Feature Engineering of Insurance Fraudulence

BlueSpace – Jingquan Miao, Yufeng Song, Zhihuan Hao

Abstract

In this project, we aimed to identify predictor variables that can facilitate building a model that is used to detect fraudulence of first-party physical damage.Our group used data data visualization, hypothesis testing and exploratory data analysis to conduct feature selection for fraud detection of first-party physical damage. Then we fitted the logistic regression model to investigate the effect of individual features on the probability of insurance fraudulence. Throughout the process of feature selection, we found that high_education_ind, past_num_of_claims, witness_present_ind, marital_status, address_change_ind, living_status, gender are the statistically meaningful explanatory variables that contribute to the predictive model of first-party physical damage fraudulence.

Introduction

Because people are generally risk averse, insurance companies exist as a safeguard for preventing people from suffering sudden huge losses from different aspects such as medical emergencies and vehicle accidents. However, as insurance evolves as such a type of business, insurance companies may also face the risks of insurance fraud from fake claims. Failure of fraud detection brings huge financial losses to insurance companies and indirectly brings harm to clients who are having valid claims. Therefore, under such circumstances, an efficient modeling for predicting and detecting insurance fraud becomes necessary. The purpose of this project is to find factors most strongly associated with fraudulent first-party physical damage claims.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   1.0.1
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.3.0      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 1.0.0
```

```
## Warning: 程辑包'forcats'是用R版本4.2.3 来建造的
```

```
## ── Conflicts ─────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(twosamples)
```

```
## Warning: 程辑包'twosamples'是用R版本4.2.3 来建造的
```

```
library(vcd)
```

```
## Warning: 程辑包'vcd'是用R版本4.2.3 来建造的
```

```
## 载入需要的程辑包：grid
```

```
library(ggmosaic)
```

```
## Warning: 程辑包'ggmosaic'是用R版本4.2.3 来建造的
```

```
##
## 载入程辑包：'ggmosaic'
##
## The following objects are masked from 'package:vcd':
##
##     mosaic, spine
```

```
library(lubridate)
```

```
##
## 载入程辑包：'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zipcodeR)
```

```
## Warning: 程辑包'zipcodeR'是用R版本4.2.3 来建造的
```

```
library(ggpubr)
```

```
## Warning: 程辑包'ggpubr'是用R版本4.2.3 来建造的
```

```
library(corrplot)
```

```
## Warning: 程辑包'corrplot'是用R版本4.2.3 来建造的
```

```
## corrplot 0.92 loaded
```

```
library(ggplot2)
library(psych)
```

```
## Warning: 程辑包'psych'是用R版本4.2.3 来建造的
```

```
##
## 载入程辑包：'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
train <- read_csv("C:/Users/郝治寰/Desktop/MA415/MA415 Project Blue Space/train_2023.csv") %>%
select(-claim_number)
```

```
## Rows: 19000 Columns: 25
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (8): gender, living_status, claim_date, claim_day_of_week, accident_sit...
## dbl (17): claim_number, age_of_driver, marital_status, safty_rating, annual_...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Data and Methods

The dataset used in this study was provided by 2023 Travelers NESS Statathon Claim fraud detection containing referred first-party physical damage claims from 2015 to 2016. The dataset has 19, 000 rows and 25 columns. Since variable “claim_number” refers to claim ID which cannot be used in building predictive models of fraud detection, which leaves us with 23 explanatory variables and 1 response variable. Among the 23 predictors, there are 9 numerical variables and 14 categorical variables. To gain insights into the potential relationships between the individual predictor variables and the response variable fraud, which is a binary variable (0=no fraud, 1=fraud) we utilized data visualization techniques as an initial step in our statistical analysis. To visualize fraud against numerical variables, we utilized various graphical methods including histogram, boxplot, violin plot, frequency polygons, and density plot. For fraud against categorical variables, we relied on bar graphs to identify any variation in the proportion of fraud across different levels within each categorical variable.

To determine whether there were statistically significant differences in the distributions of fraud and no fraud groups across levels within each predictor variable, we used various statistical tests, and we used alpha level equals to 0.05 for all tests. For numerical variables, we relied on chi-squared tests for multiple proportions and the Cramer-von Mises test (CvM) and the Anderson-Darling test (AD) to compare the distributions of fraud and no fraud within predefined subgroups. However, many visualizations of fraud against individual numerical predictors showed that there are differences in the tails. CvM places more emphasis on the tails of the distributions and is less sensitive to differences in the middle of the distribution, and AD is a modified version of the CvM test that gives more weight to the tails of the distribution. The results of these two tests are used to corroborate each other’s conclusion. For example, we suspected that certain stage of life may correlate with fraudulence, so we divide the numerical variable “age_of_driver” into 10 groups, and test the equality of the proportion of fraud among different age groups using Chi-squared Test for multiple proportions. For numerical variables with a wide range of values, for example, “safty_rating”, CvM and the AD were used to assess whether the distribution of safety ratings differed significantly between drivers who committed fraud and those who did not. For bivariate analysis, we first tried to find the relationship between the explanatory variables selected by the univariate analysis. We used chi-squared tests for categorical variables vs categorical variables; anova test for continuous variables vs categorical variables; logistic regression for categorical variables vs continuous variables and linear regression for continuous variables vs continuous variables.

Secondly, to better visualize all the differences and accuracy, we used Pearson correlation and Kendall correlation to build two correlation matrices with respect to the mix of numeric and categorical variables and only between categorical variables. The Pearson correlation is precise for correlations between continuous and categorical variables. The Kendall correlation fits for only categorical variables.

To avoid multicollinearity, we also built a linear regression between the highly correlated variables we found within the explanatory variables. We also made a plot by using ggplot function and group_by to visualize the highly correlated variables. Then, we used those variables that are correlated to responsive variables to fit the final logistic model.

For the logistic model interpretation, we made several plots to better understand how the features selected would have an effect on the responsive variable—fraud. The visualization is done by using dataframe manipulations and pivot wider to combine several categorical variables and group by to plot the proportion of fraud among each specific group so that the specific proportion will provide patterns. There are also mosaic plots, boxplot and heatmap, to better visualize and interpret explanatory variables with respect to fraud.

After detailed inter-predictor analysis of the key features identified by the logistic regression model, we further validate those features' importance by utilizing a rigorous statistical technique called Akaike Information Criterion (AIC) Model Comparison. This method enabled us to assess feature importance and compare different models to identify the most suitable combination of variables to explain the observed patterns in insurance fraud.

Results

We conduct a univariate analysis of all 23 features by first visualizing the relationship between each predictor variable and response variable `fraud` and then performing appropriate statistical tests for each feature to scrutinize its statistical significance numerically using test statistics and p-values. The set of variables below significantly influences `fraud`:

```
train_bivar <- train
train_miao_aic1 <- train
train <- train %>% transform(
  age_of_driver=as.integer(age_of_driver),
  gender=as.factor(gender),
  marital_status=as.factor(marital_status),
  safty_rating=as.integer(safty_rating),
  annual_income=as.integer(annual_income),
  high_education_ind=as.factor(high_education_ind),
  address_change_ind=as.factor(address_change_ind),
  living_status=as.factor(living_status),
  zip_code=as.factor(zip_code),
  claim_date=as.factor(claim_date),
  claim_day_of_week=as.factor(claim_day_of_week),
  accident_site=as.factor(accident_site),
  past_num_of_claims=as.factor(past_num_of_claims),
  witness_present_ind=as.factor(witness_present_ind),
  liab_prct=as.integer(liab_prct),
  channel=as.factor(channel),
  policy_report_filed_ind=as.factor(policy_report_filed_ind),
  claim_est_payout=as.double(claim_est_payout),
  age_of_vehicle=as.integer(age_of_vehicle),
  vehicle_category=as.factor(vehicle_category),
  vehicle_price=as.double(vehicle_price),
  vehicle_color=as.factor(vehicle_color),
  vehicle_weight=as.double(vehicle_weight),
  fraud=as.integer(fraud)
)
```

```
Name <- c("age_of_driver","safty_rating","annual_income","past_num_of_claims","liab_prct","claim_est_payout","age_of_vehicle","vehicle_price","vehicle_weight","claim_number","gender","marital_status","high_education_ind","address_change_ind","living_status","zip_code","claim_date","claim_day_of_week","accident_site","witness_present_ind","channel","policy_report_filed_ind","vehicle_category","vehicle_color","fraud")

Type <- c(rep("Numerical Predictor",9), rep("Categorical Predictor",15),"Response Variable")

Description <- c("Age of driver","Safety rating index of driver","Annual income of driver", "Number of claims the driver reported in past 5 years","Liability percentage of the claim"," Estimated claim payout","Age of first party vehicle","Price of first party vehicle","Weight of first party vehicle","Claim ID (cannot be used in model)","Gender of driver","Marital status of driver","Driver' s high education index","Whether or not the driver changed living address in past 1 year","Driver' s living status, own or rent","Driver' s living address zipcode","Date of first notice of claim","Day of week of first notice of claim","Accident location, highway, parking lot or local","Witness indicator of the claim","The channel of purchasing policy","Policy report filed indicator","Category of first party vehicle","Color of first party vehicle","Fraud indicator (0=no, 1=yes)")
```

```
variable_key <- data.frame(Name,Type,Description)
variable_key
```

```
##                      Name                 Type
## 1            age_of_driver   Numerical Predictor
## 2             safty_rating   Numerical Predictor
## 3            annual_income   Numerical Predictor
## 4       past_num_of_claims   Numerical Predictor
## 5                liab_prct   Numerical Predictor
## 6          claim_est_payout   Numerical Predictor
## 7            age_of_vehicle   Numerical Predictor
## 8             vehicle_price   Numerical Predictor
## 9            vehicle_weight   Numerical Predictor
## 10            claim_number Categorical Predictor
## 11                  gender Categorical Predictor
## 12          marital_status Categorical Predictor
## 13      high_education_ind Categorical Predictor
## 14      address_change_ind Categorical Predictor
## 15           living_status Categorical Predictor
## 16                zip_code Categorical Predictor
## 17              claim_date Categorical Predictor
## 18       claim_day_of_week Categorical Predictor
## 19           accident_site Categorical Predictor
## 20     witness_present_ind Categorical Predictor
## 21                 channel Categorical Predictor
## 22 policy_report_filed_ind Categorical Predictor
## 23        vehicle_category Categorical Predictor
## 24           vehicle_color Categorical Predictor
## 25                   fraud     Response Variable
##                                                        Description
## 1                                                      Age of driver
## 2                                   Safety rating index of driver
## 3                                         Annual income of driver
## 4             Number of claims the driver reported in past 5 years
## 5                                  Liability percentage of the claim
## 6                                              Estimated claim payout
## 7                                    Age of first party vehicle
## 8                                  Price of first party vehicle
## 9                                  Weight of first party vehicle
## 10                              Claim ID (cannot be used in model)
## 11                                                Gender of driver
## 12                                    Marital status of driver
## 13                                 Driver’ s high education index
## 14 Whether or not the driver changed living address in past 1 year
## 15                              Driver’ s living status, own or rent
## 16                                  Driver’ s living address zipcode
## 17                                    Date of first notice of claim
## 18                               Day of week of first notice of claim
## 19               Accident location, highway, parking lot or local
## 20                                  Witness indicator of the claim
## 21                                The channel of purchasing policy
## 22                                   Policy report filed indicator
## 23                                 Category of first party vehicle
## 24                                    Color of first party vehicle
## 25                                   Fraud indicator (0=no, 1=yes)
```

```
## fill missing values
train <- train[complete.cases(train), ]
train_yufeng <- train
```

gender: Females are more likely to make fraudulent claims than males. A two-tailed two sample z-test for proportions is used to test the equality of fraud proportions for females and males. The p-value=8.974e-10 (< 0.05), we have evidence to suggest that the proportion of fraud is statistically greater for females than for males.

marital_status: Single individuals are more likely to make fraudulent claims than married people. A two-sample z-test for proportions is used to test the equality of these two proportions. The p-value 2.2e-16 (< 0.05), we have evidence to suggest that the proportion of fraud cases is greater for unmarried people than for married people.

annual_income: Claimants of non-fraudulent cases have a more diverse range of income levels, while those committing fraudulent claims have annual incomes concentrated around $37,500. Since the test statistic for CvM is 86.45077 and the P-Value is 0.00025, we have evidence to suggest that the distributions of annual_income for fraud and no fraud are statistically significantly different.

high_education_ind: People who received high education are less likely to commit insurance fraudulence than people who are comparably less educated.A right-tailed two sample z-test for proportions is used to test the equality of p-value < 2.2e-16 (< 0.05), we have evidence to conclude that the proportion of fraud cases is statistically greater for people did not receive high education than for people who had.

address_change_ind: People who have recently changed their residence address are more prone to fabricate an insurance claim. A left-tailed two sample z-test for proportions is used to test the equality of fraud proportions for drivers who changed addresses and drivers who did not. The p-value < 2.2e-16 (< 0.05), we have evidence to suggest that the proportion of fraud cases is statistically greater for people who changed addresses than people who didn't.

living_status: People living in a rented household are more prone to stage a claim. A left-tailed two sample z-test for proportions is used to test the equality of fraud proportions for drivers who live in rented places and drivers who live in owned places. The p-value =1.025e-05 (< 0.05), we have evidence to suggest that the proportion of fraud cases for people renting a place is statistically higher than people who own a place.

witness_present_ind: The presence of a witness during an incident of vehicle damage appears to reduce the likelihood of fraud, as claims with a witness present were found to be less likely to be fraudulent. A right-tailed two sample z-test is used to test the equality of fraud proportions for drivers with and without witnesses. The p-value < 2.2e-1 (< 0.05), we have evidence to suggest that the proportions of fraud cases for no witness present is statistically higher than those with witness present.

liab_prct: Claims with liability percentages below 25%, around 50%, or above 75% were less likely to be fraudulent, while those with percentages between 25 to 50% or 50 to 75% were more likely to be fraudulent. Since the test statistic for CvM is 5.425545 and P-Value is 0.010000, we have evidence to conclude that the distributions of liab_pct for fraud and no fraud are statistically significantly different.

policy_report_filed_ind: If the policy report is filed, then the claimants are more likely to commit fraudulent claims. A left-tailed two sample z-test for proportions was used to test the equality of fraud proportions for people who filed and did not file the policy report. The p-value = 0.001284(< 0.05), we have evidence to suggest that the proportions of fraud cases for people who did not file the report is less than those who filed the report.

claim_est_payout: The estimated payout fraudulent cases have more variability between the first and third quartiles and fewer outliers than that non-fraudulent ones. Since the test statistic for CvM is 38.35846 and P-value is 0.00025, we have evidence to conclude that the distributions of claimed estimated payout for fraud and no fraud are statistically significantly different.

age_of_vehicle: The age of vehicle for both fraud and non-fraud groups has similar bell-shaped distributions, while that of the non-fraud group is slightly right-skewed, which indicates that non-fraud groups' vehicles have lower ages. Since the test statistic for CvM is 24.55885 and the P-Value is 0.00025, we have evidence to suggest that the distributions of age_of_vehicle for fraud and no fraud are statistically significantly different.

past_num_of_claims: As the number of past claims increased from 0 to 5 times, the likelihood of fraudulent claims increased but dropped when the number of claims reached 6. The chi-square test for multiple proportions is used to test the equality of fraud proportions for drivers with past number of claims ranging from 0 to 6. The p-value < 2.2e-16 (< 0.05), we have evidence to suggest that the proportion of fraud cases for drivers with different claims are statistically significantly different.

Our univariate analysis of the insurance fraud dataset has provided valuable insights into the relationship between various demographic and situational factors and the likelihood of fraudulent claims.

```
if (!is.factor(train$fraud)) {
  train$fraud <- as.factor(train$fraud)
  train$fraud <- factor(train$fraud, levels = c(0, 1), labels = c("No Fraud", "Fraud"))
}

# 1. fraud against age - Done
age_uni <- train %>%
  mutate(age_group = cut(age_of_driver, breaks = seq(0, 100, 10),
                         include.lowest = TRUE)) %>%
  ggplot(aes(age_group, fill = fraud)) +
  geom_histogram(position = "fill", stat="count") +
  scale_x_discrete(labels = c("0-10", "11-20", "21-30", "31-40", "41-50", "51-60", "61-70", "71
-80", "81-90", "91-100")) +
  scale_fill_discrete(name="Fraud") +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))
```

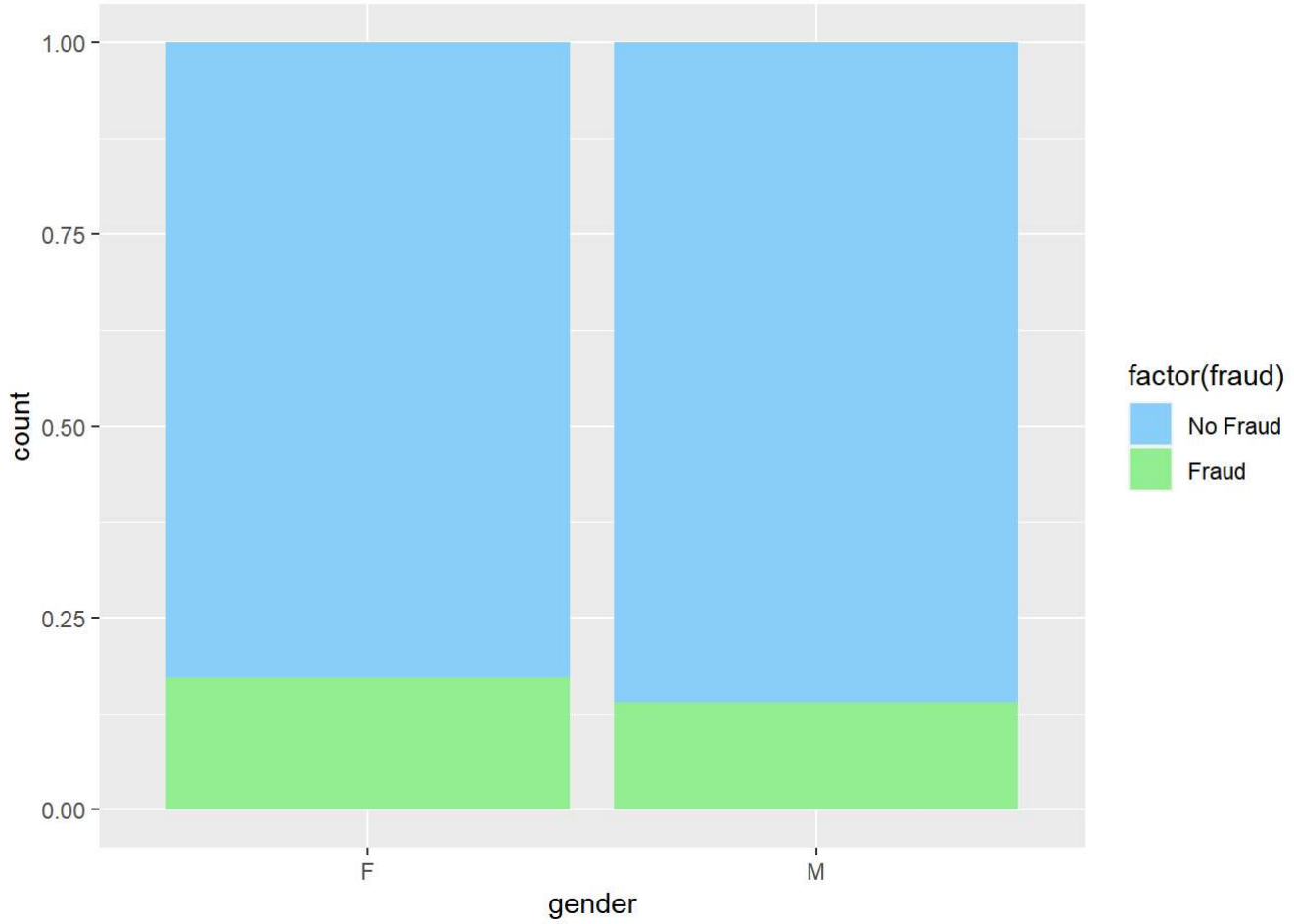
```
## Warning in geom_histogram(position = "fill", stat = "count"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
# 2. fraud against gender - Done
gender_uni <- train %>% ggplot(aes(gender, fill = factor(fraud))) +
  geom_bar(position = "fill") +
  scale_fill_discrete(name = "Fraud") +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

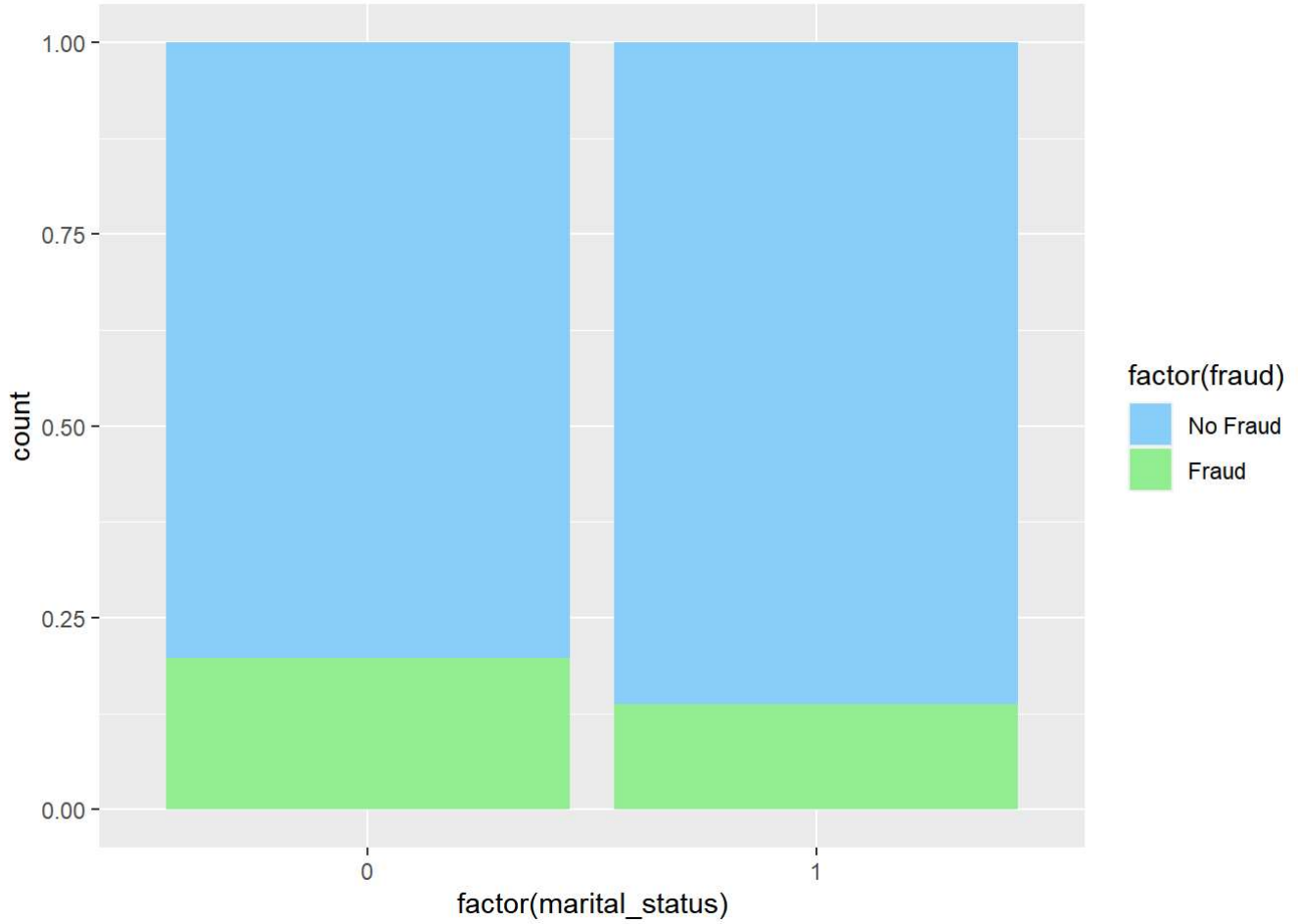
```
gender_uni
```




```
# 3. fraud against marital_status - Done
ms_uni <- train %>%
  filter(!is.na(marital_status)) %>%
  ggplot(aes(x = factor(marital_status), fill = factor(fraud))) +
  geom_bar(position = "fill")+
  scale_fill_discrete(name="Fraud")+
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

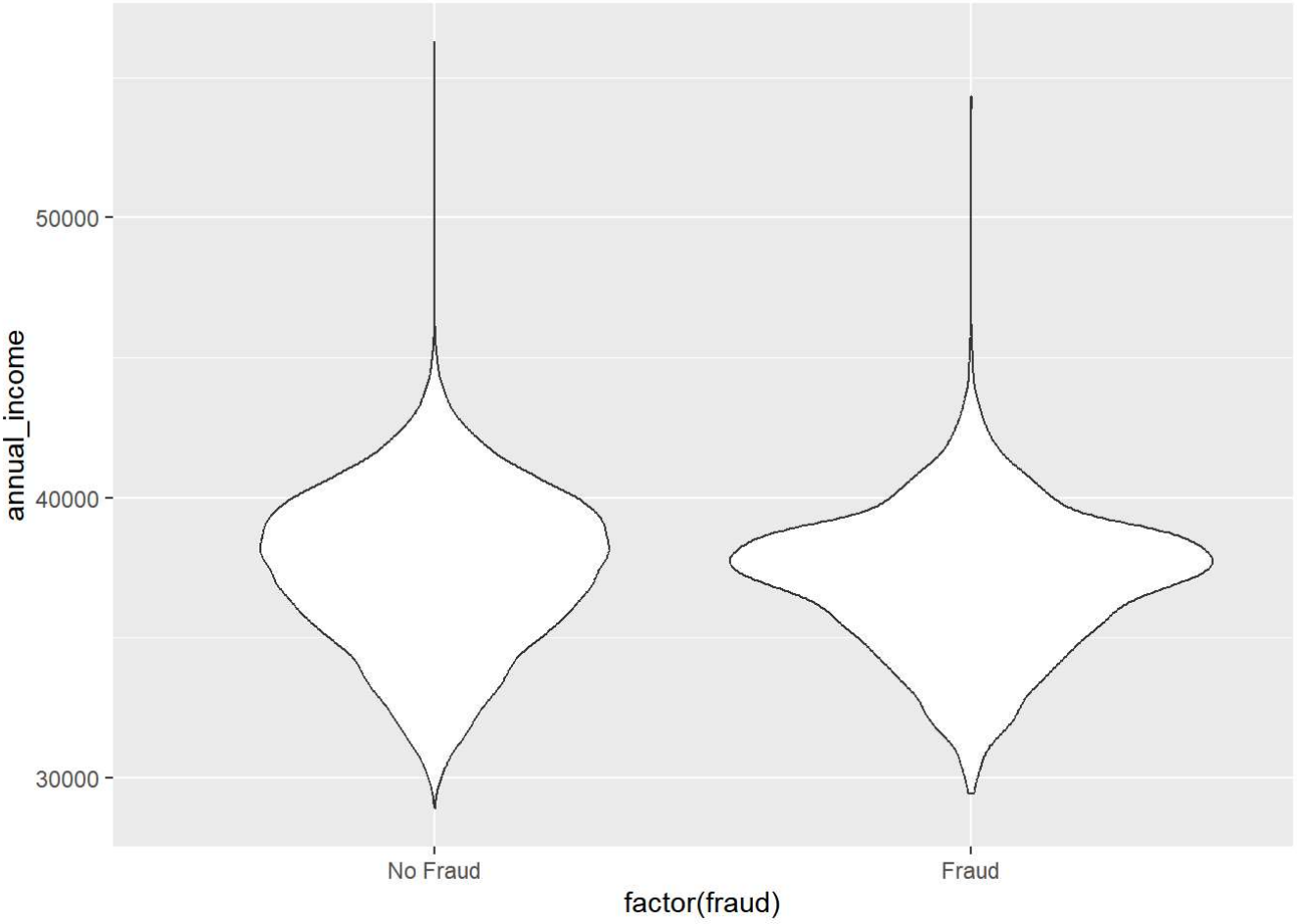
```
ms_uni
```




```
ms_z <- train %>%
  filter(!is.na(marital_status)) %>%
  mutate(fraud_numeric = ifelse(fraud == "Fraud", 1, 0),marital_status_text = ifelse(marital_st
atus == 1, "Yes", "No"))%>%
  group_by(marital_status_text) %>%
  summarize(ms_total = n(),
            ms_fraud = sum(fraud_numeric == 1))
ms_prop_z <-prop.test(ms_z$ms_fraud,ms_z$ms_total,alternative="greater")
ms_prop_z
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  ms_z$ms_fraud out of ms_z$ms_total
## X-squared = 105.56, df = 1, p-value < 2.2e-16
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.04990451 1.00000000
## sample estimates:
##    prop 1    prop 2
## 0.1980885 0.1378364
```

```
# 4. fraud against annual income - Done
income_uni <- train %>%
  filter(!is.na(annual_income), annual_income>0, annual_income >= 20000) %>%
  ggplot(aes(factor(fraud), annual_income)) +
  geom_violin() +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))

income_uni
```




```
income_uni <- train %>%
  filter(!is.na(annual_income), annual_income >= 20000) %>%
  ggplot(aes(factor(fraud),annual_income)) +
  geom_violin()
income_fraud <- train %>%
  select(fraud, annual_income) %>%
  filter(!is.na(annual_income), fraud == "Fraud")
income_no_fraud <- train %>%
  select(fraud, annual_income) %>%
  filter(!is.na(annual_income), fraud == "No Fraud")
cvm_test(income_fraud$annual_income, income_no_fraud$annual_income)
```

```
## Test Stat   P-Value
##  86.45077   0.00025
```

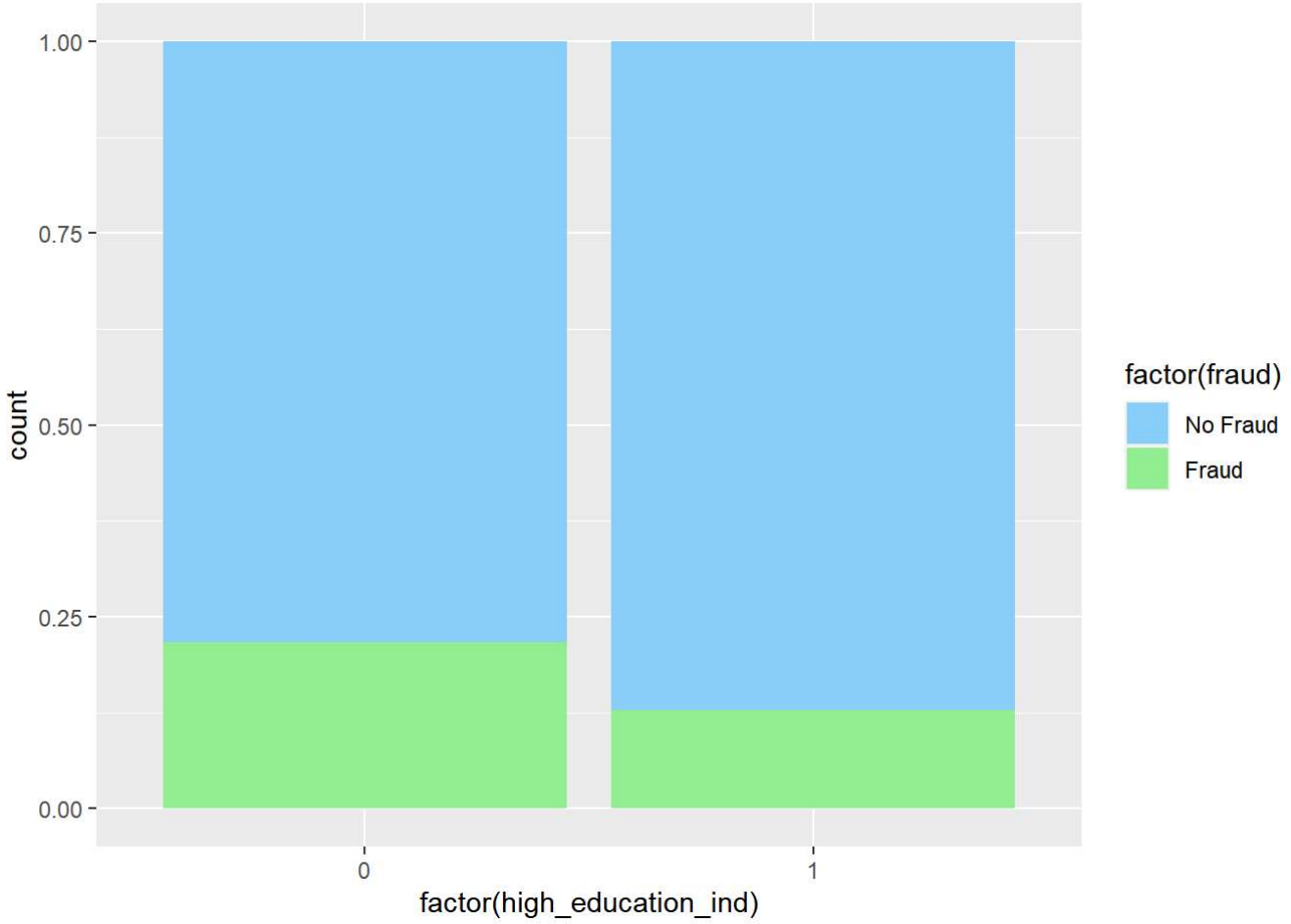
```
## No bootstrap values were more extreme than the observed value.
##  p-value = 1/(2*bootstraps) is an imprecise placeholder
```

```
ad_test(income_fraud$annual_income, income_no_fraud$annual_income)
```

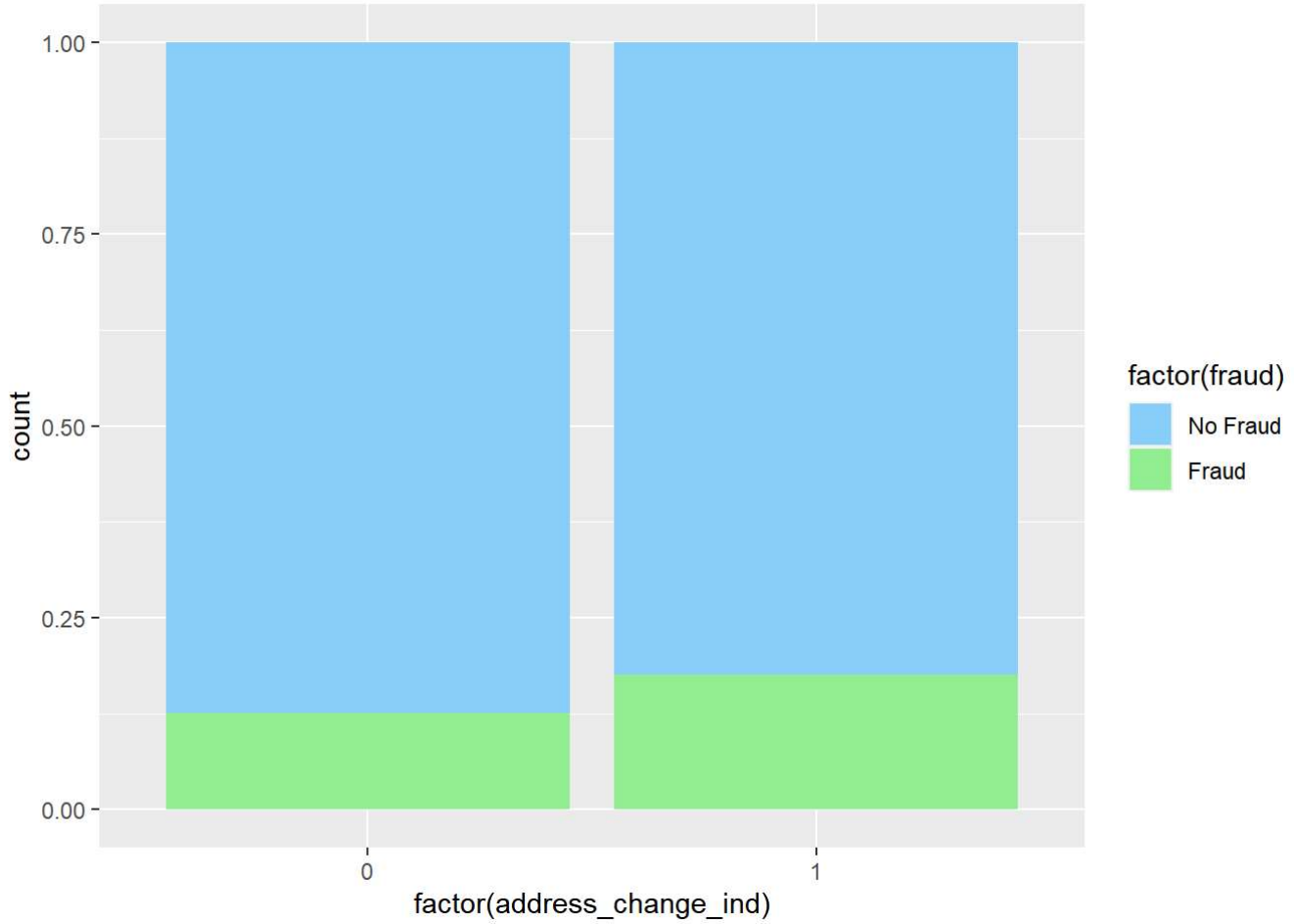
```
##    Test Stat      P-Value
## 4.328872e+06 2.500000e-04
```

```
## No bootstrap values were more extreme than the observed value.
##  p-value = 1/(2*bootstraps) is an imprecise placeholder
```

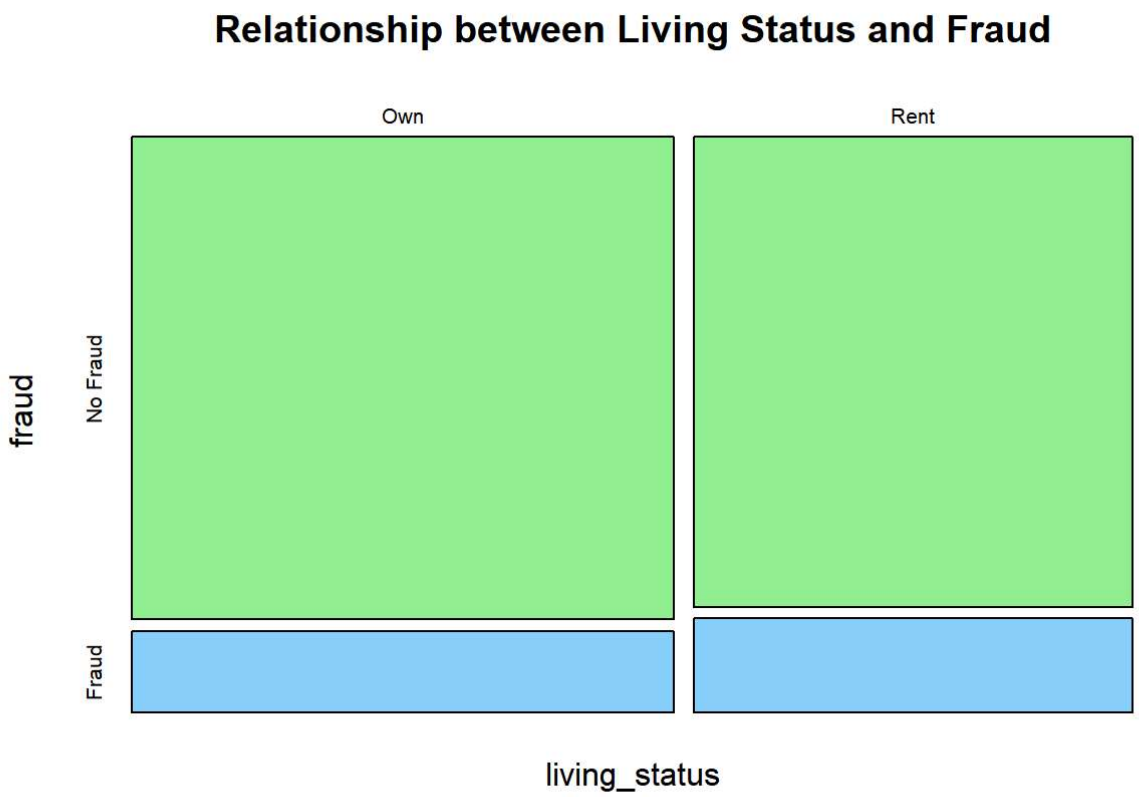
```
# 5. fraud against high_education_ind - Done
edu_uni <- train %>%
  filter(!is.na(high_education_ind)) %>%
  ggplot(aes(x = factor(high_education_ind), fill = factor(fraud))) +
  geom_bar(position = "fill")+
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))

edu_uni
```



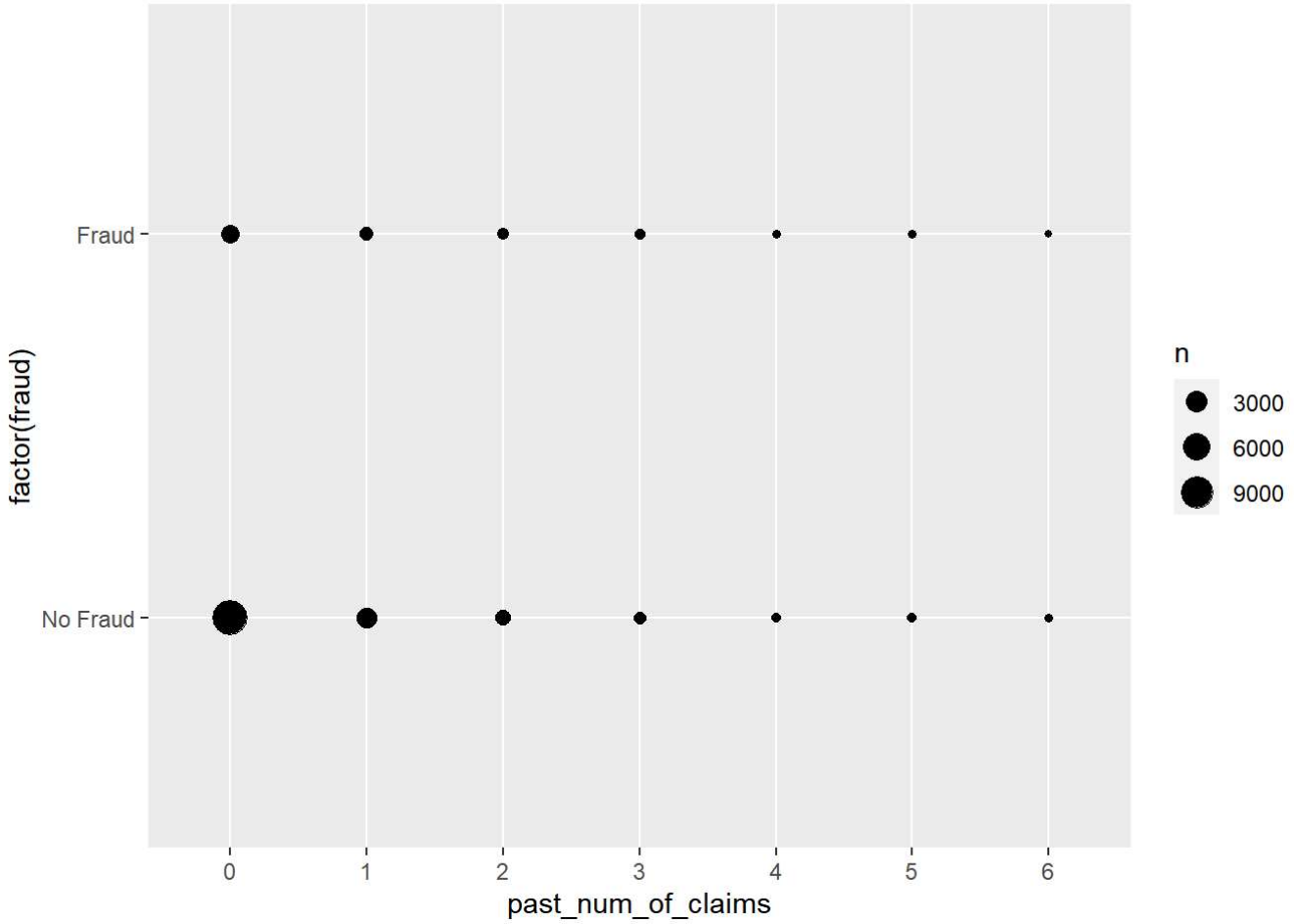

```
# 6. fraud against address_change_ind - Done
address_uni <- train %>%
  filter(!is.na(address_change_ind)) %>%
  ggplot(aes(x = factor(address_change_ind), fill = factor(fraud))) +
  geom_bar(position = "fill") +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))

address_uni
```



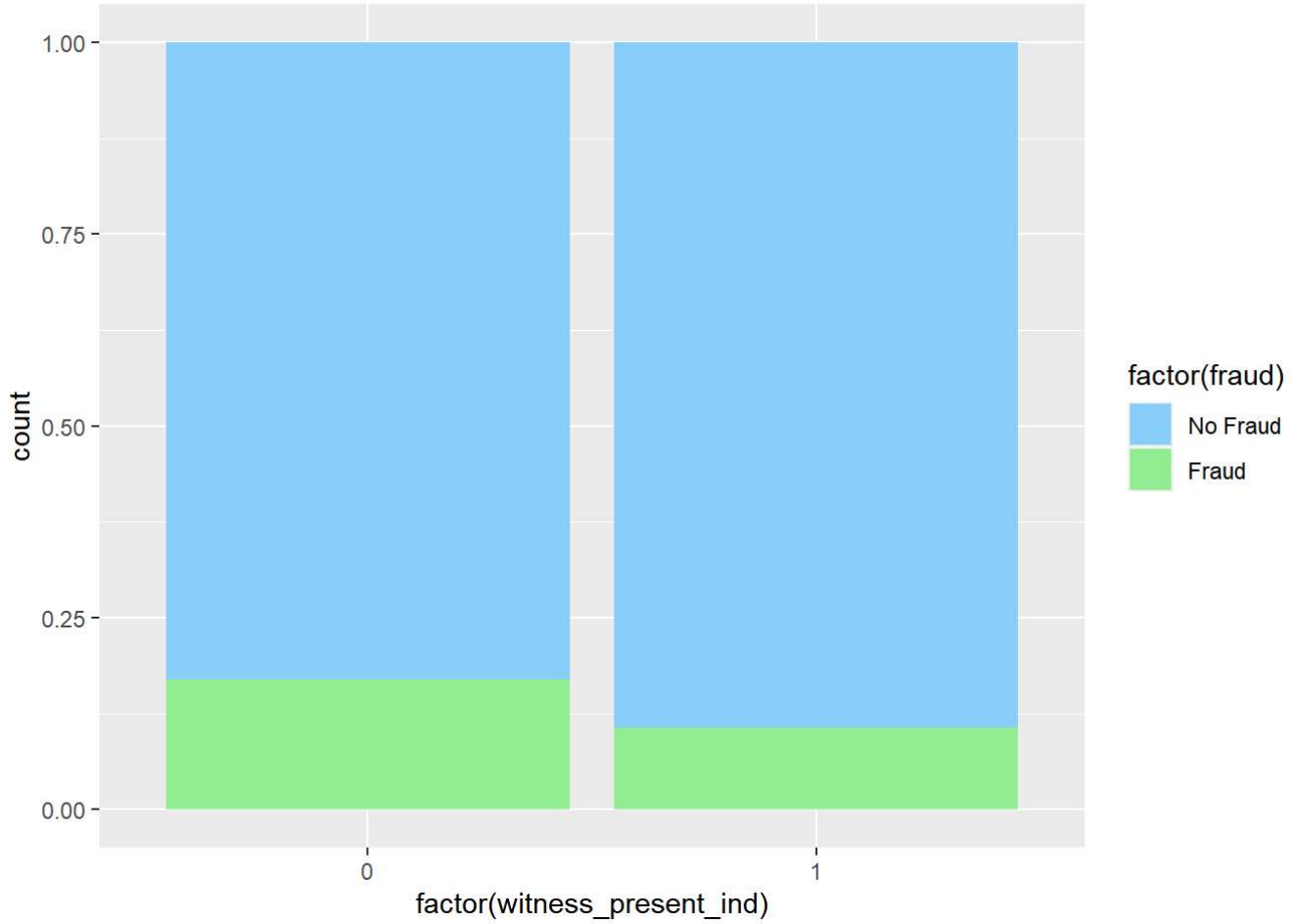

```
# 7. fraud against living_status - Done
mosaicplot(~ living_status + fraud,
           data = train,
           color = c("lightgreen", "lightskyblue"),
           main = "Relationship between Living Status and Fraud")
```



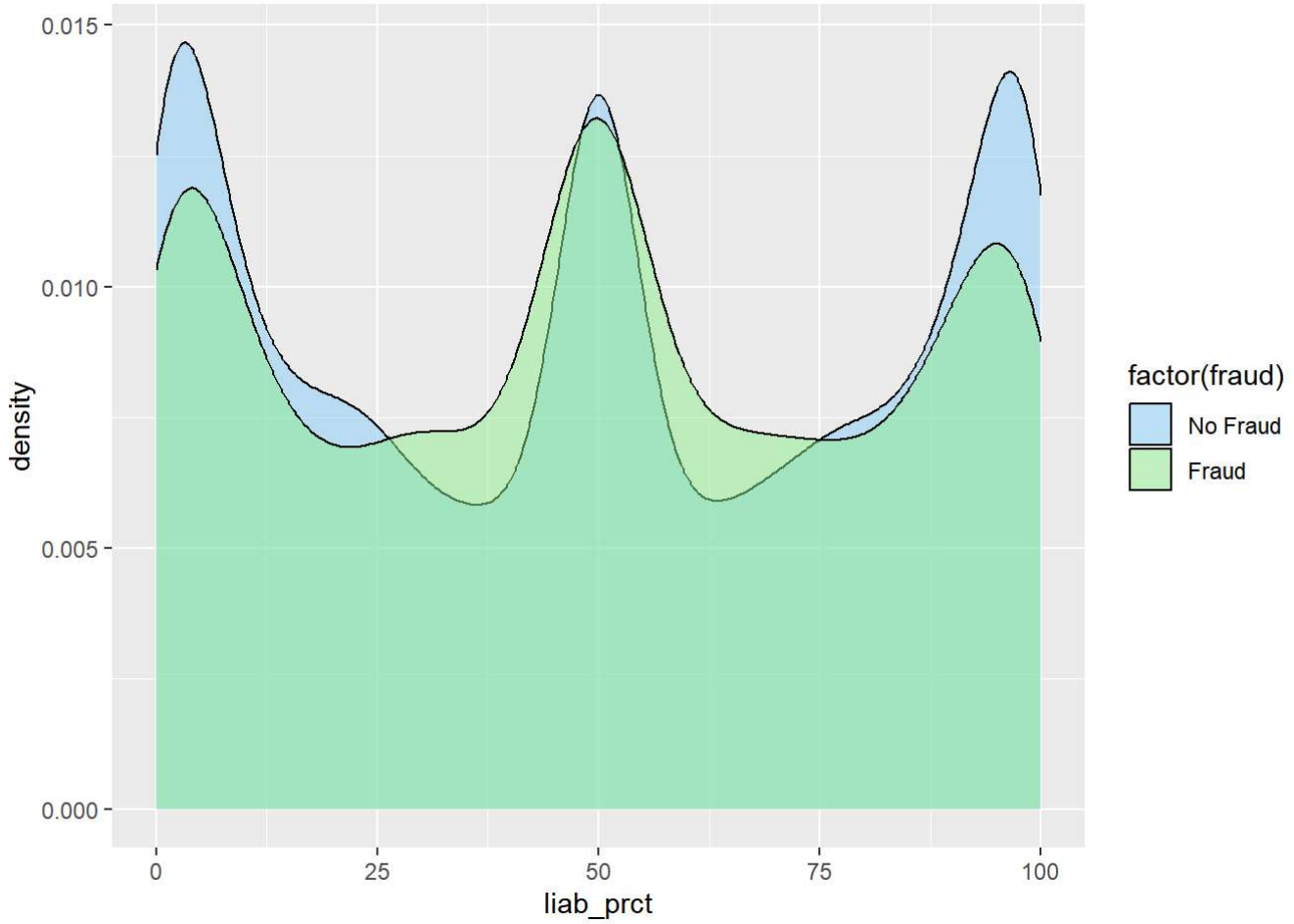

```
  # bar graph attempt
live_uni <- train %>% ggplot(aes(living_status, fill = factor(fraud))) +
  geom_bar(position = "dodge")+
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))


## 8. fraud against past_num_of_claims - Done
train %>% ggplot(aes(past_num_of_claims, factor(fraud))) +
  geom_count() +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))
```




```
  # boxplot attempt
past_uni <- train %>%
  ggplot(aes(past_num_of_claims, fill = fraud)) +
  geom_boxplot(position="fill") +
   scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))

## 9. fraud against witness_present_ind - Done
witness_uni <- train %>%
  filter(!is.na(witness_present_ind)) %>%
  ggplot(aes(x = factor(witness_present_ind), fill = factor(fraud))) +
  geom_bar(position = "fill") +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))

witness_uni
```




```
## 10. fraud against liab_prct - Not Done
train %>% ggplot(aes(liab_prct, fill = factor(fraud))) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))
```



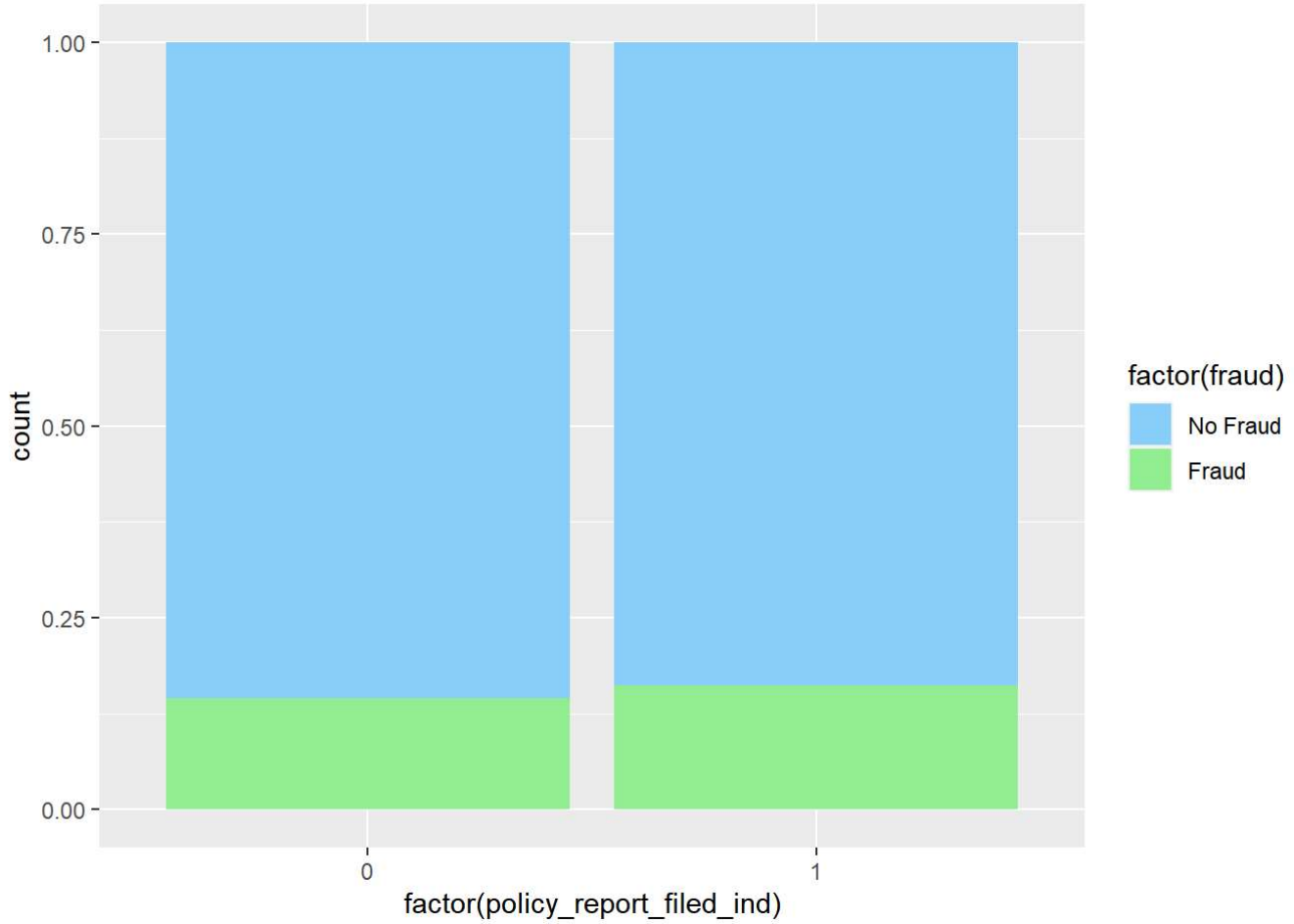

```
liab_fraud <- train %>%
  select(fraud, liab_prct) %>%
  filter(!is.na(liab_prct), fraud == "Fraud")
liab_no_fraud <- train %>%
  select(fraud, liab_prct) %>%
  filter(!is.na(liab_prct), fraud == "No Fraud")
cvm_test(liab_fraud$liab_prct, liab_no_fraud$liab_prct)
```

```
## Test Stat   P-Value
##  5.425545  0.011000
```

```
ad_test(liab_fraud$liab_prct, liab_no_fraud$liab_prct)
```

```
##  Test Stat    P-Value
## 269617.586      0.008
```

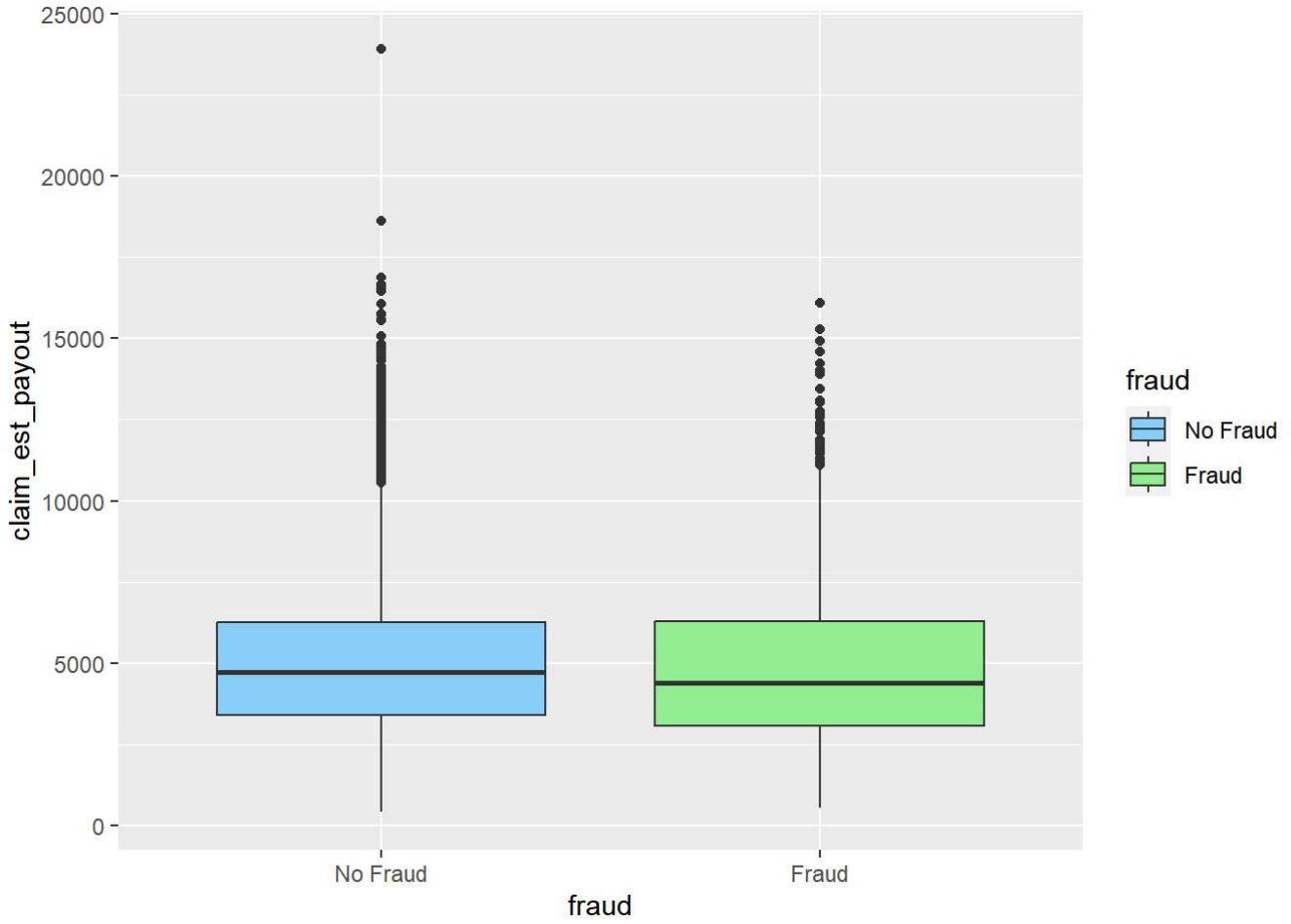
```
## 11. fraud against policy_report_filed_ind
policy_uni <- train %>%
  filter(!is.na(policy_report_filed_ind)) %>%
  ggplot(aes(x = factor(policy_report_filed_ind), fill = factor(fraud))) +
  geom_bar(position = "fill")+
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))

policy_uni
```




```
## 12. fraud against claim_est_payout
payout_uni <- train %>%
  ggplot(aes(fraud, claim_est_payout, fill = fraud)) +
  geom_boxplot() +
  scale_fill_discrete(name="Fraud")+
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))
```

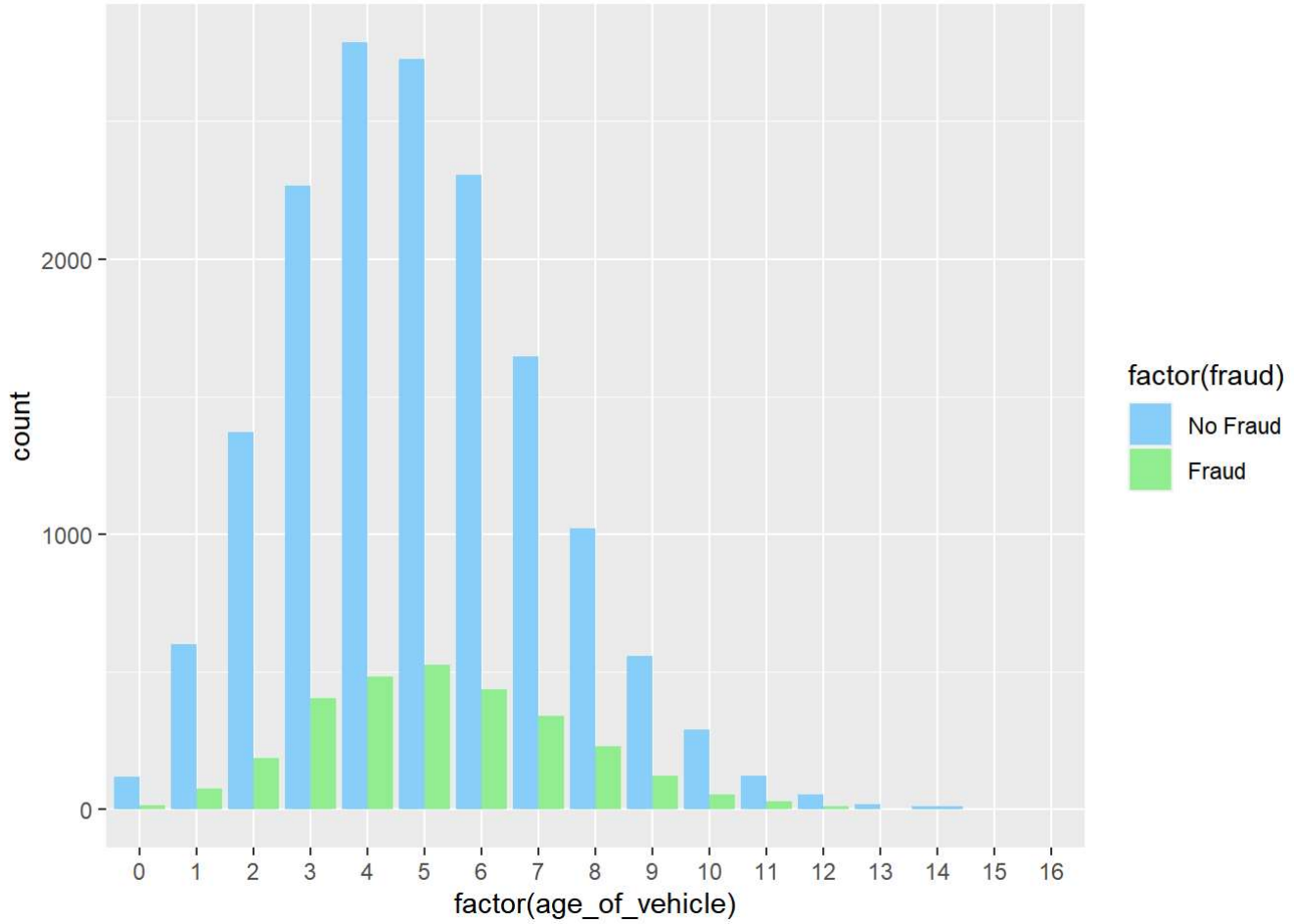
```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
payout_uni
```




```
## 13. fraud against age_of_vehicle
train %>%
  ggplot(aes(factor(age_of_vehicle), fill = factor(fraud))) +
  geom_bar(position = "dodge") +
  scale_fill_discrete(name="Fraud")+
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```




```
veh_age_fraud <- train %>%
  select(fraud, age_of_vehicle) %>%
  filter(!is.na(age_of_vehicle), fraud == "Fraud")
veh_age_no_fraud <- train %>%
  select(fraud, age_of_vehicle) %>%
  filter(!is.na(age_of_vehicle), fraud == "No Fraud")
cvm_test(veh_age_fraud$age_of_vehicle, veh_age_no_fraud$age_of_vehicle)
```

```
## Test Stat   P-Value
##  24.55885   0.00025
```

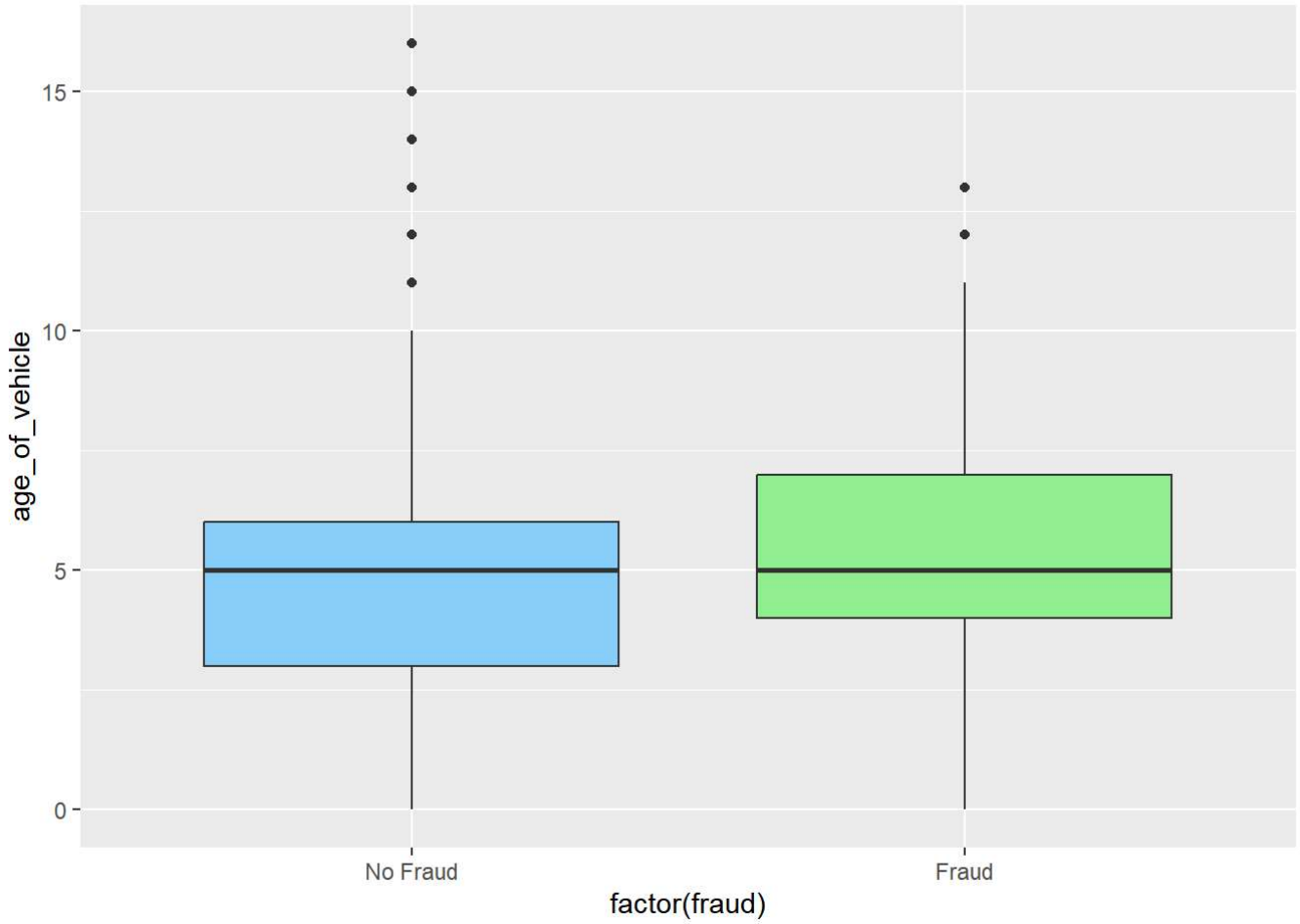
```
## No bootstrap values were more extreme than the observed value.
##  p-value = 1/(2*bootstraps) is an imprecise placeholder
```

```
ad_test(veh_age_fraud$age_of_vehicle, veh_age_no_fraud$age_of_vehicle)
```

```
##   Test Stat     P-Value
## 1.21515e+06 2.50000e-04
```

```
## No bootstrap values were more extreme than the observed value.
##  p-value = 1/(2*bootstraps) is an imprecise placeholder
```

```
  # box plot attempt
veh_age_uni <- train %>%
  ggplot(aes(factor(fraud), age_of_vehicle, fill = fraud)) +
  geom_boxplot(position = "dodge", fill=c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen"))

veh_age_uni
```



After univariate analysis, we also found some relationships between bivariate analysis with the correlation tests and correlation matrices. We found that annual income is strongly correlated with the age of the driver, but for our purpose to compare those variables to better fit the logistic regression model, we decided to drop the annual income and replace it with the age of the driver. In the logistic model, we found that the dummy variable of the high education index has a negative impact on the probability of making fraudulent claims, and marital status also has a negative impact on the fraudulent. Therefore, we made the plot by age group, selecting this two binary variables and using factors and pivots techniques to transform them into a combined categorical variables with four categories: Married and Highly Educated, Married but Not Highly Educated, Single and Highly Educated, Single but Not Highly Educated. The plot shows proportions of fraud within each group. The result shows that keeping the education factor constant, people who are single have a higher proportion of fraud; keeping the marriage factor constant, the less educated group has a higher proportion of fraud.

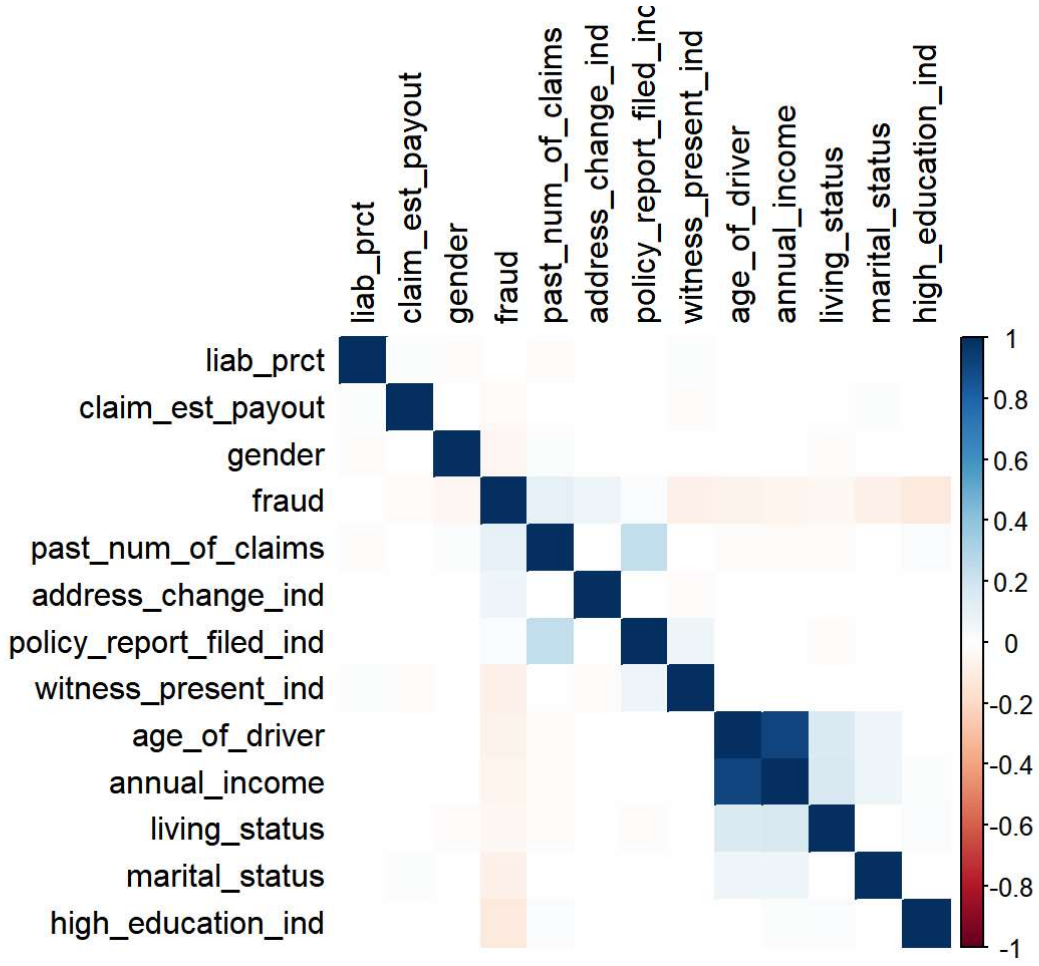
```
train_bivar <- train_bivar[complete.cases(train_bivar), ]

if (!is.factor(train_bivar$fraud)) {
  train_bivar$fraud <- as.factor(train_bivar$fraud)
  train_bivar$fraud <- factor(train_bivar$fraud, levels = c(0, 1), labels = c("No Fraud", "Frau
d"))
}

#The Bivariate Analysis Investigates Relationships among variables that were tested statistical
ly significant during Univeriate Analysis

#Categorical: gender, marital_status, high_education_ind, address_change_ind, living_status, wi
tness_present_ind, policy_report_filed_ind

#Numeric: age_of_driver, annual_income, liab_prct, claim_est_payout, age_of_vehicle, past_num_o
f_claims

#Main Goal of Bivariate Analysis: To investigate the relationship between two variables and the
ir relationships toward fraud

#Steps
#Numeric vs. Numeric
#Scatterplot
#Correlation Test & Corr if necessary
#If independent, regression
library(ggpubr)
bivar_train <- train_bivar %>% select(age_of_driver, gender, marital_status, high_education_in
d, address_change_ind, living_status, witness_present_ind, policy_report_filed_ind, annual_inco
me, liab_prct, claim_est_payout, age_of_vehicle, past_num_of_claims, fraud) %>% filter(age_of_d
river < 100) %>% mutate(log_annual_income = log(annual_income))
```

```
## Warning in log(annual_income): 产生了NaNs
```

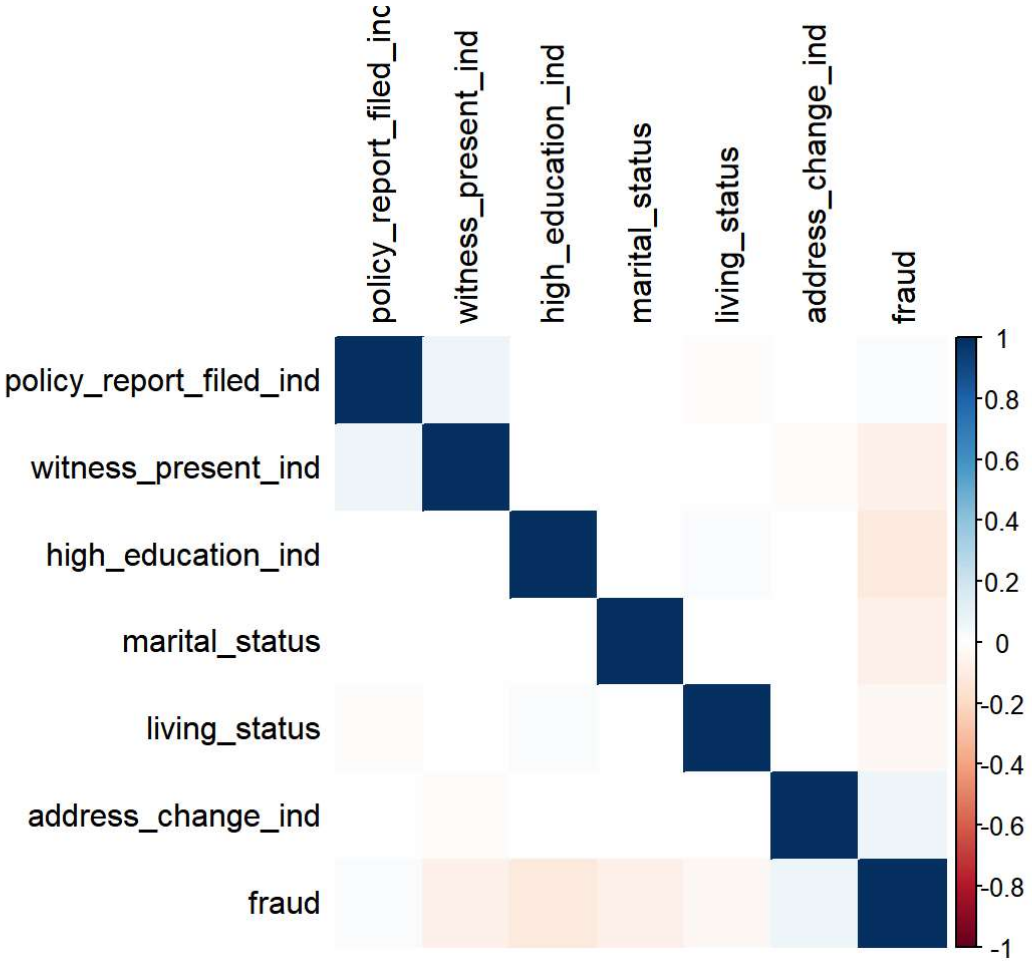
```
bivar_train_Cats <- bivar_train %>% mutate(marital_status = as.character(marital_status), high_
education_ind = as.character(high_education_ind), address_change_ind = as.character(address_cha
nge_ind), witness_present_ind = as.character(witness_present_ind), policy_report_filed_ind = a
s.character(policy_report_filed_ind)) %>% select(gender, marital_status, high_education_ind, ad
dress_change_ind, living_status, witness_present_ind, policy_report_filed_ind)
```

```
bivar_heat <- bivar_train
bivar_heat$gender <- as.factor(bivar_heat$gender)
bivar_heat$fraud <- as.factor(bivar_heat$fraud)
bivar_heat$living_status <- as.factor(bivar_heat$living_status)
bivar_heat <- bivar_heat %>%
    mutate(gender = recode(gender, "M" = as.numeric("1"), "F" = as.numeric("0")),
           fraud = recode(fraud, "No Fraud" = as.numeric("0"),"Fraud" = as.numeric("1")),
           living_status = recode(living_status, "Rent" = as.numeric("0"), "Own" = as.numeric
("1")),
           log_claim_est_payout = log(claim_est_payout)
    ) %>%
  select(age_of_driver, annual_income, gender, liab_prct, claim_est_payout, past_num_of_claims,
marital_status, high_education_ind, address_change_ind, living_status, witness_present_ind, pol
icy_report_filed_ind, fraud)

bivar_heat_Cats <- bivar_heat %>% select(marital_status, high_education_ind, address_change_in
d, living_status, witness_present_ind, policy_report_filed_ind, fraud)


NC <- cor(bivar_heat)
C <- cor(bivar_heat_Cats, method = "kendall")
```

```
corrplot(NC, method = "color", tl.col = "black", order = "AOE")
```




```
corrplot(C, method = "color", tl.col = "black", order = "AOE")
```




```
#Make the regression and the summary
bivar_lreg <- bivar_train %>% mutate(log_annual_income = log(annual_income))
```

```
## Warning in log(annual_income): 产生了NaNs
```

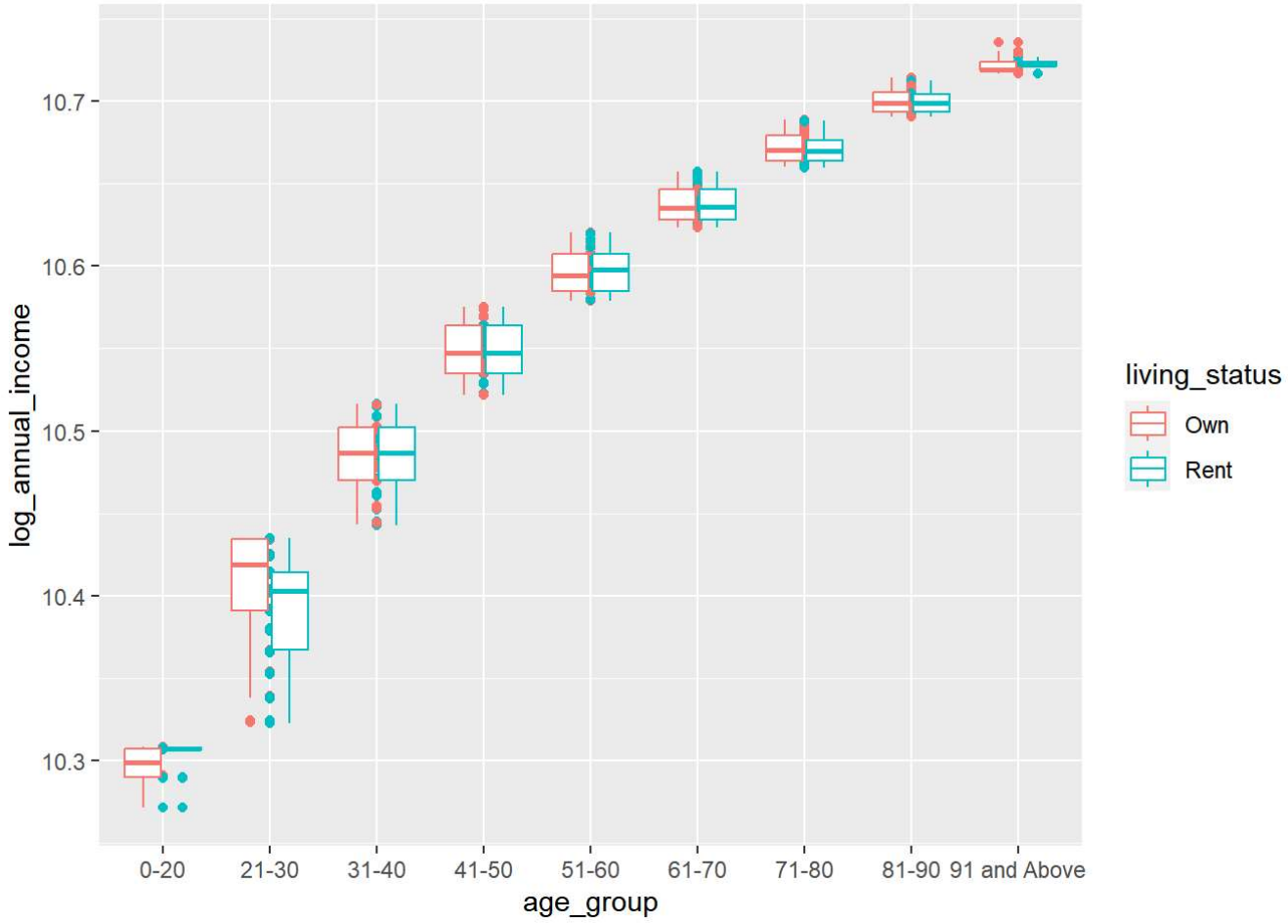
```
lreg <- lm(log_annual_income ~ age_of_driver + past_num_of_claims + living_status, bivar_lreg)
summary(lreg)
```

```
##
## Call:
## lm(formula = log_annual_income ~ age_of_driver + past_num_of_claims +
##     living_status, data = bivar_lreg)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.132668 -0.005102  0.006075  0.009852  0.017979
##
## Coefficients:
##                      Estimate Std. Error  t value Pr(>|t|)
## (Intercept)         1.026e+01  5.001e-04 20517.69  < 2e-16 ***
## age_of_driver       6.136e-03  1.036e-05   592.43  < 2e-16 ***
## past_num_of_claims -3.414e-04  1.274e-04    -2.68  0.00737 **
## living_statusRent  -5.976e-03  2.460e-04   -24.29  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01657 on 18799 degrees of freedom
##   (因为不存在，15个观察量被删除了)
## Multiple R-squared:  0.951,  Adjusted R-squared:  0.951
## F-statistic: 1.216e+05 on 3 and 18799 DF,  p-value: < 2.2e-16
```

```
#Make a ggplot that separates age groups and takes different groups to the proportion of fraud

bivar_train["age_group"] = cut(bivar_train$age_of_driver, c(0, 20, 30, 40, 50, 60, 70, 80, 90,
Inf), c("0-20", "21-30", "31-40", "41-50", "51-60", "61-70", "71-80", "81-90", "91 and Above"))
bivar_train %>% ggplot() +
  geom_point(aes(age_group, log_annual_income, color = living_status, group = living_status)) +
  geom_boxplot(aes(age_group, log_annual_income, color = living_status))
```

```
## Warning: Removed 15 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 15 rows containing missing values (`geom_point()`).
```




```
bivar_train_logi <- bivar_train
bivar_train_logi$high_education_ind <- factor(bivar_train_logi$high_education_ind)
bivar_train_logi$witness_present_ind <- factor(bivar_train_logi$witness_present_ind)
bivar_train_logi$marital_status <- factor(bivar_train_logi$marital_status)
bivar_train_logi$address_change_ind <- factor(bivar_train_logi$address_change_ind)
bivar_train_logi$living_status <- factor(bivar_train_logi$living_status)
bivar_train_logi$gender <- factor(bivar_train_logi$gender)
bivar_train_logi
```

```
## # A tibble: 18,818 × 16
##    age_of_driver gender marita…¹ high_…² addre…³ livin…⁴ witne…⁵ polic…⁶ annua…⁷
##            <dbl> <fct>  <fct>    <fct>   <fct>   <fct>   <fct>     <dbl>   <dbl>
##  1            50 F      1        1       0       Own     0             0   39117
##  2            47 M      1        1       0       Own     0             0   38498
##  3            28 M      0        0       1       Rent    1             1   33343
##  4            36 M      1        1       0       Own     1             0   35832
##  5            60 F      1        1       1       Rent    0             1   40948
##  6            50 F      1        1       1       Own     0             0   39126
##  7            28 M      1        1       1       Own     1             1   33327
##  8            55 M      1        1       0       Own     0             0   40079
##  9            47 M      1        0       1       Own     0             0   38505
## 10            34 F      0        0       1       Own     1             1   35273
## # … with 18,808 more rows, 7 more variables: liab_prct <dbl>,
## #   claim_est_payout <dbl>, age_of_vehicle <dbl>, past_num_of_claims <dbl>,
## #   fraud <fct>, log_annual_income <dbl>, age_group <fct>, and abbreviated
## #   variable names ¹marital_status, ²high_education_ind, ³address_change_ind,
## #   ⁴living_status, ⁵witness_present_ind, ⁶policy_report_filed_ind,
## #   ⁷annual_income
```

```
bivar_logi <- glm(fraud ~ gender + high_education_ind + witness_present_ind + marital_status +
address_change_ind +  living_status, bivar_train_logi, family = "binomial")
summary(bivar_logi)
```

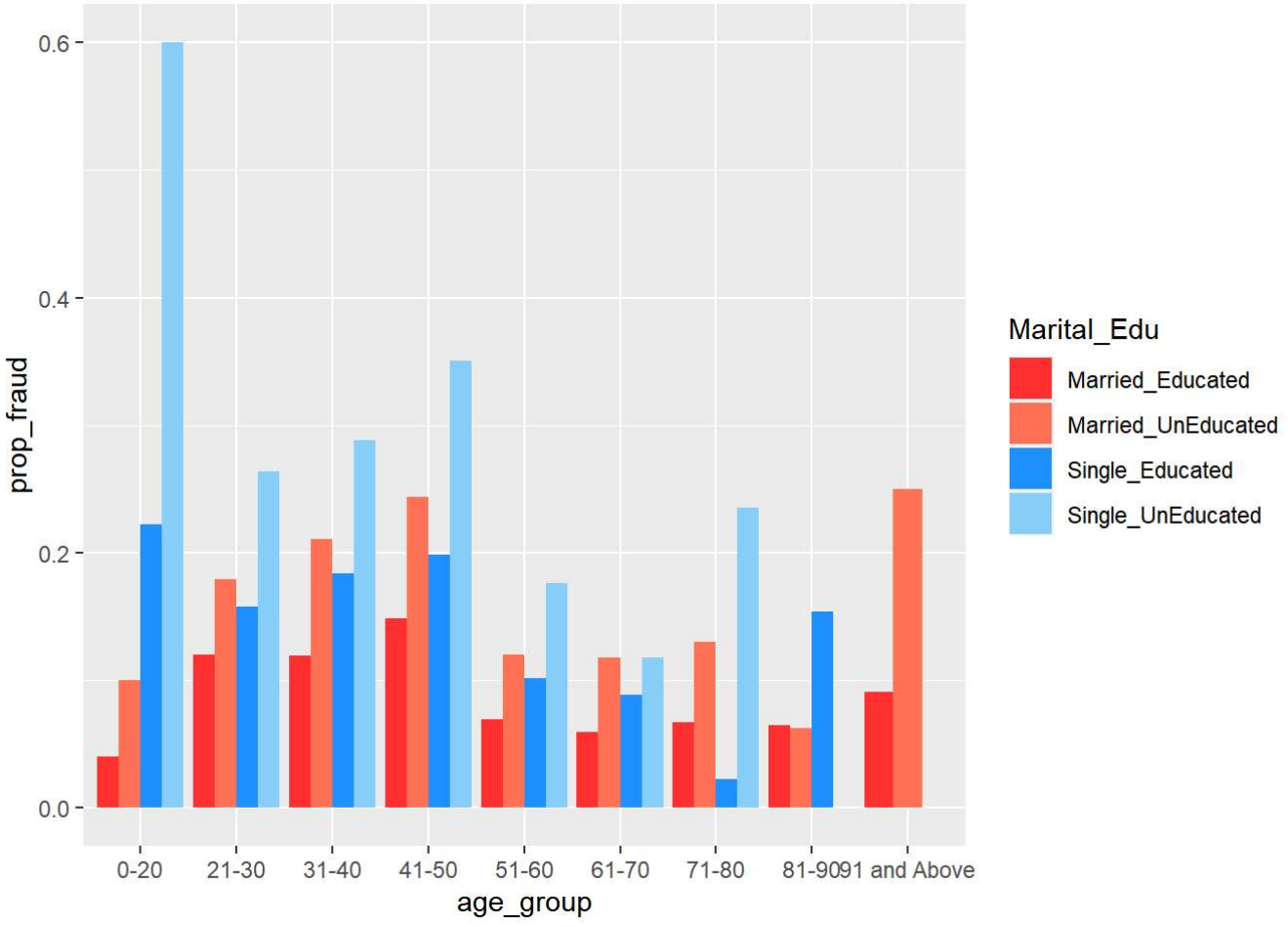
```
##
## Call:
## glm(formula = fraud ~ gender + high_education_ind + witness_present_ind +
##     marital_status + address_change_ind + living_status, family = "binomial",
##     data = bivar_train_logi)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9787  -0.6144  -0.5116  -0.4181   2.4484
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.05097    0.05903 -17.804  < 2e-16 ***
## genderM              -0.25793    0.04100  -6.290 3.17e-10 ***
## high_education_ind1  -0.63940    0.04193 -15.250  < 2e-16 ***
## witness_present_ind1 -0.55306    0.05405 -10.232  < 2e-16 ***
## marital_status1      -0.44479    0.04313 -10.314  < 2e-16 ***
## address_change_ind1   0.39610    0.04261   9.297  < 2e-16 ***
## living_statusRent     0.16758    0.04103   4.084 4.42e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16226  on 18817  degrees of freedom
## Residual deviance: 15640  on 18811  degrees of freedom
## AIC: 15654
##
## Number of Fisher Scoring iterations: 4
```

```
#Make a plot for proportion of fraud given each age group
#bivar_train
partial_EDA1 <- bivar_train %>% select(age_group, marital_status, high_education_ind, fraud)
partial_EDA1$fraud <- as.character(partial_EDA1$fraud)
partial_EDA1$age_group <- as.character(partial_EDA1$age_group)
partial_EDA1
```

```
## # A tibble: 18,818 × 4
##    age_group marital_status high_education_ind fraud
##    <chr>              <dbl>              <dbl> <chr>
##  1 41-50                  1                  1 No Fraud
##  2 41-50                  1                  1 No Fraud
##  3 21-30                  0                  0 No Fraud
##  4 31-40                  1                  1 No Fraud
##  5 51-60                  1                  1 No Fraud
##  6 41-50                  1                  1 No Fraud
##  7 21-30                  1                  1 No Fraud
##  8 51-60                  1                  1 No Fraud
##  9 41-50                  1                  0 Fraud
## 10 31-40                  0                  0 No Fraud
## # … with 18,808 more rows
```

```
partial1_group <- partial_EDA1 %>% group_by(age_group, marital_status, high_education_ind) %>%
count(fraud) %>%
  pivot_wider(names_from = fraud, values_from = n) %>%
  select_all(~gsub("\\s+|\\.", "_", .)) %>%
  mutate(prop_fraud = Fraud / (Fraud + No_Fraud))

partial1_group$Marital_Edu <- paste(partial1_group$marital_status, partial1_group$high_educatio
n_ind)

partial1_group$Marital_Edu <- replace(partial1_group$Marital_Edu, partial1_group$Marital_Edu ==
"0 0", "Single_UnEducated")
partial1_group$Marital_Edu <- replace(partial1_group$Marital_Edu, partial1_group$Marital_Edu ==
"0 1", "Single_Educated")
partial1_group$Marital_Edu <- replace(partial1_group$Marital_Edu, partial1_group$Marital_Edu ==
"1 0", "Married_UnEducated")
partial1_group$Marital_Edu <- replace(partial1_group$Marital_Edu, partial1_group$Marital_Edu ==
"1 1", "Married_Educated")


partial1_group %>% ggplot() +
  geom_col(aes(age_group, prop_fraud, fill = Marital_Edu), position = "dodge") + scale_fill_man
ual(values = c("Single_UnEducated" = "lightskyblue", "Single_Educated" = "dodgerblue", "Married
_UnEducated" = "coral1", "Married_Educated" = "firebrick1"))
```

```
## Warning: Removed 2 rows containing missing values (`geom_col()`).
```



Bivariate, or inter-predictor visualizations, can uncover hidden insights that may not be apparent through univariate analysis alone, thus providing a more comprehensive understanding of the phenomenon at hand.

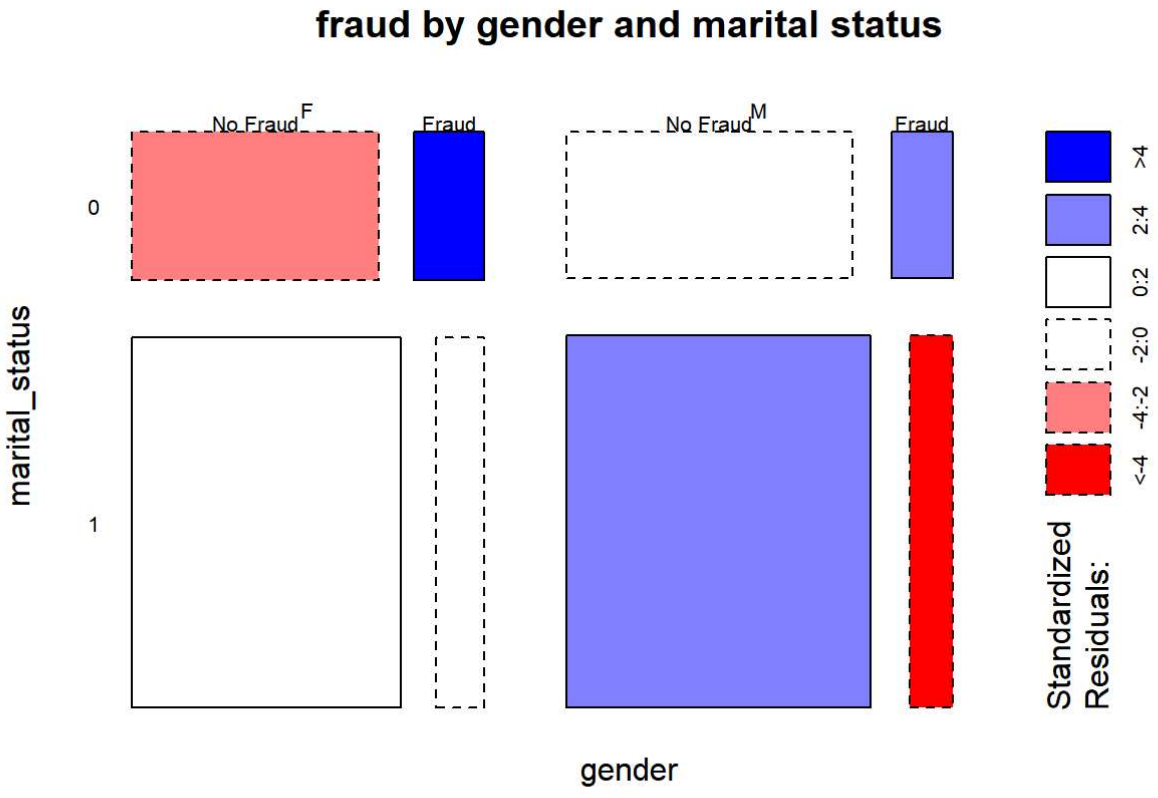
Our mosaic plot visualization of fraud by gender and marital_status sheds light on the complex interplay between these two factors. In the univariate analysis, we observed that females were more likely than males to commit insurance fraudulence. This finding is substantiated by this mosaic plot, where the fraudulent cases of single women exhibit a high standardized residual, indicating a high observed count frequency. This finding can also explain why married men have a low standardized residual, suggesting a lower observed count frequency of fraudulent cases.

However, the univariate analysis alone fails to account for the fact that single males also exhibit high standardized residuals in the fraudulent cases, implying that this group may be highly likely to make fraudulent claims as well. This observation highlights the importance of conducting bivariate visualizations, as they provide a more nuanced perspective of the data, allowing for the identification of potential risk groups that may not be evident from the univariate results.

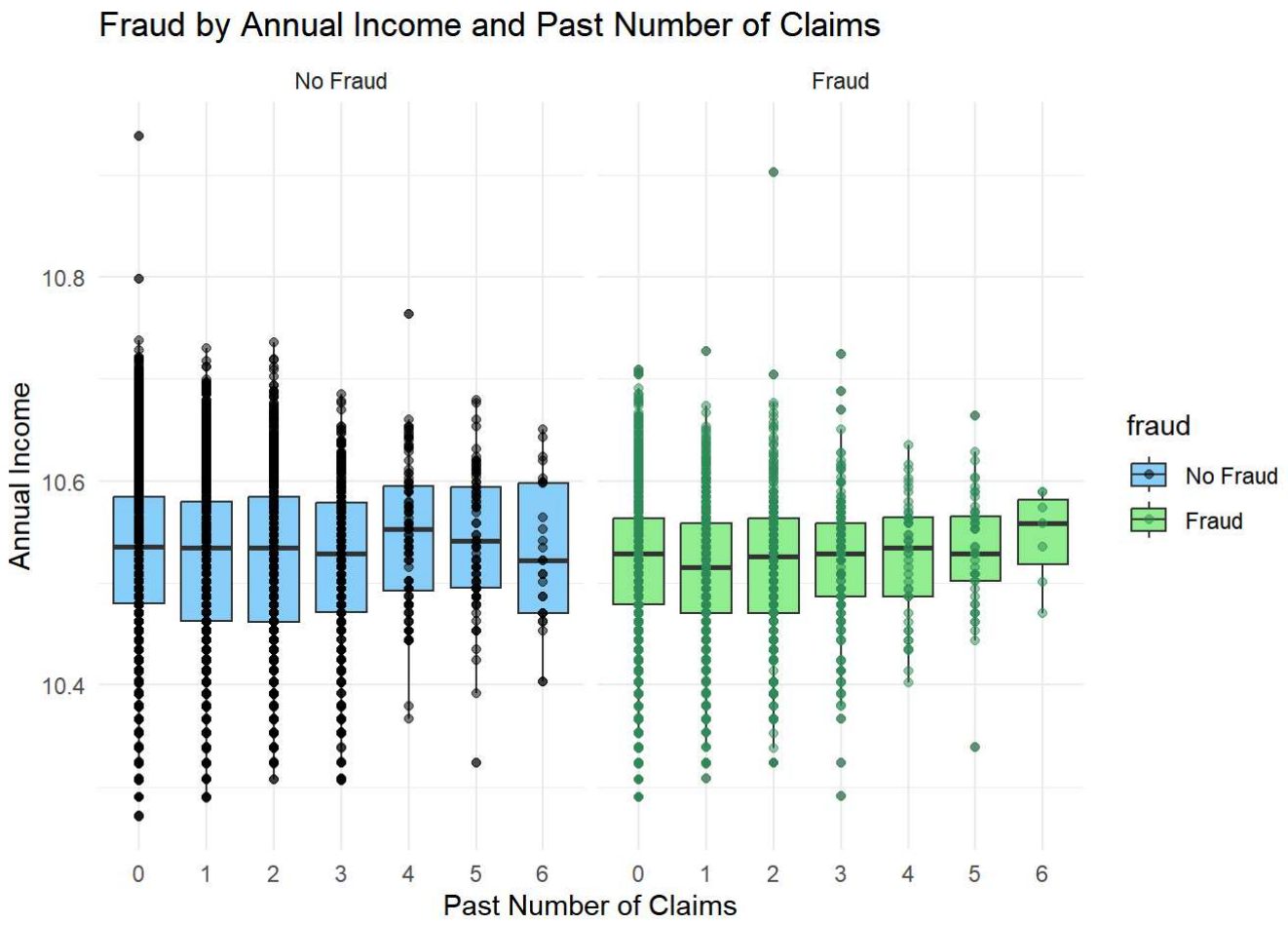
Another bivariate visualization is the facetted boxplot to analyze the relationship between fraud, past_number_of_claims, and log-transformed annual_income, where the log transformation aims to compress the range of values into a more manageable scale.

Upon examining the facetted boxplots, several observations can be made. Firstly, as the number of past claims increases, the variability of annual income appears to diminish. This suggests that drivers with a higher number of past claims exhibit less variation in their annual incomes. Secondly, the distributions of log-transformed annual income are generally similar across the Fraud and No Fraud. This finding is consistent with the results of our univariate analysis, in which we utilized violin plots to visualize the relationship between fraud and annual income. The shapes of the violins for Fraud and No Fraud were found to be similar, with data points for Fraud cases being more concentrated around the median. It is worth noting that when the past number of claims reach 2 or 3, fraudulent cases tend to have more outliers, which could imply that claimants at these two categories are susceptible to fraudulence with more variety of income levels. Thirdly, it is crucial to acknowledge that the sample sizes of fraudulent cases are significantly smaller than those of non-fraudulent ones, indicating an imbalance in the data. This issue could potentially affect the faceted box plots such as the estimation of the interquartile range and the identification of outliers. To address this limitation, resampling techniques to balance the number of instances of both groups may be employed in future analyses.

```
if (!is.factor(train_yufeng$fraud)) {
  train_yufeng$fraud <- as.factor(train_yufeng$fraud)
  train_yufeng$fraud <- factor(train_yufeng$fraud, levels = c(0, 1), labels = c("No Fraud", "Fr
aud"))
}
#correlated variables: (y,x)
#log_annual_income, marital_status;
#age_of_driver vs log_annual_income;
#log_annual_income, gender;
#living_status, log_annual_income;
#log_annual_income vs past_num_of_claims;
#age_of_driver vs past_num_claims;
#policy_report_filed_ind, past_num_of_claims;
#witness_present_ind, policy_report_filed_ind;
#living_status, policy_report_filed_ind;
#high_education_ind vs living_status;



# 1. marital_status / gender with fraud
mosaicplot(~ gender + marital_status + fraud,
           data = train,
           color = 3:5,
           las = 1,
           shade = TRUE,
           margin = list(1:2, 3),
           main = "fraud by gender and marital status")
```



```
# 2. address_change_id & living_status with fraud
  # make sure `fraud` is a factor
train_yufeng %>%
  filter(annual_income > 2000) %>%
  ggplot() +
  geom_boxplot(aes(past_num_of_claims, log(annual_income), fill = fraud), outlier.alpha = 0.5)
+
  geom_point(aes(past_num_of_claims, log(annual_income), color = fraud), alpha = 0.5) +
  facet_grid(~ fraud) +
  labs(title = "Fraud by Annual Income and Past Number of Claims",
       x = "Past Number of Claims",
       y = "Annual Income") +
  theme_minimal() +
  scale_fill_manual(values = c("No Fraud" = "lightskyblue", "Fraud" = "lightgreen")) +
  scale_color_manual(values = c("No Fraud" = "black", "Fraud" = "seagreen"))
```



The last bivariate visualization is a heatmap of fraud by marital_status, address_change_ind, and living_status, where several noteworthy patterns emerge. One striking observation is that the subgroup of married drivers who own a house and have recently changed their address exhibit a higher likelihood of committing fraudulent behaviors compared to other combinations of the three indicators. This finding presents an intriguing contradiction with our univariate results, which indicated that single people and those living in rented accommodations were more inclined to make fraudulent claims. We believe that the discrepancy between the bivariate and univariate analysis may stem from multiple factors. One potential factor is higher financial stress. Married homeowners who have recently moved residence may experience a variety of costs, such as relocation expenses, home renovations, or new furnishings, let alone larger mortgage payments or other expenses tied to their new property. As a result, they may resort to fraudulent activities to alleviate their

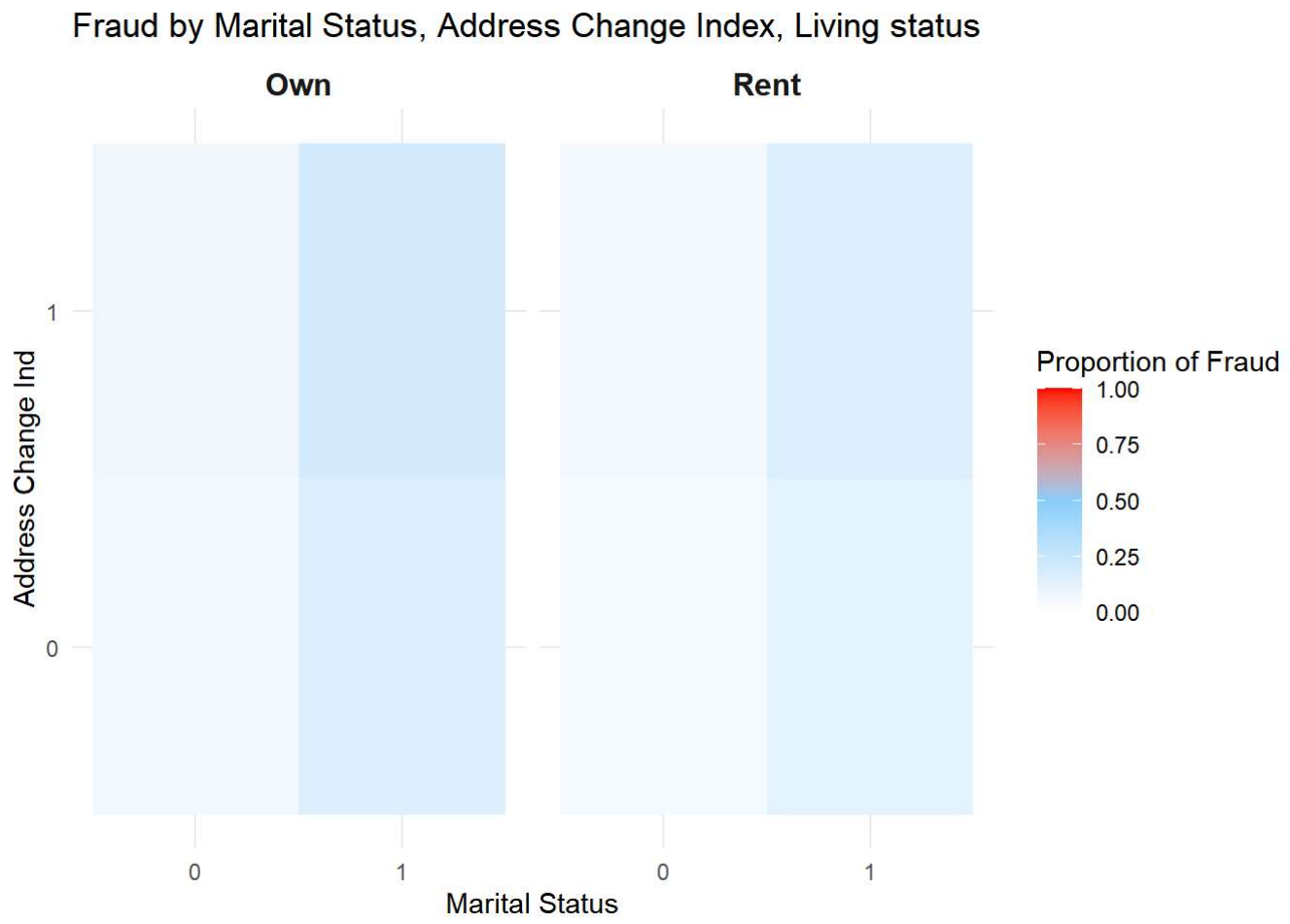
financial burdens when they encounter vehicle damage. By identifying the unique challenges associated with specific demographic subgroups, insurers can tailor their fraud detection mechanisms to better identify and mitigate fraudulent claims.

```
# 3. marital_status, address_change_ind, living_status vs fraud
train_yufeng$fraud <- ifelse(train_yufeng$fraud == "Fraud", 1, 0)
```

```
fraud_summary <- train_yufeng %>%
  group_by(marital_status, address_change_ind, living_status, fraud) %>%
  summarize(count = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'marital_status', 'address_change_ind',
## 'living_status'. You can override using the `.groups` argument.
```

```
fraud_proportion <- fraud_summary %>%
  mutate(proportion = count / sum(count)) %>%
  spread(key = fraud, value = proportion) %>%
  mutate(Fraud = ifelse(is.na(`1`), `0`, `1`)) %>%
  select(-`0`, -`1`)

fraud_proportion %>%
  ggplot(aes(marital_status, address_change_ind, fill = Fraud)) +
  geom_tile() +
  facet_wrap(~ living_status) +
  scale_fill_gradient2(low = "white", mid = "lightskyblue", high = "red", midpoint = 0.5, limit
s = c(0, 1)) +
  labs(title = "Fraud by Marital Status, Address Change Index, Living status",
       x = "Marital Status", y = "Address Change Ind", fill = "Proportion of Fraud") +
  theme_minimal() +
  theme(strip.background = element_blank(),
        strip.text.x = element_text(size = 12, face = "bold"),
        strip.text.y = element_text(size = 12, face = "bold"))
```



We further validate the key features of manual engineering by utilizing a rigorous statistical technique called Akaike Information Criterion (AIC) Model Comparison. To conduct AIC Model Comparison, we initially fitted a null model, containing only an intercept term, to the training dataset. Subsequently, we applied the stepwise model selection procedure, considering both forward and backward selection directions, to identify the best combination of predictors. During this process, the AIC criterion penalizes models with more parameters to avoid overfitting, leading to the dropping of variables that do not significantly improve the model's explanatory power. This repeated process led us to the final model, which included predictors of high_education_ind, past_num_of_claims, witness_present_ind, marital_status, address_change_ind, gender, living_status, and accident_site. Among these selected features, there are two new important features compared with our manually crucial factors: accident_site and past_num_of_claims, but generally the resultant features correspond to the insights derived from our manual feature engineering process. This dual approach of integrating manual feature engineering and AIC Model Comparison adds credibility to our analysis and findings.

```
#train_miao_aic1 <- train[complete.cases(train_miao_aic1), ]
#train_miao_aic1 <- train_miao_aic1 %>% filter(!is.nan(fraud)) %>% filter(!is.infinite(fraud))
%>% filter(!is.na(fraud))
#train_fit_null <- glm(fraud~1,data = train_miao_aic1)
#train_aic <- step(train_fit_null,scope = fraud ~ gender + marital_status + high_education_ind
+ address_change_ind + living_status + accident_site + past_num_of_claims + witness_present_in
d,direction="both")
```

Discussion

By manual engineering and AIC, we arrive at the final set of key predictors of fraudulent claims: high_education_ind, past_num_of_claims, witness_present_ind, marital_status, address_change_ind, gender, and living_status, age_of_driver.

Regarding the limitations, a major issue we encountered is the presence of missing values in the dataset. Approximately 0.5% of the 19,000 observations contain missing values. To mitigate this constraint, we may further implement appropriate missing value imputation techniques. The choice of the method (among mean, median, or model-based imputations) should depend on the characteristics of the missing data and the specific variable concerned.

The imbalanced dataset is another constraint in our EDA. We discovered a significant disparity between the number of “Fraud” and “No Fraud” instances, potentially leading to biased predictions when constructing predictive models. To overcome this problem, resampling techniques such as oversampling the minority class, undersampling the majority class, or employing synthetic data generation methods like the Synthetic Minority Over-sampling Technique (SMOTE) can be considered. These strategies aim to enhance the performance of the predictive models by counteracting the class imbalance problem.

Despite limitations such as missing values and class imbalance, our project also offers practical insights into insurance companies’ business logic. On the one hand, our identification of key attributes may assist insurance firms in risk assessment and pricing. With the knowledge of the factors contributing to insurance fraud, insurance providers can optimize their risk assessment models and adjust their pricing strategies. Consequently, agencies can formulate more accurate and competitive premiums, increasing customer satisfaction and retention.

On the other hand, our results can guide targeted investigations. Insurance firms can better deploy their investigation efforts by focusing on the most critical areas by identifying high-risk claims or claimants based on the data. This targeted approach can help minimize the overall costs associated with fraud investigations, generating significant savings for the insurance industry.

In conclusion, our exploratory data analysis generates valuable insights into the factors associated with insurance fraud while taking into account the limitations of missing values and class imbalance. These insights can counter fraudulent claims and guide insurance companies to enhance risk assessment, pricing strategies, and fraud investigation processes, ultimately benefiting both providers and policyholders—ourselves.

##Appendix NESS Statathon 2023, Traveler Insurance Company Fraud Detection
https://www.kaggle.com/competitions/2023-travelers-ness-statathon/data
(https://www.kaggle.com/competitions/2023-travelers-ness-statathon/data)