D209 Data Mining - Task 2

Scott Babcock WGU - MS, Data Analytics

Created: January 18 2025

A1, Research Question

Can customer tenure be predicted using lasso regression so that the company can predict future customer tenure and learn key attributes of long-term customers?

A2, Goals of Analysis

The analysis aims to develop a supervised machine learning model using lasso regression to predict customer tenure. Accurate predictions with this model will be valuable so the company can predict the tenure of new customers and identify key data points that signal good customers.

B1, Lasso Regression Method

The lasso regression method is an appropriate technique for this analysis. Lasso regression is similar to multiple regression; however, the key difference is that lasso regression applies a penalty to unimportant features. The lasso technique effectively performs feature selection with its lambda penalty. The features with little influence have their coefficients reduced closer to zero. Lambda must be tuned to ensure the model doesn't become overly simplistic, which can lead to a sacrifice in accuracy or overly complex, leading to overfitting. The result is a regression equation that can be applied to the new data (What is Lasso, 2024).

B2, Lasso Regression Assumption

A key assumption of lasso regression is that the true underlying model is sparse. This assumption means there is only a specific subset of features that are significant predictors, and the rest have little impact on the model. This assumption allows the lambda penalty to perform feature selection to prevent model overfitting (What is Lasso, 2024).

B3, Libraries and Packages

Several packages were used in this analysis to assist with data manipulation or calculations. First, the Dplyr package was used for data manipulation. Dplyr allowed the selection of specific features and the changing of the names of existing features. The Naniar package was used to identify missing values. The miss_var_summary function creates a table with the number of missing values and the percentage of the dataset the missing values make up. The fastDummies package was used to generate dummy columns for one hot encoding. The dummy_cols function from the package allowed for the creation of dummy variables in one simple step while also enabling the choice of removing a dummy column if needed and removing the original data. The corrplot package was used to create a correlation plot among the variables. The rsample package was used to develop training and testing splits. The initial_split function allows the user to set a proportion of the data used for the split. The caret and glmnet packages were used for the model building. Caret enabled the model template, and glmnet enabled the lasso regression technique.

```
library(dplyr)
library(naniar)
library(fastDummies)
library(corrplot)
library(rsample)
library(caret)
library(glmnet)
```

C1, Data Preprocessing

One data preprocessing step taken in the analysis was to scale the data. There was a diverse selection of features, some with larger scales. If the features were not scaled, larger values would disproportionately influence the model, leading to biased results. The scale function was used to achieve this across the non-binary numeric variables. The resulting scaled features have a mean of zero and a standard deviation of one (Bobbitt, 2021).

C2, Dataset Variables

The following variables were used in the initial subset for the analysis. The Tenure variable was used as the response variable. Tenure is a continuous numeric variable. The Children variable is numeric and needs to be scaled. The Age variable is numeric and needs to be scaled. The Income variable is numeric and needs to be scaled. Marital is a categorical variable re-expressed as numeric with one hot encoding. Gender is a categorical variable re-expressed as numeric with one hot encoding. Techie is a categorical variable re-expressed as numeric with one hot encoding. Contract is a categorical variable re-expressed as numeric with one hot encoding. Tablet is a categorical variable re-expressed as numeric with one hot encoding. DeviceProtection is a categorical variable re-expressed as numeric with one hot encoding. TechSupport is a categorical variable re-expressed as numeric with one hot encoding. StreamingTV is a categorical variable re-expressed as numeric with one hot encoding. StreamingMovies is a categorical variable re-expressed as numeric with one hot encoding. The MonthlyCharge variable is numeric and needs to be scaled. The Bandwidth_GB_Year variable is numeric and needs to be scaled.

C3, Steps for Analysis

The following steps were followed to prepare the data for the analysis.

- The appropriate libraries were loaded, and data from the churn_clean CSV file was loaded into the programming environment.
- The data was previewed, and data types were identified.
- The data was checked for duplicate values. No duplicate values were identified.
- The data was checked for missing values. No missing values were identified.
- The subset of data used in the analysis was selected, and a new data frame was created.
- The numeric variables were checked for outliers. The number of outliers were across three points in the Children variable, which were deemed reasonable. The income ranges were considered reasonable as well.
- Unique values were identified for each categorical variable. The number of values was reasonable for the creation of dummy variables.
- The categorical variables were re-expressed as numeric with one hot encoding.
- The non-binary numeric values were centered and scaled for the analysis to reduce bias.
- Summary statistics and standard deviation were viewed on each of the three variables to verify that the scaling occurred. All had a mean of zero and a standard deviation of one.
- A correlation plot was created to view the relationships between variables.
- The cleaned and transformed data was written into a CSV file.

```
# load churn dataset
churn_df <- read.csv("churn_clean.csv")

# view data structure and types
str(churn_df)
```

```
## 'data.frame':    10000 obs. of  50 variables:
##  $ CaseOrder           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Customer_id         : chr  "K409198" "S120509" "K191035" "D90850" ...
##  $ Interaction         : chr  "aa90260b-4141-4a24-8e36-b04ce1f4f77b" "fb76459f-c047-4a9d-8af9-e0f7d4
##  $ UID                 : chr  "e885b299883d4f9fb18e39c75155d990" "f2de8bef964785f41a2959829830fb8a"
##  $ City                : chr  "Point Baker" "West Branch" "Yamhill" "Del Mar" ...
##  $ State               : chr  "AK" "MI" "OR" "CA" ...
##  $ County              : chr  "Prince of Wales-Hyder" "Ogemaw" "Yamhill" "San Diego" ...
##  $ Zip                 : int  99927 48661 97148 92014 77461 31030 37847 73109 34771 45237 ...
##  $ Lat                 : num  56.3 44.3 45.4 33 29.4 ...
##  $ Lng                 : num  -133.4 -84.2 -123.2 -117.2 -95.8 ...
##  $ Population          : int  38 10446 3735 13863 11352 17701 2535 23144 17351 20193 ...
##  $ Area                : chr  "Urban" "Urban" "Urban" "Suburban" ...
##  $ TimeZone            : chr  "America/Sitka" "America/Detroit" "America/Los_Angeles" "America/Los_A
##  $ Job                 : chr  "Environmental health practitioner" "Programmer, multimedia" "Chief Fi
##  $ Children            : int  0 1 4 1 0 3 0 2 2 1 ...
##  $ Age                 : int  68 27 50 48 83 83 79 30 49 86 ...
##  $ Income              : num  28562 21705 9610 18925 40074 ...
##  $ Marital             : chr  "Widowed" "Married" "Widowed" "Married" ...
##  $ Gender              : chr  "Male" "Female" "Female" "Male" ...
##  $ Churn               : chr  "No" "Yes" "No" "No" ...
##  $ Outage_sec_perweek  : num  7.98 11.7 10.75 14.91 8.15 ...
##  $ Email               : int  10 12 9 15 16 15 10 16 20 18 ...
##  $ Contacts            : int  0 0 0 2 2 3 0 0 2 1 ...
##  $ Yearly_equip_failure: int  1 1 1 0 1 1 1 0 3 0 ...
##  $ Techie              : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Contract            : chr  "One year" "Month-to-month" "Two Year" "Two Year" ...
##  $ Port_modem          : chr  "Yes" "No" "Yes" "No" ...
##  $ Tablet              : chr  "Yes" "Yes" "No" "No" ...
##  $ InternetService     : chr  "Fiber Optic" "Fiber Optic" "DSL" "DSL" ...
##  $ Phone               : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ Multiple            : chr  "No" "Yes" "Yes" "No" ...
##  $ OnlineSecurity      : chr  "Yes" "Yes" "No" "Yes" ...
##  $ OnlineBackup        : chr  "Yes" "No" "No" "No" ...
##  $ DeviceProtection    : chr  "No" "No" "No" "No" ...
##  $ TechSupport         : chr  "No" "No" "No" "No" ...
##  $ StreamingTV         : chr  "No" "Yes" "No" "Yes" ...
##  $ StreamingMovies     : chr  "Yes" "Yes" "Yes" "No" ...
##  $ PaperlessBilling    : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ PaymentMethod       : chr  "Credit Card (automatic)" "Bank Transfer(automatic)" "Credit Card (aut
##  $ Tenure              : num  6.8 1.16 15.75 17.09 1.67 ...
##  $ MonthlyCharge       : num  172 243 160 120 150 ...
##  $ Bandwidth_GB_Year   : num  905 801 2055 2165 271 ...
##  $ Item1               : int  5 3 4 4 4 3 6 2 5 2 ...
##  $ Item2               : int  5 4 4 4 4 3 5 2 4 2 ...
##  $ Item3               : int  5 3 2 4 4 3 6 2 4 2 ...
##  $ Item4               : int  3 3 4 2 3 2 4 5 3 2 ...
##  $ Item5               : int  4 4 4 5 4 4 1 2 4 5 ...
##  $ Item6               : int  4 3 3 4 4 3 5 3 3 2 ...
##  $ Item7               : int  3 4 3 3 4 3 5 4 4 3 ...
##  $ Item8               : int  4 4 3 3 5 3 5 5 4 3 ...
```

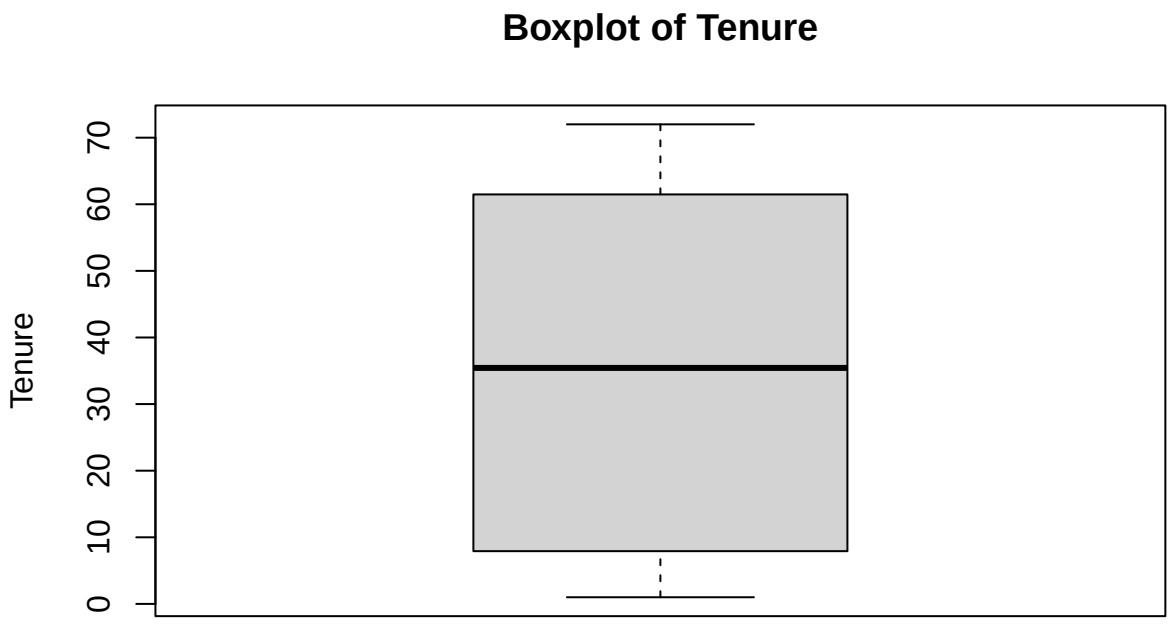
```
# check for duplicate records [In-text citation:(Getting Started with Duplicates, n.d.)]
sum(duplicated(churn_df))
```

```
## [1] 0
```

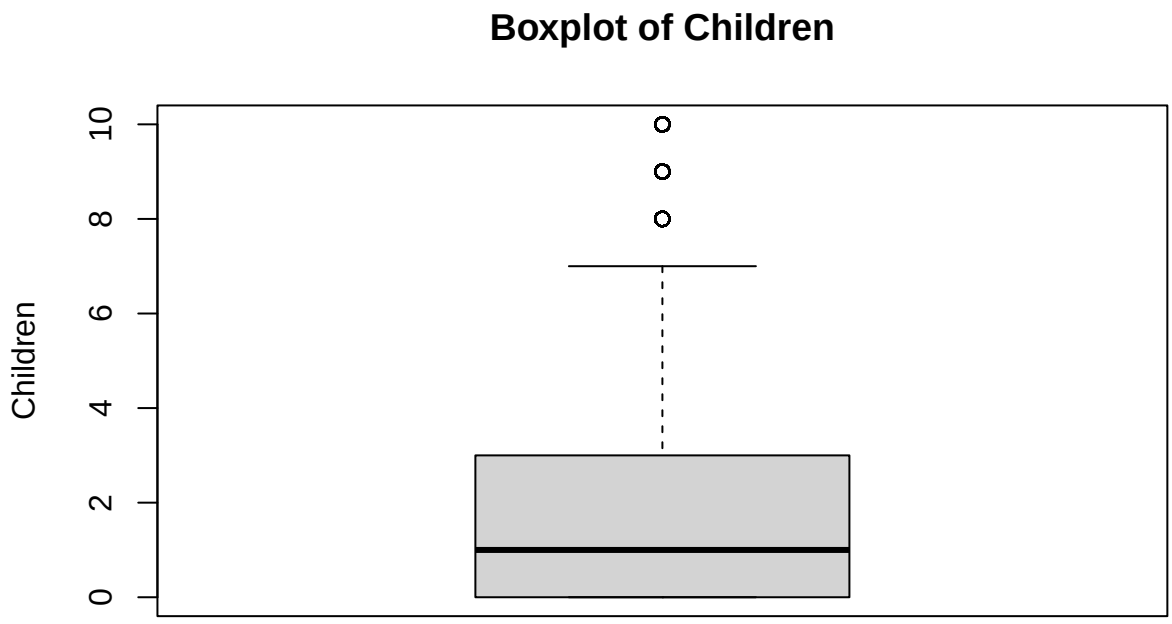
```
# check for missing values [In-text citation: (Tierney, n.d.)]
miss_var_summary(churn_df)
```

```
## # A tibble: 50 x 3
##    variable    n_miss pct_miss
##    <chr>        <int>    <num>
##  1 CaseOrder        0        0
##  2 Customer_id      0        0
##  3 Interaction      0        0
##  4 UID              0        0
##  5 City             0        0
##  6 State            0        0
##  7 County           0        0
##  8 Zip              0        0
##  9 Lat              0        0
## 10 Lng              0        0
## # i 40 more rows
```

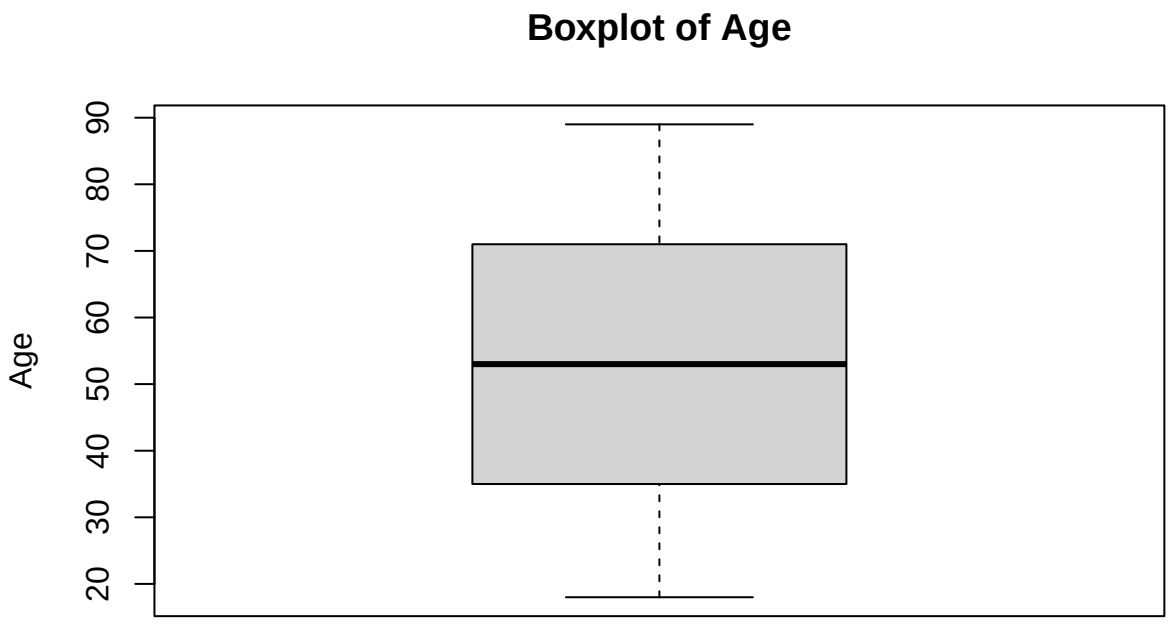
```
# select variables for analysis
churn_analysis_initial <- churn_df %>%
  select(
    Tenure,
    Children,
    Age,
    Income,
    Marital,
    Gender,
    Techie,
    Contract,
    Tablet,
    DeviceProtection,
    TechSupport,
    StreamingTV,
    StreamingMovies,
    MonthlyCharge,
    Bandwidth_GB_Year,
  )

# check for outliers in numeric variables
boxplot(churn_analysis_initial$Tenure,
        ylab = "Tenure",
        main = "Boxplot of Tenure")
```



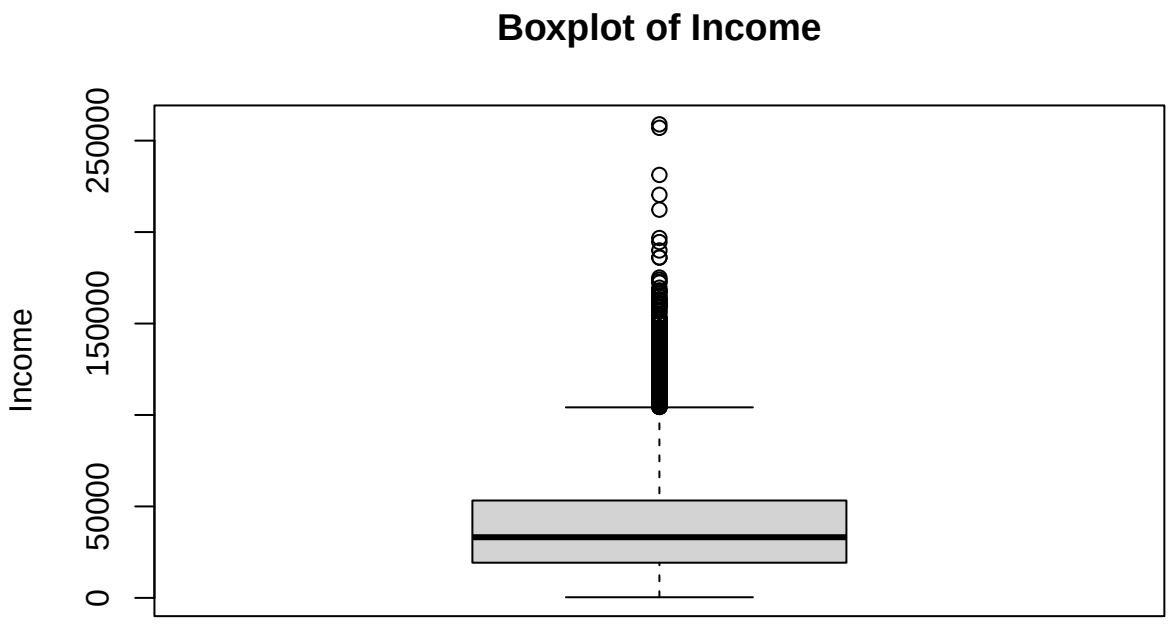

```
boxplot(churn_analysis_initial$Children,
        ylab = "Children",
        main = "Boxplot of Children")
```



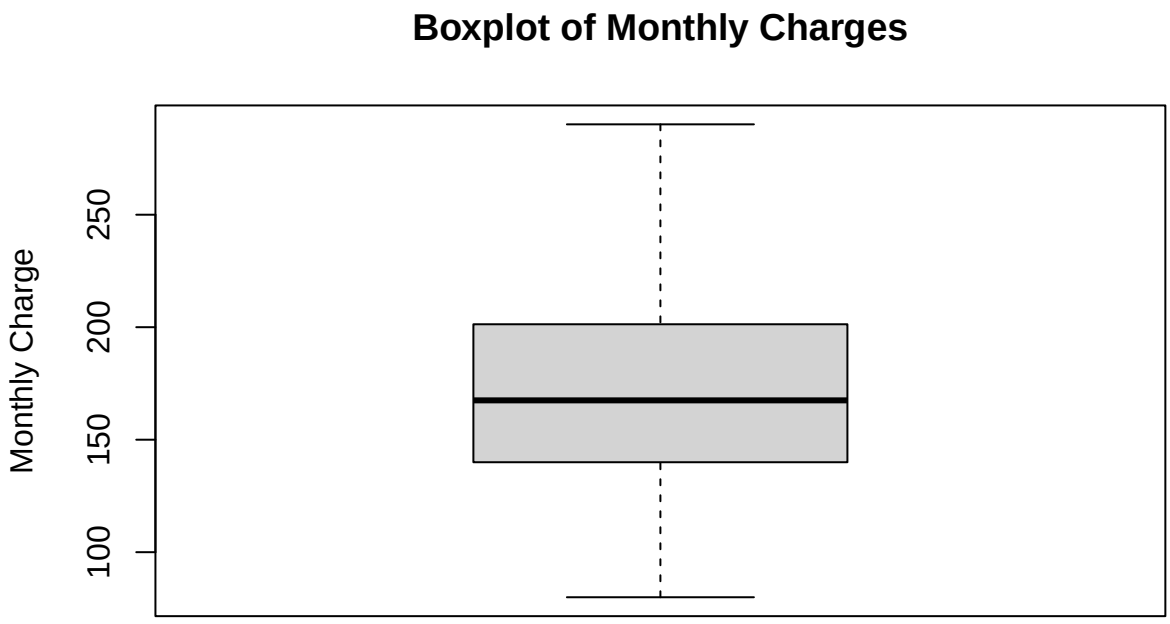

```
boxplot(churn_analysis_initial$Age,
        ylab = "Age",
        main = "Boxplot of Age")
```



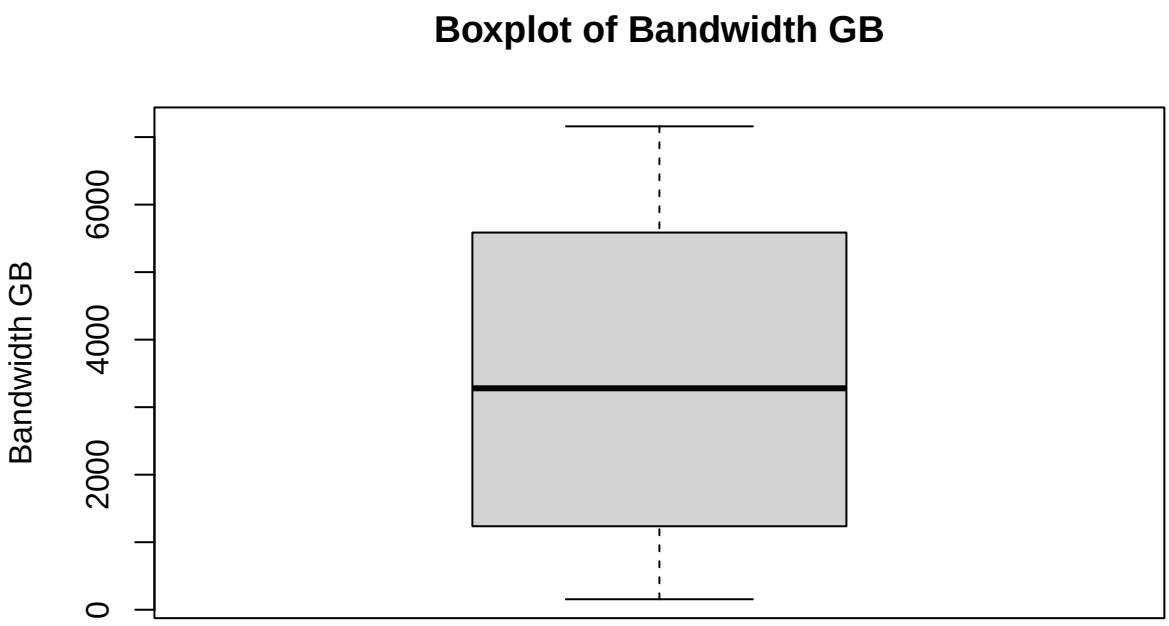

```
boxplot(churn_analysis_initial$Income,
        ylab = "Income",
        main = "Boxplot of Income")
```




```
boxplot(churn_analysis_initial$MonthlyCharge,
        ylab = "Monthly Charge",
        main = "Boxplot of Monthly Charges")
```




```
boxplot(churn_analysis_initial$Bandwidth_GB_Year,
        ylab = "Bandwidth GB",
        main = "Boxplot of Bandwidth GB")
```




```
# view the number of outliers in Children variable and distribution of Income [In-text citation: (Soete
addmargins(table(boxplot.stats(churn_analysis_initial$Children)$out))
```

```
##
##   8   9  10 Sum
## 210  92  99 401
```

```
summary(churn_analysis_initial$Income)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    348.7  19224.7  33170.6  39806.9  53246.2 258900.7
```

```
# view unique values for each categorical variable
unique(churn_analysis_initial$Marital)
```

```
## [1] "Widowed"       "Married"       "Separated"     "Never Married"
## [5] "Divorced"
```

```
unique(churn_analysis_initial$Gender)
```

```
## [1] "Male"      "Female"    "Nonbinary"
```

```r
unique(churn_analysis_initial$Techie)
```

```
## [1] "No"  "Yes"
```

```r
unique(churn_analysis_initial$Contract)
```

```
## [1] "One year"       "Month-to-month" "Two Year"
```

```r
unique(churn_analysis_initial$Tablet)
```

```
## [1] "Yes" "No"
```

```r
unique(churn_analysis_initial$DeviceProtection)
```

```
## [1] "No"  "Yes"
```

```r
unique(churn_analysis_initial$TechSupport)
```

```
## [1] "No"  "Yes"
```

```r
unique(churn_analysis_initial$StreamingTV)
```

```
## [1] "No"  "Yes"
```

```r
unique(churn_analysis_initial$StreamingMovies)
```

```
## [1] "Yes" "No"
```

```r
# transform categorical variables with one hot encoding [In-text citation: (Kaplan, 2020)]
churn_analysis <- dummy_cols(
  churn_analysis_initial,
  select_columns = c("Marital",
                     "Gender",
                     "Techie",
                     "Contract",
                     "Tablet",
                     "DeviceProtection",
                     "TechSupport",
                     "StreamingTV",
                     "StreamingMovies"),
  remove_first_dummy = TRUE,
  remove_selected_columns = TRUE
)

# center and scale non-binary numeric values [In-text citation: (Bobbitt, 2021)]
churn_analysis_scale <- churn_analysis %>%
  mutate(Children = scale(Children),
         Age = scale(Age),
```

```
        Income = scale(Income),
        MonthlyCharge = scale(MonthlyCharge),
        Bandwidth_GB_Year = scale(Bandwidth_GB_Year)
  )

summary(churn_analysis_scale$Children)
```

```
##        V1
##  Min.   :-0.9723
##  1st Qu.:-0.9723
##  Median :-0.5066
##  Mean   : 0.0000
##  3rd Qu.: 0.4249
##  Max.   : 3.6849
```

```
sd(churn_analysis_scale$Children)
```

```
## [1] 1
```

```
summary(churn_analysis_scale$Age)
```

```
##        V1
##  Min.   :-1.694700
##  1st Qu.:-0.873400
##  Median :-0.003788
##  Mean   : 0.000000
##  3rd Qu.: 0.865825
##  Max.   : 1.735437
```

```
sd(churn_analysis_scale$Age)
```

```
## [1] 1
```

```
summary(churn_analysis_scale$Income)
```

```
##        V1
##  Min.   :-1.3992
##  1st Qu.:-0.7299
##  Median :-0.2353
##  Mean   : 0.0000
##  3rd Qu.: 0.4766
##  Max.   : 7.7693
```

```
sd(churn_analysis_scale$Income)
```

```
## [1] 1
```

```r
summary(churn_analysis_scale$MonthlyCharge)
```

```
##        V1
##  Min.   :-2.1574
##  1st Qu.:-0.7602
##  Median :-0.1197
##  Mean   : 0.0000
##  3rd Qu.: 0.6546
##  Max.   : 2.7370
```

```r
sd(churn_analysis_scale$MonthlyCharge)
```

```
## [1] 1
```

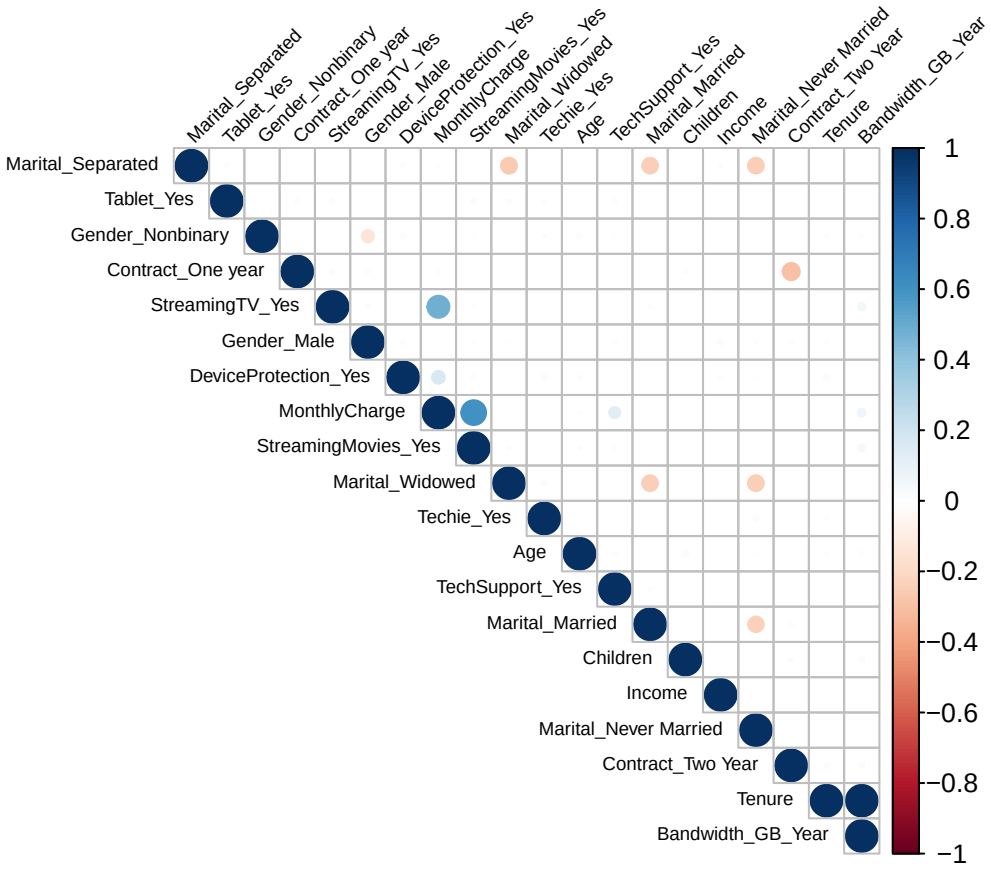
```r
summary(churn_analysis_scale$Bandwidth_GB_Year)
```

```
##        V1
##  Min.   :-1.48119
##  1st Qu.:-0.98654
##  Median :-0.05162
##  Mean   : 0.00000
##  3rd Qu.: 1.00389
##  Max.   : 1.72363
```

```r
sd(churn_analysis_scale$Bandwidth_GB_Year)
```

```
## [1] 1
```

```r
# create correlation plot [In-text citation: (Schork, n.d.)]
corrplot(cor(churn_analysis_scale),
         method = "circle",
         tl.cex = 0.6,
         tl.srt = 45,
         tl.col = "black",
         type = "upper",
         order = "hclust")
```




```
# write cleaned and transformed data to csv
write.csv(churn_analysis_scale,
          "d209_task2_babcock_churn_transformed.csv",
          row.names = FALSE)
```

C4, Cleaned Dataset

The cleaned and transformed dataset used in the analysis was written to a CSV file and is included in the submission.

D1, Splitting the Data

The data was split using a 70%/30% ratio for training and testing.

```
# create train/test data with 70/30 split [In-text citation: (Simple Training, n.d.)]
set.seed(444)

split <- initial_split(churn_analysis_scale, prop = 0.7)

train <- training(split)
test  <- testing(split)
```

The training and testing data were written to CSV files and are included in the submission.

```r
write.csv(train,
            "d209_task2_babcock_train_data.csv",
            row.names = FALSE)

write.csv(test,
          "d209_task2_babcock_test_data.csv",
          row.names = FALSE)
```

D2, Output and Calculations

After all the preprocessing steps, the model was built. The lasso regression technique was used to analyze the data. Using the caret package, the trainControl function allowed for cross-validation. Cross-validation divides the data into multiple subsets and evaluates the model on each subset. This aims to prevent overfitting (Cross validation, n.d.).

The lasso regression model was built with Tenure as the response variable against the remaining 19 explanatory variables. A tuning grid was created with alpha equal to one, indicating that lasso regression was to be used. For lambda, the model would use a sequence ranging from 0.01 to 1 with 20 values to find the best-performing value. The model was fit, and the lambda value used was 0.062, which resulted in the lowest RMSE.

The model was then used to make predictions on the unseen test data. After creating the prediction data frame, the model metrics from the training and test data were assessed. The model produced a 2.558 RMSE and 0.9907 r-squared on the training data. When run on the test data, the model produced a 2.539 RMSE and 0.9907 r-squared value. These metrics indicate a strong predictive model, which performed better on the test data given the smaller RMSE.

D3, Code Execution

The following code was used to build and train the lasso model.

```r
# train LASSO model [In-text citation: (Kuhn, n.d.)]
set.seed(444)
train_control <- trainControl(method = "cv",
                              number = 10,
                              )
tune_grid <- expand.grid(
               alpha = 1,
               lambda = seq(0.01,1,length = 20)
             )
lasso_fit <- train(Tenure ~ .,
                   data = train,
                   method = "glmnet",
                   tuneGrid = tune_grid,
                   trControl = train_control
             )
lasso_fit
```

```
## glmnet
##
## 7000 samples
```

```
##   19 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 6300, 6300, 6300, 6300, 6300, 6300, ...
## Resampling results across tuning parameters:
##
##   lambda      RMSE      Rsquared   MAE
##   0.01000000  2.558497  0.9907025  2.323408
##   0.06210526  2.558458  0.9907027  2.323314
##   0.11421053  2.570971  0.9906284  2.323371
##   0.16631579  2.591505  0.9905046  2.326053
##   0.21842105  2.619357  0.9903339  2.331687
##   0.27052632  2.654407  0.9901151  2.341503
##   0.32263158  2.696320  0.9898478  2.356016
##   0.37473684  2.744678  0.9895322  2.376020
##   0.42684211  2.796988  0.9891856  2.400535
##   0.47894737  2.848885  0.9888419  2.426657
##   0.53105263  2.898242  0.9885191  2.453284
##   0.58315789  2.950720  0.9881693  2.483406
##   0.63526316  3.007063  0.9877844  2.517464
##   0.68736842  3.067030  0.9873642  2.554961
##   0.73947368  3.128867  0.9869225  2.594449
##   0.79157895  3.185983  0.9865201  2.631452
##   0.84368421  3.236905  0.9861739  2.665276
##   0.89578947  3.272501  0.9859747  2.688664
##   0.94789474  3.309494  0.9857654  2.713317
##   1.00000000  3.347855  0.9855457  2.739890
##
## Tuning parameter 'alpha' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda = 0.06210526.
```

The code below was used to make predictions using the lasso model on the unseen test data and assess the accuracy using RMSE and r-squared.

```
# view optimal lambda value and extract coefficients [In-text citation: (Package 'glmnet', 2023)]
lasso_lambda <- lasso_fit$bestTune$lambda
lasso_coefficients <- coef(lasso_fit$finalModel, s = lasso_lambda)
lasso_coefficients
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                          s=0.06210526
## (Intercept)               37.12861460
## Children                  -0.73375757
## Age                        0.78209208
## Income                    -0.01161091
## MonthlyCharge             -0.51066220
## Bandwidth_GB_Year         26.31073372
## Marital_Married            .
## `Marital_Never Married`    .
## Marital_Separated          .
## Marital_Widowed            .
```

```
## Gender_Male              -0.79167982
## Gender_Nonbinary          .
## Techie_Yes                .
## `Contract_One year`       .
## `Contract_Two Year`       .
## Tablet_Yes                .
## DeviceProtection_Yes     -0.75245716
## TechSupport_Yes           .
## StreamingTV_Yes          -2.15065605
## StreamingMovies_Yes      -1.74776162
```

```r
# create predictions with LASSO model using test data [In-text citation: (Kuhn, n.d.)]
predictions <- predict(lasso_fit, newdata = test) %>%
  bind_cols(test) %>%
  rename_at('...1', ~'Tenure_pred')

# view RMSE and R squared
postResample(predictions$Tenure_pred,predictions$Tenure)
```

```
##      RMSE  Rsquared       MAE
## 2.5391856 0.9907533 2.3003371
```

E1, Accuracy

RMSE, or root mean squared error, is the square root of the average of squared differences between the predicted and actual values. RMSE is displayed on the same scale as the response variable, so in this case, it is in terms of tenure. A smaller value indicates there is a lower magnitude of error. R-squared, or the coefficient of determination, represents the proportion of variance in the target variable that the explanatory variables can explain. An r-squared value closer to one means that the model explains more of the variance (Task 2, n.d.). The model produced strong RMSE and r-squared values. The model produced a 2.558 RMSE and 0.9907 r-squared on the training data. When run on the test data, the model produced a 2.539 RMSE and 0.9907 r-squared value. These metrics indicate a strong predictive model, which performed better on the test data given the smaller RMSE.

E2, Results and Implications

The analysis aimed to develop a supervised machine learning model using lasso regression to predict the tenure of customers. This result was successful based on the model's RMSE and r-squared metrics. The model had a low RMSE, given the context of the response variable. The mean value of Tenure is 34.5 months, while the RMSE is 2.539. This is an acceptable value. In terms of r-squared, a 0.9907 value is very high. This indicates the model can explain nearly all the variance. This is undoubtedly a good first step. Additional models could be created with different hyper-tuning parameters to see if the metrics could be improved.

E3, Limitations

One limitation of the analysis is the trade-off in the lambda value. By selecting a lambda value that is too high, the model may penalize coefficients too much, which could lead to an underfitting model. Conversely, if the lambda value is too low, it could result in an overly complex model that does not perform well on the unseen data. A wide range of tuning values are needed to ensure the optimal value is used.

E4, Course of Action

The model produced strong RMSE and r-squared values, which indicate that it is a very predictive model. Ideally, a couple more models would be fit with different hyper-tuning parameters to see if the model could be improved. However, as it stands, the company could use the model to make predictions about the length of tenure of their customers. The model would be valuable for the company, as steps can be taken to retain customers with low predicted tenure.

F, Panopto Video

A Panopto video recording was created that covered the execution of the code. The video link can be found in the submission.

G, Sources for Code

Bobbitt, Z. (December 10, 2021). How to use the scale() function in R. Statology. Retrieved January 8, 2025, from (https://statology.org/scale-function-in-r/)

Kaplan, J. (November 28, 2020). Making dummy variables with dummy_cols(). fastDummies. Retrieved December 8, 2024, from (https://jacobkap.github.io/fastDummies/articles/making-dummy-variables.html)

Kuhn, M. (n.d.). Machine learning with caret in R [MOOC]. DataCamp. (https://app.datacamp.com/learn/courses/machine-learning-with-caret-in-r)

Package 'glmnet' (August 22, 2023). The Comprehensive R Archive Network. Retrieved January 18, 2025, from (https://cran.r-project.org/web/packages/glmnet/glmnet.pdf)

Schork, J. (n.d.). Correlation matrix in R. Statistics Globe. Retrieved December 10, 2024, from (https://statisticsglobe.com/correlation-matrix-in-r)

Simple Training/Test Set Splitting (n.d.). Rsample. Retrieved January 8, 2025, from (https://rsample.tidymodels.org/reference/initial_split.html)

Tierney, N. (n.d.). Dealing with Missing Data in R [MOOC]. DataCamp. (https://app.datacamp.com/learn/courses/dealing-with-missing-data-in-r)

WGU College of Information Technology (n.d.). Getting Started with Duplicates [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20

H, Sources for Content

Bobbitt, Z. (December 10, 2021). How to use the scale() function in R. Statology. Retrieved January 8, 2025, from (https://statology.org/scale-function-in-r/)

Cross validation in machine learning (n.d.). Geeks for Geeks. Retrieved January 13, 2025, from (https://www.geeksforgeeks.org/cross-validation-machine-learning/)

WGU College of Information Technology (n.d.). Task 2: Predictive Analysis [PowerPoint slides]. Western Governors University. (https://srm--c.vf.force.com/servlet/fileField?retURL=https%3A%2F%2Fsrm--c.vf.force.com%2Fapex%2FCourseArticle%3Fid%3DkA0S60000001DKzKAM%26groupId%3D%26searchTerm%3D%26courseCode%3DD209%26rtn%3D%252Fapex%252FCommonsExpandedSearch&entityId=ka0S60000006SzJIAU&_CONFIRMATIONTOKEN=VmpFPSxNakF5TIMwd01TMHlNRlF5TVRvMU9Eb3dPQzQxTXpCYSx3D%3D&common.udd.actions.ActionsUtilORIG_URI=%2Fservlet%2FfileField&field=FileUpload2__Body__s)

What is lasso regression? (May 15, 2024). Geeks for Geeks. Retrieved January 18, 2025, from (https://www.geeksforgeeks.org/what-is-lasso-regression/)