Practica 2 - clustering

Cédric Prieels

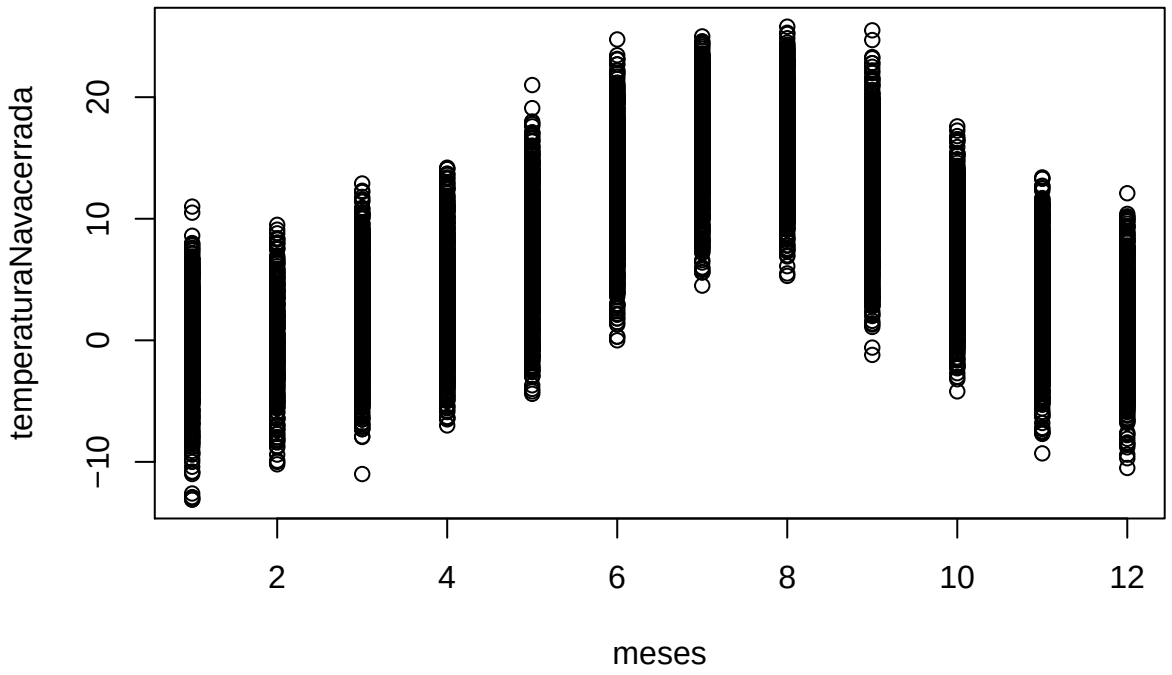
Diciembre 2016

Lo primero que se puede hacer como tarea es calcular una k-media con una k que vale 4, y que corresponde a las 4 estaciones, para ver que sale para las temperatures de Navacerrada. El objetivo principal consiste en calcular el percentage de datos de días bien clasificados en cada estación.

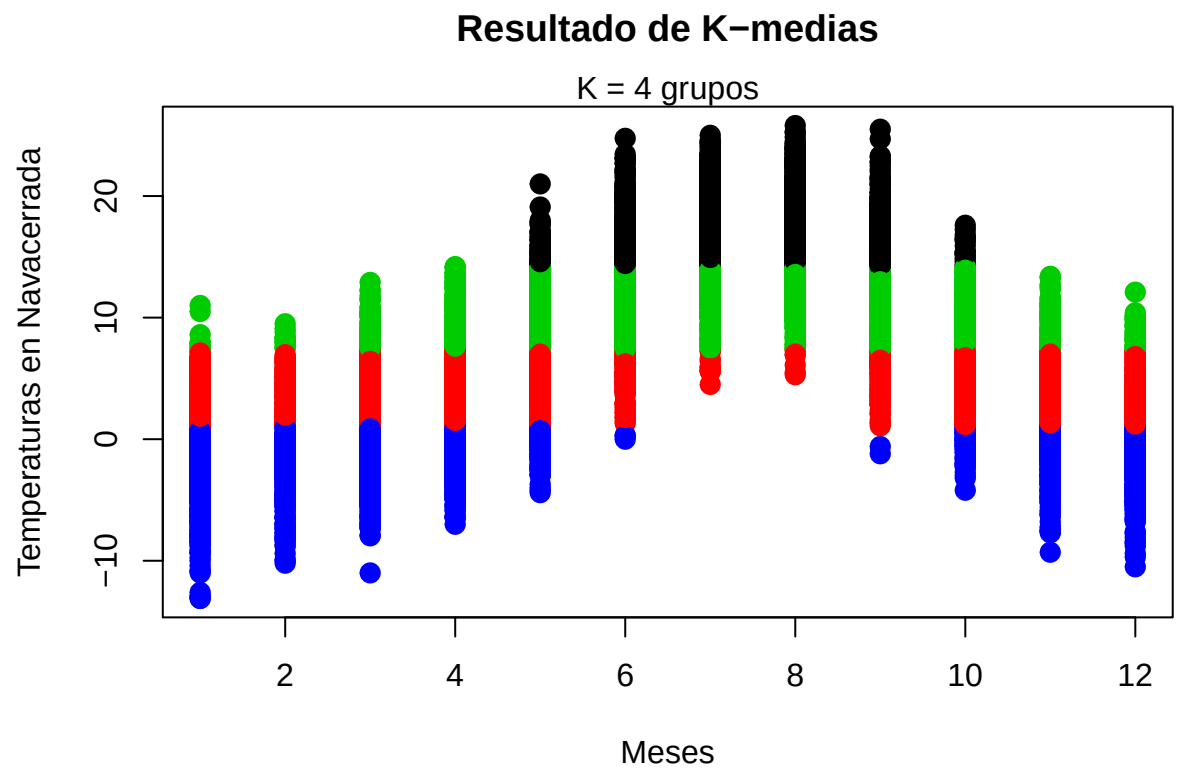
```
rm(list=ls())

setwd('/Users/ced2718/Documents/Universite/Modelizacion/')
data <- read.table("tmean.txt", header=T)
data <- data[complete.cases(data),]
meses <- data[,2]
dias <- data[,3]

temperaturaSanSebastian <- data[,4]
temperaturaBarcelona <- data[,5]
temperaturaSalamanca <- data[,6]
temperaturaNavacerrada <- data[,7]
temperaturaAlbacete <- data[,8]
temperaturaCordoba <- data[,9]

temperaturas <- data[complete.cases(data),]
```

Ahora que hemos leido la tabla de los datos con las temperatures en Navacerrada, podemos pintar lo que tenemos (la temperatura en Navacerrada en función de los meses del año). Después calculamos k-media con k=4, y pintamos el resultado de la operación.

```
par(mfrow = c(1,1))

plot(meses, temperaturaNavacerrada)
```




```
km.out <- kmeans(temperaturaNavacerrada, centers = 4)
plot(meses, temperaturaNavacerrada, col = (km.out$cluster),
     main = "Resultado de K-medias", xlab = "Meses", ylab = "Temperaturas en Navacerrada",
     pch = 20, cex = 2)
mtext("K = 4 grupos")
```



```r
head(km.out$cluster, 100)
```

```
##   [1] 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 4 4 2 4 4 2 2 2 4 2 2 4 4 4 4 4 4 4 2
##  [36] 2 2 4 4 4 2 2 3 2 4 4 4 4 4 4 4 4 4 4 2 2 2 2 4 4 4 4 4 4 4 4 4 4 4 2
##  [71] 2 4 4 4 4 4 4 4 2 2 4 4 2 2 2 2 4 4 4 4 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Lo que podemos hacer ahora es verificar si cada punto está en el buen grupo, o no. Por ejemplo, se puede intentar responder a la pregunta siguiente : de todos los días fríos, cuáles partenecen al invierno y cuáles no? Para responder a este tipo de pregunta, hay primero que escribir una función que nos devuelve la estación en función de la fecha.

```r
getSeason <- function(dia, mes) {
  WS <- as.Date("2016-12-15", format = "%Y-%m-%d") # Winter Solstice
  SE <- as.Date("2016-3-15",  format = "%Y-%m-%d") # Spring Equinox
  SS <- as.Date("2016-6-15",  format = "%Y-%m-%d") # Summer Solstice
  FE <- as.Date("2016-9-15",  format = "%Y-%m-%d") # Fall Equinox

  fecha <- as.Date(paste(dia, mes, "2016", sep="-"),format='%d-%m-%Y')

  ifelse (fecha >= WS | fecha < SE, "Invierno",
    ifelse (fecha >= SE & fecha < SS, "Primavera",
      ifelse (fecha >= SS & fecha < FE, "Verano", "Otono")))
}

#Ejemplos
getSeason(1,1)
```

```
## [1] "Invierno"
```

```r
getSeason(1,6)
```

```
## [1] "Primavera"
```

Ahora podemos calcular lo que queremos (primero, el número de días fríos que pertenecen o no al invierno) con esta función.

```r
#Invierno
diasFriosDeInvierno <- 0
diasFriosDeOtraTemporada <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 2) {
    if(getSeason(dias[i], meses[i]) == "Invierno") {
      diasFriosDeInvierno <- diasFriosDeInvierno + 1
    } else {
      diasFriosDeOtraTemporada <- diasFriosDeOtraTemporada + 1
    }
  }
}
diasFriosDeInvierno
```

```
## [1] 665
```

```
diasFriosDeOtraTemporada
```

```
## [1] 1928
```

```
percentageInvierno <-
  (diasFriosDeInvierno/(diasFriosDeInvierno+diasFriosDeOtraTemporada))*100
percentageInvierno
```

```
## [1] 25.64597
```

Volvemos a repetir lo mismo para las 3 otras estaciones.

```
#Primavera
diasMediosDePrimavera <- 0
diasMediosDeOtraTemporada1 <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 1) {
    if(getSeason(dias[i], meses[i]) == "Primavera") {
      diasMediosDePrimavera <- diasMediosDePrimavera + 1
    } else {
      diasMediosDeOtraTemporada1 <- diasMediosDeOtraTemporada1 + 1
    }
  }
}
diasMediosDePrimavera
```

```
## [1] 157
```

```
diasMediosDeOtraTemporada1
```

```
## [1] 1832
```

```
percentagePrimavera <-
  (diasMediosDePrimavera/(diasMediosDePrimavera+diasMediosDeOtraTemporada1))*100
percentagePrimavera
```

```
## [1] 7.893414
```

```
#Verano
diasCalientesDeVerano <- 0
diasCalientesDeOtraTemporada <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 4) {
    if(getSeason(dias[i], meses[i]) == "Verano") {
      diasCalientesDeVerano <- diasCalientesDeVerano + 1
    } else {
      diasCalientesDeOtraTemporada <- diasCalientesDeOtraTemporada + 1
    }
  }
}
diasCalientesDeVerano
```

```
## [1] 0
```

```
diasCalientesDeOtraTemporada
```

```
## [1] 2078
```

```
percentageVerano <-
  (diasCalientesDeVerano/(diasCalientesDeVerano+diasCalientesDeOtraTemporada))*100
percentageVerano
```

```
## [1] 0
```

```
#Otoño
diasMediosDeOtono <- 0
diasMediosDeOtraTemporada2 <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 3) {
    if(getSeason(dias[i], meses[i]) == "Otono") {
      diasMediosDeOtono <- diasMediosDeOtono + 1
    } else {
      diasMediosDeOtraTemporada2 <- diasMediosDeOtraTemporada2 + 1
    }
  }
}
diasMediosDeOtono
```

```
## [1] 685
```

```
diasMediosDeOtraTemporada2
```

```
## [1] 1522
```

```
percentageOtono <-
  (diasMediosDeOtono/(diasMediosDeOtono+diasMediosDeOtraTemporada2))*100
percentageOtono
```

```
## [1] 31.03761
```

A priorio, tendremos menos percentaje de días en el buen grupo en primavera y otoño porque hay más variabilidad en estos dos grupos. Los resultados obtenidos confirman esta intuición.

```
resultados <- matrix(c(percentageInvierno, percentagePrimavera,
                       percentageVerano, percentageOtono), ncol=4)
colnames(resultados) <- c("Invierno", "Primavera", "Verano", "Otono")
rownames(resultados) <- c("Percentages")
rtab <- as.table(resultados)
head(rtab)
```

```
##              Invierno Primavera    Verano     Otono
## Percentages 25.645970  7.893414  0.000000 31.037608
```

Ahora podemos volver a hacer lo mismo, considerando todas las ciudades a la vez (tendremos por lo tanto un espacio a 6 dimensiones, una dimensión por cada lugar de España considerado). Usamos el data frame creado llamado *temperaturas* en esta parte del ejercicio, porque tiene los datos de cada ciudad.

```
km.out <- kmeans(temperaturas, centers = 4)
head(km.out$cluster, 100)
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   2   2   2   2   2   2   2   2   3   2   2   3   3   3   3   3   3
##  19  20  21  22  23  24  25  26  27  28  29  30  31  60  61  62  63  64
##   3   3   3   3   3   3   3   3   3   3   3   3   3   2   2   2   2   2
##  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82
##   2   2   2   2   2   2   2   2   3   3   3   3   3   3   3   3   3   3
##  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   1   1   3   3   3   3   3   3   2   2   2   2   2   2   2   2   2   2
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
##   2   2   2   3   3   2   2   3   3   3   3   1   1   3   3   3   3   3
## 119 120 121 122 123 124 125 126 127 128
##   3   3   2   2   2   2   2   2   2   2
```

Ahora volvemos a repetir exactamente lo mismo que antes pero usando esta vez el km.out completo, con los resultados de las 6 cuidades de España.

```
#Invierno
diasFriosDeInvierno <- 0
diasFriosDeOtraTemporada <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 4) {
    if(getSeason(dias[i], meses[i]) == "Invierno") {
      diasFriosDeInvierno <- diasFriosDeInvierno + 1
    } else {
      diasFriosDeOtraTemporada <- diasFriosDeOtraTemporada + 1
    }
  }
}
diasFriosDeInvierno
```

```
## [1] 0
```

```
diasFriosDeOtraTemporada
```

```
## [1] 2791
```

```
percentageInvierno <-
  (diasFriosDeInvierno/(diasFriosDeInvierno+diasFriosDeOtraTemporada))*100
percentageInvierno
```

```
## [1] 0
```

```r
#Primavera
diasMediosDePrimavera <- 0
diasMediosDeOtraTemporada1 <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 1) {
    if(getSeason(dias[i], meses[i]) == "Primavera") {
      diasMediosDePrimavera <- diasMediosDePrimavera + 1
    } else {
      diasMediosDeOtraTemporada1 <- diasMediosDeOtraTemporada1 + 1
    }
  }
}
diasMediosDePrimavera
```

```
## [1] 863
```

```r
diasMediosDeOtraTemporada1
```

```
## [1] 926
```

```r
percentagePrimavera <-
  (diasMediosDePrimavera/(diasMediosDePrimavera+diasMediosDeOtraTemporada1))*100
percentagePrimavera
```

```
## [1] 48.23924
```

```r
#Verano
diasCalientesDeVerano <- 0
diasCalientesDeOtraTemporada <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 2) {
    if(getSeason(dias[i], meses[i]) == "Verano") {
      diasCalientesDeVerano <- diasCalientesDeVerano + 1
    } else {
      diasCalientesDeOtraTemporada <- diasCalientesDeOtraTemporada + 1
    }
  }
}
diasCalientesDeVerano
```

```
## [1] 32
```

```r
diasCalientesDeOtraTemporada
```

```
## [1] 2253
```

```r
percentageVerano <-
  (diasCalientesDeVerano/(diasCalientesDeVerano+diasCalientesDeOtraTemporada))*100
percentageVerano
```

```
## [1] 1.400438
```

```r
#Otoño
diasMediosDeOtono <- 0
diasMediosDeOtraTemporada2 <- 0
for(i in 1:length(km.out$cluster)){
  if(km.out$cluster[i] == 3) {
    if(getSeason(dias[i], meses[i]) == "Otono") {
      diasMediosDeOtono <- diasMediosDeOtono + 1
    } else {
      diasMediosDeOtraTemporada2 <- diasMediosDeOtraTemporada2 + 1
    }
  }
}
diasMediosDeOtono
```

```
## [1] 479
```

```r
diasMediosDeOtraTemporada2
```

```
## [1] 1523
```

```r
percentageOtono <-
  (diasMediosDeOtono/(diasMediosDeOtono+diasMediosDeOtraTemporada2))*100
percentageOtono
```

```
## [1] 23.92607
```

```r
resultados <- matrix(c(percentageInvierno, percentagePrimavera,
                       percentageVerano, percentageOtono), ncol=4)
colnames(resultados) <- c("Invierno", "Primavera", "Verano", "Otono")
rownames(resultados) <- c("Percentages")
rtab <- as.table(resultados)
head(rtab)
```

```
##              Invierno Primavera    Verano     Otono
## Percentages  0.000000 48.239240  1.400438 23.926074
```

Vemos que en general, al considerar diferentes cuidades de España, los percentages obtenidos bajan. Esto se puede entender por los diferentes climates que hay en las diferentes zonas de España.

Bibliografia

R Markdown, *Markdown basics*, http://rmarkdown.rstudio.com/authoring_basics.html. Consultado por última vez el 29 de octubre 2016.

R Development Core Team (2008). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Find which season a particular date belongs to, STACKOVERFLOW, http://stackoverflow.com/questions/9500114/find-which-season-a-particular-date-belongs-to, consultado por última vez el 2 de diciembre 2016.