Workshop 6 - Solution

Cluster Analysis with K-means

MSBX-5130: Customer Analytics

2/27/2020

1) Objectives & setup

- Workshop tasks:
  1) Define a segmentation scheme for a women's apparel brand
  2) 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_cust_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
`hh_inc`	median income of households in customer zip code ('000)

Workshop task workflow

1. Setup
   1. Download data & R Markdown files
2. Summarize and prepare data
   1. Histograms of all variables

3. Clustering steps
    1. Select variables to use for clustering
        1. Log-transformation of skewed variables
        2. Create dataframe with finalized cluster variables (only)
    2. Define distance measure between individuals
        1. Euclidean distance
        2. Gower distance
    3. Select clustering procedure
        1. K-means (Gower), 2 segments
        2. K-means (Gower), 3 segments
        3. K-means (Gower), 4 segments
    4. Select number of clusters
        1. Elbow plot
    5. Interpret and profile the clusters
        1. K-means (Gower), 2 segments
        2. K-means (Gower), 3 segments
        3. K-means (Gower), 4 segments
4. Segmentation recommendation
    1. Robustness check: compare Gower solution to Euclidean solution

1.1) Download data & R Markdown file

If you have not already done so, download the data files: `apparel_cust_data.csv` from Canvas. Also download this R markdown file, `Workshop6.Rmd`.

Now launch RStudio, and change the working directory to where you have downloaded the previously mentioned files.

2 Summarize and prepare data

First, we load the data into a dataframe named `DF_in` and generate summary statistics using `describe()`.

```
DF_in = read.csv('apparel_cust_data.csv')
library(psych)
describe(DF_in)
```

```
             vars    n    mean      sd  median trimmed     mad  min      max
iid             1 1000 5463.29 3046.41 5430.50 5486.45 3854.02 14.0 10589.00
spend_online    2 1000   72.44  177.19   14.97   32.67   22.20  0.0  1985.75
spend_retail    3 1000   78.00  182.41   27.71   39.69   41.09  0.0  2421.91
age             4 1000   40.91   11.81   41.00   40.72   11.86 18.0    88.00
white           5 1000    0.80    0.20    0.86    0.83    0.15  0.0     1.00
college         6 1000    0.54    0.22    0.56    0.55    0.25  0.0     1.00
male            7 1000    0.09    0.29    0.00    0.00    0.00  0.0     1.00
hh_inc          8 1000   96.25   50.18   87.36   91.28   45.29  2.5   250.00
                range  skew kurtosis    se
iid          10575.00 -0.05    -1.17 96.34
spend_online  1985.75  5.50    40.06  5.60
spend_retail  2421.91  6.12    50.75  5.77
```

```
age             70.00  0.23     0.13  0.37
white            1.00 -1.68     3.00  0.01
college          1.00 -0.27    -0.81  0.01
male             1.00  2.84     6.07  0.01
hh_inc         247.50  0.95     0.84  1.59
```

Discussion:

- Which (continuous) variables stand out in terms of being high-skew?

The ***spend_online*** *and* ***spend_retail*** *variables stand out as being high-skew. Hence, they are candidates to be log-transformed before doing our clustering analysis. Note that, in general, there is no motivation to transform binary or categorical variables (such as* ***male****) – discrete outcomes are handled naturally (without transformation) by distance metrics such as the Gower distance.*

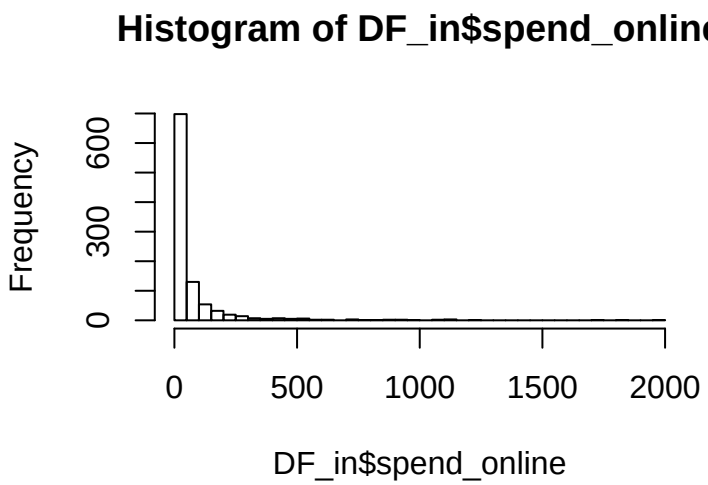
- What is the minimum value of the high-skew variables?

Zero is the minimum value of ***spend_online*** *and* ***spend_retail****. This will be important for how we log-transform the variables, since* ***log()*** *is only defined for strictly positive values. Therefore, in cases when we want to log-transform a variable whose minimum is zero (or megative), we will add a constant to all values such that the minimum value is positive.*

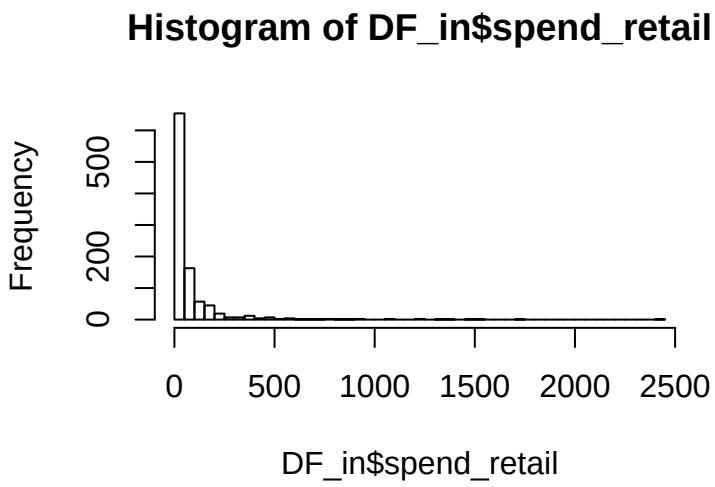
2.1 Histograms of all variables

Next, we wish to inspect the distribution of all the variables we might use for the cluster analysis. We do this by generating histograms of each variable:

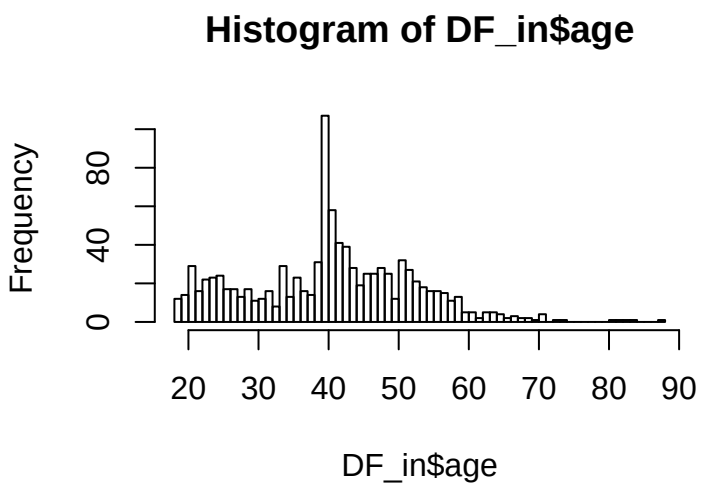
```
# histograms of variables
nbin = 50
hist(DF_in$spend_online,nbin)
```



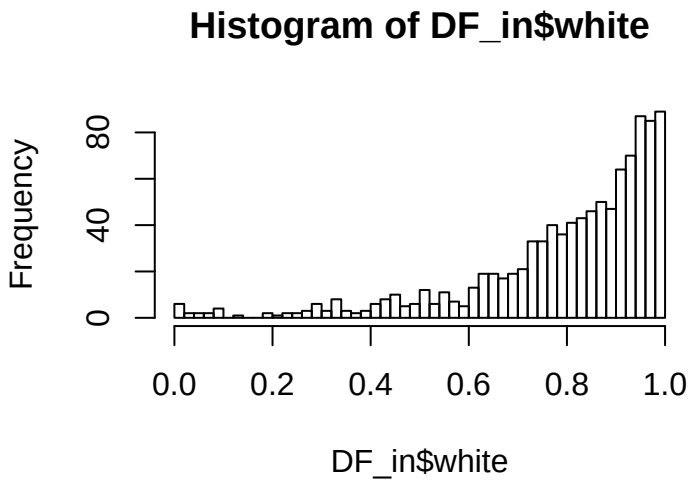

```
hist(DF_in$spend_retail,nbin)
```



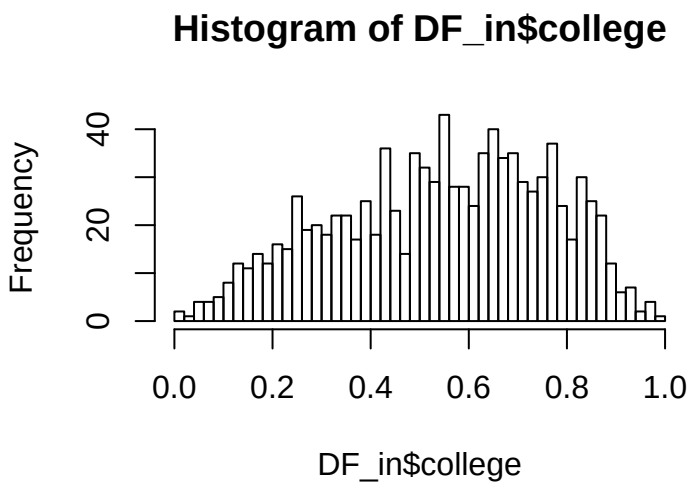

```
hist(DF_in$age,nbin)
```



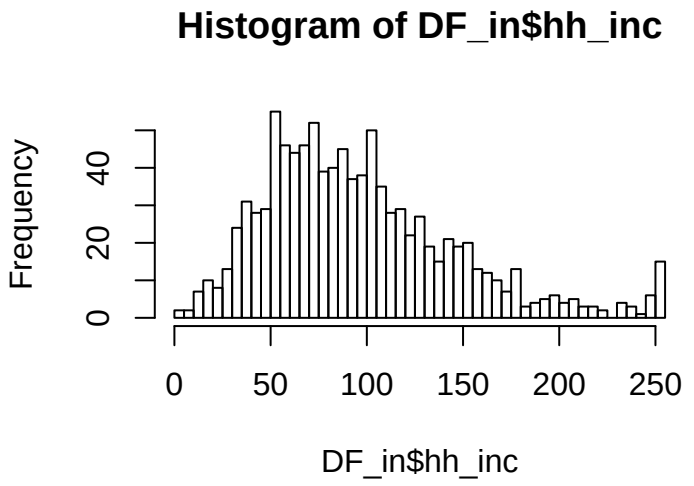

```
hist(DF_in$white,nbin)
```



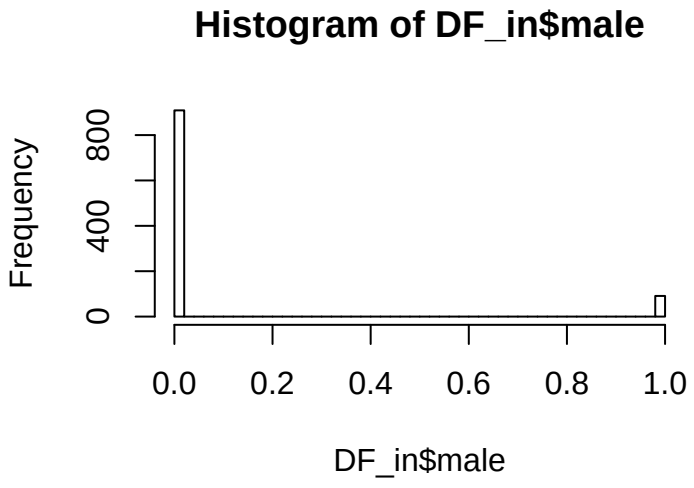
```
hist(DF_in$college,nbin)
```




```
hist(DF_in$hh_inc,nbin)
```




```
hist(DF_in$male,nbin)
```



Discussion:

- By inspecting the histograms, which variables are continuous? Binary?

`male` is a binary variable. The remaining variables are continuous. Since the dataset contains a mixture of binary and continuous variables, we will prefer the Gower distance metric for our cluster analysis (assuming all variables are used).

- Which variables demonstrate high-skew in their histograms?

As was indicated from the summary statistics, we can see visual evidence of strong rightward skew `spend_online` and `spend_retail` in the respective histograms. Recall that rightward skew implies the distribution has a long right tail (probability mass is more widely dispersed to the right of the mean than to the left).

- 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)?

The `spend_online` and `spend_retail` variables display high skew and thus should be log-transformed before being used in cluster analysis. Since we have a mixture of binary and continuous variables, we should use the Gower distance metric (as opposed to the Euclidean distance metric).

3 Clustering steps

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

3.1 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.

3.1.1 Log-transformation of skewed variables

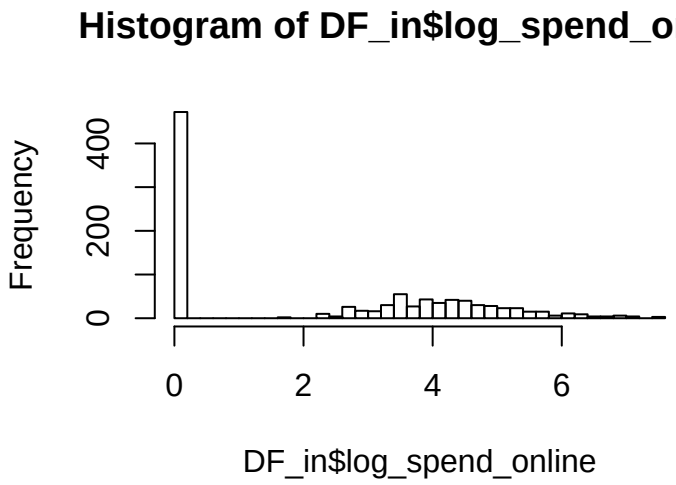
Having observed the distributions of the variables, two stand out as different from the rest: `spend_online` and `spend_retail`. These variables appear to be highly (right) skewed. Since clusting algorithms tend to perform poorly with highly skewed variables, we will transform them in a way that reduces 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 Here, we have the additional complication that the minmum value of `spend_online` and 'spend_retail is zero, and log of zero (or negative numbers) is numerically undefined. To deal with both problems, we transform the expenditure levels by taking log(1+x), where x is the untransformed variable.

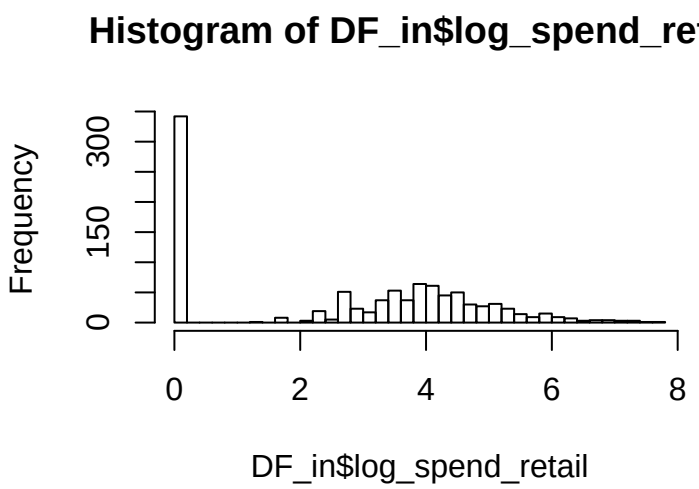
Specifically, to the dataframe `DF_in`, add variables named `log_spend_online` and `log_spend_retail` by taking the `log(1+x)` transformation of each variable. Plot histograms for `log_spend_online` and `log_spend_retail`.

```r
# log-transform skewed spend variables
DF_in$log_spend_online = log(1+DF_in$spend_online)
DF_in$log_spend_retail = log(1+DF_in$spend_retail)

# histograms
hist(DF_in$log_spend_online,nbin)
```




```r
hist(DF_in$log_spend_retail,nbin)
```



Discussion:

- 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)?

We notice that the transformed variables have a much compressed range (as expected when taking logarithms), and away from zero, they look roughly normally distributed. We also see that we have bi-modal distributions in that there is a mode (peak) at zero and another around 4 (log-scale, about exp(4) = $55). The presence of bi-modality is another reason that the Gower distance metric is more appropriate for these data than the Euclidean distance metric.

3.1.2 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.

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

```
# create dataframe with transformed variables, omit non-cluster variables
DF = DF_in
DF$iid = NULL
DF$spend_online = NULL
DF$spend_retail = NULL
```

3.2 Define distance measure between individuals

3.2.1 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).

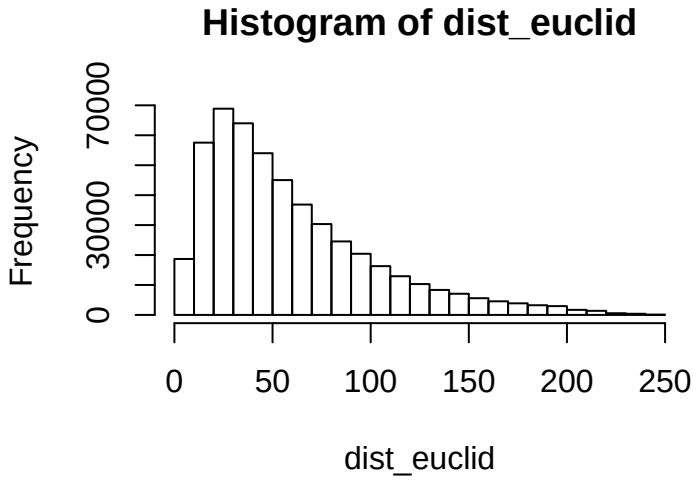
Rather than thinking of points in physical space, we can think of points in "characteristic" space. For example, the x-axis could represent a person's age and the y-axis could represent a person's income. The "distance" between two people in this case would be the square root of the squared difference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

Calculate the (unstandardized) Euclidean distance between all pairs of consumers across the variables in dataframe `DF` (as defined in 3.1.2). The `daisy()` function from the `cluster` package can be useful for this task. Call the resulting list of pairwise distances `dist_euclid`. Also, generate a histogram of `hist_euclid`.

```
library(cluster)
dist_euclid = daisy(DF,metric = "euclidean",warnType=FALSE)
hist(dist_euclid)
```



Discussion:

- Characterize the shape of the Euclidean distance distribution. Does it appear normally-distributed? Is it skewed? What are the (approximate) minimum and maximum values?

The Euclidean distance distribution is highly skewed and hence does not appear to be normally-distributed. Values range from (approximately) zero to 250.

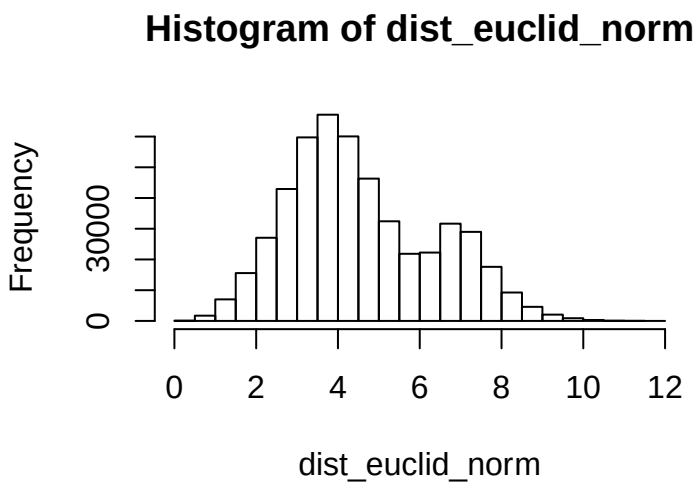
3.2.1.1 Standardized Euclidean distance

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).

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 `dist_euclid_norm`. Also, generate a histogram of `hist_euclid_norm`.

```
dist_euclid_norm = daisy(DF,metric = "euclidean",stand=TRUE,warnType=FALSE)
hist(dist_euclid_norm)
```



Discussion:

- Characterize the shape of the standardized Euclidean distance distribution. Does it appear normally-distributed? Is it skewed? What are the (approximate) minimum and maximum values? How does it compare to the non-standardized distance distribution?

The standardized Euclidean distance distribution appears to be (approximately) normally-distributed (and hence not very skewed). We would therefore expect standardized distances to deliver superior cluster solutions, as compared to non-standardized distances. Values range from (approximately) zero to 10 – the range/scale of computed distances is much smaller than the non-standardized distance distribution.

3.2.2 Gower distance

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

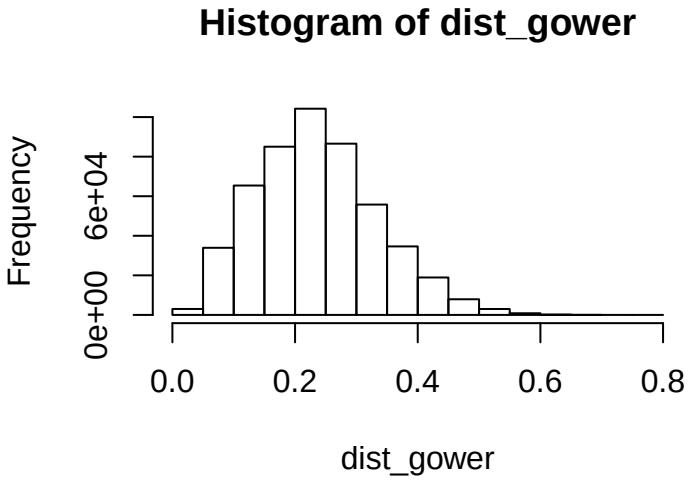
For mixed continuous & binary data, a better option is the Gower distance metric, which defines the distance between individuals i and j on variable k (e.g. age, income, etc.) as follows:

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

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.

Calculate the Gower distance between all pairs of consumers across the variables in dataframe `DF` (as defined in 3.1.2). Call the resulting list of pairwise distances `dist_gower`. Also, generate a histogram of `dist_gower`.

```
library(cluster)
dist_gower = daisy(DF,metric = "gower",warnType=FALSE)
hist(dist_gower)
```



Discussion:

- Characterize the shape of the Gower distance distribution. How does it compare to the Euclidean distance distributions?

The Gower distance distribution looks approximately normally distributed, and hence is similar to the standardized Euclidean distance distribution.

- Based on the data types in `DF` and the shapes of the distance distributions, which distance metric is most appropriate for use with our input data (`DF`)?

Since we have a mixture of binary (`male`) and continuous (all other) variables, the Gower distance metric is the preferred choice. This is epecially true given the bi-modal nature of the transformed expenditure data.

3.3 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.

Hierarchical methods (such as `hclust()` in R) typically build up clusters (groupings) of individuals by successively adding individuals to clusters, starting with the individuals associated with the smallest pairwise distances.

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 3.5.

3.3.1 K-means (Gower), 2 segments

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_in` – name the column `clu_gower_2`:

```
clu_gower_2 = kmeans(dist_gower, centers=2, nstart=10)
DF_in$clu_gower_2 = clu_gower_2$cluster
```

3.3.2 K-means (Gower), 3 segments

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_in` – name the column `clu_gower_3`:

```
clu_gower_3 = kmeans(dist_gower, centers=3, nstart=10)
DF_in$clu_gower_3 = clu_gower_3$cluster
```

3.3.3 K-means (Gower), 4 segments

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_in` – name the column `clu_gower_4`:

```
clu_gower_4 = kmeans(dist_gower, centers=4, nstart=10)
DF_in$clu_gower_4 = clu_gower_4$cluster
```

3.4 Select number of clusters

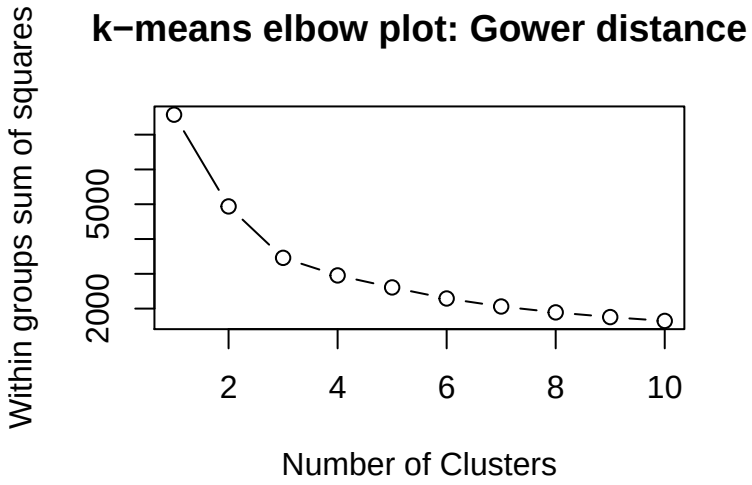
3.4.1 Elbow plot

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.

```
# elbow plot for k-means using gower distance
Nclus = 10                                  # max number of clusters to test
wss = rep(0,Nclus)                          # list to hold within-cluster sum-of-squares
for (i in 1:Nclus) {                        # loop over cluster
  res = kmeans(dist_gower, centers=i, nstart=10)
  wss[i] = sum(res$withinss)                # within-cluster sum-of-squares, summed over clusters
}
plot(1:Nclus, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares",
     main="k-means elbow plot: Gower distance")
```



Discussion:

- Using the plot, search for an "elbow" point that identifies the best "bang for buck" in terms of the number of clusters
  - Note: Often, there may be more than one candidate elbow point
  - In such cases, we typically evaluate all candidate elbow points and decide among solutions on the basis of the segmentaion criteria

While not as pronounced as the example in the slides, the "elbow" point in this graph appears to be around 3 or 4 clusters. Based on this graph, we would probably investigate clustering solutions with 3, 4 and possibly 5 segments. We would then assess these clustering solutions according to the segmentation criteria discussed in lecture.

3.5 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). Recall from the lecture slides that the `table()` function can be useful for this task.
2. Calculating cluster (segment) centroids, i.e. the mean variable values across all cluster members. Recall from the lecture slides that the `aggregate()` function can be useful for this task.

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 counter-example 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.

3.5.1 K-means (Gower), 2 segments

Calculate and print the fraction of customers assigned to each of the K = 2 segments.

Calculate and print the cluster centroids (mean values of the variables for customers in the segment). Note that we are interested in the mean values of un-transformed variables (as captured in `DF_in`).

```
table(DF_in$clu_gower_2)/dim(DF_in)[1]
```

```
    1     2
0.448 0.552
```

```
round(aggregate(
  cbind(spend_online,spend_retail,age,white,college,male,hh_inc)~clu_gower_2,
          data=DF_in,FUN=mean),3)
```

```
  clu_gower_2 spend_online spend_retail    age white college  male  hh_inc
1           1      146.864       54.800 40.022 0.776   0.482 0.123  84.861
2           2       12.038       96.838 41.629 0.819   0.594 0.065 105.500
```

Discussion:

- Attempt to label the segments in the most descriptive but brief terms possible (e.g. "online affluent")

There are no "incorrect" answers here, but some descriptions are more useful than others.

What are the main differences in segments 1 & 2? Referring to online/retail expenditures, we see segment 2 is primarily retail (limited online spending – 8:1 retail:online expenditures) while segment 1 is primarily online (1:3 retail:online expenditures). Differences in the demographic variables across the segments is moderate, with income having the largest magnitude difference. Based on these observations, we might summarize the segments simply as "1: primarily online" and "2: primarily retail".

- Which segment is biggest? smallest? How do those segments differ in characteristics?

Segment 2 is biggest, though not by much – segment 2 represents 55.2% of the sample, while segment 1 represents 44.8%

As mentioned previously, the biggest differences relate to the expenditure patterns. This is expected, as "behavioral data" (such as past purchase behavior) tends to provide more discriminating power than demographic information.

- Evaluate these segments on the basis of the segmentation criteria (substantial, actionable, differentiable)

Substantial – all (2) segments are of appreciable size, so we conclude the scheme is adequate on this dimension

Actionable – Differences in online/retail channel preferences certainly suggest one means to implement different policies. For example, for "primarily online" customers, we could use targeted e-mail promotions, while for "primarily retail" customers we could use printed coupons that are issued and redeemed only at retail outlets. Demographic information seems somewhat less useful here, as differences are smaller with the possible exception of income (which, curiously, associates higher income levels with lower overall expenditure

levels). In short, the online/retail distinction is useful but otherwise we have limited information further customize promotional appeals, etc.

Differentiable – The two segments are differentiable, primarily based on online/retail expenditure levels, but since neither segment is purely online or purely retail, we might hope for more differentiation between the 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.

3.5.2 K-means (Gower), 3 segments

Calculate and print the fraction of customers assigned to each of the K = 3 segments.

Calculate and print the cluster centroids (mean values of the varaibles for customers in the segment).

```
table(DF_in$clu_gower_3)/dim(DF_in)[1]
```

```
    1     2     3
0.470 0.107 0.423
```

```
round(aggregate(
  cbind(spend_online,spend_retail,age,white,college,male,hh_inc)~clu_gower_3,
          data=DF_in,FUN=mean),3)
```

```
  clu_gower_3 spend_online spend_retail    age white college male  hh_inc
1           1        3.731       93.536 41.649 0.819   0.592 0.00 105.128
2           2       89.758       83.117 40.505 0.694   0.498 0.85  97.577
3           3      144.403       59.455 40.189 0.804   0.501 0.00  86.059
```

Discussion:

- Attempt to label the segments in the most descriptive but brief terms possible (e.g. "online affluent")

With a third segment, we see sharper distinctions across segments. While segments 1 and 3 here are fairly close in composition (similar centroids) to segments 1 and 2 in the 2-segment clustering solution, segment 2 here is quite distinct in that the algorithm has assigned all men to this segment (plus a few "stray" women). Men apparently tend to spend fairly equally across the channels, so the male segment is also a multi-channel segment. The two other segments are entirely female, with one being almost exclusively retail and the other being multi-channel, but primarily online spending. Interestingly, higher incomes are associated with primarily retail shoppers, and lower total (online+retail) expenditure levels. So, some characteristic labels might be: "1: women - retail", "2: men - multi-channel", "3: women - multi-channel".

- Which segment is biggest? smallest? How do those segments differ in characteristics?

Segment 1 ("women - retail") is largest (47.0%), while segment 2 ("men - multi-channel") is the smallest. Again, the main differences stem from differences in the channel expenditure patterns, and income to a lesser extent.

- Evaluate these segments on the basis of the segmentation criteria (substantial, actionable, differentiable)

Substantial – segments 1 & 3 are of appreciable size, while segment 2 (men) is smaller (~11%). While we might have concern for the male segment being somewhat small for targeting purposes, 11% is not so small for this to be an acute concern. We typically will not worry about segments not being substantial until segment sizes are less than 3-5% of the total market. Plus, male expenditure levels are rather high and call for distinctive appeals (gifts) that are both "targetable" and potentially profitable.

Actionable – Moving from 2 to 3 segments has helped somewhat vis-a-vis actionability. In particular, we now have segments that are divided on gender lines (providing strong cues for different promotional appeals) as well as channel preference lines.

Differentiable – Here too matters are improved in that clusters are more distinct in terms of differences in channel expenditure patterns and gender.

3.5.3 K-means (Gower), 4 segments

Calculate and print the fraction of customers assigned to each of the K = 4 segments.

Calculate and print the cluster centroids (mean values of the varaibles for customers in the segment).

```
table(DF_in$clu_gower_4)/dim(DF_in)[1]
```

```
    1     2     3     4
0.428 0.303 0.167 0.102
```

```
round(aggregate(
  cbind(spend_online,spend_retail,age,white,college,male,hh_inc)~clu_gower_4,
          data=DF_in,FUN=mean),3)
```

```
  clu_gower_4 spend_online spend_retail    age white college  male  hh_inc
1           1        0.023       88.895 41.561 0.816   0.582 0.000 102.475
2           2      110.368        0.033 40.370 0.795   0.484 0.000  85.082
3           3      188.384      195.402 40.257 0.820   0.578 0.000  99.215
4           4       73.808       71.721 40.843 0.710   0.505 0.892  98.485
```

Discussion:

- Attempt to label the segments in the most descriptive but brief terms possible (e.g. "online affluent")

Again clusters become more distinct as we move from 3 to 4 segments. Note that here, segments 2 and 4 closely resemble segments 1 and 2 from the 3-segment solution, respectively ("women - retail", "men - multi-channel"). Segments 1 and 3 here appear to originate primarily from splitting segment 3 in the 3-segment solution.

We now have three all female segments: a multi-channel segment one with the highest expenditure levels, a online channel segment with moderate expenditure levels, and a retail channel segment with moderate expenditure levels. There is also a male multi-channel segment. So, a complete set of labels might be: "1: women - retail", "2: women: online", "3: women - multi-channel", "4: men (multi-channel)"

- Which segment is biggest? smallest? How do those segments differ in characteristics?

Segment 1 ("women - retail") is the largest, while segment 4 ("men (multi-channel)") is the smallest. These segments clearly differ in terms of gender as well as channel preference.

- Evaluate these segments on the basis of the segmentation criteria (substantial, actionable, differentiable)

Substantial – similar remarks as with 3-segment solution

Actionable – Moving from 3 to 4 segments has helped somewhat vis-a-vis actionability. In particular, within multi-channel women, we now have two segments that are distinguished by education and income. These differences can help with crafting promotional appeals.

Differentiable – Here too matters are improved in that clusters are more distinct in terms of differences in channel expenditure patterns, gender and education/income.

4 Segmentation recommendation

Discussion:

- Which segmentation scheme do you recommend? Why?

In this case, I personally favor the 4 segment solution, based on the apparent fit with the segmentation criteria.

4.1 Robustness check: compare Gower solution to Euclidean solution

Here we check what kind of solution we would get if we ignored the non-continuous and multi-modal nature of our data. Specifically, calculate and summarize the k-means cluster solution with K = 4 segments, but using the Euclidean distance metric.

```
clu_euclid_4 = kmeans(dist_euclid_norm, centers=4)
DF_in$clu_euclid_4 = clu_euclid_4$cluster
table(DF_in$clu_euclid_4)/dim(DF_in)[1]
```

```
    1     2     3     4
0.423 0.328 0.108 0.141
```

```
round(aggregate(
  cbind(spend_online,spend_retail,age,white,college,male,hh_inc)~clu_euclid_4,
          data=DF_in,FUN=mean),3)
```

```
  clu_euclid_4 spend_online spend_retail    age white college  male  hh_inc
1            1       28.929       89.285 41.619 0.857   0.609 0.000  97.485
2            2      110.745       46.789 37.976 0.741   0.382 0.000  60.768
3            3       50.661       57.107 40.917 0.666   0.485 0.843  93.784
4            4      130.548      132.787 45.596 0.863   0.768 0.000 177.001
```

Discussion:

- Does this segmentation solution appear better or worse according to the segmentation criteria?

We notice that when using the Euclidean distance measure, the clustering algorithm does not tend to recognize "purely retail" type customers (like segment 2 in the 4-segment Gower solution). The Euclidean-derived clusters are thus all multi-channel, making them less differentiated vs the Gower-derived clusters.

- Why does this make sense?

This makes sense because the Gower metric is better suited to data with mixtures of binomial and continuous data, and multi-modal data in particular.