##Assignment 4

#Q1. The dataset spending.csv

#Download spending.csv includes annual spending in monetary units on
#diverse product categories of clients of a distributor. This is available online.

#The variables are annual spending on
#fresh product (fresh),
#annual spending on milk (milk),
#annual spending on grocery (grocery),
#annual spending on fresh detergent (detergent_paper),
#annual spending on frozen (frozen),
#annual spending on delicatessen products (delicatessen),
#channel (refers to 3 channels), and regions (3 regions).

#Q1a. Conduct a cluster analysis and provide a count of data points in each cluster.
#Provide your interpretation of clusters (i.e., what they seem to capture).

```
#Install spending.csv
spending <- read.csv("C:/Users/diana/OneDrive - Texas A&M
University/Desktop/Data_Analytics/Fall_2024/ANLY608/Assignment/Assignment4/spending.csv"
, header=T)

install.packages("dplyr")
library(dplyr)
#Here, we will create a subset considering only above variables

spending1<-select(spending, c("Fresh", "Milk", "Grocery", "Frozen", "Detergents_Paper",
"Frozen", "Delicassen"))
View (spending1)

#checking descriptive and missing
install.packages("skimr")
library(skimr)
skimr::skim(spending1)
#All the variables show variances so will not remove any of them
```

```
── Variable type: numeric ──────────────────────────────────────────────────────
  skim_variable    n_missing complete_rate   mean     sd p0   p25   p50    p75   p100 hist
1 Fresh                    0             1 12000. 12647.  3 3128. 8504  16934. 112151 ▇▁▁▁▁
2 Milk                     0             1  5796.  7380. 55 1533  3627   7190.  73498 ▇▁▁▁▁
3 Grocery                  0             1  7951.  9503.  3 2153  4756. 10656.  92780 ▇▁▁▁▁
4 Frozen                   0             1  3072.  4855. 25  742. 1526   3554.  60869 ▇▁▁▁▁
5 Detergents_Paper         0             1  2881.  4768.  3  257.  816.  3922   40827 ▇▁▁▁▁
6 Delicassen               0             1  1525.  2820.  3  408.  966.  1820.  47943 ▇▁▁▁▁
```

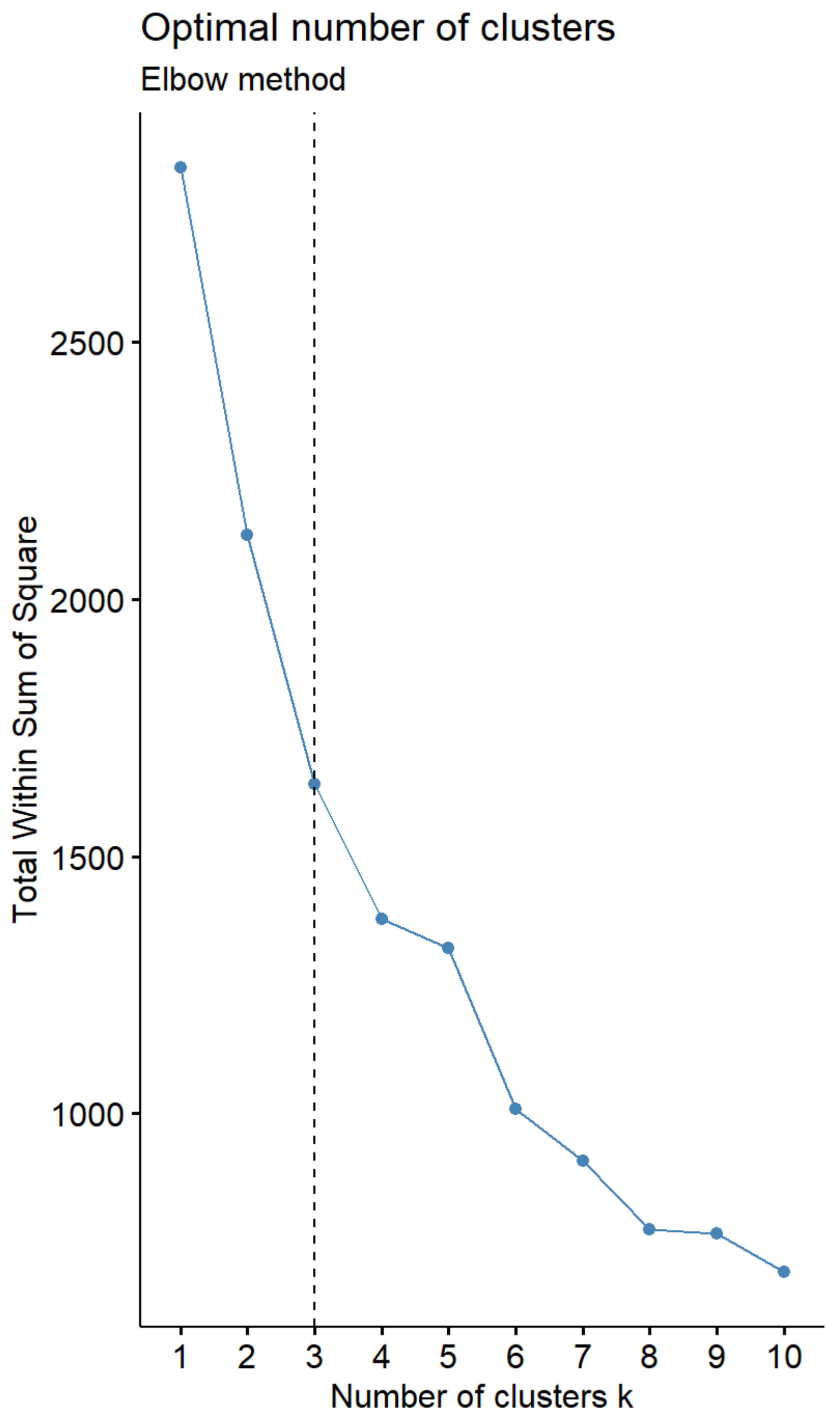
```
#Find missing if any and take only complete cases
spending_com <- na.omit(spending1)

#After data cleaning

#Step 1: Scaling (standardization)

spending_scaled <- data.frame(scale(spending_com, center = TRUE, scale = TRUE))

#Step 2: Number of clusters (This is based on Elbow method)
install.packages ("factoextra")
library(factoextra)
fviz_nbclust(spending_scaled, kmeans, method = "wss") +
  geom_vline(xintercept = 5, linetype = 2) + # add line for better viz
  labs(subtitle = "Elbow method") # add subtitle
```




```
#If we want to see the score for a specific cluster
install.packages ("cluster")
library(cluster)

km_res <- kmeans(spending_scaled, centers = 3) # defining cluster
```

```
# Add the cluster assignment to the original data
spending_scaled$cluster <- km_res$cluster

# Count data points in each cluster
cluster_counts <- spending_scaled %>%
  group_by(cluster) %>%
  summarize(count = n())

# Display the counts
print(cluster_counts)
```

```
> # Display the counts
> print(cluster_counts)
# A tibble: 3 × 2
  cluster count
    <int> <int>
1       1    52
2       2    47
3       3   341
```

```
km_res$centers
```

```
> km_res$centers
       Fresh        Milk    Grocery     Frozen Detergents_Paper  Delicassen
1  1.7822987 -0.01972441 -0.2141057  1.4530073       -0.4251979  0.72863286
2 -0.1615223 -0.44374118 -0.5006263 -0.1328292       -0.4589060 -0.20644028
3 -0.3728524  0.77595775  0.9456871 -0.3019746        0.9507125  0.09081268
```

#####INTERPRETATION###########

Cluster 1:

- High values for Fresh, Frozen, and Delicassen (1.78, 1.45, 0.73).
- May represent customers who purchase more fresh products and frozen items but less of other categories such as Milk and Grocery.

Cluster 2:

- Lower values for all categories (all negatives)
- May represent customers who tend to buy less of each category

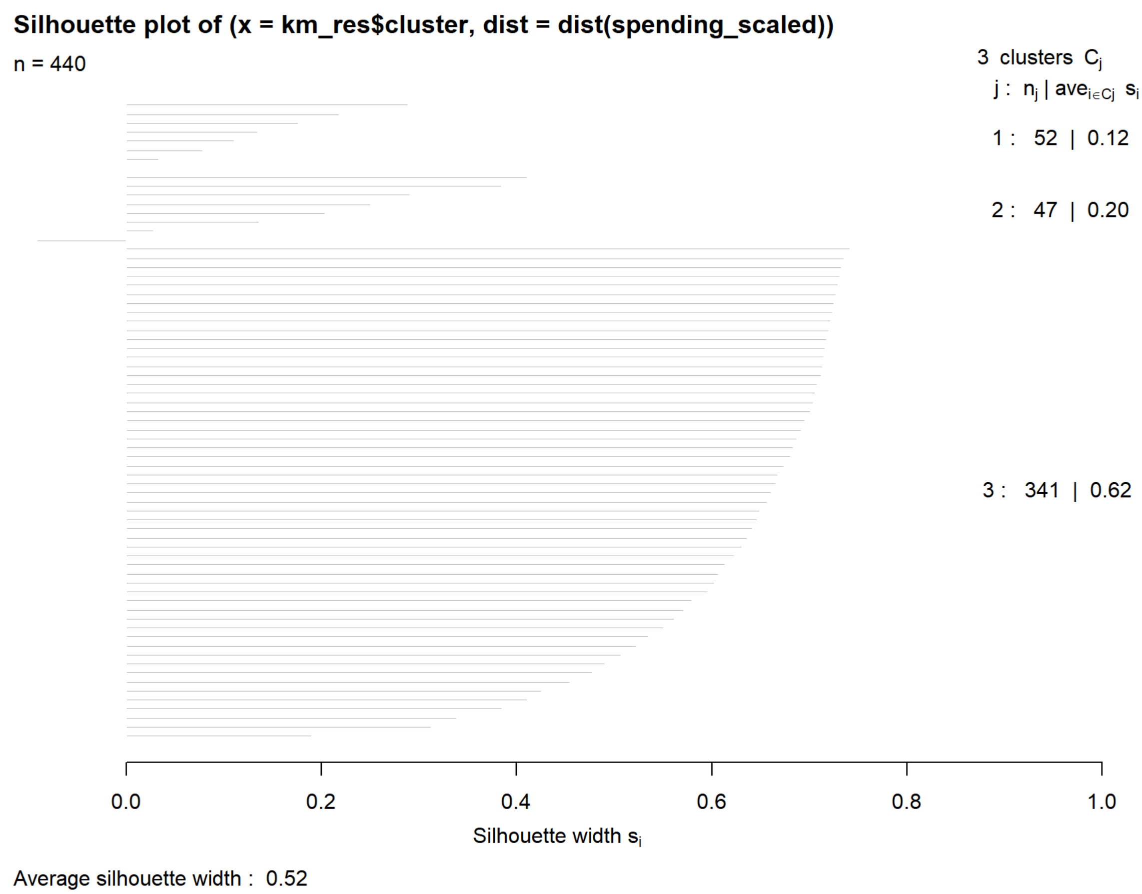
Cluster 3:

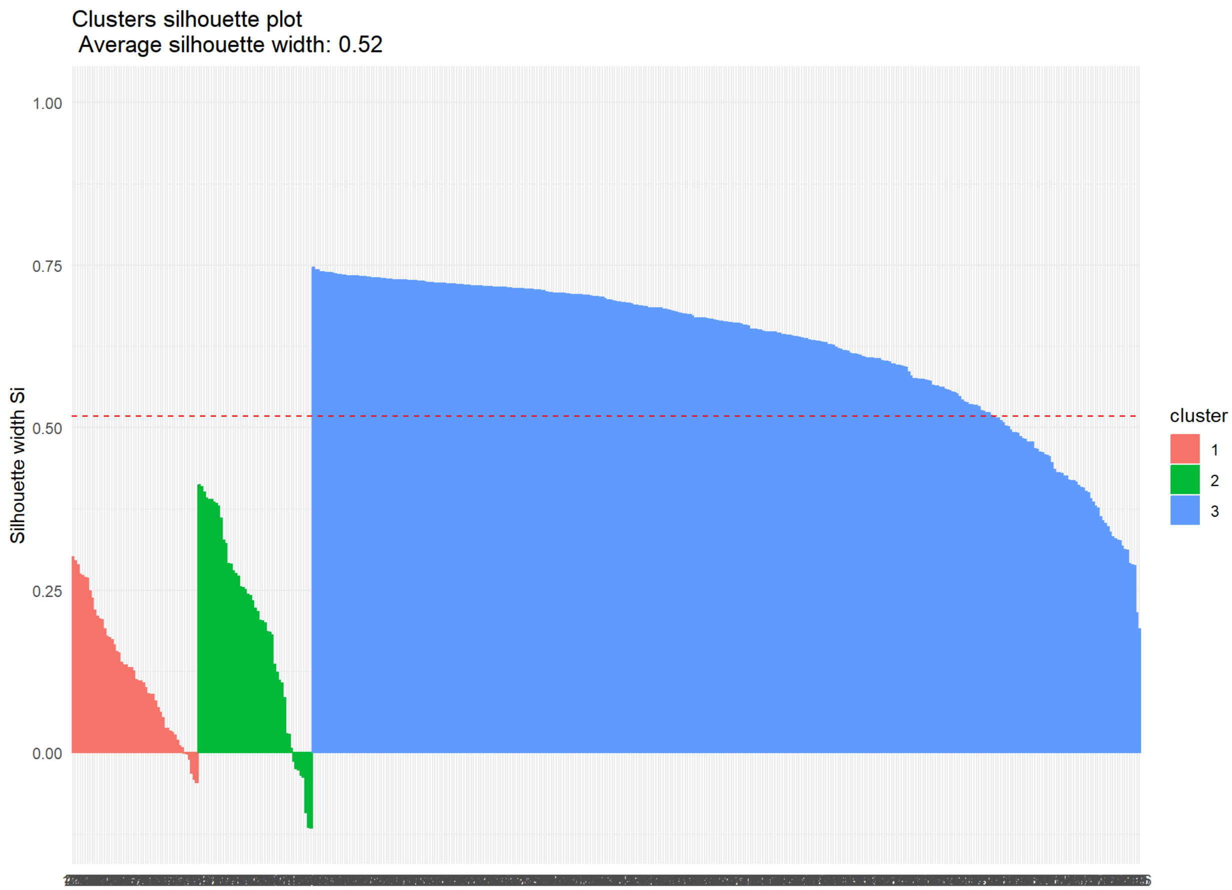
- High values for Milk (0.78), Grocery (0.94), and Detergents_Papers (0.95)
- May represent a group of customers who buy a lot of grocery items, milk, and detergents/paper products.

Cluster 1 may represent high-spending customers who buy fresh and frozen items.
Cluster 2 may represent customers who are budget-conscious.
Cluster 3 may represent customers who are from a larger household, which may explain the constant purchase of detergent products.

Cluster 3 has 341 points while Cluster 1 and 2 have 52 and 47 respectively. This indicates that the majority of the customers or typical customers are from Cluster 3. Cluster 1 and 2 may be more niche types of customers.

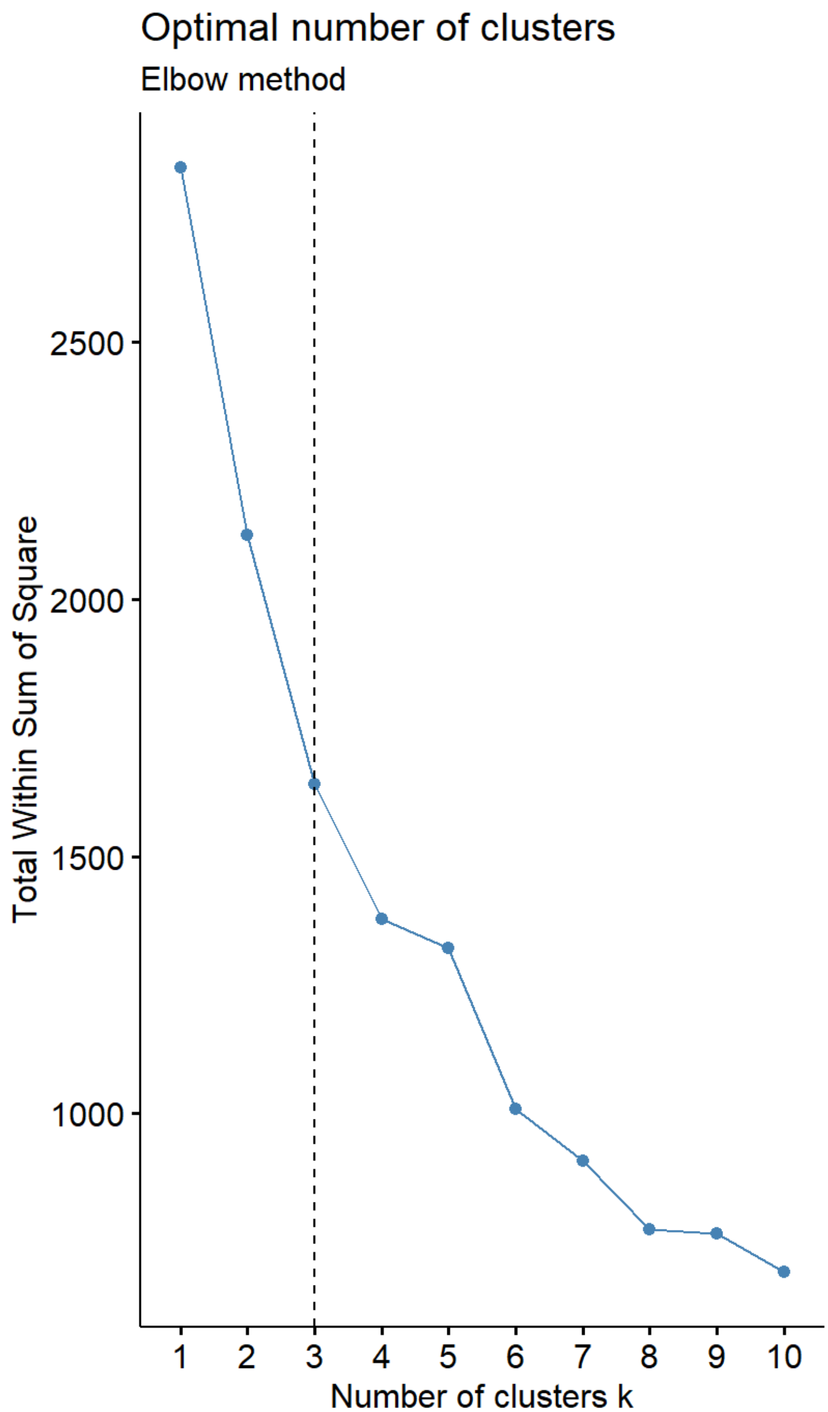
```
sil <- silhouette(km_res$cluster, dist(spending_scaled))
plot(sil)
```





#Q1b. Provide a plot that helps us to identify the optimal number of clusters.

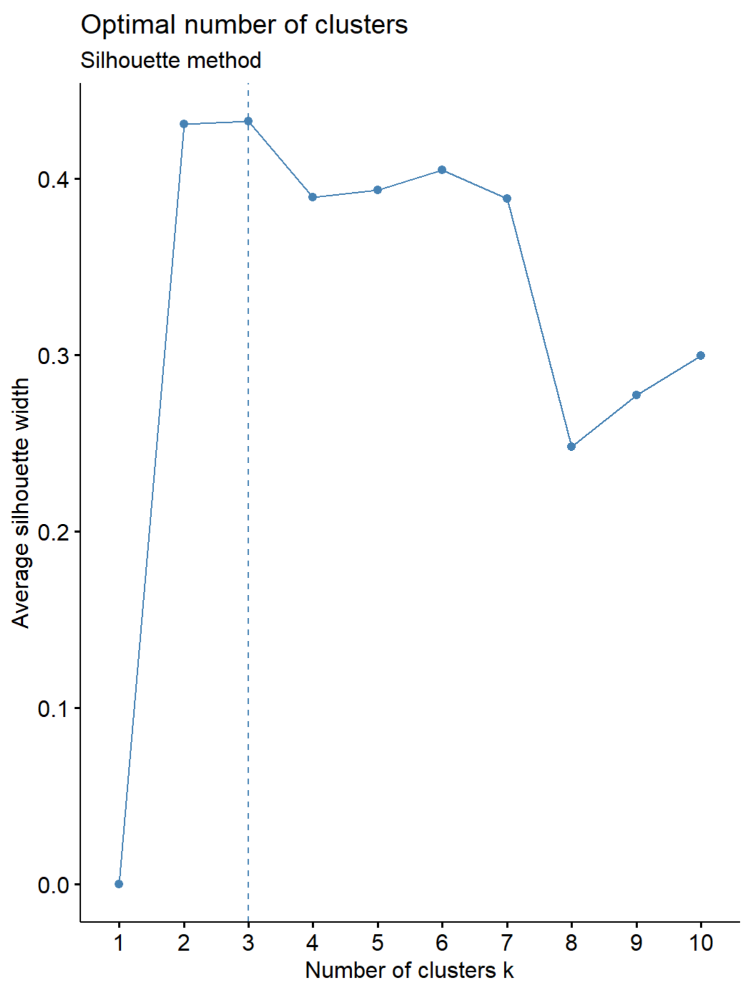
```
##Step 2: Number of clusters (This is based on Elbow method)
install.packages ("factoextra")
library(factoextra)
fviz_nbclust(spending_scaled, kmeans, method = "wss") +
  geom_vline(xintercept = 5, linetype = 2) + # add line for better viz
  labs(subtitle = "Elbow method") # add subtitle
```



#Step 3: Next, we look at the how well each data point is clustered with its own cluster compared
 #to other clusters. This is achieved through silhouette score, whose value ranges between -1 and +1.
A value close to 1=data point is perfectly clustered with its own cluster,
#0=data point is on the border between the two clusters

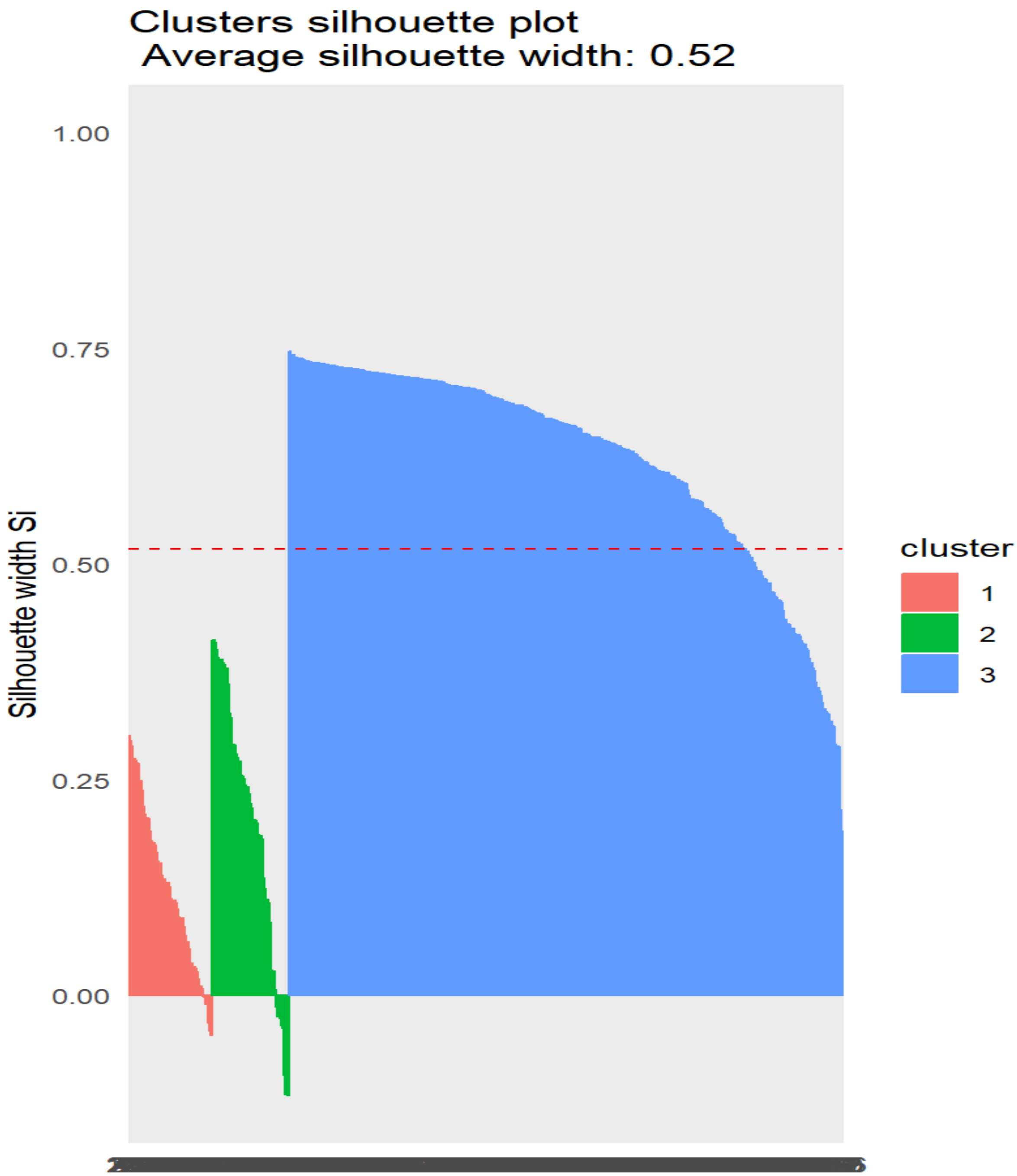
```
#-1=data point is equally well-clustered with two or more clusters
#The optimum number of clusters maximizes the avg. silhouette score

fviz_nbclust(spending_scaled, kmeans, method = "silhouette") +
  labs(subtitle = "Silhouette method")
```




```
sil <- silhouette(km_res$cluster, dist(spending_scaled))
plot(sil)
fviz_silhouette(sil) + theme_minimal()
#A positive silhouette coefficient indicates that an observation is well-matched to its own cluster.
```

```
> sil <- silhouette(km_res$cluster, dist(spending_scaled))
> plot(sil)
> fviz_silhouette(sil) + theme_minimal()
  cluster size ave.sil.width
1       1   52          0.12
2       2   47          0.20
3       3  341          0.62
```



Stone

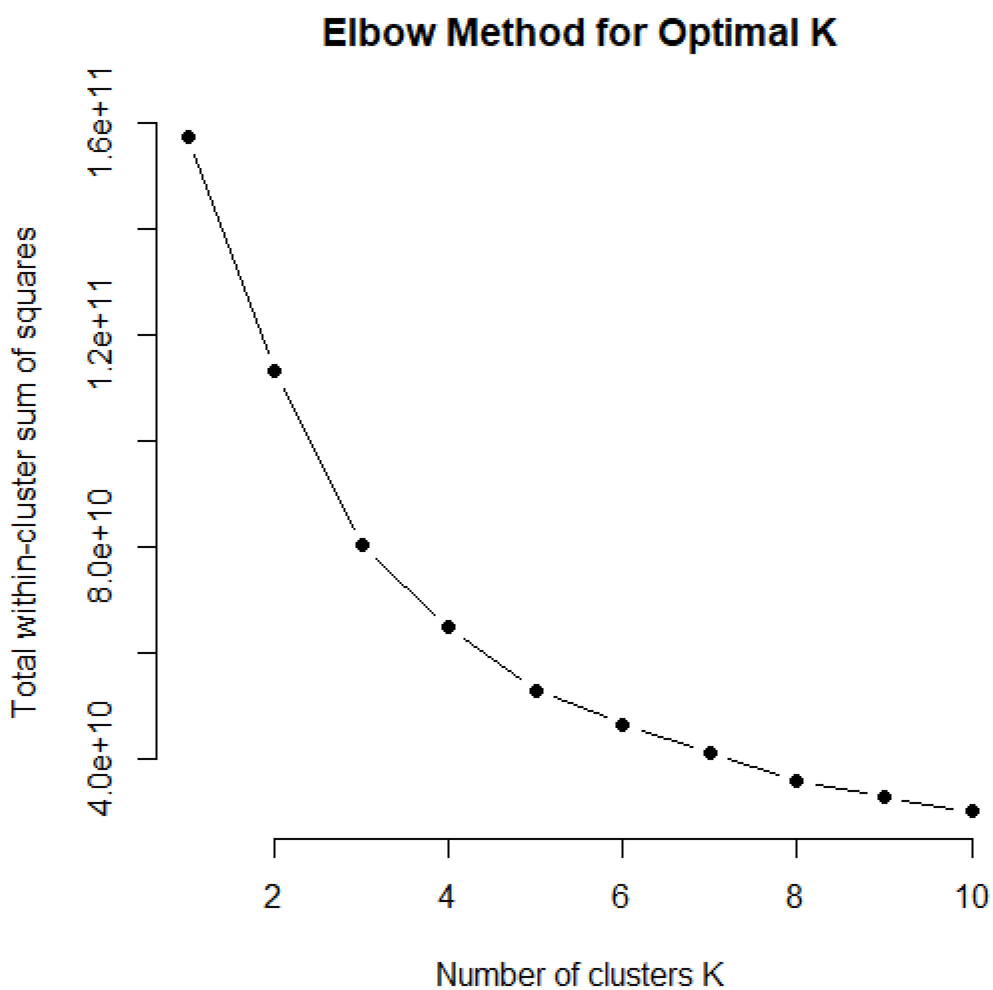
```
# Load necessary libraries
library(dplyr)
library(ggplot2)
library(cluster)

# Q1: Load the dataset
data <-
read.csv("C:/Users/leiker-s/Desktop/spending.csv")

# Q1a: Perform K-means clustering and count data points
in each cluster
numeric_data <- data %>% select_if(is.numeric)
set.seed(123)
kmeans_result <- kmeans(numeric_data, centers = 3,
nstart = 20)
data$cluster <- kmeans_result$cluster
cluster_counts <- table(data$cluster)
print(cluster_counts)
  1   2   3
330  50  60
cluster_means <- aggregate(. ~ cluster, data = data, FUN
= mean)
print(cluster_means)
```

```
  cluster  Channel   Region    Fresh      Milk   Grocery
Frozen
1       1 1.260606 2.554545  8253.47  3824.603  5280.455
2572.661
2       2 1.960000 2.440000  8000.04 18511.420
27573.900 1996.680
3       3 1.133333 2.566667 35941.40  6044.450  6288.617
6713.967
  Detergents_Paper Delicassen
1         1773.058   1137.497
2        12407.360   2252.020
3         1039.667   3049.467
```

```
# Q1b: Determine the optimal number of clusters using the
Elbow Method
wss <- sapply(1:10, function(k){
  kmeans(numeric_data, centers = k, nstart =
20)$tot.withinss
})
plot(1:10, wss, type = "b", pch = 19, frame = FALSE,
     xlab = "Number of clusters K",
     ylab = "Total within-cluster sum of squares",
     main = "Elbow Method for Optimal K")
```


Elbow Method for Optimal K
Total within-cluster sum of squares
1.6e+11
1.2e+11
8.0e+10
4.0e+10
2
4
6
8
10
Number of clusters K

Question 2

Kayla’s Code

#Q2a. Create a dataset with only the numeric variables.
#Now create a histogram for each continuous variable. Do the variables follow normal distribution?

```
df <- read.csv("C:/Users/kayla/Downloads/credit default.csv", header = T)
View(df)

install.packages('dplyr')
library(dplyr)

#numeric data
df1 <- select(df, c("Default", "duration", "amount", "installment", "residence", "age",
"cards", "liable"))

#continuous data
df11 <- select(df, c("duration", "amount", "installment", "age"))

install.packages("skimr")
library(skimr)
skimr::skim(df11)
View(df1)
```

```
── Variable type: numeric ──────────────────────────────────────────────────────────────
  skim_variable n_missing complete_rate   mean     sd  p0   p25   p50   p75  p100 hist
1 duration              0             1   20.9   12.1   4    12    18    24    72 ▇▇▂▁▁
2 amount                0             1 3271.  2823.  250 1366. 2320. 3972. 18424 ▇▂▁▁▁
3 installment           0             1    2.97   1.12  1     2     3     4     4 ▂▃▁▂▇
4 age                   0             1   35.5   11.4  19    27    33    42    75 ▇▇▃▂▁
```

#Interpretation: The continuous variables do not follow normal distribution.

#Q2c. Now predict which class of default an observation will fall if duration=6, amount=1100, installment=4, and age=67.

```
# Load necessary library
library(cluster)
```

```r
# Assuming your dataset 'data' is already loaded and the 'kmeans_model' is created

# New observation data
new_data <- data.frame(duration = 6, amount = 1100, installment = 4, age = 67)

# Scale the new observation using the same scaling parameters as the original data
new_data_scaled <- scale(new_data, center = attr(data_scaled, "scaled:center"),
                         scale = attr(data_scaled, "scaled:scale"))

# Calculate the Euclidean distance from the new observation to each cluster center
distances <- apply(kmeans_model$centers, 1, function(center) {
  sum((new_data_scaled - center)^2)
})

# Find the index of the closest cluster
predicted_cluster <- which.min(distances)

# Map predicted cluster to the default/no-default class
if (predicted_cluster == 1) {
  predicted_class <- "no default"
} else {
  predicted_class <- "default"
}

# Print the predicted class
predicted_class

#Results: The predicted class is "no default".
```

#Q2d. Now estimate a QDA model and predict which class of default an observation will fall
#if duration=6, amount=1100, installment=4, and age=67.

#We did not learn QDA in class.

Kshitij's Code

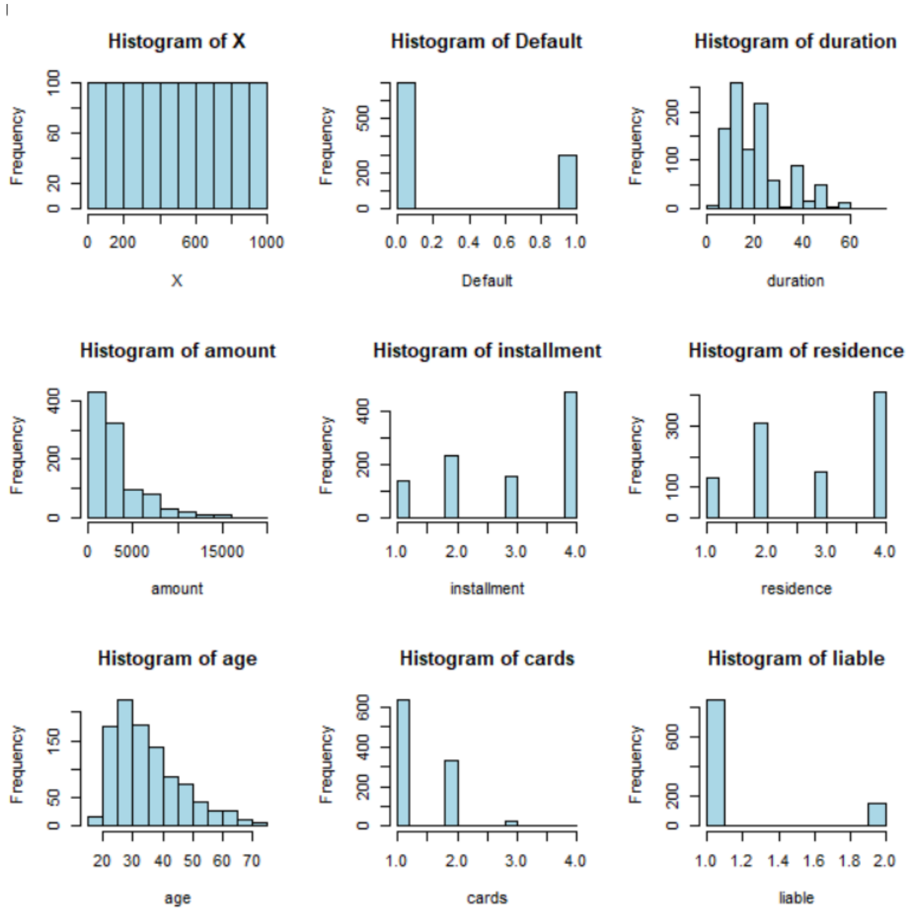
Q2A)

```
# Load necessary packages
library(ggplot2)

# Load data (replace with actual path to "credit default.csv")
data <- read.csv("C:/Users/jaink/Downloads/Class Work/ANLY608/credit default.csv")

# Select only numeric variables
numeric_data <- data[, sapply(data, is.numeric)]

# Plot histograms for each numeric variable
par(mfrow = c(3, 3)) # Set layout for multiple plots
for (col in names(numeric_data)) {
  hist(numeric_data[[col]], main = paste("Histogram of", col), xlab = col, col = "lightblue")
}
par(mfrow = c(1, 1)) # Reset layout
```



Interpretation

1. **X**: Even distribution, likely an index column.
2. **Default**: Skewed towards `0`, indicating most customers didn't default.
3. **Duration**: Positively skewed; most loans are shorter-term.
4. **Amount**: Positively skewed, with more smaller loan amounts.
5. **Installment**: Nearly uniform across categories.
6. **Residence**: Certain residence categories are more common.
7. **Age**: Positively skewed; most customers are younger.
8. **Cards**: Skewed towards lower values; most have few cards.
9. **Liable**: Concentrated at lower values, with few dependents.

Summary: Most variables are positively skewed, indicating non-normal distributions, which may require transformations for modeling.

Q2B)

```
# Load necessary library
library(MASS)

# Convert 'Default' to a factor for classification
data$Default <- as.factor(data$Default)

# Perform LDA on selected numeric variables
lda_model <- lda(Default ~ duration + amount + installment + age, data = data)

# Predict default class and create confusion matrix
lda_predictions <- predict(lda_model)
confusion_matrix <- table(Predicted = lda_predictions$class, Actual = data$Default)
print(confusion_matrix)

# Model performance interpretation
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Model accuracy:", accuracy)
```

```
> print(confusion_matrix)
         Actual
Predicted   0   1
        0 669 256
        1  31  44
>
> # Model performance interpretation
> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> cat("Model accuracy:", accuracy)
Model accuracy: 0.713>
```

Interpretation
Confusion Matrix:

- Correctly classified: 669 (no default), 44 (default).
- Misclassified: 256 (default as no default), 31 (no default as default).

Accuracy: 71.3%, showing moderate performance but with difficulty accurately predicting defaults.

Q2C)

new_observation <- data.frame(duration = 6, amount = 1100, installment = 4, age = 67)

# Predict the default class for the new observation
lda_prediction <- predict(lda_model, new_observation)
cat("Predicted class for the new observation:", lda_prediction$class)

```
> lda_prediction <- predict(lda_model, new_observation)
> cat("Predicted class for the new observation:", lda_prediction$class)
Predicted class for the new observation: 1>
```

Interpretation

The Linear Discriminant Analysis (LDA) model predicts that the new observation belongs to class **1** (default). This indicates that, based on the input values provided, the model estimates a likelihood of default for this observation.

GABEs Code

Question 2 Gabe

```
library(ggplot2)
library(dplyr)

# Load data
setwd("C:/Users/palomarez-g/Documents/MSA 608_2024/Rcode")
credit_data <- read.csv("credit default.csv")

# veiw first few rows and structure
head(credit_data)
str(credit_data)

# numeric variables change
numeric_data <- credit_data %>% select_if(is.numeric)

# View data agian to verify
head(numeric_data)

# Reshape the data to a long format
long_data <- pivot_longer(numeric_data, cols = everything(), names_to = "Variable", values_to =
"Value")

ggplot(numeric_data, aes_string(x = col)) + geom_histogram(binwidth = 10, fill = "skyblue", color
= "black", alpha = 0.7) + ggtitle(paste("Histogram of", col)) + theme_minimal() + print()


#Q2B
library(MASS)
library(caret)

#factor classification
data$Default <- as.factor(numeric_data)

# Perform LDA on selected numeric variables
lda_model <- lda(Default ~ duration + amount + installment + age, data = data)

# LDA model
lda_predictions <- predict(lda_model)

# Create confusion matrix
confusion_matrix <- table(Predicted = lda_predictions$class, Actual = data$Default)
print(confusion_matrix)

# Accuracy calculation
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
```

```r
# Print model accuracy
cat("Model accuracy:", accuracy, "\n")

#Q2c

# New data frame
new_data <- data.frame(
  duration = 6,
  amount = 1100,
  installment = 4,
  age = 67
)

# Predict new observation
lda_predictions <- predict(lda_model, new_data)

# Print the predicted class
cat("Predicted class of default:", lda_predictions$class, "\n")
```

Issues: My code is linking itself together and I can't seem to unlink it so if yall could help me
And Q2d I have no clue how to do it.