Preprocessament de les dades

PID_00247918

Esteban Vegas Lozano

Temps mínim de dedicació recomanat: 3 hores

Universitat Oberta de Catalunya

Índex

Introducció

La base de qualsevol estudi estadístic és la informació recollida, és a dir, les dades disponibles per a realitzar el model i obtenir-ne les conclusions. Actualment es disposa d'una gran varietat de dades, amb formats diversos, que poden ser recopilades en diferentes fonts, com bases de dades, o captar-les de pàgines web. En la nostra societat és fàcil registrar-ho «gairebé» tot. Per exemple, podem tenir un rellote digital que pot monitoritzar les pulsacions o altres funcions fisiològiques molt fàcilment. Som a l'era de les dades (Mayer-Schönberger i Cukier, 2013) i, en particular, de les dades massives (*big data*).

Dades massives és una nova paraula de moda que té el seu sentit per la cada vegada més gran quantitat de dades que arriba a superar la capacitat del programari (*software*) convencional per a ser administrat i analitzat en un període de temps adequat. Es caracteritza per tenir les 5 V:

- **Volum**: La quantitat de dades generades i emmagatzemades és d'un ordre de magnitud de petabytes (10^{15} bytes) o superior.
- **Velocitat**: Es refereix a la velocitat a la qual les dades són creades, emmagatzemades i processades en temps real.
- **Varietat**: Es tenen dades de diferents formats, tipus i fonts. Poden ser dades estructurades, com són les bases de dades, o no estructurades, com correus electrònics, vídeos, imatges o blogs.
- **Variabilitat**: Amb tanta quantitat i varietat de dades poden aparèixer incoherències en el conjunt, fet que provocaria problemes en el maneig i l'anàlisi de les dades.
- **Veracitat**: El grau de fiabilitat de la informació recollida pot variar molt entre les diferents fontes d'informació.

Per tant, cada vegada cobra més importància la preparació de les dades abans de fer-ne l'anàlisi estadística (Pyle, 1999; Han i Kamber, 2001). Tenir més informació no implica millors resultats. La preparació de les dades es pot dividir en:

1) Llegir les dades. Passar la informació bruta recollida i carregar-la en el programari estadístic adequadament.

2) Neteja de les dades. Corregir errors, inconsistències i altres problemes que poden aparèixer en les dades.

3) Reducció de les dades. Pot ser que es tingui tanta informació que no sigui pràctic usar tota la que es té disponible. El problema es pot deure al següent:

- tenir molts registres,

- tenir moltes variables.

En el primer cas, s'han d'aplicar tècniques per a seleccionar els registres sense que s'eliminin registres importants. En el segon cas, s'ha de reduir el nombre de variables per mitjà, habitualment, de tècniques multivariants.

La preparació de les dades se sol considerar la fase menys atractiva i més avorrida del disseny experimental; la majoria dels científics de dades prefereixen la selecció del model i la seva validació. En canvi, és l'etapa a la qual es dedica una gran part del temps i esforç en funció de la qualitat i variabilitat de les dades originals. Sempre s'han de preparar les dades. No hi ha fonts d'informacions coherents, completes i veraces. No s'ha de saltar o reduir aquesta fase perquè pot tenir conseqüències molt greus en les conclusions, com biaixos o fins i tot conclusions equivocades.

En aquest mòdul es presenten les diverses parts del procés de preparació de dades una vegada que ja s'ha recopilat la informació. En particular, es focalitza el procés de neteja de les dades.

1. Lectura de la informació

En aquesta primera etapa es prepara la informació bruta. En el cas més simple està continguda en un fitxer tabular, en un objecte del programari estadístic amb l'estructura adequada per a fer-ne l'anàlisi estadística posterior.

En el nostre cas s'usa el programari R. Algunes de les accions són:

- Càrrega de dades en diferents formats.
- Verificació de la transmissió de la informació
- Canvi d'estructura de les dades.

1.1. Càrrega de dades en diferent formats

La informació recopilada pot ser molt variada i estar en diferents formats. Per exemple, format tabular amb delimitadors, com és el cas de fitxer del tipus csv, base de dades tipus SQL, documents Excel, pàgines web amb format XML, entre altres formats.

En la taula 1 es presenta informació sobre alguns dels paquets que es poden usar en R per a llegir segons la font de dades.

Taula 1. Paquets d'R per a llegir fitxers segons el format

Font de dades	Paquet	Funció	Comentaris
Fitxer de text amb camp fix	utils	read.fwf(). Crea un data frame	Els camps estan en posicions fixes en cada línia
Fitxer de text amb delimitadors (tabulacions, punt i coma...)	utils	read.table(). Llegeix el fitxer i crea un objecte data frame	Els camps estan en format tabular separat per delimitadors
Base de dades	RJDBC	dbConnect(). Estableix connexió dbReadTable(). Llegeix taula i crea un objecte data frame dbGetQuery(). Fa una consulta a la base de dades per SQL	Necessita la màquina virtual de Java i de driver de JDBC
Full de càlcul Excel	gdata	read.xls()	Necessita perl. Hi ha altres paquets
XML o JSON	XML o rjson	readHTMLList() readHTMLTable() fromJSON()	Són més difícils d'usar
SAS, SPSS, Minitab...	foreign	read.xport() (SAS) read.spss() (SPSS) read.mtp() (Minitab)	Hi ha funcions específiques per a cada cas
Web scraping	rvest	html()	Filtrar i analitzar fitxers html

En la majoria dels casos, el tipus d'objecte R que conté les dades és un `data frame`, una estructura d'R preparada per a representar les observacions com a files i les columnes com a variables.

Un error que se sol produir quan es treballa amb fitxers de text amb delimitadors és no posar el delimitador correcte i, per tant, la càrrega de dades es fa de manera equivocada. Vegem el cas de fitxers amb extensió csv (*comma-separated values*), és a dir, els valors se separen amb comes. Un document Excel es pot exportar com a extensió csv. En la configuració d'Excel amb idioma anglès es crea un fitxer amb el separador coma (,) per als valors i el punt (.) com a separador decimal. Aquest és el format natiu. En canvi, quan la coma s'usa com a separador decimal, tal como està la configuració d'Excel d'idioma espanyol, es crea un fitxer amb el separador punt i coma (;) per als valors y amb la coma (,) com a separador decimal. En la taula 2 es mostra un exemple amb les dues variants del format csv.

Taula 2. Exemple de fitxer amb format csv segons l'idioma

Format csv anglès	Format csv espanyol
5.1,3.5,1.4,0.2	5,1;3,5;1,4;0,2
4.9,3.3,1.4,0.2	4,9;3,3;1,4;0,2
4.7,3.2,1.3,0.2	4,7;3,2;1,3;0,2
4.6,3.2,1.5,0.2	4,6;3,2;1,5;0,2
5,3.6,1.4,0.2	5;3,6;1,4;0,2

Per a aquesta situació, hi ha una funció específica en R basada en la funció `read.table()` per a cada cas. La funció `read.csv()` està preparada per a llegir fitxers amb el format natiu: la coma (,) com a separador de valors i el punt (.) com a separador decimal. En canvi, `read.csv2()` s'usa quan el separador de valor és el punt i coma (;) i la coma (,) com a separador decimal.

1.2. Verificació de la transmissió de la informació

Una vegada que s'han carregat les dades en el programari estadístic, s'ha de verificar si s'ha transmès la informació de manera correcta. Una primera verificació bàsica és comprovar si el nombre d'observacions i de variables carregades en el programari estadístic coincideix amb el nombre de registres i camps de la base de dades (o arxiu). Si passa aquest primer filtre es poden triar diversos registres a l'atzar de la base de dades (o arxiu) i comprovar que coincideix amb les observacions carregades en el programari estadístic. És una manera de verificar si l'estructura de les dades és correcta.

Una vegada feta aquesta primera verificació, s'ha de revisar si cada variable carregada en el programari estadístic té el tipus de variable estadística adequada.

1.2.1. Tipus de variables estadístiques

Es poden classificar les variables des d'un punt de vista estadístic segons el tipus de contingut:

- Categòriques o qualitatives
  - Nominal
  - Ordinal

- Quantitatives
  - Discreta
  - Contínua

El grup de variables categòriques o qualitatives indica qualitats de la variable estudiada.

Exemple de variable categòrica o qualitativa

Un exemple senzill és la variable color, que pot tenir com a valor (o atribut) un color concret, per exemple groc. Els valors de la variable color no tenen un criteri d'ordre establert; per tant, es diu que és una variable categòrica nominal. Altres exemples d'aquest tipus són el sexe: home o dona, o el grup sanguini: O (universal), A, B, AB. Si la variable qualitativa té un criteri d'ordenació llavors es diu que és qualitativa ordinal. Per exemple, en la variable nivell d'estudis es pot plantejar una ordenació de menor a major amb aquests possibles atributs: estudis primaris, estudis secundaris o estudis universitaris. No cal que els atribus estiguin igualment separats. Per exemple, nivell de dolor: lleu, moderat o fort.

D'altra banda, les variables quantitatives tenen valors numèrics que sempre tenen un sentit d'ordre. En el cas de la variable quantitativa discreta, els seus valors presenten interrupcions en l'escala de valors que pot prendre. No hi ha valors intermedis entre dos valors consecutius. En canvi, la variable quantitativa contínua pot prendre qualsevol valor dins d'un interval específic. Per això, en el primer cas els valors són nombres naturals positius o negatius, i en el segon els valors tenen decimals.

Exemples de variables quantitatives

Exemples de variables quantitatives discretes són: el nombre de fills, el nombre de productes venuts en un mes o el nombre d'accidents en un mes; i quant a les variables quantitatives contínues: el pes, l'alçada, la facturació o el benefici.

A vegades, les variables quantitatives contínues es presenten sense decimals; per exemple, el pes d'una persona es pot recollir solament en quilograms. Això pot confondre i fer creure que la variable és discreta. Per evitar aquesta confusió s'ha de recordar la naturalesa de la variable, en aquest cas el pes, i observar que es pot mesurar amb més precisió, fins i tot en grams o unitats més petites.

El programari estadístic té una sèrie de tipus bàsics de variables per a contenir la informació. En la taula 3 es mostren els tipus bàsics de variables d'R i la seva associació amb el tipus de variable estadística.

Taula 3. Tipus bàsics de variables d'R

Tipus bàsic R	Tipus variable estadística
`numeric`	Dades numèriques: quantitativa contínua
`integer`	Valors sencers: quantitatius discrets
`factor`	Qualitatius nominals
`ordered`	Qualitatius ordinals
`character`	Dades de tipus caràcter
`logical`	Dos possibles valors: TRUE/FALSE
`raw`	Dades binàries

Per tant, cal revisar el fet que, en llegir les dades, s'assigni el tipus bàsic de variable R correctament. Per conèixer el tipus de variable en R s'usa la funció `class()`. Per exemple, si l'objecte R `mydata` conté les dades en l'estructura R `data frame`, un codi R per a mostrar els tipus de variables podria ser: `sapply(mydata,class)`. Aplica la funció `class()` a cada variable de `mydata` i n'obté un vector dels tipus de variables R per a cada variable.

1.2.2. Revisió descriptiva de la matriu de dades

Una manera senzilla d'inspeccionar la informació carregada és fer una estadística descriptiva simple de cada variable que ha de ser diferent segons el tipus de variable. En el cas de variables quantitatives es poden mostrar valors mínims, màxims i alguns quartils, en contrast amb les variables qualitatives que caldria esperar, una taula de freqüència (absoluta o relativa) per categoria. Aquesta informació s'usa per al següent:

- Revisar si el contingut de les variables carregades pel programari estadístic coincideix amb la font d'informació. Si tot ha anat bé, han de donar el mateix resultat.
- Obtenir una primera descripció de les dades.

Amb el programari R és molt fàcil obtenir aquesta informació; s'usa la funció `summary()`. Un exemple del resultat de l'objecte `mydata` amb 40 observacions i 5 variables és:

```
> summary(mydata)
      pes        sexe  grupsang       nivell.est          n.fills
 Min.   :53.82   H:21   A : 6      primaris     :14    Min.   :0.00
 1st Qu.:62.52   M:19   AB:12      secundaris   :10    1st Qu.:0.75
 Median :71.65          B :11      universitaris:16    Median :1.50
 Mean   :71.66          O :11                          Mean   :2.00
 3rd Qu.:80.50                                         3rd Qu.:4.00
 Max.   :88.56                                         Max.   :4.00
```

1.2.3. Variable categòrica oculta com a variable quantitativa

Una variable quantitativa pot emmascarar una variable categòrica quan s'usa com a atributs valors numèrics en variables qualitatives nominals. Per exemple, a vegades valors amb respostes sí o no de variables com *fuma* o *producte disponible* es codifica com a 1, si és sí (fuma o el producte està disponible) o 0, si és no (no fuma o producte no disponible). El programari R assigna per defecte el tipus `numeric` a aquest camp i per tant, es necessita corregir posteriorment al tipus `factor`.

A continuació, es presenta un codi R on la variable *fuma* usa com a atribut el valor 1 per a indicar sí i 0 per a indicar no. La funció `as.factor()` converteix a tipus factor on 0 i 1 són tractats com a atributs. La funció `factor()` crea una variable del tipus factor on es poden indicar les etiquetes per a cada nivell.

```
> # Presentar el contingut de la variable fuma
> mydata$fuma
 [1] 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
> # Quin tipus de variable és?
> class(mydata$fuma)
[1] "numeric"
> # Canviar a variable factor
> mydata$fuma <- as.factor(mydata$fuma)
> mydata$fuma
 [1] 1 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
Levels: 0 1
> # Quin tipus de variable és?
> class(mydata$fuma)
[1] "factor"
>
> # Canviar a variable factor amb etiquetes
> mydata$fuma <- factor(mydata$fuma, levels=c(0,1), labels=c("No", "Si"))
> mydata$fuma
 [1] Si Si Si No Si No No No No No No Si No No No No No No No No
Levels: No Si
> # Quin tipus de variable és?
> class(mydata$fuma)
[1] "factor"
```

1.2.4. Sistema de codificació de caràcters

Nosaltres ens comuniquem amb un alfabet que depèn de l'idioma. Per exemple, l'alfabet grec és diferent de l'alfabet anglès. L'ordinador, en canvi, es basa en un sistema binari amb dos possibles valors, 0 o 1, que s'agrupa en seqüències denominades *bytes*. Així que, per a comunicar aquests dos tipus de codis, s'apliquen els sistemes de codificació. Tradueixen cada caràcter d'un alfabet a una seqüència de sis bytes. Per exemple, ASCII és un sistema de codificació

que inclou les lletres majúscules i minúscules de l'alfabet llatí, els nombres aràbics del 0 al 9, signes de puntuació i alguns símbols especials com el control de carro o nova línia. Va ser un dels primers que hi va haver. Actualment hi ha molts sistemes de codificació, però Latin1 i UTF-8 són els majoritaris. El primer habitualment el trobem en aplicacions Windows i el segon en sistemes operatius basats en Unix.

En R es pot conèixer el sistema de codificació que aplica usant la funció `Encoding()` amb un caràcter que no sigui ASCII, com per exemple una lletra accentuada.

```
> Encoding("í")
[1] "latin1"
```

Si la resposta és «unknown», indica que el sistema de codificació usat és el del sistema operatiu. El sistema de codificació local d'R es pot esbrinar amb la funció `Sys.getlocale ("LC_CTYPE")`.

Si les dades provenen d'un sistema operatiu diferent del que s'analitza, possiblement pot aparèixer algun error de codificació. En castellà o català es pot apreciar que els caràcters no ASCII, com les lletres accentuades, són mostrades de manera estranya. És un error molest que pot provocar situacions anòmales.

1.3. Canvi d'estructura de les dades

A vegades la «forma» o estructura de les dades tal com es llegeix de les fonts d'informació no és l'adequada per a analitzar-les. Una situació comuna és passar d'un format llarg i estret a un format curt i ample. Un format llarg i estret apareix quan la informació recollida d'un mateix individu està disposada en diversos registres. Per exemple, recull informació del mateix individu en condicions o moments diferents. En l'exemple següent es presenten els valors obtinguts de quatre individus en tres condicions diferents, on hi ha un registre per cada individu i condició.

```
> mydata.llarg
    individu sexe  condicio valor
1          1    M     cond1   8.1
2          1    M     cond2  11.2
3          1    M     cond3  12.3
4          2    M     cond1   6.3
5          2    M     cond2   9.6
6          2    M     cond3  10.5
7          3    H     cond1   9.8
8          3    H     cond2  12.2
9          3    H     cond3  12.9
10         4    H     cond1  10.1
11         4    H     cond2  12.2
```

```
12         4    H     cond3  13.3
```

En passar a format curt i ample, cada fila és l'observació registrada d'un individu diferent i apareixen noves columnes. L'exemple anterior quedaria com segueix, amb la creació de tres noves variables: cond1, cond2 i cond3.

```
> mydata.ample
   individu sexe cond1 cond2 cond3
1         1    M   8.1  11.2  12.3
2         2    M   6.3   9.6  10.5
3         3    H   9.8  12.2  12.9
4         4    H  10.1  12.2  13.3
```

El programari R té diverses funcions per a passar d'un format llarg i estret al format curt i ample, i viceversa. En el paquet `reshape2` la funció `dcast()` s'usa per a la primera situació. El codi R de l'exemple mostrat anteriorment és:

```
mydata.ample <- dcast(mydata.llarg, individu + sexe ~ condicio, value.var="valor")
```

La funció `melt()` serveix per a convertir en la direcció oposada. Per a usuaris en R més avançats i que coneguin l'ecosistema tidyverse, es poden usar les funcions `spread()` i `gather()` del paquet `tidyr`.

2. Neteja de les dades

Una vegada recollida la informació, s'ha de netejar de possibles errors comesos durant el procés. Per a detectar els errors cal conèixer les característiques i especificacions de les dades llegides. La qualitat final de les dades és primordial per a fer-ne un bon estudi. A continuació, es descriuen possibles problemes que poden aparèixer segons el tipus de variable:

- **Variable categòrica, tant nominal com ordinal**. Errors sintàctics, com el fet que apareguin més categories que les esperades per errors en la introducció de la informació, com per exemple tres categories en la variable sexe. Per tant, cal tenir present sempre el nombre d'atributs de cada variable i els seus valors per a verificar els possibles errors. Un altre tipus d'error és el semàntic, per exemple, contradiccions entre edat i data de naixement (que es poden identificar en creuar la informació de variables), duplicacions, com pot passar amb les adreces postals o amb claus que han de ser úniques.

- **Variable quantitativa, tant discreta com contínua**. Cal verificar que la unitat de mesura sigui igual en tots els casos; per exemple, en el cas de la variable alçada, que sempre s'usi la mateixa magnitud, en metres amb dos decimals. Un problema més habitual és la detecció de valors atípics (*outliers*), és a dir, valors que estan allunyats de la variabilitat esperada. Un altre problema comú són els valors perduts (*missing*). A vegades són valors que corresponen al valor zero, i a vegades són conseqüència de no haver pogut recollir la informació de la variable d'interès.

En resum, en aquesta fase poden aparèixer problemes com els següents:

- Errors sintàctics i normalització en variables.
- Inconsistències en variables quantitatives.
- Valors atípics (*outliers*).
- Valors perduts (*missing*).

2.1. Errors sintàctics i normalització de variables

És bastant comú cometre petits errors sintàctics fàcils de corregir en variables qualitatives, encara que pot ser que es necessiti informació complementària per a decidir com es corregeix l'error. En canvi, en variables quantitatives es pot donar el cas de tenir valors amb diferents unitats i, per tant, cal corregir-lo fixant la mateixa unitat per a tots els valors. Per exemple, tots els valors de la variable alçada en metres. Finalment, pot caldre fer transformacions de les variables quantitatives abans d'usar-les en l'anàlisi estadística.

Enumerem alguns casos segons el tipus de variable:

1) Variables qualitatives:

a) Estandarditzar les variables a minúscules o majúscules. En el llenguatge R es pot usar la funció `tolower()` o `toupper()`, respectivament. Per exemple,

```
> tolower("Cotxe")
[1] "cotxe"
> toupper("COTXE")
[1] "COCHE"
```

b) Treure espais abans i després del text o treure caràcters especials com el retorn de carro o tabulació. En el llenguatge R es pot utilitzar la funció `trimws()`. Per exemple:

```
> trimws("  El cotxe blau   \t")
[1] "El cotxe blau"
```

c) Eliminar accents. Si les paraules tenen accent es pot donar el cas que, en alguns registres, estiguin accentuades i en uns altres no. Per a solucionar el problema es poden accentuar tots els casos o treure els accents. En R es pot aplicar la funció `iconv()`. Per exemple, treure l'accent de «María» (en castellà).

```
> iconv("María", to="ASCII//TRANSLIT")
[1] "Maria"
```

d) Revisió de categories errònies per errors sintàctics. Es pot detectar si es revisa la taula de freqüències de cada categoria. En R es pot obtenir la taula de freqüències amb la funció `table()`. Un exemple senzill és tenir tres categories en la variable sexe quan se n'esperen dos.

```
> table(sexe)
sexe
 H  M  N
20 14  1
```

e) Poden aparèixer errors més complicats que requereixen analitzar el contingut de tots els valors per a verificar si compleix cert patró i, en cas contrari, poder ser rectificat. El programari R té un gran nombre de funcions per a l'anàlisi de cadenes de caràcters («string»). En els casos més complicats es crea una expressió regular per a construir el patró i després aplicar la funció més adequada per a la situació que s'ha de resoldre. Per exemple, es pot verificar que el DNI està format per vuit dígits i una lletra que pot estar en majúscules o minúscules. El codi R podria ser:

```
> expressio_regular <- "^([[:digit:]]{8})+([[:alpha:]])$"
```

```
> grep(expressio_regular, c( "12345678a", "12345678I", "123456789","123456789B" ))
[1] 1 2
```

En l'objecte R `expressio_regular` es té el patró que s'ha de verificar i amb la funció `grep()` es comprova. En aquest cas, solament el primer i el segon DNI compleixen el patró.

Hi ha diversos paquets d'R per a manejar variables tipus *string*. En la taula 4 es mostren diverses funcions del paquet `stringr` que serveixen per a corregir els errors anteriorment indicats.

Taula 4. Algunes funcions per al maneig d'*strings* del paquet `stringr`

Tasca	Funció
Estandarditzar a majúscules/minúscules/títol	`str_to_upper(),str_to_lower(), str_to_title()`
Eliminar accents	`stri_trans_general()`
Recodificar	`stri_encode()`
Eliminar espais en blanc abans o/i després de la paraula	`trimws()`
Detecció de seqüència de caràcters («string»)	`str_detect()`
Extracció/reemplaçament d'string	`str_extract()`
Separació d'string	`str_split()`
Omplir string	`str_pad()`
Separació de paraules	`word()`

2) Variables quantitatives:

a) Confusions amb separadors decimals.

- Barrejar la coma decimal i el punt decimal en la mateixa variable. Per exemple, tenir valors com 3.4; 3,2; 3,6; 4,7; 3.5 on els valors estan separats per punt i coma, però els valors 3.4 i 3.5 haurien de ser 3,4 i 3,5, respectivament.
- Tenir un separador decimal diferent en les variables. Aquest cas és similar a l'anterior, però ara una variable pot tenir com a separador decimal la coma i una altra variable el punt. És un cas estrany però possible.
- Confondre la coma (,) com a separador decimal quan és separador d'unitats de mil. Això pot ocórrer amb el format anglès de nombre. Per exemple: 23,456.

b) Estandarditzar les variables a les mateixes unitats. Si les dades s'han recollit amb diferents unitats de mesura, s'ha d'estandarditzar a una unitat concreta. Per exemple, les vendes de productes han d'estar recollides en diferents unitats de temps: per dia, per setmana o per mes. S'ha de seleccionar una unitat de mesura i estandarditzar les dades a aquesta unitat.

c) Estandarditzar les variables de data i temps. Les variables que continguin data o/i temps són variables quantitatives amb un format especial; per exemple, la data pot tenir el format dia/mes/any i el temps el format hora:minut:segon. S'ha de verificar que tots els valors tenen el mateix format i, en cas contrari, corregir-lo. En R existeix el tipus `POSIXlt` i `POSIXct` per a dates o/i temps i `Date` solament per a dates sense el temps amb moltes funcions associades. Un primer pas que s'ha de tenir en compte és canviar de tipus `character` al tipus adequat. La funció `as.Date()` o `as.POSIXct()` converteix les dates o el temps del tipus `character` al tipus `Date` o `POSIXct`, respectivament. Un exemple d'aquesta conversió és:

```
> ## data en format 'dia/mes/año'
> dates <- c("03/11/17", "07/05/16", "14/02/97", "28/10/99", "02/08/95")
class(dates)
[1] "character"
> ## Transformar el vector character amb el format 'dia/mes/año' a tipus Date
> dates_1 <- as.Date(dates, "%d/%m/%y")
> class(dates_1)
[1] "Date"
> dates_1
[1] "2017-11-03" "2016-05-07" "1997-02-14" "1999-10-28" "1995-08-02"
```

L'objecte R `dates` conté un vector de dates de tipus `character`. Amb la funció `as.Date()` la converteix al tipus `Date` amb el format de data any-mes-dia.

El paquet `lubridate()` té nombroses funcions per a facilitar les operacions bàsiques amb variables de tipus data o/i temps. El codi següent produeix el mateix resultat anterior però usant la funció `dmy()`. La funció espera que l'ordre de la data sigui dia, mes i any.

```
> library(lubridate)
> dates_3 <- dmy(dates)
> class(dates_3)
[1] "Date"
> dates_3
[1] "2017-11-03" "2016-05-07" "1997-02-14" "1999-10-28" "1995-08-02"
```

Aquest altre codi R mostra la potència de la funció `dmy()` per a extreure la data.

```
> same_date <- c("20/10/2017", "20 Oct 17", "La data és 20-10-2017", "20-10-2017")
> dmy(same_date)
```

```
[1] "2017-10-20" "2017-10-20" "2017-10-20" "2017-10-20"
```

En l'objecte R `same_date` es presenta la mateixa data amb l'ordre dia, mes i any en diferents situacions. La funció `dmy()` extreu la data de manera correcta. A més, hi ha més funcions per a la resta d'ordenacions possibles de dia, mes i any: `dmy()`, `mdy()`, `myd()`, `ymd()`, `ydm()`, `dym()`.

3) Transformació de les variables:

a) Normalització. És quan es modifiquen les variables quantitatives perquè siguin comparables entre si. La transformació més comuna és la normalització «z-score». La nova variable, *z*, es calcula amb l'expressió següent:

$$z_i = \frac{x_i - \bar{x}}{s}$$

on *x* és la variable original, $\bar{x}$ és la seva mitjana aritmètica i *s* és la seva desviació típica. Té com a propietats que el seu valor mitjà és 0 i la seva desviació típica és 1. Una altra transformació és la denominada normalització «min-max». La nova variable, *z*, s'obté aplicant l'expressió següent:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

on min(*x*) és el valor mínim de *x* i max(*x*) és el valor màxim de *x*. La nova variable es mou entre 0 i 1.

Hi ha moltes més funcions per a transformar les variables.

b) Discretització (*binning*). A vegades, la informació és recollida amb molt «soroll». Això vol dir que el valor real de la variable quantitativa pot ser bastant diferent de la dada recollida; per tant, interessa reduir aquest «soroll». Una possible solució és discretitzar. Així, una sèrie de valors són els representants de la variable quantitativa. Un exemple d'aplicació de la discretització en la representació gràfica és l'histograma.

El procés de discretització té dues fases:

- Ordenar els valors i fer la partició. Algunes formes de crear la partició són:
  - Intervals iguals (*equal-width bins*): Es divideix el rang de valors en *N* intervals iguals de grandària (max –min)/N.
  - Aproximadament igual nombre de mostres (*equal-depth bins*). Es divideix el rang de valors en *N* intervals que contenen, aproximadament, el mateix nombre de mostres.

- Suavitzar cada partició. Per a fer-ne la suavització es pot aplicar algun d'aquests criteris:

- El valor mitjà. Se substitueix cada valor de la partició pel seu valor mitjà.
- La mediana. Se substitueix cada valor de la partició per la seva mediana.
- Pels límits. Se substitueix cada valor de la partició pel valor més proper als seus valors extrems.

Per a il·lustrar com funciona el procés de discretització s'usen les dades següents ja ordenades:

12, 14, 16, 17, 18, 19, 22, 22, 23, 25, 25, 26, 27,27, 33, 34, 36,36

En primer lloc es crea una partició, per exemple de tres grups (*bins*), que queden com a:

- Cas d'intervals iguals:
  - Grup 1 ([12-20]): 12, 14, 16, 17, 18, 19
  - Grup 2 ([20-28]): 22, 22, 23, 25, 25, 26, 27, 27
  - Grup 3 ([28-36]): 33, 34, 36, 36

- Cas d'igual nombre de mostres:
  - Grup 1: 12, 14, 16, 17, 18, 19
  - Grup 2: 22, 22, 23, 25, 25, 26
  - Grup 3: 27, 27, 33, 34, 36, 36

A continuació, s'apliquen els criteris de suavització. Solament es presenta el cas de la partició amb igual nombre de mostres:

- Criteri del valor mitjà:
  - Grup 1: 16, 16, 16, 16, 16, 16
  - Grup 2: 24, 24, 24, 24, 24, 24
  - Grup 3: 32, 32, 32, 32, 32, 32

- Criteri de la mediana:
  - Grup 1: 16, 16, 16, 16, 16, 16
  - Grup 2: 23, 23, 23, 23, 23, 23
  - Grup 3: 33, 33, 33, 33, 33, 33

- Criteri dels límits:
  - Grup 1: 12, 12, 19, 19, 19, 19
  - Grup 2: 22, 22, 22, 26, 26, 26
  - Grup 3: 27, 27, 36, 36, 36, 36

Es pot observar que el resultat final de la discretització pot canviar notòriament segons el mètode aplicat.

4) Duplicació de registres:

Habitualment es deuen a errors en el procés d'adquisició de la informació. Es poden donar duplicacions de registres pels motius següents:

- Tenir més d'un registre amb el mateix identificador. Per tant, cal fusionar o seleccionar els valors per a aconseguir tenir un sol registre.
- Tenir més d'un registre amb identificadors diferents per al mateix individu. Cal comparar els valors de camps bàsics per a identificar els registres. Per exemple, si són clients es poden comparar els cognoms o la seva localització per si apareixen valors iguals o semblants.

2.2. Inconsistències en variables quantitatives

Són errors que es poden haver ocasionat en el moment de la introducció de la informació i són fàcilment recognoscibles:

- Valor invàlid. Els valors de la variable han d'estar inclosos en un determinat interval de valors. Per exemple, en la variable edat no poden aparèixer valors negatius, o en una variable data no tots els valors són possibles.
- Verificar la consistència del registre en creuar informació de diverses variables. Per exemple:
  - Si es té la data de naixement i l'edat, es pot comprovar que el valor de l'edat coincideixi amb el càlcul de l'edat a partir de la data de naixement.
  - En cas de tenir variables amb resultats parcials i la variable total, es pot confirmar que el còmput global de les variables parcials és igual a la variable total. Per exemple, si es tenen les variables de facturació de cada producte i una variable facturació total, la suma de la facturació de tots els productes ha de ser igual a la variable facturació total.
  - Un altre cas és tenir variables amb l'adreça postal i el codi postal. Es pot comprovar que l'adreça postal correspon al codi postal registrat.

2.3. Valors atípics (*outliers*)

Una definició general de *valor atípic* és aquella observació (o grups d'observacions) que semblen ser inconsistents amb el conjunt de les dades (Barnett i Lewis, 1994).

És una idea senzilla però que és molt difícil de precisar per a cada situació, ja que s'hauria de conèixer la distribució teòrica de les variables. Valors detectats com a valors atípics poden ser correctes. S'ha de meditar eliminar-les en l'anàlisi, ja que és una decisió estadística. Es poden donar dues situacions:

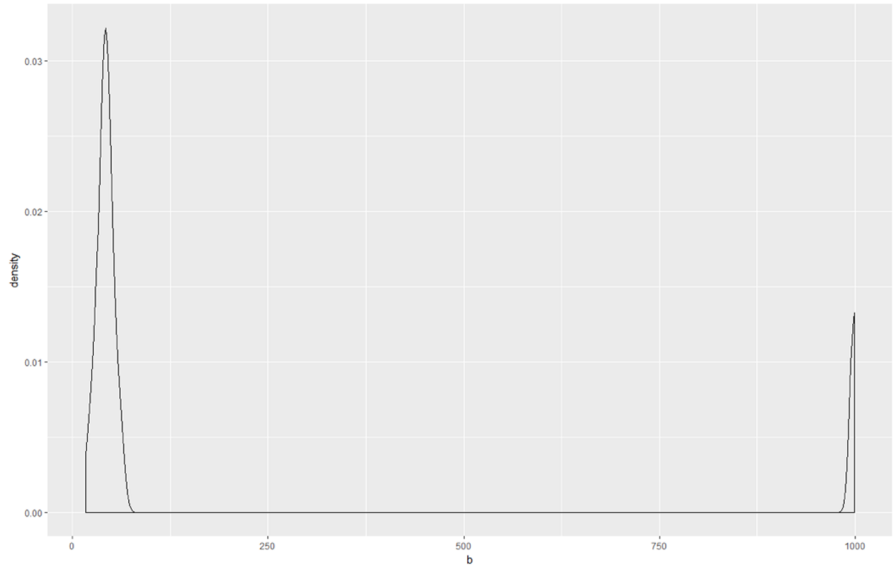
- Valor sentinella
- Valor atípic pròpiament

2.3.1. Valor sentinella

En alguns casos la detecció de valor atípic correspon a valors sentinelles, és a dir, a valors que són usats per a representar alguna situació especial en una variable numèrica. Per exemple, «valor desconegut» o «no aplicable». Sol ser un valor numèric amb totes les xifres igual a 9. Per exemple, si fos la variable edat podria ser un valor com 999 per a indicar edat desconeguda. Encara que també pot ser un valor que estigui fora del rang dels valors possibles. Per exemple, el valor -1 en la variable edat.

Una manera visual de detectar valors sentinella amb valors extrems és representar la distribució de valors de la variable. Si hi ha valors sentinella s'ha de presentar un cert pic en l'extrem de la distribució dels valors. A continuació, es presenta un codi en R que genera 50 valors sota normalitat que poden correspondre a edats. Posteriorment, s'afegeixen 10 valors igual a 999 que serveixen per a indicar «edat desconeguda». Una vegada obtinguda la sèrie d'edats, es representa la seva distribució.

```
> library(ggplot2)
> # Generar valors
> b <- trunc(rnorm(50,40,10))
> # Afegir 10 valors iguals a 999
> b <- c(b, rep(999,10))
> b
[1]  36  33  58  34  41  30  50  43  36  35  33  48  40  25  77  48  36  44  46  21
[21]  49  45  33  42  48  48  30  30  36  50  35  43  36  46  52  34  26  31  41  32
[41]  50  47  53  48  41  43  24  53  31  17 999 999 999 999 999 999 999 999 999 999
> # Representar la distribució de les dades
> ggplot(mapping= aes(x=b))+ geom_density()
```



Queda molt clar en el gràfic que hi ha dos pics, el proper a 1.000 correspon al valor sentinella.

També es pot fer una estadística descriptiva i observar quin valor és el màxim.

```
> summary(b)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  14.00   35.75   43.00  200.30   52.50  999.00
```

El codi R per representar «valor desconegut o no disponible» en una variable quantitativa s'escriu `NA` (*not available*). Un cas diferent és `NaN` (*not a number*), que apareix quan es fa alguna operació indefinida com 0/0, ∞-∞ i ∞/∞. Per tant, cal substituir el valor sentinella per `NA`.

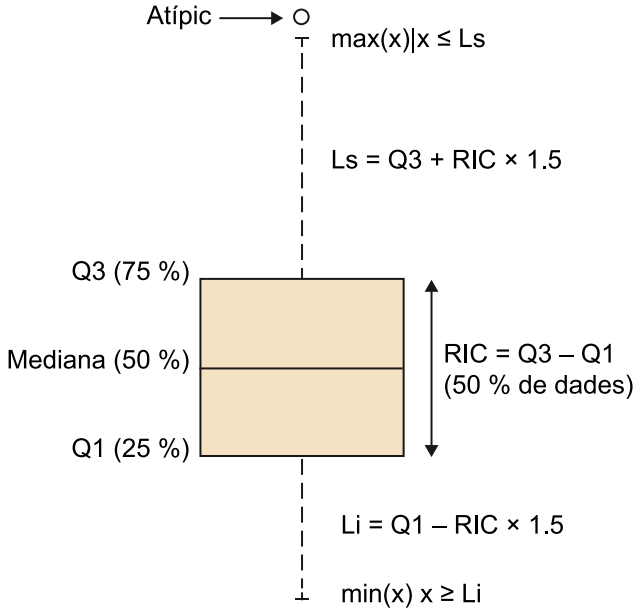
```
> val.centinela <- 999
> # Sustituir el valor centinela per NA
> b[b== val.centinela]<- NA
> # verificar resultat
> summary(b)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  15.00   32.25   38.00   37.94   43.75   55.00      10
```

En resum, en la majoria dels casos els valors sentinella que són detectats com a valor atípic corresponen a valors perduts (*missing*).

2.3.2. Valor atípic pròpiament

Una manera senzilla de veure si hi ha valors atípics és fer un diagrama de caixa (*box plot*). És un gràfic (figura següent) basat en els valors quartils on apareix una caixa central i dos segments anomenats «bigotis».

Diagrama de caixa o *box plot*



Font: Wikipedia

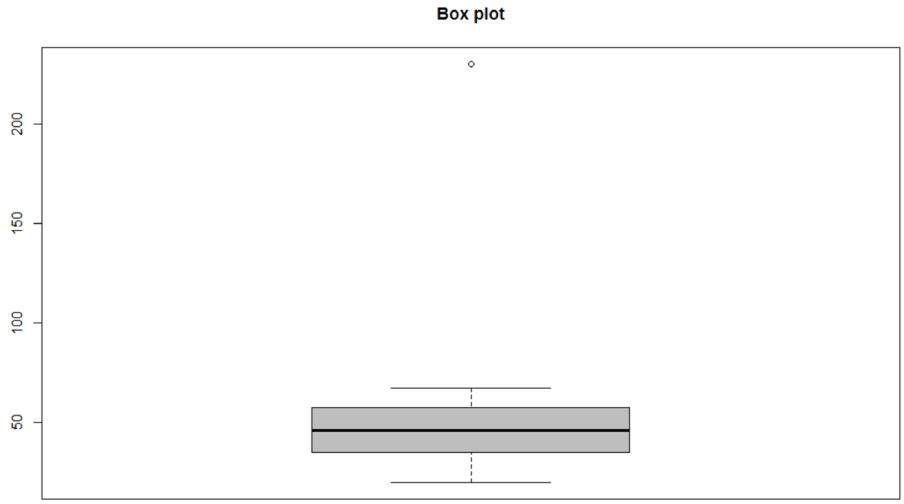
La caixa central s'estén del quartil 1 (Q1), que correspon al percentil 25 %, fins al quartil 3 (Q3), que correspon al percentil 75 %. La diferència entre Q3 i Q1 es denomina *rang interquartílic* (RIC).

Els valors atípics es representen com a punts extrems allunyats dels «bigotis» o segments exteriors a la caixa. Els «bigotis» s'estenen fins als valors mínim i màxim de la sèrie de dades o fins a 1,5 vegades el RIC. Per tant, són valors inferiors a Q1 – 1,5 * RIC o superiors a Q3 + 1,5*RIC. Per exemple, si tenim els següents valors ordenats que corresponen a les edats dels empleats d'una empresa:

20 21 25 32 33 37 40 41 44 45 47 49 50 53 55 60 61 62 67 230

el valor 230 és un clar candidat a valor atípic. El codi R següent mostra com obtenir el diagrama de caixa i els valors atípics.

```
> edat<- c(20, 21,  25, 32, 33, 37, 40, 41, 44, 45,  47, 49, 50, 53, 55, 60, 61, 62, 67, 230)
> boxplot(edat,main="Box plot", col="gray")
```




```
> boxplot.stats(edat)$out
[1] 230
```

La funció `boxplot.stats()` mostra les característiques del diagrama de caixa i, en particular, els valors atípics.

Una vegada detectat el valor atípic, es pot considerar producte d'un error i intentar corregir-lo. Per exemple, en el cas de 230, pot ser un error per haver concatenat el dígit 0 a l'edat 23. A vegades el valor és canviat per «no disponible» (NA) i, a vegades, es queda tal com està ja que pot ser un valor real però molt poc freqüent.

Existeix un subcamp de l'estadística denominat *estadística robusta* (Huber, 1981; Leroy, 1987; Maronna i altres, 2006), que desenvolupa estimadors i models estadístics alternatius a l'estadística clàssica però amb la condició que els estimadors siguin resistents (robustos) a aquest efecte dels valors atípics i els models estadístics puguin admetre certes desviacions en les condicions d'aplicabilitat. A continuació, es presenten alguns estimadors robustos de tendència central i dispersió.

L'estimador de tendència central clàssic és la mitjana aritmètica. Té molt bones propietats; és un estimador no esbiaixat, però la seva estimació està molt influïda pels valors atípics. Si es calcula la mitjana aritmètica de l'exemple anterior de la sèrie d'edats, s'obté un valor de 53,6, que és molt superior a la mitjana aritmètica sense considerar el valor 230, que és de 44,32. En canvi, el valor de la mediana és de 46 i sense considerar el valor de 230, la mediana queda en 45, la diferència és molt petita. La mediana és un dels estimadors robustos més coneguts per a mesurar la tendència central de les dades. Un altre estimador robust per a mesurar la tendència central és la mitjana retallada (*trimmed mean*). Es basa a eliminar el mateix percentatge, k %, dels valors extrems, i calcular la mitjana aritmètica de la resta de valors. Per exemple, la mitjana retallada del 5 % de la sèrie d'edats anterior correspon a eliminar el valor més petit, 20, i el valor més gran, 230, i calcular la mitjana aritmètica de

la resta de valors. Una variant de la mitjana retallada és la mitjana winsoritzada (*winsorized mean*). Els valors extrems eliminats són substituïts per altres valors. El cas més senzill és substituir pel valor més petit (o més gran) que ha quedat. Així la mitjana winsoritzada del 5 % substitueix el valor de 20 pel valor més petit que queda en la sèrie, 21, i el valor 230 pel valor més gran que queda en la sèrie, 67, i posteriorment calcula la mitjana aritmètica. Una altra opció és substituir els valors eliminats pels quantils. En aquest cas, la mitjana winsoritzada del 5 % substitueix el valor de 20 pel quantil del 5 %, 20,95, i el valor 230 pel quantil del 95 %, 75,15, i es calcula la mitjana aritmètica. Es pot observar que la mitjana retallada o winsoritzada del 50 % és equivalent al càlcul de la mediana.

Per mesurar la dispersió de les dades l'estimador més habitual és la desviació estàndard, que no és un estimador robust. La desviació estàndard de l'exemple previ amb dades d'edats és 43,67, però si es torna a calcular sense el valor 230 s'obté 13,89, un valor molt inferior a l'obtingut amb el valor atípic. El rang interquartílic (RIC) i la desviació absoluta respecte de la mediana (DAM) són alternatives robustes a la desviació estàndard. El RIC és la diferència entre el quartil 3 (percentil 75 %) i el quartil 1 (percentil 25 %) de les dades. És la grandària de la caixa del gràfic *box plot*. El valor de RIC de les dades d'edats és 20,25, no gaire diferent del valor de RIC sense el valor 230, que és 19. La DAM és un estimador anàleg de la desviació estàndard que usa la mediana en lloc de la mitjana aritmètica com a valor central de les dades; es calcula el valor de la mediana de la diferència dels valors absoluts entre les dades i la seva mediana.

$$DAM(i) = mediana(|y_i - mediana(y)|)$$

El valor de DAM amb les dades d'edats és 16,31, i sense el valor 230, és 14,83.

En la taula 5 es presenten algunes funcions d'R per a aconseguir estimacions robustes de tendència central i dispersió. A més, es mostra el resultat obtingut amb l'exemple de dades d'edat per a les diferents funcions.

Taula 5. Funcions d'R per a calcular estimacions robustes de tendència central i dispersió

Estimador	Funció	Exemple	Resultat
Mitjana aritmètica	`mean()`	`mean(edat)`	`53,6`
Mediana	`median()`	`median(edat)`	`46`
Mitjana retallada	`mean(x, trim=)`	`mean(edad,trim=0.05)`	`45,667`
Mitjana winsoritzada	`winsor.mean(x, trim=)` (Paquete psych)	`winsor.mean(edat, trim= 0.05)`	`45,905`
Desviació estàndard	`sd()`	`sd(edad)`	`43,667`
Rang interquartílic (RIC)	`IQR()`	`IQR(edad)`	`20,25`

Estimador	Funció	Exemple	Resultat
Desviació absoluta respecte de la mediana (DAM)	`mad()`	`mad(edad)`	`16,309`

2.4. Valors perduts (*missing*)

Moltes vegades la informació recollida no és completa. Falten alguns valors en certes observacions. Aquest problema és un dels més comuns i apareix molt sovint quan la informació es recull a partir d'enquestes.

Una vegada detectada la presència de dades perdudes (no observades o *missing*), cal decidir què fer. L'opció més senzilla seria eliminar les observacions que tinguessin dades perdudes, però això ocasiona un desaprofitament de la informació recopilada amb un augment de l'error estàndard de les estimacions dels paràmetres. Si són poques observacions les afectades en relació amb el total d'observacions, és una opció admissible, però en cas contrari s'ha d'estimar un valor plausible per a omplir el buit que deixa un valor no observat. Aquest procés es denomina *imputar*.

Referències bibliogràfiques

Existeix una àmplia bibliografia sobre imputació: Rubin (1976), Little i Rubin (1987), Allison (2001), De Waal, Pannekoek i Scholtus (2011).

2.4.1. Imputació de valors

L'elecció de la tècnica d'imputació depèn de la relació entre les dades perdudes i les dades. El cas més senzill és que les dades perdudes no estiguin relacionades amb cap variable present o no en les dades, és a dir, completament a l'atzar. La imputació més senzilla és

$$\hat{y}_i = \bar{y}$$

on la mitjana de les dades observades ($\bar{y}$) s'usa per a fer la imputació del valor perdut ($\hat{y}_i$). No provoca biaix en la tendència central de les dades, però sí en la mesura de dispersió. La implementació en R d'aquest procediment és molt senzilla:

```
> y[is.na(y)] <- mean(y, na.rm = TRUE)
```

També es pot optar per substituir la mitjana aritmètica per estimadors robustos de la tendència central com la mediana, mitjana retallada o mitjana winsoritzada.

Un segon model d'imputació és el denominat *ratio imputation*.

$$\hat{y}_i = \hat{R} x_i$$

on x_i és una covariable de y_i y $\hat{R}$ és una estimació del quocient de la mitjana entre x i y, que es pot obtenir com a suma de valors observats de y dividit per la suma dels corresponents valors de x. Aquest model té la propietat que si $x = 0$ el valor d'imputat és $\hat{y} = 0$. Per tant, pot ser un model adequat si es té com a restriccions, $x \geq 0$ i/o $y \geq 0$. El seu codi en R és:

```
> I <- is.na(y)
> R <- sum(y[!I])/sum(x[!I])
> y[I] <- R * x[I]
```

Un tercer model més elaborat és imputar els valors mitjançant un model de regressió lineal (o generalitzat)

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \cdots + \hat{\beta}_k x_{k,i}$$

on $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$ són les estimacions dels coeficients de regressió de cada covariable $x_1, x_2 \ldots, x_k$. Cal tenir en compte els problemes que poden existir en usar el model de regressió, com per exemple colinealitat i valors influents per a aplicar adequadament la imputació. Per exemple, en el codi R següent la funció `lm()` prediu la regressió lineal usant solament els registres complets. Posteriorment, reconeix quins registres són perduts i fa la seva predicció amb la funció `predict()`.

```
> lineal.model <- lm(y ~ x1 + x2 + x3 + x4, data= mydata)
> I <- is.na(y)
> y[I] <- predict(lineal.model, newdata = mydata[I, ])
```

Per finalitzar, es presenta un model més elaborat basat en distàncies entre registres, la imputació basada en els k veïns més propers («kNN-imputation»). La idea és trobar els k registres més propers al registre amb un o més valors perduts. Llavors, un valor (o el resultat d'una funció) dels k registres més propers serveix per a fer la imputació. Si definim la distància de Gower entre dos registres i i j, $d_g(i, j)$, com a:

$$d_g(i, j) = \frac{\sum_l w_{ijl} d_l(i, j)}{\sum_l w_{ijl}}$$

on l indica la l-èsima variable i $d_l(i, j)$ és la distància entre el valor de la variable l en el registre i i el registre j. Per a variables qualitatives, quan el valor per a la l-èsima variable és el mateix en el registre i que en j llavors $d_l(i, j) = 0$, en cas contrari és 1. En canvi, per a variables quantitatives, la distància es defineix com a $_{\mathrm{d}\,l}(i, j) = 1 - (x_i - x_j) / (\max(x) - \min(x))$.. Quan en la variable l-èsima del re-

gistre *i* o *j* és un valor perdut llavors el pes és $w_{ijl}=0$, en cas contrari és 1. El paquet `VIM` d'R té la funció `kNN()` usant la distància de Gower. Un exemple senzill és:

```
> library(VIM)
> mydata.complet <- kNN(mydata)
```

La funció `kNN()` obté els *k* (per defecte són 5) registres més propers al registre amb valors perduts. En variables qualitatives, l'atribut més freqüent en els *k* registres més propers s'usa per a fer la imputació. Si la variable és quantitativa, és el valor de la mediana dels *k* registres més propers.

3. Reducció de les dades

Algunes vegades, es té tanta quantitat d'informació que és intractable fer-ne l'anàlisi estadística.

> L'objectiu de la reducció de les dades és extreure una representació reduïda de tota la informació disponible que sigui tractable i que contingui les característiques de la totalitat de la informació.

El problema es pot dividir en les possibilitats següents:

- tenir una gran quantitat de registres o instàncies,
- tenir moltes variables o característiques.

3.1. Reducció del nombre de registres

Actualment, amb els mitjans tecnològics de què es disposa, registrar informació de forma «quasi» contínua és relativament fàcil. Això pot provocar tenir un immens nombre de registres. Una de les solucions habituals és extreure una mostra representativa de tots els registres disponibles. Les formes d'extracció més senzilles són:

- **Mostreig aleatori sense reemplaçament**. És escollir els registres amb igual probabilitat del total de registres. Un registre que s'ha extret no pot ser obtingut de nou.
- **Mostreig aleatori amb reemplaçament**. És escollir registres amb igual probabilitat del total de registre, però amb la diferència que quan s'extreu el registre torna de nou al conjunt i, per tant, es pot tornar a extreure.

En llenguatge R està la funció `sample()` per a fer aquests tipus de mostrejos. El procés es basa a fer el mostreig dels identificadors de tots els registres disponibles. Posteriorment, se seleccionen els registres complets per a fer-ne l'estudi estadístic. En l'exemple següent la variable `myregistres` conté els identificadors dels registres; en aquest cas senzill són els valors de l'1 al 100. El nombre de mostres que es vol extreure és 10. El resultat del mostreig aleatori sense reemplaçament es guarda en la variable `mas` i el resultat del mostreig aleatori amb reemplaçament es guarda en la variable `mas.R`.

```
> myregistres <-1:100
> mes<-sample(myregistres,10)
> mes
 [1] 87 45 62 47 11 76  3  6 75 17
```

```
> mes.R <-sample(myregistres,10, replace=TRUE)
> mes.R
 [1] 78 62 38 22 29 46 70 28 22 48
```

3.2. Reducció del nombre de variables

Haver compilat moltes variables sobre el problema que s'investiga no implica obligatòriament que sigui més fàcil de ser resolt. Hi pot haver moltes variables que no siguin informatives però, a més, la gestió és més complicada i algunes tècniques estadístiques no es poden aplicar. Hi ha dues possibles solucions:

1) Selecció de variables o característiques (*feature selection*)

2) Extracció de característiques (*feature extraction*)

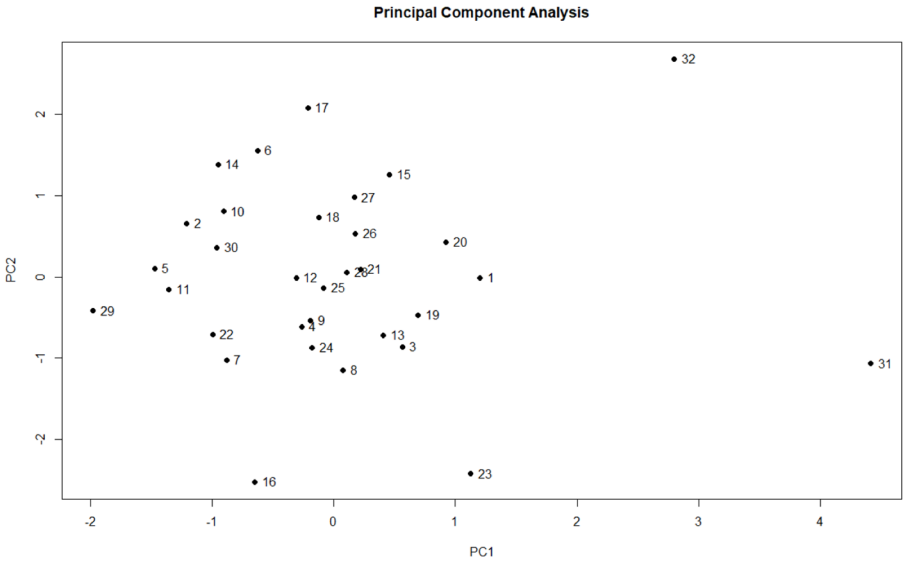
La primera es basa a escollir un grup de variables originals (també anomenades *característiques* o *atributs* des del món del *machine learning*) que contingui la major part de la informació rellevant per a resoldre el problema que s'ha de tractar. Hi ha moltes metodologies i és un camp de recerca molt important (Stanczyk i Jain, 2014).

La segona solució transforma l'espai de variables original d'alta dimensió (moltes variables) en un espai de menor dimensió, és a dir, aplica una reducció de la dimensió. Hi ha moltes tècniques de reducció de la dimensió (Everit i Dunn, 2010). Es basen a crear noves variables que siguin funció de les variables originals i que compleixin certes condicions per a retenir la màxima informació possible de les variables originals. El nombre de noves variables és molt menor que de variables originals. En molts casos, n'hi ha prou amb dos o tres noves variables. Per tant, les noves variables extreuen la informació de les variables originals i les substitueixen per a les anàlisis posteriors.

La tècnica principal en la reducció de la dimensió és l'anàlisi de components principals (ACP) o *principal component analysis* (PCA), en anglès. Es basa a construir una transformació lineal de les variables originals de tal manera que les noves variables tinguin variàncies cada vegada menors i estiguin incorelacionades entre si. Des del punt de vista algebraic, l'ACP descompon en autovalors (*eigenvalues*, en anglès) de la matriu de covariàncies (o correlacions). Amb les noves variables, denominades *components principals*, es poden projectar les observacions en un espai de dues o tres dimensions que recull la major part de la variància de les dades originals. D'aquesta manera es poden visualitzar les dades en el pla (dues dimensions) o en l'espai (tres dimensions) fàcilment. És una tècnica exploratòria bàsica que ens pot donar una idea de les relacions que poden tenir les observacions. Un cas especial és quan un punt (o diversos) estan allunyats del núvol de punts general. Aquest punt (o punts) són candidats a ser valors atípics. Un exemple d'aquesta situació es mostra en el codi R inferior.

En el llenguatge R la funció `prcomp()` fa l'ACP. A continuació, es mostra un codi R on es generen 32 observacions amb 5 variables. Les primeres 30 observacions són valors que segueixen una distribució normal de mitjana 0 i desviació típica 1. En l'observació 31 totes les seves variables segueixen una distribució normal de mitjana 2 i desviació típica 1. Finalment, en l'observació 32 les tres primeres variables segueixen una distribució normal de mitjana 2 i desviació típica 1, i les dues últimes variables segueixen una distribució normal de mitjana 0 i desviació típica 1. Per tant, la majoria d'observacions, 30, segueixen la mateixa distribució, solament hi ha dues observacions amb un comportament diferent. L'última observació té una distribució més semblant a la majoria d'observacions que la penúltima. Per tant, en la visualització de les observacions, en fer una ACP, s'observa un núvol dels 30 primers punts propers entre ells i els dos últims punts més allunyats del núvol de punts. El més allunyat és l'observació 31, amb una distribució de les variables més diferent de la majoria d'observacions. A més, el primer eix, PC1, és el que mostra més variabilitat, mentre que el segon eix, PC2, molta menys.

```
> # Creació de la mostra
> set.seed(998)
> # Primeres 30 observacions
> mat0 <- matrix(rnorm(150), 30, 5)
> dimnames(mat0) <- list(paste("Observació", 1:30), paste("Var", 1:5))
> # Observacio 31
> mat1 <- matrix(rnorm(5,2), 1, 5)
> dimnames(mat1) <- list(paste("Observació", 31), paste("Var", 1:5))
> #Observacio 32
> mat2 <- matrix(c(rnorm(3,2),rnorm(2)), 1, 5)
> dimnames(mat2) <- list(paste("Observació", 32), paste("Var", 1:5))
> # Agrupació de les 32 Observacions
> mat <- rbind(mat0,mat1,mat2)
>
> # Càlcul PCA
> myPCA <- prcomp(mat, scale. = T, center = T)
>
> # Representació del PCA
> plot(myPCA$x[,1:2], main= "Principal Component Analysis", pch=19)
> text(myPCA$x[,1:2], labels=1:32, pos=4)
```


Principal Component Analysis
PC1
PC2
-2
-1
0
1
2
3
4
32
17
6
14
15
27
10
18
2
26
20
30
5
12
21
1
11
25
29
19
9
22
13
24
3
7
8
31
16
23

Resum

En aquest mòdul es mostra la importància que té la preparació de les dades abans de l'anàlisi estadística. És una tasca poc glamurosa i divertida, encara que és bàsica per a poder aprofitar tota la informació disponible i no cometre errors en les conclusions. La quantitat de temps necessari per a la preparació de dades depèn directament de la salut de les dades. I en la majoria dels casos sempre hi ha algun tipus de problema, habitualment associat a errors o falta d'informació.

S'ha dividit la preparació de les dades en tres parts:

1) Llegir les dades.

2) Netejar les dades.

3) Reduir les dades.

La lectura de les dades té com a objectiu bàsic transmetre la informació al programari estadístic sense cometre errors i adequar-lo al programari; un punt important que s'ha de revisar és si s'ha assignat a cada variable el tipus de variable estadística adequada: quantitativa discreta (o contínua) o qualitativa nominal (o ordinal). La neteja de les dades és el punt fonamental de la preparació de les dades. Poden aparèixer:

- errors sintàctics i de normalització en variables,
- inconsistències entre variables,
- valors atípics i
- valors perduts o no observats.

Els valors atípics es poden detectar realitzant diagrama de caixa (*box plot*) i intentant minimitzar-ne els efectes amb l'aplicació d'estimadors robustos. Pel que fa als valors perduts, hi ha una gran quantitat de mètodes per a intentar imputar els valors que falten i aconseguir registres complets. Finalment, pot ser necessari reduir les dades. En el cas de tenir un gran nombre de registres, s'extreu una mostra representativa del total de registres. En canvi, si el problema és el gran nombre de variables, es poden aplicar tècniques de reducció de la dimensió, com l'ACP, per a extreure les característiques de les dades i substituir les variables originals en l'anàlisi estadística.

Bibliografia

Allison, P. (2001). *Missing values*. Thousand Oaks, CA: Sage Publications.

Barnett, V.; Lewis, T. (1994). *Outliers in Statistical Data* (3a ed.). Nova York: John Wiley & Sons.

De Waal, T.; Pannekoek, J.; Scholtus, S. (2011). *Handbook of statistical data editing and imputation. Wiley handbooks in survey methodology*. Nova York: John Wiley & Sons.

Everitt, B. S.; Dunn, G. (2010). *Applied Multivariate Data Analysis*. Nova York: John Wiley & Sons.

Han, J.; Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers, Inc.

Huber, P. (1981). *Robust Statistics*. Wiley.

Little, R.; Rubin, D. (1987). *Statistical Analysis with Missing Data*. Nova York: John Wiley & Sons.

Maronna, R. A.; Martin, R. D.; Yohai, V. J. (2006). *Robust statistics: Theory and methods*. Wiley.

Mayer-Schönberger, V.; Cukier, K. (2013). *Big Data. A Revolution That Will Transform How We Live, Work and Think*. RU: John Murray.

Pyle, D. (1999). *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc.

Rousseeuw, P. J.; Leroy, A. M. (1987). *Robust regression and outlier detection*. Nova York: John Wiley & Sons.

Rubin, D. (1976). *Inference and missing data* (vol. 3, núm. 63, pàgs. 581-592). Biometrika.

Stanczyk, U.; Jain, L. C. (2014). *Feature Selection for Data and Pattern Recognition*. Springer.