HW4_Solutions.R

Heramb

Wed Nov 29 23:58:00 2017

```
#Heramb Vijay Uttarwar
#[redacted]
#CS-422 HW-4

library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.4.2
```

```
library(curl)
```

```
## Warning: package 'curl' was built under R version 3.4.2
```

```
library(cluster)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.4.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://go
o.gl/13EFCZ
```

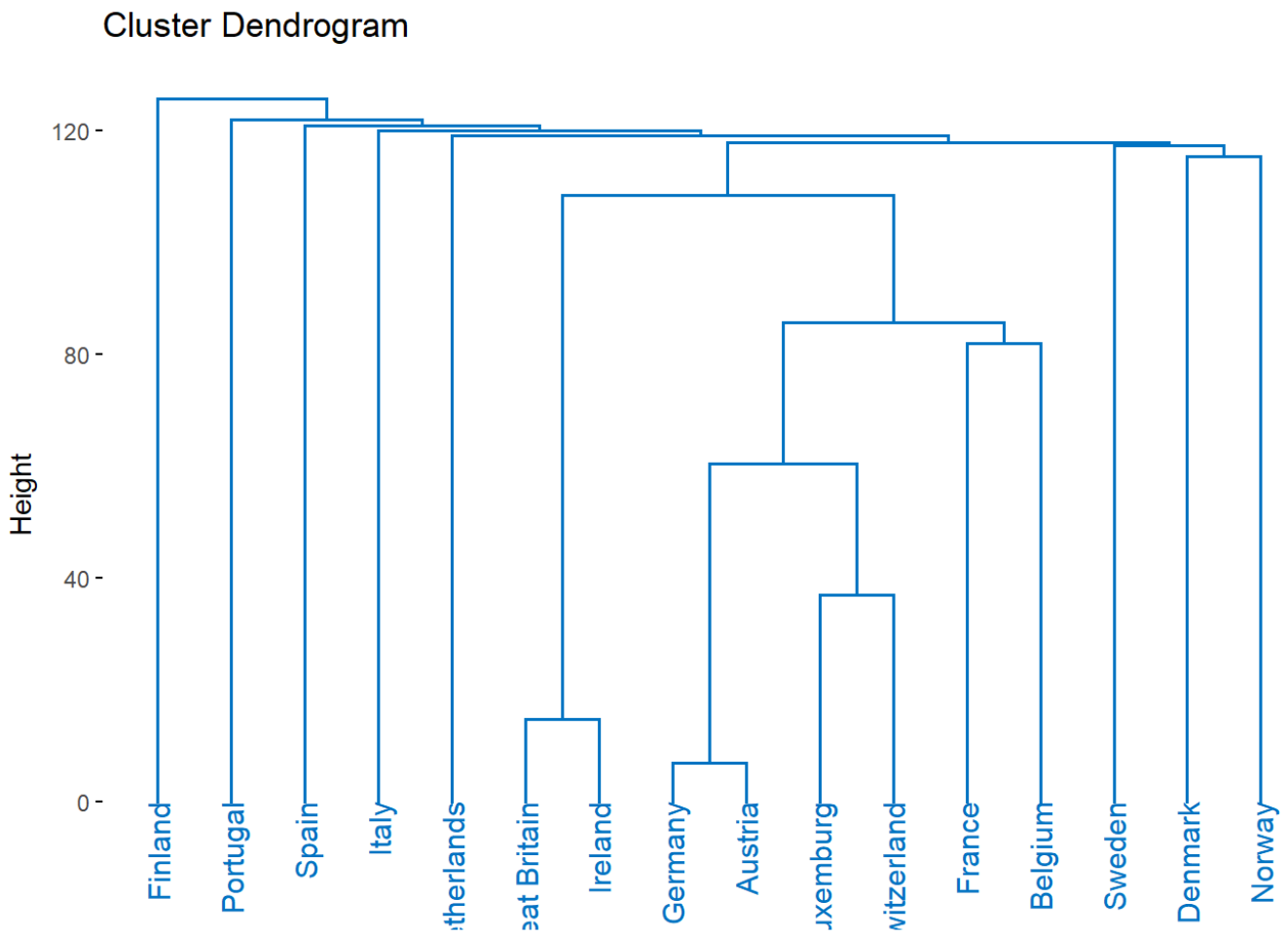
```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.4.2
```

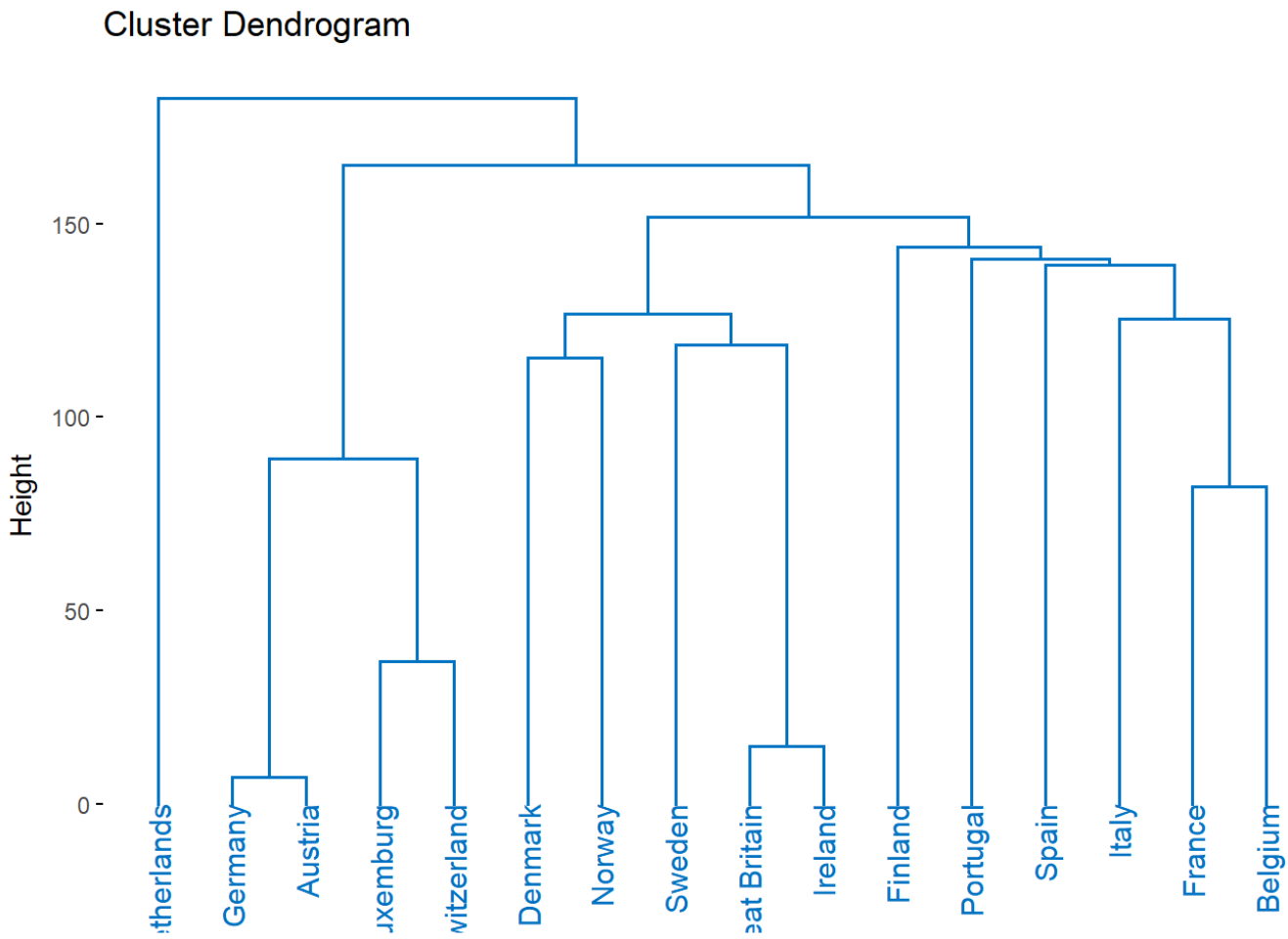
```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

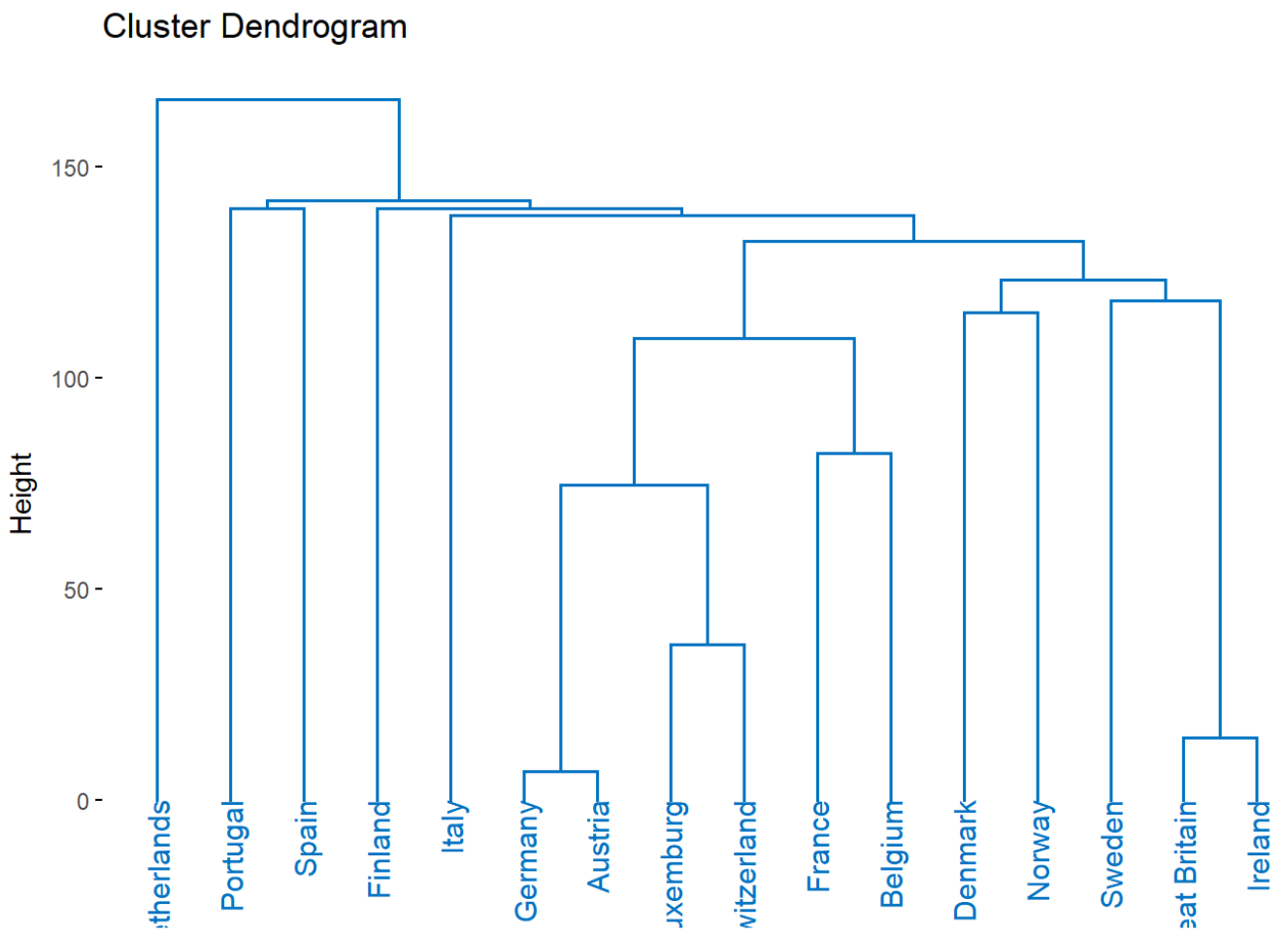
```
setwd("C:/Users/Heramb/Desktop/CS 422/HW 4/")
rm(list=ls())

lang <- fread('https://people.sc.fsu.edu/~jburkardt/datasets/hartigan/file46.txt')
#Used to fetch the data from the address

lang1 <- read.csv("lang", header = T, sep = ",", row.names = 'Country')
lang1$X=NULL

#2.1(a)
hc.single <- factoextra::eclust(lang1, "hclust", hc_method="single")
fviz_dend(hc.single, show_labels=TRUE, palette="jco", as.ggplot=T)
```




```
hc.complete <- factoextra::eclust(lang1, "hclust", hc_method="complete")
fviz_dend(hc.complete, show_labels=TRUE, palette="jco", as.ggplot=T)
```



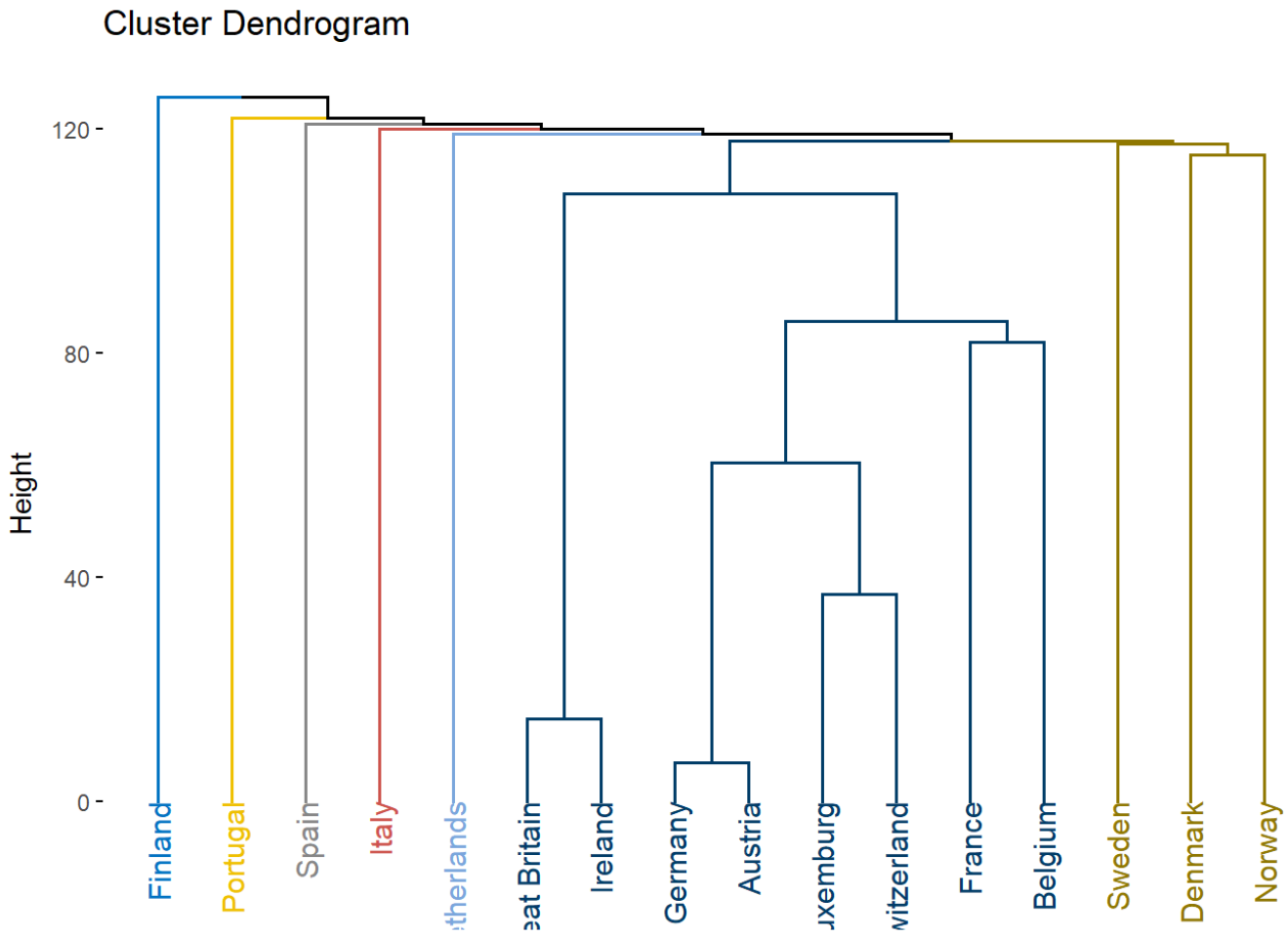

```
hc.average <- factoextra::eclust(lang1, "hclust", hc_method="average")
fviz_dend(hc.average, show_labels=TRUE, palette="jco", as.ggplot=T)
```



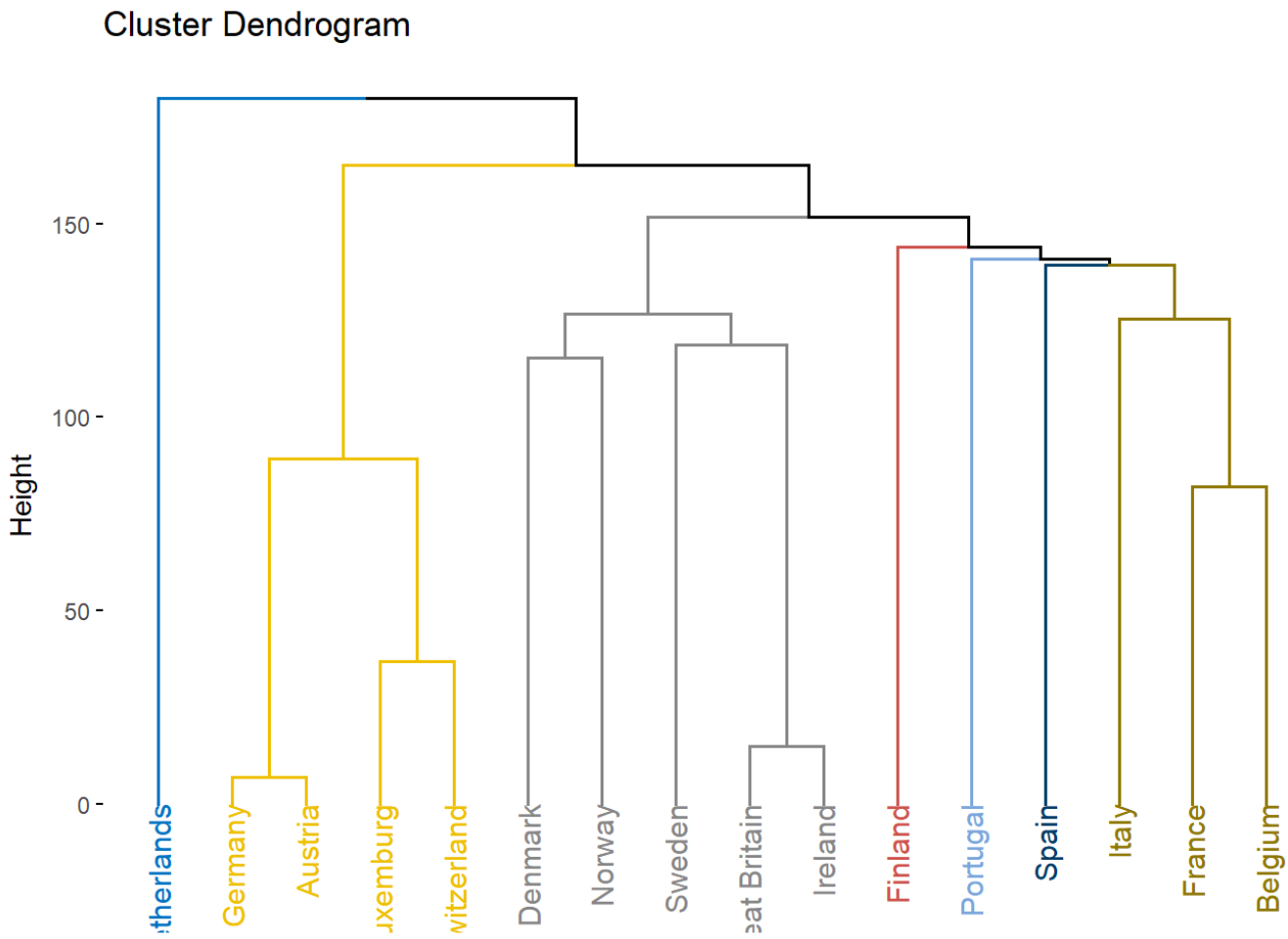

```
#2.1(b)

#1.Method->Single
#Two singleton clusters {Great Britain,Ireland},{West Germany,Austria},{Luxemburg,S
witzerland},{France, Belgium},{Denmark, Norway}

#2.Method->Complete
#Two singleton clusters {Denmark, Norway}, {Great Britain,Ireland},{West Germany,Au
stria},{Luxemburg,Switzerland}, {France, Belgium}

#3.Method->Average
#Two singleton clusters {Portugal,Spain},{Denmark, Norway},{France, Belgium}, {Grea
t Britain,Ireland},{West Germany,Austria},{Luxemburg,Switzerland}

#2.1(c)
#I think Italy should be clustered with a large cluster. Looking at the raw data, I
taly has higher dissimilarity with other countries.
#Thus, if we cut at certain cutoff will lead Italy as a outlier and will form a clu
ster with lesser dissimilarity.

#2.1(d)
#Purity as the linkage strategy that produces the most two-singleton cluster, there
is only one method i.e "Average".
#Linkage with method="Average" is pure by definition

#2.1(e)
cuttree.125<-cutree(hc.average,h=125)
table(cuttree.125)
```

```
## cuttree.125
## 1 2 3 4 5 6 7
## 6 1 1 5 1 1 1
```

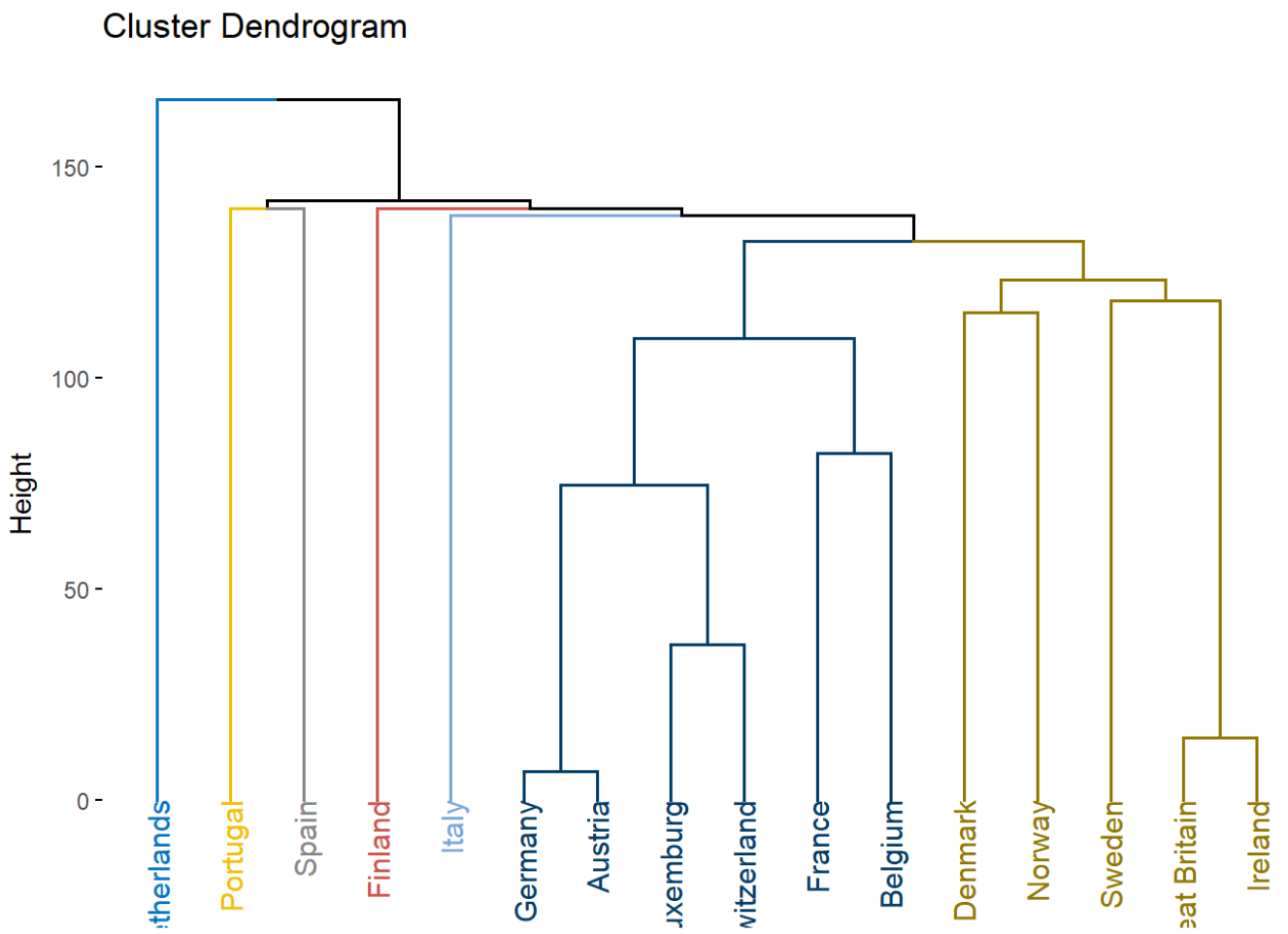
```
#There are 7 clusters at height 125.

#2.1(f)

hc.single1 <- factoextra::eclust(lang1, "hclust", k=7, hc_method="single")
fviz_dend(hc.single1, show_labels=TRUE, palette="jco", as.ggplot=T)
```




```
hc.complete1<- factoextra::eclust(lang1, "hclust", k=7, hc_method="complete")
fviz_dend(hc.complete1, show_labels=TRUE, palette="jco", as.ggplot=T)
```




```
hc.average1 <- factoextra::eclust(lang1, "hclust",k=7, hc_method="average")
fviz_dend(hc.average1, show_labels=TRUE, palette="jco", as.ggplot=T)
```




```
#2.1(g)
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 3.4.2
```

```
ct<-dist(lang1)

stats <- cluster.stats(ct, hc.single1$cluster, silhouette=TRUE)
stats$dunn
```

```
## [1] 0.7813006
```

```
stats$avg.silwidth
```

```
## [1] 0.1215148
```

```
stats1 <- cluster.stats(ct, hc.complete1$cluster, silhouette=TRUE)
stats1$dunn
```

```
## [1] 0.6768822
```

```
stats1$avg.silwidth
```

```
## [1] 0.1922308
```

```
stats2<- cluster.stats(ct, hc.average1$cluster,silhouette=TRUE)
stats2$dunn
```

```
## [1] 0.807345
```

```
stats2$avg.silwidth
```

```
## [1] 0.1698248
```

```
#2.1(h)
#Dunn index for hc.average is maximum. Thus, hc.average is the best cluster.

#2.1(i)
#silhouette width for hc.complete is maximum. Thus, hc.complete is the best cluster
.



#2.2
library(textreuse)
```

```
## Warning: package 'textreuse' was built under R version 3.4.2
```

```
files <- list.files("C:/Users/Heramb/Desktop/CS 422/HW 4/corpus", full.names = T)
minhash <- minhash_generator(n=160, seed=100)

#2.2(a)
corpus <- TextReuseCorpus(files, tokenizer = tokenize_ngrams, n = 5,
                          minhash_func = minhash, keep_tokens = TRUE)
length(unlist(tokens(corpus)))
```

```
## [1] 22075
```

```
#2.2(b)
library(magrittr)
totaltokens <- tokens(corpus)
corpusMat <- list.files("C:/Users/Heramb/Desktop/CS 422/HW 4/corpus", full.names=F)
doc_dict <- unlist(tokens(corpus)) %>% unique()
Matr <- lapply(totaltokens, function(set, dict) {   as.integer(dict %in% set)}, dic
t = doc_dict) %>% data.frame()
tempSetName <-setNames( Matr, paste( corpusMat, 1:length(corpusMat)) )
rownames(Matr) <- doc_dict
dim(Matr)
```

```
## [1] 17614   100
```

```
#i.e. 17614*100

#2.2(c)
tokens(corpus[["orig_taske"]])[1:5]
```

```
## [1] "in mathematics and computer science"
## [2] "mathematics and computer science dynamic"
## [3] "and computer science dynamic programming"
## [4] "computer science dynamic programming is"
## [5] "science dynamic programming is a"
```

```
#2.2(d)
#As we choose only 240 rows for signature matrix, The dimensions of signature matri
x will become 240*100.
# while we have generated Characteristic matrix of dimension 17614*100.
# reduction of size of problem is 98.637%.

#2.2(e)
lsh_probability(h = 240, b =  60, s = 0.3)
```

```
## [1] 0.3861342
```

```
#This probability is less than the desired one and thus increasing the bands to 80.
lsh_probability(h = 240, b =  80, s = 0.3)
```

```
## [1] 0.8880492
```

```
#We get the desired probability i.e 80 at bands=80.

#2.2(f)
buckets <- lsh(corpus, bands = 80)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.2
```

```
candidates <- lsh_candidates(buckets)
noofcandidatepair<-nrow(candidates)
#Number of candidate pairs
noofcandidatepair
```

```
## [1] 72
```

```
#2.2(g)
lshResult <- lsh_compare(candidates, corpus, jaccard_similarity)
lshResult[order(lshResult$score,decreasing = TRUE),][1:5,]
```

```
## # A tibble: 5 x 3
##            a          b     score
##        <chr>      <chr>     <dbl>
## 1 g4pC_taska orig_taska 0.7896254
## 2 g3pA_taskd orig_taskd 0.7186630
## 3 g4pB_taske orig_taske 0.4491803
## 4 g3pA_taskd g4pC_taskd 0.4142857
## 5 g2pB_taske orig_taske 0.3982906
```

```
#2.2(h)
#If we dont use Locality Sensative Hashing and directly examined every pair for sim
ilarity then
# Number of pairs of documents to be examined  = (No of Documents)C2
# Here we can write => No of Documents = 100
# No of pairs = 100C2 = 100!/(98!*2!) = 4950
# Solution for 2.2 (h)(ii)
# No of candidate pairs generated in 2.2 (f) = 72.
# The ratio of doc pair to candidate pair number is : 4950/72 = 68.75
# It shows that if we dont do Locality Sensative Hashing the number of comparisons
we have to do is 68.75 times than number of comparisons we will do after doing Loca
lity Sensative Hashing.

#2.3(a)

u.item<-read.csv("C:/Users/Heramb/Desktop/CS 422/HW 4/ml-100k/u.item", sep = "|",co
mment.char = "#")
u.item$X1=NULL
u.data<-read.csv("C:/Users/Heramb/Desktop/CS 422/HW 4/ml-100k/u.data", sep = "\t",
header = T,comment.char = "#")
#2.3(i)
user200_rownumber <- which(u.data$user== 200)
user50_rownumber <- which(u.data$user == 50)

user200 <- u.data[user200_rownumber,]
user50 <- u.data[user50_rownumber,]

movies200 <-u.item[user200[,2],]
movies50 <- u.item[user50[,2],]

movie.matrix200 <- movies200[,6:24]
genre200 <- apply(movie.matrix200,2,mean)
vector200 <- as.vector(genre200)

movie.matrix50 <- movies50[,6:24]
genre50 <- apply(movie.matrix50,2,mean)
vector50 <- as.vector(genre50)

cosine <- function(x, y) {
  # Need to do error checking:
  # 1) Ensure x and y are vectors.

  sum(x*y)/(norm(x, type="2") * norm(y, type="2"))
}

cosine(vector50,vector200)
```

```
## [1] 0.54825
```

```
#cluster_similarity(vector200, vector50, similarity="jaccard", method="independence
")

#2.3(ii)
movie127 <- u.item[127,]
vector127 <- as.vector(movie127[,6:24])
cosine(vector127,vector50)
```

```
## [1] 0.6235022
```

```
#2.3(iii)
cosine(vector127,vector200)
```

```
## [1] 0.5533398
```

```
#2.3(iv)
#The movie 127 will be recommended to user 50.

#2.3(b)
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.4.2
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 3.4.2
```

```
##
## Attaching package: 'reshape'
```

```
## The following objects are masked from 'package:reshape2':
##
##     colsplit, melt, recast
```

```
## The following object is masked from 'package:data.table':
##
##     melt
```

```
utilitymatrix<-matrix(0,6,11)

for(i in 1:length(unlist(u.data[,1])))
{
  if(u.data[i,1]==1 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],1]<-u.data[i,3]
  }

  if(u.data[i,1]==21 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],2]<-u.data[i,3]
  }
  if(u.data[i,1]==44 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],3]<-u.data[i,3]
  }
  if(u.data[i,1]==59 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],4]<-u.data[i,3]
  }
  if(u.data[i,1]==72 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],5]<-u.data[i,3]
  }
  if(u.data[i,1]==82 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],6]<-u.data[i,3]
  }
  if(u.data[i,1]==102 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],7]<-u.data[i,3]
  }
  if(u.data[i,1]==234 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],8]<-u.data[i,3]
  }
  if(u.data[i,1]==268 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],9]<-u.data[i,3]
  }
  if(u.data[i,1]==409 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],10]<-u.data[i,3]
  }
  if(u.data[i,1]==486 && u.data[i,2]<7)
  {
    utilitymatrix[u.data[i,2],11]<-u.data[i,3]
  }
}
colnames(utilitymatrix)<-c("user1","user21","user44","user59","user72","user82","us
er102","user234","user268","user409","user486")

#View(utilitymatrix)


means <- apply(utilitymatrix, 1, function(x) mean(x, na.rm=T))
```

```
means
```

```
## [1] 3.363636 1.090909 1.181818 1.545455 1.727273 1.181818
```

```
for (i in 1:dim(utilitymatrix)[1]) {
  for (j in 1:dim(utilitymatrix)[2])
  {
    if(utilitymatrix[i,j]>0)
    {
      utilitymatrix[i,j] <- utilitymatrix[i,j] - means[i]
    }
  }
}
similarmovie<-matrix(0,6,1)

for (i in 1:dim(utilitymatrix)[1])
{
  similarmovie[i,1]<-round(cosine(utilitymatrix[5,], utilitymatrix[i, ]), digits=2
)
}


rownames(similarmovie)<-c("1","2","3","4","5","6")
a<-as.numeric(rownames(similarmovie)[order(similarmovie, decreasing=TRUE)][1:6])

rating268<-((similarmovie[a[2],1]*utilitymatrix[a[2],9])+(similarmovie[a[3],1]*uti
litymatrix[a[3],9])+(similarmovie[a[4],1]*utilitymatrix[a[4],9]))/(similarmovie[a[
2],1]+similarmovie[a[3],1]+similarmovie[a[4],1])
rating268
```

```
##         2
## 0.9059123
```