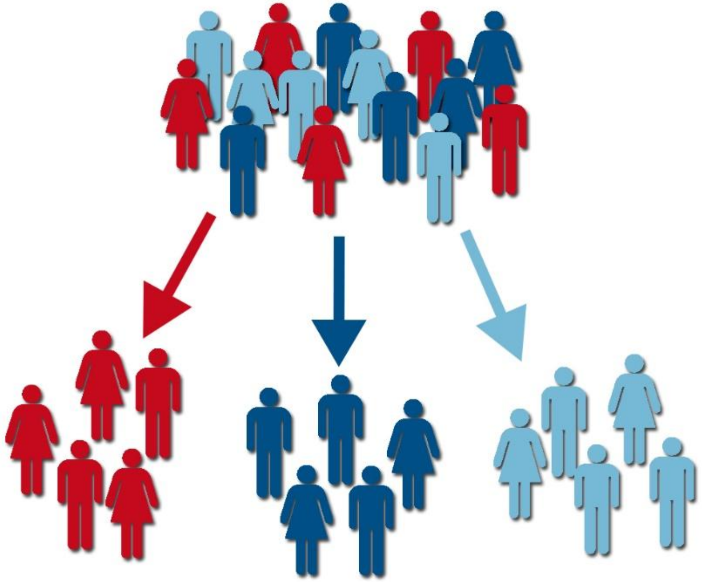
RFM Segmentation

Introduction

What is customer segmentation?

- Customer segmentation is the process of organizing the customer base in small groups that share similar characteristics



Customers can be classified according to different criteria:

- **Transactional behavior**
- Interest categories
- Products in common

This lecture

Basic R functions / dplyr

What is R?

1.	**Statistical programming language created by NZ Researchers Ross Ihaka and Robert Gentleman in 1991.**
2.	Derived from the S language, developed in the 1950's in Bell Labs.
3.	Free (both as in "free beer" and "free person") statistical package.

he

ser

interface to program in R. It's

recom ded to install both (R first

then Rstudio)

Pros and cons

PROS	CONS
Simple to use	R is not so good for intensive computations (all is done in-memory)
Extensive package library	Packages not always play well with each other… be ware!
Very good graphic libraries	

Coding time!

See 000_r_dplyr.R

K-means

What is k-means?

- K-means is a clustering algorithm.
- Clustering is the process or organizing data in groups that share certain similarities.

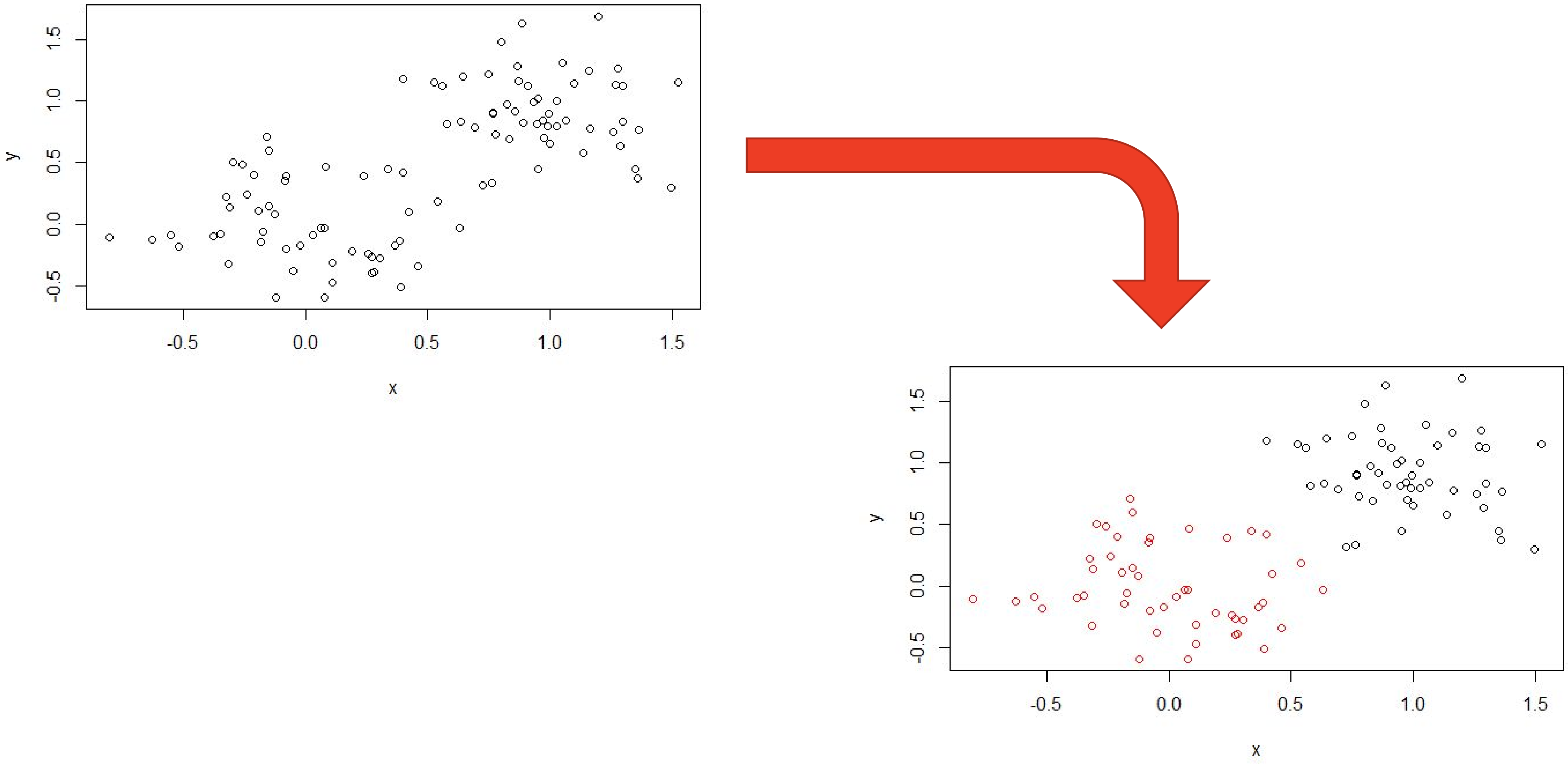
What does it do?

1. Choose **k** points from a given data set, randomly

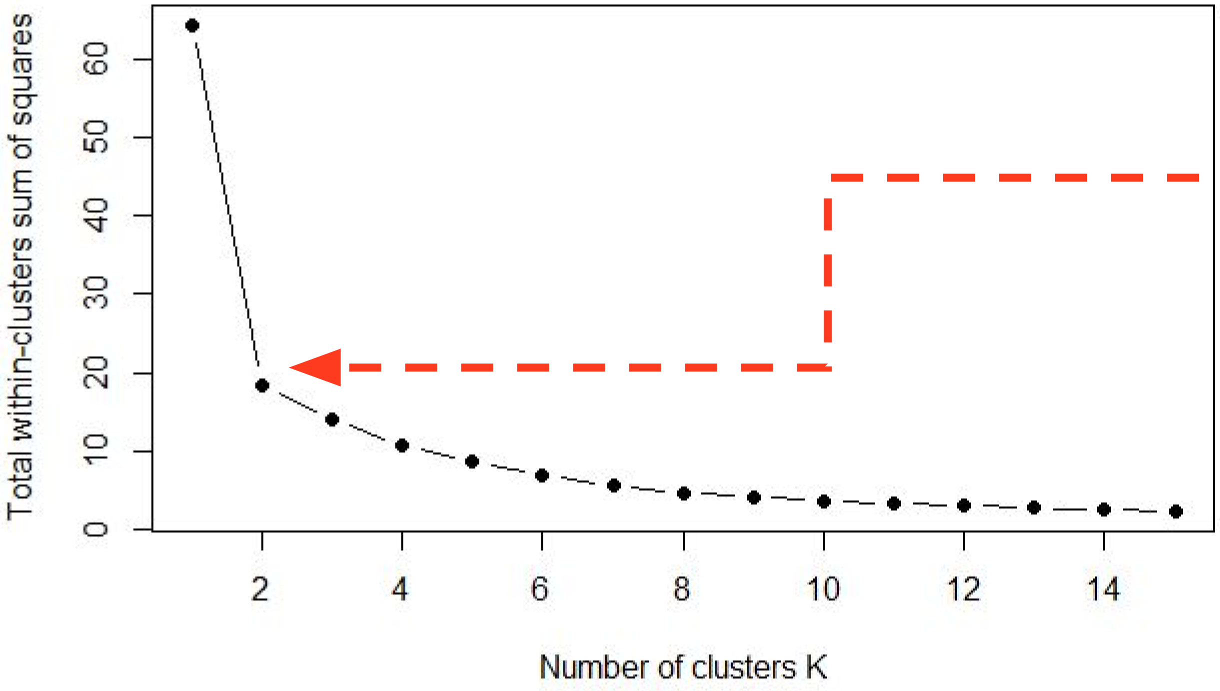
2. For every point in the data set, calculate the **similarity** to all others, and assign each point to its closes center. The groups thus formed become the candidates for clusters.

3. For every new cluster, take the average of this points as a new "center" of the cluster and repeat step 2 until the **total within-cluster square sum** stops decreasing.

Example



How many clusters are needed?



There are many methods to determine the number of clusters. The **elbow method** goes as follows:

- Run k-means for different values of k
- Store the total within cluster sum of squares
- Create the plot on the left, and visually find the inflection point or "elbow" of the curve.

Workflow for RFM Segmentation

Steps

1. Create RFM summary
2. Binning / discretizing variables
3. Clustering
4. Explain cluster results

Basic customer summary:
- **Recency** is the number of days since last purchase.
- **Frequency** is the number of purchases.
- **Monetary** is the total money spent.

This step is important to
- **Put all quantities in scale**
- **Do segmentation with business meaning:**
  - Big business difference between customers who spend 10 Euro vs those who spend 0; not so important between those spending 1000 Euro and 1010 Euro.

Step 0: Load transaction data

```
library(dplyr)
library(stringr)
df <- read.csv("./data/out.c-
pa.orders_refine.csv",
         encoding = "UTF-8")
head(df)

df$DATE_UPDATED <- as.Date(df$DATE_UPDATED)

#Assign a reference date for the analysis.
# This should be after the last date from the
# transaction history.

ref_date <- as.Date("2016-05-18")
```

```
> head(df)
  ID_ORDER         DATE_CREATED DATE_UPDATED                      CUSTOMER_ID           PRODUCT_NAME
1  1307136 2011-02-15 13:07:26   2016-05-08 ff46fb5554fc228253f0d2016bf1d88e Modelovací kostice PME 4
2  1307136 2011-02-15 13:07:26   2016-05-08 ff46fb5554fc228253f0d2016bf1d88e                Sácek PVC
3  1307136 2011-02-15 13:07:26   2016-05-08 ff46fb5554fc228253f0d2016bf1d88e         Smartflex velvet
4  1531137 2011-02-15 15:31:14   2016-05-08 9b1936c918c0968de67c3b6abecc997c            Gelové barvy
5  1531137 2011-02-15 15:31:14   2016-05-08 9b1936c918c0968de67c3b6abecc997c                 Válecek
6  1531137 2011-02-15 15:31:14   2016-05-08 9b1936c918c0968de67c3b6abecc997c     Žehlicka na marcipán
  PRODUCT_COUNT PRODUCT_PRICE
1             1           120
2            10             4
3             1           219
4             1            62
5             1           115
6             1           142
>
```

Step 1: RFM Summary

```
################################################
##      Step 1: RFM Summary              ##
################################################


rfm <- df %>%
    group_by(CUSTOMER_ID) %>%
    summarise(
    Recency=as.integer(difftime(ref_date,
    max(DATE_UPDATED), units = "days")),
Frequency = n(),
    Monetary=sum(PRODUCT_COUNT*PRODUCT_PRICE))
```

```
> head(rfm)
# A tibble: 6 x 4
                       CUSTOMER_ID Recency Frequency Monetary
                             <chr>   <int>     <int>    <dbl>
1 0001b1c6550ecbef06fbe97868c7abf0     148         8      681
2 00047b4e7881febc84760d77b94c89d2     106         5     1188
3 00092d115e79e24856bfdf1f474055fe     148         7      732
4 000b440c05bf65c64f21445abd435845      10         9      778
5 000b935b42dfb69c8cfbdb6c056a25e0     148         3      199
6 00103cbe7f765d488ec60dec86d4016e     148         1      140
>
```

Step 1: RFM Summary

```
#############################################
## Step 2: Binning
#############################################

rfm$rec_bin <- sapply(rfm$Recency, function(x){ifelse(x<50,1,ifelse(x<100,2,3))})
rfm$freq_bin <- sapply(rfm$Frequency, function(x){ifelse(x<1,1,ifelse(x<5,2,3))})
rfm$mon_bin <- sapply(rfm$Monetary, function(x){ifelse(x<300,1,ifelse(x<1000,2,3))})

# Get rid of NAs -- normally you would have
# to find out the cause of NAs from data

rfm <- na.omit(rfm)

data <- rfm %>% select(rec_bin, freq_bin, mon_bin)
```

Values on each segment obtained:
- Visual inspection
- Using quantiles

You can divide in more than 3 segments

Step 3: Clustering

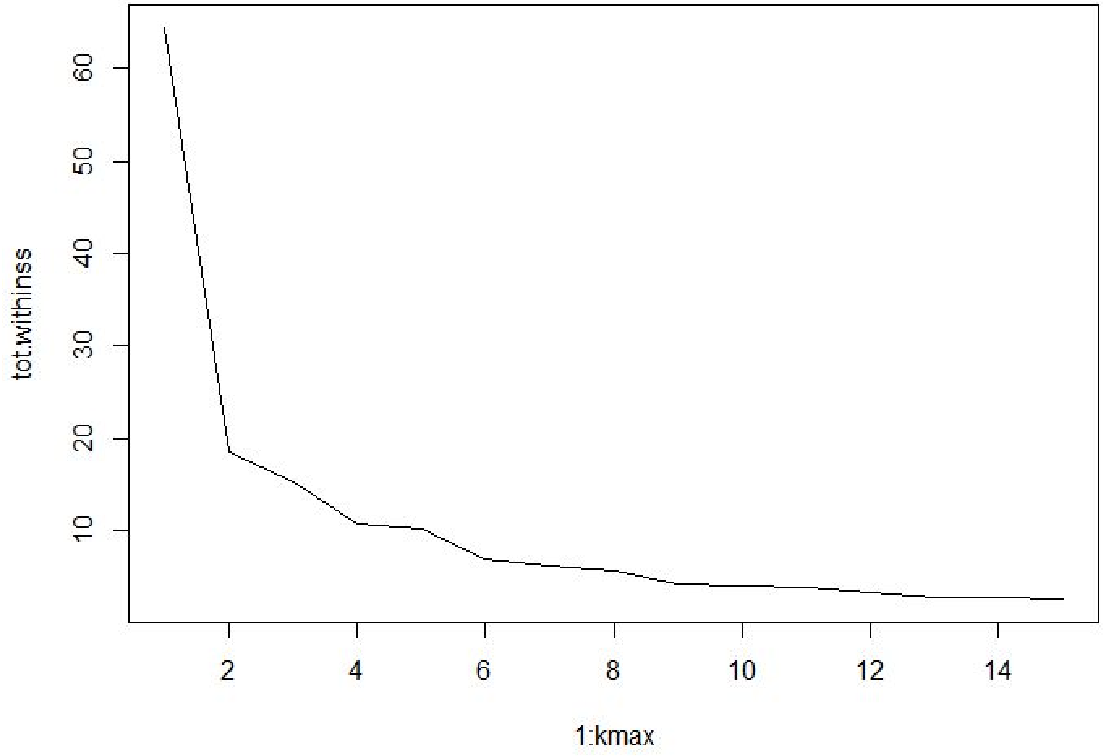
```
################################################
## Step 3: Clustering #
##############################################

## First we determine the maximum number
## of clusters

kmax <- 15
tot.withinss <- sapply(1:kmax,
           function(k){
           kmeans(data,k)$tot.withinss
           })

plot(1:kmax, tot.withinss, type = "l")

# From the plot it seems like 8 is a reasonable cho
km <- kmeans(data,8) rfm$cluster <- km$cluster
```



Step 4: Interpreting the results

```
################################################
## Step 4: Interprete cluster #
##############################################
m_rec <- mean(rfm$Recency)
m_freq <- mean(rfm$Frequency)
m_mon <- mean(rfm$Monetary)

output <- rfm %>%
    group_by(cluster) %>%
    summarise(avg_rec = mean(Recency),
avg_freq = mean(Frequency),           avg_mon =
mean(Monetary)) %>%    mutate(label =
str_c(ifelse(avg_rec>m_rec, "H","L"),
ifelse(avg_freq>m_freq, "H","L"),
ifelse(avg_mon>m_mon, "H","L") ))
```

```
> output
# A tibble: 8 x 5
  cluster   avg_rec  avg_freq   avg_mon label
    <int>     <dbl>     <dbl>     <dbl> <chr>
1       1 146.25134  3.538795  306.9001   HLL
2       2  14.63626  3.453233  403.8425   LLL
3       3  17.66839  7.248705  204.3212   LHL
4       4 138.60137  2.501197  589.7039   HLL
5       5  71.11894  1.678414  186.8943   LLL
6       6  72.84722  6.861111  614.2153   LHL
7       7 130.04491 11.876799 1708.4940   HHH
8       8  16.57364 12.522376 2201.0769   LHH
>
```

How to read the labels?

- LHH: Diamond segment, our best customers
- LHL: Frequent buyers, figure out what do they like and try to cross-sell something
- LLH: Promising
- LLL: Hard to assess
- HHH: Sleeping beauties, good customers that need reactivation
- HHL: Budget-conscious, worth reactivating if the goal is increase market share
- HLH: Worth reactivating if the goal is to increase sales
- HLL: Probably lost, may not be worth reactivating