Homework 1

Julio Cesar Enciso-Alva

January XX, 2019

Note: This documents uses the package `magrittr`, `dplyr` and `data.table`. It also uses the files `DistMat.csv`, distance matrix from question 3, and `T11-9.txt`, local copy of `cereal` database.

Question 1

a)

The multiplication will be calculated using the matrix product operator `%*%`. In order to do that, the given matrices are saved to the variables `A` and `B`.

```
A = matrix(c(7,9,20, 2,4,10),ncol=3)
B = matrix(c(1,7,12,40, 2,8,13,20, 3,9,14,21),ncol=4)
```

Then, the multiplication is calculated and displayed.

```
A.B = A %*% B
A.B
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  195  352  503  427
## [2,]  143  444  187  319
```

b)

The matrix G will be constructed using the spectral decomposition of A; if we have an ortonormal matrix C and a diagonal matrix D such that

$$A = CDC'$$

then its *square root* can be constructed (and will be proposed as the solution) using the following expression

$$G = CD^{\frac{1}{2}}C'$$

As the first step for doing so, the matrix A is saved to the variable `A`.

```
A = matrix(c(2,1,0, 1,2,0, 0,0,2),ncol=3)
```

The spectral decomposition of A is found using the function `base::eigen`. It is important to recall that the matrices C and D such that $A = CDC'$ can be constructed as

$$C = [x_1, x_2, x_3]$$
$$D = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$$

where x_i are the eigenvectors of A, and λ_i are their corresponding eigenvalues. Both values are saved to the variable `SD`.

```
SD = eigen(A)
C = SD$vectors
D = diag(SD$values)
```

The matrix G, the **proposed solution**, is constructed as described before:

```
G = C %*% sqrt(D) %*% t(C)
G
```

```
##           [,1]      [,2]     [,3]
## [1,] 1.3660254 0.3660254 0.000000
## [2,] 0.3660254 1.3660254 0.000000
## [3,] 0.0000000 0.0000000 1.414214
```

In order to verify that $G^2 = A$, both G^2 and A are displayed

```
A
```

```
##      [,1] [,2] [,3]
## [1,]    2    1    0
## [2,]    1    2    0
## [3,]    0    0    2
```

```
G %*% G
```

```
##      [,1] [,2] [,3]
## [1,]    2    1    0
## [2,]    1    2    0
## [3,]    0    0    2
```

c)

The expression requested is

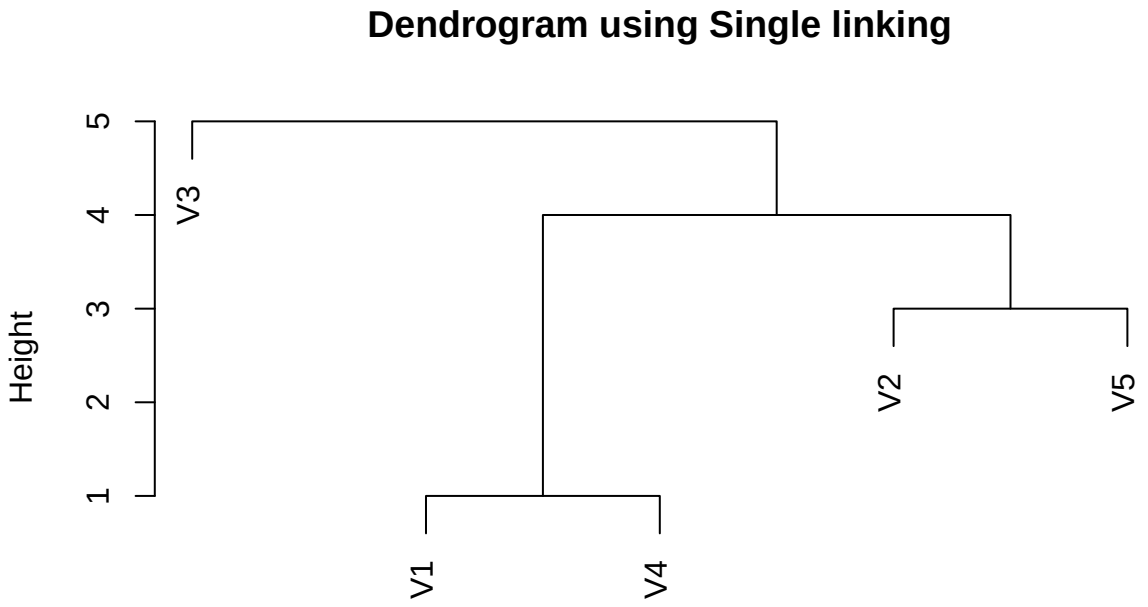
$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}\, dz$$

Question 2

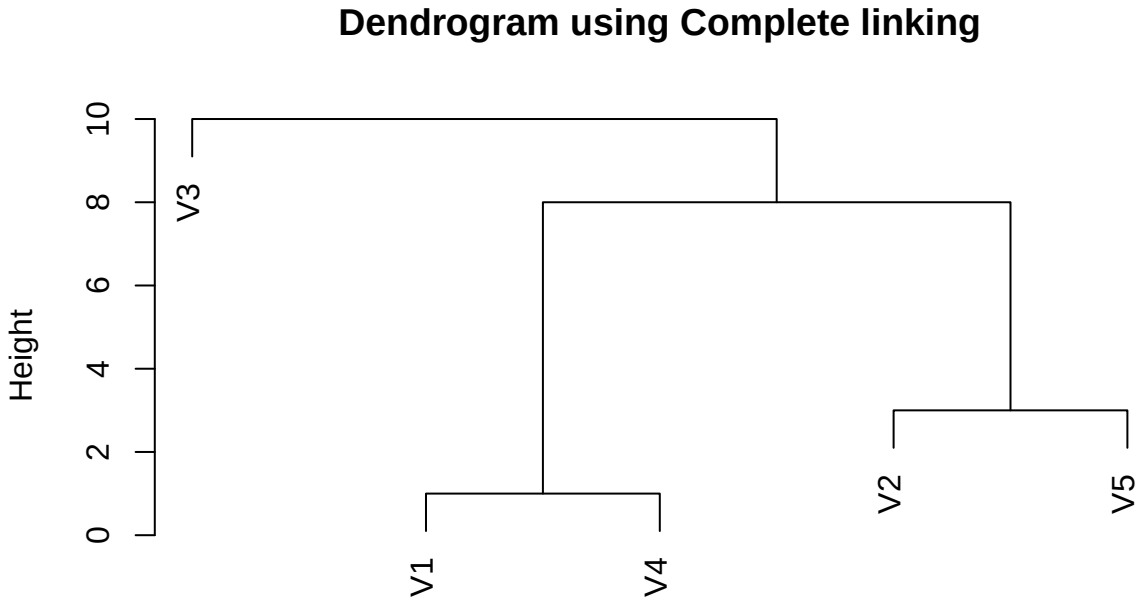
To perform the clusterings, the given distance matrix was saved on the file `DistMat.csv`; the data is then loaded and saved to the variable `d`. For ease of use, only the lower half of the matrix is in the file.

```
d = read.csv('./DistMat.csv',header=F) %>%
  as.matrix() %>% as.dist()
```

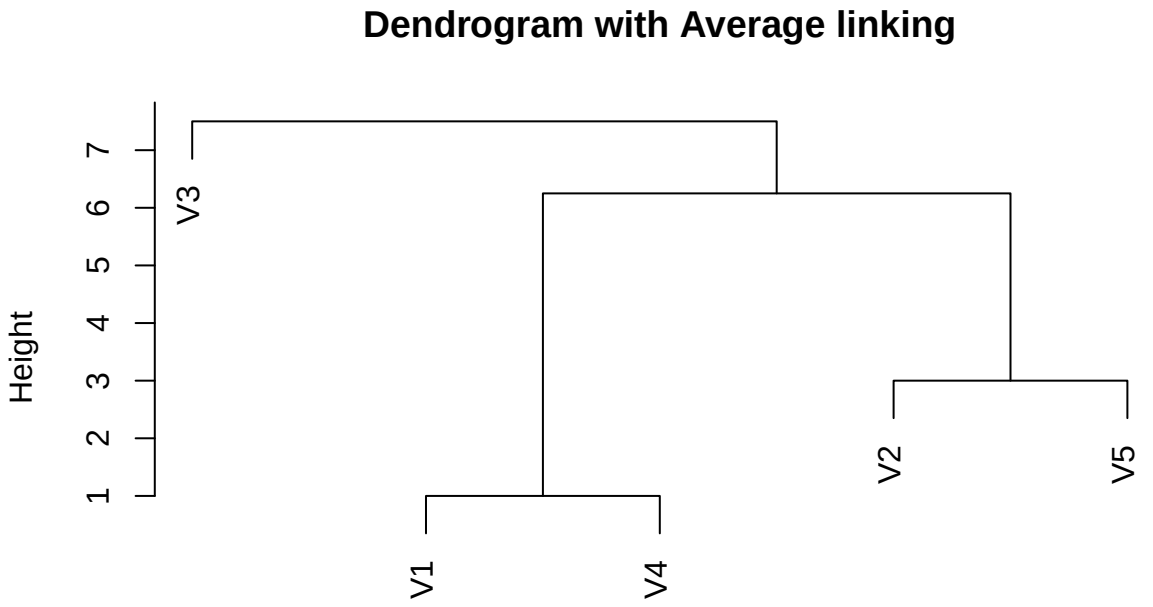
The dendrograms are generated using the function `stats::hclust`, changing the method of clustering.


Dendrogram using Single linking
Height
1
2
3
4
5
V3
V1
V4
V2
V5

.


Dendrogram using Complete linking
Height
0
2
4
6
8
10
V3
V1
V4
V2
V5

.



.

The three dendrograms have the same structure, but the nodes have different heights. This indicates that the clusters are robust.

Question 3

The dataset `cereal` is loaded from a local copy using the suggested code; the column names are also read from a file. The variables `Manufacturer` and `Group are dropped`.

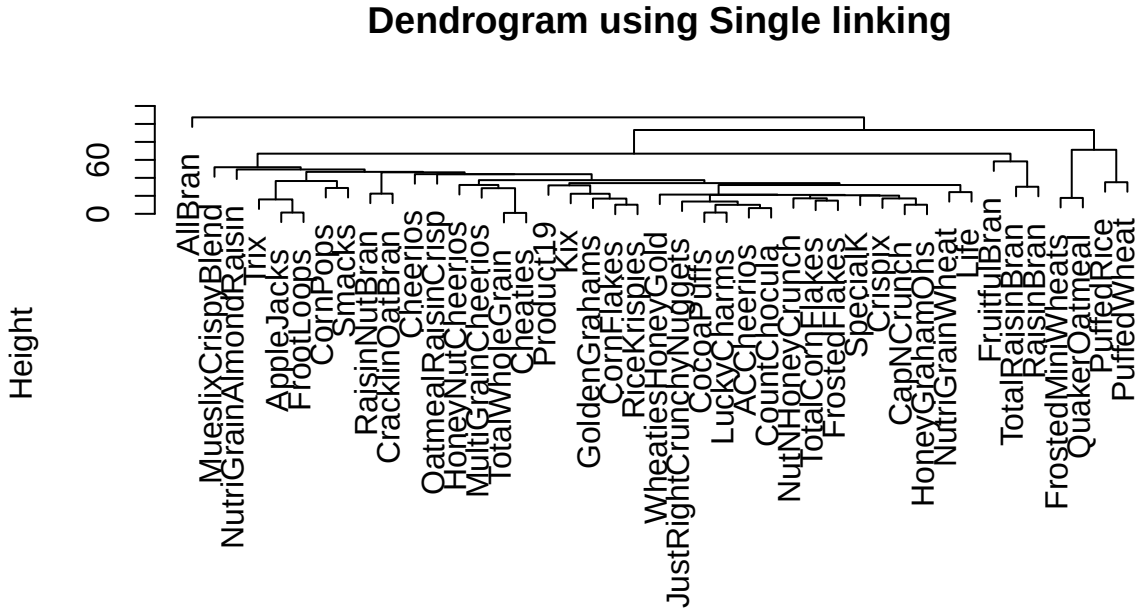
a)

The distance between vectors is calculated using the function `stats::dist` and saved to the variable `d`.

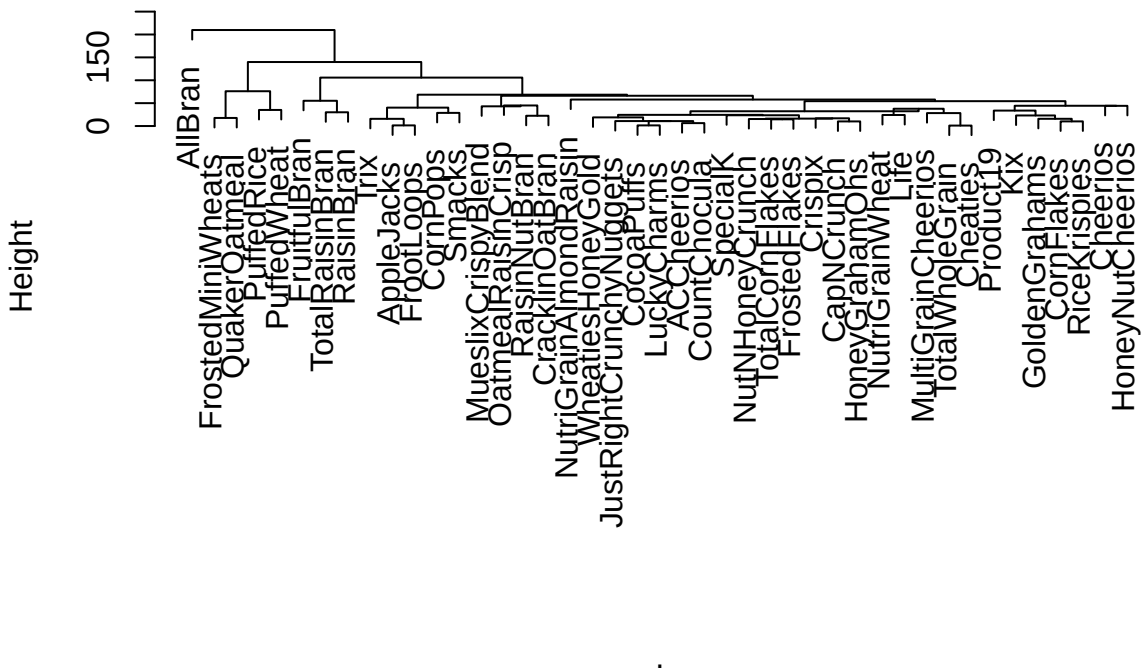
```
d = stats::dist(cereal, method='euclidean')
```

b)

The clustering is achieved using the funcion `stats::hclust`, using different methods.



.



.

The clusters differ with the two methods. There are inversions on the *centroid* clustering, possible related to ~~healthy~~ low-sugar cereals. This difference can be explained by the effect of the variables `Calories`, `Sodium` and `Potassium`; those variables have very high values because of the scale.

Question 4

The dataset `cereal` is loaded from a local copy using the suggested code; the column names are also read from a file. The variables `Manufacturer` and `Group` are dropped. **Note:** This process was done in *Question 3*, but is done again to keep the code independent.

The K-means clustering is done using the function `stats::kmeans`, and saved to a table.

```
set.seed(1)
K2 = kmeans(cereal,centers=2)$cluster
K3 = kmeans(cereal,centers=3)$cluster
```

Question 5

The required vector is generated and saved to the variable `x`, and then the required tasks are completed.

a)

There are 9498 values in `x` between -2 and 2. This value was calculated using the following code

```
length( x[ -2<x & x<2 ] )
```

b)

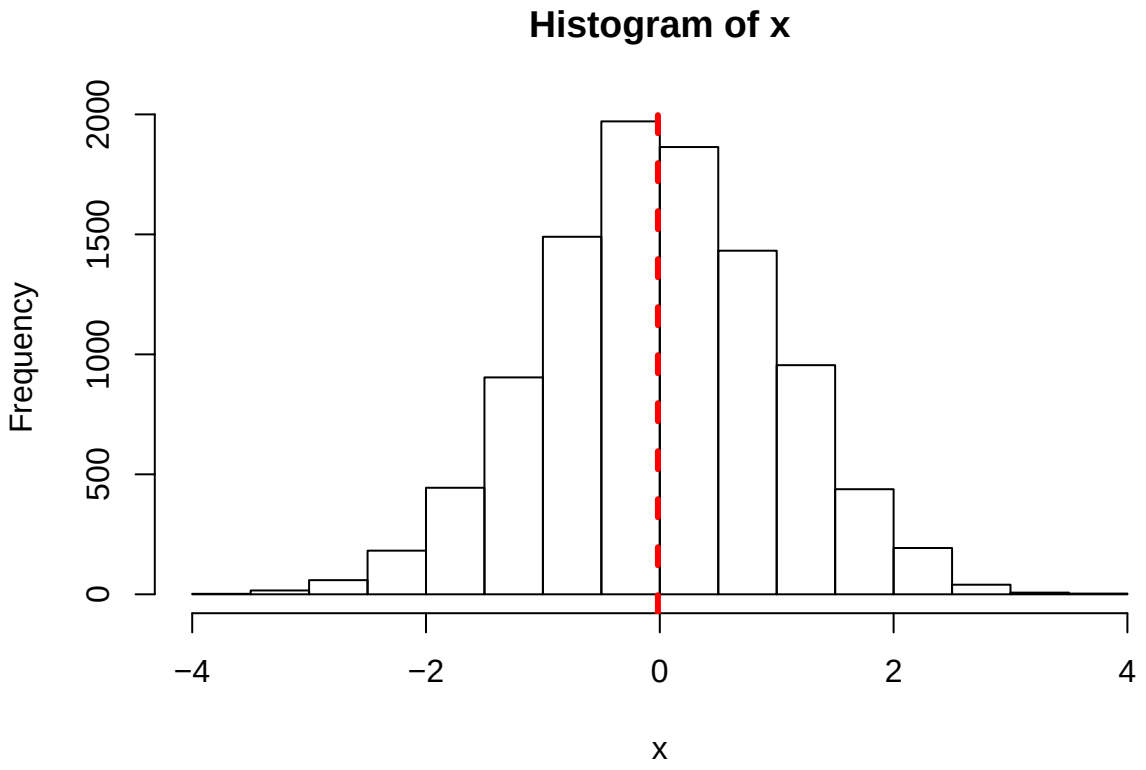
The indexes of `x`, whose values lie are on the set $(-\infty, -3] \cup [3, \infty)$, are saved to the variable `IND` and displayed.

```
IND = c( which(x<=-3 | x>=3) )
IND
```

```
##  [1]  446  495  843 1165 1295 1442 1737 2132 2460 3331 3680 3694 3751 4262
## [15] 5641 5811 5904 6108 6270 6376 7485 7873 8203 8216 8711 8757 9612 9777
```

c)

The histogram is constructed using the function `hist`.



Question 6

The dataset `cereal` is loaded from a local copy using the suggested code; the column names are also read from a file. The variable `Group` is dropped.

Note: This process was done in *Question 3* and *Question 4*, but is done again to keep the code independent, and also because variable `Manufacturer` is not dropped in this question.

a)

The table of means is constructed using the following code, and the displayed.

```
cereal %>%
  dplyr::filter( Protein>=3 & Sugar<=9 ) %>%
  dplyr::group_by(Manufacturer) %>%
  dplyr::summarize(Mean_Calories=mean(Calories)) %>%
  knitr::kable()
```

Manufacturer	Mean_Calories
G	102.5000
K	102.8571
Q	100.0000

b)

The variable `Healthy` is constructed using the following code:

```
cereal$Healthy = 1*(cereal$Protein>=3 & cereal$Sugar<=9)
```