Adult Census Income

Harsh Navin Gupta

10 January 2020

Dataset Analysis

The **Adult Census Income Dataset**, contains features that are being used to predict whether a person, earns >50K or <=50K.
The dataset, contains an observation for every individual person, and the feature to be predicted is the *income*, which is composed of two values : *<=50K, >50K*.
The **Adult Census Income Dataset** contains the following features :
1. **Age** : Stores the age of the individual.
2. **Workclass** : Stores the type of employment of the individual, whether he/she is a federal emplyee, private employee, or has his/her own buisness.
3. **fnlwgt** : Stores the sampling weight.
4. **Education** : Stores the highest degree of education, held by the individual.
5. **Education-Num** : Stores the number of years of education completed by the individual.
6. **Marital-Status** : Stores the marital status of the individual, whether they are married, divorced,etc.
7. **Occupation** : Stores a short descriptor about the type of job of the individual.
8. **Relationship** : Stores the relationship which the individual holds, if he/she is a part of a family.
9. **Race** : Stores the race of the individual.
10. **Sex** : Stores the sex of an individual.
11. **Capital-Gain**
12. **Capital-Loss**
13. **Hours-Per-Week** : Stores the number of hours the individual works in a week.
14. **Native-Country** : Stores the country to which the individual natively belongs.

Dataset Download

The dataset consists of two files, that have to be downloaded.
1. **adult.data** : This is a CSV file, that contains the training data.
2. **adult.test** : This is a CSV file, that contains the testing data.

The link to download the **adult.data** file is :
https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

The link to download the **adult.test** file is :
https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test

First, we check for the library that are required, and load the required libraries.

```r
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
```

```
##   c.quosures     rlang
##   print.quosures rlang
```

```
## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2
```

```
## -- Attaching packages ------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.1     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
if(!require(ggthemes)) install.packages("ggthemes", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: ggthemes
```

Now, we create a vector of column names, for the data frames to be used.

```r
col_names <- c("age","workclass","fnlwgt","education",
               "education_num","marriage","occupation",
               "relationship","race","sex","capital-gain",
               "capital-loss","hours",
               "country","income")
```

Now, we first download the file **adult.data** and create our *training set*.

```r
#Link To Download adult.data
train_link <- "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"
train_set <- read_csv(train_link,col_names = FALSE)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   X2 = col_character(),
##   X3 = col_double(),
##   X4 = col_character(),
```

```
##   X5 = col_double(),
##   X6 = col_character(),
##   X7 = col_character(),
##   X8 = col_character(),
##   X9 = col_character(),
##   X10 = col_character(),
##   X11 = col_double(),
##   X12 = col_double(),
##   X13 = col_double(),
##   X14 = col_character(),
##   X15 = col_character()
## )
```

```
train_set <- setNames(train_set,col_names)
head(train_set)
```

```
## # A tibble: 6 x 15
##     age workclass fnlwgt education education_num marriage occupation
##   <dbl> <chr>      <dbl> <chr>             <dbl> <chr>    <chr>
## 1    39 State-gov  77516 Bachelors            13 Never-m~ Adm-cleri~
## 2    50 Self-emp~  83311 Bachelors            13 Married~ Exec-mana~
## 3    38 Private   215646 HS-grad               9 Divorced Handlers-~
## 4    53 Private   234721 11th                  7 Married~ Handlers-~
## 5    28 Private   338409 Bachelors            13 Married~ Prof-spec~
## 6    37 Private   284582 Masters              14 Married~ Exec-mana~
## # ... with 8 more variables: relationship <chr>, race <chr>, sex <chr>,
## #   `capital-gain` <dbl>, `capital-loss` <dbl>, hours <dbl>,
## #   country <chr>, income <chr>
```

Now, we proceed to download the file **adult.test** and create our *testing set*.

```
#Link To Download adult.test
test_link <- "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test"
test_set <- read_csv(test_link,col_names = FALSE,skip = 1)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   X2 = col_character(),
##   X3 = col_double(),
##   X4 = col_character(),
##   X5 = col_double(),
##   X6 = col_character(),
##   X7 = col_character(),
##   X8 = col_character(),
##   X9 = col_character(),
##   X10 = col_character(),
##   X11 = col_double(),
##   X12 = col_double(),
##   X13 = col_double(),
```

```
##   X14 = col_character(),
##   X15 = col_character()
## )
```

```r
test_set <- setNames(test_set,col_names)
head(test_set)
```

```
## # A tibble: 6 x 15
##     age workclass fnlwgt education education_num marriage occupation
##   <dbl> <chr>      <dbl> <chr>             <dbl> <chr>    <chr>
## 1    25 Private   226802 11th                  7 Never-m~ Machine-o~
## 2    38 Private    89814 HS-grad               9 Married~ Farming-f~
## 3    28 Local-gov 336951 Assoc-ac~            12 Married~ Protectiv~
## 4    44 Private   160323 Some-col~            10 Married~ Machine-o~
## 5    18 ?         103497 Some-col~            10 Never-m~ ?
## 6    34 Private   198693 10th                  6 Never-m~ Other-ser~
## # ... with 8 more variables: relationship <chr>, race <chr>, sex <chr>,
## #   `capital-gain` <dbl>, `capital-loss` <dbl>, hours <dbl>,
## #   country <chr>, income <chr>
```

Data Preprocessing

The columns of the dataset, *workclass, occupation, and native-country*, contain unknown values.

First, we discard all observations with unknown values *(represented by ?)*, from the training set.

```r
train_set <- train_set %>%
  filter(workclass != "?") %>%
  filter(occupation != "?") %>%
  filter(country != "?")
```

Next, we discard all observations with unknown values *(represented by ?)*, from the testing set.

```r
test_set <- test_set %>%
  filter(workclass != "?") %>%
  filter(occupation != "?") %>%
  filter(country != "?")
```

First, we calculate the *Average Weekly Working Hours* of all the individuals.

```r
mean_hours <- mean(train_set$hours)
```

Now create a new column named *work_hour_group*, which can have anyone of two values :
1. **Lower** : If the works hours per week of the individual are lower than the average weekly work hours.
2. **Higher** : If the works hours per week of the individual are higher than the average weekly work hours.

We initially make the changes to *Training Set*.

```r
train_set <- train_set %>%
  mutate(work_hour_group = ifelse(hours < mean_hours,"Lower","Higher"))
```

Then, we make the changes to the *Testing Set*.

```
test_set <- test_set %>%
  mutate(work_hour_group = ifelse(hours < mean_hours,"Lower","Higher"))
```

Next, we categorise the countries in the dataset, as *Developed Countries (D)* or *Under Development Countries*.
The following countries are categorised as *Developed Countries* :
* Germany
* England
* France
* Japan
* Canada
* United-States
* Ireland
* Italy
The remaining countries are classified as *Under Development Countries (UD)*.

```
#Categorising Countries As Developed & Under Development
dc <- c("Germany","England","France","Japan",
        "Canada","United-States","Ireland","Italy")
```

Initially we make changes to the *Training Set*.

```
train_set <- train_set %>%
  mutate(country_type = ifelse(country %in% dc,"D","UD"))
```

Then, we make changes to the *Testing Set*.

```
test_set <- test_set %>%
  mutate(country_type = ifelse(country %in% dc,"D","UD"))
```

For the purpose of ease of training ML Models on the training set, we add a new column y to both training and testing sets, which contains the following values :
1. **0** : If the individual income is equal to <=50K
2. **1** : If the individual income is equal to >50K

Initially, we make changes to the *Training Set*.

```
#Creating New Column y
train_set <- train_set %>% mutate(y = ifelse(income == "<=50K",0,1))
train_set <- train_set %>% mutate(y = factor(y))
```

Then we make changes to the *Testing Set*.

```
#Creating New Column y
test_set <- test_set %>% mutate(y = ifelse(income == "<=50K.",0,1))
test_set <- test_set %>% mutate(y = factor(y))
```

Exploratory Data Analysis

```r
#Total People In The Training Set
nrow(train_set)
```

```
## [1] 30162
```

```r
#Total Countries In The Training Set
n_distinct(train_set$country)
```

```
## [1] 41
```

```r
#Total Male & Females In The Training Set
train_set %>% group_by(sex) %>%
  summarise(count = n()) %>% knitr::kable()
```

sex	count
Female	9782
Male	20380

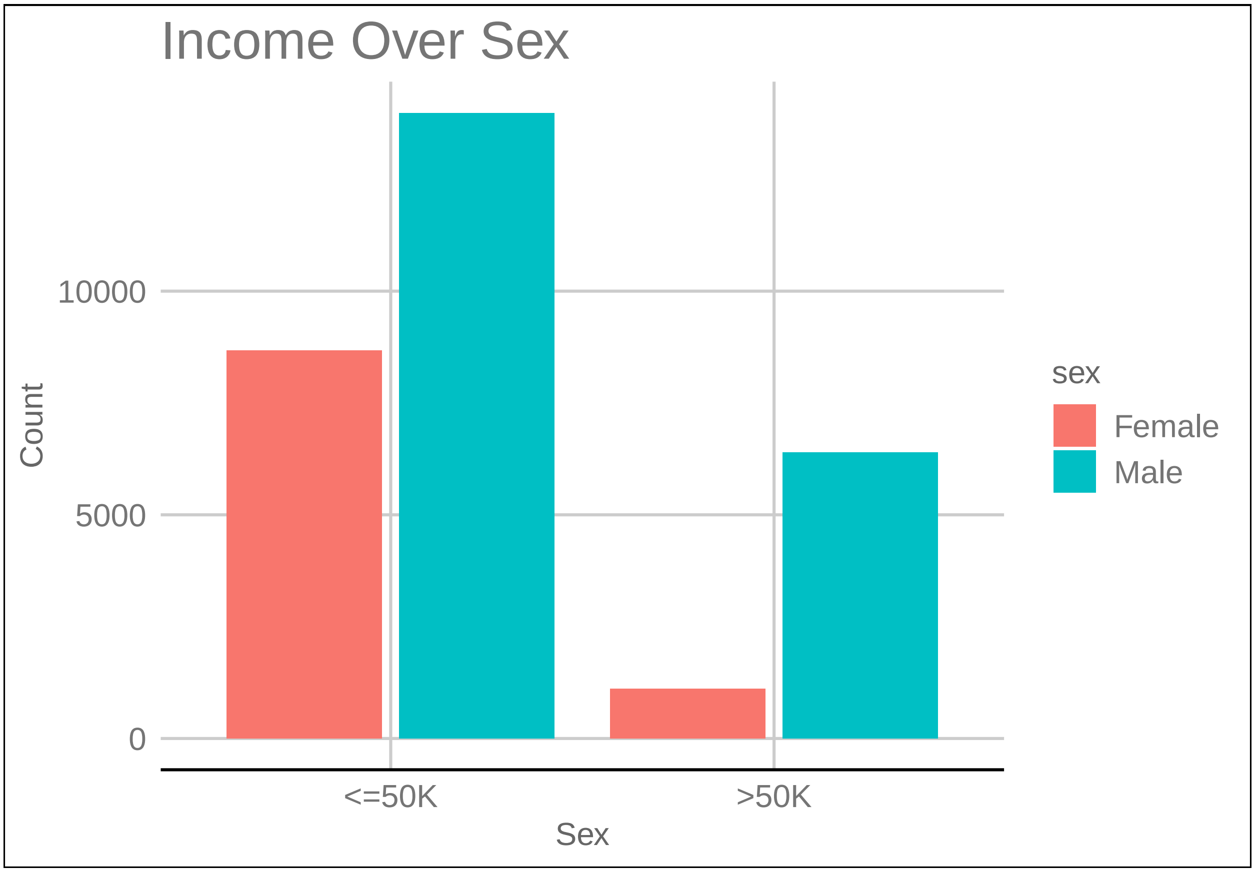
Income Over Sex

Here, we visualise the distribution of the income, over Males & Females, which are present in the training set.

```r
train_set %>% group_by(sex,income) %>%
  summarise(count = n()) %>% knitr::kable()
```

sex	income	count
Female	<=50K	8670
Female	>50K	1112
Male	<=50K	13984
Male	>50K	6396

```r
train_set %>% group_by(sex,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(income,count,fill = sex)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  xlab("Sex") + ylab("Count") +
  theme_gdocs() + ggtitle("Income Over Sex")
```



Here, it is observed that, the proportion of people earning <=50K is greater than people earning >50K for both the genders.
It can also be clearly observed that total number of Males are greater than the total number of Females.

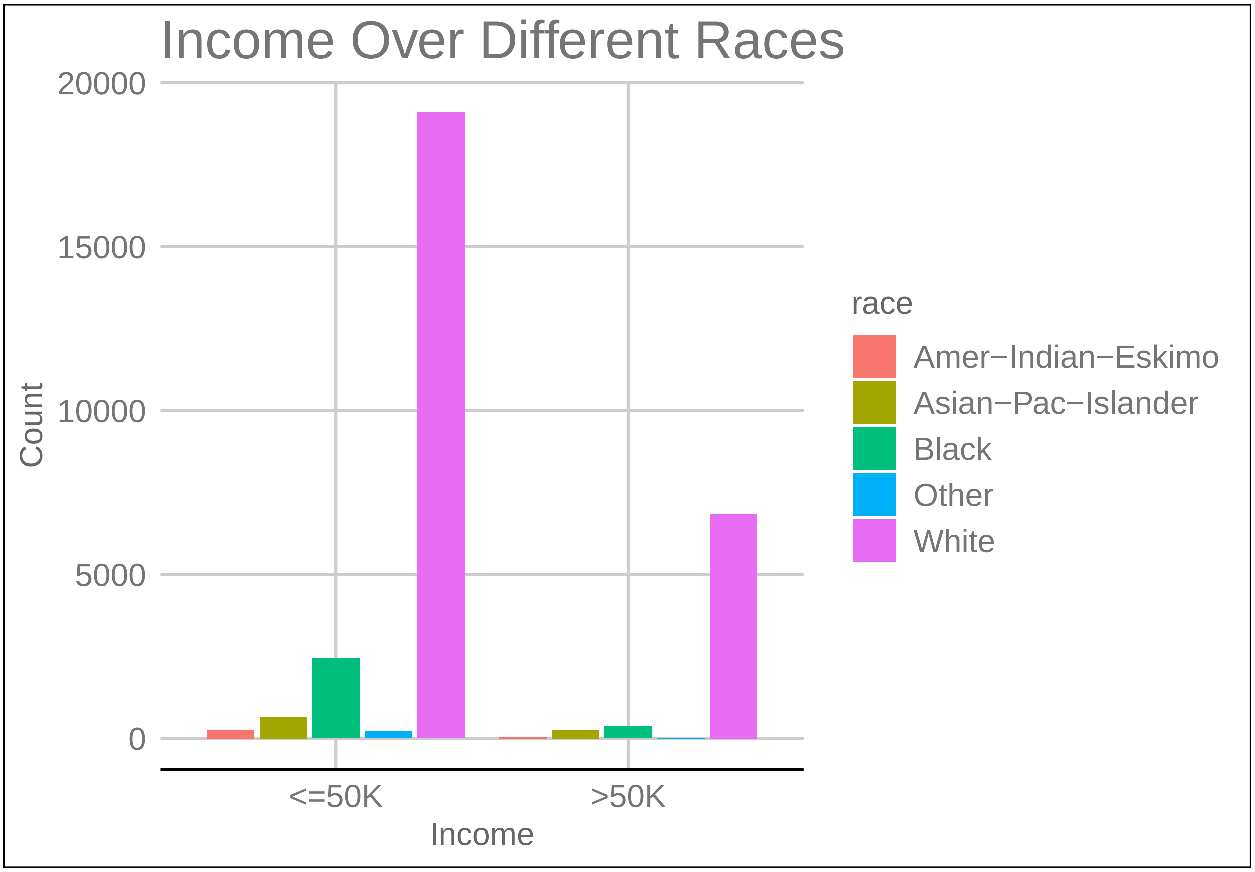
Income Over Different Races

Here, we analyse the distribution of the income, among individuals belonging to different races.

```
train_set %>% group_by(race,income) %>%
  summarise(count = n()) %>% knitr::kable()
```

race	income	count
Amer-Indian-Eskimo	<=50K	252
Amer-Indian-Eskimo	>50K	34
Asian-Pac-Islander	<=50K	647
Asian-Pac-Islander	>50K	248
Black	<=50K	2451
Black	>50K	366
Other	<=50K	210
Other	>50K	21
White	<=50K	19094
White	>50K	6839

```
train_set %>% group_by(race,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(income,count,fill = race)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  theme_gdocs() +
  xlab("Income") + ylab("Count") +
  ggtitle("Income Over Different Races")
```



Here, it can clearly observed that the dataset is dominated by people belonging to the *White* race. But, it can be observed in general, that the people with income <=50K are greater in number than people with income >50K, across all the races.

Income Over Education

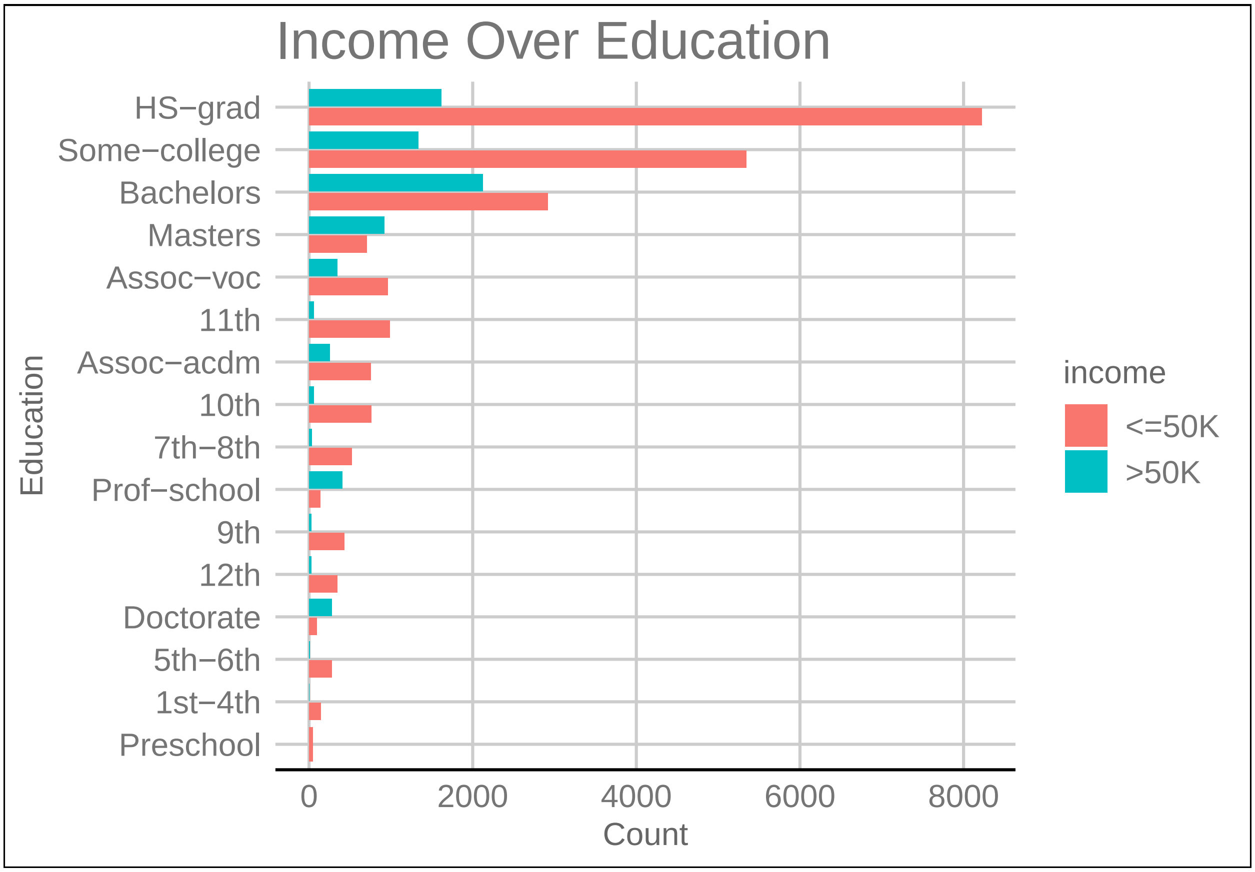
Here we analyze the distribution of the income, in comparison to the highest degree of education held by an individual.

```
train_set %>% group_by(education,income) %>%
  summarise(count = n()) %>% knitr::kable()
```

education	income	count
10th	<=50K	761
10th	>50K	59
11th	<=50K	989

education	income	count
11th	>50K	59
12th	<=50K	348
12th	>50K	29
1st-4th	<=50K	145
1st-4th	>50K	6
5th-6th	<=50K	276
5th-6th	>50K	12
7th-8th	<=50K	522
7th-8th	>50K	35
9th	<=50K	430
9th	>50K	25
Assoc-acdm	<=50K	752
Assoc-acdm	>50K	256
Assoc-voc	<=50K	963
Assoc-voc	>50K	344
Bachelors	<=50K	2918
Bachelors	>50K	2126
Doctorate	<=50K	95
Doctorate	>50K	280
HS-grad	<=50K	8223
HS-grad	>50K	1617
Masters	<=50K	709
Masters	>50K	918
Preschool	<=50K	45
Prof-school	<=50K	136
Prof-school	>50K	406
Some-college	<=50K	5342
Some-college	>50K	1336

```
train_set %>% group_by(education,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(reorder(education,count),count,fill = income)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  coord_flip() +
  theme_gdocs() +
  xlab("Education") + ylab("Count") +
  ggtitle("Income Over Education")
```



Here it can be clearly observed, that the people who hold Master's degree, or a Doctrate, or have attended Prof-School, have higher proportion of them, earning >50K, in comparison to other degrees, where the proportion of people earning, <=50K is greater.

It can also be observed that most people, in the dataset, are *High School Graduates**.

Income Over Different Occupations

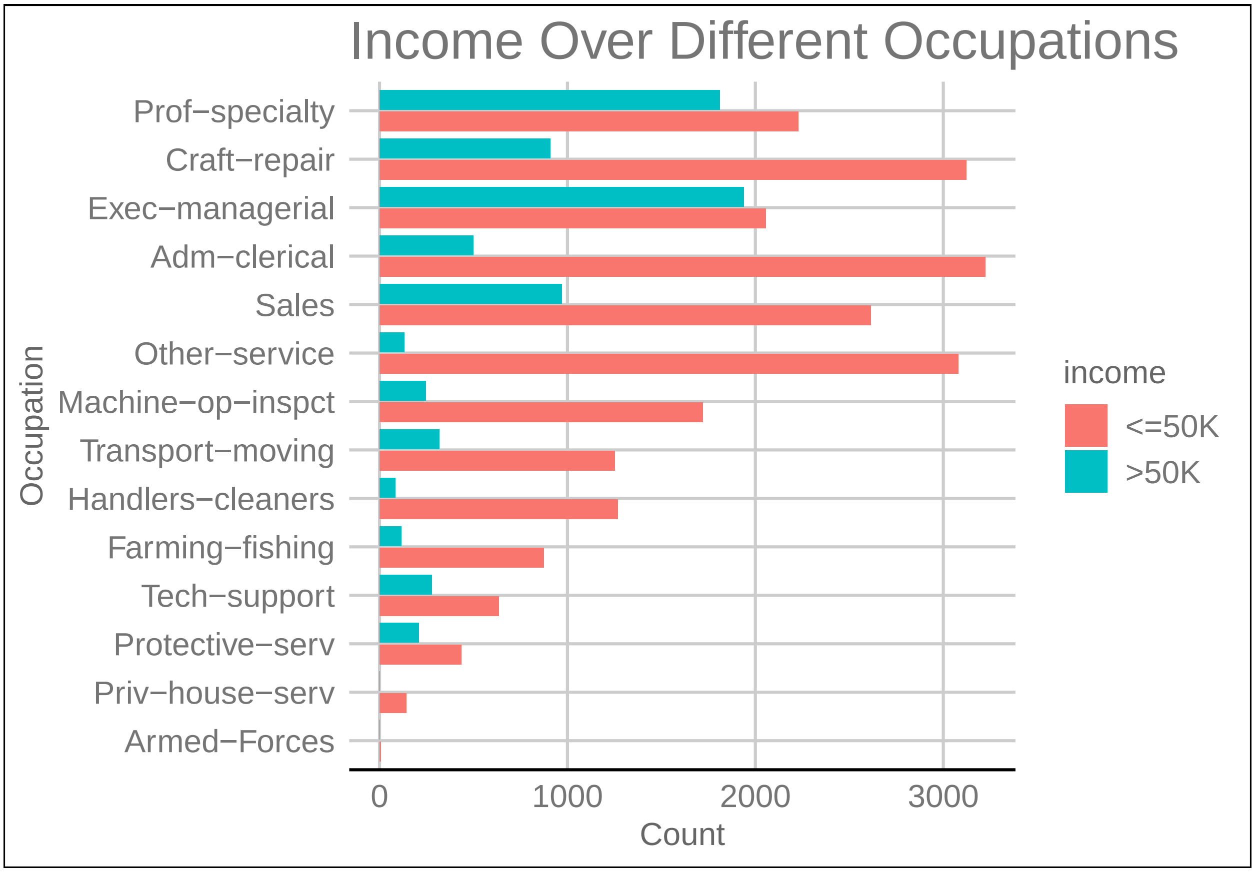
Here we analyse the distribution of incomes of the individuals in the database, on the basis of their occupation.

```
train_set %>% group_by(occupation,income) %>%
  summarise(count = n()) %>% knitr::kable()
```

occupation	income	count
Adm-clerical	<=50K	3223
Adm-clerical	>50K	498
Armed-Forces	<=50K	8
Armed-Forces	>50K	1
Craft-repair	<=50K	3122
Craft-repair	>50K	908
Exec-managerial	<=50K	2055
Exec-managerial	>50K	1937
Farming-fishing	<=50K	874

occupation	income	count
Farming-fishing	>50K	115
Handlers-cleaners	<=50K	1267
Handlers-cleaners	>50K	83
Machine-op-inspct	<=50K	1721
Machine-op-inspct	>50K	245
Other-service	<=50K	3080
Other-service	>50K	132
Priv-house-serv	<=50K	142
Priv-house-serv	>50K	1
Prof-specialty	<=50K	2227
Prof-specialty	>50K	1811
Protective-serv	<=50K	434
Protective-serv	>50K	210
Sales	<=50K	2614
Sales	>50K	970
Tech-support	<=50K	634
Tech-support	>50K	278
Transport-moving	<=50K	1253
Transport-moving	>50K	319

```r
train_set %>% group_by(occupation,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(reorder(occupation,count),count,fill = income)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  coord_flip() +
  ggtitle("Income Over Different Occupations") +
  xlab("Occupation") + ylab("Count") + theme_gdocs()
```



Here, in general, it is observed the people earning <=50K are greater than the number of people earning >50K.
However, it can also be seen that occupations such as *Prof-speciality* and *Exec-managerial*, usually requiring a *Master's Degree/ Doctrate*, have a comparitively little difference in the number of people earning <=50K & >50K.
This observation, also acts as a supporter to the conclusion drawn using the Income Over Education visualisation.

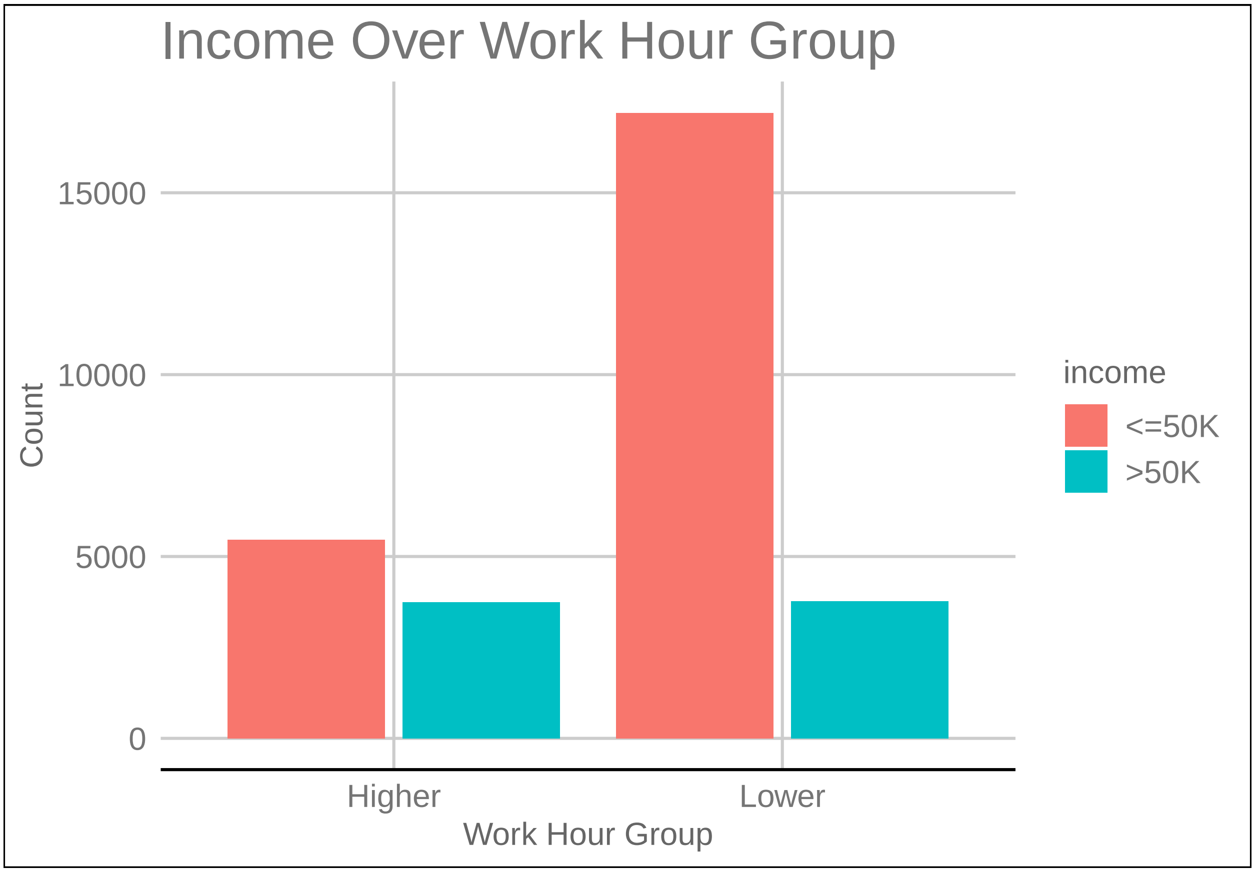
Income Over Hours Per Week

In this section, we analyse the income distribution, for the people who work less than the average hours per week, and more than the average hours per week.
We also analyse the Hours Per Week, with respect to the Gender of the individual, to determine, whether there is difference in the trend, if considered on a gender basis.

```
train_set %>% group_by(work_hour_group,income) %>%
  summarise(count = n()) %>% knitr::kable()
```

work_hour_group	income	count
Higher	<=50K	5456
Higher	>50K	3741
Lower	<=50K	17198
Lower	>50K	3767

```
train_set %>% group_by(work_hour_group,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(work_hour_group,count,fill = income)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  xlab("Work Hour Group") +
  ylab("Count") +
  ggtitle("Income Over Work Hour Group") +
  theme_gdocs()
```

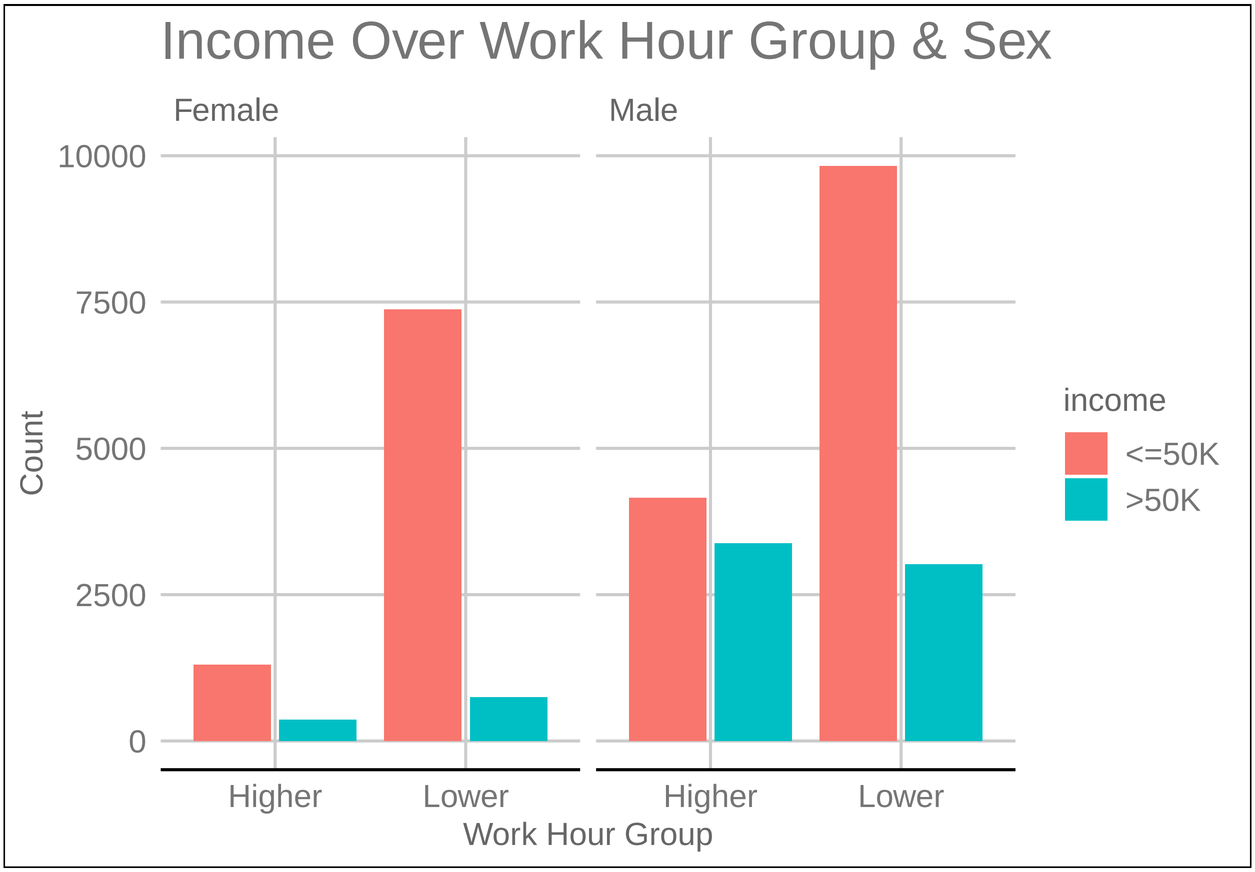


It can be clearly observed that for the people who work *Greater than the Average Hours Per Week*, the proportion of people earning <=50K, still remains larger, than the people earning >50K. However, it can also be seen that for people working *Greater than the Average Hours Per Week*, the difference in numbers is very small, when compared to the people working *Less Than The Average Hours Per Week*.

Now, we compare this trend, by also considering, the gender of the individuals.

```
train_set %>% group_by(work_hour_group,income,sex) %>%
  summarise(count = n()) %>%
  ggplot(aes(work_hour_group,count,fill = income)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  xlab("Work Hour Group") +
  ylab("Count") +
  ggtitle("Income Over Work Hour Group & Sex") +
```

```r
  theme_gdocs() + facet_grid(.~sex)
```



It can be seen that the *Gender* does not act as a *Bias*, in anyway, and the trend observed earlier, still remains true.

Income Over Maritial Status

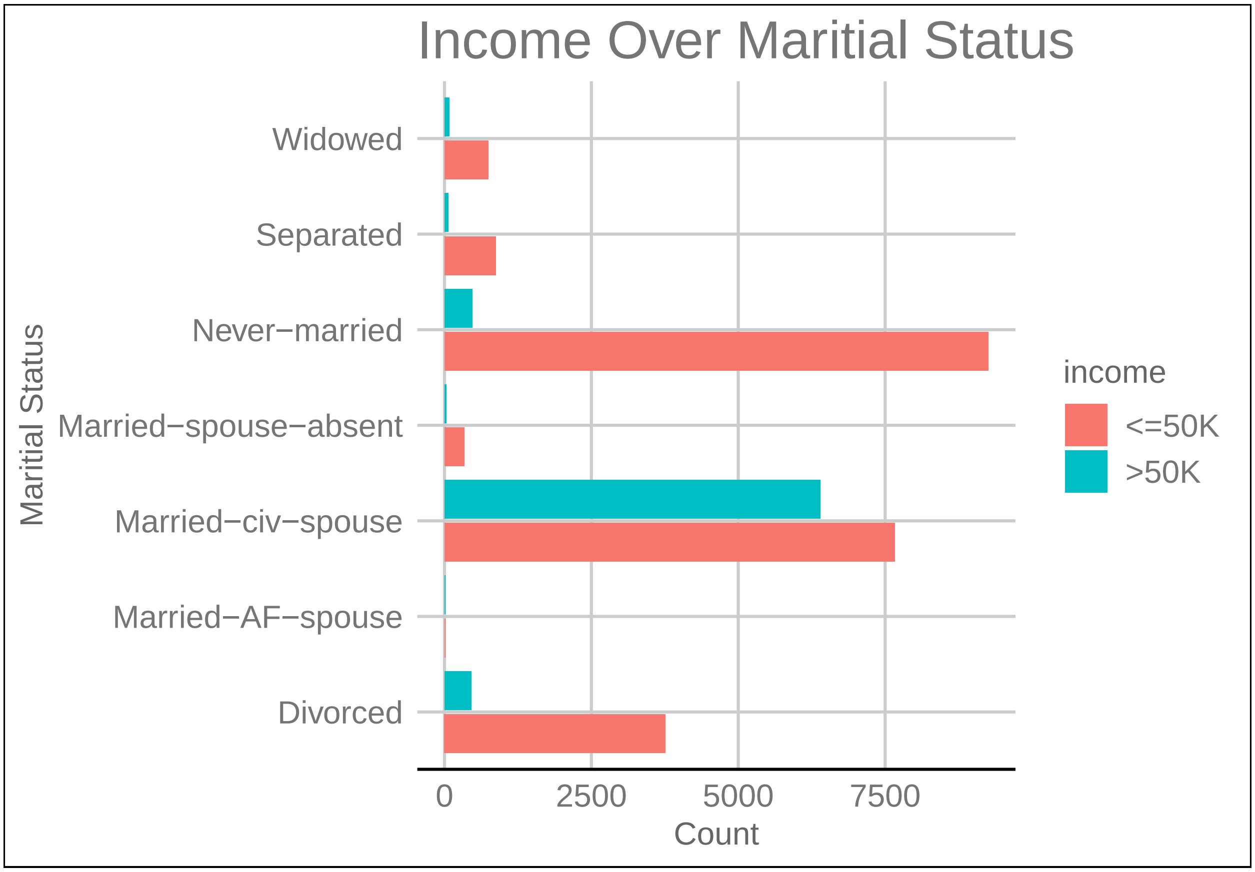
Here, we analyse the distribution of income, based on the individual's maritial status.

```r
train_set %>% group_by(marriage,income) %>%
  summarise(count = n()) %>% knitr::kable()
```

marriage	income	count
Divorced	<=50K	3762
Divorced	>50K	452
Married-AF-spouse	<=50K	11
Married-AF-spouse	>50K	10
Married-civ-spouse	<=50K	7666
Married-civ-spouse	>50K	6399
Married-spouse-absent	<=50K	339
Married-spouse-absent	>50K	31
Never-married	<=50K	9256
Never-married	>50K	470
Separated	<=50K	873

marriage	income	count
Separated	>50K	66
Widowed	<=50K	747
Widowed	>50K	80

```
train_set %>% group_by(marriage,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(marriage,count,fill = income)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  xlab("Maritial Status") +
  ylab("Count") +
  ggtitle("Income Over Maritial Status") +
  theme_gdocs() + coord_flip()
```



In the visualisaton, it can be clearly observed, that the chances of a person earning <=50K of not being married, are higher than that of being married.
The highest number of people earning <=50K, have never married.

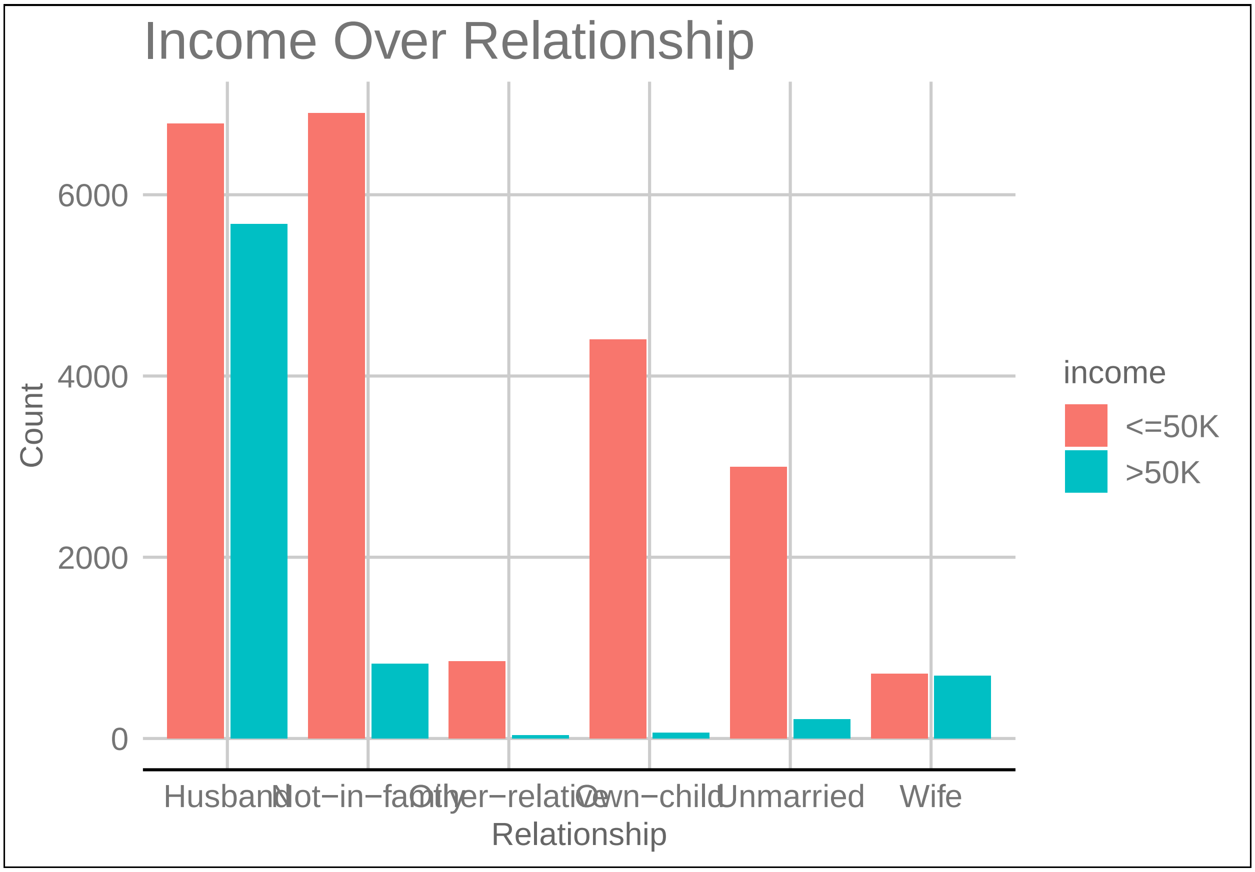
Income Over Relationship

Here, we verify the observations that have been observed in the previous visulation.

```r
train_set %>% group_by(relationship,income) %>%
  summarise(count = n()) %>% knitr::kable()
```

relationship	income	count
Husband	<=50K	6784
Husband	>50K	5679
Not-in-family	<=50K	6903
Not-in-family	>50K	823
Other-relative	<=50K	854
Other-relative	>50K	35
Own-child	<=50K	4402
Own-child	>50K	64
Unmarried	<=50K	2999
Unmarried	>50K	213
Wife	<=50K	712
Wife	>50K	694

```r
train_set %>% group_by(relationship,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(relationship,count,fill = income)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  xlab("Relationship") +
  ylab("Count") +
  ggtitle("Income Over Relationship") +
  theme_gdocs()
```

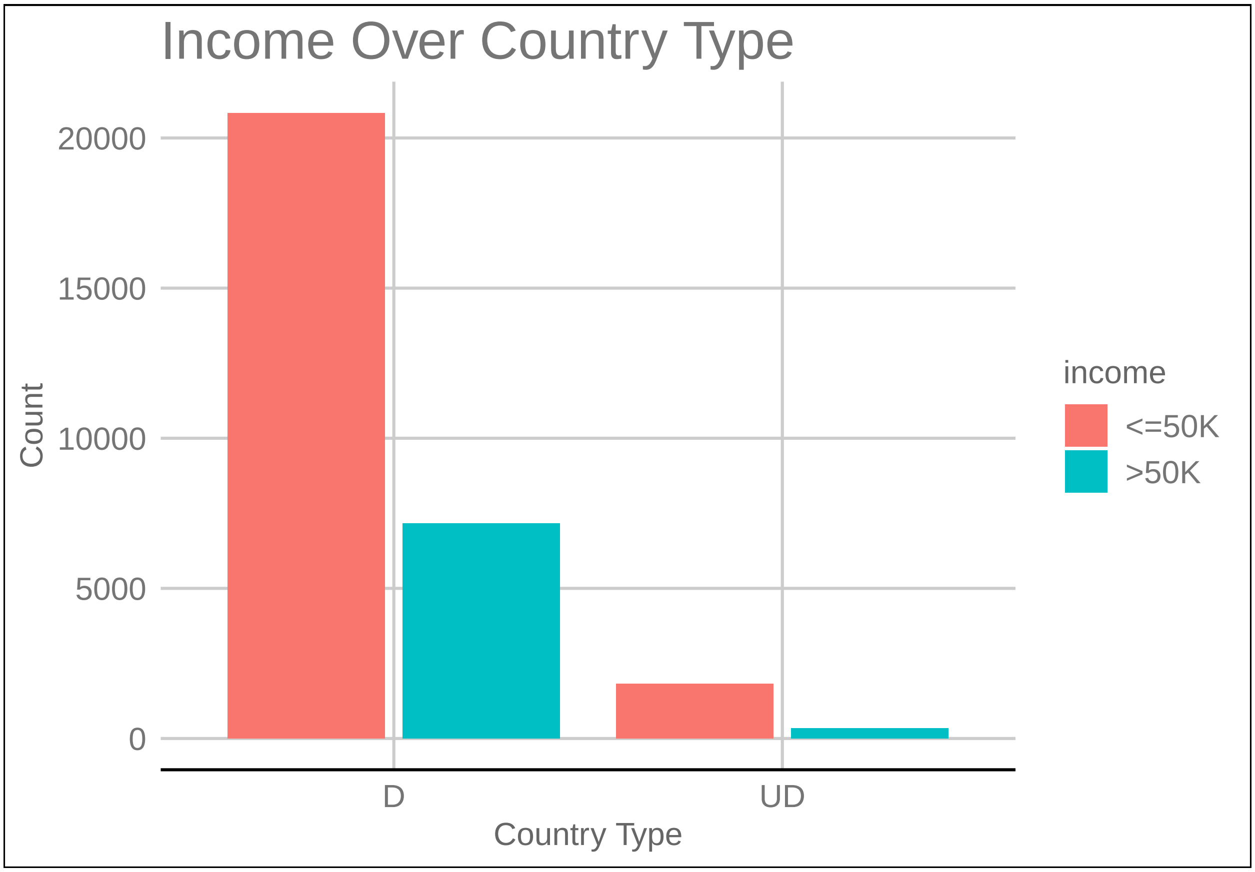


It can be clearly observed that the highest number of people earning <=50K, are current not in a family, confirming our previous conclusion, *Person earning <=50K, has a higher probability of being never married.*

Income Over Country Type

Here, we analyse whether the distribution of incomes, difference in *Developed vs Under Development Countries.*

```
train_set %>% group_by(country_type,income) %>%
  summarise(count = n()) %>%
  ggplot(aes(country_type,count,fill = income)) +
  geom_bar(stat = "identity",position = position_dodge2()) +
  ggtitle("Income Over Country Type") +
  xlab("Country Type") +
  ylab("Count") + theme_gdocs()
```



Here, it can clearly be seen, that most of the individuals of the dataset, are native to a developed country.
Secondly, be it developed or under development country, the number of people earning <=50K are significantly higher than number of people earning >50K.

Model Fitting

Based on the *Exploratory Data Analysis*, we will use only the following columns for the training of the *ML Models*.
1. **sex**
2. **age**
3. **occupation**
4. **education**
5. **relationship**
6. **race**
7. **marriage**
8. **hours**
9. **country**
10. **y**

```
selected_features <- c("sex","age","occupation",
                       "education","relationship","race",
                       "marriage","hours","country","y")
```

```r
#Selecting Columns From Training Set
train_set <- train_set %>% select(selected_features)

#Selecting Columns From Testing Set
test_set <- test_set %>% select(selected_features)
```

Using Logistic Regression

First, we define *K Fold Cross Validation*. Here, we define, that only *10 Times (K = 10)*, will be performed, and the validation set *will be 10% of the training set (p = 0.9)*.

```r
control <- trainControl(method = "cv",number = 10,p = 0.9)
```

Now, we fit the *Logistic Regression Model*, to the training set, using the train() function of the caret package.

```r
lga_fit <- train(y ~ .,data = train_set,method = "glm",trControl = control)
```

Now, we predict the income for the *Testing Set* by making of the predict() function.

```r
y_hat <- predict(lga_fit,test_set)
```

We can analyse the performance of the Model, by making use of the Confusion Matrix.

```r
confusionMatrix(y_hat,test_set$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 10412  1651
##          1   948  2049
##
##                Accuracy : 0.8274
##                  95% CI : (0.8213, 0.8334)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5025
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9165
##             Specificity : 0.5538
##          Pos Pred Value : 0.8631
##          Neg Pred Value : 0.6837
##              Prevalence : 0.7543
##          Detection Rate : 0.6914
##    Detection Prevalence : 0.8010
##       Balanced Accuracy : 0.7352
```

```
##
##        'Positive' Class : 0
##
```

The *Accuracy* for the *Logistic Regression Model*.

```
confusionMatrix(y_hat,test_set$y)$overall[1]
```

```
##  Accuracy
## 0.8274236
```

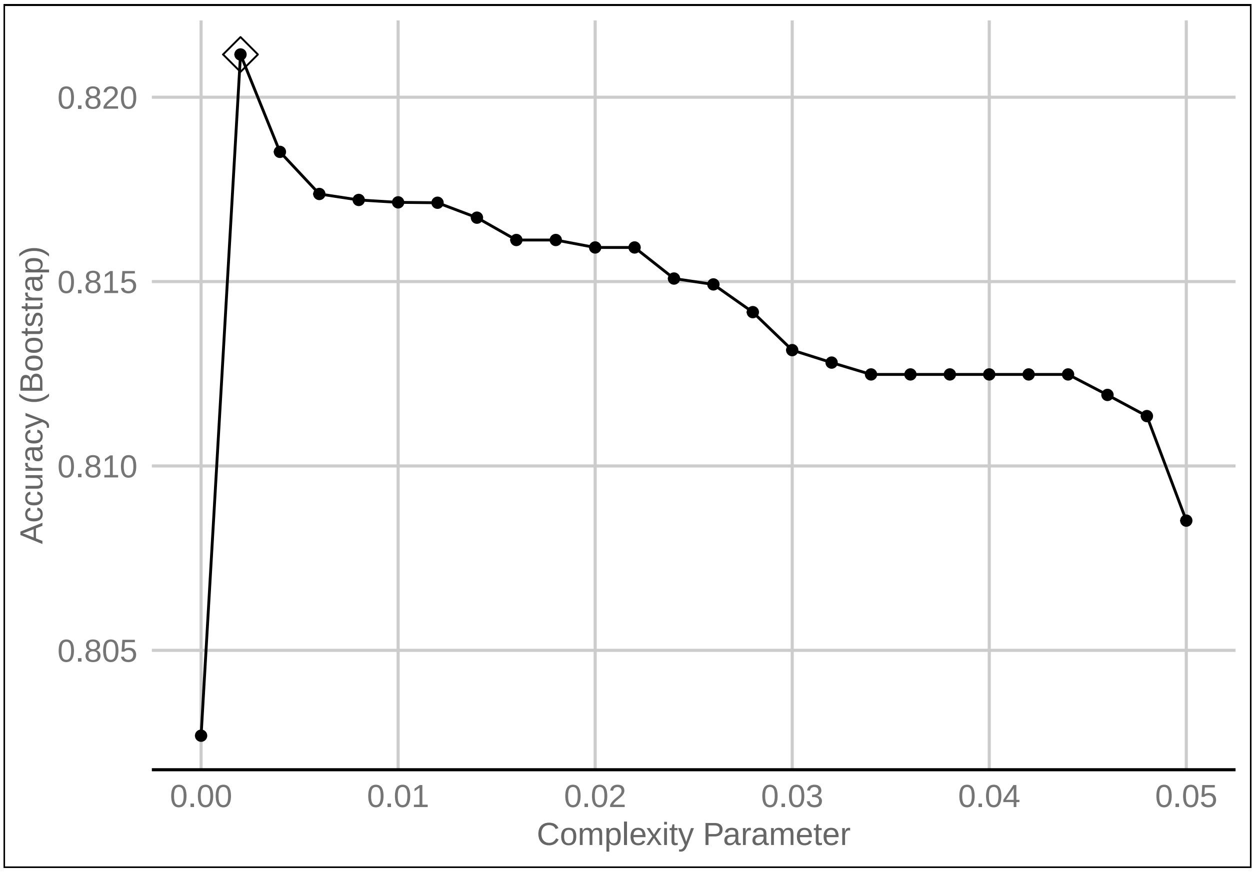
Using Classifier Tree

Here, we make of *tuneGrid* argument, to fit the Classifier tree, for multiple *cp (Complexity Parameter)* values, and fit the model using the *train()* function of the caret package.

```
tree_fit <- train(y ~ .,data = train_set,
                  method="rpart",
                  tuneGrid=data.frame(cp=seq(0,0.05,0.002)))
```

Now, we plot the *cp (Complexity Parameter) VS Accuracy*.

```
ggplot(tree_fit,highlight = TRUE) + theme_gdocs()
```



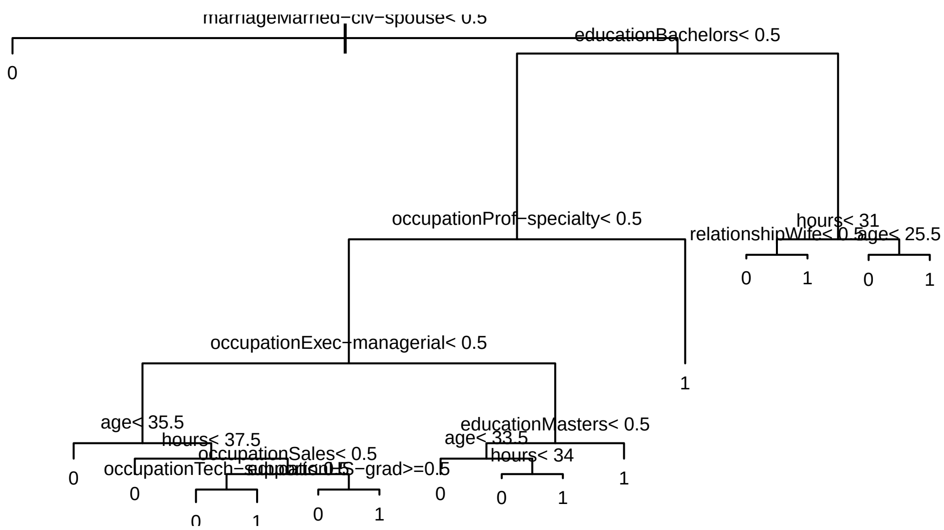
The best value of *cp* is :

```r
tree_fit$bestTune
```

```
##      cp
## 2 0.002
```

Now, we plot the *Best Fit Classifier Tree Model*.

```r
plot(tree_fit$finalModel,margin = 0.01)
text(tree_fit$finalModel,cex = 0.6)
```



Now, we predict the income for the *Testing Set* by making of the predict() function.

```r
y_hat <- predict(tree_fit,test_set)
```

We can analyse the performance of the Model, by making use of the Confusion Matrix.

```r
confusionMatrix(y_hat,test_set$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 10464  1755
##          1   896  1945
##
##                Accuracy : 0.824
```

```
##                  95% CI : (0.8178, 0.83)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4847
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9211
##             Specificity : 0.5257
##          Pos Pred Value : 0.8564
##          Neg Pred Value : 0.6846
##              Prevalence : 0.7543
##          Detection Rate : 0.6948
##    Detection Prevalence : 0.8114
##       Balanced Accuracy : 0.7234
##
##        'Positive' Class : 0
##
```

The *Accuracy* for the *Classifier Tree Model.*

```r
confusionMatrix(y_hat,test_set$y)$overall[1]
```

```
##  Accuracy
## 0.8239708
```

Conclusion

For the *Adult Census Income* dataset, both the *Logistic Regression & Classifier Tree* models, perform equally good, by making use of *Accuracy* as the comparison parameter, and hence any of the model can be used for predicting the income of an individual.

Github Link

https://github.com/guptaharshnavin/Adult_Census_Income