

Lezione n. 8

8.1 Ancora sulle stime di massima verosimiglianza

ESEMPIO 8.1 [Esponenziale traslata]

Sia $X_i = \theta + W_i$ con W_i i.i.d $\sim \text{Exp}(1)$, $\theta \in \mathbb{R}$

- (i) derivare lo stimatore di massima verosimiglianza per θ
- (ii) definire $Y = \hat{\theta} - \theta$. Determinare la distribuzione di Y e conseguentemente ricavare la distorsione di $\hat{\theta}$.

SOLUZIONE. Essendo $X_i = \theta + W_i$, $\Pr(X_i \leq x) = \Pr(W_i \leq x - \theta) = \int_0^{x-\theta} e^{-v} dv = 1 - e^{-(x-\theta)} I(x > \theta)$. Quindi X_i ha una distribuzione esponenziale negativa traslata in θ e la densità di X_i è $f(x; \theta) = e^{-(x-\theta)} I(x > \theta)$.

La verosimiglianza per θ si scrive quindi come

$$l(\theta; \underline{x}) = \prod_i e^{-(x_i - \theta)} I_{(\theta, \infty)}(x_i) =$$

$$I_{(0, \min_{1 \leq i \leq n} x_i)}(\theta) e^{-\sum_i (x_i - \theta)} = I_{(0, x_{(1)})}(\theta) e^{-n(\bar{x} - \theta)} \propto I_{(0, x_{(1)})}(\theta) e^{n\theta}.$$

La verosimiglianza ha pertanto un andamento crescente (determinato da $e^{n\theta}$) per $\theta < x_{(1)}$, cioè fino al minimo campionario $x_{(1)} = \min_{1 \leq i \leq n} x_i$; si deduce immediatamente che la funzione assume il suo massimo in corrispondenza del minimo campionario. Lo stimatore di massima verosimiglianza è dunque $\hat{\theta} = X_{(1)}$

Per derivare la distribuzione di $Y = \hat{\theta} - \theta = X_{(1)} - \theta$, osserviamo che

$$\begin{aligned} \Pr(Y \leq y) &= \Pr(X_{(1)} - \theta \leq y) = 1 - \Pr(X_{(1)} > y + \theta) \\ &= 1 - \prod_{i=1}^n \Pr(X_i > y + \theta) = 1 - \prod_{i=1}^n \Pr(X_i - \theta > y). \end{aligned}$$

La distribuzione di ogni variabile aleatoria $X_i - \theta$ è di tipo esponenziale di parametro unitario. Pertanto $\Pr(X_i - \theta > y) = e^{-y}$. Di conseguenza

$$\Pr(Y \leq y) = 1 - \prod_{i=1}^n \Pr(X_i - \theta > y) = 1 - e^{-ny}$$

cioè la distribuzione di Y è di tipo esponenziale negativo di parametro n . Il valore atteso di $Y = \hat{\theta} - \theta$ è quindi $1/n$, di conseguenza la distorsione tende a zero al crescere di n . ■

Esercizio 8.1. [Segue da Esempio 8.1]

In riferimento all'Esempio 8.1, conoscendo la distribuzione di $Y = \hat{\theta} - \theta$, calcolare analiticamente il MSE di $\hat{\theta}$.

Fissato un valore per θ (ad esempio, estraendo un valore casuale da una distribuzione a propria scelta), simulare un campione dalla distribuzione delle X_i e stimare θ .

Simulare inoltre (per lo stesso valore di θ) la distribuzione dello stimatore di massima verosimiglianza $\hat{\theta}$ e approssimarne distorsione e MSE. Confrontare questi valori con quelli ottenuti analiticamente.

In base alla simulazione effettuata (o altrimenti), che distribuzione vi sembra seguire la variabile aleatoria $\hat{\theta}$?

SUGGERIMENTO: Per ottenere la stima di θ , procedere secondo i seguenti passi:

- simulare `theta`, ad esempio estraendolo da una distribuzione uniforme (comando `runif`).
- impostare il valore della numerosità campionaria: `n_18`
- simulare il campione dalla distribuzione esponenziale traslata: `x_theta+rexp(n)`
- determinare la stima di massima verosimiglianza.

Per ottenere una approssimazione della distribuzione campionaria di $\hat{\theta}$, costruire un vettore di realizzazioni di $\hat{\theta}$ utilizzando i seguenti comandi:

```
m_5000
#(o qualunque altro valore, sufficientemente elevato,
# compatibile con le potenzialità del vostro pc)
campioni.sim_matrix(theta+rexp(n*m),nrow=n)
```

dopodiché utilizzare il comando `apply` sulla matrice `campioni.sim` per ottenere il vettore delle stime di massima verosimiglianza sugli m campioni simulati.

Ottenere, in base ai valori simulati di $\hat{\theta}$ e al valore salvato del parametro `theta` una approssimazione di $E(\hat{\theta})$ e di $MSE(\hat{\theta})$ (in corrispondenza del solo valore vero `theta`) e confrontare questi valori con quelli teorici. ■

8.1.1 Stimatori MV e ottimizzazione multidimensionale

Quando il modello ipotizzato per la popolazione X da cui il campione è estratto dipende da più di un parametro, anche nel caso in cui l'interesse verta su uno solo dei parametri incogniti, è necessario procedere alla stima di tutte le quantità incognite del modello anzidetto (parametri *di interesse* e parametri eventualmente *di disturbo*).

L'esempio più semplice riguarda il modello normale con μ, σ^2 ambedue incogniti. In generale si procede secondo i seguenti passi: innanzitutto si costruisce una funzione che permetta di calcolare la verosimiglianza in corrispondenza di una coppia qualunque di parametri e dato un generico insieme di dati. Si noti che alla funzione vengono in questo caso assegnati due argomenti, dei quali il primo rappresenta il *vettore* (a due componenti) dei parametri da stimare, mentre il secondo rappresenta l'informazione campionaria (il campione osservato, oppure il vettore della statistica sufficiente). Si noti inoltre che il risultato restituito dalla funzione è la funzione di log-verosimiglianza *cambiata di segno*. Questo dipende dal fatto che il secondo passo da compiere è derivare numericamente, piuttosto che analiticamente, le stime di massima verosimiglianza, e per l'ottimizzazione numerica si hanno a disposizione funzioni R disegnate per *minimizzare* funzioni obiettivo.

Per l'ottimizzazione numerica multidimensionale si ricorre al comando R `optim`.

Trattandosi di una funzione basata su algoritmi numerici (tutti basati su una generalizzazione del metodo di Newton-Raphson), essa va avviata da un punto iniziale. Si ricordi inoltre che la caratteristica di questi metodi è di individuare punti di minimo *locale* e che per stabilire se si tratta di un punto di minimo assoluto bisogna studiare la funzione di log verosimiglianza, oppure far partire la funzione da diversi punti iniziali e verificare che produca gli stessi risultati.

Gli argomenti principali della funzione `optim` sono i seguenti:

`fn`= la funzione da ottimizzare

`par`= i valori iniziali per il parametro. Nel caso di parametro bidimensionale, si utilizza ad esempio la sintassi `par=c(theta1,theta2)`; `theta1` e `theta2` rappresentano i valori iniziali di ciascun parametro. Si noti che la funzione da ottimizzare deve avere tra gli argomenti un unico parametro vettoriale;

`method`= il metodo di ottimizzazione da utilizzare (si veda l'`help` per la funzione `optim`).

`lower`= e `upper`= consentono di specificare vincoli sui parametri. In questo caso il punto iniziale indicato con `par` deve soddisfare i vincoli specificati da `upper` e `lower`. La specificazione di vincoli implica l'uso di `method="L-BFGS-B"`.

`maxit`= il numero massimo di iterazioni.

Infine, tra gli argomenti di `optim` vanno inclusi quelli da passare alla funzione `fn` (si veda

l'Esempio 8.2).

L'output di base della funzione è una lista contenente le seguenti informazioni:

par: Il vettore dei migliori parametri individuati per **fn**.

value: Il valore di **fn** in corrispondenza di **par**.

counts: Un vettore di due elementi. Il primo rappresenta il numero di chiamate della funzione da ottimizzare, il secondo il numero di chiamate dell'argomento opzionale **gr**, che fornisce il gradiente della funzione da ottimizzare.

convergence: Valore intero che può assumere i seguenti valori :

0 indica convergenza raggiunta.

1 Codice di errore - numero massimo di iterazioni (**maxit**) raggiunto.

10 Codice di errore per il metodo Nelder-Mead, che è quello utilizzato per default.

51 Warning per il metodo "L-BFGS-B" (seguito da un messaggio: si veda oltre **message**).

52 Codice di errore per il metodo "L-BFGS-B" (seguito da un messaggio: si veda oltre **message**).

message: Una stringa alfanumerica che, ove necessario, fornisce informazioni aggiuntive relative all'algoritmo di ottimizzazione.

Si osservi che in **optim** l'argomento **control**= consente di specificare l'opzione **fnscale**, che introduce un fattore con cui riscaldare la funzione da ottimizzare. L'ottimizzazione viene in tal caso effettuata su $\text{fn}(\text{par})/\text{fnscale}$. Di conseguenza, l'opzione **control=c(fnscale=-1)** consente di riformulare il problema come un problema di *massimizzazione*.

ESEMPIO 8.2 [*Stime di massima verosimiglianza per un modello normale*]

Si supponga di avere osservato un campione da una distribuzione normale con media e varianza incogniti. Naturalmente, il problema di trovare i parametri che massimizzano la log-verosimiglianza $\log l(\mu, \sigma; \underline{x})$ è risolvibile anche come un problema di minimizzazione di $-\log l(\mu, \sigma^2; \underline{x})$ rispetto a μ, σ^2 .

La funzione di log verosimiglianza,

$$\log l(\mu, \sigma; \underline{x}) = -n \log(\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^n (X_i - \mu)^2$$

può essere scritta in una funzione R come:

```
norm.llik <- function(vecpar, dati){
mu <- vecpar[1]
```

```
sigma2 <- vecpar[2]
sigma <- sqrt(sigma2)
n <- length(dati)
sqdiff <- (dati-mu)^2/sigma2
somma <- sum(sqdiff)
ll <- -n*log(sigma)-somma/2
return(ll)}
```

A puro scopo esemplificativo, supponiamo di aver estratto un campione di numerosità $n = 100$ da una normale con media 3 e varianza 4: scriviamo dunque `x <- rnorm(100,3.1,2)`.

Per calcolare la funzione di log-verosimiglianza in corrispondenza di una qualunque coppia di parametri, ad esempio il vettore (0,1) scriviamo:

```
norm.llik(c(0.1),dati=x).
```

La funzione di verosimiglianza normale ha una forma di paraboloide che sarà evidente da una rappresentazione grafica tridimensionale (argomento che verrà affrontato nella prossima lezione). La soluzione analitica al problema di massimo fornisce la soluzione $\hat{\mu} = \frac{1}{n} \sum_i x_i$ e $\hat{\sigma}^2 = \frac{1}{n} \sum_i (x_i - \bar{x})^2$. Il punto di massimo $(\hat{\mu}, \hat{\sigma}^2)$ si può anche ricavare numericamente con il comando `optim`; in questo caso scegliamo per l'algoritmo il punto iniziale (0,0.1):

```
optlik <- optim(par=c(0,0.1),fn=norm.llik,
               lower=c(-Inf,0),upper=c(Inf,Inf),dati=x, control=c(fnscale=-1))
```

```
>optlik
```

```
$par
```

```
[1] 3.162980 3.810393
```

```
$value
```

```
[1] -116.8854
```

```
$counts
```

```
function gradient
```

```
21 21
```

```
$convergence
```

```
[1] 0
```

```
$message
```

```
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

Warning message:

```
bounds can only be used with method L-BFGS-B in:
```

```
optim(par = c(0, 0.01), fn = norm.llik, lower = c(-Inf, 0.01),
```

Confrontiamo i risultati ottenuti numericamente (`optlik$par`) con quelli ottenibili tramite l'espressione analitica degli stimatori di massima verosimiglianza per il caso normale ($\hat{\mu} = \bar{x}$; $\hat{\sigma}^2 = 1/n \sum_i (x_i - \bar{x})^2$): a questo scopo calcoliamo

```
> optlik$par
```

```
[1] 3.162980 3.810393
```

```
> mean(x)
```

```
[1] 3.163029
```

```
> n <- length(x)
```

```
> (n-1)/n*var(x)
```

```
[1] 3.810304
```

Si ottengono risultati sostanzialmente identici.

In alcuni casi non è possibile determinare analiticamente le stime massima verosimiglianza; in tal caso ricorrere a metodi numerici è l'unica strada. Si veda l'esercizio successivo per un esempio.

Esercizio 8.2. [Stima MV per un modello Gamma]

Supponiamo che i dati nel file `surv.txt`, relativi alla durata di vita di 20 cavie sottoposte a radiazioni, siano distribuite secondo una legge Gamma: X_i i.i.d. $\sim \text{Gamma}(k, \alpha)$, $i = 1, \dots, 20$, con densità comune

$$f_X(x; \alpha, \lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{(\alpha-1)} e^{-\lambda x}, \quad x > 0$$

(nella formalizzazione di R si tratta di una Gamma con parametri `shape`= α e `rate`= λ).

Procedendo secondo quanto fatto nell'esempio 8.2, scrivere la funzione di log verosimiglianza e derivare le stime MV per α, λ sulla base del campione osservato.

SUGGERIMENTO: *La funzione di log verosimiglianza per α, λ dato un campione di n osservazioni i.i.d. da una legge Gamma,*

$$\ell(\alpha, \lambda; \underline{x}) = n\alpha \log \lambda - n \log \Gamma(\alpha) + (\alpha - 1) \sum_{i=1}^n \log(x_i) - \lambda \sum_{i=1}^n x_i$$

si può scrivere in una funzione R come

```
gamma.llik_function(pars,dati)
{a <- pars[1]
 l <- pars[2]
 sommalog <- sum(log(dati))
 somma <- sum(dati)
 n <- length(dati)
 ll <- n*a*log(l)-n*log(gamma(a))+(a-1)*sommalog-l*somma
 return(ll)}
```

oppure, sfruttando la funzione `dgamma` con l'opzione `log=T`, che fornisce il logaritmo della densità gamma, come:

```
gamma2.llik_function(pars,dati)
{ a <- pars[1]
  l <- pars[2]
  ll <- sum(dgamma(dati, a,rate=1, log=T))
  return(ll) }
```

Per leggere in R il file di dati `surv.txt`, si utilizzi il comando `scan` (che serve per leggere da file insieme di dati numerici): supponendo di avere il file nella cartella `datir` della directory personale `nomeutente`, il percorso che individua il file è `c:\winnt\profiles\nomeutente\datir\surv.txt`

Per leggere il file si scriverà dunque

```
surv.data_scan('c:\\winnt\\profiles\\nomeutente\\datir\\surv.txt')
```

oppure

```
surv.data_scan('c:/winnt/profiles/nomeutente/datir/surv.txt')
```

■

8.2 Verifica di ipotesi e ottimalità delle statistiche test

In alcuni casi esistono risultati teorici che permettono di derivare, per un dato problema di verifica di ipotesi, una statistica test ottimale. Il caso più elementare è quello descritto nel Lemma di Neyman e Pearson, in cui le ipotesi sono ambedue puntuali. Il Lemma riguarda tuttavia un problema di ipotesi particolarmente semplice; il suo merito principale consiste nel mettere in evidenza il ruolo della funzione di verosimiglianza e della sufficienza per

giungere a determinare una “buona” statistica test. In casi più complessi, spesso non è possibile individuare il test UMP. Esistono comunque problemi in cui il sistema di ipotesi e il modello parametrico sono tali da consentire di costruire un test UMP, a partire da quello suggerito dal Lemma di Neyman e Pearson. In base al teorema di Karlin e Rubin, test UMP possono essere trovati se $f_\theta(x)$ ammette una statistica sufficiente T_n per θ e il modello ha rapporto di verosimiglianza monotono in T_n . Se, per $\theta_1 < \theta_2$, il rapporto delle verosimiglianze $l(\underline{x}, \theta_1)/l(\underline{x}, \theta_2) = h(T_n, \theta_1, \theta_2)$ (che dipende dalla statistica sufficiente T_n , oltre che dai due parametri) è una funzione monotona non decrescente in T_n , allora il Teorema di Karlin e Rubin assicura che per il sistema di ipotesi $H_0 : \theta \leq \theta_0$ contro $H_1 : \theta > \theta_0$, il test con regione critica $\{\underline{x} : T_n(x) < t_\alpha\}$ è un test ottimo di livello $\alpha = \int_{-\infty}^{t_\alpha} f_{T|\theta_0}(t)dt$. In effetti in questo caso valori grandi della statistica sufficiente corrispondono a valori elevati del rapporto delle verosimiglianze, indicando pertanto evidenza sperimentale in favore dell’ipotesi nulla. Se il rapporto di verosimiglianza è monotono non crescente il test ottimo ha regione critica $\{\underline{x} : T_n(x) > t_\alpha\}$, con $\alpha = \int_{t_\alpha}^{\infty} f_{T|\theta_0}(t)dt$. In questo caso infatti valori elevati della statistica sufficiente corrispondono a valori piccoli del rapporto delle verosimiglianze cioè a evidenza sperimentale contro l’ipotesi nulla. Possiamo notare che, per un sistema di ipotesi ambedue puntuali, il Lemma di Neyman e Pearson, per il tramite della verosimiglianza, suggerisce che un buon test deve essere basato su una statistica sufficiente T_n per il parametro di interesse θ . Il risultato di Karlin e Rubin conferma tale suggerimento, estendendolo a una classe di sistemi di ipotesi (unilaterali) più generali.

ESEMPIO 8.3 [Test di ipotesi]

Per un campione di $n = 10$ elementi estratto dalla legge con densità

$$f(x; \lambda) = \lambda^2 x \exp\{-\lambda x\} I_{[0, \infty)}(x) \quad (8.1)$$

si vuole verificare il sistema di ipotesi $H_0 : \lambda \leq 1$ contro $H_1 : \lambda > 1$. Si consideri il test che rifiuta l’ipotesi H_0 per i campioni $\{\underline{x} : \bar{x} < 1.1\}$ e se ne analizzi la funzione di potenza.

Si costruisca poi il test ottimo al livello $\alpha = 0.05$ per lo stesso sistema di ipotesi e se ne disegni la funzione di potenza.

SOLUZIONE. Notiamo che la densità (8.1) è un caso di legge Gamma(α, λ) con $\alpha = 2$; la verosimiglianza di λ è proporzionale a $l(\lambda; \underline{x}) = \lambda^{2n} \prod_{i=1}^n x_i \exp\{-\lambda \sum_{i=1}^n x_i\}$. Ne deduciamo che $T_n = \sum_{i=1}^n X_i$ è una statistica sufficiente per λ .

Analizziamo prima di tutto la funzione test con regione di rifiuto $\mathcal{R} = \{\underline{x} : \bar{x} < 1.1\}$, ossia $\mathcal{R} = \{\underline{x} : \sum_i x_i < 11\}$. La sua funzione di potenza è

$$\pi(\lambda) = P_\lambda\{\underline{X} \in \mathcal{R}\} = \Pr\left\{\sum_i X_i < 11 \mid \lambda\right\} \quad (8.2)$$

e va valutata tanto per $\lambda \leq 1$ quanto per $\lambda > 1$. Per determinare la funzione di potenza del test con regione critica indicata bisogna quindi conoscere la distribuzione di $T_n = \sum_i X_i$. Essendo le osservazioni indipendenti di tipo $\text{Gamma}(2, \lambda)$, T_n ha una distribuzione ancora di tipo Gamma , con parametri $2n = 20$ e λ . La probabilità in (8.2) è dunque la probabilità della coda sinistra di una densità Gamma con i parametri anzidetti. Per un λ assegnato, ad esempio $\pi(\cdot)$ è calcolabile tramite il comando `pgamma`: ad esempio

```
#lambda=0.7 (H_0):
> pgamma(11, 20,0.7)
[1] 0.0001556389
#lambda=1 (H_0):
> pgamma(11, 20,1)
[1] 0.009289458
```

Per disegnare la funzione (8.2) scriviamo dunque:

```
curve(pgamma(11,20,x), 0.01,5,xlab='lambda',ylab='potenza')
```

Si osserva immediatamente che la funzione di potenza è una funzione crescente. Questo implica che l'ampiezza del test con regione critica $\mathcal{R} = \{\underline{x} : \bar{x} < 1.1\}$ è

$$\alpha = \sup_{\lambda \leq 1} \pi(\lambda) = \pi(1).$$

L'estremo superiore delle probabilità di rifiuto quando è vera l'ipotesi nulla è dunque raggiunto, in corrispondenza di $\lambda = 1$, e nella fattispecie è pari a

```
> pgamma(11,20,1)
[1] 0.009289458
```

Questa caratteristica della funzione di potenza si osserva in questo problema di ipotesi per qualunque test con regione critica $\mathcal{R} = \{\underline{x} : \sum_i x_i < t^*\}$ (infatti il risultato non dipende dal particolare valore fissato per t^*), ad esempio per il valore $t^* = 11$ già utilizzato.

Per determinare il test uniformemente più potente, osserviamo che siamo in presenza di una famiglia con rapporto di verosimiglianza monotono: il rapporto delle verosimiglianze valutate in due punti λ_1, λ_2 con $\lambda_1 < \lambda_2$ è infatti

$$\frac{\lambda_1^{2n} \exp\{-\lambda_1 \sum_{i=1}^n x_i\}}{\lambda_2^{2n} \exp\{-\lambda_2 \sum_{i=1}^n x_i\}} = \left(\frac{\lambda_1}{\lambda_2}\right)^{2n} \exp\{-(\lambda_1 - \lambda_2) \sum_{i=1}^n x_i\};$$

essendo $\lambda_1 - \lambda_2 < 0$, si tratta di una funzione monotona non decrescente di $\sum_i X_i$.

Siamo pertanto in presenza di rapporto di verosimiglianza monotono non decrescente rispetto alla statistica sufficiente $T_n = \sum_{i=1}^n X_i$.

In questo caso il teorema di Karlin-Rubin assicura che per il sistema di ipotesi $H_0 : \lambda \leq 1$ contro $H_1 : \lambda > 1$ esiste un test ottimo nella classe dei test aventi lo stesso livello, e tale test è basato sulla statistica sufficiente $T_n = \sum_{i=1}^n X_i$; la regione di rifiuto è del tipo $\{T < t^*\}$ dove t^* va determinato in modo che $\alpha = \Pr\{T < t^* | \lambda = 1\}$ ossia $\int_0^{t^*} f_{T|\lambda=1}(u) du = \alpha$.

Per costruire la regione di rifiuto del test uniformemente più potente per il sistema di ipotesi indicato, bisogna quindi conoscere la distribuzione di $T = \sum_i X_i$ in corrispondenza di $\lambda = 1$. Come già osservato, T ha una distribuzione di tipo Gamma, con parametri $2n$ e λ . Per trovare il valore critico t^* in modo che il test risultante abbia livello $\alpha = 0.05$, dobbiamo pertanto individuare il quantile di livello 0.05 di una gamma con parametri $(20, 1)$. Scriviamo pertanto

```
> qgamma(0.05,20,rate=1)
[1] 13.25465
```

Pertanto la regione critica del test UMP di livello $\alpha = 0.05$ per il nostro problema di ipotesi è $\mathcal{R} = \{(x_1, \dots, x_{10}) : \sum_{i=1}^{10} x_i < 13.255\}$.

Per disegnare la funzione di potenza del test ottimo di livello α generico, basta scrivere la funzione

```
potenza.gamma_function(alpha=0.05,lam0=1)
{
t.star <- qgamma(alpha,20,rate=lam0)
print(paste('valore critico del test:', t.star))
curve(pgamma(t.star,20, rate=x), 0.01,5,
main=paste('potenza del test con regione di rifiuto {T<', round(t.star,2),'}' ))
abline(h=0.05,col=4,lty=2)
abline(v=lam0,col=4,lty=2)
axis(1,seq(0,lam0,length=2),labels=c(' ',expression(lambda[0])),col.axis='blue')
axis(2,seq(0,alpha,length=2),labels=c(' ',expression(alpha)),col.axis='blue')
}
potenza.gamma()
```

A puro scopo esemplificativo, estraiamo un campione di 10 elementi da una legge $\text{Gamma}(2, 0.04)$: utilizziamo il comando `x <- rgamma(10,2,rate=0.04)`. Verifichiamo se il test UMP di livello 0.05 appena costruito porta o meno all'accettazione dell'ipotesi nulla (che per il campione estratto sappiamo essere vera).

Per verificare l'ipotesi nulla possiamo scrivere ad esempio:

```
test.gamma2_function(dati=x, alpha=0.05, lam0=1)
{
print(paste('H_0: lambda<= ', lam0))
n_length(dati)
t.star <- qgamma(alpha,2*n,rate=lam0)
print(paste('valore critico al livello alfa=',alpha,': ', t.star))
t.oss <- sum(dati)
print(paste('statistica osservata',t.oss))
test <- ifelse(t.oss<t.star, 'rifiuto H_0', 'non rifiuto H_0' )
return(test)
}

> test.gamma2(rgamma(10,2,0.04),0.05,1)
[1] "H_0: lambda<= 1"
[1] "valore critico al livello alfa= 0.05 : 13.2546515983466"
[1] "statistica osservata 595.858700053132"
[1] "non rifiuto H_0"
```

Quindi il campione estratto -correttamente, essendo il vero $\lambda = 0.04$ - non conduce al rifiuto dell'ipotesi nulla. ■

Esercizio 8.3. *[Segue da Esempio 8.3]*

Basandosi sulla costruzione sopra utilizzata per costruire la regione di rifiuto ottimale per una data ampiezza del test, verificare al livello $\alpha = 0.01$ il sistema di ipotesi indicato all'Esempio 8.3 sulla base del seguente campione di 24 unità estratto dal modello (8.1):

$$\underline{x} = (3, 13, 2, 3, 6, 22, 2, 18, 3, 7, 9, 12, 4, 1, 2, 7, 7, 3, 5, 2, 9, 18, 1, 7).$$

Rappresentare anche la funzione di potenza del test costruito.

SUGGERIMENTO: *Il campione può essere importato dal file `dati.txt` con il comando `scan()`: supponendo ad esempio che il file sia su dischetto il comando è il seguente: `dati <- scan("a:\dati.txt")`. Altrimenti si possono inserire i dati manualmente usando la sintassi `dati <- scan()` senza alcun argomento e digitando i dati uno ad uno separandoli con un invio (un doppio invio conclude il comando), oppure ancora tagliando e incollando i dati.*

In fase di costruzione della regione di rifiuto si tenga conto del fatto che la numerosità campionaria è in questo caso $n = 24$ e che $\alpha = 0.01$. ■

Esercizio 8.4. [*Ipotesi puntuali*]

Con riferimento all'esempio 8.3, supponiamo che il sistema di ipotesi sia $H_0 : \lambda = 1$ contro $H_1 : \lambda = 2$. Costruire il test UMP al livello $\alpha = 0.05$ utilizzando il lemma di Neyman e Pearson.

Che cambiamenti si ottengono nella procedura di test modificando il valore di λ specificato dall'ipotesi alternativa in modo che $\lambda_1 > \lambda_0$?

Verificare il sistema di ipotesi indicato alla luce del seguente risultato sperimentale:

$$\sum_{i=1}^{10} x_i = 8.688,$$

osservato su un campione di 10 elementi estratto dal modello (8.1).