A large, faint watermark of the University of Udine seal is visible in the background. It features a circular design with the text 'STUDIO UDI' at the top and 'UTINENSIS' at the bottom. In the center is a heraldic figure, possibly a lion or a similar creature, with its wings spread.





Università degli studi di Udine
Dipartimento di Matematica e Informatica
Laurea specialistica in Informatica

Un pacchetto R: BETAREG

Applied statistics and data analysis

Riouak Idriss
Marta Rotari

9 Gennaio 2018

-  Cribari-Neto Francisco, Achim Zeileis *Beta Regression in R* (2006).
-  Ferrari SLP, Cribari-Neto Francisco *Beta Regression for Modelling Rates and Proportions* (2004).
-  Simas AB, BarretoSouza W, Rocha AV *Improved Estimators for a General Class of Beta Regression Models* (2010)
-  Paolo Vidoni *Towards multiple linear regression and logistic regression* 2017-2018. Lecture 5. Applied Statistics and Data Analysis.



Un modello di regressione ha lo scopo di studiare le relazioni tra una variabile (y) detta variabile di **risposta**, e una o più variabili **regressori** (x). Inoltre permette di effettuare predizioni dato un nuovo valore per la variabile x .



Un modello di regressione ha lo scopo di studiare le relazioni tra una variabile (y) detta variabile di **risposta**, e una o più variabili **regressori** (x). Inoltre permette di effettuare predizioni dato un nuovo valore per la variabile x .

Nel 2004 viene definito il **Modello di regressione con variabile risposta Beta** [2] per variabili continue con valori in $(0, 1)$, come possono essere *proporzioni* e *tassi*, assumendo che la variabile risposta sia beta-distribuita.

Un modello di regressione ha lo scopo di studiare le relazioni tra una variabile (y) detta variabile di **risposta**, e una o più variabili **regressori** (x). Inoltre permette di effettuare predizioni dato un nuovo valore per la variabile x .

Nel 2004 viene definito il **Modello di regressione con variabile risposta Beta** [2] per variabili continue con valori in $(0, 1)$, come possono essere *proporzioni* e *tassi*, assumendo che la variabile risposta sia beta-distribuita.

Successivamente nel 2006, nell'articolo «Beta Regression in R» [1] ne viene fornita un'implementazione in R.

La variabile di risposta y ha una funzione di distribuzione di probabilità continua definita da due parametri sull'intervallo $(0, 1)$.

$$f(y; \mu, \phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} y^{\mu\phi-1} (1-y)^{(1-\mu)\phi-1}$$

con $0 < y < 1$ dove $0 < \mu < 1$, $\phi > 0$ e

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt.$$

Denoteremo $y \sim \mathcal{B}(\mu, \phi)$ la variabile casuale con parametri μ e ϕ (detto parametro di precisione):

$$E(y) = \mu \text{ e } V(y) = \frac{\mu(1-\mu)}{1+\phi}$$

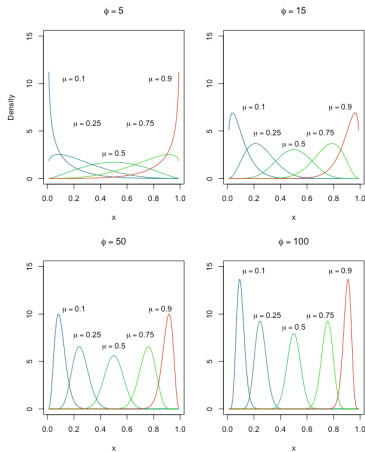


Figura: Rappresentazione grafica della distribuzione Beta

Sia y_1, y_2, \dots, y_n un **campione casuale** tale che $y_i \sim \mathcal{B}(\mu_i, \phi)$ il modello di regressione con variabile di risposta *beta distribuita* è

$$g(\mu_i) = x_i^t \beta = \eta_i$$

- $\beta = (\beta_1, \beta_2, \dots, \beta_k)^t$, con $k < n$, vettore dei coefficienti,

Sia y_1, y_2, \dots, y_n un **campione casuale** tale che $y_i \sim \mathcal{B}(\mu_i, \phi)$ il modello di regressione con variabile di risposta *beta distribuita* è

$$g(\mu_i) = x_i^t \beta = \eta_i$$

- ▶ $\beta = (\beta_1, \beta_2, \dots, \beta_k)^t$, con $k < n$, vettore dei coefficienti,
- ▶ $x_i = (x_{i1}, x_{i2}, \dots, x_{ik})^t$, per convenzione $x_{i1} = 1$,

Sia y_1, y_2, \dots, y_n un **campione casuale** tale che $y_i \sim \mathcal{B}(\mu_i, \phi)$ il modello di regressione con variabile di risposta *beta distribuita* è

$$g(\mu_i) = x_i^t \beta = \eta_i$$

- ▶ $\beta = (\beta_1, \beta_2, \dots, \beta_k)^t$, con $k < n$, vettore dei coefficienti,
- ▶ $x_i = (x_{i1}, x_{i2}, \dots, x_{ik})^t$, per convenzione $x_{i1} = 1$,
- ▶ $\eta_i = \beta_1 x_{i1} + \dots + \beta_k x_{ik}$ predittore lineare,

Sia y_1, y_2, \dots, y_n un **campione casuale** tale che $y_i \sim \mathcal{B}(\mu_i, \phi)$ il modello di regressione con variabile di risposta *beta distribuita* è

$$g(\mu_i) = x_i^t \beta = \eta_i$$

- ▶ $\beta = (\beta_1, \beta_2, \dots, \beta_k)^t$, con $k < n$, vettore dei coefficienti,
- ▶ $x_i = (x_{i1}, x_{i2}, \dots, x_{ik})^t$, per convenzione $x_{i1} = 1$,
- ▶ $\eta_i = \beta_1 x_{i1} + \dots + \beta_k x_{ik}$ predittore lineare,
- ▶ $g(\cdot) : (0, 1) \rightarrow \mathbb{R}$ è una funzione \mathcal{C}^2 , detta funzione di collegamento, avente derivata seconda costante.

Le funzioni di collegamento più utilizzate sono:

- ▶ **logit:** $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$
- ▶ **probit:** $g(\mu) = \Phi^{-1}(\mu)$, dove $\Phi(\cdot)$ è la funzione di distribuzione normale standard.
- ▶ **log-log complementare:**
 $g(\mu) = \log(-\log(1 - \mu))$
- ▶ **log-log:** $g(\mu) = \log(-\log(\mu))$
- ▶ **Cauchy:** $g(\mu) = \tan(\pi(\mu - 0.5))$

Nel 2010 è stata formulata un'estensione del modello in modello di regressione beta a dispersione variabile che considera il parametro di precisione non più costante.

Le osservazioni $y_i \sim \mathcal{B}(\mu_i, \phi_i)$ indipendenti con

$$g_1(\mu_i) = \eta_{1i} = x_i^T \beta,$$

$$g_2(\phi_i) = \eta_{2i} = z_i^T \gamma,$$

dove $\beta = (\beta_1, \dots, \beta_k)^T$ e $\gamma = (\gamma_1, \dots, \gamma_h)^T$ con $k + h < n$ insiemi dei coefficienti di regressione, x_i e z_i vettori di regressori e η_{1i} e η_{2i} predittori lineari.

- ▶ Il pacchetto **betareg** è una **collezione** di funzioni implementate in “R ”, con il quale è possibile modellare variabili dipendenti beta distribuite.
- ▶ Le versioni dalla 1.0 alla 1.2 sono state implementate da *Simas e Rocha* fino al 2006. Dalla versione 2.0, il principale contribuente è stato *Achim Zeileis* coautore dell’articolo
- ▶ La main-function `betareg()` è stata progettata e implementata per essere il più simile possibile alla funzione standard `glm()` (General Linear Model).

```
betareg(formula, data, subset, na.action, weights,
offset, link = "logit", link.phi = NULL, control =
  betareg.control(...), model = TRUE, y = TRUE, x =
  FALSE, ...)
```

- formula: descrizione simbolica del modello, e.g.: $y \sim x+z$

```
betareg(formula, data, subset, na.action, weights,  
offset, link = "logit", link.phi = NULL, control =  
  betareg.control(...), model = TRUE, y = TRUE, x =  
  FALSE, ...)
```

- ▶ formula: descrizione simbolica del modello, e.g.: $y \sim x+z$
- ▶ data, subset: sorgente dei dati.


```
betareg(formula, data, subset, na.action, weights,  
offset, link = "logit", link.phi = NULL, control =  
  betareg.control(...), model = TRUE, y = TRUE, x =  
  FALSE, ...)
```

- ▶ formula: descrizione simbolica del modello, e.g.: $y \sim x+z$
- ▶ data, subset: sorgente dei dati.
- ▶ na.action: funzione che descrive come comportarsi dinanzi a elementi NA.

```
betareg(formula, data, subset, na.action, weights,  
offset, link = "logit", link.phi = NULL, control =  
  betareg.control(...), model = TRUE, y = TRUE, x =  
  FALSE, ...)
```

- ▶ formula: descrizione simbolica del modello, e.g.: $y \sim x+z$
- ▶ data, subset: sorgente dei dati.
- ▶ na.action: funzione che descrive come comportarsi dinanzi a elementi NA.
- ▶ weights: vettore di pesi.

```
betareg(formula, data, subset, na.action, weights,
offset, link = "logit", link.phi = NULL, control =
  betareg.control(...), model = TRUE, y = TRUE, x =
  FALSE, ...)
```

- ▶ formula: descrizione simbolica del modello, e.g.: $y \sim x+z$
- ▶ data, subset: sorgente dei dati.
- ▶ na.action: funzione che descrive come comportarsi dinanzi a elementi NA.
- ▶ weights: vettore di pesi.
- ▶ link: specifica la funzione di collegamento. Il valore di default è logit. Le possibili scelte sono: probit, cloglog, cauchit, log, loglog

```
betareg(formula, data, subset, na.action, weights,
offset, link = "logit", link.phi = NULL, control =
  betareg.control(...), model = TRUE, y = TRUE, x =
  FALSE, ...)
```

- `link.phi`: specifica la funzione di collegamento per il parametro di precisione ϕ Le scelte possibili sono:

```
betareg(formula, data, subset, na.action, weights,
offset, link = "logit", link.phi = NULL, control =
  betareg.control(...), model = TRUE, y = TRUE, x =
  FALSE, ...)
```

- ▶ `link.phi`: specifica la funzione di collegamento per il parametro di precisione ϕ La scelte possibili sono:
 - ▶ `sqrt`

```
betareg(formula, data, subset, na.action, weights,
offset, link = "logit", link.phi = NULL, control =
  betareg.control(...), model = TRUE, y = TRUE, x =
  FALSE, ...)
```

- ▶ `link.phi`: specifica la funzione di collegamento per il parametro di precisione ϕ Le scelte possibili sono:
 - ▶ `sqrt`
 - ▶ `log`

```
betareg(formula, data, subset, na.action, weights,
offset, link = "logit", link.phi = NULL, control =
  betareg.control(...), model = TRUE, y = TRUE, x =
  FALSE, ...)
```

- ▶ `link.phi`: specifica la funzione di collegamento per il parametro di precisione ϕ Le scelte possibili sono:
 - ▶ `sqrt`
 - ▶ `log`
- ▶ `control`: prende come parametro un oggetto di tipo `betareg.control`.

```
betareg(formula, data, subset, na.action, weights,
offset, link = "logit", link.phi = NULL, control =
  betareg.control(...), model = TRUE, y = TRUE, x =
  FALSE, ...)
```

- ▶ `link.phi`: specifica la funzione di collegamento per il parametro di precisione ϕ Le scelte possibili sono:
 - ▶ `sqrt`
 - ▶ `log`
- ▶ `control`: prende come parametro un oggetto di tipo `betareg.control`.
- ▶ `model`, `x`, `y`: argomenti di tipo logico (`TRUE`, `FALSE`). Se impostati a `TRUE`, vengono restituiti il *model.frame*, *model matrix* e il *vettore della variabile risposta* rispettivamente.

Esempi su Shiny



Lopzione `control` prende come parametro un oggetto di tipo `betareg.control`. Tali oggetti servono per controllare la modalità con la quale veogno stimati i coefcienti del modello.

```
betareg.control(phi = TRUE, method = "BFGS", maxit =  
  5000, hessian = FALSE, trace = FALSE, start = NULL  
  , fsmaxit = 200, fstol = 1e-8, ...)
```

- ▶ `phi`: valore booleano. Indica se il parametro ϕ deve essere trattato come un parametro del modello o come un parametro di disturbo.
- ▶ `method`: specifica quale metodo numerico viene utilizzato per stimare i coefficienti. Il valore di default è BFGS.
- ▶ `maxit`: indica il numero massimo di iterate da eseguire.

- ▶ `trace`: valore booleano. Indica se deve essere mantenuta traccia delle iterazioni effettuate durante la stima dei coefficienti.
- ▶ `hessian`: valore booleano. Indica se deve essere utilizzato l'Hessiano per stimare la matrice delle covariate. L'opzione di default è FALSE.
- ▶ `start`: un vettore di valori opzionali che indica quali sono in punti di partenza per stimare i coefficienti del modello. Si veda la sezione III.i.
- ▶ `fsmaxit`: valore intero. Indica il numero massimo delle iterazioni del *punteggio di Fisher*.
- ▶ `fstol`: valore numerico. Indica la tolleranza dell'errore per raggiungere la convergenza.

L'oggetto restituito dal modello stimato della classe `betareg` è una lista simile a quella restituita dagli oggetti `glm`, dove troviamo `coefficients` e `terms`. E' possibile interrogare gli oggetti della classe `betareg` attraverso la funzione `summary()`. Nella seguente tabella è stata riportata una lista dei metodi e delle funzioni offerte dagli oggetti della classe `betareg`.

FUNZIONE	DESCRIZIONE
<code>print()</code> <code>summary()</code>	Stampa su standard output i coefficienti stimati. Stampa su standard output la stima dei coefficienti, l'errore standard e il test parziale di Wald. Restituisce un oggetto della classe <code>summary.betareg</code> .

<code>coef()</code>	Restituisce tutti i coefficienti del modello, compresi intercetta e coefficiente di precisione.
<code>vcoef()</code>	Restituisce la matrice della covarianza.
<code>predict()</code>	Funzione di predizione del valor atteso, dei predittori lineari, del parametro di precisione e delle varianze.
<code>fitted()</code>	Valori attesi stimati per un nuovo vettore di osservazioni.
<code>residual()</code>	Restituisce il vettore dei residui.
<code>terms()</code>	Restituisce i componenti del modello.
<code>model.matrix()</code>	Restituisce la <i>Matrice del Modello</i>
<code>model.frame()</code>	Restituisce l'intero frame di dati del modello.
<code>loglik()</code>	Restituisce la stima di log-verosimiglianza.

<code>plot()</code>	Stampa sul device grafico i plot dei residui, delle predizioni, dei punti di leva etc.
<code>hatvalues</code>	Restituisce un vettore di elementi rappresentanti la diagonale della matrice <i>hat</i> .
<code>cooks.distance</code>	Restituisce un'approssimazione della distanza di Cook.
<code>glevarage()</code>	Restituisce un vettore di elementi con rappresentanti il valore di leva di ogni punto.

<code>coeftest()</code> <code>waldtest()</code> <code>linear.hypotesis()</code> <code>AIC()</code>	Test parziale di Wald dei coefficienti. Test di Wald per modelli annidati. Test di Wald per ipotesi lineari. Calcola l'Aikaie Information Criteria (AIC) e altri Information Criteria come il BIC.
-------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
predict(object, newdata = NULL, type = c("response", "link", "precision", "variance", "quantile"), na.action = na.pass, at = 0.5, ...)
```

- ▶ object: fitted model object della classe betareg
- ▶ newdata: opzionale, un data frame nel quale cercare delle variabili con le quali fare predizione. Se omesso, vengono usate le osservazioni originali
- ▶ type: character che indica il tipo di predizione:
 - ▶ response: fitted means of response,
 - ▶ link:corresponding linear predictor,
 - ▶ precisionfitted precision parameter phi,
 - ▶ variancefitted variances of response,
 - ▶ quantilefitted quantile(s) of the response distribution

- `at`: vettore numerico che indica il livello al quale i quantili vengono predetti se nel parametro `textbftype="quantile"` di default è assegnata la mediana **`at=0.5`**

```
data("GasolineYield", package = "betareg")
gy2 <- betareg(yield ~ batch + temp | temp, data =
  GasolineYield)
cbind(predict(gy2, type = "response"),
predict(gy2, type = "link"),
predict(gy2, type = "precision"),
predict(gy2, type = "variance"),
predict(gy2, type = "quantile", at = c(0.25, 0.5,
  0.75)))
```

E' possibile determinare un intervallo di confidenza $(1 - \alpha)100\%$ per i coefficienti $\hat{\beta}_j$, con $j = 1, \dots, k$. Tale intervallo è:

$$\left[\hat{\beta}_j \pm \Phi^{-1} \left(1 - \frac{\alpha}{2} SE(\hat{\beta}_j) \right) \right],$$

dove $\Phi(\cdot)$ è la f.d.p di una normale standard.
Analogamente per il parametro $\hat{\phi}$ è il seguente

$$\left[\hat{\phi} \pm \Phi^{-1} \left(1 - \frac{\alpha}{2} SE(\hat{\phi}) \right) \right]$$

dove $SE(\hat{\phi}) = \sqrt{tr(D) - \phi^{-1} c^t T^t X (X^t W X)^{-1} X^t T c} = \sqrt{\hat{\gamma}}$.