Questão 4 - Prova 2 de Modelos Lineares Generalizados

Mariana Costa Freitas

2025-02-12

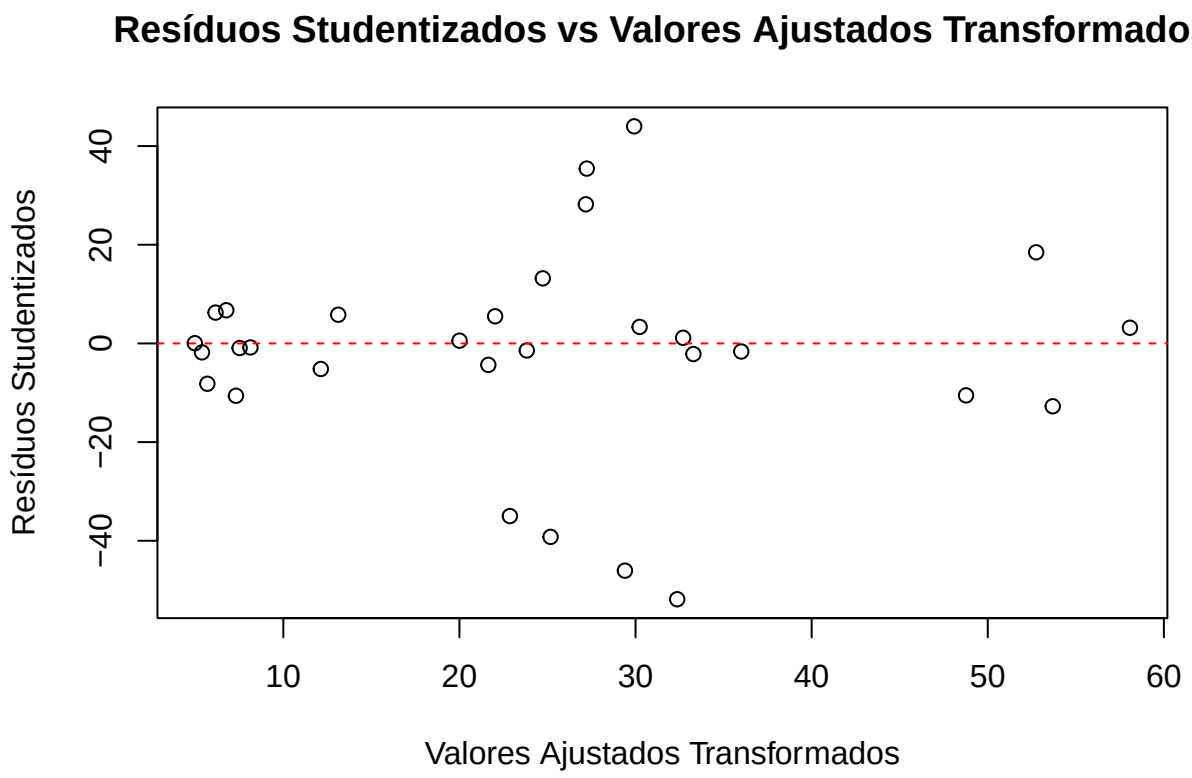
Item a)

Primeiro carregamos os dados e definimos o modelo selecionado na prova anterior.

```
#Carregando os pacotes utilizados
library(GLMsData)
#Carregando os dados
data(ccancer)
# Modelo selecionado
modelo_log <- glm(Count ~ Gender + Region + Site + Population,
data = ccancer,
family = poisson(link = "log"))
```

Um dos primeiros aspectos a serem analisados é a variabilidade, visto que é necessário checar a homocesdasticidade do resíduo, ou seja, se a variância dos resíduos é constante. Para isso, temos abaixo um gráfico de dispersão com os resíduos estudentizados representados no eixo X e o índice desses resíduos no eixo Y. Se notarmos algum padrão na distribuição desses resíduos, haverá heterocedasticidade.

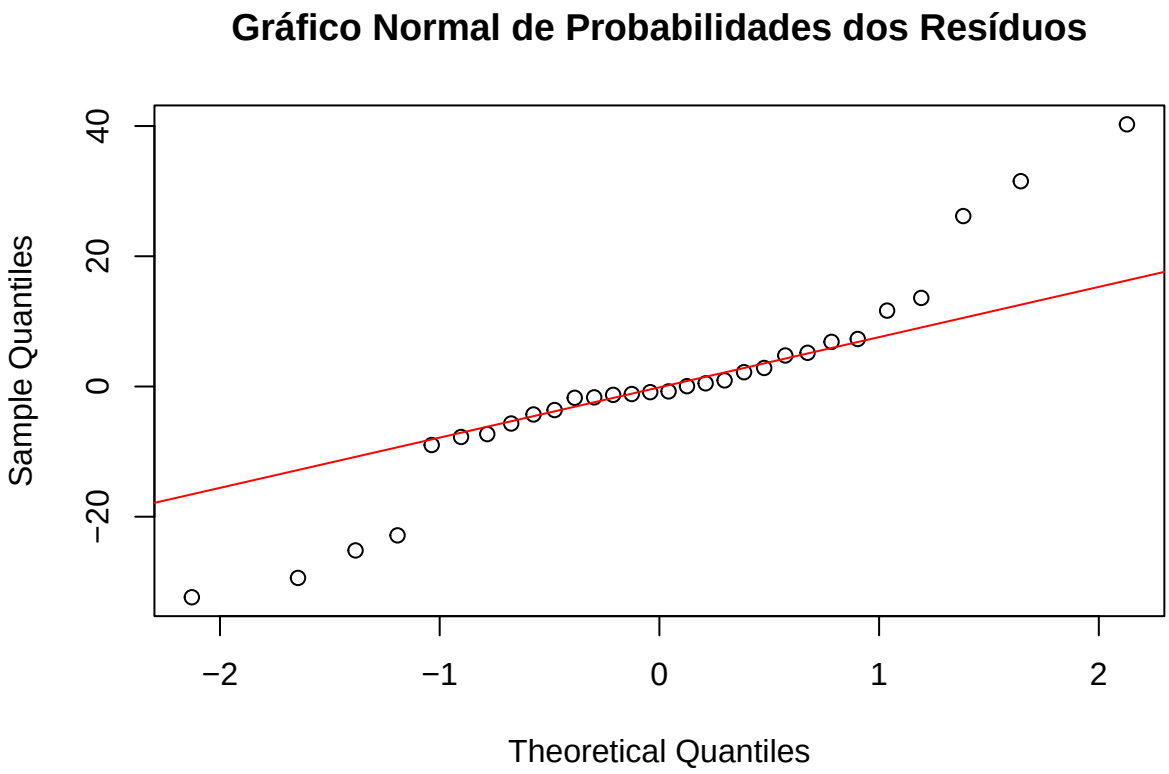
```
# Calcular resíduos studentizados
residuos_student <- rstudent(modelo_log)

# Calcular valores ajustados
valores_ajustados <- predict(modelo_log, type = "response")

# Transformar valores ajustados (raiz quadrada)
valores_ajustados_transformados <- sqrt(valores_ajustados)

# Gráfico de resíduos studentizados versus valores ajustados transformados
plot(valores_ajustados_transformados, residuos_student, main = "Resíduos Studentizados vs Valores Ajusta
abline(h = 0, col = "red", lty = 2)
```



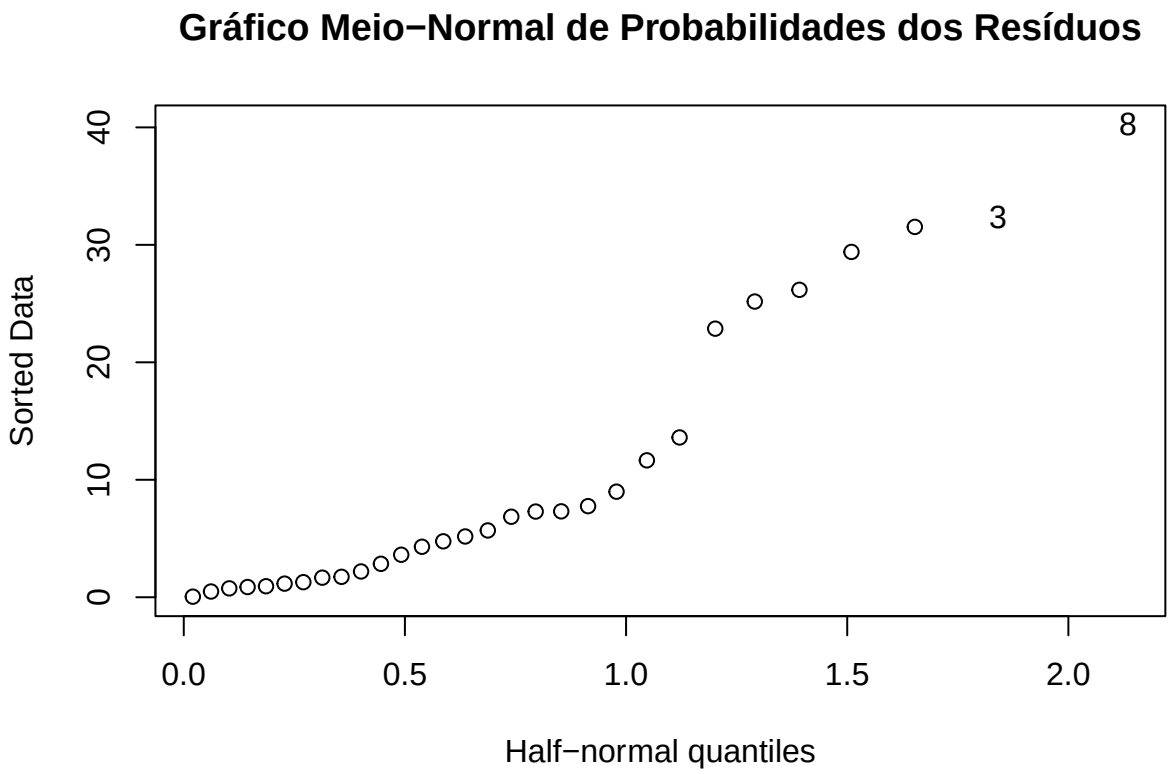
Como os resíduos parecem aleatoriamente distribuídos em torno de zero, sem padrões claros, assumismo que a variância é contante, ou seja, que há homocedasticidade.

Outro aspecto a ser considerado é verificar se os resíduos seguem normalidade, utilizando os gráficos abaixo.

```
# Resíduos padronizados
residuos_padronizados <- residuals(modelo_log, type = "pearson")

# Gráfico Normal de probabilidades dos resíduos
qqnorm(residuos_padronizados, main = "Gráfico Normal de Probabilidades dos Resíduos")
qqline(residuos_padronizados, col = "red")

# Gráfico meio-normal (half-normal) de probabilidades dos resíduos
library(faraway)
```



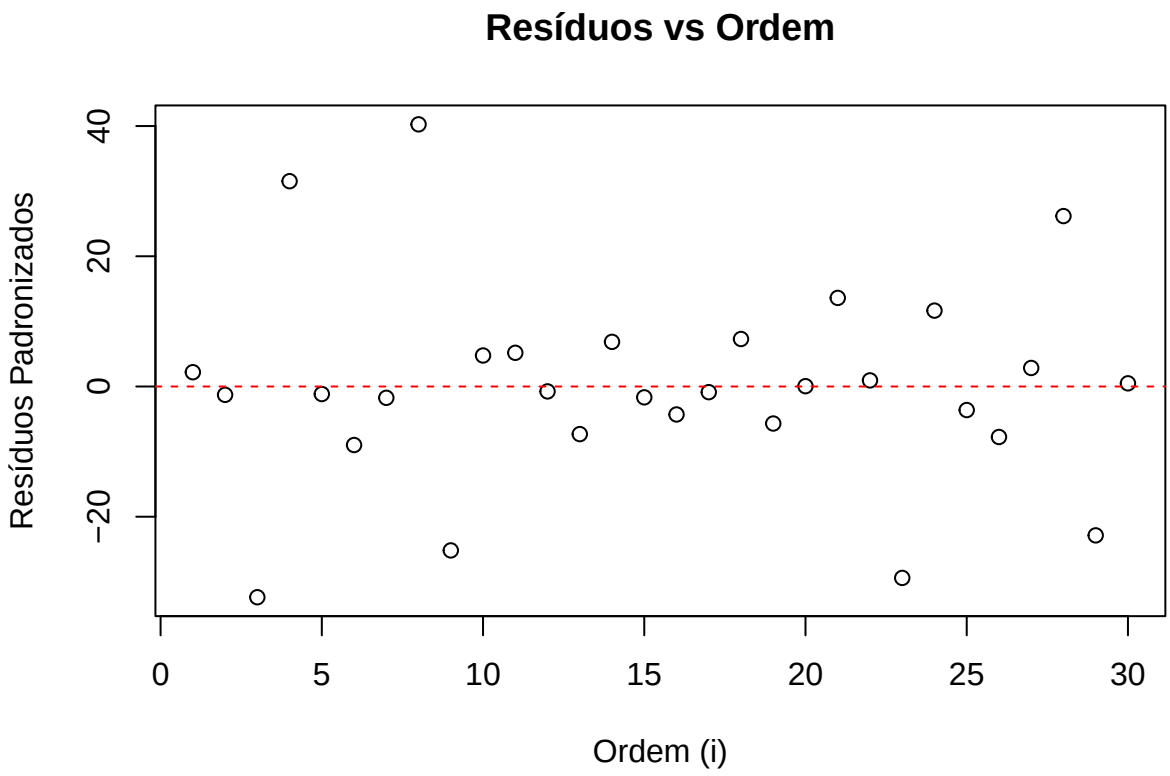

```
halfnorm(residuos_padronizados, main = "Gráfico Meio-Normal de Probabilidades dos Resíduos")
```



Já que os pontos nas extremidades do gráfico se desviam significativamente da linha vermelha (que representa a distribuição normal teórica esperada), indiando que a distribuição dos resíduos não é normal, o que é esperado para o modelo de Poisson. A forma com que os resíduos se distribuem lembra levemente um S, o que é comum para distribuições com caudas muito curtas. Os pontos 3 e 8 mostrados no segundo gráfico são observações com resíduos consideravelmente maiores do que o esperado, podendo ser pontos aberrantes.

Também é importante observar se há alguma correlação entre os resíduos e os índices das observações, o que pode ser avaliado a partir de um gráfico de dispersão de resíduos e ordem, como o mostrado abaixo.

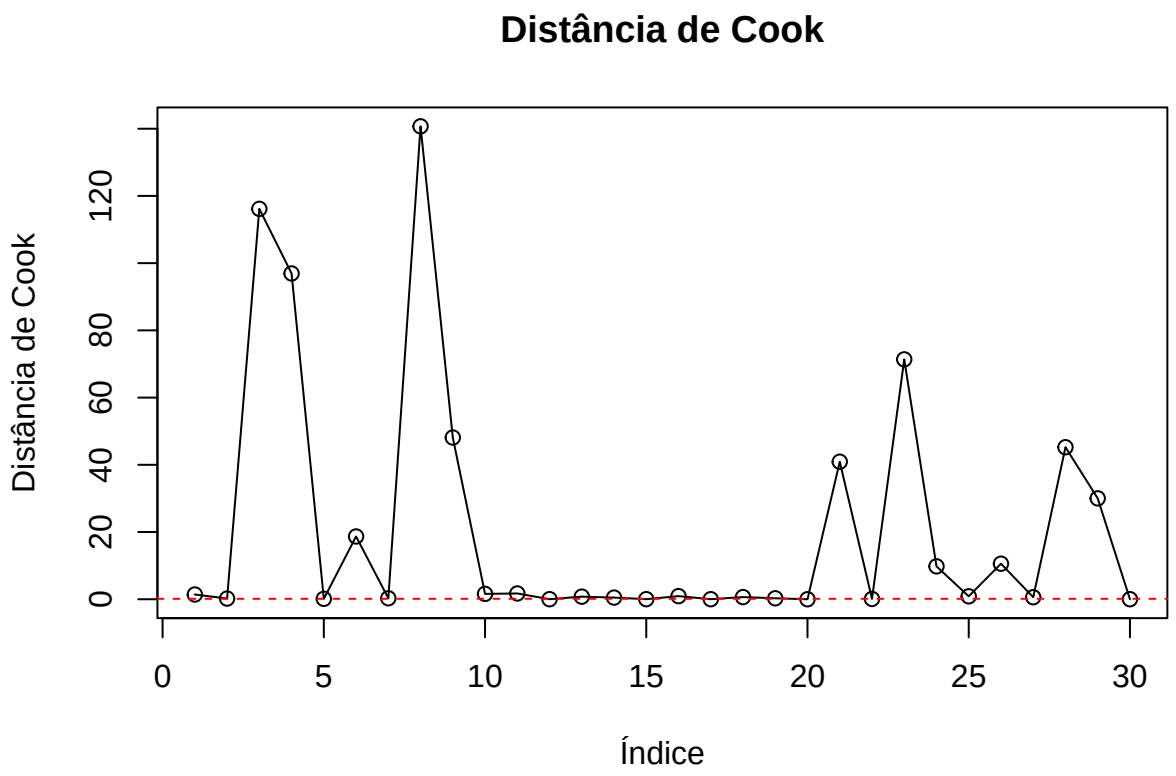
```
# Resíduos versus ordem das observações
plot(residuos_padronizados, type = "p", main = "Resíduos vs Ordem", xlab = "Ordem (i)", ylab = "Resíduos
abline(h = 0, col = "red", lty = 2)
```



Como os resíduos parecem aleatoriamente distribuídos em torno de zero, não há evidência de correlação.

Agora vamos identificar os pontos influentes, são aquelas observações que, quando removidas do conjunto de dados, causam uma mudança significativa nas estimativas dos coeficientes do modelo. Para isso vamos usar a distância de Cook, que mede o impacto de uma observação no ajuste do modelo.

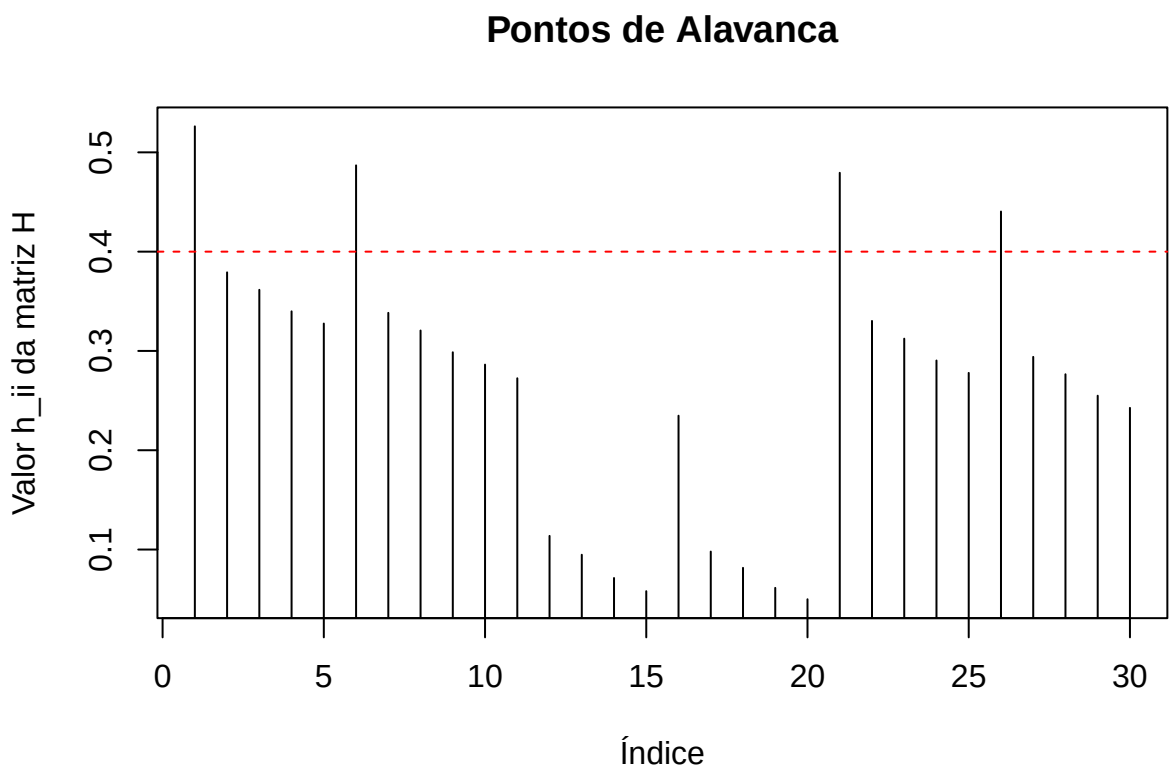
```
# Distância de Cook
distancia_cooks <- cooks.distance(modelo_log)

# Plotar distância de Cook
plot(distancia_cooks, type = "o", main = "Distância de Cook", xlab = "Índice", ylab = "Distância de Cook
abline(h = 4 / length(distancia_cooks), col = "red", lty = 2)
```



É possível observar que, das 30 observações, aquelas com índices 3, 4, 6, 8, 9, 21, 23, 24, 26, 28 e 29 são considerados influentes.

Além dos pontos influentes, podemos também identificar os pontos de alavanca, ou seja, aqueles que têm uma influência desproporcional no próprio valor ajustado. São identificados a partir da mariz de projeção H.

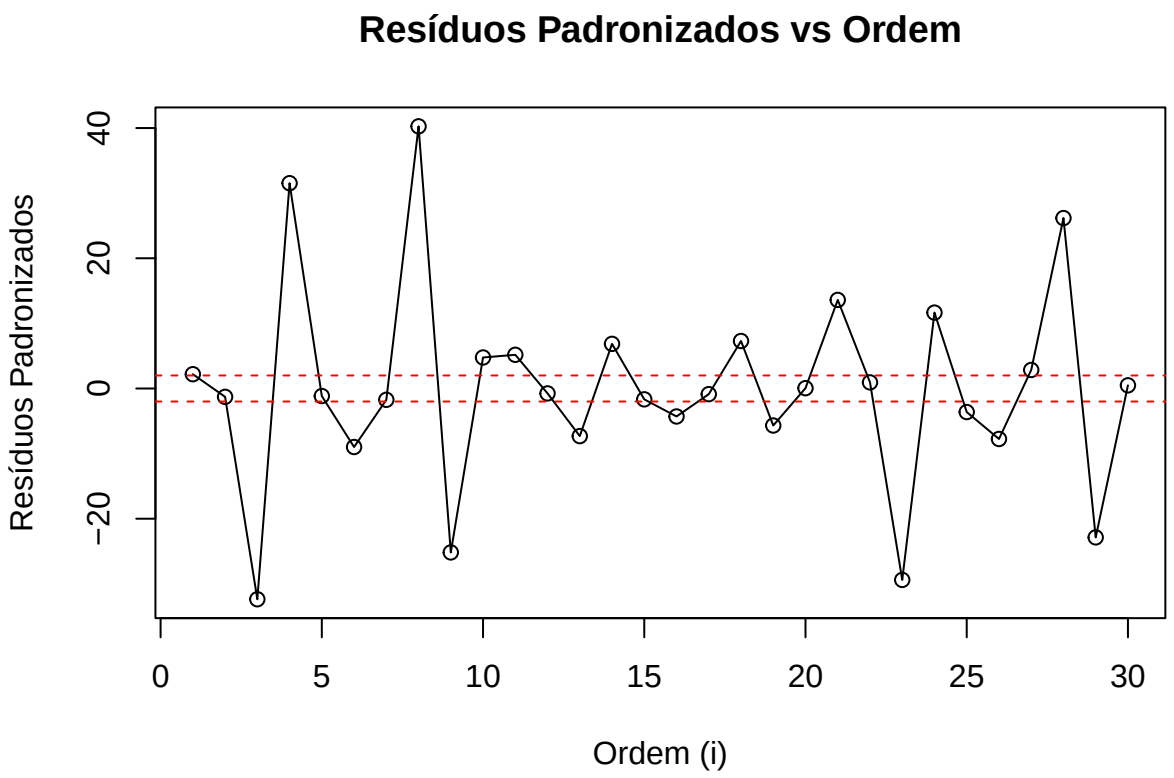
```
#cálculo dos valores de alavanca
h <- hatvalues(modelo_log)

#visualizar valores de alavanca
plot(h, type = "h", main = "Pontos de Alavanca", xlab = "Índice", ylab = "Valor h_ii da matriz H")
abline(h = 1.5 * mean(h), col = "red", lty = 2)
```



Aqui, é possível observar que os pontos de alavanca são as observações de índice 1, 6, 21 e 26.

Por último vamos identificar os pontos aberrantes, ou seja, aqueles que têm resíduos altos e parecem mal ajustados. Para isso, analisaremos no gráfico abaixo as observações que têm resíduos padronizados fora do intervalo [-2, 2].

```r
# Resíduos padronizados
residuos_padronizados <- residuals(modelo_log, type = "pearson")

plot(residuos_padronizados, type = "o", main = "Resíduos Padronizados vs Ordem", xlab = "Ordem (i)", yla
abline(h = c(-2, 2), col = "red", lty = 2)  # Limites comuns para resíduos padronizados
```



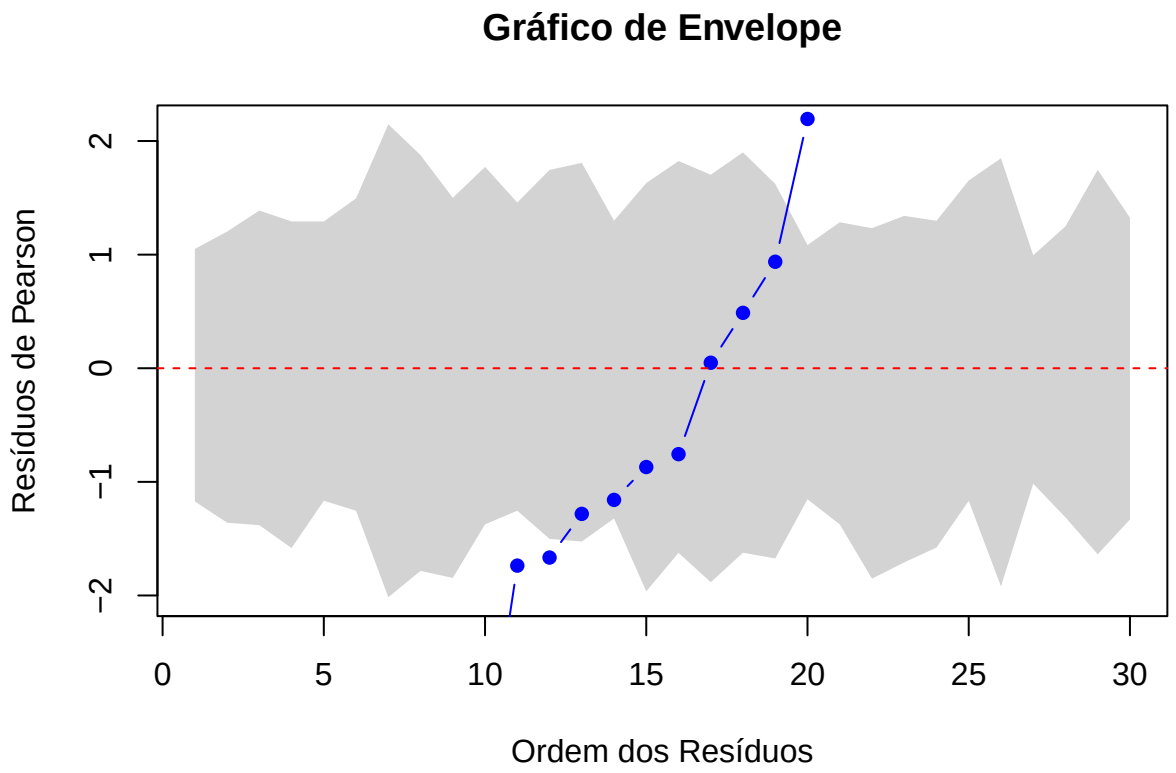
Assim, é possível observar que há uma grande presença de dados aberrantes, podendo indicar que talvez o modelo utilizado não é aquele que melhor se ajusta aos dados.

Item b)

O gráfico de evelope compara os resíduos do modelo ajustado com os resíduos simulados considerando que o modelo corredo foi escolhido. Se os resíduos observados estiverem dentro do envelope (bandas de confiança, então o modelo está bem ajustado.

```r
# Resíduos observados
residuos_obs <- residuals(modelo_log, type = "pearson")

# Valores ajustados
mu <- predict(modelo_log, type = "response")

# Simulação dos resíduos
n <- length(mu)
residuos_sim <- matrix(NA, nrow = n, ncol = 100)

for (i in 1:100) {
    y_sim <- rpois(n, lambda = mu)  # Simula novos dados
    modelo_sim <- glm(y_sim ~ ., data = modelo_log$data, family = poisson(link = "log"))
    residuos_sim[, i] <- residuals(modelo_sim, type = "pearson")
  }
```

```r
# Calcular intervalos de confiança (95%)
envelope_inf <- apply(residuos_sim, 1, quantile, probs = 0.025)
envelope_sup <- apply(residuos_sim, 1, quantile, probs = 0.975)

# Ordenar resíduos observados e esperados
ordem <- order(residuos_obs)
residuos_obs_ord <- residuos_obs[ordem]
envelope_inf_ord <- envelope_inf[ordem]
envelope_sup_ord <- envelope_sup[ordem]

# Gráfico de envelope
plot(residuos_obs_ord, type = "n", ylim = range(c(envelope_inf_ord, envelope_sup_ord)),
       main = "Gráfico de Envelope", xlab = "Ordem dos Resíduos", ylab = "Resíduos de Pearson")
  polygon(c(1:n, n:1), c(envelope_sup_ord, rev(envelope_inf_ord)), col = "lightgray", border = NA)
  lines(residuos_obs_ord, type = "b", pch = 16, col = "blue")
  abline(h = 0, col = "red", lty = 2)
```



Como alguns resíduos observados estão fora do "envelope", o modelo não está tão bem ajustado aos dados quanto gostaríamos.

O gráfico de envelope pode ajudar a verificar os pressupostos de normalidade dos resíduos e de homocedasticidade do modelo. O comportamento dos pontos mostra uma tendência crescente, o que pode indicar heterocedasticidade, enquanto os resíduos fora da faixa cinza, principalmente nas extremidades, indicam que os resíduos não seguem uma distribuição aproximadamente normal.