

Telco Predicting Customer Churn

Nicole

2025-08-23

```
library("dplyr")
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)  
library(dplyr)  
library(scales)  
library(patchwork)  
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##  
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':  
##  
##   cov, smooth, var
```

```
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
library(randomForest)
```

```
## 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
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##     combine
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##     recode
```

```
head(telco_chrun)
```

```
##   customerID gender SeniorCitizen Partner Dependents tenure PhoneService
## 1 7590-VHVEG Female             0      Yes          No        1           No
## 2 5575-GNVDE  Male             0      No           No       34           Yes
## 3 3668-QPYBK  Male             0      No           No        2           Yes
## 4 7795-CFOCW  Male             0      No           No       45           No
## 5 9237-HQITU Female            0      No           No        2           Yes
## 6 9305-CDSKC Female            0      No           No        8           Yes
##   MultipleLines InternetService OnlineSecurity OnlineBackup DeviceProtection
## 1 No phone service          DSL              No          Yes              No
## 2                No          DSL              Yes         No              Yes
## 3                No          DSL              Yes         Yes              No
## 4 No phone service          DSL              Yes         No              Yes
## 5                No    Fiber optic            No         No              No
## 6                Yes    Fiber optic            No         No              Yes
##   TechSupport StreamingTV StreamingMovies      Contract PaperlessBilling
```

## 1	No	No	No Month-to-month	Yes
## 2	No	No	No One year	No
## 3	No	No	No Month-to-month	Yes
## 4	Yes	No	No One year	No
## 5	No	No	No Month-to-month	Yes
## 6	No	Yes	Yes Month-to-month	Yes
##	PaymentMethod	MonthlyCharges	TotalCharges	Churn
## 1	Electronic check	29.85	29.85	No
## 2	Mailed check	56.95	1889.50	No
## 3	Mailed check	53.85	108.15	Yes
## 4	Bank transfer (automatic)	42.30	1840.75	No
## 5	Electronic check	70.70	151.65	Yes
## 6	Electronic check	99.65	820.50	Yes

#Understanding the nature of the variables (appropriate data types)

```
summary(telco_chrun)
```

```
## customerID          gender          SeniorCitizen      Partner
## Length:7043         Length:7043         Min.   :0.0000      Length:7043
## Class :character     Class :character    1st Qu.:0.0000      Class :character
## Mode  :character     Mode  :character    Median :0.0000      Mode  :character
##                                     Mean   :0.1621
##                                     3rd Qu.:0.0000
##                                     Max.   :1.0000
##
## Dependents          tenure          PhoneService        MultipleLines
## Length:7043         Min.   : 0.00      Length:7043         Length:7043
## Class :character     1st Qu.: 9.00      Class :character     Class :character
## Mode  :character     Median :29.00      Mode  :character     Mode  :character
##                                     Mean   :32.37
##                                     3rd Qu.:55.00
##                                     Max.   :72.00
##
## InternetService     OnlineSecurity      OnlineBackup         DeviceProtection
## Length:7043         Length:7043         Length:7043         Length:7043
## Class :character     Class :character     Class :character     Class :character
## Mode  :character     Mode  :character     Mode  :character     Mode  :character
##
##
##
## TechSupport         StreamingTV          StreamingMovies       Contract
## Length:7043         Length:7043         Length:7043         Length:7043
## Class :character     Class :character     Class :character     Class :character
## Mode  :character     Mode  :character     Mode  :character     Mode  :character
##
##
##
## PaperlessBilling    PaymentMethod        MonthlyCharges        TotalCharges
## Length:7043         Length:7043         Min.   : 18.25      Min.   : 18.8
## Class :character     Class :character     1st Qu.: 35.50      1st Qu.: 401.4
```

```
## Mode :character Mode :character Median : 70.35 Median :1397.5
## Mean : 64.76 Mean :2283.3
## 3rd Qu.: 89.85 3rd Qu.:3794.7
## Max. :118.75 Max. :8684.8
## NA's :11
## Churn
## Length:7043
## Class :character
## Mode :character
##
##
##
```

```
telco_chrun$gender <- as.factor(telco_chrun$gender)
telco_chrun$SeniorCitizen <- as.factor(telco_chrun$SeniorCitizen)
telco_chrun$Partner <- as.factor(telco_chrun$Partner)
telco_chrun$Dependents <- as.factor(telco_chrun$Dependents)
telco_chrun$PhoneService <- as.factor(telco_chrun$PhoneService)
telco_chrun$MultipleLines <- as.factor(telco_chrun$MultipleLines)
telco_chrun$InternetService <- as.factor(telco_chrun$InternetService)
telco_chrun$OnlineSecurity <- as.factor(telco_chrun$OnlineSecurity)
telco_chrun$DeviceProtection <- as.factor(telco_chrun$DeviceProtection)
telco_chrun$OnlineBackup <- as.factor(telco_chrun$OnlineBackup)
telco_chrun$TechSupport <- as.factor(telco_chrun$TechSupport)
telco_chrun$StreamingTV <- as.factor(telco_chrun$StreamingTV)
telco_chrun$StreamingMovies <- as.factor(telco_chrun$StreamingMovies)
telco_chrun$Contract <- as.factor(telco_chrun$Contract)
telco_chrun$PaperlessBilling <- as.factor(telco_chrun$PaperlessBilling)
telco_chrun$PaymentMethod <- as.factor(telco_chrun$PaymentMethod)
telco_chrun$Churn <- as.factor(telco_chrun$Churn)

summary(telco_chrun)
```

```
## customerID gender SeniorCitizen Partner Dependents
## Length:7043 Female:3488 0:5901 No :3641 No :4933
## Class :character Male :3555 1:1142 Yes:3402 Yes:2110
## Mode :character
##
##
##
## tenure PhoneService MultipleLines InternetService
## Min. : 0.00 No : 682 No :3390 DSL :2421
## 1st Qu.: 9.00 Yes:6361 No phone service: 682 Fiber optic:3096
## Median :29.00 Yes :2971 No :1526
## Mean :32.37
## 3rd Qu.:55.00
## Max. :72.00
##
## OnlineSecurity OnlineBackup
## No :3498 No :3088
## No internet service:1526 No internet service:1526
## Yes :2019 Yes :2429
```

```
##
##
##
##
##      DeviceProtection      TechSupport
## No      :3095      No      :3473
## No internet service:1526      No internet service:1526
## Yes      :2422      Yes      :2044
##
##
##
##
##      StreamingTV      StreamingMovies      Contract
## No      :2810      No      :2785      Month-to-month:3875
## No internet service:1526      No internet service:1526      One year      :1473
## Yes      :2707      Yes      :2732      Two year      :1695
##
##
##
##
## PaperlessBilling      PaymentMethod      MonthlyCharges
## No :2872      Bank transfer (automatic):1544      Min.      : 18.25
## Yes:4171      Credit card (automatic) :1522      1st Qu.: 35.50
##      Electronic check      :2365      Median : 70.35
##      Mailed check      :1612      Mean   : 64.76
##      :1612      3rd Qu.: 89.85
##      :1612      Max.    :118.75
##
##
##      TotalCharges      Churn
## Min.      : 18.8      No :5174
## 1st Qu.: 401.4      Yes:1869
## Median :1397.5
## Mean   :2283.3
## 3rd Qu.:3794.7
## Max.    :8684.8
## NA's      :11
```

```
telco_chrun_clean <- telco_chrun %>%
  select(-customerID) %>%

  mutate(MultipleLines = case_when(
    MultipleLines %in% c("No phone service", "No") ~ "No",
    TRUE ~ "Yes"
  )) %>%

  mutate(InternetService = case_when(
    InternetService == "Fiber optic" ~ "FiberOptic",
    InternetService == "DSL" ~ "DSL",
    TRUE ~ "No"
  )) %>%

  mutate(across(c(OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMo
    ~ case_when(
      . %in% c("No internet service", "No") ~ "No",
```

```

      TRUE ~ "Yes"
    ))) %>%

mutate(PaymentMethod = case_when(
  PaymentMethod == "Bank transfer (automatic)" ~ "BankTransferAuto",
  PaymentMethod == "Credit card (automatic)" ~ "CreditCardAuto",
  PaymentMethod == "Electronic check" ~ "ECheck",
  TRUE ~ "MailedCheck"
)) %>%

mutate(across(where(is.character), as.factor))

summary(telco_chrun_clean)

##      gender      SeniorCitizen Partner      Dependents      tenure      PhoneService
## Female:3488    0:5901           No :3641    No :4933    Min.   : 0.00    No : 682
## Male   :3555    1:1142           Yes:3402    Yes:2110    1st Qu.: 9.00    Yes:6361
##                                           Median :29.00
##                                           Mean   :32.37
##                                           3rd Qu.:55.00
##                                           Max.   :72.00
##
## MultipleLines  InternetService OnlineSecurity OnlineBackup DeviceProtection
## No :4072        DSL             :2421    No :5024        No :4614        No :4621
## Yes:2971        FiberOptic:3096    Yes:2019        Yes:2429        Yes:2422
##                                           No             :1526
##
##
##
## TechSupport StreamingTV StreamingMovies      Contract      PaperlessBilling
## No :4999        No :4336        No :4311        Month-to-month:3875    No :2872
## Yes:2044        Yes:2707        Yes:2732        One year             :1473    Yes:4171
##                                           Two year            :1695
##
##
##
##      PaymentMethod MonthlyCharges      TotalCharges      Churn
## BankTransferAuto:1544    Min.   : 18.25    Min.   : 18.8    No :5174
## CreditCardAuto  :1522    1st Qu.: 35.50    1st Qu.: 401.4    Yes:1869
## ECheck           :2365    Median : 70.35    Median :1397.5
## MailedCheck      :1612    Mean   : 64.76    Mean   :2283.3
##                                           3rd Qu.: 89.85    3rd Qu.:3794.7
##                                           Max.   :118.75    Max.   :8684.8
##                                           NA's    :11

#Handling NA Values

telco_chrun_clean <- na.omit(telco_chrun_clean)

```

```
dim(telco_chrun_clean)
```

```
## [1] 7032 20
```

```
#Preliminary Relationship With Chrun Plot Function
```

```
For variables with yes/no responses
```

```
create_churn_plot <- function(variable_name, data = telco_chrun_clean) {  
  
  plot_data <- data %>%  
    count(Churn, !!sym(variable_name)) %>%  
    group_by(Churn) %>%  
    mutate(percentage = n / sum(n)) %>%  
    ungroup()  
  
  plot <- ggplot(plot_data, aes(x = factor(!!sym(variable_name)),  
                                y = percentage,  
                                fill = factor(!!sym(variable_name)))) +  
    geom_col(color = "black", show.legend = FALSE) +  
    geom_text(aes(label = scales::percent(percentage, accuracy = 0.1)),  
              vjust = -0.5, size = 4, fontface = "bold") +  
    facet_wrap(~ factor(Churn, levels = c("Yes", "No"),  
                        labels = c("Churned Customers", "Retained Customers")),  
              nrow = 1) +  
    scale_y_continuous(labels = scales::percent_format(),  
                       limits = c(0, 1),  
                       expand = expansion(mult = c(0, 0.1))) +  
    scale_fill_manual(values = c("Yes" = "lightblue", "No" = "lightcoral")) +  
    labs(x = gsub("_", " ", variable_name),  
         y = "Percentage of Group") +  
    theme_minimal() +  
    theme(strip.text = element_text(face = "bold", size = 11))  
  
  return(list(plot_data = plot_data, plot = plot))  
}
```

```
#EDA for demographic variables
```

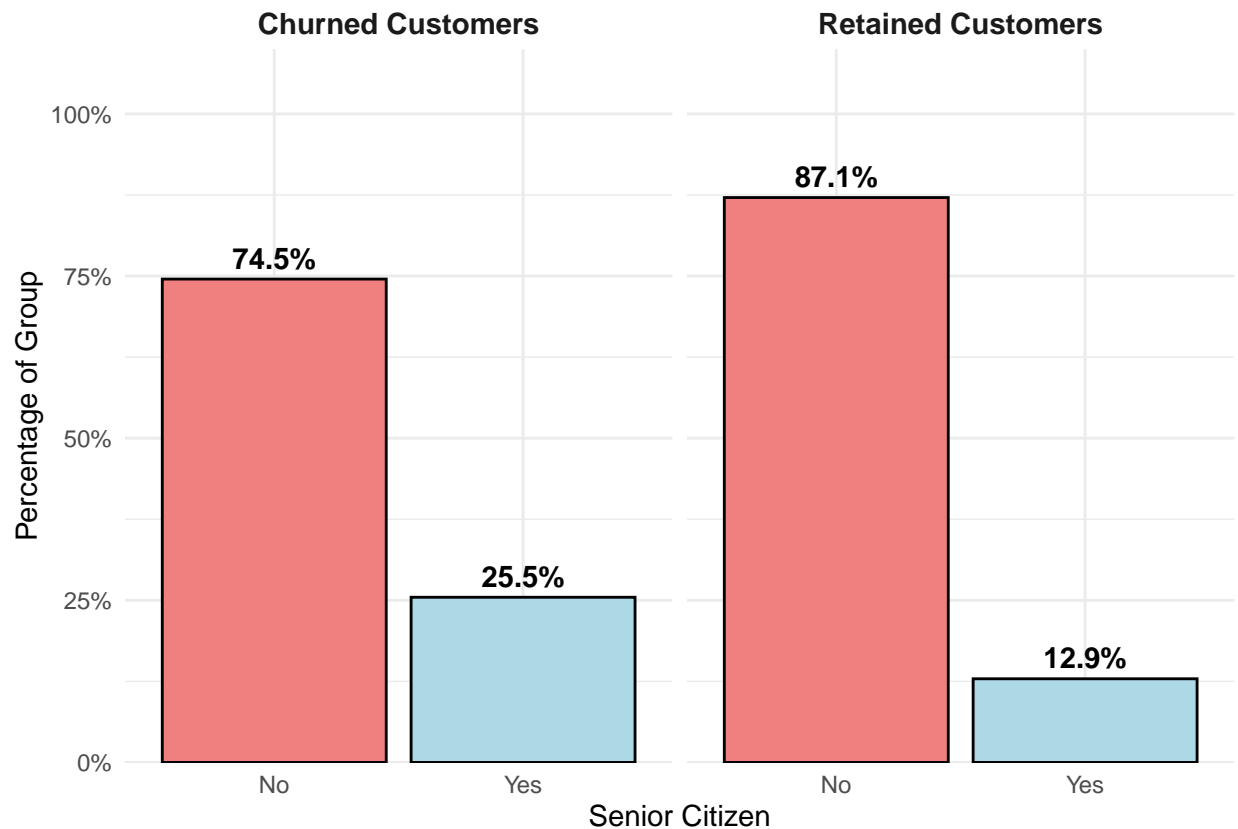
```
Senior_plot_data <- telco_chrun_clean %>%  
  count(Churn, SeniorCitizen) %>%  
  group_by(Churn) %>%  
  mutate(percentage = n / sum(n)) %>%  
  ungroup()  
  
Senior_plot <- ggplot(Senior_plot_data, aes(x = factor(SeniorCitizen, labels = c("No", "Yes")),  
                                             y = percentage,  
                                             fill = factor(SeniorCitizen, labels = c("No", "Yes")))) +  
  geom_col(color = "black", show.legend = FALSE) + # Bars with black outline, hide legend  
  geom_text(aes(label = scales::percent(percentage, accuracy = 0.1)),  
            vjust = -0.5, size = 4, fontface = "bold") +  
  facet_wrap(~ factor(Churn, levels = c("Yes", "No"),  
                        labels = c("Churned Customers", "Retained Customers")),
```

```

    nrow = 1) + # Put facets in one row, Churned first (left)
scale_y_continuous(labels = scales::percent_format(),
                  limits = c(0, 1), # Set y-axis from 0% to 100%
                  expand = expansion(mult = c(0, 0.1))) +
scale_fill_manual(values = c("Yes" = "lightblue", "No" = "lightcoral")) + # Light colors
labs(
  x = "Senior Citizen",
  y = "Percentage of Group") +
theme_minimal() +
theme(strip.text = element_text(face = "bold", size = 11))

```

Senior_plot



```

Gender_plot_data <- telco_chrun_clean %>%
  count(Churn, gender) %>%
  group_by(Churn) %>%
  mutate(percentage = n / sum(n)) %>%
  ungroup()
head(telco_chrun_clean)

```

```

##   gender SeniorCitizen Partner Dependents tenure PhoneService MultipleLines
## 1 Female           0     Yes         No        1           No           No
## 2 Male            0     No         No       34           Yes           No
## 3 Male            0     No         No        2           Yes           No
## 4 Male            0     No         No       45           No           No

```

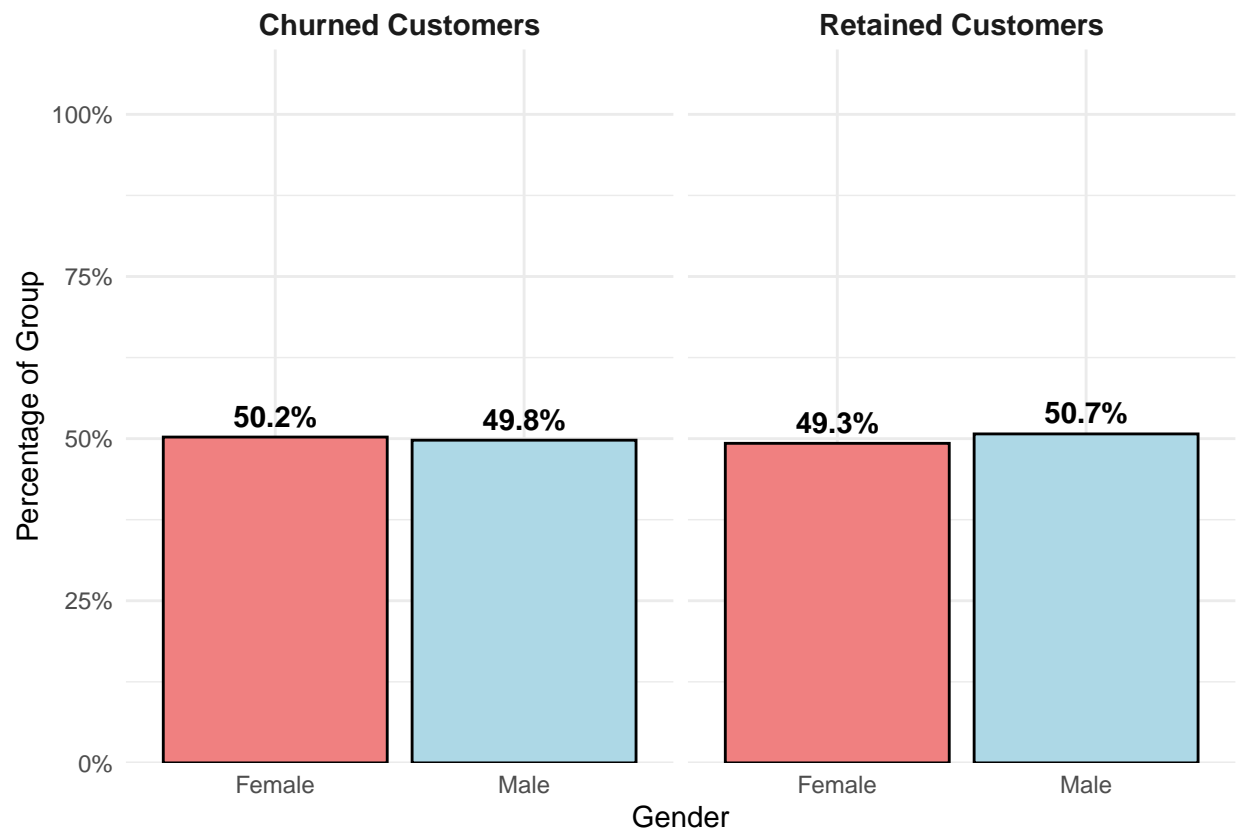

## 5	Female	0	No	No	2	Yes	No
## 6	Female	0	No	No	8	Yes	Yes
##	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport		
## 1	DSL	No	Yes	No	No	No	No
## 2	DSL	Yes	No	Yes	No	No	No
## 3	DSL	Yes	Yes	No	No	No	No
## 4	DSL	Yes	No	Yes	Yes	Yes	Yes
## 5	FiberOptic	No	No	No	No	No	No
## 6	FiberOptic	No	No	Yes	No	No	No
##	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod		
## 1	No	No	Month-to-month	Yes	ECheck		
## 2	No	No	One year	No	MailedCheck		
## 3	No	No	Month-to-month	Yes	MailedCheck		
## 4	No	No	One year	No	BankTransferAuto		
## 5	No	No	Month-to-month	Yes	ECheck		
## 6	Yes	Yes	Month-to-month	Yes	ECheck		
##	MonthlyCharges	TotalCharges	Churn				
## 1	29.85	29.85	No				
## 2	56.95	1889.50	No				
## 3	53.85	108.15	Yes				
## 4	42.30	1840.75	No				
## 5	70.70	151.65	Yes				
## 6	99.65	820.50	Yes				

```

Gender_plot <- ggplot(Gender_plot_data, aes(x = factor(gender),
      y = percentage,
      fill = factor(gender))) +
  geom_col(color = "black", show.legend = FALSE) +
  geom_text(aes(label = scales::percent(percentage, accuracy = 0.1)),
    vjust = -0.5, size = 4, fontface = "bold") +
  facet_wrap(~ factor(Churn, levels = c("Yes", "No"),
    labels = c("Churned Customers", "Retained Customers")),
    nrow = 1) +
  scale_y_continuous(labels = scales::percent_format(),
    limits = c(0, 1),
    expand = expansion(mult = c(0, 0.1))) +
  scale_fill_manual(values = c("Male" = "lightblue", "Female" = "lightcoral")) +
  labs(
    x = "Gender",
    y = "Percentage of Group") +
  theme_minimal() +
  theme(strip.text = element_text(face = "bold", size = 11))

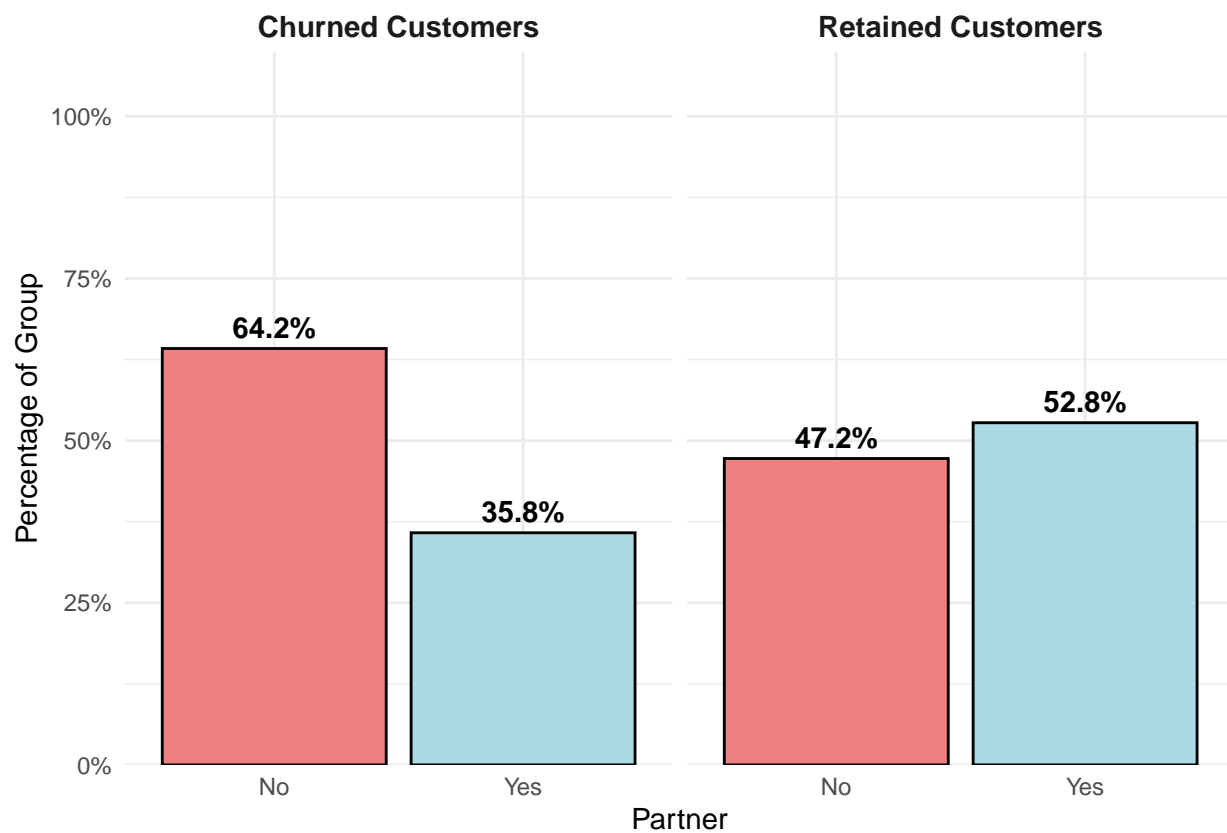
Gender_plot

```



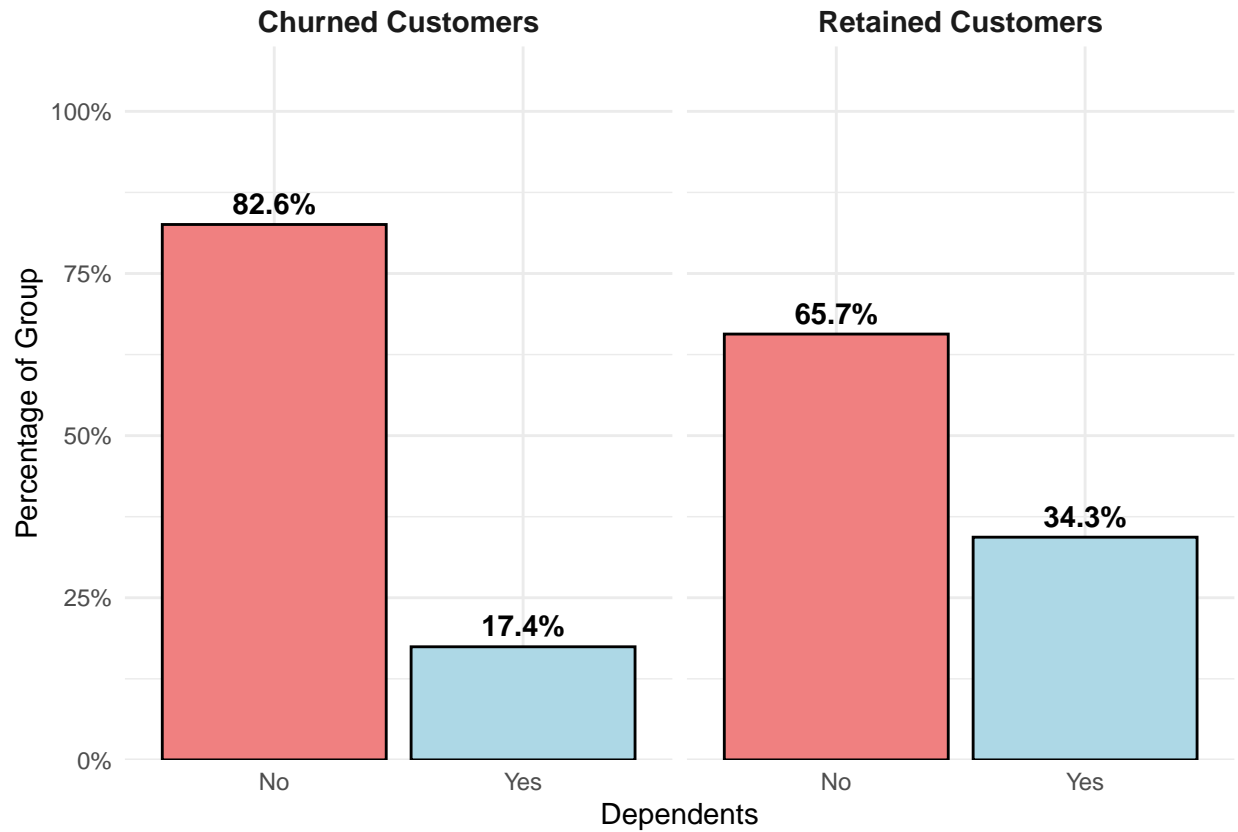
```
Partner_results <- create_churn_plot("Partner")
Partner_plot_data <- Partner_results$plot_data
Partner_plot <- Partner_results$plot

Partner_plot
```



```
Dependents_results <- create_churn_plot("Dependents")
Dependents_plot_data <- Dependents_results$plot_data
Dependents_plot <- Dependents_results$plot
```

```
Dependents_plot
```



##Plot for Report

```
combined_analysis_demographics <- (Senior_plot + Gender_plot) /
  (Partner_plot + Dependents_plot)

combined_analysis_demographics + plot_annotation(
  title = "Customer Demographic Analysis by Churn Status",
  theme = theme(plot.title = element_text(hjust = 0.5, face = "bold"))
)
```

Customer Demographic Analysis by Churn Status



#EDA for Account Information

```
avg_tenure <- telco_chrun_clean %>%
  group_by(Churn) %>%
  summarise(avg_tenure = mean(tenure))

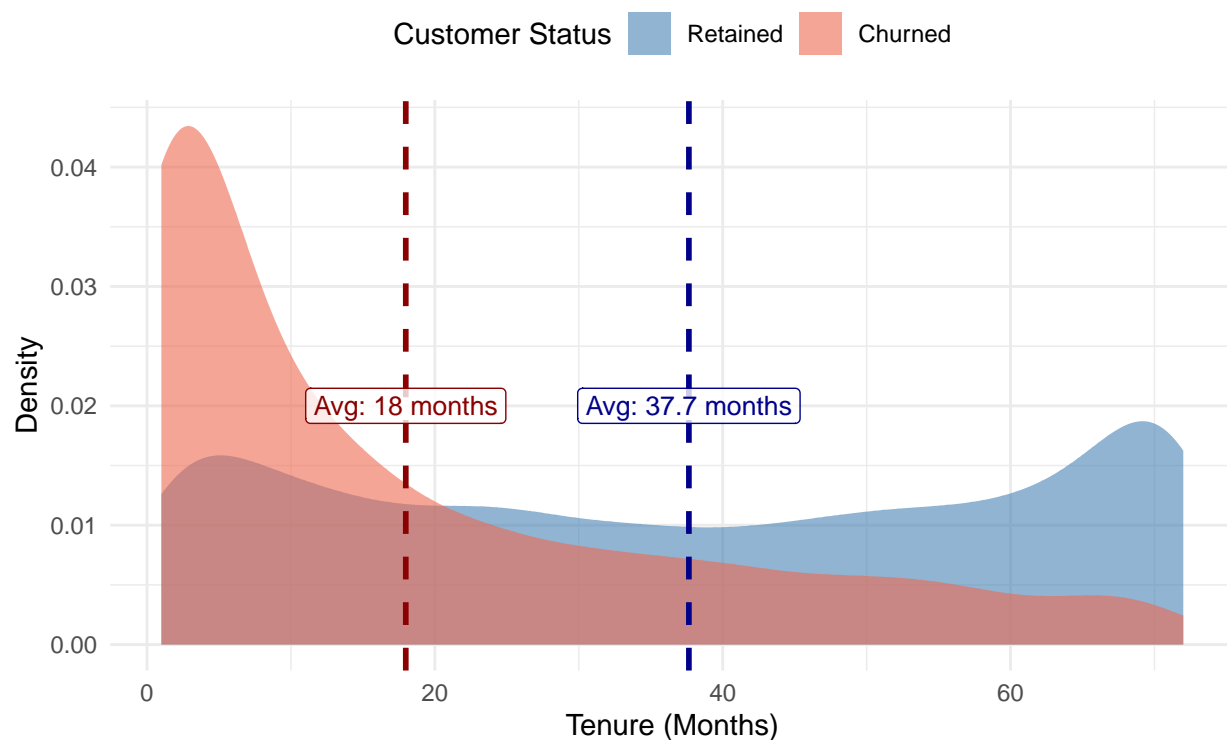
tenure_density_plot <- ggplot(telco_chrun_clean, aes(x = tenure, fill = Churn)) +
  geom_density(alpha = 0.6, color = NA) +
  geom_vline(data = avg_tenure,
            aes(xintercept = avg_tenure, color = Churn),
            linetype = "dashed", size = 1, show.legend = FALSE) +
  geom_label(data = avg_tenure,
            aes(x = avg_tenure, y = 0.02,
              label = paste("Avg:", round(avg_tenure, 1), "months"),
              color = Churn),
            fill = "white", alpha = 0.8, size = 3.5,
            show.legend = FALSE) +
  scale_fill_manual(values = c("No" = "steelblue", "Yes" = "coral2"),
                  labels = c("No" = "Retained", "Yes" = "Churned")) +
  scale_color_manual(values = c("No" = "darkblue", "Yes" = "darkred")) +
  labs(title = "Distribution of Customer Tenure by Churn Status",
       subtitle = "Dashed lines show average tenure for each group",
       x = "Tenure (Months)",
       y = "Density",
       fill = "Customer Status") +
  theme_minimal() +
  theme(legend.position = "top")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
print(tenure_density_plot)
```

Distribution of Customer Tenure by Churn Status

Dashed lines show average tenure for each group



```
print(avg_tenure)
```

```
## # A tibble: 2 x 2
##   Churn avg_tenure
##   <fct>     <dbl>
## 1 No       37.7
## 2 Yes     18.0
```

```
avg_MonthlyCharges <- telco_chrun_clean %>%
  group_by(Churn) %>%
  summarise(avg_MonthlyCharges = mean(MonthlyCharges))
```

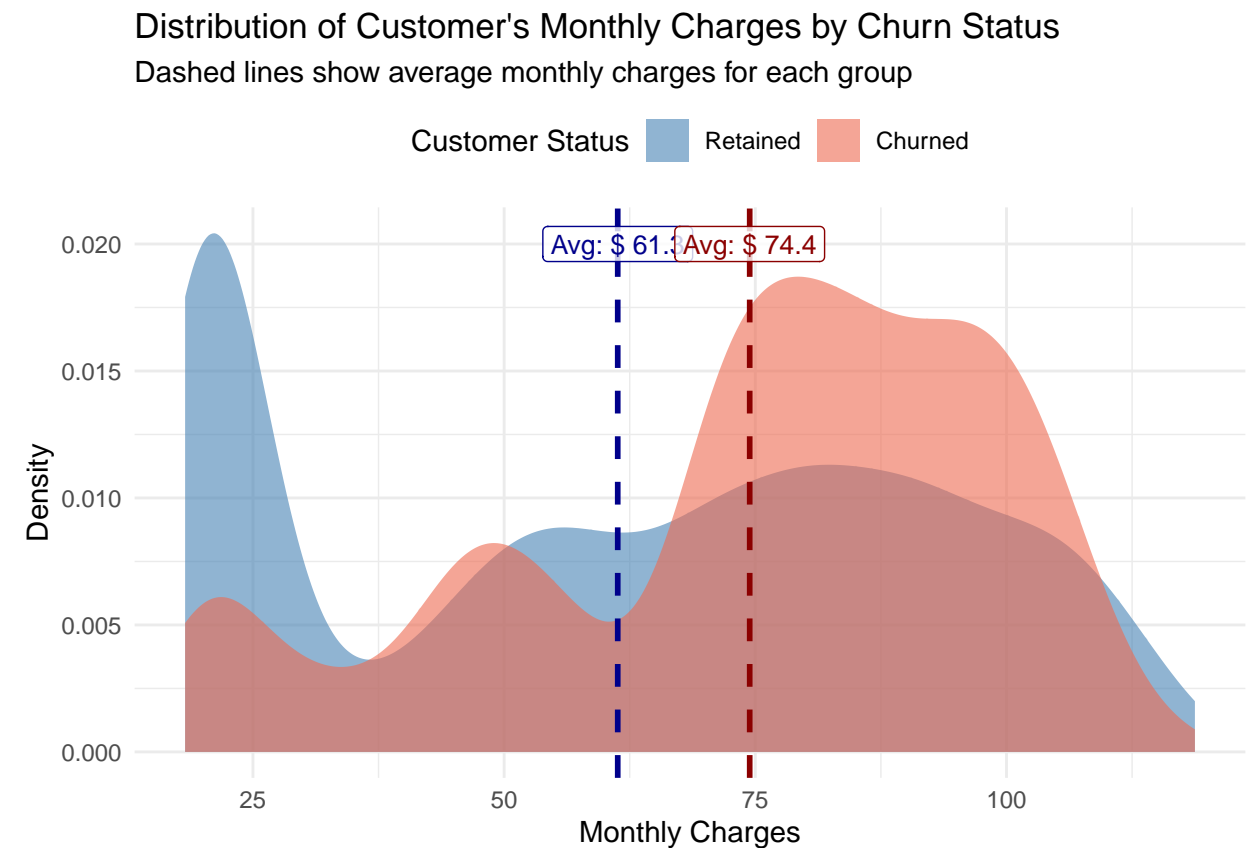
```
MonthlyCharges_density_plot <- ggplot(telco_chrun_clean, aes(x = MonthlyCharges, fill = Churn)) +
  geom_density(alpha = 0.6, color = NA) +
```

```

geom_vline(data = avg_MonthlyCharges,
  aes(xintercept = avg_MonthlyCharges, color = Churn),
  linetype = "dashed", size = 1, show.legend = FALSE) +
geom_label(data = avg_MonthlyCharges,
  aes(x = avg_MonthlyCharges, y = 0.02,
    label = paste("Avg: $", round(avg_MonthlyCharges, 1)),
    color = Churn),
  fill = "white", alpha = 0.8, size = 3.5,
  show.legend = FALSE) +
scale_fill_manual(values = c("No" = "steelblue", "Yes" = "coral2"),
  labels = c("No" = "Retained", "Yes" = "Churned")) +
scale_color_manual(values = c("No" = "darkblue", "Yes" = "darkred")) +
labs(title = "Distribution of Customer's Monthly Charges by Churn Status",
  subtitle = "Dashed lines show average monthly charges for each group",
  x = "Monthly Charges",
  y = "Density",
  fill = "Customer Status") +
theme_minimal() +
theme(legend.position = "top")

print(MonthlyCharges_density_plot)

```



```
print(avg_MonthlyCharges)
```

```
## # A tibble: 2 x 2
```

```
## Churn avg_MonthlyCharges
## <fct>          <dbl>
## 1 No          61.3
## 2 Yes         74.4
```

```
avg_TotalCharges <- telco_chrun_clean %>%
  group_by(Churn) %>%
  summarise(avg_TotalCharges = mean(TotalCharges))

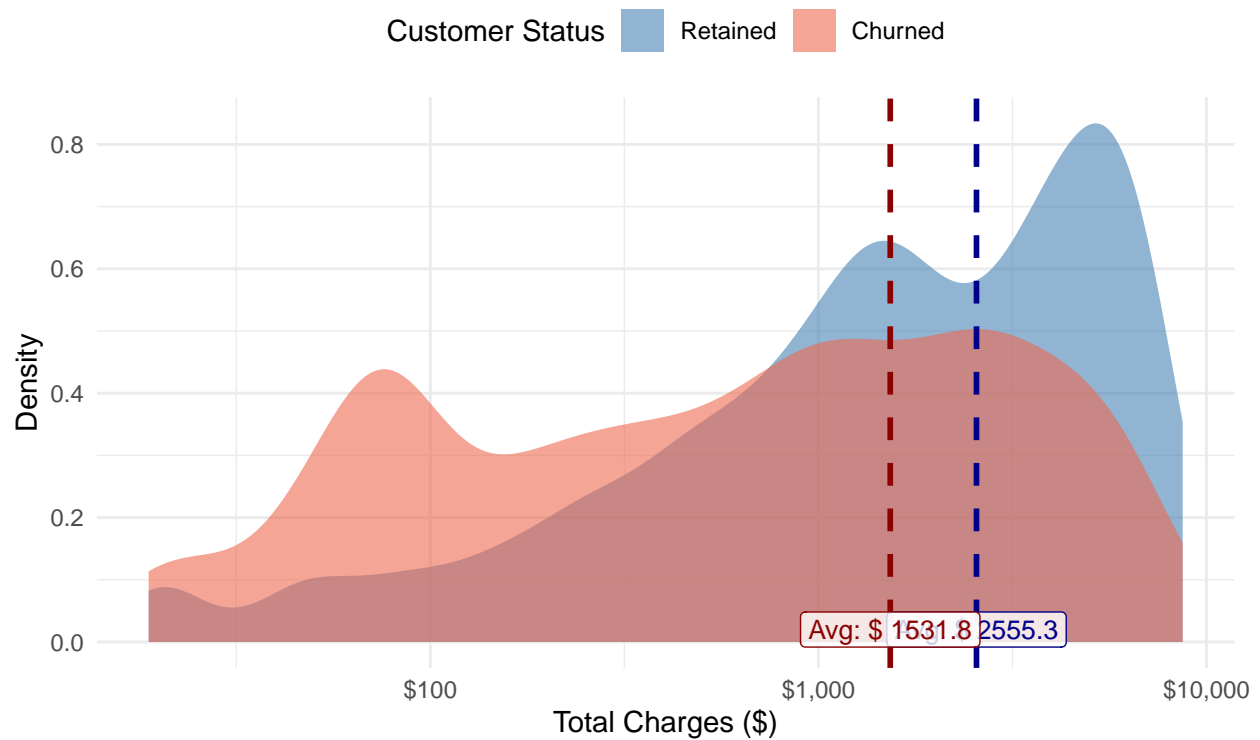
TotalCharges_density_plot <- ggplot(telco_chrun_clean, aes(x = TotalCharges, fill = Churn)) +
  geom_density(alpha = 0.6, color = NA) +
  geom_vline(data = avg_TotalCharges,
             aes(xintercept = avg_TotalCharges, color = Churn),
             linetype = "dashed", size = 1, show.legend = FALSE) +
  geom_label(data = avg_TotalCharges,
             aes(x = avg_TotalCharges, y = 0.02,
                 label = paste("Avg: $", round(avg_TotalCharges, 1)),
                 color = Churn),
             fill = "white", alpha = 0.8, size = 3.5,
             show.legend = FALSE) +
  scale_fill_manual(values = c("No" = "steelblue", "Yes" = "coral2"),
                   labels = c("No" = "Retained", "Yes" = "Churned")) +
  scale_color_manual(values = c("No" = "darkblue", "Yes" = "darkred")) +
  labs(title = "Distribution of Customer's Total Charges by Churn Status (Logarithmic Scale)",
       subtitle = "Dashed lines show average total charges for each group",
       x = "Total Charges ($)",
       y = "Density",
       fill = "Customer Status") +
  theme_minimal() +
  theme(legend.position = "top")

TotalCharges_density_plot <- TotalCharges_density_plot +
  scale_x_log10(labels = scales::dollar)

TotalCharges_density_plot
```


Distribution of Customer's Total Charges by Churn Status (Logarithmic Scale)

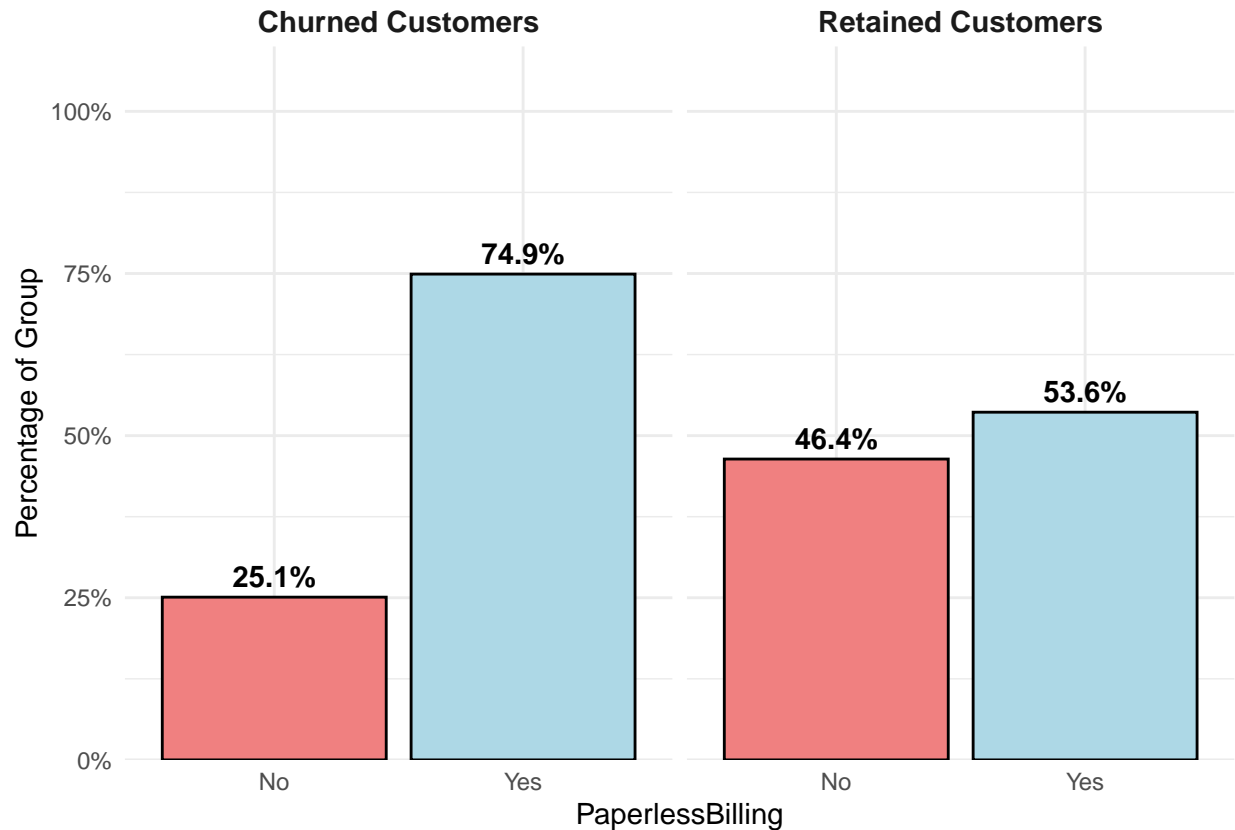
Dashed lines show average total charges for each group



Extreme Right-Skew:

```
PaperlessBilling_results <- create_churn_plot("PaperlessBilling")
PaperlessBilling_plot_data <- PaperlessBilling_results$plot_data
PaperlessBilling_plot <- PaperlessBilling_results$plot

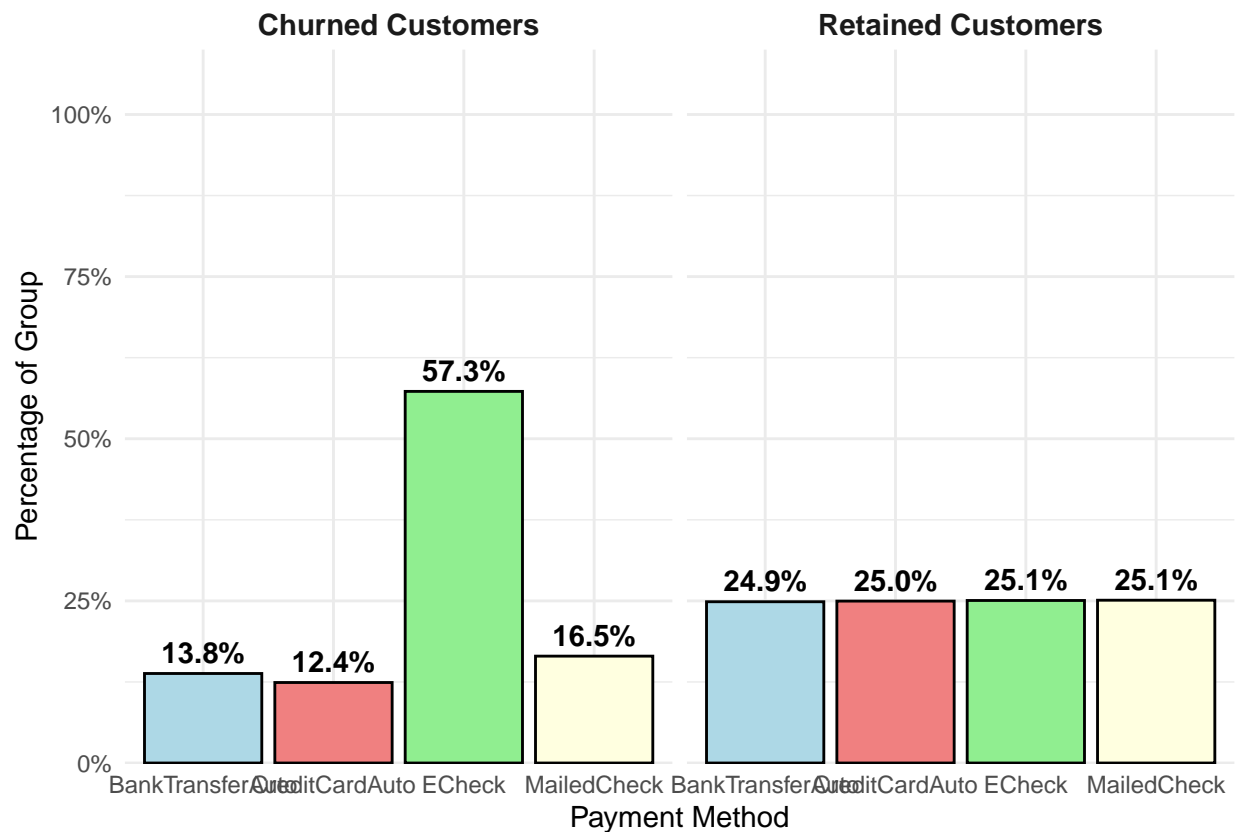
PaperlessBilling_plot
```



```
PaymentMethod_plot_data <- telco_chrun_clean %>%
  count(Churn, PaymentMethod) %>%
  group_by(Churn) %>%
  mutate(percentage = n / sum(n)) %>%
  ungroup()

PaymentMethod_plot <- ggplot(PaymentMethod_plot_data, aes(x = factor(PaymentMethod),
  y = percentage,
  fill = factor(PaymentMethod))) +
  geom_col(color = "black", show.legend = FALSE) +
  geom_text(aes(label = scales::percent(percentage, accuracy = 0.1)),
    vjust = -0.5, size = 4, fontface = "bold") +
  facet_wrap(~ factor(Churn, levels = c("Yes", "No"),
    labels = c("Churned Customers", "Retained Customers")),
    nrow = 1) +
  scale_y_continuous(labels = scales::percent_format(),
    limits = c(0, 1),
    expand = expansion(mult = c(0, 0.1))) +
  scale_fill_manual(values = c("BankTransferAuto" = "lightblue", "CreditCardAuto" = "lightcoral", "ECheckAuto" = "lightgreen"),
    labs(
      x = "Payment Method",
      y = "Percentage of Group") +
  theme_minimal() +
  theme(strip.text = element_text(face = "bold", size = 11))

PaymentMethod_plot
```



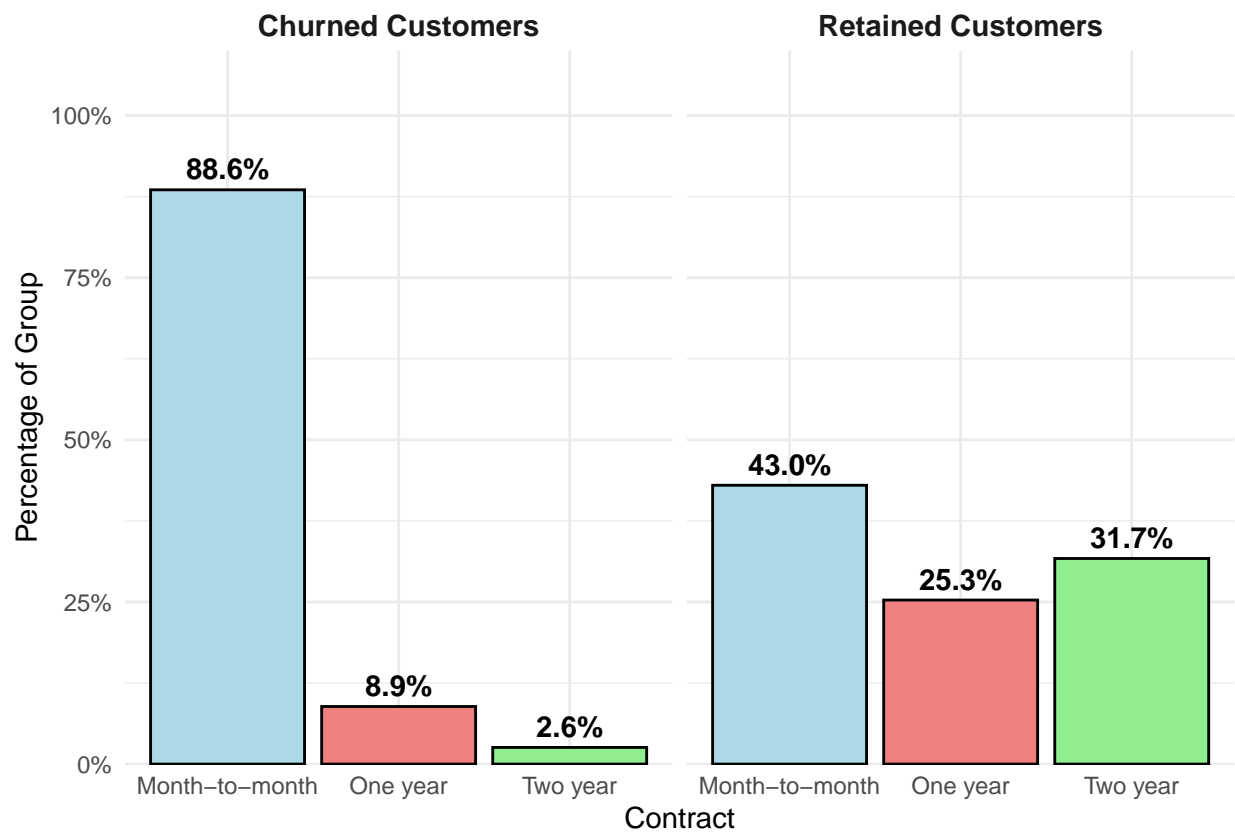
```

Contract_plot_data <- telco_chrun_clean %>%
  count(Churn, Contract) %>%
  group_by(Churn) %>%
  mutate(percentage = n / sum(n)) %>%
  ungroup()

Contract_plot <- ggplot(Contract_plot_data, aes(x = factor(Contract),
  y = percentage,
  fill = factor(Contract))) +
  geom_col(color = "black", show.legend = FALSE) +
  geom_text(aes(label = scales::percent(percentage, accuracy = 0.1)),
    vjust = -0.5, size = 4, fontface = "bold") +
  facet_wrap(~ factor(Churn, levels = c("Yes", "No"),
    labels = c("Churned Customers", "Retained Customers")),
    nrow = 1) +
  scale_y_continuous(labels = scales::percent_format(),
    limits = c(0, 1),
    expand = expansion(mult = c(0, 0.1))) +
  scale_fill_manual(values = c("Month-to-month" = "lightblue", "One year" = "lightcoral", "Two year" = "lightyellow"),
    labs(
      x = "Contract",
      y = "Percentage of Group") +
  theme_minimal() +
  theme(strip.text = element_text(face = "bold", size = 11))

```

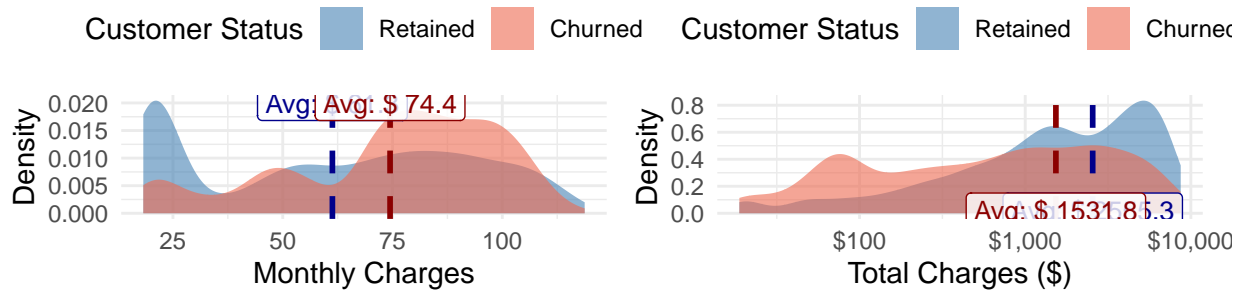
Contract_plot



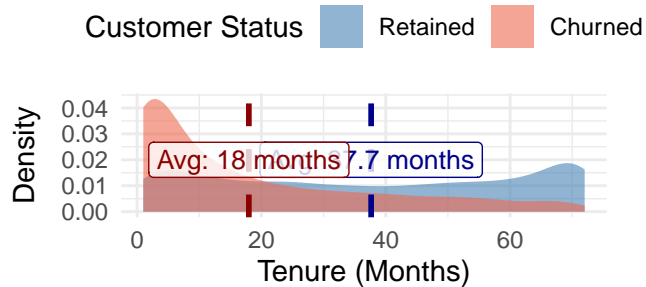
##Plots for Report

```
combined_analysis_account <- (MonthlyCharges_density_plot | TotalCharges_density_plot) /  
  (tenure_density_plot)  
  
combined_analysis_account <- wrap_plots(  
  MonthlyCharges_density_plot,  
  TotalCharges_density_plot,  
  tenure_density_plot,  
  ncol = 2,  
  nrow = 2  
)  
  
combined_analysis_account
```

Distribution of Customer's Monthly Charges by Churn Status
Dashed lines show average monthly charges for each group



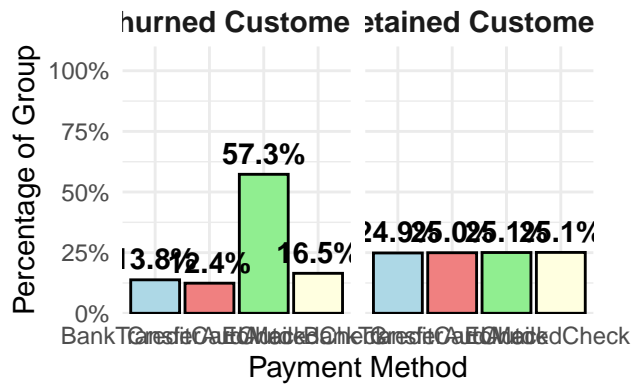
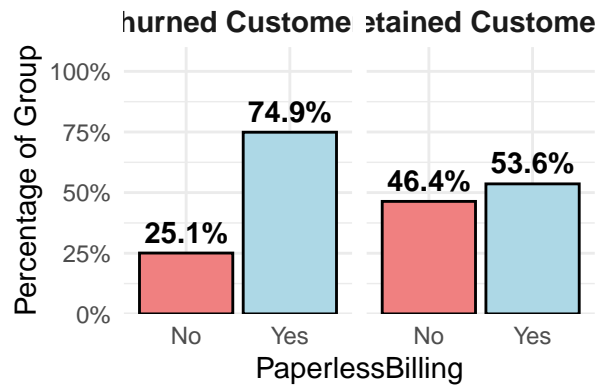
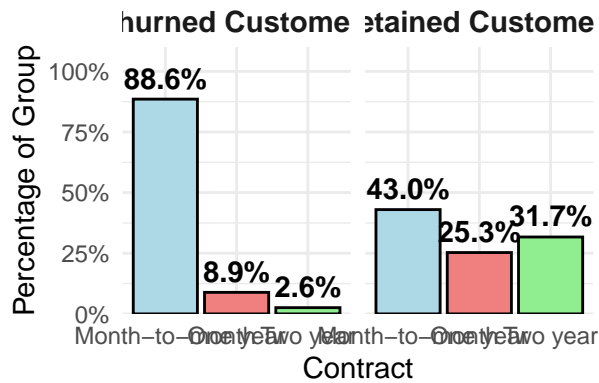
Distribution of Customer Tenure by Churn Status
Dashed lines show average tenure for each group



```
combined_analysis_account2 <- (Contract_plot | PaperlessBilling_plot) /
  (PaymentMethod_plot)

combined_analysis_account2 <- wrap_plots(
  Contract_plot,
  PaperlessBilling_plot,
  PaymentMethod_plot,
  ncol = 2,
  nrow = 2
)

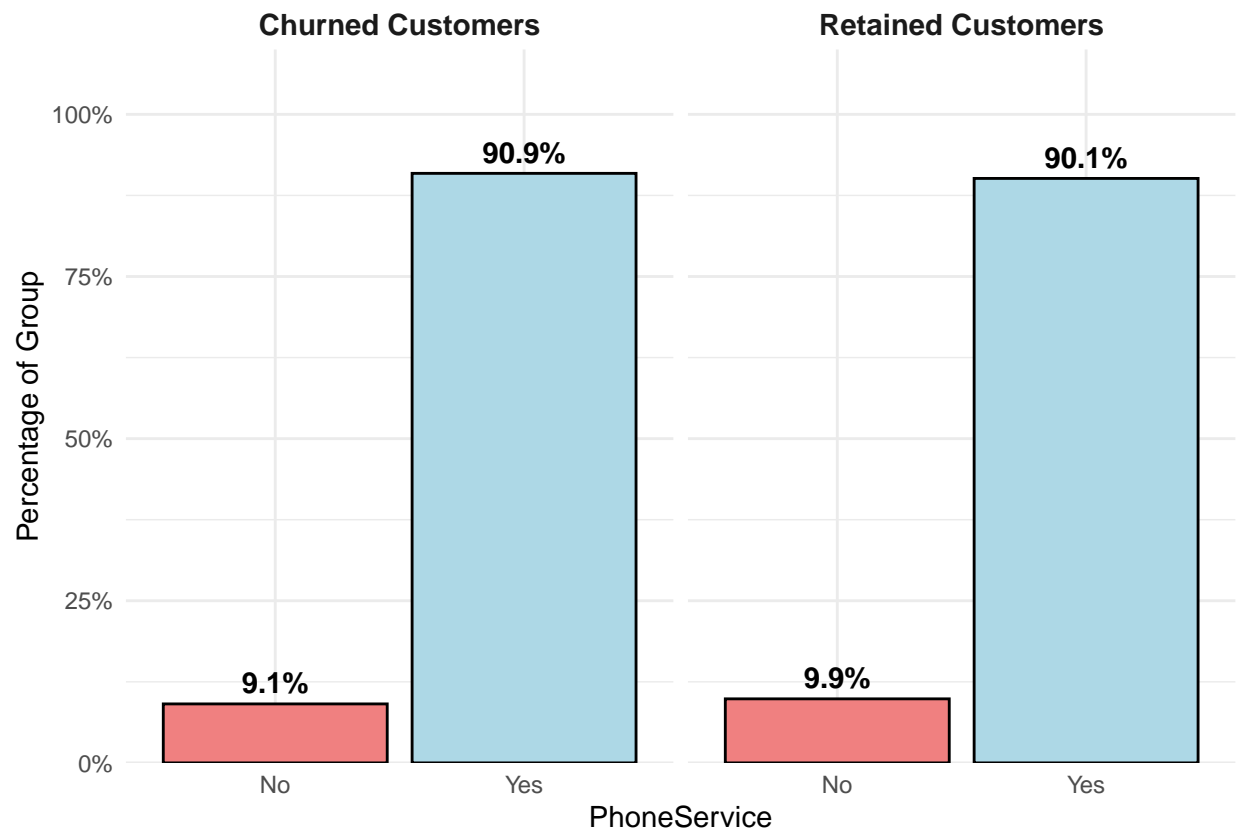
combined_analysis_account2
```



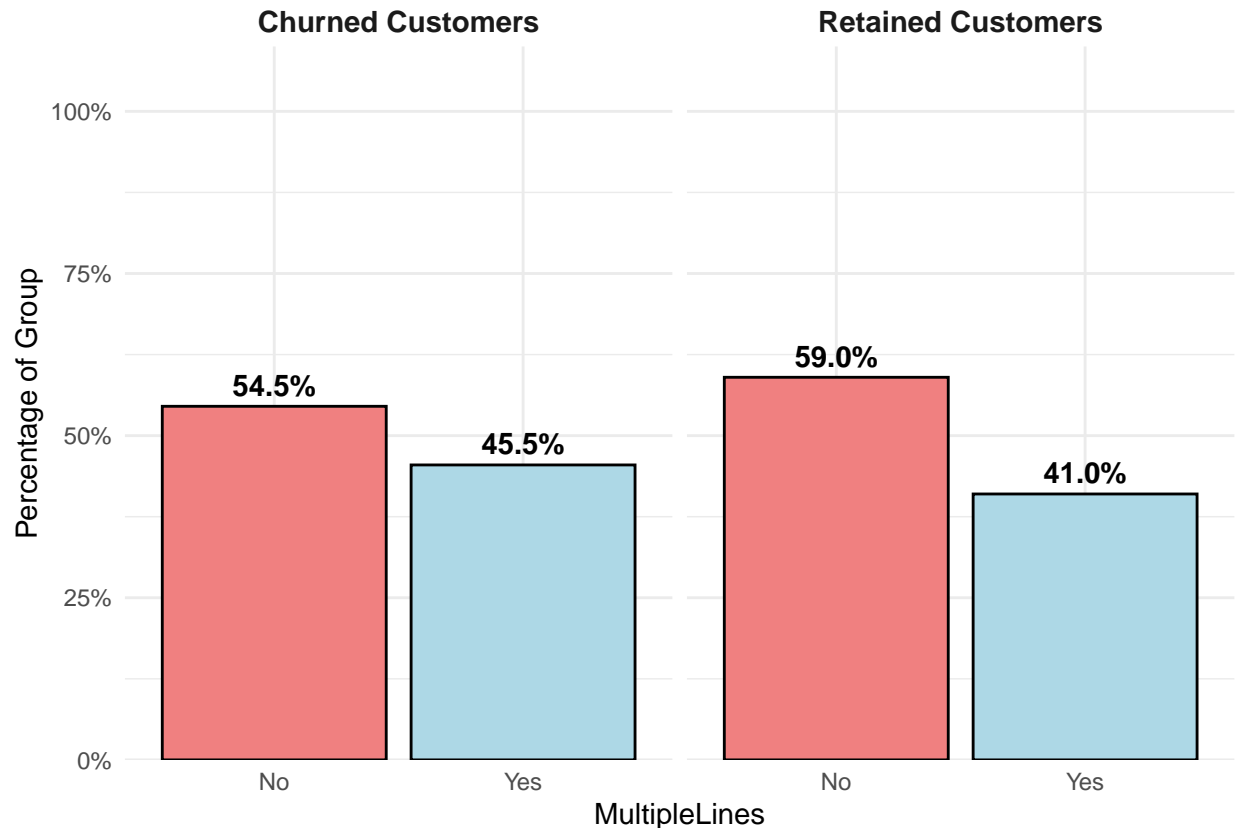
#EDA for Service Information

```
PhoneService_results <- create_churn_plot("PhoneService")
PhoneService_plot_data <- PhoneService_results$plot_data
PhoneService_plot <- PhoneService_results$plot

PhoneService_plot
```



```
MultipleLines_results <- create_churn_plot("MultipleLines")
MultipleLines_plot_data <- MultipleLines_results$plot_data
MultipleLines_plot <- MultipleLines_results$plot
MultipleLines_plot
```



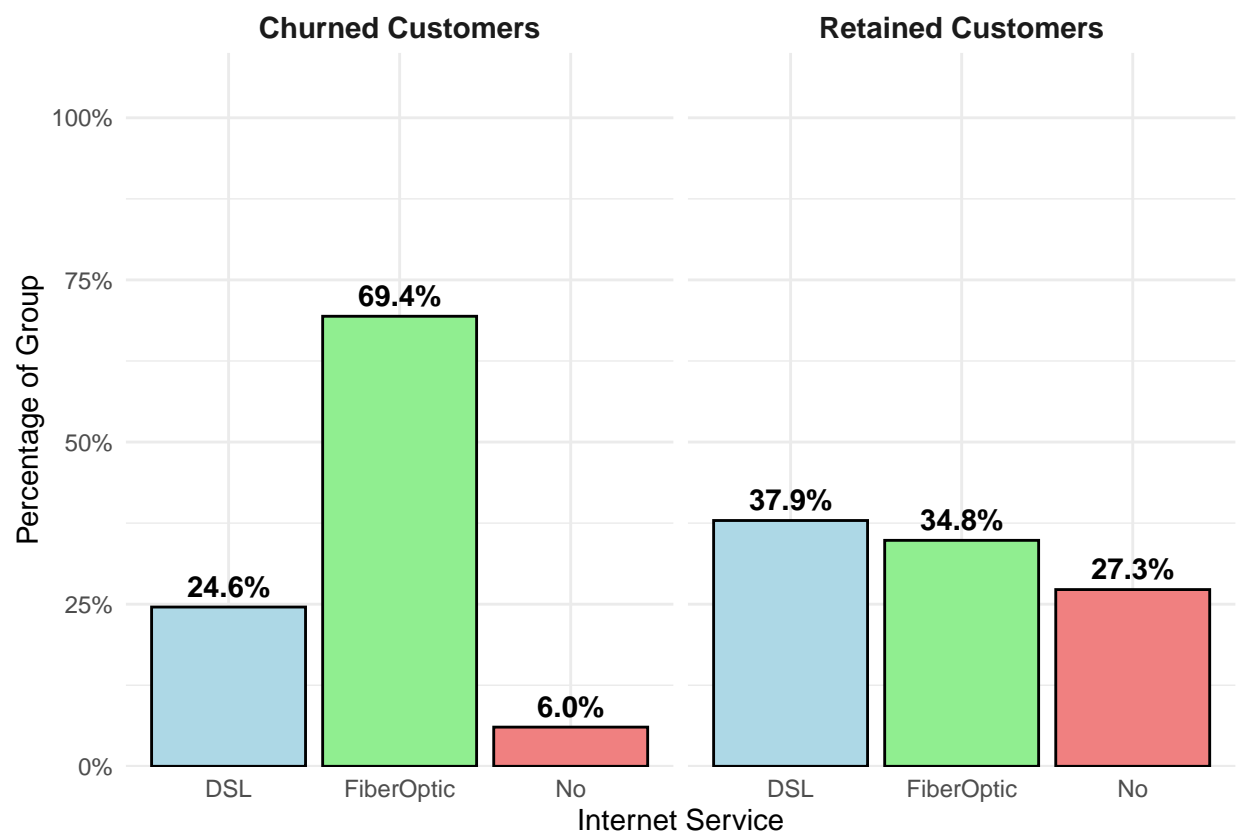
```

InternetService_plot_data <- telco_chrun_clean %>%
  count(Churn, InternetService) %>%
  group_by(Churn) %>%
  mutate(percentage = n / sum(n)) %>%
  ungroup()

InternetService_plot <- ggplot(InternetService_plot_data, aes(x = factor(InternetService),
  y = percentage,
  fill = factor(InternetService))) +
  geom_col(color = "black", show.legend = FALSE) +
  geom_text(aes(label = scales::percent(percent(percent, accuracy = 0.1)),
    vjust = -0.5, size = 4, fontface = "bold") +
  facet_wrap(~ factor(Churn, levels = c("Yes", "No"),
    labels = c("Churned Customers", "Retained Customers")),
    nrow = 1) +
  scale_y_continuous(labels = scales::percent_format(),
    limits = c(0, 1),
    expand = expansion(mult = c(0, 0.1))) +
  scale_fill_manual(values = c("DSL" = "lightblue", "No" = "lightcoral", "FiberOptic" = "lightgreen")) +
  labs(
    x = "Internet Service",
    y = "Percentage of Group") +
  theme_minimal() +
  theme(strip.text = element_text(face = "bold", size = 11))

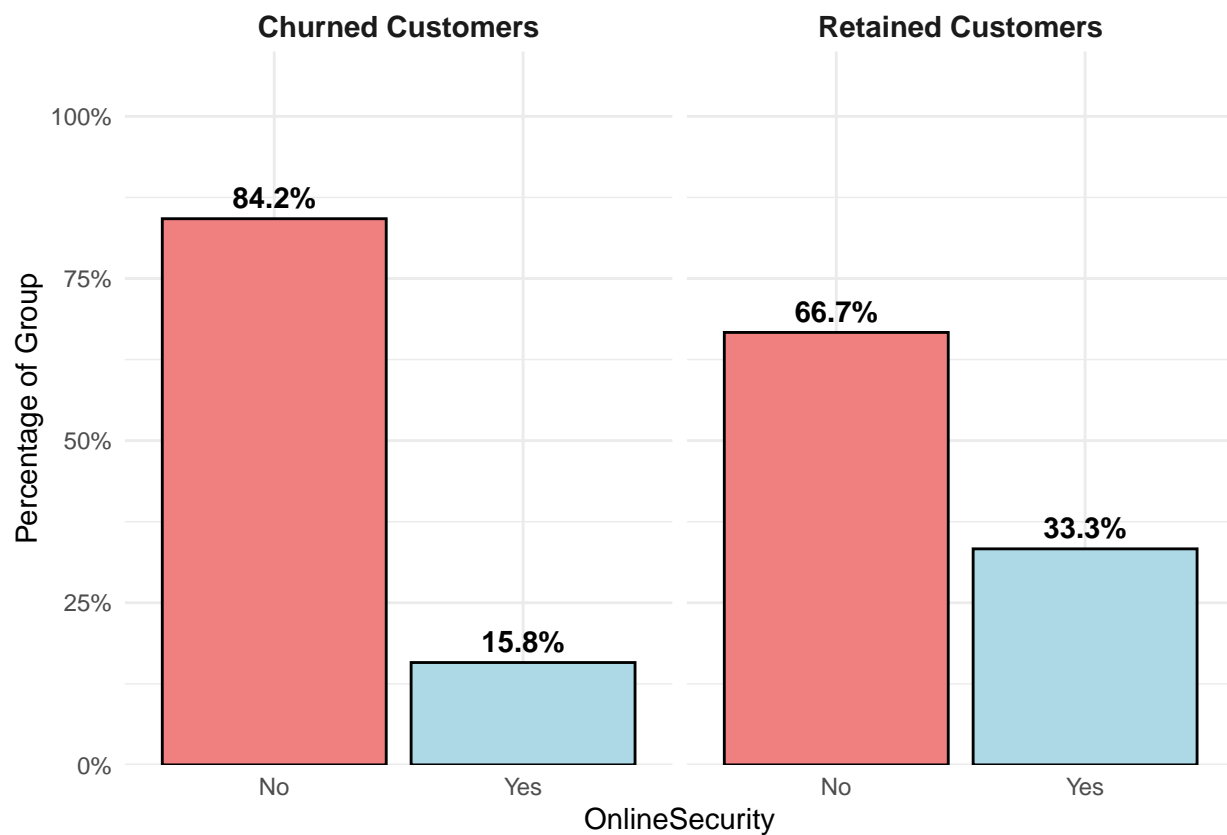
InternetService_plot

```

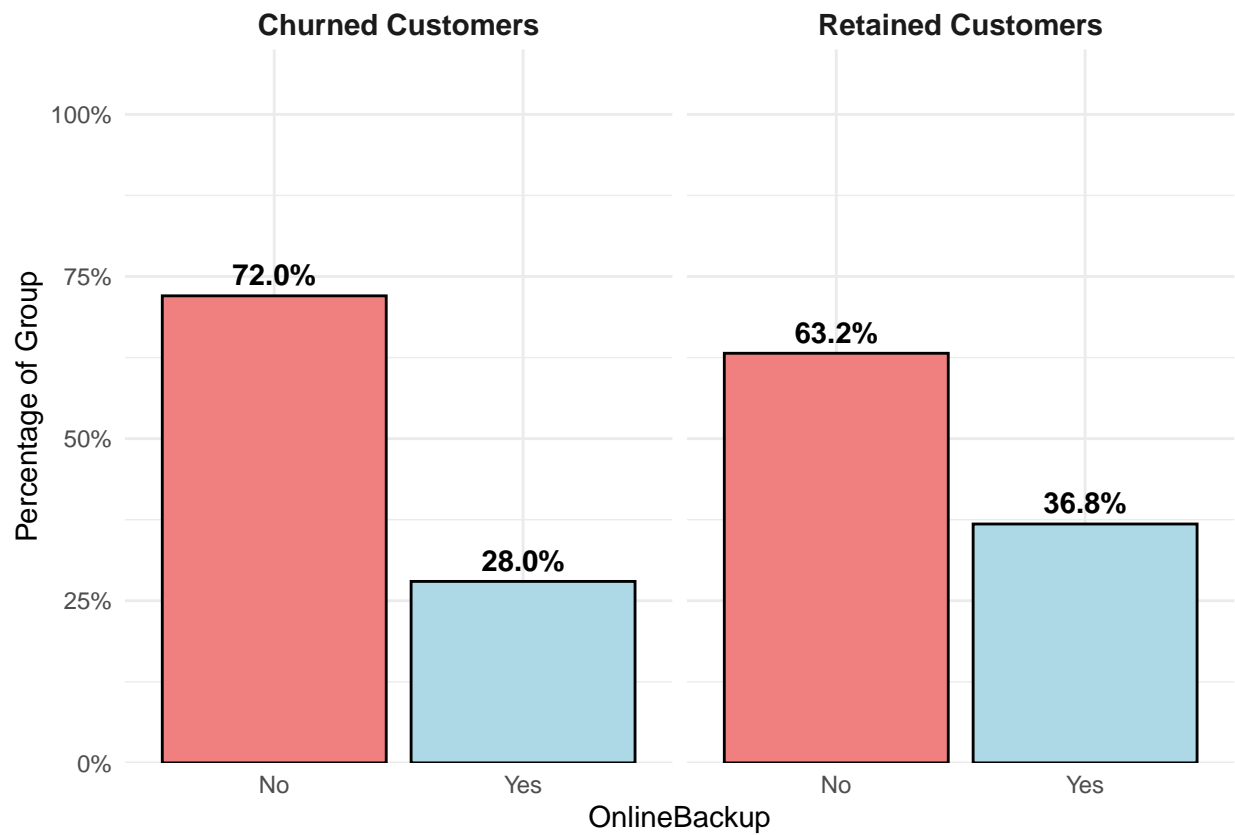
```
OnlineSecurity_results <- create_churn_plot("OnlineSecurity")
OnlineSecurity_plot_data <- OnlineSecurity_results$plot_data
OnlineSecurity_plot <- OnlineSecurity_results$plot

OnlineSecurity_plot
```



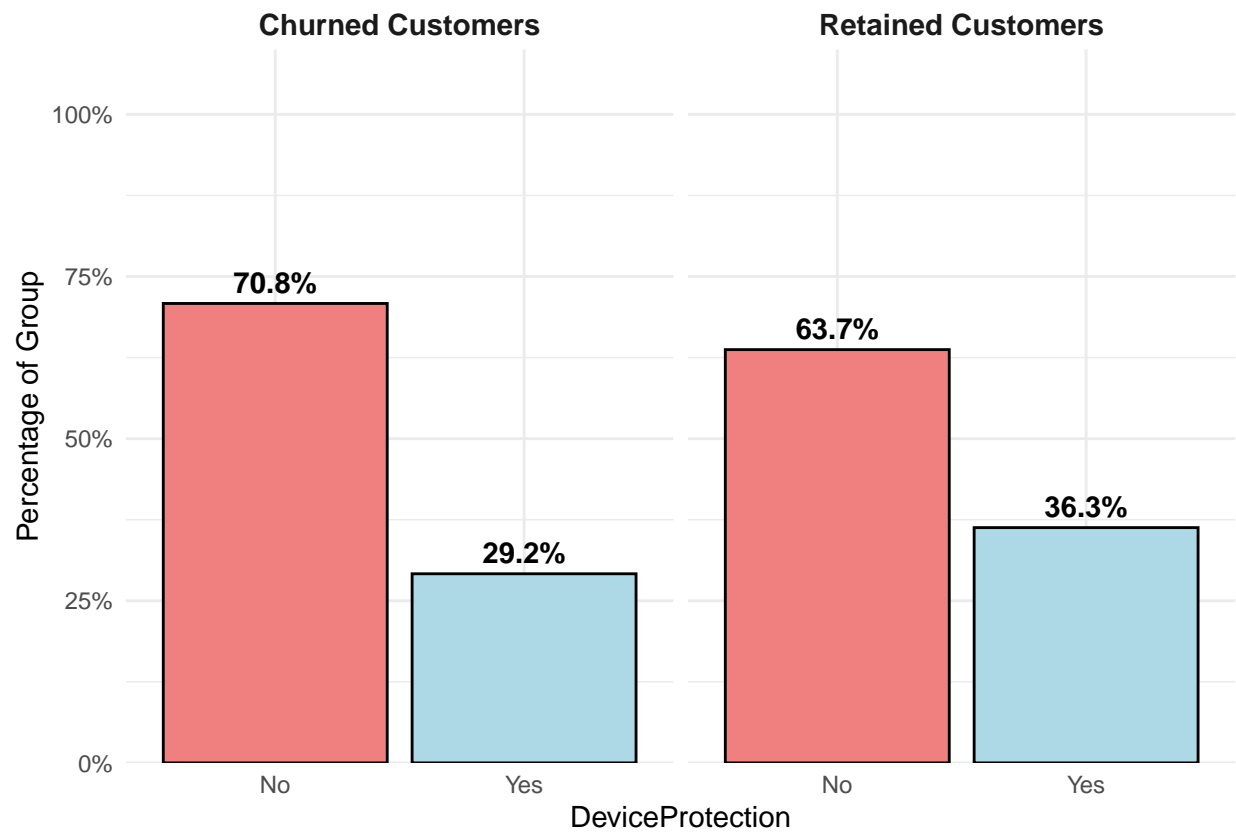
```
OnlineBackup_results <- create_churn_plot("OnlineBackup")
OnlineBackup_plot_data <- OnlineBackup_results$plot_data
OnlineBackup_plot <- OnlineBackup_results$plot

OnlineBackup_plot
```

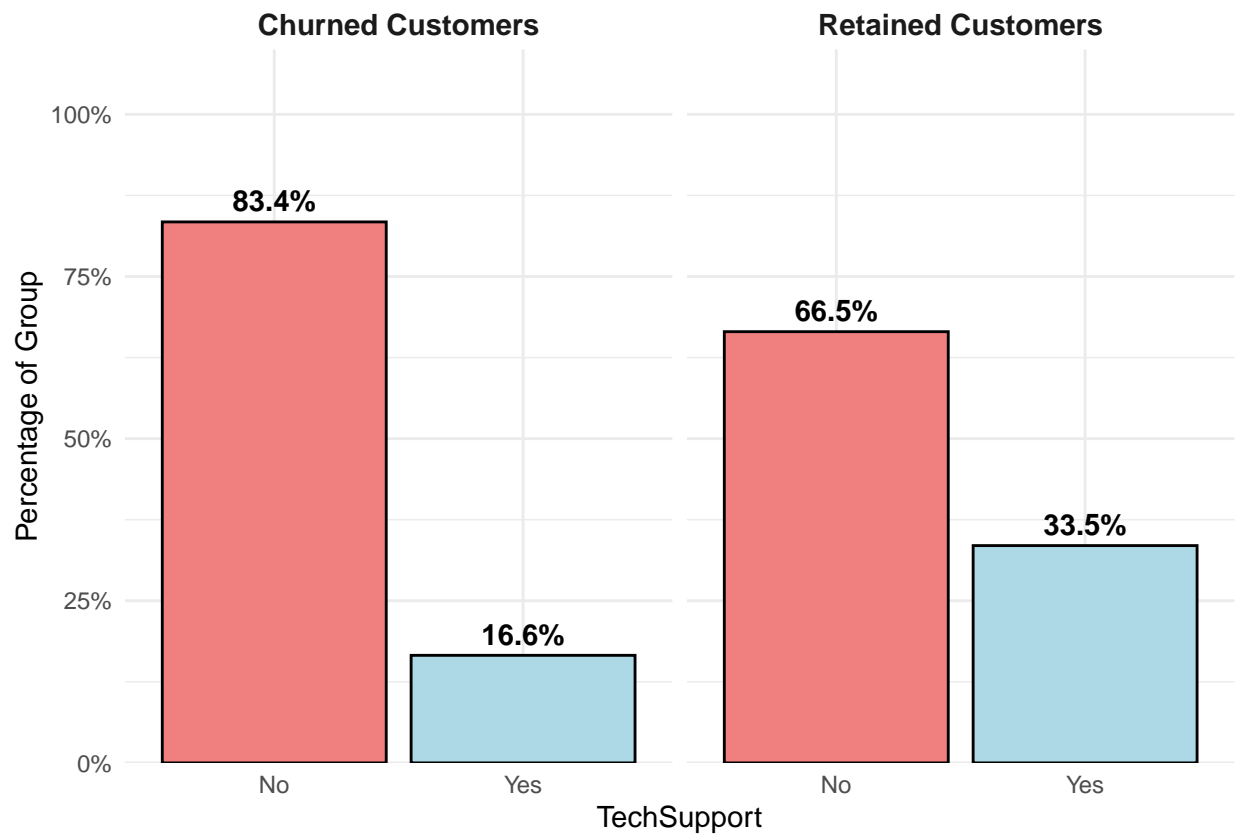


```
DeviceProtection_results <- create_churn_plot("DeviceProtection")
DeviceProtection_plot_data <- DeviceProtection_results$plot_data
DeviceProtection_plot <- DeviceProtection_results$plot

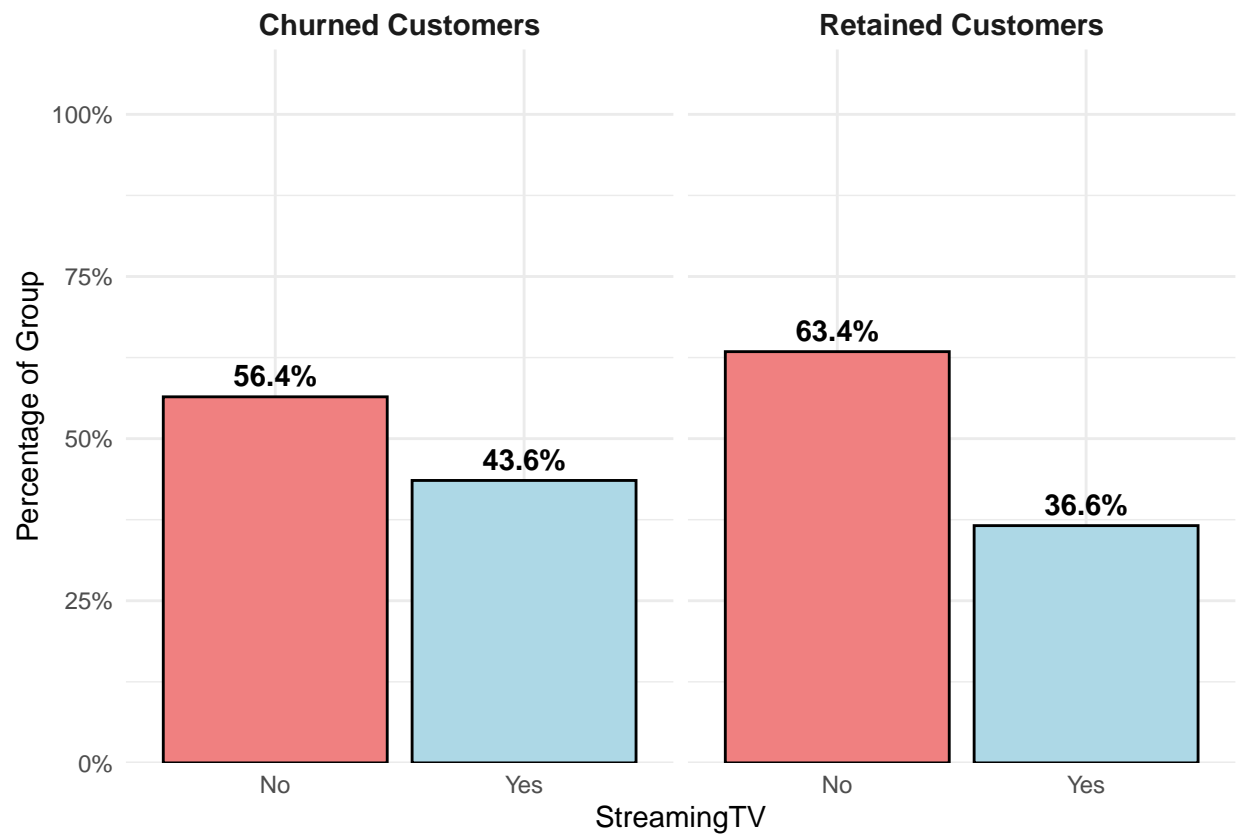
DeviceProtection_plot
```



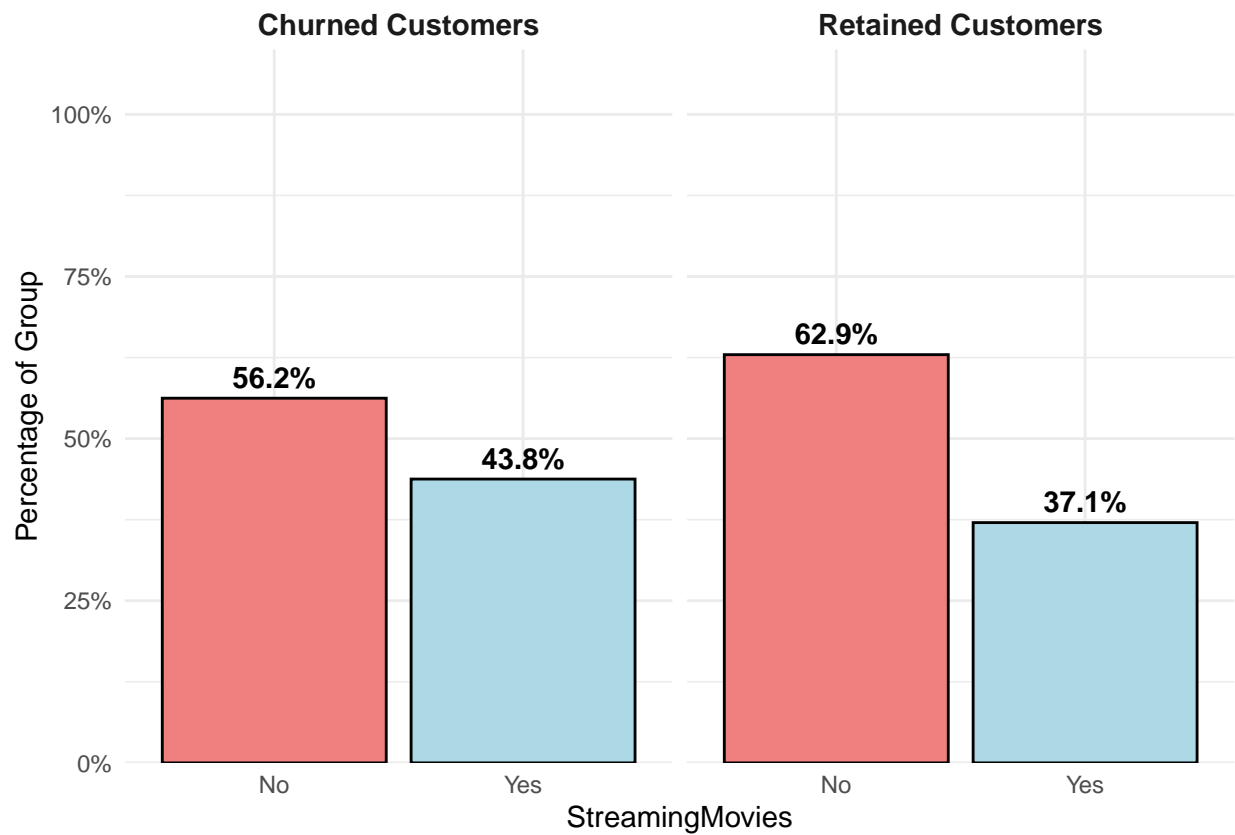
```
TechSupport_results <- create_churn_plot("TechSupport")
TechSupport_plot_data <- TechSupport_results$plot_data
TechSupport_plot <- TechSupport_results$plot
TechSupport_plot
```



```
StreamingTV_results <- create_churn_plot("StreamingTV")
StreamingTV_plot_data <- StreamingTV_results$plot_data
StreamingTV_plot <- StreamingTV_results$plot
StreamingTV_plot
```



```
StreamingMovies_results <- create_churn_plot("StreamingMovies")
StreamingMovies_plot_data <- StreamingMovies_results$plot_data
StreamingMovies_plot <- StreamingMovies_results$plot
StreamingMovies_plot
```



##Plots for Report

```
combined_analysis_service <- (PhoneService_plot + MultipleLines_plot) /
  (StreamingTV_plot + StreamingMovies_plot)

combined_analysis_service + plot_annotation(
  title = "Customer Service Analysis by Churn Status",
  theme = theme(plot.title = element_text(hjust = 0.5, face = "bold"))
)
```

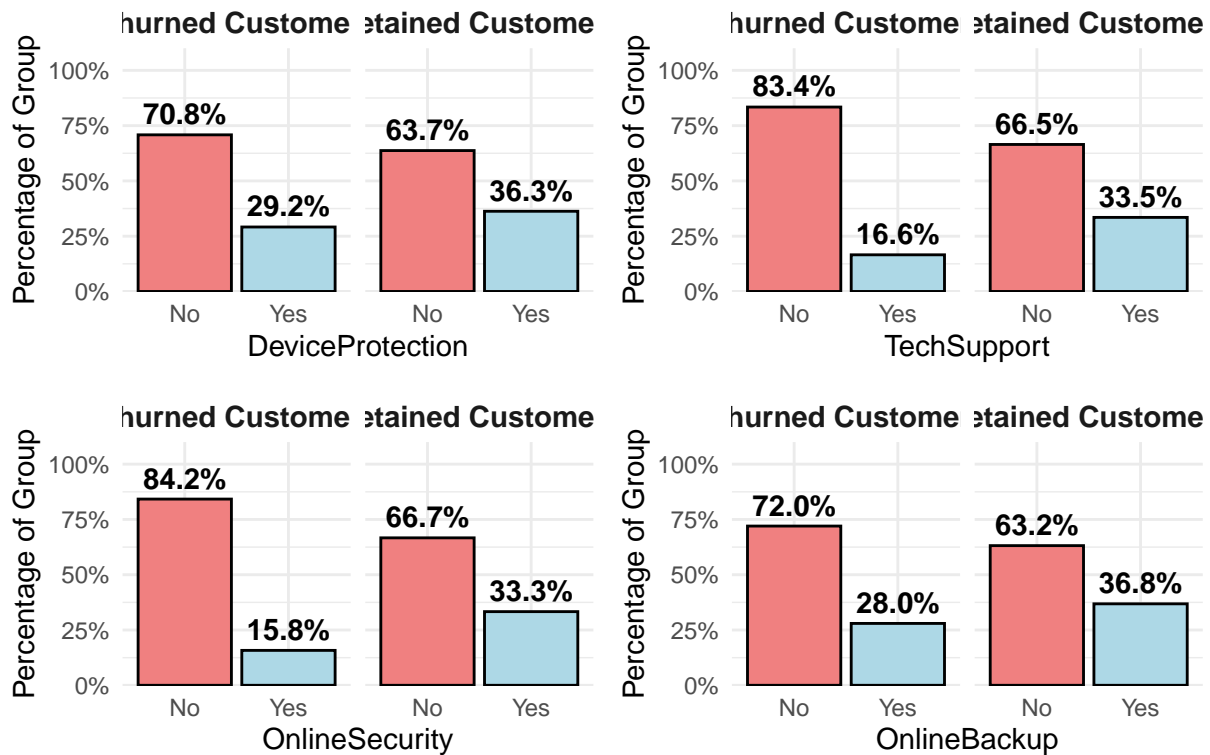
Customer Service Analysis by Churn Status



```
combined_analysis_service2 <- (DeviceProtection_plot + TechSupport_plot) /
  (OnlineSecurity_plot + OnlineBackup_plot)

combined_analysis_service2 + plot_annotation(
  title = "Customer Service Analysis by Churn Status",
  theme = theme(plot.title = element_text(hjust = 0.5, face = "bold"))
)
```


Customer Service Analysis by Churn Status



#Imbalanced Data Set addressed with Rose

```
train_index <- createDataPartition(telco_chrun_clean$Churn, p = 0.8, list = FALSE)
telco_train <- telco_chrun_clean[train_index, ]
telco_test <- telco_chrun_clean[-train_index, ]

train_balanced <- ROSE(Churn ~ ., data = telco_train, seed = 123)$data

cat("Class distribution after ROSE:\n")
```

Class distribution after ROSE:

```
print(table(train_balanced$Churn))
```

```
##
##   No   Yes
## 2834 2793
```

```
x_train_balanced <- model.matrix(Churn ~ . -1, data = train_balanced)
x_test <- model.matrix(Churn ~ . -1, data = telco_test)

y_train_balanced <- train_balanced$Churn
y_test <- telco_test$Churn
```

#Lasso Logistic Regression Model

Meeting the assumptions of the logisitc model

1. Logistic regression assumes linearity of independent variables and log odds of the dependent variable. Although this analysis does not require the dependent and independent variables to be related linearly, it requires that the independent variables are linearly related to the log odds of the dependent variable.
2. Logistic regression requires there to be little or no multicollinearity among the independent variables. Meaning, that the independent variables should not be too highly correlated with each other.
3. Binary logistic regression requires the dependent variable to be binary (which is true no code needed)
4. Logistic regression requires the observations to be independent of each other. (which is true no code needed)
5. Logistic regression typically requires a large sample size. (which is true no code needed)
6. No Extreme Outliers

Addressing Assumption 6: No Extreme Outlier

```
check_outliers <- function(data, continuous_vars) {  
  outlier_results <- list()  
  
  for (var in continuous_vars) {  
    # Calculate summary statistics  
    var_data <- data[[var]]  
    q1 <- quantile(var_data, 0.25, na.rm = TRUE)  
    q3 <- quantile(var_data, 0.75, na.rm = TRUE)  
    iqr <- q3 - q1  
    lower_bound <- q1 - 1.5 * iqr  
    upper_bound <- q3 + 1.5 * iqr  
  
    outliers <- var_data[var_data < lower_bound | var_data > upper_bound]  
    outlier_count <- length(outliers)  
    outlier_percentage <- round((outlier_count / length(var_data)) * 100, 2)  
  
    p <- ggplot(data, aes(y = !!sym(var))) +  
      geom_boxplot(fill = "lightblue", color = "darkblue") +  
      labs(title = paste("Boxplot of", var),  
           y = var) +  
      theme_minimal()  
  
    outlier_results[[var]] <- list(  
      plot = p,  
      stats = data.frame(  
        Variable = var,  
        Q1 = q1,  
        Q3 = q3,
```

```

      IQR = iqr,
      Lower_Bound = lower_bound,
      Upper_Bound = upper_bound,
      Outlier_Count = outlier_count,
      Outlier_Percentage = outlier_percentage
    )
  )
}

return(outlier_results)
}

continuous_vars <- c("tenure", "MonthlyCharges", "TotalCharges")
outlier_analysis <- check_outliers(telco_chrun_clean, continuous_vars)

for (var in continuous_vars) {
  print(outlier_analysis[[var]]$stats)
}

```

```

##      Variable Q1 Q3 IQR Lower_Bound Upper_Bound Outlier_Count Outlier_Percentage
## 25%   tenure  9 55  46         -60         124             0             0
##      Variable      Q1      Q3      IQR Lower_Bound Upper_Bound Outlier_Count
## 25% MonthlyCharges 35.5875 89.8625 54.275      -45.825      171.275             0
##      Outlier_Percentage
## 25%                  0
##      Variable      Q1      Q3      IQR Lower_Bound Upper_Bound Outlier_Count
## 25% TotalCharges 401.45 3794.738 3393.288    -4688.481      8884.669             0
##      Outlier_Percentage
## 25%                  0

```

Addressing Assumption 1: Determining the linearity of independent variables and log odds of the dependent variable.

```

check_logit_linearity <- function(data, continuous_vars) {
  plots <- list()

  for (var in continuous_vars) {
    # Create bins for the continuous variable
    data_binned <- data %>%
      mutate(bin = cut(!!sym(var), breaks = 10, include.lowest = TRUE)) %>%
      group_by(bin) %>%
      summarise(
        mean_var = mean(!!sym(var)),
        churn_rate = mean(as.numeric(Churn) - 1), # Convert to 0/1
        log_odds = log((churn_rate + 0.001) / (1 - churn_rate + 0.001)) # Avoid log(0)
      )

    # Create scatter plot
    p <- ggplot(data_binned, aes(x = mean_var, y = log_odds)) +
      geom_point(size = 3, color = "steelblue") +

```

```

    geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(title = paste("Linearity Check:", var),
         x = var, y = "Log Odds of Churn") +
    theme_minimal()

    plots[[var]] <- p
  }

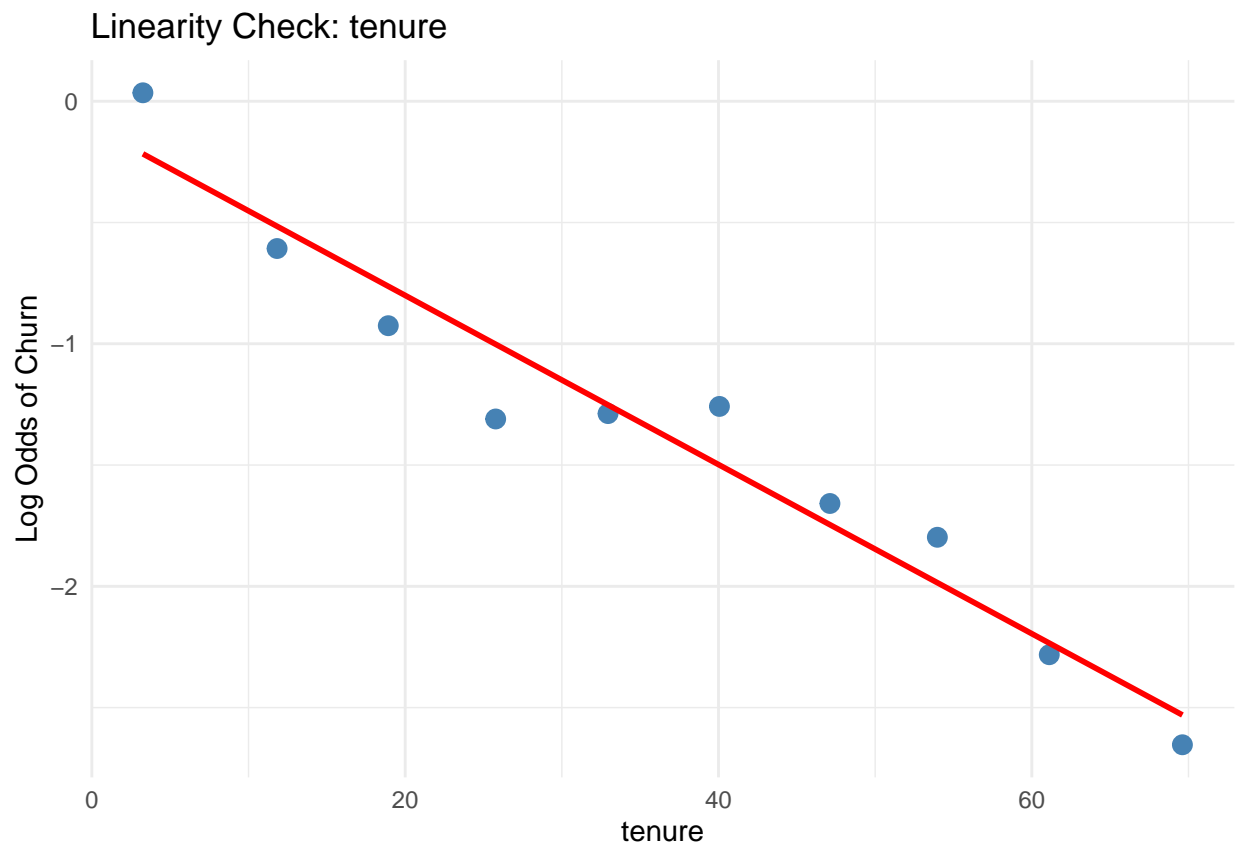
  return(plots)
}

# Check linearity for continuous variables
continuous_vars <- c("tenure", "MonthlyCharges", "TotalCharges")
linearity_plots <- check_logit_linearity(telco_chrun_clean, continuous_vars)

# Display plots
linearity_plots$tenure

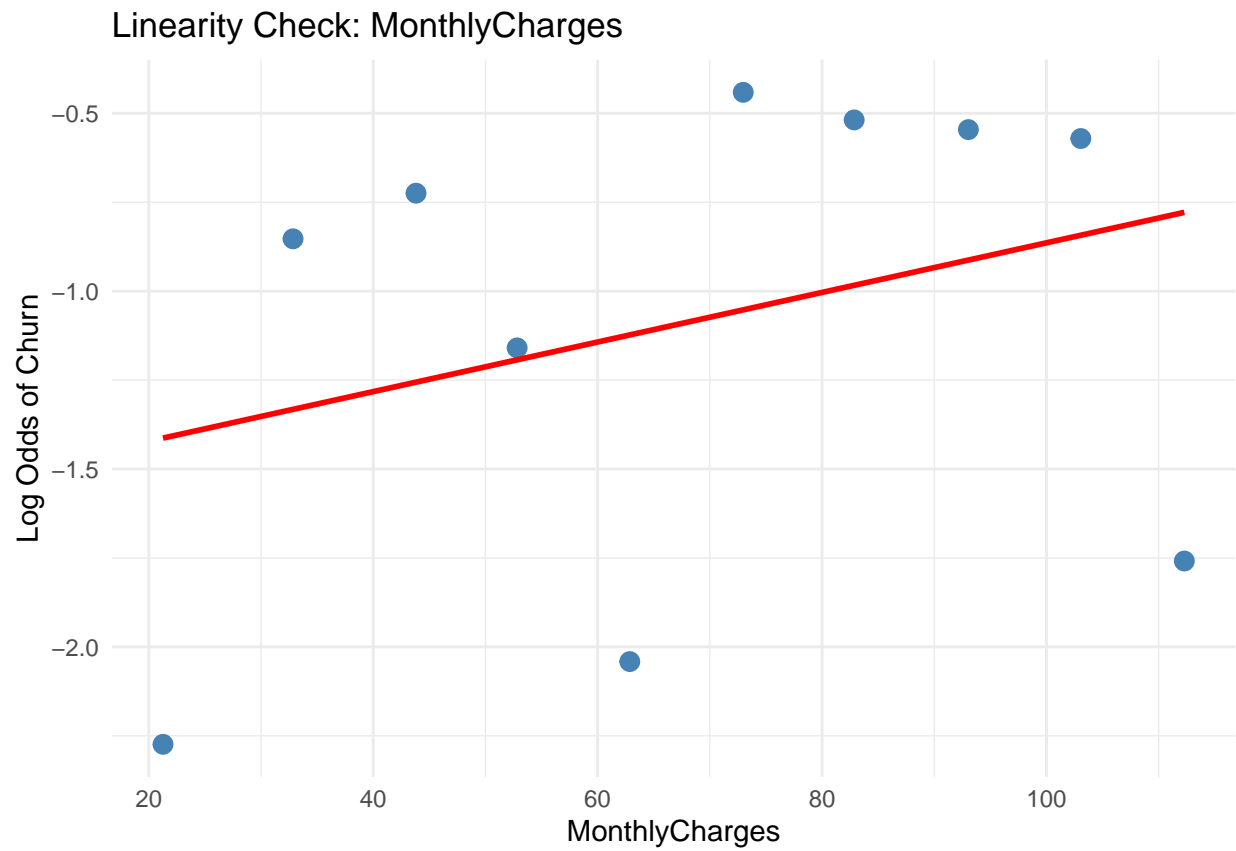
```

'geom_smooth()' using formula = 'y ~ x'



```
linearity_plots$MonthlyCharges
```

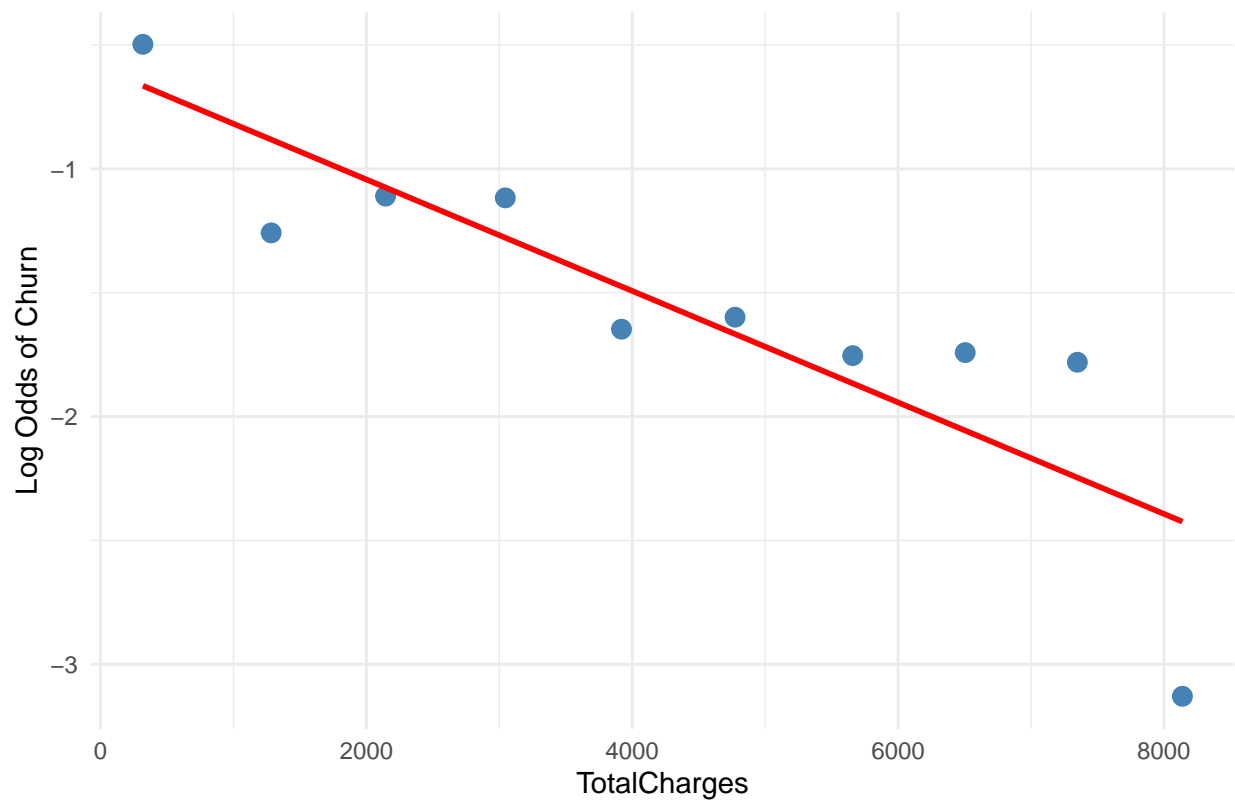
'geom_smooth()' using formula = 'y ~ x'



```
linearity_plots$TotalCharges
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

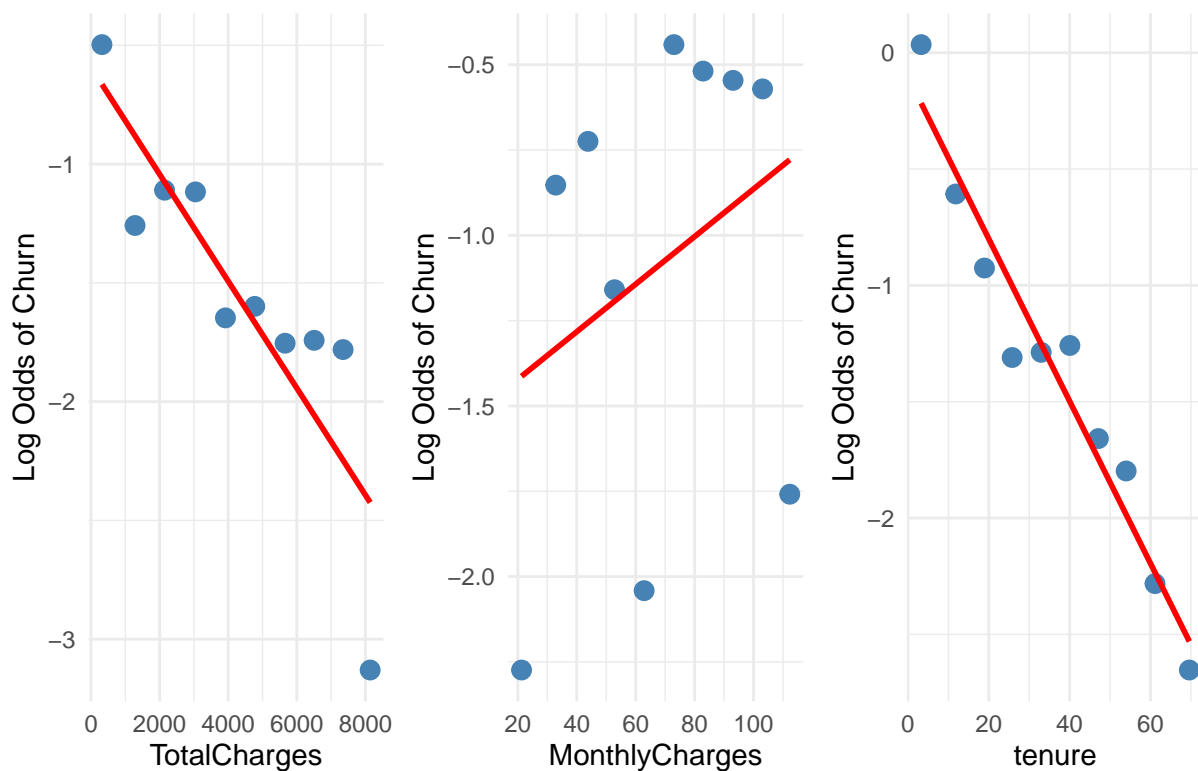
Linearity Check: TotalCharges



```
combined_linearity_plots <- (linearity_plots$TotalCharges + linearity_plots$MonthlyCharges + linearity_pl  
combined_linearity_plots
```

```
## 'geom_smooth()' using formula = 'y ~ x'  
## 'geom_smooth()' using formula = 'y ~ x'  
## 'geom_smooth()' using formula = 'y ~ x'
```

Linearity Check: TotalCharges Linearity Check: MonthlyCharges Linearity Check: tenure



Turning Monthly Charges into a categorical variable

```
telco_chrun_clean_lr <- telco_chrun_clean %>%
  mutate(
    MonthlyCharges_cat = case_when(
      MonthlyCharges < 35 ~ "Low",
      MonthlyCharges >= 35 & MonthlyCharges < 65 ~ "Medium",
      MonthlyCharges >= 65 & MonthlyCharges < 90 ~ "High",
      MonthlyCharges >= 90 ~ "Very High"
    ),
    MonthlyCharges_cat = factor(MonthlyCharges_cat,
                                levels = c("Low", "Medium",
                                             "High", "Very High"))
  )

monthly_charges_summary <- telco_chrun_clean_lr %>%
  group_by(MonthlyCharges_cat) %>%
  summarise(
    n_customers = n(),
    avg_monthly_charge = mean(MonthlyCharges),
    churn_rate = mean(Churn == "Yes"),
    .groups = 'drop'
  ) %>%
  mutate(
```

```

    percent_total = n_customers / sum(n_customers),
    churn_percent = scales::percent(churn_rate, accuracy = 0.1)
  )

print("Monthly Charges Category Summary:")

```

```
## [1] "Monthly Charges Category Summary:"
```

```
print(monthly_charges_summary)
```

```
## # A tibble: 4 x 6
##   MonthlyCharges_cat n_customers avg_monthly_charge churn_rate percent_total
##   <fct>              <int>          <dbl>         <dbl>         <dbl>
## 1 Low                1725            22.0          0.109          0.245
## 2 Medium             1405            51.7          0.232          0.200
## 3 High               2158            78.3          0.362          0.307
## 4 Very High          1744           101.          0.329          0.248
## # i 1 more variable: churn_percent <chr>
```

```

distribution_plot <- ggplot(monthly_charges_summary, aes(x = MonthlyCharges_cat, y = n_customers, fill = MonthlyCharges_cat)) +
  geom_col(show.legend = FALSE) +
  geom_text(aes(label = scales::comma(n_customers)), vjust = -0.5, size = 4, fontface = "bold") +
  labs(title = "Distribution of Customers by Monthly Charges Category",
       x = "Monthly Charges Category",
       y = "Number of Customers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

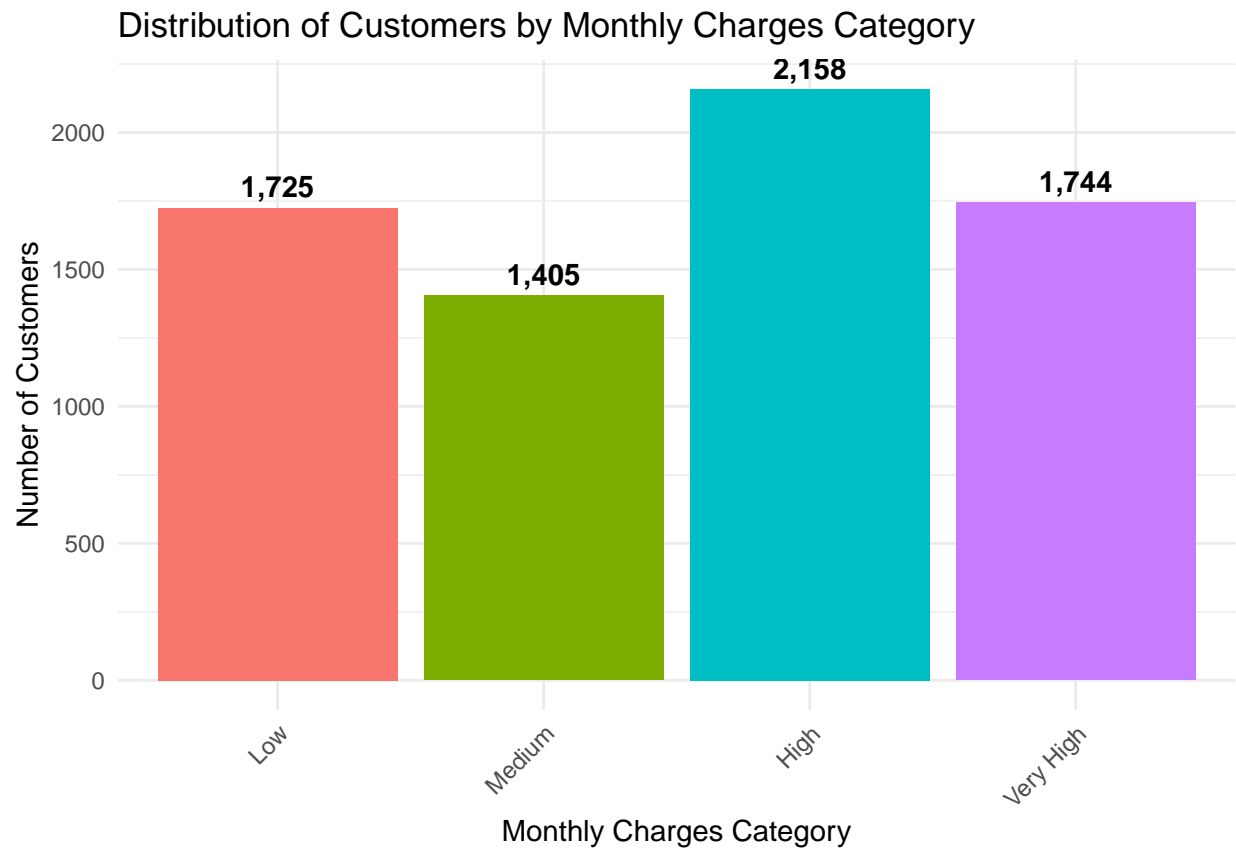
```

```

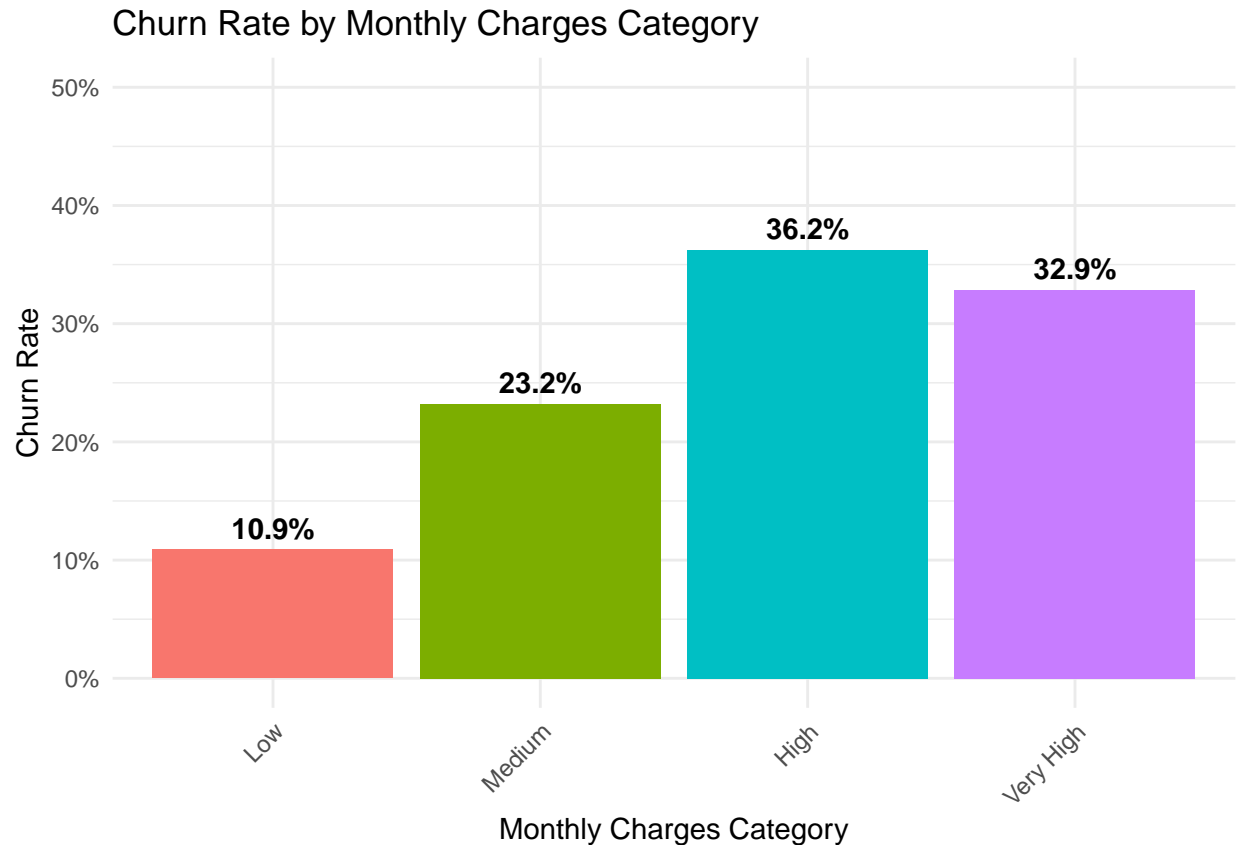
churn_plot <- ggplot(monthly_charges_summary, aes(x = MonthlyCharges_cat, y = churn_rate, fill = MonthlyCharges_cat)) +
  geom_col(show.legend = FALSE) +
  geom_text(aes(label = churn_percent), vjust = -0.5, size = 4, fontface = "bold") +
  scale_y_continuous(labels = scales::percent_format(), limits = c(0, 0.5)) +
  labs(title = "Churn Rate by Monthly Charges Category",
       x = "Monthly Charges Category",
       y = "Churn Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

```
distribution_plot
```

churn_plot

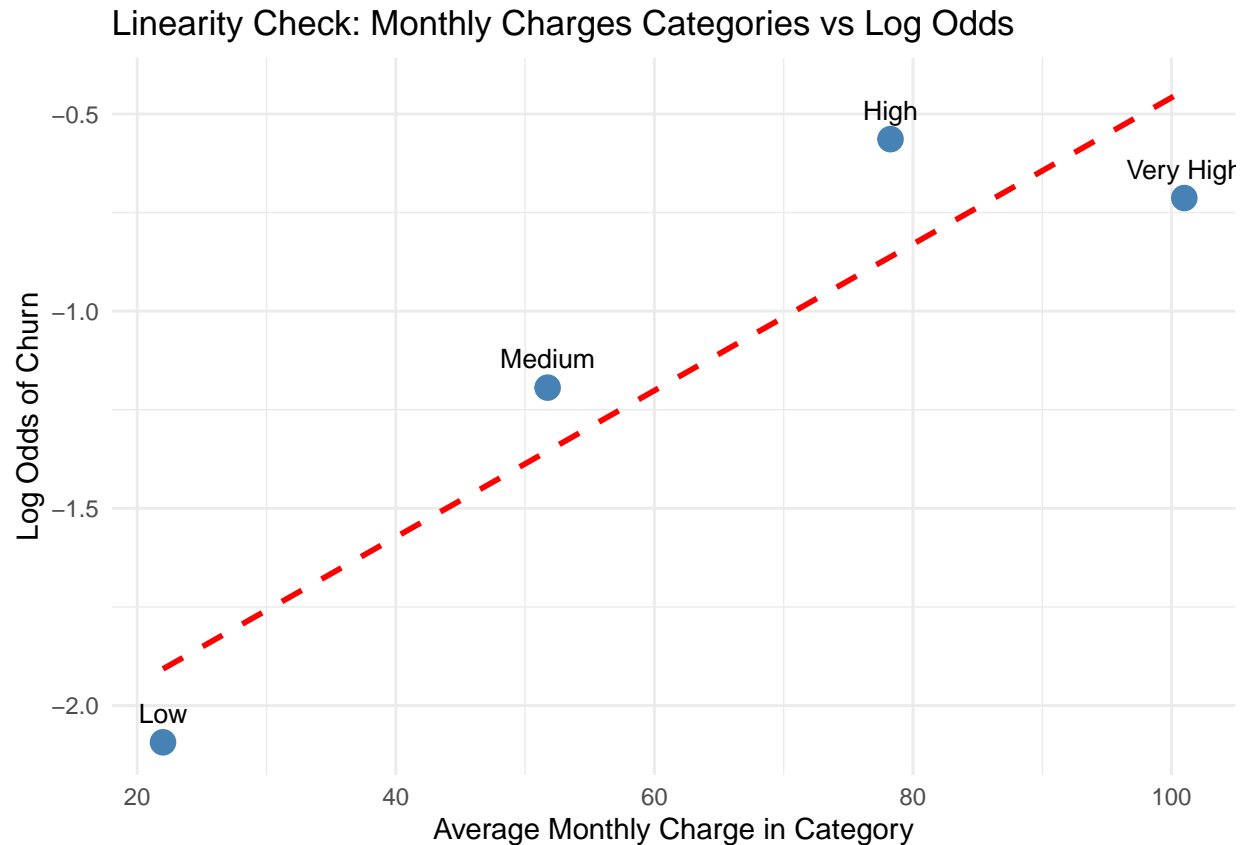


```
data_binned_cat <- telco_chrun_clean_lr %>%
  group_by(MonthlyCharges_cat) %>%
  summarise(
    avg_charge = mean(MonthlyCharges),
    churn_rate = mean(as.numeric(Churn) - 1),
    log_odds = log((churn_rate + 0.001) / (1 - churn_rate + 0.001))
  )

cat_linearity_plot <- ggplot(data_binned_cat, aes(x = avg_charge, y = log_odds)) +
  geom_point(size = 4, color = "steelblue") +
  geom_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +
  geom_text(aes(label = MonthlyCharges_cat), vjust = -1, size = 3.5) +
  labs(title = "Linearity Check: Monthly Charges Categories vs Log Odds",
       x = "Average Monthly Charge in Category",
       y = "Log Odds of Churn") +
  theme_minimal()

cat_linearity_plot
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
telco_for_modeling <- telco_chrun_clean_lr %>%
  mutate(MonthlyCharges_cat = as.factor(MonthlyCharges_cat))
```

```
telco_for_modeling <- telco_chrun_clean_lr %>%
  select(-MonthlyCharges)
```

```
cat("Final dataset structure for modeling:\n")
```

```
## Final dataset structure for modeling:
```

```
str(telco_for_modeling$MonthlyCharges_cat)
```

```
## Factor w/ 4 levels "Low","Medium",...: 1 2 2 2 3 4 3 1 4 2 ...
```

```
table(telco_for_modeling$MonthlyCharges_cat)
```

```
##
##      Low      Medium      High Very High
##    1725     1405     2158      1744
```

Lasso Variable Selection

```

set.seed(123)
train_index <- createDataPartition(telco_for_modeling$Churn, p = 0.8, list = FALSE)
telco_train <- telco_for_modeling[train_index, ]
telco_test <- telco_for_modeling[-train_index, ]

train_balanced <- ROSE(Churn ~ ., data = telco_train, seed = 1)$data

cat("Class distribution after ROSE:\n")

```

```
## Class distribution after ROSE:
```

```
print(table(train_balanced$Churn))
```

```
##
##   No  Yes
## 2866 2761
```

```

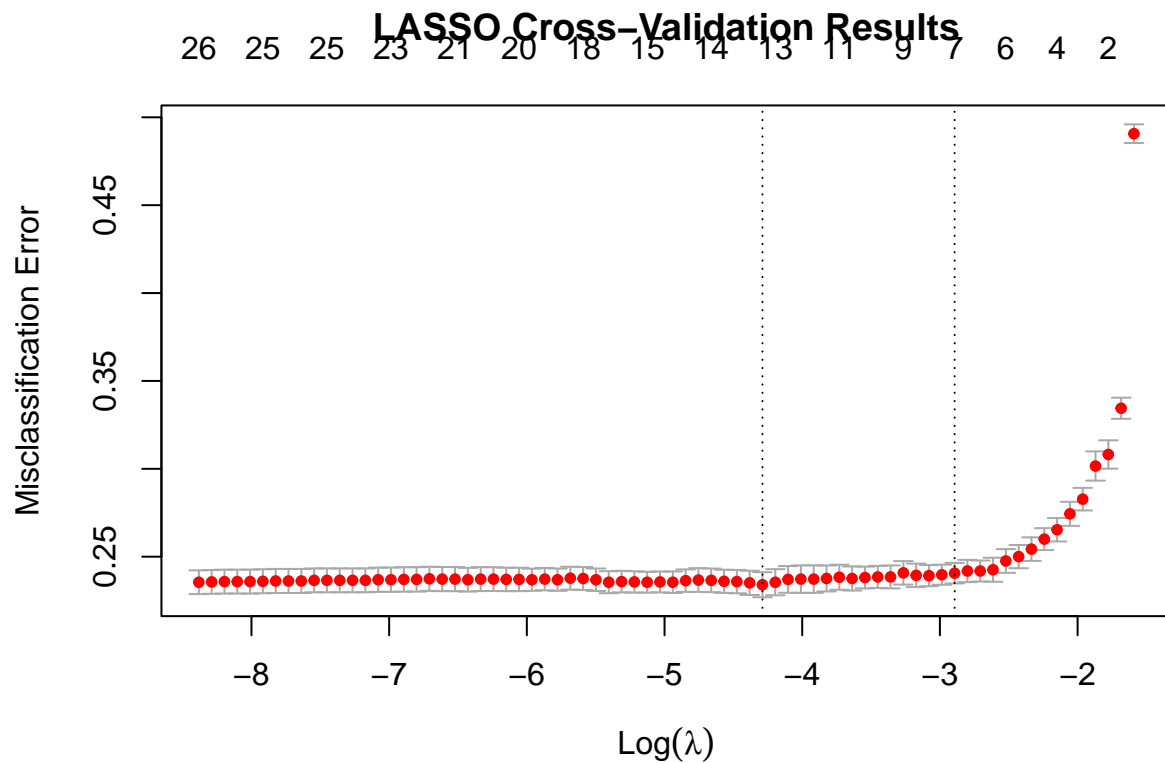
x_train <- model.matrix(Churn ~ . -1, data = train_balanced)
y_train <- as.numeric(train_balanced$Churn) - 1 # Convert to 0/1

x_test <- model.matrix(Churn ~ . -1, data = telco_test)
y_test <- as.numeric(telco_test$Churn) - 1

set.seed(123)
cv_lasso <- cv.glmnet(x_train, y_train,
                      family = "binomial",
                      alpha = 1,
                      type.measure = "class",
                      nfolds = 10)

plot(cv_lasso, main = "LASSO Cross-Validation Results")

```



```
lambda_min <- cv_lasso$lambda.min
lambda_1se <- cv_lasso$lambda.1se

cat("Minimum lambda:", round(lambda_min, 4), "\n")
```

```
## Minimum lambda: 0.0137
```

```
cat("1 SE lambda:", round(lambda_1se, 4), "\n")
```

```
## 1 SE lambda: 0.0554
```

```
lasso_model <- glmnet(x_train, y_train,
                      family = "binomial",
                      alpha = 1,
                      lambda = lambda_1se)
```

```
lasso_coef <- coef(lasso_model)
print("LASSO Coefficients:")
```

```
## [1] "LASSO Coefficients:"
```

```
print(lasso_coef)
```

```
## 27 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## (Intercept)                   0.24922638
## genderFemale                   .
## genderMale                     .
## SeniorCitizen1                 .
## PartnerYes                     .
## DependentsYes                  .
## tenure                        -0.01932266
## PhoneServiceYes                .
## MultipleLinesYes               .
## InternetServiceFiberOptic      0.59532195
## InternetServiceNo              -0.31157191
## OnlineSecurityYes              .
## OnlineBackupYes                .
## DeviceProtectionYes            .
## TechSupportYes                 .
## StreamingTVYes                 .
## StreamingMoviesYes             .
## ContractOne year               -0.26266102
## ContractTwo year               -0.77199160
## PaperlessBillingYes            0.04412032
## PaymentMethodCreditCardAuto    .
## PaymentMethodECheck            0.25277584
## PaymentMethodMailedCheck       .
## TotalCharges                   .
## MonthlyCharges_catMedium        .
## MonthlyCharges_catHigh          .
## MonthlyCharges_catVery High    .
```

```
selected_vars <- rownames(lasso_coef)[which(lasso_coef != 0)]
selected_vars <- selected_vars[selected_vars != "(Intercept)"]
cat("\nSelected variables by LASSO:", paste(selected_vars, collapse = ", "), "\n")
```

```
##
## Selected variables by LASSO: tenure, InternetServiceFiberOptic, InternetServiceNo, ContractOne year,
```

```
selected_summary <- data.frame(
  Variable = rownames(lasso_coef)[which(lasso_coef != 0)],
  Coefficient = lasso_coef[which(lasso_coef != 0)],
  Odds_Ratio = exp(lasso_coef[which(lasso_coef != 0)])
)
rownames(selected_summary) <- NULL

print("Selected Variables with Coefficients and Odds Ratios:")
```

```
## [1] "Selected Variables with Coefficients and Odds Ratios:"
```

```
print(selected_summary)
```

```
##               Variable Coefficient Odds_Ratio
## 1      (Intercept)  0.24922638  1.2830325
```

```
## 2          tenure -0.01932266  0.9808628
## 3 InternetServiceFiberOptic  0.59532195  1.8136147
## 4      InternetServiceNo -0.31157191  0.7322949
## 5      ContractOne year -0.26266102  0.7690025
## 6      ContractTwo year -0.77199160  0.4620918
## 7      PaperlessBillingYes  0.04412032  1.0451081
## 8      PaymentMethodECheck  0.25277584  1.2875946
```

Building Logistic Regression Model

```
final_model <- glm(Churn ~ tenure + Contract + InternetService + PaperlessBilling + PaymentMethod + SeniorCitizen + OnlineSecurity,
                  data = train_balanced,
                  family = "binomial")

cat("\nFinal Model Summary:\n")
```

```
##
## Final Model Summary:
```

```
summary(final_model)
```

```
##
## Call:
## glm(formula = Churn ~ tenure + Contract + InternetService + PaperlessBilling +
##      PaymentMethod + SeniorCitizen + OnlineSecurity, family = "binomial",
##      data = train_balanced)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3464  -0.7804  -0.1832   0.7817   2.7723
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.412553   0.113789   3.626 0.000288 ***
## tenure        -0.028814   0.001983 -14.527 < 2e-16 ***
## ContractOne year -0.771466   0.099382  -7.763 8.32e-15 ***
## ContractTwo year -1.618989   0.155031 -10.443 < 2e-16 ***
## InternetServiceFiberOptic  0.775648   0.079590   9.746 < 2e-16 ***
## InternetServiceNo -0.892255   0.115774  -7.707 1.29e-14 ***
## PaperlessBillingYes  0.358641   0.073916   4.852 1.22e-06 ***
## PaymentMethodCreditCardAuto  0.030231   0.112739   0.268 0.788584
## PaymentMethodECheck  0.333496   0.095442   3.494 0.000475 ***
## PaymentMethodMailedCheck -0.147393   0.111508  -1.322 0.186231
## SeniorCitizen1  0.543213   0.088006   6.172 6.72e-10 ***
## OnlineSecurityYes -0.377700   0.084062  -4.493 7.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 7798.7  on 5626  degrees of freedom
```

```
## Residual deviance: 5566.1 on 5615 degrees of freedom
## AIC: 5590.1
##
## Number of Fisher Scoring iterations: 5
```

```
odds_ratios <- exp(coef(final_model))
conf_int <- exp(confint(final_model))
```

```
## Waiting for profiling to be done...
```

```
model_summary <- data.frame(
  Variable = names(coef(final_model)),
  Coefficient = coef(final_model),
  Odds_Ratio = odds_ratios,
  CI_Lower = conf_int[,1],
  CI_Upper = conf_int[,2],
  p_value = summary(final_model)$coefficients[,4]
)

print("Model Coefficients with Odds Ratios and Confidence Intervals:")
```

```
## [1] "Model Coefficients with Odds Ratios and Confidence Intervals:"
```

```
print(model_summary, digits = 3)
```

##	Variable	Coefficient	Odds_Ratio
## (Intercept)	(Intercept)	0.4126	1.511
## tenure	tenure	-0.0288	0.972
## ContractOne year	ContractOne year	-0.7715	0.462
## ContractTwo year	ContractTwo year	-1.6190	0.198
## InternetServiceFiberOptic	InternetServiceFiberOptic	0.7756	2.172
## InternetServiceNo	InternetServiceNo	-0.8923	0.410
## PaperlessBillingYes	PaperlessBillingYes	0.3586	1.431
## PaymentMethodCreditCardAuto	PaymentMethodCreditCardAuto	0.0302	1.031
## PaymentMethodECheck	PaymentMethodECheck	0.3335	1.396
## PaymentMethodMailedCheck	PaymentMethodMailedCheck	-0.1474	0.863
## SeniorCitizen1	SeniorCitizen1	0.5432	1.722
## OnlineSecurityYes	OnlineSecurityYes	-0.3777	0.685
##	CI_Lower	CI_Upper	p_value
## (Intercept)	1.209	1.889	2.88e-04
## tenure	0.968	0.975	8.12e-48
## ContractOne year	0.380	0.561	8.32e-15
## ContractTwo year	0.145	0.267	1.58e-25
## InternetServiceFiberOptic	1.859	2.539	1.93e-22
## InternetServiceNo	0.326	0.514	1.29e-14
## PaperlessBillingYes	1.238	1.654	1.22e-06
## PaymentMethodCreditCardAuto	0.826	1.286	7.89e-01
## PaymentMethodECheck	1.157	1.683	4.75e-04
## PaymentMethodMailedCheck	0.694	1.074	1.86e-01
## SeniorCitizen1	1.450	2.047	6.72e-10
## OnlineSecurityYes	0.581	0.808	7.02e-06

Addressing Assumption 2 by Checking Multicollinearity

```
vif_final <- vif(final_model)
print("\nVariance Inflation Factors for Final Model:")
```

```
## [1] "\nVariance Inflation Factors for Final Model:"
```

```
print(vif_final)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## tenure        1.792068 1      1.338681
## Contract      1.531658 2      1.112475
## InternetService 1.512529 2      1.108986
## PaperlessBilling 1.136897 1      1.066253
## PaymentMethod  1.374494 3      1.054445
## SeniorCitizen  1.089690 1      1.043882
## OnlineSecurity  1.162539 1      1.078211
```

```
if (all(vif_final <= 5)) {
  cat("Multicollinearity issue resolved! All VIF values <= 5\n")
} else {
  high_vif_final <- names(vif_final[vif_final > 5])
  cat("Warning: Some multicollinearity remains in:", paste(high_vif_final, collapse = ", "), "\n")
}
```

```
## Multicollinearity issue resolved! All VIF values <= 5
```

Model Evaluation

```
predictions_prob <- predict(final_model, newdata = telco_test, type = "response")
predictions_class <- ifelse(predictions_prob > 0.5, 1, 0)

conf_matrix <- table(Predicted = predictions_class, Actual = y_test)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```
print(conf_matrix)
```

```
##      Actual
## Predicted  0   1
##          0 746  77
##          1 286 296
```

```

accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
sensitivity <- conf_matrix[2,2] / sum(conf_matrix[,2])
specificity <- conf_matrix[1,1] / sum(conf_matrix[,1])
precision <- conf_matrix[2,2] / sum(conf_matrix[2,])
f1_score <- 2 * (precision * sensitivity) / (precision + sensitivity)

roc_obj <- roc(y_test, predictions_prob)

```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_value <- auc(roc_obj)
```

```
cat("\nModel Performance Metrics:\n")
```

```
##
```

```
## Model Performance Metrics:
```

```
cat("Accuracy:", round(accuracy, 3), "\n")
```

```
## Accuracy: 0.742
```

```
cat("Sensitivity (Recall):", round(sensitivity, 3), "\n")
```

```
## Sensitivity (Recall): 0.794
```

```
cat("Specificity:", round(specificity, 3), "\n")
```

```
## Specificity: 0.723
```

```
cat("Precision:", round(precision, 3), "\n")
```

```
## Precision: 0.509
```

```
cat("F1 Score:", round(f1_score, 3), "\n")
```

```
## F1 Score: 0.62
```

```
cat("AUC:", round(auc_value, 3), "\n")
```

```
## AUC: 0.838
```

Random Forest

```

set.seed(123)

train_index <- createDataPartition(telco_chrun_clean$Churn, p = 0.8, list = FALSE)
telco_train <- telco_chrun_clean[train_index, ]
telco_test <- telco_chrun_clean[-train_index, ]

train_balanced <- ROSE(Churn ~ ., data = telco_train, seed = 123)$data

cat("Class distribution after ROSE:\n")

```

```
## Class distribution after ROSE:
```

```
print(table(train_balanced$Churn))
```

```
##
##      No   Yes
## 2834 2793
```

Variable Selection

```

set.seed(123)
rf_initial <- randomForest(Churn ~ .,
                           data = train_balanced,
                           ntree = 500,
                           importance = TRUE,
                           do.trace = 100)

```

```
## ntree      OOB      1      2
##   100:  17.22% 20.96% 13.43%
##   200:  17.10% 20.92% 13.21%
##   300:  16.78% 20.96% 12.53%
##   400:  16.87% 21.03% 12.64%
##   500:  16.95% 21.28% 12.57%
```

```

var_importance <- importance(rf_initial)
var_importance_df <- data.frame(
  Variable = rownames(var_importance),
  Importance = var_importance[, "MeanDecreaseGini"]
) %>%
  arrange(desc(Importance))

print("Top 15 Most Important Variables:")

```

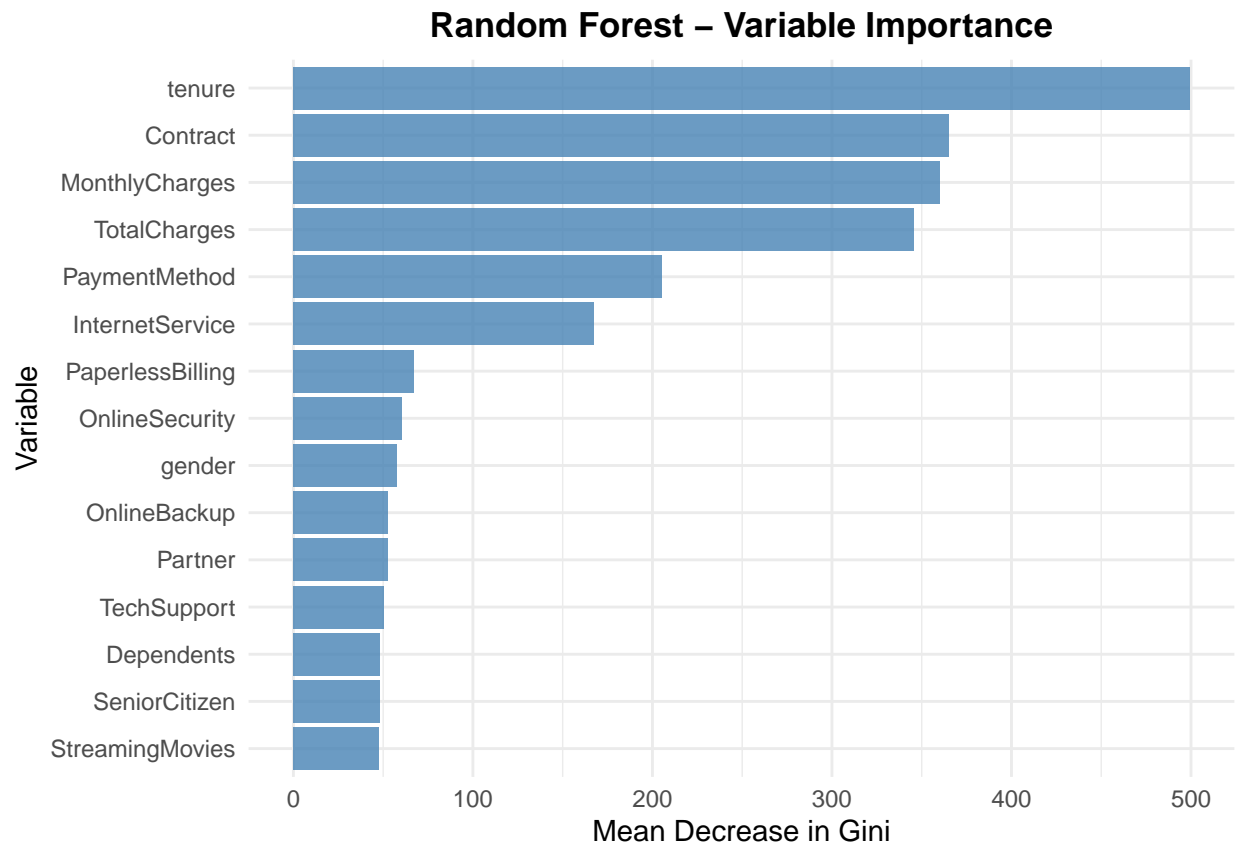
```
## [1] "Top 15 Most Important Variables:"
```

```
print(head(var_importance_df, 15))
```

##	Variable	Importance
## tenure	tenure	499.20435
## Contract	Contract	364.92943
## MonthlyCharges	MonthlyCharges	360.05316
## TotalCharges	TotalCharges	345.81818
## PaymentMethod	PaymentMethod	205.38618
## InternetService	InternetService	167.53865
## PaperlessBilling	PaperlessBilling	67.13078
## OnlineSecurity	OnlineSecurity	60.55211
## gender	gender	57.88723
## OnlineBackup	OnlineBackup	52.91792
## Partner	Partner	52.62060
## TechSupport	TechSupport	50.17606
## Dependents	Dependents	48.26610
## SeniorCitizen	SeniorCitizen	48.05320
## StreamingMovies	StreamingMovies	47.78629

```
var_imp_plot <- ggplot(head(var_importance_df, 15),
                        aes(x = reorder(Variable, Importance), y = Importance)) +
  geom_col(fill = "steelblue", alpha = 0.8) +
  coord_flip() +
  labs(title = "Random Forest - Variable Importance",
       x = "Variable",
       y = "Mean Decrease in Gini") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

print(var_imp_plot)
```



```
selected_vars_rf <- var_importance_df$Variable[1:10] # Top 10 variables
cat("\nSelected variables for final model:", paste(selected_vars_rf, collapse = ", "), "\n")
```

```
##
## Selected variables for final model: tenure, Contract, MonthlyCharges, TotalCharges, PaymentMethod, I
```

Hypertuning Parameters

```
cat("\n=== HYPERPARAMETER TUNING ===\n")
```

```
##
## === HYPERPARAMETER TUNING ===
```

```
tune_grid <- expand.grid(
  mtry = c(2, 3, 4, 5, 6)
)

ctrl <- trainControl(
  method = "cv",
  number = 5,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
```

```

    verboseIter = TRUE,
    search = "grid"
)

formula_selected <- as.formula(paste("Churn ~", paste(selected_vars_rf, collapse = " + ")))

set.seed(123)
rf_tuned <- train(
  formula_selected,
  data = train_balanced,
  method = "rf",
  metric = "ROC",
  trControl = ctrl,
  tuneGrid = tune_grid,
  ntree = 500,
  importance = TRUE
)

```

```

## + Fold1: mtry=2
## - Fold1: mtry=2
## + Fold1: mtry=3
## - Fold1: mtry=3
## + Fold1: mtry=4
## - Fold1: mtry=4
## + Fold1: mtry=5
## - Fold1: mtry=5
## + Fold1: mtry=6
## - Fold1: mtry=6
## + Fold2: mtry=2
## - Fold2: mtry=2
## + Fold2: mtry=3
## - Fold2: mtry=3
## + Fold2: mtry=4
## - Fold2: mtry=4
## + Fold2: mtry=5
## - Fold2: mtry=5
## + Fold2: mtry=6
## - Fold2: mtry=6
## + Fold3: mtry=2
## - Fold3: mtry=2
## + Fold3: mtry=3
## - Fold3: mtry=3
## + Fold3: mtry=4
## - Fold3: mtry=4
## + Fold3: mtry=5
## - Fold3: mtry=5
## + Fold3: mtry=6
## - Fold3: mtry=6
## + Fold4: mtry=2
## - Fold4: mtry=2
## + Fold4: mtry=3
## - Fold4: mtry=3
## + Fold4: mtry=4

```

```

## - Fold4: mtry=4
## + Fold4: mtry=5
## - Fold4: mtry=5
## + Fold4: mtry=6
## - Fold4: mtry=6
## + Fold5: mtry=2
## - Fold5: mtry=2
## + Fold5: mtry=3
## - Fold5: mtry=3
## + Fold5: mtry=4
## - Fold5: mtry=4
## + Fold5: mtry=5
## - Fold5: mtry=5
## + Fold5: mtry=6
## - Fold5: mtry=6
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 4 on full training set

```

```
cat("Best tuning parameters:\n")
```

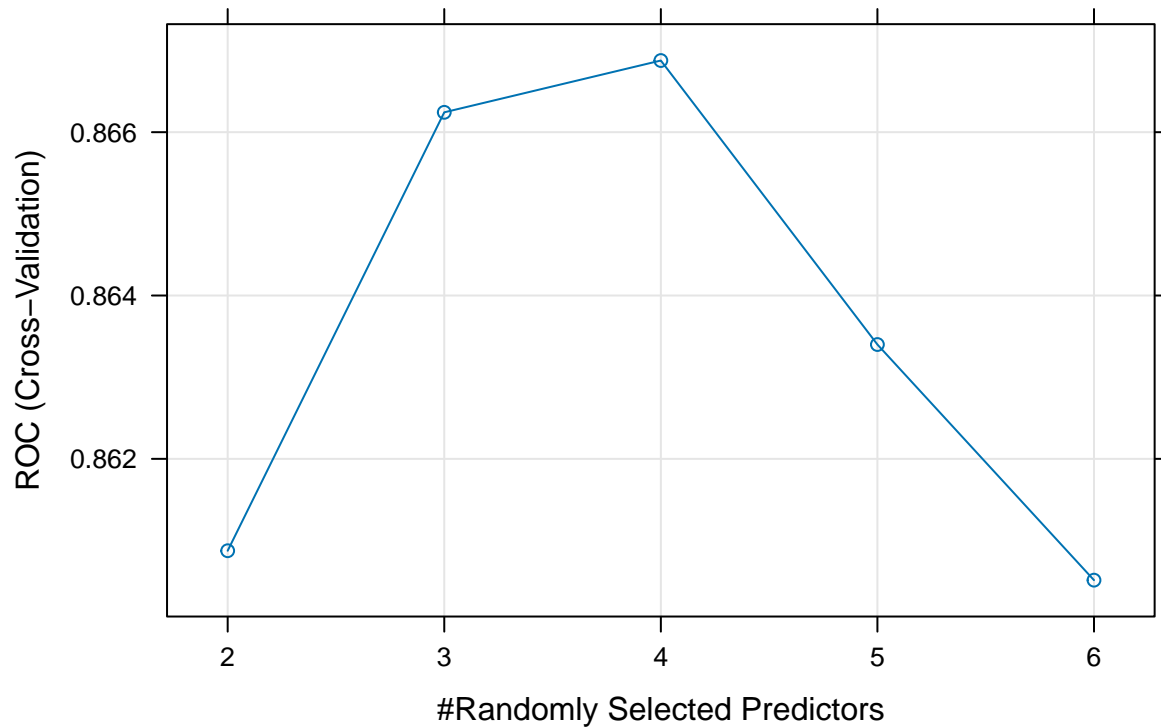
```
## Best tuning parameters:
```

```
print(rf_tuned$bestTune)
```

```
##      mtry
## 3      4
```

```
plot(rf_tuned, main = "Random Forest Tuning Results")
```

Random Forest Tuning Results



```
## Building the Model
```

```
cat("\n=== FINAL RANDOM FOREST MODEL ===\n")
```

```
##
```

```
## === FINAL RANDOM FOREST MODEL ===
```

```
final_rf <- randomForest(  
  formula_selected,  
  data = train_balanced,  
  mtry = rf_tuned$bestTune$mtry,  
  nodesize = rf_tuned$bestTune$nodesize,  
  importance = TRUE  
)
```

```
cat("Final Random Forest Model Summary:\n")
```

```
## Final Random Forest Model Summary:
```

```
print(final_rf)
```

```
##
```

```
## Call:
```

```
## randomForest(formula = formula_selected, data = train_balanced, mtry = rf_tuned$bestTune$mtry,  
## Type of random forest: classification
```



```
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##          OOB estimate of  error rate: 21.04%
## Confusion matrix:
##      No  Yes class.error
## No  2160  674   0.2378264
## Yes   510 2283   0.1825994
```

Model Evaluation

```
cat("\n=== MODEL EVALUATION ===\n")
```

```
##
## === MODEL EVALUATION ===
```

```
predictions_prob <- predict(final_rf, newdata = telco_test, type = "prob")[, "Yes"]
predictions_class <- predict(final_rf, newdata = telco_test, type = "response")
```

```
y_test_numeric <- ifelse(telco_test$Churn == "Yes", 1, 0)
pred_class_numeric <- ifelse(predictions_class == "Yes", 1, 0)
```

```
conf_matrix <- table(Predicted = pred_class_numeric, Actual = y_test_numeric)
print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```
print(conf_matrix)
```

```
##           Actual
## Predicted    0    1
##           0 755  92
##           1 277 281
```

```
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
sensitivity <- conf_matrix[2,2] / sum(conf_matrix[,2])
specificity <- conf_matrix[1,1] / sum(conf_matrix[,1])
precision <- conf_matrix[2,2] / sum(conf_matrix[2,])
f1_score <- 2 * (precision * sensitivity) / (precision + sensitivity)
```

```
roc_obj <- roc(y_test_numeric, predictions_prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_score <- auc(roc_obj)
```

```
cat("\nModel Performance Metrics:\n")
```

```
##
## Model Performance Metrics:

cat("Accuracy:", round(accuracy, 3), "\n")

## Accuracy: 0.737

cat("Sensitivity (Recall):", round(sensitivity, 3), "\n")

## Sensitivity (Recall): 0.753

cat("Specificity:", round(specificity, 3), "\n")

## Specificity: 0.732

cat("Precision:", round(precision, 3), "\n")

## Precision: 0.504

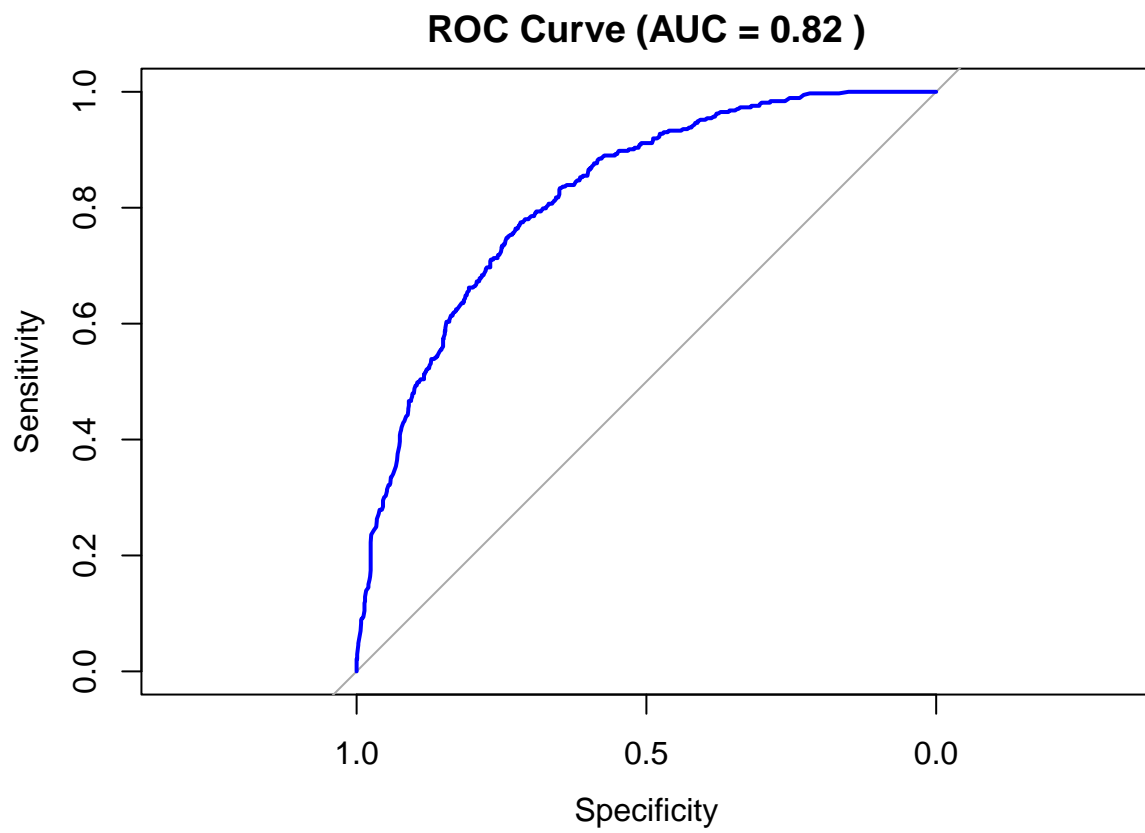
cat("F1 Score:", round(f1_score, 3), "\n")

## F1 Score: 0.604

cat("AUC:", round(auc_score, 3), "\n")

## AUC: 0.82

plot(roc_obj, main = paste("ROC Curve (AUC =", round(auc_score, 3), ")"),
     col = "blue", lwd = 2)
```



““