R code for record data clustering

Rae Zhang

10/12/2021

```
library(ggplot2)
library(reshape2)
library(ggthemes)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WB
a
```

```
library(slam)
library(ggdendro)
library(factoextra)
library(heatmaply)
```

```
## Loading required package: plotly
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
## Loading required package: viridis
```

```
## Loading required package: viridisLite
```

```
##
## ======================
## Welcome to heatmaply version 1.1.1
##
## Type citation('heatmaply') for how to cite the package.
## Type ?heatmaply for the main documentation.
##
## The github page is: https://github.com/talgalili/heatmaply/
## Please submit your suggestions and bug-reports at: https://github.com/talgalili/heatm
aply/issues
## Or contact: <tal.galili@gmail.com>
## ======================
```

```
library(htmlwidgets)
library(yaml)
library(fpc)
library(dbscan)
```

```
##
## Attaching package: 'dbscan'
```

```
## The following object is masked from 'package:fpc':
##
##     dbscan
```

```
library(ClusterR)
```

```
## Loading required package: gtools
```

```
library(amap)


setwd("/Users/raezh1/PycharmProjects/anly501/data cleaning")

df1 <- read.csv("CPT_DF.csv", na.string=c("", " "))
df2 <- read.csv("Corn_DF.csv", na.string=c("", " "))
df3 <- read.csv("Cattle_DF.csv", na.string=c("", " "))
df4 <- read.csv("Livestock_DF.csv", na.string=c("", " "))
df5 <- read.csv("Dairy_DF.csv", na.string=c("", " "))
df6 <- read.csv("Wheat_DF.csv", na.string=c("", " "))

DF <- rbind(df1, df2, df3, df4, df5, df6)
str(DF)
```

```
## 'data.frame':    65 obs. of  14 variables:
##  $ X                           : int  1 2 3 4 5 6 7 1 2 3 ...
##  $ state                       : chr  "Arkansas" "California" "Florida" "Georgia"
...
##  $ year                        : int  2019 2019 2019 2019 2019 2019 2019 2019 2019 2
019 ...
##  $ Gross.cash.income           : int  761131 929911 920986 688574 1985650 815359 710
738 606318 1015413 422046 ...
##  $ Net.farm.income             : int  258540 137872 238245 115529 608968 162883 1864
58 221718 646865 110376 ...
##  $ Other.related.income        : int  108635 311342 17459 61354 109300 124413 97188
142594 15046 42131 ...
##  $ farm.count                  : int  554 154 245 1999 207 858 3217 358 148 23811
...
##  $ Gross.cash.income.per.farm  : int  1374 6038 3759 344 9593 950 221 1694 6861 18
...
##  $ Net.farm.income.per.farm    : int  467 895 972 58 2942 190 58 619 4371 5 ...
##  $ Other.related.income.per.farm: int  196 2022 71 31 528 145 30 398 102 2 ...
##  $ gross.income.status         : chr  "more than one thousand" "more than one thousa
nd" "more than one thousand" "below one thousand" ...
##  $ Expenses                    : int  502591 792039 682741 573045 1376682 652476 524
280 384600 368548 311670 ...
##  $ Expenses.per.farm           : int  907 5143 2787 287 6651 760 163 1074 2490 13
...
##  $ profit.margin.              : int  34 15 26 17 31 20 26 37 64 26 ...
```

```
DF <- mutate_all(DF, function(x) as.numeric(as.character(x)))
```

```
## Warning in (function (x) : NAs introduced by coercion
```

```
## Warning in (function (x) : NAs introduced by coercion
```

```
str(DF)
```

```
## 'data.frame':    65 obs. of  14 variables:
##  $ X                           : num  1 2 3 4 5 6 7 1 2 3 ...
##  $ state                       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ year                        : num  2019 2019 2019 2019 2019 ...
##  $ Gross.cash.income           : num  761131 929911 920986 688574 1985650 ...
##  $ Net.farm.income             : num  258540 137872 238245 115529 608968 ...
##  $ Other.related.income        : num  108635 311342 17459 61354 109300 ...
##  $ farm.count                  : num  554 154 245 1999 207 ...
##  $ Gross.cash.income.per.farm  : num  1374 6038 3759 344 9593 ...
##  $ Net.farm.income.per.farm    : num  467 895 972 58 2942 ...
##  $ Other.related.income.per.farm: num  196 2022 71 31 528 ...
##  $ gross.income.status         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Expenses                    : num  502591 792039 682741 573045 1376682 ...
##  $ Expenses.per.farm           : num  907 5143 2787 287 6651 ...
##  $ profit.margin.              : num  34 15 26 17 31 20 26 37 64 26 ...
```

save the labels and clean the data

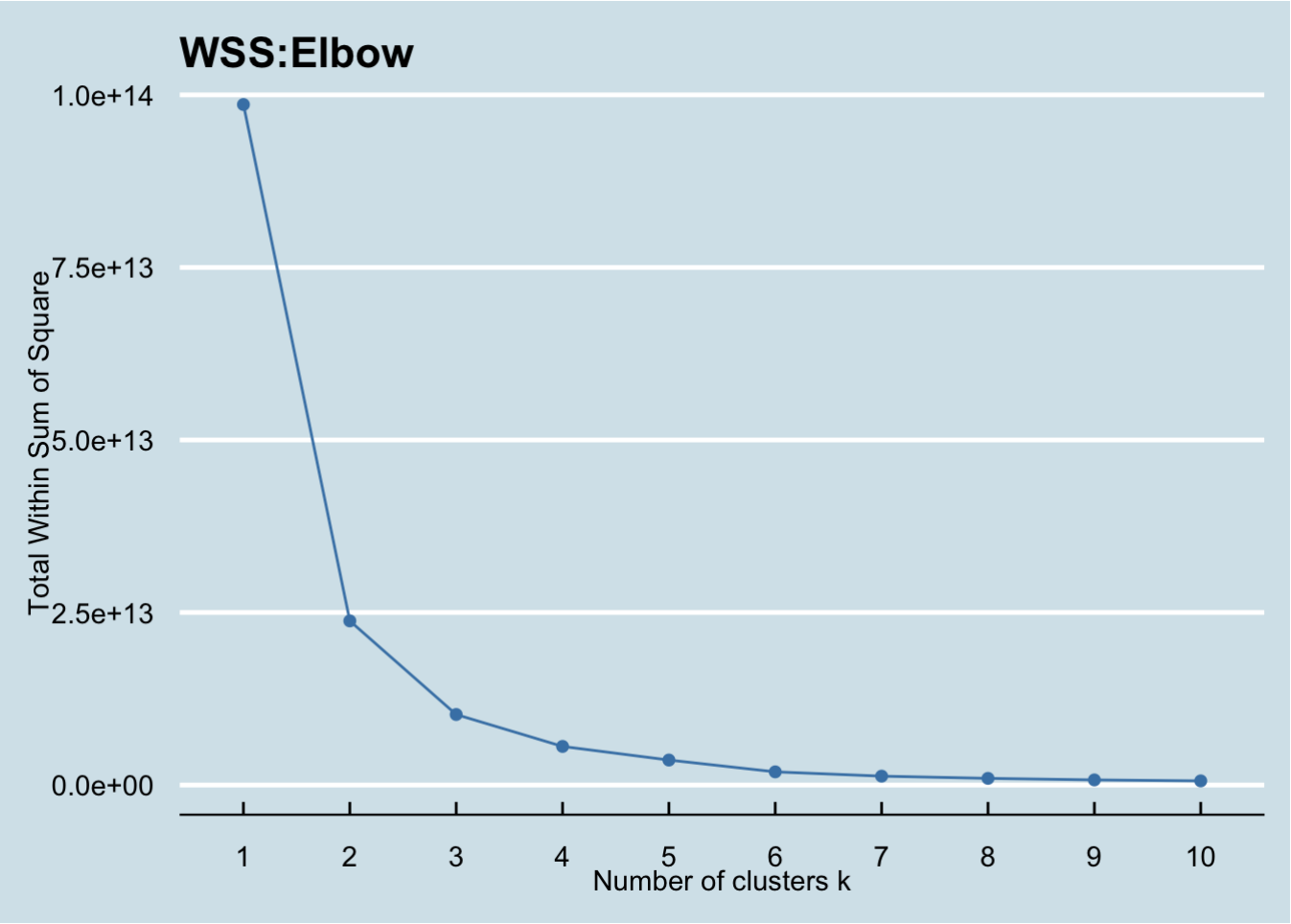
```
label1 <- DF$state
label2 <- DF$gross.income.status
DF <- DF %>% select(-"X", -"year", -"state", -"gross.income.status")
```

Find The Optimum Value Of K

1. Elbow method

Look at optimal cluster numbers using silh, elbow, gap

```
(WSS <- fviz_nbclust(DF, FUN = hcut, method = "wss", k.max = 10) +
  ggtitle("WSS:Elbow") +
    theme_economist() +
    scale_color_economist() +
    scale_fill_brewer(palette = "PuBuGn"))
```

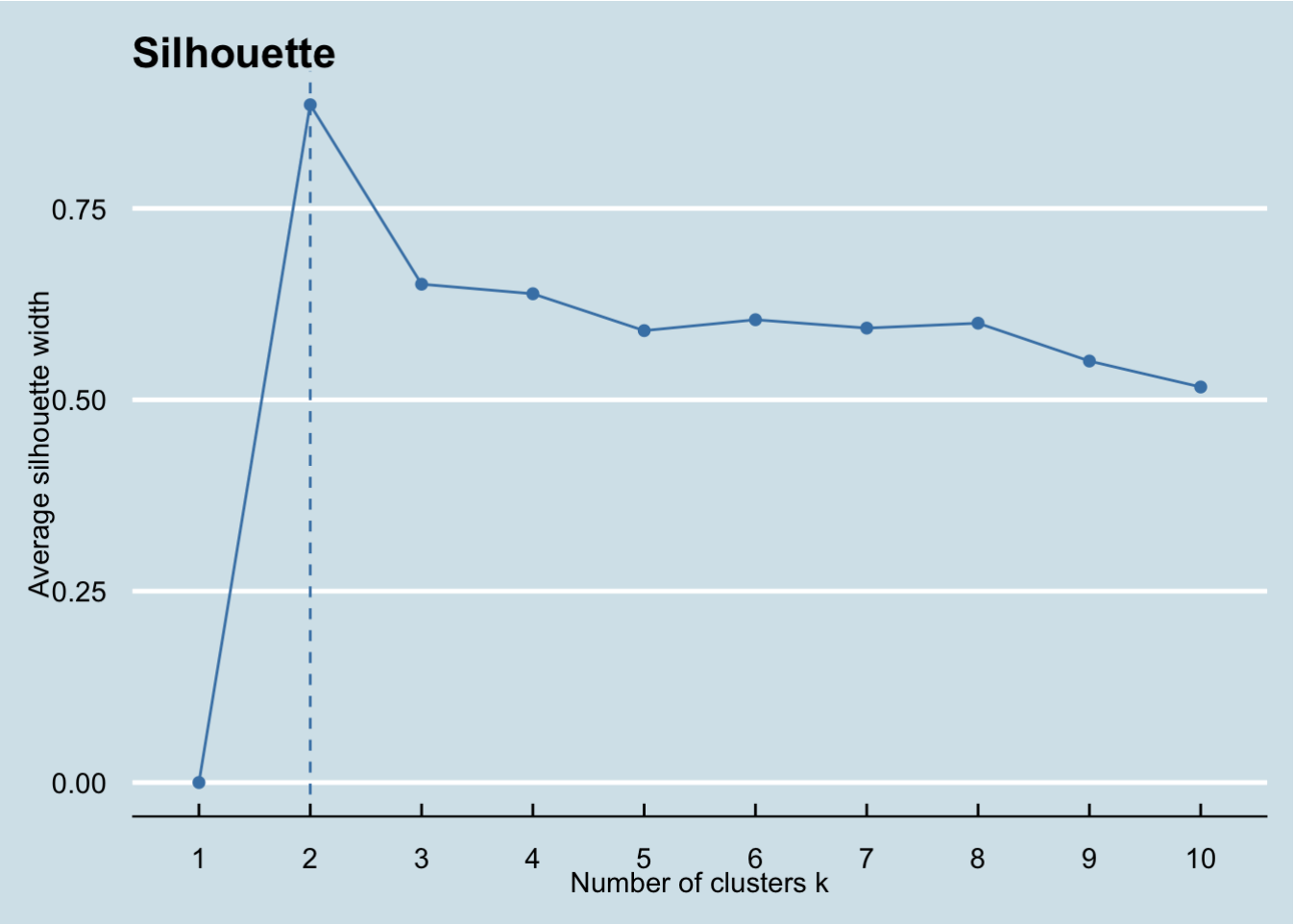


###

From the elbow graph, it shows k can be 2 or 3

2. Silhouette method

```
(SIL <- fviz_nbclust(DF, FUN = hcut, method = "silhouette", k.max = 10) +
  ggtitle("Silhouette") +
  theme_economist() +
  scale_color_economist() +
  scale_fill_brewer(palette = "PuBuGn"))
```

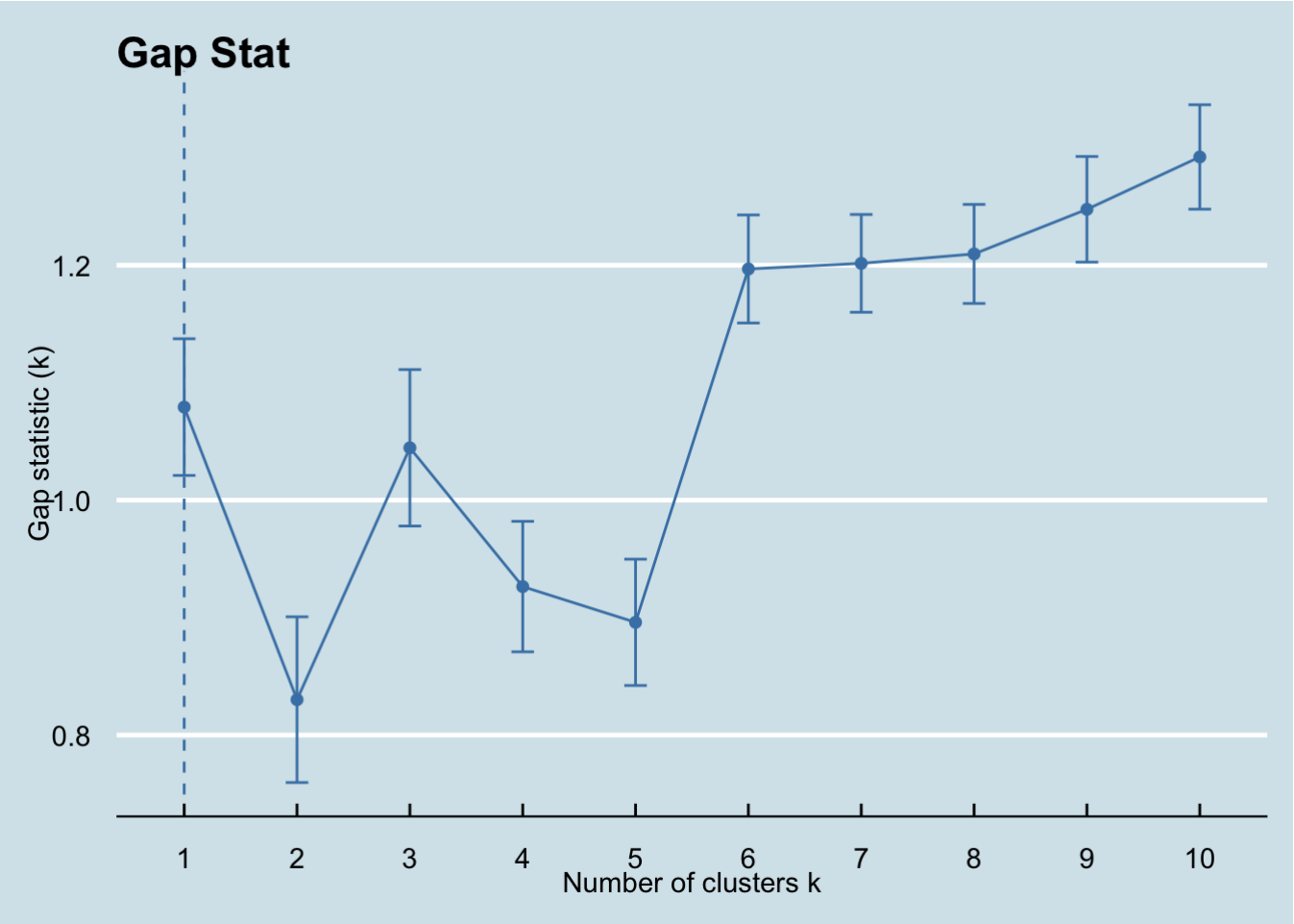


###

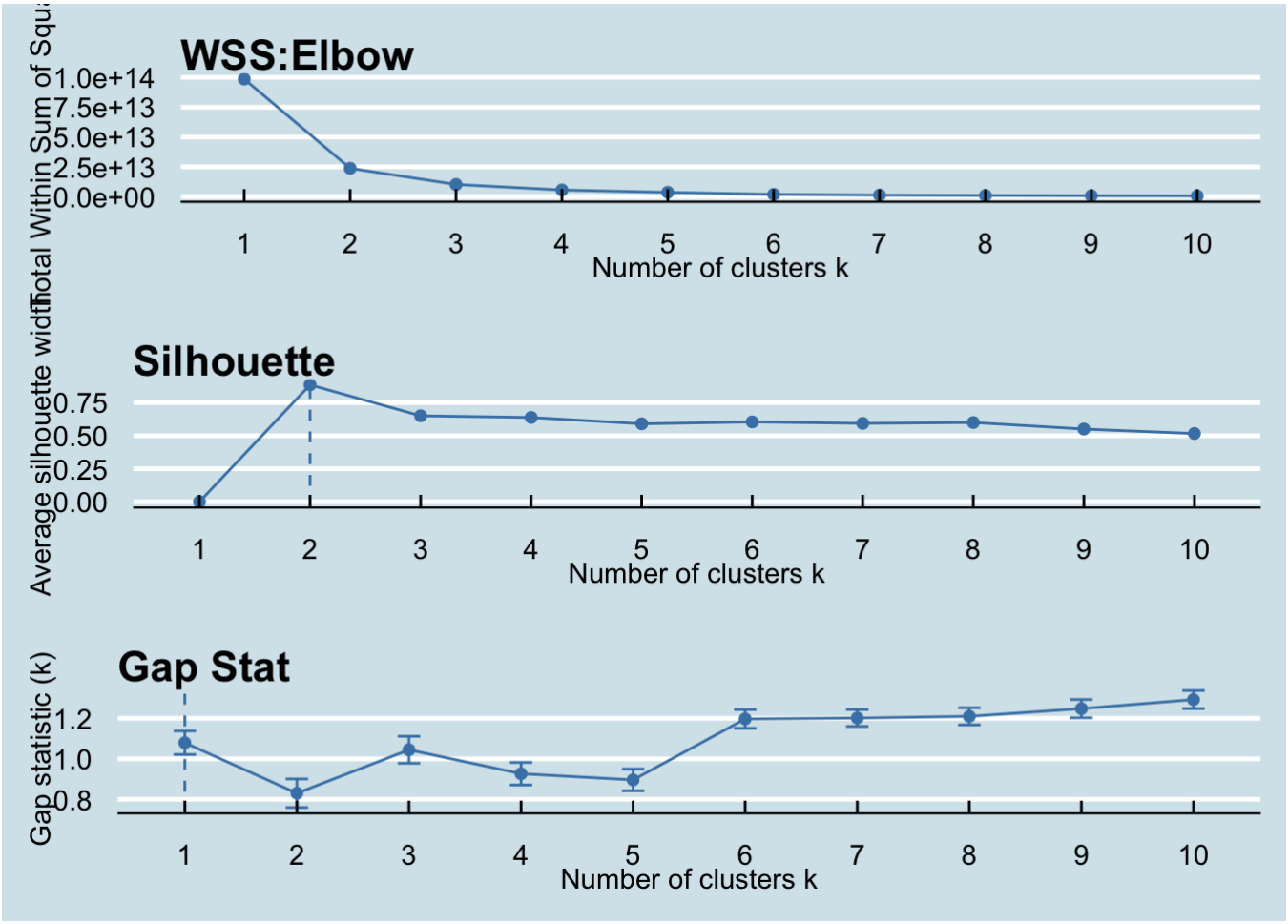
Silhouette tells that the optimum value for k is 2

3. Gap method

```
(GAP <- fviz_nbclust(DF, FUN = hcut, method = "gap_stat", k.max = 10) +
  ggtitle("Gap Stat") +
  theme_economist() +
  scale_color_economist() +
  scale_fill_brewer(palette = "PuBuGn"))
```




```
gridExtra::grid.arrange(WSS, SIL, GAP, nrow = 3) ## Three graphs together
```



##

From the gap stat, the value 1, 2, 3 are all under consideration since there are three big gaps after them

Calculate distance between rows using distance matrices

Convert dataframe to matrix first

Hierarchical clustering

```
(hc_C <- hclust(DF_DT_E, method = "complete" ))
```

```
##
## Call:
## hclust(d = DF_DT_E, method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 65
```

```
complete <- ggdendrogram(hc_C) +
  ggtitle("Complete method") +
  theme_economist() +
  scale_color_economist() +
  scale_fill_brewer(palette = "PuBuGn") +
  scale_x_continuous(breaks = NULL)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```
(hc_D <- hclust(DF_DT_M2, method = "ward.D" ))
```

```
##
## Call:
## hclust(d = DF_DT_M2, method = "ward.D")
##
## Cluster method   : ward.D
## Distance         : minkowski
## Number of objects: 65
```

```
ward.D <- ggdendrogram(hc_D)+
  ggtitle("Ward method") +
  theme_economist() +
  scale_color_economist() +
  scale_fill_brewer(palette = "PuBuGn") +
  scale_x_continuous(breaks = NULL)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```
(hc_D2 <- hclust(DF_DT_Man, method = "ward.D2" ))
```

```
##
## Call:
## hclust(d = DF_DT_Man, method = "ward.D2")
##
## Cluster method   : ward.D2
## Distance         : manhattan
## Number of objects: 65
```

```
ward.D2 <- ggdendrogram(hc_D2)+
  ggtitle("Ward method 2") +
  theme_economist() +
  scale_color_economist() +
  scale_fill_brewer(palette = "PuBuGn") +
  scale_x_continuous(breaks = NULL)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

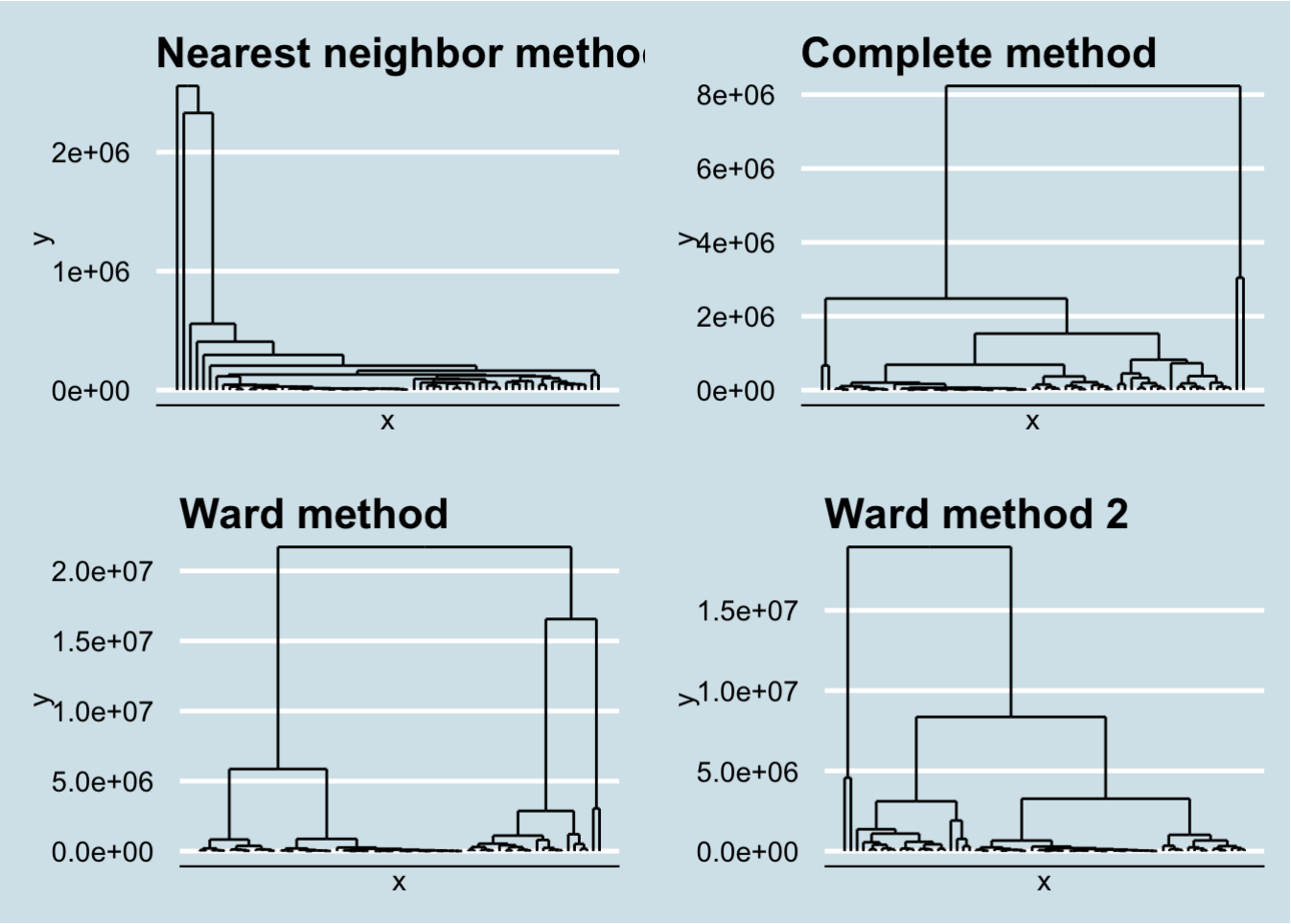
```
(hc_S <- hclust(DF_DT_M4, method = "single" ))
```

```
##
## Call:
## hclust(d = DF_DT_M4, method = "single")
##
## Cluster method   : single
## Distance         : minkowski
## Number of objects: 65
```

```
single <- ggdendrogram(hc_S)+
  ggtitle("Nearest neighbor method") +
  theme_economist() +
  scale_color_economist() +
  scale_fill_brewer(palette = "PuBuGn") +
  scale_x_continuous(breaks = NULL)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```
gridExtra::grid.arrange(single, complete, ward.D, ward.D2, nrow = 2) ## 4 graphs togethe
r
```



Try k = 2 and make visualizations

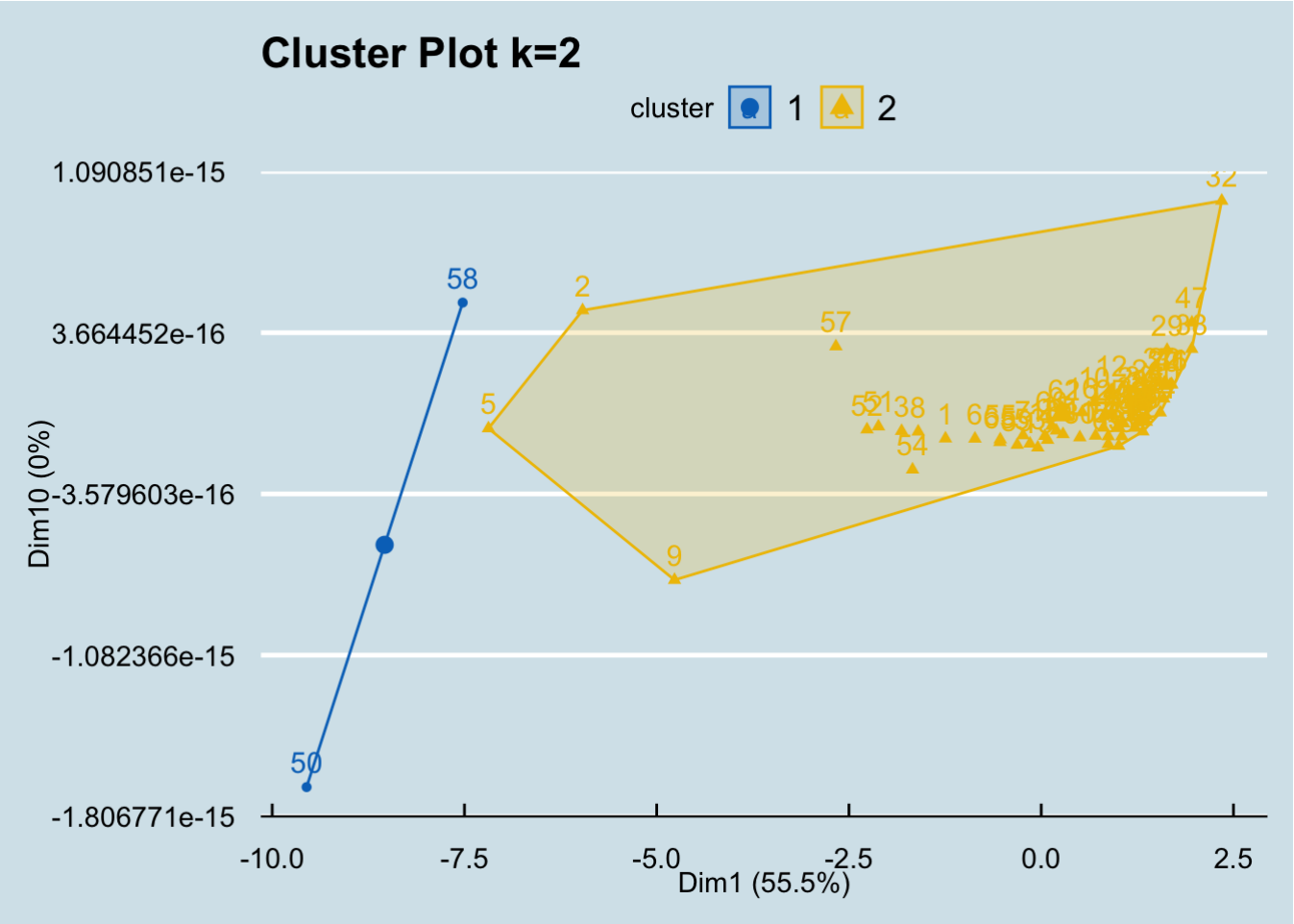
```
k <- 2
(kmeansResult_tf_scaled <- kmeans(DF, k))
```

```
## K-means clustering with 2 clusters of sizes 63, 2
##
## Cluster means:
##   Gross.cash.income Net.farm.income Other.related.income farm.count
## 1          406710.6        94123.37             36022.51   10042.54
## 2         5097631.0       775201.00            258118.50     939.50
##   Gross.cash.income.per.farm Net.farm.income.per.farm
## 1                   854.0952                 241.127
## 2                  5617.5000                 884.000
##   Other.related.income.per.farm  Expenses Expenses.per.farm profit.margin.
## 1                      71.44444  312587.2          613.0159       13.26984
## 2                     288.50000 4322430.0         4734.0000       15.50000
##
## Clustering vector:
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 1.916102e+13 4.624208e+12
##  (between_SS / total_SS =  75.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
(kmeansResult <- kmeans(DF, k))
```

```
## K-means clustering with 2 clusters of sizes 2, 63
##
## Cluster means:
##   Gross.cash.income Net.farm.income Other.related.income farm.count
## 1         5097631.0       775201.00            258118.50     939.50
## 2          406710.6        94123.37             36022.51   10042.54
##   Gross.cash.income.per.farm Net.farm.income.per.farm
## 1                  5617.5000                 884.000
## 2                   854.0952                 241.127
##   Other.related.income.per.farm  Expenses Expenses.per.farm profit.margin.
## 1                     288.50000 4322430.0         4734.0000       15.50000
## 2                      71.44444  312587.2          613.0159       13.26984
##
## Clustering vector:
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [39] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 4.624208e+12 1.916102e+13
##  (between_SS / total_SS =  75.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
(k.2_c <- fviz_cluster(kmeansResult, DF,
             ellipse.type = "convex",
             #ellipse.type = "concave",
             palette= "jco",
             main = "Cluster Plot k=2",
             axes = c(1, 10),
             ggtheme = theme_economist()))
```



Set the theme

```
gg_back_box <- theme(
  panel.background = element_rect(fill = "#d8e4ea"),
  plot.background = element_rect(fill = "#d8e4ea"),
  legend.background = element_rect(fill = "#d8e4ea")
)
```

Make a heatmap

```
mat<-t(D_sim)
heatmaply(as.matrix(D_sim), main = "Heatmap Cosine Similarity k=2",
          heatmap_layers = gg_back_box)
```

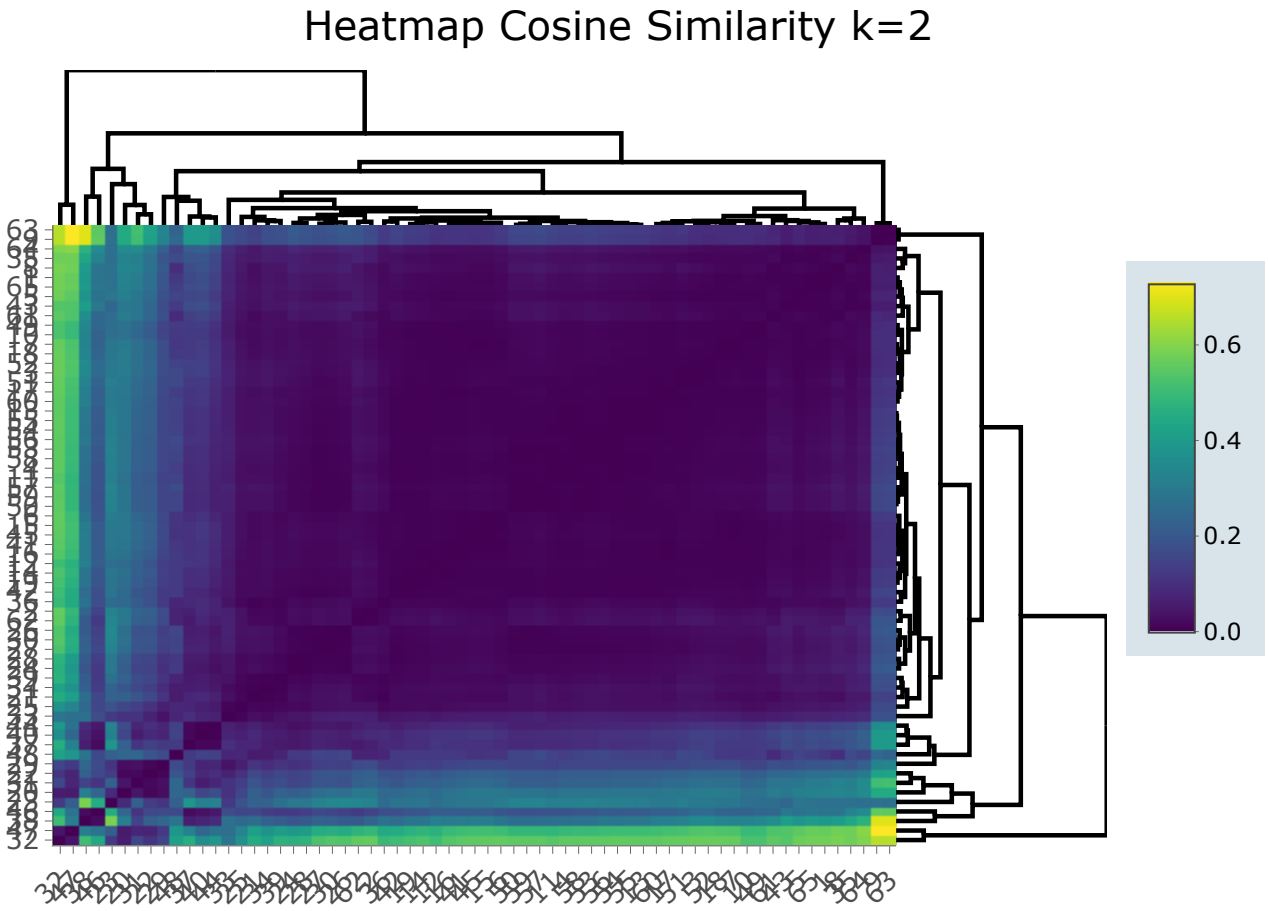
```
## Warning in doTryCatch(return(expr), name, parentenv, handler): unable to load shared
object '/Library/Frameworks/R.framework/Resources/modules//R_X11.so':
##   dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11.so, 6): Library not
loaded: /opt/X11/lib/libSM.6.dylib
##   Referenced from: /Library/Frameworks/R.framework/Versions/4.0/Resources/modules/R_X
11.so
##   Reason: image not found
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



Try k = 3 and make visualizations

```
k <- 3
(kmeansResult_tf_scaled <- kmeans(DF, k))
```

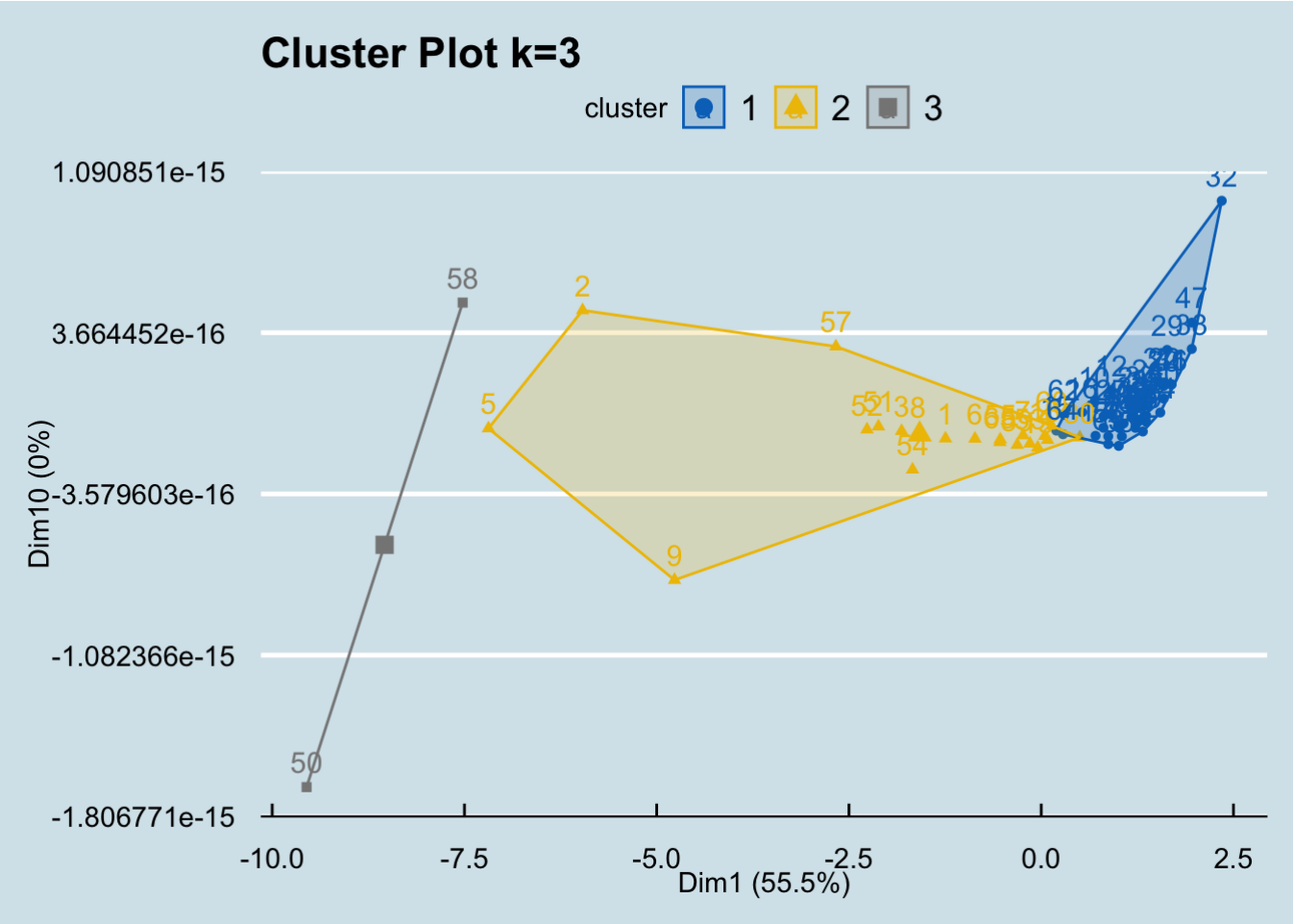
```
## K-means clustering with 3 clusters of sizes 42, 2, 21
##
## Cluster means:
##   Gross.cash.income Net.farm.income Other.related.income farm.count
## 1          149151.3        28167.26             17521.98  14189.381
## 2         5097631.0       775201.00            258118.50    939.500
## 3          921829.1       226035.57             73023.57   1748.857
##   Gross.cash.income.per.farm Net.farm.income.per.farm
## 1                   59.47619                 14.95238
## 2                5617.50000               884.00000
## 3                2443.33333               693.47619
##   Other.related.income.per.farm  Expenses Expenses.per.farm profit.margin.
## 1                        8.5000  120984.1          44.54762       7.714286
## 2                      288.5000 4322430.0        4734.00000      15.500000
## 3                      197.3333  695793.5        1749.95238      24.380952
##
## Clustering vector:
##  [1] 3 3 3 3 3 3 3 3 3 1 1 1 3 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 2 3 3 3 3 3 1 3 2 3 3 1 1 1 1 3
##
## Within cluster sum of squares by cluster:
## [1] 1.800822e+12 4.624208e+12 3.782541e+12
##  (between_SS / total_SS =  89.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
(kmeansResult <- kmeans(DF, k))
```

```
## K-means clustering with 3 clusters of sizes 42, 21, 2
##
## Cluster means:
##   Gross.cash.income Net.farm.income Other.related.income farm.count
## 1          149151.3        28167.26             17521.98  14189.381
## 2          921829.1       226035.57             73023.57   1748.857
## 3         5097631.0       775201.00            258118.50    939.500
##   Gross.cash.income.per.farm Net.farm.income.per.farm
## 1                   59.47619                 14.95238
## 2                2443.33333                693.47619
## 3                5617.50000                884.00000
##   Other.related.income.per.farm  Expenses Expenses.per.farm profit.margin.
## 1                        8.5000  120984.1          44.54762       7.714286
## 2                     197.3333  695793.5        1749.95238      24.380952
## 3                     288.5000 4322430.0        4734.00000      15.500000
##
## Clustering vector:
##  [1] 2 2 2 2 2 2 2 2 2 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 3 2 2 2 2 2 1 2 3 2 2 1 1 1 1 2
##
## Within cluster sum of squares by cluster:
## [1] 1.800822e+12 3.782541e+12 4.624208e+12
##  (between_SS / total_SS =  89.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
(fviz_cluster(kmeansResult, DF,
              ellipse.type = "convex",
              #ellipse.type = "concave",
              palette= "jco",
              axes = c(1, 10),
              title = "Cluster Plot k=3",
              ggtheme = theme_economist()))
```

```
## Warning: argument title is deprecated; please use main instead.
```



##

Make a heatmap

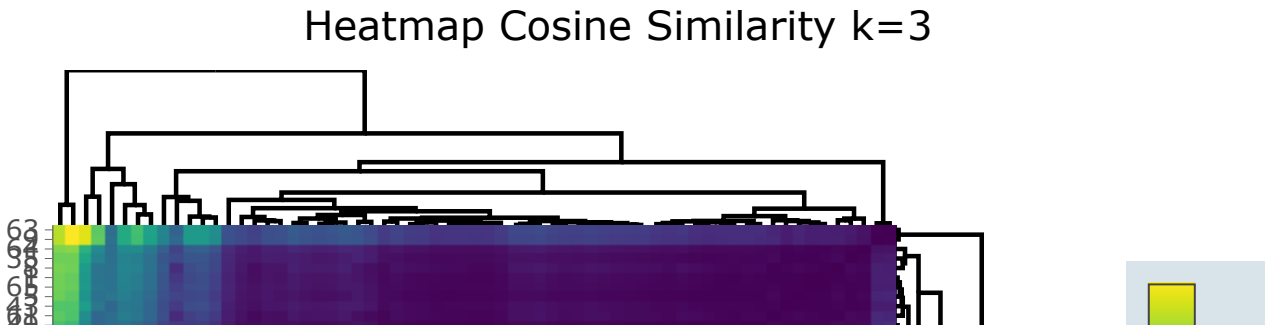
```
mat<-t(D_sim)
heatmaply(as.matrix(D_sim), main = "Heatmap Cosine Similarity k=3",
          heatmap_layers = gg_back_box)
```

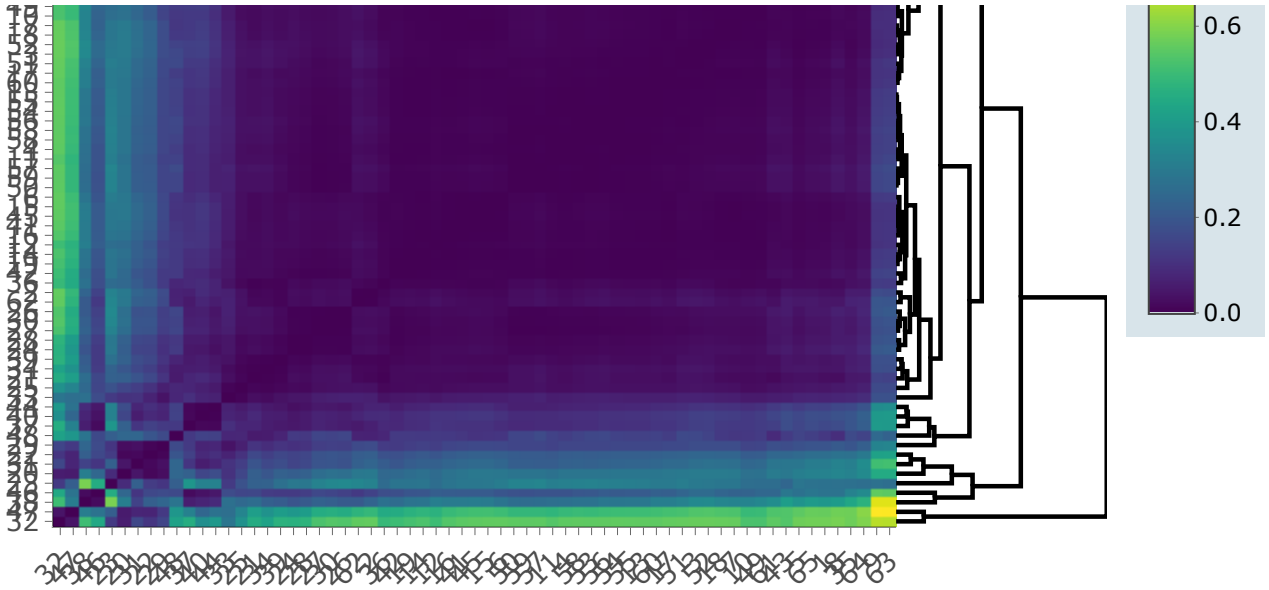
```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```





Try k = 5 and make visualizations

```
k <- 5
(kmeansResult_tf_scaled <- kmeans(DF, k))
```

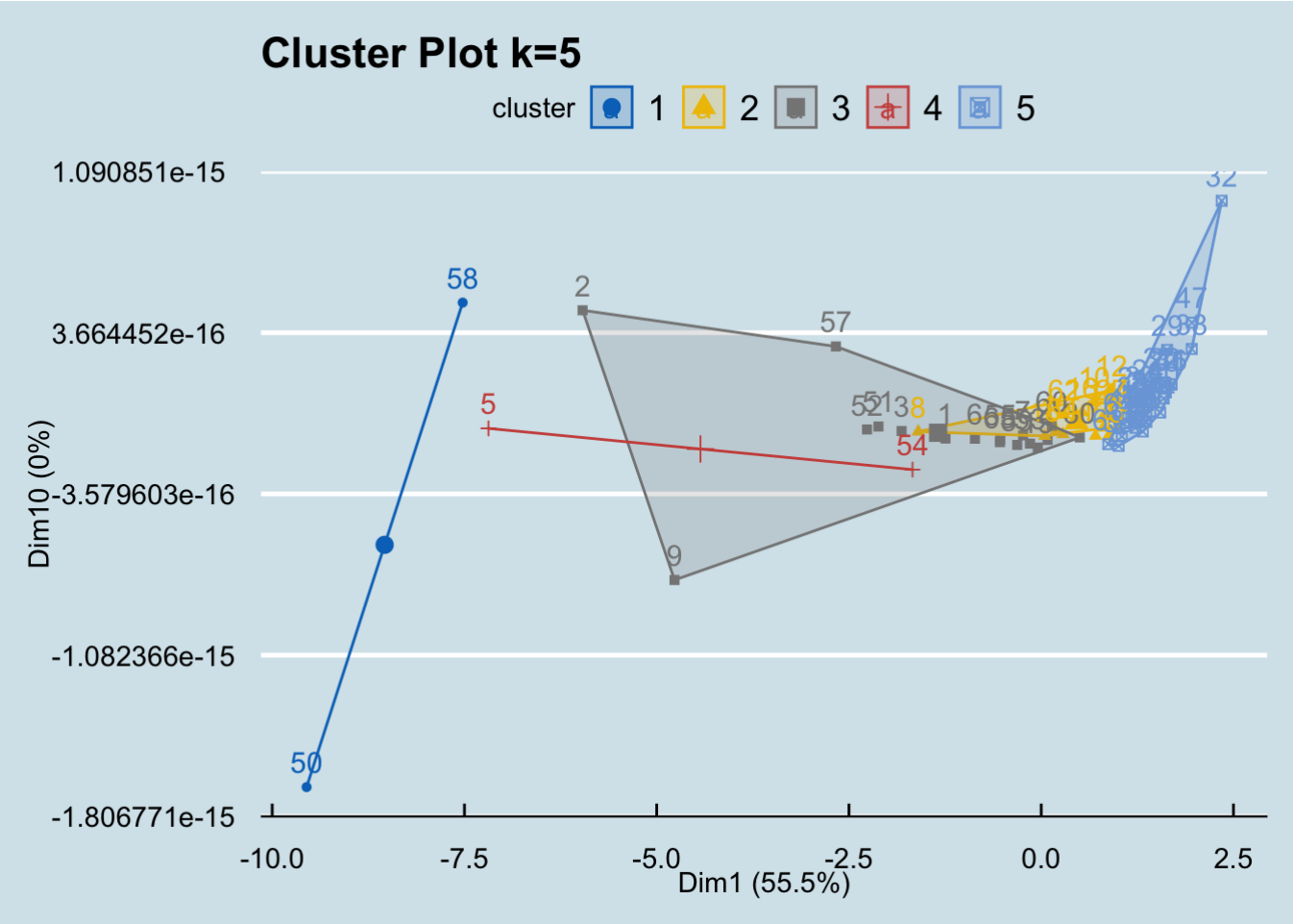
```
## K-means clustering with 5 clusters of sizes 14, 2, 30, 17, 2
##
## Cluster means:
##   Gross.cash.income Net.farm.income Other.related.income farm.count
## 1         407919.86       90548.786            47239.786   9290.786
## 2        5097631.00      775201.000           258118.500    939.500
## 3          57048.47        9461.533             9363.567  15589.700
## 4         868479.59      205832.412            66957.412   1993.529
## 5        1718141.00      439546.000            94439.000    514.000
##   Gross.cash.income.per.farm Net.farm.income.per.farm
## 1                  291.57143                85.14286
## 2                 5617.50000               884.00000
## 3                   16.33333                 5.20000
## 4                 2228.00000               621.88235
## 5                 5680.00000              1635.50000
##   Other.related.income.per.farm   Expenses Expenses.per.farm profit.margin.
## 1                     52.500000  317371.07           206.500      21.642857
## 2                    288.500000 4322430.00          4734.000      15.500000
## 3                      2.133333   47586.93            11.100       2.833333
## 4                    181.000000  662647.18          1606.294      23.411765
## 5                    312.500000 1278595.00          4044.500      25.000000
##
## Clustering vector:
##  [1] 4 4 4 4 5 4 4 1 4 1 1 1 4 1 1 1 1 1 3 3 3 3 3 3 3 1 1 3 3 4 3 3 3 3 3 3 3 3
## [39] 3 3 3 3 3 3 3 3 3 3 3 2 4 4 4 5 4 1 4 2 4 4 3 1 3 1 4
##
## Within cluster sum of squares by cluster:
## [1] 2.972874e+11 4.624208e+12 1.210606e+11 1.018697e+12 2.202615e+11
##  (between_SS / total_SS =  93.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
(kmeansResult <- kmeans(DF, k))
```

```
## K-means clustering with 5 clusters of sizes 2, 14, 17, 2, 30
##
## Cluster means:
##   Gross.cash.income Net.farm.income Other.related.income farm.count
## 1        5097631.00      775201.000           258118.500    939.500
## 2         407919.86       90548.786            47239.786   9290.786
## 3         868479.59      205832.412            66957.412   1993.529
## 4        1718141.00      439546.000            94439.000    514.000
## 5          57048.47        9461.533             9363.567  15589.700
##   Gross.cash.income.per.farm Net.farm.income.per.farm
## 1                 5617.50000               884.00000
## 2                  291.57143                85.14286
## 3                 2228.00000               621.88235
## 4                 5680.00000              1635.50000
## 5                   16.33333                 5.20000
##   Other.related.income.per.farm   Expenses Expenses.per.farm profit.margin.
## 1                     288.500000 4322430.00          4734.000      15.500000
## 2                      52.500000  317371.07           206.500      21.642857
## 3                     181.000000  662647.18          1606.294      23.411765
## 4                     312.500000 1278595.00          4044.500      25.000000
## 5                       2.133333   47586.93            11.100       2.833333
##
## Clustering vector:
##  [1] 3 3 3 3 4 3 3 2 3 2 2 2 3 2 2 2 2 2 5 5 5 5 5 5 5 2 2 5 5 3 5 5 5 5 5 5 5 5
## [39] 5 5 5 5 5 5 5 5 5 5 5 1 3 3 3 4 3 2 3 1 3 3 5 2 5 2 3
##
## Within cluster sum of squares by cluster:
## [1] 4.624208e+12 2.972874e+11 1.018697e+12 2.202615e+11 1.210606e+11
##  (between_SS / total_SS =  93.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
(fviz_cluster(kmeansResult, DF,
              ellipse.type = "convex",
              #ellipse.type = "concave",
              palette= "jco",
              axes = c(1, 10),
              title = "Cluster Plot k=5",
              ggtheme = theme_economist()))
```

```
## Warning: argument title is deprecated; please use main instead.
```



##

Make a heatmap

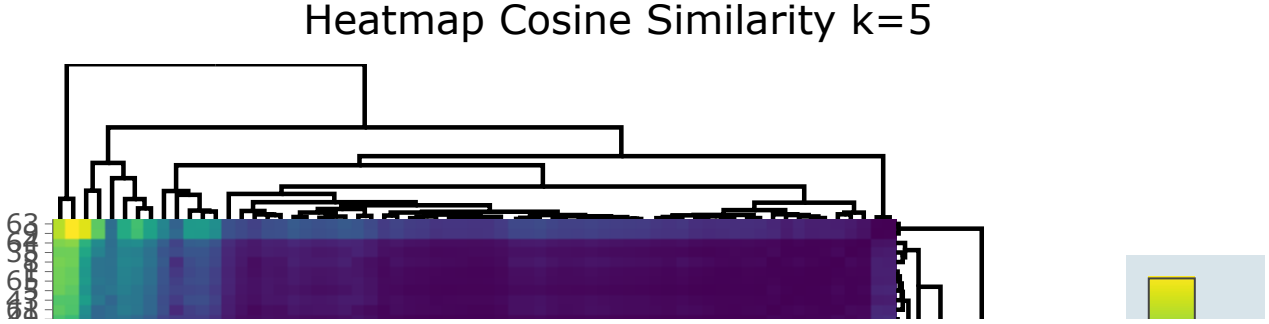
```
mat<-t(D_sim)
heatmaply(as.matrix(D_sim), main = "Heatmap Cosine Similarity k=5",
          heatmap_layers = gg_back_box)
```

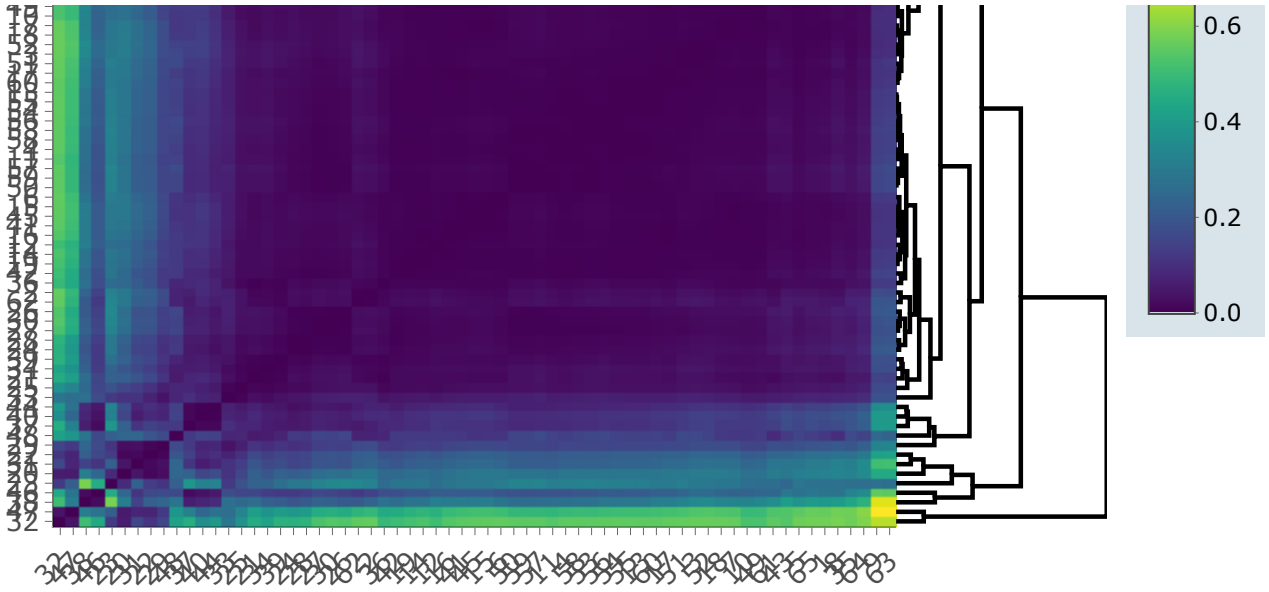
```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```




0.6
0.4
0.2
0.0