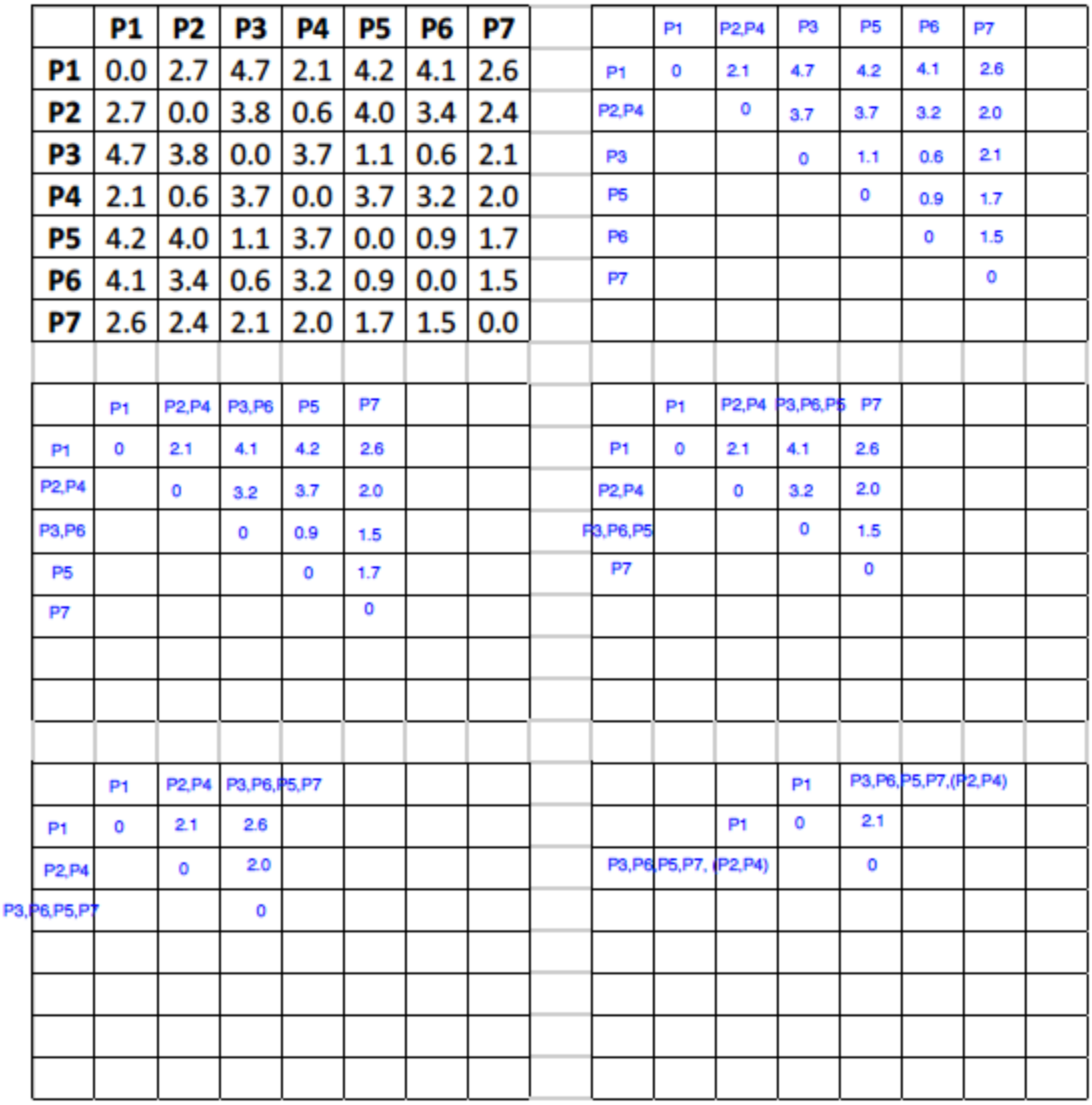
FIT3152 Data analytics. Tutorial 5: Clustering

Topics Covered:

- k-Means clustering; Hierarchical clustering

1. Work through the examples in the lecture slides.

2. Create a hierarchical cluster of the following points (P1 – P7) by applying MIN to the distance matrix below.



	P1	P2	P3	P4	P5	P6	P7
P1	0.0	2.7	4.7	2.1	4.2	4.1	2.6
P2	2.7	0.0	3.8	0.6	4.0	3.4	2.4
P3	4.7	3.8	0.0	3.7	1.1	0.6	2.1
P4	2.1	0.6	3.7	0.0	3.7	3.2	2.0
P5	4.2	4.0	1.1	3.7	0.0	0.9	1.7
P6	4.1	3.4	0.6	3.2	0.9	0.0	1.5
P7	2.6	2.4	2.1	2.0	1.7	1.5	0.0

	P1	P2,P4	P3	P5	P6	P7
P1	0	2.1	4.7	4.2	4.1	2.6
P2,P4		0	3.7	3.7	3.2	2.0
P3			0	1.1	0.6	2.1
P5				0	0.9	1.7
P6					0	1.5
P7						0

	P1	P2,P4	P3,P6	P5	P7
P1	0	2.1	4.1	4.2	2.6
P2,P4		0	3.2	3.7	2.0
P3,P6			0	0.9	1.5
P5				0	1.7
P7					0

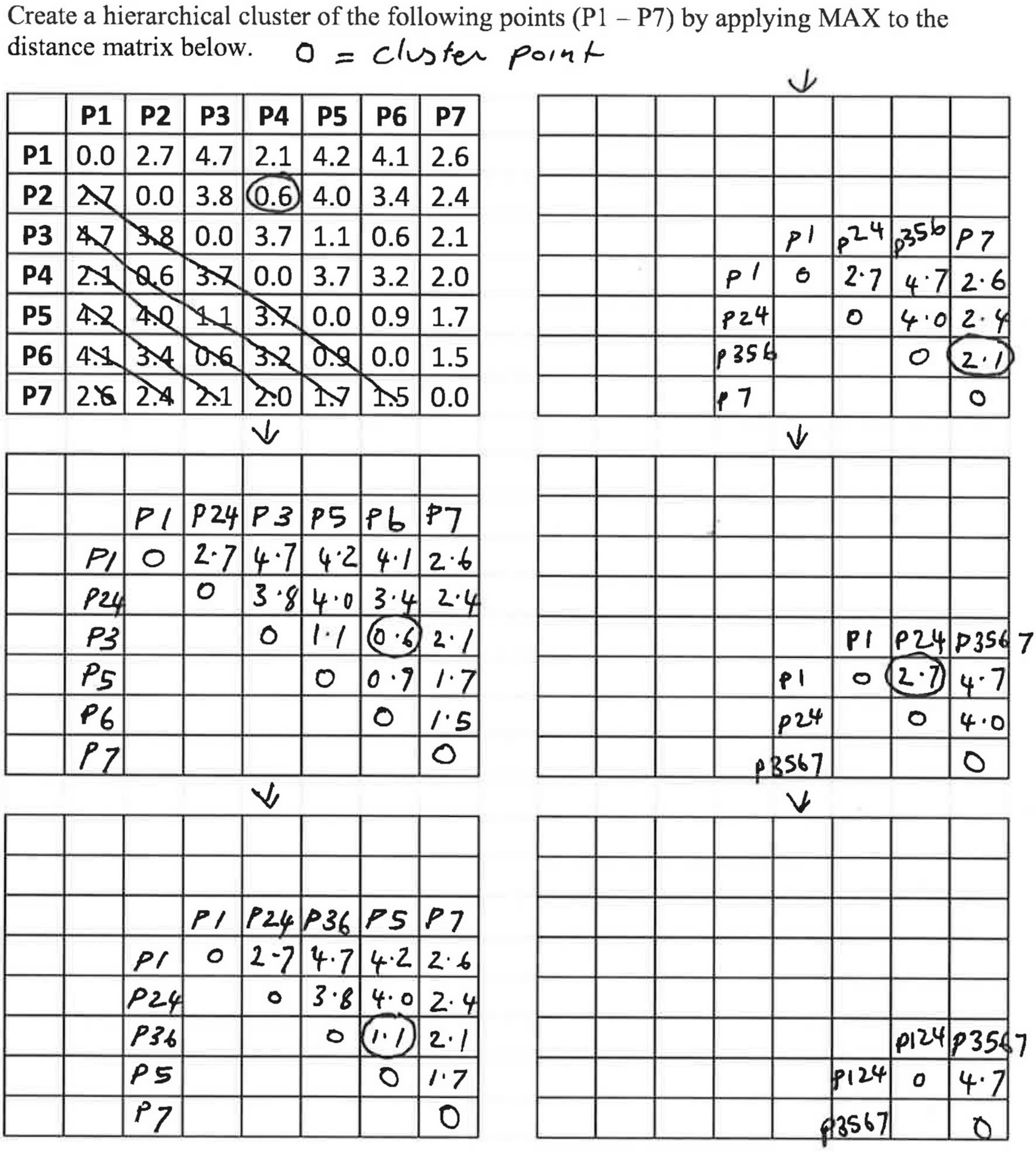
	P1	P2,P4	P3,P6,P5	P7
P1	0	2.1	4.1	2.6
P2,P4		0	3.2	2.0
P3,P6,P5			0	1.5
P7				0

	P1	P2,P4	P3,P6,P5,P7
P1	0	2.1	2.6
P2,P4		0	2.0
P3,P6,P5,P7			0

	P1	P3,P6,P5,P7,(P2,P4)
P1	0	2.1
P3,P6,P5,P7, (P2,P4)		0

3. Create a hierarchical cluster of the following points (P1 – P7) by applying MAX to the distance matrix below.

Updated solution



Create a hierarchical cluster of the following points (P1 – P7) by applying MAX to the distance matrix below. 0 = cluster point

	P1	P2	P3	P4	P5	P6	P7
P1	0.0	2.7	4.7	2.1	4.2	4.1	2.6
P2	~~2.7~~	0.0	3.8	0.6	4.0	3.4	2.4
P3	~~4.7~~	~~3.8~~	0.0	3.7	1.1	0.6	2.1
P4	~~2.1~~	~~0.6~~	~~3.7~~	0.0	3.7	3.2	2.0
P5	~~4.2~~	~~4.0~~	~~1.1~~	~~3.7~~	0.0	0.9	1.7
P6	~~4.1~~	~~3.4~~	~~0.6~~	~~3.2~~	~~0.9~~	0.0	1.5
P7	~~2.6~~	~~2.4~~	~~2.1~~	~~2.0~~	~~1.7~~	~~1.5~~	0.0

↓

	P1	P24	P3	P5	P6	P7
P1	0	2·7	4·7	4·2	4·1	2·6
P24		0	3·8	4·0	3·4	2·4
P3			0	1·1	0·6	2·1
P5				0	0·9	1·7
P6					0	1·5
P7						0

↓

	P1	P24	P36	P5	P7
P1	0	2·7	4·7	4·2	2·6
P24		0	3·8	4·0	2·4
P36			0	1·1	2·1
P5				0	1·7
P7					0

↓

	P1	P24	P356	P7
P1	0	2·7	4·7	2·6
P24		0	4·0	2·4
P356			0	2·1
P7				0

↓

	P1	P24	P3567
P1	0	2·7	4·7
P24		0	4·0
P3567			0

↓

	P124	P3567
P124	0	4·7
P3567		0

4. The Zoo data set (zoo.data.csv) contains summary data for 101 animals, as well as a classification of each animal (type) as a member of one of 7 groups.

(a) Using k-Means, cluster the data into 7 groups. Do not use "type" as an input to your clustering model. Compare the clusters you created against the actual classification of each animal type. Experiment with algorithm, scaling and normalisation to obtain the most accurate clustering compared with actual type.

(b) Repeat the activity in Part (a) using hierarchical clustering. Prune your tree to 7 clusters and compare the resulting clusters against the actual classification of each animal type. Experiment with method, scaling and normalisation to obtain the most accurate clustering compared with actual type.

(c) Which method gives the most accurate clustering of animal according to type? If clustering was used as a classifier, what would be the accuracy of your model?

(d) Calculate the optimal number of clusters using the average silhouette score. You can adapt the code for the iris data from the lecture notes.

```
# the following code (by Heshan with some additions by Anil) produces 4
      confusion matrices
# from which you can calculate the accuracy of the clustering
# note that actual label (1-7) may be different to fitted label.

rm(list=ls())
set.seed(9999)
Z=read.csv(("Zoo.data.csv"))
Z$type = factor(Z$type)
# k-means clustering
zkfit = kmeans(Z[,2:17],7,nstart = 20)
T1 = table(actual= Z$type, fitted = zkfit$cluster)
T1 = as.data.frame.matrix(T1)
T1 = T1[,c(5,2,7,4,6,3,1)]
colnames(T1) = 1:7
T1
# Accuracy = 0.7920792

# k-means clustering with scaling
ZN=Z
set.seed(9999)
ZN[,2:17]=scale(ZN[,2:17])
zkfit_N = kmeans(ZN[,2:17],7,nstart = 20)
T2 = table(actual= Z$type, fitted = zkfit_N$cluster)
T2 = as.data.frame.matrix(T2)
T2 = T2[,c(5,1,2,4,3,7,6)]
colnames(T2) = 1:7
T2
# Accuracy = 0.8316832

# hierarchical clustering
set.seed(9999)
zhfit = hclust(dist(Z[,2:17]), "ave")
plot(zhfit, hang= -1)
print(zhfit)
cut.zhfit=cutree(zhfit, k=7)
rect.hclust(zhfit, k=7, border = "red")
T3 = table(actual=Z$type, fitted = cut.zhfit)
T3 = as.data.frame.matrix(T3)
T3 = T3[,c(1,3,6,2,7,5,4)]
colnames(T3) = 1:7
T3
print(T3)
# Accuracy = 0.7524752
# hierarchical clustering with scaling
set.seed(9999)
zhfit_N = hclust(dist(ZN[,2:17]), "ave")
plot(zhfit_N, hang= -1)
print(zhfit_N)
cut.zhfit_N=cutree(zhfit, k=7)
```

```
rect.hclust(zhfit_N, k=7, border = "red")
T4 = table(actual=Z$type, fitted = cut.zhfit_N)
print(T4)
T4 = as.data.frame.matrix(T4)
T4 = T4[,c(1,3,6,2,7,5,4)]
colnames(T4) = 1:7
T4
# Accuracy = 0.7524752
print(T1)
print(T2)
print(T3)
print(T4)

# Applying the average silhouette score
# to find the optimal number of clusters

rm(list = ls())
library(cluster)

ZD <- read.csv("zoo.data.csv")
# ZD[,2:17] = scale(ZD[,2:17])

#function to get average silhouette score
# Note: try more starts to make clustering more stable
i_silhouette_score <- function(k){
  km <- kmeans(ZD[,2:17], k, nstart = 50)
  ss <- silhouette(km$cluster, dist(ZD[,2:17]))
  mean(ss[, 3])
}

#plot average silhouette score for 2-20 clusters
k <- 2:20
avg_sil <- sapply(k, i_silhouette_score)
plot(k, type='b', avg_sil, xlab='Number of clusters', ylab='Average Silhouette
      Scores')
```

5. The Cereals data set (Cereals.csv), ref: http://lib.stat.cmu.edu/DASL/Datafiles/Cereals.html, gives nutritional data for 77 different types of breakfast cereal.

(a) Using the k-Means algorithm, *and nutritional attributes only,* investigate the data and find subgroups within the data with similar nutritional attribute values.

- By inspecting the clusters can you determine whether there is a relationship between nutritional value and rating?
- Can you see any relationship between nutritional values and the nominal attributes: Manufacturer (mfr) and type?
- Experiment with the value of k and the algorithm used.

Data description for Question 5

Description: Data on several variable of different brands of cereal.
A value of -1 for nutrients indicates a missing observation.
Number of cases: 77
Variable Names:

1. Name: Name of cereal
2. mfr: Manufacturer of cereal where A = American Home Food Products; G = General Mills; K = Kelloggs; N = Nabisco; P = Post; Q = Quaker Oats; R = Ralston Purina
3. type: cold or hot
4. calories: calories per serving
5. protein: grams of protein
6. fat: grams of fat

7. sodium: milligrams of sodium
8. fiber: grams of dietary fiber
9. carbo: grams of complex carbohydrates
10. sugars: grams of sugars
11. potass: milligrams of potassium
12. vitamins: vitamins and minerals - 0, 25, or 100, indicating the typical percentage of FDA recommended
13. shelf: display shelf (1, 2, or 3, counting from the floor)
14. weight: weight in ounces of one serving
15. cups: number of cups in one serving
16. rating: a rating of the cereals

(b) Repeat the activity in Part (a) using hierarchical clustering and pruning your tree to an appropriate number of clusters. Compare your results to those obtained in Part (a).

```
# note that the two ggplots showing cluster by brand and rating. Nabisco stands
      out.
# the following code (by Heshan with some additions by Anil) produces 4
      confusion matrices will see the cereals are much harder to cluster

rm(list = ls())
library(ggplot2)
set.seed(9999)
C = read.csv("Cereals.csv")
# k-means clustering
ckfit = kmeans(C[,4:12],7,nstart = 20)
# k-means clustering with scaling
set.seed(9999)
CN=C
CN[,4:12]=scale(CN[,4:12])
ckfit_N = kmeans(CN[,4:12],7, nstart = 20)

ggplot(C,aes(mfr,type, color =factor(ckfit$cluster)))+geom_point() +
      geom_jitter(width = 0.01)
ggplot(C,aes(mfr,rating, color =factor(ckfit$cluster)))+geom_point() +
      geom_jitter(width = 0.01)

T1 = table(actual_mfr= C$mfr, fitted_clusters = ckfit$cluster)
T1 = as.data.frame.matrix(T1)
T1 = T1[,c(1,5,3,7,6,4,2)]
colnames(T1) = 1:7
print(T1)


T2 = table(actual_mfr= CN$mfr, fitted_clusters = ckfit_N$cluster)
T2 = as.data.frame.matrix(T1)
T2 = T2[,c(1,2,3,4,5,6,7)]
colnames(T2) = 1:7
print(T2)

ckfit$cluster = as.factor(ckfit$cluster)

#hierarchical clustering
set.seed(9999)
chfit = hclust(dist(C[,4:12]), "ave")
plot(chfit, hang= -1)
print(chfit)
cut.chfit=cutree(chfit, k=7)
rect.hclust(chfit, k=7, border = "red")
# hierarchical clustering with scaling
set.seed(9999)
```

```
chfit_N = hclust(dist(CN[,4:12]), "ave")
plot(chfit_N, hang= -1)
print(chfit_N)
cut.chfit_N=cutree(chfit, k=7)
rect.hclust(chfit_N, k=7, border = "red")

ggplot(C,aes(mfr,type, color =as.factor(cut.chfit_N)))+geom_point() +
      geom_jitter(width = 0.01)
ggplot(C,aes(mfr,rating, color =as.factor(cut.chfit_N)))+geom_point() +
      geom_jitter(width = 0.01)

T3 = table(actual_mfr=C$mfr, fitted_clusters = cut.chfit)
T3 = as.data.frame.matrix(T3)
T3 = T3[,c(1,4,5,2,6,7,3)]
colnames(T3) = 1:7
print(T3)

T4 = table(actual_mfr=CN$mfr, fitted_clusters = cut.chfit_N)
T4 = as.data.frame.matrix(T4)
T4 = T4[,c(1,4,5,2,6,7,3)]
colnames(T4) = 1:7
print(T4)

print(T1)
print(T2)
print(T3)
print(T4)
```