D212 Data Mining II - Task 2

Scott Babcock WGU - MS, Data Analytics

Created: April 3 2025

Load Libraries

```
library(naniar)
library(dplyr)
library(lattice)
library(reshape2)
library(ggplot2)
```

Data Load & Initial Exploration

```
# 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~
```

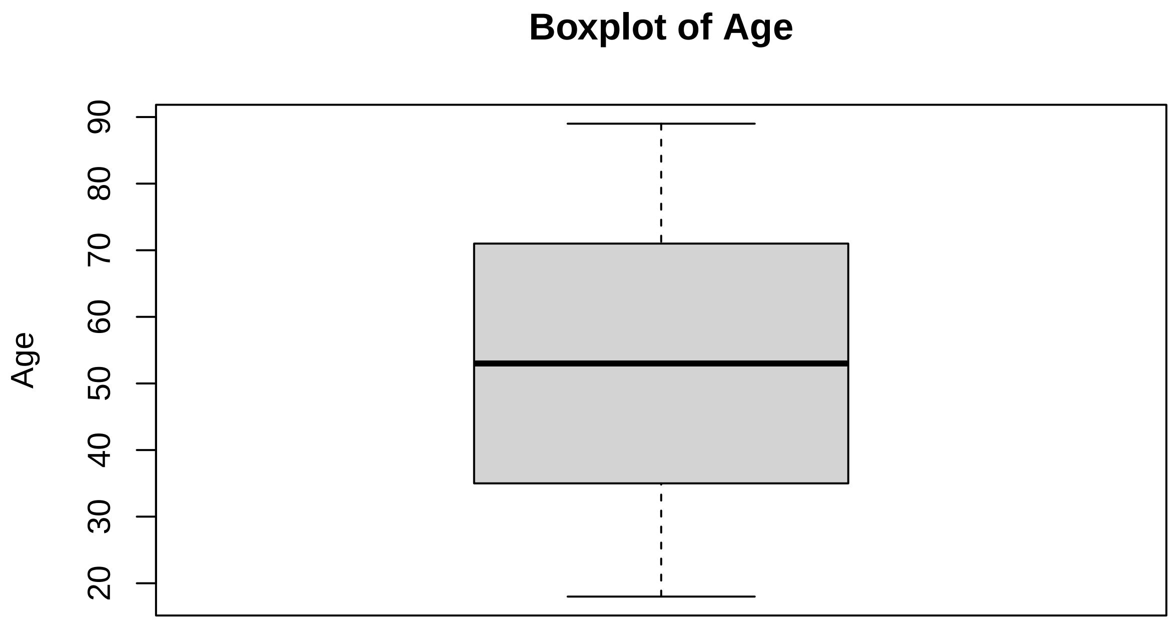
```
# check for duplicate records and missing values
#  [In-text citation:(Getting Started with Duplicates, n.d.)]
#  [In-text citation: (Tierney, n.d.)]
sum(duplicated(churn))
```

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

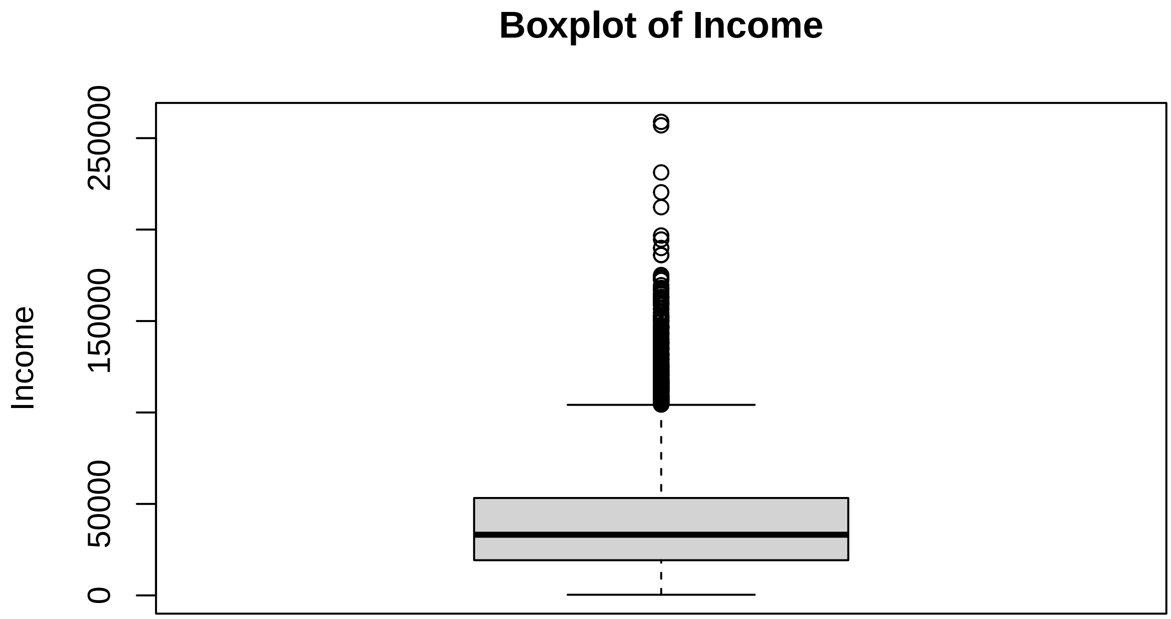
```
miss_var_summary(churn)
```

```
## # 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
```

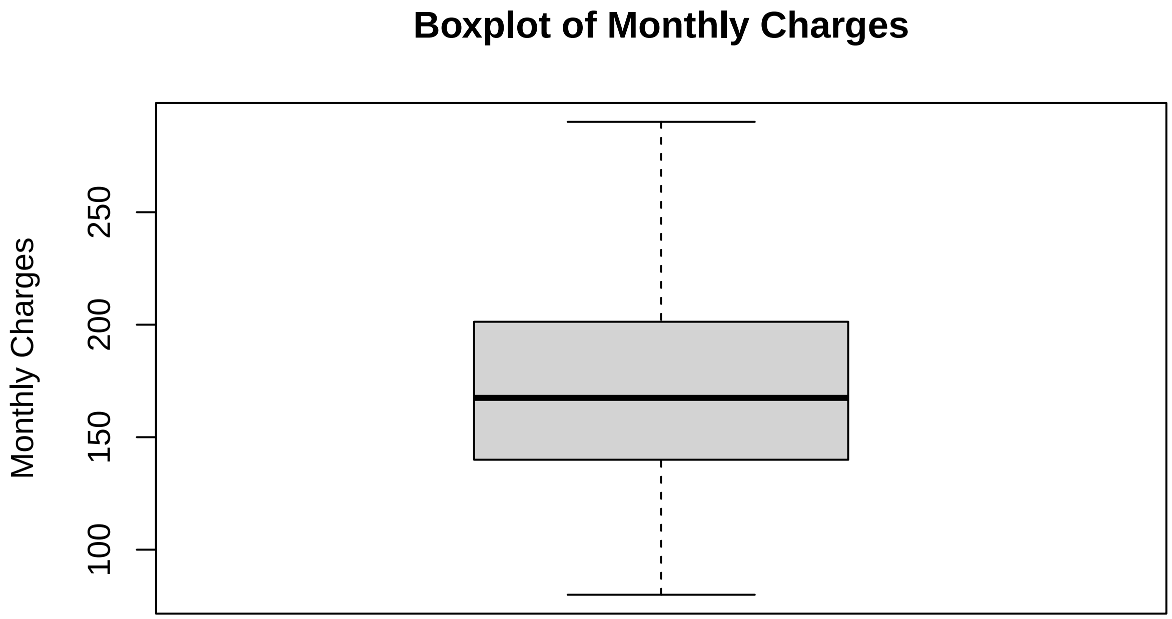
```
# create data frame with continuous numeric variables
churn_numeric <-
  churn %>%
  select(
    Age,
    Income,
    MonthlyCharge,
    Outage_sec_perweek,
    Bandwidth_GB_Year,
    Tenure,
    Lat,
    Lng
  )

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



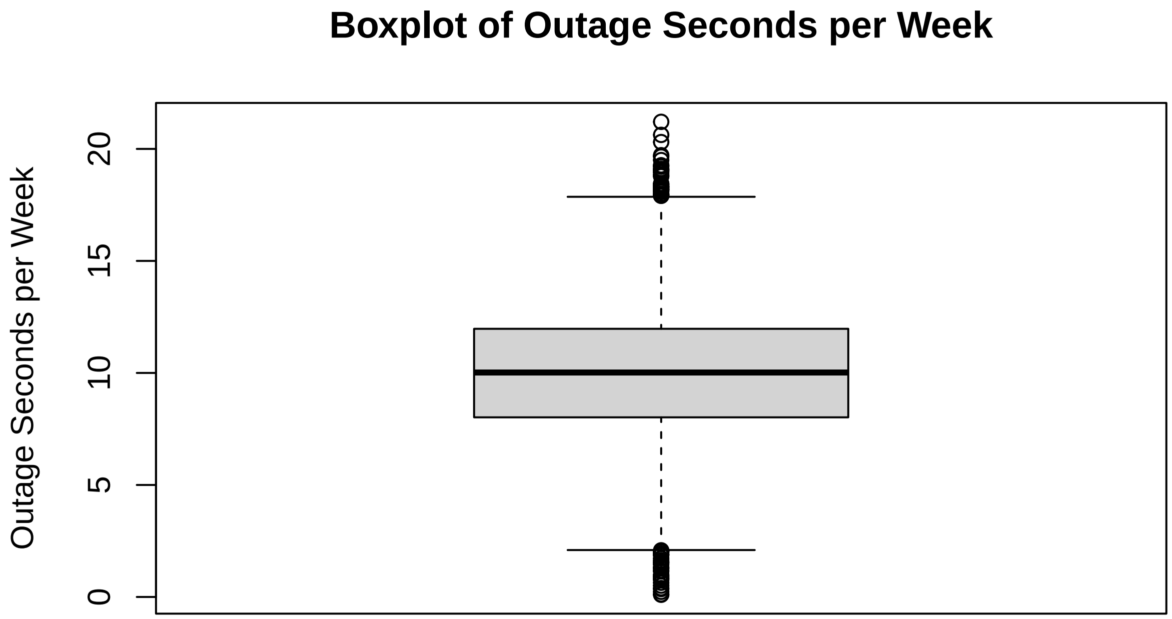
```
boxplot(churn_numeric$Income,
        ylab = "Income",
        main = "Boxplot of Income")
```



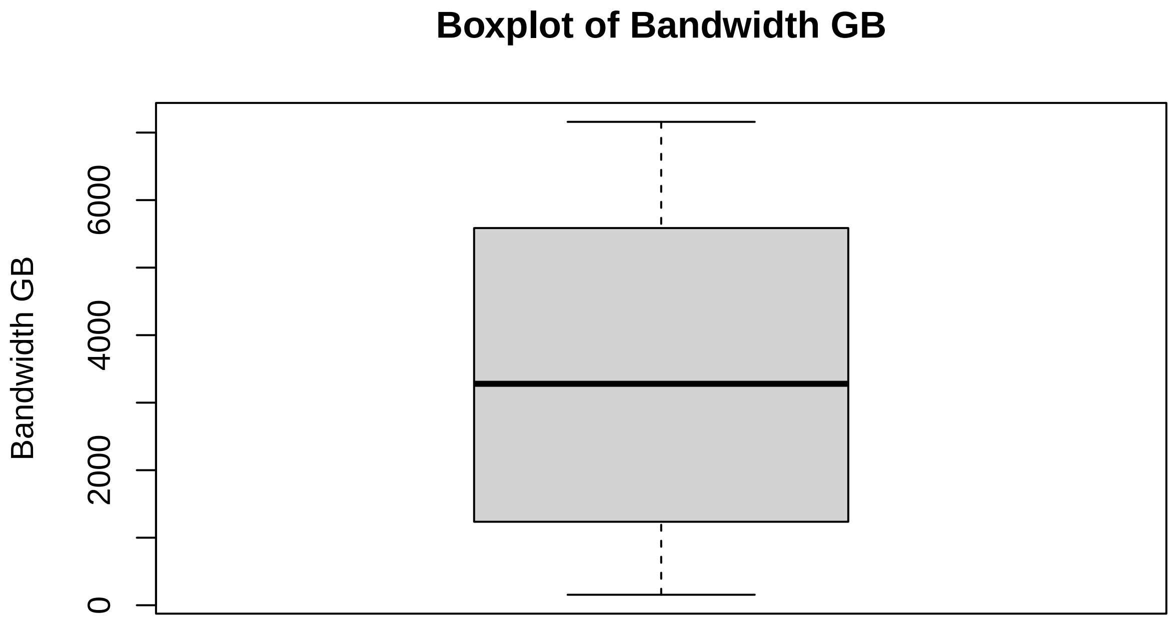
```
boxplot(churn_numeric$MonthlyCharge,
        ylab = "Monthly Charges",
        main = "Boxplot of Monthly Charges")
```



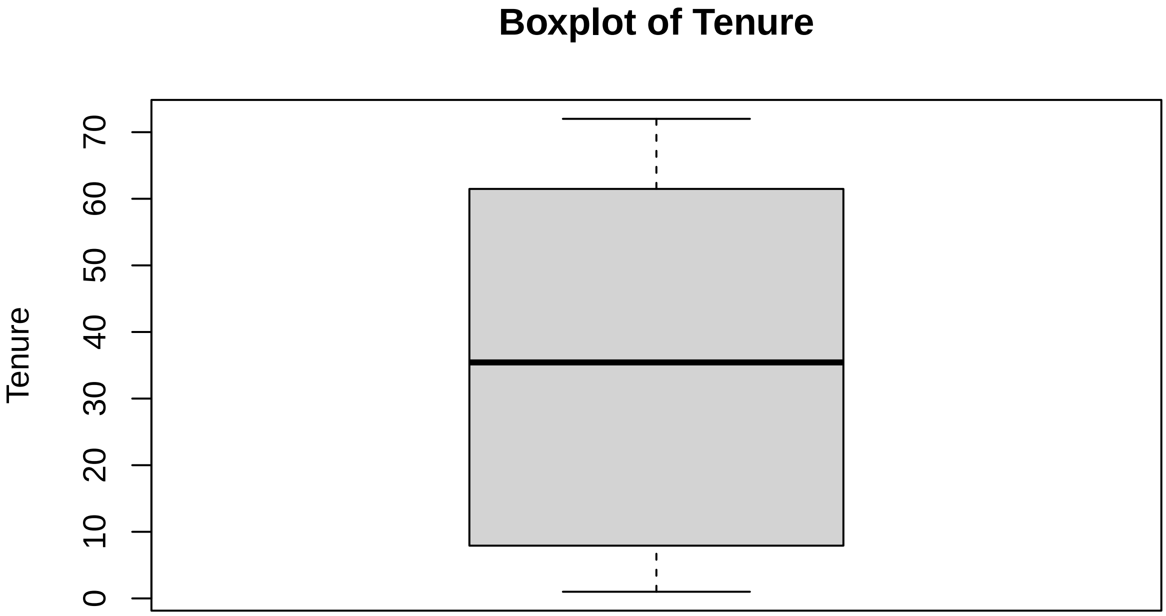
```r
boxplot(churn_numeric$Outage_sec_perweek,
        ylab = "Outage Seconds per Week",
        main = "Boxplot of Outage Seconds per Week")
```



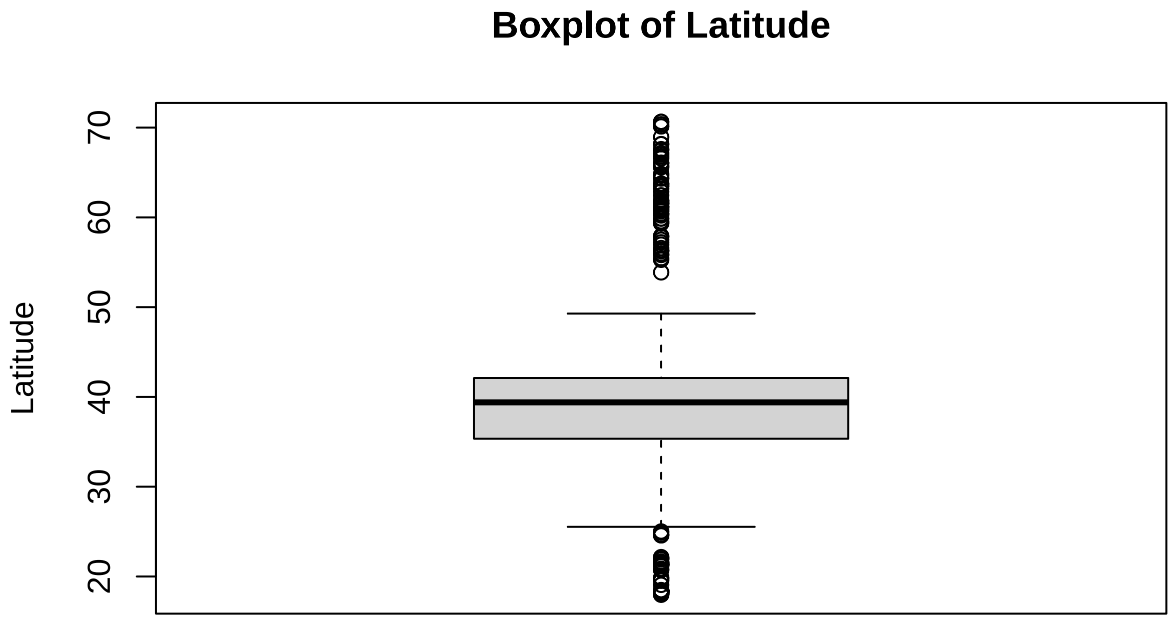
```r
boxplot(churn_numeric$Bandwidth_GB_Year,
        ylab = "Bandwidth GB",
        main = "Boxplot of Bandwidth GB")
```



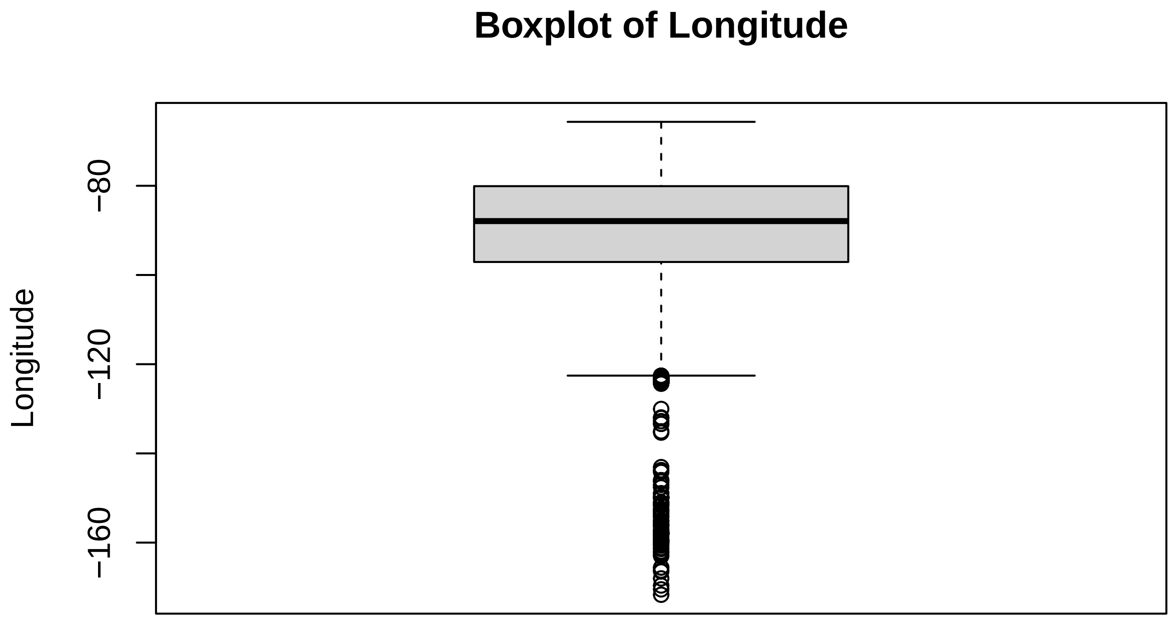
```
boxplot(churn_numeric$Tenure,
        ylab = "Tenure",
        main = "Boxplot of Tenure")
```



```r
boxplot(churn_numeric$Lat,
        ylab = "Latitude",
        main = "Boxplot of Latitude")
```



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



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

```
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
```

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

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

```
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
```

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

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

```
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
```

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

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

```
summary(churn_numeric_scale$Tenure)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -1.2679 -1.0063  0.0342  0.0000  1.0193  1.4171
```

```
sd(churn_numeric_scale$Tenure)
```

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

```
summary(churn_numeric_scale$Lat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -3.8238 -0.6282  0.1174  0.0000  0.6160  5.8637
```

```r
sd(churn_numeric_scale$Lat)
```

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

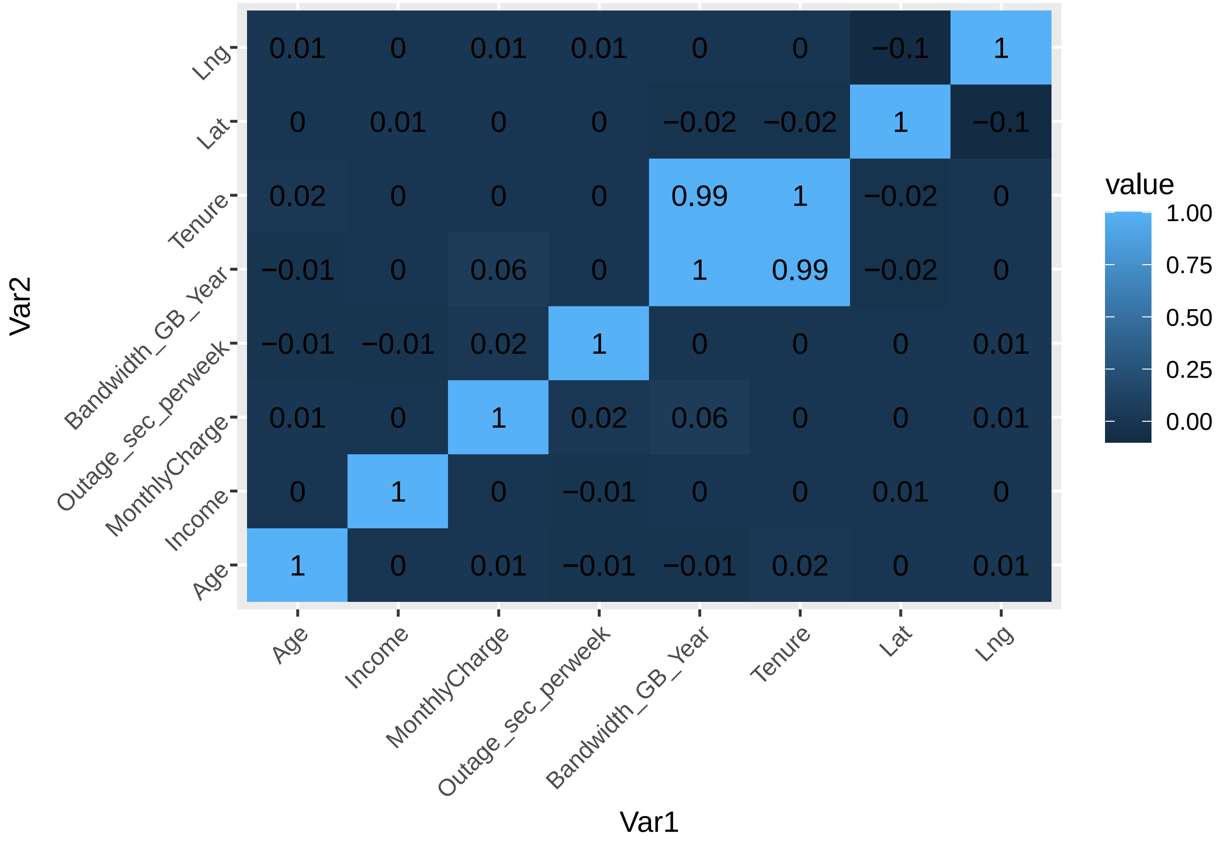
```r
summary(churn_numeric_scale$Lng)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -5.3381 -0.4157  0.1889  0.0000  0.7056  1.6571
```

```r
sd(churn_numeric_scale$Lng)
```

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

```r
# create covariance matrix [In-text citation: (Geeks for Geeks, 2022)]
corr_mat <- round(cor(churn_numeric_scale),2) %>%
  melt()

ggplot(corr_mat, aes(x = Var1, y = Var2, fill = value))+
  geom_tile()+
  geom_text(aes(Var2,Var1, label = value),color = "black", size = 4)+
  theme(axis.text = element_text(angle = 45, hjust = 1))
```



```r
# write cleaned and transformed data to csv
write.csv(churn_numeric_scale,
          "d212_task2_babcock_churn_transformed.csv",
          row.names = FALSE)
```

A1, Research Question

How effectively can principal component analysis reduce the dimensionality of the Churn dataset while preserving variance?

A2, Goals of Analysis

The analysis aims to perform principle component analysis on a wide range of continuous numeric variables to reduce the dimensionality of the dataset while maintaining as much variance as possible.

B1, PCA Technique

Principal Component Analysis (PCA) is an unsupervised machine learning technique that aims to reduce the dimensionality of a dataset. Often, with larger datasets, features are highly correlated and can influence the results if included in modeling. It can be challenging to visualize and spot patterns among a dataset with many features as well. PCA creates groupings of features, called components, that remove highly correlated data and retain as much of the variance in the data as possible. The expected result is several principle components of decreasing importance. For example, PC1 will account for the most variance in the dataset, and so on. PCA is typically performed as a pre-processing step before regression analysis or clustering (Frost, n.d.).

B2, PCA Assumption

Principal Component Analysis assumes that the components with the highest variance are the most important, or informative. The first principal component will always explain the most variance in the dataset. The ordering of the components in terms of importance, or variance explained, allows for the exclusion of components of low importance which reduces the dimensionality of the dataset (Amit, 2024).

C1, Identification of Variables

The following variables were used in the analysis. All of the variables are numeric and continuous.

- Age
- Income
- MonthlyCharge
- Outage_sec_perweek
- Bandwidth_GB_Year
- Tenure
- Lat (Latitude)
- Lng (Longitude)

C2, Standardization

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

D1, Principle Component Matrix

Once the data were standardized, PCA was performed on the variable set. The prcomp function created the principal components, which were stored in a new data frame. The print function was used to call the principle component matrix, as seen below. The result is a matrix with the variables used in rows and their contribution to each principle component in the columns. For example, PC1 is primarily influenced by Bandwidth_GB_Year and Tenure with weightings of -0.706 and -0.705, respectively.

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

# set up pca model [In-text citation: (Roark, n.d.)]
pca_initial <- prcomp(churn_numeric_scale,
                  scale. = FALSE,
                  center = FALSE)

# view weightings in each pca
pca_initial
```

```
## Standard deviations (1, .., p=8):
## [1] 1.41201448 1.04997680 1.01217074 1.00267206 0.99896312 0.98620592 0.94734602
## [8] 0.07700025
##
## Rotation (n x k) = (8 x 8):
##                             PC1         PC2           PC3           PC4
## Age                -0.002128760 -0.11224920  0.0702047415 -0.8939832717
## Income             -0.003741723  0.07464781 -0.3320622100 -0.0008919647
## MonthlyCharge      -0.040791581 -0.07974471  0.6441911207 -0.2456774421
## Outage_sec_perweek -0.005767541 -0.01939603  0.6714645129  0.3525393122
## Bandwidth_GB_Year  -0.706922372  0.01707399  0.0008937992  0.0113874344
## Tenure             -0.705626231  0.01903517 -0.0374892749 -0.0019085476
## Lat                 0.023938172  0.69188257  0.1321845656 -0.1185786756
## Lng                -0.007974697 -0.70408597 -0.0089547108  0.0442439116
##                              PC5         PC6          PC7           PC8
## Age                -0.0729626550 -0.41651070 -0.062664066  0.0230063899
## Income              0.9114041143 -0.22847831 -0.035867193 -0.0011583550
## MonthlyCharge       0.3878344928  0.59860745 -0.076242211 -0.0455837724
## Outage_sec_perweek  0.0828116602 -0.64287290 -0.065563648  0.0002183469
## Bandwidth_GB_Year   0.0003816361  0.01244832  0.008296653  0.7068343691
## Tenure             -0.0284516957 -0.03965380  0.012768278 -0.7055318875
## Lat                 0.0146082874 -0.01746318  0.699051011  0.0010659118
## Lng                 0.0756477354 -0.02584025  0.704110157  0.0004800039
```

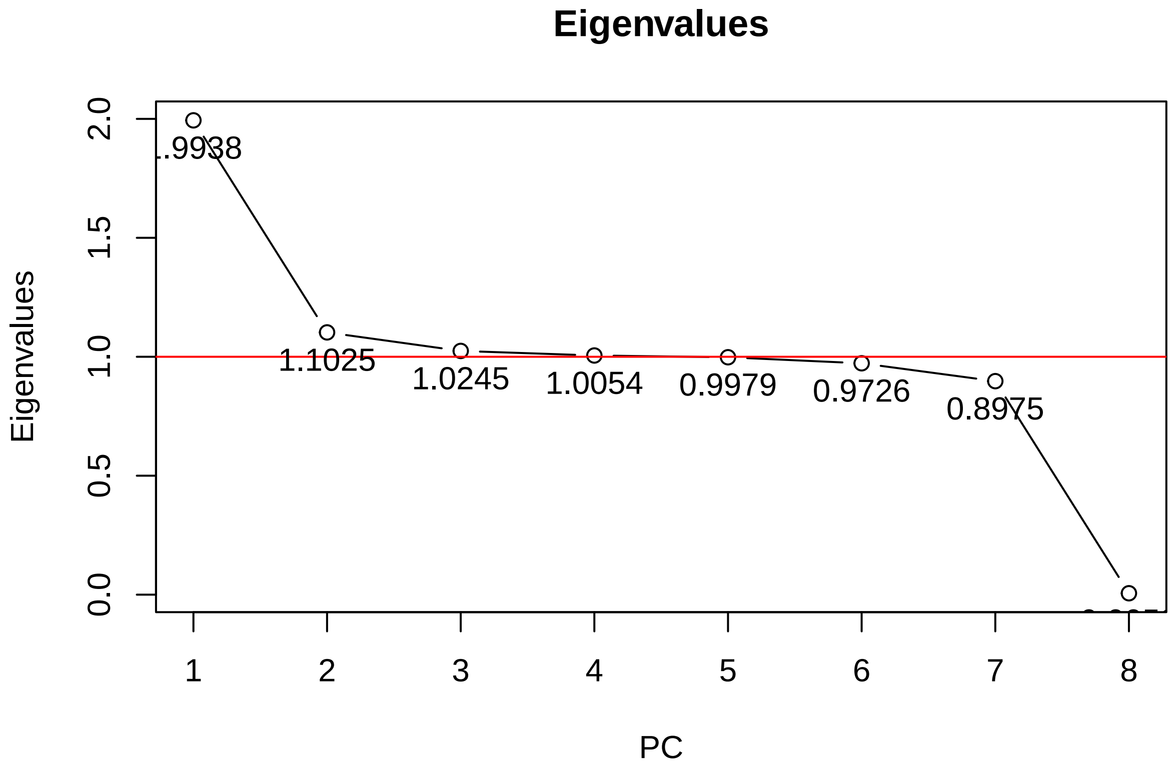
D2, Total Principle Components

The Kaiser Criterion is one method to identify the number of principal components needed. The Kaiser Criterion suggests keeping principal components with an eigenvalue greater than one. The technique reduces dimensionality while maintaining the essential components (What is the Kaiser Criterion, n.d). The standard deviation of each principal component was squared to calculate eigenvalues. These values were then plotted with a reference line at one to identify the principal components to keep easily. Overall, 8 principal components were created. Based on the Kaiser Criterion, PCs one through four would be kept, with the values beyond that falling below one. Another way to identify the number of principal components to keep is to plot the proportion of variance explained by each PC and keep the number of PCs defined by the elbow. The scree plot below would suggest keeping two principal components based on where the elbow occurs. One could argue that seven PCs should be kept as another elbow occurs there. Using the elbow method, keeping the first two PCs would capture 38.7% of the variance. Keeping the four principal components indicated by the Kaiser Criterion would result in roughly 64% of the total variance explained, which is the preferred option in this case.

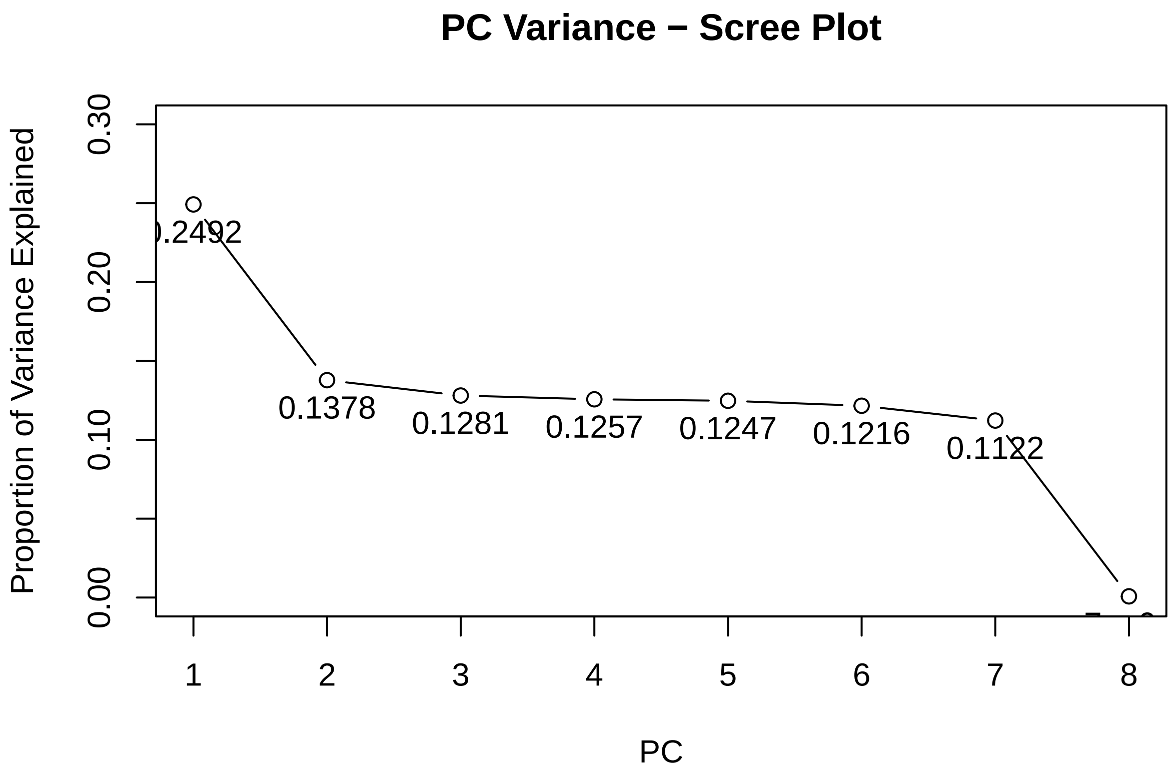
```
# calculate variance and proportion [In-text citation: (Roark, n.d.)]
pca_var <- pca_initial$sdev^2
pve <- pca_var/sum(pca_var)
print(pca_var)
```

```
## [1] 1.993784906 1.102451276 1.024489603 1.005351264 0.997927322 0.972602110
## [7] 0.897464480 0.005929038
```

```
# plot eigenvalues and variance [In-text citation: (Roark, n.d.)]
plot(pca_var, xlab = "PC", ylab = "Eigenvalues",main = "Eigenvalues",
                   type = "b")
    abline(h=1, col = "red")
    text(x = pca_var, labels = round(pca_var,4),pos = 1)
```



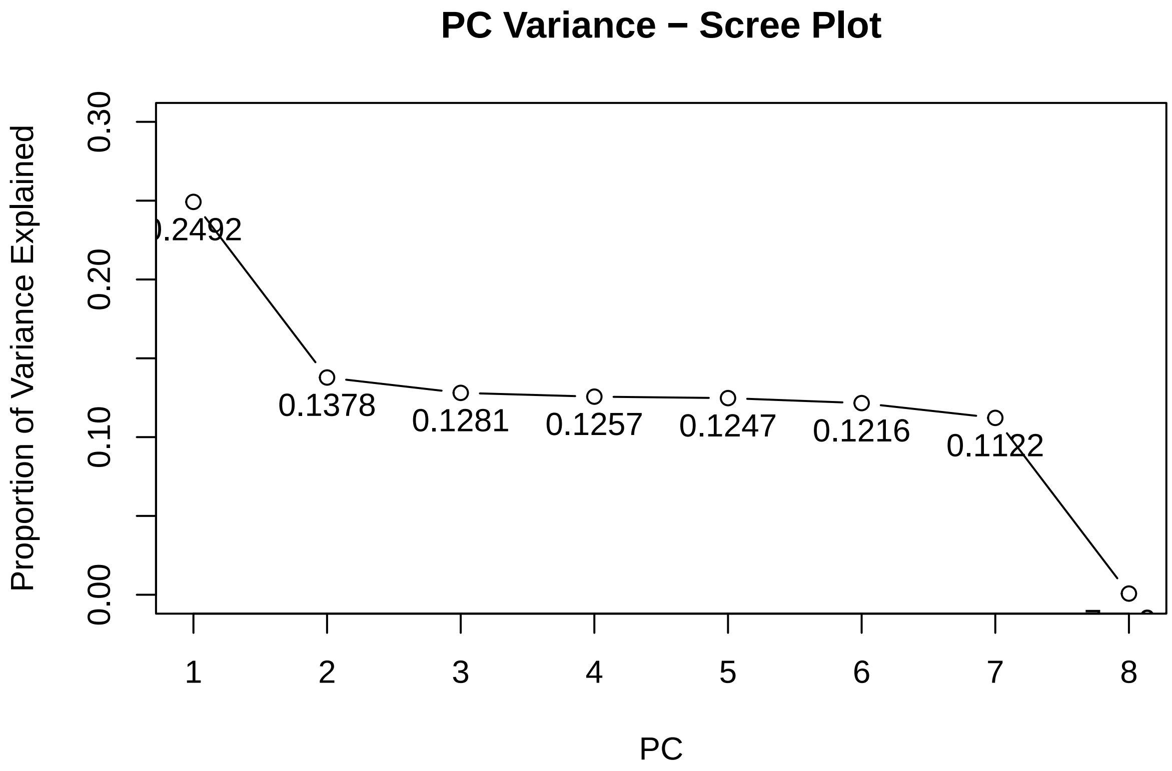
```
plot(pve, xlab = "PC", ylab = "Proportion of Variance Explained",main = "PC Variance - Scree Plot",
                 ylim = c(0,0.3),
                 type = "b")
    text(x = pve, labels = round(pve,4),pos = 1)
```



D3, Variance of Principle Components

The calculated variance is divided by the total variance to determine the proportion of variance each principal component explains. The plot below shows that PC1 is the most significant component, accounting for 24.92% of the variance. PC2 accounts for 13.78% of the variance, and PC3 through PC11 are all fairly even, ranging from 12.8% to 12.2%. PC7 and PC8 begin to dip, with PC7 accounting for 11.2% of the variance and PC8 barely above zero.

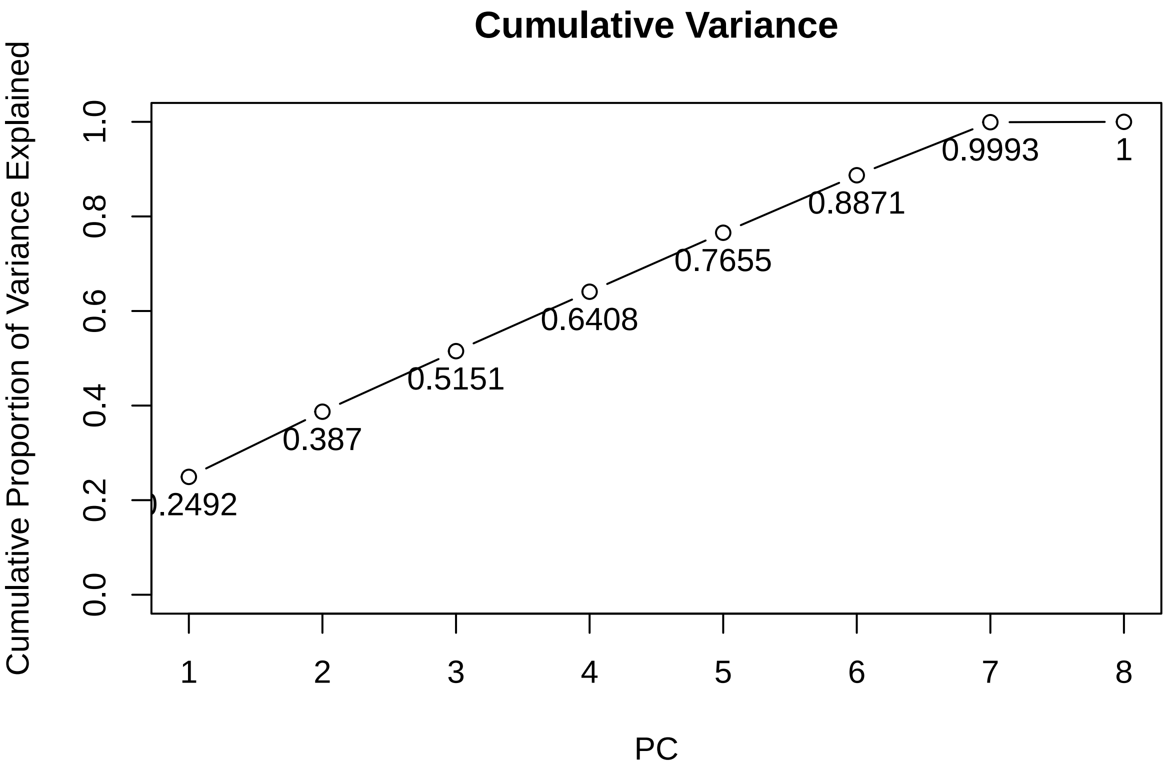
```
plot(pve, xlab = "PC", ylab = "Proportion of Variance Explained",main = "PC Variance - Scree Plot",
                 ylim = c(0,0.3),
                 type = "b")
    text(x = pve, labels = round(pve,4),pos = 1)
```



D4, Total Variance of Principle Components

The cumulative sum of the proportion of variance explained is calculated to identify the total proportion of variance explained by each principal component. PC1 will show the same proportion of variance as previously shown since it is the starting point, and the proportion of variance will be added on for each ensuing principal component until 100% of the variance is explained at PC8. As previously stated, the Kaiser Criterion recommended keeping four principal components, which would account for roughly 64% of the total variance. The two PCs suggested by the first elbow in the previous scree plot would explain just 38.7% of the total variance, while the second elbow at PC7 would result in capturing about 99% of the total variance.

```
plot(cumsum(pve), xlab = "PC", ylab = "Cumulative Proportion of Variance Explained",main = "Cumulative
                    ylim = c(0,1),
                    type = "b")
    text(x = cumsum(pve), labels = round(cumsum(pve),4),pos = 1)
```



D5, Results Summary

The goal of the analysis was to determine if principal component analysis could be used to reduce the dimensionality of the dataset while preserving as much of the variance as possible. The analysis performed PCA on 8 continuous numeric variables. Two methods were used to determine the number of principal components to keep. The Kaiser Criterion suggested keeping four principal components with eigenvalues greater than one. The elbow method suggested keeping either two or seven principal components. There are tradeoffs to both methods. Using the elbow method at two PCs reduces the dimensionality more, but the tradeoff is that the two principal components explain less of the total variance at only 38%. The second elbow at PC7 explains nearly all of the variance at about 99% but retains more dimensionality. The recommendation would be to reduce the data set to the four principal components suggested by the Kaiser Criterion, understanding that it is a middle ground between the two elbows at PC2 and PC7. These principal components could be used in regression analysis or clustering analysis.

Regardless of how many principal components were chosen, it was clear that the most important variables were Bandwidth_GB_Year and Tenure. These variables had the highest weightings in PC1, the most influential component. This result was consistent with previous clustering work, which appeared to derive clusters based primarily on the Bandwidth variable. The importance of these variables should be communicated to decision-makers, and further analysis should be conducted on them.

F, 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/)

How to create correlation heatmap in R (July 12, 2022). Geeks for Geeks. Retrieved March 30, 2025, from (https://www.geeksforgeeks.org/how-to-create-correlation-heatmap-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

G, Sources for Content

Amit, H. (December 2, 2024). Principal component analysis explained. Medium. Retrieved March 30, 2025, from (https://medium.com/data-scientists-diary/principal-component-analysis-explained-5c42371db8b8)

Frost, J. (n.d.). Principal component analysis guide & example. Statistics By Jim. Retrieved March 30, 2025, from (https://statisticsbyjim.com/basics/principal-component-analysis/)

What is the kaiser criterion? (n.d). Learn Statistics Easily. Retrieved March 30, 2025, from (https://statisticseasily.com/glossario/what-is-kaiser-criterion-detailed-explanation/)