

Trabalho Regressão Não-Linear

Caio Gomes Alves e Daniel Krügel

25/11/2023

Peso de Capulho de Algodão

O conjunto escolhido é proveniente de um experimento que tem como objetivo modelar a quantidade de capulho produzidos por plantas de algodão em diferentes estágios de desenvolvimento (**Vegetativo**, **Botão Floral**, **Florescimento**, **Maçã** e **Capulho**) utilizando diferentes covariáveis (**estag**=estágio da planta, **desf**=nível de desfolha artificial, **nnos**=número de nós no caule, dentre outras). Os dados estão disponíveis no repositório do professor Walmes Marques Zeviani, pelo link <http://www.leg.ufpr.br/~walmes/data/>.

```
# Dados de capulho de algodão:
cap <- read.table("algodao.txt",
                  header = TRUE,
                  sep = "\t",
                  encoding = "latin1")

# Ajustando os valores de desfolhamento:
cap$desf <- cap$desf/100

# Selecionando as variáveis de interesse:
cap <- subset(cap, select = c(estag, desf, pcapu))

# Ajustando os estágios de crescimento:
cap$estag <- factor(cap$estag, labels = c("Vegetativo",
                                          "Botão Floral",
                                          "Florescimento",
                                          "Maçã",
                                          "Capulho"))

# Primeiras linhas da base:
head(cap)

##           estag desf  pcapu
## 1 Vegetativo 0.00 33.160
## 2 Vegetativo 0.00 28.675
## 3 Vegetativo 0.00 31.485
## 4 Vegetativo 0.00 28.925
## 5 Vegetativo 0.00 36.430
## 6 Vegetativo 0.25 34.105

# Estrutura dos dados:
str(cap)
```

```
## 'data.frame': 125 obs. of 3 variables:
## $ estag: Factor w/ 5 levels "Vegetativo","Botão Floral",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ desf : num 0 0 0 0 0 0.25 0.25 0.25 0.25 0.25 ...
## $ pcapu: num 33.2 28.7 31.5 28.9 36.4 ...
```

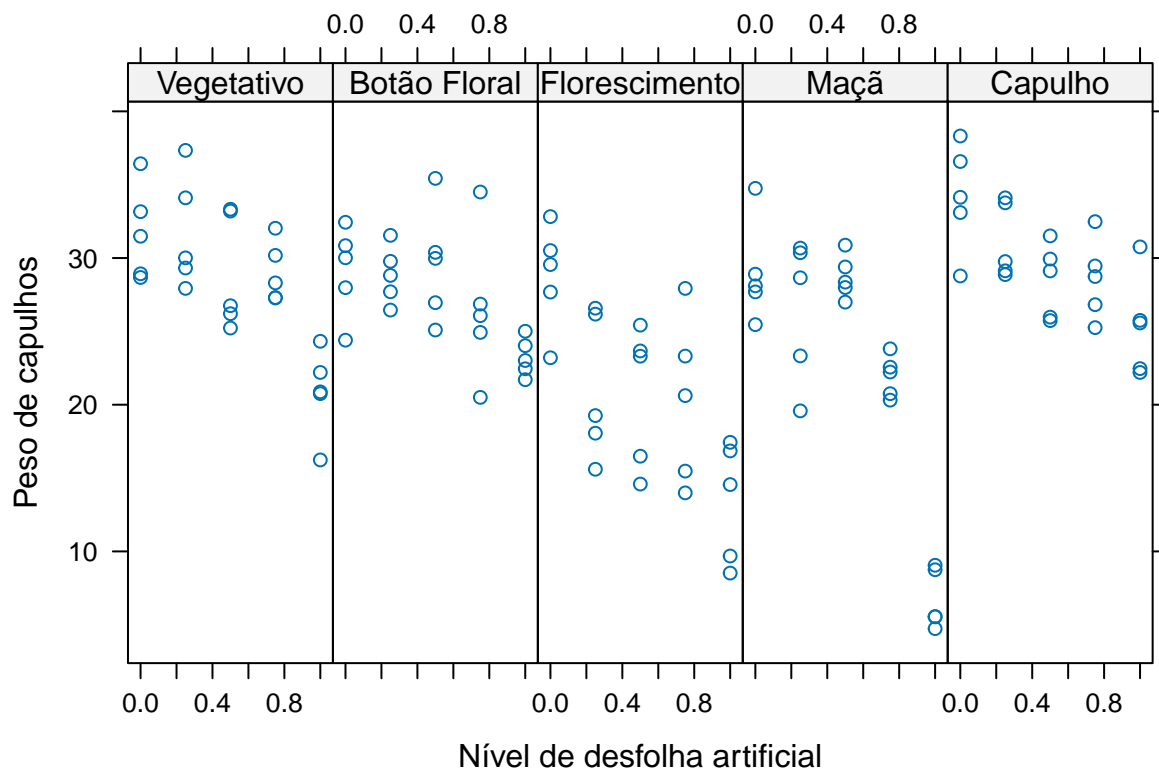
O modelo a ser ajustado é o modelo linear com potência, denotado por:

$$\text{pcapu} = f_0 - \delta(\text{desf}^{\exp(C)}) \quad (1)$$

Nesse modelo, f_0 é interpretado como o intercepto ($f(0)$), δ é dado como a diferença da função no início com o final do experimento ($f(1) - f(0)$) e C é o valor da forma como a curva decresce (caso $C = 0$, o modelo é linear, da forma $f_0 - \delta \text{desf}$). **pcapu** indica o peso dos capulhos e **desf** é o nível artificial de desfolha induzido no experimento.

Serão escolhidos dois estágios de crescimento para comparação, o **Vegetativo** e **Botão Floral** (os dois iniciais) que aparentam ser os mais comparáveis.

```
# Comparação dos estágios:
xyplot(pcapu ~ desf | estag,
       data = cap,
       layout = c(5, 1),
       xlab = "Nível de desfolha artificial",
       ylab = "Peso de capulhos")
```



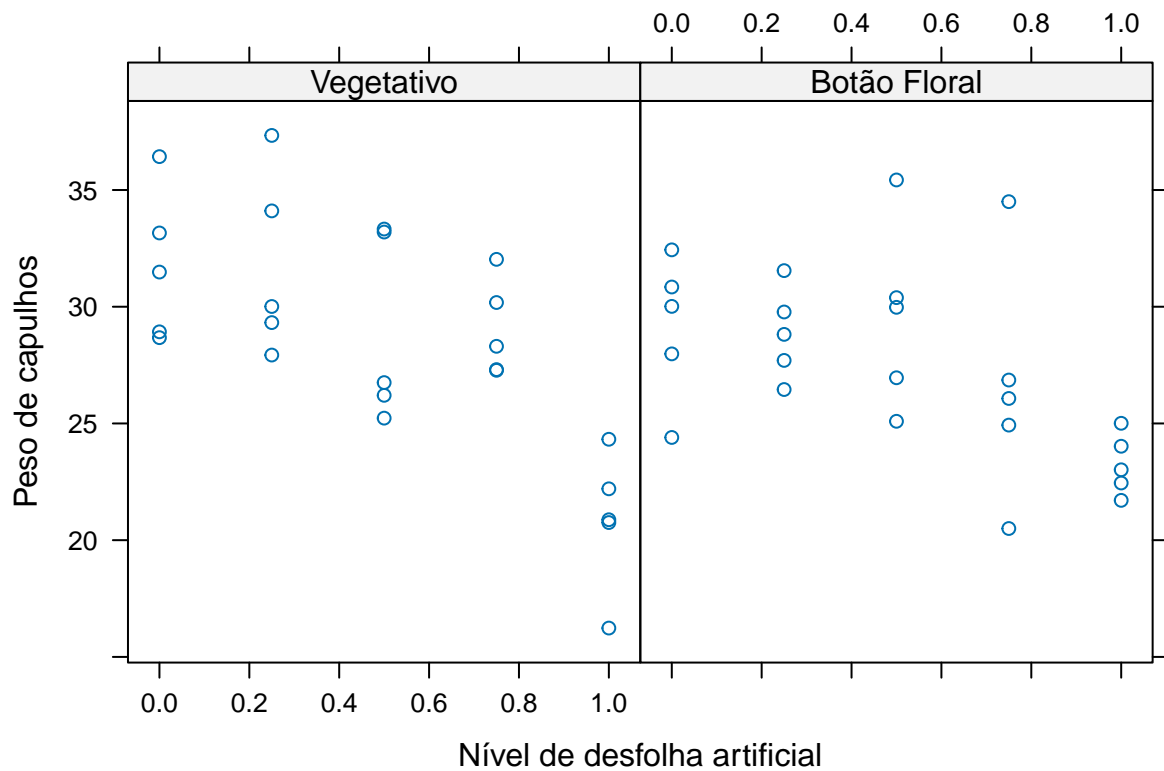
```

# Selecionando dois estágios para comparação:
cap2 <- subset(cap, estag %in% c("Vegetativo", "Botão Floral"))

# Removendo fatores não-utilizados:
cap2$estag <- droplevels(cap2$estag)

# Gráficos marginais por estágio:
xyplot(pcapu ~ desf | estag,
       data = cap2,
       xlab = "Nível de desfolha artificial",
       ylab = "Peso de capulhos")

```



```

# Separação dos dados:
cap_veg <- subset(cap, estag == "Vegetativo")
cap_bot <- subset(cap, estag == "Botão Floral")

# Modelos para cada base:
mod_veg <- nls(formula = pcapu ~ f0 + delta * desf^exp(curv), data = cap_veg,
               start = list(f0 = 30, delta = -5, curv = 1))
mod_bot <- nls(formula = pcapu ~ f0 + delta * desf^exp(curv), data = cap_bot,
               start = list(f0 = 30, delta = -5, curv = 1))

# Modelos:
summary(mod_veg)

```

```
##
## Formula: pcapu ~ f0 + delta * desf^exp(curv)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## f0      31.1492    0.9664  32.231 < 2e-16 ***
## delta -10.1663    1.7366  -5.854 6.86e-06 ***
## curv    1.5021    0.4751   3.162 0.00452 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.292 on 22 degrees of freedom
##
## Number of iterations to convergence: 10
## Achieved convergence tolerance: 9.679e-06
```

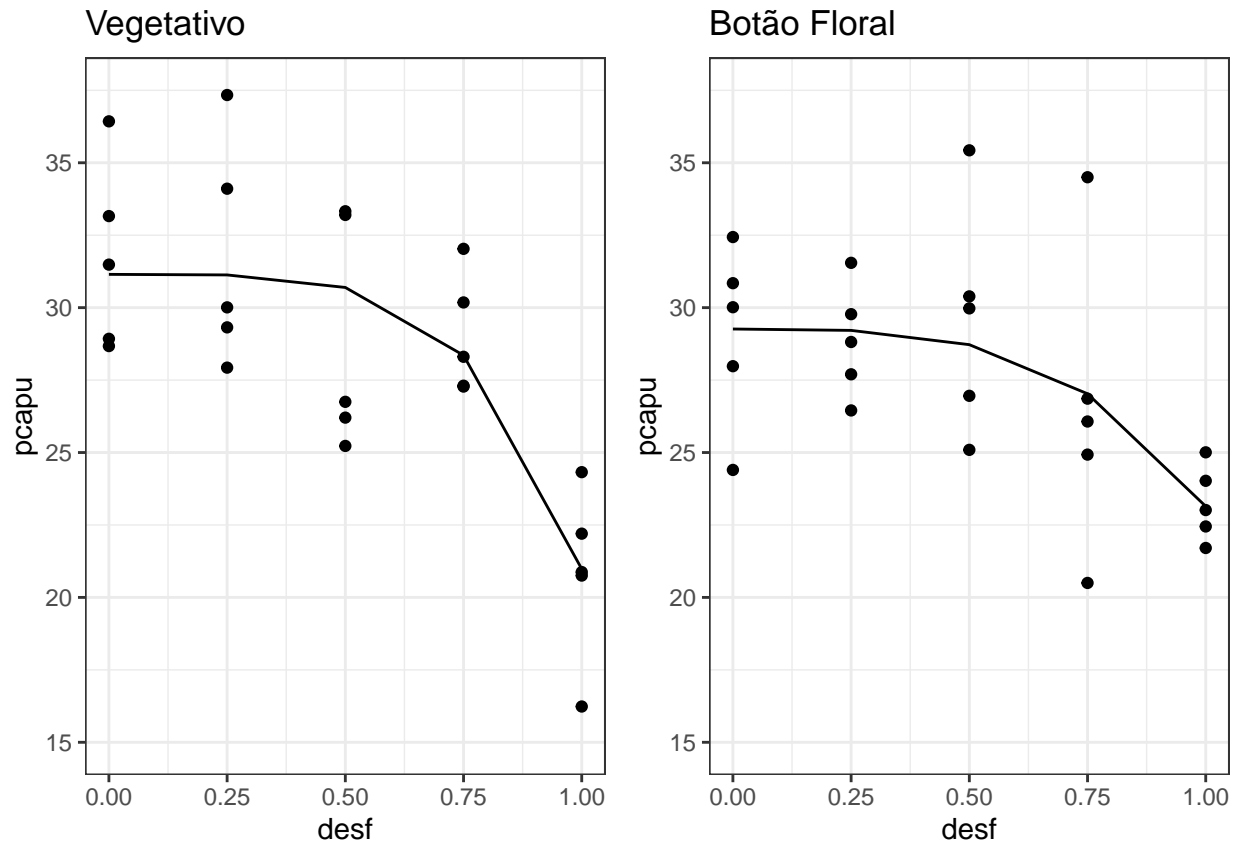
```
summary(mod_bot)
```

```
##
## Formula: pcapu ~ f0 + delta * desf^exp(curv)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## f0      29.2629    1.0180  28.747 < 2e-16 ***
## delta  -6.1174    1.7218  -3.553 0.00178 **
## curv    1.2531    0.7312   1.714 0.10065
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.237 on 22 degrees of freedom
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 6.677e-06
```

```
# Visualização dos modelos ajustados
g_veg <- ggplot(cap_veg, aes(x = desf, y = pcapu)) +
  geom_point() +
  geom_line(aes(y = fitted(mod_veg))) +
  lims(y = c(15, 37.5)) +
  labs(title = "Vegetativo") +
  theme_bw()

g_bot <- ggplot(cap_bot, aes(x = desf, y = pcapu)) +
  geom_point() +
  geom_line(aes(y = fitted(mod_bot))) +
  lims(y = c(15, 37.5)) +
  labs(title = "Botão Floral") +
  theme_bw()

ggpubr::ggarrange(g_veg, g_bot)
```



Pode-se perceber que ambos os modelos foram significativos, a um nível de 95%. Podemos visualizar esses ajustes com os dados e compará-los com um ajuste linear do segundo grau (parábola), e verificar medidas de qualidade do ajuste:

```
# Modelos lineares:
(aj_lin_veg <- lm(pcapu ~ desf + I(desf^2), data = cap_veg))
```

```
##
## Call:
## lm(formula = pcapu ~ desf + I(desf^2), data = cap_veg)
##
## Coefficients:
## (Intercept)      desf      I(desf^2)
##      31.433      5.562     -15.336
```

```
(aj_lin_bot <- lm(pcapu ~ desf + I(desf^2), data = cap_bot))
```

```
##
## Call:
## lm(formula = pcapu ~ desf + I(desf^2), data = cap_bot)
##
## Coefficients:
## (Intercept)      desf      I(desf^2)
##      28.888      5.588     -11.219
```

```

# Comparação dos modelos:
# AIC:
c("NLS" = AIC(mod_veg), "Quad" = AIC(aj_lin_veg))

##      NLS      Quad
## 135.3306 136.7096

c("NLS" = AIC(mod_bot), "Quad" = AIC(aj_lin_bot))

##      NLS      Quad
## 134.4839 134.4990

# Log-Verossimilhança:
c("NLS" = logLik(mod_veg), "Quad" = logLik(aj_lin_veg))

##      NLS      Quad
## -63.66528 -64.35478

c("NLS" = logLik(mod_bot), "Quad" = logLik(aj_lin_bot))

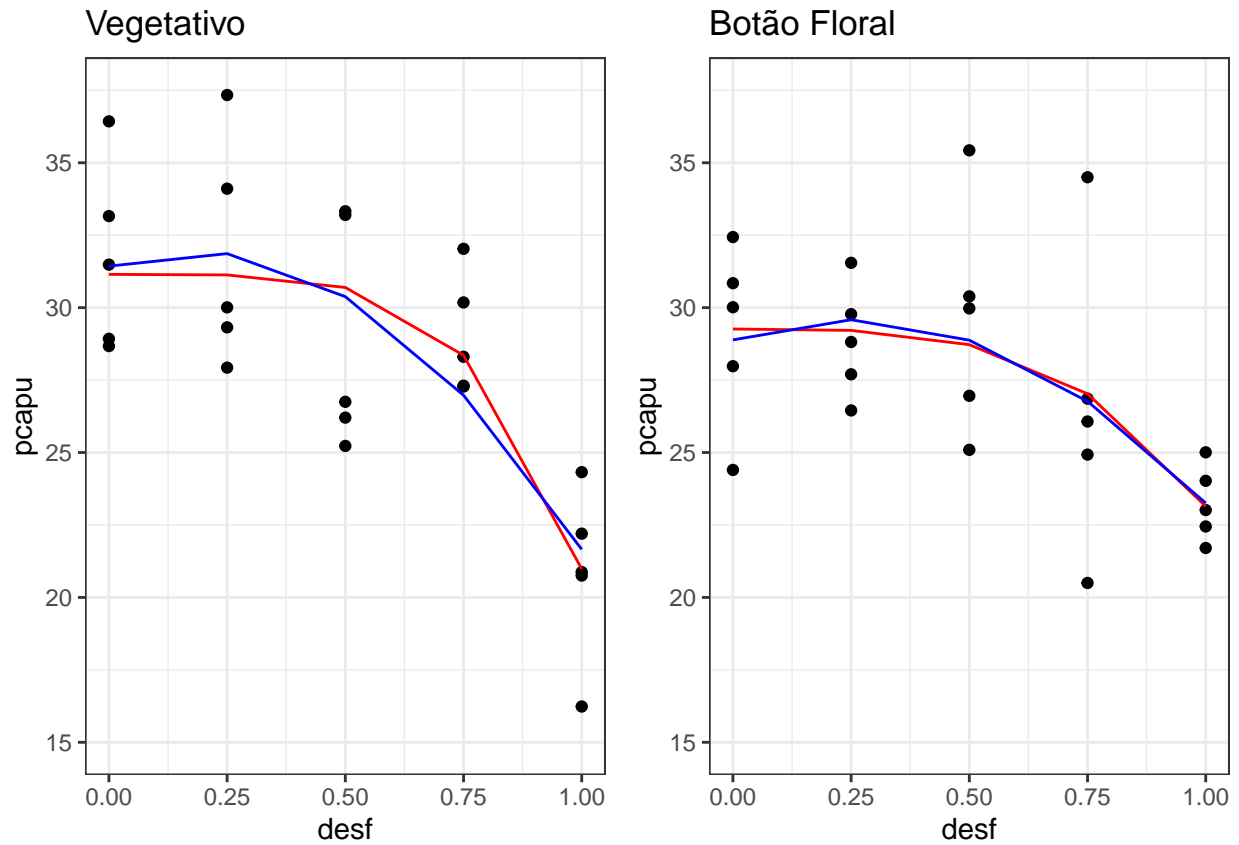
##      NLS      Quad
## -63.24194 -63.24949

# Visualização dos modelos ajustados
g_veg_2 <- ggplot(cap_veg, aes(x = desf, y = pcapu)) +
  geom_point() +
  geom_line(aes(y = fitted(mod_veg)), colour = "red") +
  geom_line(aes(y = fitted(aj_lin_veg)), colour = "blue") +
  lims(y = c(15, 37.5)) +
  labs(title = "Vegetativo") +
  theme_bw()

g_bot_2 <- ggplot(cap_bot, aes(x = desf, y = pcapu)) +
  geom_point() +
  geom_line(aes(y = fitted(mod_bot)), colour = "red") +
  geom_line(aes(y = fitted(aj_lin_bot)), colour = "blue") +
  lims(y = c(15, 37.5)) +
  labs(title = "Botão Floral") +
  theme_bw()

ggpubr::ggarrange(g_veg_2, g_bot_2) #Curva azul é o ajuste quadrático

```



Pode-se perceber que o peso dos capulhos no estágio **Botão Floral** pode ser bem ajustado por um modelo linear de segundo grau, enquanto que os pesos para as plantas no estágio **Vegetativo** são subestimados. Entretanto, como não se deve fazer inferências para os valores fora do espaço amostrado, esses dois modelos diferem bastante e podem resultar em inferências muito diferentes conforme nos afastamos do intervalo amostrado.