Data Pre-Processing Report

2023-10-10

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
getwd()
```

```
## [1] "D:/Data Science"
```

```
library(readr)
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
```

```
# Load the data
churn <- read.csv("D:/Data Science/Churn_Train.csv")


# Check for missing values
missing_values <- colSums(is.na(churn))
str(churn)
```

```
## 'data.frame':    6490 obs. of  21 variables:
##  $ CustomerID       : chr  "7590-VHVEG" "5575-GNVDE" "3668-QPYBK" "7795-CFOCW" ...
##  $ Gender           : chr  "Female" "Male" "Male" "Male" ...
##  $ Senior.Citizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner          : chr  "Yes" "No" "No" "No" ...
##  $ Dependents       : chr  "No" "No" "No" "No" ...
##  $ Tenure           : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ Phone.Service    : chr  "No" "Yes" "Yes" "No" ...
```

```
##  $ Multiple.Lines   : chr  "No phone service" "No" "No" "No phone service" ...
##  $ Internet.Service : chr  "DSL" "DSL" "DSL" "DSL" ...
##  $ Online.Security  : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Online.Backup    : chr  "Yes" "No" "Yes" "No" ...
##  $ Device.Protection: chr  "No" "Yes" "No" "Yes" ...
##  $ Tech.Support     : chr  "No" "No" "No" "Yes" ...
##  $ Streaming.TV     : chr  "No" "No" "No" "No" ...
##  $ Streaming.Movies : chr  "No" "No" "No" "No" ...
##  $ Contract         : chr  "Month-to-month" "One year" "Month-to-month" "One year" ...
##  $ Paperless.Billing: chr  "Yes" "No" "Yes" "No" ...
##  $ Payment.Method   : chr  "Electronic check" "Mailed check" "Mailed check" "Bank transfer (automati
##  $ Monthly.Charges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ Total.Charges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn            : chr  "No" "No" "Yes" "No" ...
```

```
summary(churn)
```

```
##   CustomerID           Gender          Senior.Citizen     Partner
##  Length:6490        Length:6490        Min.   :0.0000   Length:6490
##  Class :character   Class :character   1st Qu.:0.0000   Class :character
##  Mode  :character   Mode  :character   Median :0.0000   Mode  :character
##                                        Mean   :0.1627
##                                        3rd Qu.:0.0000
##                                        Max.   :1.0000
##   Dependents            Tenure      Phone.Service      Multiple.Lines
##  Length:6490        Min.   : 1.00   Length:6490        Length:6490
##  Class :character   1st Qu.: 9.00   Class :character   Class :character
##  Mode  :character   Median :29.00   Mode  :character   Mode  :character
##                     Mean   :32.41
##                     3rd Qu.:56.00
##                     Max.   :72.00
##  Internet.Service   Online.Security    Online.Backup      Device.Protection
##  Length:6490        Length:6490        Length:6490        Length:6490
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Tech.Support       Streaming.TV       Streaming.Movies     Contract
##  Length:6490        Length:6490        Length:6490        Length:6490
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Paperless.Billing  Payment.Method     Monthly.Charges  Total.Charges
##  Length:6490        Length:6490        Min.   : 18.25   Min.   :  18.8
##  Class :character   Class :character   1st Qu.: 35.41   1st Qu.: 399.3
##  Mode  :character   Mode  :character   Median : 70.40   Median :1397.1
##                                        Mean   : 64.77   Mean   :2282.9
##                                        3rd Qu.: 89.89   3rd Qu.:3786.6
##                                        Max.   :118.75   Max.   :8684.8
##     Churn
##  Length:6490
```

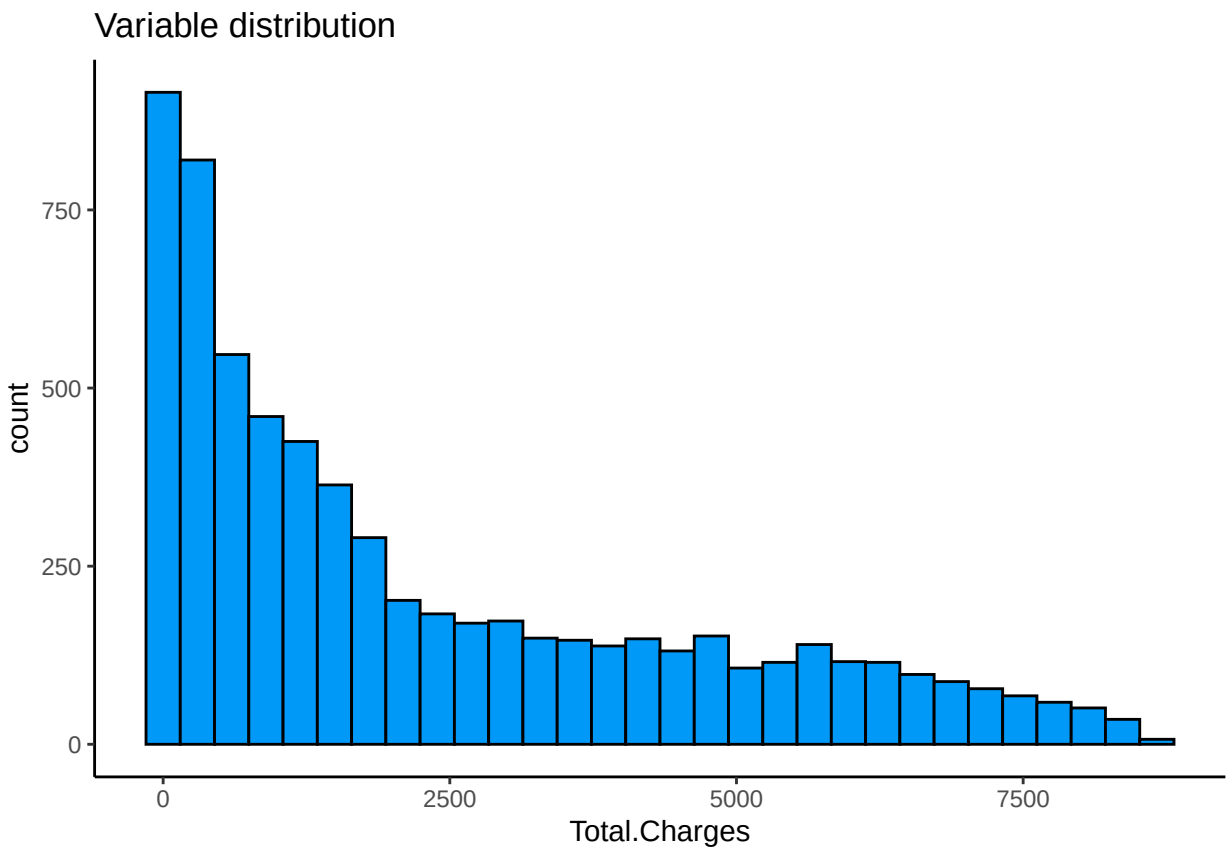
```
##  Class :character
##  Mode  :character
##
##
##
```

```
library(ggplot2)
library(cowplot)

# Replace missing values with means
churn_replace <- churn %>% mutate_all(funs(ifelse(is.na(.), mean(., na.rm = TRUE), .)))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# Create a histogram for Total Charges
ggplot(churn, aes(Total.Charges)) +
  geom_histogram(color = "#000000", fill = "#0099F8") +
  ggtitle("Variable distribution") +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

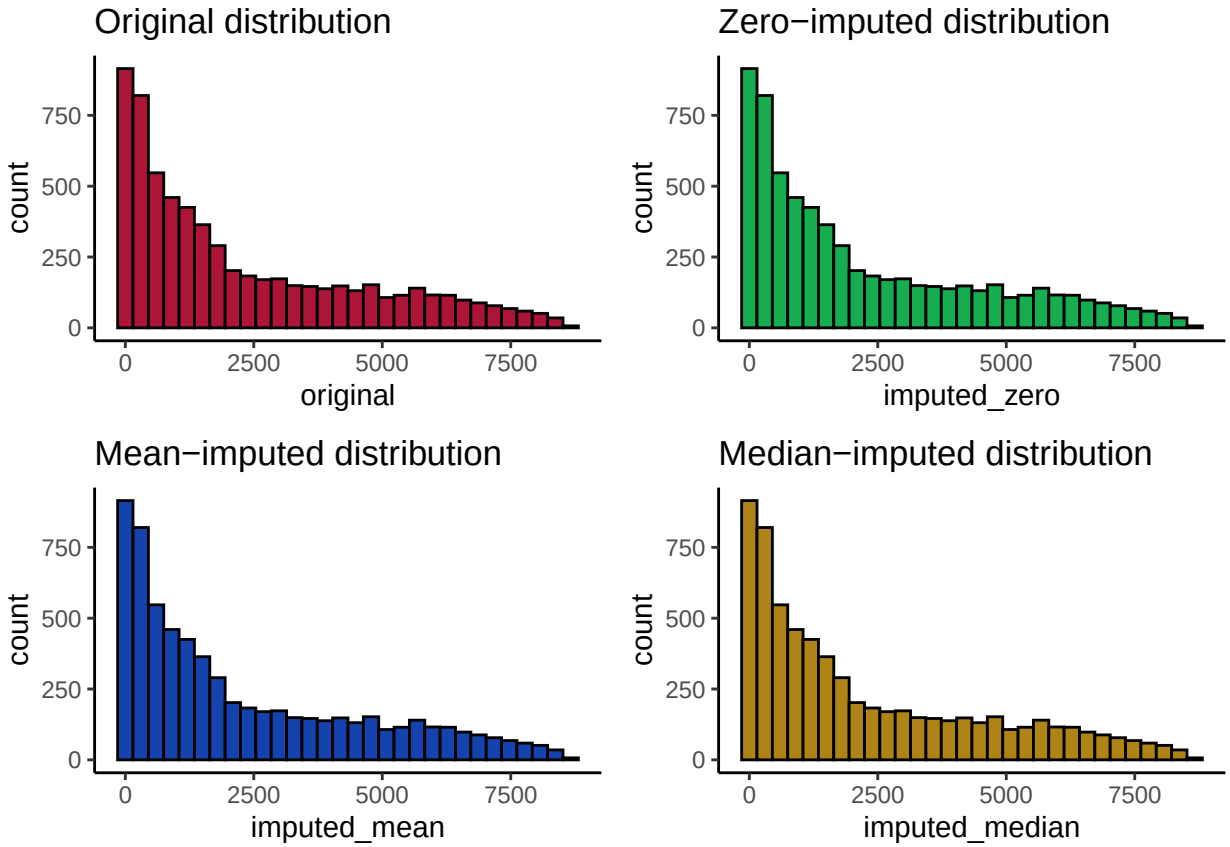



```
# Create a data frame for value imputation
value_imputed <- data.frame(
  original = churn$Total.Charges,
  imputed_zero = replace(churn$Total.Charges, is.na(churn$Total.Charges), 0),
  imputed_mean = replace(churn$Total.Charges, is.na(churn$Total.Charges), mean(churn$Total.Charges, na.
  imputed_median = replace(churn$Total.Charges, is.na(churn$Total.Charges), median(churn$Total.Charges,
)

# Create histograms for value imputation
h1 <- ggplot(value_imputed, aes(x = original)) +
  geom_histogram(fill = "#ad1538", color = "#000000", position = "identity") +
  ggtitle("Original distribution") +
  theme_classic()

h2 <- ggplot(value_imputed, aes(x = imputed_zero)) +
  geom_histogram(fill = "#15ad4f", color = "#000000", position = "identity") +
  ggtitle("Zero-imputed distribution") +
  theme_classic()

h3 <- ggplot(value_imputed, aes(x = imputed_mean)) +
  geom_histogram(fill = "#1543ad", color = "#000000", position = "identity") +
  ggtitle("Mean-imputed distribution") +
  theme_classic()

h4 <- ggplot(value_imputed, aes(x = imputed_median)) +
  geom_histogram(fill = "#ad8415", color = "#000000", position = "identity") +
```

```
  ggtitle("Median-imputed distribution") +
  theme_classic()

# Combine histograms into a grid
plot_grid(h1, h2, h3, h4, nrow = 2, ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



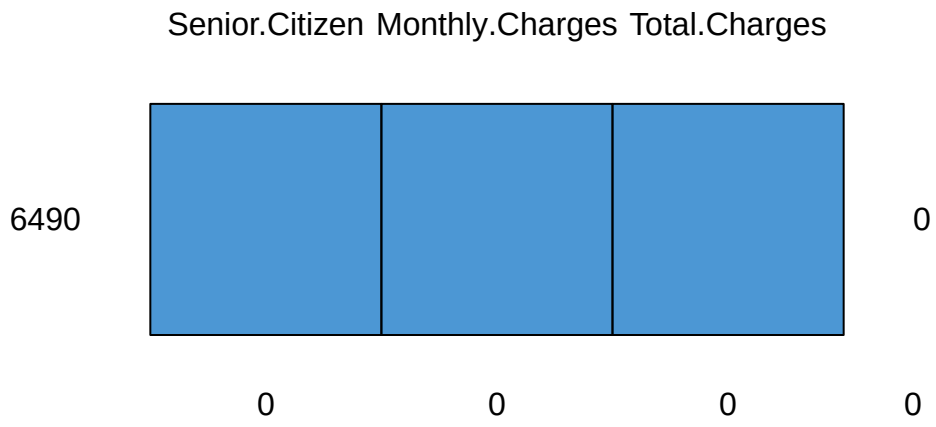

```
# Select numeric variables for imputation
churn_numeric <- churn %>%
  select (Senior.Citizen, Monthly.Charges, Total.Charges)

library(mice)
```

```
##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
```

```
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
set.seed(0)

# Generate a missing data pattern plot
md.pattern(churn_numeric)
```

```
##  /\     /\
## {  `---'  }
## {  0   0  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##      Senior.Citizen Monthly.Charges Total.Charges
## 6490              1               1             1 0
##                   0               0             0 0
```

```r
# Impute missing values using mice
mice_imputed <- data.frame(
  original = churn$Total.Charges,
```

```
  imputed_pmm = complete(mice(churn_numeric, method = "pmm"))$Total.Charges,
  imputed_cart = complete(mice(churn_numeric, method = "cart"))$Total.Charges,
  imputed_lasso = complete(mice(churn_numeric, method = "lasso.norm"))$Total.Charges
)
```

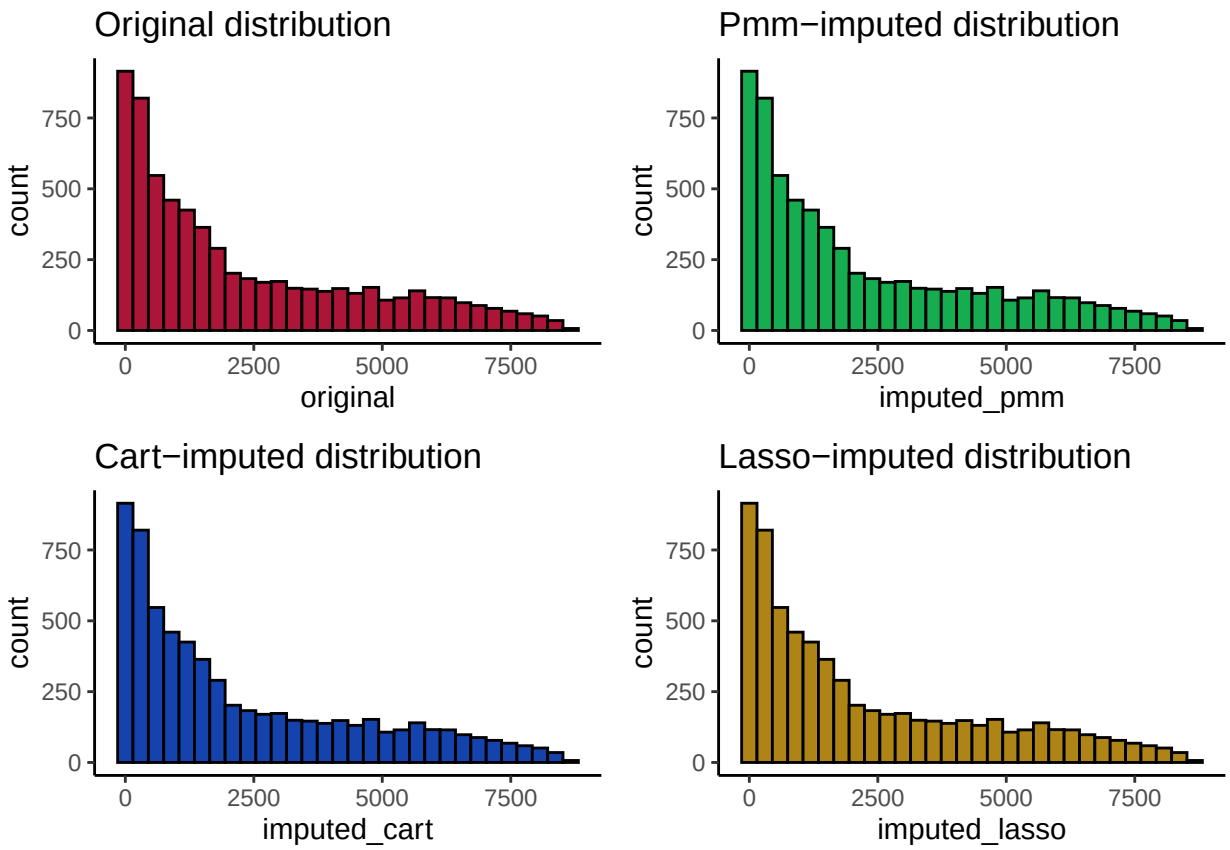
```
##
##  iter imp variable
##   1   1
##   1   2
##   1   3
##   1   4
##   1   5
##   2   1
##   2   2
##   2   3
##   2   4
##   2   5
##   3   1
##   3   2
##   3   3
##   3   4
##   3   5
##   4   1
##   4   2
##   4   3
##   4   4
##   4   5
##   5   1
##   5   2
##   5   3
##   5   4
##   5   5
##
##  iter imp variable
##   1   1
##   1   2
##   1   3
##   1   4
##   1   5
##   2   1
##   2   2
##   2   3
##   2   4
##   2   5
##   3   1
##   3   2
##   3   3
##   3   4
##   3   5
##   4   1
##   4   2
##   4   3
##   4   4
```

```
##   4   5
##   5   1
##   5   2
##   5   3
##   5   4
##   5   5
##
##  iter imp variable
##   1   1
##   1   2
##   1   3
##   1   4
##   1   5
##   2   1
##   2   2
##   2   3
##   2   4
##   2   5
##   3   1
##   3   2
##   3   3
##   3   4
##   3   5
##   4   1
##   4   2
##   4   3
##   4   4
##   4   5
##   5   1
##   5   2
##   5   3
##   5   4
##   5   5
```

```r
# Create histograms for mice imputation
h1 <- ggplot(mice_imputed, aes(x = original)) +
  geom_histogram(fill = "#ad1538", color = "#000000", position = "identity") +
  ggtitle("Original distribution") +
  theme_classic()

h2 <- ggplot(mice_imputed, aes(x = imputed_pmm)) +
  geom_histogram(fill = "#15ad4f", color = "#000000", position = "identity") +
  ggtitle("Pmm-imputed distribution") +
  theme_classic()

h3 <- ggplot(mice_imputed, aes(x = imputed_cart)) +
  geom_histogram(fill = "#1543ad", color = "#000000", position = "identity") +
  ggtitle("Cart-imputed distribution") +
  theme_classic()

h4 <- ggplot(mice_imputed, aes(x = imputed_lasso)) +
  geom_histogram(fill = "#ad8415", color = "#000000", position = "identity") +
  ggtitle("Lasso-imputed distribution") +
  theme_classic()
```

```r
# Combine histograms into a grid
plot_grid(h1, h2, h3, h4, nrow = 2, ncol = 2)
```

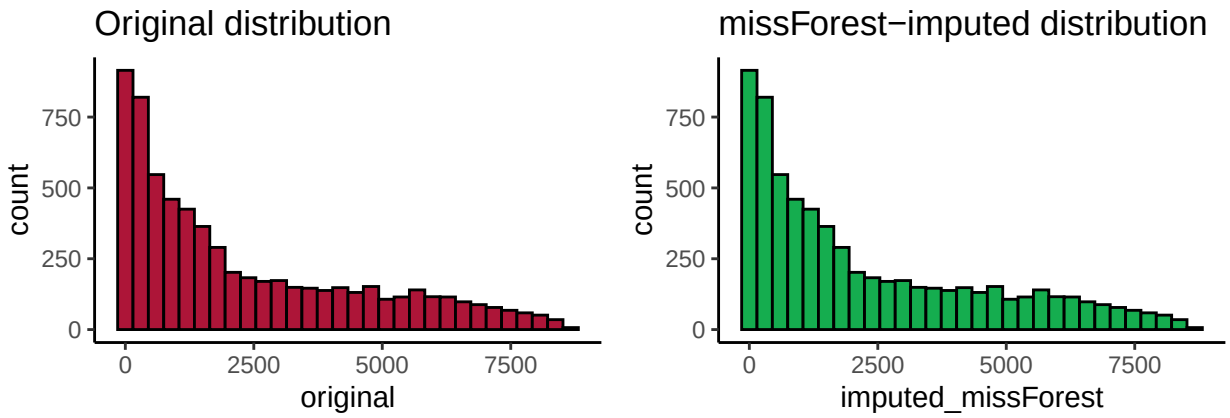
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
library(missForest)

# Impute missing values using missForest
missForest_imputed <- data.frame(
  original = churn_numeric$Total.Charges,
  imputed_missForest = missForest(churn_numeric)$ximp$Total.Charges
)

# Create histograms for missForest imputation
h1 <- ggplot(missForest_imputed, aes(x = original)) +
  geom_histogram(fill = "#ad1538", color = "#000000", position = "identity") +
  ggtitle("Original distribution") +
  theme_classic()

h2 <- ggplot(missForest_imputed, aes(x = imputed_missForest)) +
  geom_histogram(fill = "#15ad4f", color = "#000000", position = "identity") +
```

```
  ggtitle("missForest-imputed distribution") +
  theme_classic()

# Combine histograms into a grid
plot_grid(h1, h2, nrow = 2, ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Scale data using various methods
log_scale <- log(churn$Total.Charges)

library(caret)
```

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

```
process <- preProcess(as.data.frame(churn$Total.Charges), method = c("range"))
norm_scale <- predict(process, as.data.frame(churn$Total.Charges))

scale_data <- as.data.frame(scale(churn$Total.Charges))

summary(scale_data)
```

```
##        V1
##  Min.   :-0.9974
##  1st Qu.:-0.8298
##  Median :-0.3902
##  Mean   : 0.0000
##  3rd Qu.: 0.6624
##  Max.   : 2.8202
```

```
# Encode categorical variable 'Gender'
gender_encode <- ifelse(churn$Gender == "male", 1, 0)
table(gender_encode)
```

```
## gender_encode
##    0
## 6490
```

```
new_dat <- data.frame(churn$Total.Charges, churn$Gender, churn$Senior.Citizen)
summary(new_dat)
```

```
##  churn.Total.Charges churn.Gender       churn.Senior.Citizen
##  Min.   :  18.8      Length:6490        Min.   :0.0000
##  1st Qu.: 399.3      Class :character   1st Qu.:0.0000
##  Median :1397.1      Mode  :character   Median :0.0000
##  Mean   :2282.9                         Mean   :0.1627
##  3rd Qu.:3786.6                         3rd Qu.:0.0000
##  Max.   :8684.8                         Max.   :1.0000
```

```
library(caret)

dmy <- dummyVars(" ~ .", data = new_dat, fullRank = T)
dat_transformed <- data.frame(predict(dmy, newdata = new_dat))

glimpse(dat_transformed)
```

```
## Rows: 6,490
## Columns: 3
## $ churn.Total.Charges  <dbl> 29.85, 1889.50, 108.15, 1840.75, 151.65, 820.50, ~
## $ churn.GenderMale     <dbl> 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0~
## $ churn.Senior.Citizen <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
summary(new_dat$churn.Total.Charges)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    18.8   399.3  1397.1  2282.9  3786.6  8684.8
```