Homework #6: Segmentation

Aditya Arora

Grading Note

This HW is worth 40 points in total. I've made notes in this document where those points were earned.

Homework tasks:

- Define a segmentation scheme for a women's apparel brand
- Gain practice with clustering techniques
  - Euclidean and Gower distance (similarly) measures
  - K-means clustering algorithm
- The apparel customer dataset contains data on customer characteristics
  - Cross-section of observations
  - We observe last year expenditures (on all products) by channel (retail and online)
  - We directly observe the customer's age and gender (direct demographics)
  - We impute Census demographics using a zip-code matching process
    - Income, white (fraction white households), college (fraction adults w/ degree)
  - Data file is: `apparel_customer_data.csv`

The variables in the dataset are:

Variable	Description
`iid`	Identifier for customer
`spend_online`	dollars spent last 12 months on online purchases
`spend_retail`	dollars spent last 12 months on retail purchases
`age`	customer age
`male`	1 = if consumer is male
`white`	proportion of households in customer zip code that are white
`college`	proportion of households in customer zip code that have college

Variable	Description
hh_inc	median income of households in customer zip code ('000)

Read in the data

Q1 To begin, load the customer data into a dataframe named DF. Use head() and summary() to visualize the first few rows and to summarize the variables. **(1 point)**

```
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
```

```
library(ggplot2)
library(quarto)

DF <- read.csv('apparel_customer_data.csv')
head(DF)
```

```
   iid spend_online spend_retail age     white   college male  hh_inc
1  199       34.975        0.000  41 0.6403464 0.7028232    0 148.603
2 2298      220.135        0.000  41 0.2723192 0.2530814    0  25.469
3 9594       39.950        0.000  44 0.8428670 0.5984784    0  84.702
4 9542       34.975      480.355  40 0.9354839 0.5439673    0  83.125
5 1163       50.400        0.000  32 0.9256757 0.6632826    0 132.813
6 6013        0.000       46.000  40 1.0000000 0.4557109    0 128.558
```

```
summary(DF)
```

```
      iid         spend_online      spend_retail          age       
 Min.   :   14   Min.   :   0.00   Min.   :   0.00   Min.   :18.00  
 1st Qu.: 2946   1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:33.00  
 Median : 5430   Median :  14.97   Median :  27.71   Median :41.00  
 Mean   : 5463   Mean   :  72.44   Mean   :  78.00   Mean   :40.91  
 3rd Qu.: 8110   3rd Qu.:  70.72   3rd Qu.:  78.00   3rd Qu.:49.00  
 Max.   :10589   Max.   :1985.75   Max.   :2421.91   Max.   :88.00  
     white           college             male            hh_inc       
 Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :  2.499  
 1st Qu.:0.7297   1st Qu.:0.3835   1st Qu.:0.000   1st Qu.: 59.356  
 Median :0.8550   Median :0.5580   Median :0.000   Median : 87.364  
 Mean   :0.7993   Mean   :0.5437   Mean   :0.091   Mean   : 96.254  
 3rd Qu.:0.9422   3rd Qu.:0.7136   3rd Qu.:0.000   3rd Qu.:122.602  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :250.001  
```

Q2: Which (continuous) variables stand out in terms of being high-skew? *(1 point)*

Answer

spend_online, spend_retail are highly skewed because there is high difference between mean and median for these variables.

Q3: What is the minimum value of the high-skew variables? *(1 point)*

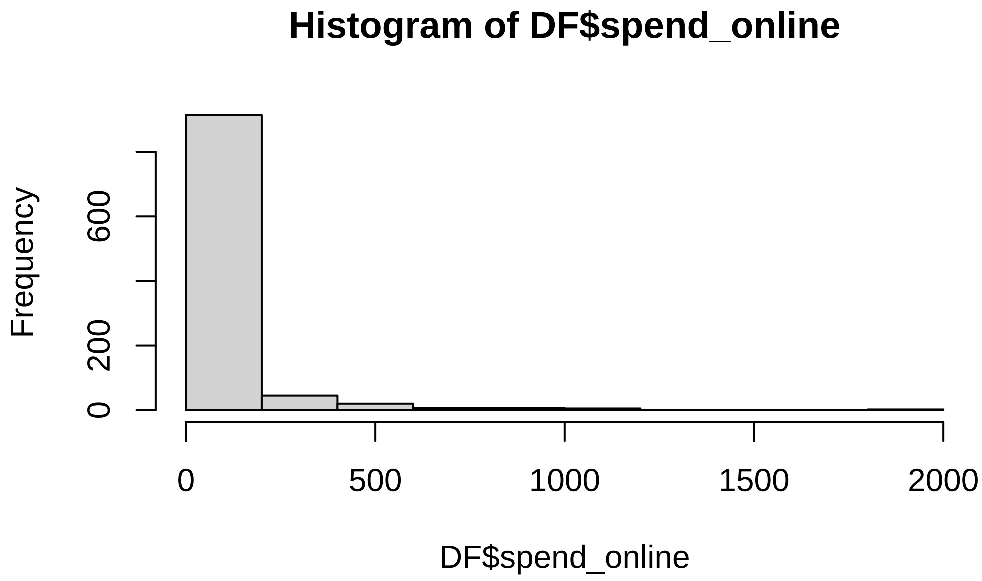
Answer

spend_online = 0.00 spend_retail = 0.00

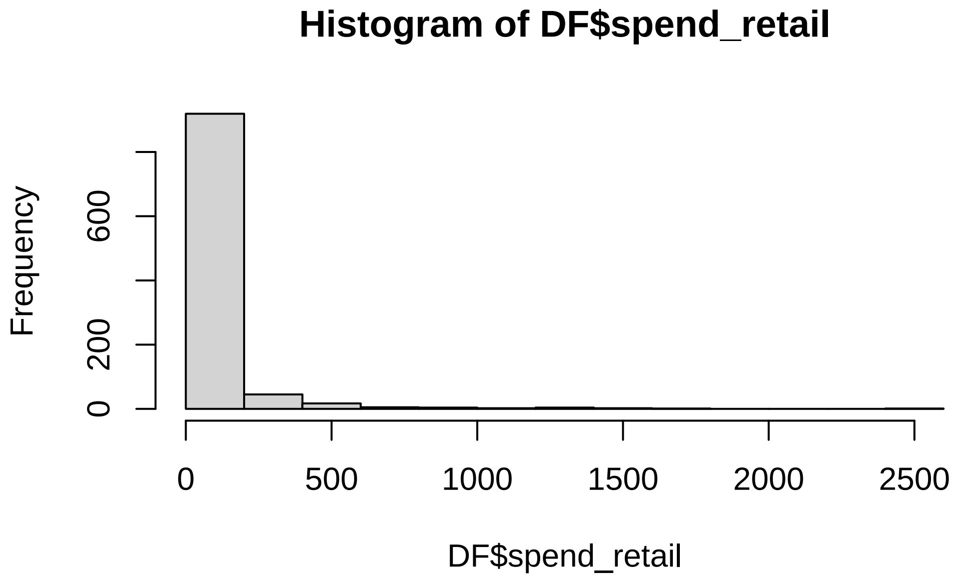
Histograms of all variables

Q4: Next, we wish to inspect the distribution of all the variables we might use for the cluster analysis. Generating histograms of each variable: *(2 points)*

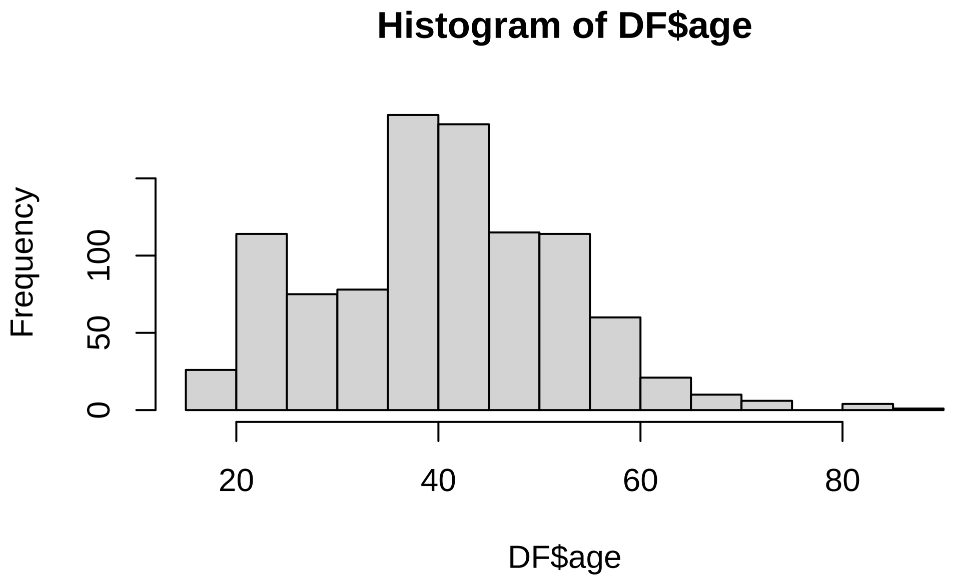
```
hist(DF$spend_online)
```



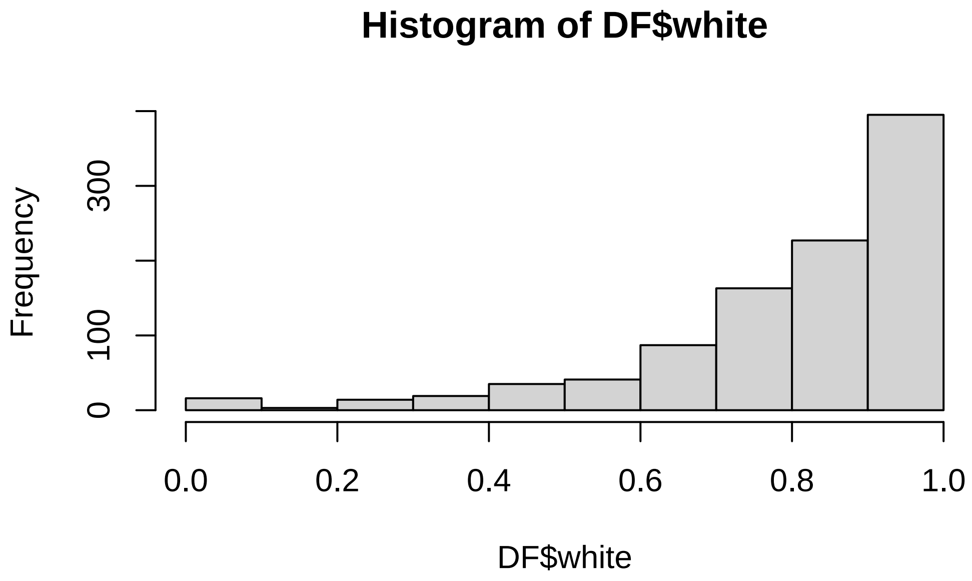
```
hist(DF$spend_retail)
```



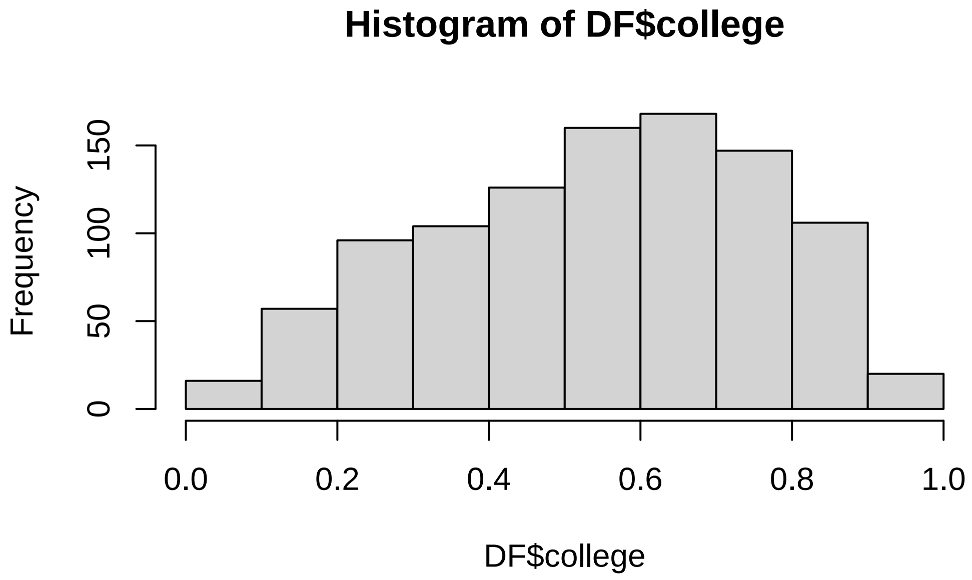
```
hist(DF$age)
```



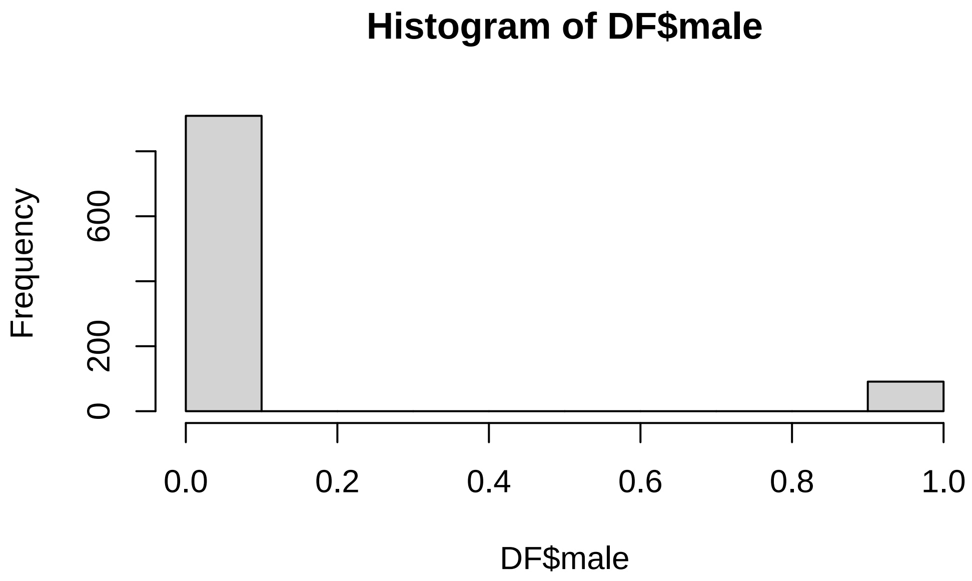
```
hist(DF$white)
```



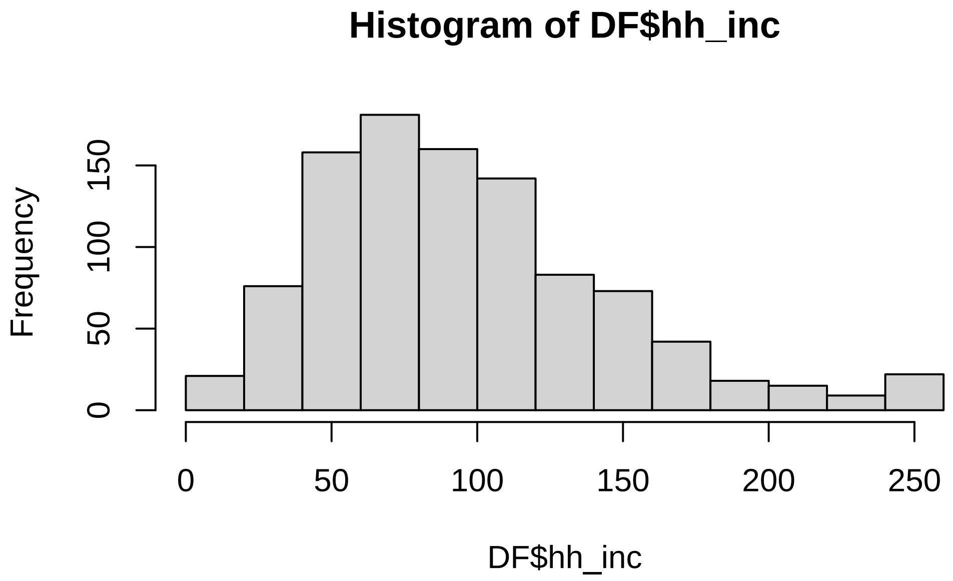
```
hist(DF$college)
```



```
hist(DF$male)
```



```
hist(DF$hh_inc)
```



Q5: By inspecting the histograms, which variables are continuous, and which are binary? *(1 point)*

Answer

Continuous - spend_online, spend_retail, age, white, college & hh_inc Binary - male

Q6: Which variables demonstrate high-skew in their histograms? *(1 point)*

Answer

spend_online, spend_retail

Q7: What do we conclude about: (a) which variables should be log-transformed, and (b) which distance metric would be appropriate for these data (assuming all variables will be used)? *(1 point)*

Answer

spend_online & spend_retail should be log transformed because of high skewness. Gower distance will be appropriate because it can be used with data that has a mix of binary and continuous variables.

Q8: Which variables should be log-transformed? *(1 point)*

Answer

spend_online & spend_retail should be log transformed.

Clustering steps

Here we go through the clustering steps outlined in the lecture slides.

1. Select variables to use for clustering

Since we have a limited number of variables, and because all look potentially relevant, we will include all variables in our initial analysis.

Often, we do this iteratively, such that we may subsequently omit variables that contribute little to distinguishing the clusters or are impractical for developing targeted marketing strategies.

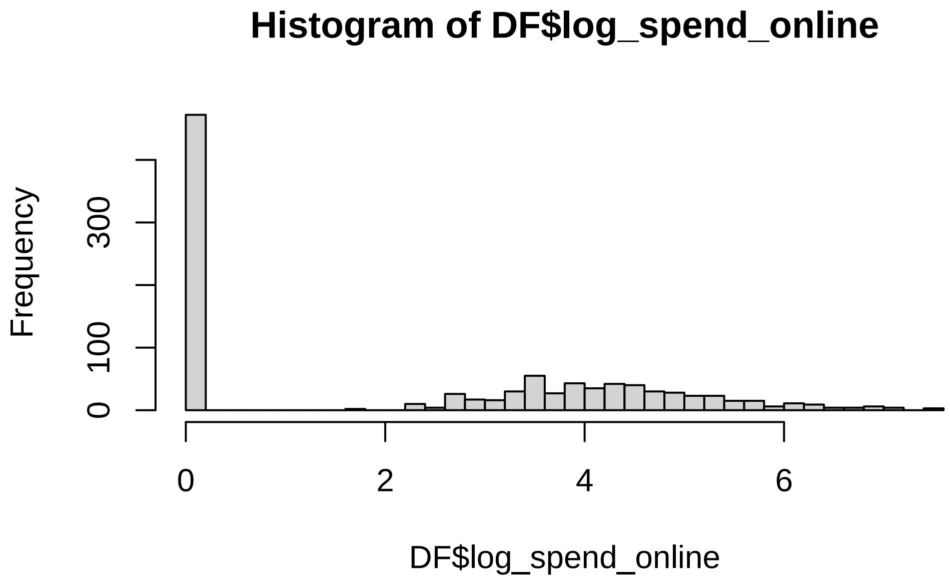
1.2. Log-transformation of skewed variables

Having observed the distributions of the variables, some stand out as different from the rest. Since clusting algorithms tend to perform poorly with highly skewed variables, we will transform them in a way that reduces skew.

The usual way to quickly handle skewed distributions such as these is to take the log-transform, which usually will give the data a more normal-shaped distribution. **Important:** The minmum value of these variables might be zero, and **log of zero (or negative numbers) is not possible. To deal with both problems, we transform the expenditure levels by taking log(1+x), where x is the untransformed variable.**

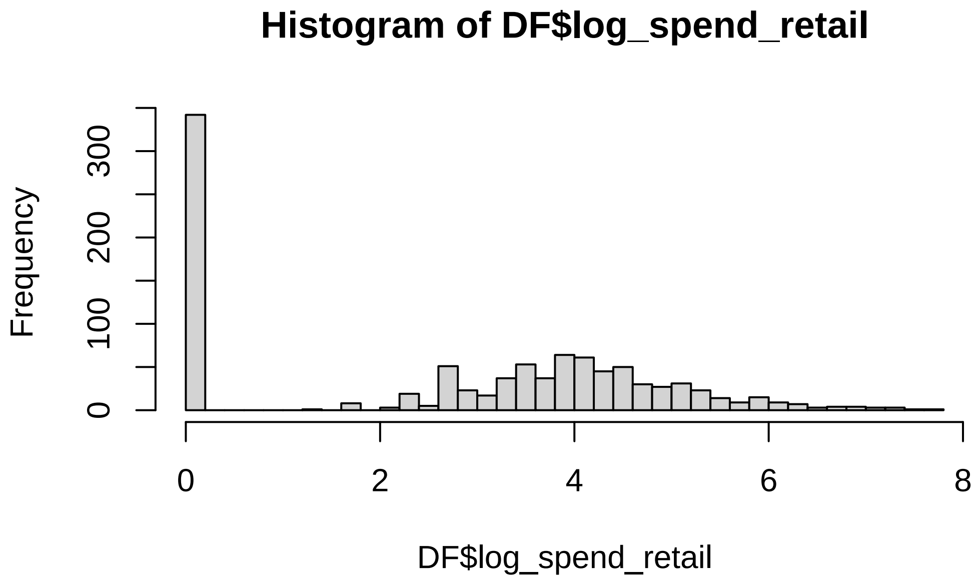
Specifically, to the dataframe `DF`, add variables named `log_variable` by taking the `log(1 DF$variable)` transformation of each variable that is skewed. Plot histograms for these. For example, `DF$spend_online` is one of the problem variables. Here is what we do:

```r
DF$log_spend_online <- log(1+DF$spend_online)
hist(DF$log_spend_online,50)
```



Q9: Repeat this for the other skewed variable *(1 point)*

```
DF$log_spend_retail <- log(1+DF$spend_retail)
hist(DF$log_spend_retail,50)
```



Q10: How would you characterize the distribution of the transformed variables? Do the distributions appear more like the normal distribution (bell curve)? Are there multiple modes (peaks)? *(1 point)*

Answer

No, the distributions do not appear like normal distribution. there are multiple peaks.

1.3 Create dataframe with finalized cluster variables (only)

To make matters easier later, create a separate dataframe with *only* the cluster variables we intend to use to generate clustering (segmentation) purposes.

This code will create a dataframe called `DF` that *only* has the following variables: `log_spend_online`, `log_spend_retail`, `age`, `white`, `college`, `male`, `hh_inc`:

```r
# create dataframe with transformed variables, omit non-cluster variables
DF <- DF
DF$spend_online <- NULL
DF$spend_retail <- NULL
```

2 Define distance measure between individuals

2.1 Euclidean distance

We measure similarity between two customers by calculating the "distance" between them in terms of their observable characteristics.

Recall from basic geometry that we can find the distance (d) between two points (x_1,y_1) and (x_2,y_2) as: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. This is simply a version of the Pythagorean theorem, which relates the length of a triangle's hypotenuse (longest edge) to the length of its sides (generally expressed $c^2 = a^2 + b^2$, where c is the hypotenuse).

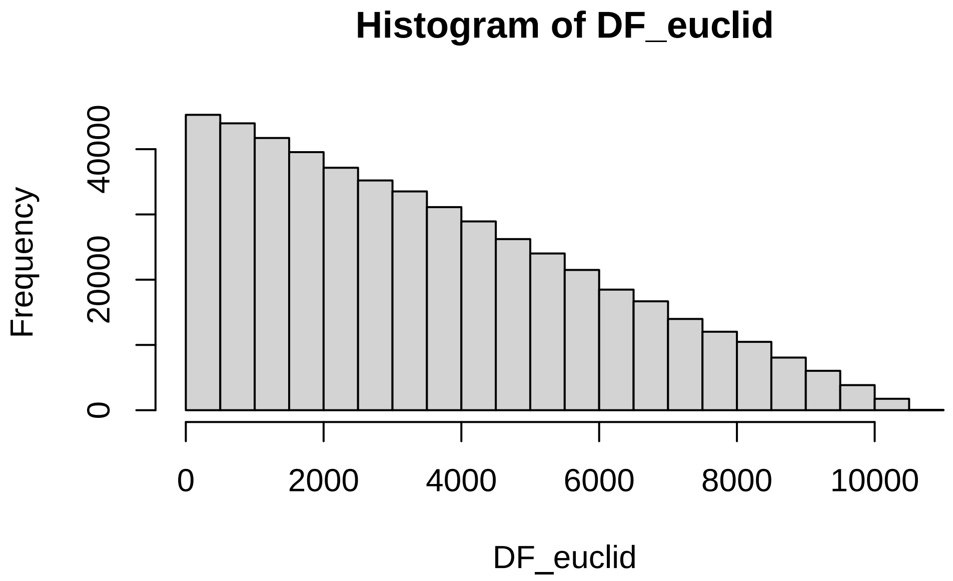
Rather than thinking of points in physical space, we can think of points in "characteristic" space. For example, the x-axis could represent a person's age and the y-axis could represent a person's income. The "distance" between two people in this case would be the square root of the squared difference in their age plus the squared difference in their income.

Consistent with its geometric origins, distance defined in this way is known as *Euclidean* distance. Note that the distance concept extends to higher dimensions, such that for $k = 1, ..., K$ dimensions, distance is given by: $d = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + ... + (x_{1K} - x_{2K})^2}$

Euclidean distance is (most) appropriately applied to a set of continuous variables. For data that is a mixture of continuous and binary/categorical variables, other distance metrics (e.g. Gower distance) are preferred.

This code calculate the (unstandardized) Euclidean distance between all pairs of consumers across the variables in dataframe `DF`.

```
library(cluster)
DF_euclid <- daisy(DF, metric = "euclidean",warnType=FALSE)
hist(DF_euclid)
```



2.2 Standardized Euclidean distance

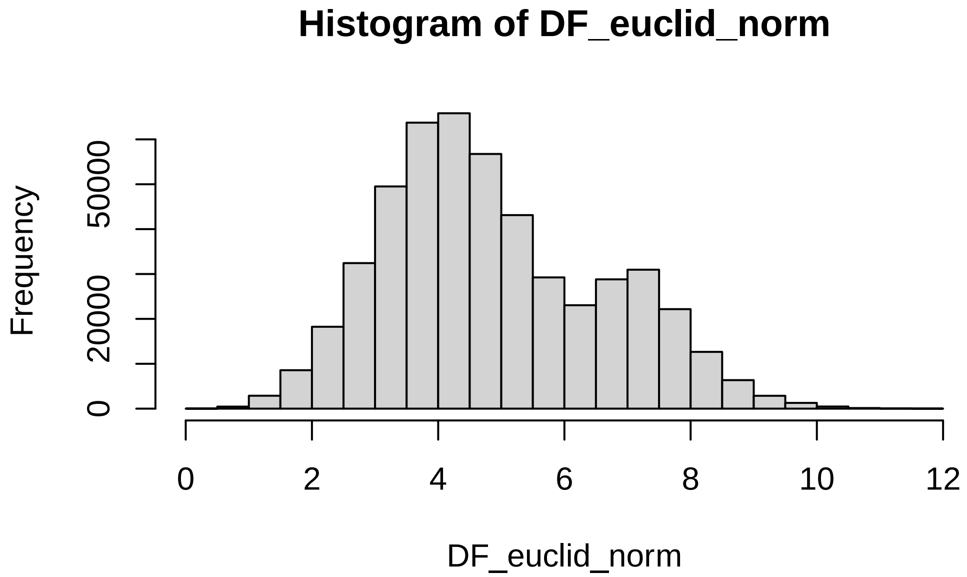
As previously mentioned, clustering algorithms tend to work best with input variables that are (approximately) normally distributed. This is principally because clustering algorithms tend to work best when the resulting *distance distribution* is normally distributed, and this tends to occur when the underlying variables are normally distributed.

In addition to log-transforming highly skewed variables, *standardizing* variables can result in distance distributions that are closer to being normally-distributed. Standardizing variables means that the variables are rescaled so that each variable has zero mean and unit (1) variance, e.g. $\tilde{x}_i = \frac{x_i - \bar{x}}{\sigma_x}$. The rationale for standardizing is that putting all variables on the "same scale" should give each variable roughly equal weight in contributing to the distance between points (consumers).

Q11: Calculate the standardized Euclidean distance between all pairs of consumers across the variables in dataframe `DF` (as defined in 3.1.2). The `stand=TRUE` option to the `daisy()` function can be useful for this task. Call the resulting list of pairwise distances `DF_euclid_norm`. Also, generate a histogram of `hist_euclid_norm`. *(2 points)*

```
DF_euclid_norm <- daisy(DF, metric = "euclidean",warnType=FALSE,stand = TRUE)
```

```r
hist(DF_euclid_norm)
```



Q12: Characterize the shape of the standardized Euclidean distance distribution. How does it compare to the non-standardized distance distribution? What does this imply for the results of our clustering later? *(2 points)*

Answer

The shape of the standardized distribution looks more normally distributed when compared to the non-standardized one. This implies that the clustering will be better because the more normally distributed the better.

2.3 Gower distance

In many cases, we have a *mixture* of continuous and binary/categorical variables. In such cases, Euclidean distance metrics can perform poorly.

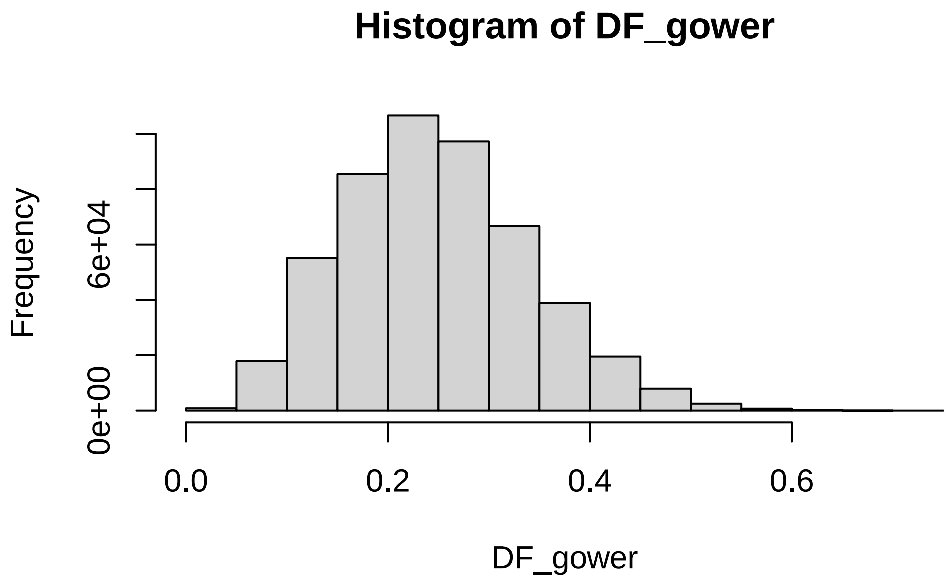
For mixed continuous & binary data, a better option is the Gower distance metric, which defines the distance between individuals i and j on variable k (e.g. age, income, etc.) as follows:

$$d_{ijk} = \begin{cases} \frac{|x_{ik}-x_{jk}|}{max(x_k)-min(x_k)} & x_k \ continuous \\ 0 & x_k \ binary, \ x_{ik} = x_{ij} \\ 1 & x_k \ binary, \ x_{ik} \neq x_{ij} \end{cases}$$

The total distance between individuals i and j is then just the sum over all observed variables, $d_{ij} = \sum_k d_{ijk}$. Note that the Gower metric “automatically” standardizes variables by construction. For continuous variables, the distance between any two individuals is normalized with respect to the maximum distance possible between any two individuals. The result is to map the original variable into the range [0,1], which is the same scale as binary variables.

Q13: Calculate the Gower distance between all pairs of consumers across the variables in dataframe `DF`. Call the resulting list of pairwise distances `DF_gower`. Also, generate a histogram of `DF_gower`. *(1 point)*

```
DF_gower <- daisy(DF, metric = "gower", warnType = FALSE)
hist(DF_gower)
```



Q14: Based on the data types in `DF` and the shapes of the distance distributions, I am going to go with the Gower distance metric for the rest of this homework. Why is that? *(2 points)*

Answer

Since, our dataset contains both continuous and binary variables, to ensure that distance is computed with equal contributions from all the variables we use Gower distance.

3 Select clustering procedure

Using the pair-wise distance measures, clustering algorithms are used to group individuals into segments (clusters). There are many different types of clustering algorithms, which generally fall into 2 categories: hierarchical and non-hierarchical.

We focus on non-hierarchical methods, and the k-means (`kmeans()`) clustering algorithm in particular. We choose k-means because it tends to be the most general purpose method in terms of applicability and performance. Non-hierarchical methods like k-means determine clusters by optimizing (maximizing/minimizing) some measure of clustering "fit".

In the case of the k-means algorithm, the objective is to minimize the total within-cluster sum of squares. That is, for a fixed number of clusters, the algorithm minimizes pairwise distances within the clusters. To determine cluster membership, the k-means algorithm begins by assigning k individuals at random to the k clusters. Then, the algorithm iterates between: (a) assigning individuals to the cluster with the closest centroid (mean variable values for all cluster members), and (b) recomputing the cluster cenrtoid values. The algorithm coverges (stops) when further iterations do not change the membership of the clusters.

In this section, we will perform k-means clustering using the **Gower distance matrix**. We will estimate cluster solutions for segments of size 2, 3 and 4. We will analyze these clustering solutions in section 5.

3.1 K-means (Gower), 2 segments

Q15: Using the Gower distance matrix, perform a k-means cluster analysis with K = 2 clusters. Use a minimum of 10 initial starting points. Save the result to `clu_gower_2`. Finally, add the cluster assignments to the orginal dataframe, `DF` – name the column `clu_gower_2`: *(1 point)*

```r
clu_gower_2 <- kmeans(DF_gower, centers = 2, nstart = 10)

DF$clu_gower_2 <- clu_gower_2$cluster
```

3.2 K-means (Gower), 3 segments

Q16: Using the Gower distance matrix, perform a k-means cluster analysis with K = 3 clusters. Save the result to `clu_gower_3`. Finally, add the cluster assignments to the orginal dataframe, `DF` – name the column `clu_gower_3`: *(1 point)*

```r
clu_gower_3 <- kmeans(DF_gower, centers = 3, nstart = 10)
DF$clu_gower_3 <- clu_gower_3$cluster
```

3.3 K-means (Gower), 4 segments

Q17: Using the Gower distance matrix, perform a k-means cluster analysis with K = 4 clusters. Save the result to `clu_gower_4`. Finally, add the cluster assignments to the orginal dataframe, `DF` – name the column `clu_gower_4`: *(1 point)*

```r
clu_gower_4 <- kmeans(DF_gower, centers = 4, nstart = 10)
DF$clu_gower_4 <- clu_gower_4$cluster
```

4 Select number of clusters

4.1 Elbow plot

Here will will use an elbow plot to assist with determining the number of clusters. Generate an elbow plot for 1 to 10 clusters.

Recall that the elbow plot graphs the within-cluster sum of squares vs. the number of clusters. You can access the within-cluster sum of squares using `$withinss`, as in `clu_gower_2$wihtinss`. Note further that the within-cluster sum of squares returned from `$withinss` is a *list*, with 1 list element per cluster – so, to get the total (across clusters) within-cluster sum of squares, we would for example calculate `sum(clu_gower_2$wihtinss)`.

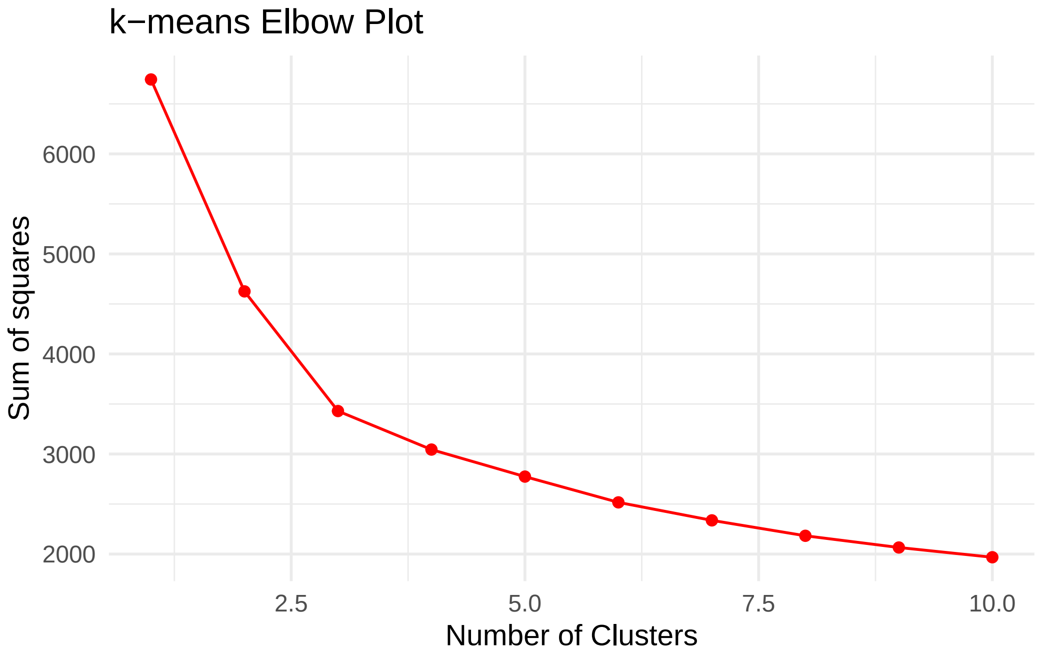
Hint: A loop is a straightforward way to approach this problem.

Q18: Make the elbow plot *(3 points)*

```r
max_clusters <- 10

wss <- rep(0, max_clusters)

for (i in 1:max_clusters) {
segments <- kmeans(DF_gower, centers = i, nstart=10)
wss[i] <- sum(segments$withinss)
}

as.data.frame(wss)
```

```
        wss
1  6743.922
2  4625.450
3  3429.508
4  3044.358
5  2774.201
6  2516.562
7  2336.829
8  2182.590
9  2065.548
10 1967.977
```

```r
elbow_data <- data.frame(k = 1:max_clusters, WCSS = wss)

ggplot(elbow_data, aes(x = k, y = WCSS)) +
  geom_line(color = "red") +
  geom_point(color = "red") +
  labs(title = "k-means Elbow Plot",
       x = "Number of Clusters",
       y = "Sum of squares") +
  theme_minimal()
```



Q19: We are going to use 3 clusters from here. Why did I choose that? *(1 point)*

Answer

We chose 3 clusters because beyond this, the reduction in the sum of squares diminishes significantly. This suggests that adding more than three clusters doesn't enhance the effectiveness of grouping the data.

5 Profile and interpret the clusters

The final stage of the cluster analysis is to profile the clusters and analyze the results. Profiling a cluster entails two things:

1. Calculating the market share associated with the cluster (segment).
2. Calculating cluster (segment) centroids, i.e. the mean variable values across all cluster members.

We analyze cluster profiles primarily by assessing them with respect to the segmentation criteria:

1. Substantial – Segment market shares are large enough to warrant serving. A counterexample for 3 segments might be market shares of 98%, 1% and 1%. Unless the 1% segments are known to be associated with very high willingness to pay customers, such a scheme would have little practical value.
2. Actionable – Segment characteristics can be translated into targeted marketing policies (e.g. using age/income differences to craft different promotional vehicles). Targeted policies must also be consistent with firm competencies.
3. Differentiable – Differences between segments should be clearly defined. That is, differences across segments must be large enough to generate different (actionable) marketing policies.

5.1 K-means (Gower), 3 segments

NOTE: In case you were wondering, the labeling of segments is arbitrary – i.e., the segment with 55.2% of the customers could have been labeled segment 1 or segment 2. Some software packages use the convention that segments are labeled in order of decreasing size – R is apparently not one of them.

Calculate and print the fraction of customers assigned to each of the K = 3 segments. **Q20: 1 point**

Calculate and print the cluster centroids (mean values of the varaibles for customers in the segment). **Q21: 1 point**

```
set.seed(123)
segments_3 <- kmeans(DF_gower, centers = 3, nstart=10)
DF$cluster <- segments_3$cluster
DF |>
  group_by(cluster) |>
  summarise(size = n(),
            proportion = round(n()/nrow(DF), 3))
```

```
# A tibble: 3 x 3
  cluster  size proportion
    <int> <int>      <dbl>
1       1   471      0.471
2       2   108      0.108
3       3   421      0.421
```

```
DF |>
  group_by(cluster) |>
  summarise(across(c(log_spend_online, log_spend_retail, age, white, college, male, hh_inc
  round(3)
```

```
# A tibble: 3 x 8
  cluster log_spend_online log_spend_retail   age white college  male hh_inc
    <dbl>            <dbl>            <dbl> <dbl> <dbl>   <dbl> <dbl>  <dbl>
1       1             0.35             3.96  41.6 0.819   0.591 0      105.
2       2             2.39             2.42  40.9 0.686   0.5   0.843   98.8
3       3             4.36             1.37  40.1 0.806   0.502 0       85.9
```

Q22: Attempt to label the segments in the most descriptive but brief terms possible (e.g. "online affluent") *(1 point)*

Answer

Segment 1 - Retail Shoppers Segment 2 - Online & Retail Shoppers Segment 3 - Highest Online Shoppers

Q23: Which segment is biggest? smallest? How do those segments differ in characteristics? *(1 point)*

Answer

Biggest segment - Cluster 1 (retail shoppers)

Smallest segment - Cluster 2 (online and retail shoppers)

Q24: Evaluate these segments on the basis of the segmentation criteria (substantial, actionable, differentiable) *(1 point)*

Substantial: Each segment possesses sufficient size to be deemed significant and potentially worth targeting.

Actionable: The segments exhibit distinct characteristics regarding online and retail spending, making them worthy for targeted marketing strategies.

Differentiable: The segments are clearly distinguishable, showcasing clear variations in their spending patterns.

Final 10 points

Repeat this task for the 2 and 4 segment solutions. Then, explain how these differ, and what scheme you would recommend using (2, 3, or 4)

```
#2 Segment
set.seed(123)
segments_2 <- kmeans(DF_gower, centers = 2, nstart=10)
DF$cluster <- segments_2$cluster
DF |>
  group_by(cluster) |>
  summarise(size = n(),
            proportion = round(n()/nrow(DF), 3))
```

```
# A tibble: 2 x 3
  cluster  size proportion
    <int> <int>      <dbl>
1       1   529      0.529
2       2   471      0.471
```

```
DF |>
  group_by(cluster) |>
  summarise(across(c(log_spend_online, log_spend_retail, age, white, college, male, hh_inc
  round(3)
```

```
# A tibble: 2 x 8
  cluster log_spend_online log_spend_retail   age white college  male hh_inc
    <dbl>            <dbl>            <dbl> <dbl> <dbl>   <dbl> <dbl>  <dbl>
1       1            0.501             4.00  41.6 0.819   0.593 0.062  105.
2       2            4.23              1.25  40.1 0.778   0.488 0.123   86.4
```

Segment 1 - Retail Shoppers Segment 2 - Online Shoppers

Biggest segment - Cluster 1 (Retail Shoppers) Smallest segment - Cluster 2 (Online shoppers)

Substantial – Each segment is big enough and potentially worth targeting. Actionable – segments display distinct characteristics of online and retail spending. Differentiable – The segments are differentiable, with clear distinctions in their spending.

```
#4 Segment
set.seed(123)
segments_4 <- kmeans(DF_gower, centers = 4, nstart=10)

DF$cluster <- segments_4$cluster

DF |>
  group_by(cluster) |>
  summarise(size = n(),
            proportion = round(n()/nrow(DF), 3))
```

```
# A tibble: 4 x 3
  cluster  size proportion
    <int> <int>      <dbl>
1       1   430      0.43
2       2   300      0.3
3       3   164      0.164
4       4   106      0.106
```

```
DF |>
  group_by(cluster) |>
  summarise(across(c(log_spend_online, log_spend_retail, age, white, college, male, hh_inc
  round(3)
```

```
# A tibble: 4 x 8
  cluster log_spend_online log_spend_retail   age white college  male hh_inc
    <dbl>            <dbl>            <dbl> <dbl> <dbl>   <dbl> <dbl>  <dbl>
1       1            0.032            3.91   41.6 0.819   0.584 0      103.
2       2            4.16             0.018  40.4 0.803   0.486 0       85.0
3       3            4.55             4.64   39.9 0.812   0.572 0       98.1
4       4            2.37             2.43   41.0 0.69    0.499 0.858   98.0
```

Segment 1 - Retail Shoppers Segment 2 - Online Shoppers Segment 3 - Online and Retail Shoppers Segment 4 - Online and Retail Shoppers

Biggest segment - Cluster 1 (Retail Shoppers) Smallest segment - Cluster 4 (Online and Retail shoppers)

Substantial – Each segment is big enough and potentially worth targeting. However, Segment 2 and 4 look very similar and there is not much difference among them. Actionable – Not all segments display distinct characteristics in terms of online and retail spending. Differentiable – all the segments are not differentiable, with clear distinctions in their.