Clustering in R

Supplemental Document to PharmaSUG 2019 ST-183

Michael Stackhouse, Alyssa Wittle

2019-04-22

Contents

1 Clustering in R

Now let's take a look at how to cluster in R. First things first: let's read in the same data we were using for clustering in Python and prepare our session.

```
# Read in the CSVs output from the Python workbook
data_2 <- as_tibble(read.csv('data_2.csv')) %>%
  mutate(shape = 1, size = 1)
data_6 <- as_tibble(read.csv('data_6.csv')) %>%
  mutate(shape = 1, size = 1)

# Set a seed for reproducibility
set.seed(0)
```

1.1 K Means

Ok - now we're ready. Let's test out K means for 2 groups.

```
# Fit and predict the clusters
km_2 <- kmeans(data_2[c("x", "y")], 2)

# Display the centers
print("Cluster centers:")
```

```
## [1] "Cluster centers:"
```

```
print(km_2$centers)
```

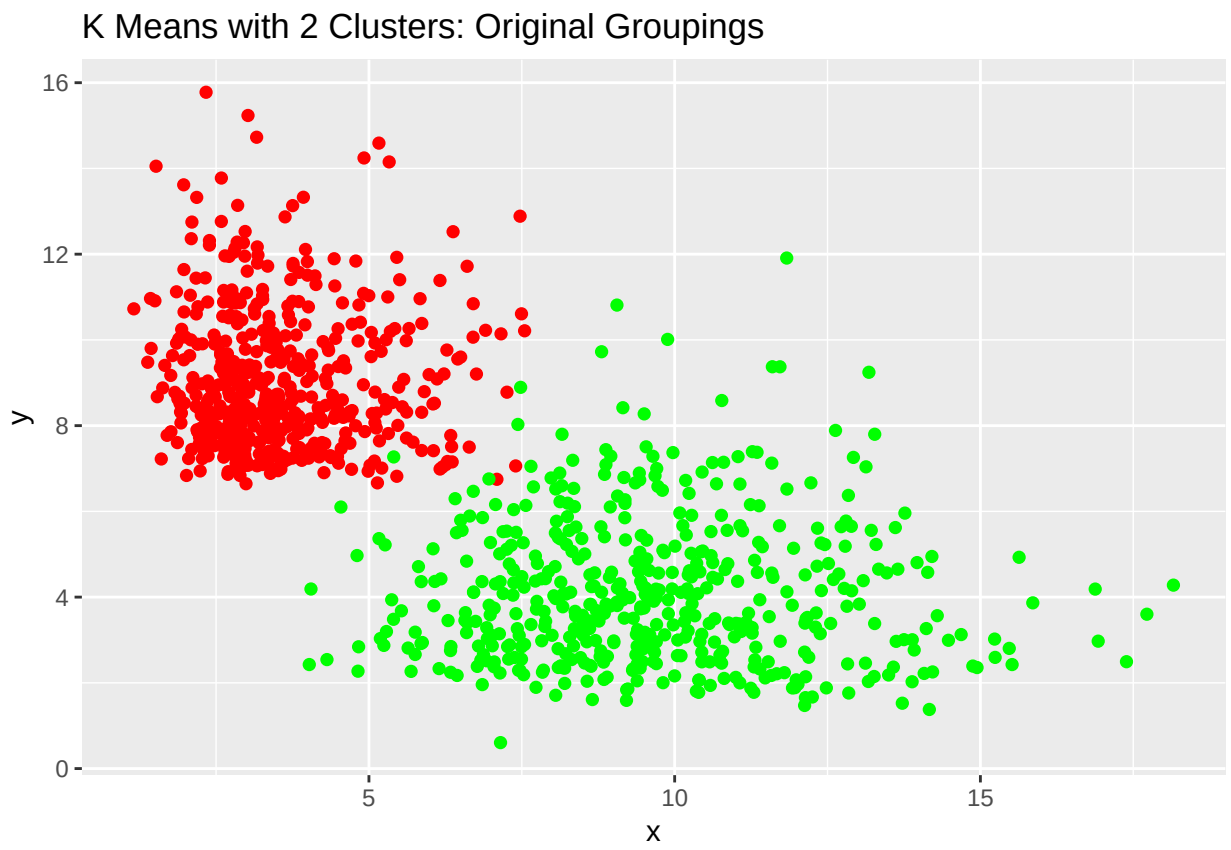
```
##          x        y
## 1 9.705509 4.065746
## 2 3.599596 9.045325
```

```
# Turn the centers into a dataset and assign a shape variable
km_2_centers <- as_tibble(km_2$centers) %>%
  mutate(pred = 99, shape = 2, size = 2)

# Make a new tibble for the predicted values
data_2p <- data_2 %>%
  mutate(pred = km_2$cluster) %>%
  bind_rows(km_2_centers)
```

Great - we have our data ready. Let's plot the original dataset and look at the groups.

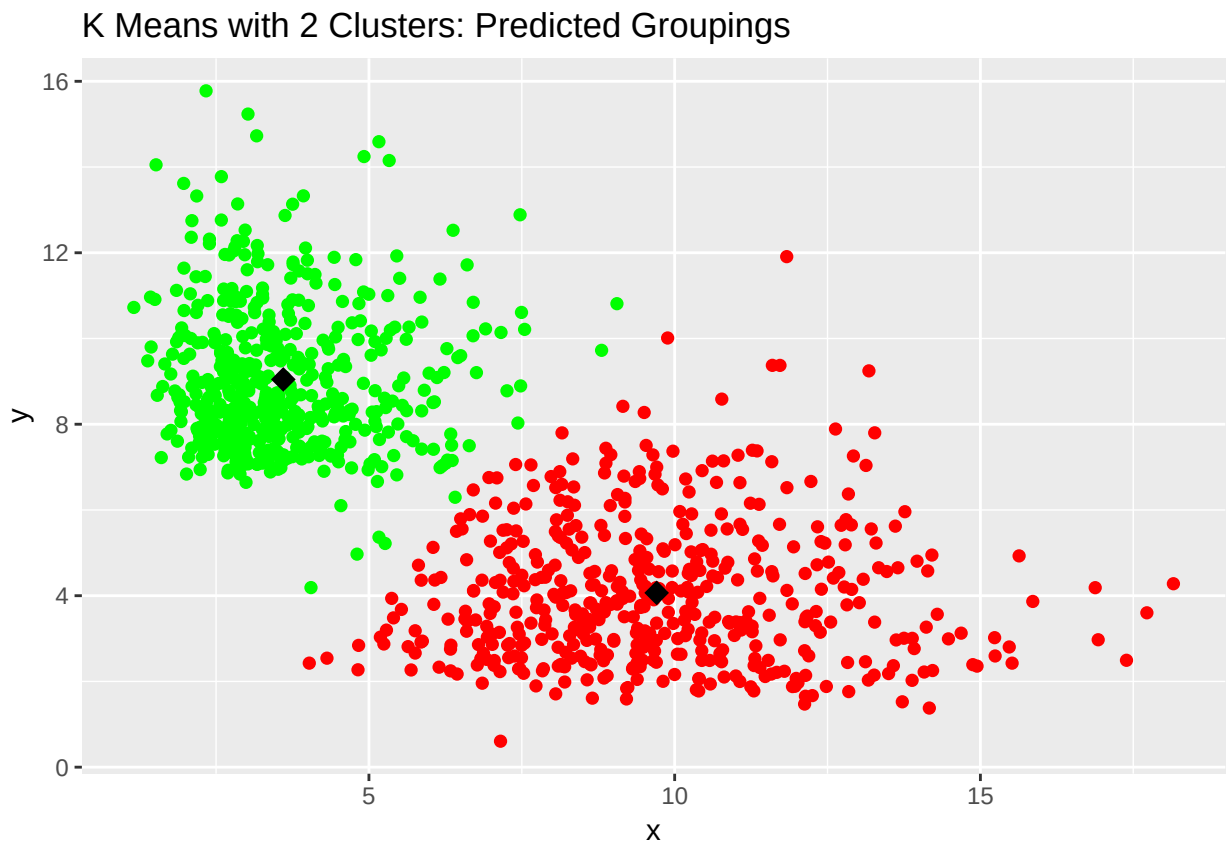
```
# Plot the original
data_2 %>%
  ggplot(aes(x, y)) +
  geom_point(aes(color = factor(z), shape = factor(shape), size = factor(size))) +
  scale_color_manual(values = c("red", "green")) +
  scale_shape_manual(values = c(16)) +
  scale_size_manual(values = c(2)) +
  theme(legend.position = "none") +
  ggtitle("K Means with 2 Clusters: Original Groupings")
```



Perfect! Looks just like before. Now, let's plot the predicted values.

```
data_2p %>%
  ggplot(aes(x, y)) +
```

```
  geom_point(aes(color = factor(pred), shape = factor(shape), size = factor(size))) +
  scale_color_manual(values = c("red", "green", "black")) +
  scale_shape_manual(values = c(16, 18)) +
  scale_size_manual(values = c(2, 4)) +
  theme(legend.position = "none") +
  ggtitle("K Means with 2 Clusters: Predicted Groupings")
```



Great! For the most part, they look the same - and we can see the center of the clusters.

Note: even though the colors are different, that doesn't matter. Clustering helps us visualize groups when we're not sure what they are, rather than classify groups that we already know exist.

Let's move on to look at the 6 group dataset now.

```
## K-Means: 6 Clusters ##
km_6 <- kmeans(data_6[c("x", "y")], 6)

# Display the centers
print("Cluster centers:")
```

```
## [1] "Cluster centers:"
```

```
print(km_6$centers)
```

```
##           x         y
```

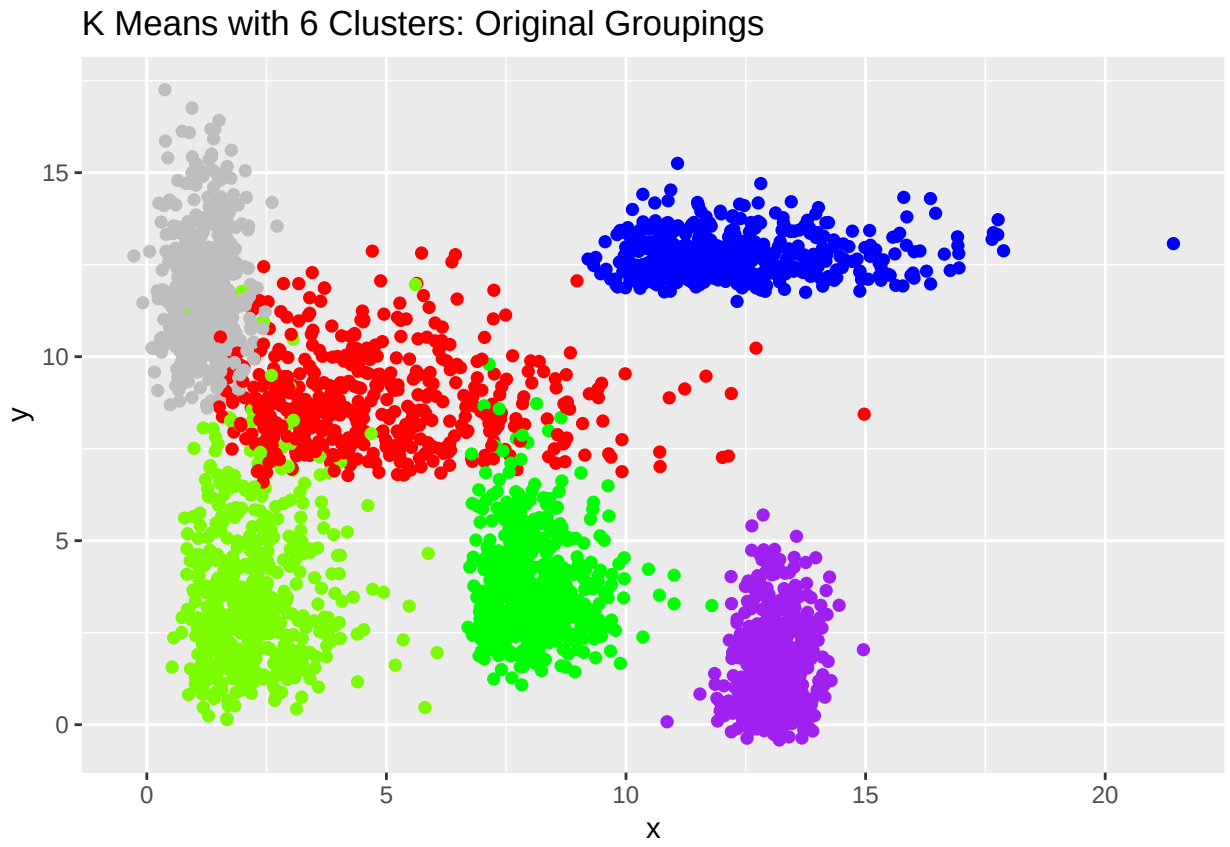
```
## 1 13.090932  1.627911
## 2  2.153296  3.041947
## 3  1.316876 11.725101
## 4  4.402932  8.452970
## 5 12.171157 12.659266
## 6  8.090763  3.810387
```

```
# Add the predicted values back to data_2
# Turn the centers into a dataset and assign a shape variable
km_6_centers <- as_tibble(km_6$centers) %>%
  mutate(pred = 99, shape = 2, size = 2)

# Make a new tibble for the predicted values
data_6p <- data_6 %>%
  mutate(pred = km_6$cluster) %>%
  bind_rows(km_6_centers)
```

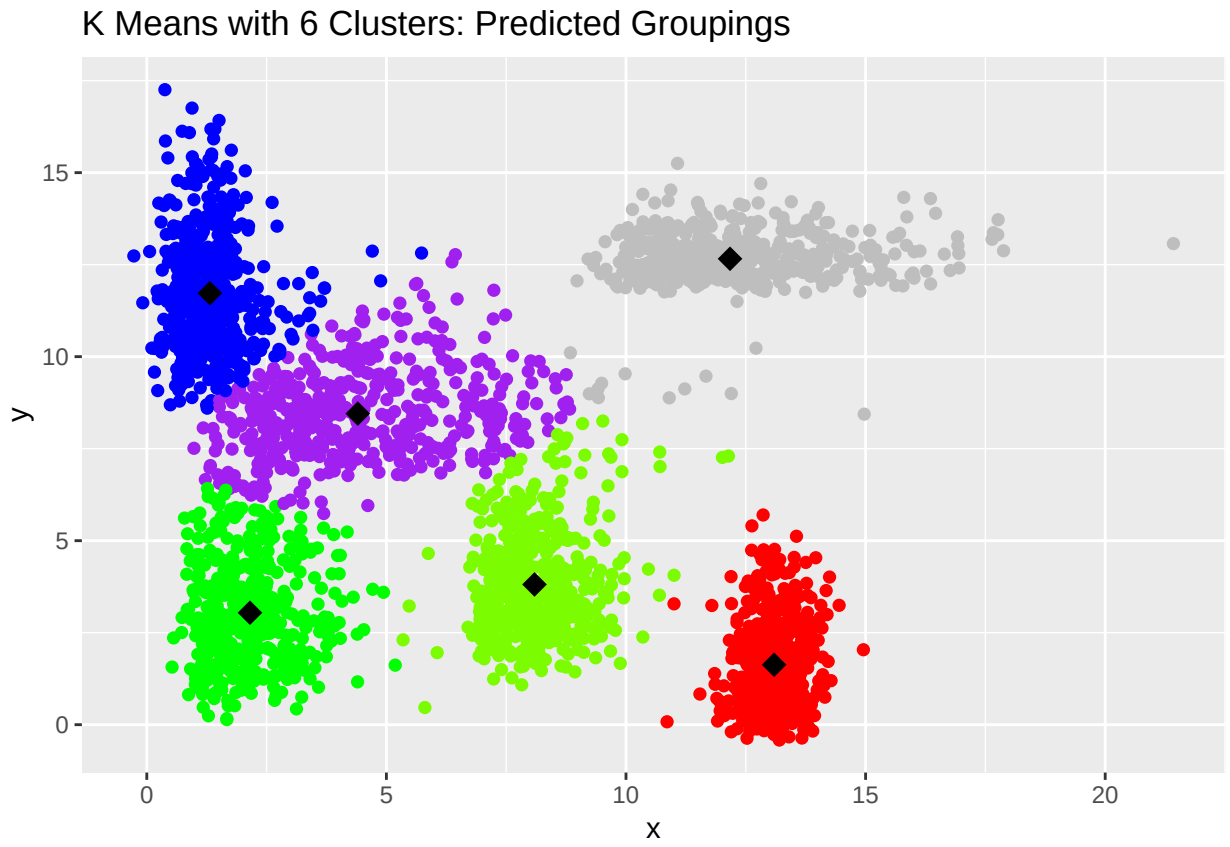
Same idea - Let's check the original data.

```
# Plot the original
data_6 %>%
  ggplot(aes(x, y)) +
  geom_point(aes(color = factor(z), shape = factor(shape), size = factor(size))) +
  scale_color_manual(values = c("red", "green", "blue", "purple", "grey", "lawngreen")) +
  scale_shape_manual(values = c(16)) +
  scale_size_manual(values = c(2)) +
  theme(legend.position = "none") +
  ggtitle("K Means with 6 Clusters: Original Groupings")
```



And now the predicted.

```
# Plot the predicted
data_6p %>%
  ggplot(aes(x, y)) +
  geom_point(aes(color = factor(pred), shape = factor(shape), size = factor(size))) +
  scale_color_manual(values = c("red", "green", "blue", "purple", "grey", "lawngreen", "black")) +
  scale_shape_manual(values = c(16, 18)) +
  scale_size_manual(values = c(2, 4)) +
  theme(legend.position = "none") +
  ggtitle("K Means with 6 Clusters: Predicted Groupings")
```



1.2 Hierarchical Clustering

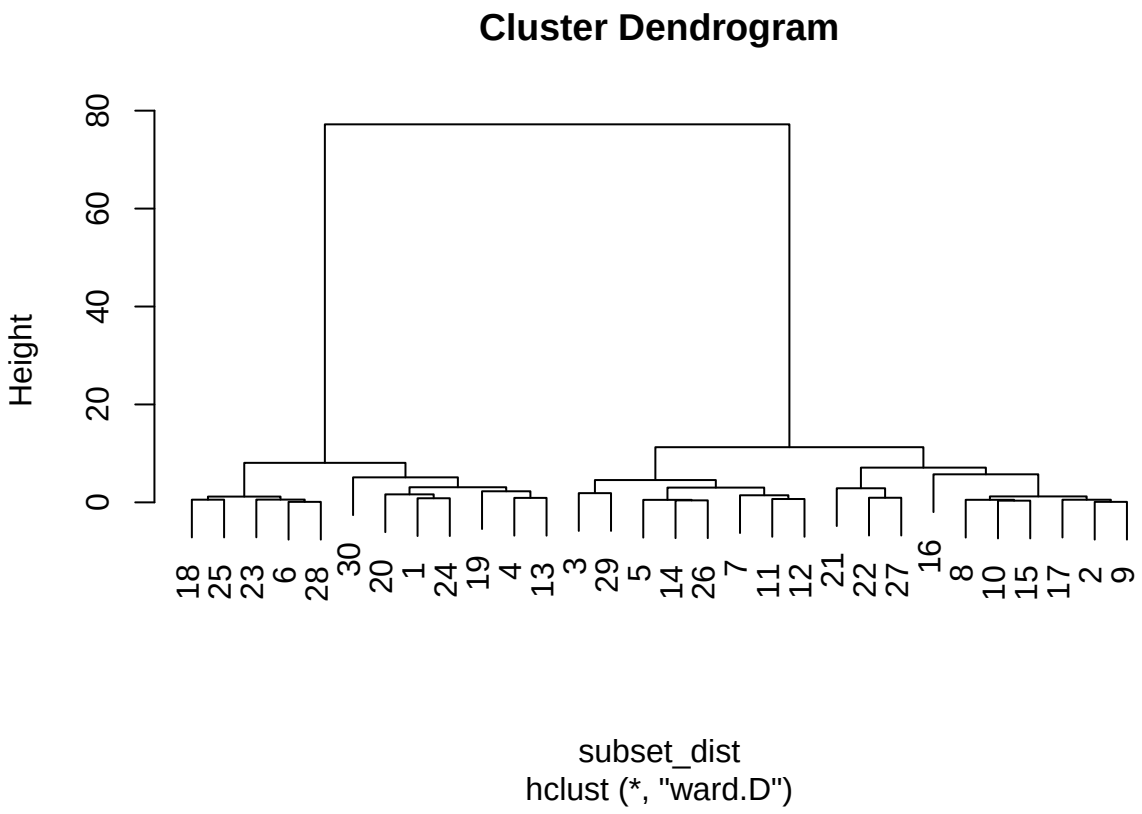
So we can see that K-means very much so works the same as it does in Python. So let's now take a look at hierarchical clustering. There are some small differences here in that we need to take a few more steps: - Measure the distances of our indepedent variables (in this case, X and Y) - Create the cluster - Cut the tree

Remember the dendogram? That's what R forms first before we get the cluster assignments that had in Python.

```
# Subset the data so the dendogram is a reasonable size
subset <- data_2[1:30, 1:2]
subset_dist <- dist(subset, method = "euclidean")

# Create the cluster object
hc <- hclust(subset_dist, method = "ward.D")

# Plot the dendogram
plot(hc)
```



A nice benefit here is how easy it is to make the dendogram. Build the cluster, and right away we can plot it with the simple command `plot()`.

Ok - so we know we want two clusters here. Let's do it.

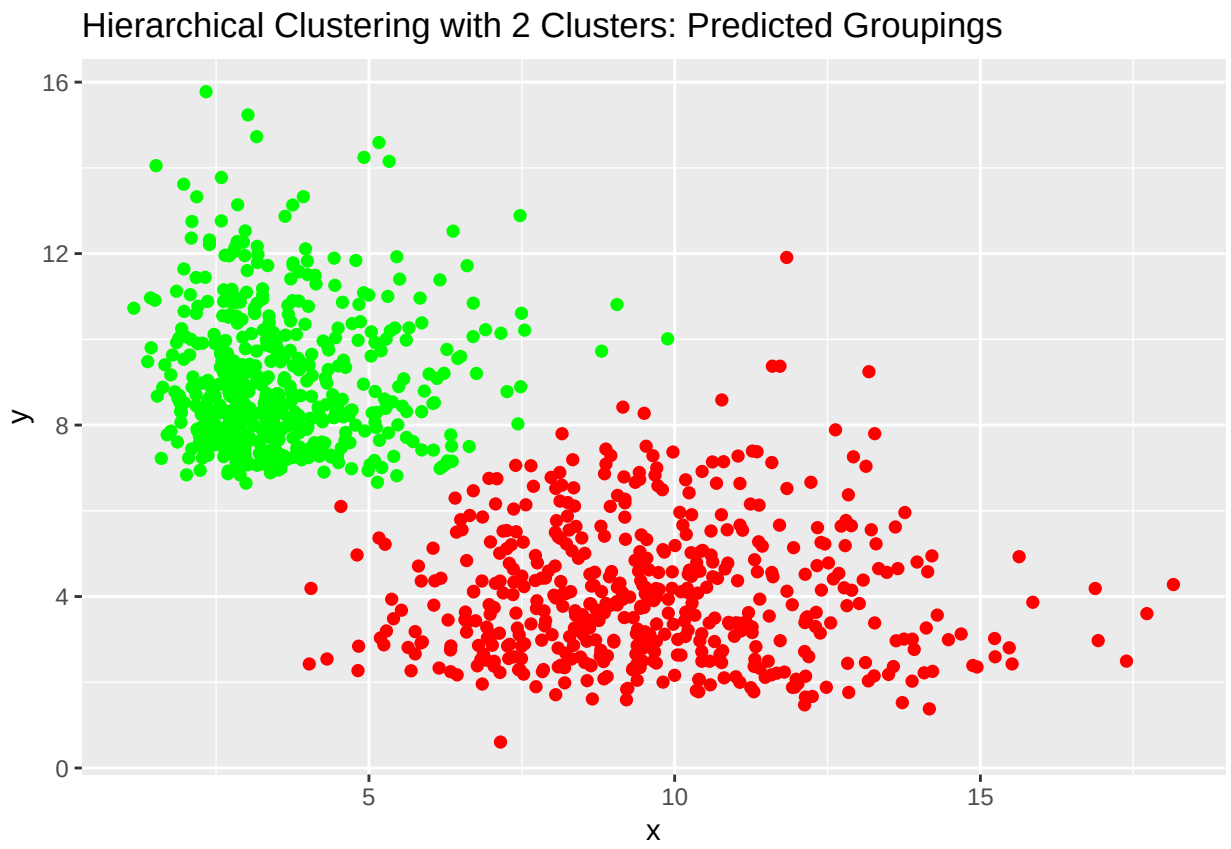
```
# Measure the distances on the 2 group dataset
data_2_dist <- dist(data_2[, 1:2], method = "euclidean")

# Cluster
hc_2 <- hclust(data_2_dist, method = "ward.D")

# Now we cut the tree
data_2_cut <- cutree(hc_2, k = 2)

# Add in the predicted groupings
data_2_hc <- data_2 %>%
  mutate(pred = data_2_cut)

# Plot
data_2_hc %>%
  ggplot(aes(x, y)) +
  geom_point(aes(color = factor(pred), shape = factor(shape), size = factor(size))) +
  scale_color_manual(values = c("red", "green")) +
  scale_shape_manual(values = c(16)) +
  scale_size_manual(values = c(2)) +
  theme(legend.position = "none") +
  ggtitle("Hierarchical Clustering with 2 Clusters: Predicted Groupings")
```



Look at that! Overall it looks like we pretty much have the groups we wanted. Now let's check out 6 groups.

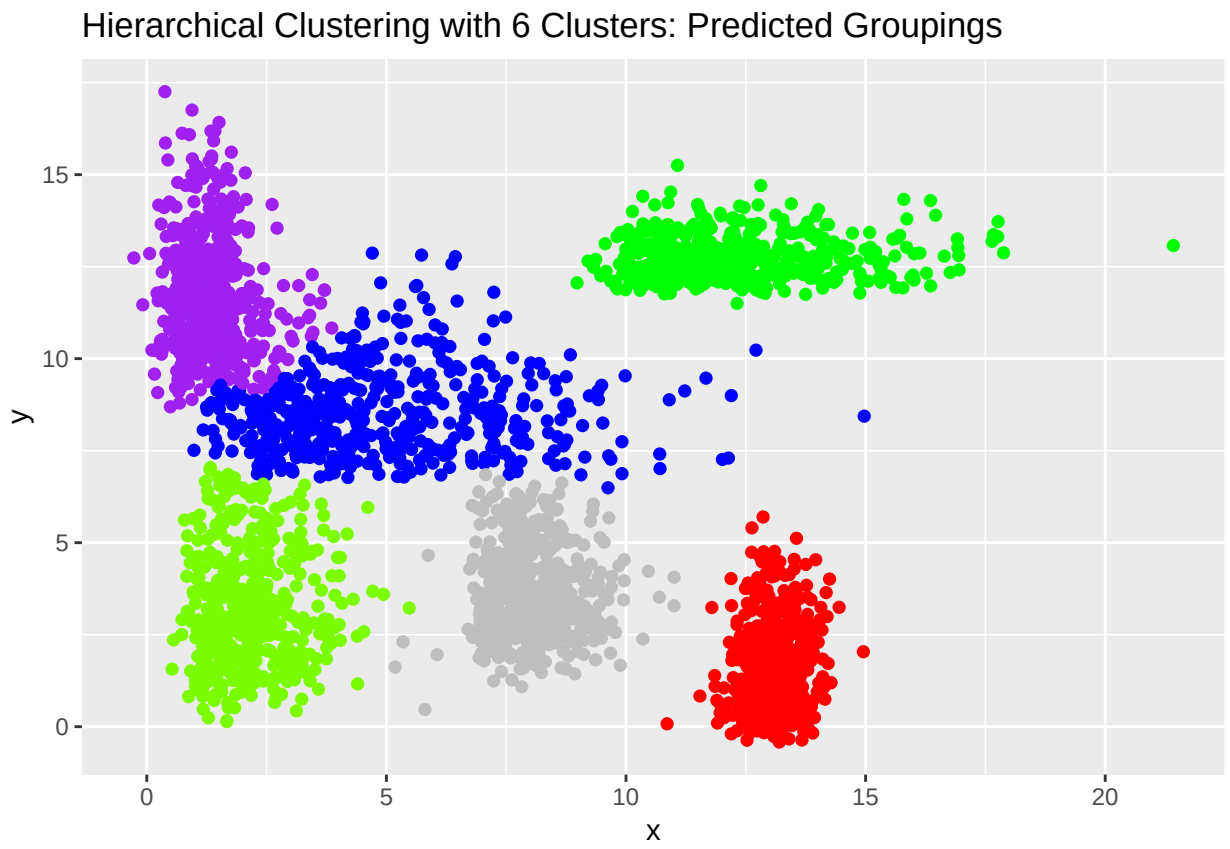
```
# Measure the distance on the 6 group dataset
data_6_dist <- dist(data_6[, 1:2], method = "euclidean")

# Cluster
hc_6 <- hclust(data_6_dist, method = "ward.D")

# Now we cut the tree
data_6_cut <- cutree(hc_6, k = 6)

# Add in the predicted groupings
data_6_hc <- data_6 %>%
  mutate(pred = data_6_cut)

# Plot
data_6_hc %>%
  ggplot(aes(x, y)) +
  geom_point(aes(color = factor(pred), shape = factor(shape), size = factor(size))) +
  scale_color_manual(values = c("red", "green", "blue", "purple", "grey", "lawngreen")) +
  scale_shape_manual(values = c(16)) +
  scale_size_manual(values = c(2)) +
  theme(legend.position = "none") +
  ggtitle("Hierarchical Clustering with 6 Clusters: Predicted Groupings")
```



And there you have it!