R workshop #4: Modelli lineari generalizzati per l'analisi di dati categorici e limitati

Nicola Romanò

Introduzione

Nei workshop precedenti ci siamo occupati principalmente di situazioni in cui è stata misurata una variabile continua e abbiamo voluto spiegare la sua variabilità in funzione di una o più variabili continue o discrete. Usiamo le variazioni del modello lineare per farlo (ricordate, ANOVA può anche essere considerato solo come un modello lineare).

Tuttavia, ci sono situazioni in cui un modello lineare non è la soluzione migliore da usare.

Alcuni esempi:

- Dati in cui misuriamo una variabile binaria (*p.es.* il soggetto ha il diabete? Sì / No) o una proporzione / probabilità (*p.es.* quali sono le probabilità di ottenere la patologia A, a seconda della variabile B?). Entrambi questi casi sono limitati tra 0 e 1 (nel primo caso la variabile può essere solo 0 o 1) o, se preferisci, 0% e 100%.
- Conteggi. Questi sono numeri interi, quindi hanno un limite inferiore uguale a 0 (non puoi contare -20 cellule!).

I modelli lineari sono molto potenti, ma sono problematici da utilizzare con dati limitati o discreti, in quanto presuppongono un intervallo continuo di valori che può assumere qualsiasi valore da $-\infty$ a $+\infty$.

In questo workshop vedremo come superare alcuni di questi problemi usando i modelli lineari generalizzati (*GLM*) [1].

[1] Alcune persone usano invece l'acronimo *GLiMs*.

Obiettivi formativi

Dopo aver completato questo workshop sarai in grado di:

- Descrivere il concetto di GLM e delle funzioni di collegamento
- Creare e interpretare l'output dei GLM per gestire i dati discreti e limitati.

Una nota su χ^2 e test di Fisher.

Il modo più semplice di trattare i dati di conteggio è quello di utilizzare i test χ^2 o Fisher. Puoi eseguire questi test in R, usando le funzioni *chisq.test* o *fisher.test*.

Introduzione ai modelli lineari generalizzati (GLM)

A questo punto, dovresti avere molta familiarità con l'equazione generica per un modello lineare:

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$

Come detto sopra, questa equazione non va bene per rappresentare i dati delimitati, diciamo una proporzione o una probabilità che va da 0 a 1.

In effetti, se dovessi modellare la proporzione di pazienti con una malattia in base a un determinato parametro X con un modello lineare potresti finire con qualcosa di simile:

$\%pazienti = 0.02 + 2.5X$

Ciò significa che se X è 50 il tuo modello dirà che il 125.02% dei pazienti ha la malattia, il che non è possibile. Allo stesso modo, se X può assumere un valore negativo, potresti ritrovarti con una percentuale negativa di pazienti che, di nuovo, non è possibile.

Pertanto, abbiamo bisogno di introdurre alcune "non-linearità" nell'equazione sopra, che ci consente, ad esempio, di limitare la nostra risposta a tra 0 e 1.

Il **Modello Lineare Generalizzato** risolve questo problema introducendo una "funzione di collegamento" f tale che $f(Y)$ sia una combinazione lineare dei predittori. Inoltre, questi modelli rilassano l'ipotesi che i residui siano normalmente distribuiti (vedi sotto).

$$f(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

Ad esempio, se f è un logaritmo, vincolerà l'output del modello a un numero positivo, imponendo quindi un limite inferiore di 0 alla nostra risposta [2].

Si noti che questo è ancora un modello lineare! Sebbene la relazione tra Y e $f(Y)$ e tra Y e i predittori X_i non sia lineare, la relazione tra $f(Y)$ e X_i lo è!

Esistono diverse funzioni di collegamento utilizzate in contesti diversi. Ne considereremo solo due in questo workshop (le funzioni di collegamento *logit* e *log*), ma il ragionamento è molto simile per qualsiasi funzione si possa finire usando [3].

[2] I modelli lineari utilizzati finora utilizzano una "funzione di collegamento identità", che è semplicemente definita come $f(x) = x$. Puoi vedere come sono un caso speciale della versione generalizzata che stiamo introducendo in questo workshop.

[3] Si noti che non possiamo usare alcuna funzione arbitraria, ma questo è oltre la portata di questo corso!

Regressione logistica

Il primo tipo di applicazione di GLM che useremo è la *regressione logistica* [4]. È un tipo di regressione usata per modellare risultati binari (0/1, sì/no), oltre a percentuali/proporzioni.

Ad esempio, potremmo voler modellare le probabilità di un evento che si verifica [5] in funzione di alcune variabili. Dal momento che vogliamo limitare la risposta a tra 0 e 1, modelliamo $log(odds)$.

Possiamo scrivere:

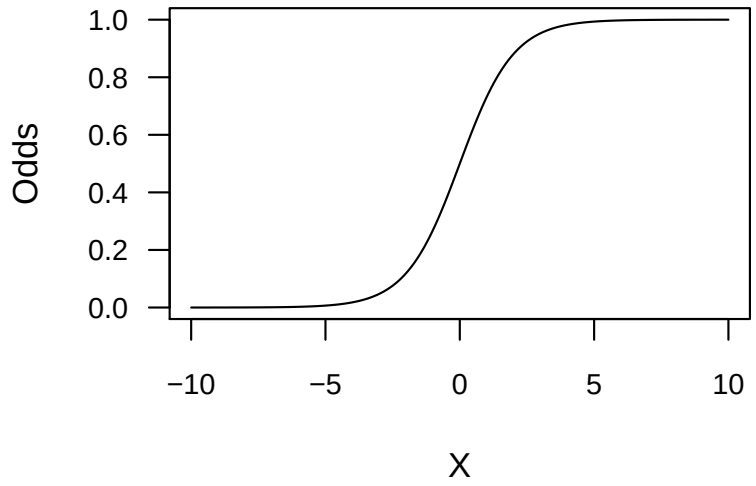
$$log(odds(Y)) = log\left(\frac{p(Y)}{1-p(Y)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

dove *p(Y)* è la probabilità che si verifichi Y.

Come spiegato sopra, questo è un GLM; la funzione di collegamento utilizzata qui (*log(odds)*) è generalmente chiamata funzione di collegamento *logit*. Possiamo anche riscrivere il modello in termini di probabilità di Y, usando la funzione di collegamento inversa [6].

$$\frac{p(Y)}{1-p(Y)} = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_n X_n}}{1+e^{\beta_0+\beta_1 X_1+...+\beta_n X_n}} + \epsilon$$

La funzione di collegamento logit è definita solo nell'intervallo (0, 1) e il suo inverso assomiglia a questo:



È quindi un'ottima scelta per modellare qualcosa che può esistere solo tra 0 e 1!

[4] A volte nominata come *regressione logit*

[5] Probabilità definita come $odds = \frac{p(X)}{1-p(X)}$

[6] In questo caso, poiché la funzione di collegamento è un logaritmo, il suo inverso è l'esponenziale.

Dati binari

Vediamo un esempio pratico. Iniziamo con un risultato binario [7], ovvero se i bambini sviluppano malattie respiratorie nel loro primo anno di vita, a seconda del loro sesso e alimentazione. In particolare, vengono considerati tre tipi di alimentazione: bottiglia (*Bottle*), seno (*Breast*), e supplementi (*Suppl*).

[7] Questi dati sono presi da Payne, 1987, e analizzati anche in Faraway, 2006

Inizia caricando il file *babyfood_workshop4.csv*

```
babyfood <- read.csv("babyfood_workshop4.csv")
babyfood
```

```
##   disease nondisease  sex   food
## 1      77        381  Boy Bottle
## 2      19        128  Boy    Mix
## 3      47        447  Boy Breast
## 4      48        336 Girl Bottle
## 5      16        111 Girl    Mix
## 6      31        433 Girl Breast
```

```
# Riordina il fattore cibo per avere Breast
# come gruppo di riferimento
babyfood$food <- factor(babyfood$food, levels = c("Breast",
    "Bottle", "Mix"))
```

Ora possiamo fittare il modello usando la funzione *glm*. Specifichiamo che i dati provengono da una distribuzione binomiale [8] e una funzione di collegamento *logit* [9].

[8] Una distribuzione binomiale è buona per rappresentare la probabilità di successo in alcune prove

[9] Si noti che *logit* è il valore predefinito, quindi si può anche omettere di specificarlo

```
model <- glm(cbind(disease, nondisease) ~ sex +
    food, family = binomial(link = logit), data = babyfood)
```

Dovresti essere abbastanza familiare con questa notazione. Passiamo entrambi gli eventi di malattia e non di malattia, usando *cbind* (binding di colonna) per "incollare" i valori insieme in una tabella con 2 colonne.

Vediamo l'output del nostro modello!

```
summary(model)
```

```
##
## Call:
## glm(formula = cbind(disease, nondisease) ~ sex + food, family = binomial(link = logit),
##     data = babyfood)
##
## Deviance Residuals:
##       1        2        3        4        5        6
##  0.1096  -0.5052   0.1922  -0.1342   0.5896  -0.2284
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.2820     0.1322 -17.259  < 2e-16 ***
## sexGirl      -0.3126     0.1410  -2.216   0.0267 *
## foodBottle    0.6693     0.1530   4.374 1.22e-05 ***
## foodMix       0.4968     0.2164   2.296   0.0217 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26.37529  on 5  degrees of freedom
## Residual deviance:  0.72192  on 2  degrees of freedom
## AIC: 40.24
##
## Number of Fisher Scoring iterations: 4
```

Vediamo che l'intercetta è diversa da 0, in quanto rappresenta le probabilità basali di malattia per il gruppo di controllo (maschi allattati al seno). Vediamo che ci sono anche effetti significativi sia di sesso che di tipo di alimentazione.

Ora, dovresti fare molta attenzione a interpretare questi coefficienti, perché ricorda che stiamo modellando *ln(odds)*, quindi dovremmo esponenziarli per ottenere le probabilità!

Quindi, per esempio, per le bambine, $\hat{\beta} = -0.3126$

```
exp(-0.3126)
```

```
## [1] 0.7315425
```

Ciò significa che essere una bambina porta le probabilità di avere malattie respiratorie al 73,2%, rispetto al livello di riferimento (bambini). Puoi calcolare gli intervalli di confidenza per le stime usando la funzione *confint* [10]. Ricordati di esponenziarli in modo che tu possa parlare di probabilità, piuttosto che *log(odds)*!

[10] In alternativa, puoi approssimare l'intervallo di confidenza del 95% usando $exp(\hat{\beta} \pm 1.96 * SE_{\hat{\beta}})$. Ad esempio per $\hat{\beta}_1$ abbiamo $exp(-0.3126 \pm 1.96 * 0.1410)$, che ci dà $[0.5549041, 0.9644088]$, molto simile a quanto calcolato da *confint*. Nota come questi intervalli non sono simmetrici, dal momento che stiamo lavorando su una scala non lineare.

```
exp(confint(model))
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %    97.5 %
## (Intercept) 0.07818602 0.1313591
## sexGirl     0.55362089 0.9629225
## foodBottle  1.45028833 2.6441703
## foodMix     1.06463534 2.4926583
```

Possiamo quindi dire che essere una bambina riduce le probabilità di avere malattie respiratorie al 73,2% (95% IC: [55,3, 96,3]) rispetto ai bambini. È possibile interpretare gli altri coefficienti in modo simile.

Infine, il sommario del modello riporta anche una misura di *devianza*. Questa è una misura di bontà del fit utile per i GLM; in generale più bassa è la devianza, meglio è.

Il riassunto riporta una devianza nulla di 26.38 su 5 gradi di libertà e una devianza residua di 0.72 su 2 gradi di libertà. La devianza nulla si riferisce al modello di sola intercetta (essenzialmente un modello nullo in cui diciamo che né il sesso né l'alimentazione hanno un effetto sulle probabilità della malattia). Dato che abbiamo 6 osservazioni, quel modello nullo ha 5 gradi di libertà. Il nostro modello attuale aggiunge 3 variabili (1 fittizia per sesso, 2 fittizie per l'alimentazione), quindi ha solo 2 gradi di libertà, ma ha una varianza molto ridotta, indicando che il nostro modello si adatta ai dati molto meglio di un modello di sola intercetta!

Come abbiamo visto in un workshop precedente, possiamo usare la funzione *drop1* per vedere il contributo di ciascun parametro del modello.

```
drop1(model)
```

```
## Single term deletions
##
## Model:
## cbind(disease, nondisease) ~ sex + food
##        Df Deviance    AIC
## <none>      0.7219 40.240
## sex     1   5.6990 43.217
## food    2  20.8992 56.417
```

Banalmente, vediamo che rimuovere Sex o Food dal modello si traduce in un aumento della devianza (e un aumento dell'`AIC`, un'altra misura di bontà di fit per la quale, ancora una volta, valori bassi sono migliori).

Ovviamente, guardando a questo tipo di dati dobbiamo sempre essere ben consapevoli che molti altri fattori di confusione (ad esempio lo stato socioeconomico) possono essere importanti da considerare.

Percentuali

Lo stesso ragionamento vale per i set di dati in cui abbiamo misurato una probabilità o una percentuale.

Ad esempio, cariciamo il file *smoking_workshop4.csv*. Questo file contiene dati sulla sopravvivenza [11] di 24321 medici britannici di sesso maschile nati tra il 1900 e il 1930, in relazione al fatto che siano fumatori o meno (questo include solo i fumatori che non abbiano mai smesso di fumare).

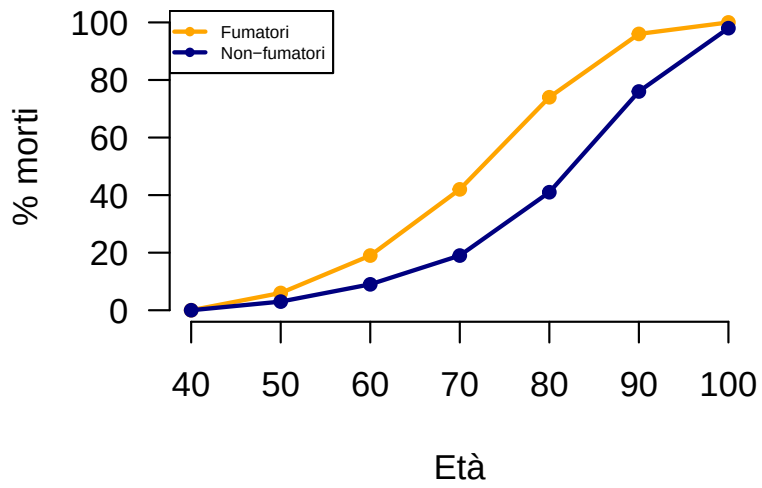
[11] Doll, 2004

```
smoking <- read.csv("smoking_workshop4.csv")
head(smoking)
```

```
##   Age Smoker Alive Dead
## 1  40      N   100    0
## 2  50      N    97    3
## 3  60      N    91    9
## 4  70      N    81   19
## 5  80      N    59   41
## 6  90      N    24   76
```

Proprio come prima, possiamo adattare un GLM.

Prova a tracciare i dati, ad esempio ho ottenuto questo grafico:



Cosa concluderesti guardando i dati?

Ora fittiamo un GLM a questi dati

```
smoking$AgeAdj <- smoking$Age - 40

model.2 <- glm(cbind(Dead, Alive) ~ AgeAdj + Smoker,
    family = binomial(link = "logit"), data = smoking)

summary(model.2)
```

```
##
## Call:
## glm(formula = cbind(Dead, Alive) ~ AgeAdj + Smoker, family = binomial(link = "logit"),
##     data = smoking)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5589  -0.7459   0.3528   1.2303   1.9902
##
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.709451   0.311152 -18.349  < 2e-16 ***
## AgeAdj       0.140632   0.007226  19.461  < 2e-16 ***
## SmokerY      1.305192   0.180313   7.238 4.54e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1024.724  on 13  degrees of freedom
## Residual deviance:   19.996  on 11  degrees of freedom
## AIC: 71.108
##
## Number of Fisher Scoring iterations: 4
```

Noterai che anziché `Age` ho modellato `Age - 40`; questo influenzerà solo l'intercetta, rendendola più facile da interpretare. Non influenzerà gli altri coefficienti [12].

L'intercetta è ~ -5.7. Questo rappresenta il `log(odds)` basale di morire per qualcuno al livello di riferimento (non fumatore) e ad `AgeAdj = 0`. Poiché `AgeAdj = Età - 40`, l'intercetta mostra il `log(odds)` basale per un non fumatore di 40 anni [13].

Quindi, le probabilità di morire per un non fumatore di 40 anni sono

```
exp(-5.709451)
```

```
## [1] 0.003314492
```

Ricorda, queste sono *odds*, quindi sono $\frac{P(morte)}{1-P(morte)}$; questo valore è molto basso, rappresentativo del fatto che tutti i soggetti erano vivi all'età di 40 anni.

Possiamo anche vedere un forte effetto dell'età sulla probabilità di morte [14], ed anche un significativo effetto del fumo. In particolare, il fumo aumenta le probabilità di morte di:

```
exp(1.305192)
```

```
## [1] 3.688397
```

Il coefficiente per età è interpretato come il rapporto di *log(odds)* per una differenza di età di 1 anno.

Cioè: $\frac{odds(morte, et\ x+1)}{odds(morte, et\ x)}$, dove le probabilità (*odds*) sono definite come sopra.

Infine, possiamo controllare graficamente che il nostro modello si adatta correttamente ai dati.

Possiamo chiedere al modello di prevedere i valori a diverse età per fumatori e non fumatori. Usiamo la funzione `predict` per questo.

[12] Prova da solo! Guarda cosa succede quando usi `Age`. Se questo è fonte di confusione, prova a farlo su un modello lineare semplice, sarà più intuitivo lì.

[13] Se modelliamo `Age` e non `AgeAdj`, l'intercetta si riferirebbe a 0 anni, che è probabilmente meno interessante.

[14] Immagino che non avessimo davvero bisogno di un modello per dirlo!

Questa funzione richiede una lista con elementi denominati come parametri del modello.

Ad esempio, se volessimo prevedere i *log(odds)* di morire per fumatori e non fumatori da 40 a 100 anni in step di 1 anno potremmo fare quanto segue:

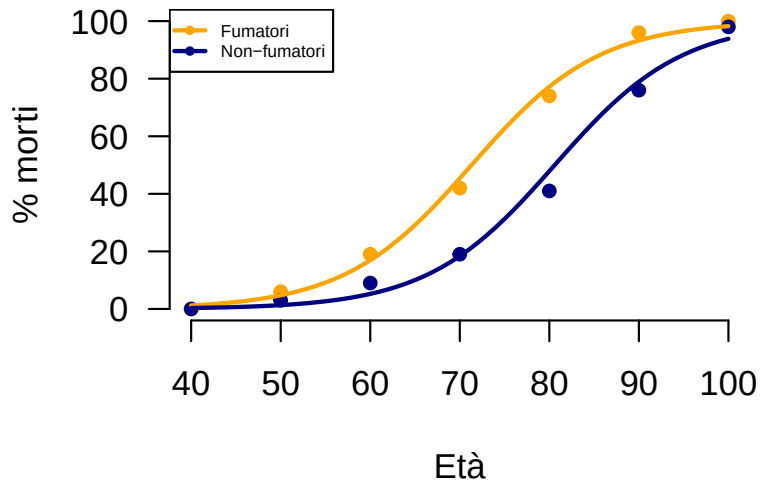
```
pred.age <- 40:100

smokers <- list(AgeAdj = pred.age - 40, Smoker = rep("Y",
    length(pred.age)))

nonsmokers <- list(AgeAdj = pred.age - 40, Smoker = rep("N",
    length(pred.age)))
```

Ora possiamo usare *predict* per chiedere al modello quale sarà il *log(odds)* per questi nuovi punti dati [15].

```
pr.smokers <- predict(model.2, type = "response",
    newdata = smokers) * 100
pr.nonsmoker <- predict(model.2, type = "response",
    newdata = nonsmokers) * 100
```

[15] Il parametro `type = "response"` ci fornisce previsioni in termini di probabilità, piuttosto che *log(odds)*. Se lo ometti dovrai esponenziare i risultati. In generale, ti permetterà di vedere la previsione del modello in termini di Y piuttosto che di $f(Y)$, dove f è la funzione di collegamento.

Ora possiamo tracciare la previsione in cima ai nostri dati, dimostrando che il modello funziona molto bene!



Quindi ... posso usare invece la regressione lineare?

Come spiegato sopra, questa è probabilmente una cattiva soluzione. Vediamo cosa succede se usiamo `lm`.

```
# Proporzione di soggetti deceduti
smoking$PercDead <- smoking$Dead/(smoking$Dead +
```

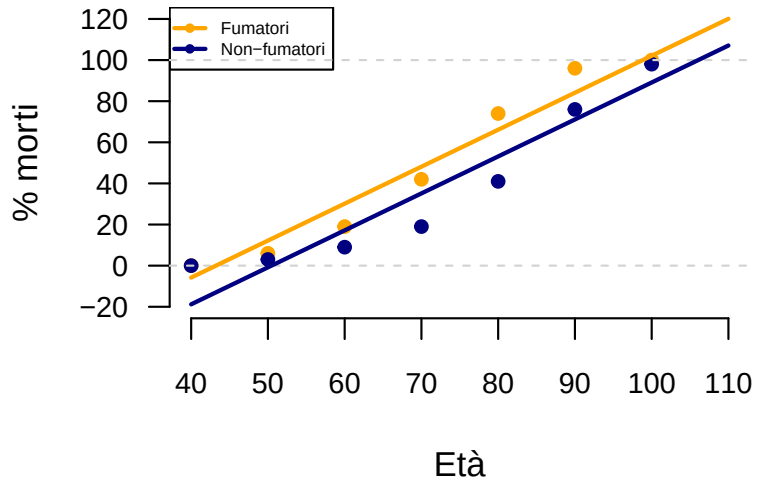
```
    smoking$Alive)
model.lm <- lm(PercDead ~ AgeAdj + Smoker, data = smoking)
```

Modellizziamo la percentuale di soggetti morti contro `Age - 40` e `Smoker`.

```
summary(model.lm)
```

```
##
## Call:
## lm(formula = PercDead ~ AgeAdj + Smoker, data = smoking)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.161429 -0.076652  0.008661  0.073571  0.188036
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.188036   0.061884  -3.038   0.0113 *
## AgeAdj       0.017982   0.001501  11.981 1.18e-07 ***
## SmokerY      0.130000   0.060037   2.165   0.0532 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1123 on 11 degrees of freedom
## Multiple R-squared:  0.9309, Adjusted R-squared:  0.9184
## F-statistic: 74.11 on 2 and 11 DF,  p-value: 4.136e-07
```

Possiamo già vedere che abbiamo un'intercetta negativa che significa che a 40 anni ... -18% dei pazienti sono morti! Un grafico delle previsioni del modello ci mostra che il modello non funziona bene, specialmente per i non fumatori.



Allo stesso modo, si consideri un fumatore di 120 anni. Qual è la probabilità che sia morto secondo i due modelli?

```
# Secondo la regressione logistica
predict(model.2, list(AgeAdj = 80, Smoker = "Y"),
    type = "response")
```

```
##         1
## 0.9989377
```

```
# Secondi la regressione lineare
predict(model.lm, list(AgeAdj = 80, Smoker = "Y"))
```

```
##        1
## 1.380536
```

Quindi, la regressione logistica ci dice che le probabilità di morte del paziente sono del 99,89%, mentre il modello lineare predice un valore del 138%!

In breve, la regressione lineare non è una buona scelta per modellare dati binari o percentuali.

Conteggi

Infine, vedremo un esempio di modello di conteggi. Questi sono spesso modellati usando quella che viene chiamata *regressione di Poisson*.

Questa regressione viene fatta con un GLM che modella i dati di Poisson e una funzione di collegamento logaritmica [16], ottenuto semplicemente specificando `family = poisson (link = log)` nella chiamata a *glm*.

Questo significa modellizzare:

$$log(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

Consideriamo i dati in *lizards-workshop4.csv*. Questo insieme di dati mostra i conteggi di tre specie di lucertole (A, B e C) in tre posizioni diverse (da Loc1 a Loc3). Per ogni posizione le lucertole erano contate in tre diversi lotti di terra.

[16] Proprio come sopra, scegliamo una distribuzione di Poisson perché è buona per modellare i conteggi, poiché è una distribuzione discreta, e la funzione di collegamento per limitare l'output a $Y > 0$. Si noti che la distribuzione di Poisson non è sempre la scelta migliore per i conteggi, altre opzioni sono disponibili. Nello specifico, potresti voler evitare la regressione di Poisson in caso di un numero elevato di zeri nei tuoi dati (le distribuzioni *zero-inflated* sono più adatte a questo) o in caso di sovraddispersione dei dati (la distribuzione binomiale negativa è più adatta a questo caso).

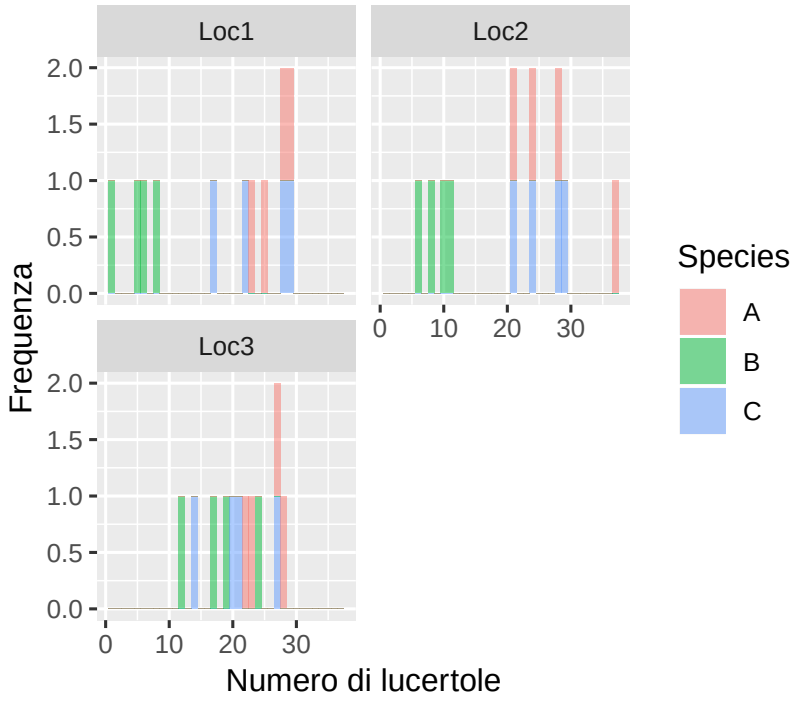
```
lizards <- read.csv("lizards-workshop4.csv")
summary(lizards)
```

```
##  Location  Plot   Species     Count
##  Loc1:12   P1:9   A:12    Min.   : 1.00
##  Loc2:12   P2:9   B:12    1st Qu.:13.50
##  Loc3:12   P3:9   C:12    Median :22.00
##            P4:9           Mean   :20.06
##                           3rd Qu.:27.25
##                           Max.   :37.00
```

```
head(lizards)
```

```
##   Location Plot Species Count
## 1     Loc1   P1       A    28
## 2     Loc1   P1       B     1
## 3     Loc1   P1       C    28
## 4     Loc1   P2       A    29
## 5     Loc1   P2       B     5
## 6     Loc1   P2       C    22
```

Possiamo iniziare tracciando i dati:



Sembra che nelle posizioni 1 e 2 le specie A e C siano in numeri simili, superiori alla specie B. Tuttavia, nella posizione 3, tutte e tre le specie sembrano avere una frequenza simile.

Questa non è una situazione ovvia da analizzare, vediamo come usare un GLM per modellarla! Per semplicità, considereremo i lotti come indipendenti, anche se dovresti aver notato che si tratta di un design annidato, quindi l'effetto casuale del lotto dovrebbe, in teoria, essere tenuto in considerazione! Se vuoi, puoi creare un GLM ad effetti misti [17], ma non lo spiegherò qui, quindi lo lascio alla tua curiosità!

[17] Ad esempio, usando la funzione *glmm* nel pacchetto *glmm* o la funzione *glmer* nel pacchetto *lme4*

Iniziamo creando il GLM. Poiché abbiamo notato una chiara interazione Specie/Posizione, la aggiungiamo al nostro modello. Possiamo iniziare tracciando i dati:

```
model.3 <- glm(Count ~ Species * Location, data = lizards,
    family = poisson(link = log))
```

```
summary(model.3)
```

```
##
## Call:
## glm(formula = Count ~ Species * Location, family = poisson(link = log),
##     data = lizards)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.18658  -0.62152   0.04753   0.54296   1.71993
```

```
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            3.26767    0.09759  33.484  < 2e-16 ***
## SpeciesB              -1.65823    0.24397  -6.797 1.07e-11 ***
## SpeciesC              -0.08961    0.14121  -0.635   0.5257
## LocationLoc2           0.04652    0.13644   0.341   0.7331
## LocationLoc3          -0.04879    0.13973  -0.349   0.7270
## SpeciesB:LocationLoc2  0.51310    0.31174   1.646   0.0998 .
## SpeciesC:LocationLoc2  0.01410    0.19707   0.072   0.9429
## SpeciesB:LocationLoc3  1.32972    0.28881   4.604 4.14e-06 ***
## SpeciesC:LocationLoc3 -0.10884    0.20527  -0.530   0.5960
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 157.26  on 35  degrees of freedom
## Residual deviance:  29.35  on 27  degrees of freedom
## AIC: 216.13
##
## Number of Fisher Scoring iterations: 4
```

Questo è un output piuttosto complesso. Decifriamolo! Prima di tutto, ricorda cosa stiamo modellizzando:

$log(Conteggi) = \beta_0 + \beta_1 * SpecieB + \beta_2 * SpecieC + \beta_3 * Posizione2 + \beta_4 * Posizione3 +$ (interazioni, con coefficienti da β_5aβ_8)

Dove *SpecieB* e *SpecieC* sono le due variabili fittizie utilizzate per rappresentare il fattore a tre livelli *Specie* e *Posizione2* e *Posizione3* sono le due variabili fittizie utilizzate per rappresentare le posizioni.

Quindi $\hat{\beta}_0$ è il $log(conteggimedi)$ per il livello basale (Specie A nella Posizione 1).

Infatti, se controlliamo la media manualmente con:

```
mean(lizards$Count[lizards$Location == "Loc1" &
    lizards$Species == "A"])
```

```
## [1] 26.25
```

Possiamo vedere che il modello si avvicina abbastanza bene!

```
exp(3.26767)  # exp(beta1)
```

```
## [1] 26.25011
```

È possibile interpretare gli altri coefficienti in modo simile. Ad esempio $\beta_{SpecieC} = -0.08961$ ci dice che l'effetto della specie C è di

diminuire i conteggi al $e^{-0.08961} \approx 0.91 \approx 91\%$ del livello di riferimento. Di nuovo, possiamo calcolare gli intervalli di confidenza del 95% usando *confint*.

```
exp(confint(model.3))
```

```
## Waiting for profiling to be done...
```

```
##                               2.5 %     97.5 %
## (Intercept)            21.5438454 31.5958139
## SpeciesB                0.1147362  0.3001501
## SpeciesC                0.6925339  1.2056592
## LocationLoc2            0.8017074  1.3697220
## LocationLoc3            0.7236934  1.2525673
## SpeciesB:LocationLoc2   0.9144204  3.1205880
## SpeciesC:LocationLoc2   0.6890984  1.4929051
## SpeciesB:LocationLoc3   2.1800599  6.7938404
## SpeciesC:LocationLoc3   0.5993073  1.3409254
```

Quindi per la specie C i conteggi sono 91% (I.C. = (69.2%, 120%)) del livello di riferimento.

Possiamo anche vedere che c'è una significativa interazione tra specie B e posizione 3. Questo non è inaspettato. Interpretare i coefficienti di interazione è sempre difficile, ma per fortuna possiamo usare il nostro amico fidato *emmeans*!

```
library(emmeans)
marginals <- emmeans(model.3, ~Species * Location)
pairs(marginals, by = "Species", type = "response")
```

```
## Species = A:
##  contrast        ratio         SE  df z.ratio p.value
##  Loc1 / Loc2 0.9545455 0.13023414 Inf  -0.341  0.9379
##  Loc1 / Loc3 1.0500000 0.14671401 Inf   0.349  0.9350
##  Loc2 / Loc3 1.1000000 0.15198684 Inf   0.690  0.7695
##
## Species = B:
##  contrast        ratio         SE  df z.ratio p.value
##  Loc1 / Loc2 0.5714286 0.16017155 Inf  -1.996  0.1130
##  Loc1 / Loc3 0.2777778 0.07021004 Inf  -5.068  <.0001
##  Loc2 / Loc3 0.4861111 0.10016757 Inf  -3.501  0.0013
##
## Species = C:
##  contrast        ratio         SE  df z.ratio p.value
##  Loc1 / Loc2 0.9411765 0.13383446 Inf  -0.426  0.9047
##  Loc1 / Loc3 1.1707317 0.17604546 Inf   1.048  0.5464
##  Loc2 / Loc3 1.2439024 0.18449653 Inf   1.471  0.3047
##
## P value adjustment: tukey method for comparing a family of 3 estimates
## Tests are performed on the log scale
```

Come previsto, l'unico rapporto di coppia statisticamente significativo è tra le posizioni 1 e 3 per la specie B. La stima è 0.28, il che significa che i conteggi per la specie B in posizione 3 sono maggiori di circa $1/0.28 \approx 3.6$ volte i conteggi della specie B in posizione 1 (o, alternativamente, i conteggi nella posizione 1 sono circa il 28% di quelli nella posizione 3) [18].

Questo workshop avrebbe dovuto darti gli strumenti di base per analizzare dati binari, proporzionali e di conteggio. Come sempre, stiamo solo grattando la superficie qui, ma questo dovrebbe essere un buon inizio, e se sei interessato a questi argomenti c'è molto da trovare! Questo è il tipo di modello lineare più avanzato che vedremo quest'anno. Il prossimo semestre esamineremo i modelli di classificazione e previsione e alcune tecniche statistiche più avanzate.

[18] Un risultato simile può essere ottenuto direttamente sommando l'esponenziato $\hat{\beta}$