# Kmean Cluster Analysis

# Learning Objectives

- Understanding the kmean cluster analysis procedure.
- Understanding the methods used to determine the optimal number of clusters.
- Managing data for the sake of conducting cluster analysis.
- Conducing kmean cluster analysis using R
- Understanding the concept of dietary patterns

# Learning Objectives

- Connecting cluster analysis results with other features of individuals
- Learning to conduct cross-tabulation analysis in R.

# Road Map

- An introduction to cluster analysis and kmean cluster analysis.
- A simple example of kmean clustering.
- Issues to consider in conducting cluster analysis.
- Kmean cluster analysis in practice: the case of dietary patterns
  - Dataset and data management
  - Optimal number of clusters
  - Identifying the clusters
  - Means, frequencies and cross-tabulation

# Machine Learning (ML)

- ML refers to methods and algorithms looking for patterns in a dataset by learning from the data itself.
- The machine, learns from data by conducting the same tasks several times until repeating the task does not improve a pre-defined criteria.
  - (mean squared error in linear regression or percentage of correct predictions in logistic regression)

# Machine Learning (ML)

- There are two types of ML methods
  - Supervised ML: where the researcher defines features (variables) of the model (e.g. random forest and support vector machine)
  - Unsupervised ML:  the researcher lets an algorithm to look for specific pattern(s) without determining what variables could possibly determine the pattern (e.g. cluster analysis and principle component analysis)

# Cluster Analysis (CL)

- CL refers to a series of methods aimed at finding the NATURAL GROUPS (CLUSTERS) in a dataset.
- There are two types of clustering methods
  - Hierarchical: refers to methods used for natural grouping in datasets that are in a top-bottom order (e.g. folders and files in your computer)
    - Hierarchical clustering is time consuming and proper for small datasets.

# Cluster Analysis (CL)

- There are two types of clustering methods
  - Hierarchical: refers to methods used to natural grouping in dataset ordered hierarchically (folders and files in your computer are ordered hierarchically)
  - Partitioning clustering: refers to the methods group the data into clusters that are not overlapping (kmean, kmedian)

# Cluster Analysis (CL)

- There are two types of clustering methods
  - Hierarchical: refers to methods used to natural grouping in dataset ordered hierarchically (folders and files in your computer are ordered hierarchically)
  - Partitioning clustering: refers to the methods group the data into clusters that are not overlapping (kmean, kmedian)
    - These methods can be used for large datasets and large sets of variables

# Cluster Analysis (CL)

- Among the methods, kmean clustering is highly popular.
- Kmean is employed in several subjects such as biology, physics marketing and nutrition studies.
- The popularity of kmean method is due to its ability in finding the patterns in data.
- For instance in **marketing** kmean CL can be used to find the shopping or expenditure patterns. In **nutrition** kmean clustering can be used to find food consumption patterns.

# Cluster Analysis (CL)

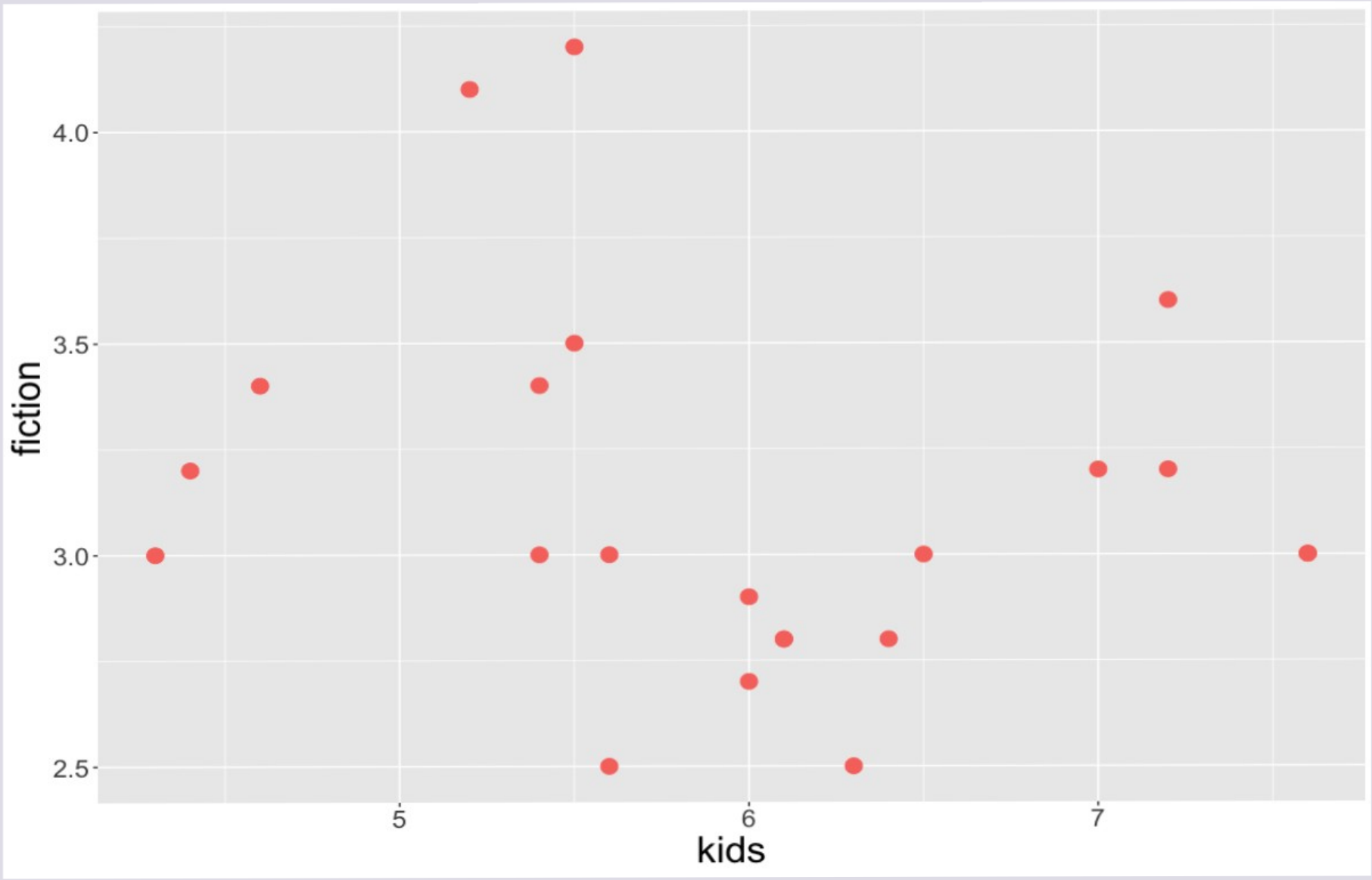
- Lets assume we have a dataset including the expenditures of 22 households on two different types of books: fiction books and kids' books.
- We would like to know if we can distinguish between the households based on their patterns of expenditures on these two types of books.
- We use kmean CL to find the clusters.

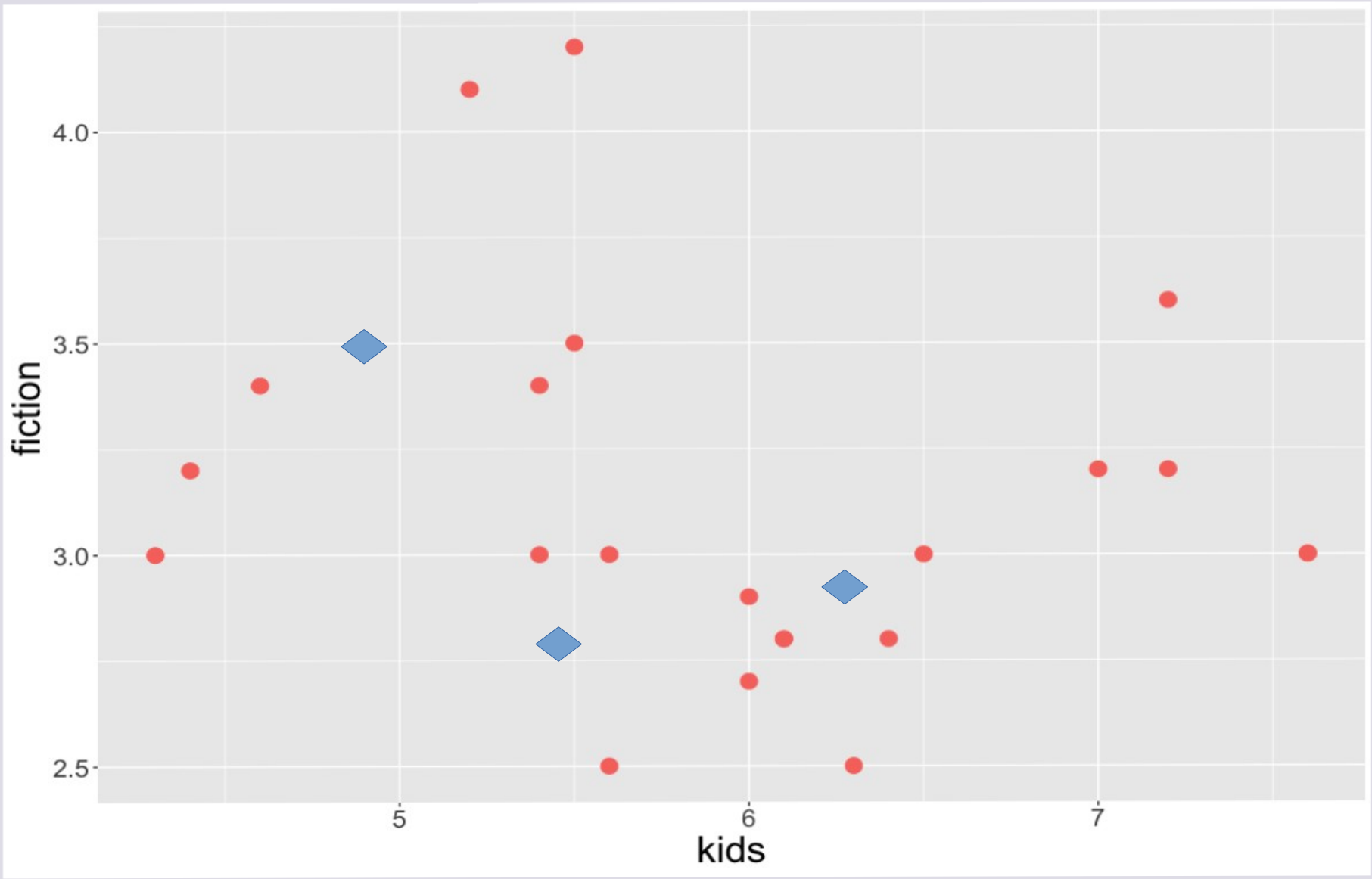
# Cluster Analysis (CL)

- Kmean CL find the natural groupings based on an iterative process.
- We have to tell the kmean clustering what are the variables that it should explores and how many groups we think exist in the dataset.
- For our dataset we tell kmean there are two variables: expenditures on fiction books and expenditures on kids' books.
- We also tell kmean that we think there are three groups of households based on their expenditures on these books.

# Cluster Analysis (CL)

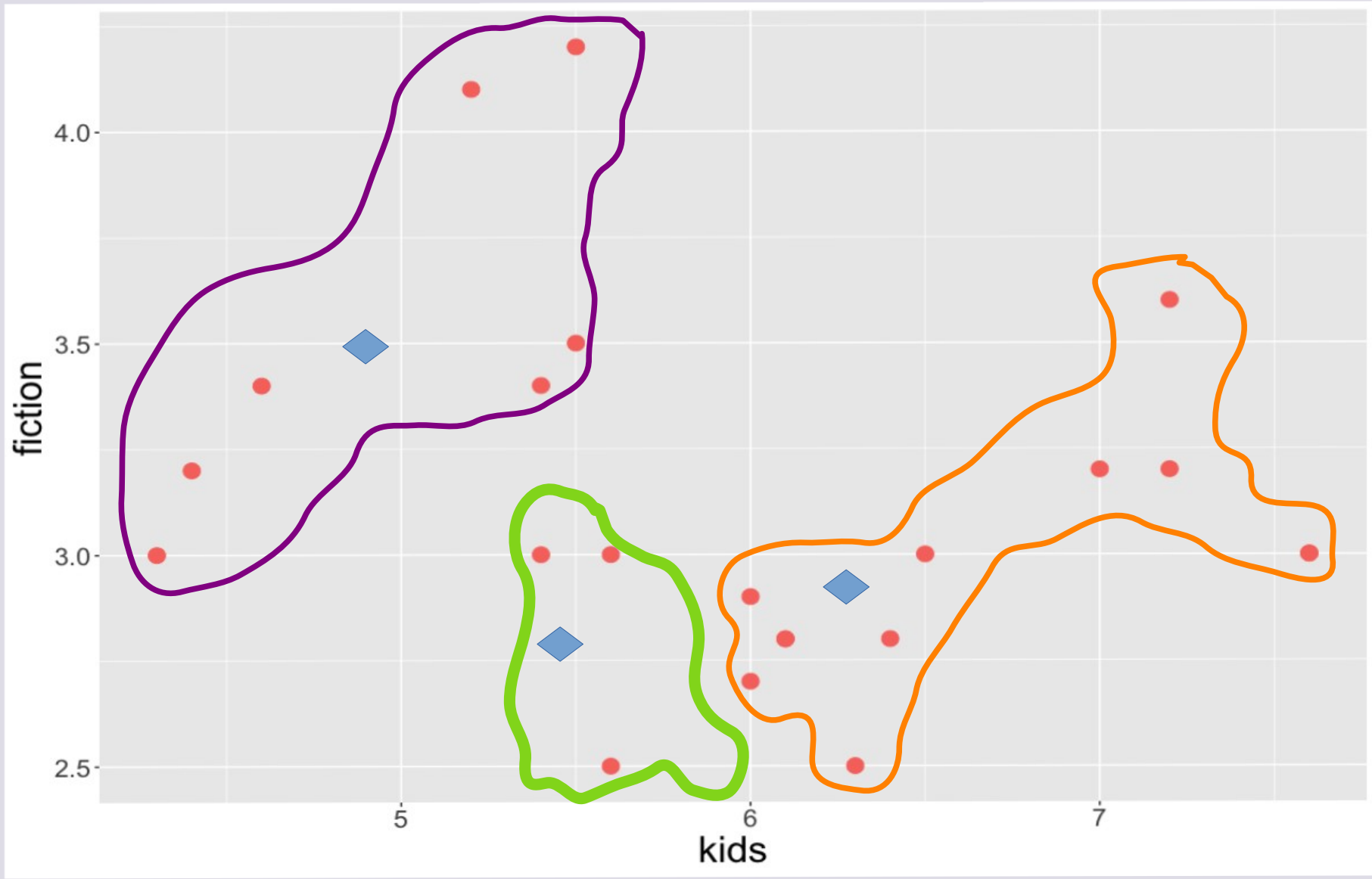
- First: kmean choose 3 random values in the data set (blue diamonds)


fiction
4.0
3.5
3.0
2.5
5
6
7
kids


fiction
4.0
3.5
3.0
2.5
5
6
7
kids

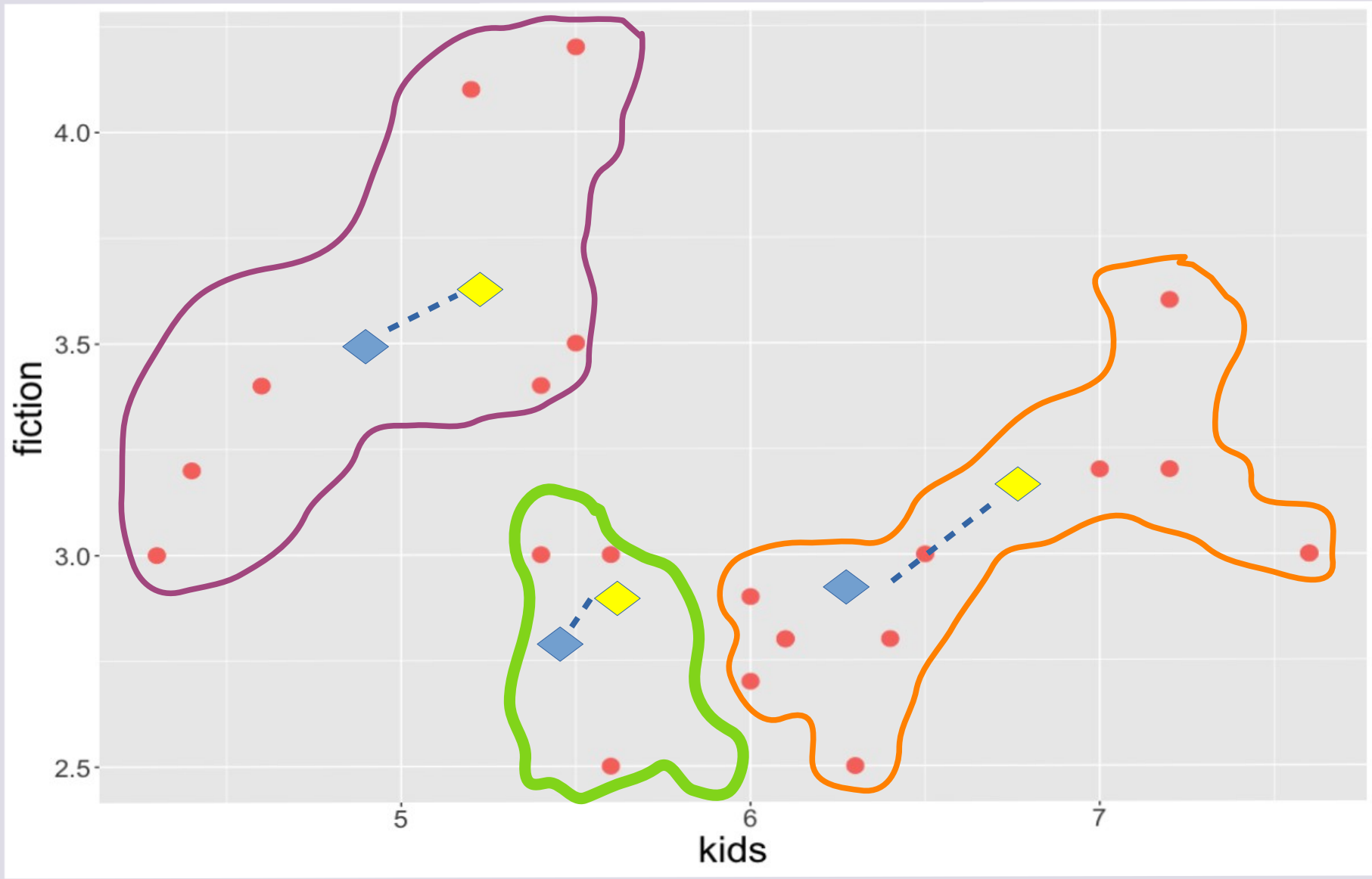
# Cluster Analysis (CL)

- Second: kmean makes three groups of observations based on their distance to the randomly assigned values (blue diamonds).
  - So the closer data points to each random value, will be grouped into one cluster (inside the curves).


4.0
3.5
3.0
2.5
fiction
5
6
7
kids

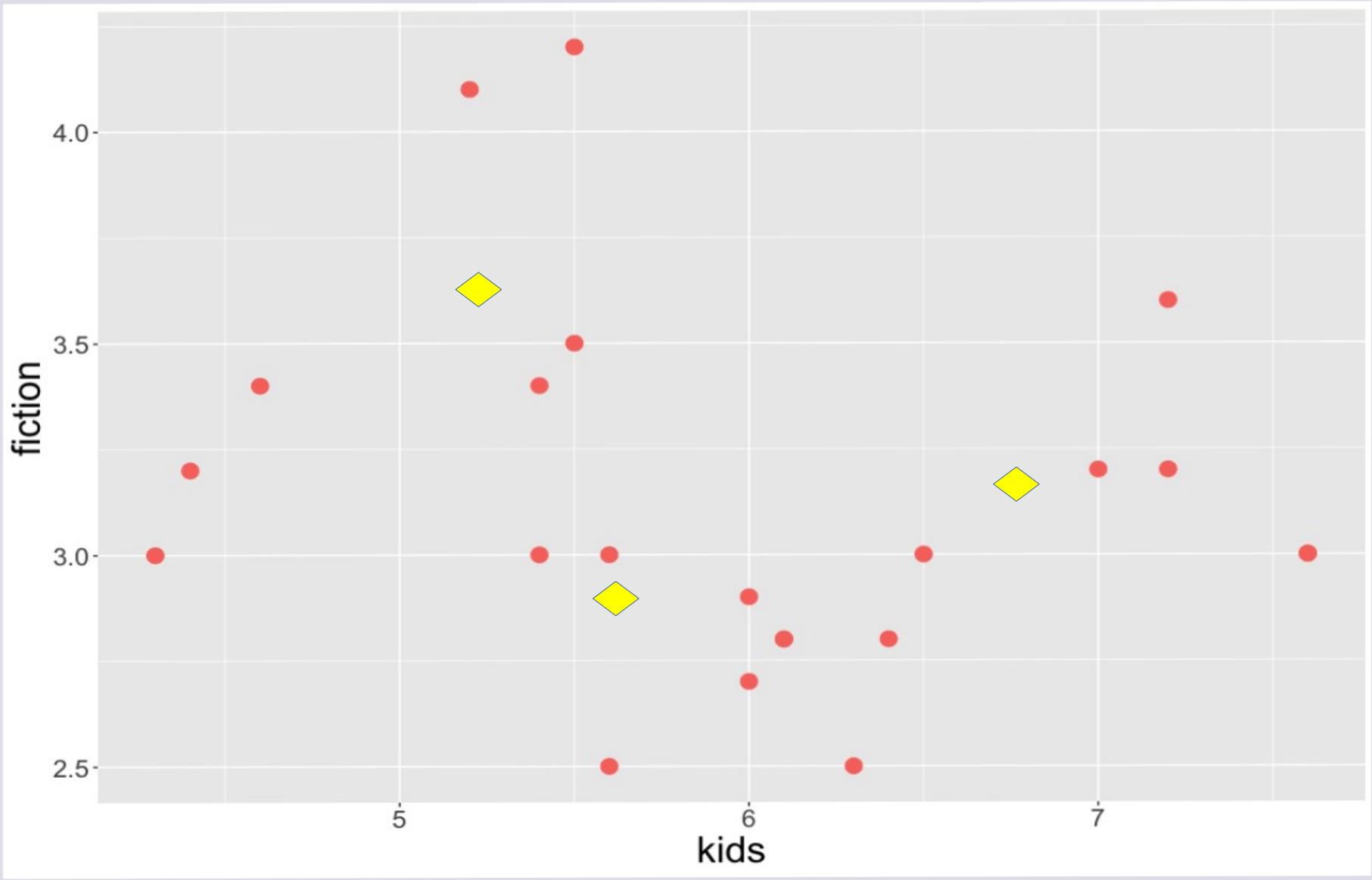
# Cluster Analysis (CL)

- Third: the mean part of kmean CL kicks in. So, the mean values of data points in each group are calculated (yellow diamonds).
- In our case we have now three mean values that are the mean of data points (red circles) in each group.


4.0
3.5
3.0
2.5
fiction
5
6
7
kids

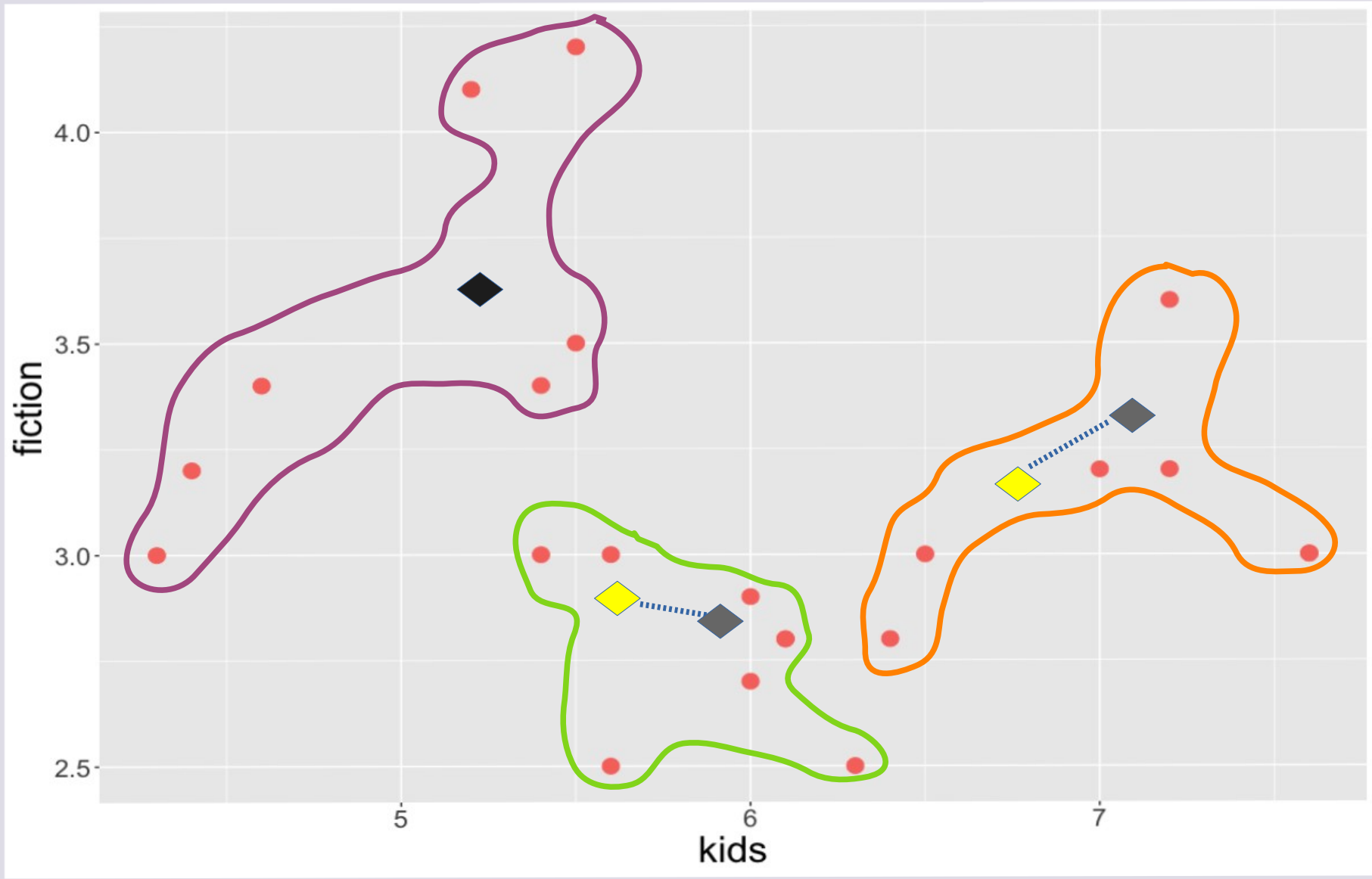
# Cluster Analysis (CL)

- Fourth: three new groups are determined based on their proximity to the mean values (yellow diamonds).
- The new mean values (yellow diamonds) play the same role as the random numbers in the first stage (blue diamonds).


fiction
4.0
3.5
3.0
2.5
5
6
7
kids

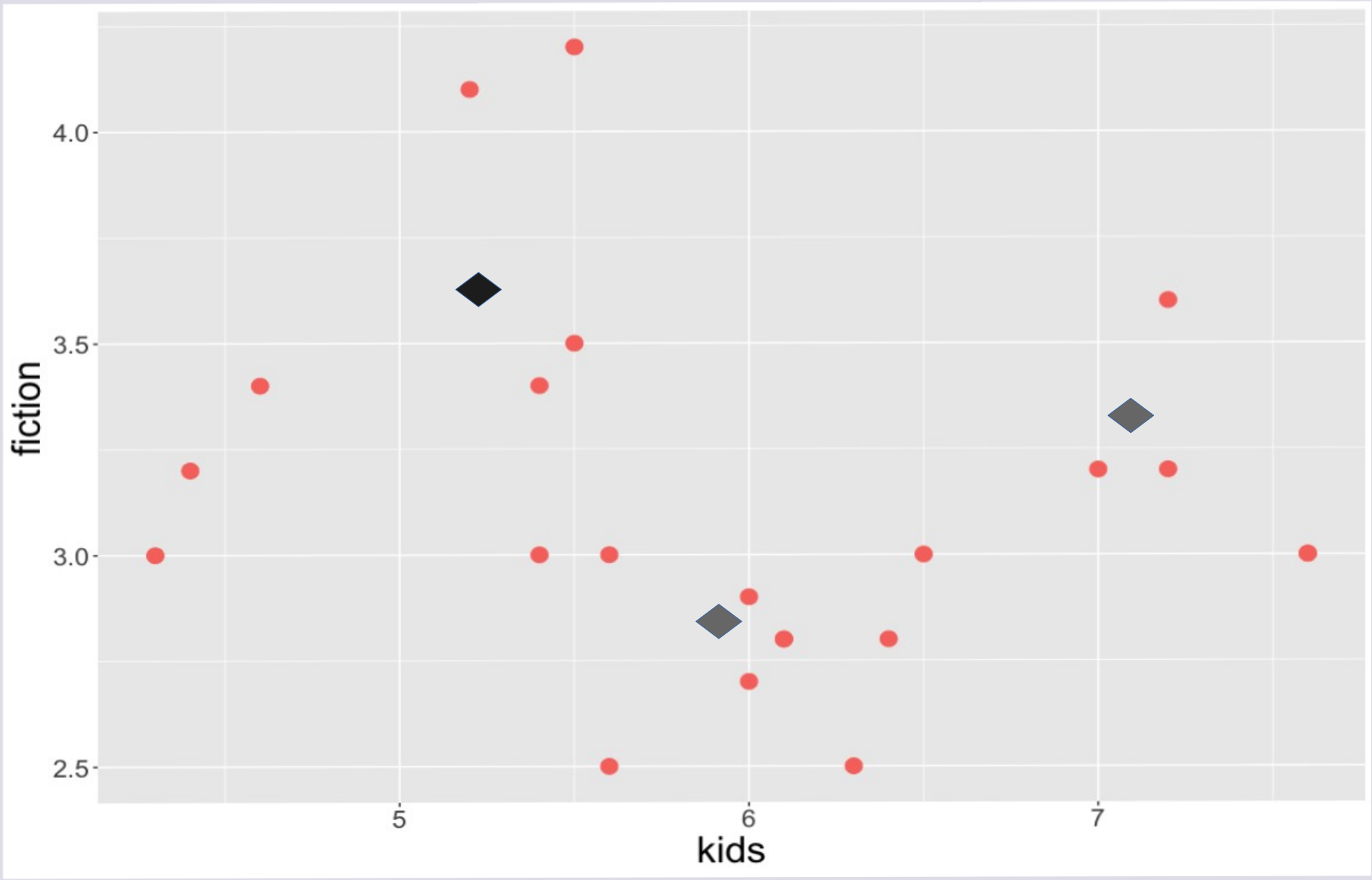
# Cluster Analysis (CL)

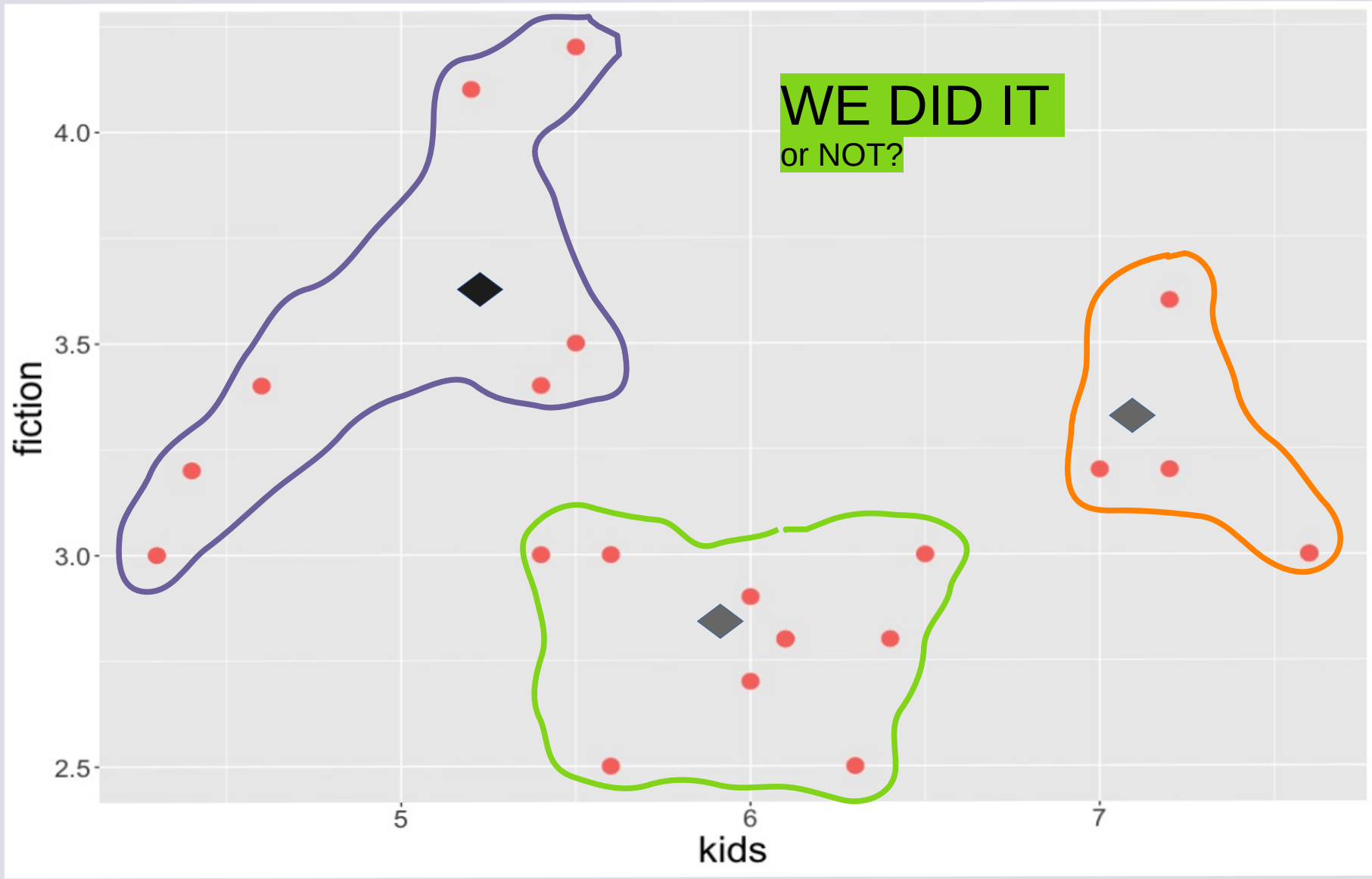
- Fifth: this process is repeated
  - new mean values are calculated.
  - new groups are identified.


fiction
4.0
3.5
3.0
2.5
5
6
7
kids

# Cluster Analysis (CL)

- Six: this process is repeated and repeated again
  - new mean values are calculated.
  - new groups are identified.
- Until: no changes are observed in the mean values
- In this stage the final clusters are identified.


fiction
4.0
3.5
3.0
2.5
5
6
7
kids


WE DID IT
or NOT?
4.0
3.5
3.0
2.5
fiction
5
6
7
kids

# Cluster Analysis (CL)

- There are five important points that should be taken into account:

  1) Kmean CL can only be used to find the natural groups among continuous variables (MEAN!!)

# Cluster Analysis (CL)

2) The units of variables should not be necessary the same
- Example: We can include expenditures on books, number of hours spent on family gathering, number of social connections and so on.
- However, we should standardize all the variables that is we should put different variables on the same scale.

Zx = [observation i of var x] – [mean of var x] / [standard deviation]

# Cluster Analysis (CL)

3) Kmean CL is highly sensitive to the presence of outliers (MEAN!!)

- Usually we should drop the outliers
- Otherwise, the results will be misleading (extra clusters or non-natural groupings)

# Cluster Analysis (CL)

4) we can evaluate kmean CL results (remember natural grouping is the primary task of kmean clustering.

- If our CL performs well, we will be able to find patterns that are consistent with theories or our expectations.

# Cluster Analysis (CL)

5) **The most important point is to determine the optimal number of clusters.**

- Remember in the first step we have to tell kmean that how many random numbers and consequently groups should it work with.
- There are several methods used to determine the optimal number of clusters

# Cluster Analysis (CL)

- **The main idea**: we conduct several cluster analysis where k (that is the number of clusters) increases from 2 to an arbitrary number.
- The maximum number of k is the number of observations where each observation is considered as one cluster.

# The Optimal Number of Clusters

- 1) Scree plot (Elbow Method)
- We need to review a few concepts to understand the method.
  - Total Sum of Squares (TSS)
  - Within Clusters Some of Squares (WCSS)

# The Optimal Number of Clusters

- Total sum of square: $\sum_{i=1}^{n} \left( x_i - \bar{x} \right)^2$
- Each sets of observations have a mean value.
- We calculate the difference between each observation and the mean and square the differences.
- We sum the values and we will get TSS

# The Optimal Number of Clusters

- Lets say we have 5 observation: c(5, 9, 2, 10, 4)
- The average of these 5 observations is equal to 6.
- TSS= 46= $(5-6)^2+(9-6)^2+(2-6)^2+(10-6)^2+(4-6)^2$

# The Optimal Number of Clusters

- WCSS measures the variability of observations within a cluster
  - Each cluster contains a series of observations.
  - Each set of observations has a mean value.
  - Total sum of square for each cluster is WCSS.
  - For two clusters with the same number of observations, the smaller WCSS means the observations are closer together

# The Optimal Number of Clusters

- Sums of WCSS is the primary measure used to determine the optimal number of clusters in elbow method.
- So we conduct several cluster analysis for a same datasets.
- For the book expenditures datasets, we assumed 3 clusters.
- Now lets use R to use elbow method to determine optimal number of clusters.

# The Optimal Number of Clusters

- We need to install and load the following packages:
- library(tidyverse)  # data manipulation
- library(cluster)    # clustering algorithms
- library(factoextra) # clustering algorithms & visualization
- library(NbClust)  #a very good package for determining the optimal number of clusters.

# The Optimal Number of Clusters

We start with k=1 (no cluster) and go up till k=5,  and record WCSSs

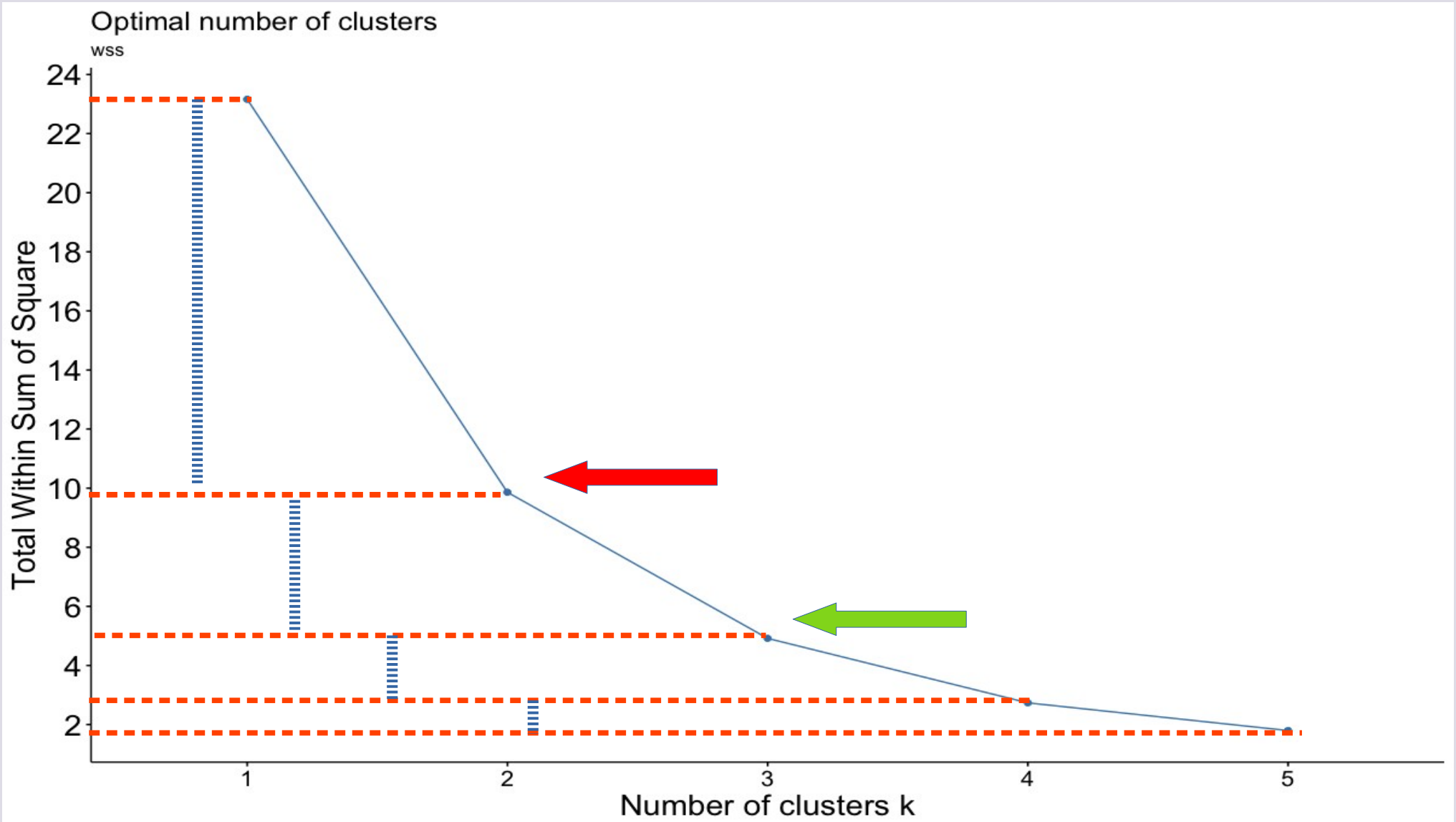
*set.seed(123)* #because kmean starts with a random procedure we use set.seed() to reproduce the results if necessary.

*book_k <- kmeans(book, k, nstart = 25)* # we store the results in **book_k**. Kmean is the function. **book** is the dataset name. **K** is the number of clusters. Finally nstart=25 is related to the initial stage of clustering. We tell the function to start with 25 initial points in the datasets (remember the blue diamonds) and choose the best ones.

# The Optimal Number of Clusters

Now we record the results (WCSS) as k goes from 1 to 5

- K=1: 23.16
- K=2: 5.07 +4.78 = 9.85
- K=3: 3.3 + 0.53 + 1.07= 4.9
- K=4: 1.5 + 0.53 + 0.12 + 0.5 = 2.75
- K=5: 0.23 + 0.05 +0.13 + 0.7 + 0.53 = 1.8


Optimal number of clusters
wss
Total Within Sum of Square
Number of clusters k
24
22
20
18
16
14
12
10
8
6
4
2
1
2
3
4
5

# The Optimal Number of Clusters

- We can see two elbows (kinks) at k=2 and k=3.
- If we are uncertain about the number of clusters we should use other methods.
- Milligan and Cooper (1985) tested 30 methods used to determine the optimal number of clusters.
- They conclude Calinski and Harabasz (CH) method outperforms other methods.

# The Optimal Number of Clusters

- CH method is more straight forward than scree plot.
- The formula for CH method also contains the information about WCSS.
- However, the optimal number of clusters is determined based on the the highest CH index.

# The Optimal Number of Clusters

- The R code for getting CH index is:
  - *ch <- NbClust(book, min.nc=2, max.nc=5,  method = "kmean", index = "ch")*
  - The results are stored in *ch <-*. *NbClust* is the function under a package with the same name. *(book*,  is the name of the dataset. *min.nc=2 , max.nc=5* are two parameters telling NbClust function to report CH index for k=2 to k=5. *method= “kmean”* is the CL method. *index=”CH”)* determine the calculation method that is CH.

# The Optimal Number of Clusters

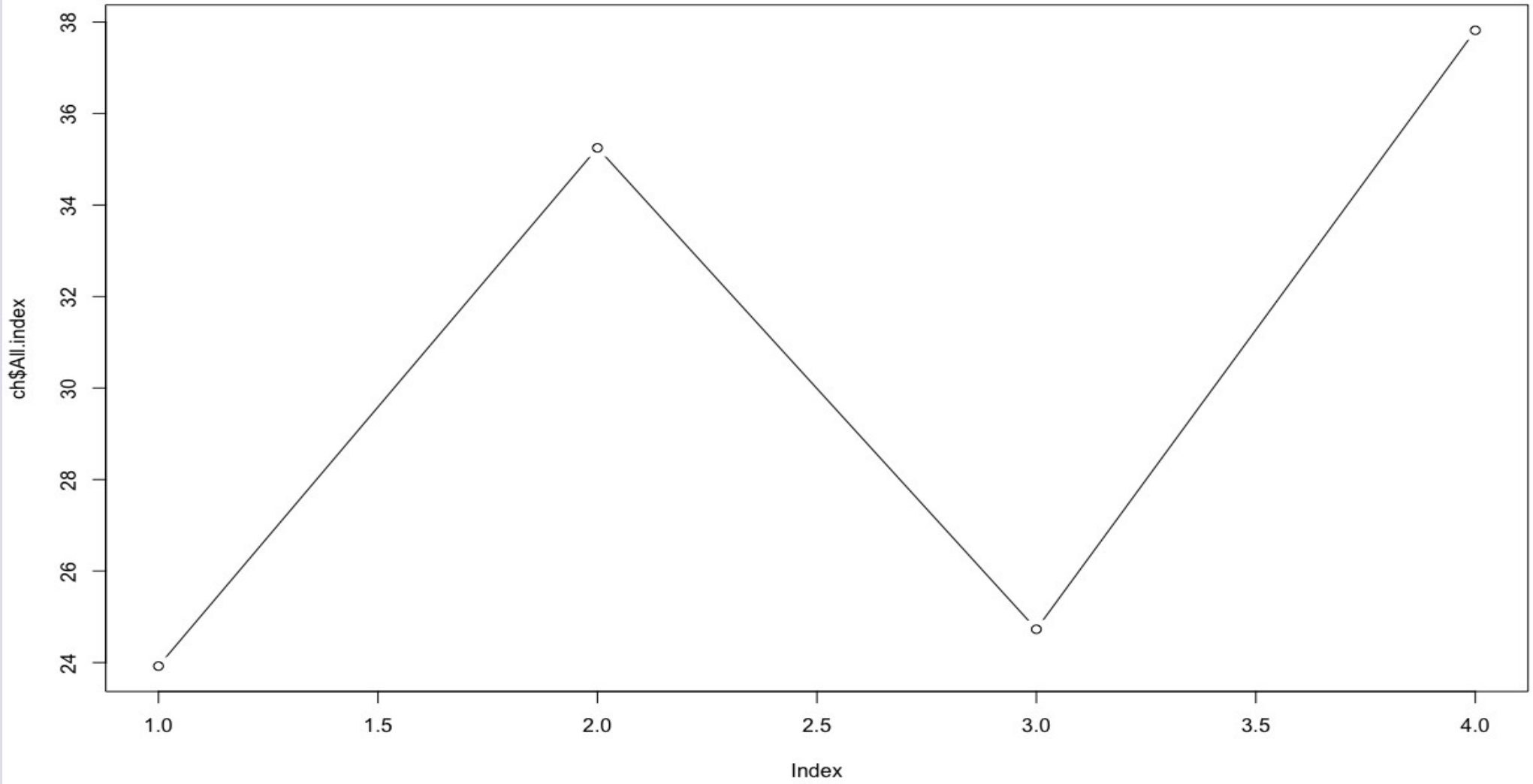
*print(ch)* renders the following results

```
> print(ch)
$All.index
       2        3        4        5
23.9236 35.2484 24.7284 37.8182

$Best.nc
Number_clusters     Value_Index
         5.0000         37.8182

$Best.partition
 [1] 3 4 4 1 3 5 2 3 4 3 3 5 3 1 5 2 1 3 5 3 3 3
```

# The Optimal Number of Clusters

We can also plot the CH index for different numbers of clusters using the following code:

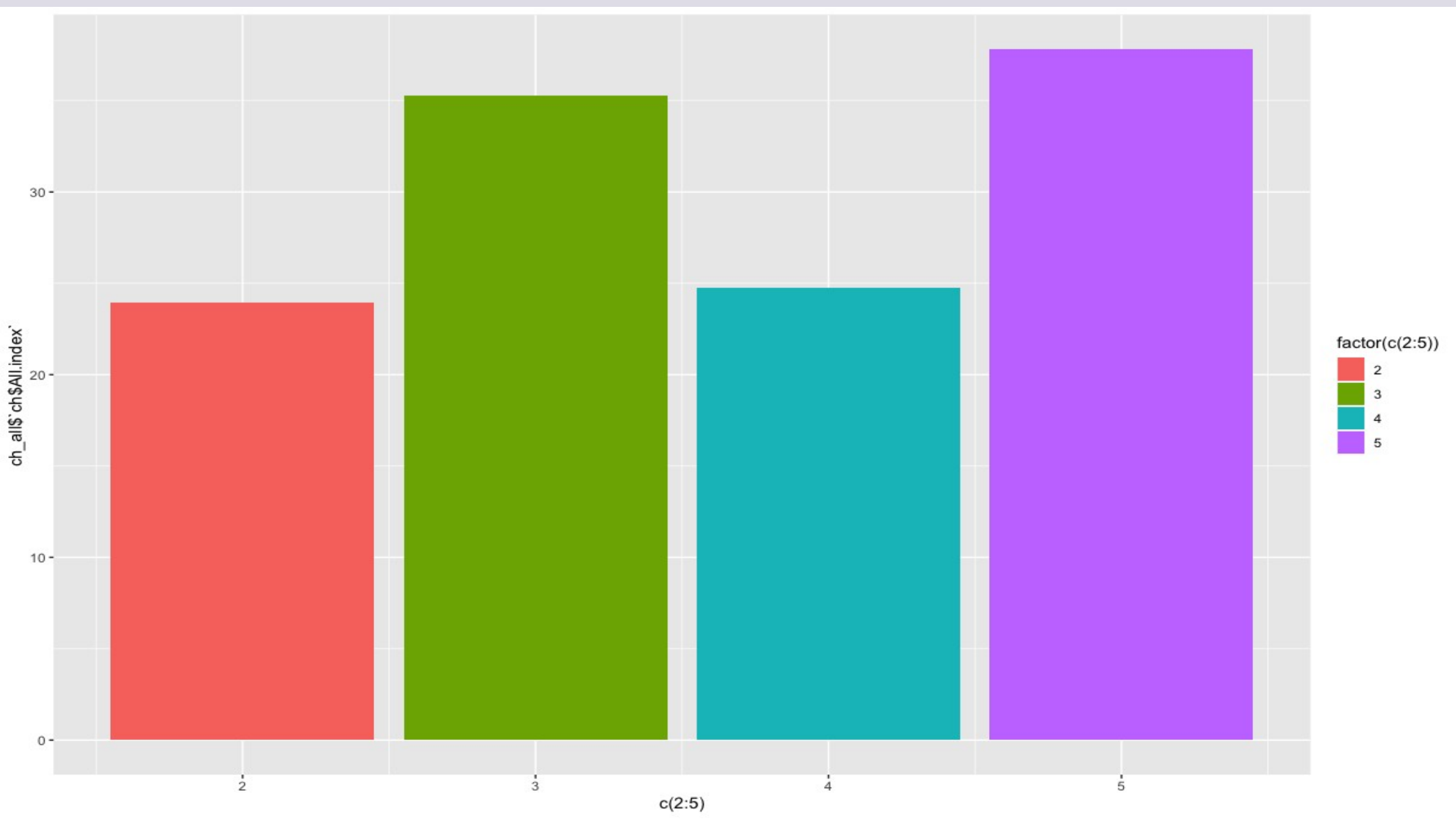
- *plot(ch$All.index, type="b").* So ch stores a series of information one of which is All.index that shows the number of clusters and their corresponding CH index. ch$All.index calls for that component. type="b" means the plot type is both line and point.


ch$All.index
24
26
28
30
32
34
36
38
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Index

# The Optimal Number of Clusters

***Just in Case you want to use ggplot***

```
ch_all <- as.data.frame(ch$All.index)

ggplot(ch_all, aes(c(2:5), ch_all$`ch$All.index`, fill=factor(c(2:5)))+
geom_col()
```


ch_all$`ch$All.index`
30
20
10
0
2
3
4
5
c(2:5)
factor(c(2:5))
2
3
4
5

# The Optimal Number of Clusters

- CH index tells us 5 is the best number of clusters (CH index for k=5 is equal to 37.8)
- However, the CH index for k=3 is equal to 35.2
- CH index for both k=3 and k=5 are close to each other.
- Considering the results of elbow method, we go with 3 clusters because both CH and elbow methods point to k=3.
- If we wanted to follow only one method, CH method is preferred

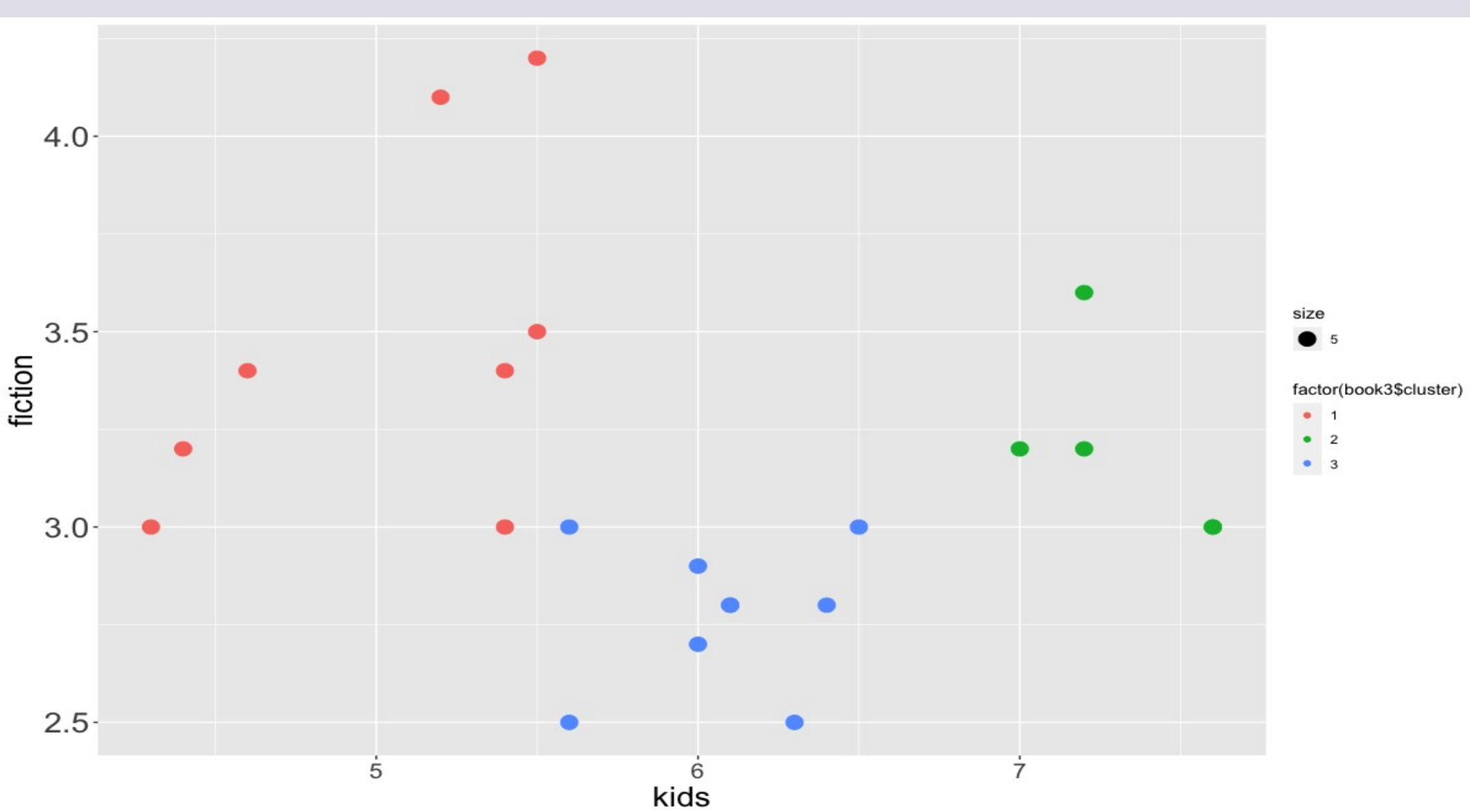
# The Optimal Number of Clusters

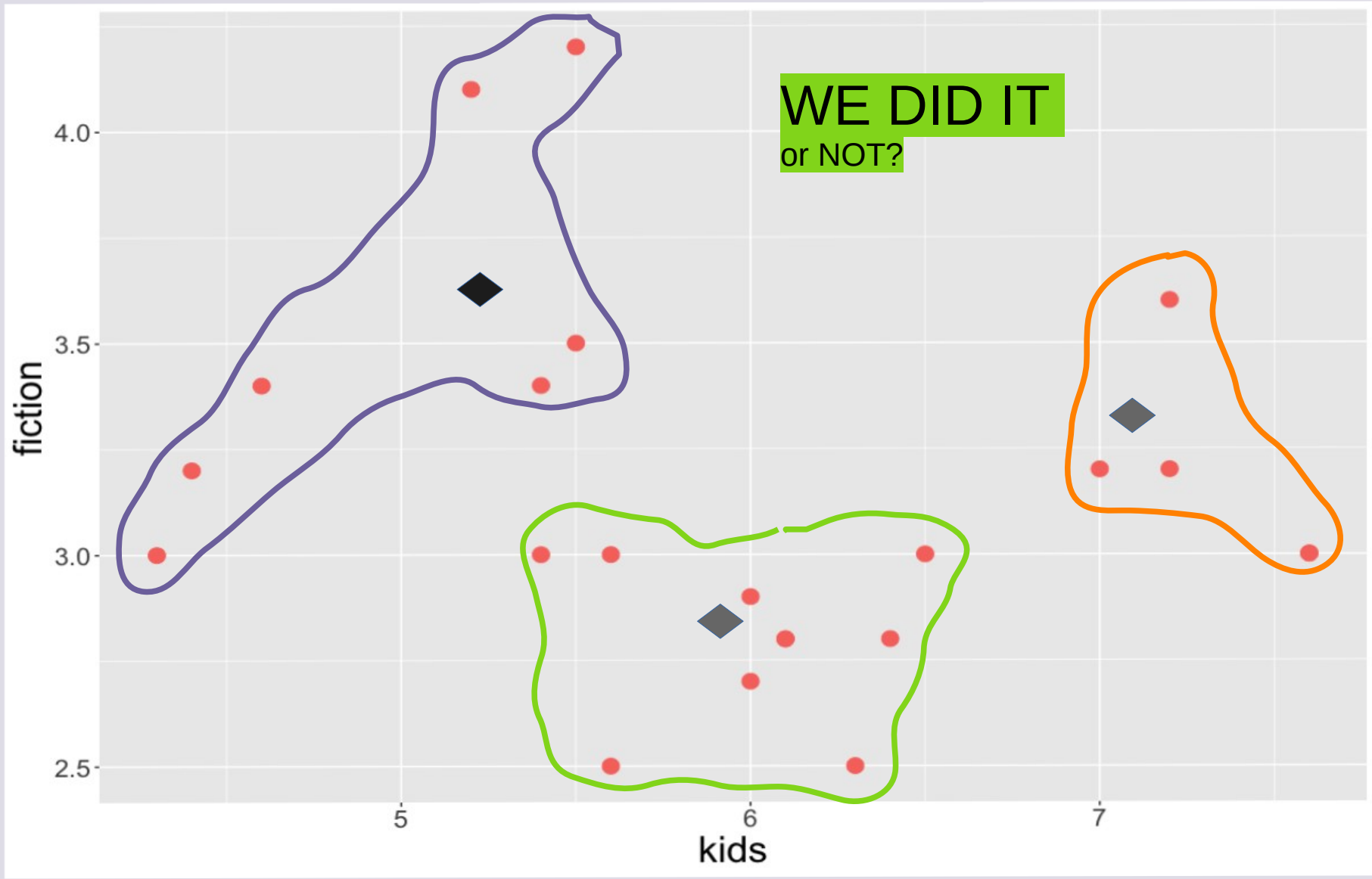
- So the final command is:
  - *book3 <- kmeans(cluster, 3, nstart = 25)*
  - *print(book3)*
  - *Cluster means: (the following table shows the mean of expenditures on both types of books across 3 clusters).*

| *kids* | *fiction* |
|---|---|
| *5.03* | *3.47* |
| *7.32* | *3.2* |
| *6.06* | *2.78* |

# The Optimal Number of Clusters

- We can also plot the CL using the following command:
- *ggplot(book, aes(kids, fiction, colour=factor(book3$cluster))) + geom_point(aes(size=5))*
- We simply ask R to use ggplot to render scatter plot for two variables of kids and fiction from book dataset. However, the colouring should be based on the cluster component of book3 (book3$cluster) where we stored the results of CL with k=3.


4.0
3.5
3.0
2.5
fiction
5
6
7
kids
size
5
factor(book3$cluster)
1
2
3


WE DID IT
or NOT?
4.0
3.5
3.0
2.5
fiction
5
6
7
kids

# CL in practice

- We now are going to extract dietary patterns of Canadian adults.
- You should be familiar with most of the coding in this part.
- We first need to look at the main dataset

# CL in practice

- The dataset includes information about foods intakes, nutrients intakes and socioeconomic status of adults in Canada.
- We use 9 variables indicating the intakes of 9 different food groups in servings for CL.
- The food intakes variables are adjusted for 2000 Kcal of energy intake (that is if an adult eats 6 servings of grains and his/her energy intake is 2500 Kcal, he/she eats 4.8 =(2000*6)/(2500) servings of grains per 2000 Kcal of energy (a solution for outliers)

# CL in practice

- The dataset called "cluster_data" includes all information. However, for CL we make a new data frame that includes the food intakes only.

# CL in practice

- *new <-cluster_data*
- we tell R to store *cluster_data* to *"new"*.
- We use the dataset called "*new*" and make the food intakes dataset from it.
- food <- new %>%
  select(starts_with("adj"))
- This command use "*new*" dataset and and then (%>%)select only those variables whose name *start with "adj"*

# CL in practice

- Looking at “food” dataset, we have three variables including adj_fruits, adj_veg_nopot  and adj_fruitveg. The variable adj_fruitveg is the sum of two other variables, so we tell R to drop the other two variables
- *food <- new %>%*

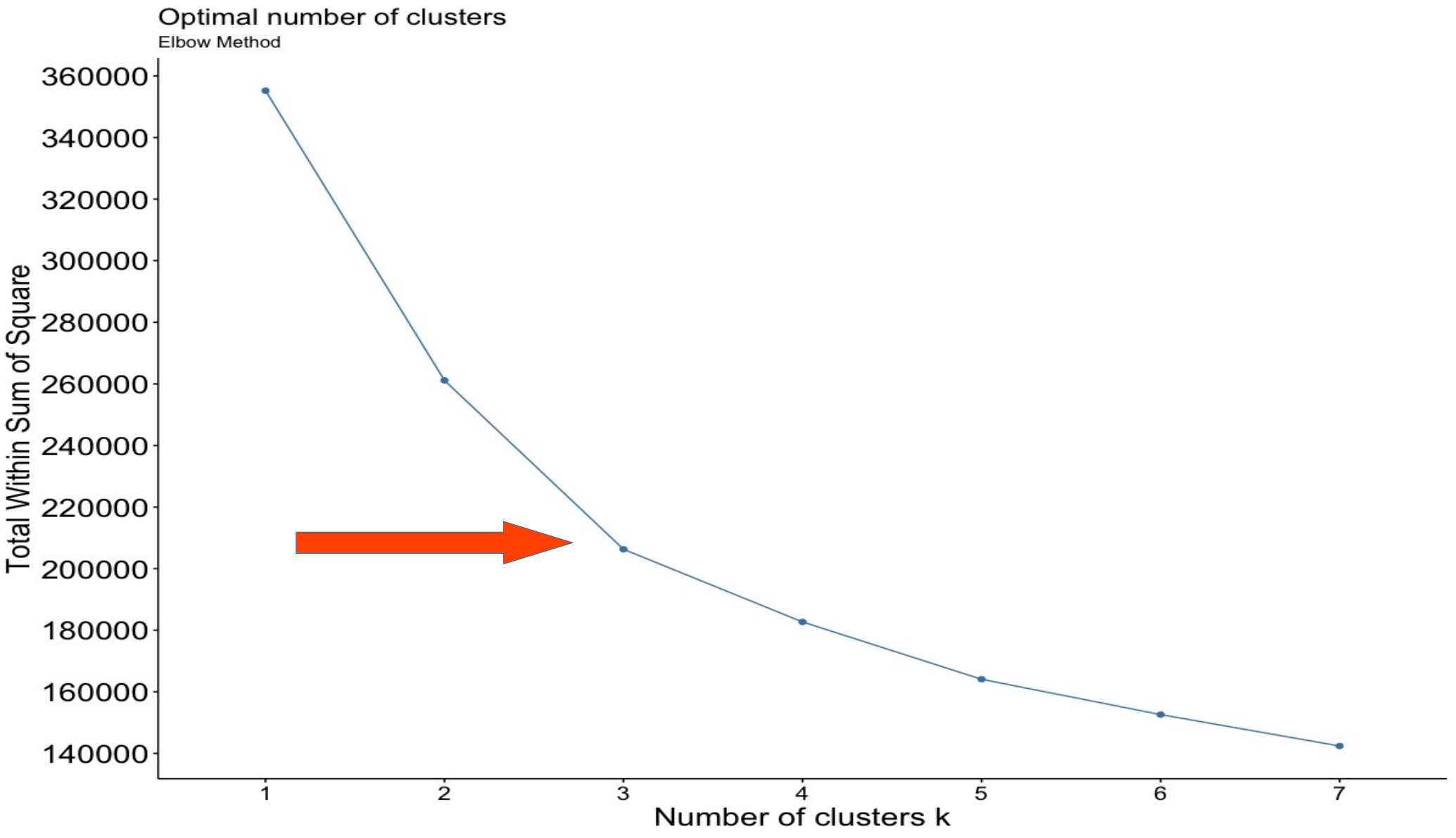
  *select(-adj_fruits, -adj_veg_nopot)*

# CL in practice

- Optimal number of clusters using *fviz_nbclust* function (elbow method)
- *fviz_nbclust(food, kmeans, method = "wss", k.max = 7) +*
  *labs(subtitle = "Elbow Method") +*
  *scale_y_continuous(breaks = scales::pretty_breaks(n = 15)) +*
  *theme(*
  *axis.title.x = element_text(size = 20), axis.text.x = element_text(size = 15), axis.text.y = element_text(size = 20), axis.title.y = element_text(size = 20), plot.title = element_text(size=18))*

# CL in practice

- We use *fviz_nbclust* and tell it to choose *food* dataset, conduct *kmean* CL, and find optimal number of clusters using method= “wss” (within cluster sum of squares).
- The rest of commands were discussed in GVC lecture and are only for better visibility of graph.


Optimal number of clusters
Elbow Method
Total Within Sum of Square
360000
340000
320000
300000
280000
260000
240000
220000
200000
180000
160000
140000
1
2
3
4
5
6
7
Number of clusters k

# CL in practice

- We also going to use CH index to confirm the results of elbow method.
- *ch_ind <- NbClust(food, min.nc=2, max.nc=7, method = "kmean", index = "ch")*
- *print(ch_ind)*
- *plot(ch_ind$All.index)*
- We use *NbClust* function and tell it to use food dataset, with *minimum* number of clusters *=2* and *maximum*=*7*, the CL method is kmean. We also tell the function that we want the "*CH*" index

# CL in practice

- Printing the results we see optimal k=3:
  - $All.index
  - 2 3 4 5 6 7
  - 3913.67 3920 3276.4 3162.07 2882.5 2703.6
  - $Best.nc
  - Number_clusters Value_Index
  - 3.00 3920

# CL in practice

- we choose k=3. We tell R to store kmean CL results in *food_cl.*
- We use *set.seed(#)* in case we want to reproduce the results.
- We tell R to conduct *kmean* CL for dataset of food where *k = 3* and *nstart=40*. Finally we want only to see the main results (next page) by printing the centres only
- *set.seed(1234)*
- *food_cl <- kmeans(food, 3, nstart = 40)*
- *print(food_cl$centers)*

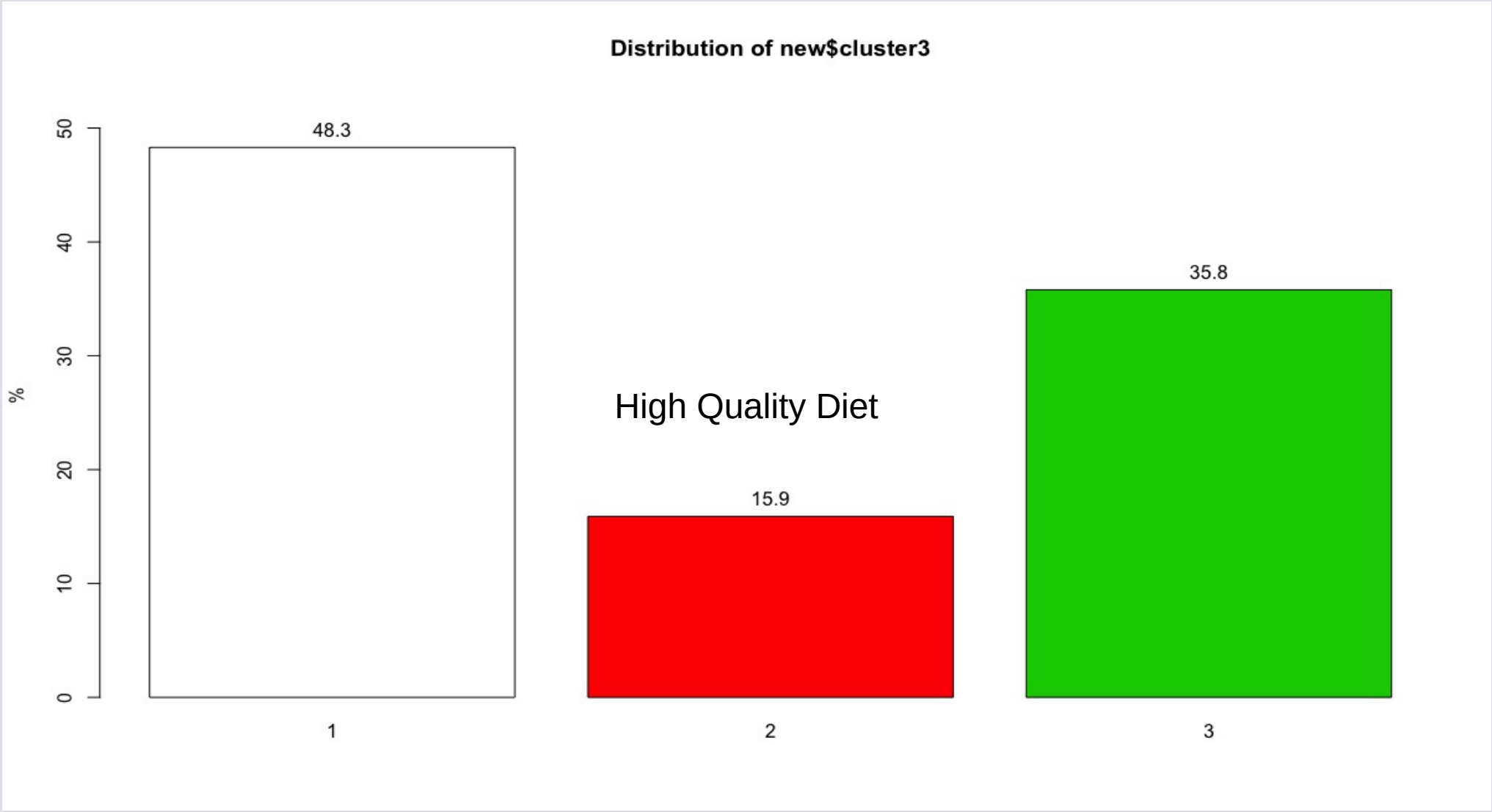
| | Cluster 1<br>Medium Quality | Cluster 2<br>High Quality | Cluster 3<br>Low Quality |
|---|---|---|---|
| Whole Grain | 1.6 | 1.3 | 0.4 |
| Refined Grain | 2.8 | 3.3 | 7.8 |
| Dairy Product | 1.7 | 1.5 | 1.4 |
| Red Meat | 0.7 | 0.6 | 0.6 |
| White Meat | 0.8 | 1.1 | 0.6 |
| Pulses and Nuts | 0.5 | 0.5 | 0.3 |
| Eggs | 0.3 | 0.3 | 0.2 |
| Processed Meat | 0.3 | 0.2 | 0.3 |
| Fruits and Vegetables | 2.7 | 10.3 | 2.5 |

# CL in practice

- Now we can evaluate the CL results by examining the prevalence of few socioeconomic characteristics across clusters
- To make everything a bit easier we add the cluster results to the "*new*" datasets
- so we tell R to make a new variable called *cluster3* in the *"new"* dataset whose values are the values of "*cluster*" variables in *food_cl* where we stored kmean CL results.
- *new$cluster3 <- food_cl$cluster*

# CL in practice

- **Cross Tabulation**
- We use “epiDisplay” package
- The following lines of codes tell R to use function *tab1* to report the distribution of *clusters* in the *“new”* data set. It also prints the results in a graph where the *bar values* are *percent* values
  - *tab1(new$cluster3, bar.values = "percent")*


Distribution of new$cluster3
%
0
10
20
30
40
50
48.3
15.9
35.8
1
2
3
High Quality Diet

# CL in practice

- We also would like to know the prevalence of males, immigrants, and those with university degrees across clusters identified.
- This is called cross tabulation and we use package “descr”.

# CL in practice

```
male <-  crosstab(new$male, new$cluster3,
                  expected = F, prop.r = T, prop.c = F, prop.t = F,
                  prop.chisq = F, chisq = T, missing.include = F,
                   format = "SPSS",
                  dnn = "label", xlab = "Clusters",
                  ylab = "Male", main = "",
                  plot = getOption("descr.plot"))
```

# CL in practice

We tell R use crosstab function, put male on column and cluster3 on rows, reporting expected values is FULSE (F), row percentages is True (T), column percentages is F,  total percentages is F, chi square of proportion is F, chi square value is T, including missing values is F, table format is the same as SPSS, the rest of codes are related to the plot shown in viewer pane.

| new$cluster3 | new$male 0 | 1 | Total |
|---|---|---|---|
| 1 | 2578 | 2667 | 5245 |
| | 49.2% | 50.8% | 48.3% |
| 2 | 1165 | 567 | 1732 |
| | 67.3% | 32.7% | 15.9% |
| 3 | 1946 | 1942 | 3888 |
| | 50.1% | 49.9% | 35.8% |
| Total | 5689 | 5176 | 10865 |

Statistics for All Table Factors

Pearson's Chi-squared test

Chi^2 = 184.172 d.f. = 2 p <2e-16

# CL in practice

- University Degree across Clusters:
- We first make a dummy variable called uni_degree. it takes value of 1 if edu_res4 is equal to 4.
- edu_res4 is a variable includes 4 levels of education where the edu_res4= 1 is high school drop out, =2 is high school diploma = 3 is trade diploma and finally =4 is university degree

# CL in practice

- *new <- new %>%*

  *mutate(uni_degree = as.numeric(edu_res4 == 4))*
- We tell R that dataset to store new variable is “new” (new<-). We also tell R to use the “new” and then (%>%) create (mutate) a new variable called “uni_degree”.
- uni_degree takes value of 1 if edu_res==4

| new$cluster3 | new$uni_degree 0 | 1 | Total |
|---|---|---|---|
| 1 | 3858 | 1359 | 5217 |
| | 74.0% | 26.0% | 48.3% |
| 2 | 1112 | 611 | 1723 |
| | 64.5% | 35.5% | 16.0% |
| 3 | 2946 | 911 | 3857 |
| | 76.4% | 23.6% | 35.7% |
| Total | 7916 | 2881 | 10797 |

Statistics for All Table Factors

Pearson's Chi-squared test

Chi^2 = 87.44402 d.f. = 2 p <2e-16

# CL in practice

- **The mean value of continuous variables over clusters**
- We tell R to  use the “new” dataset and then (%>%) group by cluster3 and then summarise (average) variable fsddekc (daily energy intakes in Kcal).
- *energy <- new %>%*

  *group_by(cluster3) %>%*

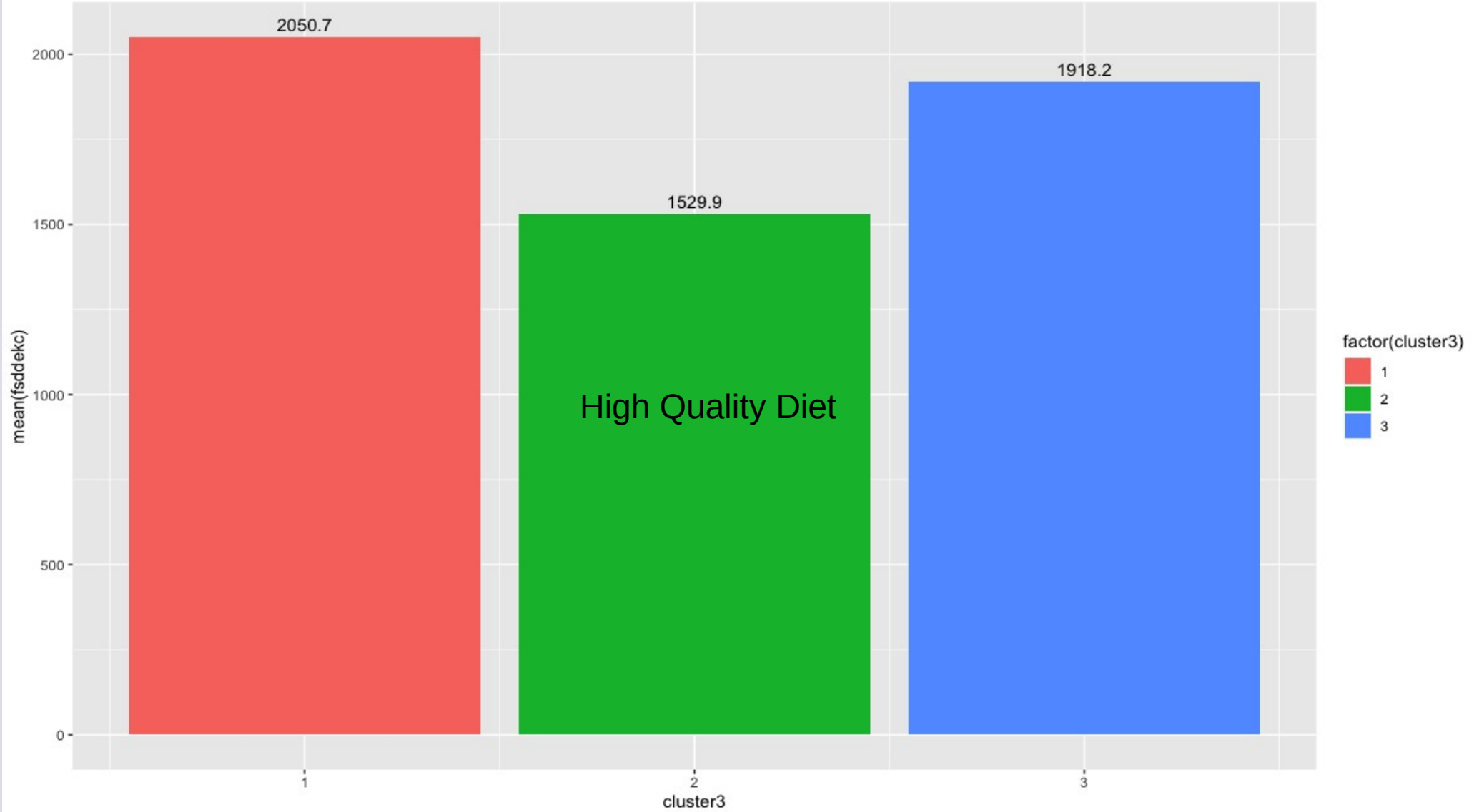
  *summarise(mean(fsddekc))*

# CL in practice

- We can also use ggplot to plot what we did in the previous stage
- So we tell R to use ggplot to make a column graph with the use of dataset energy where x is cluster3 and y is the mean of energy intake and the columns should be filled based on cluster3. The final line add values on the top of columns with geom_text.
- *ggplot(energy, aes(x=cluster3, y=`mean(fsddekc)`, fill=factor(cluster3)))+*

  *geom_col() +*

  *geom_text(aes(label = round(`mean(fsddekc)`,digits = 1), vjust = -0.5))*


2050.7
1529.9
1918.2
High Quality Diet
2000
1500
1000
500
0
mean(fsddekc)
1
2
3
cluster3
factor(cluster3)
1
2
3

# The End!!!