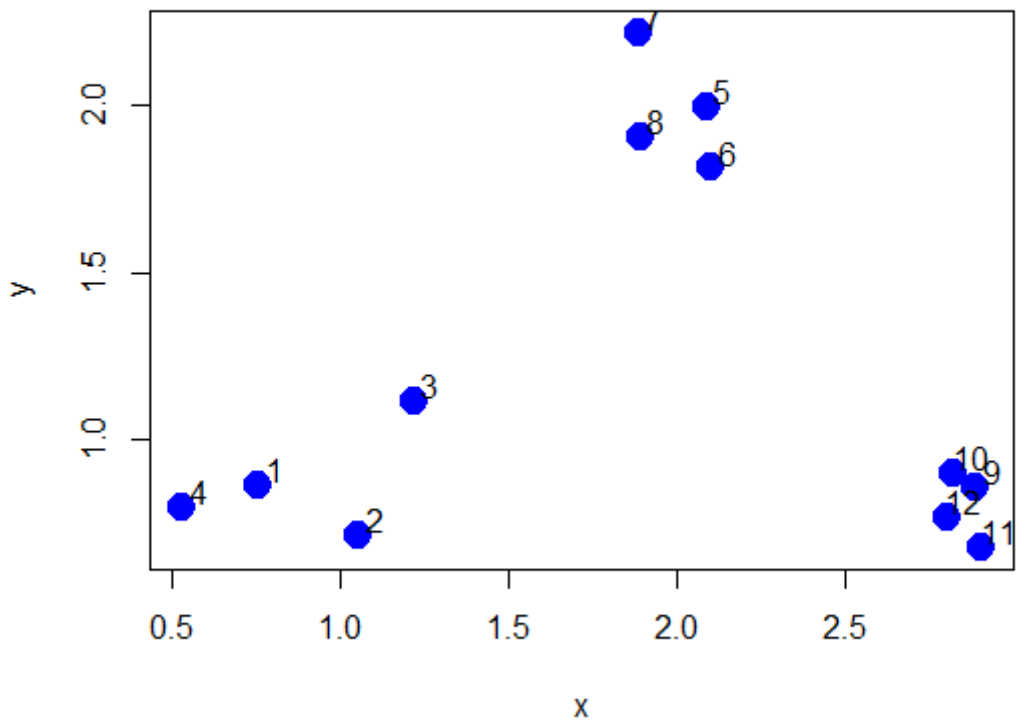
# Week 12

# Clustering

# Dataset Preparation

```
> set.seed(1234)
> x<-rnorm(12,mean=rep(1:3,each=4),sd=0.2)
> y<-rnorm(12,mean=rep(c(1,2,1),each=4),sd=0.2)
> plot(x,y,col="blue",pch=19,cex=2)
> text(x+0.05,y+0.05,labels=as.character(1:12))
> df<-data.frame(x,y)
```

# Dataset



# K-means Clustering

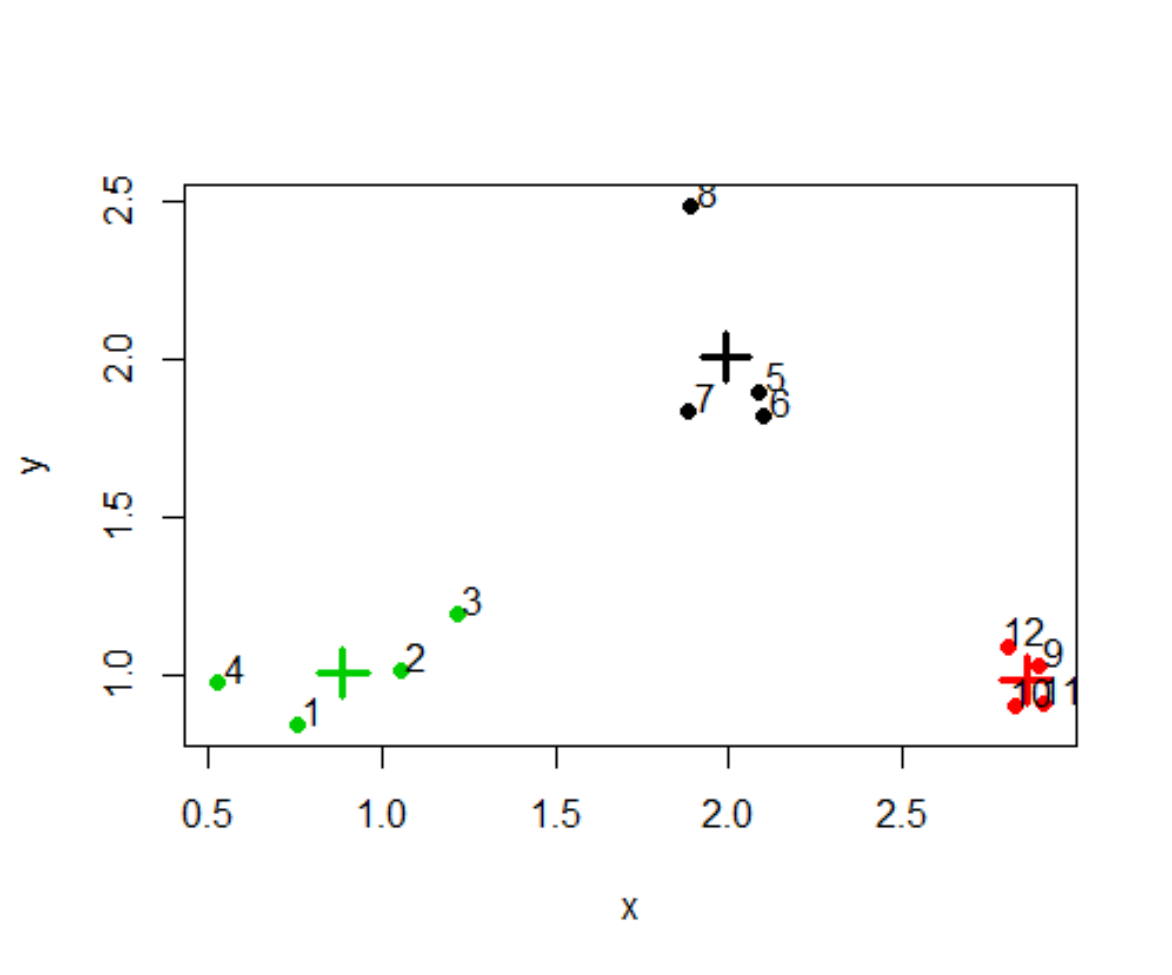
```
> k.cluster<-kmeans(df,centers=3)
> names(k.cluster)
> k.cluster$cluster
> k.cluster$centers
```

```
> k.cluster$cluster
 [1] 3 3 3 3 1 1 1 1 2 2 2 2
```

# K-means Clustering

```
> plot(x,y,col=k.cluster$cluster,
      pch=19,cex=2)
> points(k.cluster$centers,col=1:3,
      pch=3,cex=3,lwd=3)
```

# K-means Clustering



# Hierarchical Clustering

- To run Hierarchical Clustering, you need to look for the pair-wise distances between the points.

```
> df<-data.frame(x,y)
> distxy<-dist(df)
```

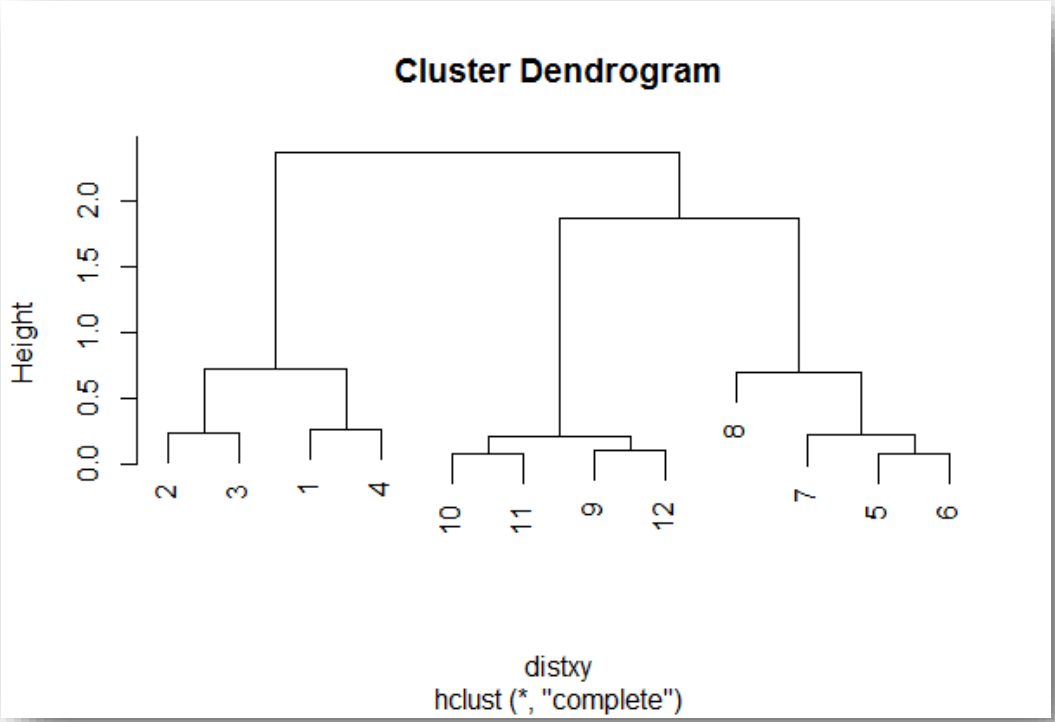
# Hierarchical Clustering

```
> dist(df)
           1          2          3          4          5          6          7          8          9         10         11
2  0.34120511
3  0.57493739 0.24102750
4  0.26381786 0.52578819 0.71861759
5  1.69424700 1.35818182 1.11952883 1.80666768
6  1.65812902 1.31960442 1.08338841 1.78081321 0.08150268
7  1.49823399 1.16620981 0.92568723 1.60131659 0.21110433 0.21666557
8  1.99149025 1.69093111 1.45648906 2.02849490 0.61704200 0.69791931 0.65062566
9  2.13629539 1.83167669 1.67835968 2.35675598 1.18349654 1.11500116 1.28582631 1.76460709
10 2.06419586 1.76999236 1.63109790 2.29239480 1.23847877 1.16550201 1.32063059 1.83517785 0.14090406
11 2.14702468 1.85183204 1.71074417 2.37461984 1.28153948 1.21077373 1.37369662 1.86999431 0.11624471 0.08317570
12 2.05664233 1.74662555 1.58658782 2.27232243 1.07700974 1.00777231 1.17740375 1.66223814 0.10848966 0.19128645 0.20802789
```

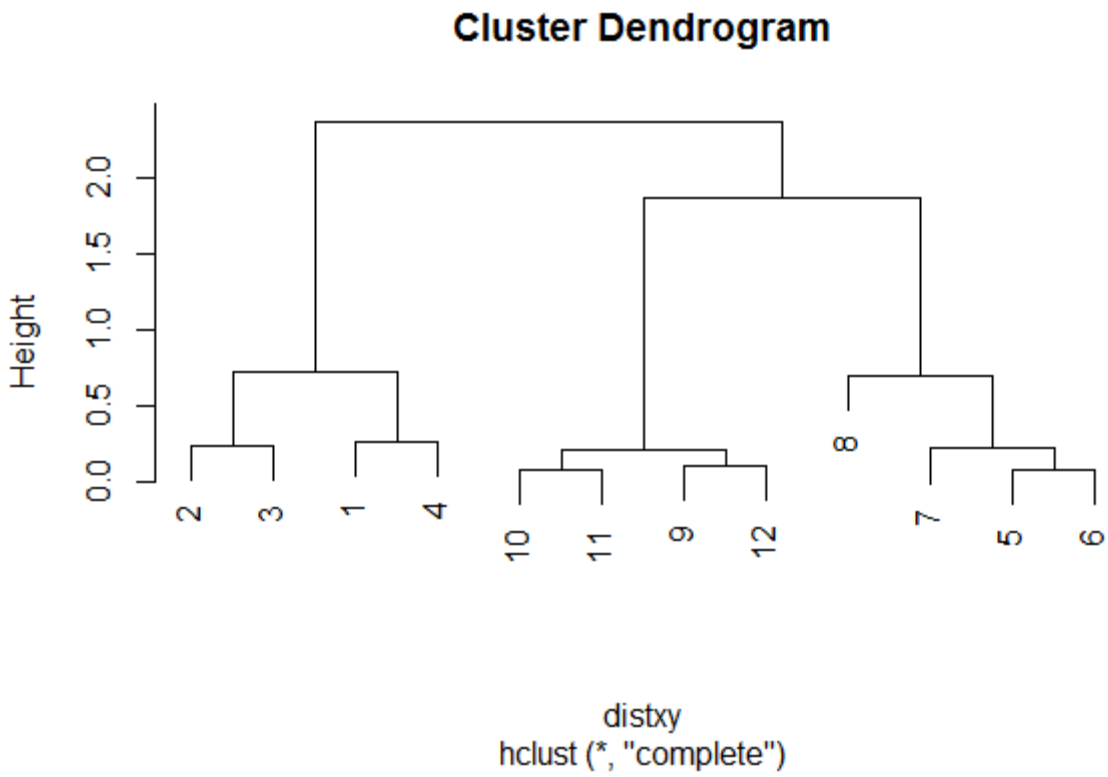
# Hierarchical Clustering

```
> clusters<-hclust(distxy)
> plot(clusters)
```

You need to determine where to "cut" the tree!



# Hierarchical Clustering



# Hierarchical Clustering

Try with different k,

```
> rect.hclust(hClusters, k=2, border="red")
> rect.hclust(hClusters, k=3, border="blue")
```

Check the plots

# Exercise – Using the Laundry Dataset

- Load the laundry dataset **`data_Lab5.csv`**
- Check for missing data.
- Remove them or perform automatic imputation of missing data.

# Distance measurement

Using Washers **W2** to **W6** only as example.

```
> dt.washers <- dt[,2:7]
```

```
> dist(dt.washers,method="euclidean")
> dist(dt.washers,method="manhattan")
```

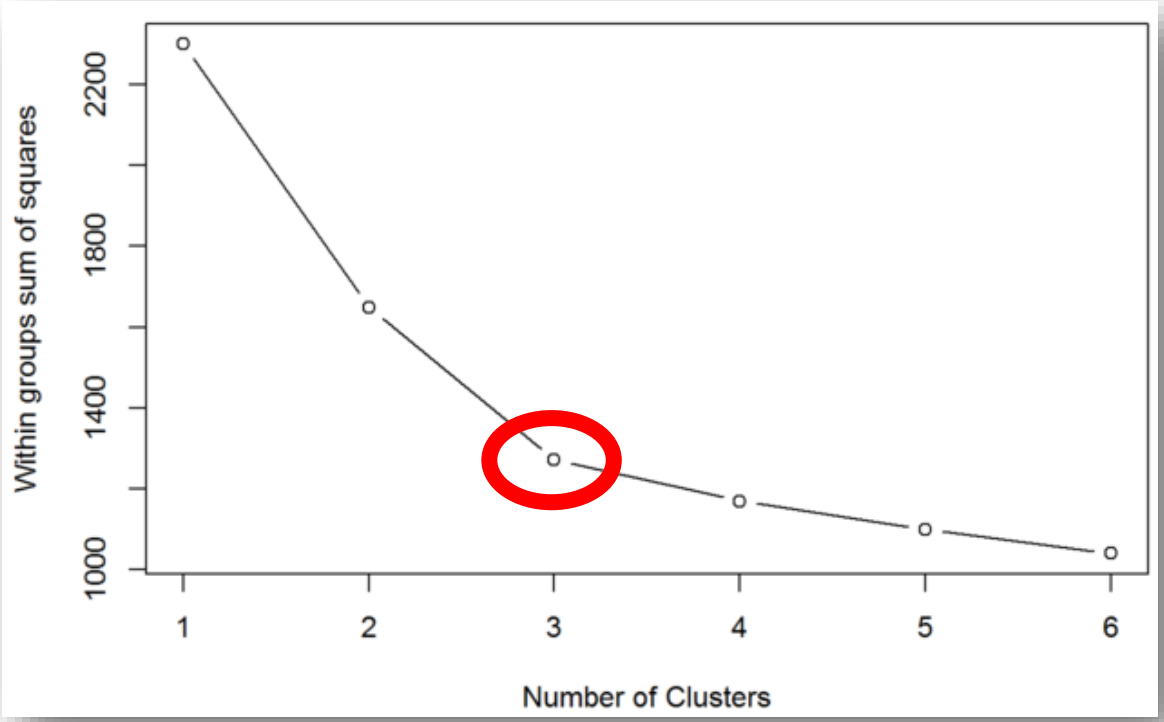
# **kmeans** function

```
> kmeansFit <-
  kmeans(dt.washers,4)
> attributes(kmeansFit)

> kmeansFit$centers
> kmeansFit$cluster
```

# Choosing the right ***k*** value

A fundamental question is how to determine the value of the parameter ***k***. If we looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion".


Within groups sum of squares
1000
1400
1800
2200
1
2
3
4
5
6
Number of Clusters

```
> wssplot <- function(data, nc=15, seed=1234){
     wss <- (nrow(data)-1)*sum(apply(data,2,var))
     for (i in 2:nc){
          set.seed(seed)
          wss[i] <- sum(kmeans(data,
                                  centers=i)$withinss)}
     plot(1:nc, wss, type="b", xlab="Number of
             Clusters‟, ylab="Within groups sum of
             squares")}

> wssplot(dt.washers, nc=6)
```

# Using the **cluster** library

- Library **clusters** allow us to represent (with the aid of PCA) the cluster solution into 2 dimensions

```
> library(cluster)
> clusplot(dt.washers,
  kmeansFit$cluster, main='2D
  representation of the Cluster
  solution',color=TRUE,
  shade=TRUE,labels=2, lines=0)
```

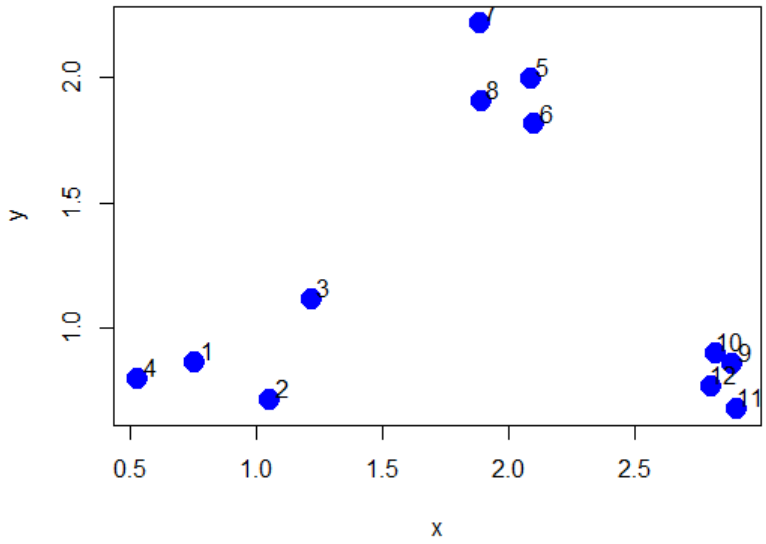
# Agglomerative Clustering

## R codes

# Dataset Preparation

```
> set.seed(1234)
> x<-rnorm(12,mean=rep(1:3,each=4),sd=0.2)
> y<-rnorm(12,mean=rep(c(1,2,1),each=4),sd=0.2)
> plot(x,y,col="blue",pch=19,cex=2)
> text(x+0.05,y+0.05,labels=as.character(1:12))
```

# Dataset



# Hierarchical Clustering

- To run Hierarchical Clustering, you need to look for the pair-wise distances between the points.

```
> df<-data.frame(x,y)
> distxy<-dist(df)
```

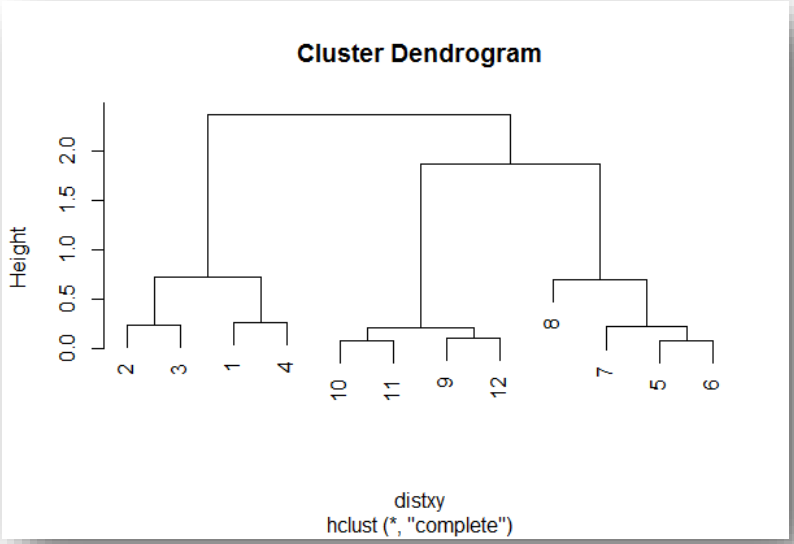
# Hierarchical Clustering

```
> dist(df)
           1          2          3          4          5          6          7          8          9         10         11
2  0.34120511
3  0.57493739 0.24102750
4  0.26381786 0.52578819 0.71861759
5  1.69424700 1.35818182 1.11952883 1.80666768
6  1.65812902 1.31960442 1.08338841 1.78081321 0.08150268
7  1.49823399 1.16620981 0.92568723 1.60131659 0.21110433 0.21666557
8  1.99149025 1.69093111 1.45648906 2.02849490 0.61704200 0.69791931 0.65062566
9  2.13629539 1.83167669 1.67835968 2.35675598 1.18349654 1.11500116 1.28582631 1.76460709
10 2.06419586 1.76999236 1.63109790 2.29239480 1.23847877 1.16550201 1.32063059 1.83517785 0.14090406
11 2.14702468 1.85183204 1.71074417 2.37461984 1.28153948 1.21077373 1.37369662 1.86999431 0.11624471 0.08317570
12 2.05664233 1.74662555 1.58658782 2.27232243 1.07700974 1.00777231 1.17740375 1.66223814 0.10848966 0.19128645 0.20802789
```

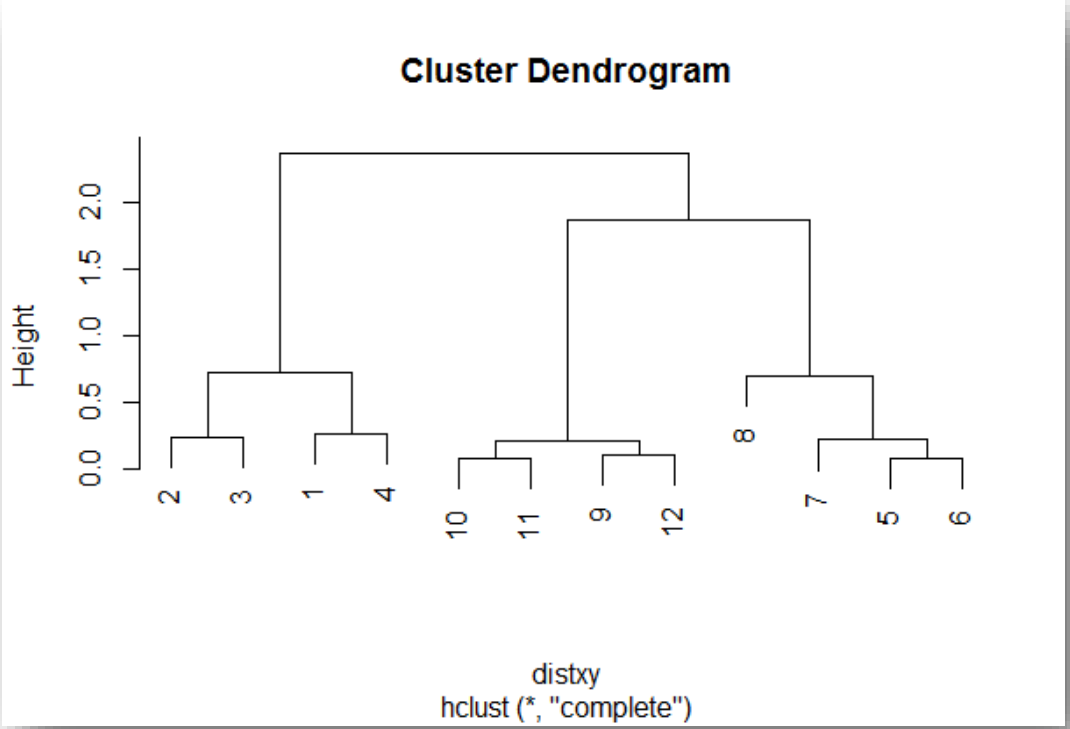
# Hierarchical Clustering

```
> clusters<-hclust(distxy)
> plot(clusters)
```

You need to determine where to "cut" the tree!



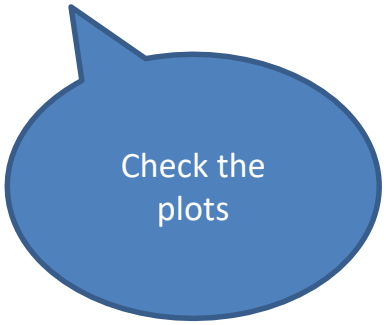
# Hierarchical Clustering



# Hierarchical Clustering

Try with different k,

```
> rect.hclust(clusters, k=2, border="red")
> rect.hclust(clusters, k=3, border="blue")
```



# Exercise: The Laundry Dataset

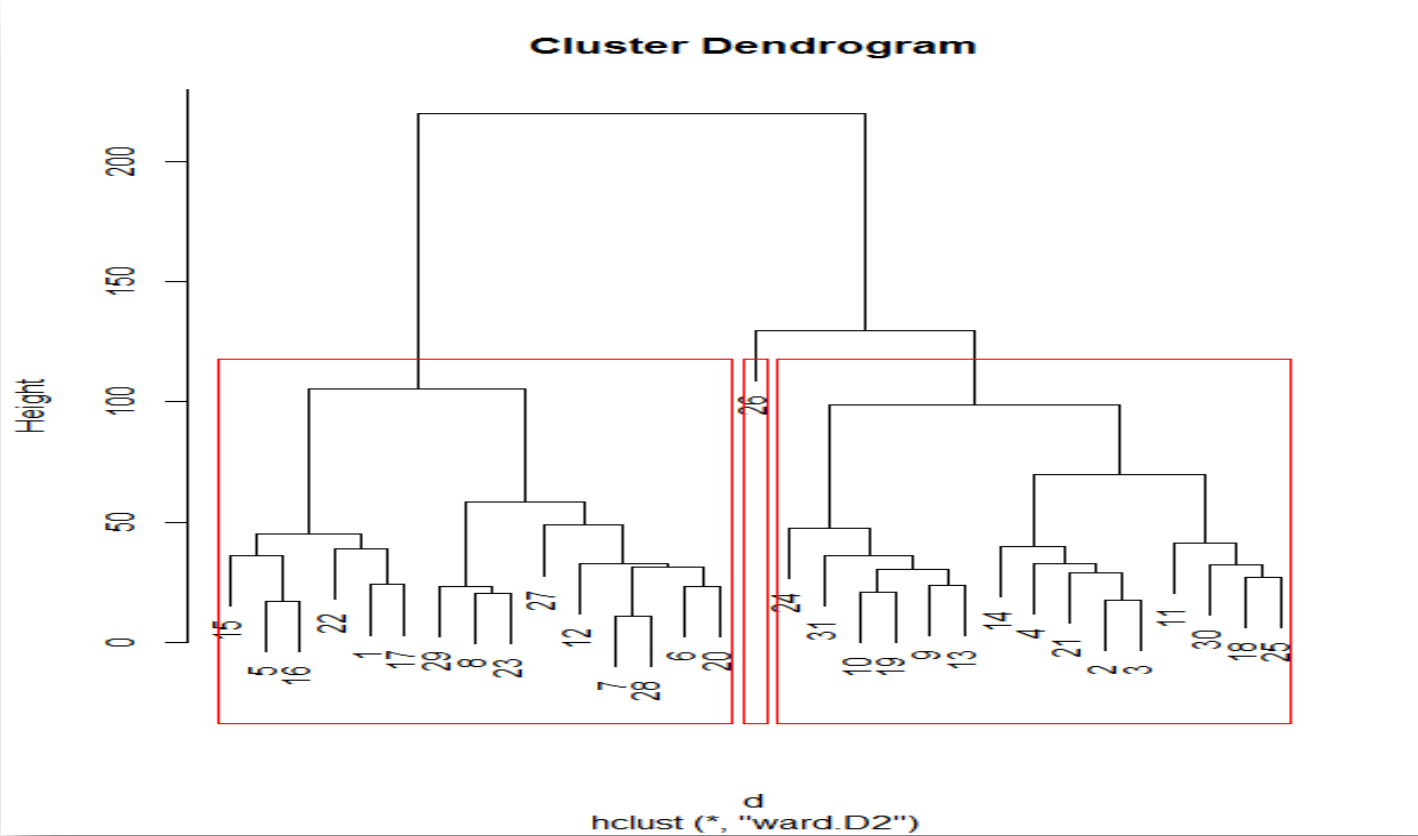
- Distance measurement

```
> d <- dist(dt.washers, method =
  "euclidean")
```

# Exercise: The Laundry Dataset

- Clustering output using ***dendrogram***

```
> H.fit <- hclust(d,
  method="ward.D2")
> plot(H.fit) # display dendogram
> groups <- cutree(H.fit, k=3)
> rect.hclust(H.fit, k=3,
  border="red")
```


Cluster Dendrogram
Height
0
50
100
150
200
15
5
16
22
1
17
29
8
23
27
12
7
28
6
20
26
24
31
10
19
9
13
14
4
21
2
3
11
30
18
25
d
hclust (*, "ward.D2")

# Case Study

# Scenario

Let us consider 25 European countries ($n$ = 25 units) and their protein intakes (in percent) from nine major food sources ($p$ = 9).

The data can be found in **`protein.csv`**.

```
> food <- read.csv('protein.csv')
> head(food)
```

# Preparing Data for Clustering

```
> set.seed(1234)
> grpMeat <-
  kmeans(food[,c("WhiteMeat","RedMeat")],
  centers=3, nstart=10)
> grpMeat
> grpMeat$cluster
> grpMeat$centers
>
plot(food$WhiteMeat,food$RedMeat,col=grpMeat$cl
uster,
  pch=19,cex=2)
>
points(grpMeat$centers,col='blue',pch=3,cex=2,
lwd=4)
```

# Reordering the results

```
> o <- order(grpMeat$cluster)
>
data.frame(food$Country[o],grpMe
at$cluster[o])
```

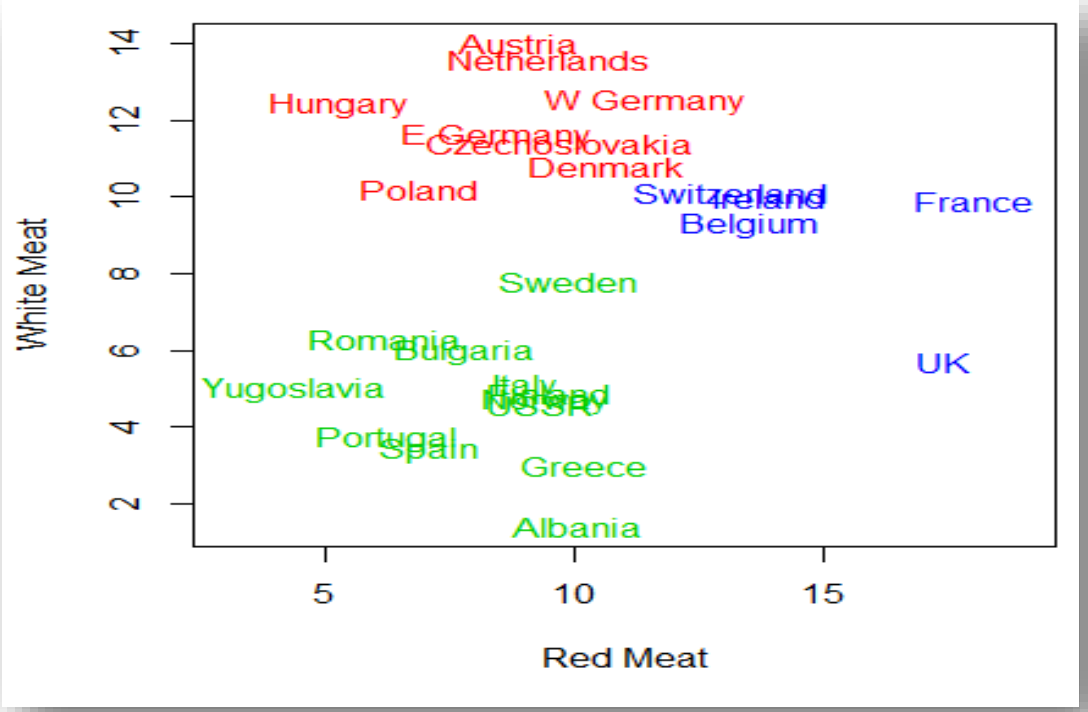
```
> o <- order(grpMeat$cluster)
> data.frame(food$Country[o],grpMeat$cluster[o])
   food.Country.o. grpMeat.cluster.o.
1          Austria                  1
2   Czechoslovakia                  1
3          Denmark                  1
4        E Germany                  1
5          Hungary                  1
6      Netherlands                  1
7           Poland                  1
8        W Germany                  1
9          Albania                  2
10        Bulgaria                  2
11         Finland                  2
12          Greece                  2
13           Italy                  2
14          Norway                  2
```

# Plotting the Results

```
> plot(food$RedMeat,
food$WhiteMeat, type="n",
  xlim=c(3,19), xlab="Red Meat",
ylab="White
  Meat")
> text(x=food$Red, y=food$White,

labels=food$Country,col=grpMeat$
cluster+1)
```

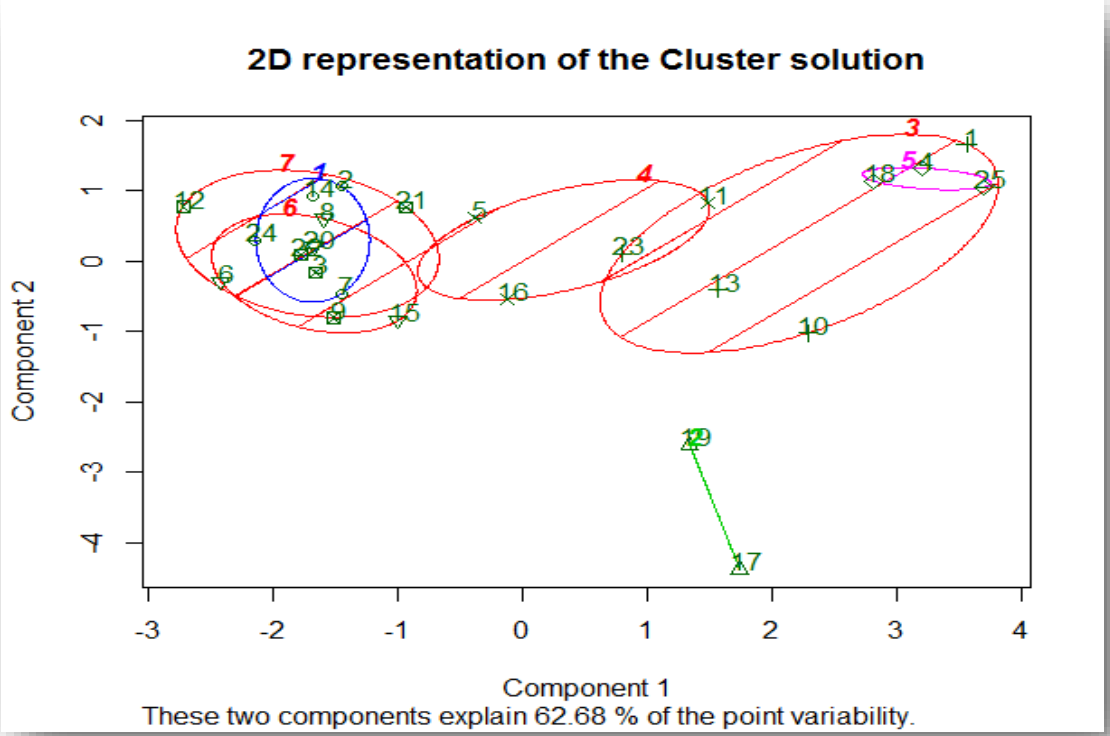
# Plotting the Results



# Clustering the Countries

```
> library(cluster)
> clusplot(food[,-1],
grpProtein$cluster,
  main='2D representation of the
Cluster
  solution', color=TRUE,
shade=TRUE, labels=2,
  lines=0)
```

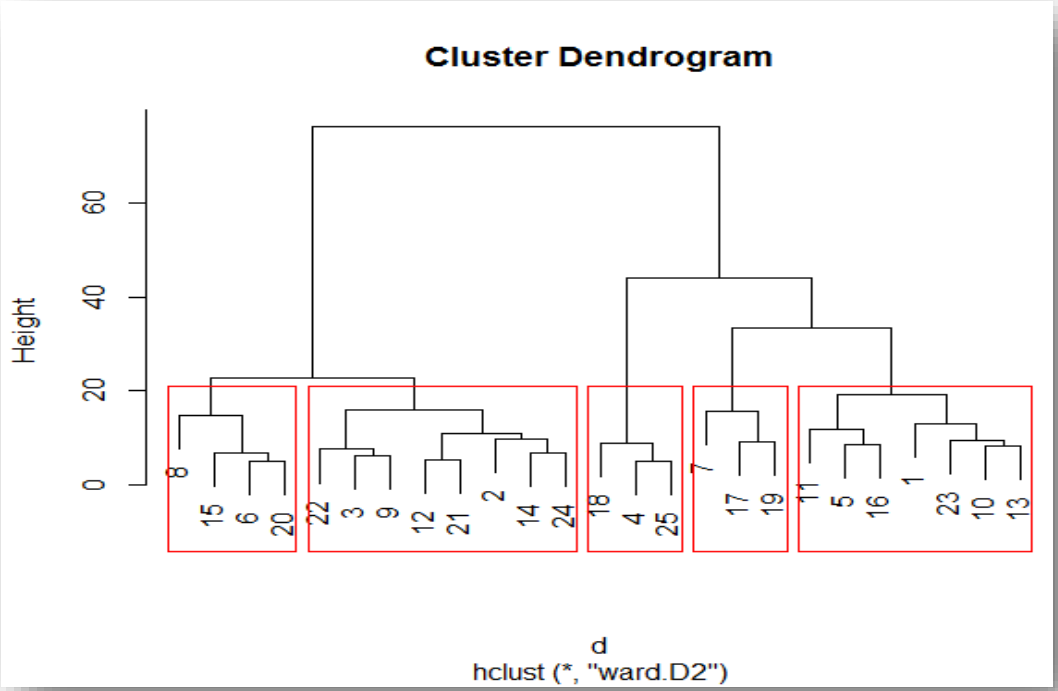
# Clustering the Countries



# Dendrogram

```
> d <- dist(food, method =
"euclidean")
> H.fit <- hclust(d,
method="ward.D2")
> plot(H.fit)
> groups <- cutree(H.fit, k=5)
> rect.hclust(H.fit, k=5,
border="red")
```

# Dendrogram


Cluster Dendrogram
Height
0
20
40
60
8
15
6
20
22
3
9
12
21
2
14
24
18
4
25
7
17
19
11
5
16
1
23
10
13
d
hclust (*, "ward.D2")