HW_9

Jared Brotamonte

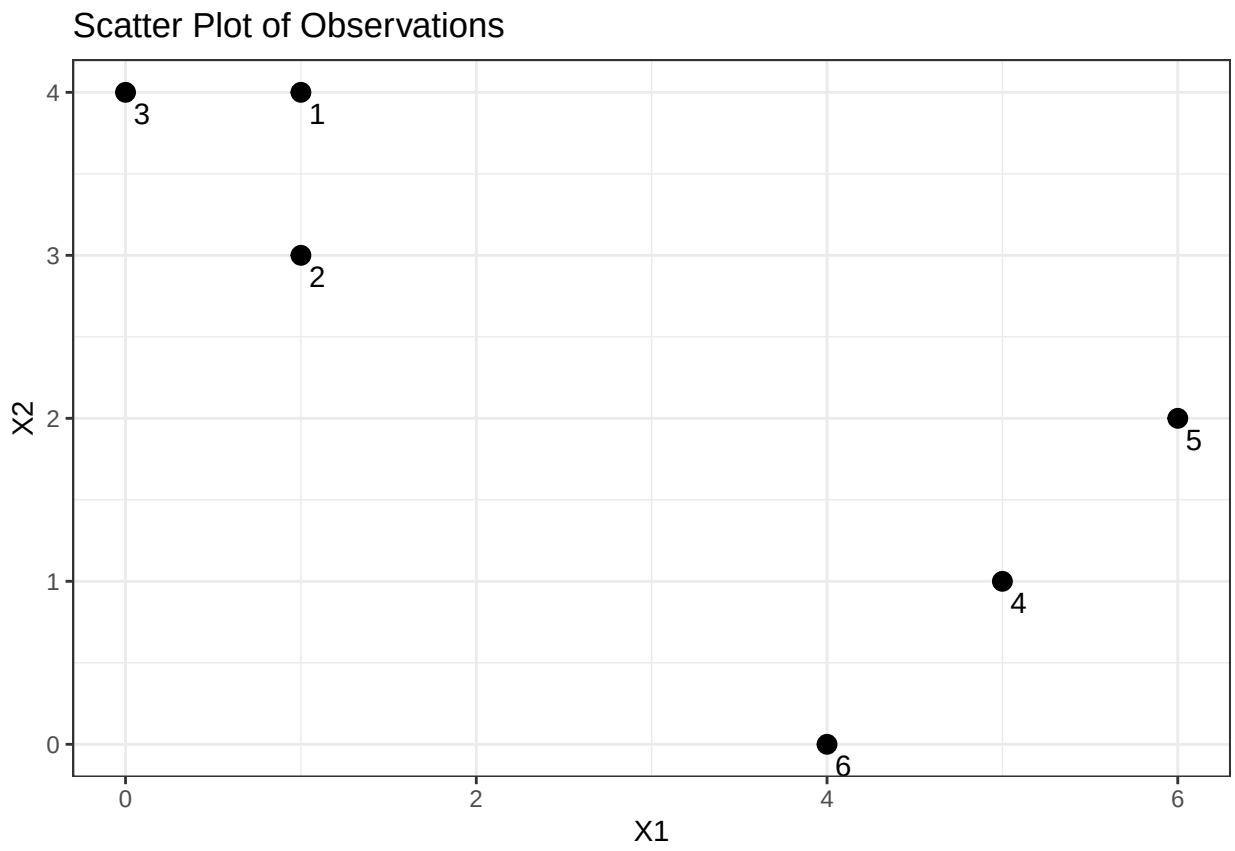
11/29/2023

Exercise 1

(a)

```
data <- data.frame(
  Obs = 1:6,
  X1 = c(1, 1, 0, 5, 6, 4),
  X2 = c(4, 3, 4, 1, 2, 0)
)

# Plot the observations
ggplot(data, aes(X1, X2)) +
  geom_point(size = 3) +
  geom_text(aes(label = Obs), vjust = 1.5, hjust = -0.5) +  # Add labels for each observation
  labs(title = "Scatter Plot of Observations", x = "X1", y = "X2") +
  theme_bw()
```



(b)

```
# Set seed for reproducibility
set.seed(123)

# Number of clusters (K)
K <- 2

# Randomly assign cluster labels to each observation
data$cluster <- sample(1:K, nrow(data), replace = TRUE)
data$cluster = as.factor(data$cluster)
head(data)
```

```
##   Obs X1 X2 cluster
## 1   1  1  4       1
## 2   2  1  3       1
## 3   3  0  4       1
## 4   4  5  1       2
## 5   5  6  2       1
## 6   6  4  0       2
```

(c)

```
centroids <- data %>%
    group_by(cluster) %>%
    summarize(mean.X1 = mean(X1), mean.X2 = mean(X2))
centroids
```

```
## # A tibble: 2 x 3
##   cluster mean.X1 mean.X2
##   <fct>     <dbl>   <dbl>
## 1 1           2      3.25
## 2 2           4.5    0.5
```

(d)

```
# check which centroid each observation is closest to and assign it to that cluster
for(i in 1:nrow(data)){
  dist_1 <- sqrt((data$X1[i]-centroids$mean.X1[1])^2 + (data$X2[i]-centroids$mean.X2[1])^2)
  dist_2 <- sqrt((data$X1[i]-centroids$mean.X1[2])^2 + (data$X2[i]-centroids$mean.X2[2])^2)
  if(dist_1==min(c(dist_1,dist_2))){
    print("assign to cluster 1")
    data$cluster[i] <- 1
  } else{
    print("assign to cluster 2")
    data$cluster[i] <- 2
  }
}
```

```
## [1] "assign to cluster 1"
## [1] "assign to cluster 1"
## [1] "assign to cluster 1"
## [1] "assign to cluster 2"
## [1] "assign to cluster 2"
## [1] "assign to cluster 2"
```

(e)

```
# Initialize a variable to track changes
changes <- TRUE

# Initialize previous_assigned_cluster
previous_assigned_cluster <- data$assigned_cluster

# Repeat (c) and (d) until the answers stop changing
while (changes) {
  # Step (c) - Compute the centroid for each cluster
  centroids <- data %>%
    group_by(cluster) %>%
```

```
    summarize(mean.X1 = mean(X1), mean.X2 = mean(X2))

  # Step (d) - Assign each observation to the centroid to which it is closest
  for(i in 1:nrow(data)){
    dist_1 <- sqrt((data$X1[i]-centroids$mean.X1[1])^2 + (data$X2[i]-centroids$mean.X2[1])^2)
    dist_2 <- sqrt((data$X1[i]-centroids$mean.X1[2])^2 + (data$X2[i]-centroids$mean.X2[2])^2)

    if(dist_1 == min(c(dist_1, dist_2))){
      print("assign to cluster 1")
      data$assigned_cluster[i] <- 1
    } else {
      print("assign to cluster 2")
      data$assigned_cluster[i] <- 2
    }
  }

  # Check if the assigned clusters have changed
  changes <- any(previous_assigned_cluster != data$assigned_cluster)

  # Update previous_assigned_cluster for the next iteration
  previous_assigned_cluster <- data$assigned_cluster
}
```

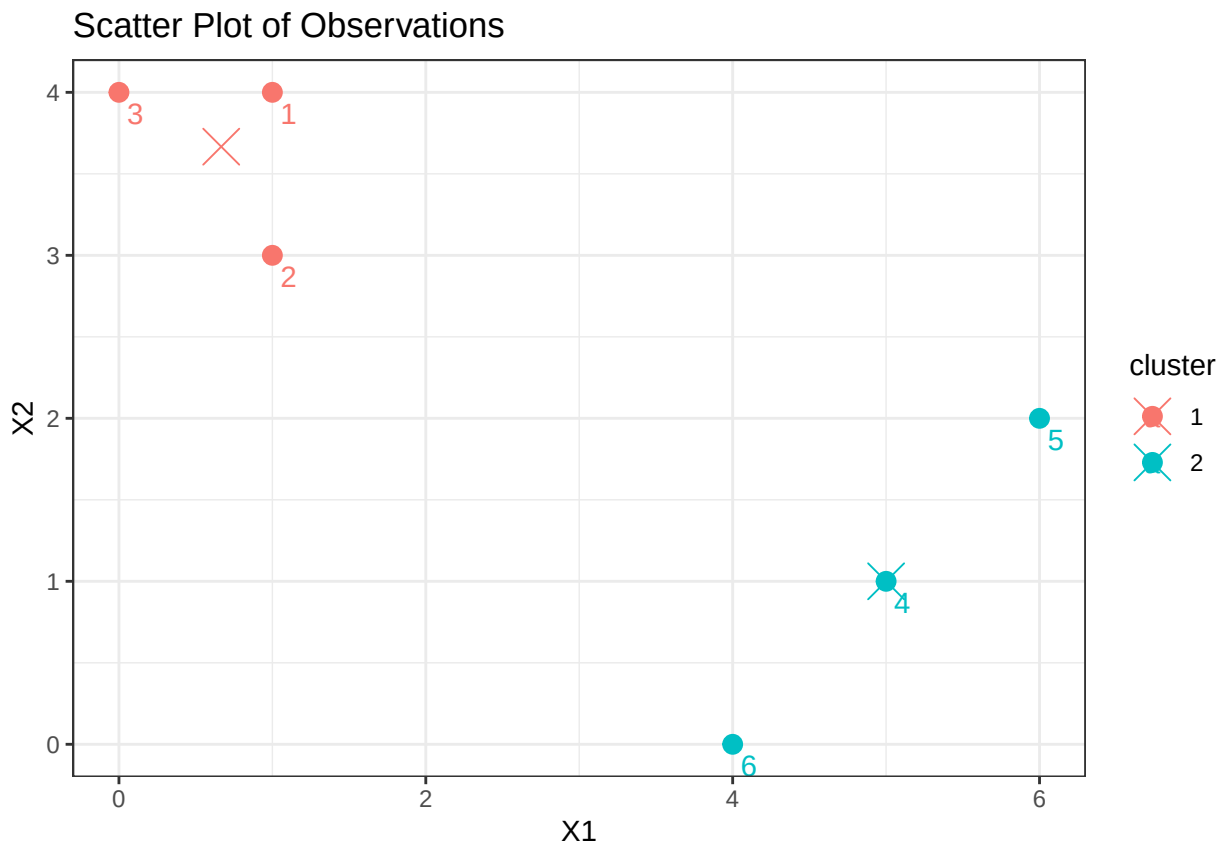
```
## [1] "assign to cluster 1"
## [1] "assign to cluster 1"
## [1] "assign to cluster 1"
## [1] "assign to cluster 2"
## [1] "assign to cluster 2"
## [1] "assign to cluster 2"
```

```
# Display the final assigned clusters
print(data$assigned_cluster)
```

```
## [1] 1 1 1 2 2 2
```

(f)

```
# Plot the observations
ggplot(data, aes(X1, X2, color=cluster)) +
  geom_point(size = 3) +
  geom_text(aes(label = Obs), vjust = 1.5, hjust = -0.5) +  # Add labels for each observation
  labs(title = "Scatter Plot of Observations", x = "X1", y = "X2") +
  theme_bw()+
  geom_point(data=centroids,
             mapping=aes(mean.X1, mean.X2),
             size=6, shape=4)
```



Exercise 2

(a)

```
file_path <- "C:/Users/jkbro/OneDrive/Desktop/STA 478/Homework/HW 9/Ch12Ex13.csv"

# Load the data using read.csv()
gene_data <- read.csv(file_path)

# Display the structure of the loaded data
head(gene_data)
```

```
##   X.0.9619334 X0.4418028 X.0.9750051  X1.417504 X0.8188148 X0.3162937
## 1 -0.29252570 -1.1392670   0.1958370 -1.2811210 -0.2514393  2.5119970
## 2  0.25878820 -0.9728448   0.5884858 -0.8002581 -1.8203980 -2.0589240
## 3 -1.15213200 -2.2131680  -0.8615249  0.6309253  0.9517719 -1.1657240
## 4  0.19578280  0.5933059   0.2829921  0.2471472  1.9786680 -0.8710180
## 5  0.03012394 -0.6910143  -0.4034258 -0.7298590 -0.3640986  1.1253490
## 6  0.08541773 -1.1130540  -0.6779688 -0.5629290  0.9381944  0.1188091
##   X.0.02496682  X.0.063966 X0.03149702 X.0.3503106 X.0.7227299 X.0.2819547
## 1  -0.92220620  0.05954277  -1.4096450  -0.6567122  -0.1157652   0.8259783
## 2  -0.06476437  1.59212400  -0.1731170  -0.1210874  -0.1875790  -1.5001630
```

```
## 3  -0.39155860  1.06361900  -0.3500090  -1.4890580  -0.2432189  -0.4330340
## 4  -0.98971500 -1.03225300  -1.1096540  -0.3851423   1.6509570  -1.7449090
## 5  -1.40404100 -0.80613040  -1.2379240   0.5776018  -0.2720642   2.1765620
## 6  -2.19222500  0.68507260   0.2623043  -1.2294590  -0.4883662  -0.7410539
##     X1.337515  X0.7019798  X1.007616 X.0.4653828 X0.6385951 X0.2867807
## 1  0.34644960 -0.56954860 -0.1315365   0.6902290 -0.9090382  1.3026420
## 2 -1.22873700  0.85598900  1.2498550  -0.8980815  0.8702058 -0.2252529
## 3 -0.03879128 -0.05789677 -1.3977620  -0.1561871 -2.7359820  0.7756169
## 4 -0.37888530 -0.67982610 -2.1315840  -0.2301718  0.4661243 -1.8004490
## 5  1.43640700 -1.02578100  0.2981582  -0.5559659  0.2046529 -1.1916480
## 6  0.25350370 -0.74905390  0.8542319   0.3547439  2.6516060 -0.3035108
##   X.0.2270782 X.0.2200452 X.1.242573 X.0.1085056 X.1.864262 X.0.5005122
## 1  -1.6726950 -0.52550400  0.7979700  -0.6897930  0.8995305   0.4285812
## 2   0.4502892  0.55144040  0.1462943   0.1297400  1.3042290  -1.6619080
## 3   0.6141562  2.01919400  1.0811390  -1.0766180 -0.2434181   0.5134822
## 4   0.6262904 -0.09772305 -0.2997108  -0.5295591 -2.0235670  -0.5108402
## 5   0.2350916  0.67096470  0.1307988   1.0689940  1.2309870   1.1344690
## 6  -1.6869130 -0.14245530 -1.1550410  -1.6636160  0.4012225  -0.4246822
##    X.1.325008   X1.063411 X.0.2963712 X.0.1216457 X0.08516605  X0.6241764
## 1 -0.67611410 -0.53409490  -1.7325070 -1.60344700 -1.08362000  0.03342185
## 2 -1.63037600 -0.07742528   1.3061820  0.79260020  1.55946500 -0.68851160
## 3 -0.51285780  2.55167600  -2.3143010 -1.27647000 -1.22927100  1.43439600
## 4  0.04600274  1.26803000  -0.7439868  0.22313190  0.85846280  0.27472610
## 5  0.55636800 -0.35876640   1.0798650 -0.20649050 -0.00616453  0.16425470
## 6  1.37574300 -0.75497810  -0.0913072  0.07828002  0.96998610  1.05213100
##   X.0.5095915  X.0.2167255 X.0.05550597 X.0.4844491 X.0.5215811  X1.949135
## 1   1.7007080  0.007289556   0.09906234   0.5638533  -0.2572752 -0.5817805
## 2  -0.6154720  0.009999363   0.94581000  -0.3185212  -0.1178895  0.6213662
## 3  -0.2842774  0.198945600  -0.09183320   0.3496279  -0.2989097  1.5136960
## 4  -0.6929984 -0.845707200  -0.17749680  -0.1664908   1.4831550 -1.6879460
## 5   1.1567370  0.241774500   0.08863952   0.1829540   0.9426771 -0.2096004
## 6  -1.7066790 -0.272883100  -1.76750600   0.4122611   0.7079067  1.0460010
##     X1.324335 X0.4681471     X1.0611   X1.65597
## 1 -0.16988710 -0.5423036  0.31293890 -1.2843770
## 2 -0.07076396  0.4016818 -0.01622713 -0.5265532
## 3  0.67118470  0.0108553 -1.04368900  1.6252750
## 4 -0.14142960  0.2007785 -0.67594210  2.2206110
## 5  0.53626210 -1.1852260 -0.42274760  0.6243603
## 6 -0.27577170 -0.1802863  0.33565780 -0.4892649
```

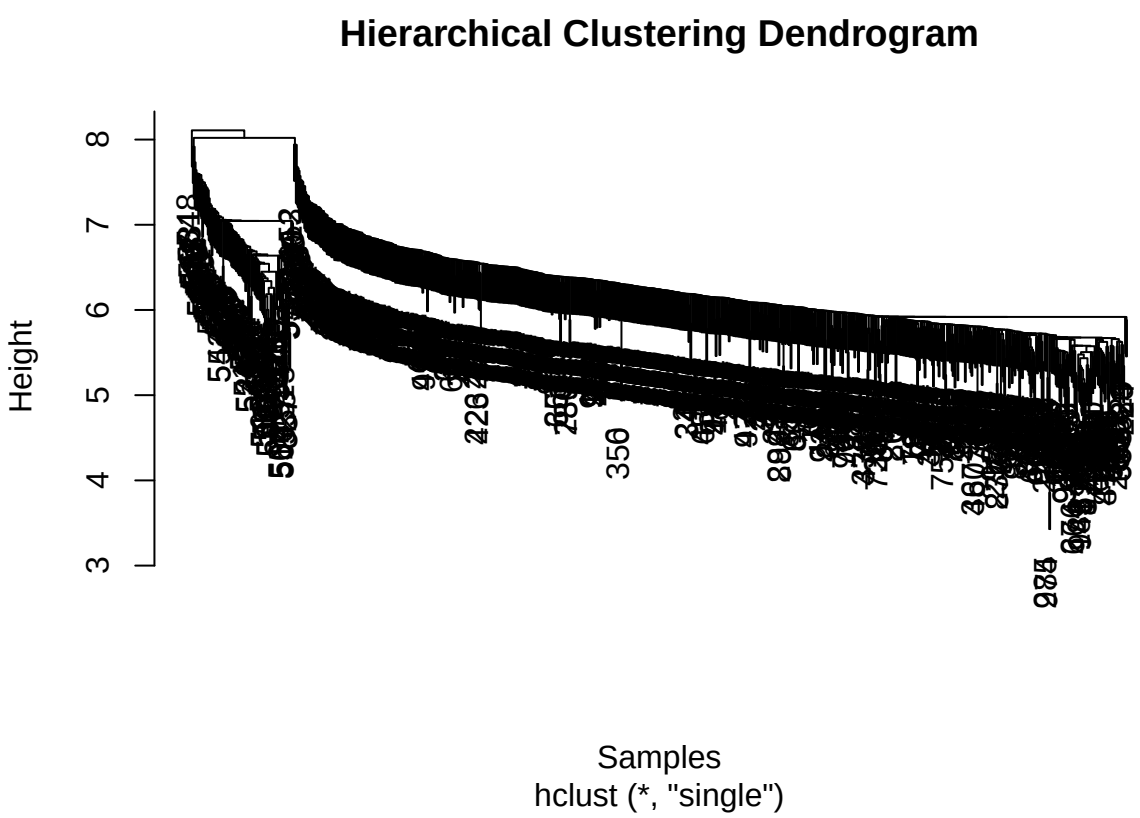
(b)

```r
# Extract gene expression data
gene_expr_data <- gene_data[, 2:ncol(gene_data)]

# Calculate Euclidean distances
distances <- dist(gene_expr_data)

# Apply hierarchical clustering using single linkage
hclust <- hclust(distances, method = "single")

plot(hclust, main = "Hierarchical Clustering Dendrogram", xlab = "Samples")
```


Hierarchical Clustering Dendrogram
Height
3
4
5
6
7
8
Samples
hclust (*, "single")