# Credit Risk Prediction using R

## 1. Introduction

Credit risk modeling is crucial in the banking and finance sectors to determine the probability of a customer defaulting on a loan. This project uses R to build a logistic regression model on the German Credit dataset to predict default status based on customer attributes.

## 2. Objective

- Load and clean the German Credit dataset  
- Explore data distribution and variable correlation  
- Build a logistic regression model  
- Evaluate model performance using confusion matrix and ROC curve  
- Optimize decision threshold and analyze feature importance

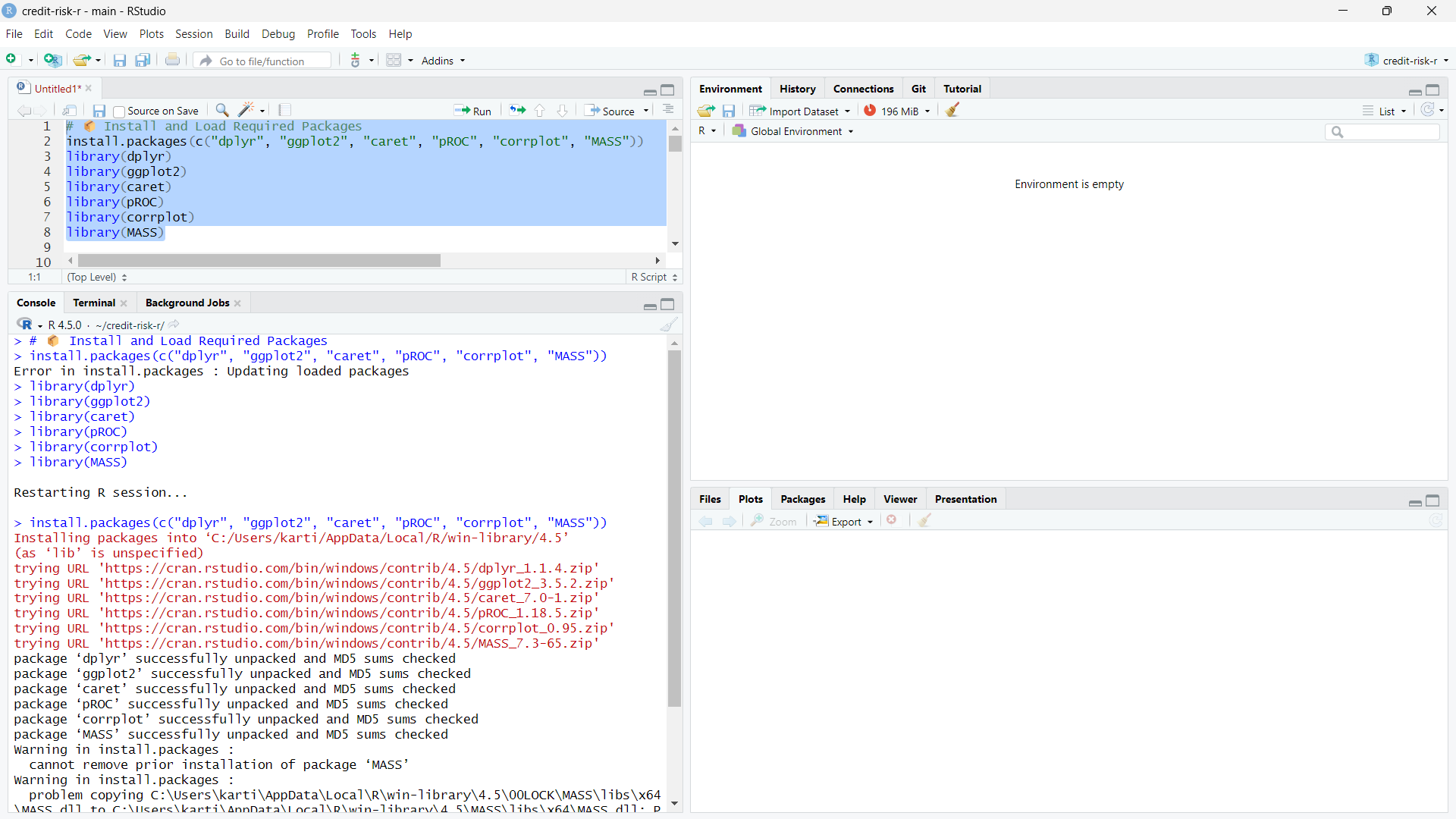
## 3. Prerequisites

Install the following R packages:

* - MASS  
  - dplyr  
  - ggplot2  
  - corrplot  
  - caret  
  - pROC

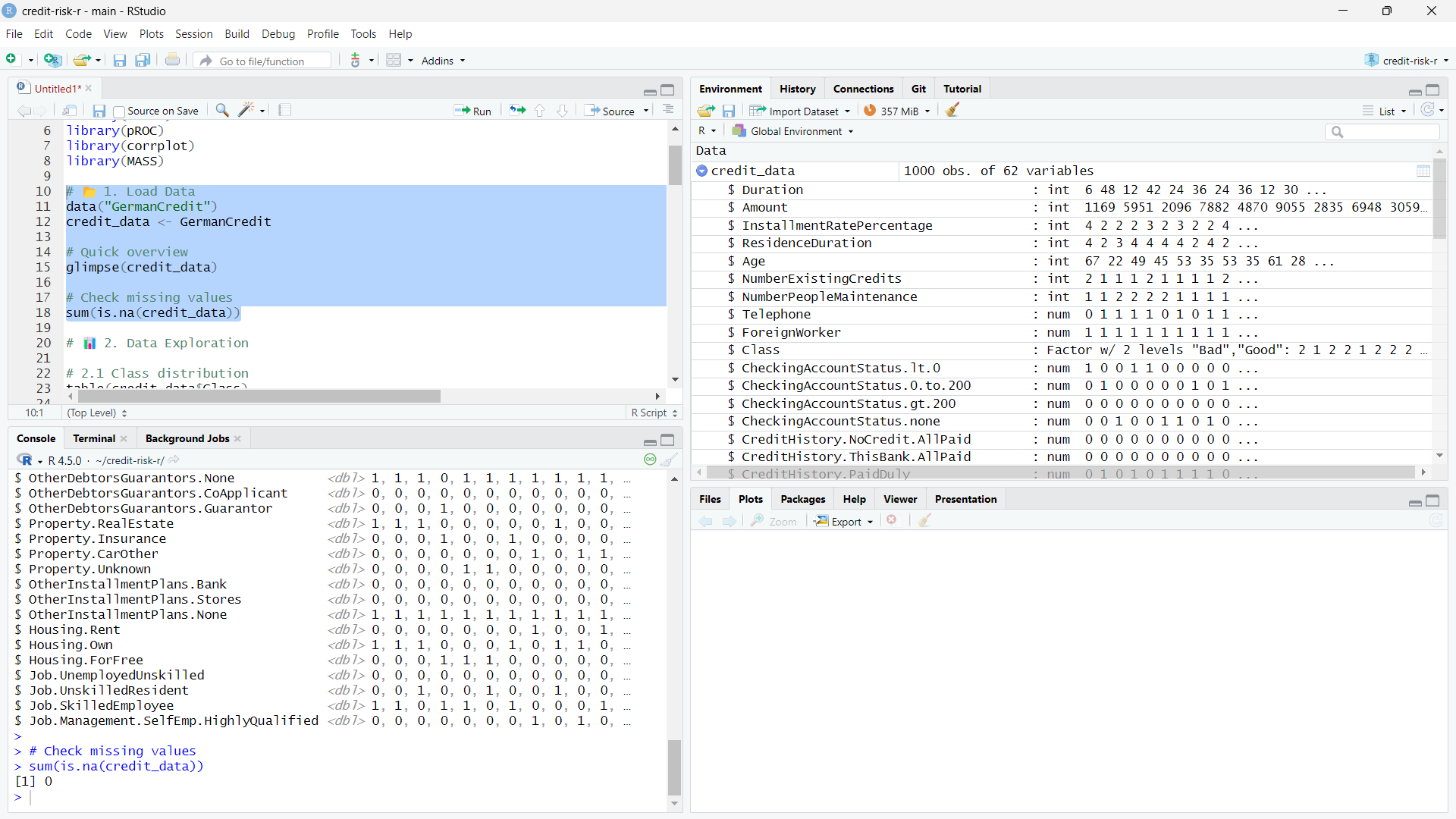
## 4. Load Libraries

```R  
library(MASS)  
library(dplyr)  
library(ggplot2)  
library(caret)  
library(pROC)  
library(corrplot)  
```



## 5. Load and Explore Data

```R  
data("GermanCredit")  
credit\_data <- GermanCredit  
glimpse(credit\_data)  
sum(is.na(credit\_data))  
```



## 6. Data Exploration

```R  
table(credit\_data$Class)  
prop.table(table(credit\_data$Class))  
ggplot(credit\_data, aes(x = Class, y = Amount, fill = Class)) +  
 geom\_boxplot() +  
 theme\_minimal()  
```

## 7. Correlation Matrix

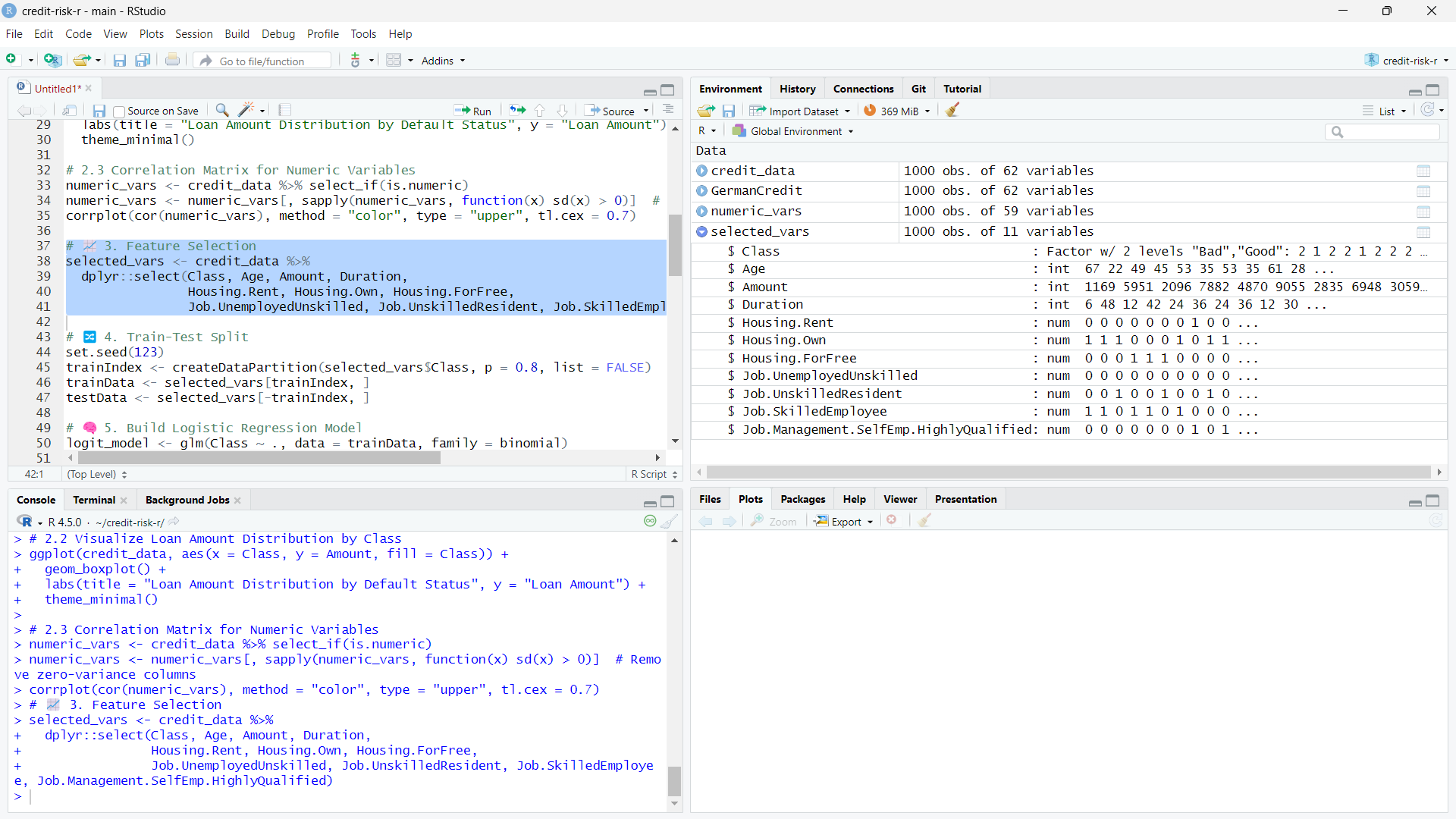
```R  
numeric\_vars <- credit\_data %>% select\_if(is.numeric)  
numeric\_vars <- numeric\_vars[, sapply(numeric\_vars, function(x) sd(x) > 0)]  
corrplot(cor(numeric\_vars), method = "color", type = "upper")  
```

A screenshot of a computer

AI-generated content may be incorrect.

## 8. Logistic Regression Model

```R  
selected\_vars <- credit\_data %>% select(Class, Age, Amount, Duration,  
 Housing.Rent, Housing.Own, Housing.ForFree,  
 Job.UnemployedUnskilled, Job.UnskilledResident,  
 Job.SkilledEmployee, Job.Management.SelfEmp.HighlyQualified)



set.seed(123)  
trainIndex <- createDataPartition(selected\_vars$Class, p = 0.8, list = FALSE)  
trainData <- selected\_vars[trainIndex, ]  
testData <- selected\_vars[-trainIndex, ]

A screenshot of a computer

AI-generated content may be incorrect.

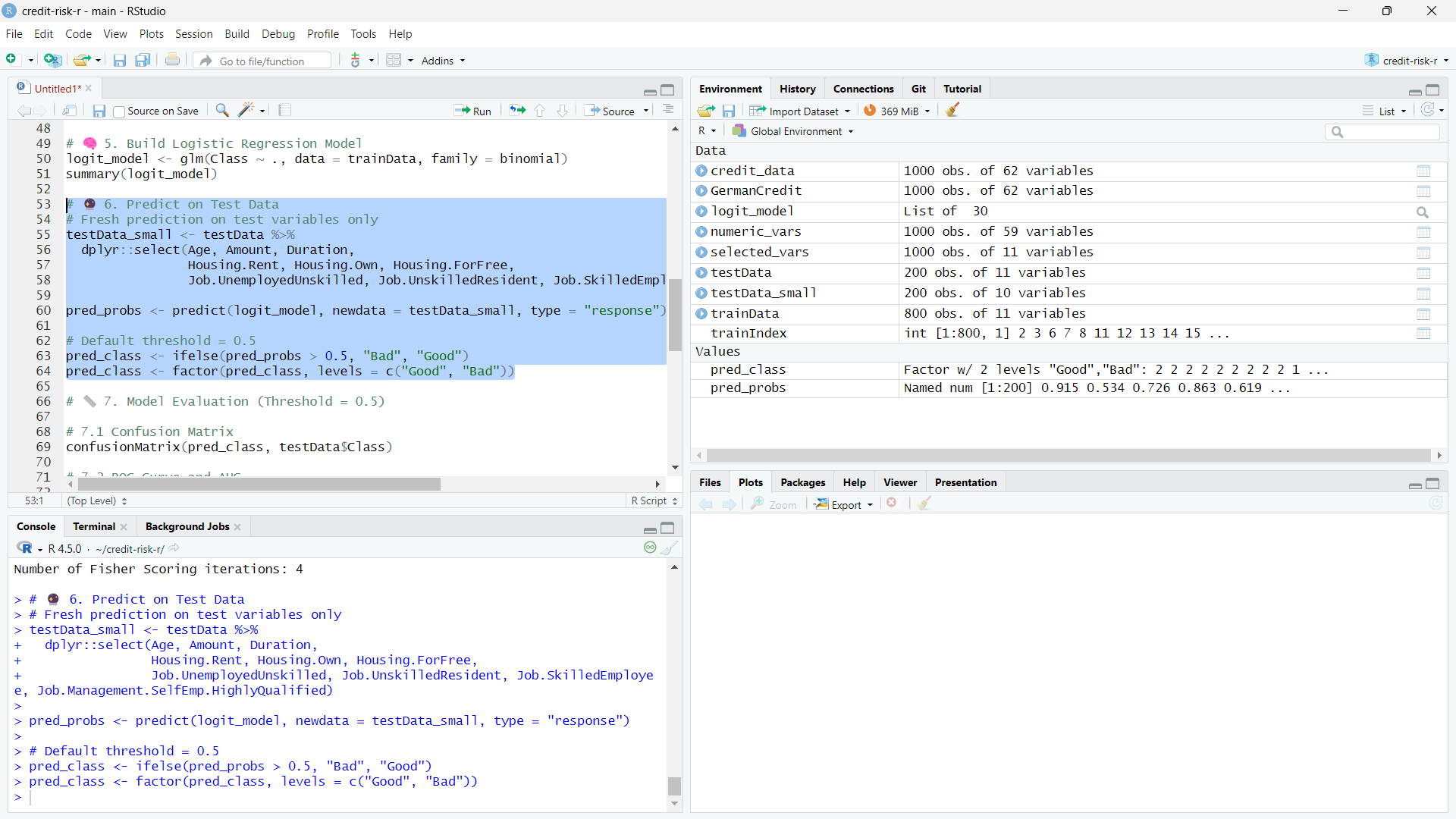
logit\_model <- glm(Class ~ ., data = trainData, family = binomial)  
summary(logit\_model)  
```

A screenshot of a computer

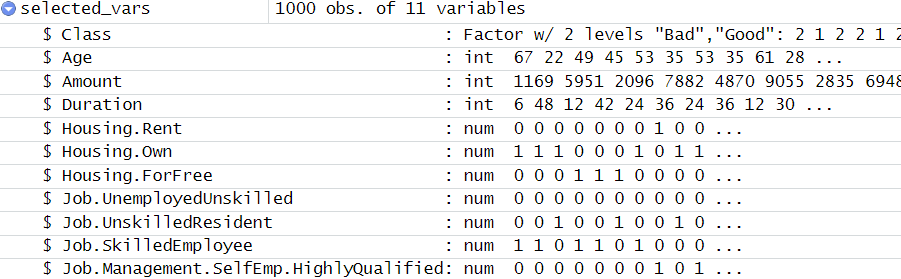
AI-generated content may be incorrect.

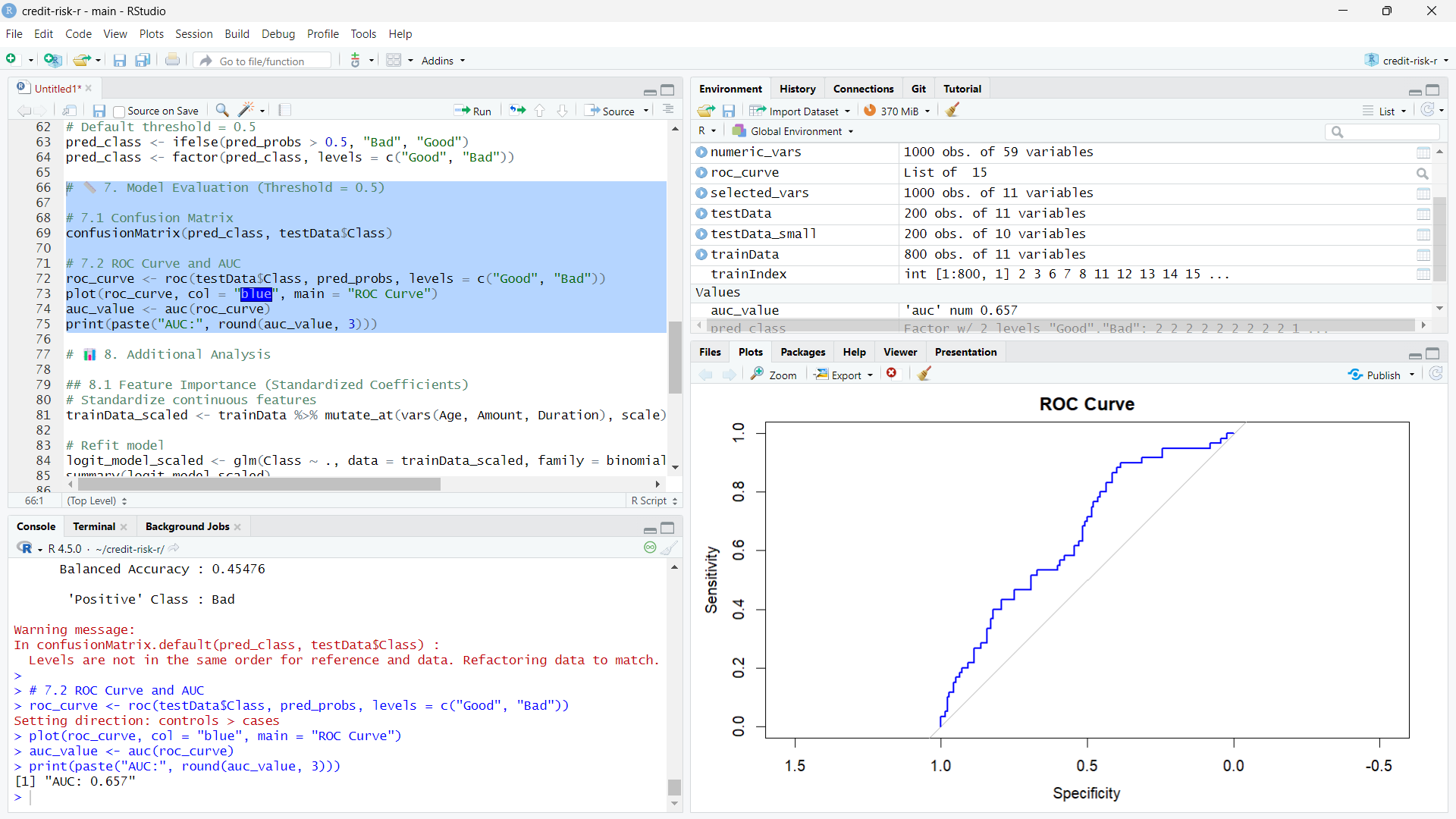
## 9. Prediction and Evaluation

```R  
testData\_small <- testData %>% select(-Class)  
pred\_probs <- predict(logit\_model, newdata = testData\_small, type = "response")  
pred\_class <- ifelse(pred\_probs > 0.5, "Bad", "Good")  
pred\_class <- factor(pred\_class, levels = c("Good", "Bad"))



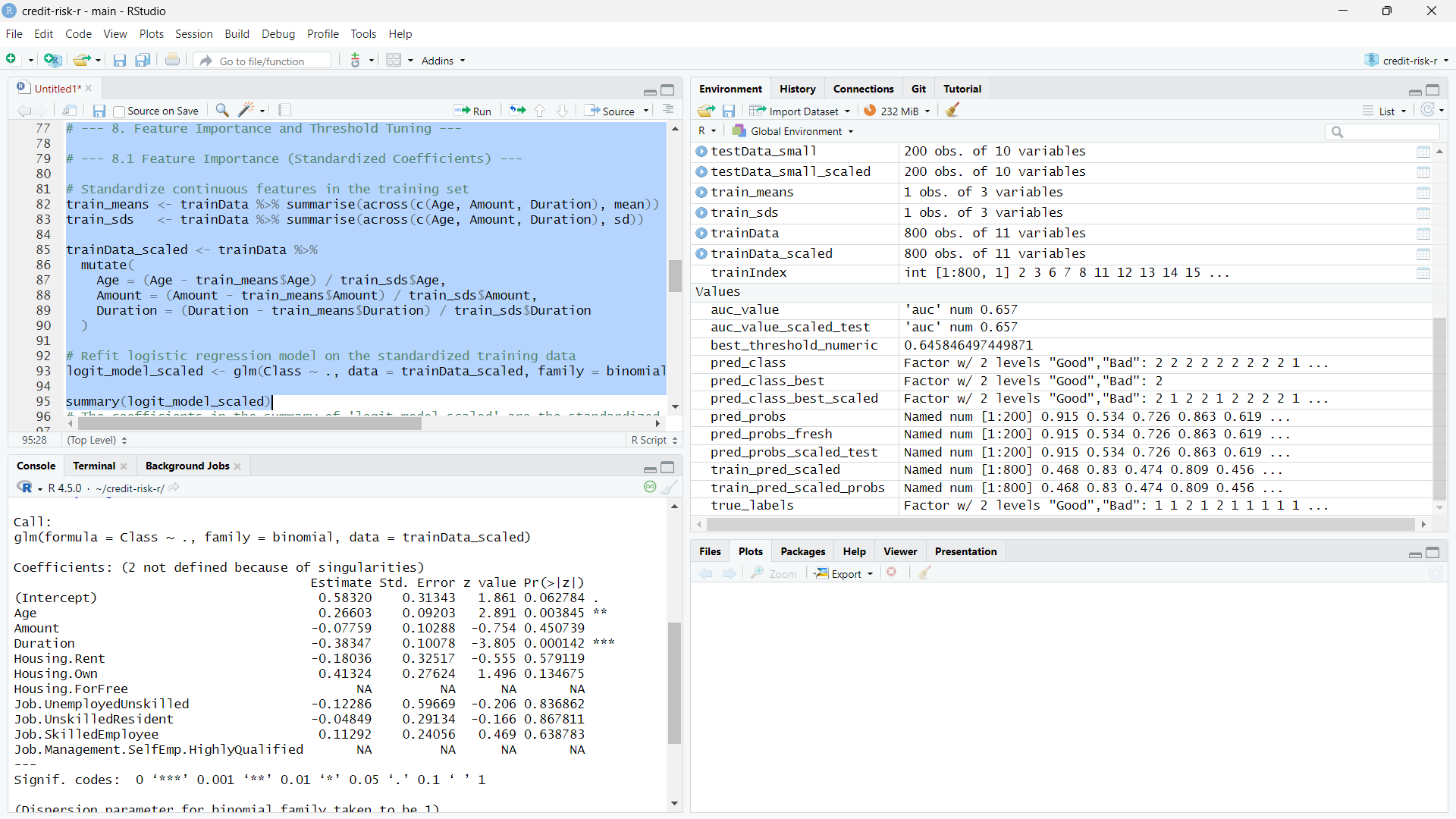
confusionMatrix(pred\_class, testData$Class)  
roc\_curve <- roc(testData$Class, pred\_probs, levels = c("Good", "Bad"))  
plot(roc\_curve, col = "blue")  
auc(roc\_curve)  
```



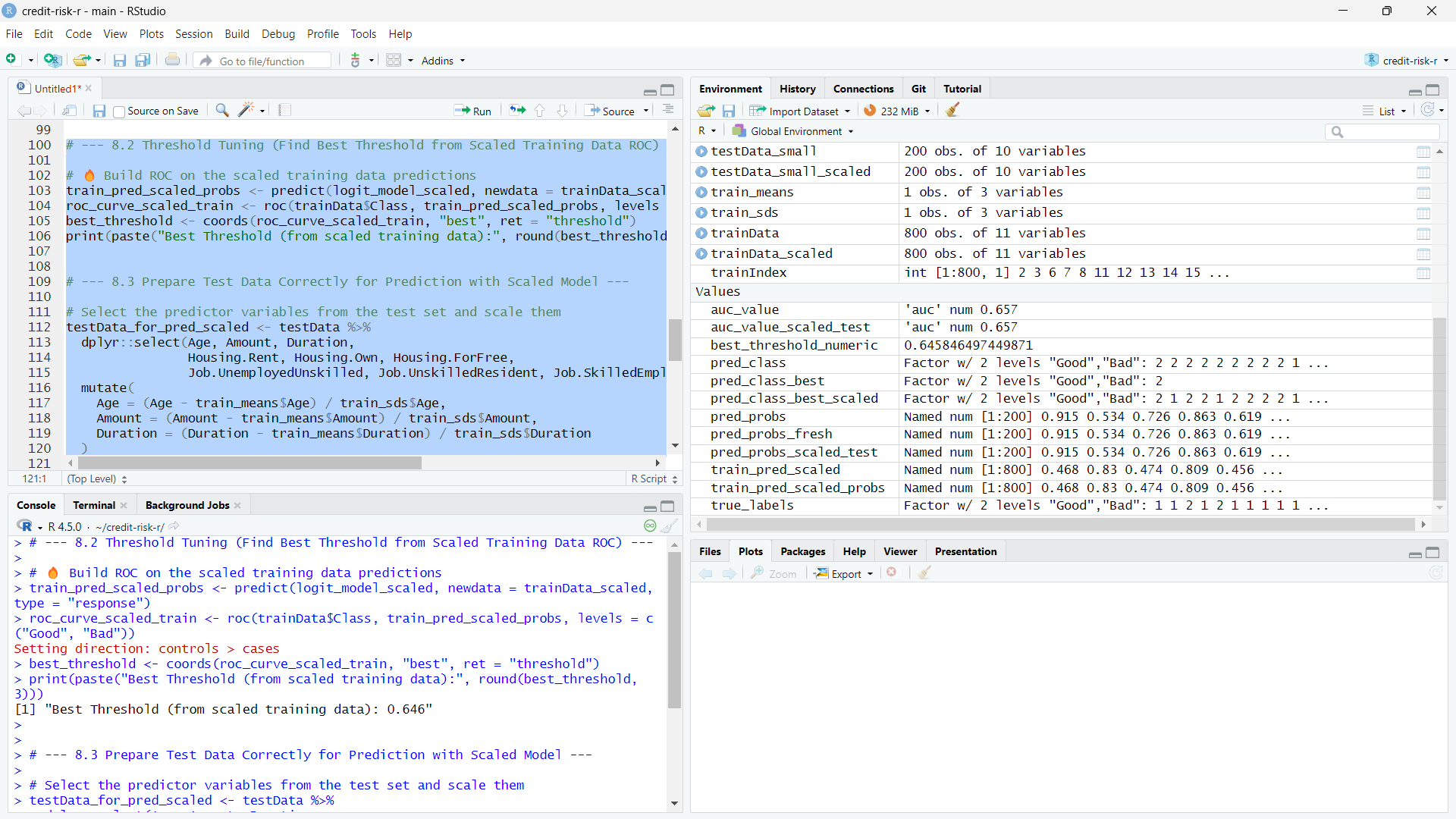


## 10. Threshold Tuning and Final Evaluation

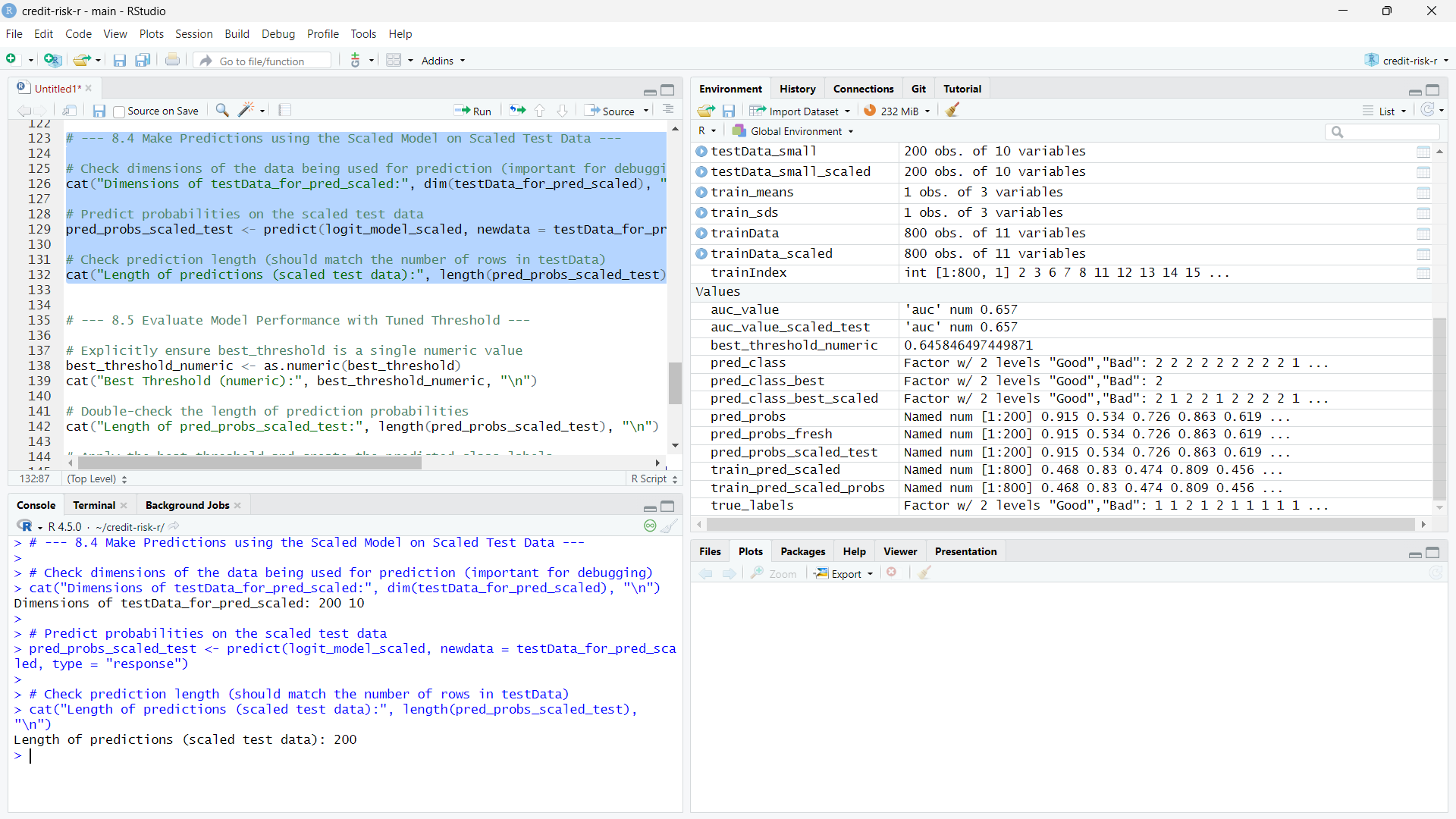
```R  
train\_means <- trainData %>% summarise(across(c(Age, Amount, Duration), mean))  
train\_sds <- trainData %>% summarise(across(c(Age, Amount, Duration), sd))  
trainData\_scaled <- trainData %>% mutate(  
 Age = (Age - train\_means$Age) / train\_sds$Age,  
 Amount = (Amount - train\_means$Amount) / train\_sds$Amount,  
 Duration = (Duration - train\_means$Duration) / train\_sds$Duration  
)  
logit\_model\_scaled <- glm(Class ~ ., data = trainData\_scaled, family = binomial)



train\_pred\_scaled\_probs <- predict(logit\_model\_scaled, newdata = trainData\_scaled, type = "response")  
roc\_curve\_scaled\_train <- roc(trainData$Class, train\_pred\_scaled\_probs, levels = c("Good", "Bad"))  
best\_threshold <- coords(roc\_curve\_scaled\_train, "best", ret = "threshold")  
testData\_for\_pred\_scaled <- testData %>% select(-Class) %>% mutate(  
 Age = (Age - train\_means$Age) / train\_sds$Age,  
 Amount = (Amount - train\_means$Amount) / train\_sds$Amount,  
 Duration = (Duration - train\_means$Duration) / train\_sds$Duration  
)



pred\_probs\_scaled\_test <- predict(logit\_model\_scaled, newdata = testData\_for\_pred\_scaled, type = "response")  
pred\_class\_best\_scaled <- factor(ifelse(pred\_probs\_scaled\_test > as.numeric(best\_threshold), "Bad", "Good"), levels = c("Good", "Bad"))



confusionMatrix(pred\_class\_best\_scaled, factor(testData$Class, levels = c("Good", "Bad")))  
roc\_curve\_test\_scaled <- roc(testData$Class, pred\_probs\_scaled\_test, levels = c("Good", "Bad"))  
plot(roc\_curve\_test\_scaled, col = "darkgreen")  
auc(roc\_curve\_test\_scaled)  
```

