CE073 - Análise de Dados Categóricos

Trabalho No. 2

Luiz Henrique Barretta Francisco - GRR20213026

setembro/2023

1-a) Discuta por que essa suposição, de independência entre lançamentos, é necessária e os possíveis problemas com ela.

A suposição de independência entre lançamentos é um dos requisitos para a utilização do modelo de regressão logística aplicado, o que gera:

1. Simplificação do modelo: Ao assumir independência entre os O-rings em cada lançamento, os pesquisadores podem utilizar técnicas estatísticas mais simples, como o uso de distribuições binomiais para modelar a probabilidade de falha. Isso torna a análise mais acessível e fácil de calcular.
2. Facilidade de interpretação: A independência simplifica a interpretação do modelo, tornando mais claro como as variáveis (temperatura e pressão, no caso do Challenger) afetam a probabilidade de falha em cada lançamento.

No entanto, há problemas potenciais associados a essa suposição:

1. Correlações não consideradas: A independência dos O-rings pode não ser uma suposição válida em todos os casos. Por exemplo, se houver problemas sistêmicos de fabricação ou montagem que afetem todos os O-rings em um conjunto de lançamentos, a suposição de independência não será adequada.
2. Violação da independência: Se os O-rings forem de fato dependentes, isso pode levar a uma subestimação da probabilidade de falha. Em outras palavras, a suposição de independência pode levar a uma análise otimista e não refletir com precisão o risco real.
3. Consequências significativas: Em situações onde a falha de um O-ring pode ter consequências extremamente graves, como a destruição de um ônibus espacial, é importante ser extremamente cauteloso ao fazer suposições de independência. A falha em levar em conta a dependência entre os O-rings pode levar a decisões inadequadas de segurança.

1-b) Estime o modelo de regressão logística usando as variáveis explicativas de forma linear.

```
challenger <-  read.csv(file = "http://leg.ufpr.br/~lucambio/CE073/20222S/challenger.csv")
#Transformando todos valores maiores que 0 em 1
challenger$O.ring[challenger$O.ring > 0] <- 1
mod1 <- glm(formula = O.ring ~ Temp + Pressure,
               family = binomial(link = logit), data = challenger)
summary(mod1)
```

```
##
## Call:
## glm(formula = O.ring ~ Temp + Pressure, family = binomial(link = logit),
##     data = challenger)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 13.292360   7.663968   1.734   0.0828 .
## Temp        -0.228671   0.109988  -2.079   0.0376 *
## Pressure     0.010400   0.008979   1.158   0.2468
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 18.782  on 20  degrees of freedom
## AIC: 24.782
##
## Number of Fisher Scoring iterations: 5
```

1-c) Realize testes da razão de verossimilhanças (LRTs) para julgar a importância das variáveis explicativas no modelo.

```
library(package = car)
Anova(mod1 , test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: O.ring
##          LR Chisq Df Pr(>Chisq)
## Temp       7.7542  1   0.005359 **
## Pressure   1.5331  1   0.215648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1-d) Os autores optaram por remover Pressure do modelo com base nos LRTs. Com base em seus resultados, discuta por que você acha que isso foi feito. Há algum problema potencial com a remoção dessa variável?

Com base nos resultados do teste da razão de verossimilhanças (LRTs), os autores optaram por remover a variável "Pressure" do modelo. A decisão de remover essa variável foi tomada porque o valor-p associado ao teste de "Pressure" é relativamente alto (0.215648), indicando que essa variável não é estatisticamente significativa para prever a variável resposta "O.ring" após levar em consideração a variável "Temp" no modelo.

No entanto, é importante notar que a remoção de uma variável do modelo também tem suas próprias implicações:

Perda de informação: A remoção de variáveis pode resultar na perda de informações potencialmente relevantes. É importante considerar o contexto e os objetivos do estudo ao tomar essa decisão.

Simplificação excessiva: A simplificação excessiva do modelo pode levar a um modelo muito simplificado, nesse caso com apenas uma variável explicativa, que pode não capturar nuances importantes nos dados.

2-a) Estime o modelo

```
mod2 <- glm(formula = O.ring ~ Temp,
            family = binomial(link = logit), data = challenger)
summary(mod2)
```

```
##
## Call:
## glm(formula = O.ring ~ Temp, family = binomial(link = logit),
```

```
##     data = challenger)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  15.0429     7.3786   2.039   0.0415 *
## Temp         -0.2322     0.1082  -2.145   0.0320 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 20.315  on 21  degrees of freedom
## AIC: 24.315
##
## Number of Fisher Scoring iterations: 5
```

2-b) Construa dois gráficos: (1) π vs. Temp e (2) Número esperado de falhas vs. Temp. Use uma faixa de temperatura de 31°F a 81°F no eixo x, mesmo que a temperatura mínima no conjunto de dados seja 53°F.

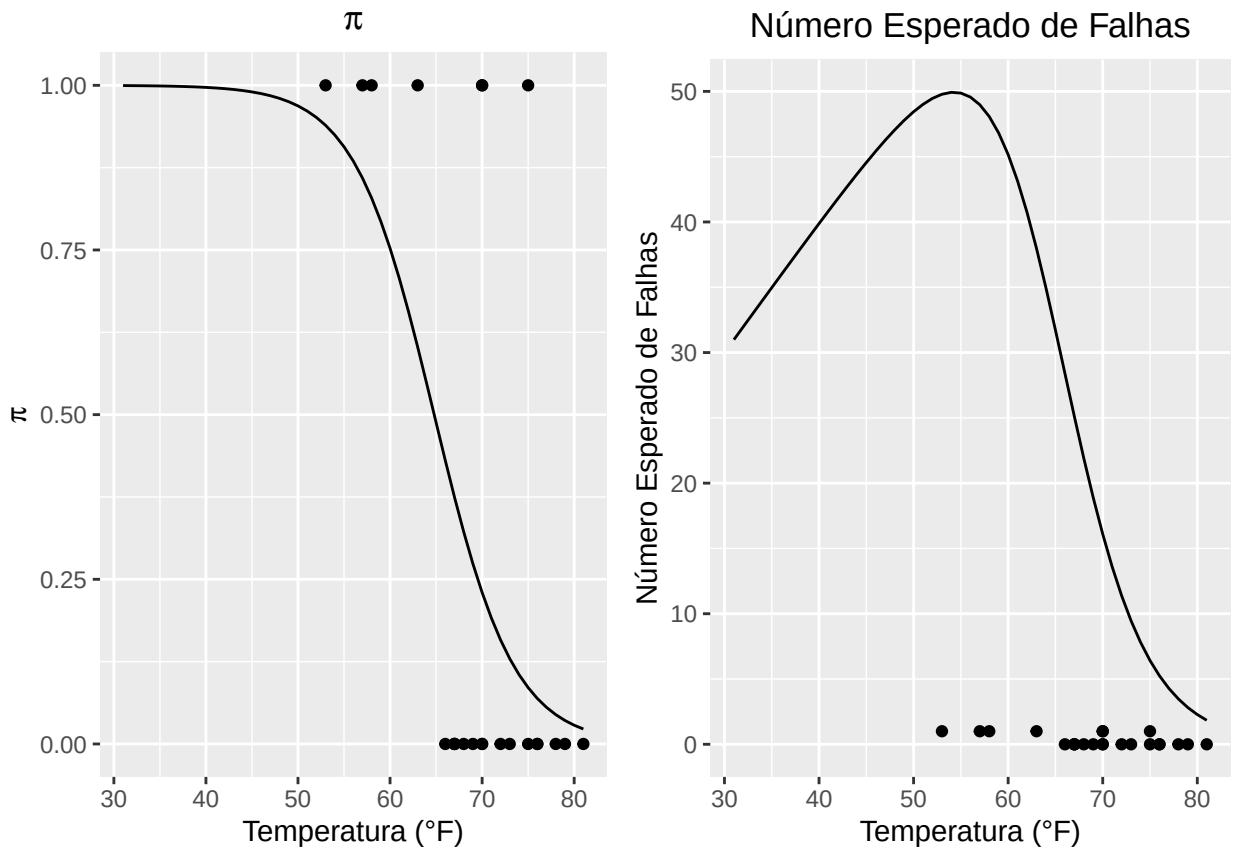
```r
library(ggplot2)
library(gridExtra)

temps <- seq(31, 81, by = 1)
probs <- predict(mod2, newdata = data.frame(Temp = temps), type = "response")
dados_originais <- data.frame(Temp = challenger$Temp, O.ring = challenger$O.ring)
num_esp <- temps * probs

grafico1 <- ggplot(data = data.frame(Temp = temps, Probabilidade = probs),
                   aes(x = Temp, y = Probabilidade)) +
  geom_line() +
  geom_point(data = dados_originais,
             aes(x = Temp, y = O.ring), color = "black", shape = 19) +
  labs(x = "Temperatura (°F)", y = expression(pi)) +
  ggtitle(expression(pi)) + theme(plot.title = element_text(hjust = 0.5))

grafico2 <- ggplot(data = data.frame(Temp = temps, num_esp = num_esp),
                   aes(x = Temp, y = num_esp)) +
  geom_line() +
  geom_point(data = dados_originais,
             aes(x = Temp, y = O.ring), color = "black", shape = 19) +
  labs(x = "Temperatura (°F)", y = "Número Esperado de Falhas") +
  ggtitle("Número Esperado de Falhas") +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(grafico1, grafico2, ncol = 2)
```



2-c) Inclua as bandas do intervalo de confiança de Wald de 95% para no gráfico. Porque as bandas são muito mais largas para temperaturas mais baixas do que para temperaturas mais altas?

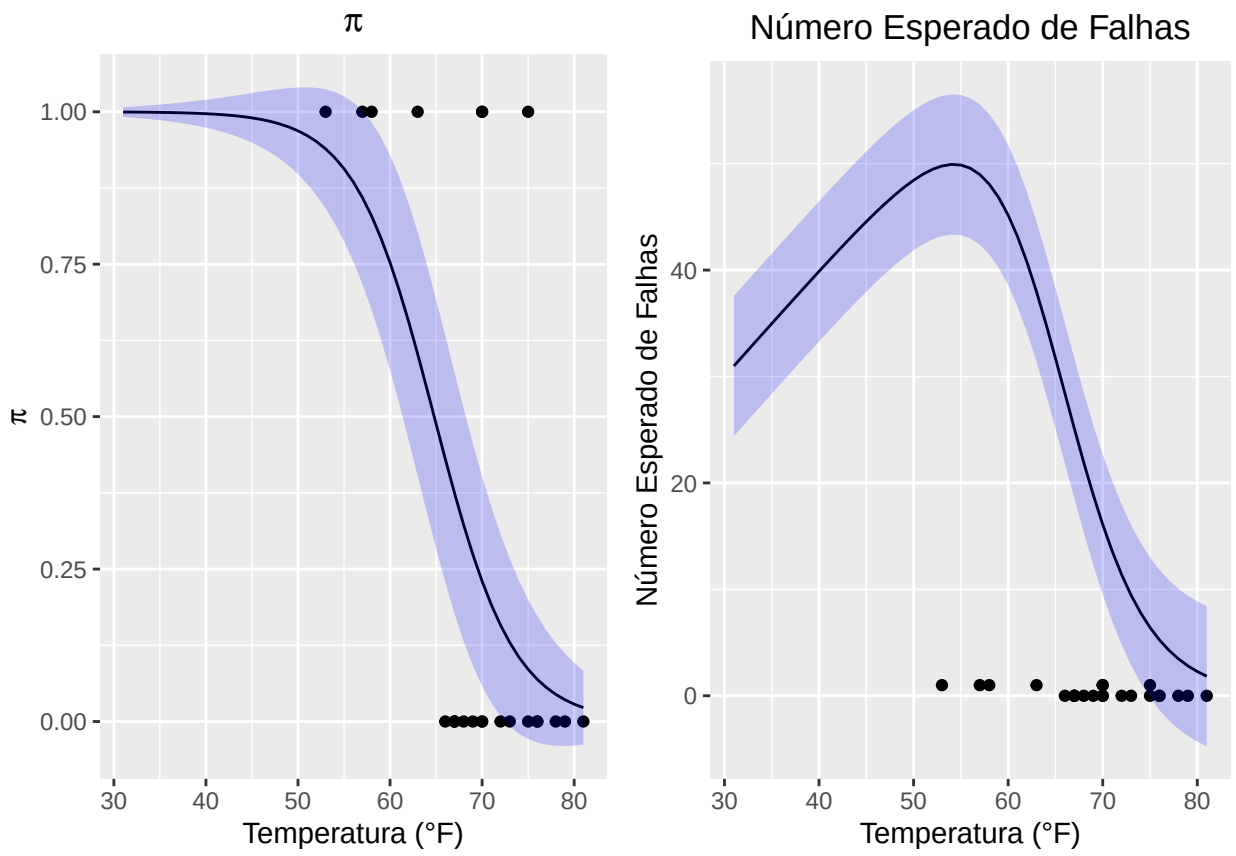
```
se <- sqrt(probs * (1 - probs) / length(dados_originais$O.ring))
z <- qnorm(0.975)  # Valor crítico para o intervalo de confiança de 95%
lower <- probs - z * se
upper <- probs + z * se

grafico3 <- grafico1 +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2, fill = "blue")

se_esp <- sqrt(var(num_esp) / length(dados_originais$O.ring))
lower_esp <- num_esp - z * se_esp
upper_esp <- num_esp + z * se_esp

grafico4 <- grafico2 +
  geom_ribbon(aes(ymin = lower_esp, ymax = upper_esp), alpha = 0.2, fill = "blue")

grid.arrange(grafico3, grafico4, ncol = 2)
```



2-d) A temperatura era 31°F no lançamento do Challenger em 1986. Estime a probabilidade de uma falha do O-ring usando esta temperatura e calcule um intervalo de confiança correspondente. Discuta quais suposições precisam ser feitas para aplicar os procedimentos de inferência.

1. Linearidade: O modelo de regressão logística assume uma relação linear entre as variáveis independentes (no caso, a temperatura) e o logito das chances. É importante considerar se essa suposição é razoável para a faixa de temperaturas em questão.
2. Homoscedasticidade: O modelo pressupõe homoscedasticidade, o que significa que a variância do erro é constante em todas as temperaturas.

```
temp_int <- 31
dados_int <- data.frame(Temp = temp_int)
prob_int <- predict(mod2, newdata = dados_int, type = "response")

paste("Estimativa da probabilidade de falha: ", round(prob_int,4))
```

```
## [1] "Estimativa da probabilidade de falha:  0.9996"
```

```
se_int <- sqrt(prob_int * (1 - prob_int)/length(dados_originais$O.ring))
lower_int <- prob_int - z * se_int
upper_int <- prob_int + z * se_int

cat("Intervalo de confiança: ", paste(round(c(lower_int,upper_int),4)))
```

```
## Intervalo de confiança:  0.9915 1.0077
```

Com base nos resultados, podemos concluir que, de acordo com o modelo, a probabilidade de uma falha do O-ring em 31°F é muito alta e está muito próxima de 1.

2-e)Usando o bootstrap paramétrico, calcule intervalos de confiança de 90% separadamente em temperaturas de 31°F e 72°F.

```
library(boot)


n_amostras_bootstrap <- 1000
set.seed(123)
estimar_probabilidade_falha_31 <- function(data, indices) {
  amostra_bootstrap <- data[indices, ]
  mod_bootstrap <- glm(formula = O.ring ~ Temp, family = binomial(link = logit),
                       data = amostra_bootstrap)
  temp_int <- 31
  dados_int <- data.frame(Temp = temp_int)
  prob_int <- predict(mod_bootstrap, newdata = dados_int, type = "response")
  return(prob_int)}

resultado_bootstrap <- boot(data = challenger, statistic = estimar_probabilidade_falha_31,
                            R = n_amostras_bootstrap)
intervalo_confianca <- boot.ci(resultado_bootstrap, type = "basic", conf = 0.90)
print(intervalo_confianca)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = resultado_bootstrap, conf = 0.9, type = "basic")
##
## Intervals :
## Level      Basic
## 90%   ( 0.9992,  1.0297 )
## Calculations and Intervals on Original Scale
```

```
estimar_probabilidade_falha_72 <- function(data, indices) {
  amostra_bootstrap <- data[indices, ]
  mod_bootstrap <- glm(formula = O.ring ~ Temp, family = binomial(link = logit),
                       data = amostra_bootstrap)
  temp_int <- 72
  dados_int <- data.frame(Temp = temp_int)
  prob_int <- predict(mod_bootstrap, newdata = dados_int, type = "response")
  return(prob_int)}

resultado_bootstrap <- boot(data = challenger, statistic = estimar_probabilidade_falha_72,
                            R = n_amostras_bootstrap)

intervalo_confianca <- boot.ci(resultado_bootstrap, type = "basic", conf = 0.90)
print(intervalo_confianca)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = resultado_bootstrap, conf = 0.9, type = "basic")
##
## Intervals :
## Level      Basic
## 90%   (-0.0251,  0.3046 )
## Calculations and Intervals on Original Scale
```

2-f) Determine se um termo quadrático é necessário no modelo para a temperatura.

```
mod3 <- glm(formula = O.ring ~ Temp + I(Temp^2), family = binomial(link = logit),
            data = challenger)

lrt_statistic <- 2 * (deviance(mod2) - deviance(mod3))
lrt_statistic
```

```
## [1] 1.85298
```

```
p_valor <- 1 - pchisq(lrt_statistic, df.residual(mod2) - df.residual(mod3))
p_valor
```

```
## [1] 0.1734372
```

O valor p obtido é maior que o nível de significância comum de 0.05. Isso indica que não há evidência estatisticamente significativa para rejeitar a hipótese nula de que o modelo linear (sem o termo quadrático) é tão bom quanto o modelo com termo quadrático. Em outras palavras, a inclusão do termo quadrático não melhora significativamente o ajuste do modelo.

Portanto, com base nos resultados do teste LRT, não é necessário incluir um termo quadrático no modelo para a temperatura. O modelo linear parece ser adequado para descrever a relação entre a temperatura e a probabilidade de falha, uma vez que a adição do termo quadrático não resulta em uma melhoria estatisticamente significativa no ajuste do modelo.

3-a) Investigue se a probabilidade estimada de falha teria mudado significativamente se seu modelo de regressão probit ou log-log complementar fosse usado em vez de ummodelo de regressão logística.

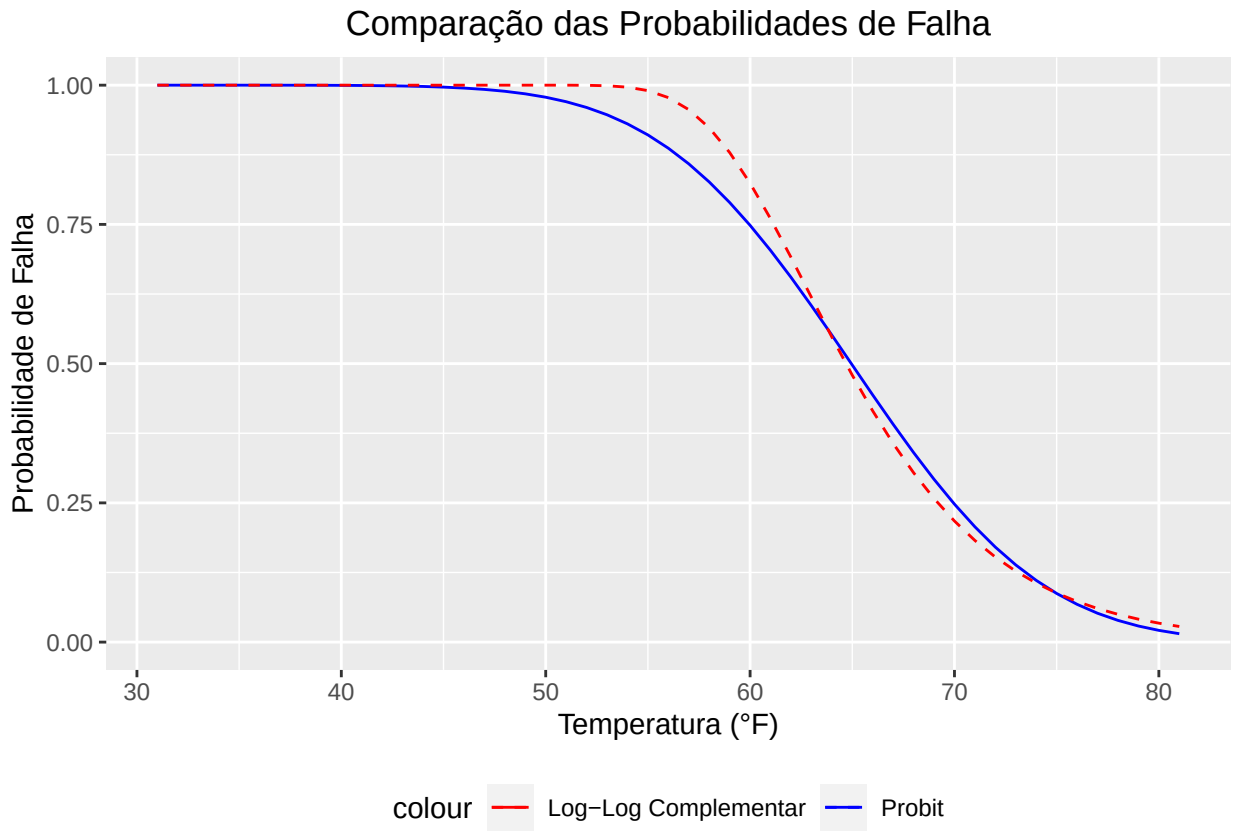
```
mod_probit <- glm(formula = O.ring ~ Temp, family = binomial(link = probit),
                  data = challenger)
probs_probit <- predict(mod_probit, newdata = data.frame(Temp = temps), type = "response")

mod_loglog <- glm(formula = O.ring ~ Temp, family = binomial(link = cloglog), data = challenger)
probs_loglog <- predict(mod_loglog, newdata = data.frame(Temp = temps), type = "response")

df_probs <- data.frame(Temp = temps, Probit = probs_probit, LogLogComplementar = probs_loglog)
```

```r
grafico_comparativo <- ggplot(df_probs, aes(x = Temp)) +
  geom_line(aes(y = Probit, color = "Probit"), linetype = "solid") +
  geom_line(aes(y = LogLogComplementar, color = "Log-Log Complementar"), linetype = "dashed") +
  labs(x = "Temperatura (°F)", y = "Probabilidade de Falha") +
  ggtitle("Comparação das Probabilidades de Falha") +
  scale_y_continuous(limits = c(0, 1)) +
  scale_color_manual(values = c("Probit" = "blue", "Log-Log Complementar" = "red")) +
  theme(plot.title = element_text(hjust = 0.5),
        legend.position = "bottom")

grafico_comparativo
```



3-b) Investigue também se aplicar uma função de ligação paramétrica aprimora a probabilidade de falha.

```r
aic_logit <- AIC(mod2)
aic_probit <- AIC(mod_probit)
aic_cloglog <- AIC(mod_loglog)
valores_aic <- c(aic_logit, aic_probit, aic_cloglog)
tabela_aic <- data.frame(Modelo = c("Logit", "Probit", "Log-Log Complementar"), AIC = valores_aic)

library(knitr)
kable(tabela_aic, format = "markdown")
```

Modelo	AIC
Logit	24.31519
Probit	24.37774
Log-Log Complementar	23.53146

Ao comparar os valores de AIC para os três modelos, observamos que o modelo "Log-Log Complementar" possui o valor mais baixo de AIC (23.53146), indicando que ele fornece o melhor ajuste aos dados em comparação com os outros modelos. Isso sugere que a aplicação da função de ligação paramétrica específica desse modelo aprimorou a probabilidade de falha em relação aos modelos "Logit" e "Probit". Portanto, com base no critério AIC, o modelo "Log-Log Complementar" é a escolha preferida, uma vez que oferece um ajuste estatisticamente superior e uma melhor descrição dos dados, considerando a complexidade do modelo.

4-a) Estime um modelo de regressão logística usando a quantidade de picloram como variável explicativa e o número de ervas daninhas mortas como variável de resposta.

```
dados <-  read.csv(file = "http://leg.ufpr.br/~lucambio/ADC/picloram.csv")
dados$kill <- dados$kill / dados$total
mod_picloram <- glm(formula = kill ~ picloram, family = binomial(link = logit), data = dados)
summary(mod_picloram)
```

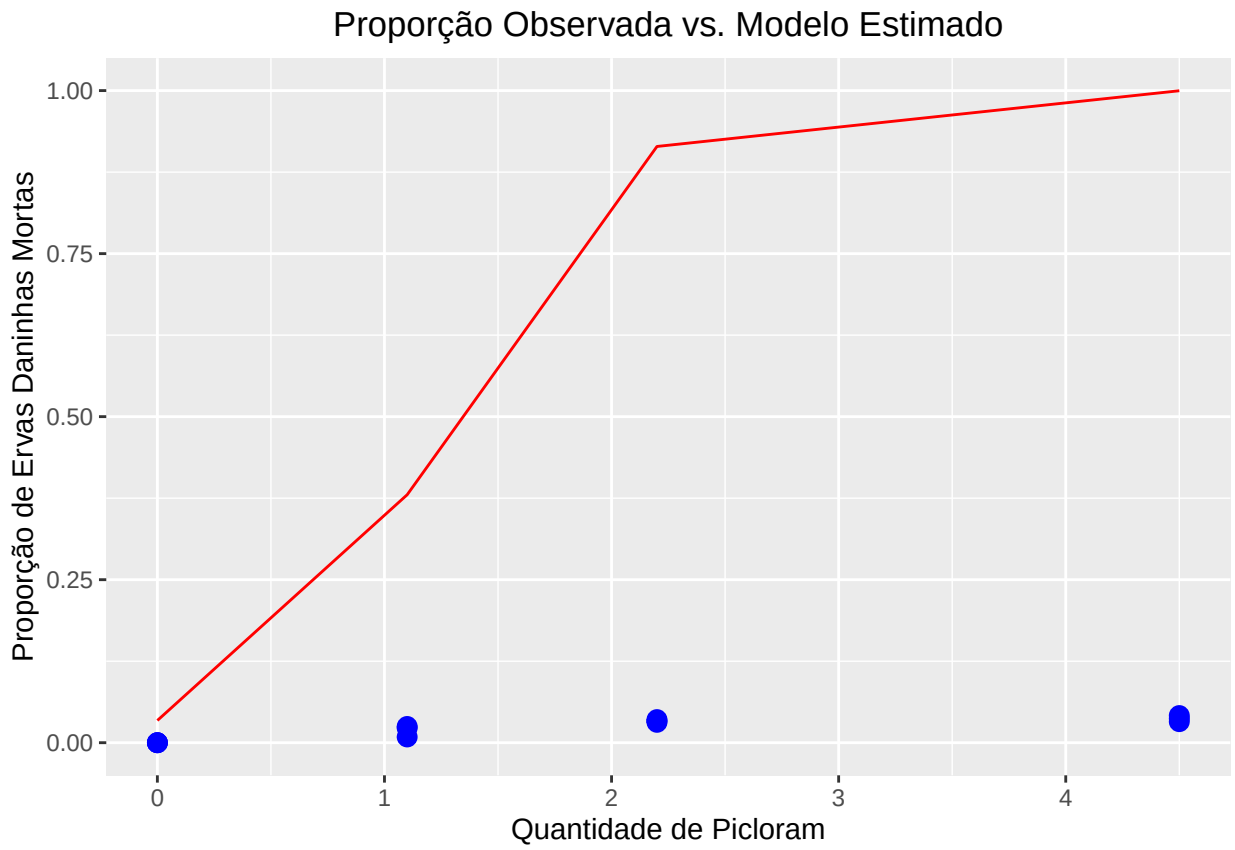
```
##
## Call:
## glm(formula = kill ~ picloram, family = binomial(link = logit),
##     data = dados)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.345      2.224  -1.504    0.133
## picloram       2.597      1.602   1.621    0.105
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10.39178  on 11  degrees of freedom
## Residual deviance:  0.72588  on 10  degrees of freedom
## AIC: 9.5712
##
## Number of Fisher Scoring iterations: 7
```

4-b) Trace a proporção observada de ervas daninhas mortas e o modelo estimado. Descreva o quão bem o modelo se ajusta aos dados. Investigue também se aplicar uma função de ligação paramétrica aprimora a probabilidade de falha.

```
library(ggplot2)
df_plot <- data.frame(Picloram = dados$picloram,
                      ProporcaoObservada = dados$kill / dados$total)
df_plot$ProbabilidadeEstimada <- predict(mod_picloram, newdata = dados, type = "response")

ggplot(data = df_plot, aes(x = Picloram)) +
```

```
  geom_point(aes(y = ProporcaoObservada), color = "blue", size = 3) +
  geom_line(aes(y = ProbabilidadeEstimada), color = "red") +
  labs(x = "Quantidade de Picloram", y = "Proporção de Ervas Daninhas Mortas") +
  ggtitle("Proporção Observada vs. Modelo Estimado") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
mod_probit <- glm(formula = kill ~ picloram, family = binomial(link = probit), data = dados)
mod_cloglog <- glm(formula = kill ~ picloram, family = binomial(link = cloglog), data = dados)

aic_logit <- AIC(mod_picloram)
aic_probit <- AIC(mod_probit)
aic_cloglog <- AIC(mod_cloglog)
valores_aic <- c(aic_logit, aic_probit, aic_cloglog)
tabela_aic <- data.frame(Modelo = c("Logit", "Probit", "Log-Log Complementar"), AIC = valores_aic)

kable(tabela_aic, format = "markdown")
```

Modelo	AIC
Logit	9.571192
Probit	9.575085
Log-Log Complementar	10.207891

Com base nos valores de AIC calculados, o modelo preferível é o "Logit" (com uma função de ligação logit). Isso ocorre porque ele apresenta o menor valor de AIC (9.571192), o que indica um melhor ajuste aos dados

em comparação com os modelos “Probit” e “Log-Log Complementar”. Portanto, o modelo logit é a escolha preferida para modelar a relação entre a quantidade de picloram e o número de ervas daninhas mortas neste conjunto de dados.

No que diz respeito à aplicação de diferentes funções de ligação paramétrica, não parece haver uma melhoria significativa na probabilidade de falha ao alternar entre o logit e o probit, uma vez que os valores de AIC são muito próximos entre esses dois modelos. No entanto, o modelo log-log complementar apresenta um valor de AIC significativamente mais alto, o que sugere que essa função de ligação não é a mais adequada para descrever a relação nos dados.

Portanto, escolher o modelo logit como preferido indica que a função de ligação logit é a mais apropriada para modelar a probabilidade de falha neste contexto.

4-c) Estime o nível de taxa de morte de 0.9 para o picloram. Adicione linhas ao gráfico em (b).

```
ld90_dose <- approx(dados$kill, dados$picloram, xout = 0.9)$y
plot(dados$picloram, dados$kill, type = "o", xlab = "Dose de Picloram", ylab = "Taxa de Morte",
     main = "Estimativa do LD90 para Picloram")
```