Telecom Customer Churn Prediction Analysis

[Your Name]

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

```
# 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
```

```r
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

```r
# Read the datasets
telecom_churn <- read.csv("telecom_customer_churn.csv", stringsAsFactors = TRUE)
zipcode_population <- read.csv("telecom_zipcode_population.csv")
data_dictionary <- read.csv("telecom_data_dictionary.csv", encoding = "CP1252")

# 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:

```r
# Display the structure of the first few columns
str(telecom_churn[, 1:10])
```

```
## 'data.frame':    7043 obs. of  10 variables:
##  $ customer_id         : Factor w/ 7043 levels "0002-ORFBO","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","Acton",..: 347 369 223 588 140 609 547 654 9
##  $ zip_code            : int  93225 91206 92627 94553 93010 95345 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 ...
```

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

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

```r
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.

```
# 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

```
# 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_charges")])
```

```
##       age        tenure_in_months number_of_dependents avg_monthly_gb_download
##  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

```
# 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
```

```r
# Handle missing values for avg_monthly_gb_download using median imputation
telecom_churn$avg_monthly_gb_download[is.na(telecom_churn$avg_monthly_gb_download)] <-
  median(telecom_churn$avg_monthly_gb_download, 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. We'll handle these appropriately:

```r
# 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 Int
}

# 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
}

# Check if there are still missing values
missing_values_after <- colSums(is.na(telecom_churn))
missing_values_after[missing_values_after > 0]
```

```
## avg_monthly_long_distance_charges
##                               682
```

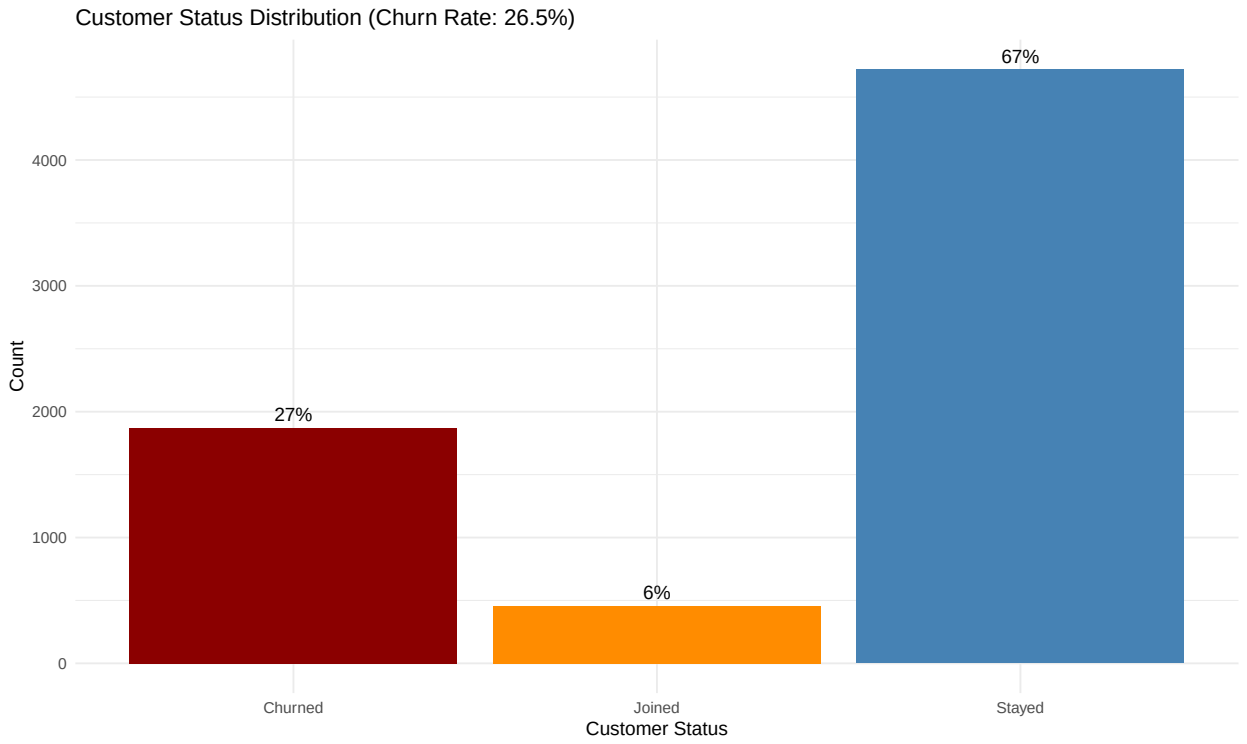
4. Exploratory Data Analysis

4.1 Overall Churn Rate

```
# 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 %
```

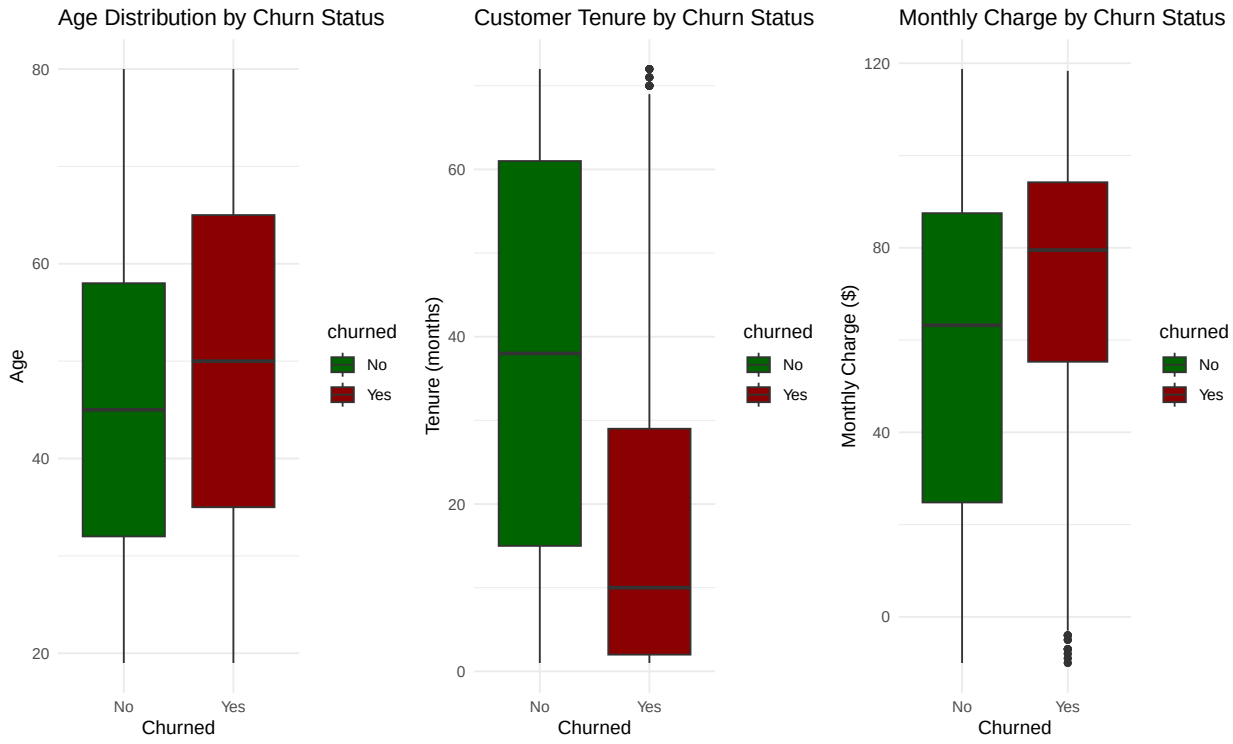
```
# Visualize the churn distribution
ggplot(telecom_churn, aes(x = customer_status)) +
  geom_bar(fill = c("darkred", "darkorange", "steelblue")) +
  geom_text(stat = "count", aes(label = scales::percent(after_stat(count)/sum(after_stat(count)))),
            vjust = -0.5) +
  labs(title = paste0("Customer Status Distribution (Churn Rate: ", round(churn_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:

```r
# 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("darkgreen", "darkred")) +
  theme_minimal()

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

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

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



```r
# 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"))
```

Table 1: 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:

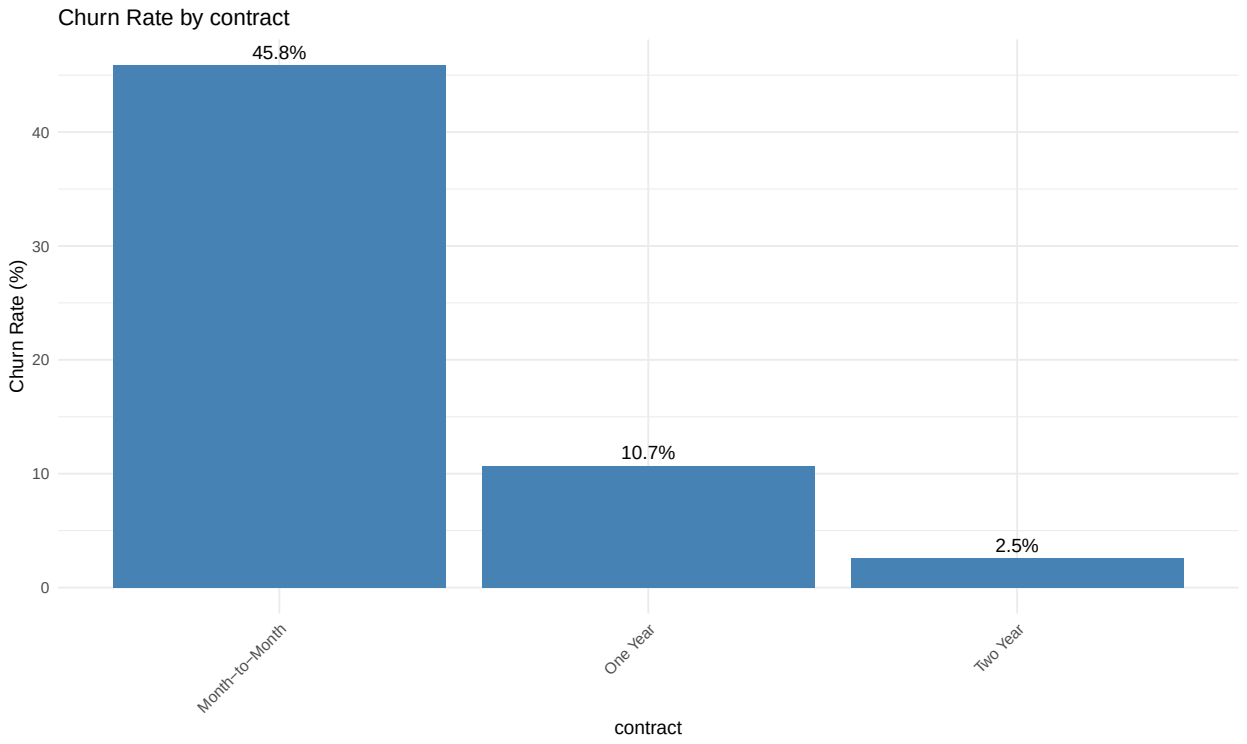
1. **Age**: Churned customers tend to be slightly older on average.
2. **Tenure**: There's a substantial difference in tenure between churned and retained customers. Customers who churn have much shorter tenure on average.
3. **Monthly Charge**: Churned customers have higher monthly charges on average.

4.3 Categorical Variables and Churn

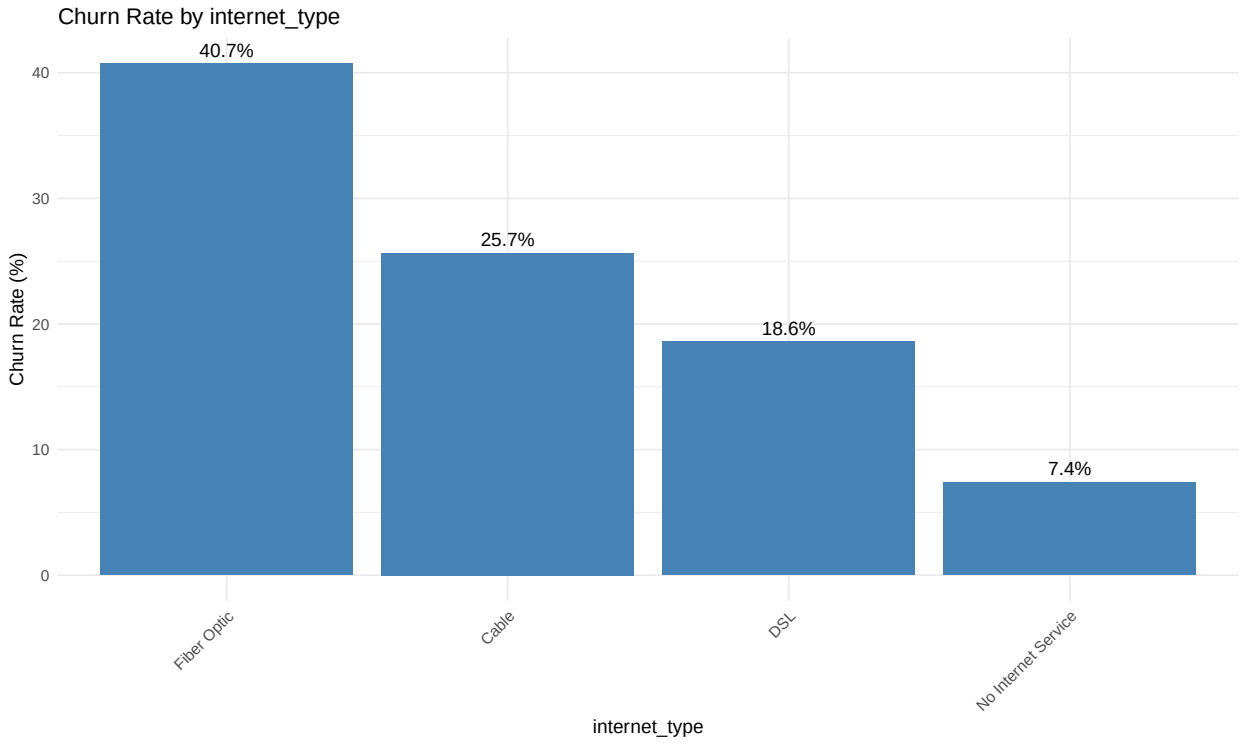
Let's analyze how categorical variables relate to churn:

```r
# 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 = churn_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() +
```

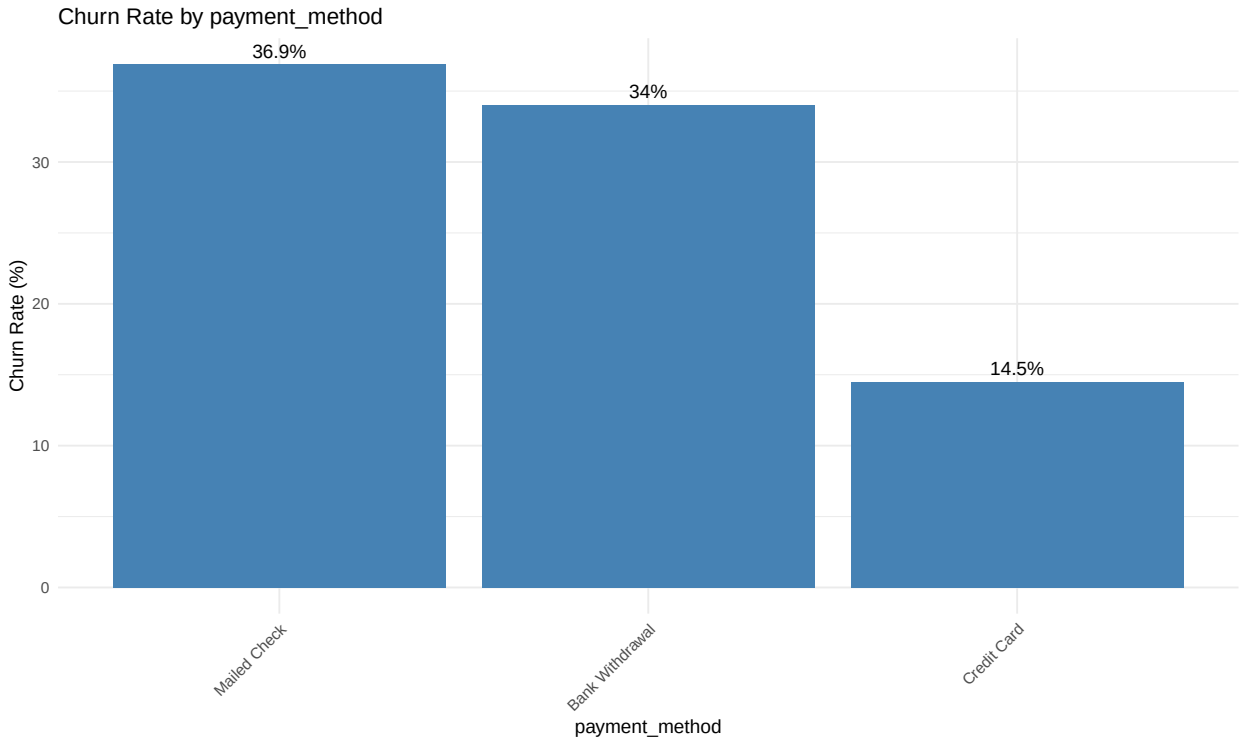
```
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

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




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




```
# 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 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:

```
# 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 Status") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

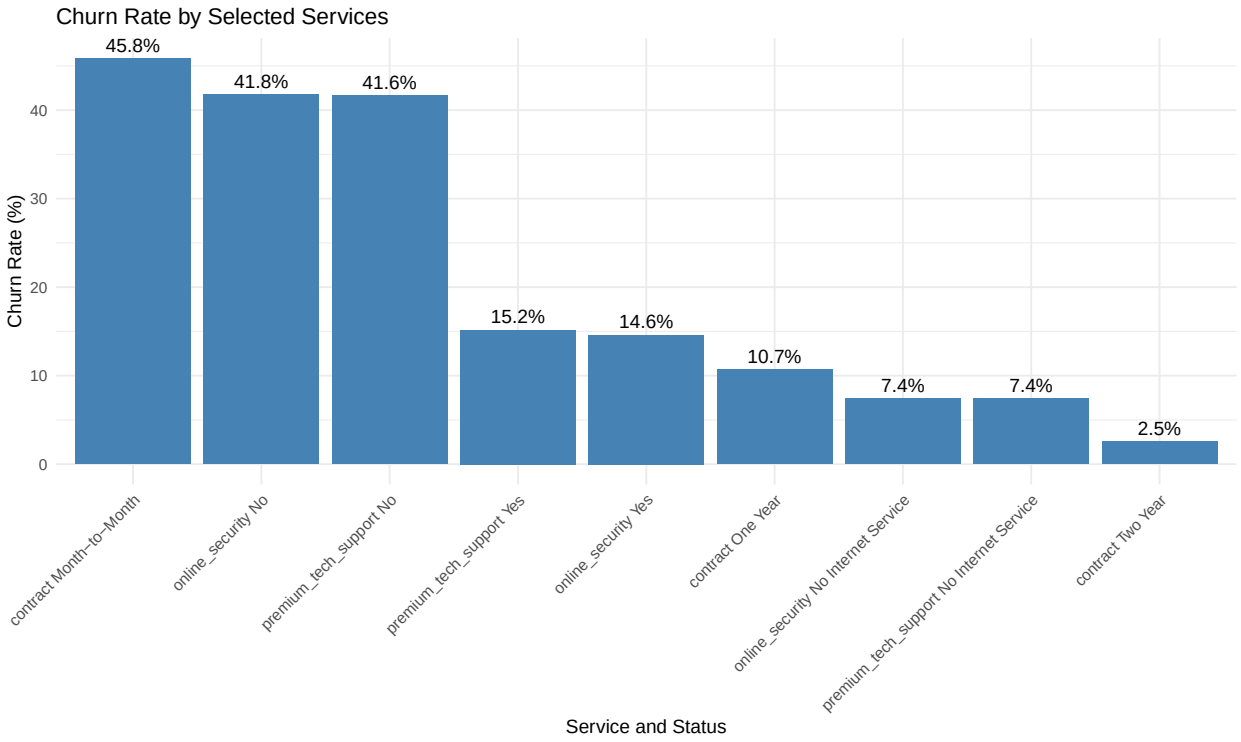
Table 2: Churn Rate by Online Security Status

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:

```
# 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
)
```

```r
# 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), -churn_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))
```



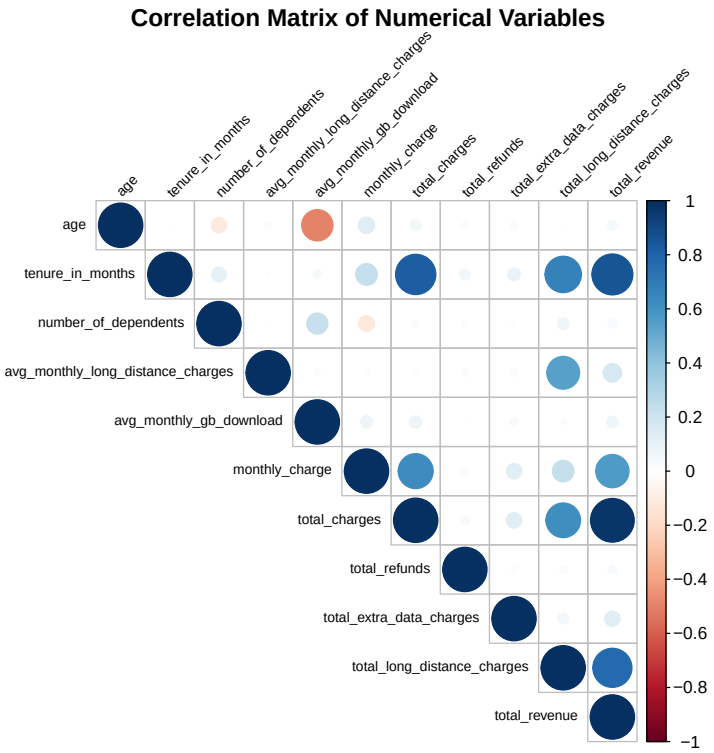
Observations:

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:

```
# Select numerical columns for correlation analysis
numeric_cols <- telecom_churn %>%
  select(age, tenure_in_months, number_of_dependents, avg_monthly_long_distance_charges,
         avg_monthly_gb_download, monthly_charge, total_charges, total_refunds,
         total_extra_data_charges, total_long_distance_charges, total_revenue) %>%
  names()

# Calculate correlation matrix
correlation_matrix <- cor(telecom_churn[numeric_cols], use = "pairwise.complete.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 and total long distance charges.

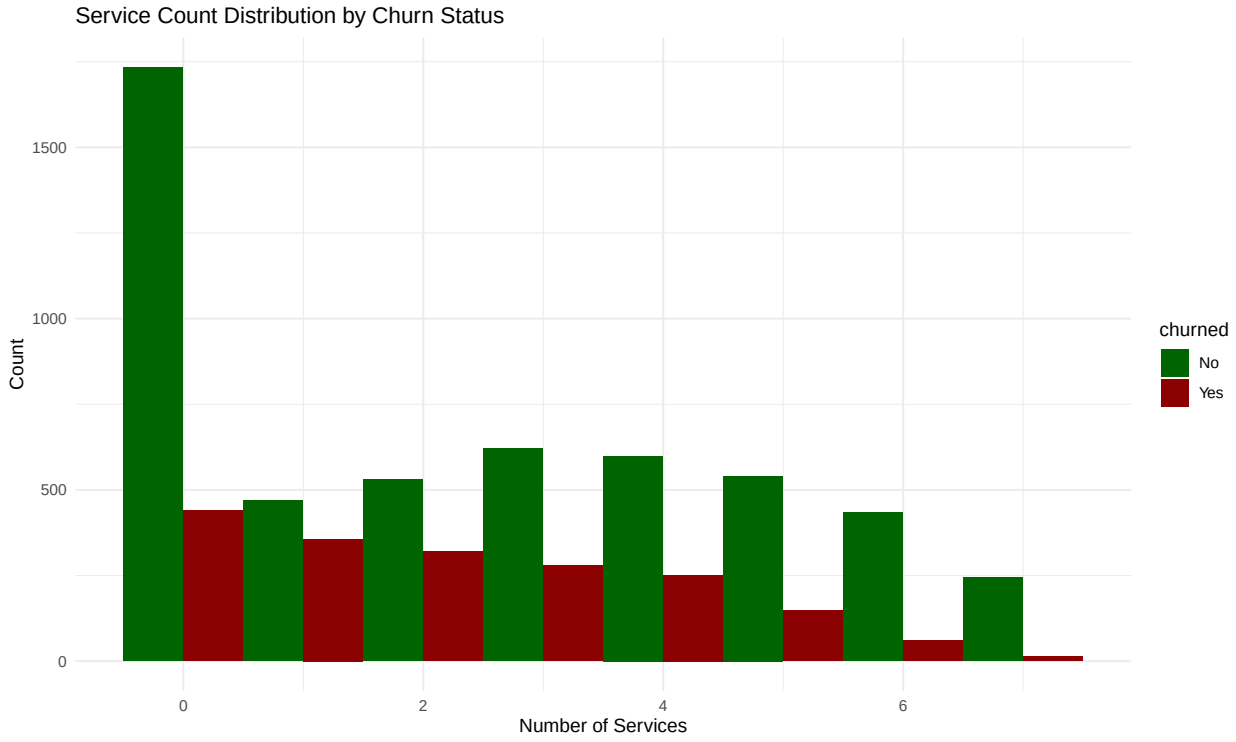
5. Feature Engineering

5.1 Create Derived Features

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

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

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

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

# Visualize service count distribution by churn status
```

```r
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("darkgreen", "darkred")) +
  theme_minimal()
```



5.2 Add Population Data

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

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

# 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"))
```

Table 3: 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:

```
# 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] # Remove 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"
```

```
# 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_encoded), 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:

```r
# 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:

```r
# 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
```

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

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

```r
# 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:

```r
# 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, collapse = " + ")))

# Fit logistic regression model
logistic_model <- glm(logistic_formula, family = binomial(link = "logit"), data = 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 0.143585
## tenure_in_months                       -0.0259894  0.0025595 -10.154  < 2e-16
## contractOne Year                       -1.1716349  0.1237280  -9.469  < 2e-16
## contractTwo Year                       -2.3476807  0.1944431 -12.074  < 2e-16
## monthly_charge                          0.0113124  0.0037202   3.041 0.002359
## internet_typeDSL                       -0.4513211  0.1414717  -3.190 0.001422
## internet_typeFiber Optic                0.0868914  0.1637040   0.531 0.595569
## internet_typeNo Internet Service       -1.1764044  0.1970953  -5.969 2.39e-09
## online_securityYes                     -0.6103785  0.1067801  -5.716 1.09e-08
## online_securityNo Internet Service             NA         NA      NA       NA
## premium_tech_supportYes                -0.5384246  0.1102700  -4.883 1.05e-06
## premium_tech_supportNo Internet Service        NA         NA      NA       NA
## payment_methodCredit Card              -0.4846154  0.0920324  -5.266 1.40e-07
## payment_methodMailed Check              0.6103819  0.1706563   3.577 0.000348
## paperless_billingYes                    0.3915585  0.0904218   4.330 1.49e-05
## service_count                           0.0527388  0.0370910   1.422 0.155063
## customer_lifetime_value                 0.0001035  0.0021870   0.047 0.962272
##
## (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              ***
```