Bhavesh_project

Bhavesh

2023-05-14

```
setwd("C:\\Users\\visha\\OneDrive\\Desktop\\STAT 515\\Final Project 2")
```

```
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.2.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.2.3
```

```
# Load the required libraries
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

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

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

```
## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.2.3
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.2.3
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.2.3
```

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.2.3
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'tidyr'
```

```
## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
library(purrr)
```

```
## Warning: package 'purrr' was built under R version 4.2.3
```

```
##
## Attaching package: 'purrr'
```

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

```
## The following object is masked from 'package:plyr':
##
##     compact
```

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.2.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:xgboost':
##
##     slice
```

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

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

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
X.adult=read.csv("adult.data.csv")

colnames(X.adult) = c("age", "workclass", "fnlwgt", "education", "education_num",
                      "marital_status", "occupation", "relationship", "race", "sex",
                      "capital_gain", "capital_loss", "hours_per_week", "native_country",
                      "income_class")
```

```
str(X.adult)
```

```
## 'data.frame':    32560 obs. of  15 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : chr  "Self-emp-not-inc" "Private" "Private" "Private" ...
##  $ fnlwgt        : int  83311 215646 234721 338409 284582 160187 209642 45781 159449 28046
4 ...
##  $ education     : chr  "Bachelors" "HS-grad" "11th" "Bachelors" ...
##  $ education_num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital_status: chr  "Married-civ-spouse" "Divorced" "Married-civ-spouse" "Married-civ-
spouse" ...
##  $ occupation    : chr  "Exec-managerial" "Handlers-cleaners" "Handlers-cleaners" "Prof-sp
ecialty" ...
##  $ relationship  : chr  "Husband" "Not-in-family" "Husband" "Wife" ...
##  $ race          : chr  "White" "White" "Black" "Black" ...
##  $ sex           : chr  "Male" "Male" "Male" "Female" ...
##  $ capital_gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native_country: chr  "United-States" "United-States" "United-States" "Cuba" ...
##  $ income_class  : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

```
X.adult$workclass = as.character(X.adult$workclass)
X.adult$occupation = as.character(X.adult$occupation)
X.adult$native_country = as.character(X.adult$native_country)
X.adult$race = as.character(X.adult$race)
X.adult$marital_status = as.character(X.adult$marital_status)
```

```
str(X.adult)
```

```
## 'data.frame':    32560 obs. of  15 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : chr  "Self-emp-not-inc" "Private" "Private" "Private" ...
##  $ fnlwgt        : int  83311 215646 234721 338409 284582 160187 209642 45781 159449 28046
4 ...
##  $ education     : chr  "Bachelors" "HS-grad" "11th" "Bachelors" ...
##  $ education_num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital_status: chr  "Married-civ-spouse" "Divorced" "Married-civ-spouse" "Married-civ-
spouse" ...
##  $ occupation    : chr  "Exec-managerial" "Handlers-cleaners" "Handlers-cleaners" "Prof-sp
ecialty" ...
##  $ relationship  : chr  "Husband" "Not-in-family" "Husband" "Wife" ...
##  $ race          : chr  "White" "White" "Black" "Black" ...
##  $ sex           : chr  "Male" "Male" "Male" "Female" ...
##  $ capital_gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native_country: chr  "United-States" "United-States" "United-States" "Cuba" ...
##  $ income_class  : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

```
unique(X.adult$education)
```

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

```
X.adult$education <- ifelse(X.adult$education %in% c("1st-4th", "5th-6th", "7th-8th", "9th", 
"10th", "11th", "12th", "HS-grad", "Preschool"), "School-Level", X.adult$education)

X.adult$education <- ifelse(X.adult$education %in% c("Bachelors", "Some-college"), "UG", X.ad
ult$education)

X.adult$education <- ifelse(X.adult$education %in% c("Masters", "Doctorate"), "PG", X.adult$e
ducation)

X.adult$education <- ifelse(X.adult$education %in% c("Assoc-acdm", "Assoc-voc","Prof-schoo
l"), "other", X.adult$education)
```

```
unique(X.adult$education)
```

```
## [1] "UG"           "School-Level" "PG"           "other"
```

```
X.adult$marital_status[X.adult$marital_status=="Never-married"] = "Never-Married"
X.adult$marital_status[X.adult$marital_status=="Married-AF-spouse"] = "Married"
X.adult$marital_status[X.adult$marital_status=="Married-civ-spouse"] = "Married"
X.adult$marital_status[X.adult$marital_status=="Married-spouse-absent"] = "Not-Married"
X.adult$marital_status[X.adult$marital_status=="Separated"] = "Not-Married"
X.adult$marital_status[X.adult$marital_status=="Divorced"] = "Not-Married"
X.adult$marital_status[X.adult$marital_status=="Widowed"] = "Widowed"

X.adult$native_country[X.adult$native_country=="Cambodia"] = "SE-Asia"
X.adult$native_country[X.adult$native_country=="Canada"] = "British-Commonwealth"
X.adult$native_country[X.adult$native_country=="China"] = "China"
X.adult$native_country[X.adult$native_country=="Columbia"] = "South-America"
X.adult$native_country[X.adult$native_country=="Cuba"] = "Other"
X.adult$native_country[X.adult$native_country=="Dominican-Republic"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Ecuador"] = "South-America"
X.adult$native_country[X.adult$native_country=="El-Salvador"] = "South-America"
X.adult$native_country[X.adult$native_country=="England"] = "British-Commonwealth"
X.adult$native_country[X.adult$native_country=="France"] = "Euro_1"
X.adult$native_country[X.adult$native_country=="Germany"] = "Euro_1"
X.adult$native_country[X.adult$native_country=="Greece"] = "Euro_2"
X.adult$native_country[X.adult$native_country=="Guatemala"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Haiti"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Holand-Netherlands"] = "Euro_1"
X.adult$native_country[X.adult$native_country=="Honduras"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Hong"] = "China"
X.adult$native_country[X.adult$native_country=="Hungary"] = "Euro_2"
X.adult$native_country[X.adult$native_country=="India"] = "British-Commonwealth"
X.adult$native_country[X.adult$native_country=="Iran"] = "Other"
X.adult$native_country[X.adult$native_country=="Ireland"] = "British-Commonwealth"
X.adult$native_country[X.adult$native_country=="Italy"] = "Euro_1"
X.adult$native_country[X.adult$native_country=="Jamaica"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Japan"] = "Other"
X.adult$native_country[X.adult$native_country=="Laos"] = "SE-Asia"
X.adult$native_country[X.adult$native_country=="Mexico"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Nicaragua"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Outlying-US(Guam-USVI-etc)"] = "Latin-Americ
a"
X.adult$native_country[X.adult$native_country=="Peru"] = "South-America"
X.adult$native_country[X.adult$native_country=="Philippines"] = "SE-Asia"
X.adult$native_country[X.adult$native_country=="Poland"] = "Euro_2"
X.adult$native_country[X.adult$native_country=="Portugal"] = "Euro_2"
X.adult$native_country[X.adult$native_country=="Puerto-Rico"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="Scotland"] = "British-Commonwealth"
X.adult$native_country[X.adult$native_country=="South"] = "Euro_2"
X.adult$native_country[X.adult$native_country=="Taiwan"] = "China"
X.adult$native_country[X.adult$native_country=="Thailand"] = "SE-Asia"
X.adult$native_country[X.adult$native_country=="Trinadad&amp;Tobago"] = "Latin-America"
X.adult$native_country[X.adult$native_country=="United-States"] = "United-States"
X.adult$native_country[X.adult$native_country=="Vietnam"] = "SE-Asia"
X.adult$native_country[X.adult$native_country=="Yugoslavia"] = "Euro_2"

X.adult$workclass = gsub("^Federal-gov","Federal-Govt",X.adult$workclass)
X.adult$workclass = gsub("^Local-gov","Other-Govt",X.adult$workclass)
X.adult$workclass = gsub("^State-gov","Other-Govt",X.adult$workclass)
X.adult$workclass = gsub("^Private","Private",X.adult$workclass)
```

```
X.adult$workclass = gsub("^Self-emp-inc","Self-Employed",X.adult$workclass)
X.adult$workclass = gsub("^Self-emp-not-inc","Self-Employed",X.adult$workclass)
X.adult$workclass = gsub("^Without-pay","Not-Working",X.adult$workclass)
X.adult$workclass = gsub("^Never-worked","Not-Working",X.adult$workclass)

X.adult$occupation = gsub("^Adm-clerical","Admin",X.adult$occupation)
X.adult$occupation = gsub("^Armed-Forces","Military",X.adult$occupation)
X.adult$occupation = gsub("^Craft-repair","Blue-Collar",X.adult$occupation)
X.adult$occupation = gsub("^Exec-managerial","White-Collar",X.adult$occupation)
X.adult$occupation = gsub("^Farming-fishing","Blue-Collar",X.adult$occupation)
X.adult$occupation = gsub("^Handlers-cleaners","Blue-Collar",X.adult$occupation)
X.adult$occupation = gsub("^Machine-op-inspct","Blue-Collar",X.adult$occupation)
X.adult$occupation = gsub("^Other-service","Service",X.adult$occupation)
X.adult$occupation = gsub("^Priv-house-serv","Service",X.adult$occupation)
X.adult$occupation = gsub("^Prof-specialty","Professional",X.adult$occupation)
X.adult$occupation = gsub("^Protective-serv","Other-Occupations",X.adult$occupation)
X.adult$occupation = gsub("^Sales","Sales",X.adult$occupation)
X.adult$occupation = gsub("^Tech-support","Other-Occupations",X.adult$occupation)
X.adult$occupation = gsub("^Transport-moving","Blue-Collar",X.adult$occupation)

X.adult$race[X.adult$race=="White"] = "White"
X.adult$race[X.adult$race=="Black"] = "Black"
X.adult$race[X.adult$race=="Amer-Indian-Eskimo"] = "Amer-Indian"
X.adult$race[X.adult$race=="Asian-Pac-Islander"] = "Asian"
X.adult$race[X.adult$race=="Other"] = "Other"





is.na(X.adult) = X.adult=='?'
is.na(X.adult) = X.adult==' ?'
#X.adult = na.omit(X.adult)
```

```
str(X.adult)
```

```
## 'data.frame':    32560 obs. of  15 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : chr  "Self-Employed" "Private" "Private" "Private" ...
##  $ fnlwgt        : int  83311 215646 234721 338409 284582 160187 209642 45781 159449 28046
4 ...
##  $ education     : chr  "UG" "School-Level" "School-Level" "UG" ...
##  $ education_num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital_status: chr  "Married" "Not-Married" "Married" "Married" ...
##  $ occupation    : chr  "White-Collar" "Blue-Collar" "Blue-Collar" "Professional" ...
##  $ relationship  : chr  "Husband" "Not-in-family" "Husband" "Wife" ...
##  $ race          : chr  "White" "White" "Black" "Black" ...
##  $ sex           : chr  "Male" "Male" "Male" "Female" ...
##  $ capital_gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native_country: chr  "United-States" "United-States" "United-States" "Other" ...
##  $ income_class  : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

```
# Replace "?" with NA
X.adult <- data.frame(lapply(X.adult, function(x) ifelse(x == "?", NA, x)))
```

```
# Check for missing values in each column
before_na_removing=sapply(X.adult, function(x) sum(is.na(x)))
before_na_removing
```

```
##            age      workclass         fnlwgt      education  education_num
##              0           1836              0              0              0
## marital_status     occupation   relationship           race            sex
##              0           1843              0              0              0
##   capital_gain   capital_loss hours_per_week native_country   income_class
##              0              0              0            583              0
```

```
# Count number of missing values in each column
missing_values <- colSums(is.na(X.adult))

# Get names of columns with missing values
names_with_na <- names(missing_values[missing_values > 0])
names_with_na
```

```
## [1] "workclass"      "occupation"     "native_country"
```

```
# Replace NA in workclass with the mode
X.adult$workclass[is.na(X.adult$workclass)] <- mode(X.adult$workclass)

# Replace NA in occupation with the mode
X.adult$occupation[is.na(X.adult$occupation)] <- mode(X.adult$occupation)

# Replace NA in native_country with the mode
X.adult$native_country[is.na(X.adult$native_country)] <- mode(X.adult$native_country)
```

```
# Count number of missing values in each column
missing_values <- colSums(is.na(X.adult))

missing_values
```

```
##            age      workclass         fnlwgt      education  education_num
##              0              0              0              0              0
## marital_status     occupation   relationship           race            sex
##              0              0              0              0              0
##   capital_gain   capital_loss hours_per_week native_country   income_class
##              0              0              0              0              0
```

```
# Get names of columns with missing values
names_with_na <- names(missing_values[missing_values > 0])
names_with_na
```

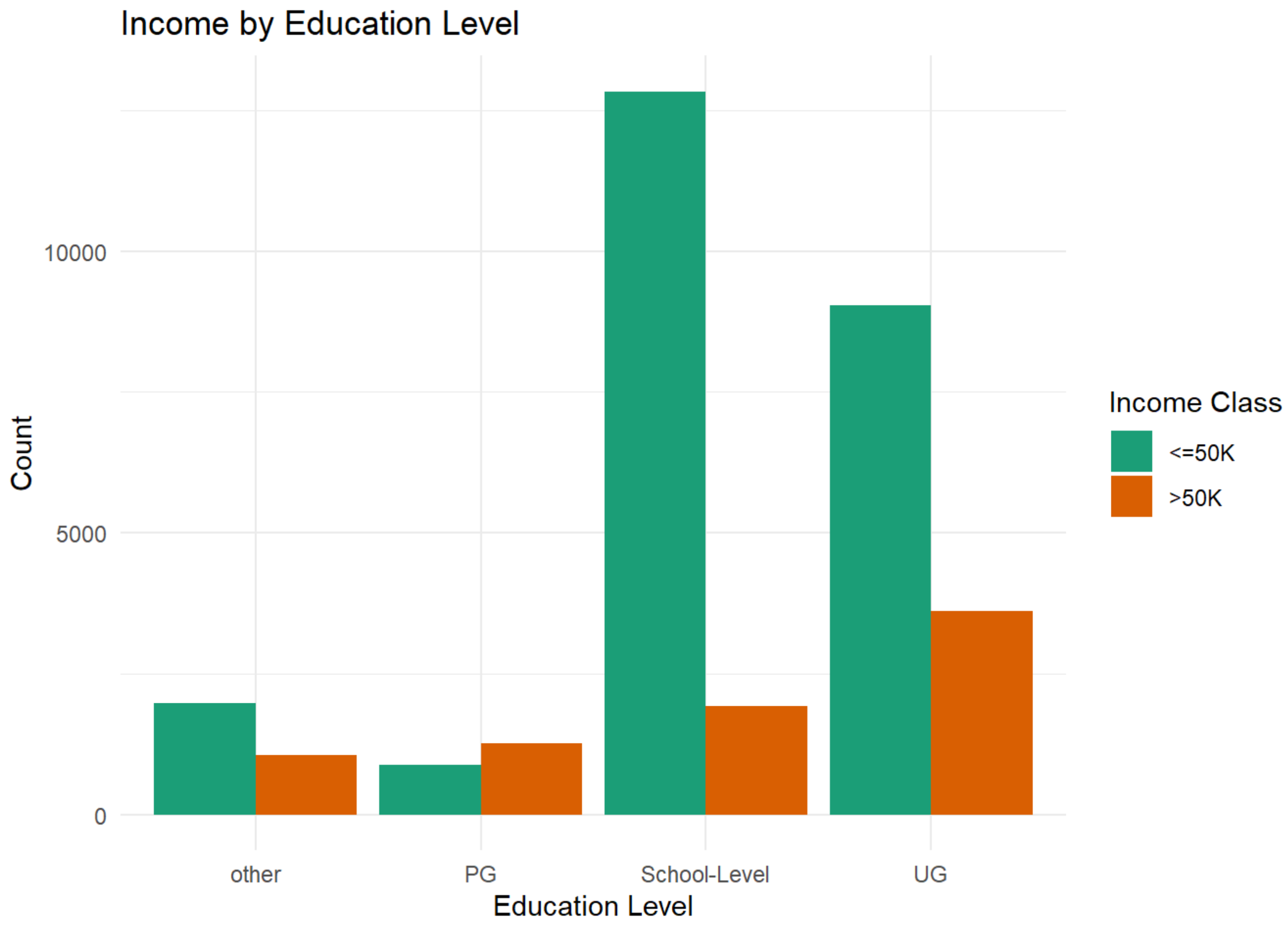
```
## character(0)
```

###EDA

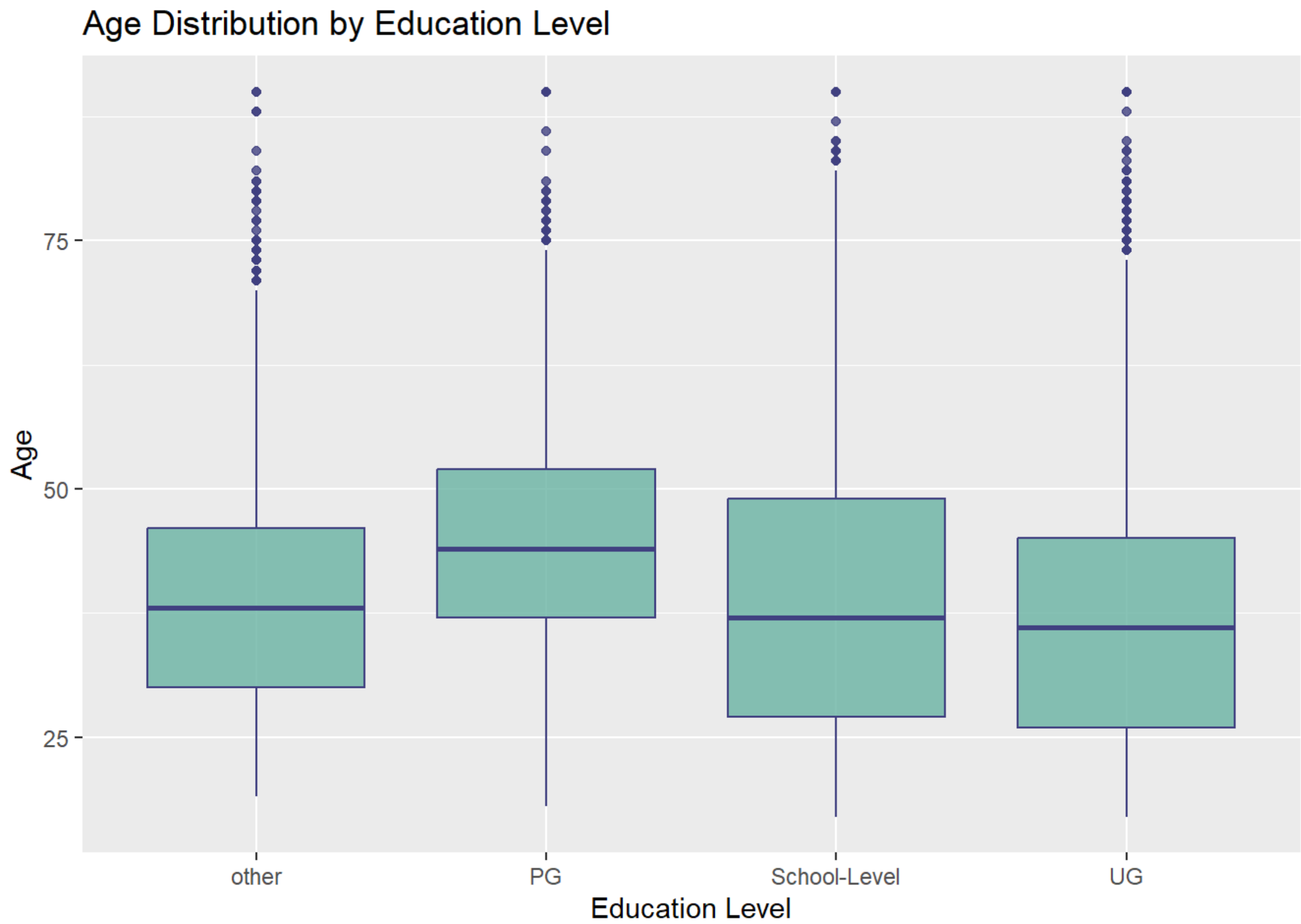
```
X.adult$workclass = as.character(X.adult$workclass)
X.adult$occupation = as.character(X.adult$occupation)
X.adult$native_country = as.character(X.adult$native_country)
X.adult$race = as.character(X.adult$race)
X.adult$marital_status = as.character(X.adult$marital_status)
X.adult$education=as.character((X.adult$education))
X.adult$income_class=as.character((X.adult$income_class))
```

```
library(ggplot2)

income_by_education<-ggplot(X.adult, aes(x = education, fill = income_class)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("#1b9e77", "#d95f02"),
                    name = "Income Class",
                    labels = c("<=50K", ">50K")) +
  labs(title = "Income by Education Level", x = "Education Level", y = "Count") +
  theme_minimal()
income_by_education
```

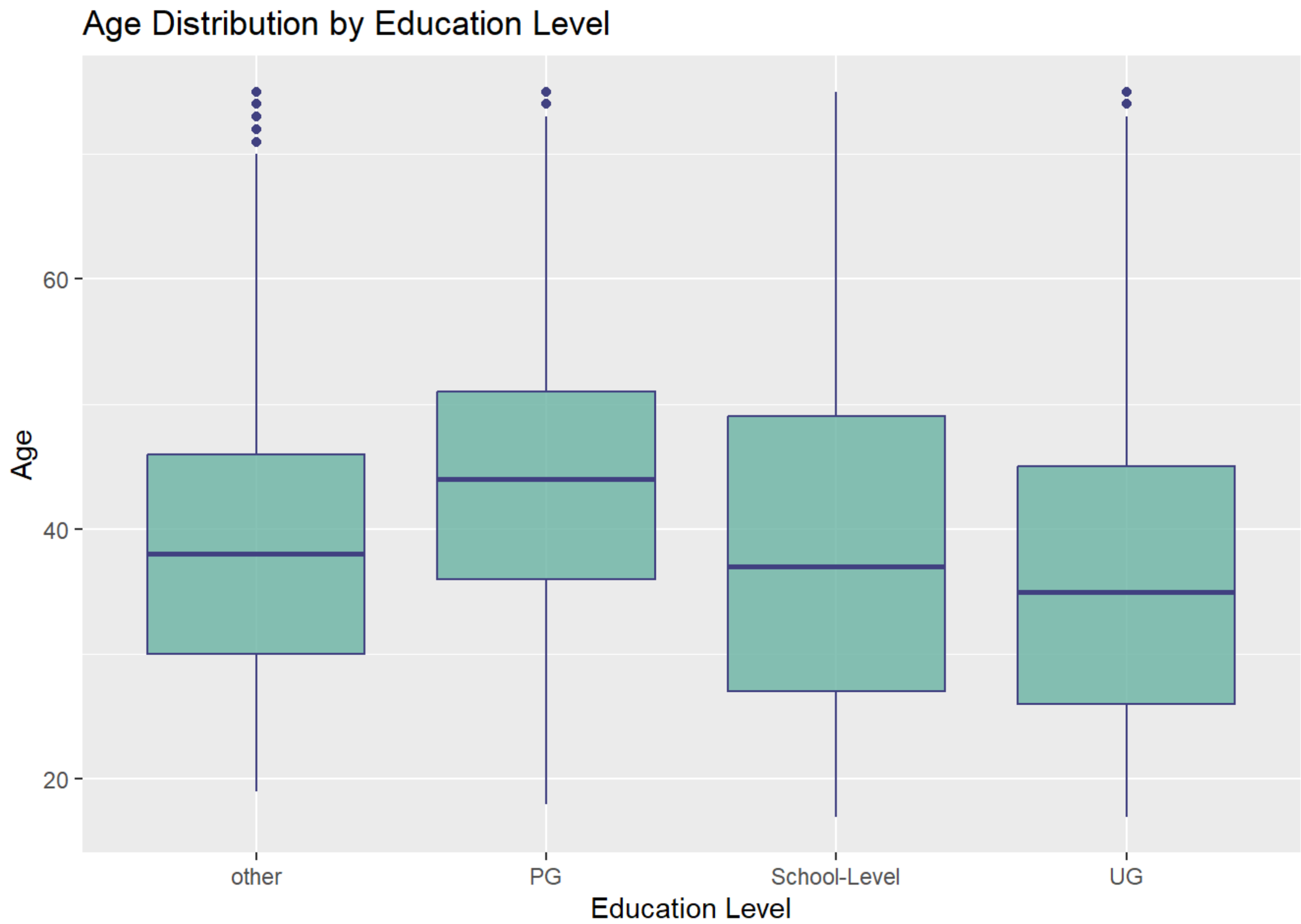



```
Age_Distribution_by_Education_Level<-ggplot(X.adult, aes(x = education, y = age)) +
  geom_boxplot(fill = "#69b3a2", color = "#404080", alpha = 0.8) +
  labs(title = "Age Distribution by Education Level", x = "Education Level", y = "Age")
Age_Distribution_by_Education_Level
```

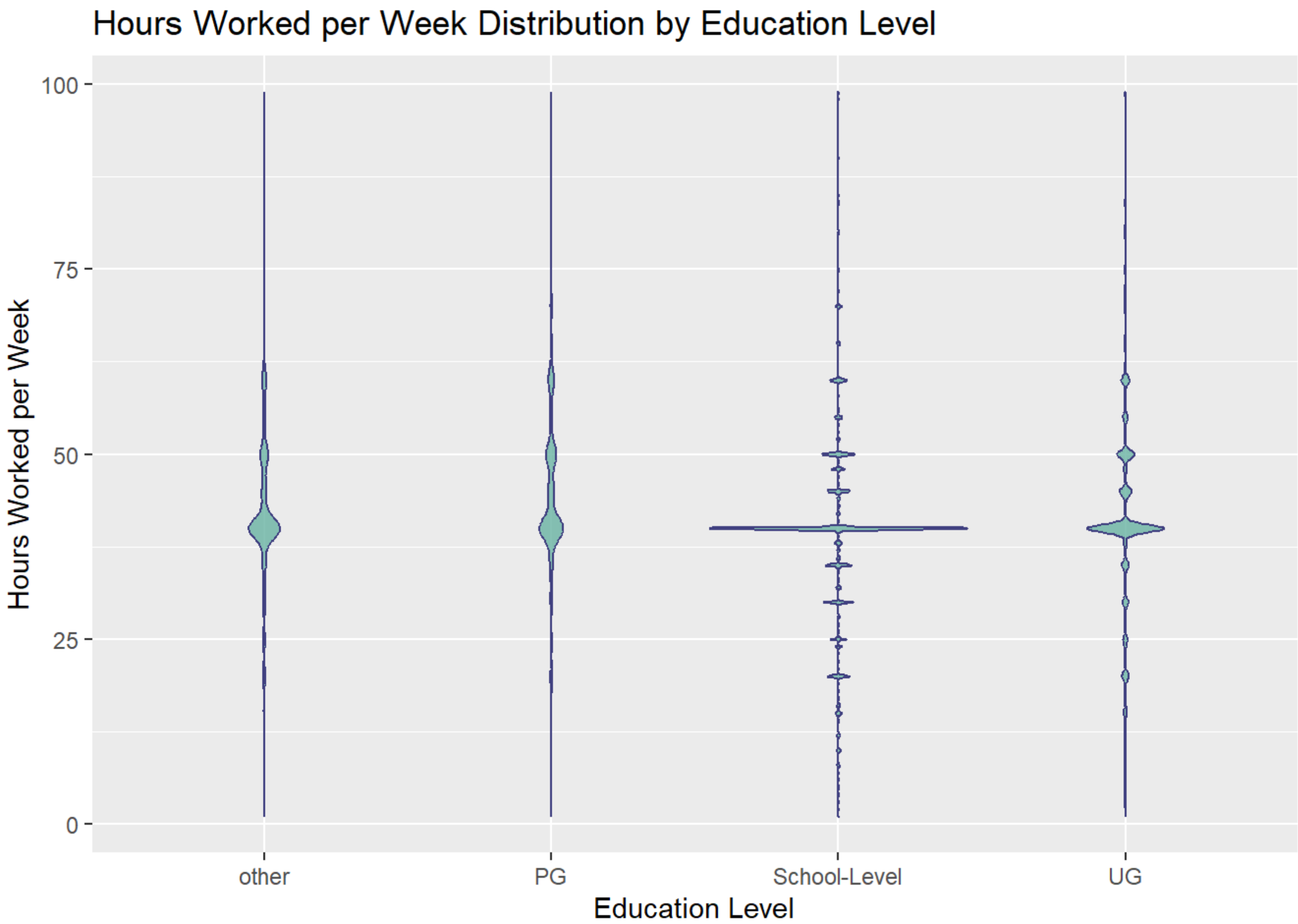



```
X.adult <- X.adult[X.adult$age <= 75, ]
```

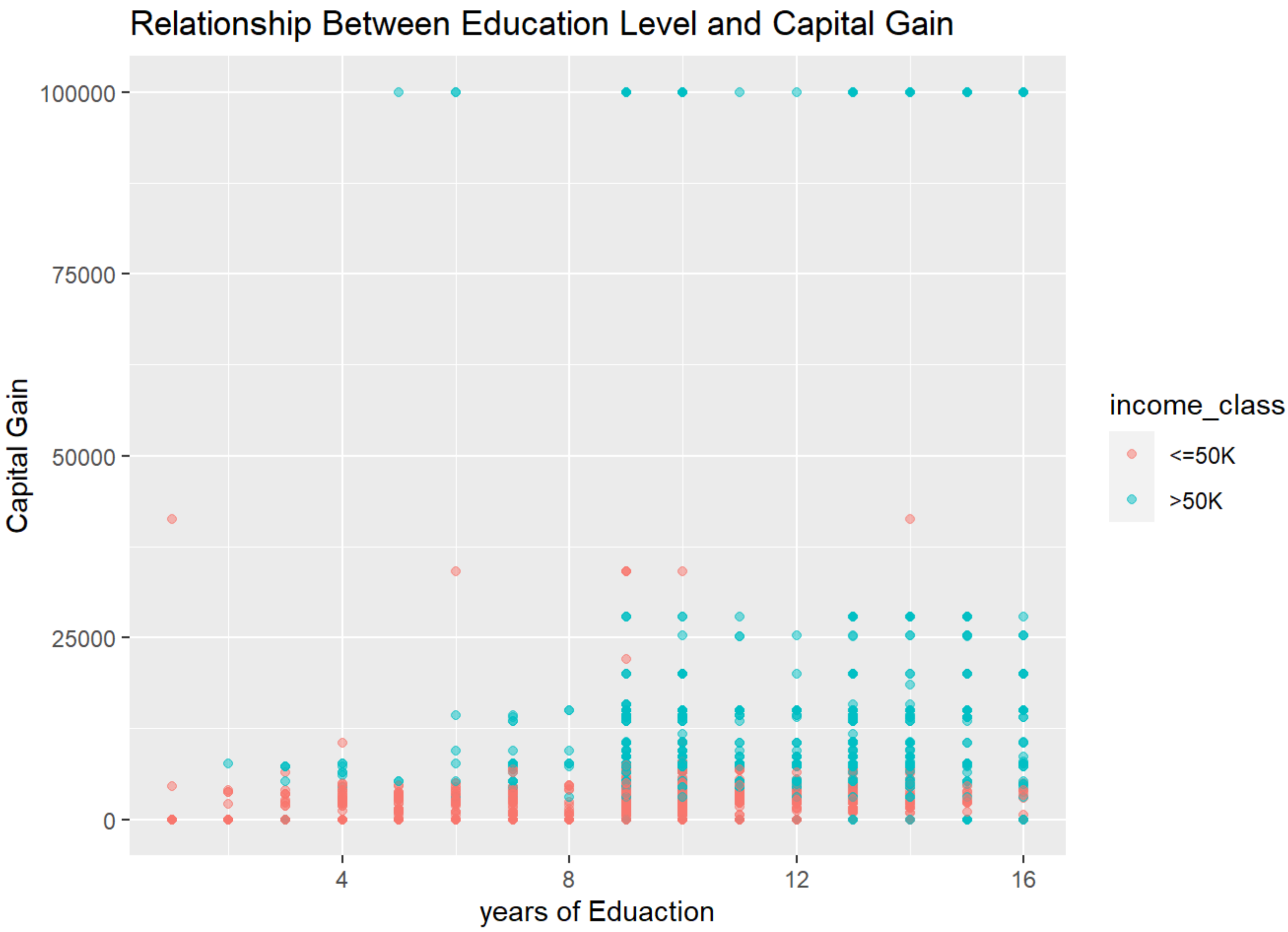
```
Age_Distribution_by_Education_Level_rmna<-ggplot(X.adult, aes(x = education, y = age)) +
  geom_boxplot(fill = "#69b3a2", color = "#404080", alpha = 0.8) +
  labs(title = "Age Distribution by Education Level", x = "Education Level", y = "Age")
Age_Distribution_by_Education_Level_rmna
```




```
ggplot(X.adult, aes(x = education, y = hours_per_week)) +
  geom_violin(fill = "#69b3a2", color = "#404080", alpha = 0.8) +
  labs(title = "Hours Worked per Week Distribution by Education Level", x = "Education Leve
l", y = "Hours Worked per Week")
```




```
ggplot(X.adult, aes(x = education_num, y = capital_gain, color = income_class)) +
  geom_point(alpha = 0.5) +
  labs(title = "Relationship Between Education Level and Capital Gain", x = "years of Eduacti
on", y = "Capital Gain")
```




```
X.adult$marital_status = factor(X.adult$marital_status)
X.adult$native_country = factor(X.adult$native_country)
X.adult$workclass = factor(X.adult$workclass)
X.adult$occupation = factor(X.adult$occupation)
X.adult$race = factor(X.adult$race)
X.adult$sex = factor(X.adult$sex)
X.adult$relationship = factor(X.adult$relationship)
X.adult$income_class = factor(X.adult$income_class)
X.adult$education=factor(X.adult$education)
```

```
str(X.adult)
```

```
## 'data.frame':    32319 obs. of  15 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 6 levels "character","Federal-Govt",..: 6 5 5 5 5 5 6 5 5 5 ...
##  $ fnlwgt        : int  83311 215646 234721 338409 284582 160187 209642 45781 159449 280464 ...
##  $ education     : Factor w/ 4 levels "other","PG","School-Level",..: 4 3 3 4 2 3 3 2 4 4 ...
##  $ education_num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital_status: Factor w/ 4 levels "Married","Never-Married",..: 1 3 1 1 1 3 1 2 1 1 ...
##  $ occupation    : Factor w/ 9 levels "Admin","Blue-Collar",..: 9 2 2 6 9 8 9 6 9 9 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 1 2 1 6 6 2 1 2 1 1 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian",..: 5 5 3 3 5 3 5 5 5 3 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 2 2 ...
##  $ capital_gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native_country: Factor w/ 11 levels "British-Commonwealth",..: 11 11 11 7 11 6 11 11 11 11 ...
##  $ income_class  : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 2 2 2 ...
```

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

# Split data into training and testing sets
trainIndex <- sample(1:nrow(X.adult), size = 0.7*nrow(X.adult), replace = FALSE)
train <- X.adult[trainIndex,]
test <- X.adult[-trainIndex,]

# Create the random forest model
rf_model <- randomForest(income_class ~ ., data = train, importance = TRUE, ntree = 500)

# Predict on the test data
rf_pred <- predict(rf_model, newdata = test)

# Evaluate the performance of the model
confusionMatrix(rf_pred, test$income_class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  6991  850
##      >50K    406 1449
##
##                Accuracy : 0.8705
##                  95% CI : (0.8636, 0.8771)
##     No Information Rate : 0.7629
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6164
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9451
##             Specificity : 0.6303
##          Pos Pred Value : 0.8916
##          Neg Pred Value : 0.7811
##              Prevalence : 0.7629
##          Detection Rate : 0.7210
##    Detection Prevalence : 0.8087
##       Balanced Accuracy : 0.7877
##
##        'Positive' Class : <=50K
##
```

```
str(X.adult)
```

```
## 'data.frame':    32319 obs. of  15 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 6 levels "character","Federal-Govt",..: 6 5 5 5 5 5 6 5 5 5
...
##  $ fnlwgt        : int  83311 215646 234721 338409 284582 160187 209642 45781 159449 28046
4 ...
##  $ education     : Factor w/ 4 levels "other","PG","School-Level",..: 4 3 3 4 2 3 3 2 4 4
...
##  $ education_num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital_status: Factor w/ 4 levels "Married","Never-Married",..: 1 3 1 1 1 3 1 2 1 1
...
##  $ occupation    : Factor w/ 9 levels "Admin","Blue-Collar",..: 9 2 2 6 9 8 9 6 9 9 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 1 2 1 6 6 2 1 2 1 1
...
##  $ race          : Factor w/ 5 levels "Amer-Indian",..: 5 5 3 3 5 3 5 5 5 3 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 2 2 ...
##  $ capital_gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native_country: Factor w/ 11 levels "British-Commonwealth",..: 11 11 11 7 11 6 11 11 11
11 ...
##  $ income_class  : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 2 2 2 ...
```

```
# Load the necessary libraries
library(glmnet)
library(caret)

# Split data into training and testing sets
set.seed(123)
trainIndex <- sample(1:nrow(X.adult), size = 0.7*nrow(X.adult), replace = FALSE)
train <- X.adult[trainIndex,c(1,3,5,11,12,13,15)]
test <- X.adult[-trainIndex,c(1,3,5,11,12,13,15)]
train1 <- X.adult[trainIndex,c(1,3,5,11,12,13,15)]
test1 <- X.adult[-trainIndex,c(1,3,5,11,12,13,15)]

# Check for missing values in test data
if (anyNA(test)) {
  stop("There are missing values in test dataset.")
}
```

```
# Create the logistic regression model using LASSO regularization
lasso_model <- cv.glmnet(as.matrix(train[, -7]), train$income_class, family = "binomial", alp
ha = 1)

# Find the optimal lambda value
lasso_best_lambda <- lasso_model$lambda.min

# Fit the model using the optimal lambda value
lasso_pred <- predict(lasso_model, newx = as.matrix(test[, -7]), s = lasso_best_lambda, type
= "response")

# Convert predicted probabilities to predicted classes
lasso_pred <- ifelse(lasso_pred > 0.5, ">50K", "<=50K")

lasso_pred=as.factor(lasso_pred)



str(lasso_pred)
```

```
##  Factor w/ 2 levels "<=50K",">50K": 1 1 2 1 1 2 1 1 1 1 ...
```

```
str(test)
```

```
## 'data.frame':    9696 obs. of  7 variables:
##  $ age           : int  50 49 31 23 40 56 49 22 48 31 ...
##  $ fnlwgt        : int  83311 160187 45781 122272 121772 216851 193366 311512 242406 50787
5 ...
##  $ education_num : int  13 5 14 13 11 13 9 10 7 5 ...
##  $ capital_gain  : int  0 0 14084 0 0 0 0 0 0 0 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: int  13 16 50 30 40 40 40 15 40 43 ...
##  $ income_class  : Factor w/ 2 levels "<=50K",">50K": 1 1 2 1 2 2 1 1 1 1 ...
```

```r
# Evaluate the performance of the model
confusionMatrix(lasso_pred, test$income_class)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction <=50K >50K
##      <=50K  7026 1363
##      >50K    371  936
## 
##                Accuracy : 0.8212
##                  95% CI : (0.8134, 0.8287)
##     No Information Rate : 0.7629
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.4193
## 
##  Mcnemar's Test P-Value : < 2.2e-16
## 
##             Sensitivity : 0.9498
##             Specificity : 0.4071
##          Pos Pred Value : 0.8375
##          Neg Pred Value : 0.7161
##              Prevalence : 0.7629
##          Detection Rate : 0.7246
##    Detection Prevalence : 0.8652
##       Balanced Accuracy : 0.6785
## 
##        'Positive' Class : <=50K
## 
```

```r
# Create a contingency table of income class and native country
cont_table <- table(X.adult$income_class, X.adult$native_country)

# Perform a chi-squared test of independence
chisq_test <- chisq.test(cont_table)
```

```
## Warning in chisq.test(cont_table): Chi-squared approximation may be incorrect
```

```r
# Print the results of the test
print(chisq_test)
```

```
## 
##  Pearson's Chi-squared test
## 
## data:  cont_table
## X-squared = 274.26, df = 10, p-value < 2.2e-16
```

```
# Create a contingency table of income class and native country
cont_table <- table(X.adult$income_class, X.adult$occupation)

# Perform a chi-squared test of independence
chisq_test <- chisq.test(cont_table)
```

```
## Warning in chisq.test(cont_table): Chi-squared approximation may be incorrect
```

```
# Print the results of the test
print(chisq_test)
```

```
##
##  Pearson's Chi-squared test
##
## data:  cont_table
## X-squared = 3812.2, df = 8, p-value < 2.2e-16
```

```
# Subset the data into two groups based on income class
group1 <- X.adult$hours_per_week[X.adult$income == "<=50K"]
group2 <- X.adult$hours_per_week[X.adult$income == ">50K"]

# Perform a t-test
t.test(group1, group2)
```

```
##
##  Welch Two Sample t-test
##
## data:  group1 and group2
## t = -44.901, df = 14494, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.863163 -6.289007
## sample estimates:
## mean of x mean of y
##  38.95615  45.53224
```

```
str(X.adult)
```

```
## 'data.frame':    32319 obs. of  15 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 6 levels "character","Federal-Govt",..: 6 5 5 5 5 5 6 5 5 5
...
##  $ fnlwgt        : int  83311 215646 234721 338409 284582 160187 209642 45781 159449 28046
4 ...
##  $ education     : Factor w/ 4 levels "other","PG","School-Level",..: 4 3 3 4 2 3 3 2 4 4
...
##  $ education_num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital_status: Factor w/ 4 levels "Married","Never-Married",..: 1 3 1 1 1 3 1 2 1 1
...
##  $ occupation    : Factor w/ 9 levels "Admin","Blue-Collar",..: 9 2 2 6 9 8 9 6 9 9 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 1 2 1 6 6 2 1 2 1 1
...
##  $ race          : Factor w/ 5 levels "Amer-Indian",..: 5 5 3 3 5 3 5 5 5 3 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 2 2 ...
##  $ capital_gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native_country: Factor w/ 11 levels "British-Commonwealth",..: 11 11 11 7 11 6 11 11 11
11 ...
##  $ income_class  : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 2 2 2 ...
```

```
# Load the dataset
data <- X.adult

# Split the dataset into training and testing sets
set.seed(123) # for reproducibility
train_index <- sample(1:nrow(data), size = 0.8 * nrow(data), replace = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Fit the model on the training set
model <- lm(capital_gain ~ age + education_num + hours_per_week, data = train_data)

# Make predictions on the testing set
predictions <- predict(model, newdata = test_data)

# Compute the root mean squared error (RMSE)
rmse <- sqrt(mean((test_data$capital_gain - predictions)^2))

# Print the RMSE
rmse
```

```
## [1] 7224.371
```

```r
# Define a new instance
new_instance <- data.frame(age = 35, education_num = 12, hours_per_week = 40)

# Make a prediction for the new instance
prediction <- predict(model, newdata = new_instance)

# Print the prediction
prediction
```

```
##       1
## 1508.43
```