EDA

AUTHOR
Skylar Hildebrand

```r
# Load required libraries
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.2     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.1.0
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
```

```r
library(corrplot)
```

```
corrplot 0.95 loaded
```

```r
library(ggplot2)
library(gridExtra)
```

```

Attaching package: 'gridExtra'

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

    combine
```

```r
library(scales)
```

```

Attaching package: 'scales'

The following object is masked from 'package:purrr':

    discard

The following object is masked from 'package:readr':

    col_factor
```

```r
library(knitr)

# Read the data
train_data <- read.csv("cell2celltrain.csv")
holdout_data <- read.csv("cell2cellholdout.csv")

# =============================================================================
# 1. DATA OVERVIEW & BASIC STATISTICS
# =============================================================================

cat("=== DATA OVERVIEW ===\n")
```

```
=== DATA OVERVIEW ===
```

```r
cat("Training data dimensions:", dim(train_data), "\n")
```

```
Training data dimensions: 51047 58
```

```r
cat("Holdout data dimensions:", dim(holdout_data), "\n\n")
```

```
Holdout data dimensions: 20000 58
```

```r
# Basic structure
cat("Training data structure:\n")
```

```
Training data structure:
```

```r
str(train_data)
```

```
'data.frame':   51047 obs. of  58 variables:
 $ CustomerID                : int  3000002 3000010 3000014 3000022 3000026 3000030
3000038 3000042 3000046 3000050 ...
 $ Churn                     : chr  "Yes" "Yes" "No" "No" ...
 $ MonthlyRevenue            : num  24 17 38 82.3 17.1 ...
 $ MonthlyMinutes            : int  219 10 8 1312 0 682 26 98 24 1056 ...
 $ TotalRecurringCharge      : int  22 17 38 75 17 52 30 66 35 75 ...
 $ DirectorAssistedCalls     : num  0.25 0 0 1.24 0 0.25 0.25 2.48 0 0 ...
 $ OverageMinutes            : int  0 0 0 0 0 0 0 0 0 0 ...
 $ RoamingCalls              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PercChangeMinutes         : int  -157 -4 -2 157 0 148 60 24 20 43 ...
 $ PercChangeRevenues        : num  -19 0 0 8.1 -0.2 -3.1 4 6.8 -0.3 2.4 ...
 $ DroppedCalls              : num  0.7 0.3 0 52 0 9 0 0 0 0 ...
 $ BlockedCalls              : num  0.7 0 0 7.7 0 1.7 1 0.3 0 0 ...
 $ UnansweredCalls           : num  6.3 2.7 0 76 0 13 2.3 4 1 0 ...
 $ CustomerCareCalls         : num  0 0 0 4.3 0 0.7 0 4 0 0 ...
 $ ThreewayCalls             : num  0 0 0 1.3 0 0 0 0 0 0 ...
 $ ReceivedCalls             : num  97.2 0 0.4 200.3 0 ...
 $ OutboundCalls             : num  0 0 0.3 370.3 0 ...
```

```
 $ InboundCalls             : num  0 0 0 147 0 0 0 0 1.7 0 ...
 $ PeakCallsInOut           : num  58 5 1.3 555.7 0 ...
 $ OffPeakCallsInOut        : num  24 1 3.7 303.7 0 ...
 $ DroppedBlockedCalls      : num  1.3 0.3 0 59.7 0 10.7 1 0.3 0 0 ...
 $ CallForwardingCalls      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ CallWaitingCalls         : num  0.3 0 0 22.7 0 0.7 0 0 0 0 ...
 $ MonthsInService          : int  61 58 60 59 53 53 57 59 53 55 ...
 $ UniqueSubs               : int  2 1 1 2 2 1 2 2 3 1 ...
 $ ActiveSubs               : int  1 1 1 2 2 1 2 2 3 1 ...
 $ ServiceArea              : chr  "SEAPOR503" "PITHOM412" "MILMIL414" "PITHOM412" ...
 $ Handsets                 : int  2 2 1 9 4 3 2 3 4 9 ...
 $ HandsetModels            : int  2 1 1 4 3 2 2 3 3 5 ...
 $ CurrentEquipmentDays     : int  361 1504 1812 458 852 231 601 464 544 388 ...
 $ AgeHH1                   : int  62 40 26 30 46 28 52 46 36 46 ...
 $ AgeHH2                   : int  0 42 26 0 54 0 58 46 34 68 ...
 $ ChildrenInHH             : chr  "No" "Yes" "Yes" "No" ...
 $ HandsetRefurbished       : chr  "No" "No" "No" "No" ...
 $ HandsetWebCapable        : chr  "Yes" "No" "No" "Yes" ...
 $ TruckOwner               : chr  "No" "No" "No" "No" ...
 $ RVOwner                  : chr  "No" "No" "No" "No" ...
 $ Homeownership            : chr  "Known" "Known" "Unknown" "Known" ...
 $ BuysViaMailOrder         : chr  "Yes" "Yes" "No" "Yes" ...
 $ RespondsToMailOffers     : chr  "Yes" "Yes" "No" "Yes" ...
 $ OptOutMailings           : chr  "No" "No" "No" "No" ...
 $ NonUSTravel              : chr  "No" "No" "No" "No" ...
 $ OwnsComputer             : chr  "Yes" "Yes" "No" "No" ...
 $ HasCreditCard            : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ RetentionCalls           : int  1 0 0 0 0 0 0 0 0 0 ...
 $ RetentionOffersAccepted  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ NewCellphoneUser         : chr  "No" "Yes" "Yes" "Yes" ...
 $ NotNewCellphoneUser      : chr  "No" "No" "No" "No" ...
 $ ReferralsMadeBySubscriber: int  0 0 0 0 0 0 0 0 0 0 ...
 $ IncomeGroup              : int  4 5 6 6 9 1 9 6 9 5 ...
 $ OwnsMotorcycle           : chr  "No" "No" "No" "No" ...
 $ AdjustmentsToCreditRating: int  0 0 0 0 1 1 1 0 0 1 ...
 $ HandsetPrice             : chr  "30" "30" "Unknown" "10" ...
 $ MadeCallToRetentionTeam  : chr  "Yes" "No" "No" "No" ...
 $ CreditRating             : chr  "1-Highest" "4-Medium" "3-Good" "4-Medium" ...
 $ PrizmCode                : chr  "Suburban" "Suburban" "Town" "Other" ...
 $ Occupation               : chr  "Professional" "Professional" "Crafts" "Other" ...
 $ MaritalStatus            : chr  "No" "Yes" "Yes" "No" ...
```

```r
cat("\n")
```

```r
# Check for missing values
cat("Missing values in training data:\n")
```

```
Missing values in training data:
```

```
missing_summary <- sapply(train_data, function(x) sum(is.na(x)))
print(missing_summary[missing_summary > 0])
```

```
      MonthlyRevenue        MonthlyMinutes  TotalRecurringCharge 
                 156                   156                   156 
DirectorAssistedCalls        OverageMinutes          RoamingCalls 
                 156                   156                   156 
   PercChangeMinutes    PercChangeRevenues              Handsets 
                 367                   367                     1 
       HandsetModels  CurrentEquipmentDays                AgeHH1 
                   1                     1                   909 
              AgeHH2 
                 909 
```

```
# Basic statistics for key numerical variables
key_vars <- c("MonthlyRevenue", "MonthlyMinutes", "CustomerCareCalls",
              "DroppedCalls", "BlockedCalls", "MonthsInService",
              "OverageMinutes", "PercChangeMinutes", "PercChangeRevenues")

cat("\nBasic statistics for key variables:\n")
```

```
Basic statistics for key variables:
```

```
print(summary(train_data[key_vars]))
```

```
 MonthlyRevenue    MonthlyMinutes   CustomerCareCalls  DroppedCalls    
 Min.   :  -6.17   Min.   :   0.0   Min.   :  0.000   Min.   :  0.000  
 1st Qu.:  33.61   1st Qu.: 158.0   1st Qu.:  0.000   1st Qu.:  0.700  
 Median :  48.46   Median : 366.0   Median :  0.000   Median :  3.000  
 Mean   :  58.83   Mean   : 525.7   Mean   :  1.869   Mean   :  6.011  
 3rd Qu.:  71.06   3rd Qu.: 723.0   3rd Qu.:  1.700   3rd Qu.:  7.700  
 Max.   :1223.38   Max.   :7359.0   Max.   :327.300   Max.   :221.700  
 NA's   :156       NA's   :156                                         
  BlockedCalls     MonthsInService OverageMinutes    PercChangeMinutes
 Min.   :  0.000   Min.   : 6.00   Min.   :   0.00   Min.   :-3875.00  
 1st Qu.:  0.000   1st Qu.:11.00   1st Qu.:   0.00   1st Qu.:  -83.00  
 Median :  1.000   Median :16.00   Median :   3.00   Median :   -5.00  
 Mean   :  4.086   Mean   :18.76   Mean   :  40.03   Mean   :  -11.55  
 3rd Qu.:  3.700   3rd Qu.:24.00   3rd Qu.:  41.00   3rd Qu.:   66.00  
 Max.   :384.300   Max.   :61.00   Max.   :4321.00   Max.   : 5192.00  
                                   NA's   :156       NA's   :367       
 PercChangeRevenues
 Min.   :-1107.700 
 1st Qu.:   -7.100 
 Median :   -0.300 
 Mean   :   -1.192 
 3rd Qu.:    1.600 
```

```
 Max.   : 2483.500
 NA's   :367
```

```r
# ============================================================================
# 2. TARGET VARIABLE ANALYSIS
# ============================================================================

# Churn distribution
churn_summary <- train_data %>%
  group_by(Churn) %>%
  summarise(
    Count = n(),
    Percentage = n() / nrow(train_data) * 100
  )

cat("\n=== CHURN DISTRIBUTION ===\n")
```

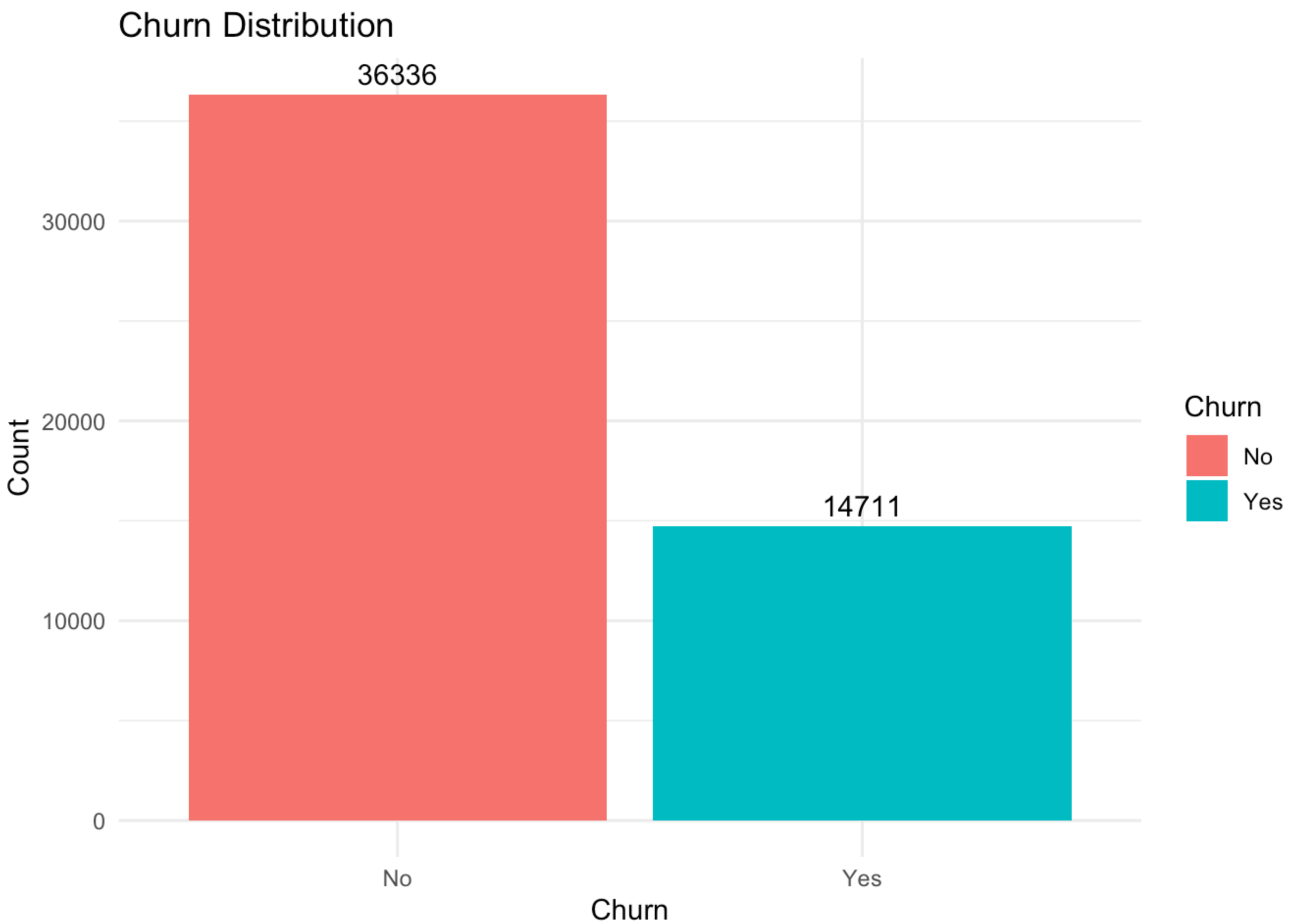
```
=== CHURN DISTRIBUTION ===
```

```r
print(churn_summary)
```

```
# A tibble: 2 × 3
  Churn Count Percentage
  <chr> <int>      <dbl>
1 No    36336       71.2
2 Yes   14711       28.8
```

```r
# Plot churn distribution
p1 <- ggplot(train_data, aes(x = Churn, fill = Churn)) +
  geom_bar() +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5) +
  labs(title = "Churn Distribution", x = "Churn", y = "Count") +
  theme_minimal()

print(p1)
```

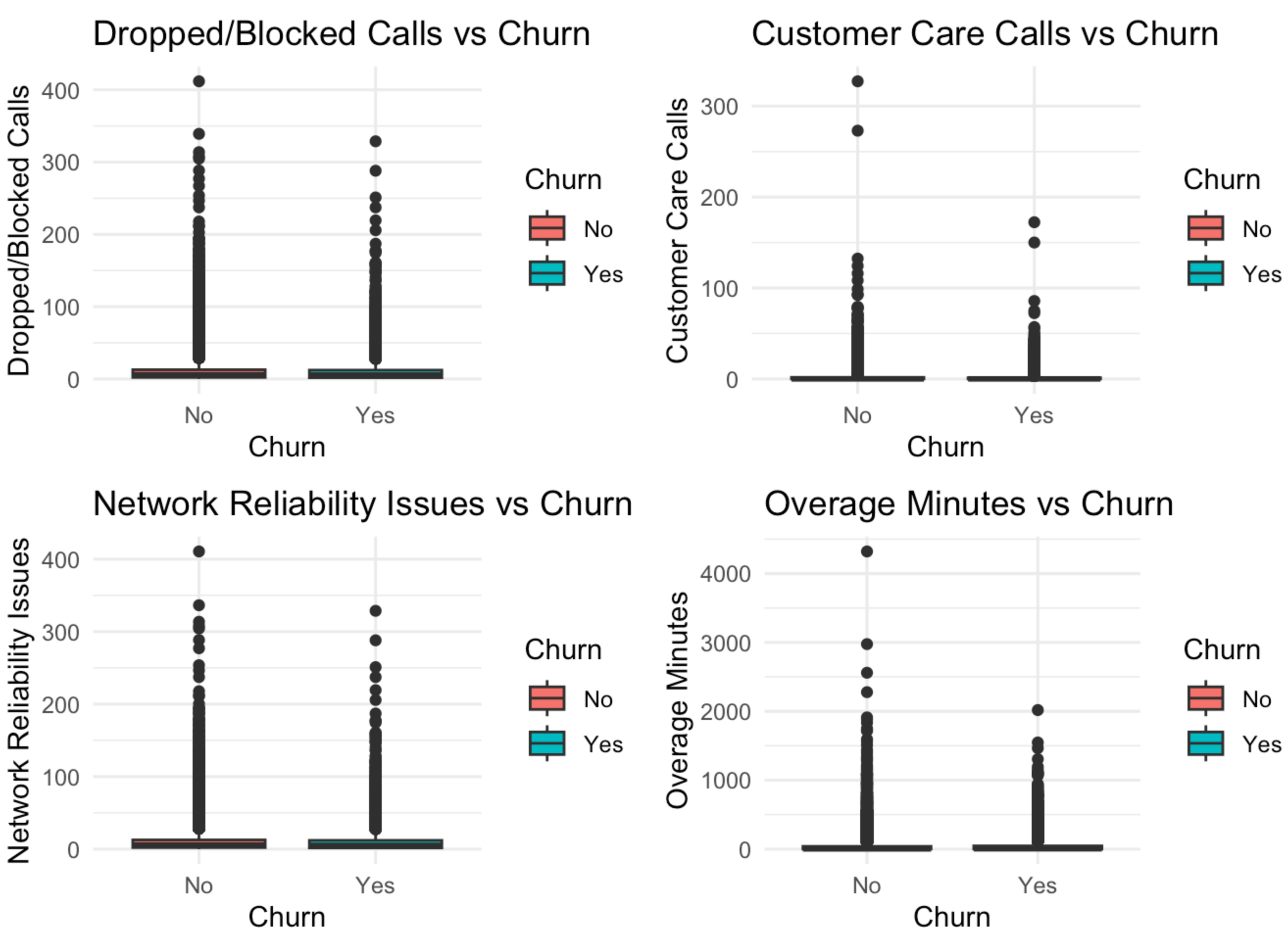
```
Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
ℹ Please use `after_stat(count)` instead.
```




```
# ============================================================================
# 3. SERVICE QUALITY FACTORS ANALYSIS
# ============================================================================

# Create composite network reliability metric
train_data <- train_data %>%
  mutate(
    NetworkReliability = DroppedCalls + BlockedCalls,
    CallQualityIssues = ifelse(DroppedBlockedCalls > median(DroppedBlockedCalls, na.rm =
    HighCareCalls = ifelse(CustomerCareCalls > median(CustomerCareCalls, na.rm = TRUE), "
  )

# Service quality vs churn
service_plots <- list()

# Dropped/Blocked Calls vs Churn
service_plots[[1]] <- ggplot(train_data, aes(x = Churn, y = DroppedBlockedCalls, fill = C
  geom_boxplot() +
  labs(title = "Dropped/Blocked Calls vs Churn", y = "Dropped/Blocked Calls") +
  theme_minimal()

# Customer Care Calls vs Churn
service_plots[[2]] <- ggplot(train_data, aes(x = Churn, y = CustomerCareCalls, fill = Chu
```

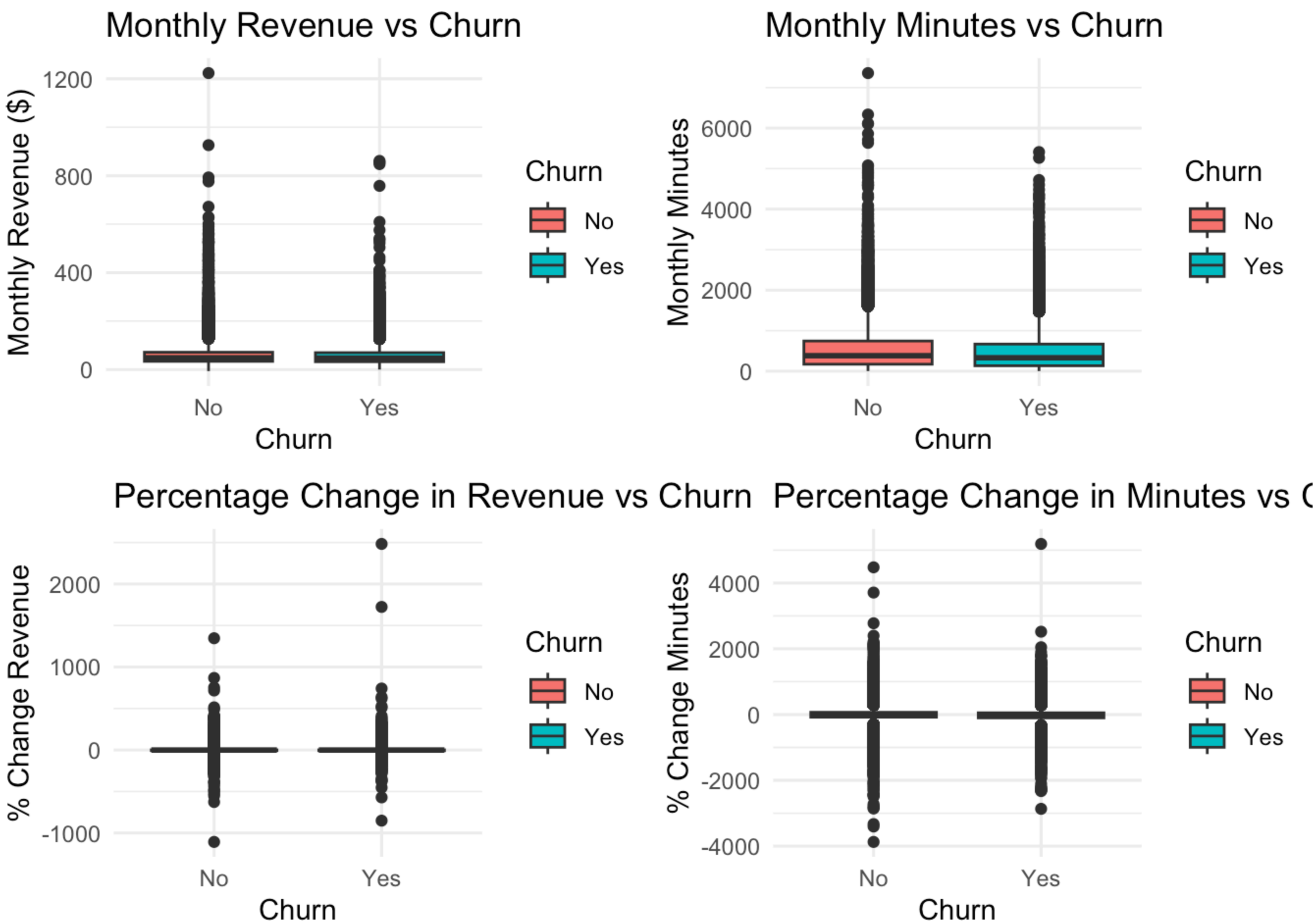
```r
  geom_boxplot() +
  labs(title = "Customer Care Calls vs Churn", y = "Customer Care Calls") +
  theme_minimal()

# Network Reliability vs Churn
service_plots[[3]] <- ggplot(train_data, aes(x = Churn, y = NetworkReliability, fill = Ch
  geom_boxplot() +
  labs(title = "Network Reliability Issues vs Churn", y = "Network Reliability Issues") +
  theme_minimal()

# Overage Minutes vs Churn
service_plots[[4]] <- ggplot(train_data, aes(x = Churn, y = OverageMinutes, fill = Churn)
  geom_boxplot() +
  labs(title = "Overage Minutes vs Churn", y = "Overage Minutes") +
  theme_minimal()

# Display service quality plots
grid.arrange(grobs = service_plots, ncol = 2)
```

```
Warning: Removed 156 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



```
# ============================================================================
# 4. REVENUE & USAGE ANALYSIS
# ============================================================================

revenue_plots <- list()

# Monthly Revenue vs Churn
revenue_plots[[1]] <- ggplot(train_data, aes(x = Churn, y = MonthlyRevenue, fill = Churn)
  geom_boxplot() +
  labs(title = "Monthly Revenue vs Churn", y = "Monthly Revenue ($)") +
  theme_minimal()

# Monthly Minutes vs Churn
revenue_plots[[2]] <- ggplot(train_data, aes(x = Churn, y = MonthlyMinutes, fill = Churn)
  geom_boxplot() +
  labs(title = "Monthly Minutes vs Churn", y = "Monthly Minutes") +
  theme_minimal()

# Percentage Change in Revenue vs Churn
revenue_plots[[3]] <- ggplot(train_data, aes(x = Churn, y = PercChangeRevenues, fill = Ch
  geom_boxplot() +
  labs(title = "Percentage Change in Revenue vs Churn", y = "% Change Revenue") +
  theme_minimal()

# Percentage Change in Minutes vs Churn
revenue_plots[[4]] <- ggplot(train_data, aes(x = Churn, y = PercChangeMinutes, fill = Chu
  geom_boxplot() +
  labs(title = "Percentage Change in Minutes vs Churn", y = "% Change Minutes") +
  theme_minimal()

# Display revenue plots
grid.arrange(grobs = revenue_plots, ncol = 2)
```

```
Warning: Removed 156 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 156 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 367 rows containing non-finite outside the scale range
(`stat_boxplot()`).
Removed 367 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```




```r
# ==========================================================================
# 5. CUSTOMER SEGMENTATION ANALYSIS
# ==========================================================================

# Create customer segments - FIXED CODE
# Calculate quantiles for revenue segmentation
revenue_quantiles <- quantile(train_data$MonthlyRevenue, probs = c(0, 0.33, 0.66, 1), na.
tenure_breaks <- c(0, 12, 36, max(train_data$MonthsInService, na.rm = TRUE))

train_data <- train_data %>%
  mutate(
    RevenueSegment = cut(MonthlyRevenue, breaks = revenue_quantiles,
                         labels = c("Low", "Medium", "High"), include.lowest = TRUE),
    TenureSegment = cut(MonthsInService, breaks = tenure_breaks,
                        labels = c("New", "Medium", "Long"), include.lowest = TRUE),
    HighValue = ifelse(MonthlyRevenue > median(MonthlyRevenue, na.rm = TRUE), "High", "Lo
  )

# Churn rate by segments
segment_analysis <- train_data %>%
  group_by(RevenueSegment) %>%
  summarise(
    Total_Customers = n(),
```

```
    Churn_Rate = sum(Churn == "Yes") / n() * 100
  )

cat("\n=== CHURN RATE BY REVENUE SEGMENT ===\n")
```

```
=== CHURN RATE BY REVENUE SEGMENT ===
```

```
print(segment_analysis)
```

```
# A tibble: 4 × 3
  RevenueSegment Total_Customers Churn_Rate
  <fct>                    <int>      <dbl>
1 Low                      16796       30.3
2 Medium                   16792       28.1
3 High                     17303       27.9
4 <NA>                       156       44.9
```

```
# Service issues by customer segments
segment_service <- train_data %>%
  group_by(HighValue) %>%
  summarise(
    Avg_DroppedCalls = mean(DroppedCalls, na.rm = TRUE),
    Avg_CareCalls = mean(CustomerCareCalls, na.rm = TRUE),
    Churn_Rate = sum(Churn == "Yes") / n() * 100
  )

cat("\n=== SERVICE ISSUES BY CUSTOMER VALUE ===\n")
```

```
=== SERVICE ISSUES BY CUSTOMER VALUE ===
```

```
print(segment_service)
```

```
# A tibble: 3 × 4
  HighValue Avg_DroppedCalls Avg_CareCalls Churn_Rate
  <chr>                <dbl>         <dbl>      <dbl>
1 High                  8.99          2.68       28.3
2 Low                   3.04          1.07       29.3
3 <NA>                  4.31         0.568       44.9
```

```
# ============================================================================
# 6. CORRELATION ANALYSIS
# ============================================================================

# Select numerical variables for correlation
numerical_vars <- train_data %>%
  select(where(is.numeric)) %>%
```

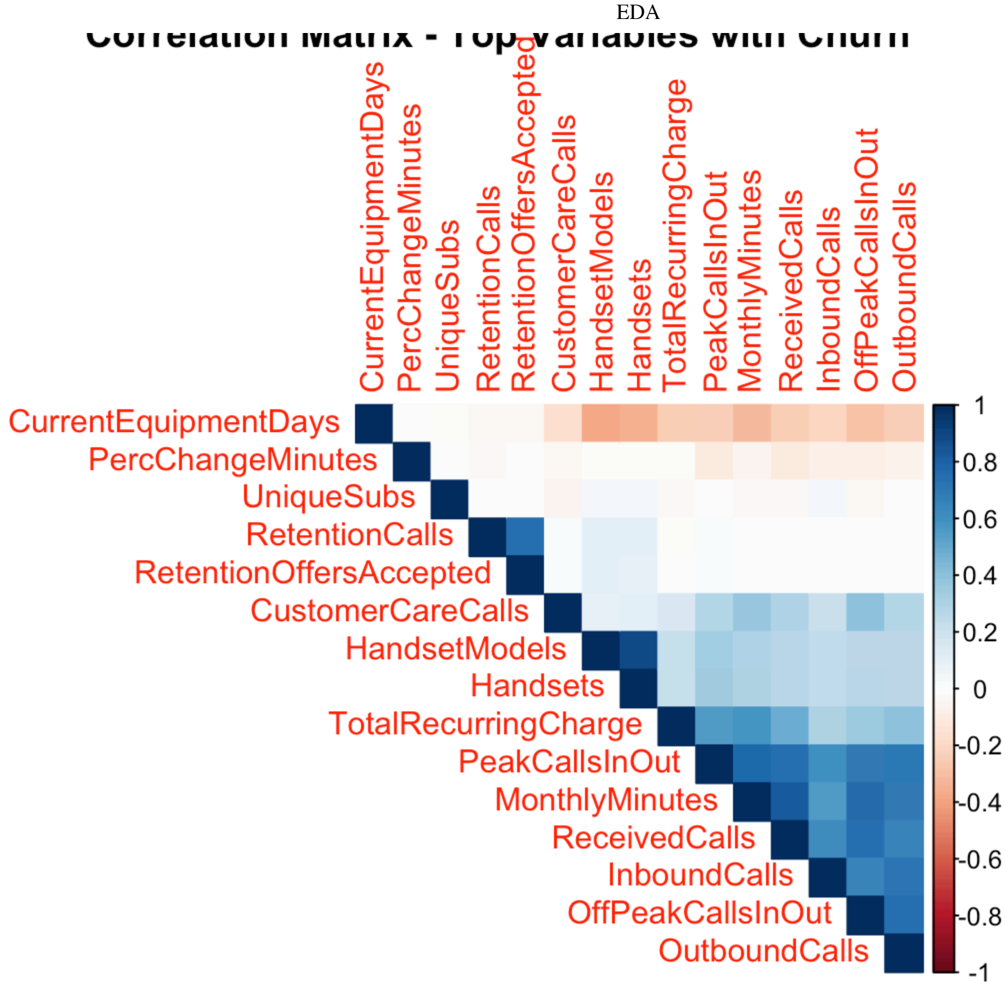
```r
  select(-CustomerID)  # Remove ID column

# Calculate correlation matrix (using only complete cases for simplicity)
cor_matrix <- cor(numerical_vars, use = "complete.obs")

# Focus on correlations with churn (convert Churn to numeric for correlation)
train_data_numeric <- train_data %>%
  mutate(Churn_Numeric = ifelse(Churn == "Yes", 1, 0))

# Get numerical variables including the new Churn_Numeric
numerical_vars_with_churn <- train_data_numeric %>%
  select(where(is.numeric)) %>%
  select(-CustomerID)

churn_correlations <- cor(numerical_vars_with_churn, use = "complete.obs") %>%
  as.data.frame() %>%
  select(Churn_Numeric) %>%
  arrange(desc(abs(Churn_Numeric)))

cat("\n=== TOP CORRELATIONS WITH CHURN ===\n")
```

```
=== TOP CORRELATIONS WITH CHURN ===
```

```r
print(head(churn_correlations, 15))
```

```
                        Churn_Numeric
Churn_Numeric              1.00000000
CurrentEquipmentDays       0.10247171
TotalRecurringCharge      -0.05908656
RetentionCalls             0.05850470
MonthlyMinutes            -0.05002993
OffPeakCallsInOut         -0.04077639
HandsetModels             -0.04053673
PeakCallsInOut            -0.03946775
ReceivedCalls             -0.03708503
CustomerCareCalls         -0.03481686
RetentionOffersAccepted    0.03467547
Handsets                  -0.03384803
InboundCalls              -0.03378622
PercChangeMinutes         -0.03302091
UniqueSubs                 0.03294253
```

```r
# Correlation plot (top 15 variables with churn)
top_vars <- rownames(churn_correlations)[2:16]  # Skip Churn_Numeric itself
cor_top <- cor(numerical_vars_with_churn[top_vars], use = "complete.obs")
corrplot(cor_top, method = "color", type = "upper", order = "hclust",
         title = "Correlation Matrix - Top Variables with Churn")
```




```
# ============================================================================
# 7. INTERACTION EFFECTS ANALYSIS (For Pathway 2)
# ============================================================================

# Analyze how service issues affect different customer segments
interaction_analysis <- train_data %>%
  group_by(HighValue, CallQualityIssues) %>%
  summarise(
    Count = n(),
    Churn_Rate = sum(Churn == "Yes") / n() * 100,
    .groups = 'drop'
  )

cat("\n=== INTERACTION: CUSTOMER VALUE × CALL QUALITY ===\n")
```

```
=== INTERACTION: CUSTOMER VALUE × CALL QUALITY ===
```

```
print(interaction_analysis)
```

```
# A tibble: 6 × 4
  HighValue CallQualityIssues Count Churn_Rate
```

```
  <chr>     <chr>             <int>      <dbl>
1 High      High              17786       27.8
2 High      Low                7656       29.5
3 Low       High               7606       28.8
4 Low       Low               17843       29.5
5 <NA>      High                 38       26.3
6 <NA>      Low                 118       50.8
```

```r
# Customer care calls impact by tenure
care_calls_impact <- train_data %>%
  mutate(Tenure_Group = ifelse(MonthsInService <= 12, "New",
                               ifelse(MonthsInService <= 36, "Medium", "Long"))) %>%
  group_by(Tenure_Group, HighCareCalls) %>%
  summarise(
    Count = n(),
    Churn_Rate = sum(Churn == "Yes") / n() * 100,
    .groups = 'drop'
  )

cat("\n=== INTERACTION: TENURE × CUSTOMER CARE CALLS ===\n")
```

```
=== INTERACTION: TENURE × CUSTOMER CARE CALLS ===
```

```r
print(care_calls_impact)
```

```
# A tibble: 6 × 4
  Tenure_Group HighCareCalls Count Churn_Rate
  <chr>        <chr>         <int>      <dbl>
1 Long         High           1026       22.6
2 Long         Low            1894       29.9
3 Medium       High          13257       27.8
4 Medium       Low           17895       31.9
5 New          High           8744       24.9
6 New          Low            8231       28.4
```

```r
# ============================================================================
# 8. KEY INSIGHTS SUMMARY
# ============================================================================

cat("\n=== KEY EDA INSIGHTS FOR PATHWAY 2 ===\n")
```

```
=== KEY EDA INSIGHTS FOR PATHWAY 2 ===
```

```r
# Calculate key metrics
overall_churn_rate <- mean(train_data$Churn == "Yes") * 100
high_care_churn <- train_data %>%
  filter(HighCareCalls == "High") %>%
```

```r
  summarise(Churn_Rate = mean(Churn == "Yes") * 100) %>% pull(Churn_Rate)
high_issues_churn <- train_data %>%
  filter(CallQualityIssues == "High") %>%
  summarise(Churn_Rate = mean(Churn == "Yes") * 100) %>% pull(Churn_Rate)

cat(sprintf("Overall churn rate: %.1f%%\n", overall_churn_rate))
```

```
Overall churn rate: 28.8%
```

```r
cat(sprintf("Churn rate for high customer care calls: %.1f%%\n", high_care_churn))
```

```
Churn rate for high customer care calls: 26.5%
```

```r
cat(sprintf("Churn rate for high call quality issues: %.1f%%\n", high_issues_churn))
```

```
Churn rate for high call quality issues: 28.0%
```

```r
# Final summary
cat("\n=== INITIAL RECOMMENDATIONS FOR PATHWAY 2 ===\n")
```

```

=== INITIAL RECOMMENDATIONS FOR PATHWAY 2 ===
```

```r
cat("1. Focus on network reliability improvements\n")
```

```
1. Focus on network reliability improvements
```

```r
cat("2. Enhance customer service efficiency\n")
```

```
2. Enhance customer service efficiency
```

```r
cat("3. Monitor high-value customers with service issues\n")
```

```
3. Monitor high-value customers with service issues
```

```r
cat("4. Implement early warning systems for new customers\n")
```

```
4. Implement early warning systems for new customers
```

```r
cat("5. Develop targeted retention for different customer segments\n")
```

```
5. Develop targeted retention for different customer segments
```

```r
# Additional visualizations for service factors
# Plot interaction effects
p_interaction1 <- ggplot(interaction_analysis,
```

```
                          aes(x = HighValue, y = Churn_Rate, fill = CallQualityIssues)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Churn Rate: Customer Value × Call Quality",
       x = "Customer Value", y = "Churn Rate (%)") +
  theme_minimal()

p_interaction2 <- ggplot(care_calls_impact,
                          aes(x = Tenure_Group, y = Churn_Rate, fill = HighCareCalls)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Churn Rate: Tenure × Customer Care Calls",
       x = "Tenure Group", y = "Churn Rate (%)") +
  theme_minimal()

grid.arrange(p_interaction1, p_interaction2, ncol = 2)
```