WBA_EDA

Benjamin Tan

2022-08-26

Contents

1 Libraries and Reading in Data **1**

2 Data Cleaning **2**

2.1 Cleaning Frequency Values 2

2.2 Wide to Long Format 2

3 Optimal NBumber of Clusters **3**

3.1 Clustering Variables 3

3.2 Distance Matrix 5

```
rm(list=ls())
```

1 Libraries and Reading in Data

```
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(magrittr)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```
library(ggplot2)
library(stringr)
```

```
wba_data <- read.csv("WBA_data.csv")
```

2 Data Cleaning

2.1 Cleaning Frequency Values

```
cnames <- colnames(wba_data)

# Columns 40 and beyond are frequencies for WBA
cnames[40:length(cnames)] %<>%
  str_replace_all("f", "") %>%
  str_sub(2,-2)

colnames(wba_data) <- cnames
```

2.2 Wide to Long Format

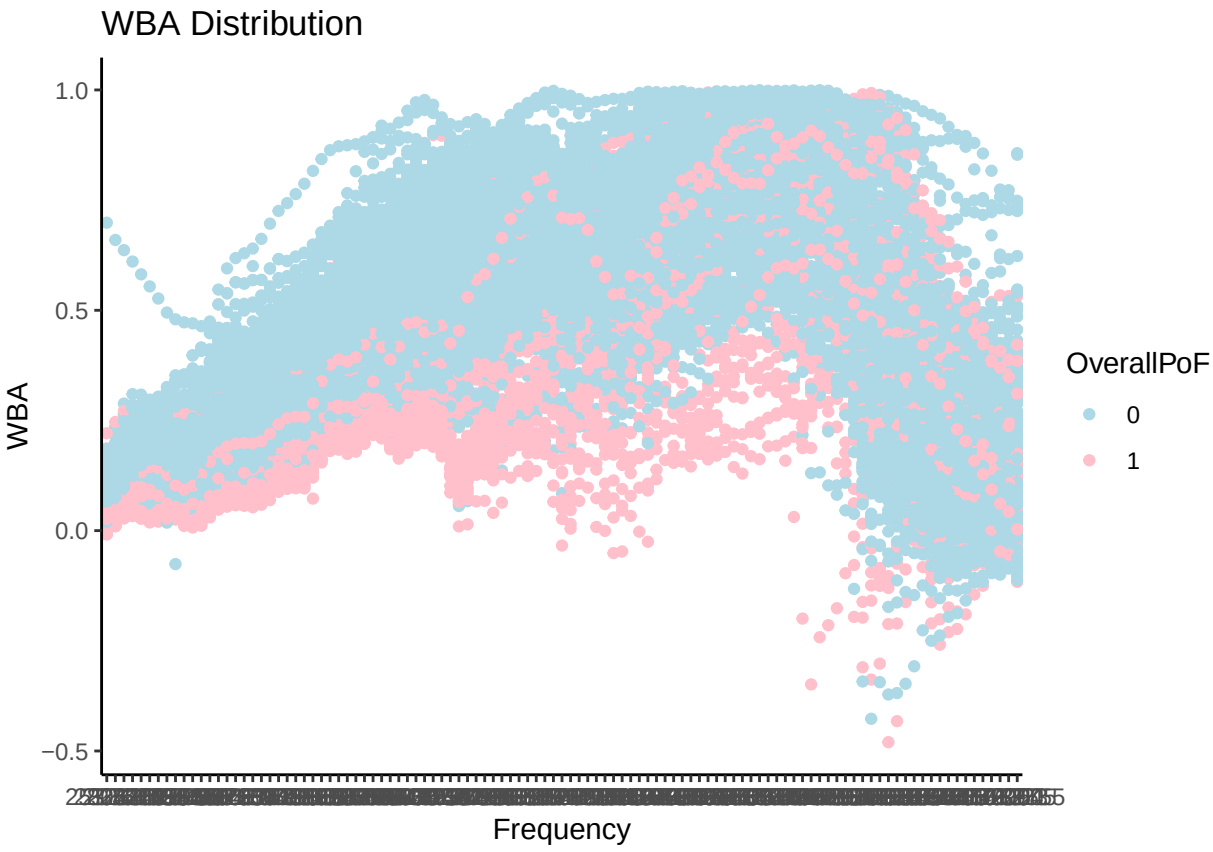
```
frequencies <- colnames(wba_data)[40:length(colnames(wba_data))]
```

```
wba_long <- wba_data %>%
  pivot_longer(cols=frequencies,
               names_to = "freq",
               values_to="wba")
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(frequencies)‘ instead of ‘frequencies‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```r
wba_long$OverallPoF <- as.factor(wba_long$OverallPoF)
wba_long$freq <- as.numeric(wba_long$freq)
wba_long$freq <- as.factor(wba_long$freq)

wba_long %>%
  # filter(OverallPoF == 0) %>%
  ggplot(mapping=aes(x=freq, y=wba, colour=OverallPoF)) +
  geom_point() +
  scale_color_manual(values=c("0" = "lightblue", "1" = "pink")) +
  labs(title="WBA Distribution", x="Frequency",y="WBA") +
  theme_classic()
```



3 Optimal NBumber of Clusters

3.1 Clustering Variables

Specifying which variables to cluster by

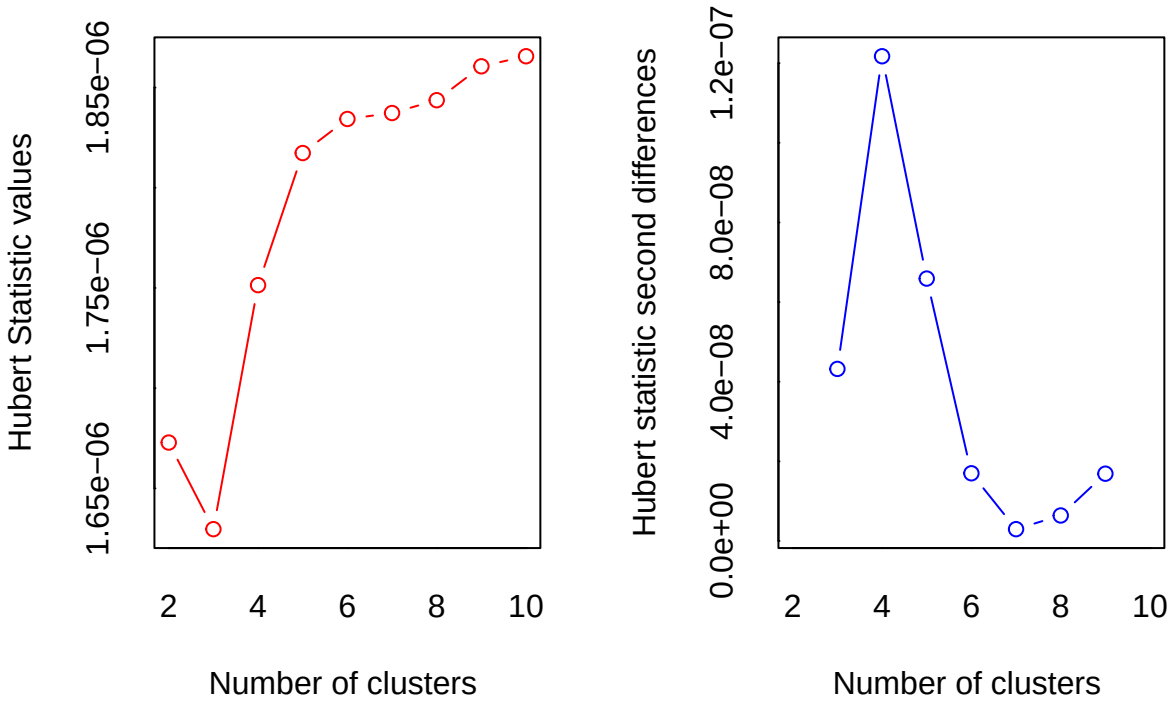
```r
frequencies <- colnames(wba_data)[40:length(colnames(wba_data))]

cluster_vars <- c("Gender", "AgeY", "ECV", "TPP")
```

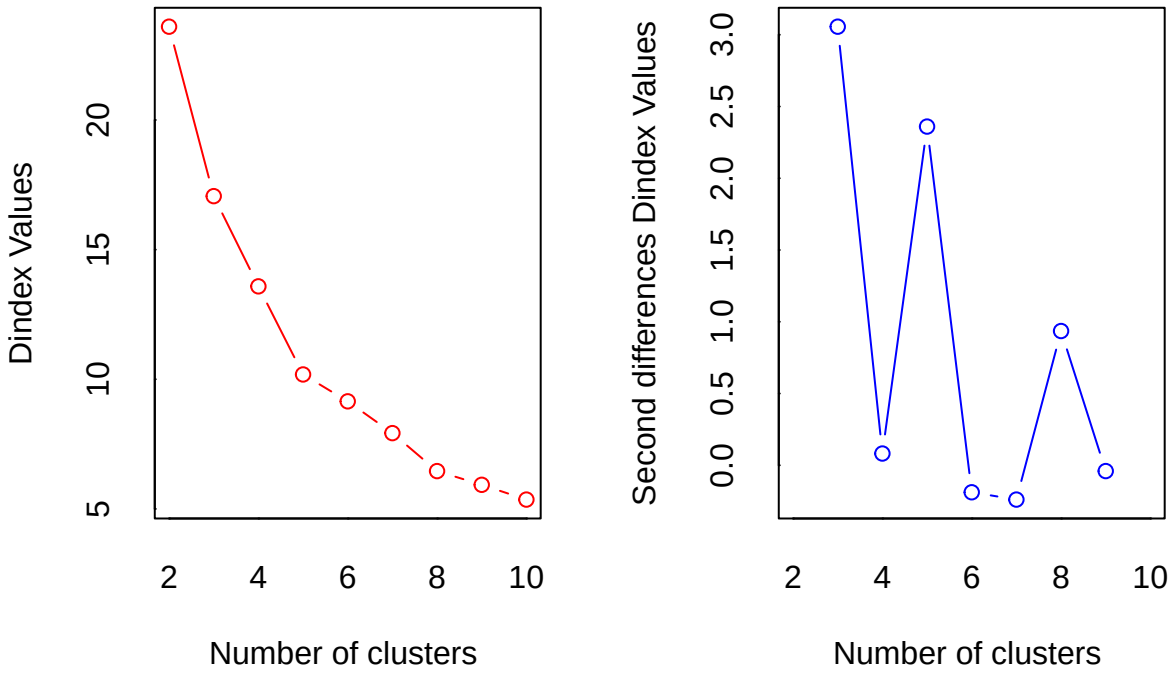
```r
optimal_clusters <-
  wba_data[cluster_vars] %>%
  NbClust::NbClust(
    data = .,
    distance = "euclidean",

    # Assessing 2-10 clusters
    min.nc = 2,
    max.nc = 10,
    method = "ward.D2",

    # All optimal cluster criterion
    index = "all"
  )
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 8 proposed 2 as the best number of clusters
## * 3 proposed 3 as the best number of clusters
## * 6 proposed 4 as the best number of clusters
## * 4 proposed 10 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

3.2 Distance Matrix

```
dist_matrix <- dist(wba_data[cluster_vars], method="euclidean")
```