EDA

2023-11-02

GOAL of this project:

Exploratory data analysis

Source = https://www.kaggle.com/datasets/arashnic/imbalanced-df-practice/

The df is already segmented into a train set and a test set. My goal is to only touch the test df only after I have done EDA, trained a model.

```
library(ggplot2)
library(e1071)
library(corrplot)
```

```
## Warning: package ’corrplot’ was built under R version 4.3.2
```

```
## corrplot 0.92 loaded
```

```
library(randomForest)
```

```
## Warning: package ’randomForest’ was built under R version 4.3.2
```

```
## 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
```

```
df = read.csv('aug_train.csv')
head(df)
```

```
##       id Gender Age Driving_License Region_Code Previously_Insured Vehicle_Age
## 1 167647   Male  22               1           7                  1    < 1 Year
## 2  17163   Male  42               1          28                  0    1-2 Year
## 3  32023 Female  66               1          33                  0    1-2 Year
## 4  87447 Female  22               1          33                  0    < 1 Year
## 5 501933   Male  28               1          46                  1    < 1 Year
## 6 295775 Female  25               1          25                  1    < 1 Year
##   Vehicle_Damage Annual_Premium Policy_Sales_Channel Vintage Response
## 1             No           2630                  152      16        0
## 2            Yes          43327                   26     135        0
## 3            Yes          35841                  124     253        0
## 4             No          27645                  152      69        0
## 5             No          29023                  152     211        0
## 6             No          27954                  152      23        0
```

Description of features:

Gender: Describes the gender of a customer Age: Describes the age of a customer Driving_license: An binary feature which takes the value of 1 when the customer possess a driving license Region_code: The region a customer belongs to (Ranging from 0 to 52) Previously_Insured: A binary feature which takes the value of 1 when the customer possessed a vehicle insurance before. Vehicle_Age: Describes the age of a customer's vehicle Vehicle_Damage: A feature taking the values of either Yes or No according to if a customer's vehicle is damaged or not Annual_Premium: Amount of premium a customer will pay if they buy a vehicle insurance (Currency undefined) Policy_Sales_Channel: The unique identifier of referral agency Vintage: The number of days the customer has been associated with the company Response: A binary feature which takes the value of 1 when the customer purchased a vehicle insurance.

```r
#Dropping the id column since we already have dfframes indices as their unique ID
df = df[, !names(df) %in% 'id']
```

Finding any missing values

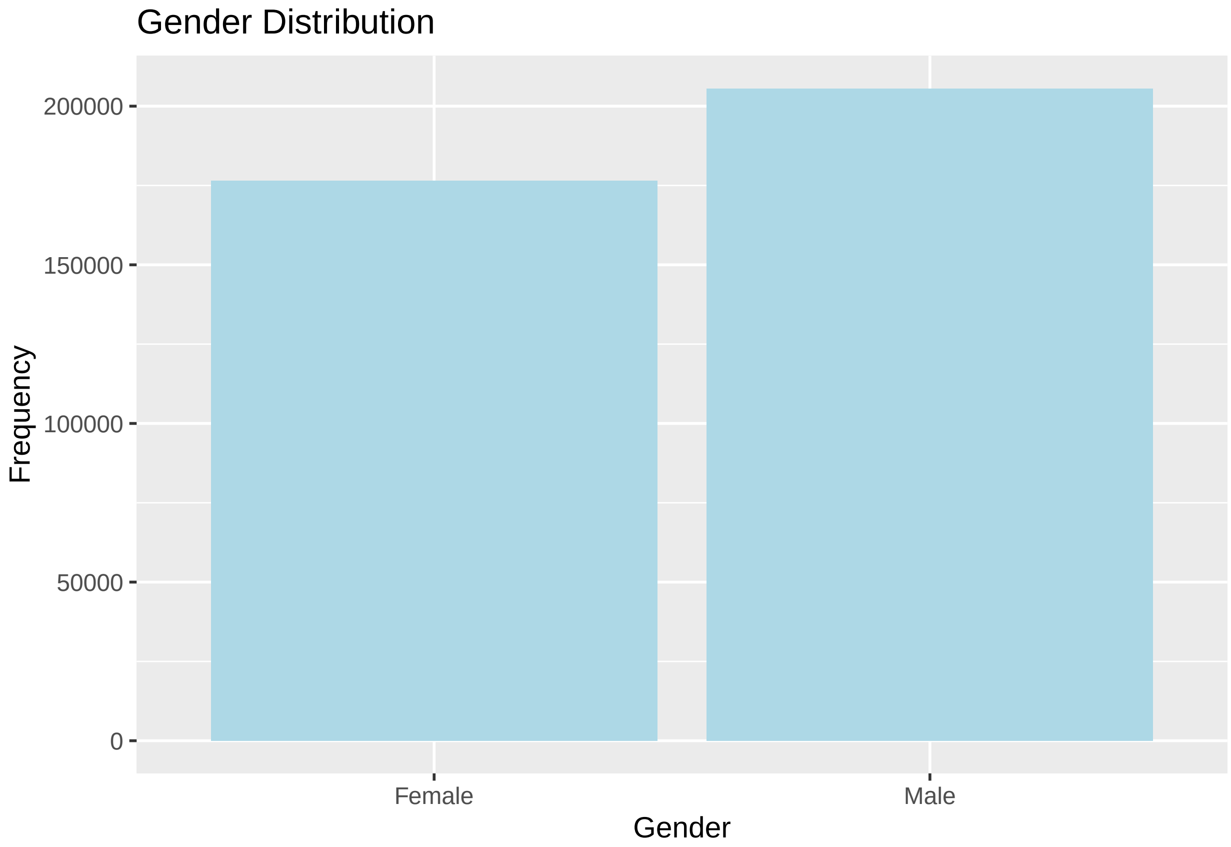
```r
#A 0 indicates that I have no missing values
print(sum(is.na(df)))
```

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

Univariate Analysis

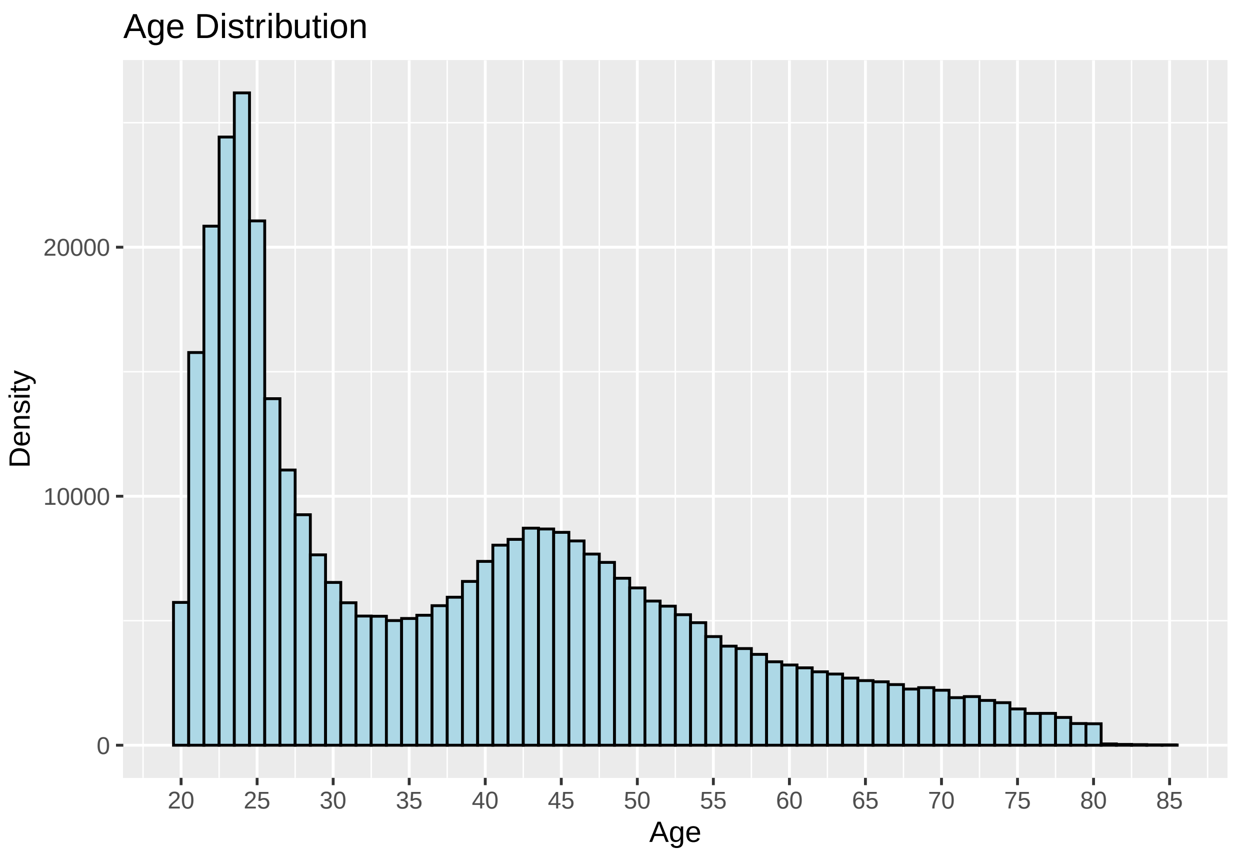
I will first study each feature's characterisitics individually.

```r
ggplot(df, aes(x = Gender)) + geom_bar(fill = "lightblue") +
  labs(title = "Gender Distribution", x = "Gender", y = "Frequency")
```

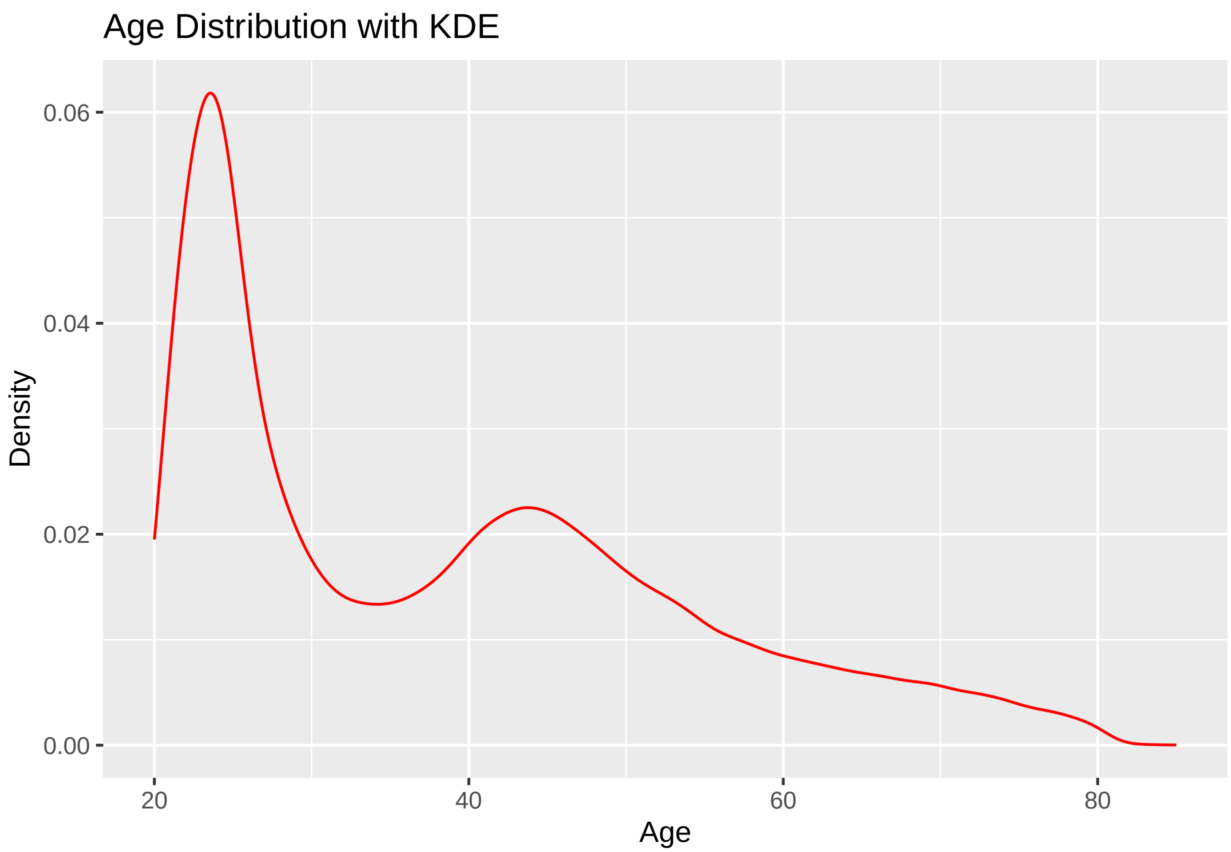


Comment: Gender seems to be well balanced. Although there are slightly more males in the data.

```r
ggplot(df, aes(x = Age)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "black") +
  labs(title = "Age Distribution",
       x = "Age",
       y = "Density") + scale_x_continuous(breaks = seq(0, 100, by = 5))
```



```r
ggplot(df, aes(x = Age)) +
  geom_density(color = "red") +
  labs(title = "Age Distribution with KDE",
       x = "Age",
       y = "Density")
```



Printing the summary statistics for Age:

```
summary(df$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   20.00   25.00   36.00   38.55   49.00   85.00
```

Outliers using the Z-score method with 3 standard deviations

```
df$Age_ZScore <- scale(df$Age)

#I chose 2
zscore_threshold <- 3
age_outliers <- df[abs(df$Age_ZScore) >= zscore_threshold, ]

print(age_outliers)
```

```
##        Gender Age Driving_License Region_Code Previously_Insured Vehicle_Age
## 5659     Male  85               1          28                  0    1-2 Year
## 43367  Female  85               1          28                  1    1-2 Year
## 150859 Female  85               1           8                  0    1-2 Year
## 238853 Female  85               1          28                  0    1-2 Year
## 267456 Female  85               0          28                  1    1-2 Year
## 294610 Female  85               1          50                  0    1-2 Year
## 306506 Female  85               1          28                  0    1-2 Year
## 350121 Female  85               1          28                 0    1-2 Year
## 351274 Female  85               1           8                 1    1-2 Year
## 362812   Male  85               1          48                 0    1-2 Year
##        Vehicle_Damage Annual_Premium Policy_Sales_Channel Vintage Response
## 5659              Yes          34005                  122     279        0
## 43367              No          42530                   26      64        0
## 150859            Yes          65268                  124     245        0
## 238853            Yes          32366                   26     293        0
## 267456             No          27057                   26      65        0
## 294610            Yes          26475                  124     114        0
## 306506            Yes          51045                  124     147        0
## 350121            Yes           2630                   26     290        0
## 351274             No          41080                    7     129        0
## 362812            Yes           2630                  124      37        0
##        Age_ZScore
## 5659     3.050806
## 43367    3.050806
## 150859   3.050806
## 238853   3.050806
## 267456   3.050806
## 294610   3.050806
## 306506   3.050806
## 350121   3.050806
## 351274   3.050806
## 362812   3.050806
```

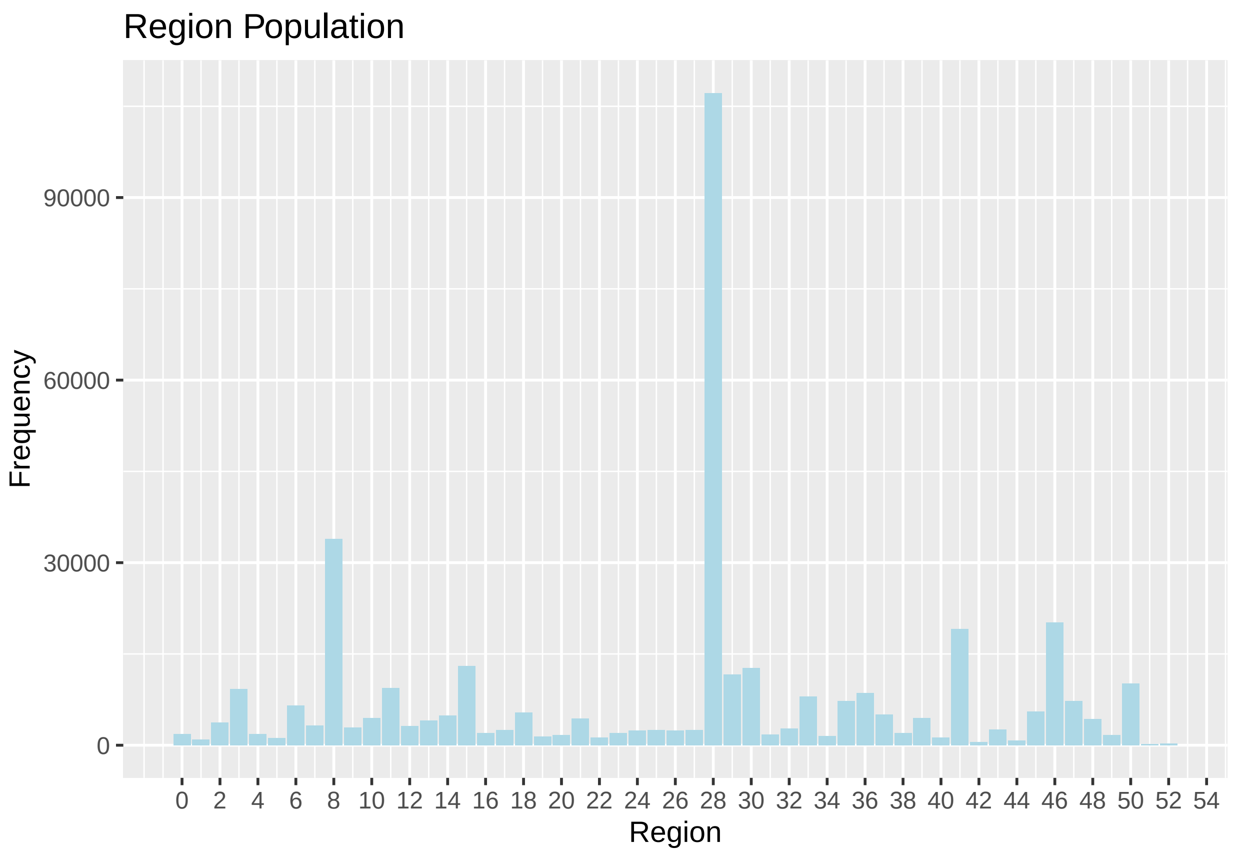
Comment (Age): Eyeballing the Age distribution, I can see that most of the customers are of age 25 and around. Tells me that most of the customers are young. There are also a significant number of customers in the range of 40-50. Using the Z-score method, I find there are outliers in Age, customers with age of 85.Age is an interesting variable because younger people are more likely to be in a car accident, but also since they are young and less wealth, they are less likely to buy insurance. Maybe this variable can be transformed using a log transformation since the distribution looks right skewed.

```
driving_license_proportions <- prop.table(table(df$Driving_License))
round(driving_license_proportions * 100, 2)
```

```
##
##     0     1
##  0.19 99.81
```

Comment (Driving_License): This feature tells me that 99.81% of the customer possess a drivings license. Since this feature has very low variance, this might cause convergence problems later when I am modelling

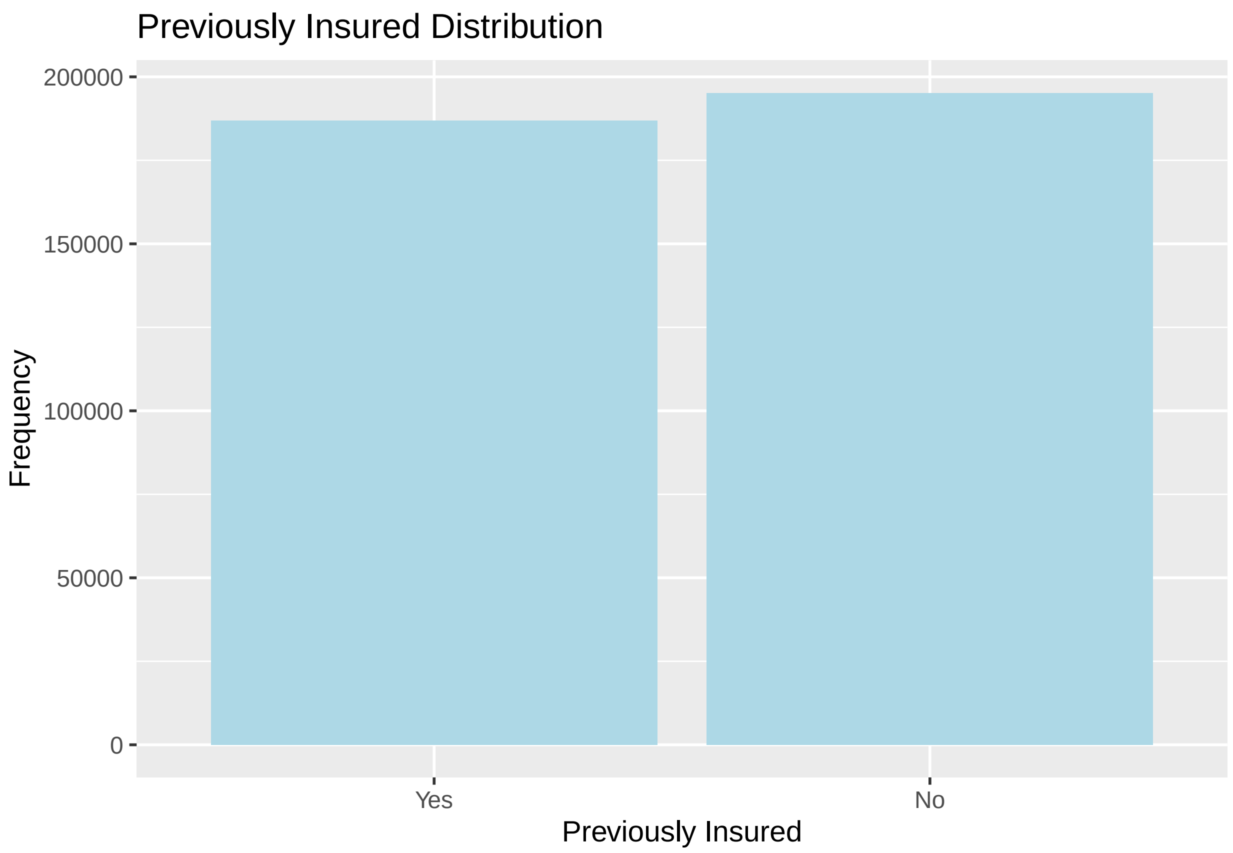
```
#df$Region_Code

ggplot(df, aes(x = Region_Code)) + geom_bar(fill = "lightblue") +
  labs(title = "Region Population", x = "Region", y = "Frequency") + scale_x_continuous(breaks = seq(0,
```



Comment (Region): The bar plot shows region disparity, where most of the customer belong to region 28. This can be due to various regions like economic disparity, or if the company is based in that region. Also, shows me that there is a marketing potential in region 28 since the company is more likely to acquire customers there.This plot also tells me that there are regions with very low population of customers which can later be combined into one category so that we dont come across low variance problem in their dummy. Specifically, “1” “42” “44” “51” “52”, are the regions which have less than a 1000 customers.

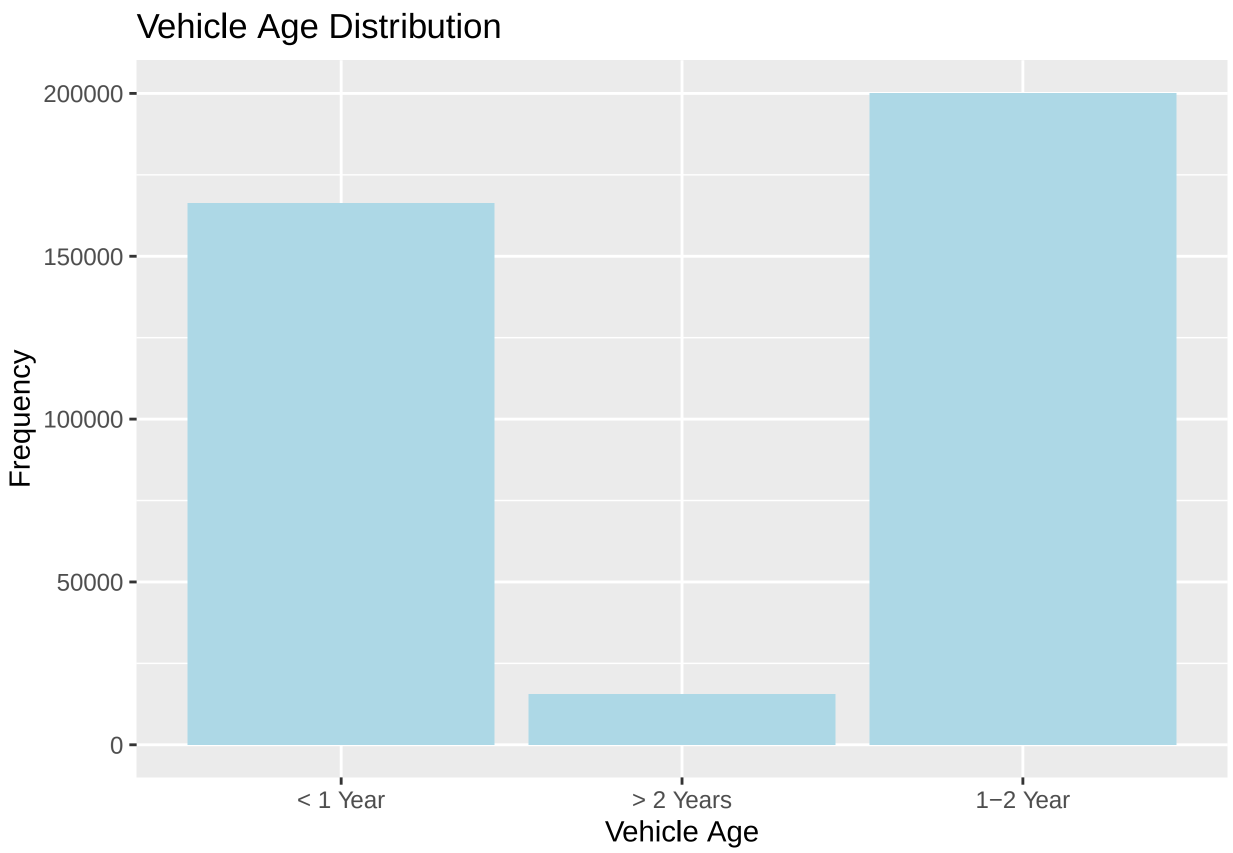
I am assuming there will be relation with Policy_sales_channel feature since it might be because most of these sales channel operate in region 28.

```
ggplot(df, aes(x = factor(Previously_Insured, levels = c(1, 0), labels = c("Yes", "No")))) + geom_bar(fi
  labs(title = "Previously Insured Distribution", x = "Previously Insured", y = "Frequency")
```



Comment (Previously Insured): This feature is also well balanced, with almost equal proportions of customers having a vehicle insurance in the past.

```r
ggplot(df, aes(x = Vehicle_Age)) + geom_bar(fill = "lightblue") +
  labs(title = "Vehicle Age Distribution", x = "Vehicle Age", y = "Frequency")
```

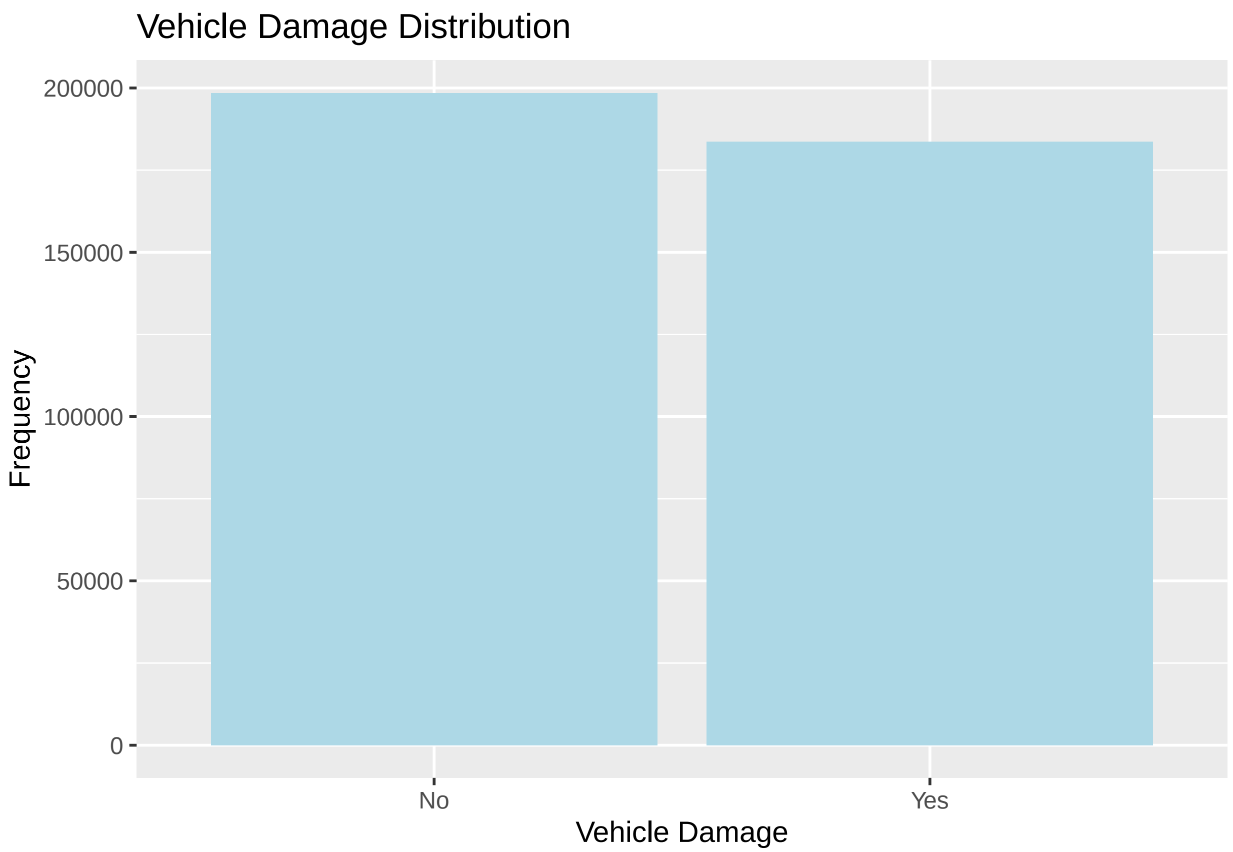


```r
round(prop.table(table(df$Vehicle_Age)) * 100, 2)
```

```
##
##  < 1 Year > 2 Years  1-2 Year
##     43.53      4.09     52.38
```

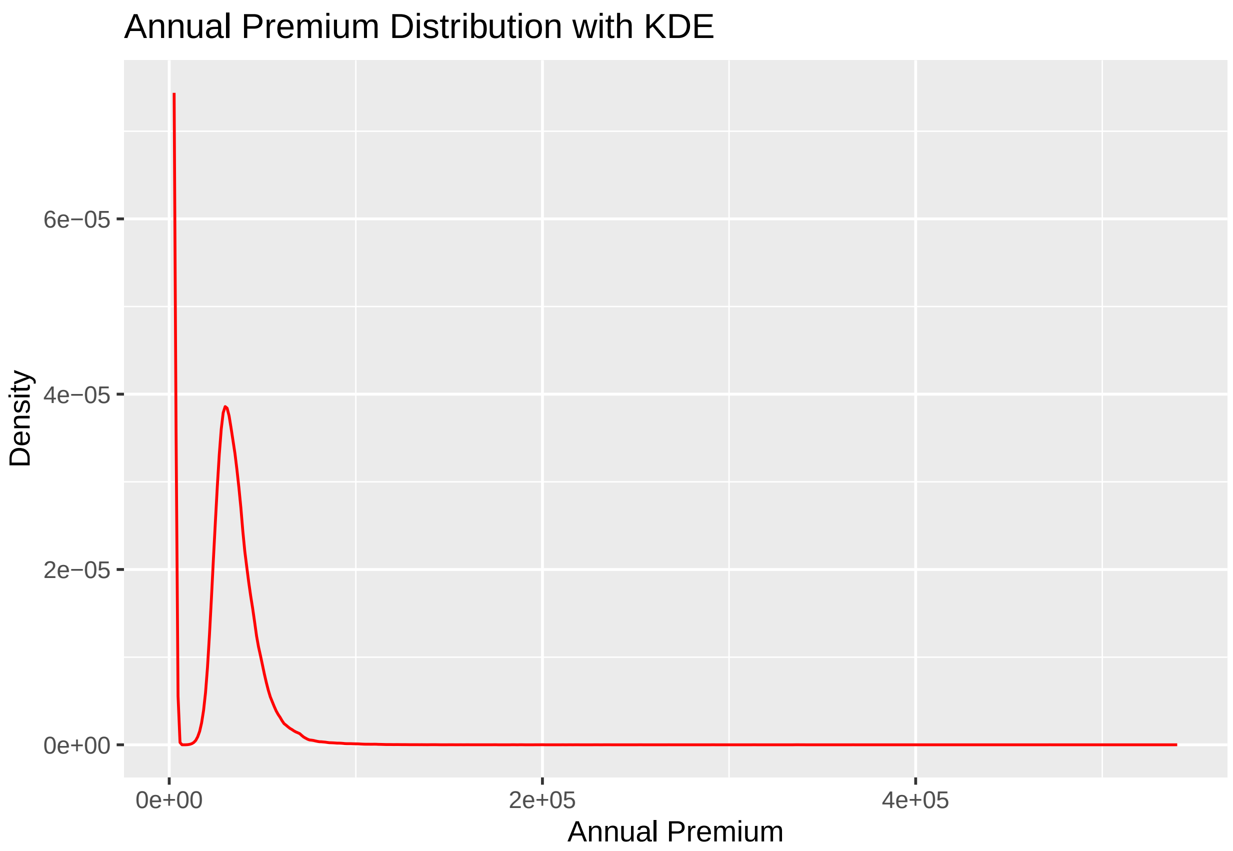
Comment (Vehicle Age): This feature shows the most of the customers' cars are new. Because only 4.1% of the customers possess cars that are greater than 2 years old while 43.5 of the customers have cars of age less than 1 year. Furthermore, majority of the customers (52.4%) have cars of age between 1 to 2 years. This feature holds importance since, people with newer cars are more likely to buy car insurance since the expected loss should be higher.

```r
ggplot(df, aes(x = Vehicle_Damage)) + geom_bar(fill = "lightblue") +
  labs(title = "Vehicle Damage Distribution", x = "Vehicle Damage", y = "Frequency")
```

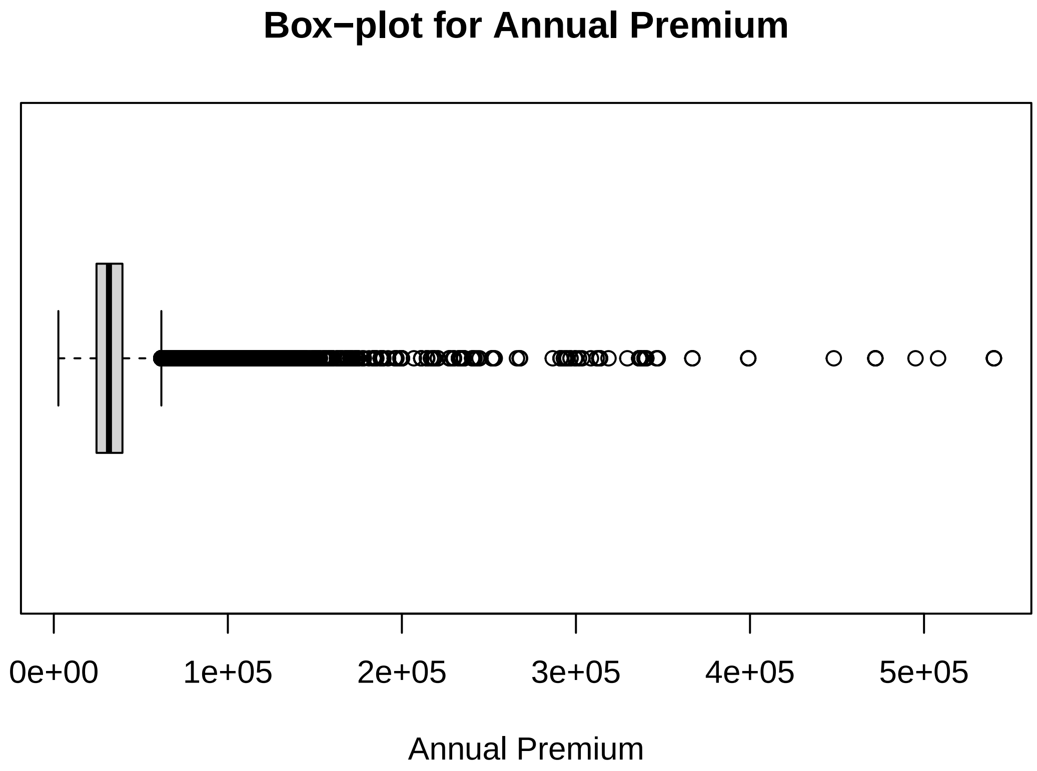


Comment (Vehicle Damage): The feature is also a well balanced feature. And it also relates to how someone possessing a damaged vehicle might be more likely to buy an insurance since they have gone through bad experience/s in the past.

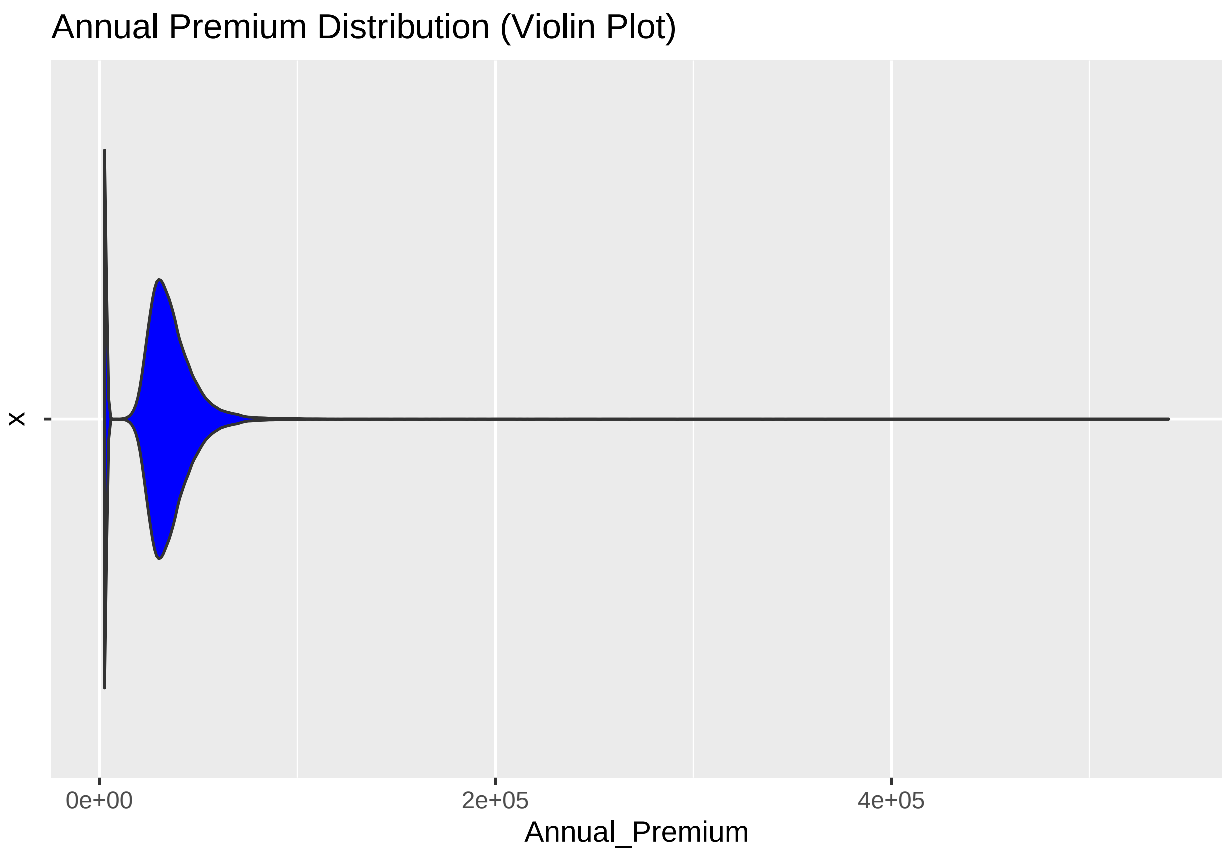
```
ggplot(df, aes(x = Annual_Premium)) +
  geom_density(color = "red") +
  labs(title = "Annual Premium Distribution with KDE",
       x = "Annual Premium",
       y = "Density")
```



```
boxplot(df$Annual_Premium, horizontal=TRUE, main="Box-plot for Annual Premium", xlab="Annual Premium")
```



```r
ggplot(df, aes(x = "", y = Annual_Premium)) +
  geom_violin(fill = "blue") + coord_flip() +
  labs(title = "Annual Premium Distribution (Violin Plot)")
```



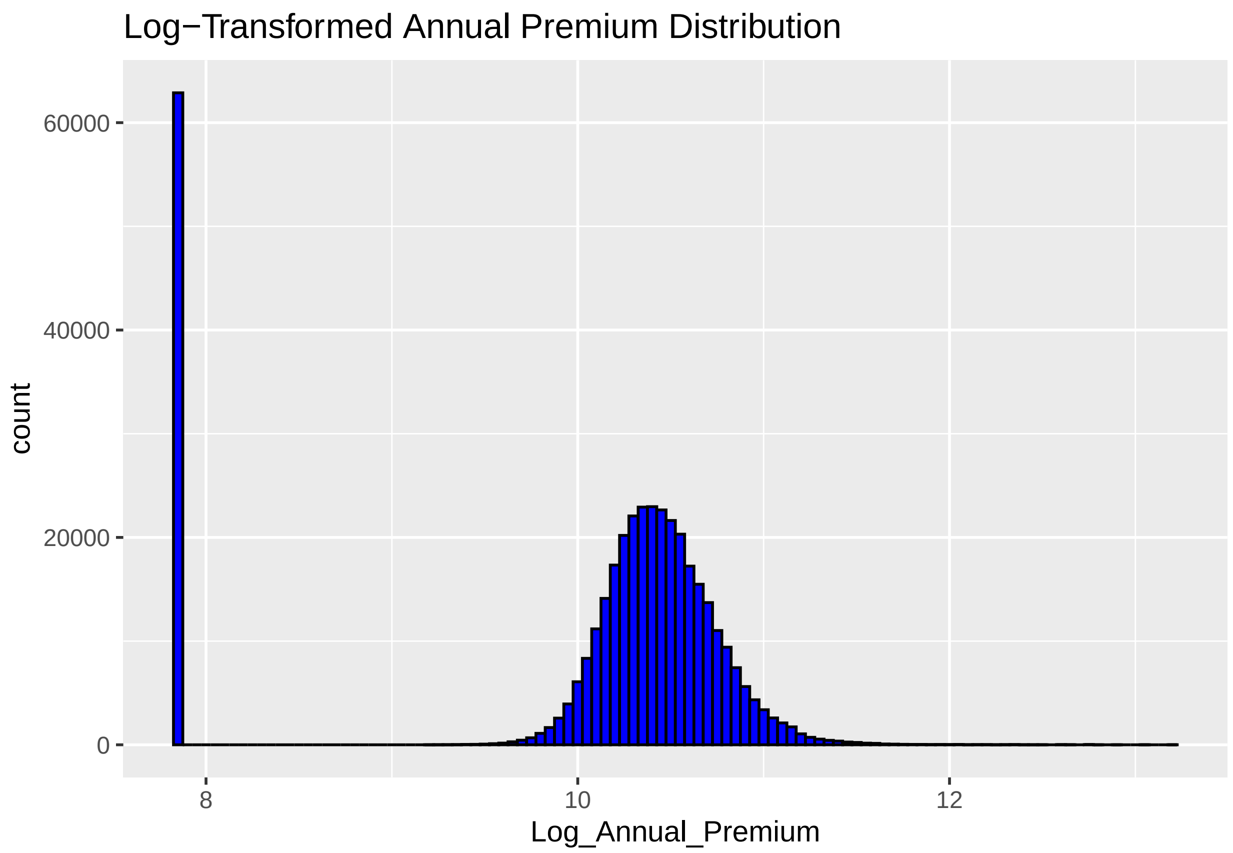
```
skew_value <- skewness(df$Annual_Premium)
print("Skewness Statistics")
```

```
## [1] "Skewness Statistics"
```

```
print(skew_value)
```

```
## [1] 1.720438
```

```
df$Log_Annual_Premium <- log(df$Annual_Premium)
ggplot(df, aes(x = Log_Annual_Premium)) +
  geom_histogram(binwidth = 0.05, fill = "blue", color = "black") +
  labs(title = "Log-Transformed Annual Premium Distribution")
```



```
skew_value <- skewness(df$Annual_Premium)
print(skew_value)
```

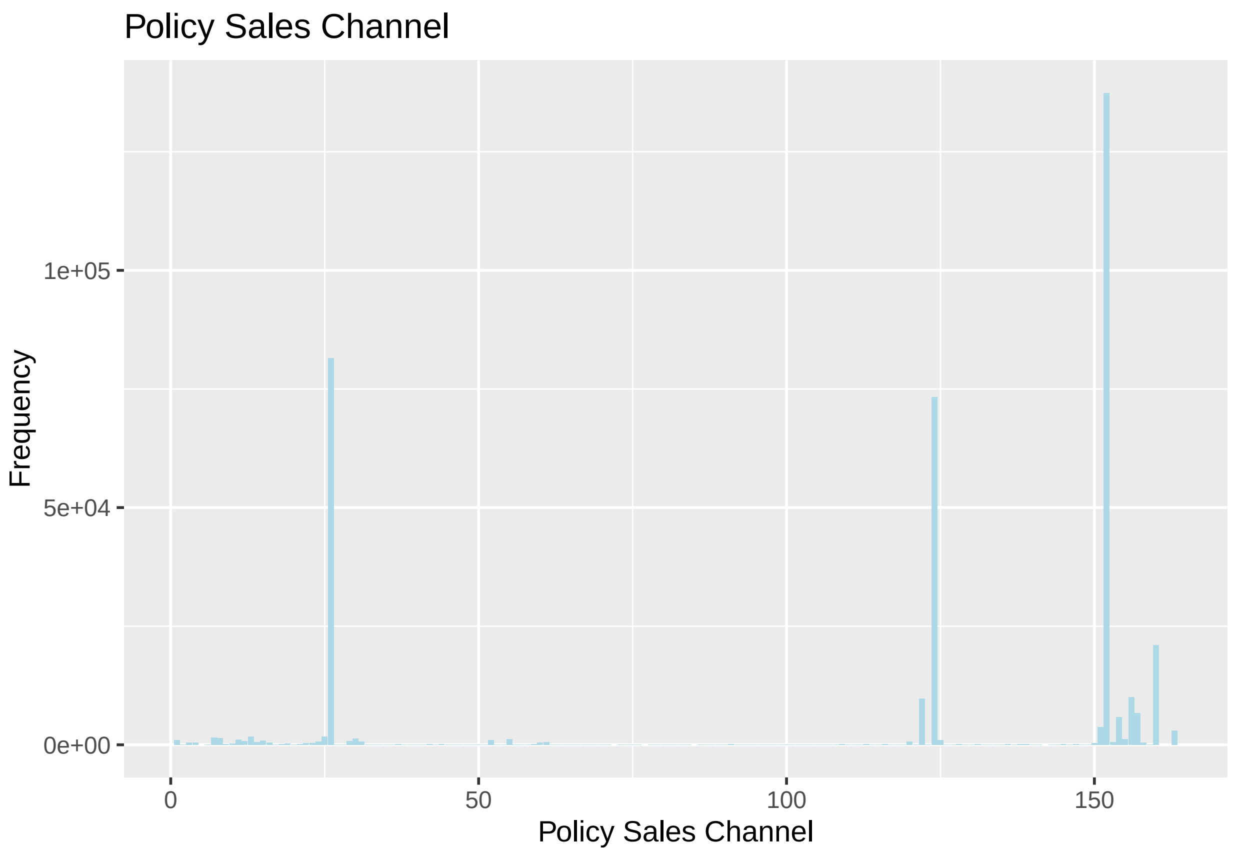
```
## [1] 1.720438
```

Comment (Annual Premium): The feature represents the amount the customer will pay for the vehicle insurance per year. Plotting the histogram for annual premium showed very weird results so I resorted to plotting Kernel Density Estimates, a violin plot and a box plot. All of them showed the the data was highly skewed (Right). Furthermore, I also found the skewness of this feature and it was positive, also indicating a right skew.

Since it is a best practice to transform features so that they follow a normal distribution, I applied a log transformation and the result was a nice normal distribution. The reason to transform is that the original feature has extreme values which might influence the predictive power of the model and if I select variable, this variable might get chosen because of the extreme effect from its extreme values. In other words, reducing the effects of outlier values. This also helps in making the relationship between log-odds and the feature "more linear".

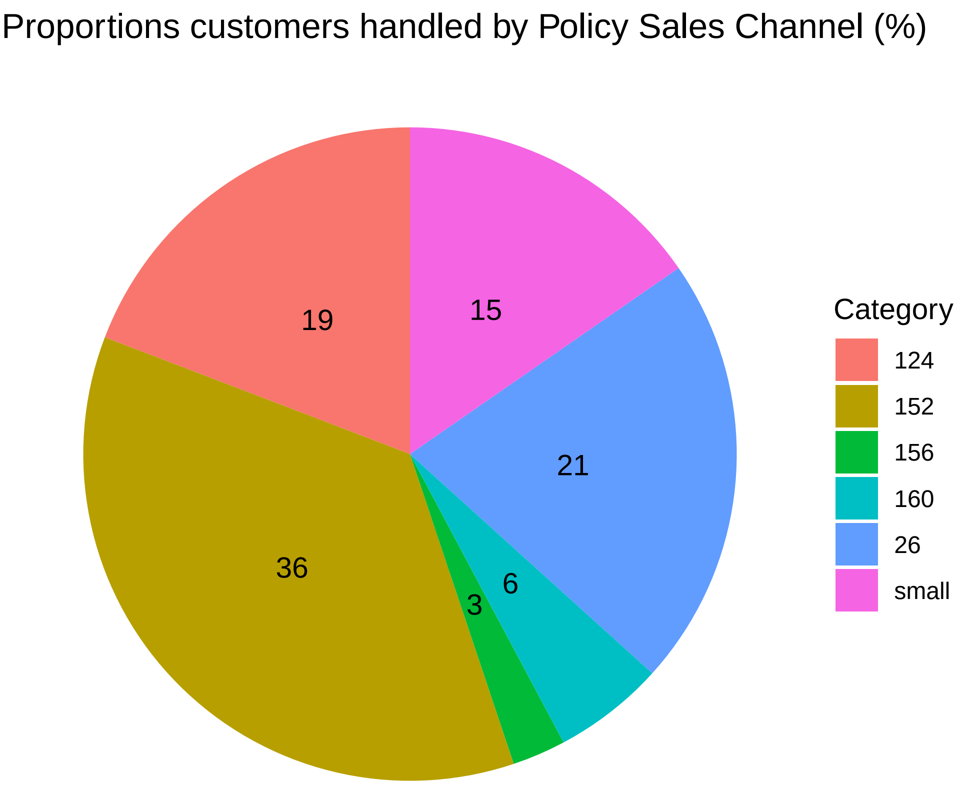
Its important to note that now the interpretation of the coefficient of this new feature has changed. But since I am trying to predict instead of inferring, I am not concerned with the interpretations.

```
#Policy_Sales_Channel
ggplot(df, aes(x = Policy_Sales_Channel)) + geom_bar(fill = "lightblue") +
  labs(title = "Policy Sales Channel", x = "Policy Sales Channel", y = "Frequency")
```



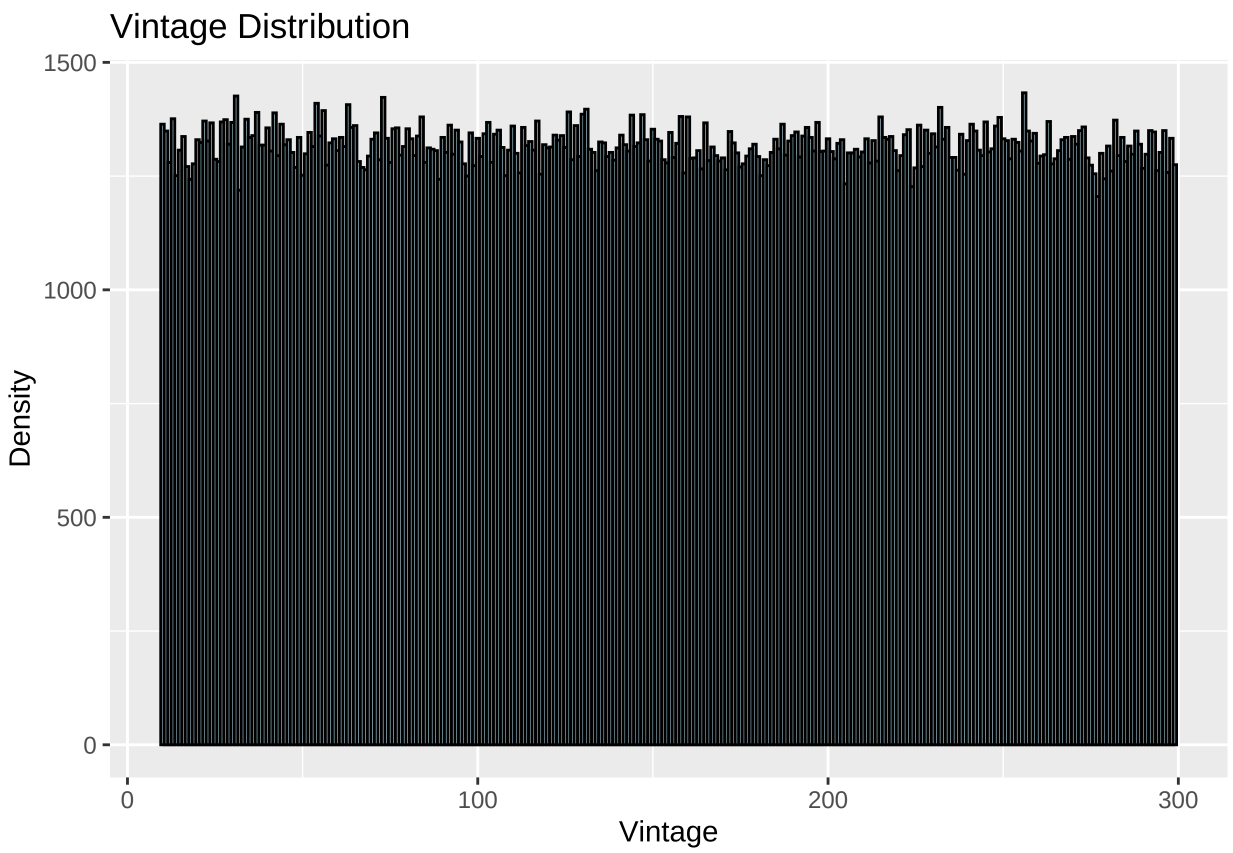
```r
df$new_cat_policy_channel <- df$Policy_Sales_Channel

channel_counts <- table(df$Policy_Sales_Channel)
small_channels <- as.numeric(names(channel_counts[channel_counts < 10000]))

df$new_cat_policy_channel[df$new_cat_policy_channel %in% small_channels] <- "small"
```

```r
category_proportions <- prop.table(table(df$new_cat_policy_channel))


pie_data <- data.frame(
  Category = names(category_proportions),
  Proportion = category_proportions
)

 ggplot(pie_data, aes(x = "", y = Proportion.Freq, fill = Category, label = scales::percent(Proportion.F
  geom_bar(stat = "identity") +
  geom_text(aes(label = as.character(round(Proportion.Freq * 100))), position = position_stack(vjust = 0
  coord_polar("y") +
  labs(title = "Proportions customers handled by Policy Sales Channel (%)") +
  theme_void()  # Remove unnecessary elements
```

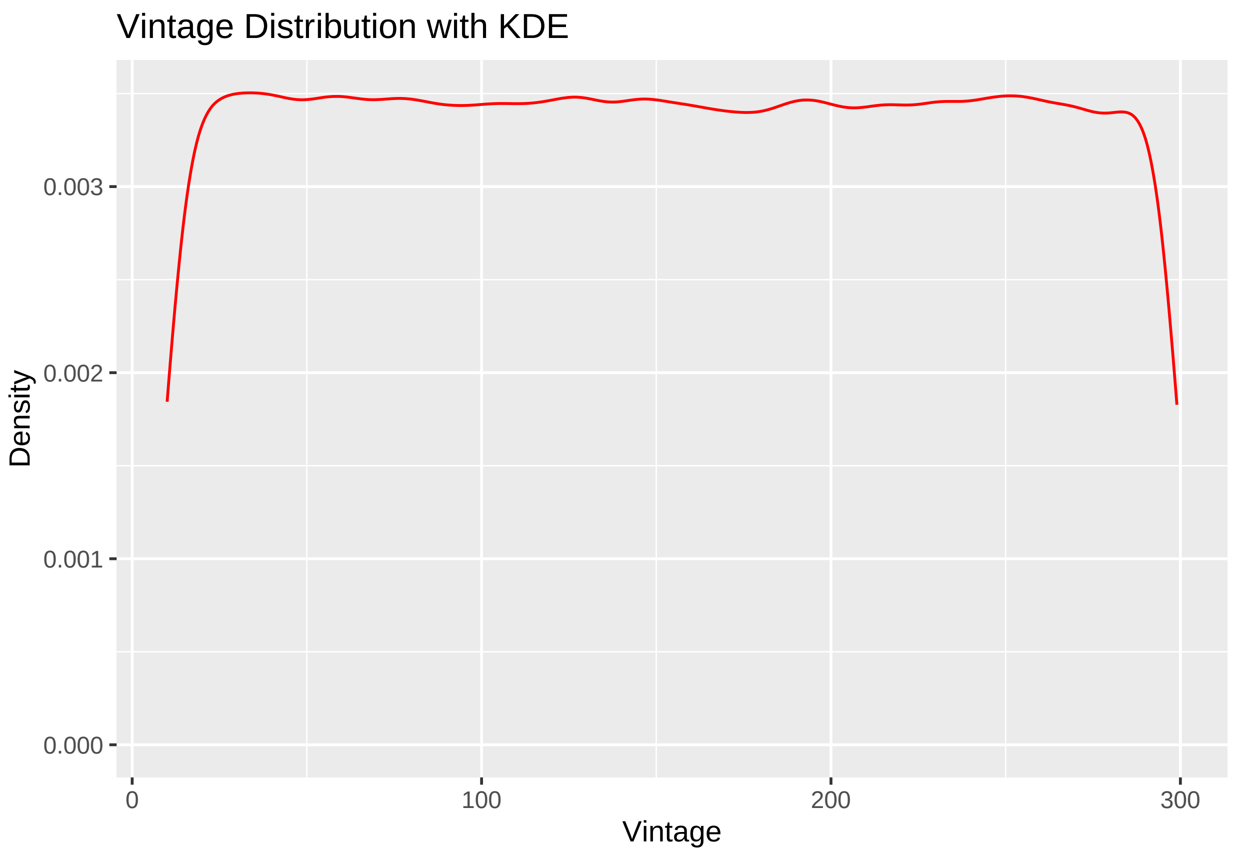


Comment (Policy_Sales_Channel): I can see that there are only 3 major contributers, 124, 152, and 26. Grouping all the sales channel with less than 10000 sales, I can generate the pie chart above. Sales channels in general act as a good predictor of whether you can sell a person an insurance but with categories with very little variance, this might cause a problem later. If it does, I would have to combine the small categories like I did in this piechart.

```r
ggplot(df, aes(x = Vintage)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "black") +
  labs(title = "Vintage Distribution",
       x = "Vintage",
       y = "Density")
```



```r
ggplot(df, aes(x = Vintage)) +
  geom_density(color = "red") +
  labs(title = "Vintage Distribution with KDE",
       x = "Vintage",
       y = "Density")
```

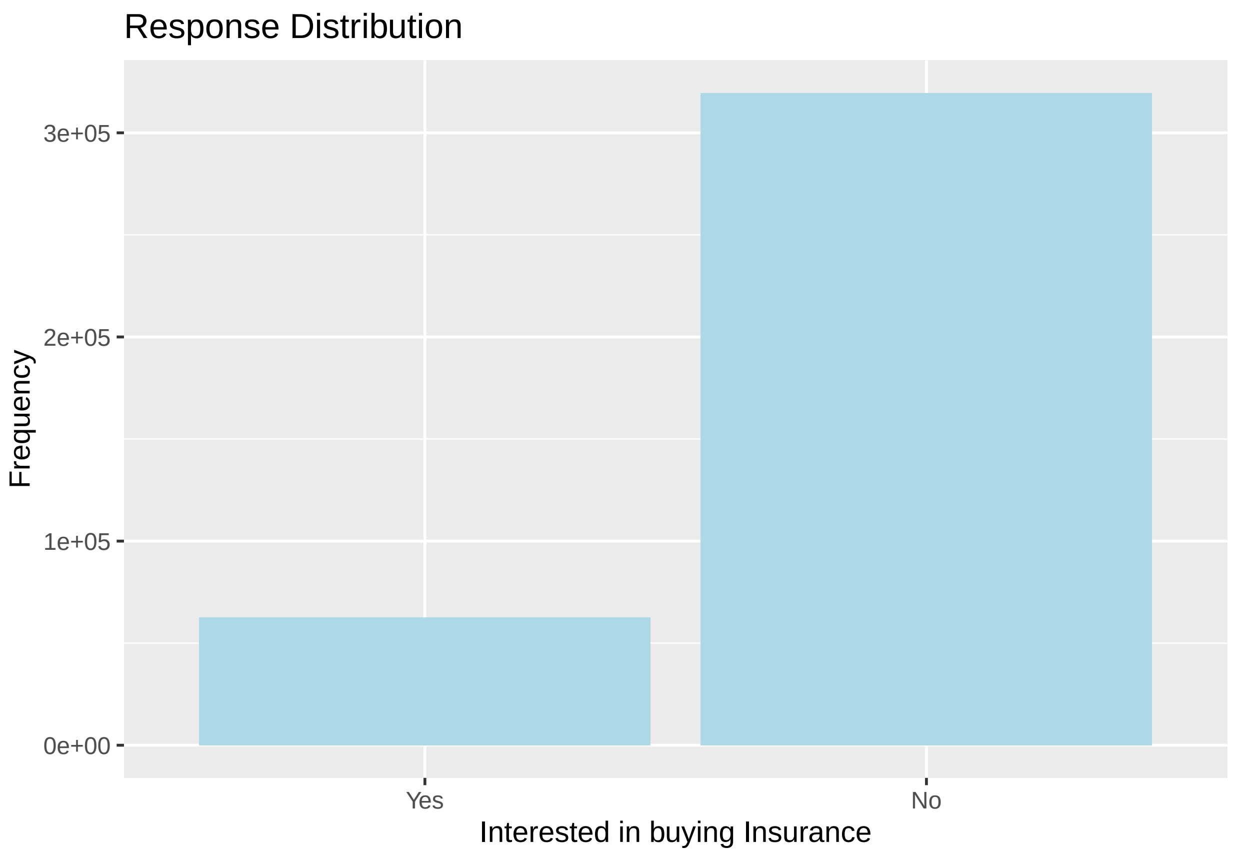


```
#Outliers
df$Vintage_ZScore <- scale(df$Vintage)

#I chose 2
zscore_threshold <- 2
Vintage_outliers <- df[abs(df$Vintage_ZScore) >= zscore_threshold, ]

print(Vintage_outliers)
```

```
##  [1] Gender                 Age                    Driving_License
##  [4] Region_Code            Previously_Insured     Vehicle_Age
##  [7] Vehicle_Damage         Annual_Premium         Policy_Sales_Channel
## [10] Vintage                Response               Age_ZScore
## [13] Log_Annual_Premium     new_cat_policy_channel Vintage_ZScore
## <0 rows> (or 0-length row.names)
```

Comment (Vintage): This is an interesting feature, as it seems to be uniformly distributed with no outliers according to the Zscore method. Since there is very little variation, all values are equally likely, it shouldnt add much to the predictive accuracy of the model. Which means it maybe possible to leave this variable out of the model (Prime Candidate).

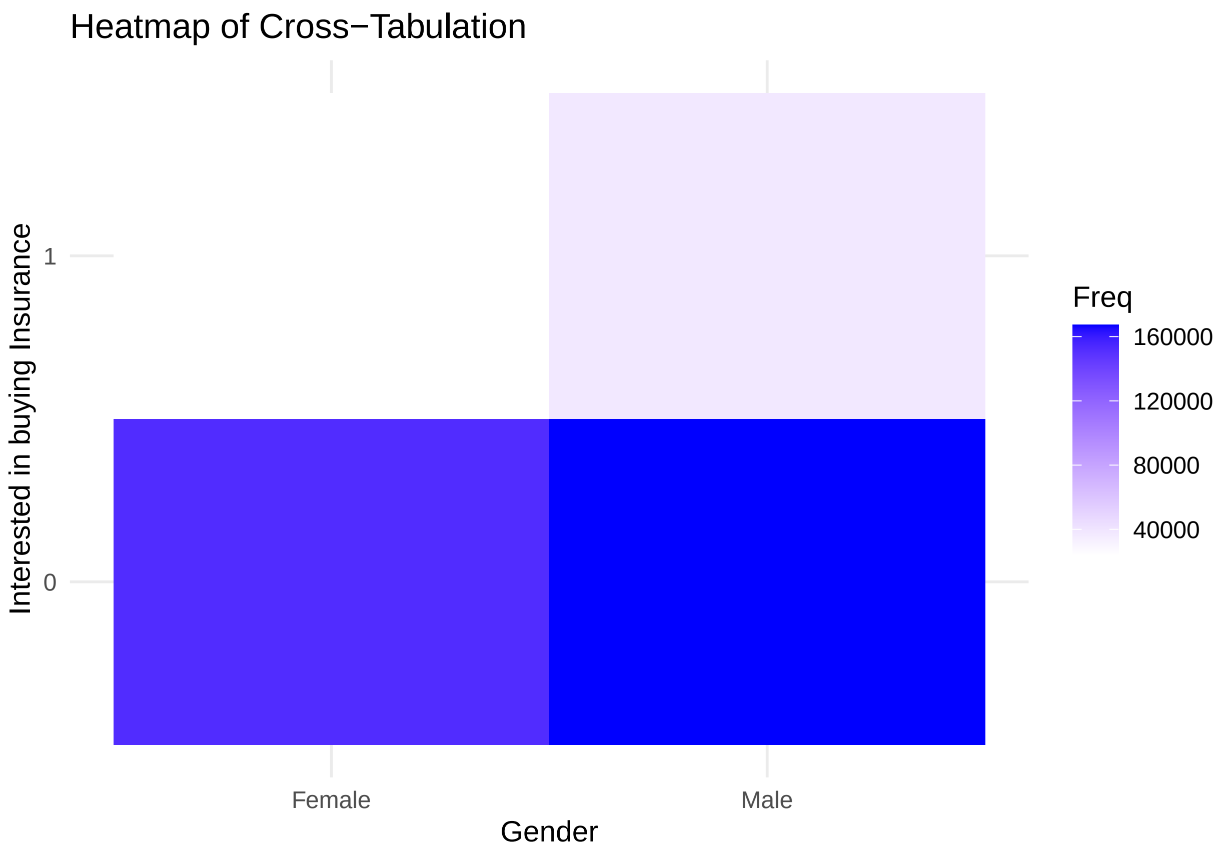
```
ggplot(df, aes(x = factor(Response, levels = c(1, 0), labels = c("Yes", "No")))) + geom_bar(fill = "lig
  labs(title = "Response Distribution", x = "Interested in buying Insurance", y = "Frequency")
```



Comment (Response): The response is highly imbalanced and is likely to make the model accurately classify 0s or No. This bias in the response can affect the models' performance. Especially their calibration. My goal is to ensure that whatever the output of the model is, it is based on the data so that when calculating expected profits, I am using probabilities and not scores.

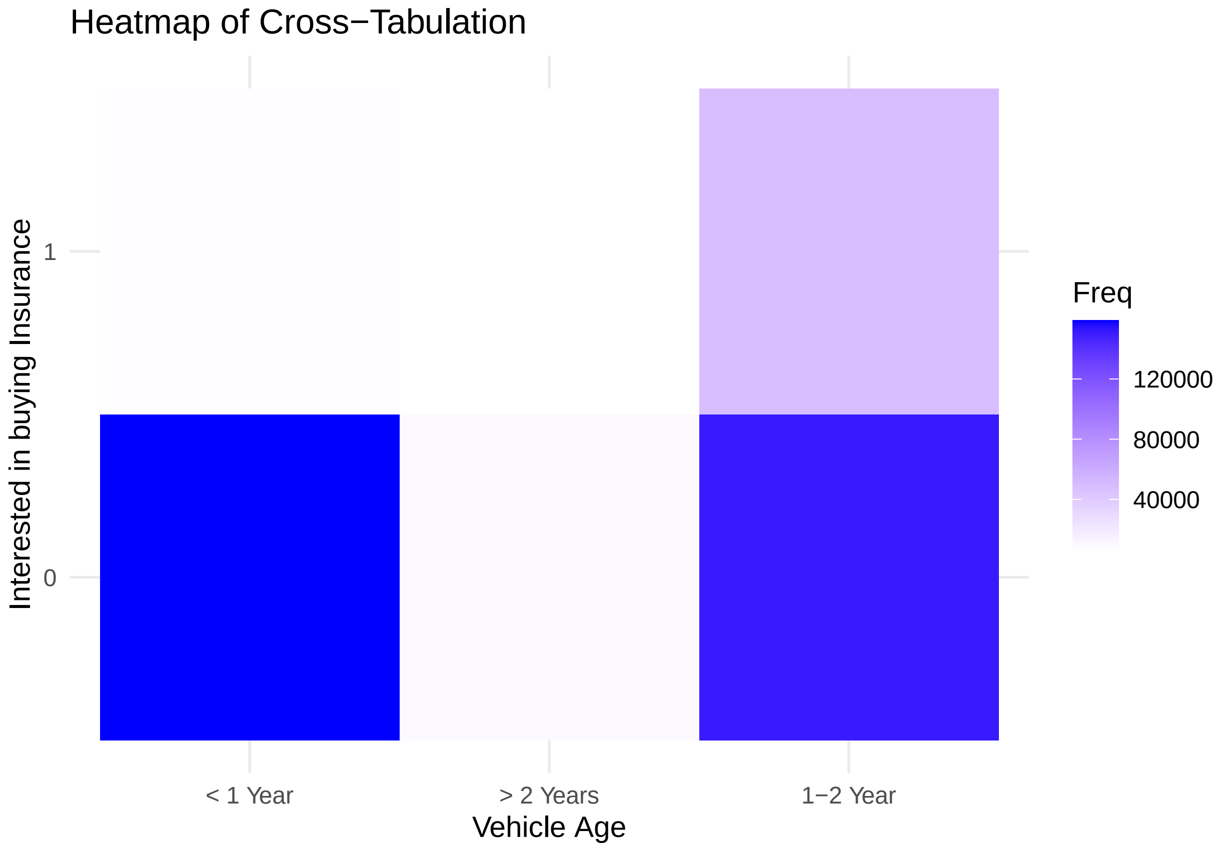
Bivariate analysis

```r
cross_tabulation <- as.data.frame(table(df$Gender, df$Response))

heatmap_plot <- ggplot(cross_tabulation, aes(x = Var1, y = Var2, fill = Freq)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "blue") +  # Define color scale
  labs(title = "Heatmap of Cross-Tabulation", x='Gender', y='Interested in buying Insurance') +
  theme_minimal()  # Customize the theme as needed

heatmap_plot
```



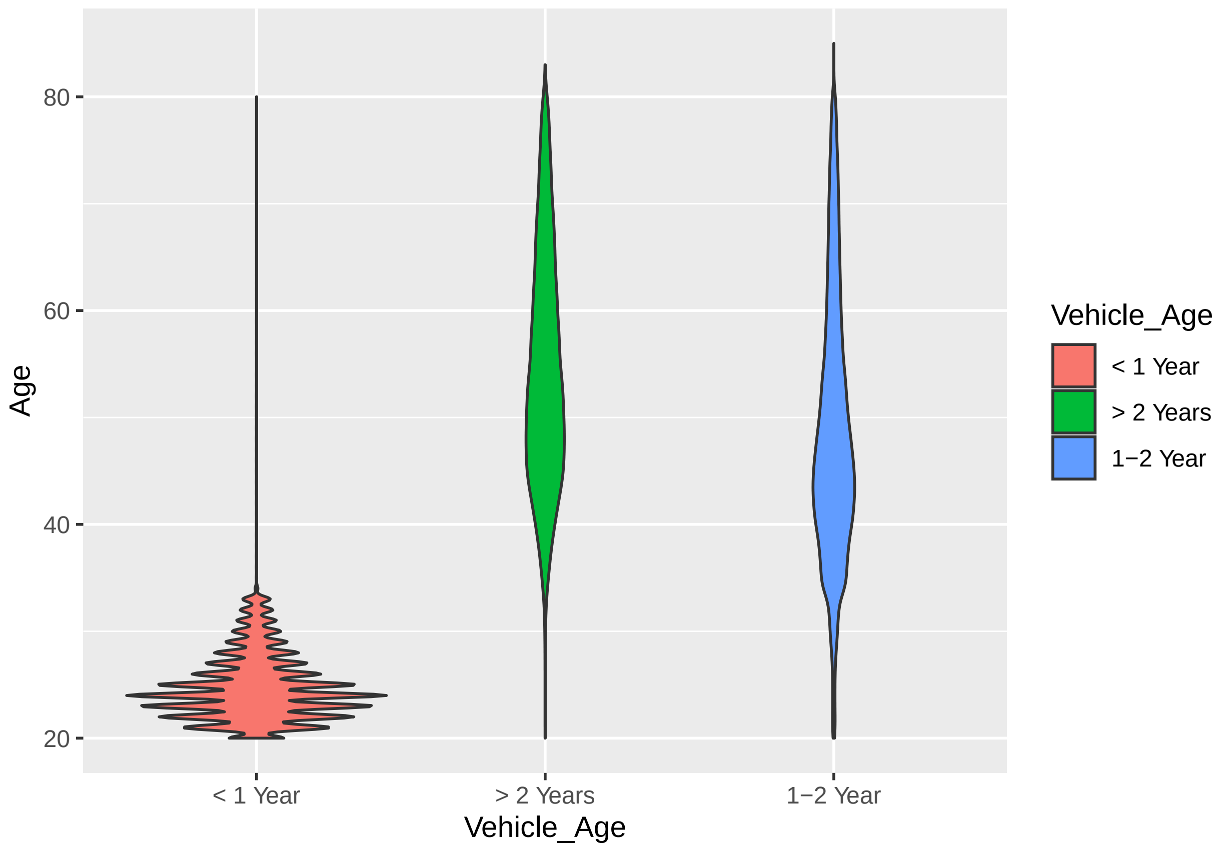
Comment: Gender seems to be like an important feature as I can see that males are more likely to buy a vehicle insurance. Also points towards the presence of potential confounding variables.

```
cross_tabulation <- as.data.frame(table(df$Vehicle_Age, df$Response))

heatmap_plot <- ggplot(cross_tabulation, aes(x = Var1, y = Var2, fill = Freq)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "blue") +  # Define color scale
  labs(title = "Heatmap of Cross-Tabulation", x='Vehicle Age', y='Interested in buying Insurance') +
  theme_minimal()  # Customize the theme as needed

heatmap_plot
```



Comment: Most people with car with age between 1 to 2 years are interested in buying insurance. This indicates that Vehicle age might be an important feature

```r
ggplot(df, aes(x = Vehicle_Age, y = Age, fill = Vehicle_Age)) +
  geom_violin()
```



Comment: It is interesting to see that most of the newer cars are owned by younger people. Which contradicts the general notion of young people not being rich enough to afford new cars. This again can point towards confounding variables.

Feature Importance

I will use a random forest to rank all the features according to their importance. Features which are present at the top of a tree, tends to be more explainatory hence having more importance.

```
df$Response <- as.factor(df$Response)
```

```
rf_model <- randomForest(Response ~ ., data = df, ntree = 100)
```

```
feature_importance <- importance(rf_model)
sorted_feature_importance <- feature_importance[order(-feature_importance), ]
sorted_feature_importance
```

```
##        Vehicle_Damage     Previously_Insured         Vintage_ZScore
##            9987.65826             9583.04267             6277.18057
##               Vintage         Annual_Premium     Log_Annual_Premium
##            6275.50424             6164.14940             6162.27304
##            Age_ZScore                    Age            Region_Code
##            5994.88872             5813.48440             4494.28631
##   Policy_Sales_Channel            Vehicle_Age new_cat_policy_channel
##            3326.57600             2920.12245             1958.44060
##                Gender        Driving_License
##             894.30694               39.79387
```