Laboratorio di ST1
Lezione 2

Claudia Abundo

Dipartimento di Matematica
Università degli Studi Roma Tre

Frequenze in R

ESEMPIO

Fiori preferiti da n=6 ragazze

In R:

```
fiori=c("rosa", "orchidea", "violetta", "rosa", "rosa", "orchidea")
table(fiori)
length(fiori)
table(fiori)/length(fiori)
```

il comando length conta quanti sono i fiori

table(fiori)/length(fiori) rapporta le frequenze assolute alla numerosità
quindi fornisce le frequenze relative!

```
tit=c("Licenza media", "Biennio o triennio scuola sup.", "Diploma di
istr. secondaria sup.", "Dipl. universitario", "Laurea")
num=c(2,0,26, 2, 79)
TIT=data.frame(tit, num)
prop.table(TIT[,2])
TIT
cumsum(TIT[,2])
```

un data frame è una struttura per dati di qualsiasi tipo: è possibile inserire in un data frame dati sia caratteri, che numeri, che operatori logici (true/false)
“TIT” crea una struttura di dati
prop.table(TIT[,2])
fornisce le frequenze relative cumsum(TIT[,2])
fornisce le frequenze cumulate

read.table

Come si usa “read.table”:

```
read.table("nome_file.formato", header = TRUE/FALSE, sep = " t\ /
;",  dec = ".")
```

Una serie di comandi separati da virgole:

- si inserisce il nome del file tra virgolette
- “header” indica se sul data set é presente il nome delle variabili (TRUE se si, FALSE se no)
- “sep” indica come i diversi dati sono separati tra loro (es. \t se c’è uno spazio)
- “dec” indica come sono definiti gli eventuali numeri decimali

Partite del campionato di calcio italiano 68- 69 distinte in:
{“In parità’, “Vince la squadra che gioca in casa”, “Vince la squadra che gioca fuori casa”}

- Creiamo una cartella di nome “5 Marzo” sul desktop
- cambiamo la directory di lavoro:

  ```
  setwd("C:/Documents and Settings/claudia/Desktop/5 MARZO")
  ```

  oppure da file → cambia directory...

- ```
  esiti=read.table("esiti.txt", sep=" ")
  ```
- frequenze relative

  ```
  prop.table(esiti[,2])
  ```
- le frequenze cumulate non hanno senso perchè la variabile non è ordinata!
  ```

Sul file “dataset_1” abbiamo a disposizione sesso ed età relativamente ad un collettivo di n=40 persone

```
coll=read.table("dataset_1.txt", sep="\t", header= T)
```

Possiamo determinare le frequenze di ogni variabile attraverso il comando

```
table(coll$nome.variabile)
```

ESEMPIO

```
table(coll$SESSO)
table(coll$ETA)
```

Suddividere in classi

```
coll$ETA2=cut(coll$ETA, breaks=c(60, 70, 80, 90))
```

si definiscono dei “breaks” a 60, 70, 80, 90 anni

```
table(coll$ETA2)
```

fornisce la nuova tabella

Diagramma circolare

x=c(elenco di percentuali separate da una virgola: numeri che sommano a 100 o a 1)
pie(x)

Esempio:

```
x=c(10, 10, 20, 40, 20)
```

oppure

```
x=c(0.1, 0.1, 0.2, 0.4, 0.2)
pie(x)
```

Salvare i grafici

Quando si elabora un grafico, questo appare in una nuova finestra.
Cliccando in alto a sinistra su File appare Salva con nome, puntandoci sopra il mouse appaiono tanti modi diversi di salvare il grafico: postscript, pdf, bmp, jpeg...

Attenzione! Se facciamo un altro grafico si sovrapporrà al primo: se vogliamo tenerceli è importante salvarli subito.

Per inserire il grafico in un documento, ad esempio Word, basta cliccare sul grafico col tasto destro e poi sinistro su copia come bitmap oppure copia come metafile e poi incolla sul documento aperto.

```
a=c(0.1, 0.3, 0.25, 0.15, 0.2)
pie(a)
b=c(0.2, 0.2, 0.2, 0.3, 0.1)
pie(b)
```

Affiancare più grafici in una stessa finestra

- due grafici su una stessa riga

```
par(mfrow=c(1, 2))
pie(a)
pie(b)
```

- due grafici su una stessa colonna

```
par(mfrow=c(2, 1))
pie(a)
pie(b)
```

Spike Plot

Grafico per distribuzioni discrete (numero finito di modalità)

Dal data set “cereal.txt”
plottiamo le frequenze della quarta variabile (calorie)

```
cereal=read.table("cereal.txt", sep=" ", dec=".")
table(cereal[,4])
plot((table(cereal[,4])),xlab="Calorie",ylab="Frequenze",col="red",
lwd=5)
```

Istogrammi

Quando la variabile é (virtualmente) continua; o discreta ma con un elevato numero di modalità (le misurazioni sono sempre discrete!), tabelle o grafici che riportino tutti i valori distinti e le relative densità possono essere poco sintetiche o informative

Negli istogrammi

- l'asse orizzontale é un segmento dell'asse reale suddiviso in intervalli (classi)
- l'area dei rettangoli (barre) rappresenta la frequenza relativa
- L'area totale dell'istogramma é uguale 1

Esempio: Il file *libri.csv* contiene l'ammontarare in dollari speso da 40 studenti per l'acquisto di libri all'inizio dell'anno accademico (Johnson, R., and Bhattachar yya, G. K., p. 32).

```
>libri=read.table("libri.csv",header=T,sep=";",dec=",")
> # a fini illustrativi visualizziamo di dati
> # disposti in matrici solo per motivi di spazio
> matrix(sort(libri$ordini),nrow=5)
      [,1]   [,2]   [,3]   [,4]   [,5]   [,6]  [,7]   [,8]
[1,] 16.00 142.20 219.70 262.35 319.00 372.60 428.5 458.10
[2,] 58.50 145.35 247.55 268.20 332.00 383.40 432.4 493.95
[3,] 68.20 186.70 249.19 269.60 343.20 389.20 444.6 511.95
[4,] 78.00 209.05 256.00 270.15 350.75 404.55 446.4 521.05
[5,] 79.45 216.75 257.15 284.45 354.90 420.40 456.8 621.35
```

Il comando `sort(x)` produce un ordinamento crescente del vettore (fattore), `sort(x,decreasing=T)` produce un ordinamento decrescente

Il comando `hist()`

- suddivide la distribuzione in classi, determinandone numero ed estremi
- calcola le frequenze dei casi che ricadono in ciascuna classe
- procede alla generazione del grafico corrispondente

Calcolo frequenze relative e altezze

```
> #tabella frequenze assolute
>  ordini2=cut(libri$ordini,breaks=seq(0,700,100))
> table(ordini2)
ordini2
  (0,100] (100,200] (200,300] (300,400] (400,500] (500,600] (600,700]
        5         3        12         8         9         2         1
> #tabella frequenze relative
> table(ordini2)/sum(table(ordini2))
ordini2
  (0,100] (100,200] (200,300] (300,400] (400,500] (500,600] (600,700]
    0.125     0.075     0.300     0.200     0.225     0.050     0.025
> #tabella altezze rettangoli
> table(ordini2)/sum(table(ordini2))/diff(seq(0,700,100))
ordini2
  (0,100] (100,200] (200,300] (300,400] (400,500] (500,600] (600,700]
  0.00125   0.00075   0.00300   0.00200   0.00225   0.00050   0.00025
```

La tabelle viste nella slide precedente sono state ottenute usando comandi già noti, **R** fornisce con un solo comando tutte le informazioni necessarie

```
> ist.val=hist(libri$ordini, plot=F)
> str(ist.val) #str() mostra il contenuto di un oggetto
List of 7
 $ breaks     : num [1:8] 0 100 200 300 400 500 600 700
 $ counts     : int [1:7] 5 3 12 8 9 2 1
 $ intensities: num [1:7] 0.00125 0.00075 0.00300 0.00200 0.00225 ...
 $ density    : num [1:7] 0.00125 0.00075 0.00300 0.00200 0.00225 ...
 $ mids       : num [1:7] 50 150 250 350 450 550 650
 $ xname      : chr "libri$ordini"
 $ equidist   : logi TRUE
 - attr(*, "class")= chr "histogram"
> #Esempio su come usare queste informazioni:
> sum(ist.val$counts)
[1] 40
# ed otteniamo la numerosità campionaria
```

L'opzione `plot=F` impedisce la creazione del grafico, che si può ottenere successivamente con `plot(ist.val)`
Provate anche `ist.val`

Classi di ampiezza diversa

```
> min(libri$ordini)
[1] 16
> max(libri$ordini)
[1] 621.35
> hist(libri$ordini,breaks=c(min(libri$ordini),250,500,
+max(libri$ordini)))
```

Di *default* sulle ordinate sono visualizzate le altezze dei rettangoli!

Alcune opzioni utili

```
hist(libri$ordini,breaks=3)
```

Si indica il numero delle classi (la funzione genera gli estremi)

```
hist(libri$ordini,freq=F)
```

Anche per classi di uguale ampiezza abbiamo sulle ordinate le altezze dei rettangoli, le aree sommano ad uno. Di *default* `freq=T` e sulle ordinate si hanno le frequenze assolute (sempre che i breaks siano equidistanti)

Per altre opzioni `?hist` é particolarmente ben fatto

La Media aritmetica (semplice)

Peso é un vettore contenente le rilevazioni del peso di 26 persone

```
> tests=read.table("Tests.txt",header=T,sep=";")
> peso=tests$Peso
> Mpeso=sum(peso)/length(peso)
> Mpeso
[1] 55.15385
> mean(peso)
[1] 55.15385
```

Proprietà della media

```
> # Gli scarti sommano a zero
> #
> sum(peso-Mpeso)
[1] 1.421085e-14
> #
> #La media é compresa tra min e max
> c(min(peso),max(peso))
[1]  11 100
> #
> #Invarianza rispetto a trasformazioni lineari affini
> mean(5+2*peso)
[1] 115.3077
> 5+2*mean(peso)
[1] 115.3077
```

Boxplot

Il dataset *internet.csv* registra il numero di interruzioni nel funzionamento della rete internet di una Università, rilevato in un mese (Johnson, R., and Bhattachar yya, G. K., p. 28)

```
> crash=read.table("internet.csv",header=T,sep=";")
> par(mfrow=c(1,2))
> hist(crash$rotture,breaks=0:7,labels=T,right=F,freq=F)
> boxplot(crash$rotture,horizontal=T)
> #
> # Come PROMEMORIA riporto la tabella per modalità
> table(crash$rotture)

 0  1  2  3  4  6
 9 10  5  3  2  1
```

Il comando `boxplot` é molto semplice, `horizontal=T` ruota il la scatola
Le righe di comando sopra ci consentono di metterlo a confronto con l'istogramma, per evidenziare come i due grafici illustrano l'andamento di uno stesso fenomeno asimmetrico.
`labels=T` scrive le frequenze sopra le barre.

Come si legge il Boxplot?

1. La linea centrale evidenzia la mediana
2. Sopra e sotto la mediana ci sono il primo e il terzo quartile
3. la larghezza della scatola é detta **intervallo o scarto interquartile**, $\mathbf{SIQ = Q_3 - Q_1}$. Nel SIQ si posizionano almeno il 50% dei valori
4. Le linee al termine dei baffi sono poste a $Q_1 - 1.5SIQ$ (a sinistra) e $Q_3 + 1.5SIQ$ (a destra)
5. Le osservazione esterne alle linee che terminano il baffo sono indicate da pallini (potenziaili **outliers**)
6. Se non ci valori fuori dai baffi, il baffo termina nel valore intero precedente (successivo) alla osservazione più piccola (più grande), come nel nostro caso (baffo a sinistra)

Il box plot evidenzia come l'istogramma l'asimmetria della distribuzione. Tuttavia Q_1 e Q_2 sono equidistanti rispetto alla mediana, poiché circa l'80% della distribuzione é compresa nell'intervallo [0,2]

Un altro esempio

```
> cereals=read.table("cereals.txt",header=T,sep=";")
> summary(cereals$calorie)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   50.0   100.0   110.0   106.9   110.0   160.0
> summary(cereals$proteine)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   2.545   3.000   6.000
> summary(cereals$potassio)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   15.0    45.0    90.0   104.2   125.0   335.0
> par(mfrow=c(3,2))
> hist(cereals$calorie)
> boxplot(cereals$calorie,main="calorie")
> hist(cereals$proteine)
> boxplot(cereals$proteine,main="proteine")
> hist(cereals$potassio)
> boxplot(cereals$potassio,main="potassio")
```

Provate il comando `summary(cereals)`