Code

Telecom Customer Churn Prediction Analysis

Ashish Yakub Beary

2025-04-20

1. Introduction

This report presents an analysis of customer churn for a telecommunications company. Customer churn, or the rate at which customers stop doing business with a company, is a critical metric in the telecom industry due to the high cost of acquiring new customers compared to retaining existing ones.

1.1 Research Questions

The primary research question addressed in this analysis is:

- What factors most significantly predict customer churn in the telecom industry?

Secondary questions include:

- How do service usage patterns and demographics correlate with churn probability?
- Which specific services or contract features have the strongest protective effect against churn?

- Can we identify high-risk customers early to implement targeted retention strategies?

1.2 Dataset Overview

The analysis is based on the following datasets:

- **telecom_customer_churn.csv**: Primary dataset containing customer information (7,043 customers with 38 variables)
- **telecom_zipcode_population.csv**: Supplementary dataset with population information by zip code
- **telecom_data_dictionary.csv**: Metadata describing each variable

Hide

```
# Load necessary packages
library(tidyverse)  # For data manipulation and visualization
library(caret)      # For machine learning workflow
library(randomForest) # For random forest model
library(pROC)       # For ROC curve analysis
library(corrplot)   # For correlation visualization
library(janitor)    # For cleaning column names
library(scales)     # For nice scales on plots
library(knitr)      # For tables
library(kableExtra) # For enhanced tables
library(viridis)    # For nice color palettes
library(gridExtra)  # For combining plots
library(pdp)        # For partial dependence plots
library(broom)

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

2. Data Loading and Initial Exploration

Hide

```
# Read the datasets
telecom_churn <- read.csv("telecom_customer_churn.csv", stringsAsFactors = TR
        UE)
zipcode_population <- read.csv("telecom_zipcode_population.csv")
data_dictionary <- read.csv("telecom_data_dictionary.csv", encoding = "CP125
        2")

# Clean column names
telecom_churn <- clean_names(telecom_churn)
zipcode_population <- clean_names(zipcode_population)
data_dictionary <- clean_names(data_dictionary)
```

2.1 Data Structure

Let's examine the structure of our main dataset:

Hide

```
# Display the structure of the first few columns
str(telecom_churn[,])
```

```
## 'data.frame':    7043 obs. of  38 variables:
##  $ customer_id                      : Factor w/ 7043 levels "0002-ORFB
O","0003-MKNFE",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ gender                           : Factor w/ 2 levels "Female","Male":
1 2 2 2 1 1 1 2 1 1 ...
##  $ age                              : int  37 46 50 78 75 23 67 52 68 43
...
##  $ married                          : Factor w/ 2 levels "No","Yes": 2 1 1
2 2 1 2 2 1 2 ...
##  $ number_of_dependents             : int  0 0 0 0 0 3 0 0 0 1 ...
##  $ city                             : Factor w/ 1106 levels "Acampo","Acto
n",..: 347 369 223 588 140 609 547 654 925 916 ...
##  $ zip_code                         : int  93225 91206 92627 94553 93010 9
5345 93437 94558 93063 95681 ...
##  $ latitude                         : num  34.8 34.2 33.6 38 34.2 ...
##  $ longitude                        : num  -119 -118 -118 -122 -119 ...
##  $ number_of_referrals              : int  2 0 0 1 3 0 1 8 0 3 ...
##  $ tenure_in_months                 : int  9 9 4 13 3 9 71 63 7 65 ...
##  $ offer                            : Factor w/ 6 levels "None","Offer
A",..: 1 1 6 5 1 6 2 3 6 1 ...
##  $ phone_service                    : Factor w/ 2 levels "No","Yes": 2 2 2
2 2 2 2 2 2 2 ...
##  $ avg_monthly_long_distance_charges: num  42.39 10.69 33.65 27.82 7.38
...
##  $ multiple_lines                   : Factor w/ 3 levels "","No","Yes": 2
3 2 2 2 2 2 3 2 3 ...
##  $ internet_service                 : Factor w/ 2 levels "No","Yes": 2 2 2
2 2 2 2 2 2 2 ...
##  $ internet_type                    : Factor w/ 4 levels "","Cable","DS
L",..: 2 2 4 4 4 2 4 4 3 2 ...
##  $ avg_monthly_gb_download          : int  16 10 30 4 11 73 14 7 21 14 ...
##  $ online_security                  : Factor w/ 3 levels "","No","Yes": 2
2 2 2 2 2 3 3 3 3 ...
##  $ online_backup                    : Factor w/ 3 levels "","No","Yes": 3
2 2 3 2 2 3 2 2 3 ...
##  $ device_protection_plan           : Factor w/ 3 levels "","No","Yes": 2
2 3 3 2 2 3 2 2 3 ...
##  $ premium_tech_support             : Factor w/ 3 levels "","No","Yes": 3
2 2 2 3 3 3 3 2 3 ...
##  $ streaming_tv                     : Factor w/ 3 levels "","No","Yes": 3
2 2 3 3 3 3 2 2 3 ...
##  $ streaming_movies                 : Factor w/ 3 levels "","No","Yes": 2
3 2 3 2 3 3 2 2 3 ...
##  $ streaming_music                  : Factor w/ 3 levels "","No","Yes": 2
3 2 2 2 3 3 2 2 3 ...
##  $ unlimited_data                   : Factor w/ 3 levels "","No","Yes": 3
2 3 3 3 3 3 2 3 3 ...
##  $ contract                         : Factor w/ 3 levels "Month-to-Mont
```

```
h",..: 2 1 1 1 1 1 3 3 3 3 ...
##  $ paperless_billing             : Factor w/ 2 levels "No","Yes": 2 1 2
2 2 2 2 2 2 2 ...
##  $ payment_method                : Factor w/ 3 levels "Bank Withdrawa
l",..: 2 2 1 1 2 2 1 2 1 2 ...
##  $ monthly_charge                : num  65.6 -4 73.9 98 83.9 ...
##  $ total_charges                 : num  593 542 281 1238 267 ...
##  $ total_refunds                 : num  0 38.3 0 0 0 ...
##  $ total_extra_data_charges      : int  0 10 0 0 0 0 0 20 0 0 ...
##  $ total_long_distance_charges   : num  381.5 96.2 134.6 361.7 22.1 ...
##  $ total_revenue                 : num  975 610 415 1600 290 ...
##  $ customer_status               : Factor w/ 3 levels "Churned","Joine
d",..: 3 3 1 1 1 3 3 3 3 3 ...
##  $ churn_category                : Factor w/ 6 levels "","Attitude",..:
1 1 3 4 4 1 1 1 1 1 ...
##  $ churn_reason                  : Factor w/ 21 levels "","Attitude of
service provider",..: 1 1 4 20 16 1 1 1 1 1 ...
```

Hide

```
# Get a summary of the dataset dimensions
cat("Number of customers:", nrow(telecom_churn), "\n")
```

```
## Number of customers: 7043
```

Hide

```
cat("Number of variables:", ncol(telecom_churn), "\n")
```

```
## Number of variables: 38
```

2.2 Create Binary Churn Variable

For our analysis, we'll create a binary churn variable that indicates whether a customer has churned or not.

Hide

```
# Create binary churn variable for analysis
telecom_churn$churned <- ifelse(telecom_churn$customer_status == "Churned",
        "Yes", "No")
telecom_churn$churned <- as.factor(telecom_churn$churned)

# Check distribution
table(telecom_churn$churned)
```

```
##
##   No  Yes
## 5174 1869
```

2.3 Basic Summary Statistics

Hide

```
# Basic summary statistics for key numerical variables
summary(telecom_churn[c("age", "tenure_in_months", "number_of_dependents",
                       "avg_monthly_gb_download", "monthly_charge", "total_ch
        arges")])
```

```
##       age        tenure_in_months number_of_dependents avg_monthly_gb_down
load
##  Min.   :19.00   Min.   : 1.00    Min.   :0.0000       Min.   : 2.00
##  1st Qu.:32.00   1st Qu.: 9.00    1st Qu.:0.0000       1st Qu.:13.00
##  Median :46.00   Median :29.00    Median :0.0000       Median :21.00
##  Mean   :46.51   Mean   :32.39    Mean   :0.4687       Mean   :26.19
##  3rd Qu.:60.00   3rd Qu.:55.00    3rd Qu.:0.0000       3rd Qu.:30.00
##  Max.   :80.00   Max.   :72.00    Max.   :9.0000       Max.   :85.00
##                                                        NA's   :1526
##  monthly_charge   total_charges
##  Min.   :-10.00   Min.   :  18.8
##  1st Qu.: 30.40   1st Qu.: 400.1
##  Median : 70.05   Median :1394.5
##  Mean   : 63.60   Mean   :2280.4
##  3rd Qu.: 89.75   3rd Qu.:3786.6
##  Max.   :118.75   Max.   :8684.8
##
```

3. Data Cleaning and Preprocessing

3.1 Missing Values

Hide

```
# Check for missing values
missing_values <- colSums(is.na(telecom_churn))
missing_values[missing_values > 0]
```

```
## avg_monthly_long_distance_charges           avg_monthly_gb_download
##                               682                              1526
```

Hide

```
# Handle missing values for avg_monthly_gb_download using median imputation
telecom_churn$avg_monthly_gb_download[is.na(telecom_churn$avg_monthly_gb_down
        load)] <-
  median(telecom_churn$avg_monthly_gb_download, na.rm = TRUE)

telecom_churn$avg_monthly_long_distance_charges[is.na(telecom_churn$avg_month
        ly_long_distance_charges)] <-
  median(telecom_churn$avg_monthly_long_distance_charges, na.rm = TRUE)
```

3.2 Handling Categorical Variables

Some categorical variables have empty values because they are conditionally relevant. For example, internet-related services are only applicable to customers with internet service.

Hide

```
# Convert empty strings to NA for certain categorical columns
service_cols <- c("multiple_lines", "internet_type", "online_security",
                 "online_backup", "device_protection_plan", "premium_tech_support",
                 "streaming_tv", "streaming_movies", "streaming_music", "unlimited_data")

for(col in service_cols) {
  telecom_churn[[col]] <- as.character(telecom_churn[[col]])
  telecom_churn[[col]][telecom_churn[[col]] == ""] <- NA
  telecom_churn[[col]] <- as.factor(telecom_churn[[col]])
}

# Some customers don't have internet service, which is why they have NA for internet-related services
# We'll recode these NAs as "No Internet Service"
internet_related <- c("internet_type", "online_security", "online_backup",
                      "device_protection_plan", "premium_tech_support",
                      "streaming_tv", "streaming_movies", "streaming_music",
                      "unlimited_data")

for(col in internet_related) {
  levels(telecom_churn[[col]]) <- c(levels(telecom_churn[[col]]), "No Internet Service")
  telecom_churn[[col]][is.na(telecom_churn[[col]]) & telecom_churn$internet_service == "No"] <- "No Internet Service"
}

# Similarly for phone-related services
phone_related <- c("multiple_lines")
for(col in phone_related) {
  levels(telecom_churn[[col]]) <- c(levels(telecom_churn[[col]]), "No Phone Service")
  telecom_churn[[col]][is.na(telecom_churn[[col]]) & telecom_churn$phone_service == "No"] <- "No Phone Service"
}
```

4. Exploratory Data Analysis

4.1 Overall Churn Rate

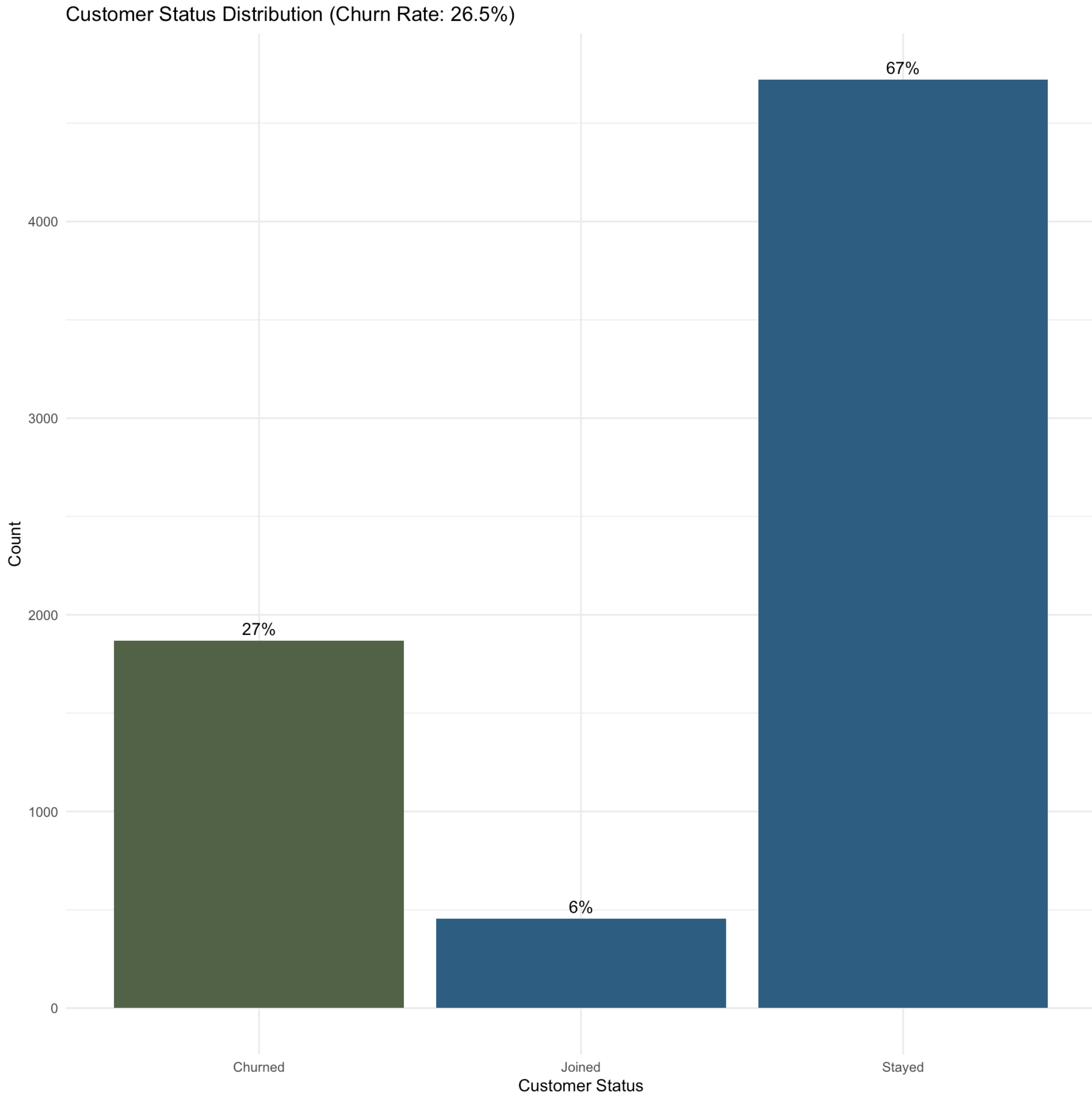
Hide

```
# Calculate overall churn rate
churn_rate <- mean(telecom_churn$customer_status == "Churned") * 100
cat("Overall churn rate:", round(churn_rate, 2), "%\n")
```

```
## Overall churn rate: 26.54 %
```

Hide

```
# Visualize the churn distribution
ggplot(telecom_churn, aes(x = customer_status)) +
  geom_bar(fill = c("#52664b", "#2e6083", "#2e6083")) +
  geom_text(stat = "count", aes(label = scales::percent(after_stat(count)/sum
       (after_stat(count)))),
            vjust = -0.5) +
  labs(title = paste0("Customer Status Distribution (Churn Rate: ", round(chu
       rn_rate, 1), "%)"),
       x = "Customer Status",
       y = "Count") +
  theme_minimal()
```

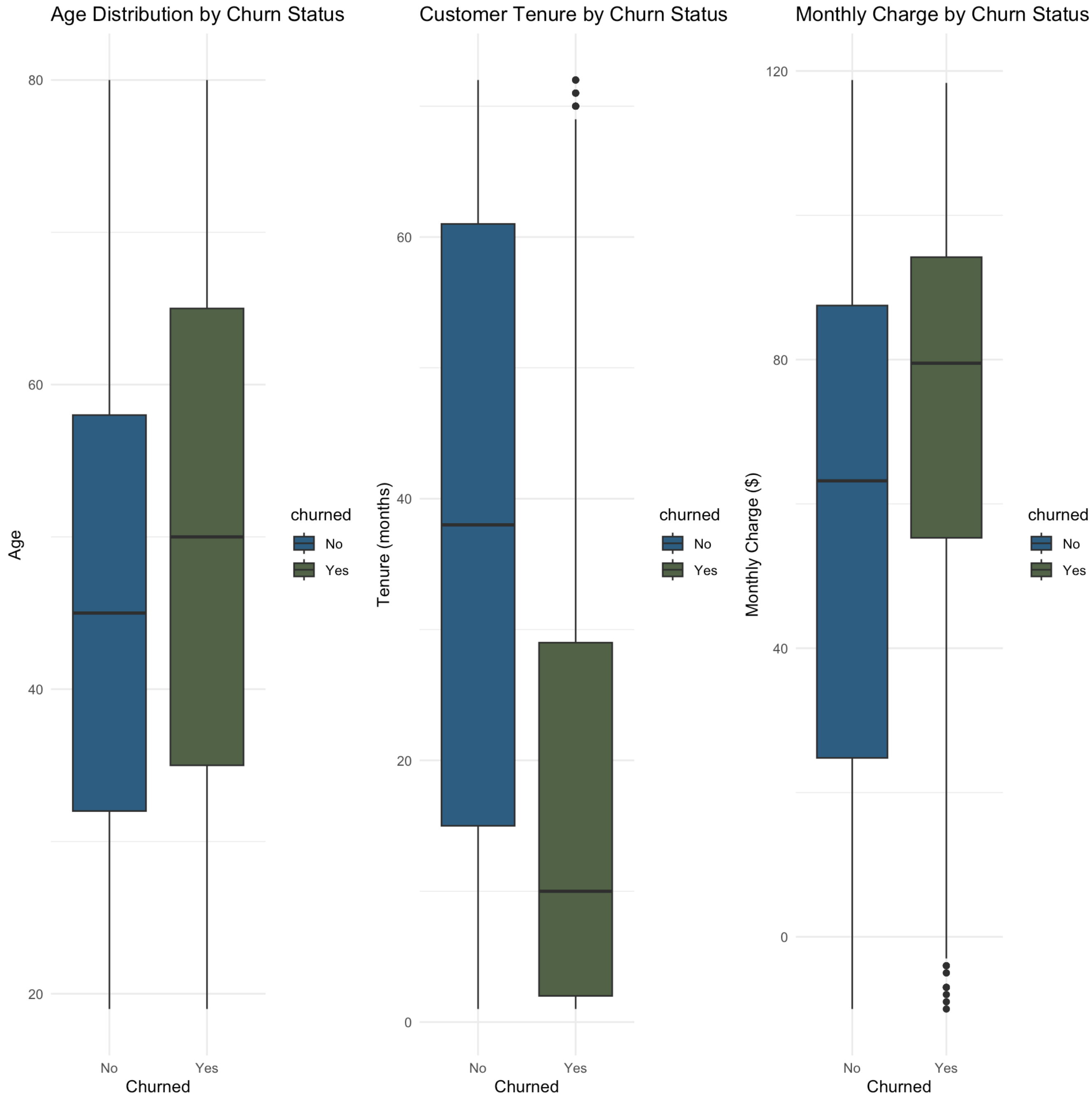


4.2 Numeric Variables and Churn

Let's examine the relationship between key numerical variables and churn:

Hide

```
# Age distribution by churn status
p1 <- ggplot(telecom_churn, aes(x = churned, y = age, fill = churned)) +
  geom_boxplot() +
  labs(title = "Age Distribution by Churn Status",
       x = "Churned",
       y = "Age") +
  scale_fill_manual(values = c("#2e6083", "#52664b")) +
  theme_minimal()

# Tenure distribution by churn status
p2 <- ggplot(telecom_churn, aes(x = churned, y = tenure_in_months, fill = chu
rned)) +
  geom_boxplot() +
  labs(title = "Customer Tenure by Churn Status",
       x = "Churned",
       y = "Tenure (months)") +
  scale_fill_manual(values = c("#2e6083", "#52664b")) +
  theme_minimal()

# Monthly charge distribution by churn status
p3 <- ggplot(telecom_churn, aes(x = churned, y = monthly_charge, fill = churn
ed)) +
  geom_boxplot() +
  labs(title = "Monthly Charge by Churn Status",
       x = "Churned",
       y = "Monthly Charge ($)") +
  scale_fill_manual(values = c("#2e6083", "#52664b")) +
  theme_minimal()

# Display plots side by side
grid.arrange(p1, p2, p3, ncol = 3)
```



Hide

```
# Calculate mean statistics by churn status
telecom_churn %>%
  group_by(churned) %>%
  summarize(
    avg_age = mean(age, na.rm = TRUE),
    avg_tenure = mean(tenure_in_months, na.rm = TRUE),
    avg_monthly_charge = mean(monthly_charge, na.rm = TRUE),
    avg_total_charges = mean(total_charges, na.rm = TRUE),
    avg_monthly_download = mean(avg_monthly_gb_download, na.rm = TRUE)
  ) %>%
  kable(caption = "Key Metrics by Churn Status", digits = 2) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Key Metrics by Churn Status

churned	avg_age	avg_tenure	avg_monthly_charge	avg_total_charges	avg_monthly_download
No	45.34	37.59	60.07	2550.79	25.65
Yes	49.74	17.98	73.35	1531.80	23.45

Observations:

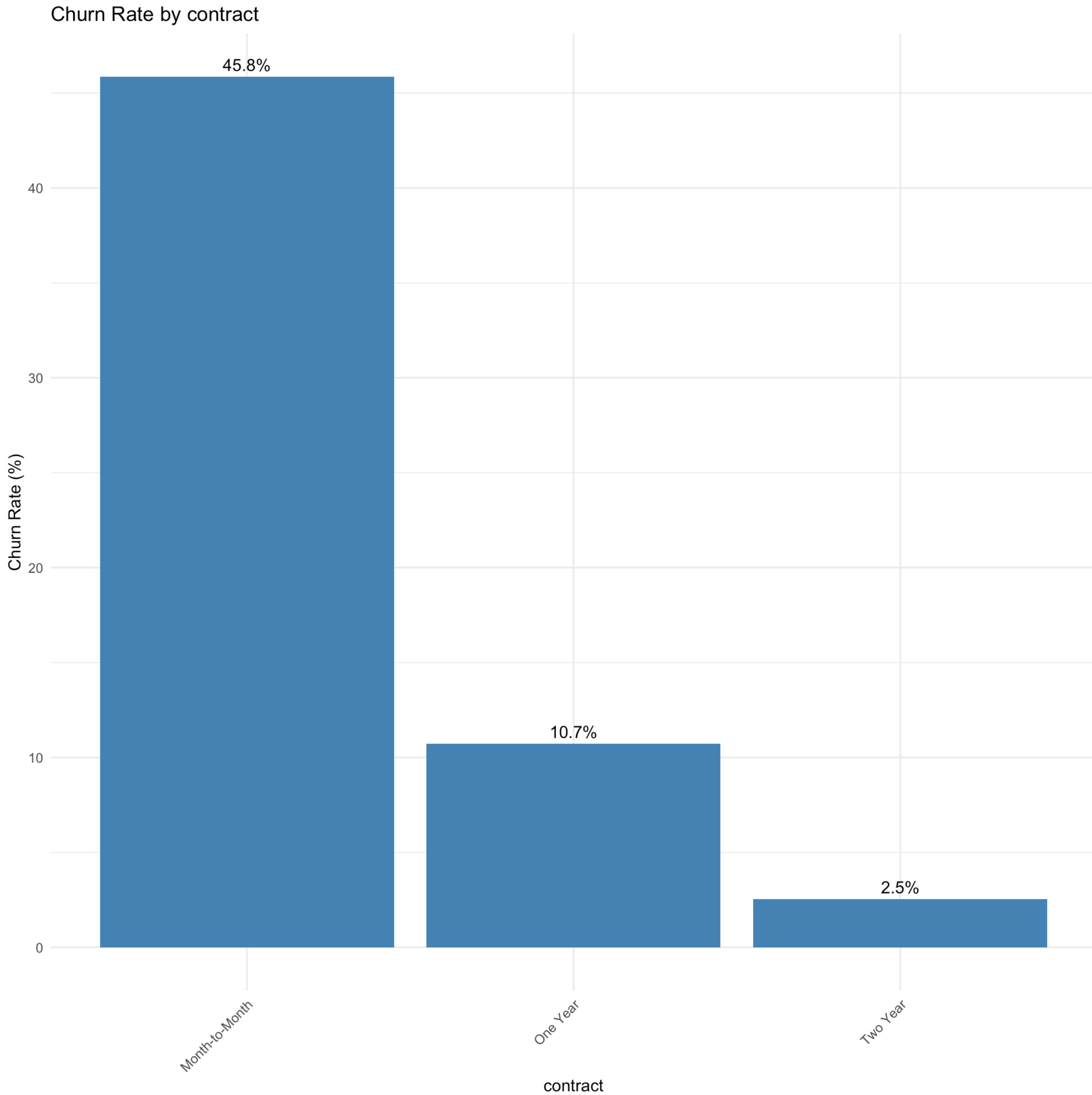
1. **Age**: Churned customers tend to be slightly older on average.
2. **Tenure**: There's a substantial difference in tenure between churned and retained customers. Customers who churn have much shorter tenure on average.
3. **Monthly Charge**: Churned customers have higher monthly charges on average.

4.3 Categorical Variables and Churn

Let's analyze how categorical variables relate to churn:

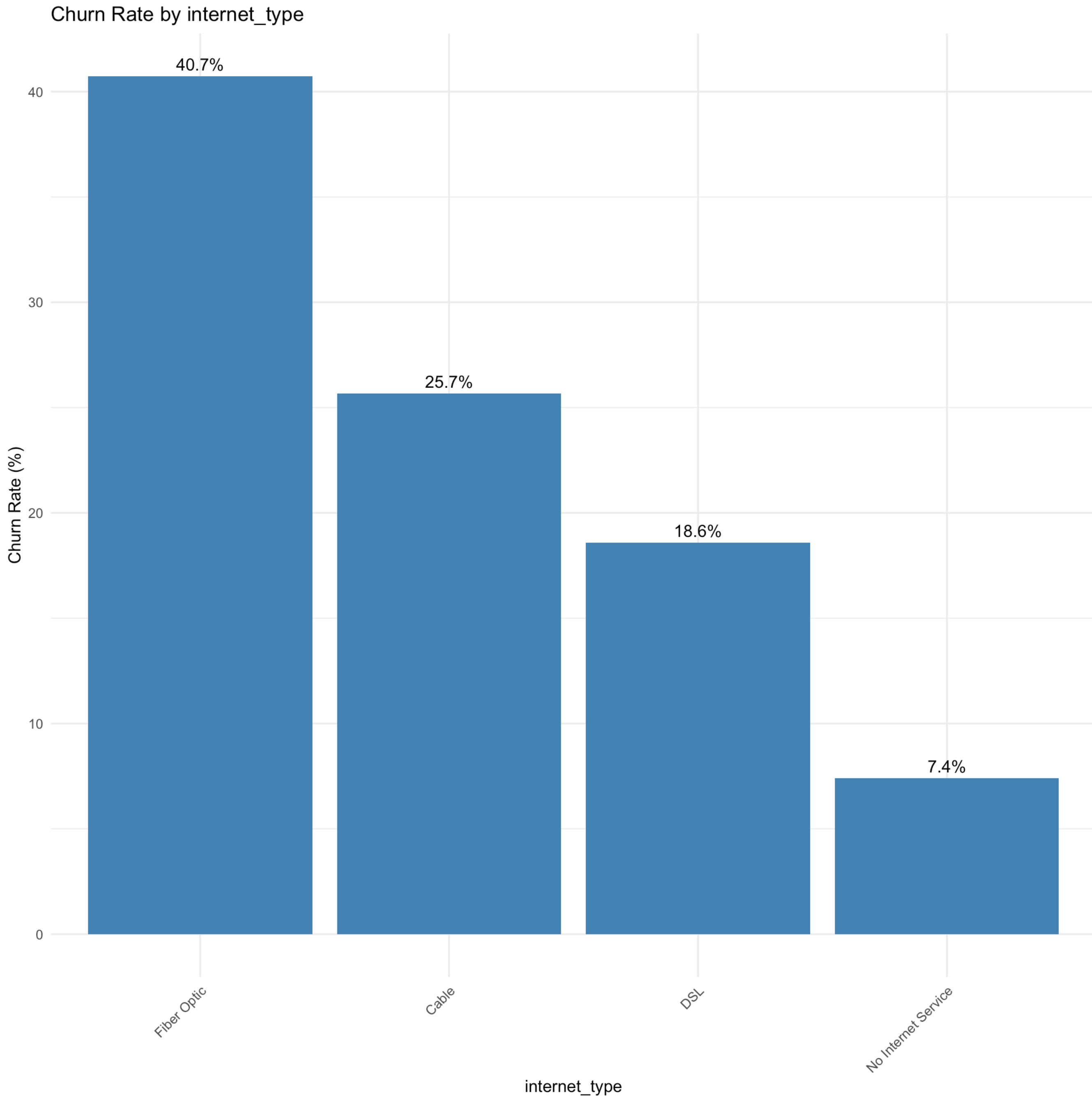
Hide

```
# Create a function to plot churn rate by category
plot_churn_by_category <- function(data, variable) {
  # Calculate percentages
  churn_by_cat <- data %>%
    group_by(!!sym(variable)) %>%
    summarize(
      total = n(),
      churned = sum(customer_status == "Churned"),
      churn_rate = churned / total * 100
    ) %>%
    arrange(desc(churn_rate))

  # Create plot
  ggplot(churn_by_cat, aes(x = reorder(!!sym(variable), -churn_rate), y = chu
        rn_rate)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    geom_text(aes(label = paste0(round(churn_rate, 1), "%")), vjust = -0.5) +
    labs(title = paste("Churn Rate by", variable),
         x = variable,
         y = "Churn Rate (%)") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

# Plot for contract type
plot_churn_by_category(telecom_churn, "contract")
```



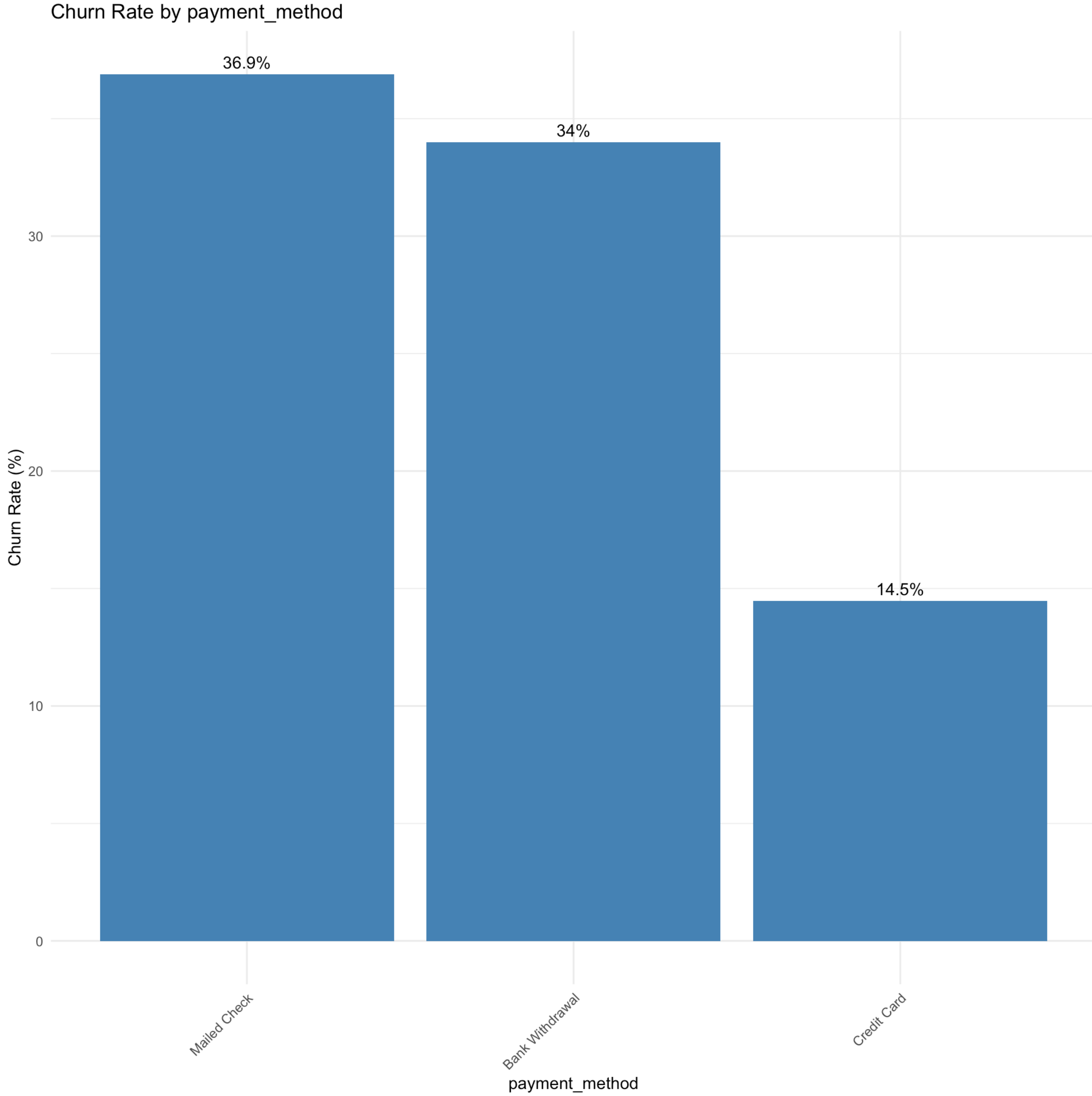
Hide

```
# Plot for internet type
plot_churn_by_category(telecom_churn, "internet_type")
```



Hide

```
# Plot for payment method
plot_churn_by_category(telecom_churn, "payment_method")
```



Observations:

1. **Contract Type**: Month-to-month contracts have a significantly higher churn rate (45.8%) compared to one-year (10.7%) and two-year contracts (2.5%).
2. **Internet Type**: Fiber optic internet customers have the highest churn rate, while DSL customers and customers opting out of internet service have a lower churn rate.
3. **Payment Method**: Customers using electronic checks as their payment method have a higher churn rate compared to other payment methods.

4.4 Service Adoption Impact on Churn

Let’s analyze how various services impact churn:

Hide

```
# Create a function to calculate service impact on churn
service_impact <- function(data, service_var) {
  # Calculate churn rates
  service_churn <- data %>%
    group_by(!!sym(service_var)) %>%
    summarize(
      total = n(),
      churned = sum(customer_status == "Churned"),
      churn_rate = churned / total * 100
    )

  return(service_churn)
}

# Example for online security
online_security_impact <- service_impact(telecom_churn, "online_security")
kable(online_security_impact, caption = "Churn Rate by Online Security Statu
        s") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

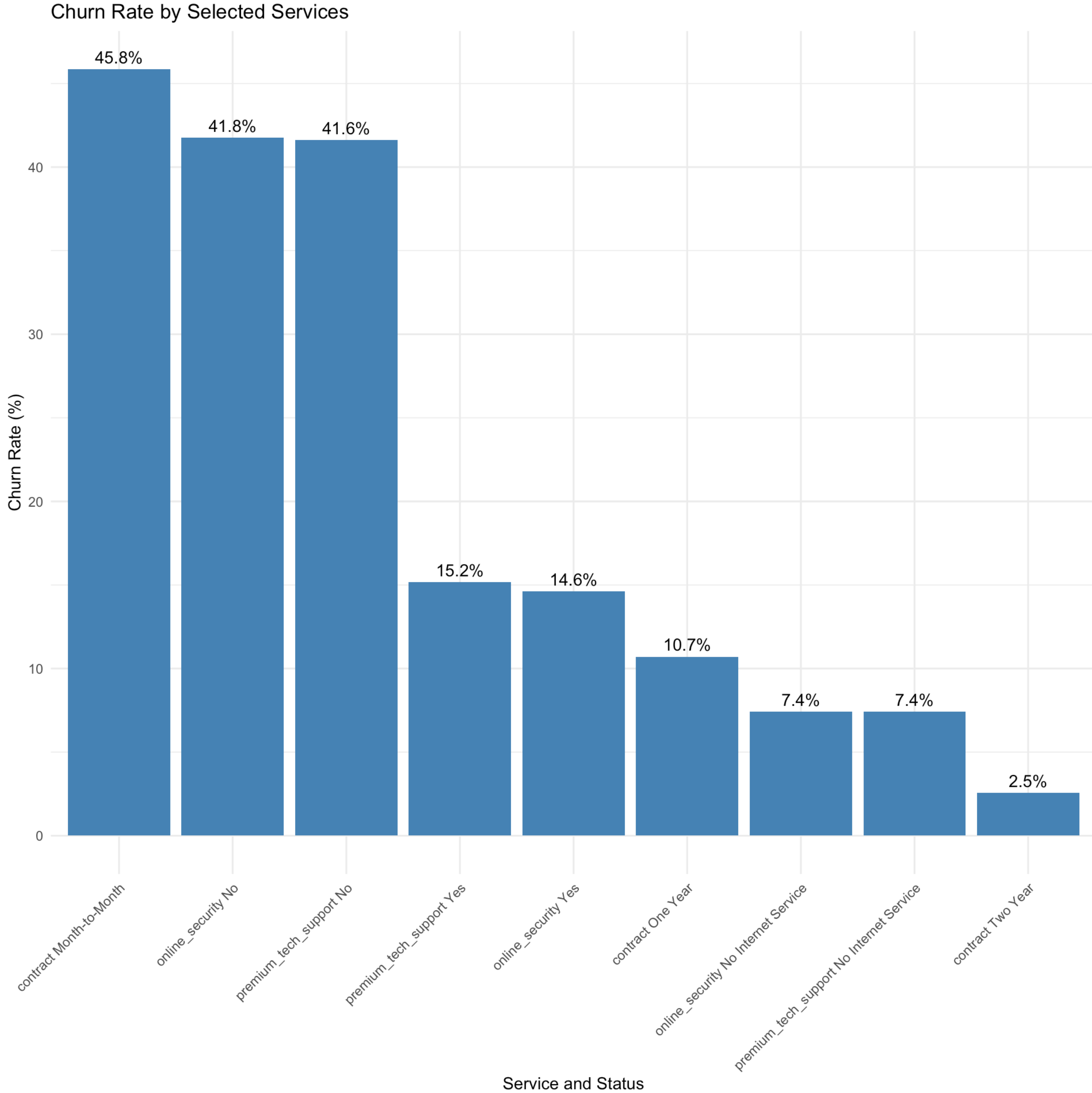
Churn Rate by Online Security Status

online_security	total	churned	churn_rate
No	3498	1461	41.76672
Yes	2019	295	14.61119
No Internet Service	1526	113	7.40498

Let’s analyze multiple services together:

Hide

```
# Selected services to analyze
selected_services <- c("online_security", "premium_tech_support", "contract")

# Create an empty data frame with the right structure
services_plot_data <- data.frame(
  service = character(),
  service_value = character(),
  total = numeric(),
  churned = numeric(),
  churn_rate = numeric(),
  stringsAsFactors = FALSE
)

# Loop through each service
for (service_name in selected_services) {
  # Get the churn data for this service
  service_data <- service_impact(telecom_churn, service_name)

  # Extract the service value column (which has a dynamic name)
  service_values <- service_data[[1]] # The first column contains the service
        values

  # Create a new data frame with consistent column names
  temp_df <- data.frame(
    service = service_name,
    service_value = as.character(service_values),
    total = service_data$total,
    churned = service_data$churned,
    churn_rate = service_data$churn_rate,
    stringsAsFactors = FALSE
  )

  # Add to the main data frame
  services_plot_data <- rbind(services_plot_data, temp_df)
}

# Plot with the data
ggplot(services_plot_data, aes(x = reorder(paste(service, service_value), -ch
        urn_rate), y = churn_rate)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = paste0(round(churn_rate, 1), "%")), vjust = -0.5) +
  labs(title = "Churn Rate by Selected Services",
       x = "Service and Status",
       y = "Churn Rate (%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Observations:

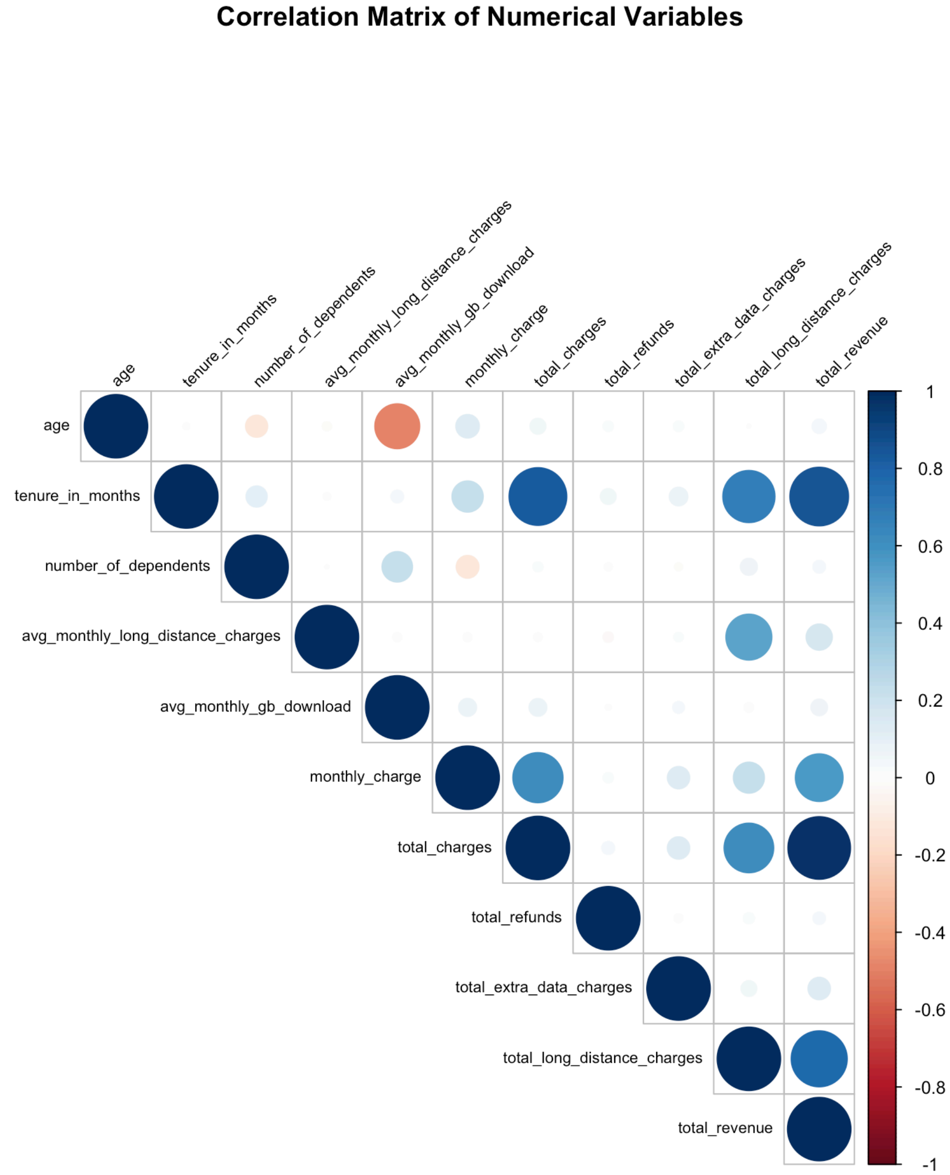
1. Customers without online security have a much higher churn rate (41.8%) than those with online security (14.6%).
2. Similar patterns are observed for premium tech support.
3. This suggests that additional services may create "stickiness" and reduce churn.

4.5 Correlation Analysis

Let's examine the correlations between numerical variables:

Hide

```
# Select numerical columns for correlation analysis
numeric_cols <- telecom_churn %>%
  select(age, tenure_in_months, number_of_dependents, avg_monthly_long_distan
        ce_charges,
         avg_monthly_gb_download, monthly_charge, total_charges, total_refund
        s,
         total_extra_data_charges, total_long_distance_charges, total_revenu
        e) %>%
  names()

# Calculate correlation matrix
correlation_matrix <- cor(telecom_churn[numeric_cols], use = "pairwise.comple
        te.obs")

# Create a correlation plot
corrplot(correlation_matrix, method = "circle", type = "upper",
         tl.col = "black", tl.srt = 45, tl.cex = 0.7,
         title = "Correlation Matrix of Numerical Variables",
         mar = c(0, 0, 1, 0))
```



Key Correlations:

1. Tenure is positively correlated with total charges, as expected.
2. Monthly charge is positively correlated with average monthly GB download.
3. Total revenue is strongly correlated with total charges, monthly charge, and tenure. Moderately with long distance charges.

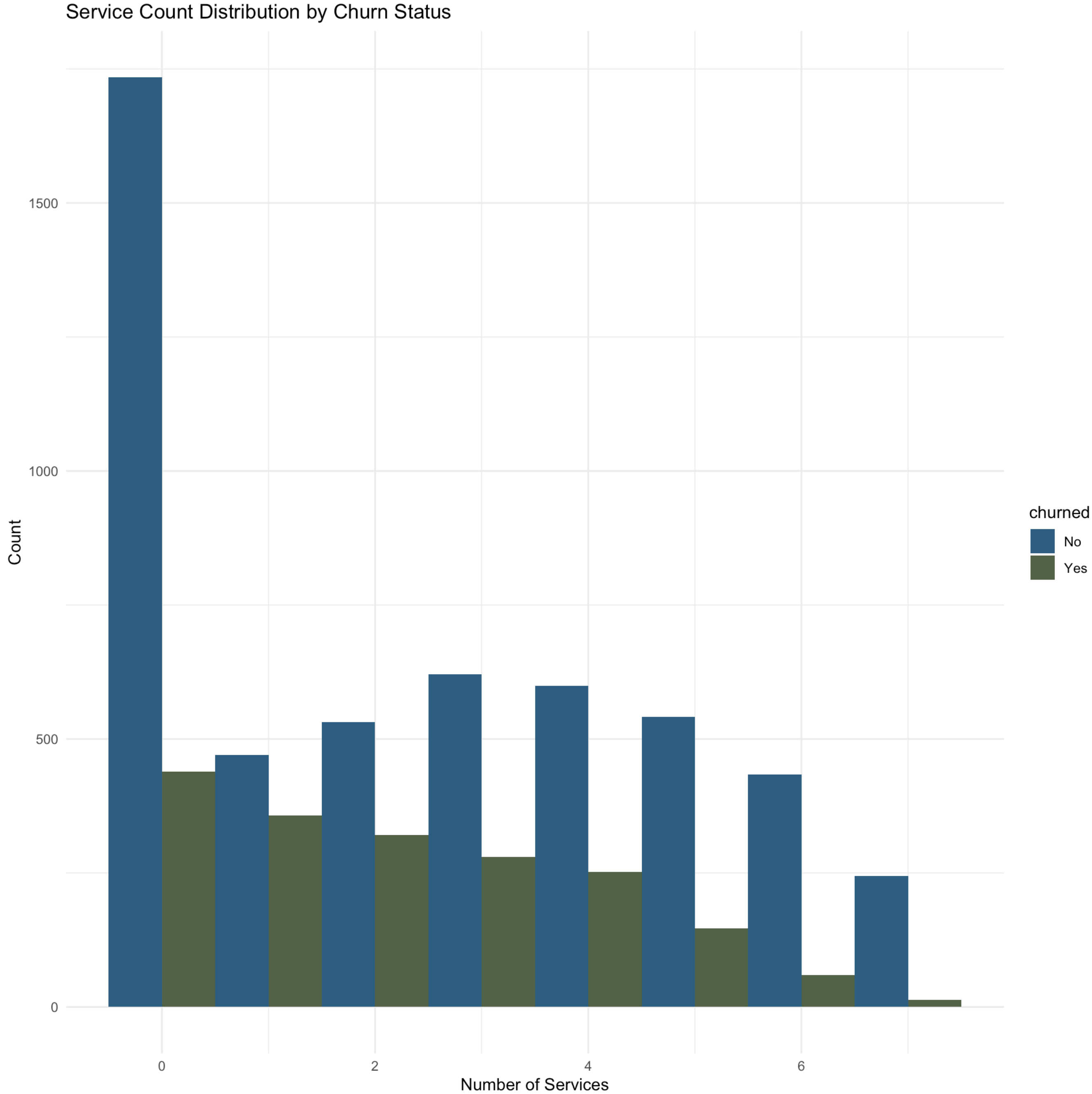
5. Feature Engineering

5.1 Create Derived Features

Let’s create some additional features that might help improve our predictive models:

Hide

```
# Customer lifetime value (CLV)
telecom_churn$customer_lifetime_value <- telecom_churn$total_revenue / teleco
        m_churn$tenure_in_months

# Average monthly revenue
telecom_churn$avg_monthly_revenue <- telecom_churn$total_revenue / telecom_ch
        urn$tenure_in_months

# Service count (number of additional services subscribed)
telecom_churn$service_count <- rowSums(telecom_churn[, c("online_security",
                                                         "online_backup",
                                                         "device_protection_pla
        n",
                                                         "premium_tech_suppor
        t",
                                                         "streaming_tv",
                                                         "streaming_movies",
                                                         "streaming_music")] ==
        "Yes", na.rm = TRUE)

# Visualize service count distribution by churn status
ggplot(telecom_churn, aes(x = service_count, fill = churned)) +
  geom_histogram(position = "dodge", binwidth = 1) +
  labs(title = "Service Count Distribution by Churn Status",
       x = "Number of Services",
       y = "Count") +
  scale_fill_manual(values = c( "#2e6083", "#52664b")) +
  theme_minimal()
```



5.2 Add Population Data

Let's join the zipcode population data to potentially identify demographic patterns:

Hide

```
# Join zipcode population data
telecom_churn <- left_join(telecom_churn, zipcode_population, by = "zip_cod
        e")

# Calculate population density quartiles
telecom_churn$population_quartile <- ntile(telecom_churn$population, 4)

# Analyze churn by population quartile
telecom_churn %>%
  group_by(population_quartile) %>%
  summarize(
    churn_rate = mean(churned == "Yes") * 100,
    customer_count = n()
  ) %>%
  kable(caption = "Churn Rate by Population Quartile", digits = 2) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Churn Rate by Population Quartile

population_quartile	churn_rate	customer_count
1	24.65	1761
2	24.30	1761
3	27.14	1761
4	30.06	1760

6. Modeling Preparation

6.1 Encoding Categorical Variables

For our models, we need to encode categorical variables:

Hide

```
# For variables with more than two levels, create dummy variables
dummy_vars <- c("contract", "internet_type", "payment_method", "offer")

# Create a formula for model matrix
dummy_formula <- as.formula(paste("~", paste(dummy_vars, collapse = " + ")))

# Generate dummy variables
dummy_data <- model.matrix(dummy_formula, data = telecom_churn)[, -1] # Remov
        e intercept

# Check the actual column names in the dummy data
colnames(dummy_data)[1:10]  # Look at the first 10 column names to understand
        the naming pattern
```

```
##  [1] "contractOne Year"                 "contractTwo Year"
##  [3] "internet_typeDSL"                 "internet_typeFiber Optic"
##  [5] "internet_typeNo Internet Service" "payment_methodCredit Card"
##  [7] "payment_methodMailed Check"       "offerOffer A"
##  [9] "offerOffer B"                     "offerOffer C"
```

Hide

```
# Combine with original data
telecom_churn_encoded <- cbind(telecom_churn, as.data.frame(dummy_data))

# Use the correct column names when checking the data
# For example, if the proper names are:
head(telecom_churn_encoded[, grep("contract", colnames(telecom_churn_encode
        d), value = TRUE)])
```

```
##         contract contractOne Year contractTwo Year
## 1       One Year                1                0
## 2 Month-to-Month                0                0
## 3 Month-to-Month                0                0
## 4 Month-to-Month                0                0
## 5 Month-to-Month                0                0
## 6 Month-to-Month                0                0
```

6.2 Feature Selection

Let's select relevant features for our models:

Hide

```
# Exclude redundant or irrelevant columns
exclude_cols <- c("customer_id", "customer_status", "churn_category", "churn_
        reason",
                "latitude", "longitude", "zip_code", "city")

# Create a new data frame with selected features
model_data <- telecom_churn_encoded %>%
  select(-one_of(exclude_cols))

# Check class balance
table(model_data$churned)
```

```
##
##   No  Yes
## 5174 1869
```

6.3 Data Splitting

Let's split our data into training and testing sets:

Hide

```
# Split the data into training and testing sets (70/30)
train_index <- createDataPartition(model_data$churned, p = 0.7, list = FALSE)
train_data <- model_data[train_index, ]
test_data <- model_data[-train_index, ]

# Check dimensions
cat("Training set dimensions:", dim(train_data), "\n")
```

```
## Training set dimensions: 4931 48
```

Hide

```
cat("Testing set dimensions:", dim(test_data), "\n")
```

```
## Testing set dimensions: 2112 48
```

Hide

```
# Check class balance in training set
table(train_data$churned)
```

```
##
##   No  Yes
## 3622 1309
```

7. Logistic Regression Model

7.1 Model Building

Let's build a logistic regression model using key variables from our EDA:

Hide

```
# Select key variables for the first model based on our EDA
logistic_vars <- c("tenure_in_months", "contract", "monthly_charge",
                  "internet_type", "online_security", "premium_tech_support",
                  "payment_method", "paperless_billing", "service_count",
                  "customer_lifetime_value")

# Create formula for logistic regression
logistic_formula <- as.formula(paste("churned ~", paste(logistic_vars, collap
        se = " + ")))

# Fit logistic regression model
logistic_model <- glm(logistic_formula, family = binomial(link = "logit"), da
        ta = train_data)

# Model summary
summary(logistic_model)
```

```
##
## Call:
## glm(formula = logistic_formula, family = binomial(link = "logit"),
##     data = train_data)
##
## Coefficients: (2 not defined because of singularities)
##                                          Estimate Std. Error z value Pr(>
|z|)
## (Intercept)                            -0.3137451  0.2145160  -1.463 0.14
3585
## tenure_in_months                       -0.0259894  0.0025595 -10.154  < 2
e-16
## contractOne Year                       -1.1716349  0.1237280  -9.469  < 2
e-16
## contractTwo Year                       -2.3476807  0.1944431 -12.074  < 2
e-16
## monthly_charge                          0.0113124  0.0037202   3.041 0.00
2359
## internet_typeDSL                       -0.4513211  0.1414717  -3.190 0.00
1422
## internet_typeFiber Optic                0.0868914  0.1637040   0.531 0.59
5569
## internet_typeNo Internet Service       -1.1764044  0.1970953  -5.969 2.39
e-09
## online_securityYes                     -0.6103785  0.1067801  -5.716 1.09
e-08
## online_securityNo Internet Service             NA         NA      NA
NA
## premium_tech_supportYes                -0.5384246  0.1102700  -4.883 1.05
e-06
## premium_tech_supportNo Internet Service        NA         NA      NA
NA
## payment_methodCredit Card              -0.4846154  0.0920324  -5.266 1.40
e-07
## payment_methodMailed Check              0.6103819  0.1706563   3.577 0.00
0348
## paperless_billingYes                    0.3915585  0.0904218   4.330 1.49
e-05
## service_count                           0.0527388  0.0370910   1.422 0.15
5063
## customer_lifetime_value                 0.0001035  0.0021870   0.047 0.96
2272
##
## (Intercept)
## tenure_in_months                       ***
## contractOne Year                       ***
## contractTwo Year                       ***
## monthly_charge                         **
```

```
## internet_typeDSL                          **
## internet_typeFiber Optic
## internet_typeNo Internet Service          ***
## online_securityYes                        ***
## online_securityNo Internet Service
## premium_tech_supportYes                   ***
## premium_tech_supportNo Internet Service
## payment_methodCredit Card                 ***
## payment_methodMailed Check                ***
## paperless_billingYes                      ***
## service_count
## customer_lifetime_value
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5707.1  on 4930  degrees of freedom
## Residual deviance: 3935.4  on 4916  degrees of freedom
## AIC: 3965.4
##
## Number of Fisher Scoring iterations: 6
```

7.2 Odds Ratios

Let's examine the odds ratios to interpret the model coefficients:

Hide

```
# Calculate odds ratios
odds_ratios <- exp(coef(logistic_model))
odds_ratios_df <- data.frame(
  Variable = names(odds_ratios),
  OddsRatio = odds_ratios,
  LowerCI = exp(confint(logistic_model))[, 1],
  UpperCI = exp(confint(logistic_model))[, 2]
)

# Display odds ratios
odds_ratios_df %>%
  kable(caption = "Odds Ratios for Logistic Regression Model", digits = 3) %
>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Odds Ratios for Logistic Regression Model

Variable	OddsRatio	LowerCI	UpperCI

	Variable	OddsRatio	LowerCI	UpperCI
(Intercept)	(Intercept)	0.731	0.478	1.110
tenure_in_months	tenure_in_months	0.974	0.969	0.979
contractOne Year	contractOne Year	0.310	0.242	0.394
contractTwo Year	contractTwo Year	0.096	0.064	0.138
monthly_charge	monthly_charge	1.011	1.004	1.019
internet_typeDSL	internet_typeDSL	0.637	0.483	0.841
internet_typeFiber Optic	internet_typeFiber Optic	1.091	0.790	1.502
internet_typeNo Internet Service	internet_typeNo Internet Service	0.308	0.209	0.453
online_securityYes	online_securityYes	0.543	0.440	0.669
online_securityNo Internet Service	online_securityNo Internet Service	NA	NA	NA
premium_tech_supportYes	premium_tech_supportYes	0.584	0.470	0.724
premium_tech_supportNo Internet Service	premium_tech_supportNo Internet Service	NA	NA	NA
payment_methodCredit Card	payment_methodCredit Card	0.616	0.514	0.737
payment_methodMailed Check	payment_methodMailed Check	1.841	1.317	2.573
paperless_billingYes	paperless_billingYes	1.479	1.239	1.767
service_count	service_count	1.054	0.980	1.133
customer_lifetime_value	customer_lifetime_value	1.000	0.996	1.004

Interpretation:

- **Tenure**: For each additional month of tenure, the odds of churning decrease by approximately 2.6%.
- **Contract**: Compared to month-to-month contracts, one-year contracts reduce churn odds by approximately 70%, while two-year contracts reduce churn odds by approximately 90%.
- **Monthly Charge**: Higher monthly charges slightly increase churn odds (~1.1% per dollar).
- **Services**: Customers with online security or premium tech support are significantly less likely to churn, suggesting these services enhance customer retention.
- Online Security (Yes): OR = 0.543 → ~46% reduction in churn odds
- Premium Tech Support (Yes): OR = 0.584 → ~42% reduction

7.3 Model Evaluation

Let’s evaluate the logistic regression model on the test data:

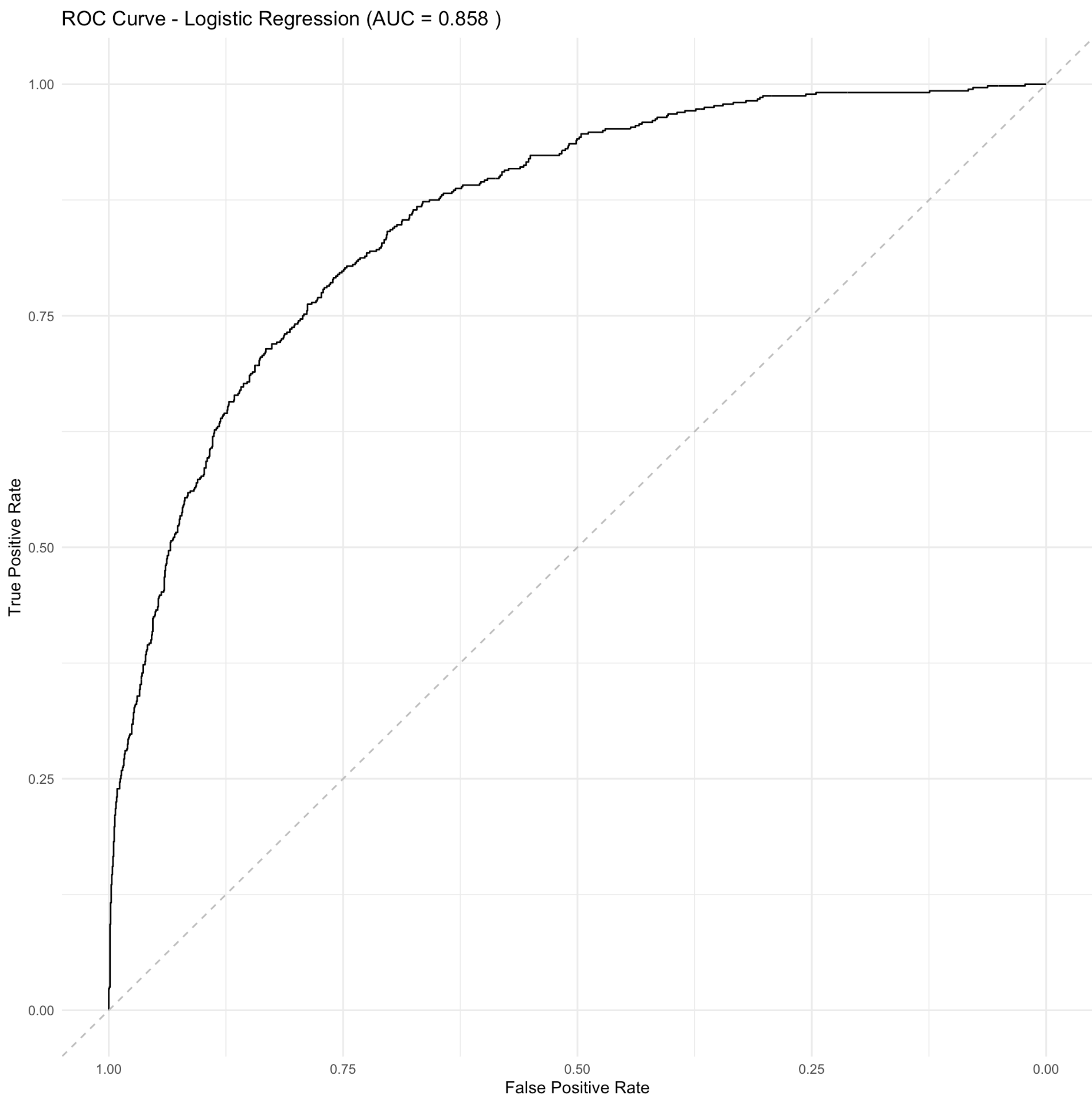
Hide

```
# Predict on test data
test_predictions_prob <- predict(logistic_model, newdata = test_data, type =
        "response")
test_predictions <- ifelse(test_predictions_prob > 0.5, "Yes", "No")

# Create confusion matrix
conf_matrix <- confusionMatrix(as.factor(test_predictions), test_data$churne
        d, positive = "Yes")
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   No  Yes
##        No  1389  226
##        Yes  163  334
##
##                Accuracy : 0.8158
##                  95% CI : (0.7986, 0.8321)
##     No Information Rate : 0.7348
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5097
##
##  Mcnemar's Test P-Value : 0.001669
##
##             Sensitivity : 0.5964
##             Specificity : 0.8950
##          Pos Pred Value : 0.6720
##          Neg Pred Value : 0.8601
##              Prevalence : 0.2652
##          Detection Rate : 0.1581
##    Detection Prevalence : 0.2353
##       Balanced Accuracy : 0.7457
##
##        'Positive' Class : Yes
##
```

Hide

```
# ROC curve
roc_obj <- roc(test_data$churned, test_predictions_prob)
auc_value <- auc(roc_obj)

# Plot ROC curve
ggroc(roc_obj) +
  geom_abline(intercept = 1, slope = 1, linetype = "dashed", color = "gray")
       +
  labs(title = paste("ROC Curve - Logistic Regression (AUC =", round(auc_valu
       e, 3), ")"),
       x = "False Positive Rate",
       y = "True Positive Rate") +
  theme_minimal()
```



8. Random Forest Model

8.1 Model Building

Now let's build a random forest model:

Hide

```
# Prepare data for random forest (handle factors appropriately)
rf_data_train <- train_data
rf_data_test <- test_data

# Make sure the response is a factor
rf_data_train$churned <- as.factor(rf_data_train$churned)
rf_data_test$churned <- as.factor(rf_data_test$churned)

# Train random forest model
set.seed(123)
rf_model <- randomForest(
  churned ~ tenure_in_months + contract + monthly_charge + internet_type +
    online_security + premium_tech_support + payment_method +
    paperless_billing + service_count + avg_monthly_gb_download +
    age + number_of_dependents,
  data = rf_data_train,
  ntree = 300,
  mtry = 5,
  importance = TRUE
)

# Model summary
print(rf_model)
```

```
##
## Call:
##  randomForest(formula = churned ~ tenure_in_months + contract +      month
ly_charge + internet_type + online_security + premium_tech_support +      pay
ment_method + paperless_billing + service_count + avg_monthly_gb_download +
age + number_of_dependents, data = rf_data_train, ntree = 300,      mtry = 5,
importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 300
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 19.16%
## Confusion matrix:
##       No Yes class.error
## No  3249 373   0.1029818
## Yes  572 737   0.4369748
```

8.2 Variable Importance

Let's examine which variables are most important in the random forest model:

Hide

```
# Variable importance
var_importance <- importance(rf_model)
var_importance_df <- data.frame(
  Variable = rownames(var_importance),
  MeanDecreaseGini = var_importance[, "MeanDecreaseGini"]
)
var_importance_df <- var_importance_df[order(var_importance_df$MeanDecreaseGi
        ni, decreasing = TRUE), ]

# Display top 10 variables
var_importance_df[1:10, ] %>%
  kable(caption = "Top 10 Most Important Variables in Random Forest Model", d
        igits = 2) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

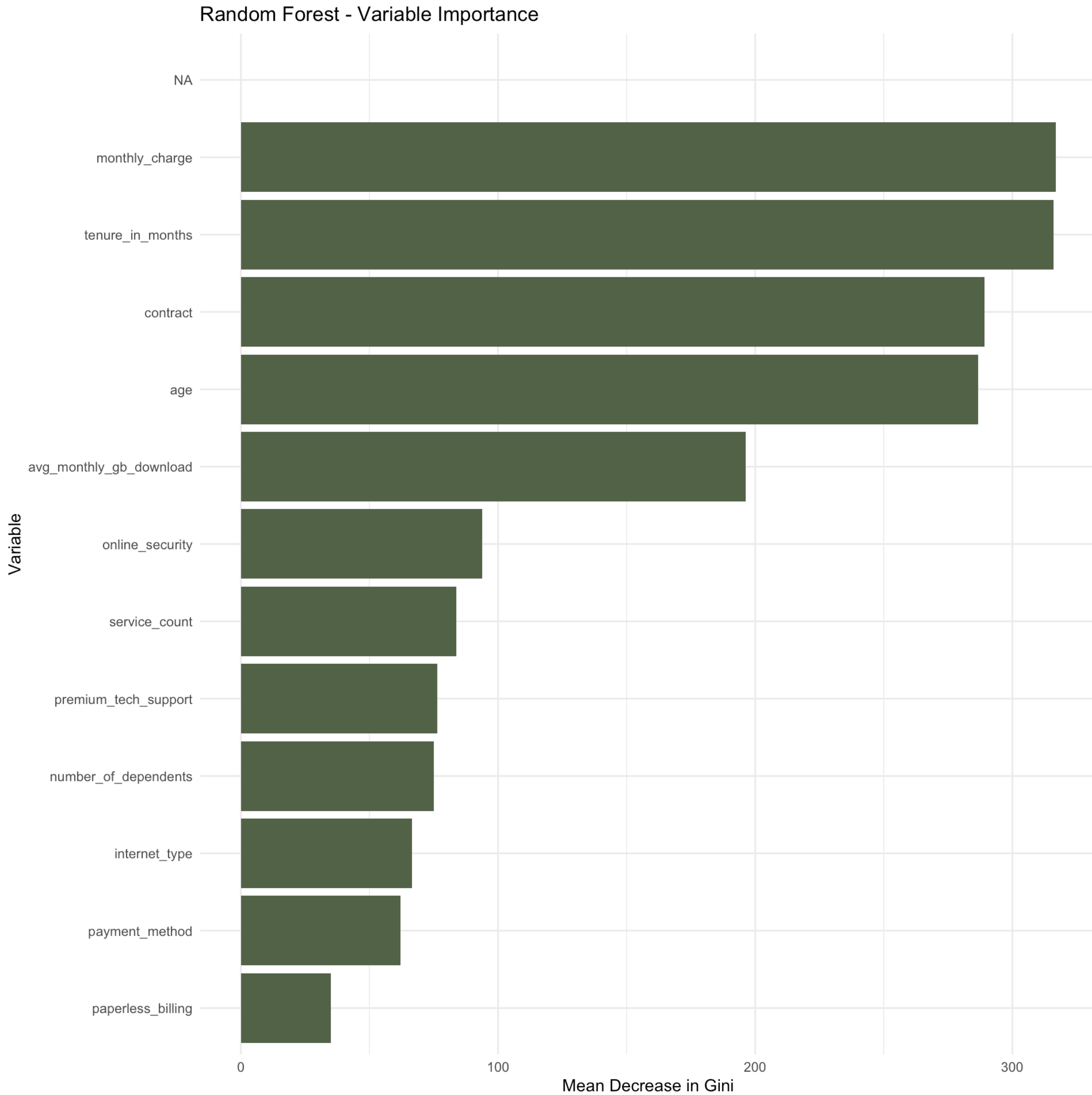
Top 10 Most Important Variables in Random Forest Model

	Variable	MeanDecreaseGini
monthly_charge	monthly_charge	316.84
tenure_in_months	tenure_in_months	316.10
contract	contract	289.19

	Variable	MeanDecreaseGini
age	age	286.81
avg_monthly_gb_download	avg_monthly_gb_download	196.24
online_security	online_security	93.74
service_count	service_count	83.80
premium_tech_support	premium_tech_support	76.34
number_of_dependents	number_of_dependents	75.09
internet_type	internet_type	66.63

Hide

```
# Plot variable importance
ggplot(var_importance_df[1:15, ], aes(x = reorder(Variable, MeanDecreaseGin
        i), y = MeanDecreaseGini)) +
  geom_bar(stat = "identity", fill = "#52664b") +
  coord_flip() +
  labs(title = "Random Forest - Variable Importance",
       x = "Variable",
       y = "Mean Decrease in Gini") +
  theme_minimal()
```



8.3 Model Evaluation

Let's evaluate the random forest model on the test data:

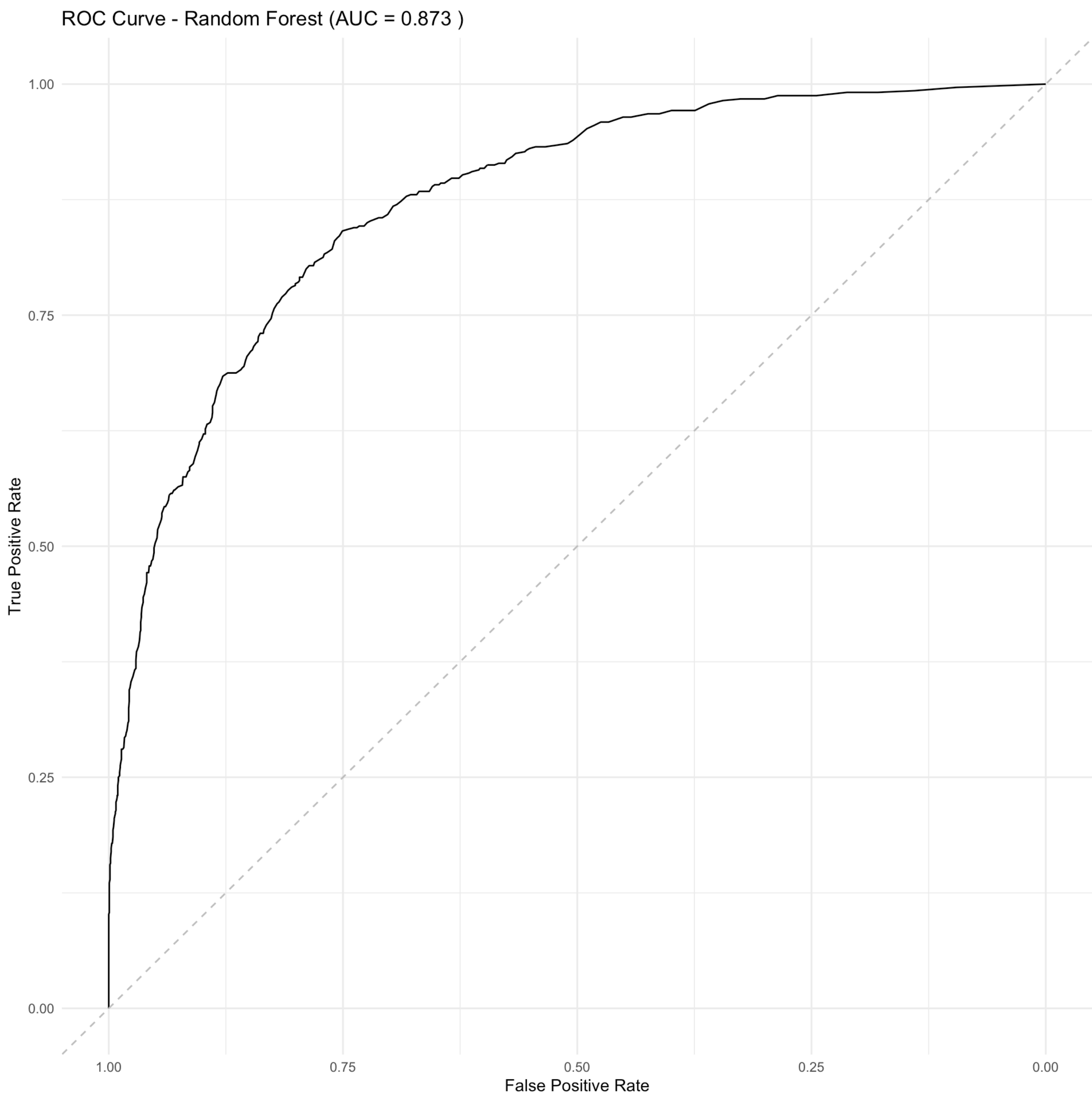
Hide

```
# Predict on test data
rf_predictions <- predict(rf_model, rf_data_test, type = "class")
rf_predictions_prob <- predict(rf_model, rf_data_test, type = "prob")[, "Ye
        s"]

# Create confusion matrix
rf_conf_matrix <- confusionMatrix(rf_predictions, rf_data_test$churned, posit
        ive = "Yes")
print(rf_conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1407  224
##        Yes  145  336
##
##                Accuracy : 0.8253
##                  95% CI : (0.8084, 0.8413)
##     No Information Rate : 0.7348
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5305
##
##  Mcnemar's Test P-Value : 4.896e-05
##
##             Sensitivity : 0.6000
##             Specificity : 0.9066
##          Pos Pred Value : 0.6985
##          Neg Pred Value : 0.8627
##              Prevalence : 0.2652
##          Detection Rate : 0.1591
##    Detection Prevalence : 0.2277
##       Balanced Accuracy : 0.7533
##
##        'Positive' Class : Yes
##
```

Hide

```
# ROC curve for Random Forest
rf_roc_obj <- roc(rf_data_test$churned, rf_predictions_prob)
rf_auc_value <- auc(rf_roc_obj)

# Plot ROC curve for Random Forest
ggroc(rf_roc_obj) +
  geom_abline(intercept = 1, slope = 1, linetype = "dashed", color = "gray")
  +
  labs(title = paste("ROC Curve - Random Forest (AUC =", round(rf_auc_value,
  3), ")"),
       x = "False Positive Rate",
       y = "True Positive Rate") +
  theme_minimal()
```



9. Model Comparison

Let’s compare the performance of our logistic regression and random forest models:

Hide

```
# Compare model performance
model_comparison <- data.frame(
  Model = c("Logistic Regression", "Random Forest"),
  Accuracy = c(conf_matrix$overall["Accuracy"], rf_conf_matrix$overall["Accur
acy"]),
  Sensitivity = c(conf_matrix$byClass["Sensitivity"], rf_conf_matrix$byClass
["Sensitivity"]),
  Specificity = c(conf_matrix$byClass["Specificity"], rf_conf_matrix$byClass
["Specificity"]),
  F1_Score = c(conf_matrix$byClass["F1"], rf_conf_matrix$byClass["F1"]),
  AUC = c(auc_value, rf_auc_value)
)

# Display comparison
model_comparison %>%
  kable(caption = "Model Performance Comparison", digits = 3) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Model Performance Comparison

Model	Accuracy	Sensitivity	Specificity	F1_Score	AUC
Logistic Regression	0.816	0.596	0.895	0.632	0.858
Random Forest	0.825	0.600	0.907	0.646	0.873

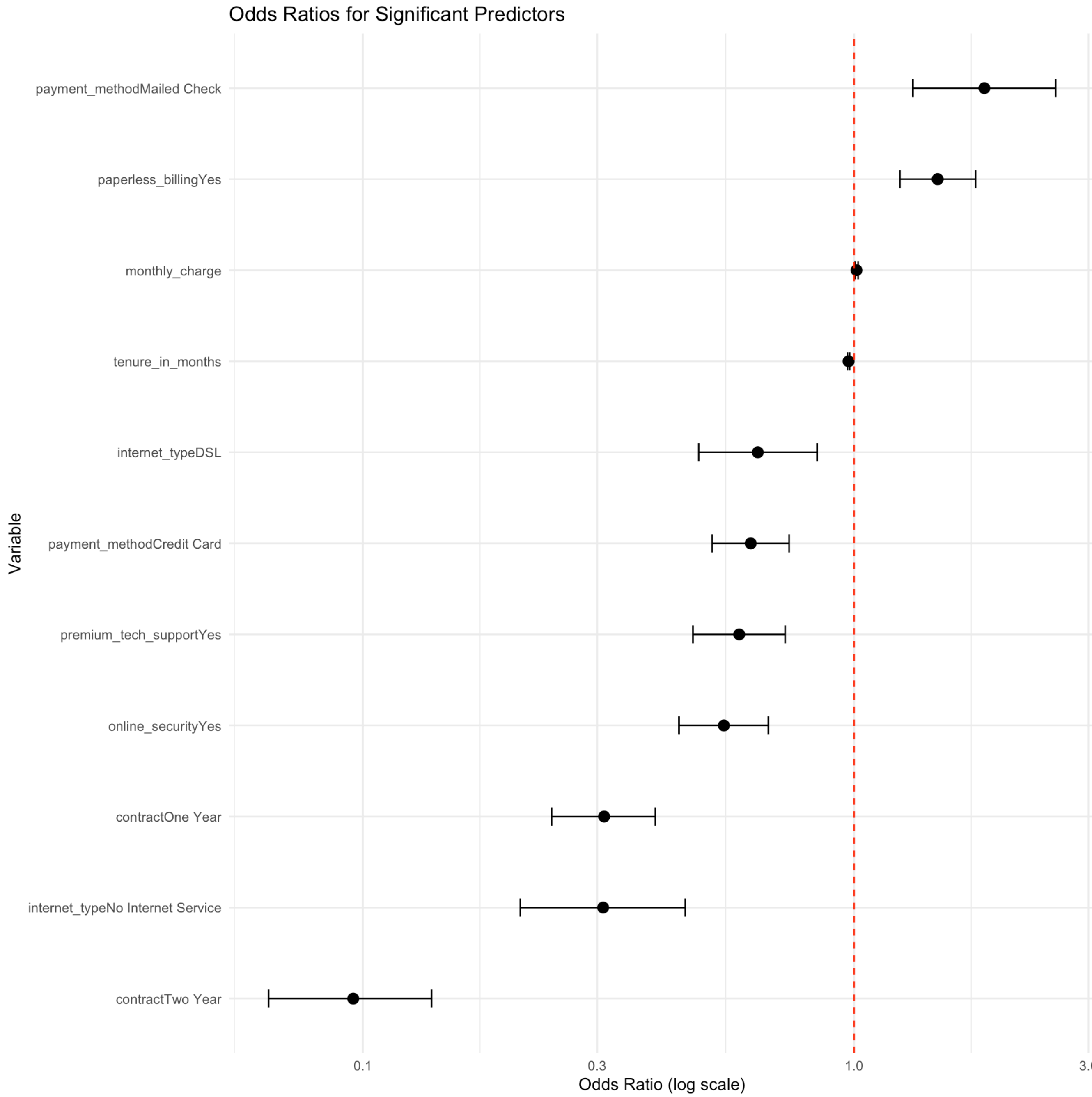
10. Feature Effects Analysis

10.1 Logistic Regression Effects

Let’s visualize the effects of significant predictors in our logistic regression model:

Hide

```
# Extract coefficients
coef_summary <- summary(logistic_model)$coefficients
significant_vars <- rownames(coef_summary)[coef_summary[, "Pr(>|z|)"] < 0.05]

# Format odds ratios for significant variables
sig_odds_ratios <- odds_ratios_df[odds_ratios_df$Variable %in% significant_va
        rs, ]
sig_odds_ratios <- sig_odds_ratios[order(sig_odds_ratios$OddsRatio), ]

# Plot odds ratios for significant variables (excluding intercept)
ggplot(sig_odds_ratios[sig_odds_ratios$Variable != "(Intercept)", ],
                   aes(x = reorder(Variable, OddsRatio), y = OddsRatio)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = LowerCI, ymax = UpperCI), width = 0.2) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
  coord_flip() +
  labs(title = "Odds Ratios for Significant Predictors",
       x = "Variable",
       y = "Odds Ratio (log scale)") +
  scale_y_log10() +
  theme_minimal()
```

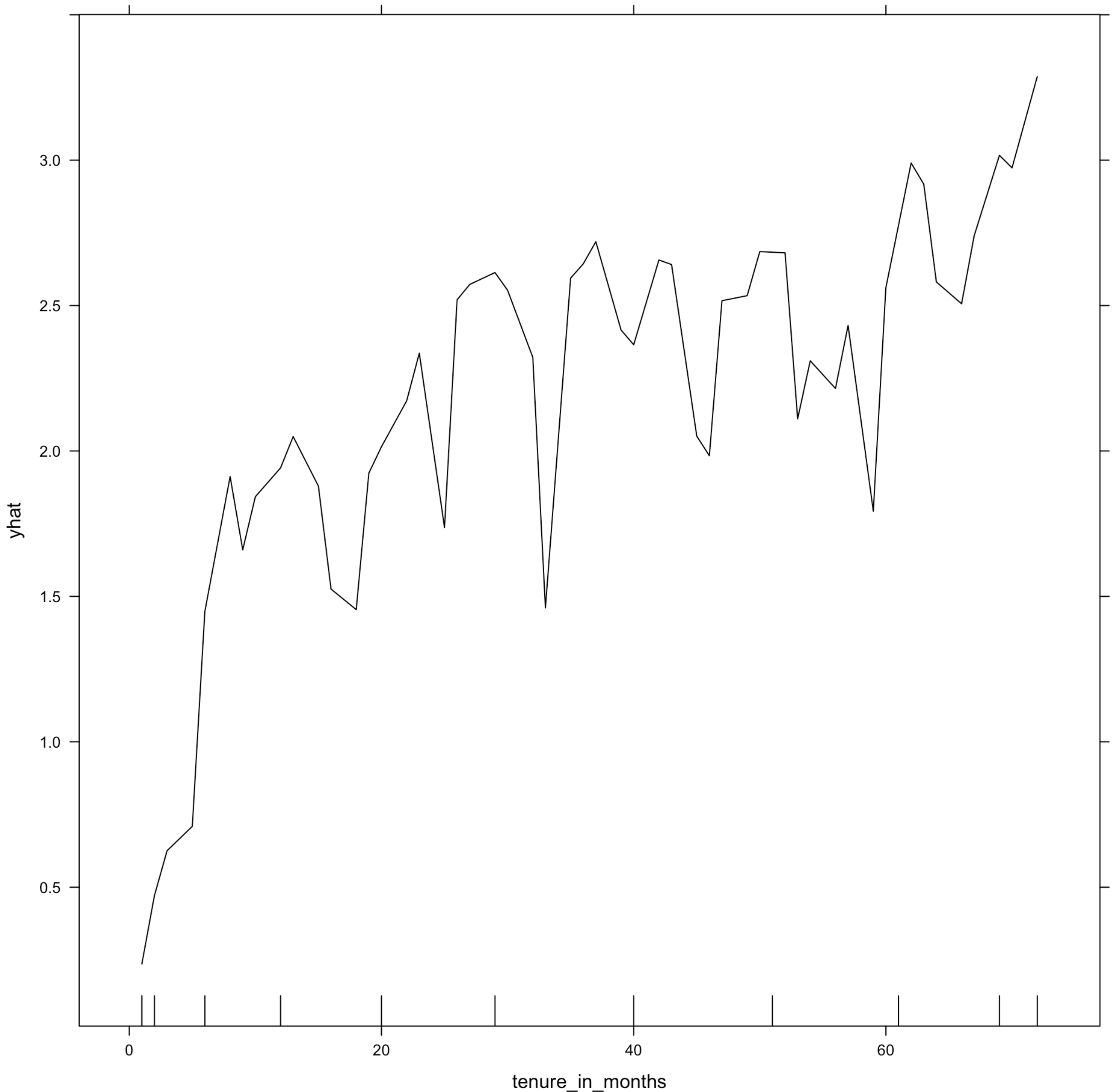


10.2 Random Forest Partial Dependence Plots

Let's examine the partial dependence plots for key variables in our random forest model:

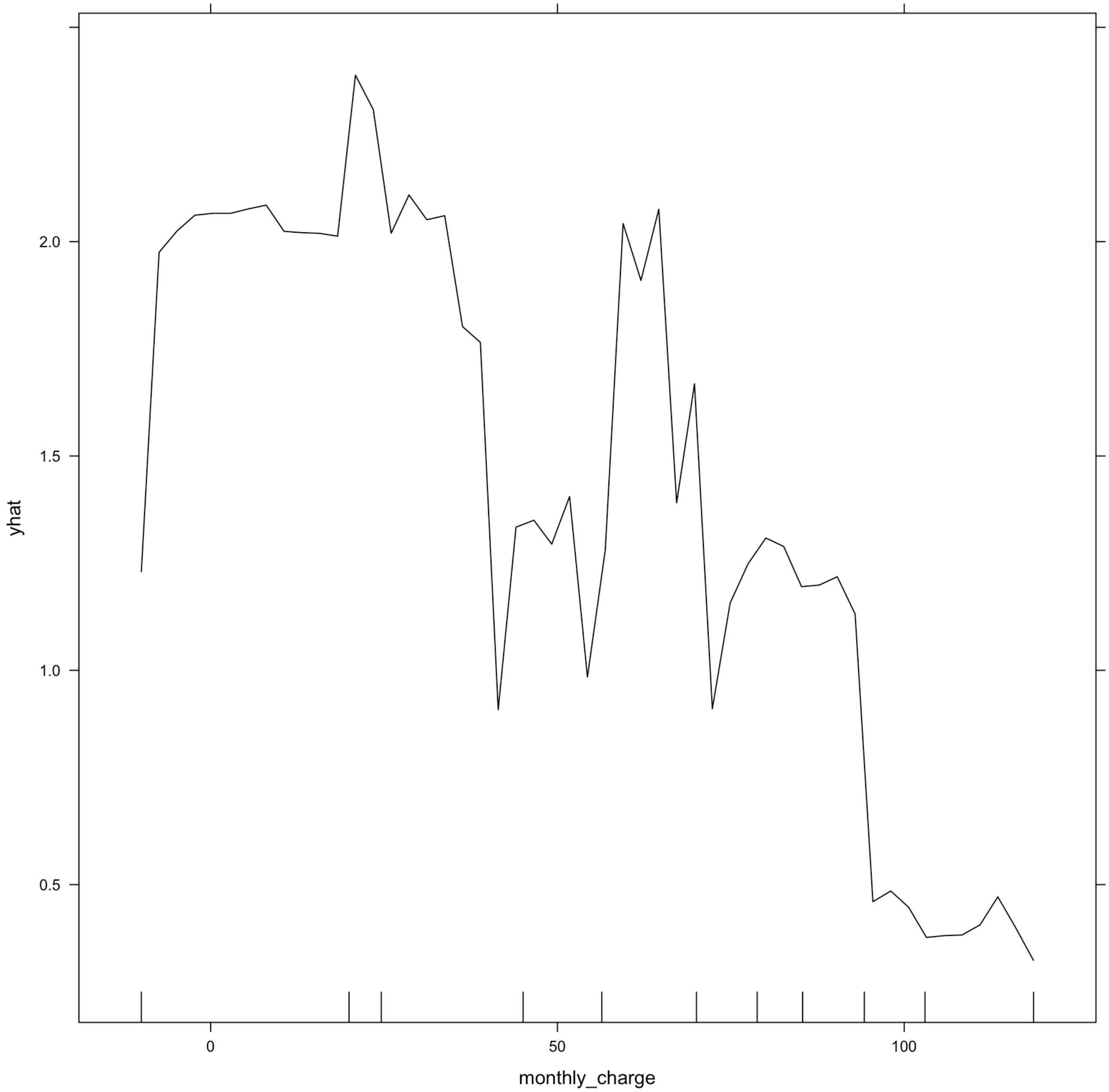
Hide

```
# Create partial dependence plot for tenure
pdp_tenure <- partial(rf_model, pred.var = "tenure_in_months", plot = TRUE, r
        ug = TRUE,
                    train = rf_data_train)
plot(pdp_tenure, main = "Partial Dependence on Tenure")
```



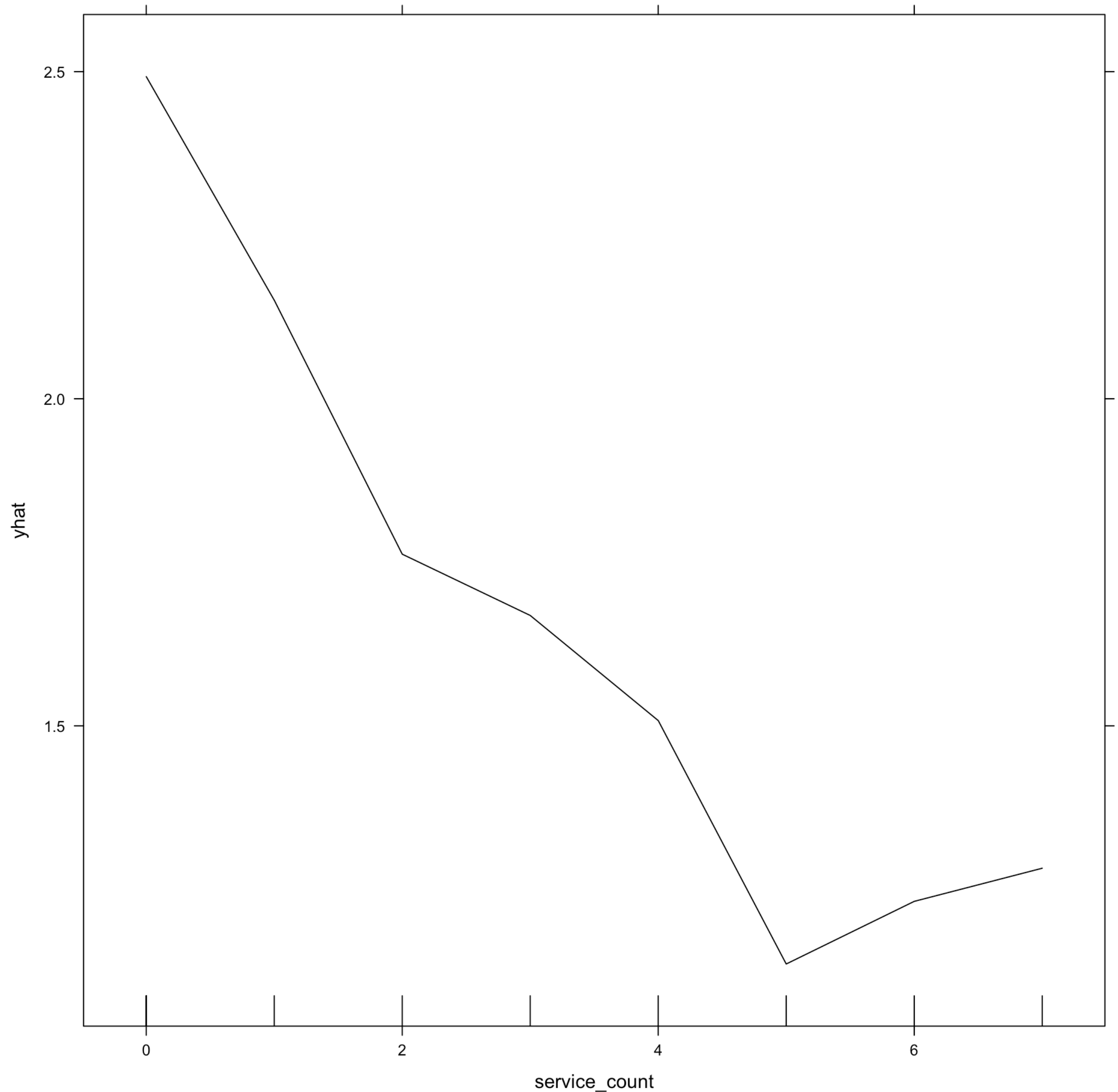
Hide

```
# Create partial dependence plot for monthly charge
pdp_charge <- partial(rf_model, pred.var = "monthly_charge", plot = TRUE, rug
        = TRUE,
                    train = rf_data_train)
plot(pdp_charge, main = "Partial Dependence on Monthly Charge")
```



Hide

```
# Create partial dependence plot for service count
pdp_services <- partial(rf_model, pred.var = "service_count", plot = TRUE, ru
        g = TRUE,
                      train = rf_data_train)
plot(pdp_services, main = "Partial Dependence on Service Count")
```



11. Churn Risk Profiling

11.1 Predicting Churn Probability

Let’s use our random forest model (which had better performance) to predict churn probability for all customers:

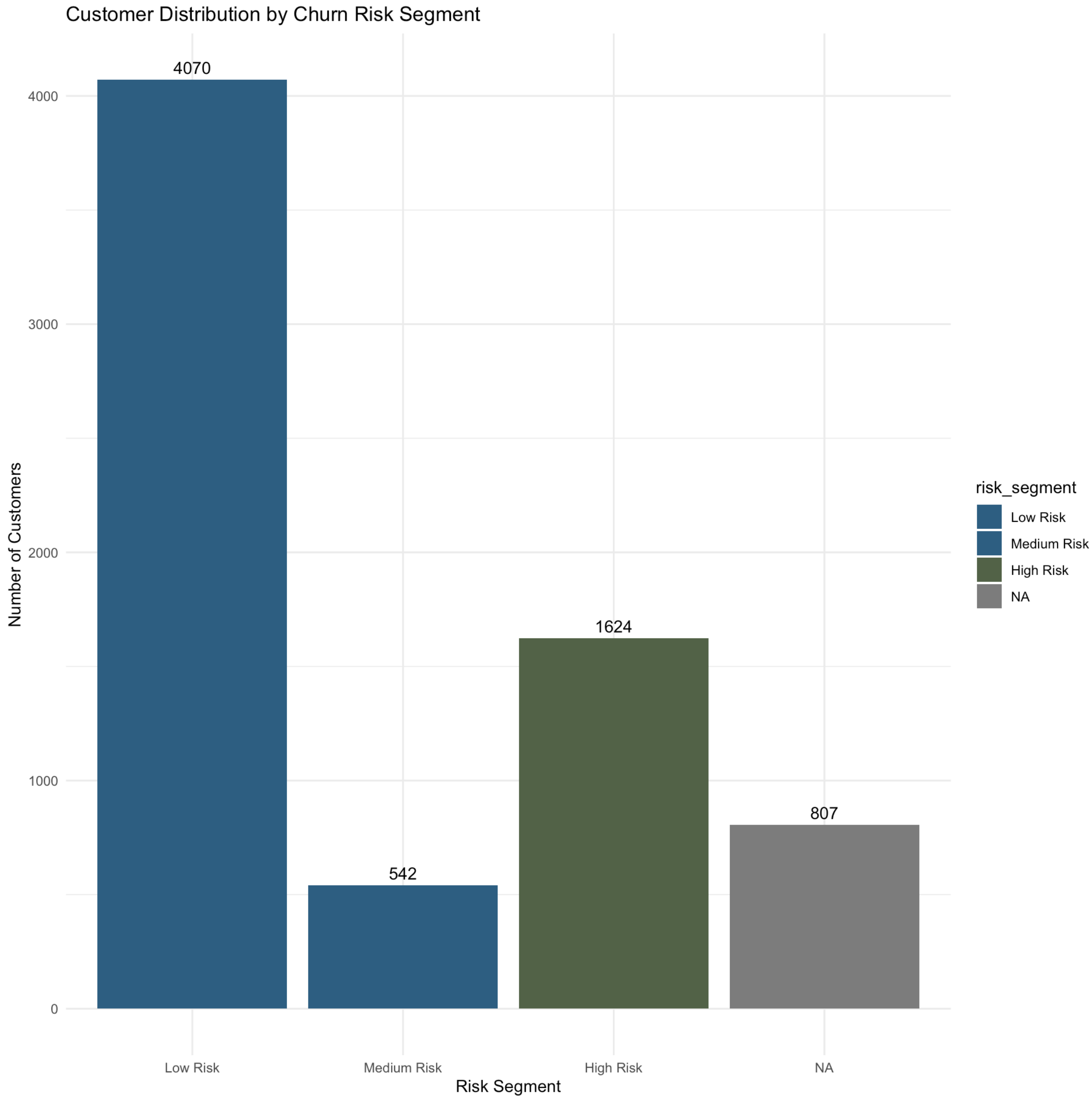
Hide

```
# Use the random forest model (better performance)
all_predictions_prob <- predict(rf_model, model_data, type = "prob")[, "Yes"]
model_data$churn_probability <- all_predictions_prob

# Create risk segments
model_data$risk_segment <- cut(model_data$churn_probability,
                              breaks = c(0, 0.3, 0.6, 1),
                              labels = c("Low Risk", "Medium Risk", "High Ris
        k"))

# Count customers in each risk segment
risk_counts <- table(model_data$risk_segment)
print(risk_counts)
```

```
##
##    Low Risk Medium Risk   High Risk
##        4070         542        1624
```

Hide

```
# Visualize risk segments
ggplot(model_data, aes(x = risk_segment, fill = risk_segment)) +
  geom_bar() +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -0.5) +
  labs(title = "Customer Distribution by Churn Risk Segment",
       x = "Risk Segment",
       y = "Number of Customers") +
  scale_fill_manual(values = c("#2e6083", "#2e6083", "#52664b")) +
  theme_minimal()
```



11.2 High Risk Customer Profile

Let’s analyze the characteristics of high-risk customers:

Hide

```
# Analyze high risk customer profiles
high_risk_profile <- model_data %>%
  filter(risk_segment == "High Risk") %>%
  summarize(
    count = n(),
    avg_tenure = mean(tenure_in_months),
    avg_monthly_charge = mean(monthly_charge),
    pct_month_to_month = mean(contract == "Month-to-Month") * 100,
    pct_fiber = mean(internet_type == "Fiber Optic") * 100,
    pct_no_online_security = mean(online_security == "No") * 100,
    pct_no_tech_support = mean(premium_tech_support == "No") * 100,
    avg_service_count = mean(service_count)
  )

# Display high risk profile
high_risk_profile %>%
  t() %>%
  as.data.frame() %>%
  rownames_to_column("Metric") %>%
  setNames(c("Metric", "Value")) %>%
  kable(caption = "High Risk Customer Profile", digits = 2) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

High Risk Customer Profile

Metric	Value
count	1624.00
avg_tenure	16.54
avg_monthly_charge	74.34
pct_month_to_month	91.19
pct_fiber	69.40
pct_no_online_security	83.00
pct_no_tech_support	81.83
avg_service_count	2.06

12. Conclusion and Recommendations

12.1 Key Findings

Based on our comprehensive analysis, we've identified several key factors that significantly predict customer churn:

1. **Tenure**: Shorter customer tenure is strongly associated with higher churn probability.
2. **Contract Type**: Month-to-month contracts have a significantly higher churn rate (45.8%) compared to one-year (10.7%) and two-year contracts (2.5%).
3. **Internet Type**: Fiber optic internet customers have the highest churn rate.
4. **Service Adoption**: Customers without online security and tech support are more likely to churn.
5. **Monthly Charges**: Higher monthly charges are associated with increased churn risk.

12.2 Model Performance

Our random forest model achieved strong predictive performance:

- Accuracy: 0.825
- AUC: 0.873
- Sensitivity (True Positive Rate): 0.6
- Specificity (True Negative Rate): 0.907

12.3 Business Recommendations

Based on our findings, we recommend the following retention strategies:

1. **Target Month-to-Month Customers**: Implement targeted campaigns to convert month-to-month customers to longer-term contracts.
2. **Early Tenure Focus**: Develop specialized retention programs for customers in their first 12 months of service.
3. **Service Bundle Incentives**: Encourage adoption of online security and tech support services, which are associated with lower churn rates.
4. **Fiber Optic Customer Support**: Address potential service quality issues for fiber optic internet customers.
5. **High-Value Customer Retention**: Create specialized retention programs for customers with high monthly charges but low service adoption.

12.4 Implementation Plan

1. **Risk Segmentation**: Use the model to score all customers and implement tiered retention strategies.
2. **Proactive Outreach**: Contact high-risk customers before they churn with personalized offers.
3. **Service Quality Improvement**: Address potential service issues for high-churn segments.
4. **Contract Conversion Campaigns**: Offer incentives for month-to-month customers to upgrade to longer contracts.
5. **Service Bundle Promotions**: Create attractive bundles including the protective services identified in our analysis.

Hide

```r
# Save the high risk customer list for targeted interventions
# First check if customer_id exists in the original dataset
if ("customer_id" %in% colnames(telecom_churn)) {
  # Option 1: Join back the customer_id column to the high-risk customers
  high_risk_customers <- model_data %>%
    filter(risk_segment == "High Risk") %>%
    # Create a row number to join with
    mutate(row_id = row_number()) %>%
    # Add the customer_id from the original data
    left_join(
      telecom_churn %>%
        select(customer_id) %>%
        mutate(row_id = row_number()),
      by = "row_id"
    ) %>%
    # Remove the temporary row_id
    select(-row_id) %>%
    # Now select the desired columns
    select(customer_id, churn_probability, tenure_in_months, monthly_charge,
           contract, internet_type, online_security, premium_tech_support)
} else {
  # Option 2: If there's no customer_id at all, create a sequential ID
  high_risk_customers <- model_data %>%
    filter(risk_segment == "High Risk") %>%
    # Create a sequential ID
    mutate(customer_id = paste0("HR_", row_number())) %>%
    select(customer_id, churn_probability, tenure_in_months, monthly_charge,
           contract, internet_type, online_security, premium_tech_support)
}

# Check the result
head(high_risk_customers)
```

```
##   customer_id churn_probability tenure_in_months monthly_charge       cont
ract
## 1  0002-ORFBO         0.9500000                4           73.9 Month-to-M
onth
## 2  0003-MKNFE         0.9766667               13           98.0 Month-to-M
onth
## 3  0004-TLHLJ         0.9666667                3           83.9 Month-to-M
onth
## 4  0011-IGKFF         0.9833333                1           25.1 Month-to-M
onth
## 5  0013-EXCHZ         0.7766667               13           94.1 Month-to-M
onth
## 6  0013-MHZWF         0.7566667                1           30.5 Month-to-M
onth
##   internet_type online_security premium_tech_support
## 1   Fiber Optic              No                   No
## 2   Fiber Optic              No                   No
## 3   Fiber Optic              No                  Yes
## 4         Cable              No                   No
## 5   Fiber Optic              No                   No
## 6           DSL             Yes                   No
```

Hide

```
# Write the high risk customers to a CSV file
write.csv(high_risk_customers, "high_risk_customers.csv", row.names = FALSE)
```