Activitat 1: Exploració i preprocés de dades

Marc Cervera Rosell

05-04-2024

1: Lectura de dades i examinació del tipus de variale

1.1 Carregar el fitxer de dades

Llegir el fitxer de dades i consultar el nom de les columnes del fitxer.

```
tryCatch({
  data <- read.csv("pisa2009-1.csv", header = TRUE)
  print("El fitxer s'ha llegit correctament")
}, error = function(e){
  print("ERROR en el moment de llegir el document:",conditionMessage(e), "\n")
})
```

```
## [1] "El fitxer s'ha llegit correctament"
```

Si per alguna raó el fitxer no es pot llegir, en haver un block tryCatch, es treurà per pantalla un missatge d'error. En cas de poder-se llegir sense problema i que tot vagi bé, a més de llegir-se el fitxer es treurà un missatge per pantalla indicant que el fitxer s'ha llegit correctament.

```
columns <- names(data)
print(columns)
```

```
##  [1] "grade"                 "male"                  "raceeth"
##  [4] "preschool"             "expectBachelors"       "motherHS"
##  [7] "motherBachelors"       "motherWork"            "fatherHS"
## [10] "fatherBachelors"       "fatherWork"            "selfBornUS"
## [13] "motherBornUS"          "fatherBornUS"          "englishAtHome"
## [16] "computerForSchoolwork" "read30MinsADay"        "minutesPerWeekEnglish"
## [19] "studentsInEnglish"     "schoolHasLibrary"      "publicSchool"
## [22] "urban"                 "schoolSize"            "readingScore"
```

1.2 Examinar el tipus de vasriables

Indicar quines variables són de naturalesa numèrica, caràcter i categòrica.En cas que el tipus de variable que ha atorgat R no coincideixi amb el tipus que li correspondria, indicar de quines variables es tracta. Considereu que les variables binàries prenen valors 1 o 0. La transformació corresponent, si és necessària, s'aplicarà en els apartats següents, una vegada normalitzades les variables.

```
type <- sapply(data, class)

for (i in seq_along(columns)) {
  cat("La columna",columns[i], "és de tipus", type[i], "\n")
}
```

```
## La columna grade és de tipus integer
## La columna male és de tipus integer
## La columna raceeth és de tipus character
## La columna preschool és de tipus integer
## La columna expectBachelors és de tipus integer
## La columna motherHS és de tipus integer
## La columna motherBachelors és de tipus integer
## La columna motherWork és de tipus integer
## La columna fatherHS és de tipus integer
## La columna fatherBachelors és de tipus integer
## La columna fatherWork és de tipus integer
## La columna selfBornUS és de tipus integer
## La columna motherBornUS és de tipus integer
## La columna fatherBornUS és de tipus integer
## La columna englishAtHome és de tipus integer
## La columna computerForSchoolwork és de tipus integer
## La columna read30MinsADay és de tipus character
## La columna minutesPerWeekEnglish és de tipus integer
## La columna studentsInEnglish és de tipus integer
## La columna schoolHasLibrary és de tipus integer
## La columna publicSchool és de tipus integer
## La columna urban és de tipus integer
## La columna schoolSize és de tipus integer
## La columna readingScore és de tipus character
```

La funció *sapply* ens permet obtenir el tipus de dades que hi ha en cada columna. Per a fer més senzilla la visualització de la categoría i el seu tipus, s'utilitza un bucle de tipus *for* per a treure per pantalla una frase on es relaciona cada categoria amb el seu tipus.

2: Normalització de variables qualitatives (text)

2.1 Variable raceeth

Mostreu les categories de la variable raceeth. En cas d'inconsistències o errors, corregiu la informació. A continuació, mostreu el percentatge d'estudiants a cada categoria i dibuixeu un gràfic circular (pie chart).

```
categories_raceeth <- unique(data$raceeth)
print(categories_raceeth)
```

```
##  [1] NA
##  [2] "White"
##  [3] "Black"
##  [4] "Hispanic"
##  [5] "Assian"
##  [6] "white"
##  [7] "More than one race"
##  [8] "Asian"
##  [9] "American Indian/Alaska Native"
## [10] "Native Hawaiian/Other Pacific Islander"
## [11] "Asiann"
## [12] "whit"
```

La funció *unique()* ens permet obtenir els diferents valors únics que hi ha en el conjunt de dades que estem estudiant, en aquest cas, la columna *raceeth*.

```
percentages_rounded <- round(prop.table(table(data$raceeth)) * 100, 2)
for (i in seq_along(categories_raceeth)) {
  cat("La categoria",categories_raceeth[i], "té un percetatge de ",
      percentages_rounded[i],"%", "\n")
}
```

```
## La categoria NA té un percetatge de  1.02 %
## La categoria White té un percetatge de  3.75 %
## La categoria Black té un percetatge de  0.14 %
## La categoria Hispanic té un percetatge de  0.06 %
## La categoria Assian té un percetatge de  12.24 %
## La categoria white té un percetatge de  22.99 %
## La categoria More than one race té un percetatge de  3.42 %
## La categoria Asian té un percetatge de  0.85 %
## La categoria American Indian/Alaska Native té un percetatge de  0.08 %
## La categoria Native Hawaiian/Other Pacific Islander té un percetatge de  0.17 %
## La categoria Asiann té un percetatge de  55.29 %
## La categoria whit té un percetatge de  NA %
```

En aquest últim apartat cal posar èmfasi en que s'han arrodonit els percentatges i per tant no són exactes.

Com s'ha fet enteriorment, per tal de facilitar la visualització dels percentatges, s'ha tret per pantalla una frase amb cada categoria i el tant per cent que representa.

Abans de representar les dades en el diagrama de pastís, es pot observar que en el conjunt de dades hi ha alguns errors. Per exemple: "White", "white", "whit", per tant, abans de representar les dades en el diagrama s'han de rectificar els errors del *dataset*

```
data$raceeth <- gsub("\\bwhite\\b", "White", data$raceeth, ignore.case = TRUE)
data$raceeth <- gsub("\\bwhit\\b", "White", data$raceeth, ignore.case = TRUE)
data$raceeth <- gsub("\\bAssian\\b", "Asian", data$raceeth, ignore.case = TRUE)
data$raceeth <- gsub("\\bAsiann\\b", "Asian", data$raceeth, ignore.case = TRUE)
write.csv(data, "pisa_clean.csv", row.names = FALSE)
print("File has been corrected succesfully")
```

```
## [1] "File has been corrected succesfully"
```

Totes les correccions de les dades qualitatives es guardaràn en el nou fitxer corregit en el qual es guardaràn totes les correccions que anirem fent. Aquest nou fitxer s'anomena *pisa_clean.csv* Tornem a repetir els passos de lectura i consulta de les categories però ara amb el fitxer corregit.

```
tryCatch({
  data2 <- read.csv("pisa_clean.csv", header = TRUE)
  print("El fitxer amb les dades corregides s'ha llegit correctament")
}, error = function(e){
  print("ERROR en el moment de llegir el document:",conditionMessage(e), "\n")
})
```

```
## [1] "El fitxer amb les dades corregides s'ha llegit correctament"
```

```
categories_raceeth2 <- unique(data2$raceeth)
print(categories_raceeth2)
```

```
## [1] NA
## [2] "White"
## [3] "Black"
## [4] "Hispanic"
## [5] "Asian"
```

```
## [6] "More than one race"
## [7] "American Indian/Alaska Native"
## [8] "Native Hawaiian/Other Pacific Islander"
```

Com es pot obbservar, un cop corregides les categories de la variable qualitativa *raceeth*, apareixen menys categories a representar en el diagrama de pastís.

```
percentages_rounded2 <- round(prop.table(table(data2$raceeth)) * 100, 2)
for (i in seq_along(categories_raceeth2)) {
  cat("La categoria",categories_raceeth2[i], "té un percetatge de ",
      percentages_rounded2[i],"%", "\n")
}
```

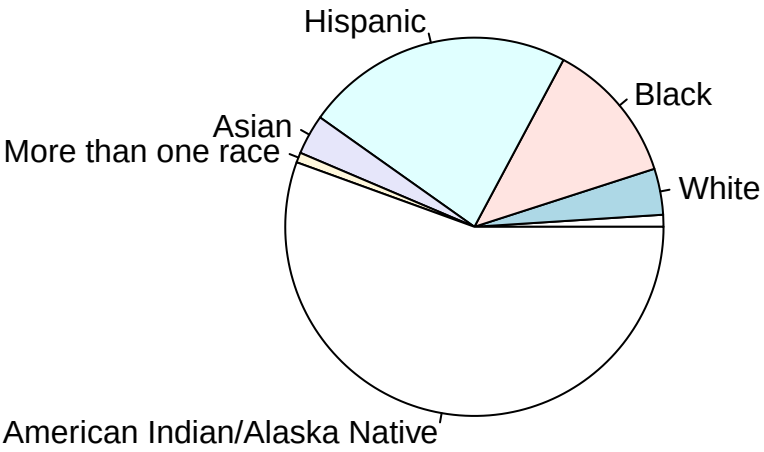
```
## La categoria NA té un percetatge de  1.02 %
## La categoria White té un percetatge de  3.94 %
## La categoria Black té un percetatge de  12.24 %
## La categoria Hispanic té un percetatge de  22.99 %
## La categoria Asian té un percetatge de  3.42 %
## La categoria More than one race té un percetatge de  0.85 %
## La categoria American Indian/Alaska Native té un percetatge de  55.54 %
## La categoria Native Hawaiian/Other Pacific Islander té un percetatge de  NA %
```

```
percentages_not_rounded_aux <- prop.table(table(data2$raceeth)) * 100
print(percentages_not_rounded_aux)
```

```
##
##          American Indian/Alaska Native                                Asian
##                              1.0198456                            3.9415656
##                                  Black                             Hispanic
##                             12.2381477                           22.9878721
##                       More than one race Native Hawaiian/Other Pacific Islander
##                              3.4178611                            0.8544653
##                                  White
##                             55.5402426
```

En aquesta última cel · la es veuen els percentatges exactes de les categories un cop corregits els noms mal escrits

```
percentages_not_rounded <- prop.table(table(data2$raceeth)) * 100
pie(percentages_not_rounded, labels=categories_raceeth2)
```



En el diagrama anterior es representen les categories corregides. Com s'observa hi ha una categoria del diagrama que no té etiqueta, això és degut a que en el fitxer csv aquesta categoría correspon a NA.

3: Normalització i descripció de variables binàries

El conjunt de dades conté un nombre elevat de variables binàries. Reviseu els seus valors i en cas d'errors o inconsistències, corregiu els valors a partir dels criteris indicats. A continuació, resumiu en una taula la proporció d'estudiants per als valors positius (1) i els valors negatius (0) d'aquestes variables. Interpreteu breument.

Requisits:

- **La taula ha de contenir una variable a cada fila i quatre columnes: nombre d'estudiants amb valor 0 a la variable, nombre d'estudiants amb valor 1, proporció d'estudiants amb valor 0 i proporció d'estudiants amb valor 1.**

- **Es recomana generar la taula de forma automàtica, sense haver de fer el càlcul manualment per a cada variable. Podeu fer servir funcions de la família *apply* per automatitzar aquest càlcul.**

```
total_data <- nrow(data)
calculate_stats <- function(col_name){
  number_of_1 <- sum(col_name == 1, na.rm = TRUE)
  number_of_0 <- sum(col_name == 0, na.rm = TRUE)
  ratio_1 <- number_of_1 / total_data
  ratio_0 <- number_of_0 / total_data
  c(number_of_1, number_of_0, ratio_1, ratio_0)
```

```
}
```

En la cèl · la anterior, es mostra l'obtenció del nombre de files que hi ha en el fitxer csv i la definició d'una funció que ens permetrà realitzar el càlcul de les diferents estadístiques. El paràmetre *na.rm* s'estableix a *TRUE* ja que en les dades tenim valors del tipus *NA* i per tant si no establim aquest paràmetre a cert, els càlculs donaràn com a resultat *NA*. Aquest paràmetre establert a *TRUE* fa que s'ignorin els valors *NA*.

```
binary_columns <- data[, c("male","preschool", "expectBachelors", "motherHS",
                           "motherBachelors","motherWork", "fatherHS",
                           "fatherBachelors","fatherWork", "selfBornUS",
                           "motherBornUS", "fatherBornUS", "englishAtHome",
                           "computerForSchoolwork","schoolHasLibrary",
                           "publicSchool", "urban")]
```

Un cop executada l'última cèl · la, ja tenim seleccionades les columnes amb variables binàries a les quals volem aplicar els diferents càlculs.

```
calculations <- t(apply(binary_columns, 2, calculate_stats))
# 2 = Apply function in each column
```

Ara ja tenim els càlculs realitzats i en una taula, però si treiem per pantalla la taula, tal i com es mostra en la següent cèl · la, les columnes no tenen un nom que permeti identificar correctament el càlcul realitzat.

```
print(calculations)
```

```
##                       [,1] [,2]      [,3]       [,4]
## male                  1872 1791 0.5110565 0.48894349
## preschool             2607 1000 0.7117117 0.27300027
## expectBachelors       2830  771 0.7725908 0.21048321
## motherHS              3138  428 0.8566749 0.11684412
## motherBachelors       1137 2129 0.3104013 0.58121758
## motherWork            2622  948 0.7158067 0.25880426
## fatherHS              2937  481 0.8018018 0.13131313
## fatherBachelors       1027 2067 0.2803713 0.56429156
## fatherWork            2926  504 0.7987988 0.13759214
## selfBornUS            3347  247 0.9137319 0.06743107
## motherBornUS          2775  817 0.7575758 0.22304122
## fatherBornUS          2722  828 0.7431067 0.22604423
## englishAtHome         3131  461 0.8547639 0.12585313
## computerForSchoolwork 3236  362 0.8834289 0.09882610
## schoolHasLibrary      3406  114 0.9298389 0.03112203
## publicSchool          3421  242 0.9339339 0.06606607
## urban                 1410 2253 0.3849304 0.61506962
```

```
colnames(calculations) <- c("Number of 1", "Number of 0", "Poportion 1", "Proportion 0")
```

Si tornem a treure per pantalla la taula amb els càlculs, podrem observar que, ara sí, les columnes apareixen amb el nom que els hi pertoca.

```
print(calculations)
```

```
##                       Number of 1 Number of 0 Poportion 1 Proportion 0
## male                         1872        1791   0.5110565   0.48894349
## preschool                    2607        1000   0.7117117   0.27300027
## expectBachelors              2830         771   0.7725908   0.21048321
## motherHS                     3138         428   0.8566749   0.11684412
## motherBachelors              1137        2129   0.3104013   0.58121758
```

```
## motherWork                2622         948   0.7158067   0.25880426
## fatherHS                  2937         481   0.8018018   0.13131313
## fatherBachelors           1027        2067   0.2803713   0.56429156
## fatherWork                2926         504   0.7987988   0.13759214
## selfBornUS                3347         247   0.9137319   0.06743107
## motherBornUS              2775         817   0.7575758   0.22304122
## fatherBornUS              2722         828   0.7431067   0.22604423
## englishAtHome             3131         461   0.8547639   0.12585313
## computerForSchoolwork     3236         362   0.8834289   0.09882610
## schoolHasLibrary          3406         114   0.9298389   0.03112203
## publicSchool              3421         242   0.9339339   0.06606607
## urban                     1410        2253   0.3849304   0.61506962
```