D212 Data Mining II - Task 1

Scott Babcock WGU - MS, Data Analytics

Created: March 23 2025

A1, Research Question

Can groups of customers with similar patterns and characteristics be identified using k-means clustering?

A2, Goals of Analysis

The analysis aims to develop an unsupervised machine-learning model using k-means clustering to identify groups of customers with similar characteristics. These clusters will then be compared to the churn variable to determine if there are any commonalities.

B1, K-means Technique

K-means is an unsupervised machine-learning algorithm that groups similar data points into clusters. It performs the grouping by picking an assigned number of centroids, referred to as k. Each data point must belong to one of the k clusters. The algorithm calculates the distance between each data point and the randomly picked centroids. The goal is to find the groups with the least distance from the centroid. It will iterate until the cluster assignments stop changing (Kumar, n.d.). The expected outcome is a cluster assignment for each record of data. The clusters should have some distinct defining qualities.

B2, K-Means Assumption

K-means assumes that the data is unlabeled. The unlabeled feature differentiates it from supervised machine learning techniques like KNN. The k-means technique is considered unsupervised because the algorithm finds the groups/similarities independently (Kumar, n.d.).

B3, Libraries and Packages

Three 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. Dplyr was also used to add features from other data frames. 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 k-means clustering was performed using the base package in R.

```
library(naniar)
library(dplyr)
library(purrr)
```

C1, Data Preprocessing

One data preprocessing step taken in the analysis was to scale the data. Because k-means relies on distances between data points, the data must be scaled so it does not give preference to larger numeric values. The scale function was used to achieve this across five continuous 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 analysis. These variables are all numeric and continuous:

- Age
- Income
- MonthlyCharge
- Outage_sec_perweek
- Bandwidth_GB_Year

C3, Steps for Analysis

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

- The 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 records and missing values. No duplicate records were identified. No missing values were identified.
- The subset of data used in the analysis was selected, and a new data frame was created.
- The variables were checked for outliers. Outliers were identified in the Income and Outage_sec_perweek variables but ultimately deemed reasonable.
- A new data frame was created where the values were centered and scaled to reduce bias.
- Summary statistics and standard deviation were viewed on each variable to verify that the scaling occurred. All had a mean of zero and a standard deviation of one.
- The cleaned and transformed data was written to a CSV file.

```
# load churn data into environment
churn <- read.csv("churn_clean.csv")

# view data types and structure
glimpse(churn)
```

```
## Rows: 10,000
## Columns: 50
## $ CaseOrder           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15~
## $ Customer_id         <chr> "K409198", "S120509", "K191035", "D90850", "K6627~
## $ Interaction         <chr> "aa90260b-4141-4a24-8e36-b04ce1f4f77b", "fb76459f~
## $ UID                 <chr> "e885b299883d4f9fb18e39c75155d990", "f2de8bef9647~
## $ City                <chr> "Point Baker", "West Branch", "Yamhill", "Del Mar~
## $ State               <chr> "AK", "MI", "OR", "CA", "TX", "GA", "TN", "OK", "~
## $ County              <chr> "Prince of Wales-Hyder", "Ogemaw", "Yamhill", "Sa~
## $ Zip                 <int> 99927, 48661, 97148, 92014, 77461, 31030, 37847, ~
## $ Lat                 <dbl> 56.25100, 44.32893, 45.35589, 32.96687, 29.38012,~
## $ Lng                 <dbl> -133.37571, -84.24080, -123.24657, -117.24798, -9~
## $ Population          <int> 38, 10446, 3735, 13863, 11352, 17701, 2535, 23144~
## $ Area                <chr> "Urban", "Urban", "Urban", "Suburban", "Suburban"~
## $ TimeZone            <chr> "America/Sitka", "America/Detroit", "America/Los_~
## $ Job                 <chr> "Environmental health practitioner", "Programmer,~
## $ Children            <int> 0, 1, 4, 1, 0, 3, 0, 2, 2, 1, 7, 2, 0, 5, 1, 3, 0~
## $ Age                 <int> 68, 27, 50, 48, 83, 83, 79, 30, 49, 86, 23, 56, 8~
## $ Income              <dbl> 28561.99, 21704.77, 9609.57, 18925.23, 40074.19, ~
## $ Marital             <chr> "Widowed", "Married", "Widowed", "Married", "Sepa~
```

```
## $ Gender               <chr> "Male", "Female", "Female", "Male", "Male", "Fema~
## $ Churn                <chr> "No", "Yes", "No", "No", "Yes", "No", "Yes", "Yes~
## $ Outage_sec_perweek   <dbl> 7.978323, 11.699080, 10.752800, 14.913540, 8.1474~
## $ Email                <int> 10, 12, 9, 15, 16, 15, 10, 16, 20, 18, 9, 17, 9, ~
## $ Contacts             <int> 0, 0, 0, 2, 2, 3, 0, 0, 2, 1, 0, 1, 0, 1, 3, 1, 1~
## $ Yearly_equip_failure <int> 1, 1, 1, 0, 1, 1, 1, 0, 3, 0, 2, 1, 0, 0, 0, 0, 0~
## $ Techie               <chr> "No", "Yes", "Yes", "Yes", "No", "No", "Yes", "Ye~
## $ Contract             <chr> "One year", "Month-to-month", "Two Year", "Two Ye~
## $ Port_modem           <chr> "Yes", "No", "Yes", "No", "Yes", "Yes", "No", "No~
## $ Tablet               <chr> "Yes", "Yes", "No", "No", "No", "No", "No", "No",~
## $ InternetService      <chr> "Fiber Optic", "Fiber Optic", "DSL", "DSL", "Fibe~
## $ Phone                <chr> "Yes", "Yes", "Yes", "Yes", "No", "Yes", "Yes", "~
## $ Multiple             <chr> "No", "Yes", "Yes", "No", "No", "Yes", "No", "No"~
## $ OnlineSecurity       <chr> "Yes", "Yes", "No", "Yes", "No", "Yes", "No", "No~
## $ OnlineBackup         <chr> "Yes", "No", "No", "No", "No", "Yes", "No", "Yes"~
## $ DeviceProtection     <chr> "No", "No", "No", "No", "No", "Yes", "No", "No", ~
## $ TechSupport          <chr> "No", "No", "No", "No", "Yes", "No", "Yes", "No",~
## $ StreamingTV          <chr> "No", "Yes", "No", "Yes", "Yes", "No", "Yes", "No~
## $ StreamingMovies      <chr> "Yes", "Yes", "Yes", "No", "No", "Yes", "Yes", "N~
## $ PaperlessBilling     <chr> "Yes", "Yes", "Yes", "Yes", "No", "No", "No", "Ye~
## $ PaymentMethod        <chr> "Credit Card (automatic)", "Bank Transfer(automat~
## $ Tenure               <dbl> 6.795513, 1.156681, 15.754144, 17.087227, 1.67097~
## $ MonthlyCharge        <dbl> 172.45552, 242.63255, 159.94758, 119.95684, 149.9~
## $ Bandwidth_GB_Year    <dbl> 904.5361, 800.9828, 2054.7070, 2164.5794, 271.493~
## $ Item1                <int> 5, 3, 4, 4, 4, 3, 6, 2, 5, 2, 4, 4, 1, 5, 3, 3, 3~
## $ Item2                <int> 5, 4, 4, 4, 4, 3, 5, 2, 4, 2, 4, 4, 2, 6, 3, 3, 4~
## $ Item3                <int> 5, 3, 2, 4, 4, 3, 6, 2, 4, 2, 4, 3, 1, 5, 4, 3, 4~
## $ Item4                <int> 3, 3, 4, 2, 3, 2, 4, 5, 3, 2, 7, 4, 4, 2, 2, 2, 3~
## $ Item5                <int> 4, 4, 4, 5, 4, 4, 1, 2, 4, 5, 3, 4, 3, 4, 3, 4, 5~
## $ Item6                <int> 4, 3, 3, 4, 4, 3, 5, 3, 3, 2, 3, 4, 2, 5, 4, 3, 4~
## $ Item7                <int> 3, 4, 3, 3, 4, 3, 5, 4, 4, 3, 3, 3, 3, 4, 4, 5, 4~
## $ Item8                <int> 4, 4, 3, 3, 5, 3, 5, 5, 4, 3, 4, 4, 3, 4, 2, 2, 3~
```

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

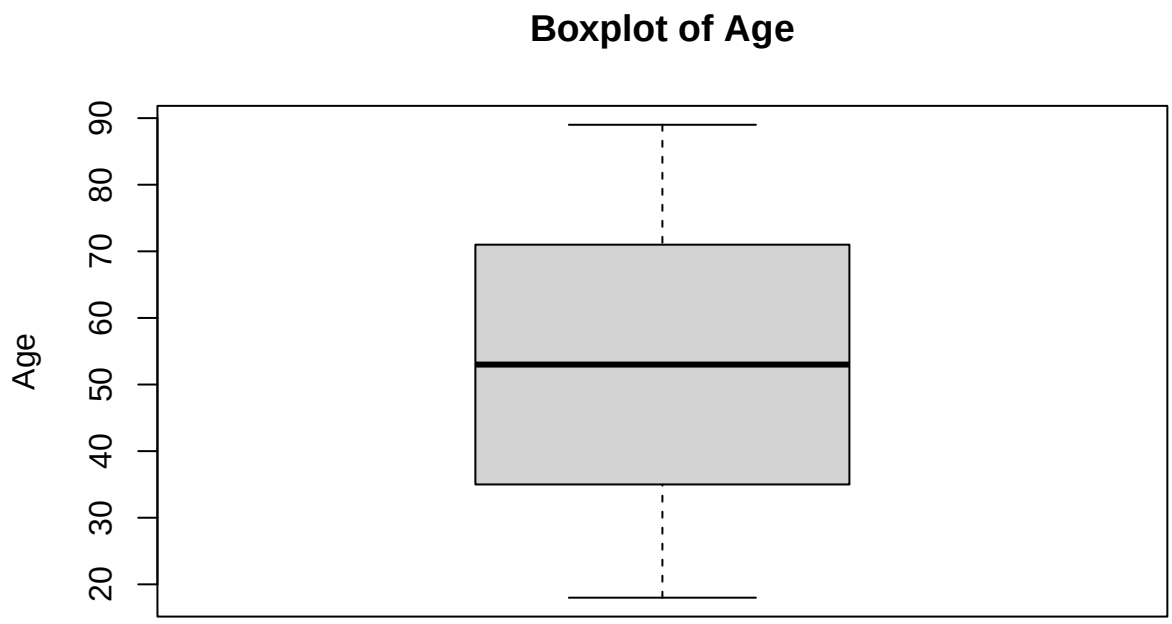
```
## [1] 0
```

```
miss_var_summary(churn)
```

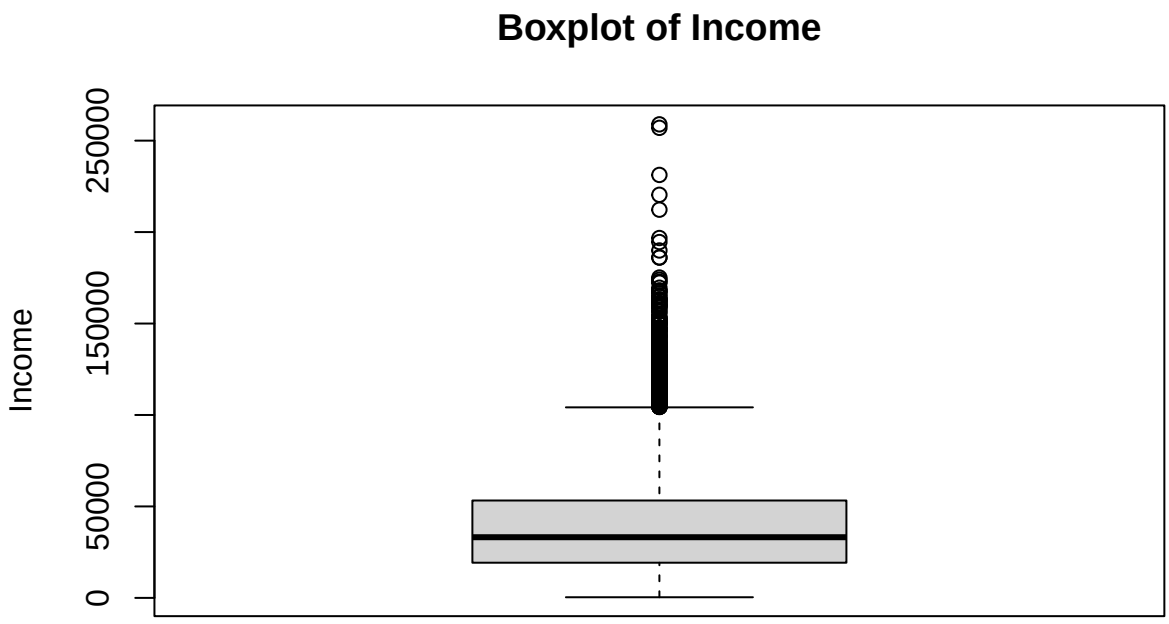
```
## # A tibble: 50 x 3
##    variable    n_miss pct_miss
##    <chr>        <int>    <num>
##  1 CaseOrder        0        0
##  2 Customer_id      0        0
##  3 Interaction      0        0
##  4 UID              0        0
##  5 City             0        0
##  6 State            0        0
##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

```
##  9 Lat              0        0
## 10 Lng              0        0
## # i 40 more rows
```

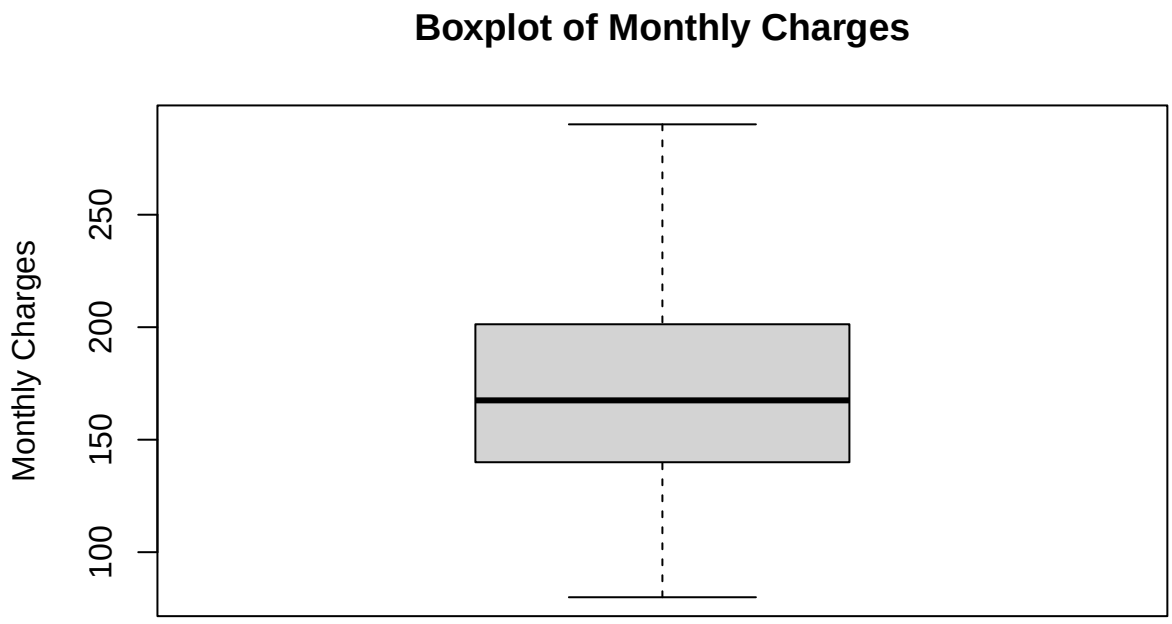
```
# create data frame with continuous numeric variables
churn_numeric <-
  churn %>%
    select(
      Age,
      Income,
      MonthlyCharge,
      Outage_sec_perweek,
      Bandwidth_GB_Year
    )

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



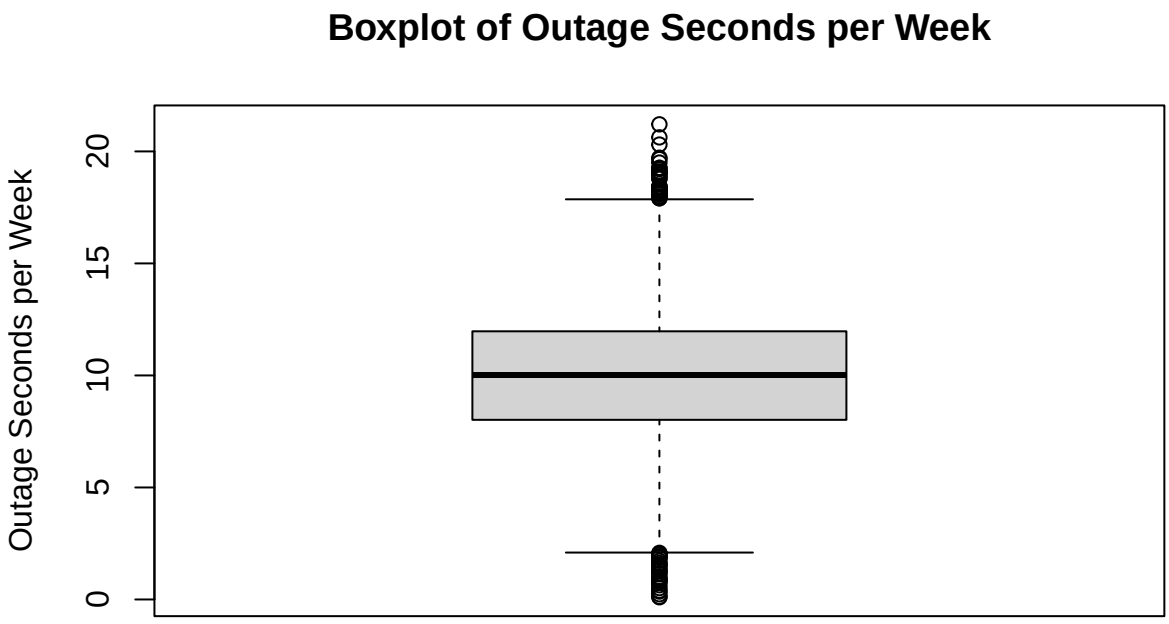

```
boxplot(churn_numeric$Income,
        ylab = "Income",
        main = "Boxplot of Income")
```



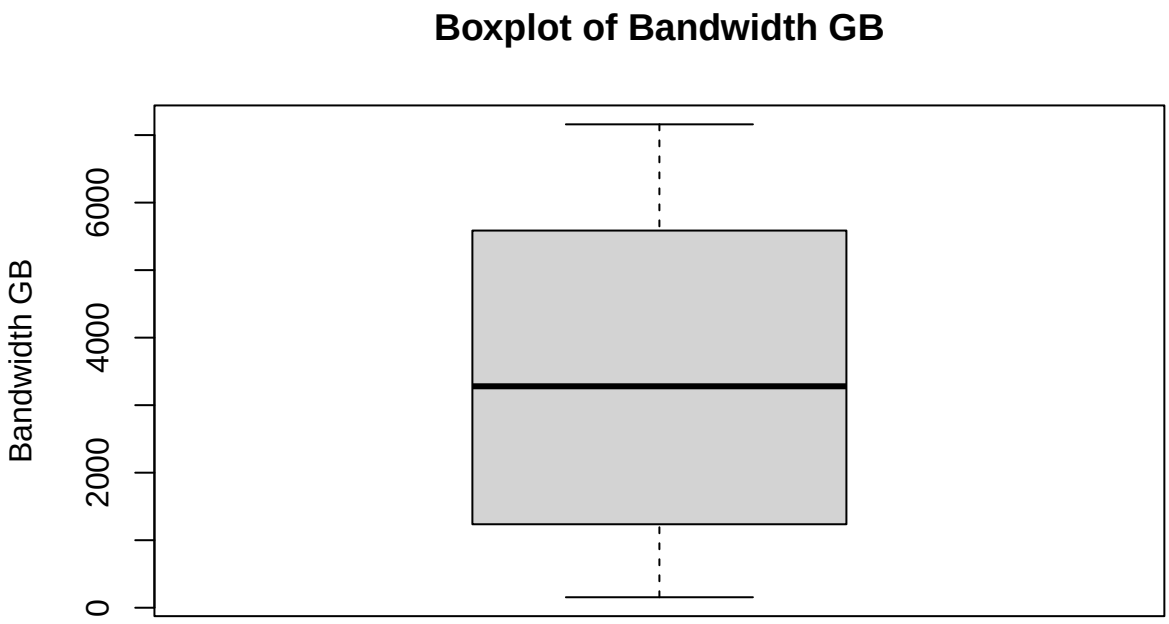

```
boxplot(churn_numeric$MonthlyCharge,
        ylab = "Monthly Charges",
        main = "Boxplot of Monthly Charges")
```




```
boxplot(churn_numeric$Outage_sec_perweek,
        ylab = "Outage Seconds per Week",
        main = "Boxplot of Outage Seconds per Week")
```




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




```
# create data frame with scaled variables [In-text citation: (Bobbitt, 2021)]
churn_numeric_scale <-
  churn_numeric %>%
    scale() %>%
    data.frame()

# view mean and standard deviation of each variable to confirm scaling took place
summary(churn_numeric_scale$Age)
```

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

```
sd(churn_numeric_scale$Age)
```

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

```
summary(churn_numeric_scale$Income)
```

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

```
sd(churn_numeric_scale$Income)
```

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

```r
summary(churn_numeric_scale$MonthlyCharge)
```

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

```r
sd(churn_numeric_scale$MonthlyCharge)
```

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

```r
summary(churn_numeric_scale$Outage_sec_perweek)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.327298 -0.666539  0.005616  0.000000  0.661164  3.765225
```

```r
sd(churn_numeric_scale$Outage_sec_perweek)
```

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

```r
summary(churn_numeric_scale$Bandwidth_GB_Year)
```

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

```r
sd(churn_numeric_scale$Bandwidth_GB_Year)
```

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

```r
# write cleaned and transformed data to csv
write.csv(churn_numeric_scale,
          "d212_task1_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, Optimal k

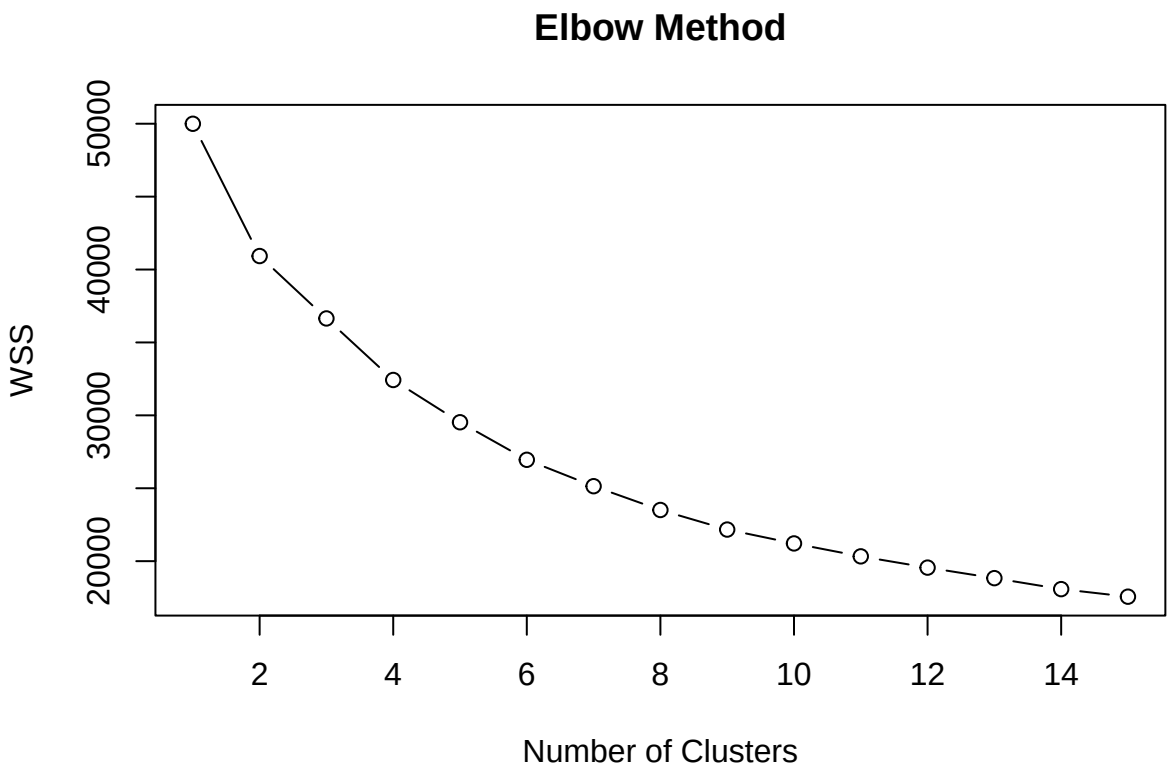
The k-means algorithm requires a value for k, which is the number of groups to be created within the dataset. One can arbitrarily assign a value for k, but it may not provide the best output. The elbow and silhouette methods are two common ways to identify the optimal k value. This analysis chose the elbow method to determine the optimal k value.

The within-cluster-sum of squared errors (WSS) is calculated for a range of k values in the elbow method. WSS is the sum of the squared errors for all points in the dataset relative to the predicted cluster center.

The WSS value is then plotted for each k value, and the optimal k value can be found by identifying where the WSS value starts to diminish, often visible as an elbow (Mahendru, 2019).

To find the optimal k value, a range from 1:15 was tested and plotted. The plot below shows that the elbow appears most distinct at a k value of two. It begins to flatten out for each ensuing value of k. Therefore, a k value of two was used in the analysis.

```
# set random seed for replication
set.seed(44)

# elbow method
# initialize total within sum of squares error [In-text citation: (Roark, n.d.)]
wss <- 0

# for 1 to 15 cluster centers [In-text citation: (Roark, n.d.)]
for(i in 1:15) {
  km_elbow <- kmeans(churn_numeric_scale, centers = i, nstart = 20)
  #save total within sum of squares to wss variable
  wss[i] <- km_elbow$tot.withinss
}

# plot total withinss vs num clusters [In-text citation: (Roark, n.d.)]
plot(1:15, wss, type = "b",
     xlab = "Number of Clusters",
     ylab = "WSS",
     main = "Elbow Method")
```



D2, Clustering Code

The following code was used to perform the k-means clustering. As shown above, the set.seed() function was used for replication purposes. A WSS variable with a zero value was created to be used in the loop code. A k-means model was built for each value of k ranging from 1:15. The model loops through and assigns the within-cluster-sum of squared error value for each respective k value. A plot was then created to show the WSS for each k value. Once the optimal k value was determined, two, in this case, a final model was created.

```
# set k equal to elbow
k <- 2

# build model with k clusters [In-text citation: (Roark, n.d.)]
km_final <- kmeans(churn_numeric_scale, centers = k, nstart = 20, iter.max = 50)
```

E1, Cluster Quality

Because k-means aims to identify groups with similarities and not necessarily a target outcome, there is no accuracy metric on which to base the quality. However, by looking at the means of each variable within the clusters, how the algorithm segments the data becomes apparent. Four of the five variables have nearly identical means in the two groups. Bandwidth_GB_Year is the primary driver of the two clusters. The quality of the clusters is acceptable based on this distinction. However, the remaining four variables may not be necessary in the model.

E2, Results and Implications

After running the k-means model, the clusters were appended to the original data to begin analyzing them. The churn variable was also brought in to provide additional context. The data was grouped and summarized by cluster to identify patterns among the two groups. The count of records was also calculated, along with the count of churned customers and the respective churn rate of each cluster.

It was determined that the primary distinction between the clusters was the Bandwidth_GB_Year variable. Cluster 1 could be considered low-bandwidth customers, and Cluster 2 could be considered high-bandwidth customers. Further, the pattern becomes more apparent when assessing the count of churned customers and churn rate within the two clusters. The group of heavy bandwidth users, which is comprised of nearly 5k customers, has a significantly lower churn rate compared to the group with lower bandwidth usage.

```
# bind clusters and churn variable to original cluster data
churn_numeric <-
  churn_numeric %>%
    mutate(
      Cluster = km_final$cluster,
      Churn = churn$Churn
    )

# Calculate the means for each cluster and summarize churn counts
churn_numeric %>%
  group_by(Cluster) %>%
  summarise(
    across(where(is.numeric),mean),
    cluster_count = n(),
    churn_count = sum(Churn == "Yes"),
```

```
    churn_pct = sum(Churn == "Yes")/n()
  )
```

```
## # A tibble: 2 x 9
##   Cluster   Age Income MonthlyCharge Outage_sec_perweek Bandwidth_GB_Year
##     <int> <dbl>  <dbl>         <dbl>              <dbl>             <dbl>
## 1       1  52.7 39756.          173.              10.00             1316.
## 2       2  53.5 39858.          173.              10.0              5478.
## # i 3 more variables: cluster_count <int>, churn_count <int>, churn_pct <dbl>
```

E3, Limitations

One limitation of the analysis could be the methodology used in the data cleaning. If there were initially missing values in the raw data and they were imputed using mean or median, it could render the clusters less valid. The imputation strategy could explain the slight variance between the cluster means in four of the five variables.

E4, Course of Action

The analysis aimed to identify groups of customers with similar characteristics using the k-means algorithm. Two clusters were identified within the five variables, and the distinction between them was driven primarily by the bandwidth used. As an additional step, the count of churned customers and churn rate were analyzed for each cluster, and there was a clear pattern between the two groups. The group of customers who used more bandwidth had a significantly lower churn rate than those with lower bandwidth usage.

The takeaway is that the company needs to get customers to engage more with the product and, in this case, use more bandwidth. The usage may help the customers realize the value they get by paying for the product. Said differently, customers who are not heavy bandwidth users may not want to pay a premium for the service and will seek alternative options. Additional analysis should be run to determine what features correlate with bandwidth usage.

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/)

Roark, H. (n.d.). Unsupervised learning in R [MOOC]. DataCamp. (https://app.datacamp.com/learn/courses/unsupervised-learning-in-r)

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

Kumar, V. (n.d.). What is k-means algorithm and how it works. TowardsMachineLearning. Retrieved March 23, 2025, from (https://towardsmachinelearning.org/k-means/)

Mahendru, K. (June 17, 2019). How to determine the optimal value for k-means. Medium. Retrieved March 23, 2025, from (https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb)