Anàlisi de dades i estadística descriptiva amb R i R-Commander

Daniel Liviano Solís
Maria Pujol Jover

PID_00208267

UOC
Universitat Oberta de Catalunya

Índex

Introducció

Aquest mòdul té dos grans objectius. D'una banda, el primer capítol vol introduir l'anàlisi estadística descriptiva utilitzant R-Commander, la qual cosa inclou la introducció i importació de dades, la creació i transformació de variables, el càlcul d'estadístics bàsics(univariants i multivariants), i l'elaboració de gràfics. Aquests continguts corresponen, aproximadament, a les assignatures introductòries d'estadística cursades a la UOC.

D'altra banda, els capítols segon i tercer aprofundeixen en les possibilitats que ofereix R per a fer anàlisis estadístiques descriptives sense fer servir R-Commander, és a dir, només amb codi. Evidentment, és una anàlisi més complexa però ofereix moltes més possibilitats, tant en el càlcul d'estadístics i l'extracció d'informació estadística en general com en el camp de l'anàlisi gràfica.

Objectius

1. Ser capaç d'introduir dades i variables directament.
2. Conèixer els càlculs principals d'estadística descriptiva i saber implementar-los amb R-Commander.
3. Saber escollir en cada cas el tipus de gràfic necessari segons l'anàlisi que es porti a terme.
4. Poder exportar els resultats obtinguts amb R i amb R-Commander en diferents formats.

1. Estadística descriptiva amb R-Commander

1.1. Iniciar sessió amb R-Commander

Comencem recordant que per a iniciar R-Commander hem d'iniciar una sessió d'R, i tot seguit introduir en la consola la instrucció següent:

```
> library(Rcmdr)
```

És important que el programa R-Commander ja estigui instal·lat en l'ordinador[1]. Si tot és correcte, obtindrem una finestra amb la interfície R-Commander, tal com es mostra en la figura 1:

Figura 1: Interfície d'R-Commander

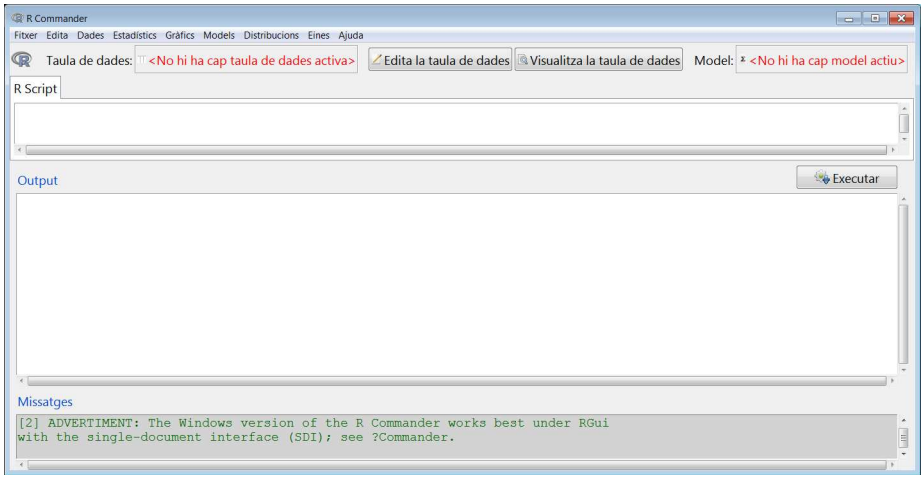


Figura 1

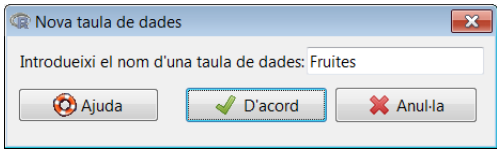
Quan apareix la finestra d'R-Commander, la sessió d'R continua oberta, de manera que totes dues finestres es mantenen obertes i operatives simultàniament.

1.2. Introducció de dades

Una de les opcions que té l'usuari és introduir manualment les dades de l'anàlisi. En aquest cas, la ruta que cal seguir serà la següent:

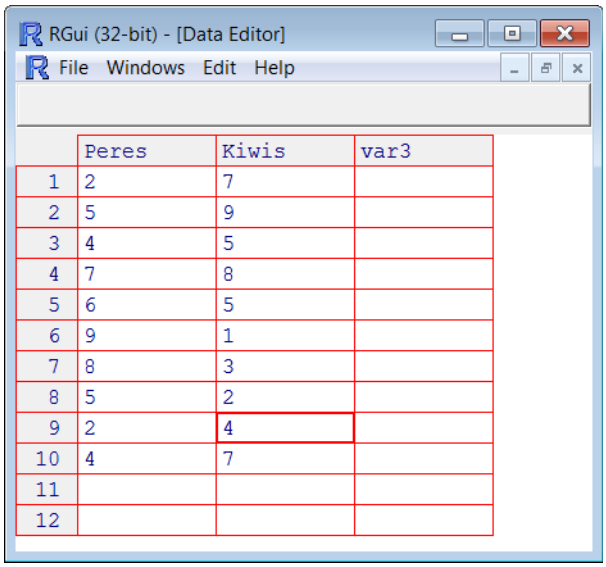
Dades / Nova taula de dades

Aleshores apareixerà la finestra següent, en la qual haurem d'introduir el nom que vulguem donar a la nostra taula de dades (per exemple, *Fruites*):



[1] El primer mòdul explica detalladament com es fa la instal·lació d'R-Commander.

Un cop introduït el nom de la taula de dades, premem *D'acord*. Tot seguit, apareixerà un full de càlcul amb cel·les buides en què haurem d'introduir les nostres variables en columnes, especificant també el nom de cada variable en la primera fila. En el nostre exemple, crearem dues variables fictícies: *Peres* i *Kiwis*. Per a introduir els noms en cada columna, premerem els encapçalaments *var1, var2,...* i hi introduirem el nom que vulguem en cada cas, després premerem l'opció *numeric* o *character*, en funció de si es tracta d'una variable numèrica o de text. Els valors s'introdueixen amb el teclat:

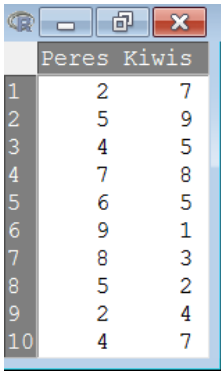


	Peres	Kiwis	var3
1	2	7	
2	5	9	
3	4	5	
4	7	8	
5	6	5	
6	9	1	
7	8	3	
8	5	2	
9	2	4	
10	4	7	
11			
12			

Introduir i desar dades

Un cop hem introduït les variables en columnes, i els hem donat un nom en la primera cel·la de cada columna, les dades es desen simplement tancant aquesta finestra. Si volem tornar a accedir a aquest editor de dades, simplement hem d'accedir a l'opció *Edita la taula de dades*.

Quan tinguem creada la nostra taula de dades, simplement tanquem la finestra, i les variables amb els seus valors quedaran desades. En el moment en què vulguem veure les dades en memòria, ho farem amb l'opció *Visualiza la taula de dades*. En el nostre cas obtindríem la finestra següent:



	Peres	Kiwis
1	2	7
2	5	9
3	4	5
4	7	8
5	6	5
6	9	1
7	8	3
8	5	2
9	2	4
10	4	7

Visualitzar les dades

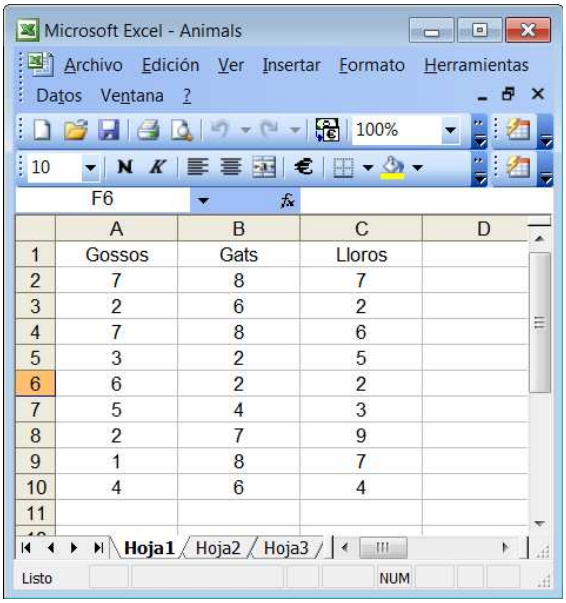
Aquesta finestra no es pot editar ni modificar, ja que simplement mostra les variables que componen la taula de dades.

De la mateixa manera, podem introduir i/o modificar dades amb l'opció *Edita la taula de dades*.

1.3. Importació de dades

Una altra opció que ens ofereix R-Commander és importar dades d'un arxiu extern, és a dir, creat prèviament amb qualsevol programa. Hi ha diferents maneres de fer

aquesta operació. En aquest manual, ens limitem a explicar la manera més fàcil i directa. Suposem que les dades originals estan en format Excel (amb extensió *xls*), i que les variables estan disposades en columnes, amb la primera fila reservada per al nom de cada variable. En aquest cas, tenim 9 observacions i 3 variables: *Gossos*, *Gats* i *Lloros*:



	A	B	C	D
1	Gossos	Gats	Lloros	
2	7	8	7	
3	2	6	2	
4	7	8	6	
5	3	2	5	
6	6	2	2	
7	5	4	3	
8	2	7	9	
9	1	8	7	
10	4	6	4	
11				

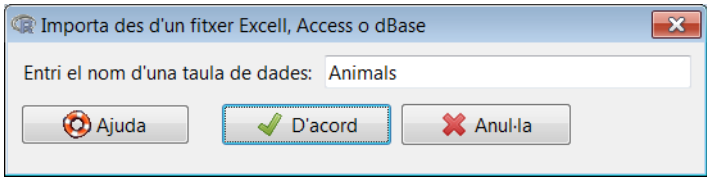
Importar dades amb decimals

Segurament tindrem el programa Excel configurat perquè faci servir les comes com a separadors de decimals. Això no representa cap problema, R-Commander convertirà automàticament aquestes comes en punts, ja que, com s'ha comentat, R fa servir el punt com a separador decimal.

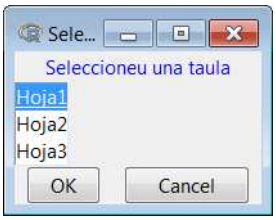
Per a carregar aquestes dades des d'R-Commander, haurem de seguir la ruta següent:

Dades / Importa dades / des d'una taula de dades d'Excel, Access o dBase...

Igual que abans, apareixerà una finestra que ens preguntarà el nom que volem donar a la nostra taula de dades. En aquest exemple, l'anomenarem *Animals*:



En prémer *D'acord*, apareixerà la pantalla següent, en què hem d'especificar la ruta d'accés a l'arxiu Excel amb les dades originals, i també en quin full són les dades que volem utilitzar:



Compte amb els fulls en documents d'Excel!

Molts cops ens trobarem amb arxius d'Excel amb més d'un full amb dades. És fonamental tenir-los identificats per a fer aquest pas correctament i no importar dades que no corresponen.

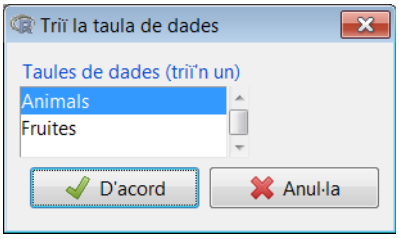
Ara podem visualitzar, com hem fet abans, la taula de dades que acabem d'importar per a comprovar que s'han carregat correctament:

Animals

	Gossos	Gats	Lloros
1	7	8	7
2	2	6	2
3	7	8	6
4	3	2	5
5	6	2	2
6	5	4	3
7	2	7	9
8	1	8	7
9	4	6	4

1.4. Anàlisi descriptiva

Sovint tindrem diferents taules de dades carregades, però R només ens permet mantenir-ne un d'activa, que és la taula amb la qual treballarem. Per tant, el primer pas a l'hora de fer una anàlisi descriptiva és activar una taula de dades entre les que hàgim importat o introduït. Això ho farem prement *Taula de dades* i triant-ne una. En el nostre exemple, treballarem amb la taula de dades *Animals*:



Un primer grup d'estadístics bàsics de les variables incloses en una taula de dades és el que en ofereix el menú desplegable següent:

Estadístics / Resums / Taula de dades activa

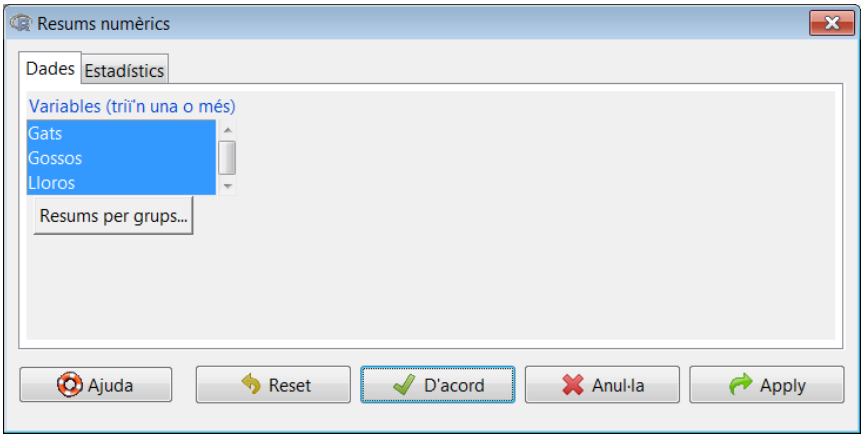
El resultat que s'obté en la consola es mostra a continuació. Veiem que inclou els valors mínim i màxim de cada variable, la mitjana aritmètica i els tres quartils (el segon és la mitjana):

```
> summary(Animals)
     Gossos           Gats           Lloros
 Min.   :1.000   Min.   :2.000   Min.   :2
 1st Qu.:2.000   1st Qu.:4.000   1st Qu.:3
 Median :4.000   Median :6.000   Median :5
 Mean   :4.111   Mean   :5.667   Mean   :5
 3rd Qu.:6.000   3rd Qu.:8.000   3rd Qu.:7
 Max.   :7.000   Max.   :8.000   Max.   :9
```

Sovint no en tindrem prou amb els estadístics bàsics i voldrem obtenir mesures addicionals com l'asimetria, la curtosi, el coeficient de variació, la desviació típica o alguns quantils. Per a això hi ha una opció en la qual es pot escollir entre una conjunt d'estadístics. Per a accedir a aquesta opció, la ruta que hem de seguir serà la següent:

Estadístics / Resums / Resums numèrics

Obtindrem el menú següent, en el qual seleccionarem els estadístics que vulguem obtenir entre les variables que ens interessin.



Càlcul de quantils

Per a calcular diferents quantils, hem d'introduir específicament els que volem calcular, tenint en compte que el separador de decimals és el punt i que els diferents quantils se separen amb comes. En aquest exemple, calculem el mínim (0), els tres quartils $(0,25,\ 0,5$ i $0,75)$ i el màxim (1).

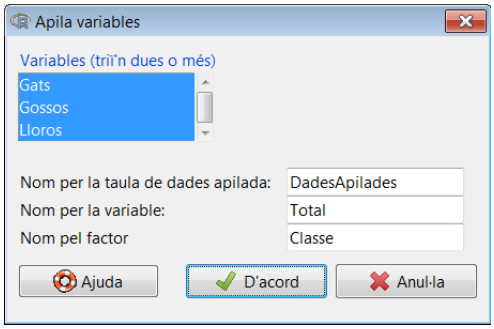
El resultat que s'obté en la finestra de resultats es mostra a continuació:

```
> numSummary(Animals[,c("Gats", "Gossos", "Lloros")],
+ statistics=c("mean", "sd", "quantiles"),
+ quantiles=c(0,.25,.5,.75,1))
           mean        sd 0% 25% 50% 75% 100% n
Gats    5.666667 2.449490  2   4   6   8    8 9
Gossos  4.111111 2.260777  1   2   4   6    7 9
Lloros  5.000000 2.449490  2   3   5   7    9 9
```

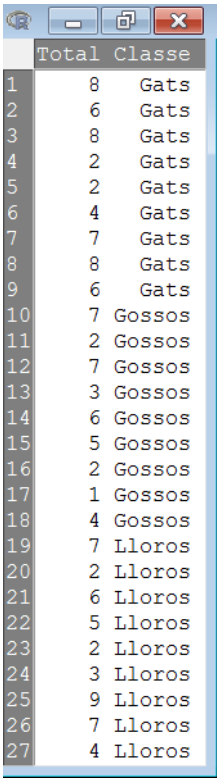
S'ha de dir que, de vegades, no tenim les dades disposades de la manera adequada per a obtenir les anàlisis que volem. Suposem que volem agrupar les tres variables en una de sola, fent que tingui $9 \times 3 = 27$ observacions, i que volem crear una variable qualitativa associada (o factor) que indiqui, per a cada una de les 27 observacions, quina classe d'animal és. En R-Commander les instruccions més senzilles que es poden seguir per a dur-ho a terme són les següents:

Dades / Taula de dades activa / Apilar variables de la taula de dades activa

A continuació apareixerà el quadre de diàleg següent, que ens permetrà donar nom a la nova taula de dades, a la variable agrupada i al factor. La nova taula de dades l'anomenarem *DadesApilades*, la nova variable amb 27 observacions s'anomenarà *Total*, i el factor, que és una variable qualitativa amb caràcters, s'anomenarà *Classe*, i especificarà per a cada observació quin animal és.



Si visualitzem la nova taula de dades amb l'opció *Visualiza la taula de dades*, veiem que consta de dues variables: una de numèrica (*Total*) i el factor associat (*Classe*):



	Total	Classe
1	8	Gats
2	6	Gats
3	8	Gats
4	2	Gats
5	2	Gats
6	4	Gats
7	7	Gats
8	8	Gats
9	6	Gats
10	7	Gossos
11	2	Gossos
12	7	Gossos
13	3	Gossos
14	6	Gossos
15	5	Gossos
16	2	Gossos
17	1	Gossos
18	4	Gossos
19	7	Lloros
20	2	Lloros
21	6	Lloros
22	5	Lloros
23	2	Lloros
24	3	Lloros
25	9	Lloros
26	7	Lloros
27	4	Lloros

Crear noves taules de dades

Cal tenir en compte que, en fer aquesta operació, creem una nova taula de dades diferent de l'anterior. En aquest cas, la taula *DadesApilades* només tindrà dues variables: una amb les observacions (*Total*) i una altra amb el tipus d'animal que és (*Classe*). **Cal tenir sempre present quina de les dues taules de dades està activa**, ja que els càlculs que efectuem només s'aplicaran sobre la taula de dades que estigui activa.

Vegem les estadístiques bàsiques d'aquestes noves variables, i com hi ha 9 animals de cada classe. Recordem la ruta que hem de seguir per a obtenir aquest resultat:

Estadístics / Resums / Taula de dades activa

```
Total           Classe
 Min.   :1.000   Gats  :9
 1st Qu.:2.500   Gossos:9
 Median :5.000   Lloros:9
 Mean   :4.926
 3rd Qu.:7.000
 Max.   :9.000
```

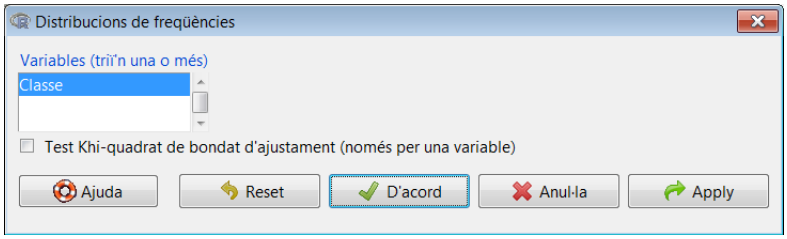
Resum de factors

Fixem-nos que, per a la variable *Classe*, que és un factor amb caràcters, aquesta funció ens calcula el nombre d'observacions, en aquest cas 9 observacions de cada tipus d'animal.

Si el que volem és una taula de freqüències del factor (variable qualitativa) que acabem de crear, utilitzarem la ruta següent:

Estadístics / Resums / Distribucions de freqüències

Com podem veure, R-Commander ens ofereix l'opció de marcar la possibilitat de realització del test Khi-quadrat de bondat de l'ajust, el qual s'estudiarà més endavant:



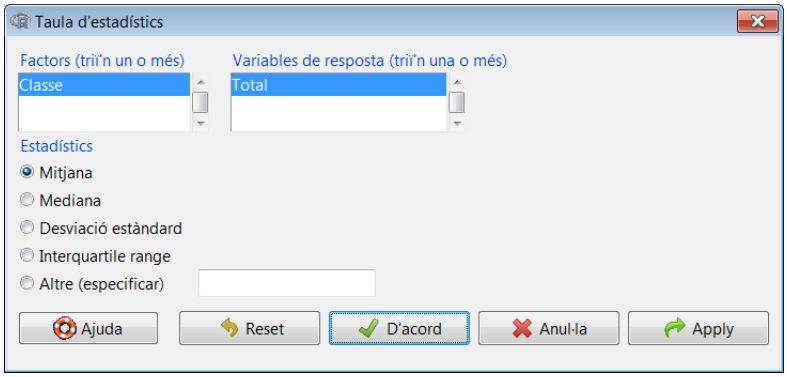
Això és el que obtenim en la finestra de resultats. Veiem que conté la freqüència absoluta (9 animals de cada classe) i relativa (un 33% de cada classe):

```
> .Table <- table(DadesApilades$Classe)

> .Table  # counts for Classe

  Gats Gossos Lloros
     9      9      9

> round(100*.Table/sum(.Table), 2)  # percentages for Classe

  Gats Gossos Lloros
 33.33  33.33  33.33

> remove(.Table)
```

També és possible obtenir estadístics específics de la variable apilada segons el factor. Això es fa seguint aquesta ruta:

Estadístics / Resums / Taula d'estadístics

Amb això obtenim un menú en què s'ha d'escollir la variable d'interès, el factor (en aquest cas només hi ha la classe d'animal), i l'estadístic que ens interessa. Per exemple, calcularem la mitjana aritmètica:



Diferents factors per a una variable

En aquest menú hem d'escollir la variable a analitzar i els factors. És possible que tinguem més d'un factor associat a una variable, amb la qual cosa és important no confondre's i escollir el que ens interessa.

Veiem que el resultat ens mostra com la mitjana aritmètica és superior per a les observacions que corresponen als gats:

```
> tapply(DadesApilades$Total, list(Classe=DadesApilades$Classe),
+ mean, na.rm=TRUE)
Classe
    Gats   Gossos   Lloros
5.666667 4.111111 5.000000
```

1.5. Anàlisi gràfica

Abans de res, cal deixar clar que el menú desplegable d'R-Commander només ens ofereix una part minúscula de tot el potencial d'R pel que fa a anàlisi gràfica. Tot i així, R-Commander és més que suficient per a cobrir l'anàlisi que es fa en els cursos d'estadística en els graus de la UOC.

1.5.1. Histograma

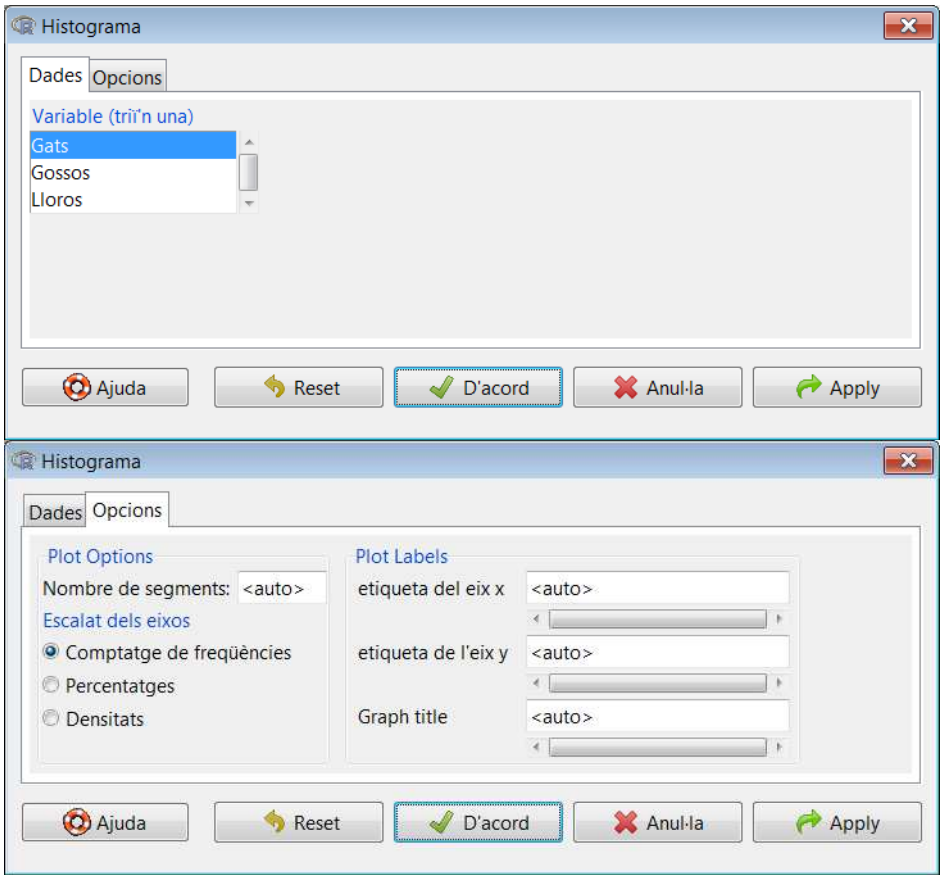
Començarem l'anàlisi amb l'histograma, que és una representació gràfica d'una variable contínua en forma de barres verticals, en què la superfície de cada barra és proporcional a la freqüència dels valors representats. En el nostre exemple, calcularem l'histograma de la variable *Gats*. El primer pas serà activar la taula de dades *Animals*, i després accedirem a la ruta següent del menú desplegable:

Gràfics / Histograma

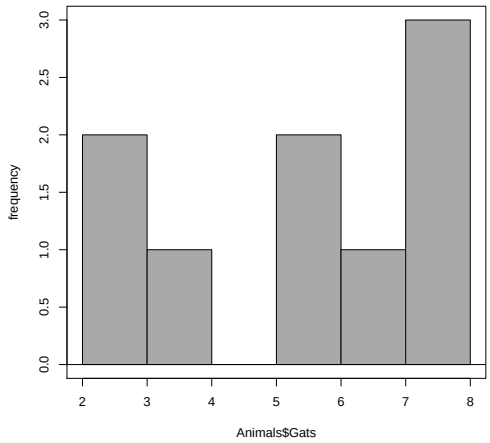
En fer-ho apareixerà un menú en què haurem d'escollir la variable a representar i el tipus d'histograma, això és, si volem freqüències absolutes (les quals sumarien 9), percentatges o densitats.

Histograma i gràfic de barres

Tot i que s'assemblen, aquests dos tipus de gràfic no són idèntics. L'histograma es fa servir per a representar **dades quantitatives contínues**, mentre que el **gràfic de barres** es fa servir per a representar gràficament dades quantitatives discretes o dades qualitatives.



El resultat, amb les freqüències absolutes, es mostra a continuació. És important tenir en compte que el gràfic apareix en la consola d'R i no en la finestra d'R-Commander, de manera que haurem d'acudir a la sessió inicial d'R per a visualitzar-lo.



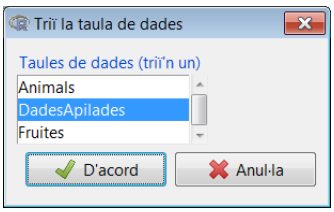
Desar i exportar gràfics

En la consola original d'R, un cop seleccionada la finestra del gràfic, si anem al menú *Archivo*, es desplegarà un menú amb opcions per a desar el gràfic en un document. Veurem que podem escollir entre diferents formats (EPS, JPG, PDF, etc.).

1.5.2. Diagrama de barres

Com s'ha comentat més amunt, el diagrama de barres es fa servir amb variables discretes o qualitatives, com són els factors. En el nostre exemple, aquest gràfic ens mostrarà quantes observacions estan incloses en cada una de les categories del factor. Abans de

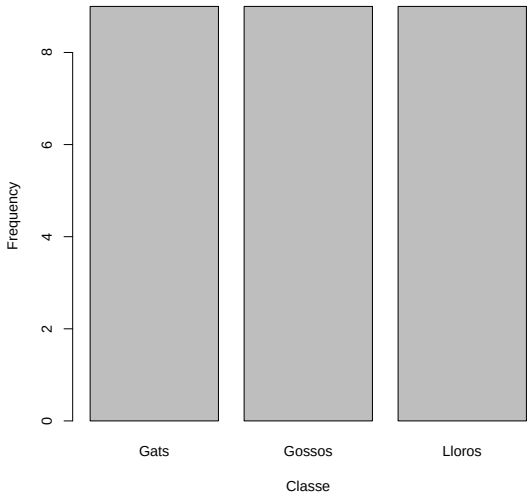
procedir, és important canviar la taula de dades actiu, això és, activar la taula *DadesApilades*:



Ara es pot procedir accedint a la ruta següent del menú desplegable:

Gràfics / Gràfic de barres

Com podem veure, hi ha 9 observacions per a cada una de les categories del factor *Classe*.



Opcions del menú no disponibles

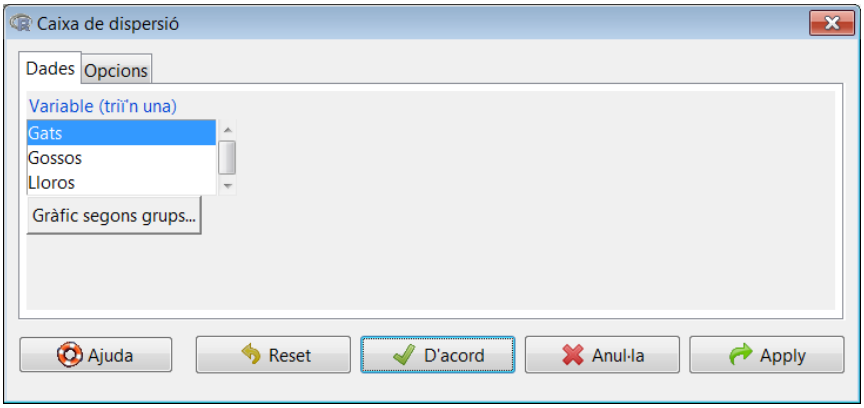
Si intentéssim calcular un diagrama de barres mentre la taula de dades *Animals* està activa, l'opció del menú *Gràfics / Gràfic de barres* estaria desactivada, ja que en aquesta taula no hi ha cap factor per a ser representat.

1.5.3. Diagrama de caixa

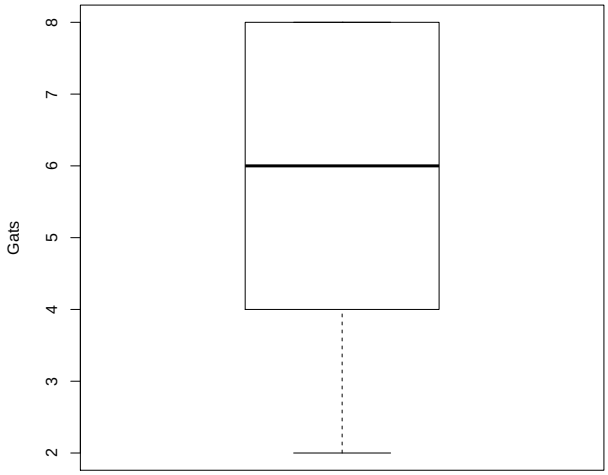
El diagrama de caixa proporciona informació sobre els valors mínim i màxim, els quartils i possibles valors atípics, a més de la simetria de la distribució de les dades. Calcularem aquest gràfic de la variable *Gats*, de manera que hem de tornar a canviar la taula de dades activa a *Animals*. Un cop fet això, la ruta que hem de seguir és aquesta:

Gràfics / Caixa de dispersió

Ens apareixerà un quadre de diàleg en què hem de triar la variable per a la qual volem calcular el diagrama de caixa:



En el gràfic resultant podem veure els tres quartils, més el màxim i el mínim, i com la distribució mostra una asimetria negativa.

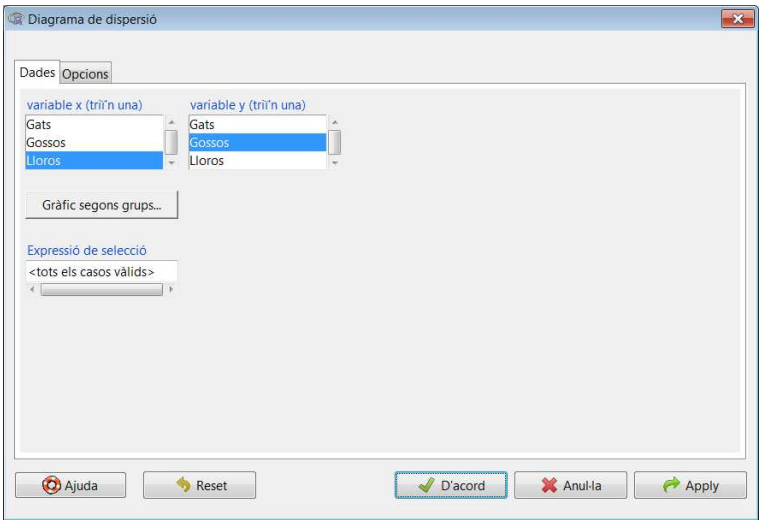


1.5.4. Diagrama de dispersió

Per acabar, veurem com es calcula un diagrama de dispersió per a dues variables, consistent en coordenades cartesianes en què cada observació és un punt diferent, i la posició dels quals la determinen dos valors: un en l'eix horitzontal i un altre en l'eix vertical, és a dir, un per a cada variable. La disposició dels punts revelarà si hi ha algun tipus de correlació o no entre totes dues variables. Per exemple, calcularem aquest diagrama per a les variables *Gossos* i *Lloros*. Accedint a la ruta següent:

Gràfics / Diagrama de dispersió

Obtindrem un quadre de diàleg complet amb múltiples opcions gràfiques, com veiem a continuació. En el nostre cas, només activarem l'opció *Línia de mínims quadrats*, per visualitzar la recta que millor s'ajusta als punts.



Com podem observar a continuació, el resultat mostra que no hi ha una correlació clara entre totes dues variables, ja que el pendent és força pla i la majoria dels punts estan molt allunyats de la recta estimada.

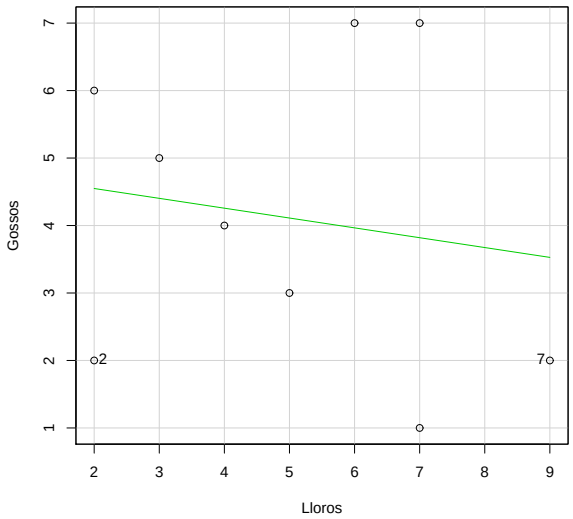


Diagrama de dispersió

Aquest tipus de gràfic és molt útil a l'hora de fer una exploració prèvia de les dades, és a dir, una anàlisi descriptiva, ja que proporciona informació visual sobre com es comporten les variables i com estan relacionades entre elles.

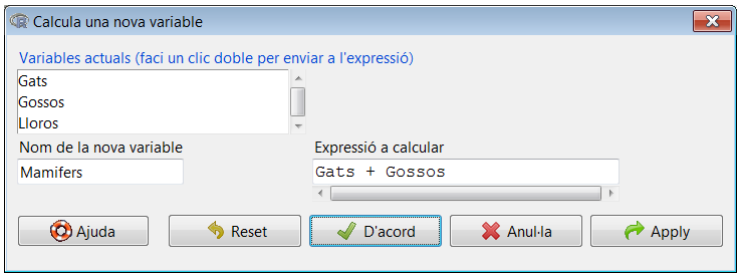
1.6. Transformació de variables

A l'hora d'efectuar una anàlisi quantitativa, és fonamental tenir l'opció de fer transformacions de les variables que estem estudiant. Això inclou sumar, restar, elevar a una potència, arrels quadrades, logaritmes i un llarg etcètera. En els dos primers mòduls hem repassat profundament aquesta mena d'operacions, les quals també es poden implementar amb R-Commander. En l'opció *Dades* de la barra de menús es poden efectuar totes aquestes transformacions.

Una primera transformació és el càlcul d'una variable a partir d'altres variables. En aquest exemple, calcularem la variable *Mamífers*, que serà la suma de les variables *Gats* i *Gossos*. Per a fer aquest càlcul, recorrerem a aquesta ruta:

Dades / Modifica variables de la taula de dades activa / Crea una nova variable

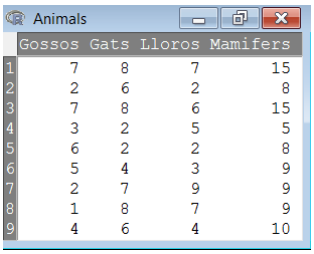
Apareixerà el quadre de diàleg següent en el qual hem d'introduir, a l'esquerra, el nom de la nova variable, i a la dreta la seva expressió. En aquesta segona finestra podem introduir qualsevol operació algebraica o funció d'R.



Creant noves variables

En aquest exemple, hem creat una nova variable simplement sumant dues variables originals.

Si tornem a visualitzar la taula de dades, veurem com s'ha incorporat en la columna dreta la variable que hem creat recentment.



Animals

	Gossos	Gats	Lloros	Mamifers
1	7	8	7	15
2	2	6	2	8
3	7	8	6	15
4	3	2	5	5
5	6	2	2	8
6	5	4	3	9
7	2	7	9	9
8	1	8	7	9
9	4	6	4	10

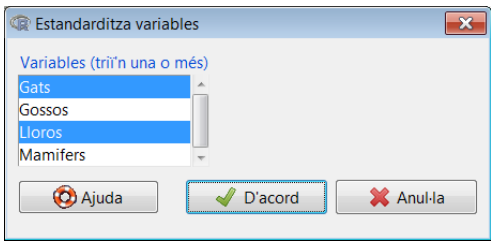
Una transformació molt utilitzada a l'hora de fer inferència estadística és l'estandardització o tipificació. En aquesta, partint d'una variable aleatòria X que segueix una distribució Normal, tal que $X \sim N(\mu, \sigma)$, aquesta es transforma en una variable Z tal que

$$Z = \frac{X - \mu}{\sigma}.$$

Els valors de la variable resultant permetran calcular àrees de probabilitat en la distribució normal estàndard, ja que $Z \sim N(0, 1)$. En R-Commander, per a estandarditzar les variables *Gats* i *Lloros*, seguirem la ruta següent:

Dades / Modifica variables de la taula de dades activa / Estandarditza variables

Ens apareixerà un menú en el qual hem de triar les variables que volem estandarditzar.



Estandarditzar amb R

La funció d'R `scale` s'utilitza per a estandarditzar vectors i matrius.

Si tornem a visualitzar la taula de dades activa, comprovem que en les columnes de la dreta s'han afegit les noves variables que s'han creat, amb una *Z* al davant que indica que aquestes variables han estat estandarditzades.

Animals

	Gossos	Gats	Lloros	Mamifers	Z.Gats	Z.Lloros
1	7	8	7	15	0.9525793	0.8164966
2	2	6	2	8	0.1360828	-1.2247449
3	7	8	6	15	0.9525793	0.4082483
4	3	2	5	5	-1.4969104	0.0000000
5	6	2	2	8	-1.4969104	-1.2247449
6	5	4	3	9	-0.6804138	-0.8164966
7	2	7	9	9	0.5443311	1.6329932
8	1	8	7	9	0.9525793	0.8164966
9	4	6	4	10	0.1360828	-0.4082483

2. Anàlisi del mercat de treball a Espanya

2.1. Importació i maneig de dades

En aquesta secció fem una anàlisi estadística del mercat de treball a Espanya. Per a això treballem amb dades de l'EPA procedents de l'Institut Nacional d'Estadística (INE). Específicament, per al 4t. trimestre de 2012 i per a cada província considerem les variables següents:

Enquesta de població activa (EPA)

L'EPA està elaborada per l'Institut Nacional d'Estadística (INE) amb periodicitat trimestral, i té com a objectiu oferir indicadors sobre la situació del mercat laboral a Espanya. Es considera el millor indicador de l'evolució de l'ocupació i la desocupació a Espanya.

- **act:** població activa.
- **inact:** població inactiva.
- **ocup:** població activa ocupada.
- **atu:** població activa a l'atur.

El primer pas en l'anàlisi és especificar la ruta al directori de treball. Això és fonamental, ja que les dades s'extrauran des d'aquí, i tot el que es desi també s'emmagatzemarà en aquesta carpeta. La funció que s'utilitzarà serà `setwd` (*set working directory*). Si el nostre directori de treball es troba a *C:/directori*, simplement haurem d'introduir la instrucció següent, **no oblidant introduir el directori entre cometes**.

Organització!

És fonamental disposar de la informació i els arxius ben organitzats en una carpeta. Això ens facilitarà la feina a l'hora d'importar i exportar dades procedents de l'anàlisi estadística.

```
> setwd("C:/directori")
```

El pas següent és importar un document de text (format *txt*) o full de càlcul (format *csv*, això és, valors separats per comes). En aquest exemple considerem la primera opció, és a dir, un document de text. Aquest document ha de tenir les variables en columnes i acostuma a contenir el nom de cada variable en la primera línia. És important que l'última línia d'aquest arxiu no estigui en blanc perquè la importació de les dades es faci correctament.

La funció més adequada per a importar dades en aquest format és `read.delim2`. El primer element que cal introduir és el nom de l'arxiu entre cometes. Si la primera línia està ocupada pel nom de la variable (és a dir, és un text i no un nombre), especificarem `header=TRUE`. A més, si en els valors numèrics els decimals fossin punts en lloc de comes, ho hauríem d'especificar amb l'ordre `dec=","`. La funció `read.delim2` desa les dades en un objecte tipus base de dades, el qual, fent gala d'originalitat, anomenarem `base.dades`:

Si el document de text que conté les dades que hem d'importar no conté el nom de les variables en la primera fila, haurem d'introduir l'opció `header=FALSE` en la funció d'importació de dades `read.delim2`.

```
> base.dades <- read.delim2("dades.txt",header=TRUE)
```

Una manera de comprovar que les dades s'han importat correctament és amb la funció `head`, la qual mostra en consola les sis primeres observacions en files, amb les variables en columnes.

Visualitzar bases de dades

Les funcions `head` i `tail` (cap i cua) ens permeten visualitzar el principi i el final d'una taula de dades respectivament, i se solen utilitzar per a comprovar que les dades s'han importat o introduït correctament.

```
> head(base.dades)
  Provincia   act  ocup   atu inact
1  Albacete 182.9 120.7  62.2 145.1
2  Alicante 894.8 639.2 255.6 707.7
3   Almeria 371.7 235.7 136.0 190.5
4     Alava 158.4 129.4  29.0  98.0
5  Asturias 480.4 366.2 114.1 437.0
6     Avila  76.1  59.2  16.9  64.4
```

De la mateixa manera, la funció `tail` mostra les últimes sis observacions.

```
> tail(base.dades)
    Provincia    act  ocup   atu inact
47   Valencia 1304.6 941.2 363.4 774.3
48 Valladolid  269.0 215.7  53.2 178.4
49     Zamora   74.0  55.3  18.7  90.5
50   Zaragoza  488.7 391.6  97.0 310.7
51      Ceuta   33.4  20.8  12.6  27.1
52    Melilla   32.2  23.1   9.1  26.6
```

La funció `dim` s'aplica tant a matrius com a bases de dades, i ens indica la dimensió de l'objecte. En aquest cas, comprovem que tenim 5 variables, cada una amb 52 observacions.

La funció `dim`

Aquesta funció, aplicada a una base de dades, ens mostra en primer lloc el nombre de files (observacions), i en segon lloc el nombre de columnes (variable).

```
> dim(base.dades)
[1] 52  5
```

Sempre és recomanable començar una anàlisi amb una visió general de les variables. Per a això, com hem vist en fer l'anàlisi descriptiva amb R-Commander, hi ha diferents funcions entre les quals es troba `summary`. Recordem que aquesta funció mostra per a cada variable el valor mínim, màxim, la mitjana i els tres quartils.

```
> summary(base.dades)
    Provincia       act              ocup
 Alava   : 1   Min.   :  32.2   Min.   :  20.8
 Albacete: 1   1st Qu.: 151.9   1st Qu.: 121.3
 Alicante: 1   Median : 298.0   Median : 215.0
 Almeria : 1   Mean   : 440.8   Mean   : 326.1
 Asturias: 1   3rd Qu.: 499.3   3rd Qu.: 362.8
 Avila   : 1   Max.   :3347.4   Max.   :2682.0
 (Other) :46
      atu             inact
 Min.   :  7.00   Min.   :  26.6
 1st Qu.: 29.98   1st Qu.: 117.0
 Median : 75.35   Median : 203.7
 Mean   :114.71   Mean   : 296.4
 3rd Qu.:132.32   3rd Qu.: 326.9
 Max.   :665.30   Max.   :1900.3
```

El símbol $

Recordem que aquest símbol s'utilitza per a accedir als diferents components d'un objecte. En aquest cas, es fa servir per a accedir a les diferents variables que componen una base de dades.

Per a fer referència a una variable d'una base de dades, la sintaxi serà `base.dades$variable`. Així doncs, la variable `atur` s'extreu de la manera següent.

```
> base.dades$atu
 [1]  62.2 255.6 136.0  29.0 114.1  16.9 117.4 144.1 651.5
[10]  94.3  32.7  56.2 233.6  53.5  81.1  69.6 131.1 114.0
[19]  21.9  39.5  91.5 162.9  35.0  89.5  14.8 111.6  48.1
[28]  37.3  25.9 665.3 275.4 216.6  52.6  30.3  14.1 198.8
[37] 107.1  28.0  34.1 169.6  15.9 302.5   7.0 104.8  10.1
[46] 107.9 363.4  53.2  18.7  97.0  12.6   9.1
```

De tota manera, això pot resultar pesat a l'hora de programar una anàlisi amb les variables. Per això, és recomanable fer servir la funció `attach`. D'aquesta manera, es podrà accedir a les variables simplement donant-ne els noms, sense fer referència a la base de dades.

La funció `attach`

Aquesta funció, aplicada a una base de dades, incrusta totes les variables d'aquesta en la memòria en la forma d'objectes (vectors) independents, de manera que per a referir-nos a aquestes variables no faci falta utilitzar repetidament el símbol $.

```
> attach(base.dades)
```

Continuarem l'anàlisi creant noves variables a partir de la base de dades importada. Els indicadors principals del mercat laboral són els següents:

$$\text{Taxa d'activitat} = 100 \cdot \frac{\text{Població activa}}{\text{Població} > 16 \text{ anys}}$$

$$\text{Taxa d'ocupació} = 100 \cdot \frac{\text{Població ocupada}}{\text{Població} > 16 \text{ anys}}$$

$$\text{Taxa d'atur} = 100 \cdot \frac{\text{Població en atur}}{\text{Població activa}}$$

En R, la creació d'aquestes variables és immediata.

```
> pob <- act + inact
> t.act  <- 100*act/pob
> t.ocup <- 100*ocup/pob
> t.atur <- 100*atu/act
```

Cada una d'aquestes variables té 52 observacions (tantes com províncies). Per a calcular les taxes que hem descrit abans en l'àmbit estatal, haurem de sumar per províncies.

La funció `sum`

Recordem que la funció `sum` aplica l'operació suma a tots els components d'un objecte, en aquest cas les variables de la nostra anàlisi.

```
> # Taxa d'activitat:
> 100*sum(act)/sum(pob)
[1] 59.79814

> # Taxa d'ocupació:
> 100*sum(ocup)/sum(pob)
[1] 44.2359

> # Taxa d'atur:
> 100*sum(atu)/sum(act)
[1] 26.02201
```

Si volguéssim aquests indicadors en tant per un, n'hi hauria prou de no multiplicar per 100.

2.2. Estadística descriptiva

R ofereix moltes possibilitats per a fer una anàlisi descriptiva de dades. A més de disposar de moltes llibreries amb estadístics i procediments, la seva flexibilitat fa que puguem programar la nostra pròpia anàlisi descriptiva a mida. Per començar, vegem com podem calcular la mitjana aritmètica de la taxa d'activitat.

```
> mean(t.act)
[1] 57.74514
```

Abans de fer una anàlisi descriptiva de les tres taxes que hem calculat més amunt, és recomanable unir-les per columnes per a formar una matriu 52×3. A més, amb la funció `row.names` assignem un nom a cada observació (fila), amb la variable `Provincia`, la qual conté els noms de les províncies.

Les funcions `cbind` i `rbind`

Aquestes funcions ens permeten unir objectes per columnes i per files, respectivament.

```
> indic <- cbind(t.act,t.ocup,t.atur)
> row.names(indic) <- Provincia
```

Com hem vist anteriorment, la funció `summary` mostra el valor mínim, màxim, la mitjana i els tres quartils de cada variable.

```
> summary(indic)
     t.act           t.ocup          t.atur
 Min.   :44.98   Min.   :33.62   Min.   :12.59
 1st Qu.:54.93   1st Qu.:37.88   1st Qu.:19.64
 Median :58.13   Median :42.96   Median :24.26
 Mean   :57.75   Mean   :42.93   Mean   :25.56
 3rd Qu.:60.92   3rd Qu.:47.75   3rd Qu.:31.65
 Max.   :66.63   Max.   :51.11   Max.   :40.63
```

Una funció molt útil a l'hora de fer un càlcul per a una taula de variables és `apply`. En aquet cas, aplicarem a la matriu de variables `indic`, i només per a la dimensió 2 (columnes), la funció `mean` (mitjana aritmètica). El resultat serà un vector amb les mitjanes aritmètiques de les tres variables d'interès.

```
> apply(indic,2,mean)
   t.act   t.ocup   t.atur
57.74514 42.93311 25.56296
```

Un càlcul una mica més complex però potencialment útil és el de crear una funció pròpia per a obtenir una taula de estadístics per a un grup de variables. En el nostre cas, anomenarem aquesta funció `estad.basic`, i consistirà en la mitjana, la variància, la desviació estàndard, el mínim, els tres quartils i el màxim (aquests últims calculats amb la funció `quantile`). El resultat l'arrodonirem a 2 decimals amb la funció `round`.

La sintaxi de la creació de funcions es descriu àmpliament en el primer mòdul.

```
> estad.basic <- function(x){
+   est <- cbind(mean(x),var(x),sd(x),t(quantile(x)))
+   colnames(est) <- c("mitjana","var","desv.est","min",
+   "Q1","Q2","Q3","max")
+   return(round(est,2))
+ }
```

Si apliquem aquesta funció a la variable *taxa d'activitat* (`t.act`), obtenim el resultat següent:

```
> estad.basic(t.act)
     mitjana  var desv.est   min    Q1    Q2    Q3   max
[1,]   57.75 22.2     4.71 44.98 54.93 58.13 60.92 66.63
```

Si combinem les funcions `apply` i `estad.basic`, podrem obtenir amb un sol càlcul els estadístics seleccionats aplicats a totes les variables simultàniament, i els resultats queden registrats en una matriu.

```
> est.total <- apply(indic,2,estad.basic)
> rownames(est.total)<- c("mitjana","var","desv.est","min",
+ "Q1","Q2","Q3","max")
> print(est.total)
         t.act t.ocup t.atur
mitjana  57.75  42.93  25.56
var      22.20  25.99  53.43
desv.est  4.71   5.10   7.31
min      44.98  33.62  12.59
Q1       54.93  37.88  19.64
Q2       58.13  42.96  24.26
Q3       60.92  47.75  31.65
max      66.63  51.11  40.63
```

2.3. Representació gràfica

Les possibilitats gràfiques que ofereix R són enormes, molt més àmplies que les que ofereix R-Commander. En aquesta secció només veurem una petita part de tot aquest potencial. L'objectiu és obtenir, amb les dades analitzades fins ara, els tipus de gràfics més utilitzats en estadística.

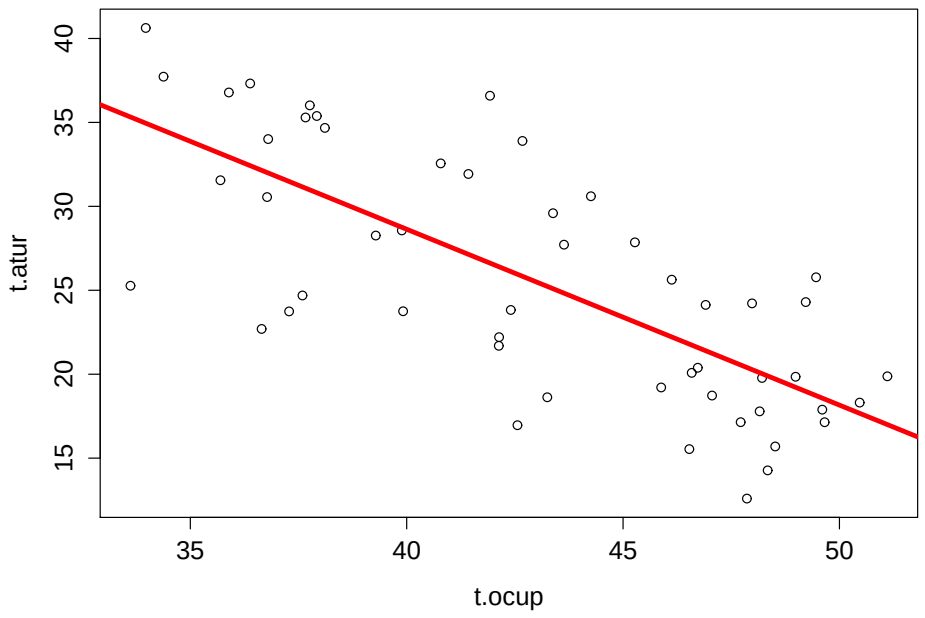
2.3.1. Diagrama de dispersió

Un diagrama de dispersió és un gràfic bidimensional que utilitza les coordenades cartesianes per a mostrar els valors de dues variables, una en cada eix. Si volem aquest diagrama per a les variables *taxa d'ocupació* i *taxa d'atur*, n'hi haurà prou d'introduir-les com a arguments de la funció `plot`. A més, opcionalment podem superposar més elements a aquest gràfic, per exemple rectes mitjançant la funció `abline`. En el nostre exemple, afegirem una recta ajustada als punts, per a la qual cosa farem servir la funció `lm` (*linear model*). Dibuixarem aquesta recta en vermell (`red`) i amb un gruix `lwd=4`.

Les opcions `col` i `lwd`

Aquestes opcions són molt útils a l'hora de personalitzar els gràfics, ja que permeten definir el color i el gruix dels punts i línies representades.

```
> plot(t.ocup,t.atur)
> abline(lm(t.atur~t.ocup),col="red",lwd=4)
```

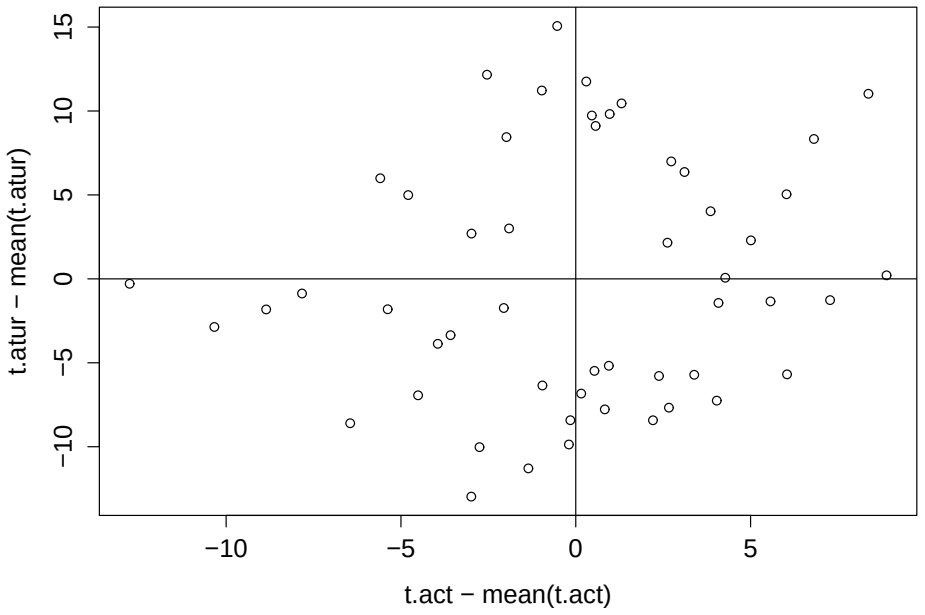


L'opció `abline`

Aquesta opció ens permet afegir qualsevol tipus de línies a gràfics ja creats, com capes superposades.

Un altre exemple de diagrama de dispersió és el de les variables centrades en el valor zero, per a la qual cosa s'ha d'aplicar la transformació $x_i - \bar{x}$, és a dir, restar la mitjana aritmètica. Addicionalment es pot afegir una línia horitzontal `h=0` i vertical `v=0` en el valor zero.

```
> plot(t.act-mean(t.act),t.atur-mean(t.atur))
> abline(h=0)
> abline(v=0)
```

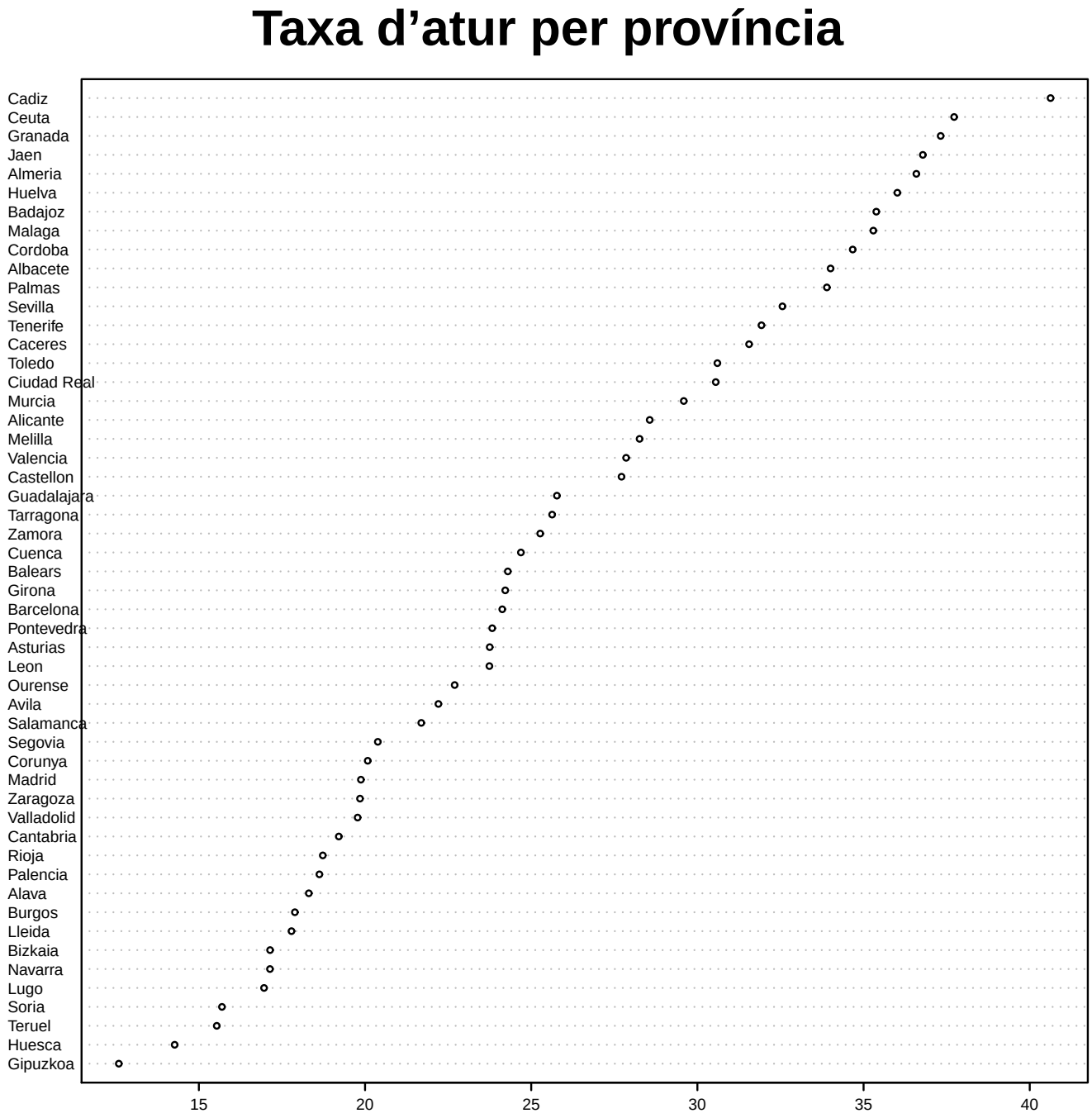


La funció `dotchart` dibuixa un diagrama de punts unidimensional, i identifica cada punt amb el seu nom. Per a fer-lo més complet, utilitzarem la funció `sort` perquè les dades apareguin ordenades de més petites a més grans, i considerarem els noms de la matriu `indic`. La instrucció `cex` fa referència a l'escala del gràfic respecte a 1. A aquest gràfic, hi afegirem un títol mitjançant la funció `title`.

Les opcions `labels` i `cex`

Aquestes opcions permeten introduir etiquetes i regular l'escala del gràfic.

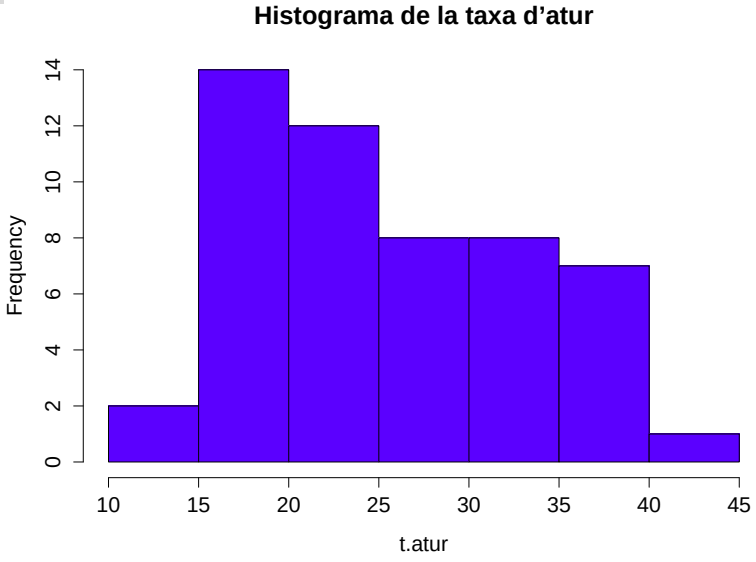
```
> dotchart(sort(indic[,3]),labels=names(indic),cex=.4)
> title("Taxa d'atur per província")
```



2.3.2. Histograma i funció de densitat

Aquest tipus de gràfic es fa servir per a visualitzar la distribució d'una variable. L'histograma es representa mitjançant la funció `hist` de la manera següent:

```
> hist(t.atur,main="Histograma de la taxa d'atur",col="blue")
```



Histograma i diagrama de barres

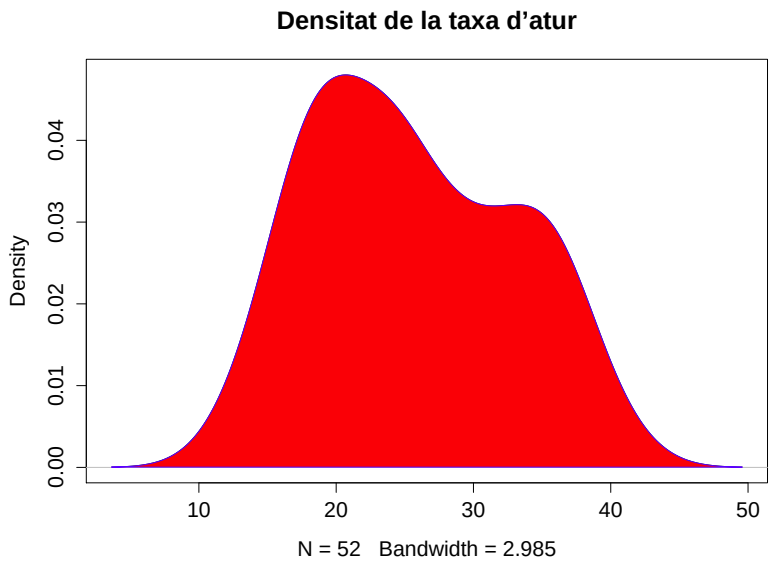
Encara que s'assemblen, aquests dos tipus de gràfic no són idèntics. L'histograma es fa servir per a representar **dades quantitatives contínues**, mentre que el **diagrama de barres** es fa servir per a representar gràficament dades quantitatives discretes o dades qualitatives.

L'opció `main`

Aquesta opció s'utilitza per a introduir un títol general a un gràfic.

La funció de densitat, calculada amb tècniques no paramètriques, és la generalització de l'histograma assumint que el gruix de les barres tendeix a zero. Vist d'una altra manera, l'histograma és discret i la funció de densitat contínua. El primer pas serà estimar la funció de densitat mitjançant la funció `density`, el resultat de la qual serà una taula de valors. Per a representar-los gràficament, farem servir la funció `plot`. Opcionalment, a més, podrem acolorir aquesta funció (tant la vora com l'interior) amb la funció `polygon`. És important veure com R elabora gràfics complexos afegint capes a partir d'una instrucció inicial.

```
> den.atur<- density(t.atur)
> plot(den.atur,main="Densitat de la taxa d'atur")
> polygon(den.atur, col="red", border="blue")
```

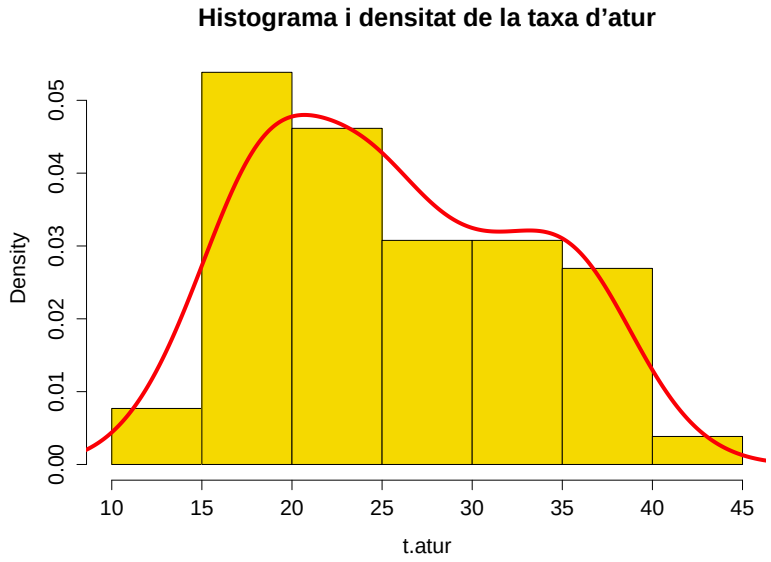


Una altra opció és combinar un histograma amb l'estimació de la funció de densitat. Amb la instrucció `freq=FALSE` establim que en l'eix d'abscisses no aparegui la freqüència, sinó la densitat de probabilitat. A més, amb la funció `lines` afegim al gràfic una capa addicional amb l'estimació de la funció de densitat.

```
> hist(t.atur,main="Histograma i densitat de la taxa d'atur",
+ col="gold",freq=FALSE)
> lines(den.atur,col="red",lwd=4)
```

Combinant un histograma i la funció de densitat

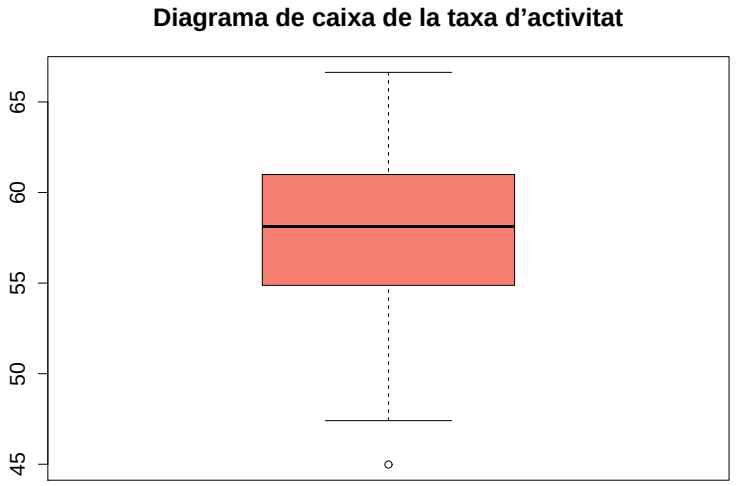
Aquestes línies de codi mostren com es poden combinar tots dos tipus de gràfics, i també com en R la creació de gràfics és seqüencial, això és, línia a línia d'una manera superposada.



2.3.3. Diagrama de caixa

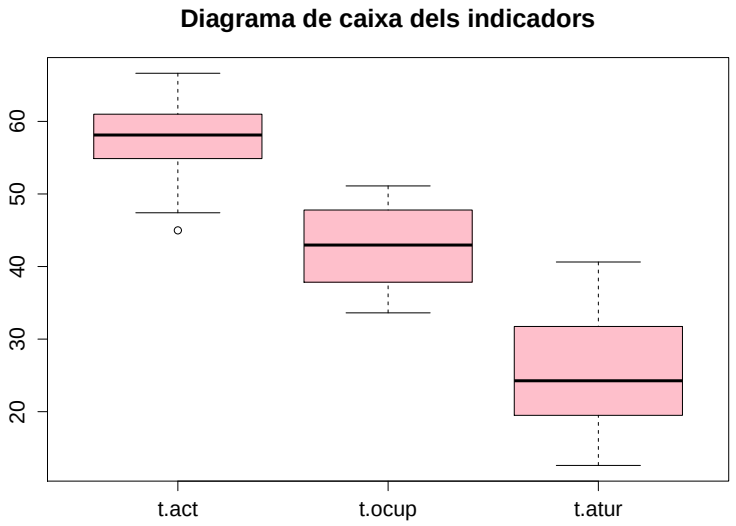
El diagrama de caixa consisteix en un gràfic basat en quartils, amb el qual es pot visualitzar la simetria de la distribució de les dades. La funció d'R que produeix aquest tipus de gràfic és `boxplot`. Vegem el diagrama de caixa de la taxa d'activitat:

```
> boxplot(t.act,col="salmon",main="Diagrama de caixa de la taxa
+ d'activitat")
```



Si el que ens interessa és la comparació de diferents diagrames de caixa de diverses variables en un mateix gràfic, introduirem com a primer argument una matriu (o base de dades) amb les variables disposades en columnes. En el nostre cas, la matriu `indic` inclou les tres variables del mercat laboral:

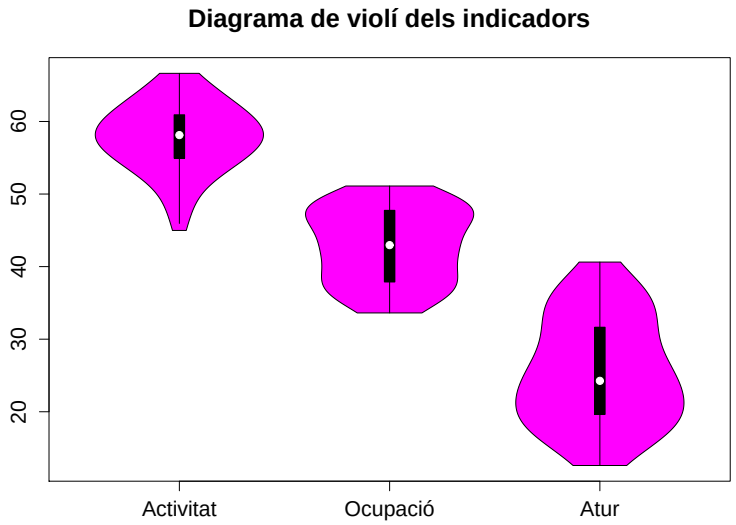
```
> boxplot(indic,col="pink",main="Diagrama de caixa dels indicadors")
```



Una altra opció interessant és el diagrama de violí, que combina el diagrama de caixa amb la funció de densitat en un sol gràfic. Abans haurem d'instal·lar el paquet `vioplot` i carregar la llibreria. El diagrama de violí de les tres variables d'interès pren la forma següent:

Per a poder carregar un paquet (*library*), abans s'ha de tenir instal·lat. Recordem també que només és necessari instal·lar els paquets una sola vegada.

```
> library(vioplot)
> vioplot(t.act,t.ocup,t.atur,names=c("Activitat","Ocupació","Atur"))
> title("Diagrama de violí dels indicadors")
```



2.3.4. Gràfics compostos

A vegades, per qüestió d'espai o de síntesi, necessitarem combinar diversos gràfics en un sol arxiu o imatge. Això es fa amb la instrucció `atur`. Així, en l'exemple següent disposarem quatre gràfics en dues files i dues columnes mitjançant la instrucció `mfrow=c(2,2)`. Els gràfics que introduïm a continuació s'aniran col·locant per files. Un cop hàgim obtingut el gràfic, per a restaurar els valors inicials dels gràfics, acabarem amb la instrucció `dev.off()`, amb la qual cosa s'esborrarà el gràfic compost.

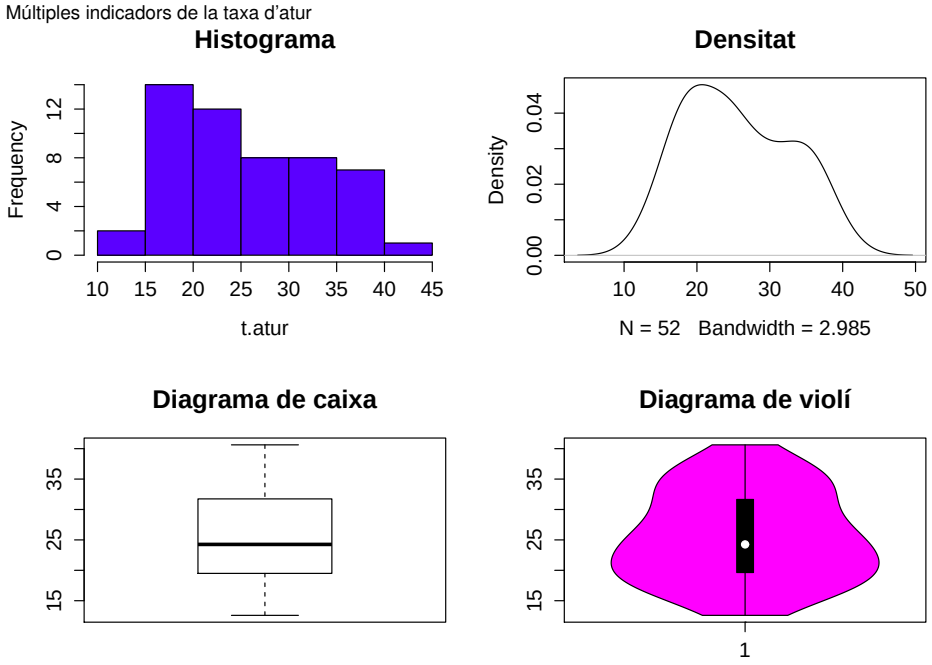
Elaborant gràfics compostos

Mitjançant la funció `par` podem combinar diversos gràfics en un sol marc. Amb l'opció `mfrow` establim com es disposen els diferents gràfics. En el nostre cas, en dues files i dues columnes respectivament.

```
> par(mfrow=c(2,2))
> hist(t.atur,col="blue",main="Histograma")
> plot(density(t.atur),main="Densitat")
> boxplot(t.atur,main="Diagrama de caixa")
> violplot(t.atur)
> title("Diagrama de violí")
> dev.off()
null device
          1
```

La instrucció `dev.off()`

Aquesta instrucció esborra el gràfic elaborat i restableix els valors gràfics per defecte. Si volem desar el gràfic, ho haurem de fer **abans** d'introduir aquesta última instrucció.



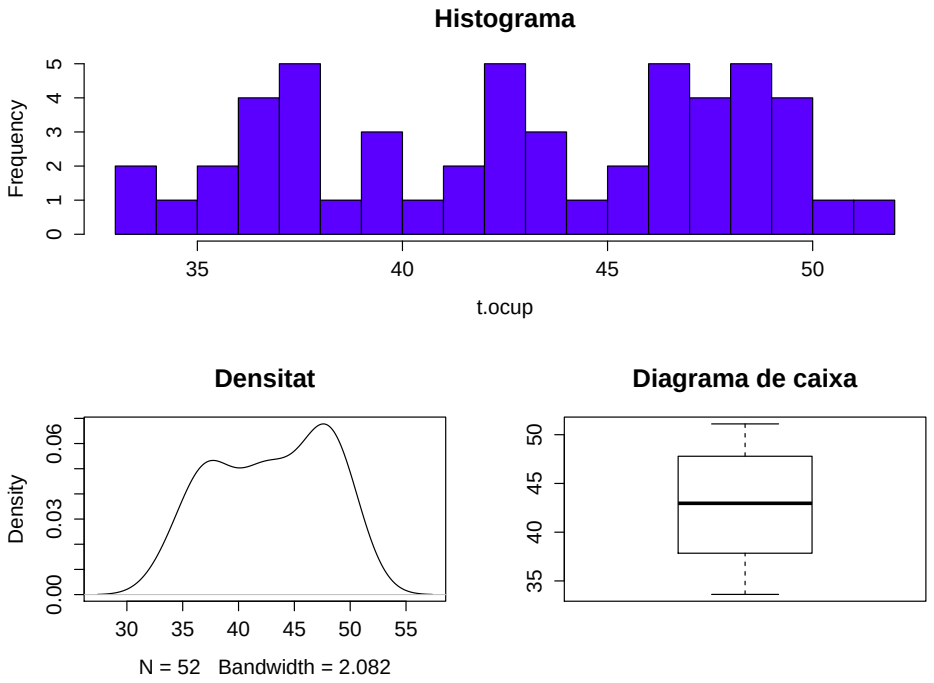
Per a fer composicions més complexes disposem de la funció `layout`. En l'exemple següent crearem 3 gràfics en una composició 2×2, és a dir, dues files i dues columnes. Establim que el primer gràfic ocupi tota la primera fila (és a dir, les dues columnes), mentre que els dos gràfics que queden ocupen cada un una columna de la segona fila. En el resultat es pot apreciar com el primer gràfic apareix allargat horitzontalment perquè ocupa el doble d'espai que els altres dos gràfics.

La funció `layout`

Aquesta funció permet disposar gràfics d'una manera molt flexible, com es mostra en aquest exemple.

```
> layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))
> hist(t.ocup,col="blue",main="Histograma",breaks=20)
> plot(density(t.ocup),main="Densitat")
> boxplot(t.ocup,main="Diagrama de caixa")
> dev.off()
null device
          1
```

Múltiples indicadors de la taxa d'ocupació



3. Anàlisi demogràfica a Catalunya

Aquesta secció està dedicada a l'estudi de dades demogràfiques dels municipis de Catalunya. Amb aquest estudi explicarem com es pot fer un estudi estadístic descriptiu introduint variables discretes, les quals s'analitzen amb factors. La base de dades que hem d'analitzar incorpora les variables següents:

- *municipi:* nom del municipi.
- *sup:* superfície municipal en km^2.
- *edat:* mitjana d'edat dels habitants.
- *pobl:* població total del municipi.
- *immig:* percentatge de població immigrant.
- *capital:* variable dicotòmica que pren el valor 1 si el municipi és capital de comarca, i 0 en cas contrari.
- *costa:* variable dicotòmica que pren el valor 1 si el municipi està situat a la costa, i 0 en cas contrari.
- *bcn:* variable dicotòmica que pren el valor 1 si el municipi està situat a l'àrea metropolitana de Barcelona, i 0 en cas contrari.

3.1. Maneig de bases de dades

Per començar l'anàlisi, carregarem al principi les llibreries que utilitzarem durant l'anàlisi. La primera és una llibreria que permet crear gràfics de dades amb factors, i la segona permet calcular moments estadístics.

```
> library(lattice)
> library(moments)
```

És recomanable tenir instal·lada una versió recent d'R per a garantir que les llibreries més recents es carreguin correctament.

El pas següent, anàlogament al cas anterior, serà importar les dades des d'un arxiu extern, en aquest cas amb extensió *R*.

```
> base.dades <- read.delim2("dades_demografia.R",header=TRUE)
```

Vegem les sis primeres observacions de la base de dades per a comprovar que s'ha fet correctament.

```
> head(base.dades)
             municipi   sup  edat  pobl immig capital costa bcn
1              Abrera  1.53 40.67 11278  9.47       0     0   1
2  Aguilar de Segarra  0.11 48.90   249  5.22       0     0   0
3              Alella  2.67 44.47  9260  9.01       0     0   1
4              Alpens  0.25 45.95   312  6.41       0     0   0
5  Ametlla del Vallès  4.44 40.92  7796  7.21       0     0   1
6       Arenys de Mar  2.04 46.61 14449 10.97       0     1   1
```

El sumari de la base de dades ens proporciona estadístics bàsics de la variable: mínim i màxim, mitjana aritmètica i quartils.

```
> summary(base.dades)
                      municipi         sup              edat
    Abrera                  :  1   Min.   : 0.010   Min.   :36.81
    Aguilar de Segarra      :  1   1st Qu.: 0.130   1st Qu.:43.90
    Alella                  :  1   Median : 0.360   Median :47.47
    Alpens                  :  1   Mean   : 1.183   Mean   :47.95
    Ametlla del Vallès (L'):  1   3rd Qu.: 1.120   3rd Qu.:51.37
    Arenys de Mar           :  1   Max.   :75.290   Max.   :68.43
 (Other)                         :935
      pobl             immig           capital
 Min.   :     29   Min.   : 0.000   Min.   :0.00000
 1st Qu.:    329   1st Qu.: 4.510   1st Qu.:0.00000
 Median :    957   Median : 8.400   Median :0.00000
 Mean   :   7826   Mean   : 9.811   Mean   :0.04357
 3rd Qu.:   3659   3rd Qu.:13.190   3rd Qu.:0.00000
 Max.   :1615908   Max.   :49.930   Max.   :1.00000

    costa               bcn
Min.   :0.00000   Min.   :0.0000
1st Qu.:0.00000   1st Qu.:0.0000
Median :0.00000   Median :0.0000
Mean   :0.07439   Mean   :0.2306
3rd Qu.:0.00000   3rd Qu.:0.0000
Max.   :1.00000   Max.   :1.0000
```

Interpretació del resum de les variables

En aquest cas, és interessant veure com s'interpreta la mitjana aritmètica de les variables dicotòmiques. Veiem com el $4,3\%$ dels municipis són capital, el $7,4\%$ estan situats a la costa i el 23% estan situats a l'àrea metropolitana de Barcelona.

És recomanable fer servir la funció `attach`. D'aquesta manera, es podrà accedir a les variables simplement donant-ne els noms, sense fer referència a la base de dades.

```
> attach(base.dades)
```

Molt sovint, per a agilitar l'explotació de les dades que volem analitzar, ens trobem amb la necessitat de crear bases de dades més reduïdes a partir de la base de dades original. En R això es pot fer amb la funció `subset`. Comencem amb un primer exemple molt senzill: de la base de dades inicial només extraurem dues variables (el municipi i l'edat), i només les observacions que compleixin la condició `edat>65`. Així doncs, la base de dades resultant només conté dues variables i tres observacions que compleixen aquesta condició.

```
> subset(base.dades,edat>65,select=c(municipi,edat))
                 municipi  edat
571                Bausen 67.75
819                 Forès 68.43
916  Vallfogona de Riucorb 66.99
```

La funció `subset`

Aquesta és una funció fonamental en l'anàlisi de dades, ja que permet crear bases de dades menors a partir d'una base de dades original introduint condicions.

Continuarem amb un altre exemple una mica més complex. En aquest cas extraurem tres variables (*municipi*, *edat* i *taxa d'immigració*), i només ens interessen les observacions que compleixin dues condicions: mitjana d'edat menor de 40 anys i taxa

d'immigració superior al 30%. Veiem que la base de dades resultant només conté dos municipis.

```
> subset(base.dades,edat<40 & immig>30,
+        select=c(municipi,edat,immig))
     municipi  edat immig
452      Salt 39.54 39.20
624  Guissona 38.50 43.46
```

El pas següent és veure com podem reordenar les observacions d'una base de dades segons l'ordre creixent o decreixent d'una variable. Aquí el vector `order(edat)` serà un vector que conté els valors 1 : 941 segons l'ordre creixent de la variable *edat*. Si ordenem la base de dades segons aquest vector, i en veiem els sis primers valors (`head`), obtenim el resultat següent:

```
> head(base.dades[order(edat),])
                 municipi  sup  edat pobl immig capital costa bcn
857      Pallaresos (Els) 1.00 36.81 3828  2.59       0     0   0
167               Polinyà 2.44 37.09 7403  6.85       0     0   1
866 Pobla de Mafumet (La) 2.77 38.29 2108  7.78       0     0   0
355                 Celrà 0.93 38.46 4329 15.64       0     0   0
624              Guissona 1.02 38.50 5683 43.46       0     0   0
504         Vall-llobrega 0.11 38.77  825  9.58       0     0   0
```

La funció `order`

Aquesta funció serveix per a ordenar vectors en ordre ascendent. Si es col·loca el signe negatiu davant del vector, el resultat és l'ordre descendent.

Ara fem un càlcul anàleg al cas anterior però aquesta vegada ordenant les observacions *en ordre decreixent* (mitjançant el signe negatiu) de la variable *immigració* `order(-immig)`:

```
> head(base.dades[order(-immig),])
                municipi  sup  edat  pobl immig capital costa bcn
353  Castelló d'Empúries 5.85 43.83 11653 49.93       0     1   0
624             Guissona 1.02 38.50  5683 43.46       0     0   0
940                Salou 4.11 40.02 25754 40.26       0     1   0
398        Lloret de Mar 8.75 40.58 37734 39.58       0     1   0
499                 Ullà 0.18 41.25  1067 39.27       0     0   0
452                 Salt 1.62 39.54 28763 39.20       0     0   0
```

3.2. Creació i anàlisi de variables

Per a l'anàlisi que farem a continuació és necessari crear la variable `densitat`, definida com la població dividida per la superfície i per 1.000.

```
> densitat <- pobl/(1000*sup)
```

Com en la secció anterior, crearem la funció `estad.basic`, consistent en la mitjana, la variància, la desviació estàndard, el mínim, els tres quartils i el màxim (aquests últims calculats amb la funció `quantile`), arrodonint a 2 decimals amb la funció `round`.

```
> estad.basic <- function(x){
+   est <- cbind(mean(x),var(x),sd(x),t(quantile(x)))
+   colnames(est) <- c("mitjana","var","desv.est","min",
+   "Q1","Q2","Q3","max")
+   return(round(est,2))
+ }
```

Tot seguit crearem la matriu `demo`, unint per columnes les variables *densitat*, *mitjana d'edat* i *taxa d'immigració* de cada municipi. A aquesta matriu apliquem la funció `estad.basic` per columnes, i obtenim les estadístiques bàsiques de cada una d'aquestes tres variables per al total de municipis.

```
> demo <- cbind(densitat,edat,immig)

> est.total <- apply(demo,2,estad.basic)
> rownames(est.total) <- c("mitjana","var","desv.est","min",
+ "Q1","Q2","Q3","max")

> print(est.total)
         densitat  edat immig
mitjana      5.43 47.95  9.81
var         95.68 25.69 51.31
desv.est     9.78  5.07  7.16
min          0.13 36.81  0.00
Q1           1.98 43.90  4.51
Q2           3.22 47.47  8.40
Q3           5.38 51.37 13.19
max        159.90 68.43 49.93
```

La matriu de correlació lineal d'aquestes variables es calcula amb la funció `cor`:

```
> cor(demo)
            densitat        edat       immig
densitat  1.00000000 -0.02328298  0.04810744
edat     -0.02328298  1.00000000 -0.28050864
immig     0.04810744 -0.28050864  1.00000000
```

La funció `cor`

Aquesta funció és vital per a calcular quina és la correlació lineal entre dues o més variables. Introduint en la consola `help(cor)` veurem les diferents maneres de calcular aquesta matriu.

3.3. Creació i anàlisi de factors

R ofereix moltes possibilitats d'anàlisi de variables quantitatives i qualitatives. Especialment interessant és la de les variables discretes, és a dir, que prenen un nombre limitat de nombres enters. Les variables qualitatives i discretes acostumen a contenir informació sobre les característiques de les variables. En el nostre exemple disposem de tres variables dicotòmiques o binàries: *capital*, *costa* i *bcn*. És recomanable crear factors amb la funció `factor` a partir d'aquestes variables, afegint també una etiqueta com es mostra a continuació.

Variables discretes, dicotòmiques i binàries

Una variable **discreta** pot prendre un nombre limitat de valors enters, i una variable **dicotòmica** o **binària** només pren dos valors. És a dir, una variable dicotòmica és una variable discreta, però el contrari no és necessàriament cert.

```
> f_cap <- factor(capital,labels=c("no_cap","cap"))
> f_cos <- factor(costa,labels=c("no_costa","costa"))
> f_bcn <- factor(bcn,labels=c("no_bcn","bcn"))
```

És fonamental saber crear i manejar factors quan utilitzem variables qualitatives en la nostra anàlisi.

La funció `table` utilitza els factors per a la construcció d'una taula de contingència per a cada combinació de nivells dels factors. Per exemple, les 941 observacions es divideixen en quatre grups segons siguin capital o no i siguin a la costa o no:

```
> table(f_cap,f_cos)
        f_cos
f_cap    no_costa costa
  no_cap      836    64
  cap          35     6
```

Una funció que permet aplicar a una variable una operació diferenciant pel valor d'un factor és `tapply`. Per exemple, la instrucció `tapply(edat,f_cos,mean)` produeix

la mitjana de la variable *edat* per als dos grups (nivells) del factor *costa*. En l'exemple següent comprovem que el nombre de municipis situats a la costa és de 70, i que tant la mitjana d'edat com la desviació estàndard són menors a la costa que a l'interior.

```
> mostra <- tapply(edat,f_cos,length)
> mitjana <- tapply(edat,f_cos,mean)
> desv.est <- tapply(edat,f_cos,sd)
> Rtat <- rbind(mostra,mitjana,desv.est)
> row.names(Rtat) <- c("N","Mitjana","Desv. Est.")
> print(Rtat,digits=3)
           no_costa costa
N            871.00 70.00
Mitjana       48.21 44.75
Desv. Est.     5.12  2.79
```

L'opció `digits`

Aquesta opció ens permet ajustar el nombre de decimals del resultat. Específicament, ens permet establir el nombre mínim de dígits significatius per a tots els valors mostrats.

Si combinen la funció `tapply` amb la funció d'estadístics bàsics (`estad.basic`) que hem creat, obtindrem una sèrie d'estadístics per a la variable *edat*, depenent de si el municipi és a l'àrea de Barcelona o no. Comprovem que la mitjana d'edat a l'àrea metropolitana de Barcelona és més baixa que a la resta de Catalunya.

```
> tapply(edat,f_bcn,estad.basic)
$no_bcn
     mitjana   var desv.est   min    Q1    Q2    Q3   max
[1,] 48.95   22.57     4.75 36.81 45.44 48.66 52.05 68.43

$bcn
     mitjana   var desv.est   min    Q1    Q2    Q3   max
[1,] 44.62   21.82     4.67 37.09 41.62 43.42 45.49 66.99
```

La funció `tapply`

Aquesta funció ens permet obtenir informació estadística d'una variable segmentant-la segons els valors d'un factor.

Essent una mica més ambiciosos, per a la variable *edat* calcularem una taula amb els estadístics de la funció `estad.basic` per als tres factors disponibles.

```
> estad.edat <- rbind(
+   estad.basic(edat),
+   tapply(edat,f_cap,estad.basic)$no_cap,
+   tapply(edat,f_cap,estad.basic)$cap,
+   tapply(edat,f_cos,estad.basic)$no_costa,
+   tapply(edat,f_cos,estad.basic)$costa,
+   tapply(edat,f_bcn,estad.basic)$no_bcn,
+   tapply(edat,f_bcn,estad.basic)$bcn
+ )
> rownames(estad.edat) <- c("Total","No Capital","Capital",
+ "Interior","Costa","No BCN","BCN")
```

```
> print(estad.edat)
           mitjana   var desv.est   min    Q1    Q2    Q3   max
Total        47.95 25.69     5.07 36.81 43.90 47.47 51.37 68.43
No Capital   48.10 26.13     5.11 36.81 44.09 47.59 51.52 68.43
Capital      44.55  4.10     2.03 40.42 43.38 44.08 45.70 48.99
Interior     48.21 26.25     5.12 36.81 44.16 47.68 51.60 68.43
Costa        44.75  7.77     2.79 40.02 43.40 44.22 45.51 56.44
No BCN       48.95 22.57     4.75 36.81 45.44 48.66 52.05 68.43
BCN          44.62 21.82     4.67 37.09 41.62 43.42 45.49 66.99
```

Una altra funcionalitat d'R és la de la creació de variables discretes mitjançant la partició de variables en intervals. Per exemple, vegem quin és el rang de la taxa d'immigració:

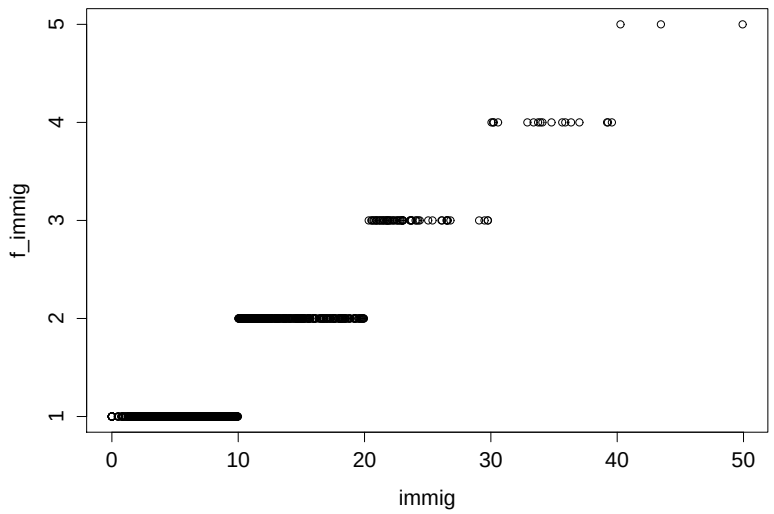
```
> range(immig)
[1]  0.00 49.93
```

Veiem que està limitada, aproximadament, entre el 0% i el 50%. El que farem serà crear un factor que prengui 5 possibles valors discrets, i que es correspongui amb els intervals $[0, 10)$, $[10, 20)$, ...,$[40, 50]$ de la variable `immig`. Després de crear els punts de tall, amb la funció `cut` trossegem la variable `immig` de manera que obtinguem 5 nivells diferents en la variable `f_immig`. Un diagrama de dispersió entre les dues variables permet veure gràficament la relació entre totes dues.

La funció `cut`

Aquesta funció permet obtenir una variable discreta a partir d'una variable contínua. El que fa és segmentar aquesta variable segons uns intervals definits i crear diverses categories. Amb l'opció `right=FALSE` especifiquem que els intervals siguin oberts per la dreta (i tancats per l'esquerra).

```
> int <- seq(0,50,by=10)
> f_immig <- cut(immig,breaks=int,right=FALSE)

> plot(immig,f_immig)
```



Segmentació de variables

En aquest gràfic es pot veure clarament l'ús de la funció `cut`. L'eix horitzontal representa la variable contínua original, i l'eix vertical correspon al factor creat, el qual només pren cinc valors, corresponents als cinc intervals establerts.

Un cop creat el factor `f_immig`, el podem creuar amb el factor `f_bcn` per a veure com es reparteixen els municipis segons la taxa d'immigrants i la seva pertinença a l'àrea metropolitana de Barcelona. Per a això, utilitzarem la funció `table`.

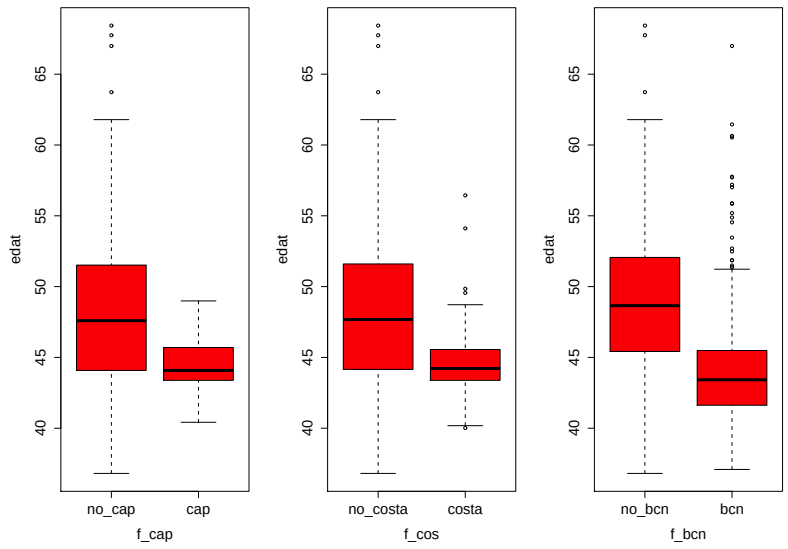
```
> table(f_immig,f_bcn)
         f_bcn
f_immig   no_bcn bcn
  [0,10)     417 149
  [10,20)    237  61
  [20,30)     50   7
  [30,40)     17   0
  [40,50)      3   0
```

3.4. Representació gràfica

3.4.1. Gràfics amb component factorial

La presència de variables qualitatives, discretes o dicotòmiques en les dades ens dóna molt joc a l'hora de compondre gràfics en què es mostren possibles relacions entre aquestes. Començarem amb la representació de tres diagrames de caixa per a la variable `edat`, un per a cada factor (capitalitat, costa i Barcelona).

```
> par(mfrow=c(1,3))
> plot(edat~f_cap,col="red")
> plot(edat~f_cos,col="red")
> plot(edat~f_bcn,col="red")
> dev.off()
null device
          1
```

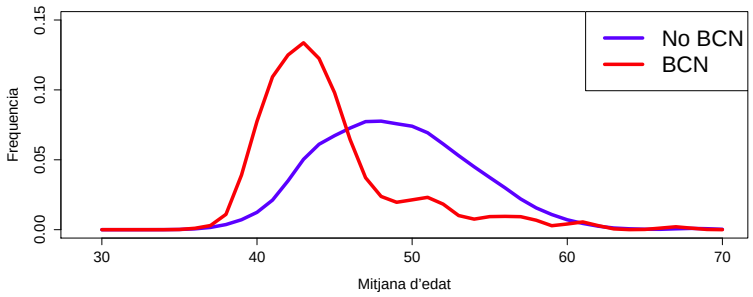


En l'exemple següent farem una cosa una mica més complexa, representarem gràficament la densitat empírica de la mitjana d'edat diferenciant entre municipis dins i fora de l'àrea metropolitana de Barcelona. El primer pas consistirà a crear dues variables per a la mitjana d'edat amb la finalitat de calcular-ne posteriorment les densitats respectives. Després crearem un marc per al gràfic, especificant els valors del rang i el domini, a més del nom dels dos eixos. Aquest marc, amb la instrucció `type="n"`, el deixem buit de contingut, ja que tot seguit hi afegim les dues capes mitjançant la funció `lines`, és a dir, l'omplim amb les dues funcions de densitat. Finalment, incorporem els valors de l'eix d'abscisses i la llegenda.

```
> edat_no_bcn <- edat[f_bcn=="no_bcn"]
> edat_bcn <- edat[f_bcn=="bcn"]
> d_no_bcn <- density(edat_no_bcn,from=30,to=70,n=41)
> d_bcn <- density(edat_bcn,from=30,to=70,n=41)

> plot(c(0,41),c(0,0.15),type="n",xaxt="n",xlab="Mitjana d'edat",
+ ylab="Frequencia")
> lines(d_no_bcn$y,type="l",col="blue",lwd=5)
> lines(d_bcn$y,type="l",col="red",lwd=5)
> lab <- 1+10*0:4
> axis(1,at=lab,labels=10*3:7)
> legend("topright",c("No BCN","BCN"),col=c("blue","red"),
+ lty=1,lwd=5,cex=1.5)
```

La funció `density`

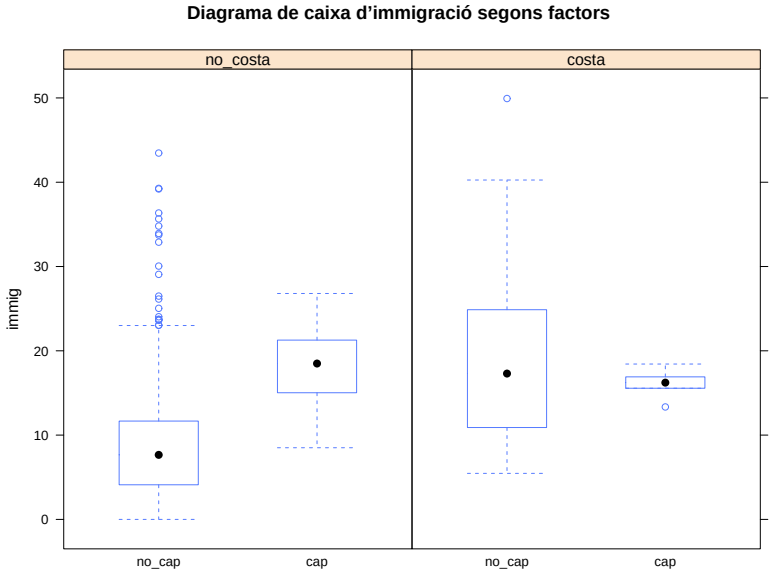
Aquesta funció es pot limitar a un rang i a un nombre de punts determinat mitjançant l'opció `n`.



3.4.2. La llibreria *Lattice*

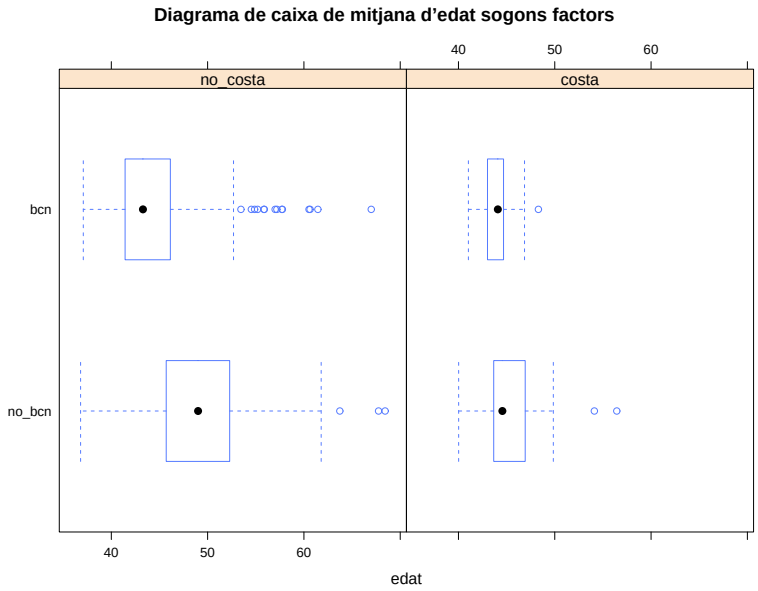
La llibreria `Lattice` és un sistema de visualització de dades d'alt nivell. És potent i elegant, amb un èmfasi en les dades multivariants, com és el cas d'aquest estudi. La primera funció d'aquesta llibreria és `bwplot`, que permet representar un diagrama de caixa per a diferents nivells de factors. Això és, especificant `immig ~ f_cap|f_cos` obtenim el diagrama per a la variable `immig` segons els factors capital i costa.

```
> bwplot(immig ~ f_cap | f_cos, main="Diagrama de caixa
+ d'immigració segons factors")
```



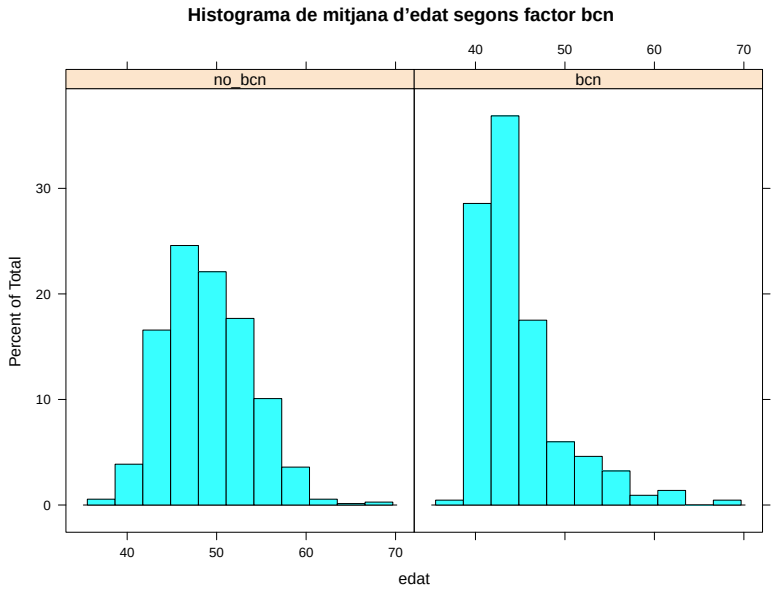
Si canviem l'ordre dels factors (`f_bcn ~ edat|f_cos`) l'ordre del gràfic també canvia.

```
> bwplot(f_bcn ~ edat | f_cos, main="Diagrama de caixa
+ caixa de mitjana d'edat sogons factors")
```



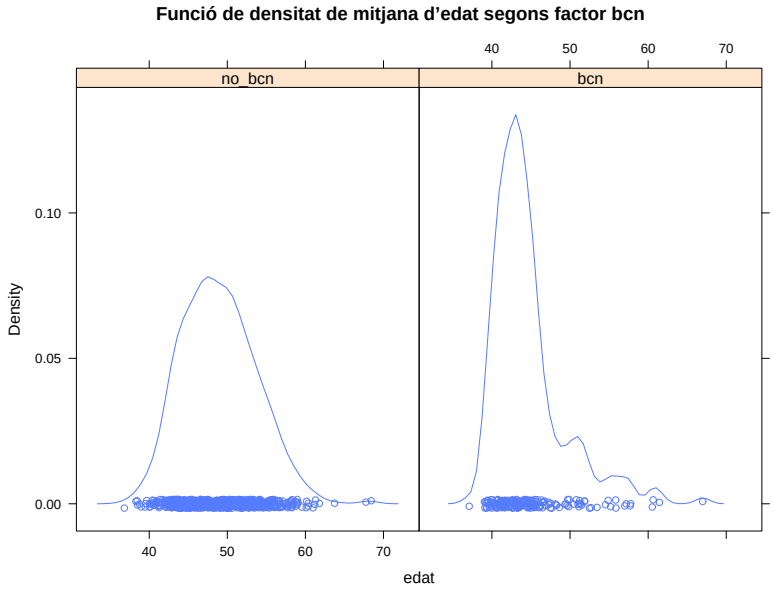
En l'exemple següent, dibuixarem la funció de densitat de la variable `edat` dividint-la en dos subgrups a partir de la variable `f_bcn`. Això és, per als municipis que pertanyen a l'àrea metropolitana de Barcelona (`f_bcn=1`) i per als que no (`f_bcn=0`).

```
> histogram(~ edat | f_bcn, main="Histograma de
+ mitjana d'edat segons factor bcn")
```



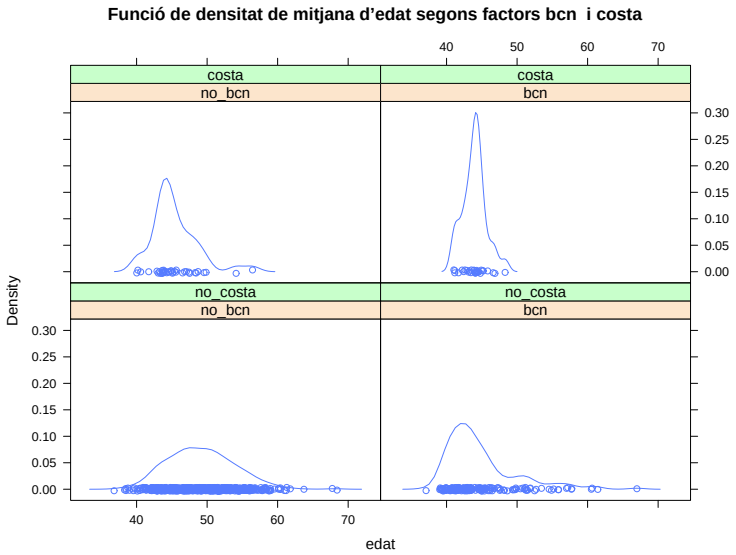
Un gràfic similar a l'anterior és el següent, però en comptes de calcular l'histograma de la variable calcula la funció de densitat empírica, a més de situar les observacions en l'eix d'abscisses.

```
> densityplot(~ edat | f_bcn, main="Funció de densitat
+ de mitjana d'edat segons factor bcn")
```



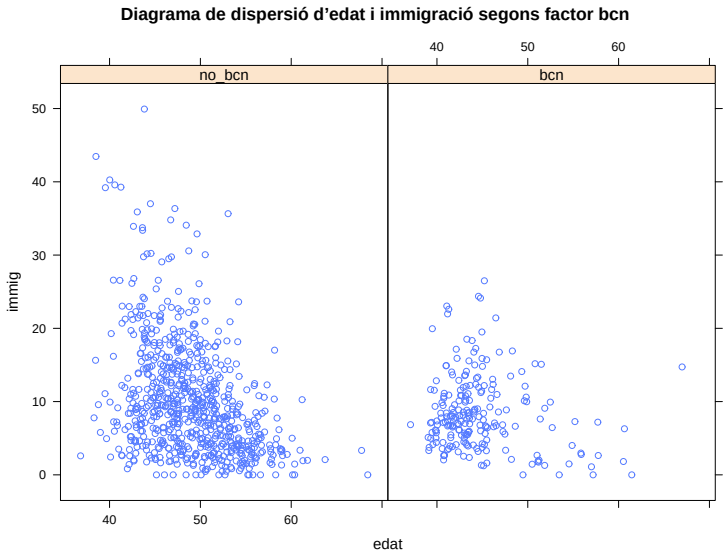
Podem complementar el gràfic anterior afegint un altre factor. Si introduïm `f_bcn*f_cos`, estem de fet creuant dos factors: la pertinença a l'àrea metropolitana de Barcelona i la localització costanera del municipi, de manera que el resultat seran quatre gràfics de la funció de densitat empírica d'`edat` amb les quatre possibles combinacions de `f_bcn` i `f_cos`.

```
> densityplot(~ edat | f_bcn*f_cos, main="Funció de
+ densitat de mitjana d'edat segons factors bcn i costa")
```



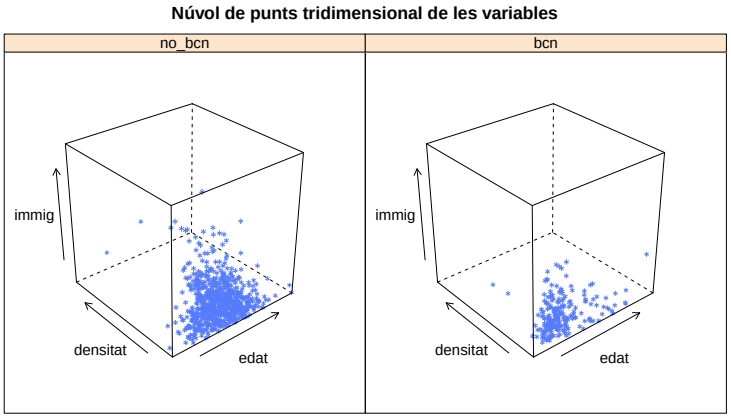
De manera similar als gràfics anteriors, el diagrama de dispersió d'una variable segons els valors discrets d'un factor es fa amb la funció `xyplot`. Per exemple, si volem obtenir dos diagrames de dispersió de les variables *immigració* i *edat*, un per a valors `f_bcn=1` i un altre per a valors `f_bcn=0`, n'hi haurà prou d'introduir les instruccions següents:

```
> xyplot(immig ~ edat | f_bcn, main="Diagrama de dispersió
+ d'edat i immigració segons factor bcn")
```



Un equivalent tridimensional del diagrama de dispersió és el núvol de punts, consistent en una taula de vèrtexs en un sistema de coordenades tridimensional. S'hi representa el valor de cada observació com un punt referenciat per tres eixos (X, Y i Z) corresponents a tres variables. Vegem com es calcula el núvol de punts de les variables *immigració*, *edat* i *densitat* segons la pertinença o no a l'àrea metropolitana de Barcelona:

```
> cloud(immig ~ edat*densitat | f_bcn, main="Núvol de punts
+ tridimensional de les variables")
```



Finalment, vegem una estructura que és molt útil a l'hora d'analitzar les relacions entre diverses variables. Aquesta estructura i la seva interpretació s'assemblen a les d'una matriu de correlacions, però en comptes de valors numèrics, cada element és un diagrama de dispersió.

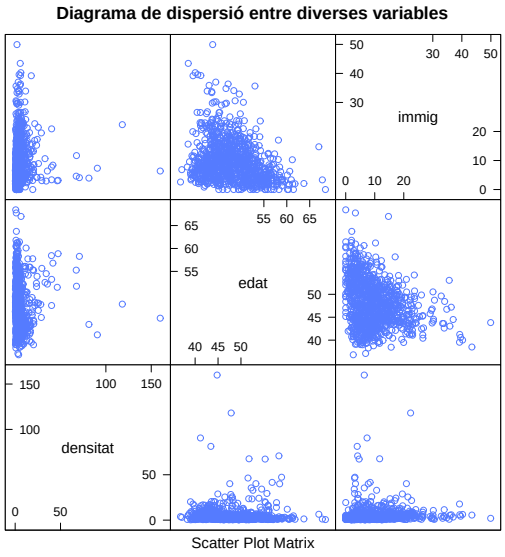
Recordem que el diagrama de dispersió d'una variable amb si mateixa donaria com a resultat una sèrie de punts alineats perfectament sobre la diagonal $x = y$.

Vegem-ho en un exemple pràctic: per a les variables *immigració*, *edat* i *densitat* tindrem una estructura 3×3, en què els elements de la diagonal estan buits. Fixem-nos que els diagrames de la diagonal superior són iguals que els de la diagonal inferior invertits.

```
> splom(demo, main="Diagrama de dispersió entre diverses
+ variables")
```

La matriu `demo`

Recordem que, en aquest mòdul, hem creat aquesta matriu que inclou tres variables relatives al percentatge d'immigració, mitjana d'edat i densitat dels municipis de la base de dades.



Bibliografia

Gibernans Bàguena, J.; Gil Estallo, À. J.; Rovira Escofet, C. (2009). *Estadística*. Barcelona: Material didàctic UOC.