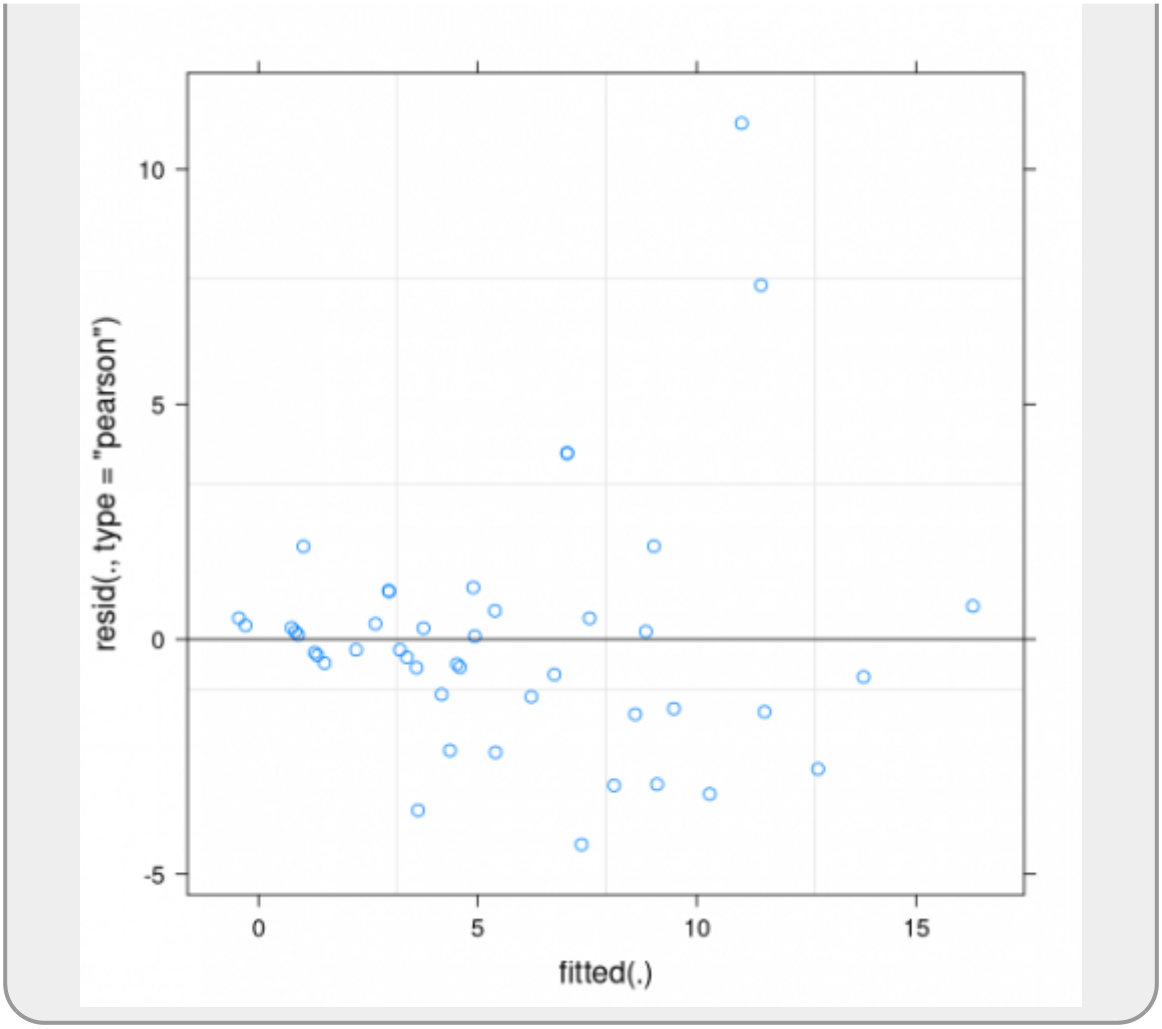
Modelos Generalizados Mistos (GLMM)

Este roteiro é uma continuação do roteiro de Modelos Lineares Mistos (LMM), o qual deve ser feito e entendido bem antes de continuar. Também é importante que você já tenha estudado o roteiro de Modelos Lineares Generalizados (GLMs). Os modelos GLMM são uma extensão dessas duas abordagens reunidas.

Ao final do roteiro de LMMs nós observamos que os resíduos do modelo final não atendiam às premissas de homogeneidade de variâncias e normalidade do resíduos. No próprio roteiro, lembramos que a riqueza de espécies (número de espécies no local) é um dado de contagem (valores sempre inteiros e positivos - discretos) e, portanto, não poderíamos utilizar um modelo que tem como premissa a distribuição normal (valores contínuos). Logo, precisaremos utilizar um modelo com uma distribuição diferente da normal. Neste exemplo, vamos recorrer a um GLM com distribuição de `Poisson` - que lida com os dados de contagem, assim como fizemos no roteiro GLM contagem .

Gráfico básico de diagnóstico do Rcmdr com a distribuição dos resíduos do modelo selecionado no roteiro Modelos Lineares Mistos.

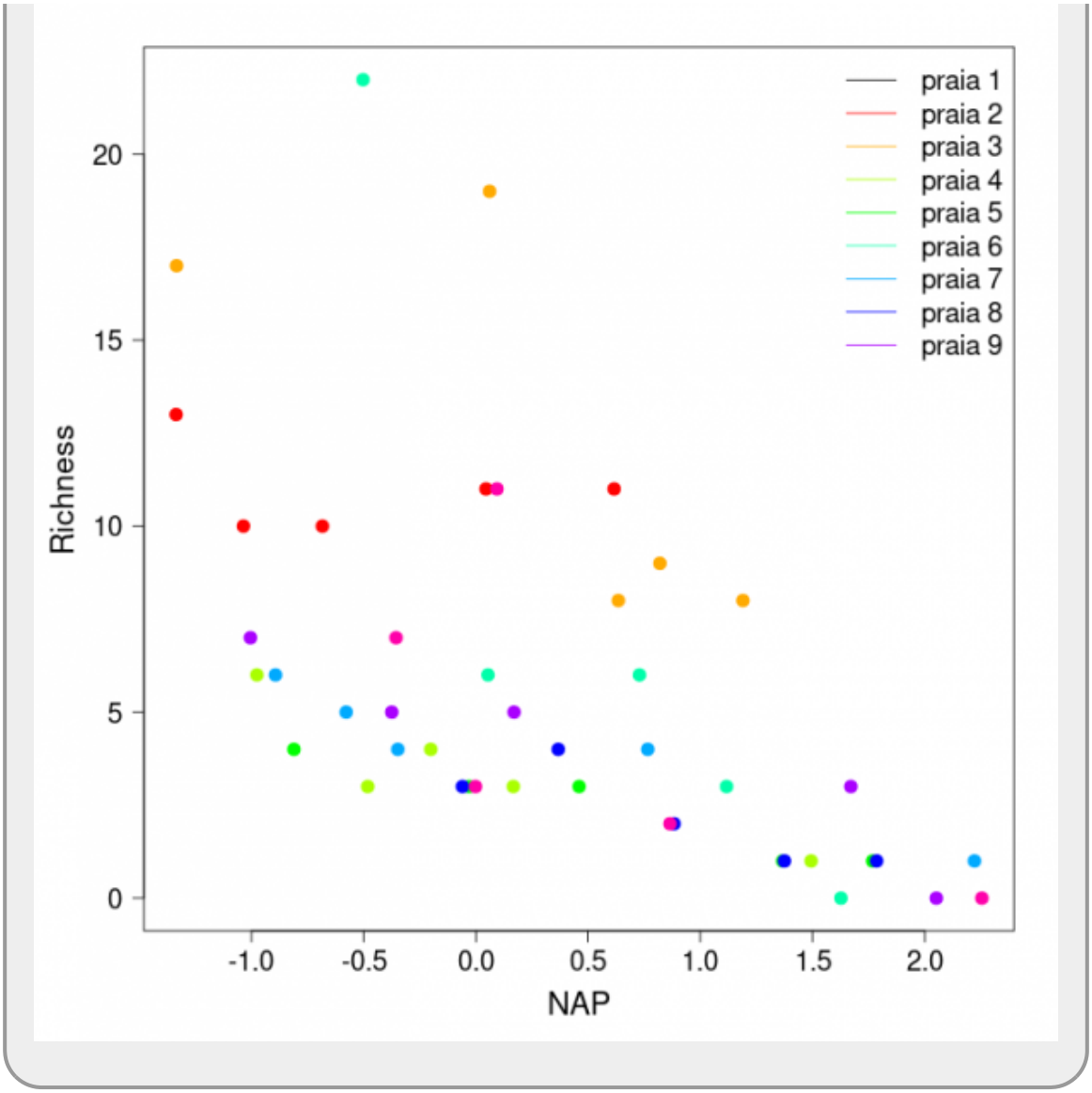


Voltando aos dados da riqueza de espécies das praias

Vamos retornar então ao exemplo do livro do Zuur et al. (2009), no qual queremos entender se a riqueza de espécies da macro-fauna em 9 praias na costa da Holanda é influenciado pela altura da estação de amostragem em relação à altura média da maré (NAP) [1)]. As unidades amostrais estão aninhadas dentro das praias, ou seja, temos 5 valores por praia que não podem ser consideradas independentes, e por isso incluímos no nosso modelo a praia como um efeito aleatório[2)].

Dados de contagem, ou seja, o número de espécies em cada amostra, são geralmente modelados com a distribuição de **Poisson**, que assume também que a média é igual à variância. Ou seja, se a média aumenta, a variância também.

No nosso exemplo, essa relação variância-média pode ser visualizada olhando o gráfico de dispersão dos dados:



Perceba que os pontos vão ficando menos dispersos à medida que os valores de riqueza diminuem (e os de `NAP` aumentam).

Modelando GLMMs: contagem

Lendo os dados no Rcmdr

Depois de ler os dados no Rcmdr precisamos modificar a variável `fExp` para nomear os níveis. Como fizemos no roteiro do **LMM**.

- baixe o arquivo

  vamos à praia

  ;
- no Rcmdr, importe o arquivo, que está formatado com campos separados por tabulação e decimal como ponto;
- atribua o nome de `praia` ao conjunto de dados;
- visualize o conjunto de dados para se familiarizar com as variáveis.

Para ler os dados e fazer as modificações no console do R use o código abaixo:

```
praia <- read.table("praia.txt", header = TRUE, sep = "\t" , as.is =
TRUE)
praia$fExposure <- factor(praia$fExp, levels = c("low", "high"))
head(praia) #observando as primeiras linhas de dados
str(praia) # vendo se a estrutura dos dados está ok!
```

Agora, vamos modificar o modelo do roteiro de LMM apenas em relação à distribuição de probabilidades dos resíduos do modelo. Operacionalmente isso é bastante simples, basta usar a função `glmer()` do pacote `lme4` com a mesma formulação do modelo usada anteriormente, explicitando a distribuição (`family = "poisson"`).

```
library(lme4)
mod.riq <- glmer(Richness ~ NAP +  (1 | Beach), data = dados,
                 family = "poisson")
```

Vamos inspecionar o modelo:

```
summary(mod.riq)
```

Predito pelo Modelo

Lembrando que, como num GLM, os valores dos coeficientes agora estão na escala da função de ligação, que para a distribuição de poisson é a função log.

Para fazer o gráfico de nosso modelo ajustado, com a reta média (predição) dos efeitos fixos e as retas do efeito aleatório, preditas para cada praia, precisamos primeiro dos valores dos coeficientes do preditor linear. A partir dele podemos estimar qualquer valor predito (esperança) com um certo NAP em uma dada praia. Se tivermos uma sequencia de valores de NAP, em um dado intervalo, podermos construir as curvas preditas e com isso representar o modelo. O código abaixo cria os seguintes objetos:

- `randcoef`: guarda os coeficientes da estrutura aleatória
- `fixcoef`: guarda os coeficientes da estrutura fixa do modelo
- `seqNap`: cria uma sequencia de 100 valores no intervalo de -1.5 a 2.5
- beachXX: estima o valore esperado para a praia XX para os valores de `seqNAP`

```
## coeficientes da estrutura aleatória
randcoef <- coef(mod.riq)$Beach
randcoef
## coeficientes da estrutura fixa do modelo
fixcoef  <- fixef(mod.riq)
fixcoef
```

```
## valores da preditora em uma sequencia
seqNAP <- seq(-1.5, 2.5,len=100)

## estimativas para cada praia
beach01 <- randcoef[1,1] + randcoef[1,2] * seqNAP
beach02 <- randcoef[2,1] + randcoef[2,2] * seqNAP
beach03 <- randcoef[3,1] + randcoef[3,2] * seqNAP
beach04 <- randcoef[4,1] + randcoef[4,2] * seqNAP
beach05 <- randcoef[5,1] + randcoef[5,2] * seqNAP
beach06 <- randcoef[6,1] + randcoef[6,2] * seqNAP
beach07 <- randcoef[7,1] + randcoef[7,2] * seqNAP
beach08 <- randcoef[8,1] + randcoef[8,2] * seqNAP
beach09 <- randcoef[9,1] + randcoef[9,2] * seqNAP
beachFix <- fixcoef[1] + fixcoef[2] * seqNAP
```

Predito na escala da observação

Os preditos calculados acima estão na escala do preditor linear, no caso o logaritmo da contagem de espécies. Para colocar os valores na escala de observação precisamos exponenciar. O código abaixo faz isso para cada praia e para o predito pelo estrutura fixa do modelo (a praia média).

```
beach01a <- exp(beach01)
beach02a <- exp(beach02)
beach03a <- exp(beach03)
beach04a <- exp(beach04)
beach05a <- exp(beach05)
beach06a <- exp(beach06)
beach07a <- exp(beach07)
beach08a <- exp(beach08)
beach09a <- exp(beach09)
beachFixa <- exp(beachFix)
```

Gráfico Base

Primeiro, vamos fazer o gráfico básico com os dados coletados. O código a seguir contem as seguintes linhas de códigos:

- `cores …`: cria um objeto com nove cores tiradas de uma paleta do arco-iris com a função `rainbow(…)`
- `par(…)`: função que modifica parâmetros da janela gráfica, no caso modificamos as bordas
- `plot(…)`: função que cria o gráfico
    - 1° linha: os dados
    - 2° e 3° linha: modificações de parâmetros do gráfico

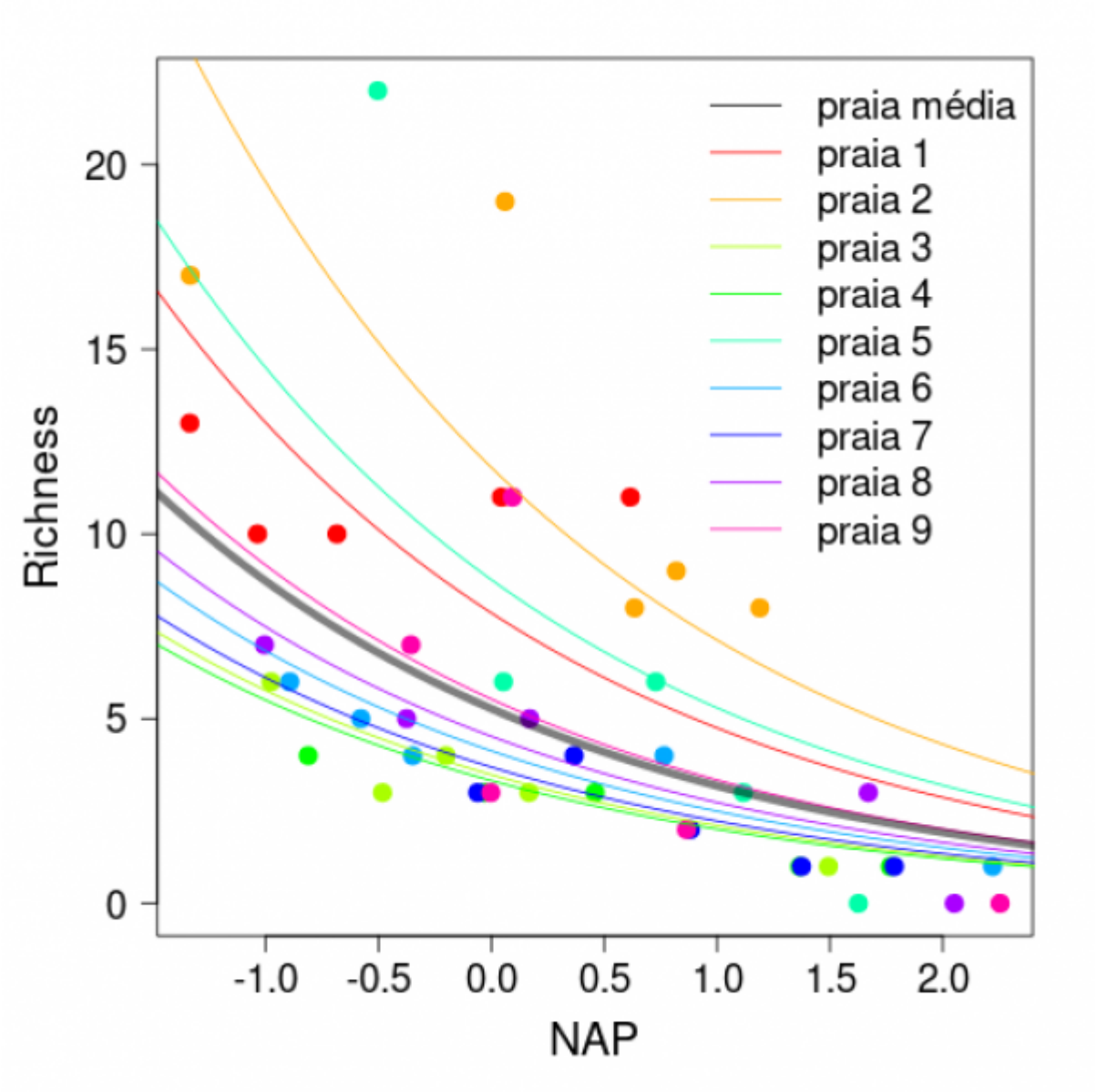
```
cores <- rainbow(9)
par( mar=c(5,5,2,2))
plot(Richness ~ NAP , data = dados,
```

```
     pch = 19, col = cores[as.factor(Beach)] , las = 1,
     cex=1.5, cex.lab= 1.7, cex.axis = 1.5
     )
```

Predito para cada Praia

Agora podemos juntar ao gráfico base as predições para cada praia e o predito pela estrutura fixa do modelo. As primeiras linhas abaixo incluem essas predições em cores diferentes e a última linha insere uma legenda no gráfico.

```
lines(beach01a ~ seqNAP, col = cores[1])
lines(beach02a ~ seqNAP, col = cores[2])
lines(beach03a ~ seqNAP, col = cores[3])
lines(beach04a ~ seqNAP, col = cores[4])
lines(beach05a ~ seqNAP, col = cores[5])
lines(beach06a ~ seqNAP, col = cores[6])
lines(beach07a ~ seqNAP, col = cores[7])
lines(beach08a ~ seqNAP, col = cores[8])
lines(beach09a ~ seqNAP, col = cores[9])
lines(beachFixa ~ seqNAP, col = rgb(0,0,0,0.5), lwd=5)
legend("topright", c( "praia média",paste("praia", 1:9)) , lty=1, col=
c(1,cores), bty="n", cex=1.5)
```



Podemos ver nessa figura a predição do nosso modelo em relação aos parâmetros fixos (curva em preto) e as predições para cada praia separadamente.

Compare este gráfico ao gráfico das predições no roteiro de LMM. Qual parece se ajustar melhor aos dados?

Já ajustamos nosso primeiro GLMM! Porém nossas análises têm mais uma variável preditora que poderia entrar no modelo. Então, o próximo passo é incorporar a variável fixa `Exposure` e fazer a escolha dos efeitos aleatórios antes de prosseguir com a seleção de modelos.

Efeitos aleatórios

Os passos para a escolha dos efeitos aleatórios no GLMM, quando temos mais de uma possibilidade de efeitos aleatórios, são iguais aos apresentados no roteiro de LMM.

- 1. Criar o modelo completo, com todos os efeitos fixos a serem utilizados. Aqui incluímos também a variável `fExposure` e a interação entre `fExposure` e NAP. Nosso modelo é:

*Richness = fExposure * NAP + "efeito(s) aleatório(s)"*

- 2. Incluir os efeitos aleatórios plausíveis. No nosso exemplo, escolhemos duas possibilidades de efeito aleatório, variações no intercepto entre praias (`(1|Beach)`) e a interação entre praia e NAP, resultando também na variação de inclinação entre praias (`(NAP|Beach)`). Além disso, colocamos para "competir" um modelo glm sem efeitos aleatórios.

```
m0 <- glm(Richness ~ fExposure*NAP, data = dados, family= poisson)
m1 <- glmer(Richness ~ fExposure * NAP + (1|Beach), data = dados,
            family = poisson)
m2 <- glmer(Richness ~ fExposure * NAP + (NAP|Beach), data = dados,
            family = poisson)
```

Um ponto importante, assim como nos LMM, é que estes modelos serão ajustados utilizando uma forma diferente de estimação, ao invés da máxima verossimilhança (ML), vamos usar a **máxima verossimilhança restrita** (REML). Isso acontece porque a ML é enviesada para as estimativas de variância do modelo e a REML corrige este enviesamento. Na prática, nós não precisamos fazer nada, pois a função `glmer` já usa, por padrão, a REML (argumento `REML = TRUE`).

- 3. Utilizar o comando `anova`, como temos feito para tomar a decisão sobre qual modelo aninhado reter, buscando a simplificação do modelo ao mínimo adequado. Aqui utilizamos o argumento `refit = FALSE` para garantir que a função irá manter a estimativa do REML.

```
anova(m1, m2, refit= FALSE)
```

Aqui o valor de p é relativamente baixo, indicando que os modelos apresentam diferenças marcantes, sendo assim, devemos reter o modelo mais complexo, ou seja com a estrutura de erro `(NAP|Beach)` .

Bom, agora sabemos que a melhor estrutura de efeito aleatório é a variação na relação de inclinação de NAP entre as praias. Assim, podemos prosseguir com a verificação dos efeitos fixos através de teste de hipóteses e/ou seleção de modelos.

Inferência por seleção de modelos pela tabela de ANOVA

Agora, vamos avaliar os modelos mistos quanto à sua estrutura fixa através da comparação de modelos por ANOVA, da mesma maneira que foi feito no roteiro de LMM.

Comparando modelo com interação entre `fExposure` (lembrando que estamos usando como variável categórica com 3 níveis) e NAP e sem interação para ver se a interação é importante:

```
# modelo com interação entre fExposure e NAP
m3 <- glmer(Richness ~ fExposure * NAP + (NAP|Beach),
            data = dados, family="poisson")

# modelo sem interação entre fExposure e NAP
m4 <- glmer(Richness ~ fExposure + NAP + (NAP|Beach),
            data = dados, family="poisson")
```

E aplicamos a função `anova` nos nossos modelos aninhados:

```
anova(m3,m4)
```

Como o resultado da comparação entre os modelos não foi significativo, ou seja, o modelo com interação não é significativamente diferente do modelo sem interação, nós ficamos então com o modelo sem interação e continuamos a compará-lo com modelos com apenas uma das variáveis preditoras por vez:

```
# removendo a variável NAP
m5 <- glmer(Richness ~ fExposure + (NAP|Beach),
            data = dados, family="poisson")
anova(m4, m5)
```

O resultado acima nos mostra que não parece razoável remover a variável NAP do modelo, pois há uma diferença significativa entre os modelos com e sem NAP.

Agora, vamos comparar novamente o modelo com as duas variáveis preditoras com um modelo sem a variável `fExposure`.

```
# removendo a variável `fExposure`
m6<- glmer(Richness ~ NAP + (NAP|Beach),
            data = dados, family="poisson")
anova(m4, m6)
```

Também não parece razoável remover `fExposure` e ficar apenas com NAP, então nosso modelo final contém apenas o efeito aditivo das duas variáveis. Para finalizar, vamos comparar este modelo com o nulo.

```
# modelo nulo
m7 <- glmer(Richness ~ 1 + (NAP|Beach), data = dados, family="poisson")

anova(m4,m7)
```

Então podemos concluir que tanto `fExposure` quanto NAP influenciam na riqueza de espécies, mas de maneira aditiva.

Vamos observar o *output* deste modelo:

```
summary(m4)
```

Diagnósticos dos modelos

A forma de se fazer o diagnóstico de um GLMM é ligeiramente diferente de um LMM, isso porque não esperamos normalidade nem homocedasticidade dos dados, e mesmo de um GLM, já que resíduos escalonados como o resíduo deviance e de Pearson podem não informar bem se o modelo está mal especificado. Sugerimos fortemente olhar esta breve apresentação do pacote `DHARMa` que explica rapidamente os problemas de usar técnicas de diagnóstico usuais em LMM e GLM para um GLMM e sugere uma solução elegante para o problema.

Nós vamos seguir o diagnóstico proposto por Florian Hartig usando o pacote `DHARMa`. Essa

abordagem é baseada em simulações para criar resíduos escalonados, com valores entre 0 e 1, interpretáveis de um modelo GLMM. Aqui, os detalhes estatísticos e teóricos da forma de realizar estas simulações e criar os resíduos escalonados vai ficar de lado (mas está nas referências do pacote `DHARMa`) para que possamos focar em como olhar os gráficos produzidos e interpretá-los.

Criando os resíduos

Os resíduos escalonados (também chamados de resíduos quantílicos) são calculados com a função `simulateResiduals()`, na qual indicamos o nosso modelo GLMM e o número de simulações. Para o número de simulações, o padrão da função é 250 pois pode demora muito, mas para uma análise concreta faça pelo menos 1000. Como os nossos dados são poucos e o modelo simples, vamos utilizar 1000 (mas se se computador demorar, pare o processo e faça com menos simulações).

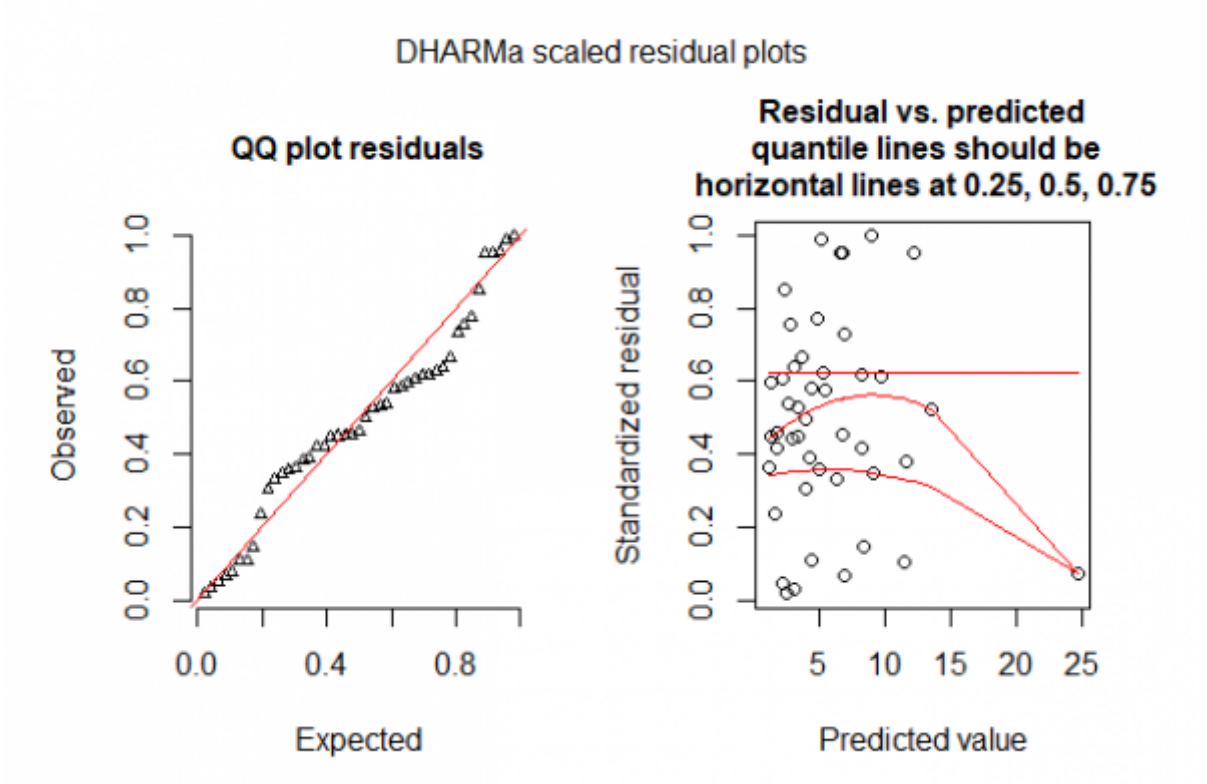
```
#install.packages("DHARMa") #não esquecer de instalar o pacote
library(DHARMa)

residuo <- simulateResiduals(fittedModel = m4, n = 1000)
```

Plotando os resíduos

Vamos fazer a inspeção visual dos nossos resíduos:

```
plotSimulatedResiduals(residuo)
```



À esquerda, temos um gráfico de quantil-quantil, que nos ajuda a ver se os nossos resíduos estão de

acordo com o que é esperado dado o nosso modelo. Portanto, nossos pontos deveriam estar todos alinhados em cima da linha vermelha. Se não estiverem, aí temos problemas! A forma desse não-alinhamento nos indica quais são os possíveis problemas de especificação do modelo. No nosso modelo, não parece haver desvios muito fortes dos valores esperados.

À direita, temos um gráfico dos resíduos contra os valores preditos pelo modelo. Este gráfico nos ajuda a detectar desvios da uniformidade na direção y. As linhas vermelhas são regressões quantílicas que mostram os quantis 0.25, 0.50, 0.75. Estas linhas deveriam ser retas horizontais para cada quantil. **Note** que alguns desvios desta reta são esperados ao acaso, mesmo para um modelo perfeito, especialmente quando o tamanho amostral é pequeno. No nosso caso, temos um valor predito muito alto que puxa as retas vermelhas pra baixo. Vamos avaliar melhor a uniformidade dos resíduos a seguir com o teste de uniformidade.

Predito pelo Modelo Selecionado

Finalmente, vamos plotar um gráfico com os dados e as curvas de predição para o modelo que foi selecionado - efeito aditivo de `Exposure` e `NAP`. Vamos seguir os passos anteriores para calcular os valores preditos para cada praia e para a estrutura fixa do modelo. Note que agora temos duas curvas para cada praia, representando as exposições alta e baixa.

```
rand04 <- coef(m04)$Beach
rand04
fix04<- fixef(m04)
fix04
beach01L <- exp(rand04[1,1] + rand04[1,3] * seqNAP)
beach01H <- exp(rand04[1,1] +  rand04[1,2]  + rand04[1,3] * seqNAP)
beach02L <- exp(rand04[2,1] + rand04[2,3] * seqNAP)
beach02H <- exp(rand04[2,1] +  rand04[2,2]  + rand04[2,3] * seqNAP)
beach03L <- exp(rand04[3,1] + rand04[3,3] * seqNAP)
beach03H <- exp(rand04[3,1] +  rand04[3,2]  + rand04[3,3] * seqNAP)
beach04L <- exp(rand04[4,1] + rand04[4,3] * seqNAP)
beach04H <- exp(rand04[4,1] +  rand04[4,2]  + rand04[4,3] * seqNAP)
beach05L <- exp(rand04[5,1] + rand04[5,3] * seqNAP)
beach05H <- exp(rand04[5,1] +  rand04[5,2]  + rand04[5,3] * seqNAP)
beach06L <- exp(rand04[6,1] + rand04[6,3] * seqNAP)
beach06H <- exp(rand04[6,1] +  rand04[6,2]  + rand04[6,3] * seqNAP)
beach07L <- exp(rand04[7,1] + rand04[7,3] * seqNAP)
beach07H <- exp(rand04[7,1] +  rand04[7,2]  + rand04[7,3] * seqNAP)
beach08L <- exp(rand04[8,1] + rand04[8,3] * seqNAP)
beach08H <- exp(rand04[8,1] +  rand04[8,2]  + rand04[8,3] * seqNAP)
beach09L <- exp(rand04[9,1] + rand04[9,3] * seqNAP)
beach09H <- exp(rand04[9,1] +  rand04[9,2]  + rand04[9,3] * seqNAP)
beachFixL <- exp(fix04[1] + fix04[3] * seqNAP)
beachFixH <- exp(fix04[1] + fix04[2]  + fix04[3] * seqNAP)
```

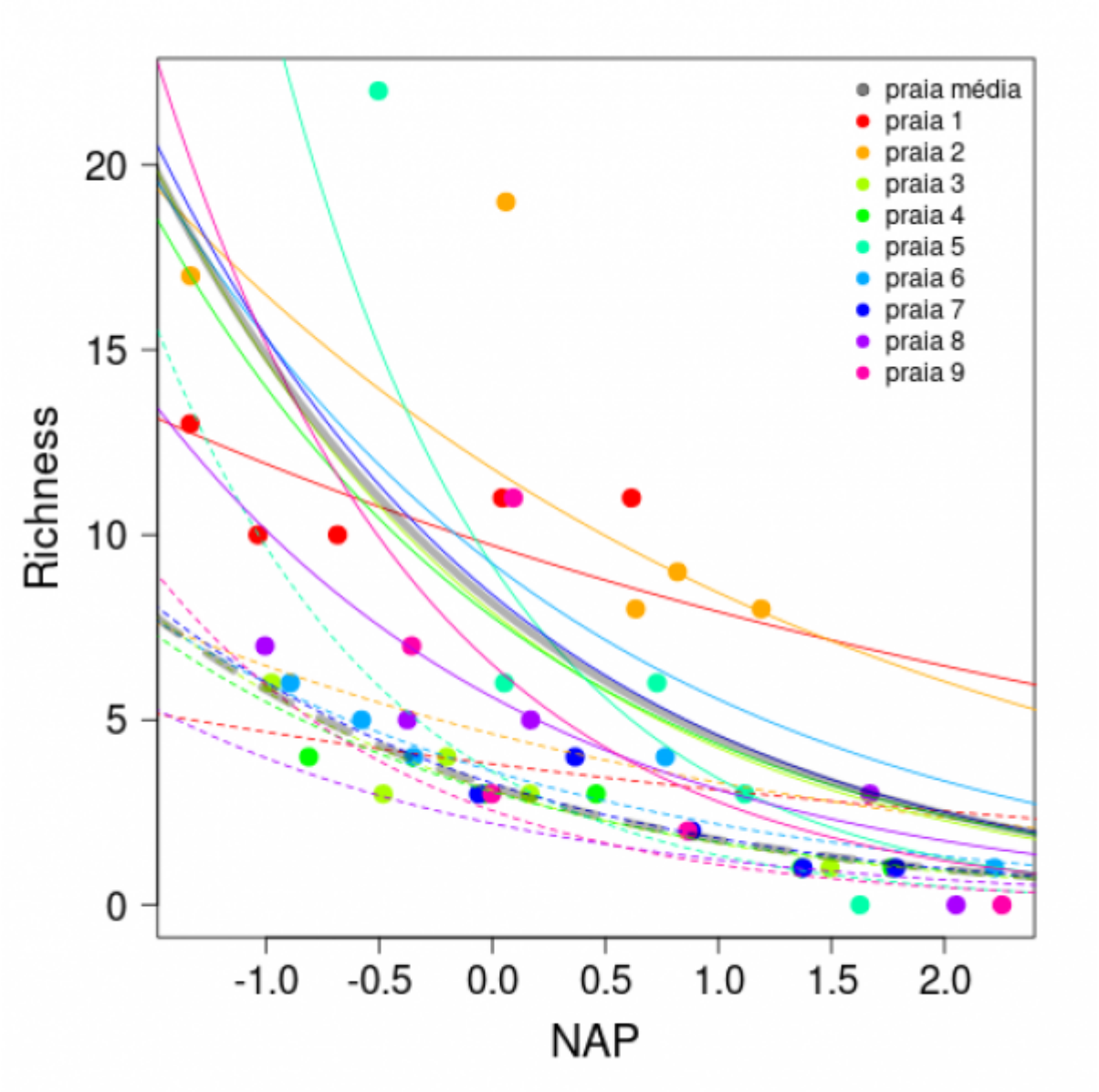
Gráfico do Modelo

Para fazer o gráfico, seguimos os mesmos passos do código do gráfico anterior, com a diferença que

agora a exposição alta é representada por uma linha tracejada(`lty = 2`).

```
par( mar=c(5,5,2,2))
plot(Richness ~ NAP ,data = dados,
     pch = 19, col = cores[as.factor(Beach)] , las = 1,
     cex=1.5, cex.lab= 1.7, cex.axis = 1.5
     )
lines(beach01L ~ seqNAP, col= cores[1], lty=1)
lines(beach01H ~ seqNAP, col= cores[1], lty=2)
lines(beach02L ~ seqNAP, col= cores[2], lty=1)
lines(beach02H ~ seqNAP, col= cores[2], lty=2)
lines(beach03L ~ seqNAP, col= cores[3], lty=1)
lines(beach03H ~ seqNAP, col= cores[3], lty=2)
lines(beach04L ~ seqNAP, col= cores[4], lty=1)
lines(beach04H ~ seqNAP, col= cores[4], lty=2)
lines(beach05L ~ seqNAP, col= cores[5], lty=1)
lines(beach05H ~ seqNAP, col= cores[5], lty=2)
lines(beach06L ~ seqNAP, col= cores[6], lty=1)
lines(beach06H ~ seqNAP, col= cores[6], lty=2)
lines(beach07L ~ seqNAP, col= cores[7], lty=1)
lines(beach07H ~ seqNAP, col= cores[7], lty=2)
lines(beach08L ~ seqNAP, col= cores[8], lty=1)
lines(beach08H ~ seqNAP, col= cores[8], lty=2)
lines(beach09L ~ seqNAP, col= cores[9], lty=1)
lines(beach09H ~ seqNAP, col= cores[9], lty=2)
lines(beachFixL ~ seqNAP, col= rgb(0,0,0,0.3), lty=1, lwd=5)
lines(beachFixH ~ seqNAP, col= rgb(0,0,0,0.3), lty=2, lwd=5)

legend("topright", c( "praia média",paste("praia", 1:9)) , pch =19,  col=
c(rgb(0,0,0,0.5),cores), bty="n", cex=1)
```



Referências recomendadas

Bates, et al. 2014..**Fitting linear mixed-effects models using lme4**. arXiv preprint arXiv:1406.5823. (publicação do pacote `lme4`)

Florian Hartig (2016). DHARMa: Residual Diagnostics for Hierarchical (Multi-Level/Mixed) Regression Models. R package. version 0.1.2. https://github.com/florianhartig/DHARMa

Zuur, A., Ieno, E., Walker, N., Saveliev, A. & Smith, G. 2009. Mixed effects models and extensions in ecology with R. (Livro muito bom e completo sobre modelos mistos e aditivos)

1)

vamos começar com o modelo mais simples

2)

veja o roteiro Modelos Lineares Mistos

From:
http://labtrop.ib.usp.br/ - **Laboratório de Ecologia de Florestas Tropicais**

Permanent link:
http://labtrop.ib.usp.br/doku.php?id=cursos:planeco:roteiro:12-glmm_rcmdr

Last update: **2021/04/30 15:48**