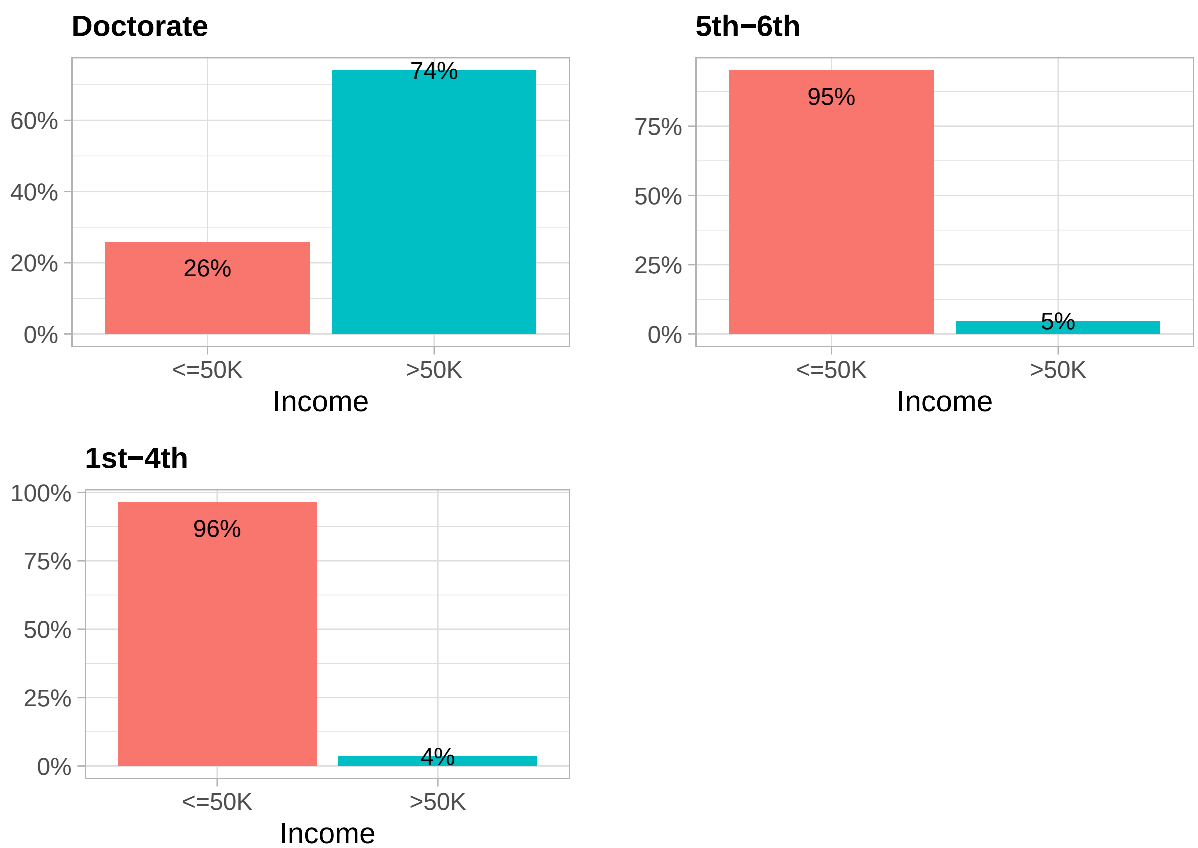

```
grid.arrange(grobs = lg.mod.edu[5:8], ncol = 2)
```




```
grid.arrange(grobs = lg.mod.edu[9:12], ncol = 2)
```




```
grid.arrange(grobs = lg.mod.edu[13:15], ncol = 2)
```



The categories " 1st-4th“," 5th-6th“," 7th-8th“," 9th“," 10th“," 11th" and " 12th" have a very small percentage of people with income greater than 50K a year. The percentage of people with a high school degree who earn more than 50K is also relatively small - 16%. 19% of the individuals in the category " Some-college" earn more than

50K. The biggest percentage of employees (74%), who have an annual income higher than 50K, belongs to the category " Doctorate". The"Prof-school" group is next with 73%, followed by the categories " Masters" - 56% and "Bachelors" - 41%.

```
table(df$marital.status)
```

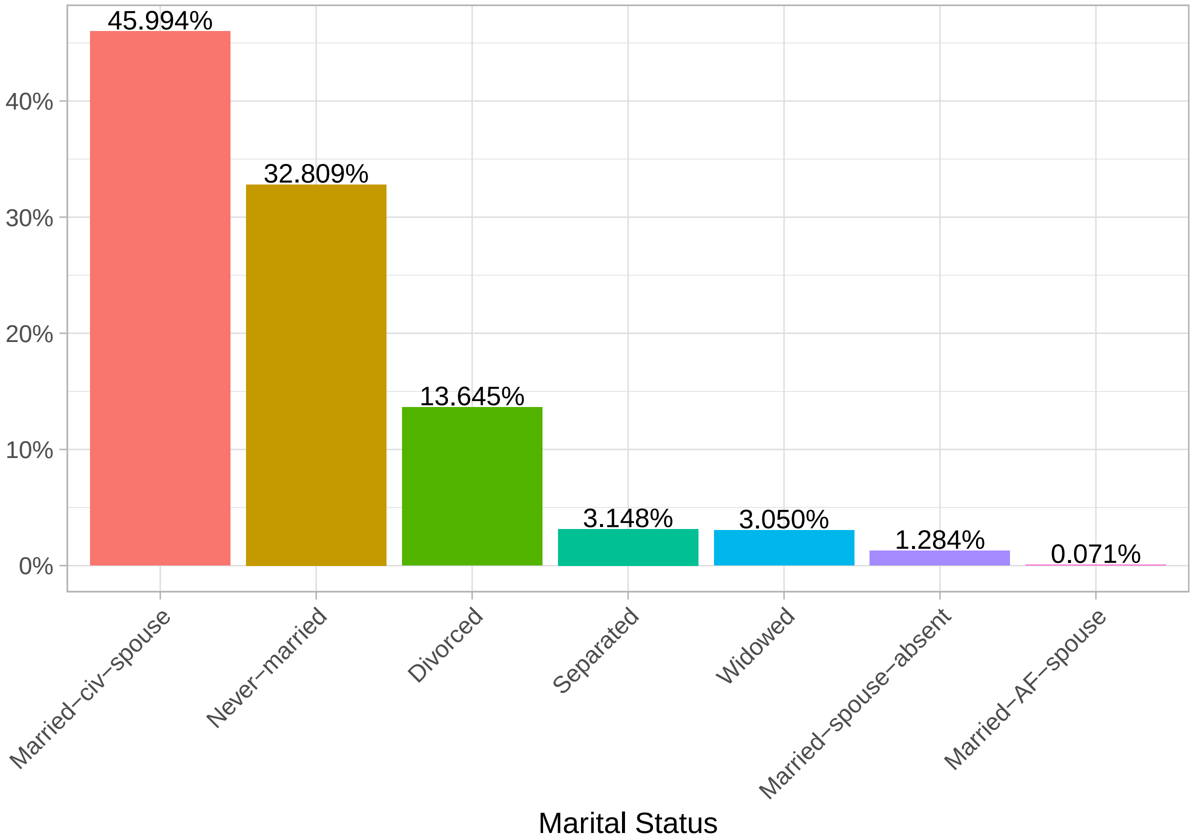
```
##
##              Divorced     Married-AF-spouse    Married-civ-spouse
##                  4443                    23                 14976
## Married-spouse-absent         Never-married             Separated
##                   418                 10683                  1025
##               Widowed
##                   993
```

The biggest number of people are married to a civilian spouse - 14976. A significant number of individuals belong to the group " Never-married" - 10683, followed by divorced people - 4443. A very small number of participants in the study are married to an army spouse - 23.

Below we visualize the percentage of people belonging to each category:

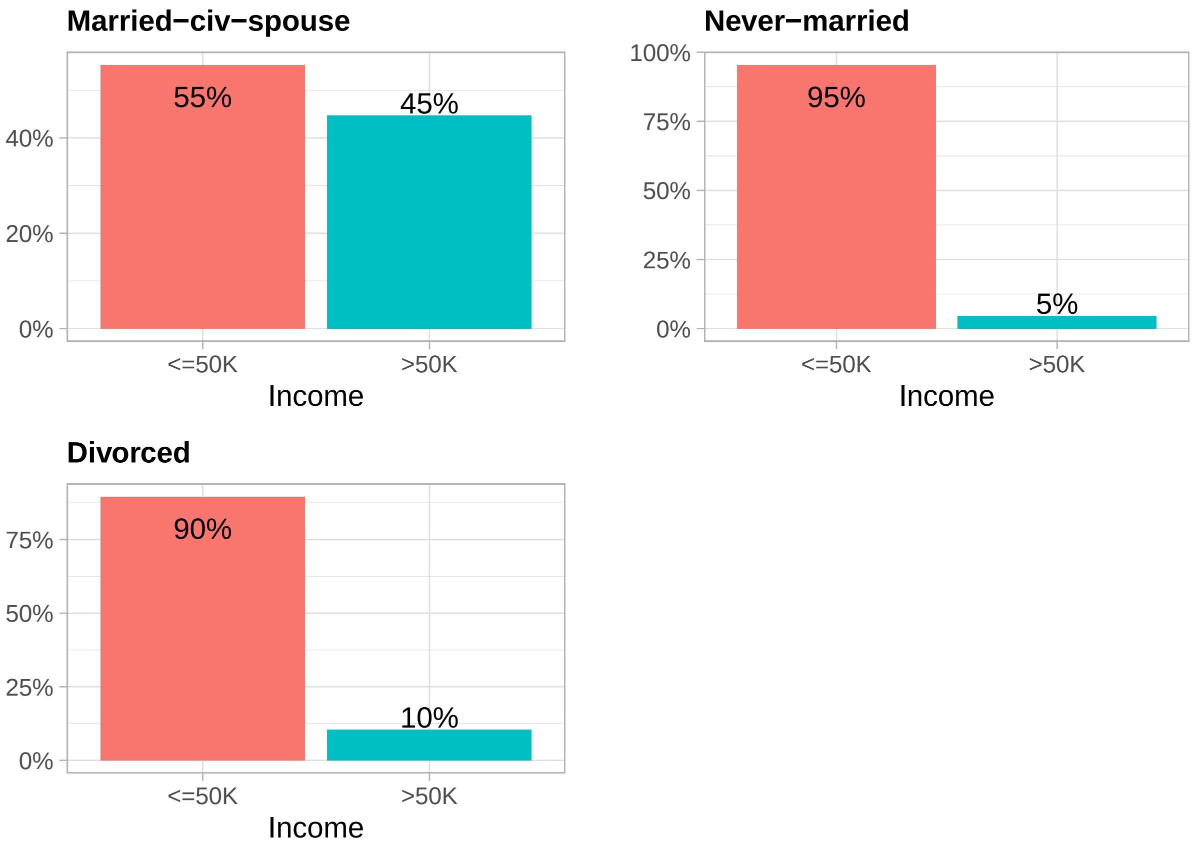
```
df$marital.status <- factor(df$marital.status,
                                levels =
                                    names(sort(table(df$marital.status),


ggplot(df,
       aes(x = df$marital.status, fill = df$marital.status)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = -.1,
            size = 3.5) +
  labs(x = "Marital Status",
       y = "",
       fill = "Marital Status") +
  theme(legend.position = 'none',
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = percent)
```

```
## Warning: Use of `df$marital.status` is discouraged. Use `marital.status` instead.
## Use of `df$marital.status` is discouraged. Use `marital.status` instead.
## Use of `df$marital.status` is discouraged. Use `marital.status` instead.
## Use of `df$marital.status` is discouraged. Use `marital.status` instead.
```



Below we give the bar plots of income grouped by marital status:

```
lp_marital <- lapply(levels(df$marital.status), function(v){

  ggplot(data = subset(df, df$marital.status == v),
         aes(x = subset(df, df$marital.status == v)$income,
             fill = subset(df, df$marital.status == v)$income)) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                  y = (..count..)/sum(..count..) ),
              stat = "count",
              vjust = c(2, -0.1)) +
    labs(x = "Income",
         y = "",
         fill = "Income") +
    ggtitle(v) +
    theme(legend.position = 'none',
          plot.title = element_text(size = 11, face = "bold")) +
    scale_y_continuous(labels = percent) })


grid.arrange(grobs = lp_marital[1:3], ncol = 2)
```



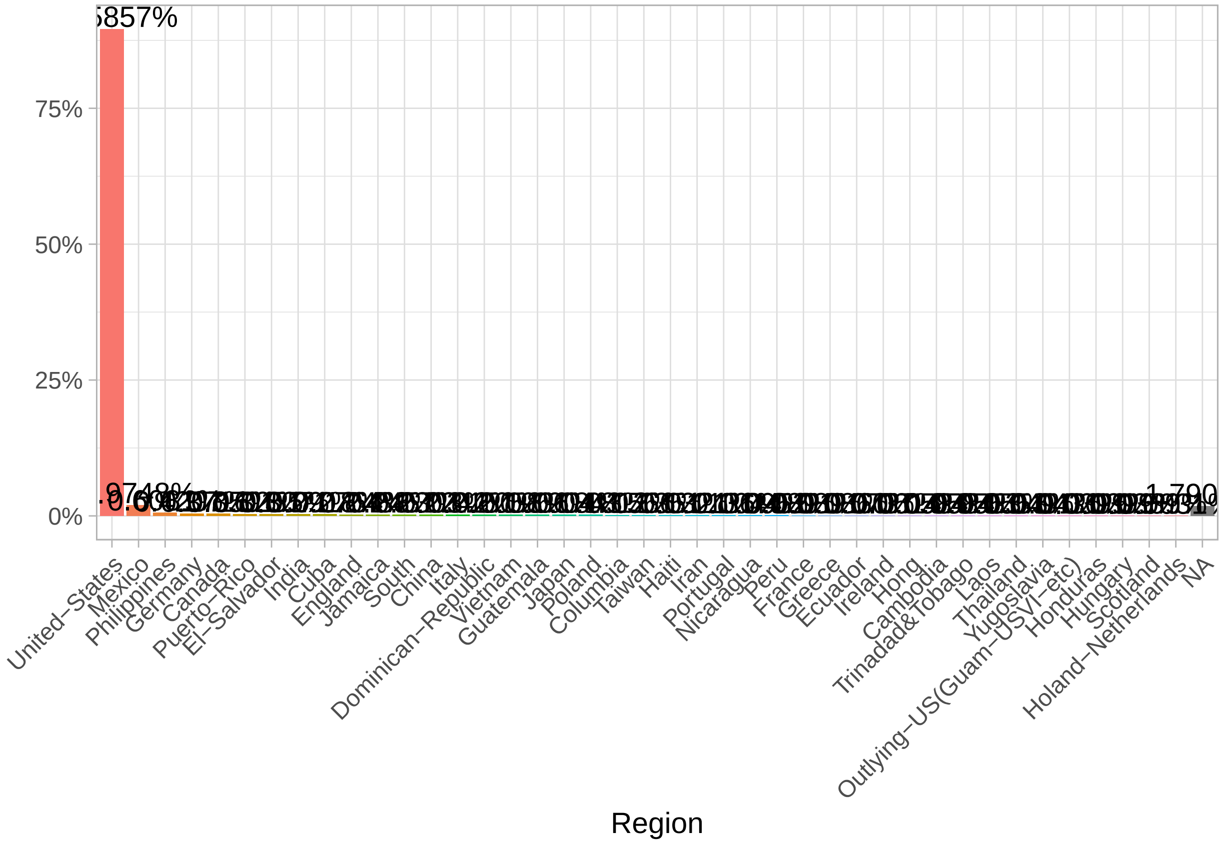

```
grid.arrange(grobs = lp_marital[4:7], ncol = 2)
```

As we see from the graphs above, the biggest percentage of employees with income higher than 50K are those from the category “Married-civ-spouse”. But “Married-AF-spouse“, since there are only 23 observations in this category, we cannot draw trustworthy conclusions regarding the income of the individuals belonging to this group. On the other hand, the random sample for the category" Married-civ-spouse" amounts to 14065 individuals and can be considered representative. For this category, the percentage of people with income of more than 50K is very high - 45%. The same cannot be said for the groups " Divorced"," Never-married"," Married-spouse-absent"," Separated" and " Widowed", where the percentage of people with income higher than 50K varies between 5% and 10%. One explanation as to why people who never got married earn less than married people is that the former group probably contains mostly young individuals who work part-time (for example, students saving for college), as well as younger people as a whole, who are in the beginning of their professional career. This conclusion is also in agreement with the results for the variable"age", where we noticed that the greater the age of an individual, the higher their income. However, the same logic cannot be applied to the other categories with low percentage of individuals with income greater than 50K –" Divorced"," Married-spouse-absent"," Separated" and " Widowed". Therefore these results provide evidence that there is a correlation between income and marital status, which cannot be explained only with the confounding"age" variable.

4 Update 3

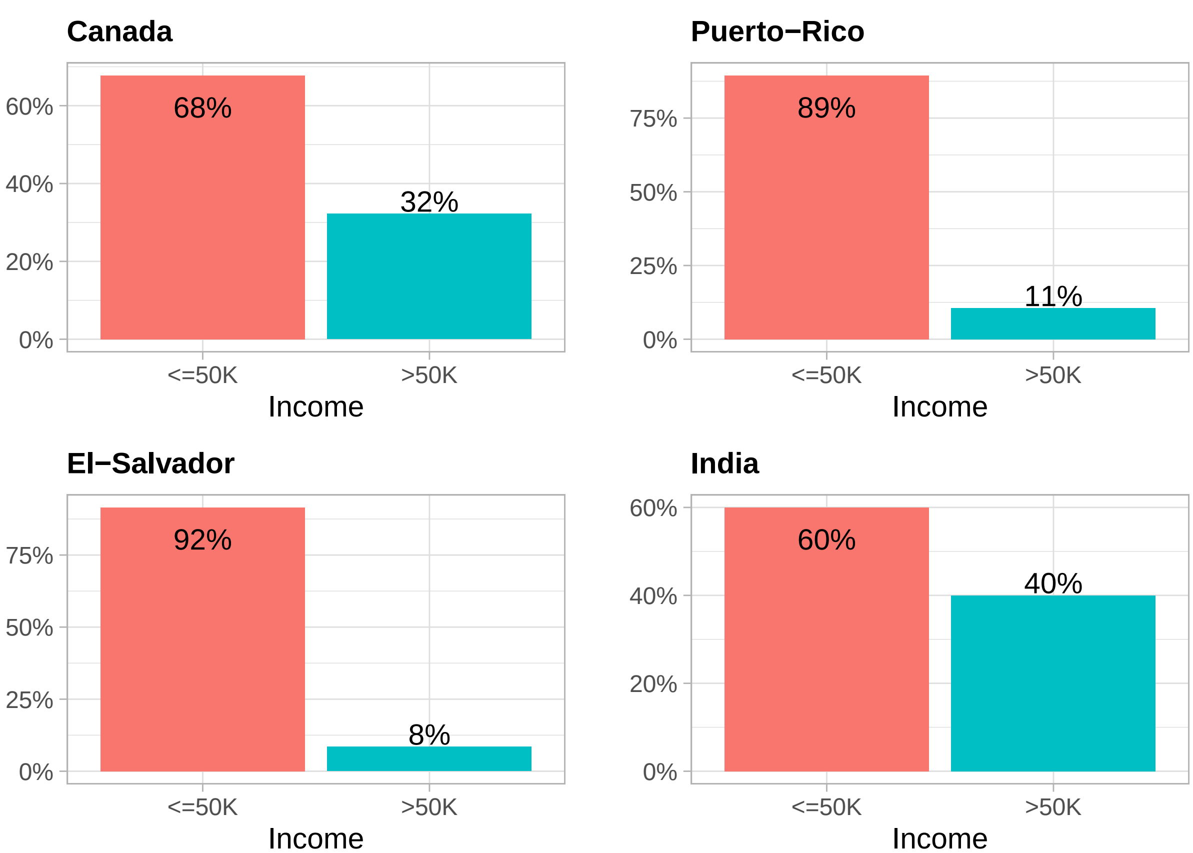
```
df$native.country <- factor(df$native.country,
                                 levels =
                                     names(sort(table(df$native.country),

ggplot(df,
       aes(x = df$native.country, fill = df$native.country)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = -.1) +
  labs(x = "Region",
       y = "",
       fill = "Regions") +
  theme(legend.position = 'none',
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = percent)
```

```
## Warning: Use of `df$native.country` is discouraged. Use `native.country` instead.
## Use of `df$native.country` is discouraged. Use `native.country` instead.
## Use of `df$native.country` is discouraged. Use `native.country` instead.
## Use of `df$native.country` is discouraged. Use `native.country` instead.
```




```
lp_region <- lapply(levels(df$native.country), function(v){

    df <- subset(df, df$native.country == v)

    ggplot(data = df,
           aes(x = income,
               fill = income)) +
      geom_bar(aes(y = (..count..)/sum(..count..))) +
      geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                    y = (..count..)/sum(..count..) ),
                stat = "count",
                vjust = c(2, -0.1),
                size = 4) +
      labs(x = "Income",
           y = "",
           fill = "Income") +
      ggtitle(v) +
      theme(legend.position = 'none',
            plot.title = element_text(size = 11, face = "bold")) +
      scale_y_continuous(labels = percent) })


grid.arrange(grobs = lp_region[1:4], ncol = 2)
```


```
grid.arrange(grobs = lp_region[5:8], ncol = 2)
```



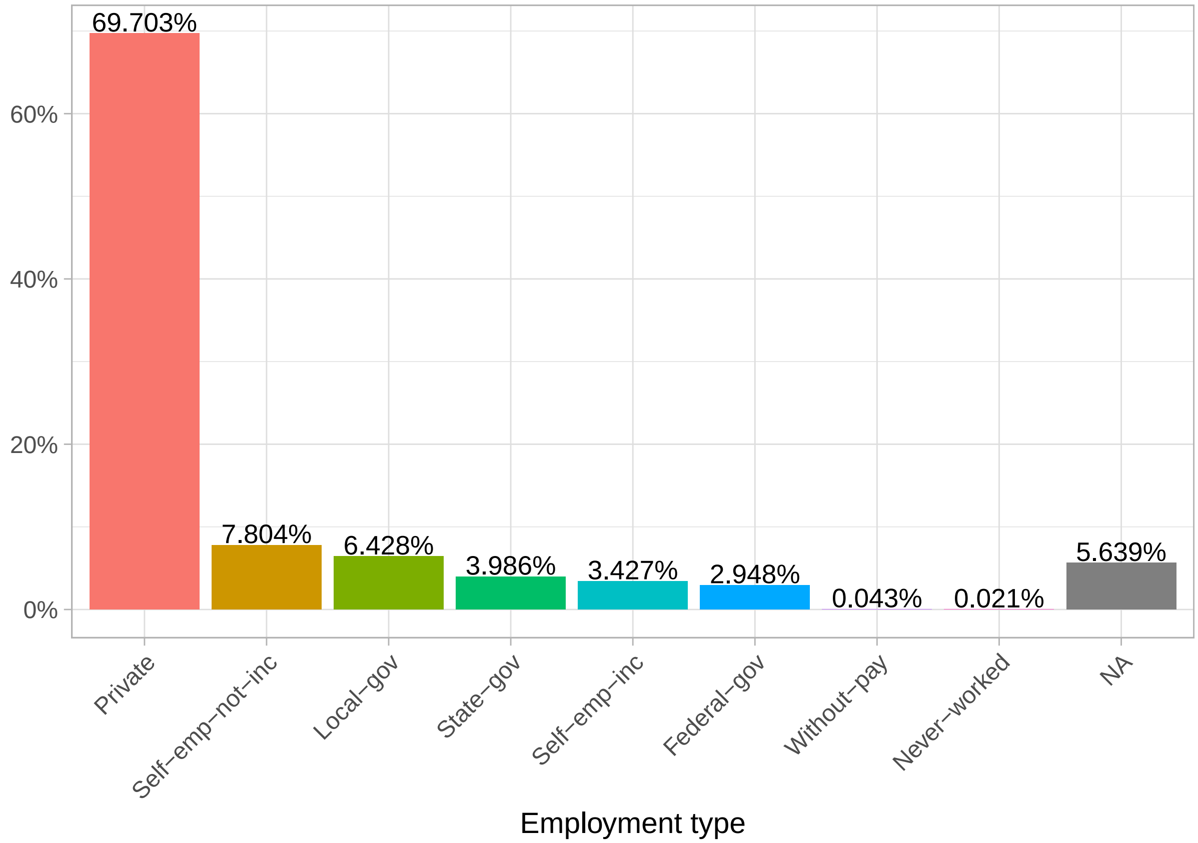

```
table(df$workclass)
```

```
##
```

```
##      Federal-gov        Local-gov     Never-worked          Private
##              960             2093                7            22696
##     Self-emp-inc Self-emp-not-inc        State-gov      Without-pay
##             1116             2541             1298               14
```

```
df$workclass <- factor(df$workclass,
                             levels =
                                     names(sort(table(df$workclass),

ggplot(df,
       aes(x = df$workclass, fill = df$workclass)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = -.1,
            size = 3.5) +
  labs(x = "Employment type",
       y = "",
       fill = "Employment type") +
  theme(legend.position = 'none',
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = percent)
```

```
## Warning: Use of `df$workclass` is discouraged. Use `workclass` instead.
## Use of `df$workclass` is discouraged. Use `workclass` instead.
## Use of `df$workclass` is discouraged. Use `workclass` instead.
## Use of `df$workclass` is discouraged. Use `workclass` instead.
```




```
nrow(subset(df , df$workclass == " Never-worked"))
```

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

```
nrow(subset(df , df$workclass == " Without-pay" &
                       df$income == " >50K"))
```

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

```
modified.work <- levels(df$workclass)

modified.work
```

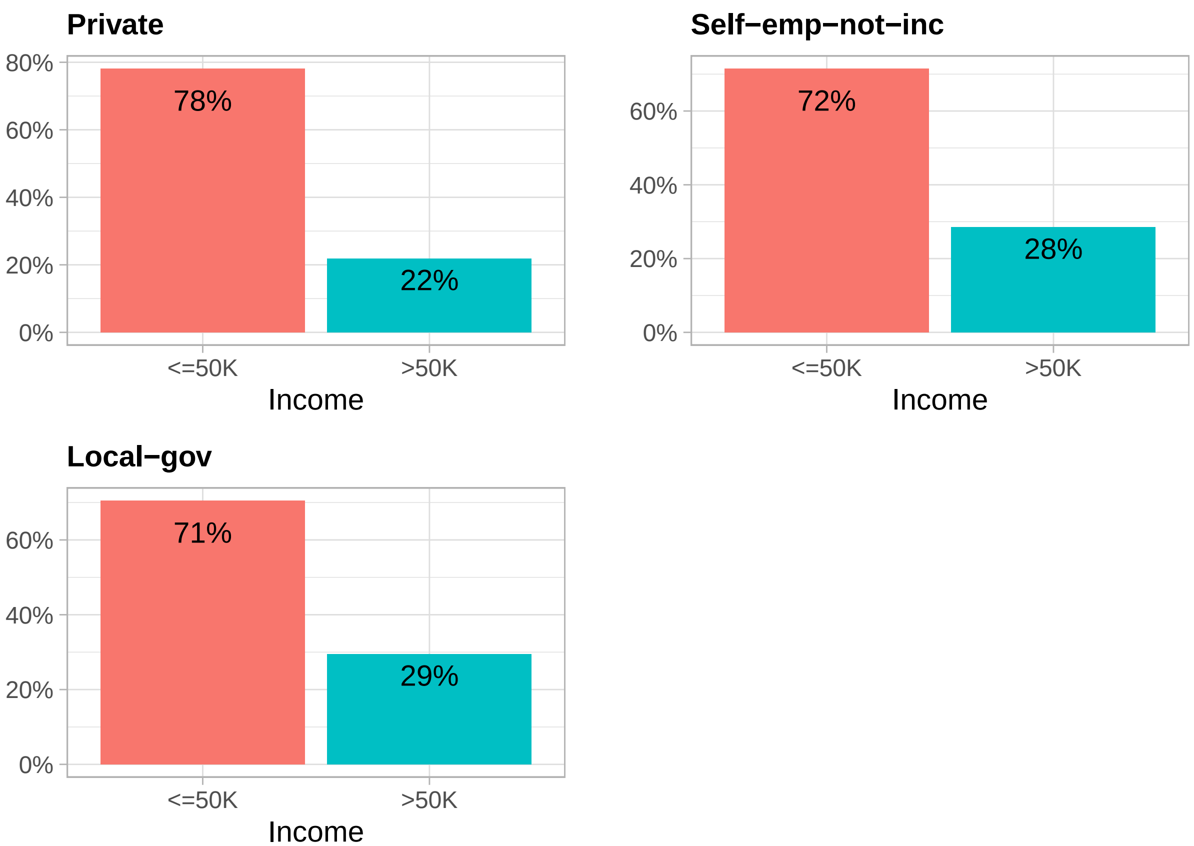
```
## [1] "Private"          "Self-emp-not-inc" "Local-gov"        "State-gov"
## [5] "Self-emp-inc"     "Federal-gov"      "Without-pay"      "Never-worked"
```

```
modified.work <- modified.work[!is.element(modified.work,
                                           c("Never-worked",
                                             "Without-pay"))]

modified.work
```

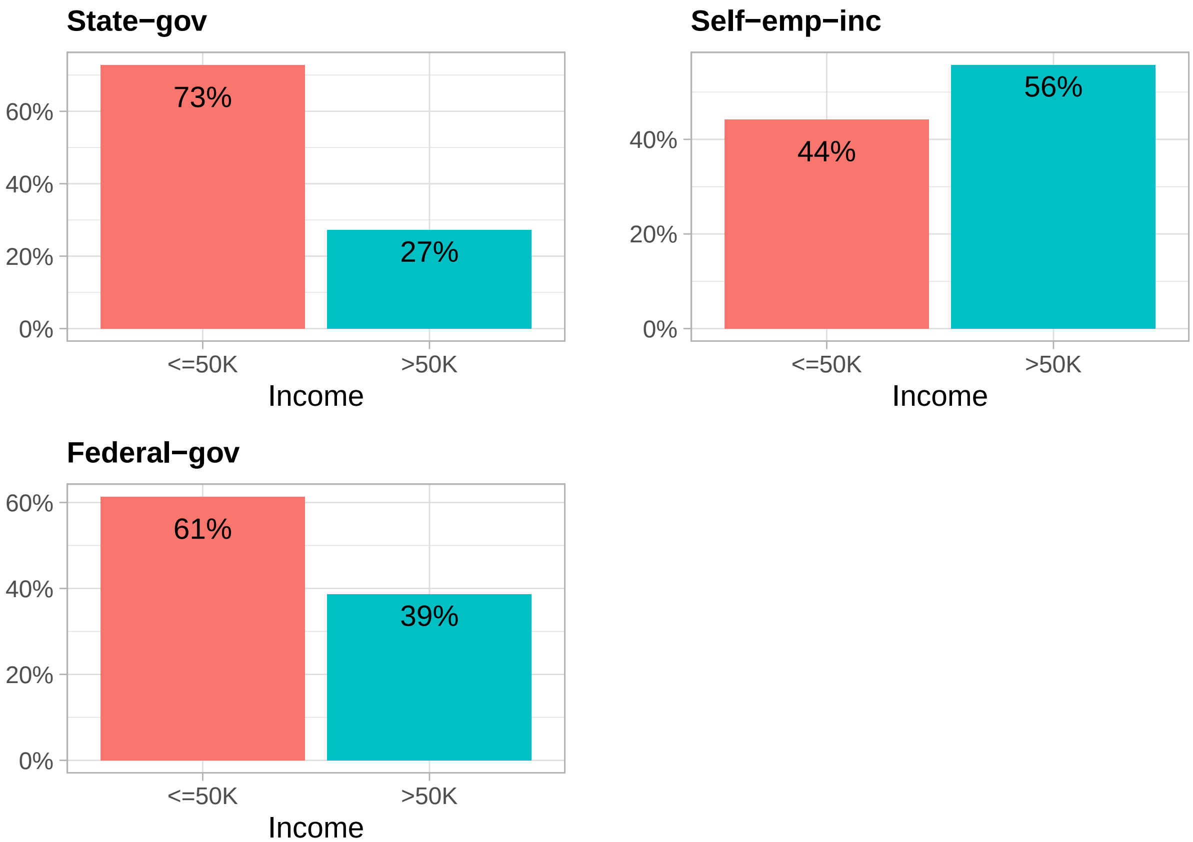
```
## [1] "Private"          "Self-emp-not-inc" "Local-gov"        "State-gov"
## [5] "Self-emp-inc"     "Federal-gov"
```

```
lg.workclass.mod <- lapply(modified.work, function(v){

  ggplot(data = subset(df, df$workclass == v),
         aes(x = subset(df, df$workclass == v)$income,
             fill = subset(df, df$workclass == v)$income)) +
```

```
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
              stat = "count",
              vjust = c(2, 1.5)) +
    labs(x = "Income",
         y = "",
         fill = "Income") +
    ggtitle(v) +
    theme(legend.position = 'none',
          plot.title = element_text(size = 11, face = "bold")) +
    scale_y_continuous(labels = percent) })

grid.arrange(grobs = lg.workclass.mod[1:3], ncol = 2)
```




```
grid.arrange(grobs = lg.workclass.mod[4:6], ncol = 2)
```



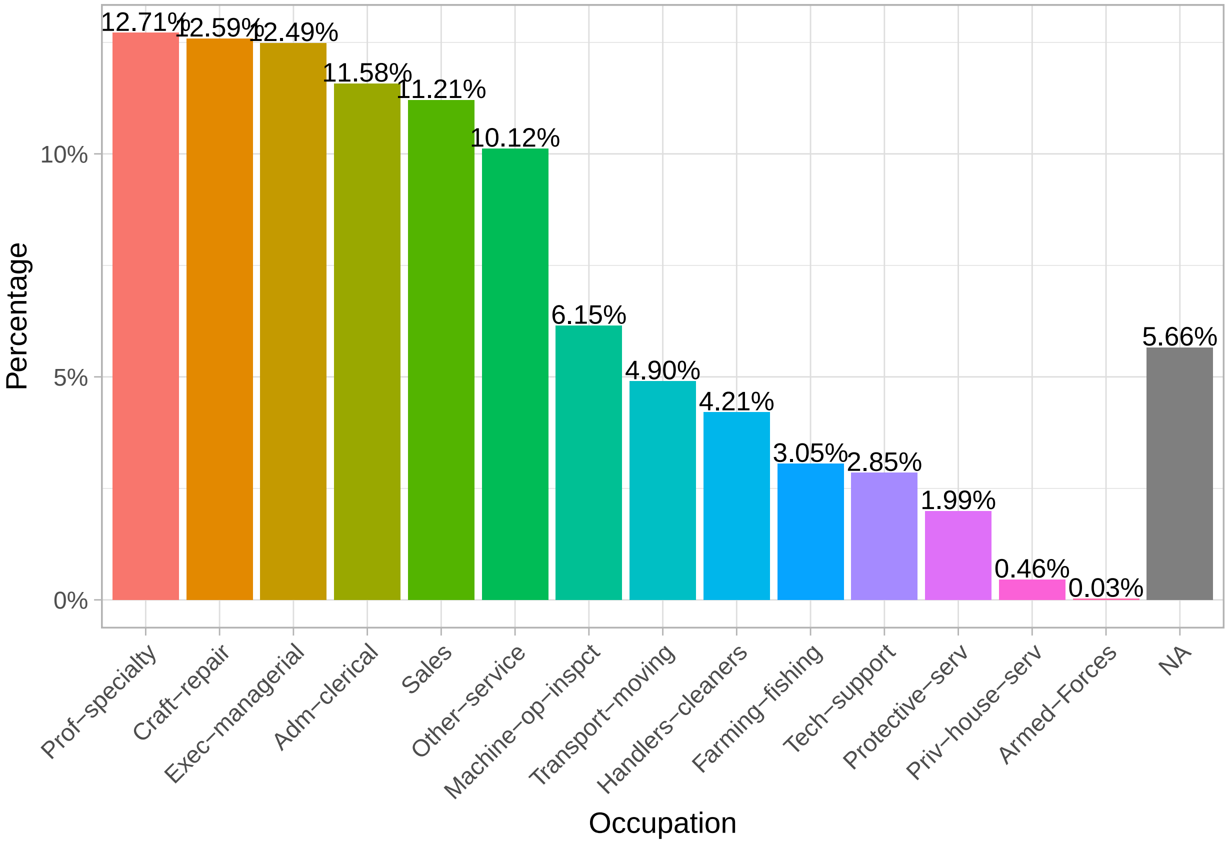

```
table(df$occupation)
```

```
##
##      Adm-clerical      Armed-Forces      Craft-repair   Exec-managerial
##              3770                 9              4099              4066
##   Farming-fishing Handlers-cleaners Machine-op-inspct     Other-service
##               994              1370              2002              3295
##   Priv-house-serv    Prof-specialty   Protective-serv             Sales
##               149              4140               649              3650
##      Tech-support  Transport-moving
##               928              1597
```

```
df$occupation <- factor(df$occupation,
                                 levels =
                                     names(sort(table(df$occupation),

ggplot(df,
       aes(x = df$occupation, fill = df$occupation)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = -.1,
            size = 3.5) +
  labs(x = "Occupation",
       y = "Percentage",
       fill = "Occupation") +
```

```r
  theme(legend.position = 'none',
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = percent)
```

```
## Warning: Use of `df$occupation` is discouraged. Use `occupation` instead.
## Use of `df$occupation` is discouraged. Use `occupation` instead.
## Use of `df$occupation` is discouraged. Use `occupation` instead.
## Use of `df$occupation` is discouraged. Use `occupation` instead.
```




```r
nrow(subset(df, df$sex == " Female" &
                      df$occupation == " Armed-Forces"))
```

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

```r
nrow(subset(df, df$sex == " Male" &
                      df$occupation == " Priv-house-serv" &
                      df$income == " >50K"))
```

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

```r
modified.occup.f <- levels(df$occupation)
modified.occup.f
```

```
##  [1] "Prof-specialty"    "Craft-repair"      "Exec-managerial"
##  [4] "Adm-clerical"      "Sales"             "Other-service"
##  [7] "Machine-op-inspct" "Transport-moving"  "Handlers-cleaners"
## [10] "Farming-fishing"   "Tech-support"      "Protective-serv"
## [13] "Priv-house-serv"   "Armed-Forces"
```

```r
modified.occup.f <- modified.occup.f[!is.element(modified.occup.f,
                                                 c("Armed-Forces"))]
modified.occup.f
```

```
##  [1] "Prof-specialty"    "Craft-repair"      "Exec-managerial"
##  [4] "Adm-clerical"      "Sales"             "Other-service"
##  [7] "Machine-op-inspct" "Transport-moving"  "Handlers-cleaners"
## [10] "Farming-fishing"   "Tech-support"      "Protective-serv"
## [13] "Priv-house-serv"
```

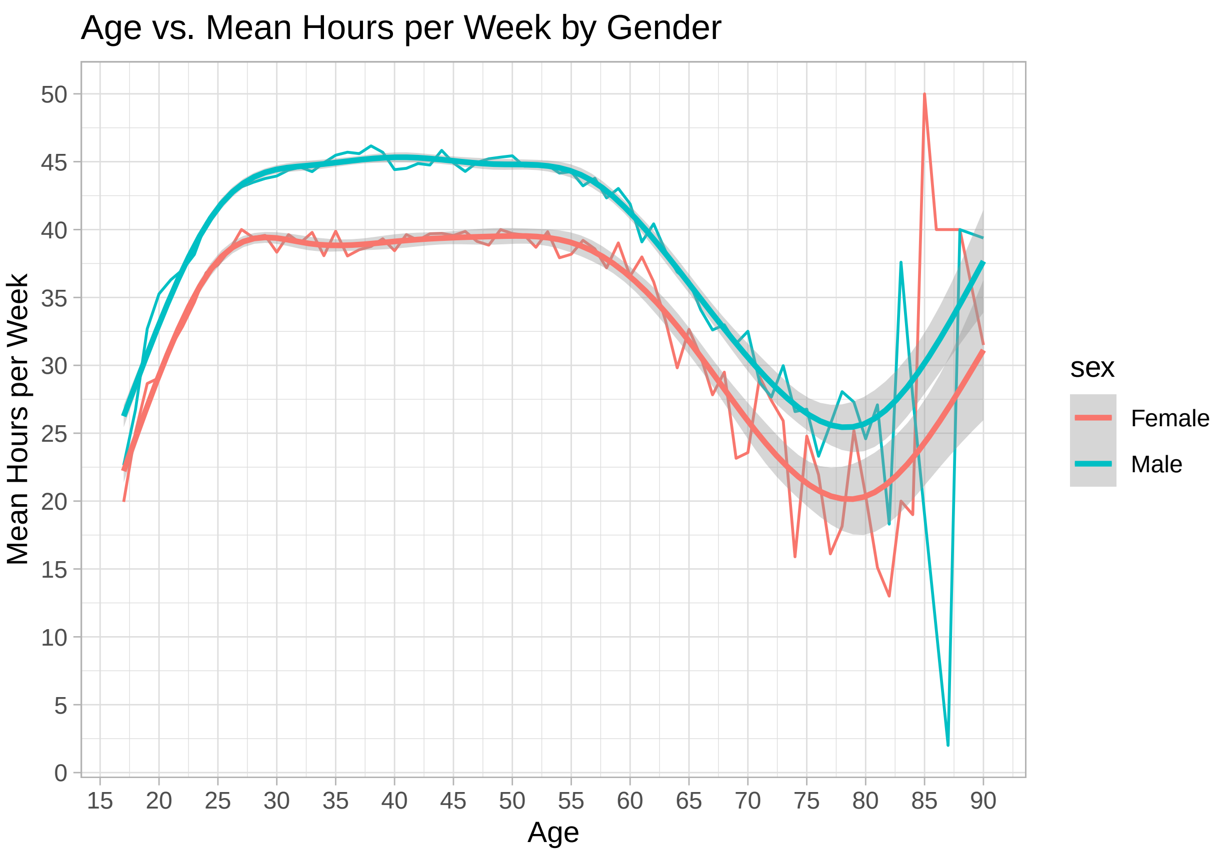
5 Update 4

```r
ggplot(aes(x = age, y = hours.per.week),
       data = df) +
  geom_line(mapping = aes(color = sex),
            stat = 'summary',
            fun.y = mean) +
  geom_smooth(mapping = aes(color = sex)) +
  scale_x_continuous(breaks = seq(10, 100, 5)) +
  scale_y_continuous(breaks = seq(0, 55, 5)) +
  labs(x = "Age", y = "Mean Hours per Week") +
  ggtitle("Age vs. Mean Hours per Week by Gender")
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
summary(df$hours.per.week)
```

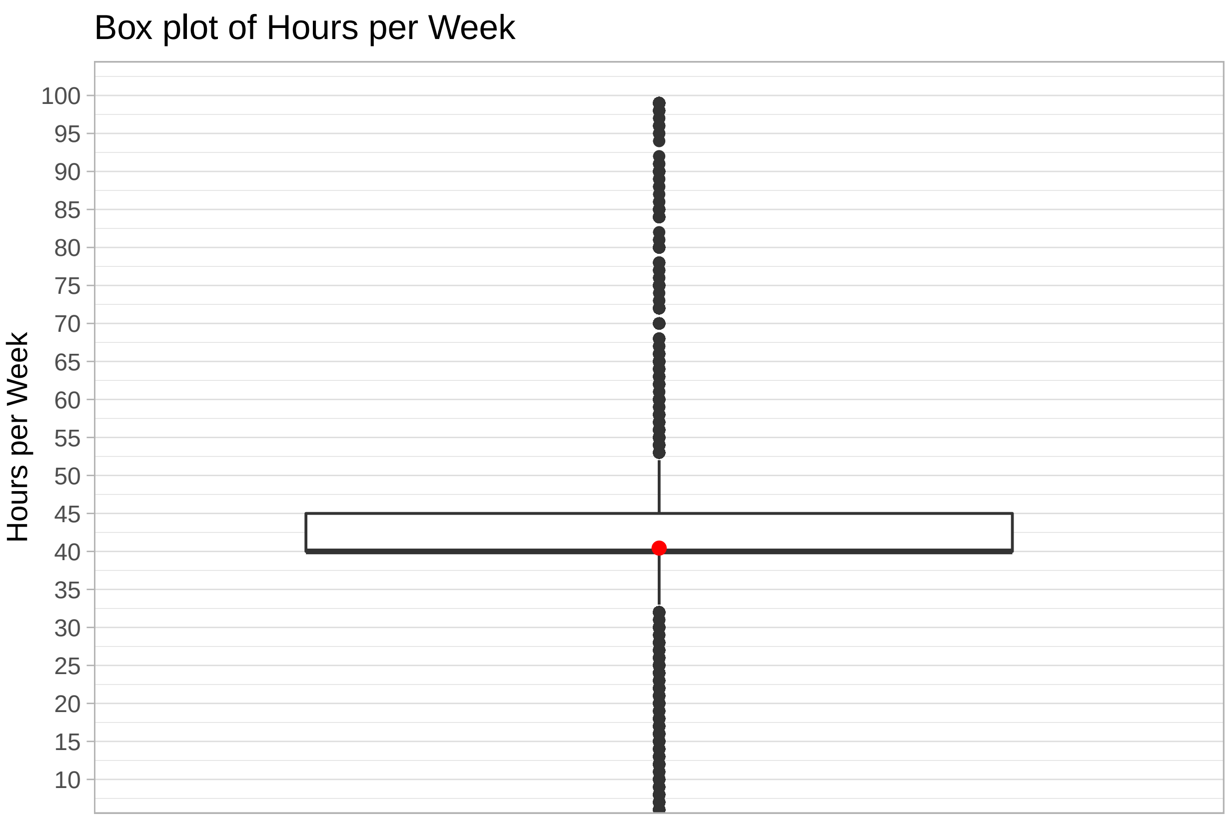
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   40.00   40.00   40.44   45.00   99.00
```

```
IQR(df$hours.per.week)
```

```
## [1] 5
```

```
ggplot(aes(x = factor(0), y = hours.per.week),
       data = df) +
  geom_boxplot() +
  stat_summary(fun.y = mean,
               geom = "point",
               shape = 19,
               color = "red",
               cex = 2) +
  coord_cartesian(ylim = c(10, 100)) +
  scale_x_discrete(breaks = NULL) +
  scale_y_continuous(breaks = seq(10, 100, 5)) +
  ylab("Hours per Week") +
  xlab("") +
  ggtitle("Box plot of Hours per Week")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```




```
ggplot(aes(x = income, y = hours.per.week),
       data = df) +
  geom_boxplot() +
  stat_summary(fun.y = mean,
               geom = 'point',
               shape = 19,
               color = "red",
               cex = 2) +
  coord_cartesian(ylim = c(10, 100))+
  scale_y_continuous(breaks = seq(10, 100, 10)) +
  ylab("Hours per Week") +
  xlab("Income") +
  ggtitle("Box plot of Hours per Week by Income")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

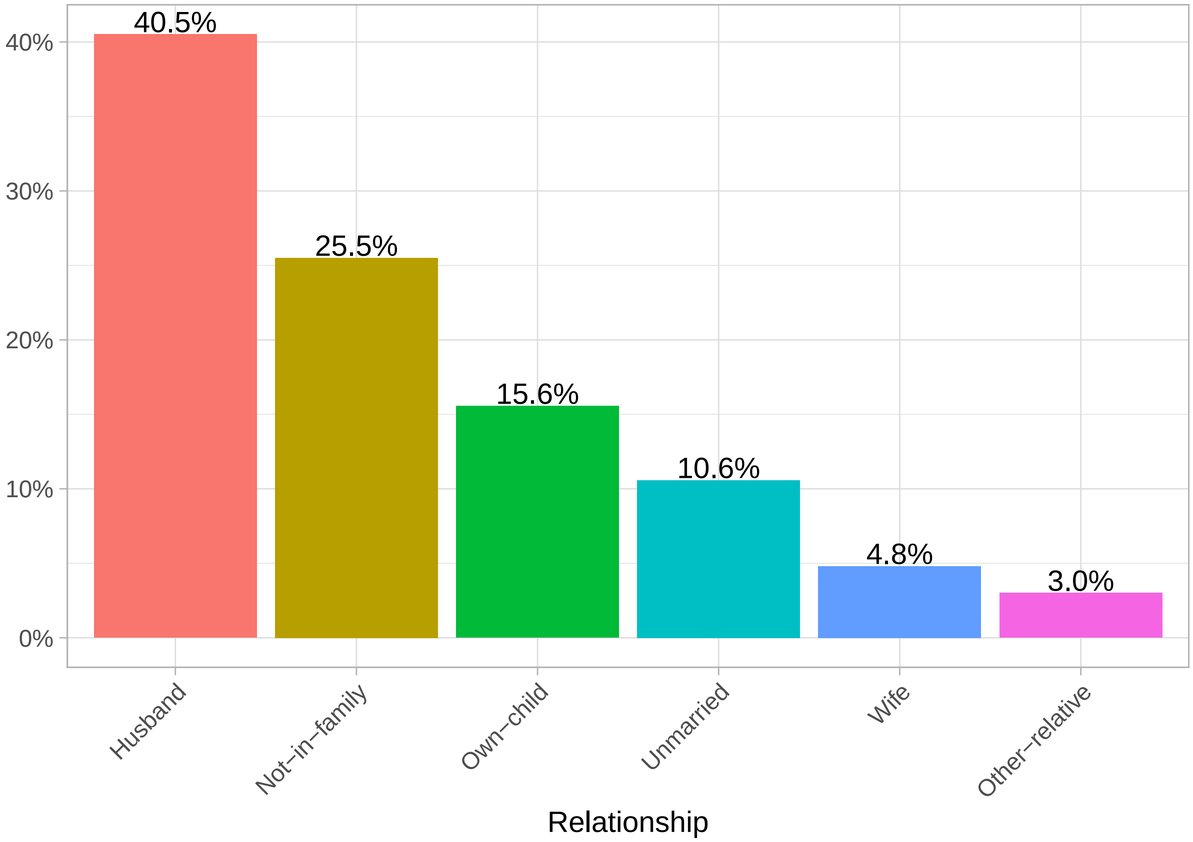

```
table(df$relationship)
```

```
##
##        Husband  Not-in-family Other-relative      Own-child      Unmarried
##          13193           8305            981           5068           3446
##           Wife
##           1568
```

```
df$relationship <- factor(df$relationship,
                                 levels =
                                     names(sort(table(df$relationship),

ggplot(df,
       aes(x = df$relationship, fill = df$relationship)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = -.1) +
  labs(x = "Relationship",
       y = "",
       fill = "Relationship") +
  theme(legend.position = 'none',
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = percent)
```

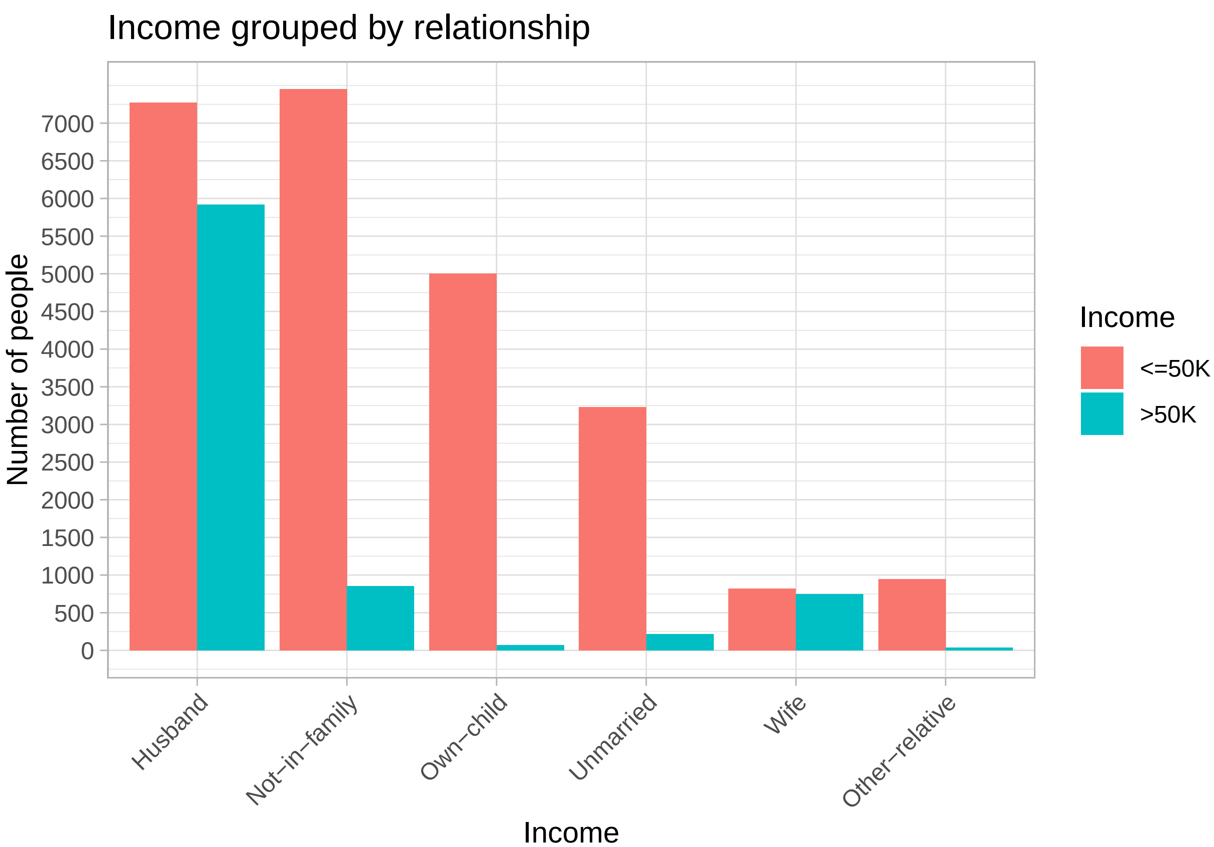
```
## Warning: Use of `df$relationship` is discouraged. Use `relationship` instead.
```

```
## Use of `df$relationship` is discouraged. Use `relationship` instead.
## Use of `df$relationship` is discouraged. Use `relationship` instead.
## Use of `df$relationship` is discouraged. Use `relationship` instead.
```

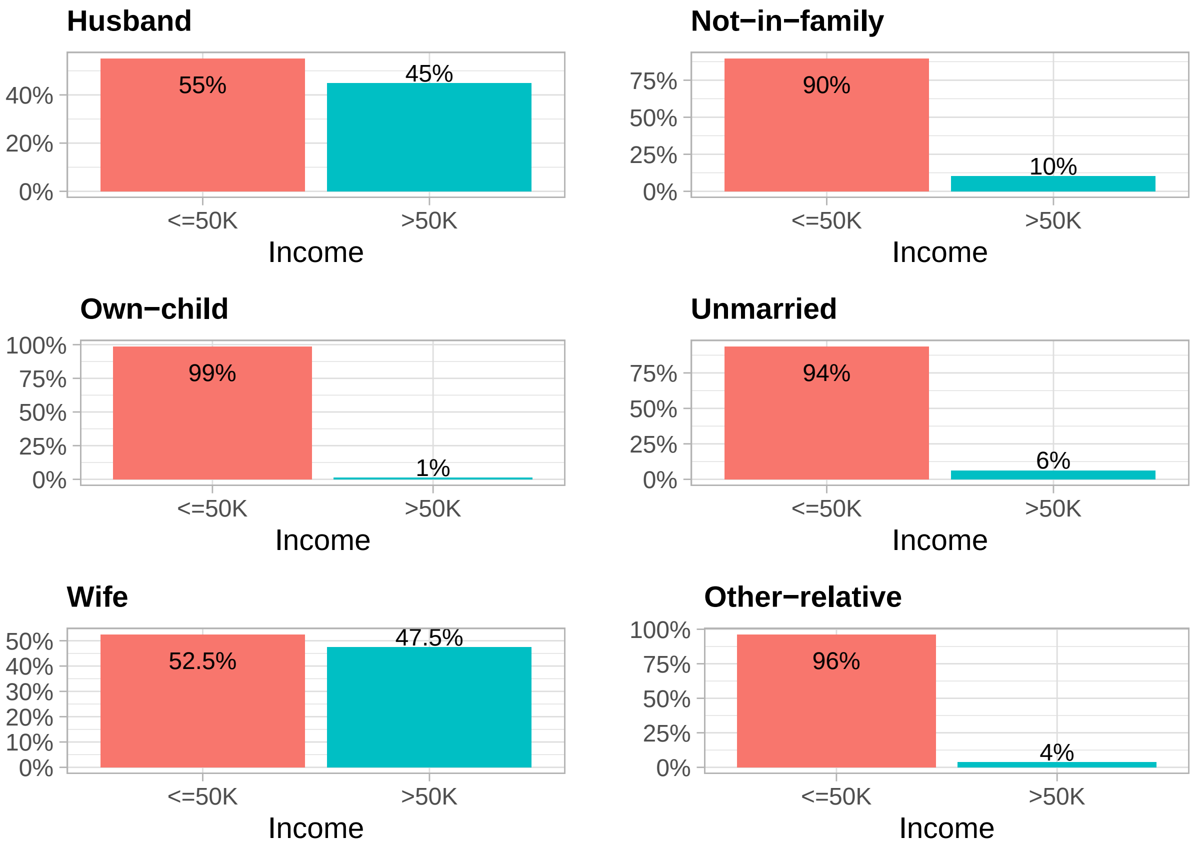



```
ggplot(df, aes(x=df$relationship, fill=df$income)) +
  geom_bar(position=position_dodge()) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Income",
       y = "Number of people",
       fill = "Income") +
  ggtitle("Income grouped by relationship") +
  scale_y_continuous(breaks = seq(0,7000,500))
```

```
## Warning: Use of `df$relationship` is discouraged. Use `relationship` instead.

## Warning: Use of `df$income` is discouraged. Use `income` instead.
```




```
lg.relationship <- lapply(levels(df$relationship), function(v){

ggplot(data = subset(df, df$relationship == v),
         aes(x = subset(df, df$relationship == v)$income,
             fill = subset(df, df$relationship == v)$income)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = c(2, -0.1),
            size = 3) +
  labs(x = "Income",
       y = "",
       fill = "Income") +
  ggtitle(paste(v)) +
  theme(legend.position = 'none',
        plot.title = element_text(size = 11, face = "bold")) +
  scale_y_continuous(labels = percent) })


grid.arrange(grobs = lg.relationship[1:6], ncol = 2)
```



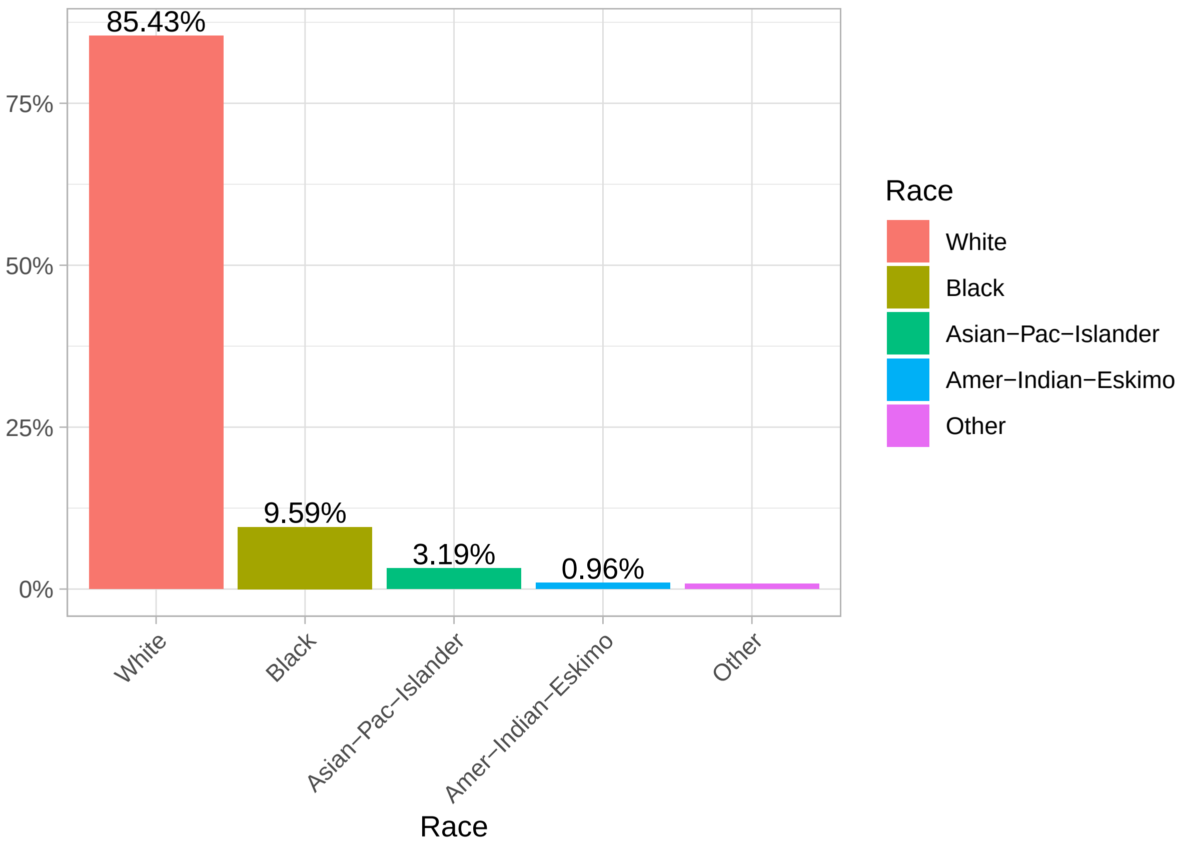

```
table(df$race)
```

```
##
## Amer-Indian-Eskimo Asian-Pac-Islander              Black              Other
##                311               1039               3124                271
##              White
##              27816
```

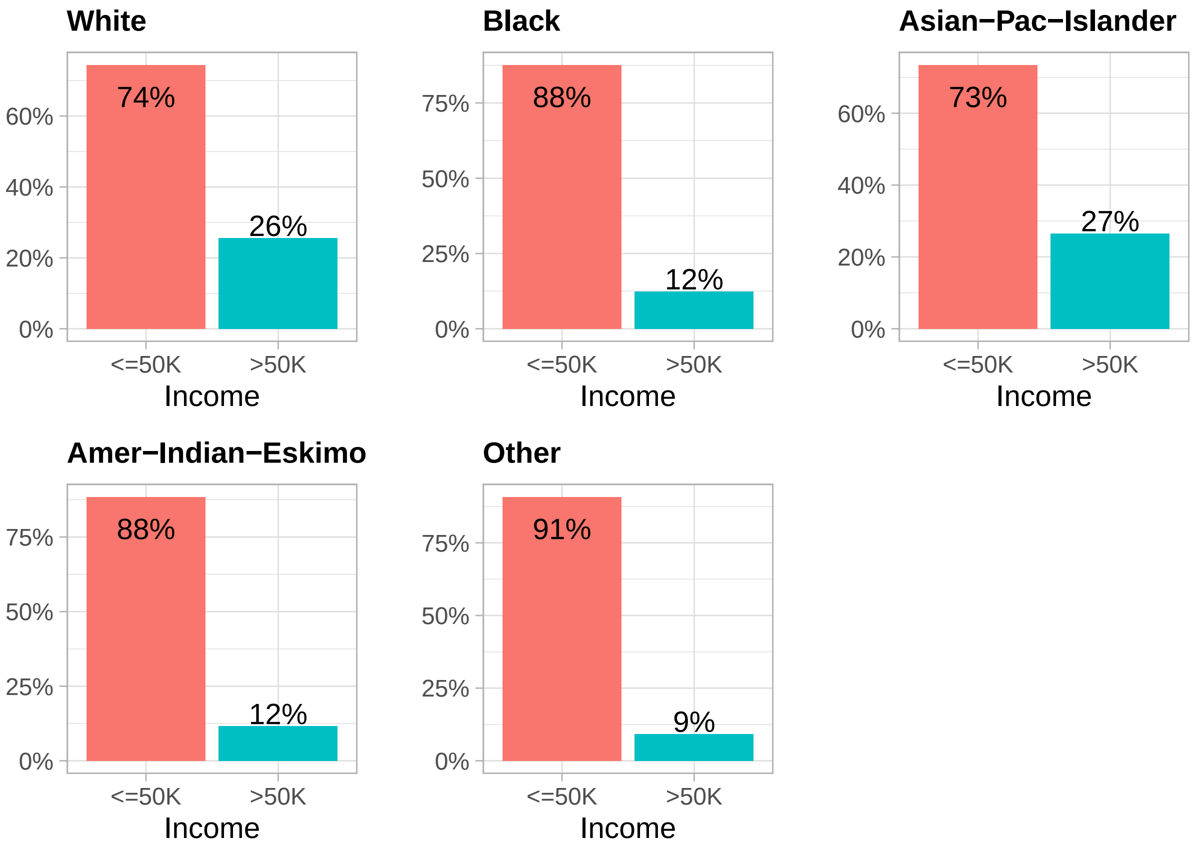
```
df$race <- factor(df$race,
                                   levels =
                                       names(sort(table(df$race),

ggplot(df,
       aes(x = df$race, fill = df$race)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = c(-0.2, -0.2, -0.2, -0.2, 3)) +
  labs(x = "Race",
       y = "",
       fill = "Race") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = percent)
```

```
## Warning: Use of `df$race` is discouraged. Use `race` instead.
## Use of `df$race` is discouraged. Use `race` instead.
```

```
## Use of `df$race` is discouraged. Use `race` instead.
## Use of `df$race` is discouraged. Use `race` instead.
```




```
lg.race <- lapply(levels(df$race), function(v){

  ggplot(data = subset(df, df$race == v),
         aes(x = subset(df, df$race == v)$income,
             fill = subset(df, df$race == v)$income)) +
    geom_bar(aes(y = (..count..)/sum(..count..))) +
    geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                  y = (..count..)/sum(..count..) ),
              stat = "count",
              vjust = c(2, -0.1)) +
    labs(x = "Income",
         y = "",
         fill = "Income") +
    ggtitle(paste(v)) +
    theme(legend.position = 'none',
          plot.title = element_text(size = 11, face = "bold")) +
    scale_y_continuous(labels = percent) })


grid.arrange(grobs = lg.race, ncol = 3)
```

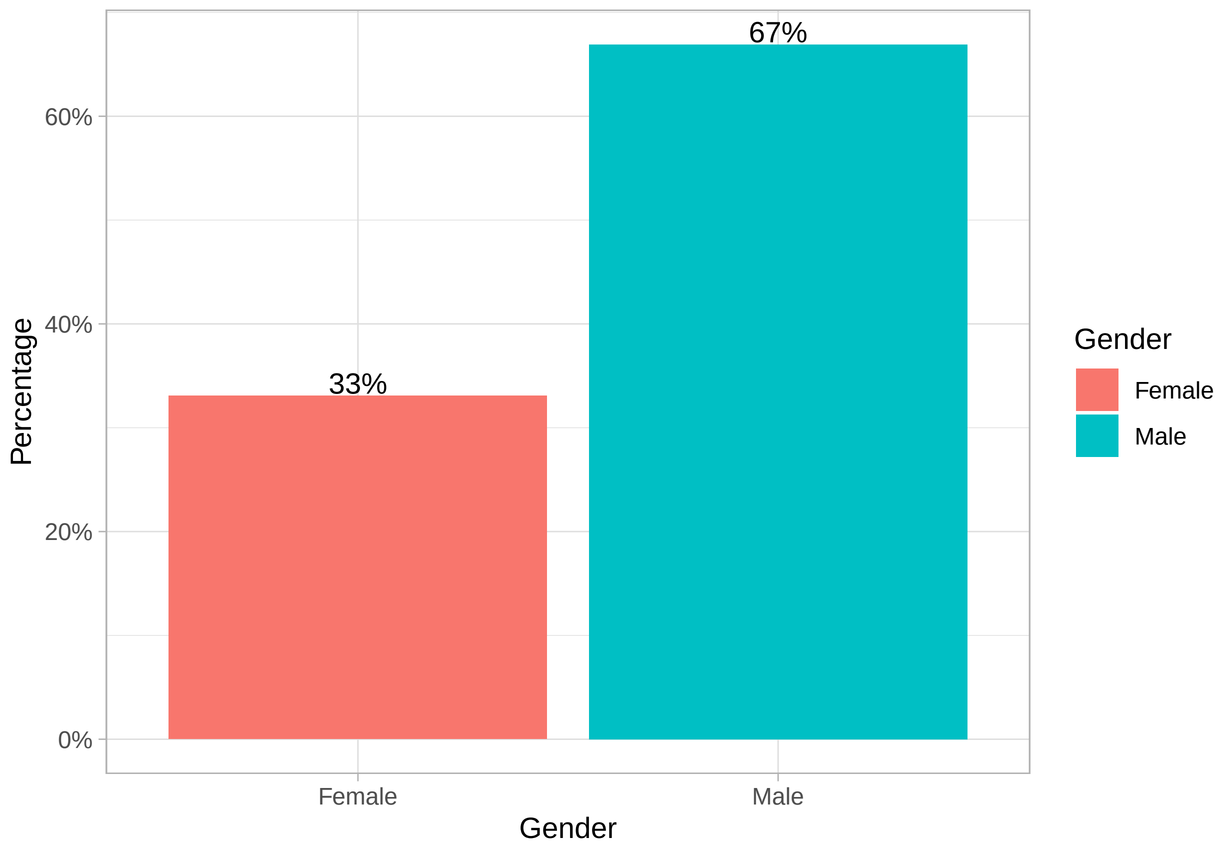



```
table(df$sex)
```

```
##
## Female   Male
##  10771  21790
```

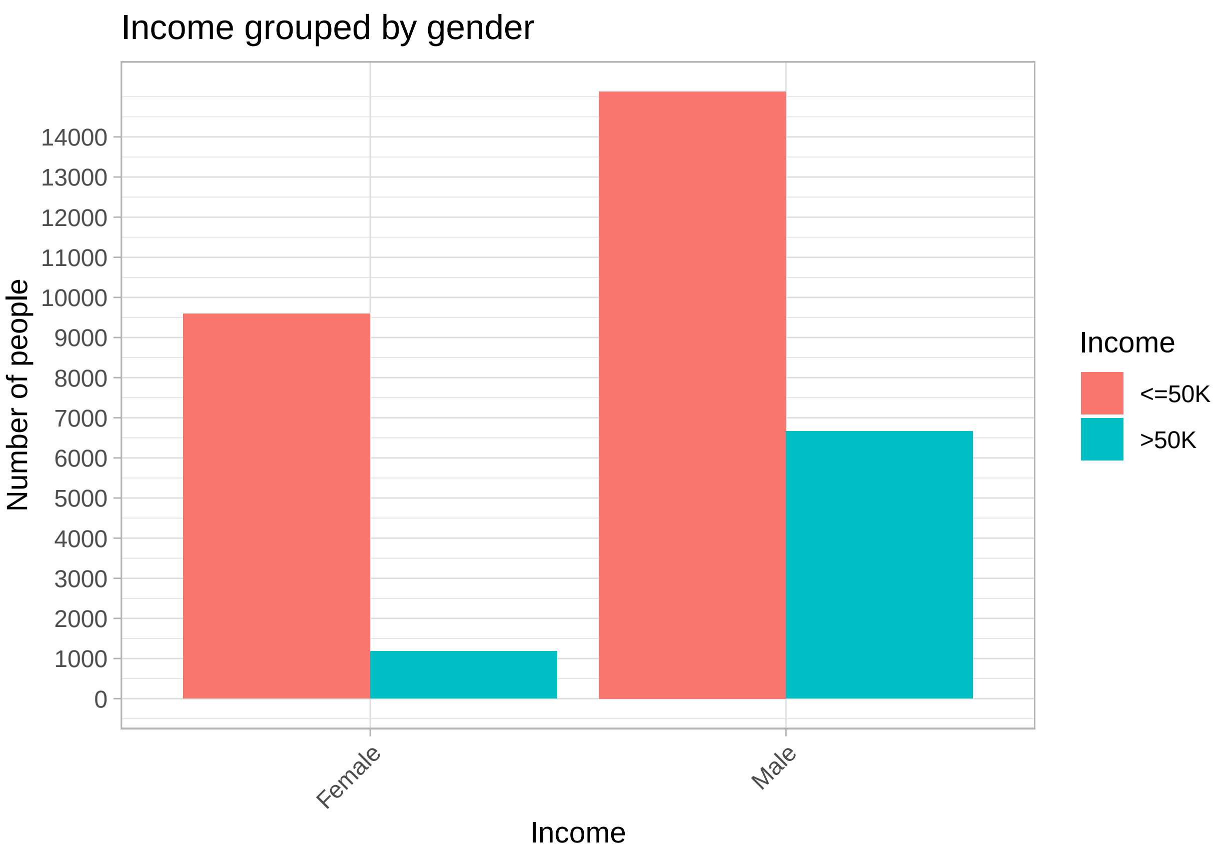
```
ggplot(df,
       aes(x = df$sex, fill = df$sex)) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(aes(label = scales::percent((..count..)/sum(..count..)),
                y = (..count..)/sum(..count..) ),
            stat = "count",
            vjust = -.1) +
  labs(x = "Gender",
       y = "Percentage",
       fill = "Gender") +
  scale_y_continuous(labels = percent)
```

```
## Warning: Use of `df$sex` is discouraged. Use `sex` instead.
## Use of `df$sex` is discouraged. Use `sex` instead.
## Use of `df$sex` is discouraged. Use `sex` instead.
## Use of `df$sex` is discouraged. Use `sex` instead.
```




```
ggplot(df, aes(x = df$sex, fill = df$income)) +
  geom_bar(position = position_dodge()) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Income",
       y = "Number of people",
       fill = "Income") +
  ggtitle("Income grouped by gender") +
  scale_y_continuous(breaks = seq(0,14500,1000))
```

```
## Warning: Use of `df$sex` is discouraged. Use `sex` instead.

## Warning: Use of `df$income` is discouraged. Use `income` instead.
```




```
table(df$sex, df$income)
```

```
##
##          <=50K  >50K
##   Female  9592  1179
##   Male   15128  6662
```

```
chisq.test(df$occupation, df$income)
```

```
## Warning in chisq.test(df$occupation, df$income): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  df$occupation and df$income
## X-squared = 3744.9, df = 13, p-value < 2.2e-16
```

```
chisq.test(df$occupation, df$income)$expected
```

```
## Warning in chisq.test(df$occupation, df$income): Chi-squared approximation may
## be incorrect
```

```
##                      df$income
## df$occupation               <=50K         >50K
##   Prof-specialty     3108.975845 1031.024155
##   Craft-repair       3078.186470 1020.813530
##   Exec-managerial    3053.404779 1012.595221
##   Adm-clerical       2831.120516  938.879484
```

```
##   Sales              2741.005274  908.994726
##   Other-service      2474.414350  820.585650
##   Machine-op-inspct 1503.422619  498.577381
##   Transport-moving  1199.283677  397.716323
##   Handlers-cleaners 1028.815678  341.184322
##   Farming-fishing    746.454587  247.545413
##   Tech-support       696.891204  231.108796
##   Protective-serv    487.373266  161.626734
##   Priv-house-serv    111.893092   37.106908
##   Armed-Forces         6.758643    2.241357
```

```
chisq.test(df$education, df$income)
```

```
##
##  Pearson's Chi-squared test
##
## data:  df$education and df$income
## X-squared = 4429.7, df = 15, p-value < 2.2e-16
```

6 Excuetive Summary

- Summarize the key (This could be a bulleted list)
  - information about your data set
  - major data cleaning
  - findings from EDA
  - Model output
  - Overall conclusions

7 Abstract

- Summary of the nature, finding and meaning of your data analysis project.
- 1 paragraph written summary of your data analysis project

8 Introduction

- Background and motivation of the Data Science question. The "Why'' of the research
- Explanation of your data
  - Where is your data from
  - What are the variables
- What data would be necessary to improve your analysis?

9 Data Science Methods

- To be applied (such as image processing, time-series analysis, spectral analysis etc
- Define critical capabilities and identify packages you will draw upon

10 Exploratory Data Analysis

10.1 Explanation of your data set

- How many variables?
- What are the data classes?
- How many levels of factors for factor variables?
- Is your data suitable for a project analysis?
- Write you databook, defining variables, units and structures

10.2 Data Cleaning

- What you had to do to clean your data

10.3 Data Vizualizations

- Vizualizations of your data

10.4 Variable Correlations

- Pairwise correlation plots, etc.

11 Statistical Learning: Modeling & Prediction

- DSCI 451 will accomplish at least 1 simple linear model (or simple logistic model)
- DSCI 352/352M/452 requires the appropriate modeling for your data set including machine learning
- Types of modeling to try
- Statistical prediction/modeling
- Model selection
- Cross-validation, Predictive R2
- Interpret results
- Challenge results

12 Discussion

- Discussion of the answers to the data science questions framed in the introduction

13 Conclusions

14 Acknowledgments

15 References

- Include a bib file in the markdown report

- Or hand written citations.