Rozdział 6 - Segmentacja - ćwiczenia

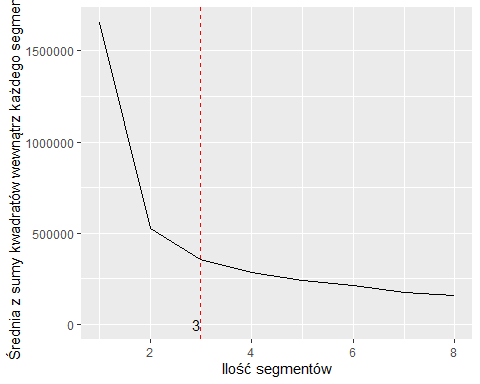
# Wczytanie i przygotowanie danych

rm(list=ls())  
Dane<-read.csv("./Dane/07 - reklama.csv")  
Dane<-Dane[,-1]  
# dodanie wartości odstających  
Dane<-rbind(Dane,c(170,100,100,200))

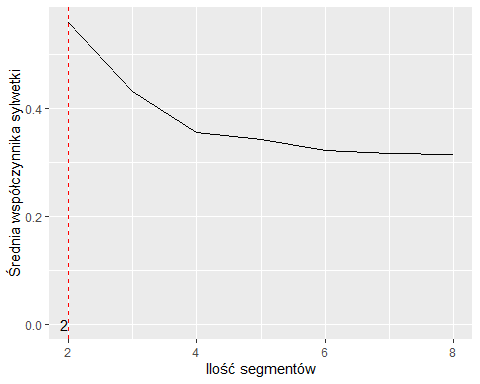
# Segmentacja danych metodą k-średnich

## Dobór ilości segmentów metodą łokcia

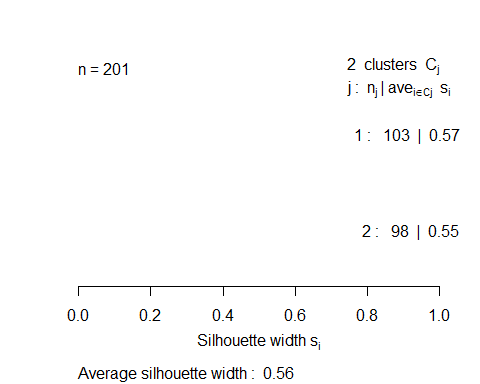
k<-8  
wss<-sapply(1:k,function(k){kmeans(Dane, k)$tot.withinss})  
  
#optymalny dobór to tam gdzie jest zagięcie wykresu - tzw łokieć  
library(ggplot2)  
ggplot()+geom\_line(aes(x=1:8,y=wss))+xlab("Ilość segmentów")+  
 ylab("Średnia z sumy kwadratów wewnątrz każdego segmentu")+  
 geom\_vline(xintercept = 3,col="red",linetype="dashed")+  
 geom\_text(aes(x=3,y=0,label="3"),hjust = "right")

 ## Dobór ilości segmentów metodą średniej sylwetki

library(cluster)  
k<-8  
sc<-sapply(2:k,function(k){summary(silhouette(  
 kmeans(Dane, k)$cluster,  
 dist(Dane,"euclidean")))$avg.width})  
  
#optymalny dobór to tam gdzie wykres przyjmuje wartość max  
ggplot()+geom\_line(aes(x=2:8,y=sc))+xlab("Ilość segmentów")+  
 ylab("Średnia współczynnika sylwetki")+  
 geom\_vline(xintercept = 2,col="red",linetype="dashed")+  
 geom\_text(aes(x=2,y=0,label="2"),hjust = "right")



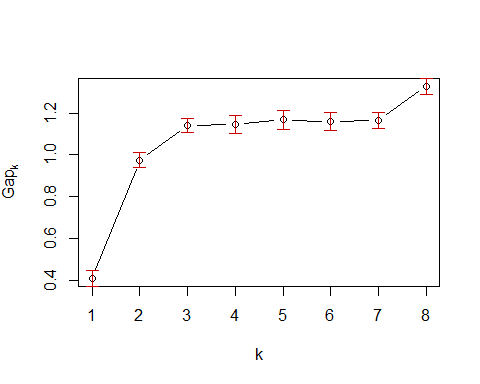
plot(silhouette(kmeans(Dane, 2)$cluster,dist(Dane,"euclidean")),main="")

 ## Dobór ilości segmentów metodą statystyki luk

Gap<-clusGap(Dane,FUN = kmeans,K.max = 8,B=100,spaceH0 = "original",  
 d.power = 2)  
print(Gap,method="firstSEmax")

## Clustering Gap statistic ["clusGap"] from call:  
 ## clusGap(x = Dane, FUNcluster = kmeans, K.max = 8, B = 100, d.power = 2, spaceH0 = "original")  
 ## B=100 simulated reference sets, k = 1..8; spaceH0="original"  
 ## --> Number of clusters (method 'firstSEmax', SE.factor=1): 3  
 ## logW E.logW gap SE.sim  
 ## [1,] 13.62682 14.03497 0.4081515 0.03774985  
 ## [2,] 12.47751 13.45183 0.9743235 0.03632266  
 ## [3,] 12.08538 13.22614 1.1407645 0.03488940  
 ## [4,] 11.86341 13.00866 1.1452456 0.04248368  
 ## [5,] 11.69298 12.86161 1.1686322 0.04586951  
 ## [6,] 11.58974 12.74854 1.1588015 0.04183574  
 ## [7,] 11.50512 12.67030 1.1651842 0.03855072  
 ## [8,] 11.27100 12.59905 1.3280419 0.03941550

plot(Gap,main="")

 # Podział na 3 segmenty

podzial<-kmeans(Dane,3)  
Dane<-cbind(Dane,as.factor(podzial$cluster))  
colnames(Dane)[5]<-"Segment\_kmeans"

# Segmentacja danych metodą k-median

library(Gmedian)  
podzial<-kGmedian(Dane[,-5],3)

## Warning: did not converge in 10 iterations

Dane<-cbind(Dane,as.factor(podzial$cluster))  
colnames(Dane)[6]<-"Segment\_kmedian"  
levels(Dane$Segment\_kmedian)<-c("2","1","3") #zmiana numeracji segmentów

# Segmentacja danych metodą PAM

podzial<-pam(Dane[,c(-5,-6)],3)  
Dane<-cbind(Dane,as.factor(podzial$cluster))  
colnames(Dane)[7]<-"Segment\_pam"  
levels(Dane$Segment\_pam)<-c("1","2","3") #zmiana numeracji segmentów

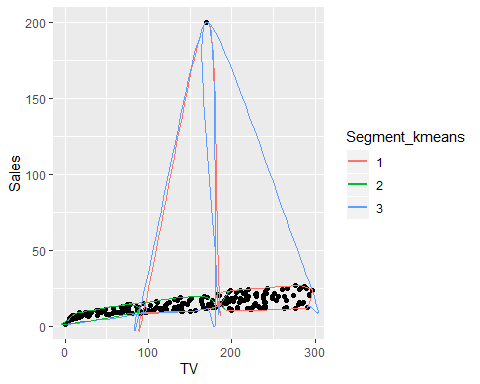
# Wizualizacja segmentacji

library(ggalt)

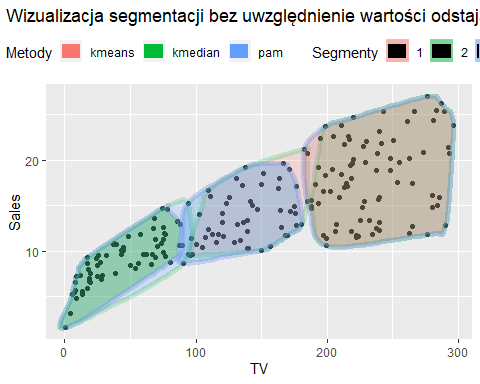
## Warning: package 'ggalt' was built under R version 3.6.2

## Registered S3 methods overwritten by 'ggalt':  
 ## method from   
 ## grid.draw.absoluteGrob ggplot2  
 ## grobHeight.absoluteGrob ggplot2  
 ## grobWidth.absoluteGrob ggplot2  
 ## grobX.absoluteGrob ggplot2  
 ## grobY.absoluteGrob ggplot2

ggplot() + geom\_point(data=Dane,aes(x=TV,y=Sales))+  
 geom\_encircle(data=Dane,aes(x=TV,y=Sales,col=Segment\_kmeans),expand=0)+  
 geom\_encircle(data=Dane,aes(x=TV,y=Sales,col=Segment\_kmedian),expand=0)+  
 geom\_encircle(data=Dane,aes(x=TV,y=Sales,col=Segment\_pam),expand=0)



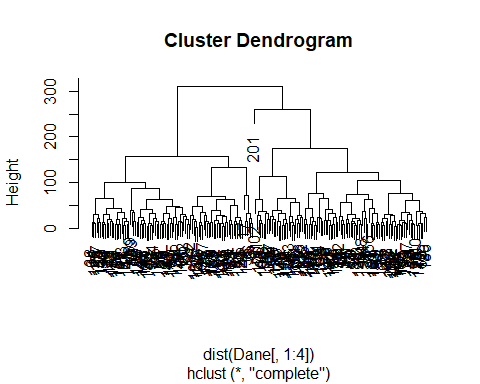
#bez wartości odstającej  
ggplot() + geom\_point(data=Dane[-201,],aes(x=TV,y=Sales))+  
 geom\_encircle(data=Dane[-201,],  
 aes(x=TV,y=Sales,fill=Segment\_kmeans,col="kmeans"),  
 expand=0,alpha=0.1,size=5)+  
 geom\_encircle(data=Dane[-201,],  
 aes(x=TV,y=Sales,fill=Segment\_kmedian,col="kmedian"),  
 expand=0,alpha=0.2,size=5)+  
 geom\_encircle(data=Dane[-201,],  
 aes(x=TV,y=Sales,fill=Segment\_pam,col="pam"),  
 expand=0,alpha=0.3,size=5)+  
 labs(fill = "Segmenty")+theme(legend.position = "top")+labs(col="Metody")+  
 labs(title="Wizualizacja segmentacji bez uwzględnienie wartości odstającej")+  
 theme(plot.title = element\_text(hjust = 0.5))



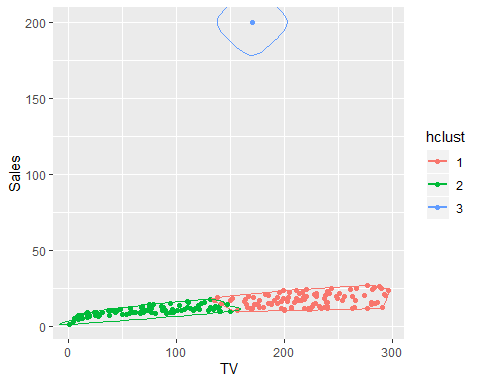
# Segmentacja hierarchiczna

## Podzial funkcją hclust

podzial<-hclust(dist(Dane[,1:4]),method = "complete")  
plot(podzial)

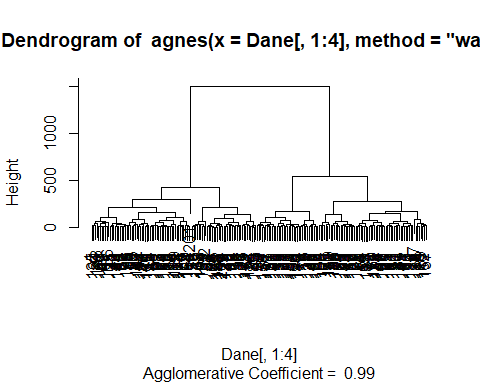
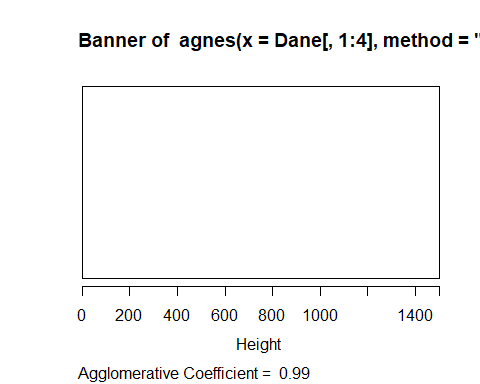


segmenty<-cutree(podzial,k=3)  
Dane<-cbind(Dane,hclust=as.factor(segmenty))  
# wizualizacja  
ggplot() + geom\_point(data=Dane,aes(x=TV,y=Sales,col=hclust))+  
 geom\_encircle(data=Dane,aes(x=TV,y=Sales,col=hclust),expand=0)

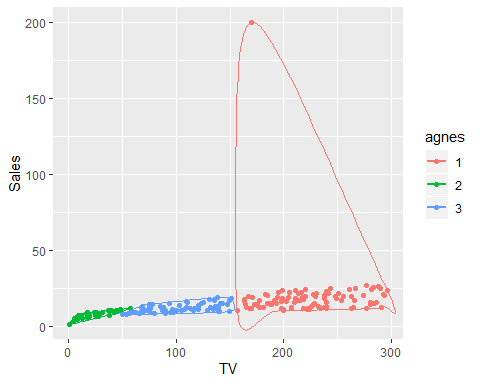


## Podzial funkcją agnes

podzial<-agnes(Dane[,1:4],method = "ward")  
plot(podzial)



segmenty<-cutree(podzial,k=3)  
Dane<-cbind(Dane,agnes=as.factor(segmenty))  
# wizualizacja  
ggplot() + geom\_point(data=Dane,aes(x=TV,y=Sales,col=agnes))+  
 geom\_encircle(data=Dane,aes(x=TV,y=Sales,col=agnes),expand=0)



# Ćwiczenia —————————————————————

#Wykonaj segmentacje danych z pliku 03 - Analiza płac.csv

#Wykonaj segmentacje danych z pliku 04 - kredyty.xls

#Wykonaj segmentacje krai europejskich pod względem przestepczości z pliku crim\_hom\_soff.tsv

#Wykonaj segmentacje danych z pliku 05 - choroba.xlsx

#Wykonaj segmentacje danych z pliku 06 - nadciśnienie.xlsx

## Analiza statystyczna opisowa pliku *04 - kredyty.xls*

# 1. Wczytanie i transformacja pliku csv *04 - kredyty.xls*.

rm(list=ls())  
library(dplyr)

##   
 ## Attaching package: 'dplyr'

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

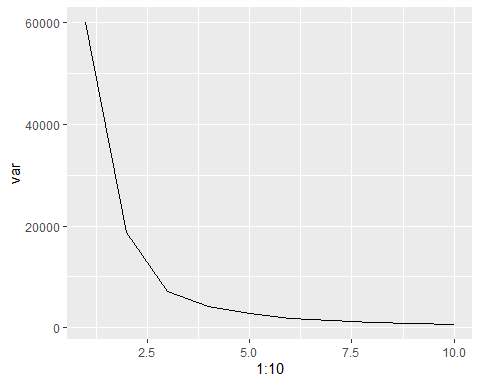
## The following objects are masked from 'package:base':  
 ##   
 ## intersect, setdiff, setequal, union

library(tidyr)  
library(readxl)  
Dane<-read\_xls("./Dane/04 - kredyty.xls")  
Dane<- Dane %>% select(LIMIT\_BAL,SEX,EDUCATION,MARRIAGE,AGE) %>%   
 mutate\_at(c("SEX","EDUCATION","MARRIAGE"),as.factor)  
levels(Dane$SEX)<-c("Mężczyzna","Kobieta")  
levels(Dane$MARRIAGE)<-c("W związku","Rozwiedziony","Kawaler/Panna","Inny")  
tbl\_df(head(Dane))

## # A tibble: 6 x 5  
 ## LIMIT\_BAL SEX EDUCATION MARRIAGE AGE  
 ## <dbl> <fct> <fct> <fct> <dbl>  
 ## 1 20000 Kobieta 2 Rozwiedziony 24  
 ## 2 120000 Kobieta 2 Kawaler/Panna 26  
 ## 3 90000 Kobieta 2 Kawaler/Panna 34  
 ## 4 50000 Kobieta 2 Rozwiedziony 37  
 ## 5 50000 Mężczyzna 2 Rozwiedziony 57  
 ## 6 50000 Mężczyzna 1 Kawaler/Panna 37

## 2. Segmentacja k-średnich

DaneS<-Dane %>% select(LIMIT\_BAL,AGE) %>% mutate\_all(scale)  
  
var<-sapply(1:10,function(x) mean(kmeans(DaneS,x)$withinss))  
library(ggplot2)  
ggplot()+geom\_line(aes(x=1:10,y=var))



podzial<- Dane %>% select(LIMIT\_BAL,AGE) %>% mutate\_all(scale) %>% kmeans(3)  
Dane<-bind\_cols(Dane,data.frame(SEGMENT=podzial$cluster))  
  
Podsumowanie<-Dane %>% group\_by(SEGMENT,cut(AGE,c(0,40,60,80)),cut(LIMIT\_BAL,breaks=c(0,100000,1000000))) %>% summarise(ILOSC=n())  
levels(Podsumowanie$`cut(AGE, c(0, 40, 60, 80))`)<-c("młody","średni","stary")   
colnames(Podsumowanie)[2]<-"Wiek"  
colnames(Podsumowanie)[3]<-"Limit kredytowy"  
levels(Podsumowanie$`Limit kredytowy`)<-c("Mały","Duży")  
spread(Podsumowanie,Wiek,"ILOSC")

## # A tibble: 5 x 5  
 ## # Groups: SEGMENT [3]  
 ## SEGMENT `Limit kredytowy` młody średni stary  
 ## <int> <fct> <int> <int> <int>  
 ## 1 1 Duży 5641 1745 25  
 ## 2 2 Mały 690 3356 93  
 ## 3 2 Duży 843 2901 154  
 ## 4 3 Mały 8359 NA NA  
 ## 5 3 Duży 6193 NA NA