Practice3

Dasha Asienga

Due by midnight, Monday, October 17

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (anything besides our textbook(s), course materials in Moodle, R help menu) to complete this assignment, please acknowledge them below using a bulleted list.

I acknowledge the following individuals with whom I worked on this assignment:

Name(s) and corresponding problem(s)

-

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

-

Prompt

Research scientists have been studying mice to understand which proteins are useful to learning. The data set we have access to is from UCI's machine learning repository, and the data citation is:

Higuera C, Gardiner KJ, Cios KJ (2015) Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome. PLoS ONE 10(6): e0129126.

You should read more about it here in order to understand the variables involved.

Consider this a "data challenge" presented to you as a novice statistician to demonstrate your understanding of clustering. In other words, when writing your intro, you can frame it in the following way. We are being challenged to use at least two different clustering methods to find clusters of mice based on their protein expression levels. Then, after choosing a solution, the challenge is to describe the clusters we found, and comment on the strength/validity of our clustering solution.

(You cannot use self-organizing maps (SOMs) - we didn't cover that method in class.)

Two methods refers to algorithms from different classes, not just changing linkages or numbers of clusters, though you are free to explore those options as well. If you are interested in more flexible distance matrices, you can check out the *daisy* function, or look more into the properties of *dist* via the help menu. Note that in Methods, you will describe the plan to choose k, the number of clusters, and then in the results, actually decide on a value of k. In other words, methods need not include your final decision about a value of k, but should describe the process you will use to pick it.

It is VERY important to note that each mouse participated in 15 experiments, so there are 15 observations per mouse. In the Data Pre-processing chunk below, I have loaded the original data set, created a data set of the average expression levels per mouse, and created a data set of just the first experiment values per mouse. You need to discuss and specify which of these data sets or some other data set of your construction (for example, maybe you want the 5th experiment or median values, not means) you are using for your analysis and why you chose that data set.

The pre-processed data sets contain variables that are NOT protein expression levels, in case you want to use them to help describe the clusters you found. Be careful to use ONLY protein expression level variables in the clustering. There are also MANY variables possible to use. You have to determine which you will include in finding the cluster solution, just be sure they are the protein expression variables.

Be sure to leave the pre-processing section in place, though you may edit as you desire so it only includes the data set you will be using, and note that you'll need to refer back up to it in your write-up when describing the data.

Data Pre-Processing

```
#reads in the data
mice <- read_csv("https://awagaman.people.amherst.edu/stat240/Data_Cortex_Nuclear.csv")
```

```
## Rows: 1080 Columns: 82
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): MouseID, Genotype, Treatment, Behavior, class
## dbl (77): DYRK1A_N, ITSN1_N, BDNF_N, NR1_N, NR2A_N, pAKT_N, pBRAF_N, pCAMKII...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#this adds columns with just the unique mouse ID and experiment number to the data set
splitting <- strsplit(mice$MouseID, "_")
splitdata <- matrix(unlist(splitting), ncol = 2, byrow = TRUE)
#does this by breaking apart the MouseID which contains both pieces of info
mice <- mutate(mice, MouseNum = splitdata[, 1], ExpNum = splitdata[, 2])
```

```r
#Create data set of just average values per mouse
miceavg <- mice %>%
  group_by(MouseNum) %>%
  #computes average values for each mouse
  summarize_at(vars(DYRK1A_N:CaNA_N), mean, na.rm = TRUE)


#now we want to join in the Genotype, Treatment, Behavior, and class info for reference
#values for each mouse are repeated so just take from Experiment 1
miceinfo <- mice %>%
  filter(ExpNum == 1) %>%
  select(MouseNum, Genotype, Treatment, Behavior, class)

miceavg <- inner_join(miceavg, miceinfo)
```

```
## Joining, by = "MouseNum"
```

Introduction

Purpose of the Analysis

For this analysis, we are being challenged to use at least 2 different clustering algorithms to find clusters of mice based on their protein expression levels. There are 77 variables coding protein expression levels in the data set, 1 identifier variable `MouseID`, and 4 categorical variables `Genotype`, `Treatment`, `Behavior`, and `class`. Our aim is to statistically assess whether we can find natural groups/ clusters of mice given information on mice protein expression levels. We can then assess how strong our clustering solution is and compare it with some of the categorical variables (pre-defined groups) available to us in the data set. In the end, our goal is to demonstrate our understanding of clustering by running it on this data set, describe our clustering solution, and finally, test for its validity and robustness.

Aggregating the Full Data Set

The data set `mice` that we are using for this analysis is from UCI's machine learning repository. However, each mouse participated in 15 experiments, so there are 15 observations per mouse. Because our goal is to classify mice based on their protein expression levels, we need only one row per mouse. I have decided to proceed with `miceavg`, which records the average value for each protein expression across the 15 experiments. I made the choice to perform my analysis on the aggregate data because it will capture information from all the experiments and allow for a more representative analysis.

Statistical Techniques

For this analysis, we will be using clustering as our main statistical technique. Clustering refers to any set of methods used to find natural groups of objects. We will specifically be running agglomerative hierarchical clustering and k-means clustering algorithms to find our clustering solution, both of which will be explained in further detail in the methods section.

Because we have many different variables, we will also be employing Principal Components Analysis (PCA), which is a dimension reduction technique, to visualize our final clustering solution.

Preliminary Analysis

Picking Variables

As mentioned in the introduction, there are 77 different quantitative variables coding the mice protein expression levels. Because these are many variables, it will not only be difficult to interpret our solution, but it will also be difficult to reduce the dimensionality of the data set to visualize our solution. Additionally, the number of variables is greater than the number of observations, and so, we cannot run PCA on the data as is. As a result, we will begin by picking which variables to proceed with in order to reduce the dimensionality.

First, we will drop all columns with missing data so that we remain only with the variables for which we have all the information.

```
miceavg <- miceavg %>%
  select_if(~ !any(is.na(.)))
```

Next, we will attempt to pick our variables based on which seem to be the most variable, and hence, would be most likely to explain differences across natural groups.

```
variance <- as.vector(NULL)
var_num <- as.vector(NULL)

for(i in 1:ncol(miceavg)) {
  var_num[i] <- i
  variance[i] <- var(miceavg[ , i])
}

temp <- cbind(var_num, variance) %>%
  as.data.frame()

colnames(temp) = c("num", "var")
```

Let's find the ten most variable variables and use them in our final data set, including our 5 categorical variables.

```
sortedtemp <- temp[order(temp$var, decreasing = TRUE), ]
head(sortedtemp, 10)
```

```
##    num       var
## 9    9 1.5061535
## 6    6 0.4128483
## 46  46 0.3216449
## 19  19 0.2130044
## 25  25 0.2087920
## 11  11 0.1585047
## 12  12 0.1071425
## 70  70 0.0916543
## 49  49 0.0916507
## 32  32 0.0730208
```

Subsetting the Data Set

Based on the method above, we will subset our data into 14 variables, the 10 most variable quantitative variables and our 4 categorical variables. We will also have a second subset with just the quantitative

variables in our data set. Note that the categorical variables will not be included in our clustering analysis. The aim is to assess whether our clustering solution can recover some of those groups.

```
miceavg_sub <- miceavg[ , c(6,9,11,12,19,25,32,46,49,70,71,72,73,74)]
head(miceavg_sub, 2)
```

```
## # A tibble: 2 x 14
##   NR2A_N pCAMKII_N pELK_N pERK_N pPKCAB_N ERK_N SOD1_N pPKCG_N ADARB1_N CaNA_N
##    <dbl>     <dbl>  <dbl>  <dbl>    <dbl> <dbl>  <dbl>   <dbl>    <dbl>  <dbl>
## 1   3.46      2.54   1.56  0.613    1.27   2.78  0.352    1.38     1.33   1.33
## 2   3.47      3.83   1.20  0.303    0.957  2.37  1.12     1.59     1.18   1.26
## # ... with 4 more variables: Genotype <chr>, Treatment <chr>, Behavior <chr>,
## #   class <chr>
```

```
# quantitative variables
miceavg_sub2 <- miceavg_sub[ , 1:10]
head(miceavg_sub2, 2)
```

```
## # A tibble: 2 x 10
##   NR2A_N pCAMKII_N pELK_N pERK_N pPKCAB_N ERK_N SOD1_N pPKCG_N ADARB1_N CaNA_N
##    <dbl>     <dbl>  <dbl>  <dbl>    <dbl> <dbl>  <dbl>   <dbl>    <dbl>  <dbl>
## 1   3.46      2.54   1.56  0.613    1.27   2.78  0.352    1.38     1.33   1.33
## 2   3.47      3.83   1.20  0.303    0.957  2.37  1.12     1.59     1.18   1.26
```

Exploring the Data Set

Based on the following glimpse of our data set, we can see that some of the variables are on different scales. It will be important to make sure that we scale our variables when running our clustering algorithm so that some variables don't disproportionately dominate the others.

```
glimpse(miceavg_sub)
```

```
## Rows: 72
## Columns: 14
## $ NR2A_N    <dbl> 3.46156, 3.46626, 3.83898, 4.35474, 4.23471, 5.49896, 5.2983~
## $ pCAMKII_N <dbl> 2.54137, 3.82561, 5.13009, 1.92237, 1.93167, 2.40630, 2.5022~
## $ pELK_N    <dbl> 1.556977, 1.203226, 1.365092, 1.422933, 1.874307, 1.815423, ~
## $ pERK_N    <dbl> 0.612801, 0.302583, 0.407521, 0.531628, 0.854832, 0.861139, ~
## $ pPKCAB_N  <dbl> 1.265044, 0.957446, 1.167634, 1.899062, 1.428326, 1.813830, ~
## $ ERK_N     <dbl> 2.77828, 2.37146, 2.21875, 2.62453, 2.95802, 3.82994, 3.4494~
## $ SOD1_N    <dbl> 0.352095, 1.123893, 0.565798, 0.338898, 0.381469, 0.368435, ~
## $ pPKCG_N   <dbl> 1.378694, 1.585696, 1.547511, 1.756119, 0.945970, 0.733212, ~
## $ ADARB1_N  <dbl> 1.329512, 1.176663, 0.999535, 1.088090, 1.011126, 1.790362, ~
## $ CaNA_N    <dbl> 1.328343, 1.257729, 1.058736, 1.674160, 1.193285, 1.624791, ~
## $ Genotype  <chr> "Ts65Dn", "Ts65Dn", "Control", "Control", "Control", "Contro~
## $ Treatment <chr> "Saline", "Memantine", "Memantine", "Memantine", "Memantine"~
## $ Behavior  <chr> "C/S", "S/C", "S/C", "C/S", "C/S", "C/S", "C/S", "C/S", "S/C~
## $ class     <chr> "t-CS-s", "t-SC-m", "c-SC-m", "c-CS-m", "c-CS-m", "c-CS-m", ~
```

We also notice that most of the final variables have moderate correlational relationships with each other, with some having very strong relationships. This suggests that PCA will be an appropriate secondary technique to visualize our clusters and reduce the dimensionality of our final data subset.

```
cor(miceavg_sub2)
```

```
##              NR2A_N  pCAMKII_N     pELK_N     pERK_N   pPKCAB_N     ERK_N
## NR2A_N     1.000000  0.1395626  0.4358315  0.3055200  0.3844351  0.757067
## pCAMKII_N  0.139563  1.0000000 -0.1605706 -0.2965942  0.0576737 -0.229919
## pELK_N     0.435831 -0.1605706  1.0000000  0.8647215  0.2275467  0.433294
## pERK_N     0.305520 -0.2965942  0.8647215  1.0000000  0.3687243  0.425776
## pPKCAB_N   0.384435  0.0576737  0.2275467  0.3687243  1.0000000  0.464335
## ERK_N      0.757067 -0.2299194  0.4332944  0.4257758  0.4643354  1.000000
## SOD1_N    -0.129692  0.2067143 -0.2240327 -0.5205629 -0.5140298 -0.268082
## pPKCG_N   -0.202095  0.3564289 -0.2376241 -0.1500576  0.5946875 -0.171389
## ADARB1_N   0.646673  0.1205852  0.0891529  0.0635559  0.4999548  0.550089
## CaNA_N     0.192505 -0.4493505  0.2529130  0.5471591  0.6174624  0.555588
##               SOD1_N    pPKCG_N   ADARB1_N    CaNA_N
## NR2A_N    -0.1296922 -0.2020954  0.6466735  0.192505
## pCAMKII_N  0.2067143  0.3564289  0.1205852 -0.449350
## pELK_N    -0.2240327 -0.2376241  0.0891529  0.252913
## pERK_N    -0.5205629 -0.1500576  0.0635559  0.547159
## pPKCAB_N  -0.5140298  0.5946875  0.4999548  0.617462
## ERK_N     -0.2680816 -0.1713892  0.5500894  0.555588
## SOD1_N     1.0000000 -0.0496458 -0.1347015 -0.587567
## pPKCG_N   -0.0496458  1.0000000  0.0450470  0.169984
## ADARB1_N  -0.1347015  0.0450470  1.0000000  0.172543
## CaNA_N    -0.5875665  0.1699835  0.1725427  1.000000
```

We notice that there are 8 different classes, which seems to be the most granular classification of our mice. Each group has about 7-10 mice so they are all of similar sizes.

```
tally(~ class, data = miceavg_sub)
```

```
## class
## c-CS-m c-CS-s c-SC-m c-SC-s t-CS-m t-CS-s t-SC-m t-SC-s
##     10      9     10      9      9      7      9      9
```

There are only 2 types of genotypes among our mice, with the control having 30 mice and the other group having 34 mice.

```
tally(~ Genotype, data = miceavg_sub)
```

```
## Genotype
## Control  Ts65Dn
##      38      34
```

Similarly, there are only 2 types of treatments among our mice.

```
tally(~ Treatment, data = miceavg_sub)
```

```
## Treatment
## Memantine    Saline
##        38        34
```

Lastly, the mice can have 2 types of behaviors, and both groups are of similar size.

```
tally(~ Behavior, data = miceavg_sub)
```

```
## Behavior
## C/S S/C
##  35  37
```

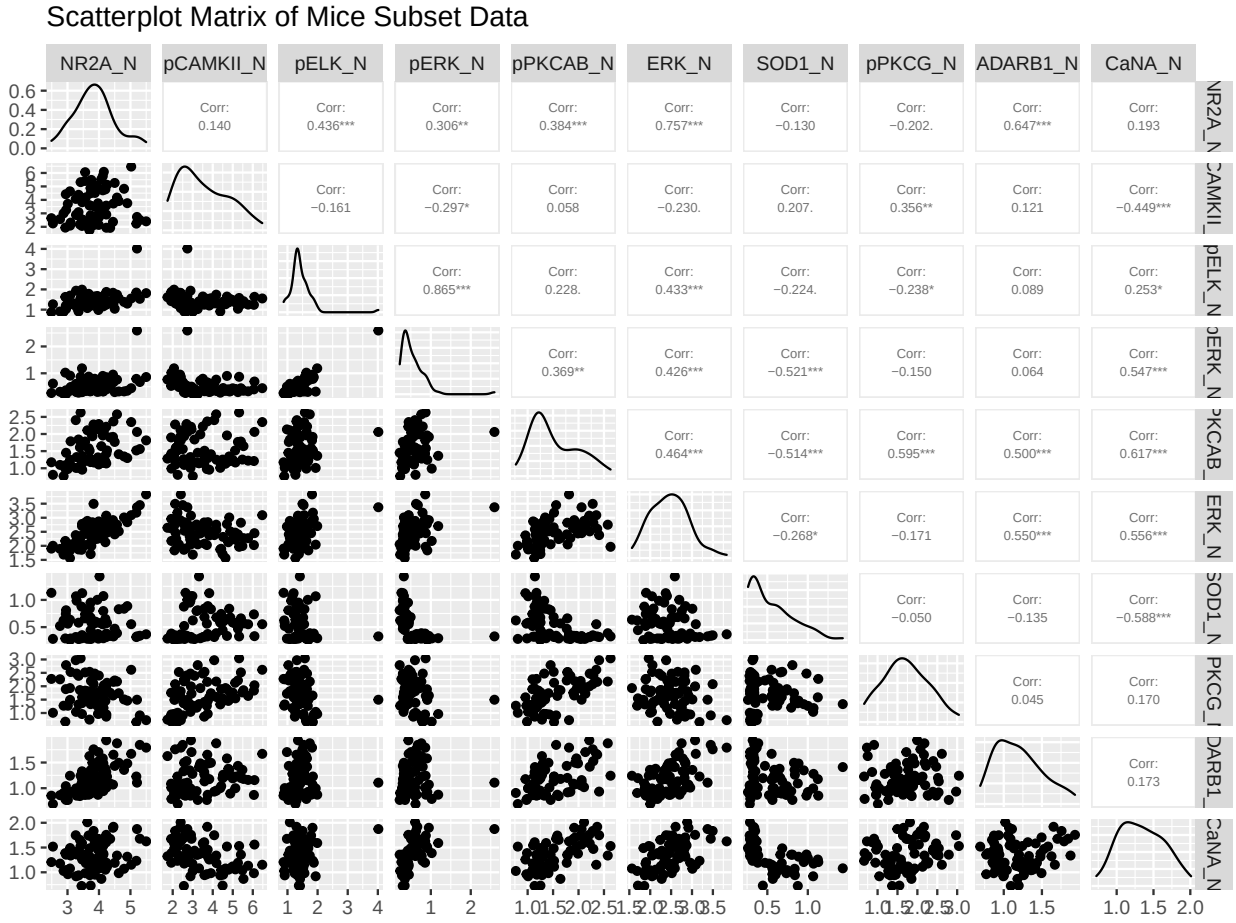
After running our clustering algorithms, we will assess whether any of these groups have been recovered.

Univariate and Bivariate Analysis

The matrix below allows us to see much more clearly that there are evident relationships between the protein expression variables in our data set, with some correlations being very strong. This is important because it suggests that we can reduce the dimensionality of this data set and PCA could prove especially useful.

Most of our variables are either normally distributed or right-skewed. This suggests influential outliers on the higher end, which we can see on some of the scatterplots. The distribution of the variables is not a major concern because we are performing an exploratory analysis. However, the outlier is of importance because some of the linkage methods used in hierarchical clustering are sensitive to outliers. We will need to take note of it as we assess our solution.

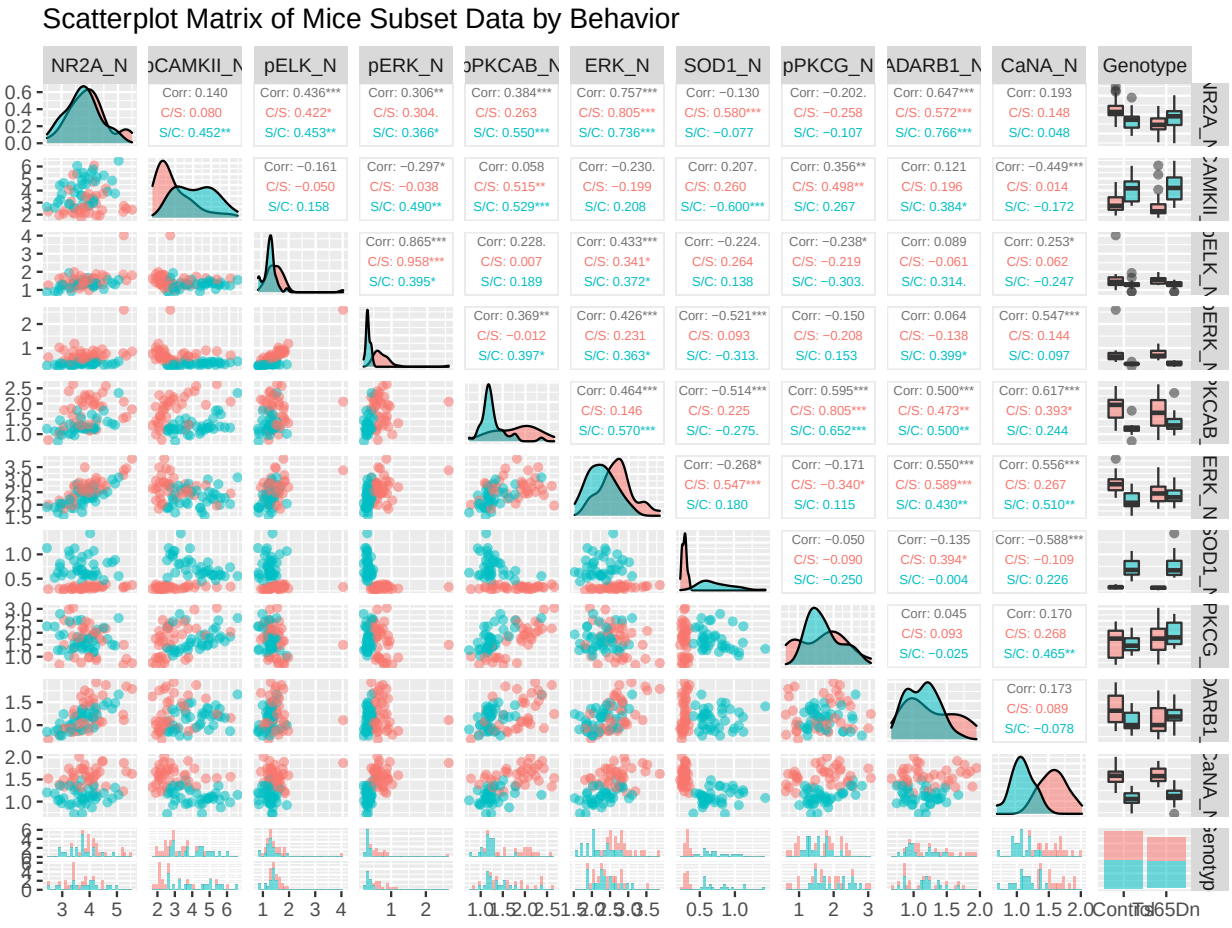
```
ggpairs(miceavg_sub2,
        upper = list(continuous = wrap("cor", size = 2)),
        title = "Scatterplot Matrix of Mice Subset Data")
```



Multivariate Analysis

Finally, in picking one of our categorical variables, `Behavior`, we can see different protein expression levels. This is a motivation for performing clustering, in an aim to see whether we can recover some of these groups through unsupervised learning.

```
ggpairs(miceavg_sub, columns = c(1:11),
        upper = list(continuous = wrap("cor", size = 2)),
        mapping = aes(color = Behavior, alpha = 0.7),
        title = "Scatterplot Matrix of Mice Subset Data by Behavior")
```


Scatterplot Matrix of Mice Subset Data by Behavior
NR2A_N
pELK_N
pERK_N
ERK_N
SOD1_N
pPKCG_N
CaNA_N
Genotype
Corr: 0.140
C/S: 0.080
S/C: 0.452**
Corr: 0.436***
C/S: 0.422*
S/C: 0.453**
Corr: 0.306**
C/S: 0.304.
S/C: 0.366*
Corr: 0.384***
C/S: 0.263
S/C: 0.550***
Corr: 0.757***
C/S: 0.805***
S/C: 0.736***
Corr: −0.130
C/S: 0.580***
S/C: −0.077
Corr: −0.202.
C/S: −0.258
S/C: −0.107
Corr: 0.647***
C/S: 0.572***
S/C: 0.766***
Corr: 0.193
C/S: 0.148
S/C: 0.048
Corr: −0.161
C/S: −0.050
S/C: 0.158
Corr: −0.297*
C/S: −0.038
S/C: 0.490**
Corr: 0.058
C/S: 0.515**
S/C: 0.529***
Corr: −0.230.
C/S: −0.199
S/C: 0.208
Corr: 0.207.
C/S: 0.260
S/C: −0.600***
Corr: 0.356**
C/S: 0.498**
S/C: 0.267
Corr: 0.121
C/S: 0.196
S/C: 0.384*
Corr: −0.449***
C/S: 0.014
S/C: −0.172
Corr: 0.865***
C/S: 0.958***
S/C: 0.395*
Corr: 0.228.
C/S: 0.007
S/C: 0.189
Corr: 0.433***
C/S: 0.341*
S/C: 0.372*
Corr: −0.224.
C/S: 0.264
S/C: 0.138
Corr: −0.238*
C/S: −0.219
S/C: −0.303.
Corr: 0.089
C/S: −0.061
S/C: 0.314.
Corr: 0.253*
C/S: 0.062
S/C: −0.247
Corr: 0.369**
C/S: −0.012
S/C: 0.397*
Corr: 0.426***
C/S: 0.231
S/C: 0.363*
Corr: −0.521***
C/S: 0.093
S/C: −0.313.
Corr: −0.150
C/S: −0.208
S/C: 0.153
Corr: 0.064
C/S: −0.138
S/C: 0.399*
Corr: 0.547***
C/S: 0.144
S/C: 0.097
Corr: 0.464***
C/S: 0.146
S/C: 0.570***
Corr: −0.514***
C/S: 0.225
S/C: −0.275.
Corr: 0.595***
C/S: 0.805***
S/C: 0.652***
Corr: 0.500***
C/S: 0.473**
S/C: 0.500**
Corr: 0.617***
C/S: 0.393*
S/C: 0.244
Corr: −0.268*
C/S: 0.547***
S/C: 0.180
Corr: −0.171
C/S: −0.340*
S/C: 0.115
Corr: 0.550***
C/S: 0.589***
S/C: 0.430**
Corr: 0.556***
C/S: 0.267
S/C: 0.510**
Corr: −0.050
C/S: −0.090
S/C: −0.250
Corr: −0.135
C/S: 0.394*
S/C: −0.004
Corr: −0.588***
C/S: −0.109
S/C: 0.226
Corr: 0.045
C/S: 0.093
S/C: −0.025
Corr: 0.170
C/S: 0.268
S/C: 0.465**
Corr: 0.173
C/S: 0.089
S/C: −0.078
Control
Ts65Dn

Methods

The Challenge

As explained in the introduction, we are being challenged to use at least 2 different clustering algorithms to find clusters of mice based on their protein expression levels. Clustering refers to any set of methods used to find natural groups of objects through unsupervised learning. For the purpose of this analysis, we will use agglomerative hierarchical clustering and K-Means clustering.

There are 77 variables coding protein expression levels in the data set, 1 identifier variable `MouseID`, and 4 categorical variables `Genotype`, `Treatment`, `Behavior`, and `class`. Our aim is to statistically assess whether we can find natural groups/ clusters of mice given information on mice protein expression levels.

The Final Variables

As detailed in the preliminary analysis section, we have managed to reduce the dimensionality of our aggregated data set (average of all 15 experiments for each mice) to 10 protein expression variables (quantitative) – `NR2A_N`, `pCAMKII_N`, `pELK_N`, `pERK_N`, `pPKCAB_N`, `ERK_N`, `SOD1_N`, `pPKCG_N`, `ADARB1_N`, and `CaNA_N` – along with the 4 categorical variables `Genotype`, `Treatment`, `Behavior`, and `class`. We chose the 10 protein variables from the 77 options available to us based on which had the most variation, which we hope will be more likely to explain differences between groups/ clusters. Note that this solution is likely to include outliers because we used variables with the most variability, and we spotted one influential one in our scatterplot matrix.

Additionally, note that we will not include the 4 categorical variables when running our clustering algorithm. We will only use the 10 protein expression variables, and later assess whether our clustering solutions was able to recover any of the 4 pre-defined groups (the categorical variables).

Because our protein expression levels have different scales, as observed in the preliminary analysis, we will scale/ standardize them in the clustering analysis.

Agglomerative Hierarchical Clustering

In agglomerative hierarchical clustering, all observations start off in a cluster alone. The pair of clusters closest to each other are then merged. The distances between each of the current clusters is updated through linkage. This process is repeated until all observations are in one cluster. The ideal number of clusters is then chosen.

Distance measure: We will use Euclidian distance as the default when running this algorithm.

Linkage: Linkage is the way in which we update our distance matrix in hierarchical clustering. Chaining can be a problem for single linkage, which finds the smallest distance between clusters and is sensitive to outliers. We will use single linkage to check for outliers in our data set. Complete linkage, which finds the smallest maximum distance between clusters, avoids chaining but tends to give clusters of similar diameter. Average linkage is a compromise between the two.

For this analysis, we will use Ward's method, which is very common. This method merges clusters that result in the smallest Within Group Sum of Squares (WGSS), a metric used to determine what to merge.

Choosing the Number of Clusters: We will observe a dendogram of the clusters and choose a cut-off based on where the joins are large.

K-Means Clustering

K-Means clustering, on the other hand, is a partitioning algorithm that uses an iterative process. We need to determine the number of clusters, k, that we wish to have. K-Means clustering then finds a set of k clusters that minimize a criterion called the WGSS (Within Group Sum of Squares). The algorithm picks k data points as starting cluster centers randomly and adjusts cluster centers and moves points between clusters until WGSS is minimized. This technique tends to give spherical clusters. Because it can get stuck with poor starting points, we will specify that the algorithm should try 10 different random starts.

Distance measure: We will use Euclidian distance as the default when running this algorithm.

Choosing the Number of Clusters: We will generate a WGSS elbow plot for K-Means. By looking at the elbow – that is, where the slope stops changing significantly –, we will be able to determine the number of clusters to set for our K-Means solution.

Assessing the Strength and Validity of the Clustering Solution

For both solutions, will use silhouette coefficients to assess the strength and validity of the clustering solution. In particular, we will look at the silhouette value per observation in each cluster, cluster average silhouette values, as well as the silhouette coefficient for the entire solution, which averages over all clusters. This will allow us to assess the structure of our solution. Values above 0.7 will indicate a strong structure, between 0.51 and 0.7 will indicate a reasonable structure, between 0.26 and 0.5 will indicate a weak, artificial structure, and below 0.25 will indicate no real structure.

We will also assess whether our solutions matched any of the 4 groups in our data set: `Genotype`, `Treatment`, `Behavior`, and `class`.

Principal Components Analysis

Lastly, we will run PCA on our data set to visualize our preferred clustering solution in the PC space using the first 2 principal components, which will explain most of the variability in the data set. In the preliminary analysis, we established that there are moderate to strong relationships between our variables, so PCA is appropriate. We will use the correlation matrix to run the PCA.

Results

Agglomerative Hierarchical Clustering

First, let's create the distance matrix that will be used by the clustering algorithm. By default, this uses Euclidian distance.

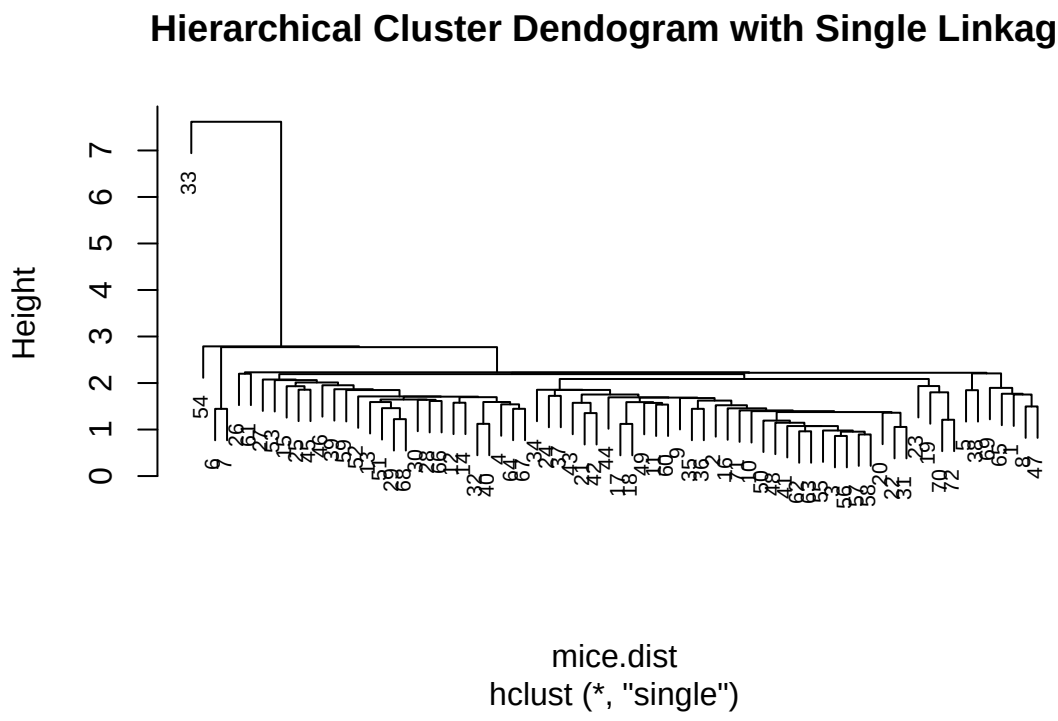
```
mice.dist <- dist(scale(select(miceavg_sub, -c(Behavior, Treatment, Genotype, class))))
```

Next, let's use single linkage to check for outliers in our solution. We notice that there is a chaining problem. There are 2 main clusters with 31 and 16 observations each. At the top left, we see one observation in isolation. I would be interested in examining that observation further. The remaining clusters have less than 4 observations each. In fact, 14 of them have a single observation. As a result, single linkage isn't the best option for this data set.

```
hcsingle <- hclust(mice.dist, method = "single")
list(hcsingle)
```

```
## [[1]]
##
## Call:
## hclust(d = mice.dist, method = "single")
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 72
```

```
plot(hcsingle, cex = 0.7, main = "Hierarchical Cluster Dendogram with Single Linkage")
```



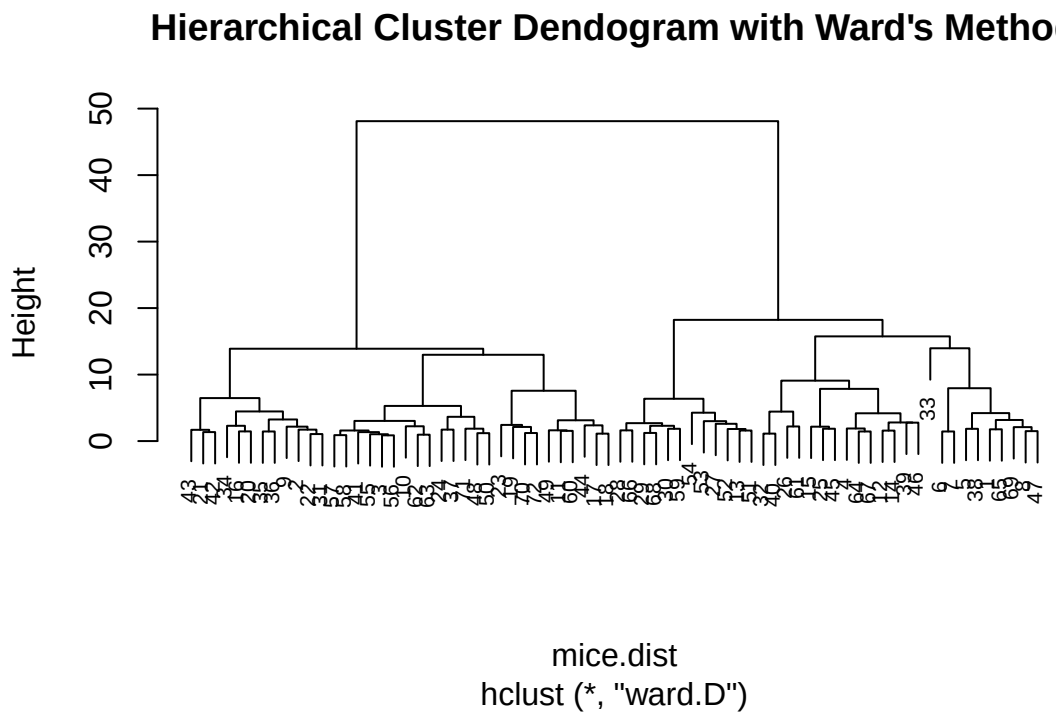
```
# example cut to show a solution
singleSol <- (cutree(hcsingle, k = 20))
summary(as.factor(singleSol))
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  4 31 16  2  2  1  3  1  1  1  1  1  1  1  1  1  1  1  1  1
```

Now, let's use Ward's method to fit our final hierarchical cluster. This solution is much better. We see that there are 2 main clusters of mice, which each split into 3 and 4 smaller clusters respectively. Note that the influential outlier, observation 33, is in its own cluster within cluster 2. In looking at the dendogram, based on where the joins are very large, I would choose $k = 2$ for this solution.

$k = 7$ is a potential solution as well, which seems like it would split the mice up into their respective classes, which had 8 groups. We haven't examined this yet.

```
hcward <- hclust(mice.dist, method = "ward.D")
plot(hcward, cex = 0.7, main = "Hierarchical Cluster Dendogram with Ward's Method")
```



Having chosen $k = 2$, let's look at the summary solution. Ward's method tends to give clusters with similar diameter. It seems this solution has split the mice into 2 even clusters of 36 mice each.

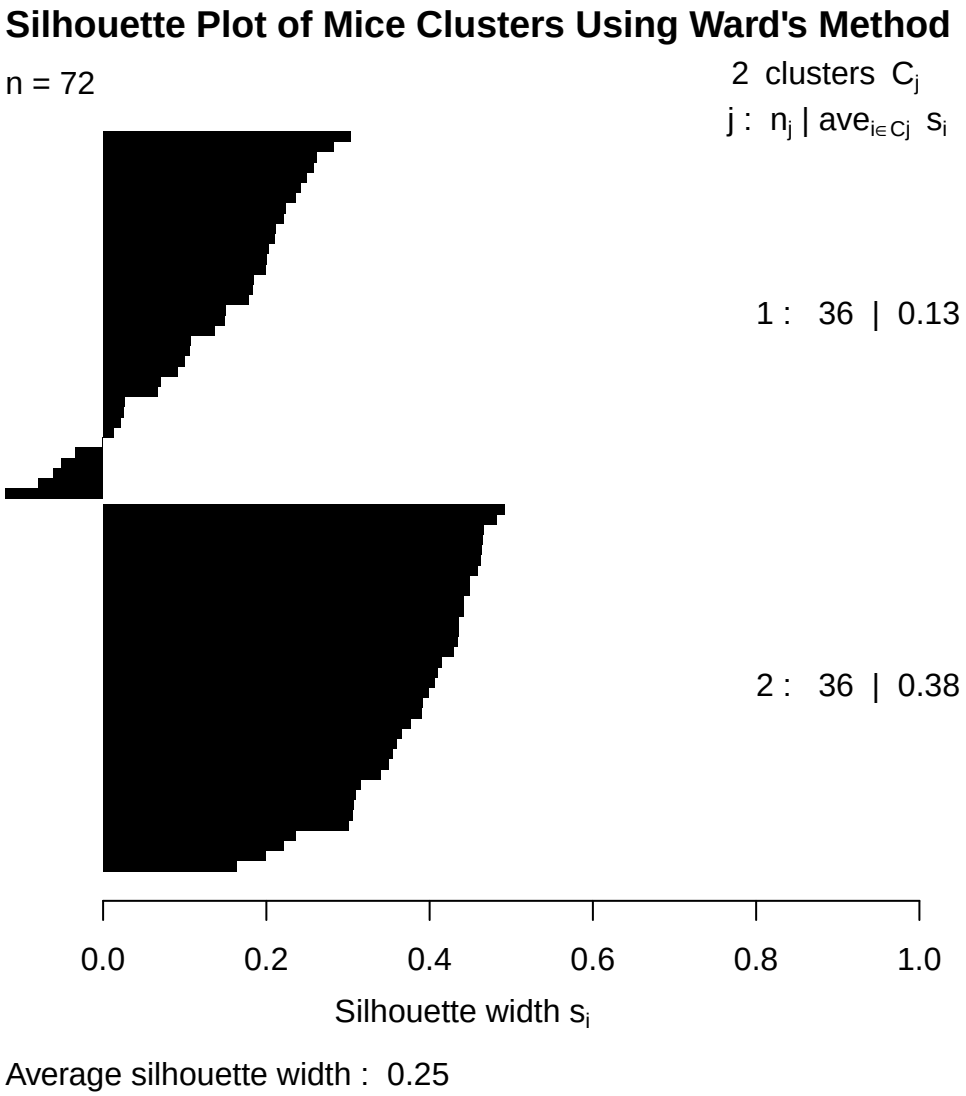
```
wardSol <- (cutree(hcward, k = 2))
summary(as.factor(wardSol))
```

```
##  1  2
## 36 36
```

Finally, let's assess the strength and validity of this solution. In looking at the summary, cluster 1 has a silhouette coefficient of 0.13 and cluster 2 has a silhouette coefficient of 0.38. Cluster 1 has no real structure, with some observations having negative values, suggesting that they do not fit into this cluster at all. Cluster 2 has a weak structure, and suggests that it may be artificial. Most observations in this cluster seem to have a weak structure as well.

```
wardSil <- silhouette(wardSol, mice.dist)

plot(wardSil, col = "black", main = "Silhouette Plot of Mice Clusters Using Ward's Method")
```



Overall, our clustering solution has a silhouette coefficient of 0.25. We conclude that this solution has no real structure. Let's look at K-Means clustering next.

K-Means Clustering

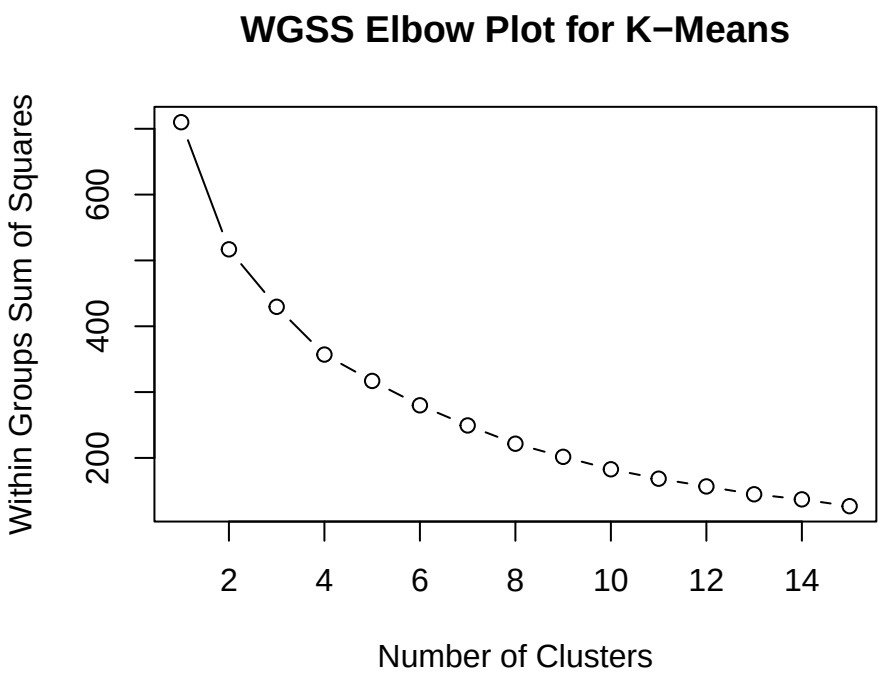
First, let's begin by determining the number of clusters, k, that we will use for the K-Means clustering algorithm by generating the WGSS elbow plot. It's difficult to see where the slope changes, but it seems the

sharpest changes are at $k = 2$ or $k = 4$. Because the joins were largest for $k = 2$ in the hierarchical solution, let's proceed with $k = 2$.

```
set.seed(240)
nclustmax <- 15

wss <- rep(0, nclustmax)
for(i in 1:nclustmax){
  wss[i] <- sum(kmeans(scale(select(miceavg_sub, -c(Behavior, Treatment, Genotype, class))),
                       centers = i, nstart = 10)$withinss)
}

plot(1:nclustmax, wss, type = "b",
     xlab = "Number of Clusters",
     ylab = "Within Groups Sum of Squares",
     main = "WGSS Elbow Plot for K-Means")
```



Now, let's run K-Means with 2 clusters. We will use 10 random starts to ensure our algorithm doesn't get stuck. For this solution, we notice that the cluster sizes are very different with cluster 1 having 30 mice and cluster 2 having 42 mice.

```
set.seed(304)

Ksol1 <- kmeans(scale(select(miceavg_sub, -c(Behavior, Treatment, Genotype, class))),
                centers = 2, nstart = 10)
list(Ksol1$size)
```

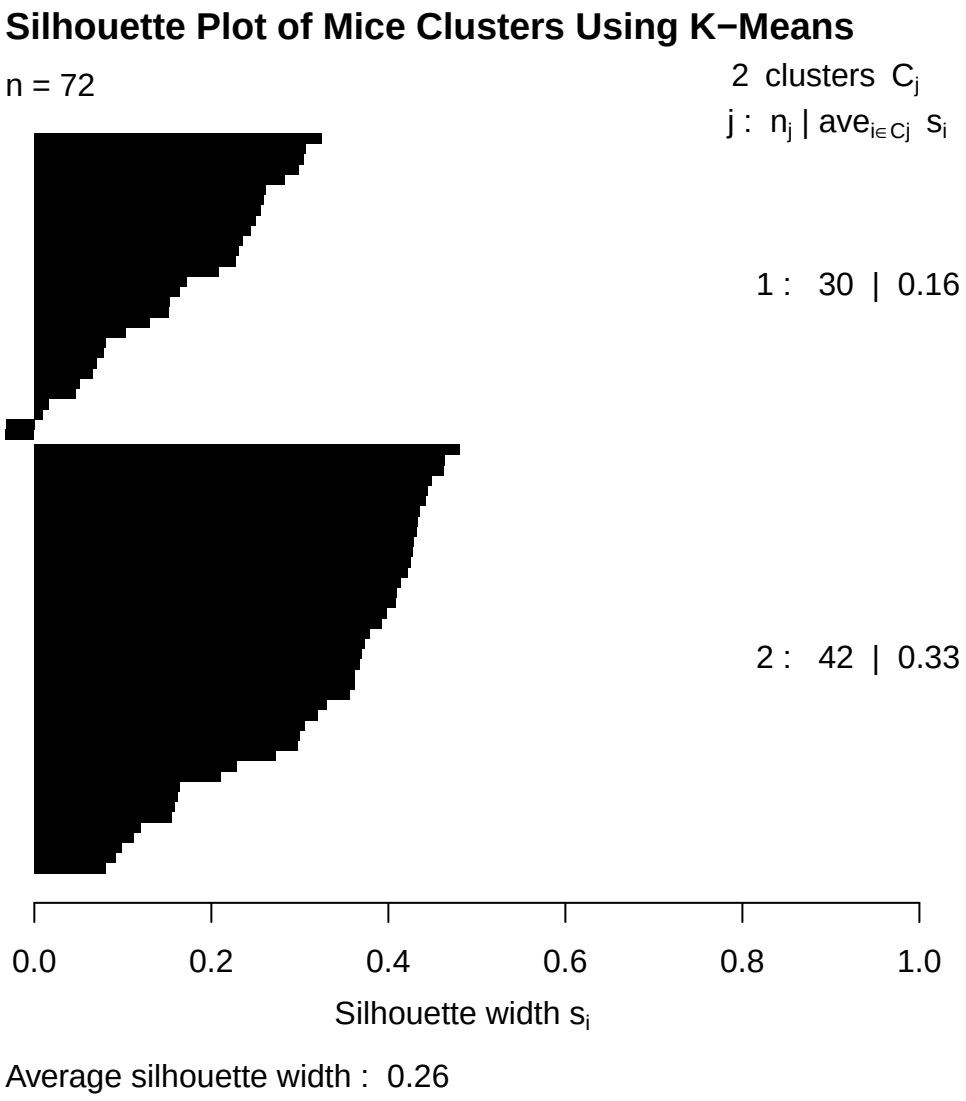
```
## [[1]]
## [1] 30 42
```

At the moment, our clustering output is on standardized variables. Therefore, we will need to save the clusters to the original data set to perform any further analysis.

Finally, let's assess the strength and validity of this solution. In looking at the summary, cluster 1 has a silhouette coefficient of 0.16 and cluster 2 has a silhouette coefficient of 0.33. Cluster 1 has no real structure, with 2 observations having negative values, suggesting that they do not fit into this cluster at all. Cluster 2 has a weak structure, and suggests that it may be artificial. Most observations in this cluster seem to have a weak structure as well.

```
kmeansSil <- silhouette(Ksol1$cluster,
                        dist(scale(select(miceavg_sub,
                                          -c(Behavior, Treatment, Genotype, class)))))

plot(kmeansSil, col = "black", main = "Silhouette Plot of Mice Clusters Using K-Means" )
```



Overall, our clustering solution has a silhouette coefficient of 0.26. This solution has a weak structure.

Preferred Solution

Our hierarchical clustering solution had no real structure and our K-Means clustering solution had a weak structure; however, the difference was quite small. Both solutions are not strong solutions.

A possible explanation could be the fact that we included the most variable variables in our data set. Perhaps the distances between our observations were too large, and so, most observations did not fit too well into their cluster. We can assess this by looking at the visualization in the next section.

Because the silhouette coefficients are very similar, we can attempt to find a stronger solution by looking at whether any of our solutions recovered any of our original groups. Because we have 2 clusters in both solutions, we will look at `Genotype`, `Treatment`, and `Behavior`, which all had only 2 groups as well. The `class` variable had 8 groups.

First, let's create a data set that encodes the clusters obtained in the hierarchical solution.

```
miceavg_sub3 <- mutate(miceavg_sub, wardSol = factor(wardSol))
```

Now, let's look at `Treatment`. It seems that both solutions are split across both treatments with no real pattern.

```
tally(miceavg_sub$Treatment ~ Ksol1$cluster)
```

```
##                       Ksol1$cluster
## miceavg_sub$Treatment  1  2
##             Memantine 15 23
##             Saline    15 19
```

```
tally(Treatment ~ wardSol, data = miceavg_sub3)
```

```
##            wardSol
## Treatment    1  2
##   Memantine 19 19
##   Saline    17 17
```

Next, let's look at `Behavior`. It seems that there is a much better split between different behaviors. The K-Means solution shows some overlap. However, the hierarchical solution perfectly matches the `Behavior` group with the exception of 1 mouse. This may be the outlier observation. The hierarchical solution seems to have recovered the `Behavior` group.

```
tally(miceavg_sub$Behavior ~ Ksol1$cluster)
```

```
##                      Ksol1$cluster
## miceavg_sub$Behavior  1  2
##                  C/S 29  6
##                  S/C  1 36
```

```
tally(Behavior ~ wardSol, data = miceavg_sub3)
```

```
##         wardSol
## Behavior  1  2
##      C/S 35  0
##      S/C  1 36
```

Lastly, `Genotype` shows a similiar even split to `Treatment`. There is no discernible pattern.

```
tally(miceavg_sub$Genotype ~ Ksol1$cluster)
```

```
##                     Ksol1$cluster
## miceavg_sub$Genotype  1  2
##              Control 17 21
##              Ts65Dn  13 21
```

```
tally(Genotype ~ wardSol, data = miceavg_sub3)
```

```
##          wardSol
## Genotype   1  2
##   Control 19 19
##   Ts65Dn  17 17
```

Based on the `Behavior` variable, we will proceed to visualize the hierarchical solution.

Principal Components Analysis

We have 10 variables in our data set. In order to visualize our clustering solution, we will turn to PCA to reduce the dimensionality of our data set.

We will use the correlation matrix. The first 2 PC's explain 58% of the variance. We will plot our solution in the PC space using PC1 and PC2.

```
micePCAs <- princomp(select(miceavg_sub, -c(Behavior, Treatment, Genotype, class)),
                     cor = TRUE)
summary(micePCAs)
```

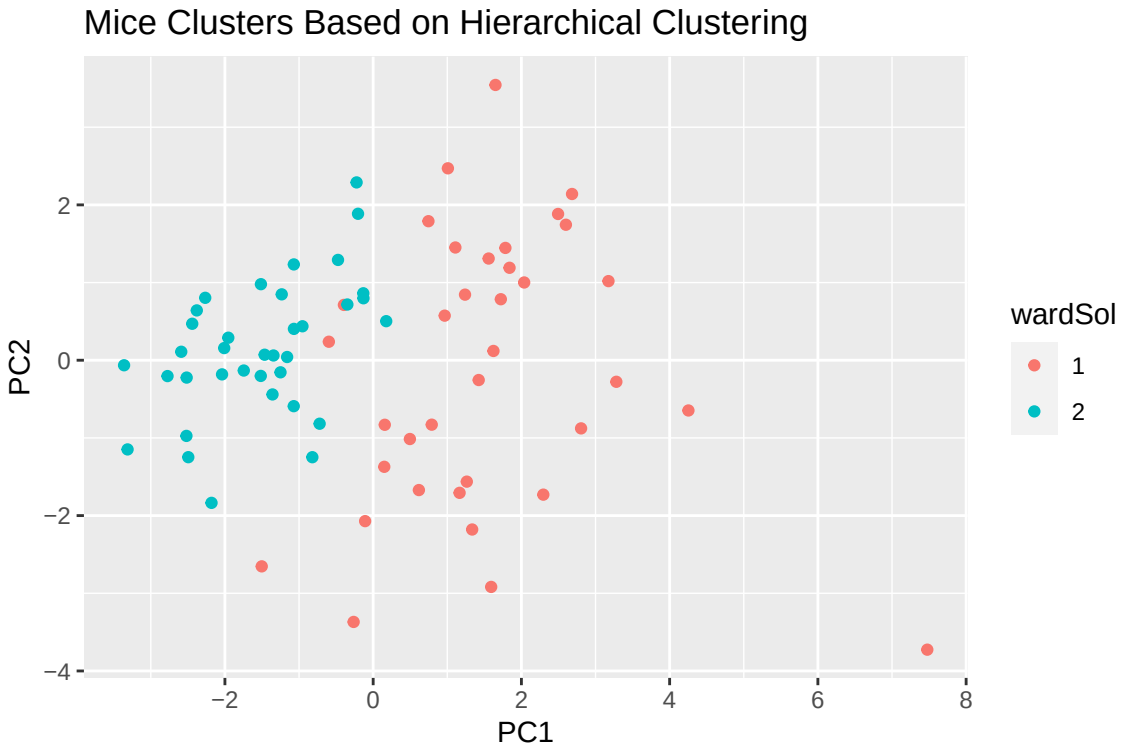
```
## Importance of components:
##                           Comp.1   Comp.2   Comp.3   Comp.4    Comp.5   Comp.6
## Standard deviation     1.976784 1.389357 1.304259 1.023196 0.7779984 0.613702
## Proportion of Variance 0.390767 0.193031 0.170109 0.104693 0.0605282 0.037663
## Cumulative Proportion  0.390767 0.583799 0.753908 0.858601 0.9191291 0.956792
##                            Comp.7    Comp.8     Comp.9    Comp.10
## Standard deviation     0.4371973 0.3515842 0.26899733 0.21205210
## Proportion of Variance 0.0191142 0.0123611 0.00723596 0.00449661
## Cumulative Proportion  0.9759063 0.9882674 0.99550339 1.00000000
```

Now, let's encode the scores in our data set that already contains the hierarchical clusters, so that we can use them to plot. We will only include the scores for PC_1 and PC_2.

```
miceavg_sub3 <- mutate(miceavg_sub3, PC1 = micePCAs$scores[, 1],
                       PC2 = micePCAs$scores[, 2],
                       kmeanssol = Ksol1$cluster)
```

Now, let's plot the clusters.

```
gf_point(PC2 ~ PC1, data = miceavg_sub3, color = ~ wardSol) %>%
  gf_labs(title = "Mice Clusters Based on Hierarchical Clustering")
```



As hypothesized above, it appears that there is a lot of variability in our data set and large distances between observations.

In conclusion, the clusters are neither strong nor robust.

Conclusion

The purpose of this analysis was to use at least 2 different clustering algorithms to find clusters of mice based on their protein expression levels. Our aim was to statistically assess whether we can find natural groups/ clusters of mice given information on mice protein expression levels, describe the clusters, assess the validity, strength, and robustness of the clustering solution, and compare it with some of the categorical variables available to us in the data set.

Because the original data set had 15 observations per mouse, we took the average across all experiments to obtain one row per mouse, while capturing information across all the experiments. Additionally, there were 77 protein expression variables. We chose the 10 most variable ones.

We then ran 2 clustering algorithms on our data set: hierarchical clustering and K-Means clustering. Both of our clustering solutions were not strong and had an artificial structure. However, the hierarchical clustering solution was able to recover the `Behavior` group in the original data set. We then used principal components analysis (PCA) to reduce the dimensionality of the data set and visualize our solution. There was a lot of variability in our data set and one influential outlier, likely because we chose the most variable variables.

Ultimately, even though our clustering solution wasn't strong, we were able to complete the challenge by running 2 different clustering algorithms on our data set and we demonstrated our understanding of clustering. If there was a future clustering data challenge regarding this data set, I would use a different method to pick a few variables and investigate how the clustering solutions would differ if the variables were picked differently.