Métodos Computacionais para Inferência Estatística

Capítulo 2 - Modelos de Regressão

Wagner Hugo Bonat
Paulo Justiniano Ribeiro Jr
Elias Teixeira Krainski
Walmes Marques Zeviani

LEG: Laboratório de Estatística e Geoinformação
Universidade Federal do Paraná

30 de julho de 2012

Objetivo e estrutura

1. Modelos de regressão;
2. Regressão de Poisson;
3. Regressão Contagem-Gama (contagem com subdispersão);
4. Regressão não linear (reparametrizações do modelo logístico);

Laboratório de Estatística e Geoinformação

Objetivo e estrutura

- Procura-se explicar a variação em uma variável aleatória dado conhecimento sobre covariáveis;

Y: variável aleatória

$f()$: distribuição de probabilidades

$$Y|x \sim f(\mu, \phi)$$

ϕ: parâmetros subjacentes

μ: parâmetros de interesse

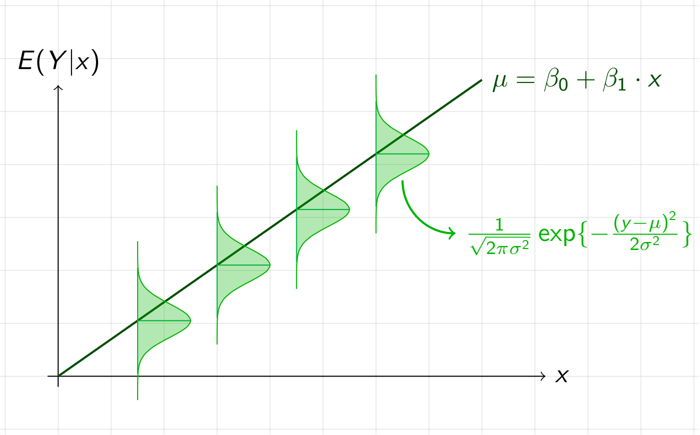
x: covariáveis conhecidas

$g()$: função de relação

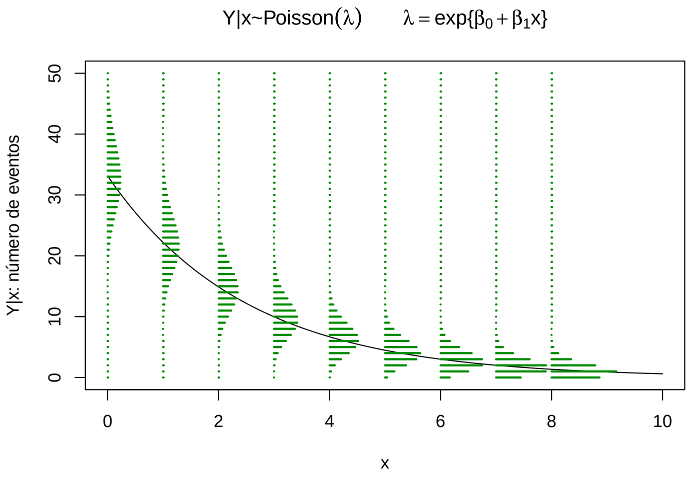
β: parâmetros

$$\mu = g(x, \beta)$$

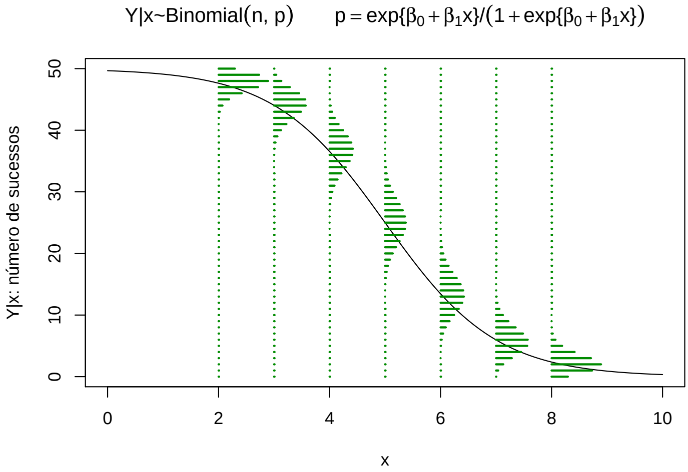
Modelo de regressão linear simples



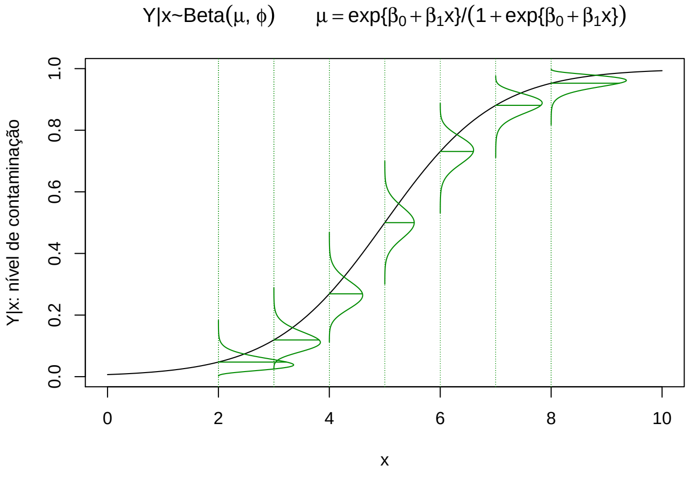
Regressão de Poisson (GLM Poisson)



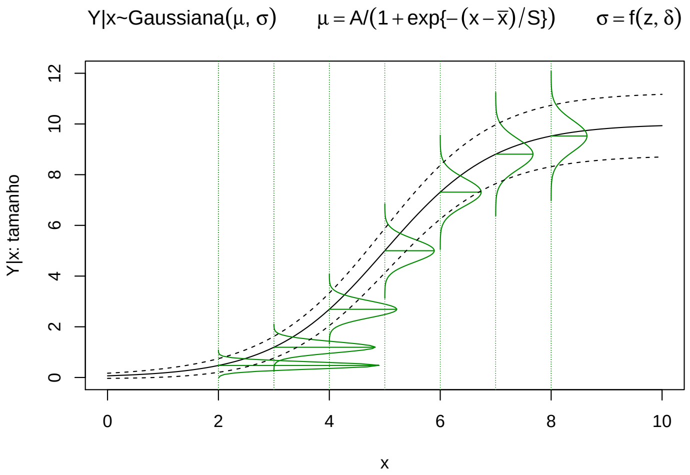
Regressão logística (GLM Binomial)



Regressão Beta



Regressão não linear com heterogeneidade de variância



Regressão de Poisson: verossimilhança

- Função de probabilidade

$$Pr(Y = y) = \frac{\exp\{-\lambda\}\lambda^y}{y!}, \qquad \lambda > 0, \quad y = 0, 1, 2, \ldots;$$

- Momentos

$$E(Y) = \lambda \quad \text{e} \quad V(Y) = \lambda;$$

- Função de relação

$$\lambda = g(x, \beta) : \mathbb{R} \rightarrow \mathbb{R}_*^+,$$

- Usualmente

$$\lambda = \exp\{X\beta\}.$$

Laboratório de Estatística e Geoinformação

Regressão de Poisson: log-verossimilhança

- Função de verossimilhança

$$L(\lambda; y) = \prod_{i=1}^{n} \frac{\exp\{-\lambda\}\lambda_i^{y}}{y_i!};$$

- Função de log-verossimilhança $\ell(.) = \log L(.)$, $\lambda_i = \exp\{x_i^\top \beta\}$,

$$\ell(\beta; y, X) = \sum_{i=1}^{n} -\exp\{x_i^\top \beta\} + y_i x_i^\top \beta - \log(y_i!);$$

- Em forma vetorial

$$\ell(\beta; y, X) = -1^\top \exp\{X\beta\} + y^\top X\beta - 1^\top \log(y!);$$

Laboratório de Estatística e Geoinformação

Regressão de Poisson: função escore e hessiana

- $\ell(.)$ vetorial

$$\ell(\beta; y, X) = -1^\top \exp\{X\beta\} + y^\top X\beta - 1^\top \log(y!);$$

- Função escore $U(\beta) = \frac{\partial \ell}{\partial \beta}$,

$$U(\beta) = (y - \exp\{X\beta\})^\top X;$$

- Matriz de informação observada $I_o(\beta) = \frac{\partial U}{\partial \beta}$,

$$I_o(\beta) = X^\top (diag(\exp\{X\beta\}))X;$$

Regressão de Poisson: simulação

Simulando dados

$$Y|x \sim \text{Poisson}(\lambda)$$

$$\lambda = \exp\{\beta_0 + \beta_1 x\}$$

```
> simula.poisson <- function(formula, beta){
+   X <- model.matrix(formula)
+   lambda <- exp(X%*%beta)
+   y <- rpois(nrow(X), lambda=lambda)
+   return(data.frame(y=y, X=X))
+ }
```

```
> set.seed(123)
> cov <- seq(0, 5, length=10)
> dados10 <- simula.poisson(~cov, beta=c(2, 0.5))
> str(dados10)

'data.frame':        10 obs. of  3 variables:
 $ y            : num  6 12 12 23 22 30 49 54 57 73
 $ X..Intercept.: num  1 1 1 1 1 1 1 1 1 1
 $ X.cov        : num  0 0.556 1.111 1.667 2.222 ...
```

Regressão de Poisson: cálculo do escore e hessiana

Vetor escore e matriz hessiana

```
> escore <- function(par, formula, dados){
+   X <- model.matrix(as.formula(formula), data=dados)
+   esco <- t(dados$y-exp(X%*%c(par[1], par[2])))%*%X
+   return(as.vector(esco))
+ }
```

$U(\beta) = (y - \exp\{X\beta\})^\top X$

```
> hessiano <- function(par, formula, dados){
+   X <- model.matrix(as.formula(formula), data=dados)
+   mat <- diag(length(dados$y))
+   diag(mat) <- -exp(X%*%c(par[1], par[2]))
+   H <- t(X)%*%mat%*%X
+   return(H)
+ }
```

$I_o(\beta) = -X^\top (diag(\exp\{X\beta\}))X$

```
> hessiano <- function(par, formula, dados){
+   X <- model.matrix(as.formula(formula), data=dados)
+   H <- crossprod(X*-(exp(drop(X%*%par))), X)
+   return(H)
+ }
```

+eficiente

Regressão de Poisson: estimação

- estimação pelo Newton-Raphson

```
> estimativa <- NewtonRaphson(initial=c(0,0), escore=escore, hessiano=hessiano,
                              max.iter=100, n.dim=2, formula="~cov", dados=dados10)
> estimativa

[1] 2.2285674 0.4276769
```

- informação

```
> Io <- -hessiano(par=estimativa, formula="~cov", dados=dados10)
> Io

            (Intercept)      cov
(Intercept)     338.000 1182.222
cov            1182.222 4796.149
```

- intervalos de confiança

```
> desvio.padrao <- sqrt(diag(solve(Io)))
> estimativa[1]+c(-1,1)*qnorm(0.975)*desvio.padrao[1]
> estimativa[2]+c(-1,1)*qnorm(0.975)*desvio.padrao[2]

[1] 1.941420 2.515715
[1] 0.3514484 0.5039054
```

Regressão de Poisson: estimação pela `glm()`

- estimativas

```
> reg.glm <- glm(y~cov, data=dados10, family=poisson)
> summary(reg.glm)$coeff

             Estimate Std. Error  z value     Pr(>|z|)
(Intercept) 2.2285674 0.14650663 15.21138 2.972301e-52
cov         0.4276769 0.03889281 10.99630 3.981440e-28
```

- IC assintótico

```
> confint.default(reg.glm)

                2.5 %    97.5 %
(Intercept) 1.9414197 2.5157151
cov         0.3514484 0.5039054
```

- IC perfilhado

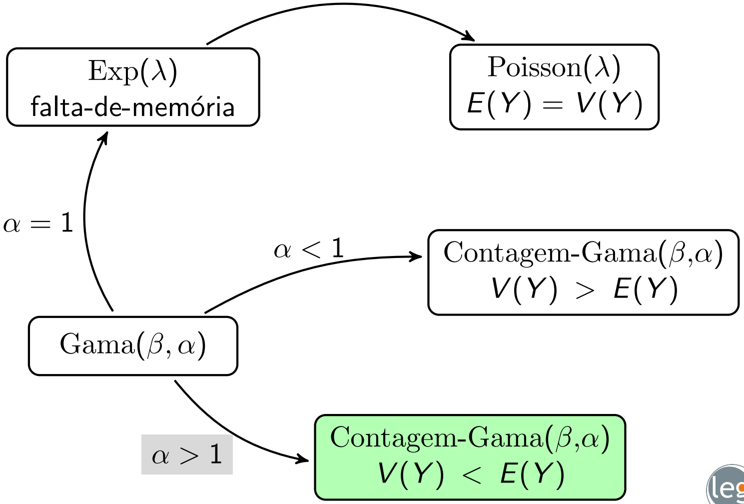
```
> confint(reg.glm)

Waiting for profiling to be done...
                2.5 %    97.5 %
(Intercept) 1.9327201 2.5074004
cov         0.3525055 0.5050645
```

Laboratório de Estatística e Geoinformação

Regressão contagem-Gama: motivação

- Poisson implica relação média-variância: $E(Y) = V(Y)$;
- Superdispersão ($V(Y) > E(Y)$): Binomial negativa, Quasi-Poisson, GLMM, ...;
- Subdispersão ($V(Y) < E(Y)$): carente de modelos;
- Experimentos agronômicos: frutos, sementes, brotos, raízes, nós, filhotes, ovos, pústulas, lesões, nódulos, ...;

Regressão contagem-Gama: introdução



Regressão contagem-Gama: introdução

$$\boxed{\mathrm{Gama}(y;\beta,\alpha) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\exp\{-\beta y\}\cdot y^{\alpha-1}}$$

$$\Big\downarrow \; \alpha = 1$$

$$\boxed{\mathrm{Exp}(\beta) = \beta\exp\{-\beta y\}}$$

Laboratório de Estatística e Geoinformação

Regressão contagem-Gama: desenvolvimento

- intervalos entre tempo $\tau \sim$ Gama(α,β),

$$f(\tau, \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \cdot \tau^{\alpha-1} \cdot \exp\{-\beta\tau\}$$

- tempo até o n-ésimo evento $\vartheta_n = \tau_1 + \cdots + \tau_n \ \sim$ Gama$(n\alpha, \beta)$,

$$f_n(\vartheta, \alpha, \beta) = \frac{\beta^{n\alpha}}{\Gamma(n\alpha)} \cdot \vartheta^{n\alpha-1} \cdot \exp\{-\beta\vartheta\}$$

- $N_T < n$ se e somente se $\vartheta_n \geq T$,

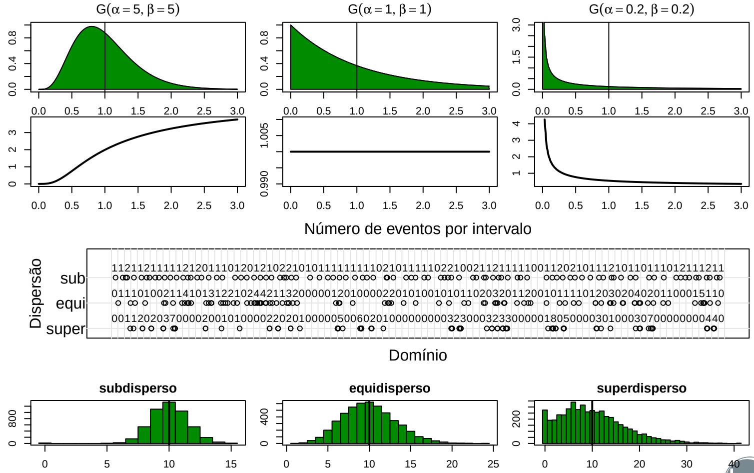
$$P(N_T < n) = P(\vartheta_n \geq T) = 1 - F_n(T);$$

- $P(N_T = n) = P(N_T < n+1) - P(N_T < n)$ então

$$P(N_T = n) = F_n(T) - F_{n+1}(T).$$

Laboratório de Estatística e Geoinformação

Regressão contagem-Gama: simulação



Laboratório de Estatística e Geoinformação

Regressão contagem-Gama: verossimilhança

- $F_n(T) = G(T, n\alpha, \beta) = \frac{1}{\Gamma(n\alpha)} \int_0^T u^{n\alpha - 1} \cdot \exp\{-\beta u\}\, du$
- $P(N_T = n) = G(T, n\alpha, \beta) - G(T, (n+1)\alpha, \beta)$
- $E(\tau|x) = \frac{\alpha}{\beta} = \exp\{-x^\top \gamma\}$
- $\ell(y; x, \alpha, \gamma, T) =$

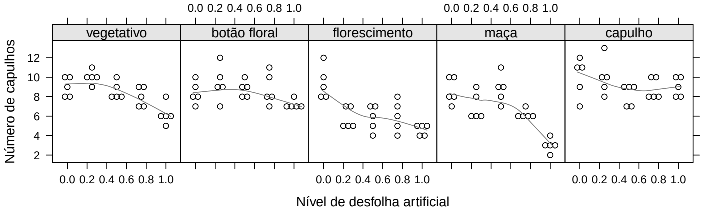
$$\sum_{i=1}^{n} \ln\left(G(T, y_i\alpha, \alpha\exp\{x_i^\top\gamma\}) - G(T, (y_i+1)\alpha, \alpha\exp\{x_i^\top\gamma\})\right)$$

```
ll <- function(theta, y, X, T=1){
  eXb <- exp(crossprod(X, theta[-1]))
  sum(log(pgamma(T, theta[1]*y, theta[1]*eXb)-
          pgamma(T, (theta[1]+1)*y, theta[1]*eXb)))
}
```

Laboratório de Estatística e Geoinformação

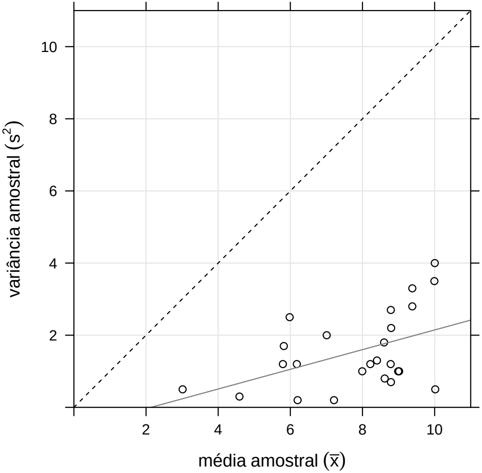
Regressão contagem-Gama: estudo de caso

- Número de capulhos do algodão $\sim$ nível de desfolha $+$ estágio fenológico;
- Experimento em vasos, fatorial 5×5 com 5 repetições;



Laboratório de Estatística e Geoinformação

Regressão contagem-Gama: análise exploratória



Regressão contagem-Gama: estimação

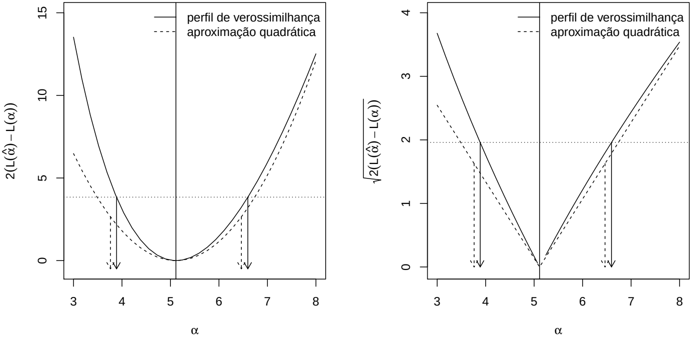
```
> X <- model.matrix(~est:(des+I(des^2)), data=cap)
> rp <- glm(nc~est:(des+I(des^2)), data=cap, family=poisson)
> cbind(estimativas=coef(rp)) # estimativas

                        estimativas
(Intercept)             2.189560352
estvegetativo:des       0.436859418
estbotao:des            0.289715410
...
estcapulho:I(des^2)    -0.019970497

> gam <- coef(rp)
> chutes <- c(alpha=1, gam)
> # estimação por máxima verossimilhança
> op <- optim(chutes, ll, y=cap$nc, X=X, hessian=TRUE,
+             method="BFGS", control=list(fnscale=-1))
> cbind(estimativas=op$par)        # estimativas

                        estimativas
alpha                   5.112297805
(Intercept)             2.234239342
estvegetativo:des       0.412024360
estbotao:des            0.274377741
...
estcapulho:I(des^2)    -0.018586566

> # 2*diferença da log-verossimilhança
> dll <- c(diff.ll=2*abs(op$value-c(logLik(rp)))); dll

 diff.ll
94.83326
```

Regressão contagem-Gama: perfil para α

```
ll.alpha <- function(theta, alpha, y, X){
  eXb <- exp(X%*%theta) #*theta[1]
  sum(log(pgamma(1, alpha*y, alpha*eXb)-
          pgamma(1, alpha*y+alpha, alpha*eXb)))
}
```

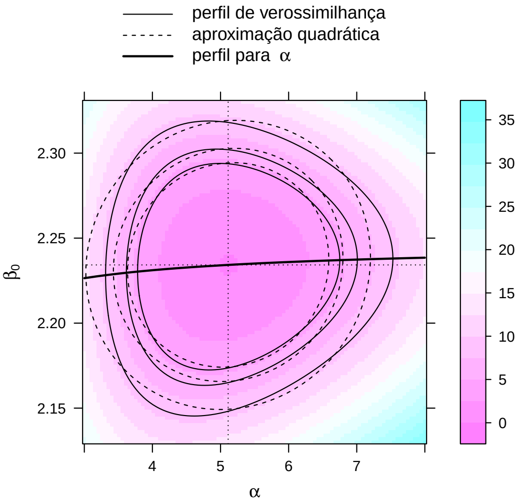
```
> alpha <- sort(c(seq(3,8,l=30), op$par[1])) # grid de valores para alpha
> perfil <- sapply(alpha,
+                  function(a){
+                    op <- optim(coef(rp), ll.alpha, alpha=a, y=cap$nc, X=X,
+                                method="BFGS", control=list(fnscale=-1))
+                    c(op$value, op$par[1])
+                  })
> coef <- op$par; vcov <- -solve(op$hessian); llik <- op$value
> alp <- coef["alpha"]; sd.alp <- sqrt(vcov["alpha","alpha"])
> dev.perf <- 2*(llik-perfil[1,])  # deviance da log-ver perfilhada
> dev.quad <- (alp-alpha)^2/sd.alp # deviance da apro quadrática
> require(rootSolve)
> qchi <- qchisq(0.95, df=1)
> fperf <- approxfun(alpha, dev.perf-qchi)
> lim <- uniroot.all(fperf, c(0, 10)) # limites do IC perf
> lim2 <- alp+c(-1,1)*1.96*sd.alp     # limites do IC assint
```

Regressão contagem-Gama: perfil para α

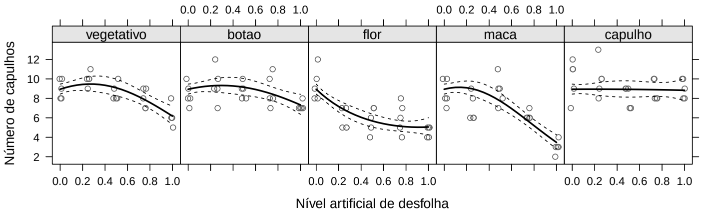


Laboratório de Estatística e Geoinformação

Regressão contagem-Gama: perfil para α e β_0



Regressão contagem-Gama: resultado final




```
> tabcoef <- data.frame(Estimativas=coef, ErroPadrão=sqrt(diag(vcov)))
> tabcoef$zvalor <- with(tabcoef, Estimativas/ErroPadrão)
> tabcoef$pvalor <- with(tabcoef, pnorm(abs(zvalor), lower=FALSE)*2)
> tabcoef

                        Estimativas ErroPadrão      zvalor       pvalor
alpha                   5.112297805 0.68872753  7.42281614 1.146559e-13
(Intercept)             2.234239342 0.02802741 79.71622031 0.000000e+00
estvegetativo:des       0.412024360 0.22796029  1.80743922 7.069382e-02
estbotao:des            0.274377741 0.22448099  1.22227609 2.216032e-01
estflor:des            -1.182180751 0.26654192 -4.43525263 9.196438e-06
estmaca:des             0.319589495 0.24988237  1.27895977 2.009112e-01
estcapulho:des          0.007104167 0.22267231  0.03190413 9.745485e-01
estvegetativo:I(des^2) -0.762638914 0.25818749 -2.95381824 3.138688e-03
estbotao:I(des^2)      -0.464149443 0.25044536 -1.85329623 6.383991e-02
estflor:I(des^2)        0.645341332 0.30030943  2.14892127 3.164064e-02
estmaca:I(des^2)       -1.198887094 0.29689851 -4.03803680 5.390040e-05
estcapulho:I(des^2)    -0.018586566 0.24424267 -0.07609877 9.393405e-01
```

Laboratório de Estatística e Geoinformação

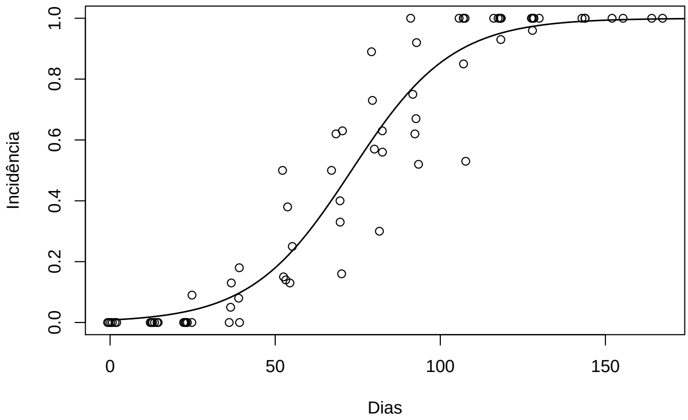
Modelo logístico: parametrizações

$$\text{logis}(x, \theta, \beta) = \frac{\theta}{1 + f(\exp\{x\}, \beta)} \tag{1}$$

- $f_a(x) = \exp\{(a_1 - x)/a_2\}$;
- $f_b(x) = b_1 \exp\{b_2 x\}$;
- $f_c(x) = \exp\{c_1 + c_2 x\}$;
- $f_d(x) = (-1 + 1/d_1) \exp\{-d_2 x\}$.

$$\begin{array}{lll} b_1 = \exp\{a_1/a_2\} & c_1 = a_1/a_2 & d_1 = 1/(1 + \exp\{a_1/a_2\}) \\ b_2 = -1/a_2 & c_2 = -1/a_2 & d_2 = 1/a_2. \end{array}$$

Modelo logístico: incidência de doença em plantas



Modelo logístico: estimação

```
> ll <- function(th, y, x, model){
+   ex <- do.call(model, list(x=x, th=th))
+   sd <- sqrt(crossprod(y-ex)/length(x))
+   ll <- sum(dnorm(y, mean=ex, sd=sd, log=TRUE))
+   ll
+ }
```

```
> # parametrizações
> f.a <- function(x, th){ 1/(1+exp((th[1]-x)/th[2])) }
> f.b <- function(x, th){ 1/(1+th[1]*exp(th[2]*x)) }
> f.c <- function(x, th){ 1/(1+exp(th[1]+th[2]*x)) }
> f.d <- function(x, th){ 1/(1+(-1+1/th[1])*exp(-th[2]*x)) }
> # dados
> y <- dados$inc2; x <- dados$dia
> # lista com valores iniciais e modelo
> init.list <- list(A=list(par=c(80,13), model=f.a), B=list(par=c(120,-0.06), model=f.b),
+                   C=list(par=c(5,-0.06), model=f.c), D=list(par=c(0.008, 0.065), model=f.d))
> fixed.list <- list(fn=ll, x=x, y=y, method="BFGS", control=list(fnscale=-1))
> # otimização em série dos modelos
> op.all <-
+   lapply(init.list,
+          function(i){
+            op <- do.call(optim, c(i, fixed.list)); op
+          })
> # estimativas dos parâmetros e log-verossimilhança
> pars <- sapply(op.all, "[[", "par"); pars
> ll0 <- sapply(op.all, "[[", "value"); ll0
```

```
            A            B           C           D
[1,] 73.12710 120.01001195  4.81592735 0.008455968
[2,] 15.23597  -0.06548208 -0.06584788 0.065166770
       A        B        C        D
57.13773 57.13735 57.13700 57.13430
```

Modelo logístico: contornos de confiança