Churn Data Exploration

Andres Perez

Introduction

This report explores the churn dataset. We will look at what churn means, how many customers have churned vs. not churned, and explore the predictors with summary statistics and visualizations. The goal is to understand the data before building any models.

Data Dictionary

Below is a brief description of each variable in the dataset:

- **ID**: Unique customer identifier
- **Customer.Months**: Number of months the customer has been with the company
- **Churn**: 1 if the customer churned, 0 otherwise
- **CHI.Score.Mon0**: Customer Health Index score (current month)
- **CHI.Score**: Customer Health Index score (previous month)
- **Support.Cases.Mon0**: Support cases (current month)
- **Support.Cases**: Support cases (previous month)
- **SP.Mon0**: Service points (current month)
- **SP**: Service points (previous month)
- **Logins**: Number of logins
- **Blog.Articles**: Number of blog articles read
- **Views**: Number of views
- **Days.Since.Last.Login**: Days since last login

Understanding Temporal Predictors and Negative Values

The dataset contains two versions of several predictors: one for the current month (marked with .Mon0) and one for the previous month. This temporal structure allows us to analyze how customer behavior changes over time. Here's what you need to know:

1. **Temporal Structure**:
    - Variables with .Mon0 suffix (e.g., CHI.Score.Mon0) represent the current month's values
    - Variables without .Mon0 (e.g., CHI.Score) represent the previous month's values
    - This structure helps us understand how customer behavior and metrics change over time
2. **Negative Values**:
    - Negative values in the non-Mon0 variables (previous month) represent changes or differences from the current month
    - For example:
        - A negative CHI.Score indicates the customer's health index was lower in the previous month
        - A negative Support.Cases means there were fewer support cases in the previous month
        - A negative Days.Since.Last.Login suggests the customer logged in more recently in the previous month
    - These negative values are not errors but rather meaningful indicators of temporal changes in customer behavior
3. **Why This Matters**:
    - The presence of both current and previous month values allows us to:
        - Track changes in customer behavior over time
        - Identify patterns that might predict churn
        - Understand how customer engagement evolves
    - The negative values help us quantify these changes and their direction

Load the Data

```
data <- read_csv('data/ChurnData.csv')
```

```
## Rows: 5713 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (13): ID, Customer.Months, Churn, CHI.Score.Mon0, CHI.Score, Support.Cas...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(data)
```

```
## Rows: 5,713
## Columns: 13
## $ ID                    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1…
## $ Customer.Months       <dbl> 67, 67, 55, 63, 57, 58, 57, 46, 56, 56, 53, 56, …
## $ Churn                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ CHI.Score.Mon0        <dbl> 0, 62, 0, 231, 43, 138, 180, 116, 78, 78, 91, 40…
## $ CHI.Score             <dbl> 0, 4, 0, 1, -1, -10, -5, -11, -7, -37, -1, 14, 1…
## $ Support.Cases.Mon0    <dbl> 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, …
## $ Support.Cases         <dbl> 0, 0, 0, -1, 0, 0, 1, 0, -2, 0, 0, 0, 0, 0, 0, 0…
## $ SP.Mon0               <dbl> 0, 0, 0, 3, 0, 0, 3, 0, 3, 0, 0, 0, 0, 0, 0, 0, …
## $ SP                    <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 3.0, 0.0, 0.0, 0.0…
## $ Logins                <dbl> 0, 0, 0, 167, 0, 43, 13, 0, -9, -7, 14, 0, 71, 0…
## $ Blog.Articles         <dbl> 0, 0, 0, -8, 0, 0, -1, 0, 1, 0, 3, 0, 9, 0, 1, 0…
## $ Views                 <dbl> 0, -16, 0, 21996, 9, -33, 907, 38, 0, 30, 0, 15,…
## $ Days.Since.Last.Login <dbl> 31, 31, 31, 0, 31, 0, 0, 6, 7, 14, 0, 31, 0, 31,…
```

Data Quality Checks

```
# Check for missing values
missing_summary <- sapply(data, function(x) sum(is.na(x)))
print("Missing values per column:")
```

```
## [1] "Missing values per column:"
```

```
print(missing_summary)
```

```
##                     ID       Customer.Months                 Churn
##                      0                     0                     0
##        CHI.Score.Mon0             CHI.Score    Support.Cases.Mon0
##                      0                     0                     0
##         Support.Cases               SP.Mon0                    SP
##                      0                     0                     0
##                Logins         Blog.Articles                 Views
##                      0                     0                     0
## Days.Since.Last.Login
##                      0
```

```
# Check for duplicate IDs
num_duplicates <- sum(duplicated(data$ID))
cat("Number of duplicate IDs:", num_duplicates, "\n")
```

```
## Number of duplicate IDs: 0
```

Note: There are no missing values or duplicate IDs in the dataset. This confirms the data is clean and ready for analysis. No imputation or removal was performed; this step is only to demonstrate that data quality was checked.

Churn Distribution

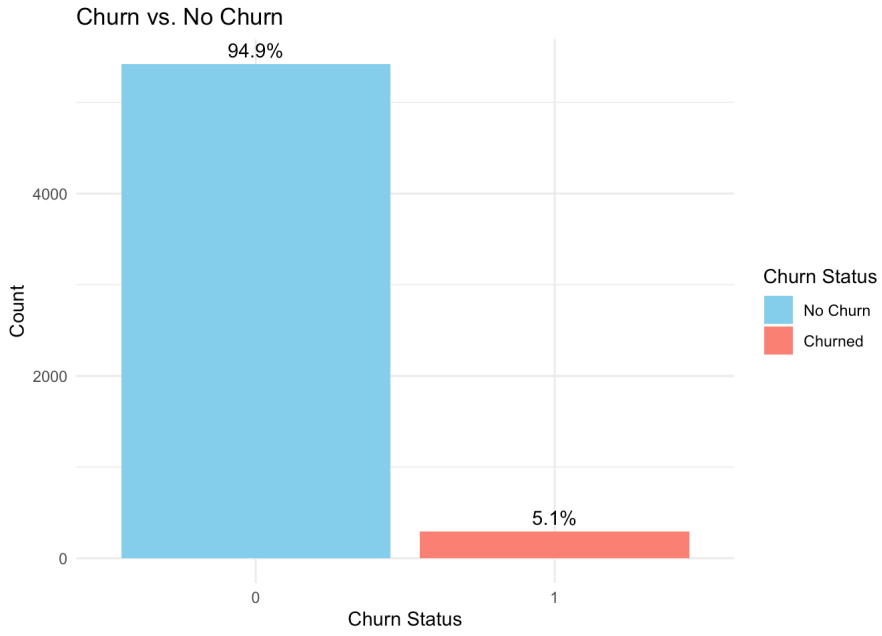
Let's see how many customers have churned vs. not churned, and the churn rate.

```
churn_table <- data %>% count(Churn) %>% mutate(Percent = n / sum(n) * 100)
kable(churn_table, caption = "Churn Counts and Percentages")
```

Churn Counts and Percentages

Churn	n	Percent
0	5422	94.906354
1	291	5.093646

```
ggplot(churn_table, aes(x = factor(Churn), y = n, fill = factor(Churn))) +
  geom_bar(stat = 'identity') +
  geom_text(aes(label = paste0(round(Percent,1), "%")), vjust = -0.5) +
  scale_fill_manual(values = c('0' = 'skyblue', '1' = 'salmon'),
                    labels = c('0' = 'No Churn', '1' = 'Churned')) +
  labs(title = 'Churn vs. No Churn',
       x = 'Churn Status',
       y = 'Count',
       fill = 'Churn Status') +
  theme_minimal()
```



Interpretation: - This shows how many customers have churned and how many have stayed, as well as the churn rate.

Univariate Analysis

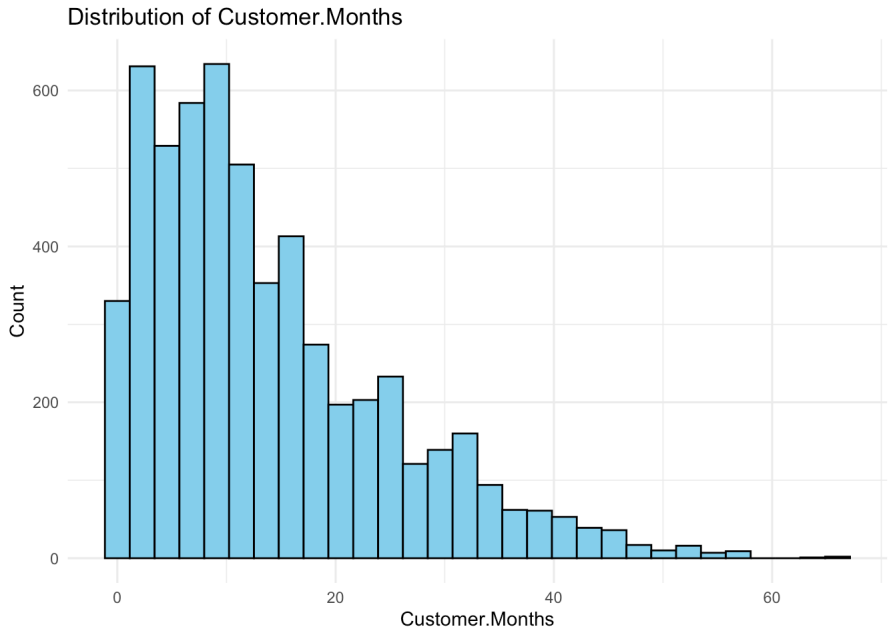
Numeric Predictors

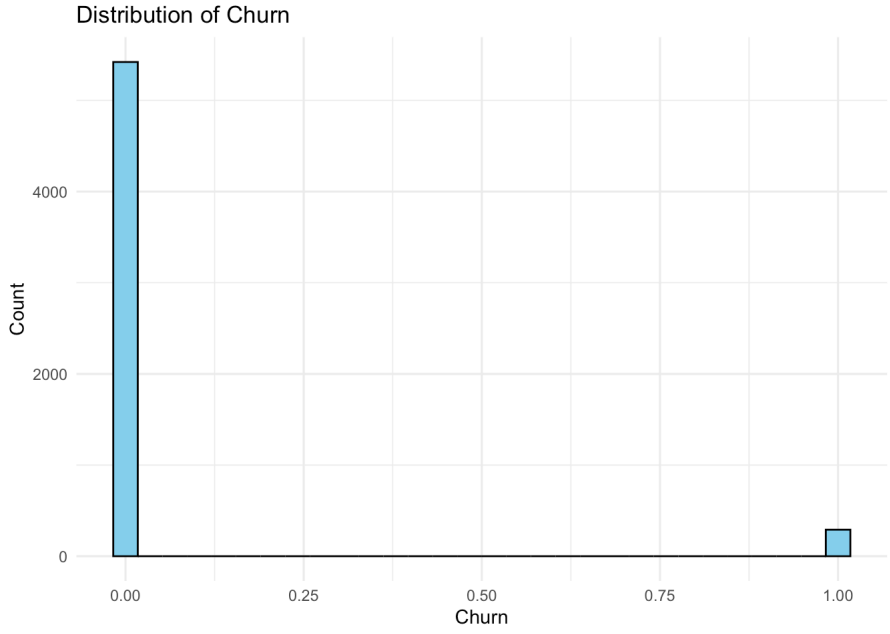
```
df_num <- data %>% select(where(is.numeric), -ID)
summary_stats <- df_num %>% summary()
kable(summary_stats, caption = "Summary Statistics for Numeric Predictors")
```

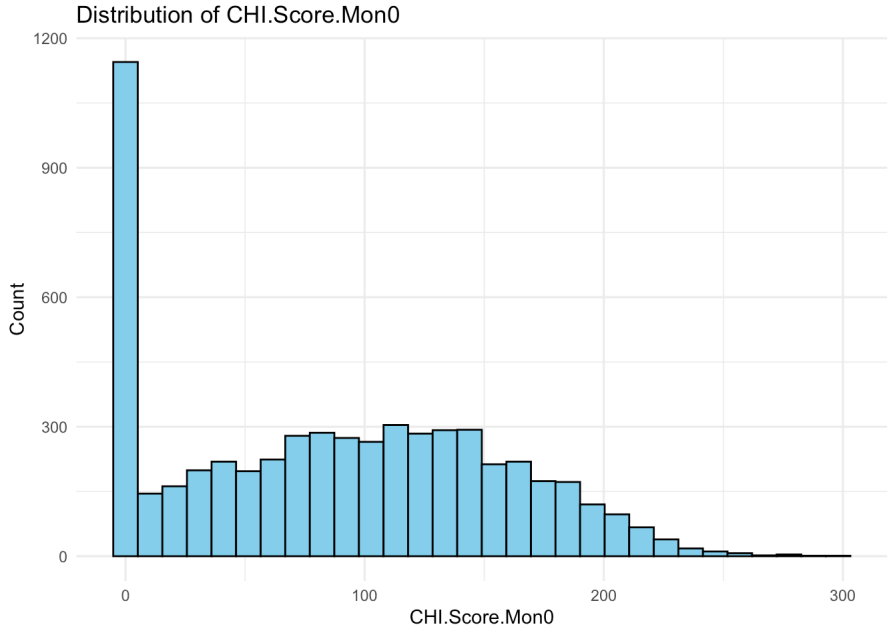
Summary Statistics for Numeric Predictors

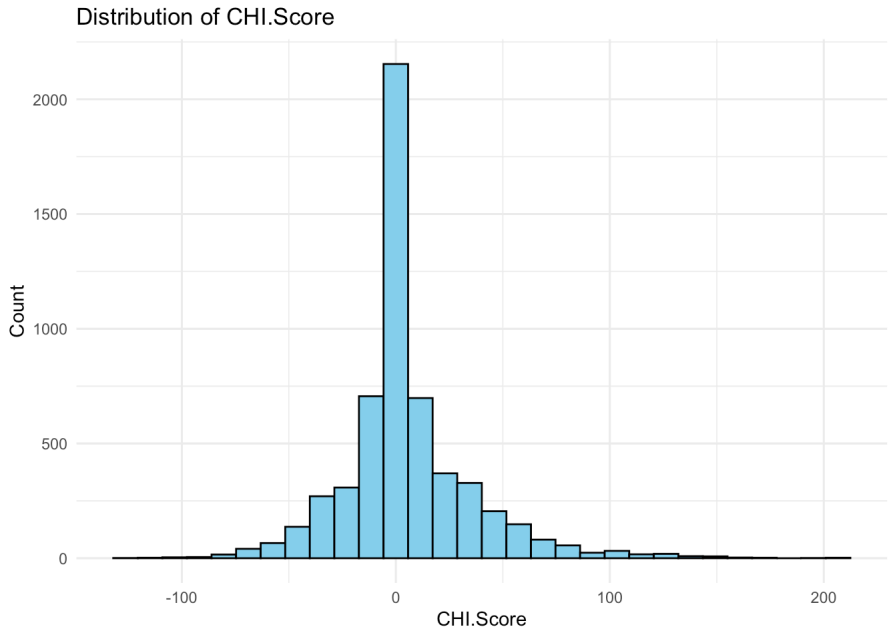
Customer.Months	Churn	CHI.Score.Mon0	CHI.Score	Support.Cases.Mon0	Support.Cases	SP.Mon0	SP	Logins	Blog.Articles	Vie
Min. : 1.00	Min. :0.00000	Min. : 0.00	Min. :-125.00	Min. : 0.0000	Min. :-17.00000	Min. :0.0000	Min. :-4.00000	Min. :-293.00	Min. :-75.0000	Mir :-2
1st Qu.: 5.00	1st Qu.:0.00000	1st Qu.: 25.00	1st Qu.: -8.00	1st Qu.: 0.0000	1st Qu.: 0.00000	1st Qu.:0.0000	1st Qu.: 0.00000	1st Qu.: -1.00	1st Qu.: 0.0000	1st -12
Median :11.00	Median :0.00000	Median : 88.00	Median : 0.00	Median : 0.0000	Median : 0.00000	Median :0.0000	Median : 0.00000	Median : 2.00	Median : 0.0000	Me 0.0
Mean :13.91	Mean :0.05094	Mean : 87.45	Mean : 5.06	Mean : 0.7098	Mean : -0.01243	Mean :0.8123	Mean : 0.02592	Mean : 15.68	Mean : 0.1679	Me 96.
3rd Qu.:20.00	3rd Qu.:0.00000	3rd Qu.:139.00	3rd Qu.: 15.00	3rd Qu.: 1.0000	3rd Qu.: 0.00000	3rd Qu.:2.6667	3rd Qu.: 0.00000	3rd Qu.: 23.00	3rd Qu.: 0.0000	3rd 27.
Max. :67.00	Max. :1.00000	Max. :298.00	Max. : 208.00	Max. :32.0000	Max. : 31.00000	Max. :4.0000	Max. : 4.00000	Max. : 865.00	Max. :217.0000	Ma :23

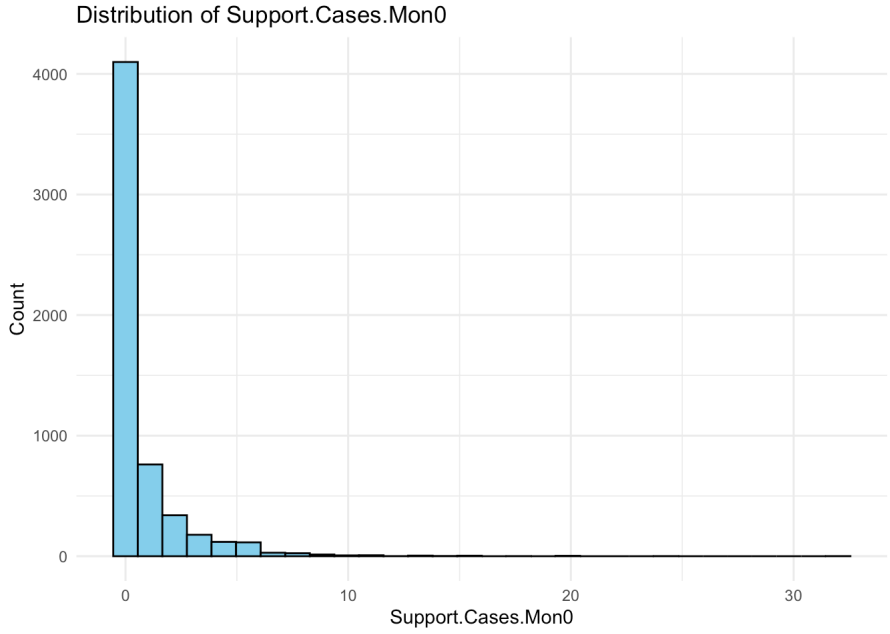
```
# Histograms
for (col in names(df_num)) {
  print(
    ggplot(data, aes_string(x = col)) +
      geom_histogram(bins = 30, fill = 'skyblue', color = 'black') +
      labs(title = paste('Distribution of', col), x = col, y = 'Count') +
      theme_minimal()
  )
}
```

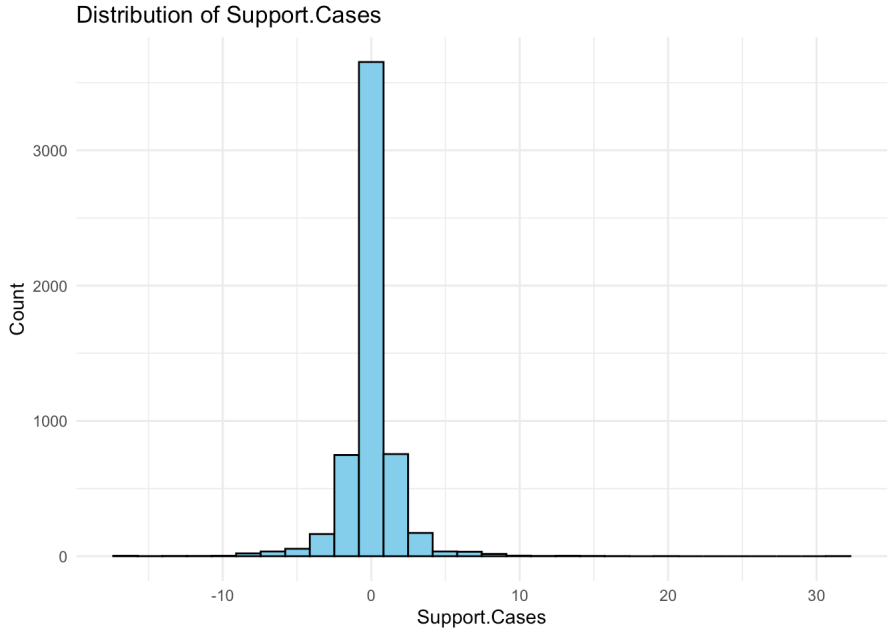

Distribution of Customer.Months
Count
600
400
200
0
0
20
40
60
Customer.Months

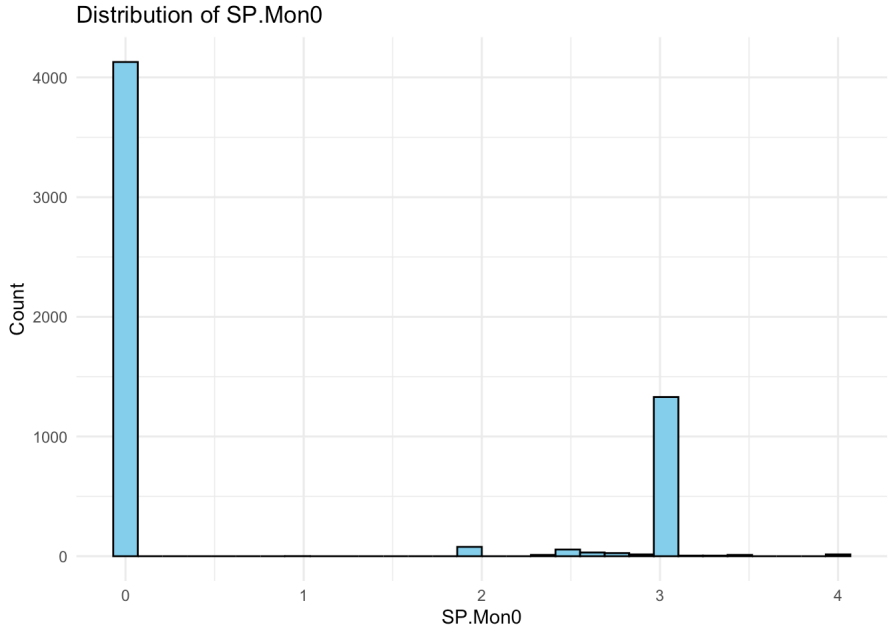

Distribution of Churn
Count
4000
2000
0
0.00
0.25
0.50
0.75
1.00
Churn

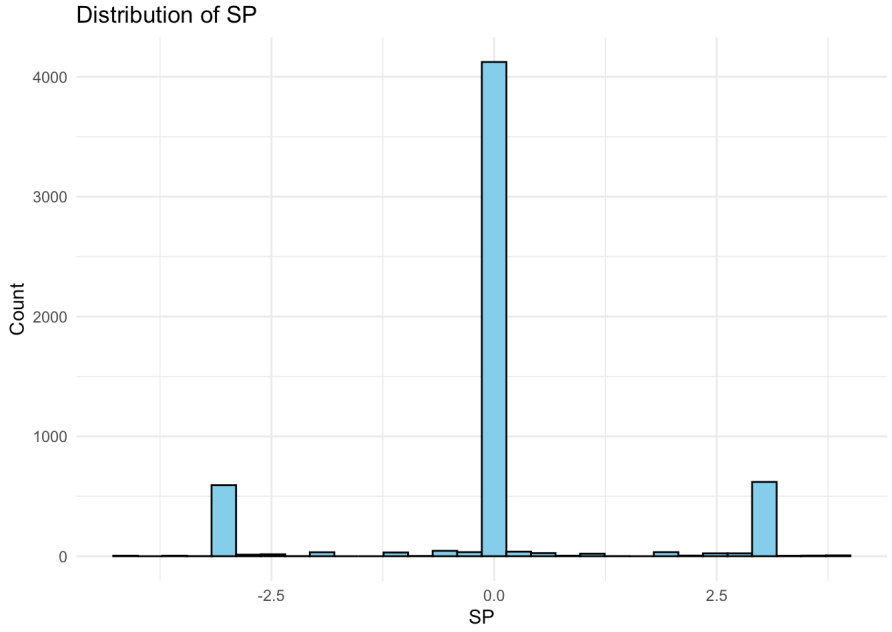

Distribution of CHI.Score.Mon0
Count
1200
900
600
300
0
0
100
200
300
CHI.Score.Mon0

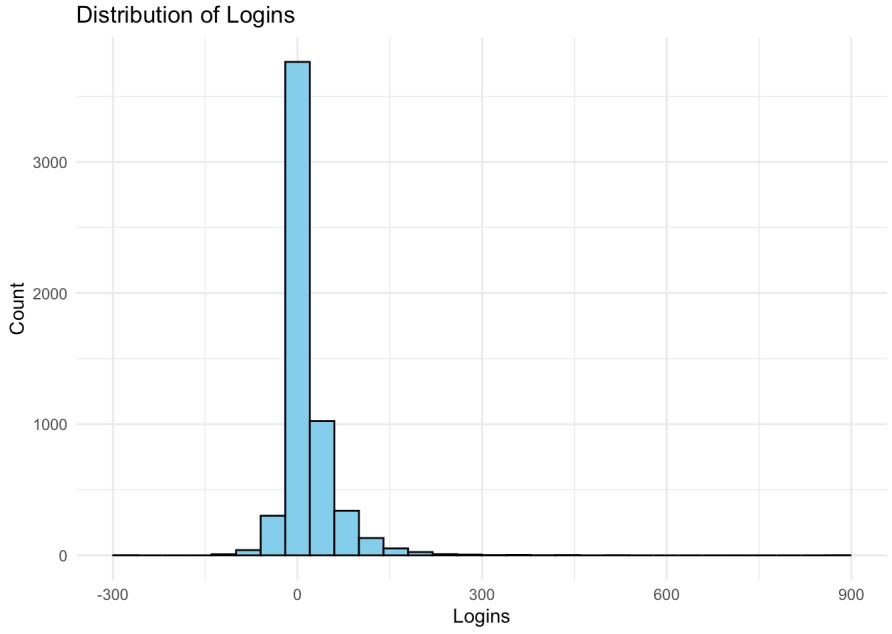

Distribution of CHI.Score
Count
2000
1500
1000
500
0
-100
0
100
200
CHI.Score

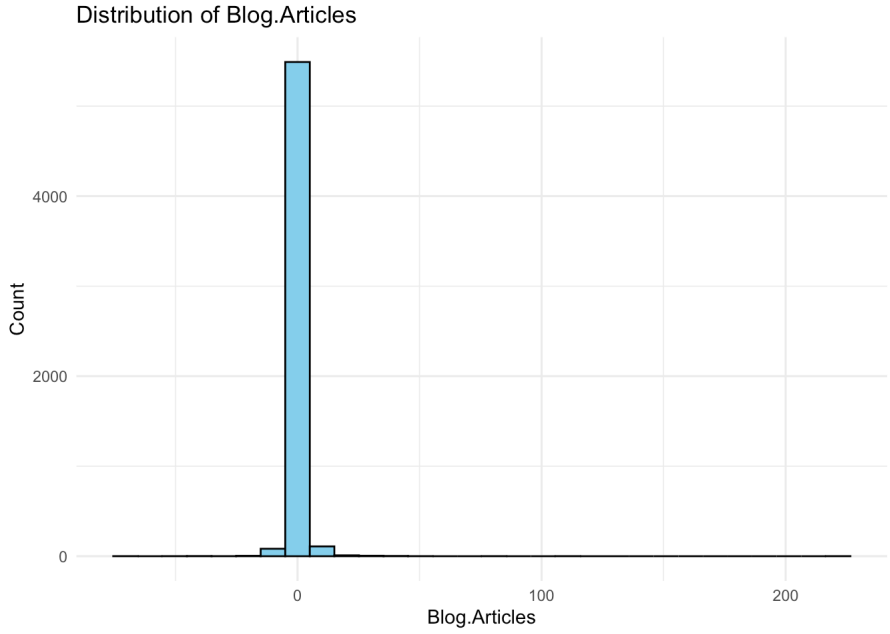

Distribution of Support.Cases.Mon0
Count
4000
3000
2000
1000
0
0
10
20
30
Support.Cases.Mon0

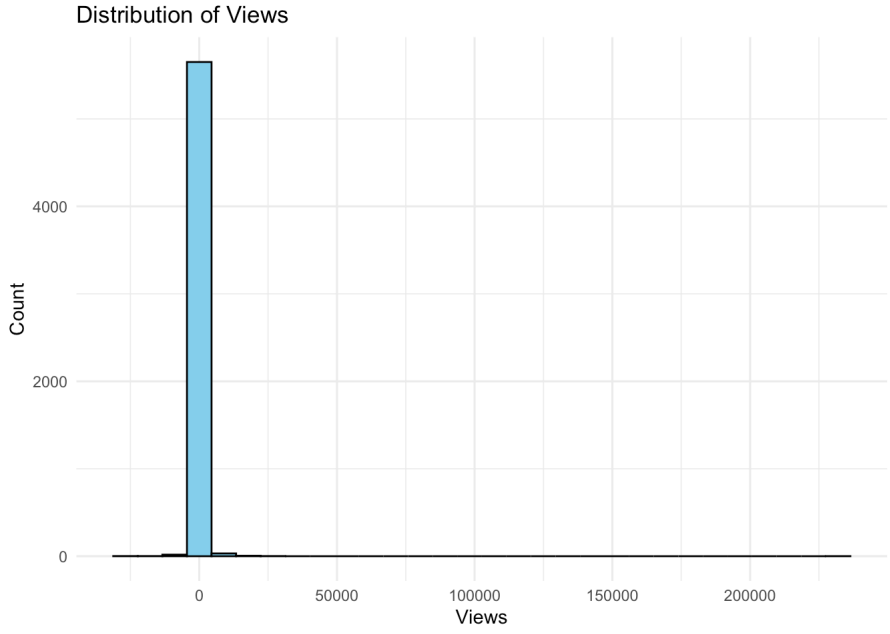

Distribution of Support.Cases
Count
3000
2000
1000
0
-10
0
10
20
30
Support.Cases

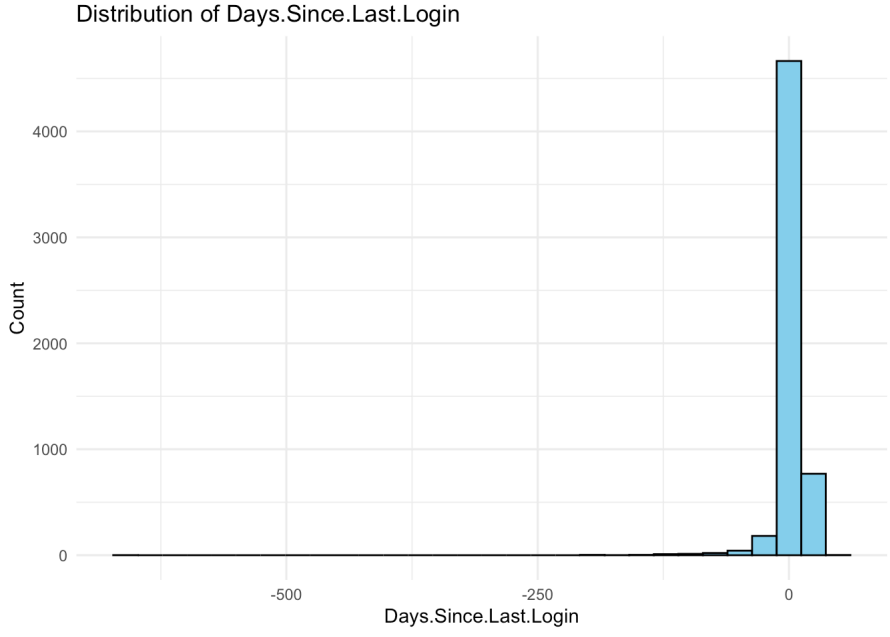

Distribution of SP.Mon0
Count
4000
3000
2000
1000
0
0
1
2
3
4
SP.Mon0


Distribution of SP
Count
4000
3000
2000
1000
0
-2.5
0.0
2.5
SP


Distribution of Logins
Count
3000
2000
1000
0
-300
0
300
600
900
Logins


Distribution of Blog.Articles
Count
4000
2000
0
0
100
200
Blog.Articles


Distribution of Views
Count
4000
2000
0
0
50000
100000
150000
200000
Views


Distribution of Days.Since.Last.Login
Count
4000
3000
2000
1000
0
-500
-250
0
Days.Since.Last.Login

Categorical Predictors

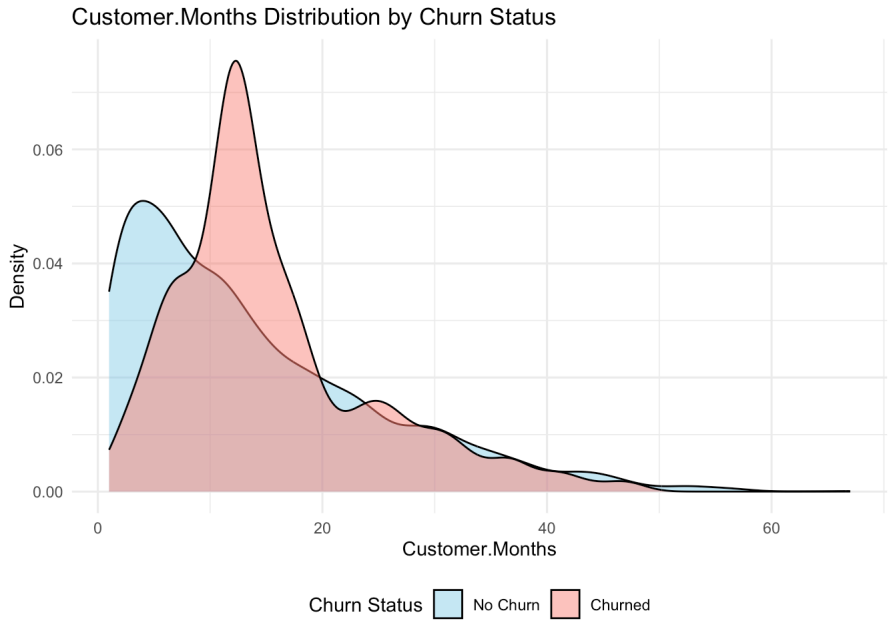
```
df_cat <- data %>% select(where(~!is.numeric(.)), -ID)
if (ncol(df_cat) > 0) {
  for (col in names(df_cat)) {
    print(
      ggplot(data, aes_string(x = col, fill = col)) +
        geom_bar() +
        labs(title = paste('Distribution of', col), x = col, y = 'Count') +
        theme_minimal() +
        theme(legend.position = 'none')
    )
  }
} else {
  print("No categorical predictors other than Churn.")
}
```

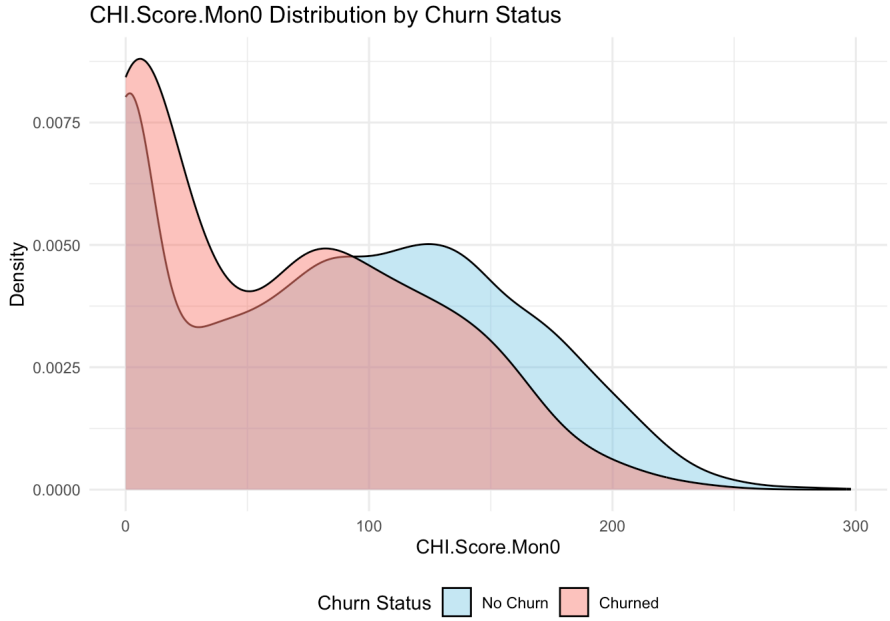
```
## [1] "No categorical predictors other than Churn."
```

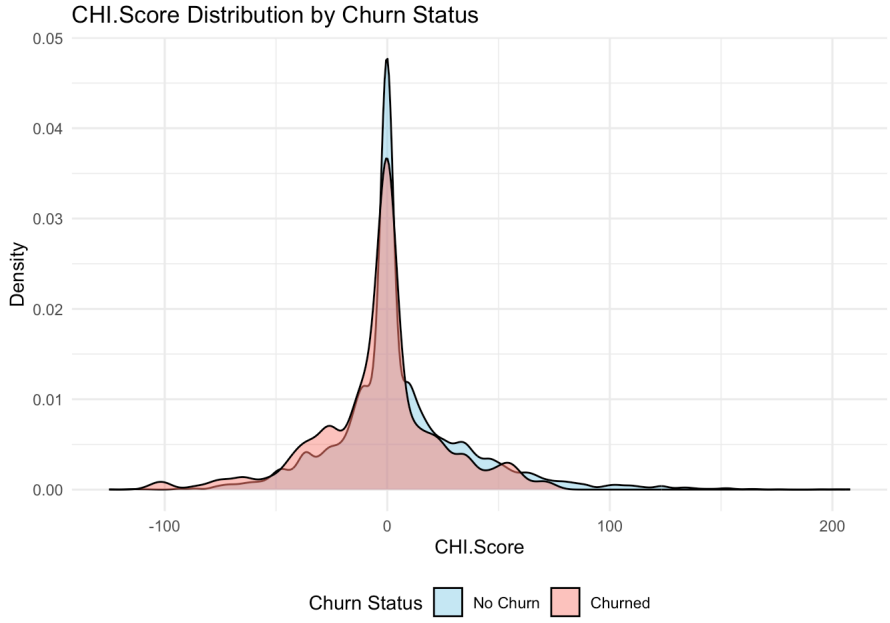
Bivariate Analysis: Predictors vs. Churn

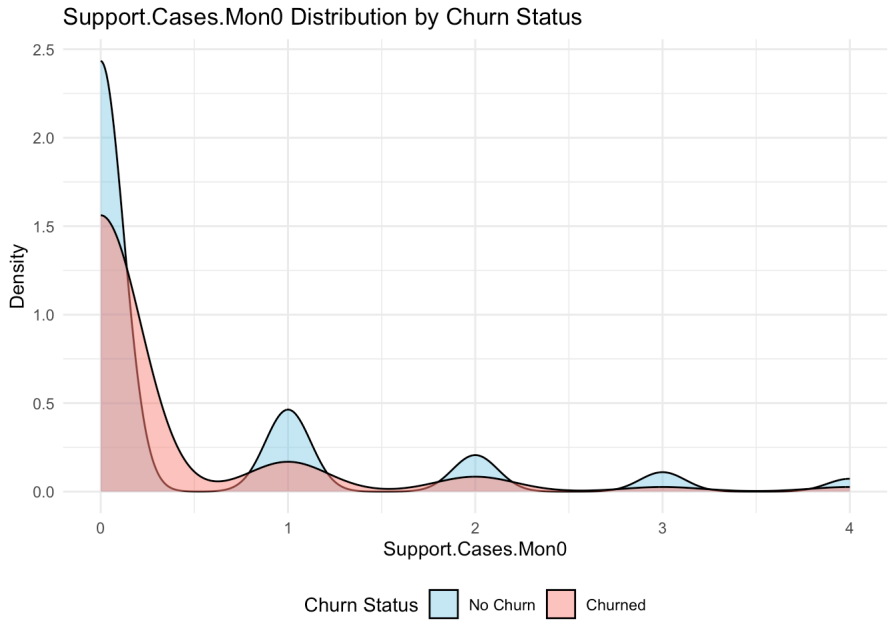
Numeric Predictors vs. Churn

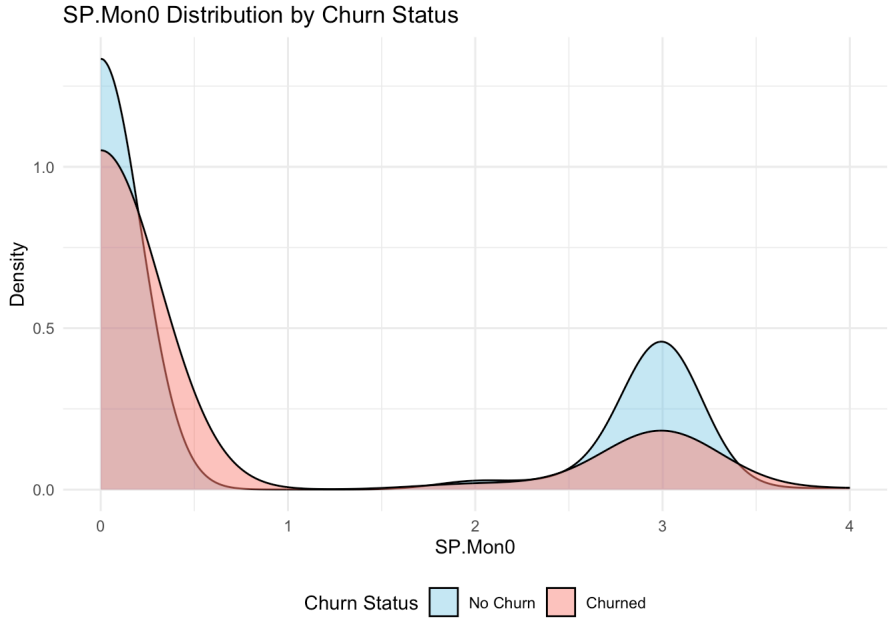
```
# Function to remove outliers (less aggressive: 3*IQR)
remove_outliers <- function(x) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm=TRUE)
  H <- 3 * IQR(x, na.rm=TRUE)
  x[x < (qnt[1] - H) | x > (qnt[2] + H)] <- NA
  return(x)
}

# Create a copy of data for plotting
plot_data <- data

# Remove outliers for specific variables
outlier_vars <- c('Support.Cases.Mon0', 'Support.Cases', 'Logins',
                  'Blog.Articles', 'Views', 'Days.Since.Last.Login')
plot_data[outlier_vars] <- lapply(plot_data[outlier_vars], remove_outliers)

# Create density plots, skipping 'Churn', 'Support.Cases', and 'Blog.Articles'
skip_vars <- c('Churn', 'Support.Cases', 'Blog.Articles')
for (col in names(df_num)) {
  if (col %in% skip_vars) next
  print(
    ggplot(plot_data, aes_string(x = col, fill = 'factor(Churn)')) +
      geom_density(alpha = 0.5) +
      labs(title = paste(col, 'Distribution by Churn Status'),
           x = col,
           y = 'Density',
           fill = 'Churn Status') +
      scale_fill_manual(values = c('0' = 'skyblue', '1' = 'salmon'),
                        labels = c('0' = 'No Churn', '1' = 'Churned')) +
      theme_minimal() +
      theme(legend.position = 'bottom')
  )
}
```

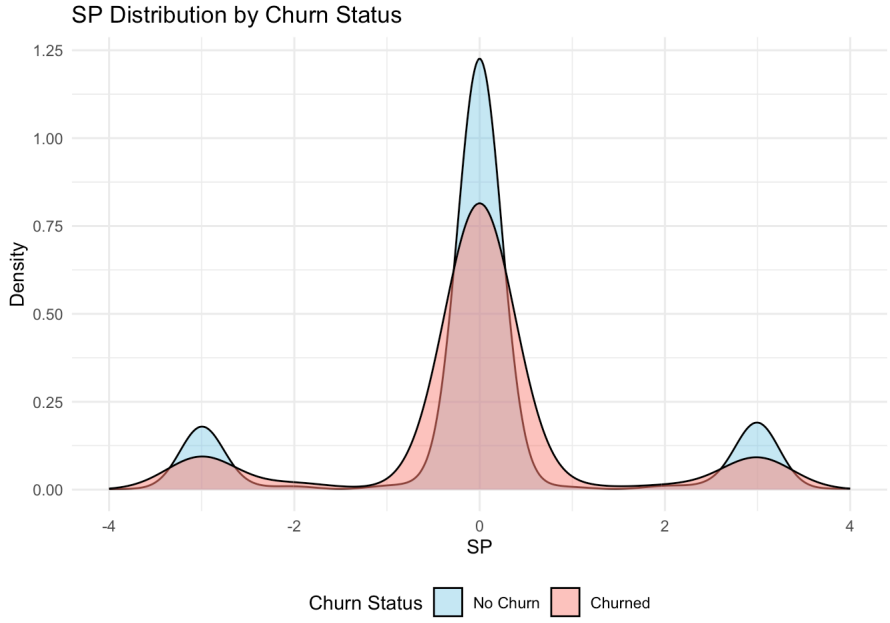

Customer.Months Distribution by Churn Status
Density
0.06
0.04
0.02
0.00
0
20
40
60
Customer.Months
Churn Status
No Churn
Churned

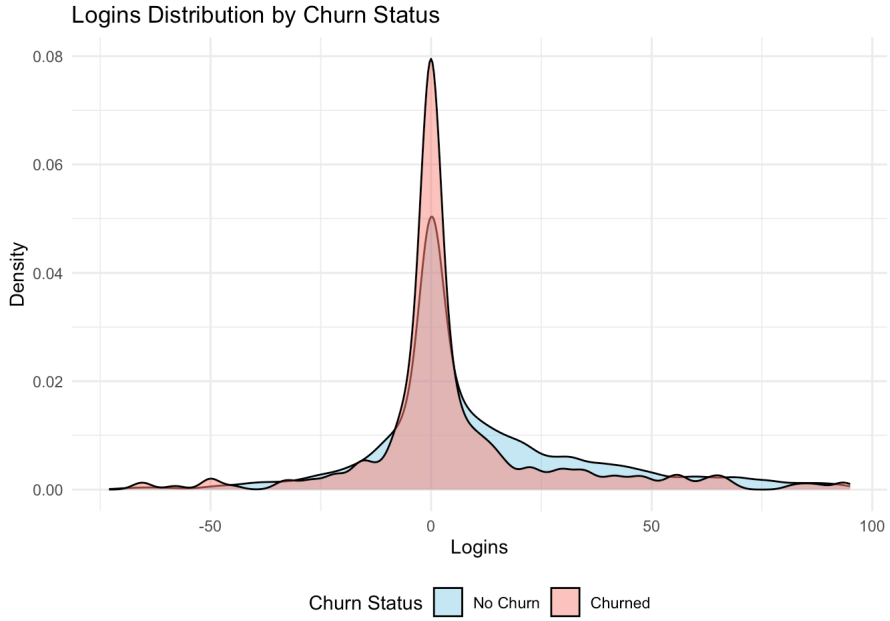

CHI.Score.Mon0 Distribution by Churn Status
Density
0.0075
0.0050
0.0025
0.0000
0
100
200
300
CHI.Score.Mon0
Churn Status
No Churn
Churned

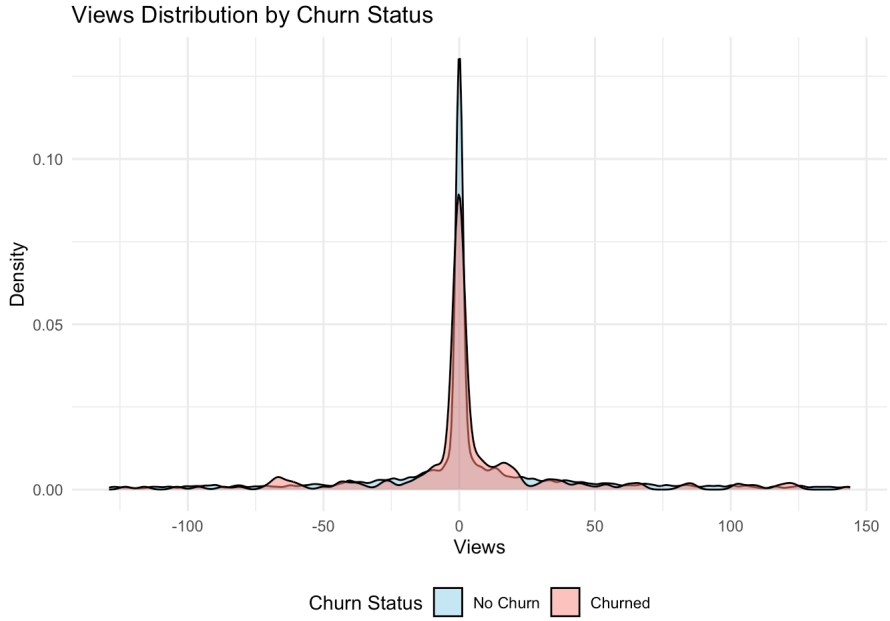

CHI.Score Distribution by Churn Status
Density
0.05
0.04
0.03
0.02
0.01
0.00
-100
0
100
200
CHI.Score
Churn Status
No Churn
Churned

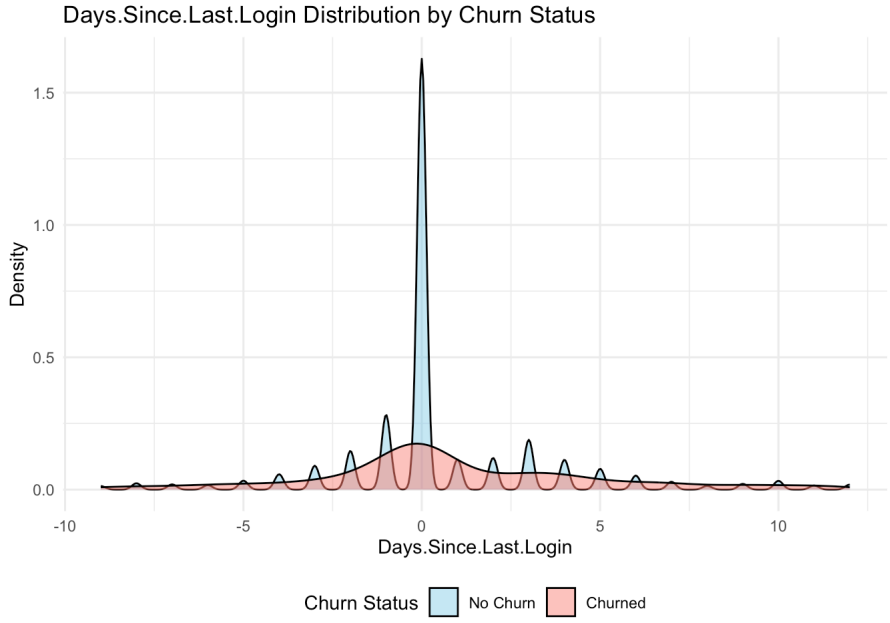

Support.Cases.Mon0 Distribution by Churn Status
Density
Support.Cases.Mon0
Churn Status
No Churn
Churned


SP.Mon0 Distribution by Churn Status
Density
SP.Mon0
Churn Status
No Churn
Churned


SP Distribution by Churn Status
Density
SP
Churn Status
No Churn
Churned


Logins Distribution by Churn Status
Density
0.08
0.06
0.04
0.02
0.00
-50
0
50
100
Logins
Churn Status
No Churn
Churned


Views Distribution by Churn Status
Density
0.10
0.05
0.00
-100
-50
0
50
100
150
Views
Churn Status
No Churn
Churned


Days.Since.Last.Login Distribution by Churn Status
Density
1.5
1.0
0.5
0.0
-10
-5
0
5
10
Days.Since.Last.Login
Churn Status
No Churn
Churned

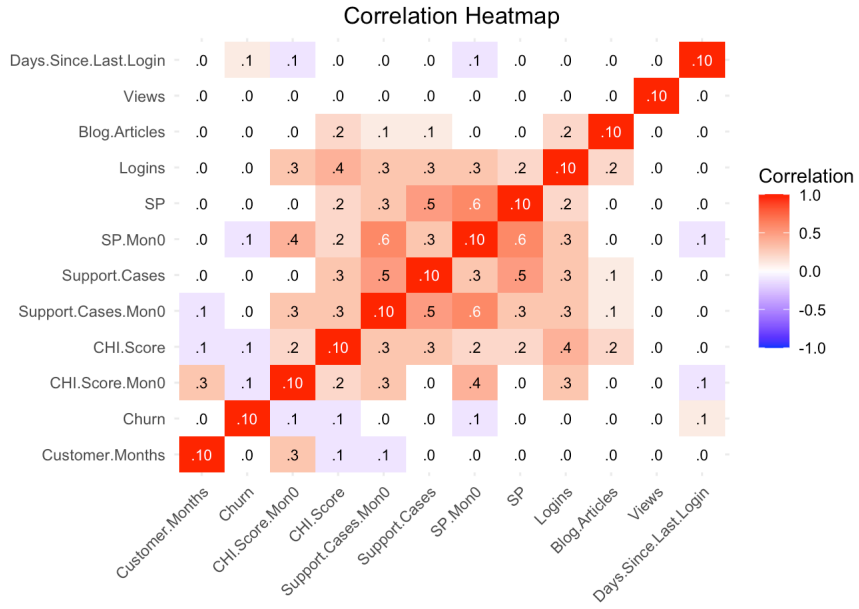
Note: Density plots for 'Support.Cases' and 'Blog.Articles' are not included because, after outlier removal, these variables have too few non-missing values to produce meaningful plots. This is likely due to their highly skewed distributions and the large number of zero values in the data.

Interpretation of the Distribution Plots:

The density plots for the numeric predictors reveal that the distributions for churned and non-churned customers are extremely similar across all variables. There is substantial overlap between the two groups for each predictor, indicating that, on their own, these variables do not provide strong separation between customers who churn and those who do not. This suggests that individual features may not be sufficient to distinguish churners from non-churners, and that more complex relationships or combinations of variables may be necessary to effectively predict churn.

Correlation Analysis

```
# Calculate correlation matrix
corr_matrix <- cor(df_num, use = 'complete.obs')
corr_matrix_rounded <- round(corr_matrix, 1)

# Create heatmap
ggplot(melt(corr_matrix_rounded), aes(Var1, Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(label = sprintf(".%d", abs(value*10))),
            color = ifelse(abs(melt(corr_matrix_rounded)$value) > 0.5, "white", "black"),
            size = 3) +
  scale_fill_gradient2(low = 'blue', high = 'red', mid = 'white',
                       midpoint = 0, limit = c(-1, 1)) +
  labs(title = 'Correlation Heatmap',
       x = '',
       y = '',
       fill = 'Correlation') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.text.y = element_text(hjust = 1),
        plot.title = element_text(hjust = 0.5))
```



Interpretation of Correlation Heatmap

The correlation heatmap reveals that most variables in our dataset have weak or very weak correlations with each other, with the majority of coefficients falling between -0.1 and 0.3. Only a few pairs show moderate to strong positive correlations:

- **Support.Cases.Mon0 with SP.Mon0 (0.6) and SP with SP.Mon0 (0.6):** These strong correlations suggest that both the number of support cases and the service priority (SP) in the current month are closely linked to the service priority in the current month. This indicates that customers who have more support cases tend to have higher service priority, and that service priority is consistent within the same month.
- **Support Cases and Support Cases Mon0 (0.5), SP and Support Cases (0.5):** These moderate correlations further reinforce the connection between support activity and service priority.
- **CHI.Score.Mon0 and SP.Mon0 (0.4), Logins and CHI.Score (0.4):** These relationships suggest that customer health and engagement (logins) are somewhat related, but not strongly so.

The lack of strong correlations among most other variables is beneficial for our analysis, as it means that each variable is likely to provide unique information about customer behavior and churn risk. This reduces the risk of redundancy in our predictive models and helps ensure that each feature can contribute independently to our understanding and prediction of churn.

Correlations with Churn:

The direct correlations between the predictor variables and the target variable “Churn” are very weak. The strongest correlations observed are only ±0.1: - Days Since Last Login with Churn: +0.1 - SP.Mon0 with Churn: -0.1 - CHI.Score with Churn: -0.1 - CHI.Score.Mon0 with Churn: -0.1 - All other variables have a correlation of 0 with Churn.

This is somewhat concerning, as it suggests that none of the individual predictors have a strong linear relationship with customer churn. In other words, no single variable stands out as a clear indicator of whether a customer will churn or not.

What does this mean for our analysis? - It indicates that churn is likely influenced by more complex, possibly non-linear relationships or interactions between variables, rather than by any single factor alone. - Predictive models that can capture interactions and non-linearities (such as decision trees, random forests, or ensemble methods) may be more effective than simple linear models. - Feature engineering, such as creating new variables that combine information from multiple predictors, may help improve model performance. - It also highlights the importance of not relying solely on correlation for feature selection, as important predictors may not show strong linear relationships with the outcome.

How this helps our analysis:
The heatmap helps us quickly identify which variables are closely related and which are not. This guides our feature selection for modeling, allowing us to avoid including highly redundant variables and to focus on those that provide distinct insights into customer behavior. It also reassures us that multicollinearity is not a major concern for most predictors in this dataset.

Conclusion

Summary Table: Key Findings and Next Steps

Aspect	Key Finding / Action
Data Quality	No missing values or duplicate IDs detected. Data is clean and ready for analysis.
Churn Distribution	Churned and non-churned customers are clearly labeled; class imbalance is visually presented.
Univariate Distributions	Numeric predictors show very similar distributions for churned and non-churned customers.
Outlier Handling	Outliers removed for select variables using 3*IQR; some variables (Support.Cases, Blog.Articles) too sparse after removal for density plots.
Correlation Analysis	Most variables are weakly correlated; a few moderate correlations among support/service variables.
Correlation with Churn	All predictors have very weak linear correlation with churn (max ±0.1).
Modeling Implications	No strong univariate predictors; recommend using models that capture interactions/non-linearities.
Next Steps	Proceed to feature engineering and predictive modeling (e.g., decision trees, random forests, etc.).

‴