Analysis of Factors Influencing Individual Income Levels

Group 30

```r
# load packages
library(knitr)
library(tidyverse)
library(caret)
library(party)
library(pROC)
library(rpart)
library(rpart.plot)
library(randomForest)
library(randomForestExplainer)
```

```r
# set global theme
theme_set(theme_minimal())
```

```r
# function to save ggplot2 plots
save_fig <- function(filename, ...) {
  ggsave(
    filename = file.path('Plots', paste0(filename, '.png')),
    ...
  )
}
```

1 Introduction

This analysis aims to identify key socio-economic factors that influence whether an individual earns more than $50,000 per year, using data from **U.S. Census Bureau, 1994**. The dataset includes demographic and employment-related variables including **age, education**

level, **marital status, occupation, sex, hours worked per week, and nationality**, with income categorised into two groups: ≤$50k and >$50k per year.

To address this, a **Generalised Linear Model** is applied to evaluate the impact of these factors on income levels. The findings provide insights into the most significant predictors of higher earnings, contributing to a deeper understanding of income distribution patterns.

2 Data Processing

```
# import and clean the data
df <- read.csv("dataset30.csv", na.strings = '?,',
               stringsAsFactors = FALSE) %>%
  drop_na() %>%
  mutate(
    across(everything(), ~ str_remove_all(., ",")),
    Income = Income %>% factor(levels = c('<=50K', '>50K')),
    across(c(Hours_PW, Age), as.numeric),
    across(where(is.character), as.factor))
```

```
# split train and test data
set.seed(1111)
trainIndex <- sample(nrow(df), 0.8 * nrow(df))
train_data <- df[trainIndex,]
test_data <- df[-trainIndex,]
```

This data analysis first involves data cleaning, and transforming. Then, the dataset is split into **80% training set and 20% test set** to ensure data quality and enhance the model's generalization ability.

3 Exploratory Data Analysis

```
# config plot colours
income_colour <- c(
  '<=50K' = 'slateblue',
  '>50K' = 'skyblue')
```

3.1 Numerical Variables

```r
# draw box-plot for numerical variable
num_plot <- function(data, col) {
  ggplot(data, aes(x = Income, y = {{col}}, fill = Income)) +
  geom_boxplot() +
  scale_colour_manual(values = income_colour,
                      aesthetics = 'fill') +
  xlab("Income Levels")
}
```

```r
g <-
  num_plot(df, Age) +
  ylab("Age")
save_fig(
  'age',
  plot = g + ggtitle('Box-Plot of Age vs. Income'))
g
```
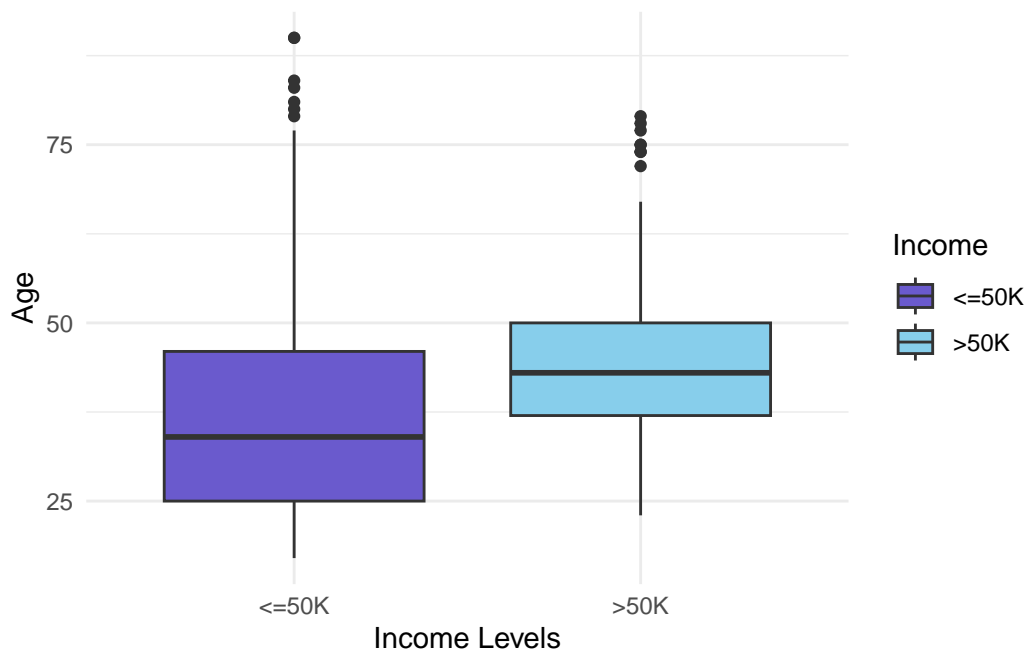

Figure 1: Box-Plot of Age vs. Income

As can be seen from Figure 1, there is a certain positive correlation between age and income:

the median age of the high-income group (income >$50k) is higher than that of the low-income group (income ≤$50k), indicating that older people are more likely to has the higher income.

```r
g <-
  num_plot(df, Hours_PW) +
  ylab("Hours per Week")
save_fig(
  'hour',
  plot = g + ggtitle("Box-Plot of Worked Hours vs. Income"))
g
```
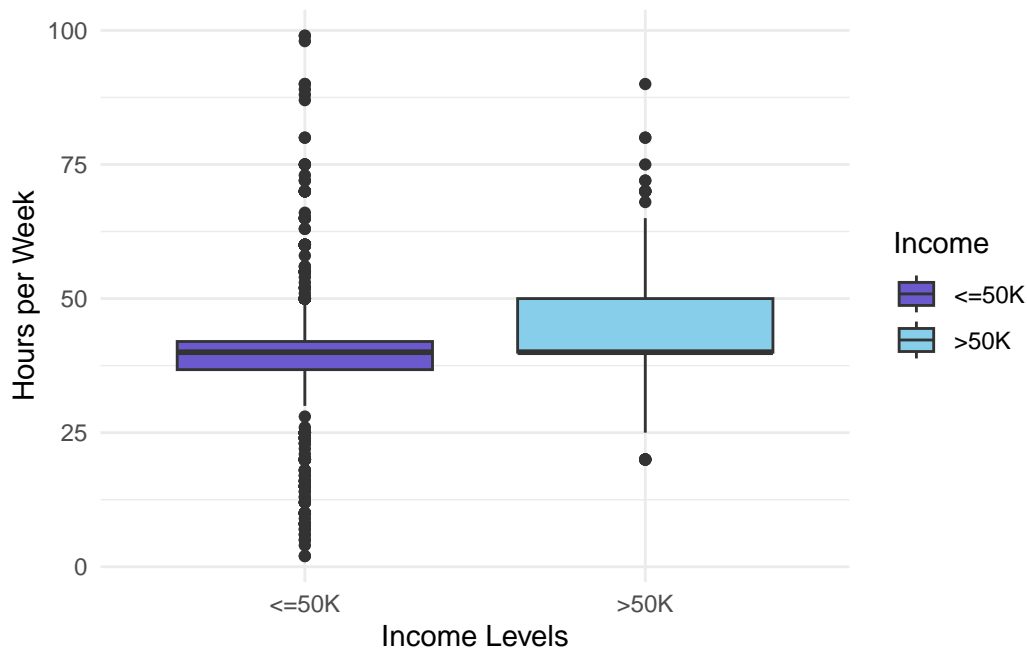

Figure 2: Box-Plot of Worked Hours vs. Income

Working hours have a significant impact on income, higher earners tend to work longer hours, but hours alone do not fully determine income. The working hours of low-income groups are more concentrated, but there are more outliers: individuals with extremely low working hours (<25 hours) and extremely high working hours (>50 hours).

3.2 Categorical Variables

```r
# draw bar-plot for categorical variable
cat_plot <- function(data, col) {
  data %>%
    select({{col}}, Income) %>%

    # calculate count and proportion grouped by variable
    summarise(
      '>50K' = sum(Income == '>50K'),
      '<=50K' = sum(Income == '<=50K'),
      prop = mean(Income == '>50K'),
      .by = {{col}}) %>%

    # transform wide-form to long-form
    pivot_longer(
      cols = c('>50K', '<=50K'),
      names_to = 'Income') %>%

  ggplot(aes(
    # order x-axis by proportion of high-income
    x = reorder({{col}}, prop),
    fill = Income)) +
  geom_col(aes(y = value)) +
  scale_colour_manual(values = income_colour,
                      aesthetics = 'fill') +
  ylab("Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
}
```

```r
g <-
  cat_plot(df, Education) +
  xlab("Education Level")
save_fig(
  'education',
  plot = g + ggtitle("Bar-Plot of Education Level by Income"))
g
```
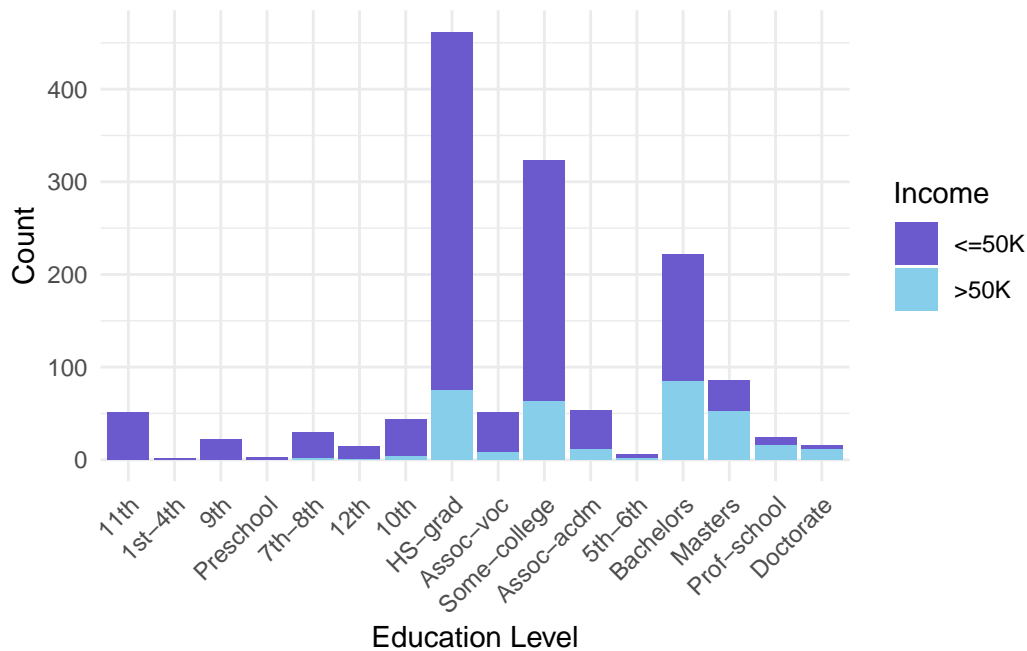
Figure 3: Bar-Plot of Education Level by Income

Education level is positively correlated to income, and higher degree is more likely to earn a high income. In particular, the proportion of high income of master's, doctor's and professional school graduates is significantly higher, while those with low education level are mainly concentrated in low-income groups. This suggests that education level plays an important role in income.

```
g <-
  cat_plot(df, Marital_Status) +
  xlab("Marital Status")
save_fig(
  'marital status',
  plot = g + ggtitle("Bar-Plot of Marital Status by Income"))
g
```
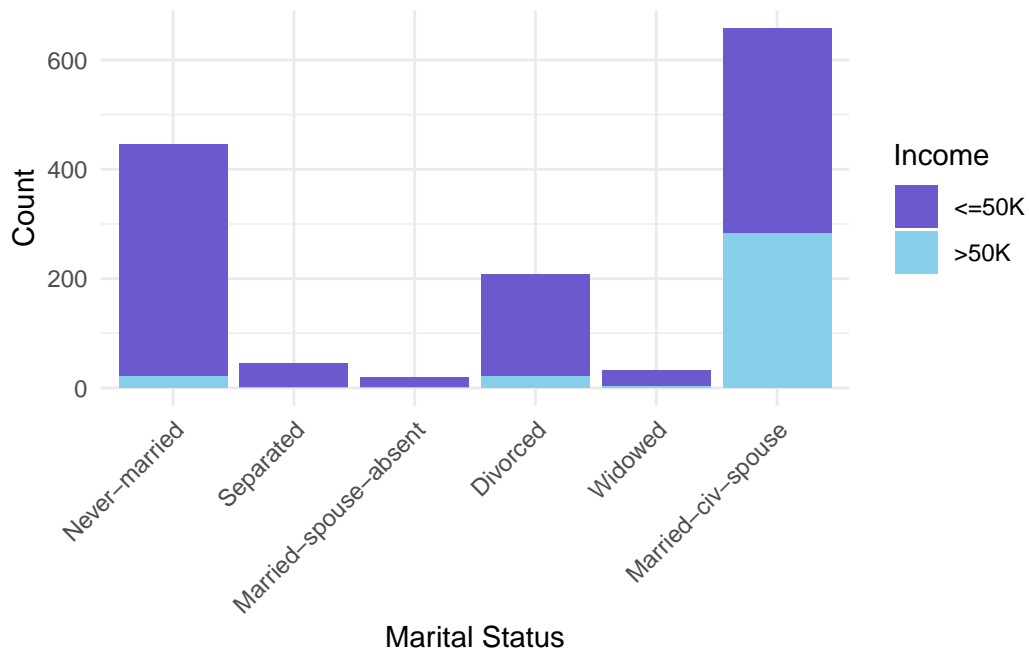
Figure 4: Bar-Plot of Marital Status by Income

Figure 4 shows that people married with civil spouse have higher income than other group.

```
g <-
  cat_plot(df, Occupation) +
  xlab("Occupation")
save_fig(
  'occupation',
  plot = g + ggtitle("Bar-Plot of Occupation by Income"))
g
```
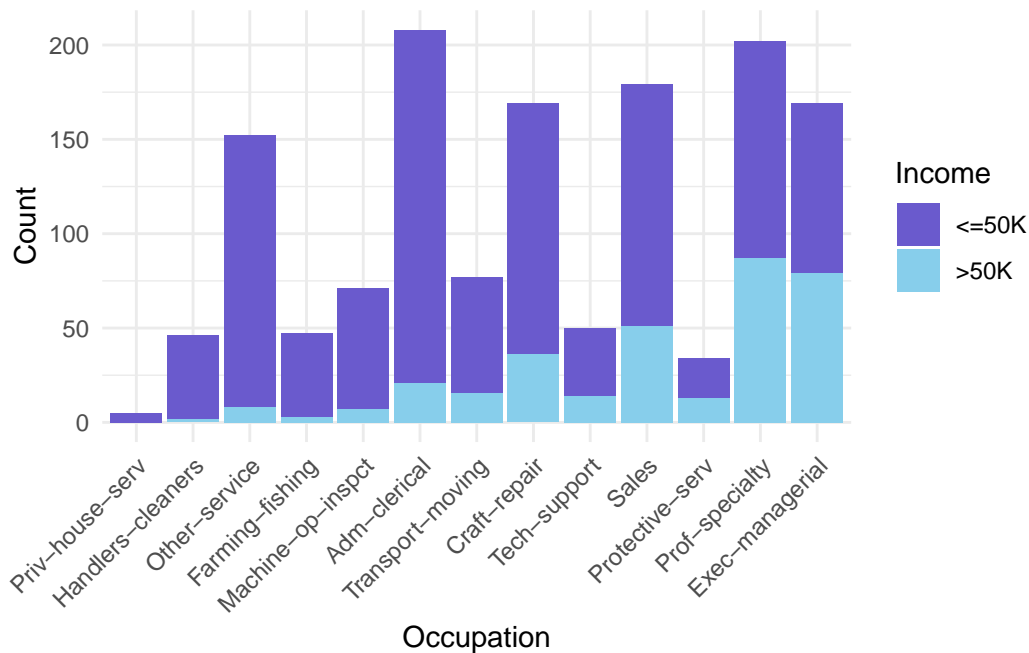
Figure 5: Bar-Plot of Occupation by Income

Figure 5 shows clear income differences between different occupational categories. A higher percentage of "Exec-managerial" and "Prof-specialty" occupations make more than 50,000, indicating that these occupations are more likely to generate higher income. By contrast, the proportion of "Priv-house-serv" and "Handlers-cleaners" is significantly lower.

```
g <-
  cat_plot(df, Sex) +
  xlab("Gender")
save_fig(
  'gender',
  plot = g + ggtitle("Bar-Plot of Gender by Income"))
g
```
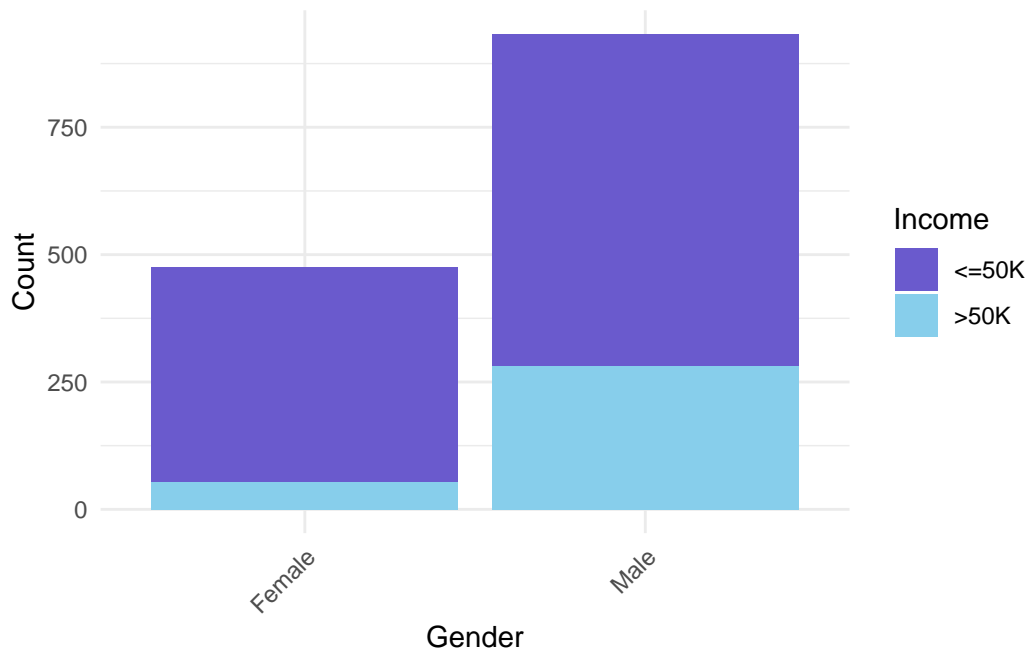
Figure 6: Bar-Plot of Gender by Income

There are significant differences in income distribution between men and women. The proportion of men earning more than $50,000 is significantly higher than that of women, while the women income less than $50,000 is relatively higher.

```
g <-
  cat_plot(df, Nationality) +
  xlab("Nationality")
save_fig(
  'nationality',
  plot = g + ggtitle("Bar-Plot of Nationality by Income"))
g
```
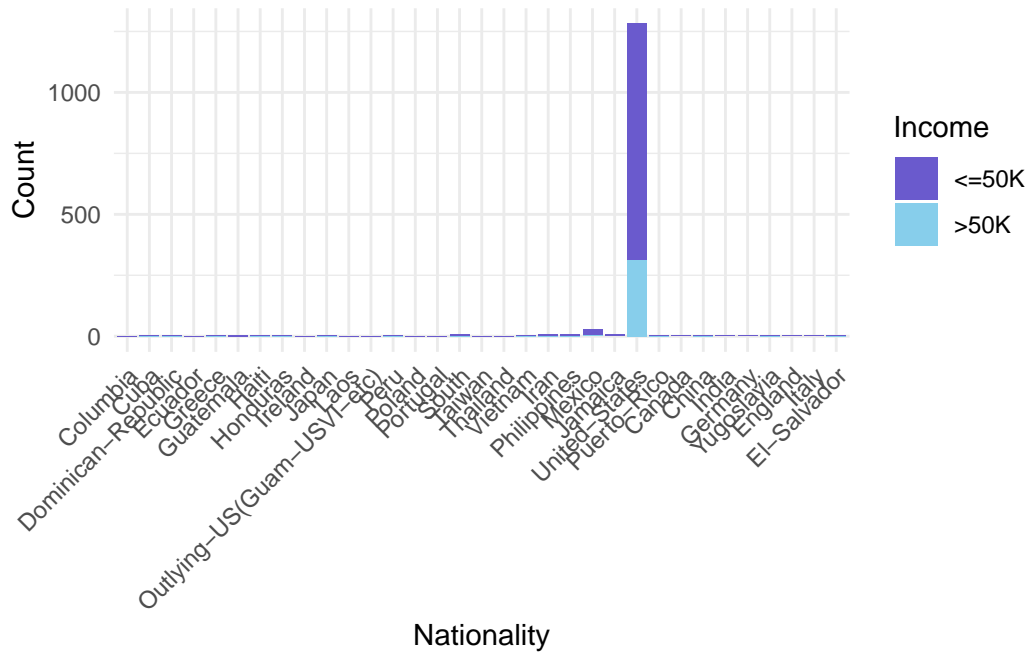
Figure 7: Bar-Plot of Nationality by Income

```
df %>%
  # calculate number of observations in each group
  summarise(
    Count = n(),
    .by = Nationality
  ) %>%
  # calculate proportion
  mutate(
    'Proportion in Observations (%)' = Count / sum(Count) * 100
  ) %>%
  top_n(3, Count) %>%
  kable(digits = 2)
```

Table 1: Nationality Summary

Nationality	Count	Proportion in Observations (%)
United-States	1281	90.92
Mexico	27	1.92
Jamaica	9	0.64

These summaries show that the sample size from groups except United States is quite small,

which may adversely impact model accuracy if nationality is used as an explanatory variable.

4 Formal Data Analysis

4.1 Generalised Linear Model

```
# Fit a full logistic regression model using all predictors
full_model <- glm(Income ~ ., data = train_data, family = binomial)
```

The **logistic regression** model is built to predict income level (**Income**) by considering various socio-economic factors, including **age, education, marital status, occupation, sex,** and **hours worked per week**.

```
# Perform stepwise regression to simplify the model,
# starting from the full model and removing non-significant predictors
stepwise_model <- step(full_model, direction = "backward")
```

To further refine the model, **stepwise regression** is applied using **backward elimination**, which systematically removes variables that do **not significantly** contribute to the prediction based on the **Akaike Information Criterion (AIC)**.

The reduced model is given by formula Income ~ Age + Education + Marital_Status + Occupation + Hours_PW.

The initial model's **AIC** is 906.71, while the optimized model's **AIC** was reduced to 877.85, indicating that the model becomes more concise while maintaining a good fit. As a result, this approach not only simplifies the model, but also helps to reduce over-fitting and to enhance interpretability by retaining only the most significant predictors.

```
# compute the accuracy, and confusion matrix
test_predictions <-
  predict(stepwise_model, newdata = test_data, type = "response")
test_predicted_classes <- ifelse(test_predictions > 0.5, '>50K', '<=50K')
conf_matrix <-
  confusionMatrix(factor(test_predicted_classes), test_data$Income)
conf_matrix$table %>% kable()
```

Table 2: Confusion Matrix of GLM

	<=50K	>50K
<=50K	202	31
>50K	12	37

```
# draw the curve of roc and compute the auc
roc_curve_test <- roc(test_data$Income, test_predictions)
plot(roc_curve_test, col = "red",
     xlim = c(1, 0), ylim = c(0, 1), asp = 1)
```
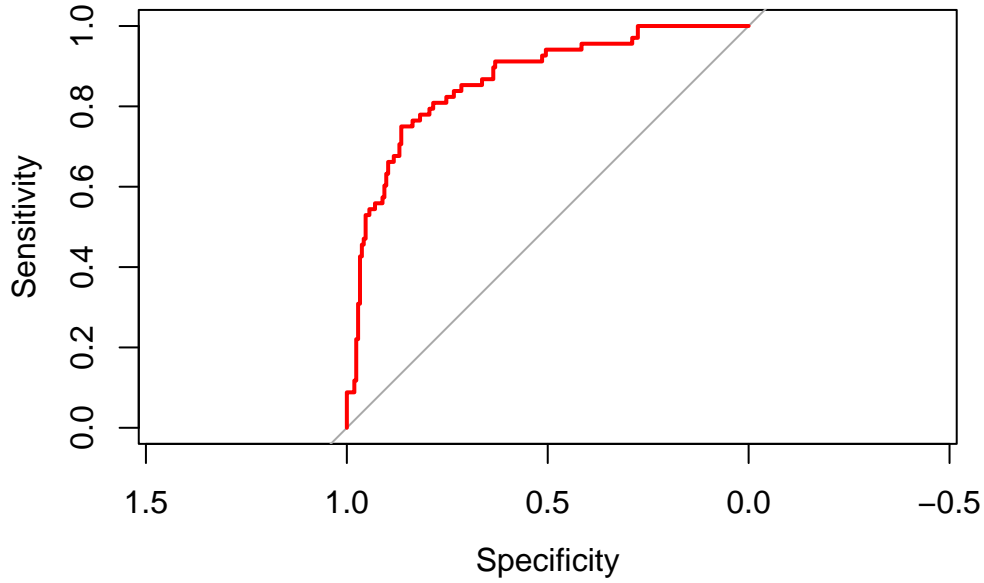

Figure 8: ROC Curve for Test Data

The evaluation results of this logistic regression model on the test set indicate a high classification capability, with an **accuracy** of 84.75% within the **95% confidence interval** (80.02% - 88.74%), demonstrating stable predictive performance. The confusion matrix further shows that the model has a strong ability to identify low-income individuals, achieving a **sensitivity** of 94.39%, meaning that most low-income individuals are correctly classified. However, the **specificity** is only 54.41%, suggesting that the model has some difficulty in correctly identifying high-income individuals, with a considerable number being misclassified as low-income. In terms of overall classification performance, the **AUC value** at 0.864 indicates that the model performs well in distinguishing between high-income and low-income individuals, and the **ROC curve** demonstrates strong discriminatory power.

```
# save model performance for comparison
models_sum <-
  data.frame(GLM = c(
    ACC = conf_matrix$overall[["Accuracy"]],
    Sensitivity = conf_matrix$byClass[["Sensitivity"]],
    Specificity = conf_matrix$byClass[["Specificity"]],
    AUC = auc(roc_curve_test)
  ))
```

4.2 Random Forest

```
#reset the seed and reset the data
df = read.csv("dataset30.csv")
set.seed(1111)

#clear the data

df = df %>%
  mutate(across(everything(), ~str_remove_all(., ","))) %>%
  mutate(Hours_PW = as.numeric(Hours_PW),
         Age = as.numeric(Age))
df = df %>% filter(Occupation != "?" & Nationality != "?")
df = df %>% select(Age, Education, Marital_Status, Occupation, Sex, Hours_PW, Income)
df = df %>% mutate(across(where(is.character), as.factor))
x = df[,1:6]
y = df[,7]
#test train splite

x_pre = x[1:1228,]
x_te = x[1229:1409,]
y_pre = y[1:1228]
y_te = y[1229:1409]
#predict the RF model and set the confusionmatrics
pre = cbind(y_pre,x_pre)
rf_model = randomForest(y_pre~.,data = pre,ntree = 100,importance = T)
predictions <- predict(rf_model, newdata = x_te)
tb = table(predictions,y_te)
TP = tb[1,1]
FP = tb[1,2]
FN = tb[2,1]
```

```
TN = tb[2,2]
tb %>% kable()
```

Table 3: Confusion Matrix of Random Forest

	<=50K	>50K
<=50K	124	19
>50K	12	26

After using the method of Randomforest , we do not need to do any judgement on the value of predictions ,the randomforest automatic done the **decision tree** for the data. The table shows that most of the value full in the range TP and FP, which mean most of the data fit well.

```
#the acc is use to text the ggod fit of the model and the TPR and TNR is for the sensitivity
acc = (TP + TN)/(TP + FP + FN + TN)
TPR = TP/(TP + FN)
TNR = TN/(FP + TN)
print(acc)
```

```
[1] 0.8287293
```

```
print(TPR)
```

```
[1] 0.9117647
```

```
print(TNR)
```

```
[1] 0.5777778
```

The accuracy of randomforest is up tp **82.87%** which shows that it give lots of information to the predictors, and th achieving a **sensitivity of 86.71%** and **specificity of 68.42%** ,which mean it have a low ability in predict the negative size of value which is the income >50%.

```
#the plot of ROC which show the good fit for the POS. side.
pred_probs = predict(rf_model,newdata = x_te, type = "prob")[,2]
roc_c = roc(y_te,pred_probs)
plot(roc_c,col = "red")
```
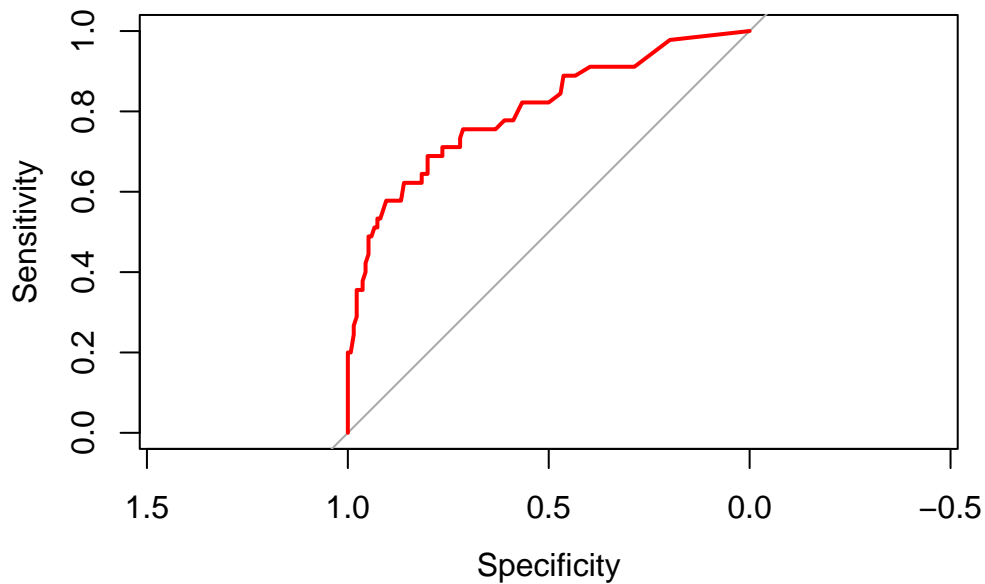
Figure 9: ROC Curve for Test Data

```
auc(roc_c)
```

```
Area under the curve: 0.8018
```

using the **ROC and AUC** to detect the balance of the model ,from the plot of ROC we see that the curve are far away from the straight line ,which can also reflect by the number of AUC is **0.8018**, that means the model show more information in the True Positive range than False Negative range ,so it give a good fit to the data.

```
models_sum <- models_sum %>% cbind(RF = c(
  acc, TPR, TNR, auc(roc_c)
))
```

```
# plot the rf dicsiontree
# more minimal depth have high weight in predict
plot_min_depth_distribution(rf_model)
```
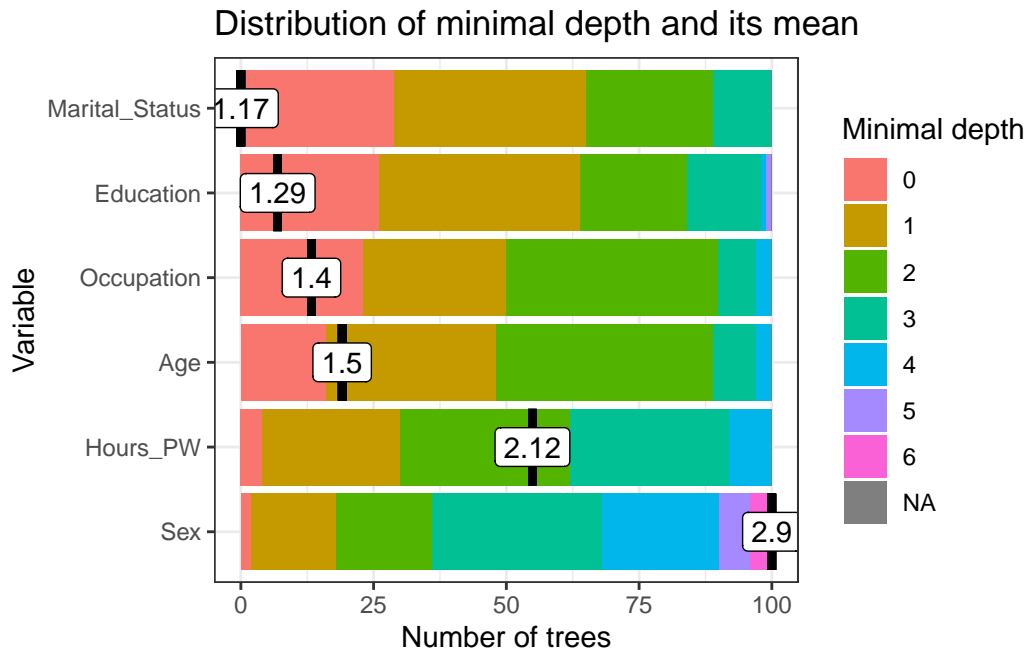
Figure 10: Minimal Depth of Random Forest

This is the plot of **minimal depth** ,the plot give outlay the first time of the "tree branch" split by each variable ,the closer the split from the root the more influence it comes .So form the plot **Marital_Status** give the most information to the model ,then comes the **Education**. However that didn't mean that the other variable didn't appeard on the plot don't make any influence ,we still need to consider the times that it split for each variable.

```
rpart_tree <- rpart(y_pre ~ ., data=pre)
# plot(rpart_tree,compress = TRUE)
# text(rpart_tree, use.n = TRUE,cex = 0.5)
rpart.plot(rpart_tree,type = 1,box.palette = "light green")
```
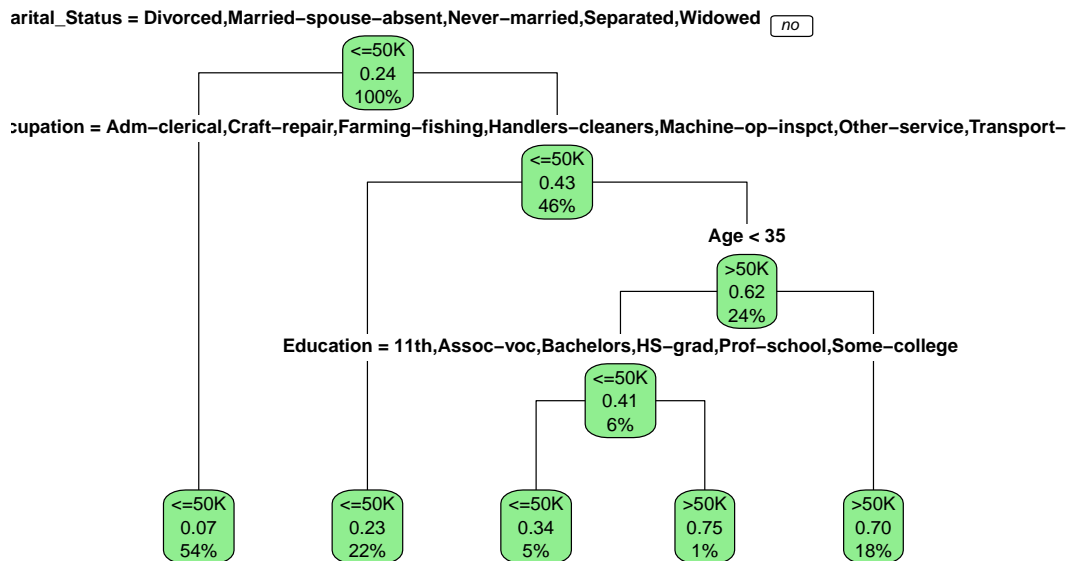
Figure 11: Random Forest Decision Tree

The decision tree plot is the plot that give the most obvious information to the model. We can see clearly from each branch that how the data are alienation to different slop. From this data we can see that there are no such much split in the plot maybe cause by the small amount of data or there are no strong influence with most of the variable.

4.2.1 The Resample for Model

```
#the resample part using K-fold.stable the VAR
K = 5
set.seed(1111)
folds = cut(1:1409, breaks=K, labels=FALSE)
sen = sep = acc =numeric(K)
for(k in 1:K){
  x.train = x[which(folds!=k),]
  x.text = x[which(folds==k),]
  y.train = y[which(folds!=k)]
  y.text = y[which(folds==k)]
  pre = cbind(y.train,x.train)
  rf_fit = randomForest(y.train~.,data = pre,ntree = 100,importance = T)
  predictions = predict(rf_fit,newdata =x.text)
  tb = confusionMatrix(y.text,predictions)
  tb.class = tb$byClass
```

17

```
  tb.overall = tb$overall
  sen[k] = tb.class[1]
  sep[k] = tb.class[2]
  acc[k] = tb.overall[1]
}
# the mean of 5 time predict-text outcome
sen_m = sum(sen)/5
sep_m = sum(sep)/5
acc_m =sum(as.numeric(acc))/5
print(sen_m)
```

```
[1] 0.8622353
```

```
print(sep_m)
```

```
[1] 0.6302956
```

```
print(acc_m)
```

```
[1] 0.8140506
```

the re-sample are usually used to do repeat experiment ,this is for helping the model to give a more precise ACC and any other data. The method shows here is the **K-fold** for k equal to 5 . which mean split the hole data into 5 part and set each part as test data . which will give 5 different outcome ,to make the outcome useful ,we take **mean** for these outcome (thus ACC:**81.4%**),this would be a more representative outcome then the acc before.

4.3 Comparative Analysis between the GLM and Random Forest

From the data we get above we can have a compare to two model.

```
models_sum %>% kable(digits = 4)
```

Table 4: Summary for Models

	GLM	RF
ACC	0.8475	0.8287

Table 4: Summary for Models

	GLM	RF
Sensitivity	0.9439	0.9118
Specificity	0.5441	0.5778
AUC	0.8640	0.8018

Table 4 shows that two model are giving the near ACC and AUC ,and the GLM is higher. The RF model is higher in Specificity and lower in Sensitivity, which indicates that it have a better detection on the Negative side but less in Positive side. It can be seen that GLM are giving more information to the data, but maybe after changing the depth of Random Forest ,the fitting rate of it would be higher than the GLM. So we can say that both of the model can be used in this data fitting, but GLM is a better one.

5 Conclusions

This study utilized data from the **1994 U.S. Census** to analyze key socioeconomic factors influencing individual income levels. The results indicate that **education level, occupation, marital status, and age** are crucial determinants of income, with higher education and specialized professions significantly increasing earning potential. Additionally, **sex**, as men are more likely to earn higher incomes. Furthermore, **weekly working hours** have a certain impact on income, though with diminishing marginal returns.