

Reunião Orientação 23mar2017

Danilo Pereira Mori

16 de março de 2017

META

Construção do modelo de trabalho para a variável GOF

Ordem:

- i) realizar a transformação Z (ou score Z) nas variáveis preditoras do modelo (p e kernel) = (observado - média)/desvio padrão;
- ii) seleção da função de ligação (protocolo de Zuur et al. 2009);
- iii) Seleção/confirmação da estrutura aleatória (idem);
- iv) Avaliação do modelo (protocolo de Bolker et al. 2008);

Seleção do modelo cheio - protocolo de Zuur et al. 2009 (pag 121-122)

Para iniciar a modelagem vou utilizar a distribuição binomial e vou selecionar qual a melhor função de ligação utilizando como critério AIC.

Janela de Código 2 Seleção da função de ligação

```
l_md1 <- vector("list", length = 3)
names(l_md1) <- c("logit", "probit", "cloglog")
l_md1[[1]] <- glm(cbind(GOF, 100 - GOF) ~ p.z * kernel.z, family = "binomial",
  df_temp)
l_md1[[2]] <- glm(cbind(GOF, 100 - GOF) ~ p.z * kernel.z, family = binomial(link = probit),
  df_temp)
l_md1[[3]] <- glm(cbind(GOF, 100 - GOF) ~ p.z * kernel.z, family = binomial(link = cloglog),
  df_temp)
AICctab(l_md1, weights = TRUE)
```

```
##          dAICc df weight
## logit    0.0  4  0.987
## probit   8.7  4  0.013
## cloglog 20.7  4  <0.001
```

A função de ligação mais plausível é a logito. Agora vou selecionar a estrutura aleatória. Consideramos duas estruturas aleatórias: i) (1 | Site); ii) (kernel | Site).

Janela de Código 3 Comparação da estrutura aleatória

```
l_md2 <- vector("list", length = 4)
names(l_md2) <- c("(1|Site)", "(kernel|Site)", "(1 | kernel)", "(p.z | kernel)")
l_md2[[1]] <- glmer(cbind(GOF, 100 - GOF) ~ p.z * kernel.z + (1 | Site), family = binomial,
  data = df_temp)
l_md2[[2]] <- glmer(cbind(GOF, 100 - GOF) ~ p.z * kernel.z + (kernel.z | Site),
  family = binomial, data = df_temp)
l_md2[[3]] <- glmer(cbind(GOF, 100 - GOF) ~ p.z + (1 | kernel.z), family = binomial,
  data = df_temp)
l_md2[[4]] <- glmer(cbind(GOF, 100 - GOF) ~ p.z + (p.z | kernel.z), family = binomial,
```

```
data = df_temp)
AICtab(l_md2, weights = TRUE)
```

```
##              dAIC    df weight
## (kernel|Site)    0.0  7    1
## (1|Site)        3552.4  5 <0.001
## (p.z | kernel) 20655.1  5 <0.001
## (1 | kernel)    21215.2  3 <0.001
```

A estrutura aleatória selecionada considera interação entre kernel e sítio

Janela de Código 4 Comparação da estrutura fixa

```
l_md3 <- vector("list", length = 4)
names(l_md3) <- c("p * kernel", "p + kernel", "p", "1")
l_md3[[1]] <- glmer(cbind(GOF,100-GOF) ~ p.z * kernel.z + (kernel.z | Site),
                    family = binomial, data = df_temp)
l_md3[[2]] <- glmer(cbind(GOF,100-GOF) ~ p.z + kernel.z + (kernel.z | Site),
                    family = binomial, data = df_temp)
l_md3[[3]] <- glmer(cbind(GOF,100-GOF) ~ p.z + (kernel.z | Site), # precisar confirmar que isso faz sen
                    family = binomial, data = df_temp)
l_md3[[4]] <- glmer(cbind(GOF,100-GOF) ~ 1 + (1 | Site),
                    family = binomial, data = df_temp)
AICctab(l_md3, weights = TRUE)
```

```
##              dAICc  df weight
## p * kernel    0.0  7  0.64
## p + kernel    2.4  6  0.20
## p             2.7  5  0.16
## 1            4381.7  2 <0.001
```

```
RVAideMemoire::plotresid(l_md3[[1]])
```

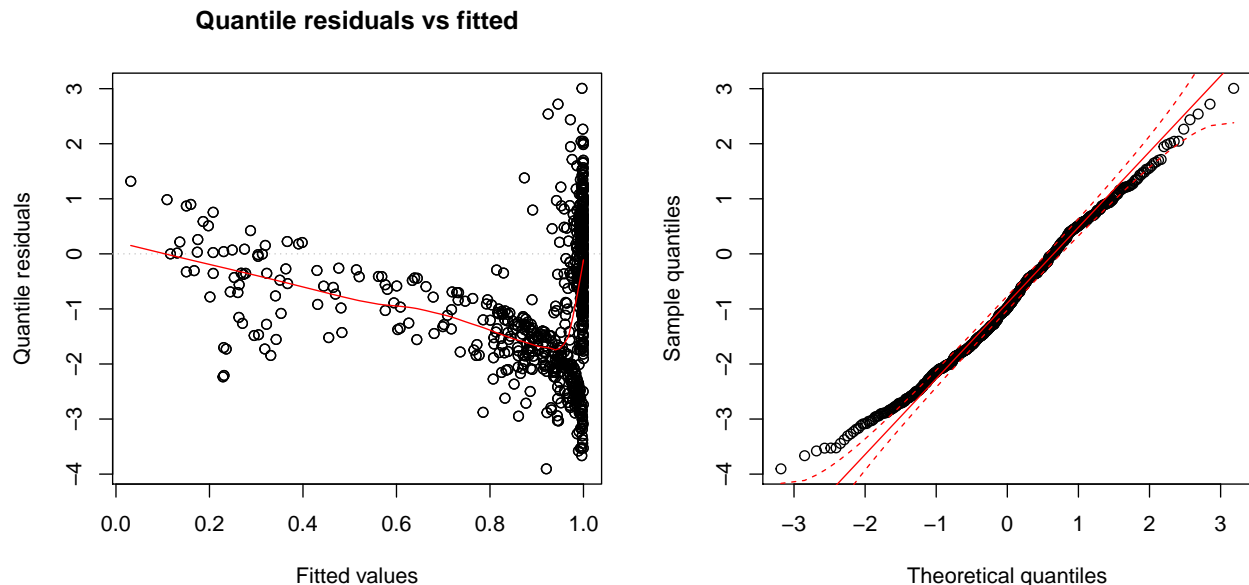


Figura 1 Gráficos diagnóstico do modelo selecionado $\text{cbind}(\text{GOF}, 100 - \text{GOF}) \sim p.z * \text{kernel.z} + (\text{kernel.z} | \text{Site})$

A distribuição dos resíduos é claramente não homogênea ao longo dos valores ajustados (figura 1, primeiro gráfico). A distribuição teórica parece descrever bem os dados, com notória exceção à cauda (figura 1, segundo

gráfico).

```
l_md3[[1]]
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: cbind(GOF, 100 - GOF) ~ p.z * kernel.z + (kernel.z | Site)
## Data: df_temp
##      AIC      BIC    logLik deviance df.resid
## 4771.869 4803.605 -2378.934  4757.869      681
## Random effects:
## Groups Name      Std.Dev. Corr
## Site (Intercept) 2.618
##      kernel.z    1.209    -0.15
## Number of obs: 688, groups: Site, 86
## Fixed Effects:
## (Intercept)      p.z      kernel.z p.z:kernel.z
##      4.13364    -0.04439    -0.22830      0.30015
```

Avaliação do modelo cheio - protocolo de Bolker et al. 2008 (box 4)

1)“Specify fixed and random effects; only important interactions”

$GOF \sim p * kernel + (kernel | site)$

2)“Choose an error distribution and link function”

Binomial com logit

3)“Graphical checking: are variances of data (transformed by the link function) homogeneous across categories? Are responses of transformed data linear with respect to continuous predictors? Are there outlier individuals or groups? Do distributions within groups match the assumed distribution?”

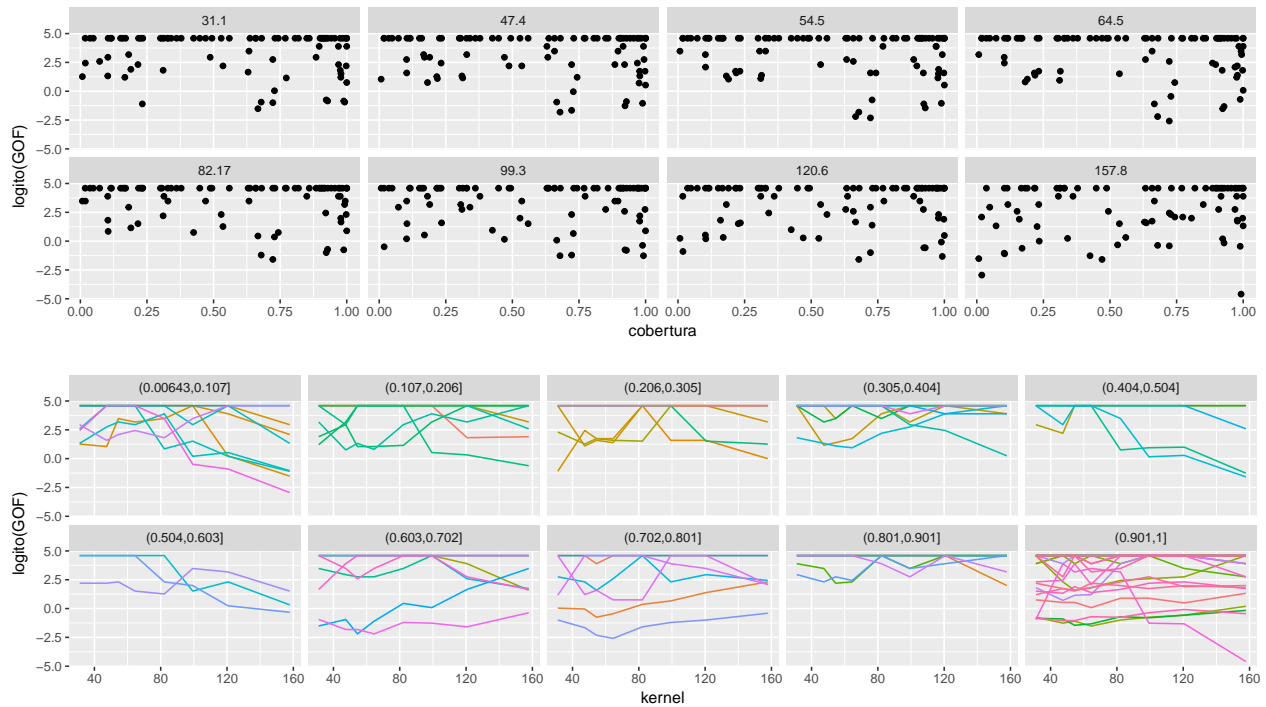


Figura 2. Estrutura aleatória do modelo por classes de cobertura vegetal. No eixo x as diferentes classes de kernel, no eixo y o logito do número de réplicas/simulação que apresentam bons ajustes segundo o teste de Kolmogorov-Smirnov com p valor > 0.05.

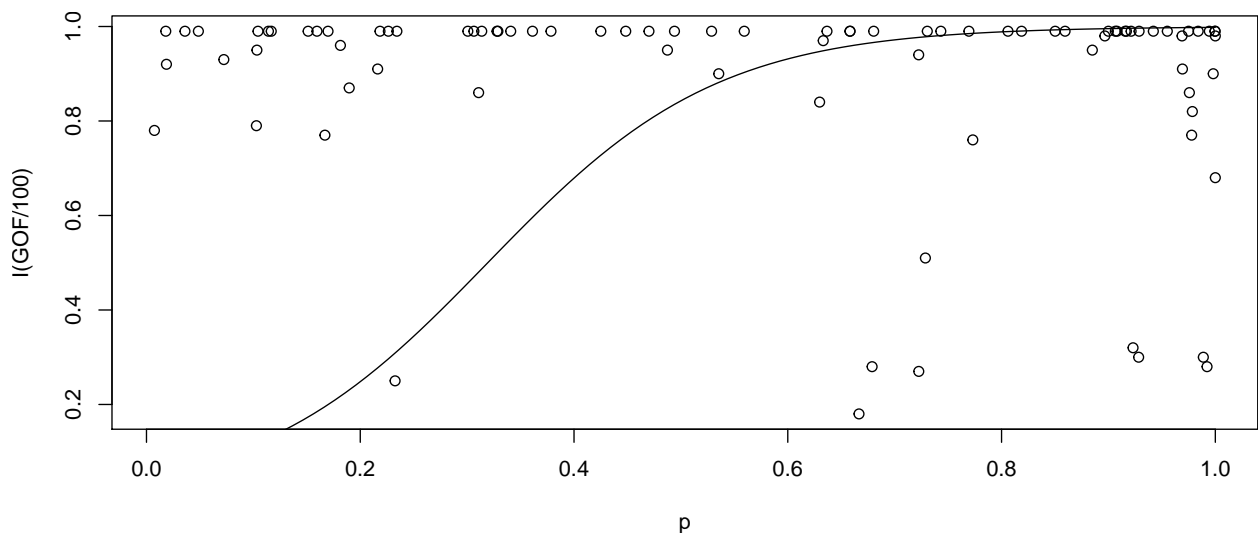
```

coefs <- fixef(l_md3[[1]])

fun_logit <- function(x, a0, a1) {
  exp(a0 + a1 * x)/(1 + exp(a0 + a1 * x))
}

plot(I(GOF/100) ~ p, df_temp1, subset = kernel == 31.1)
curve(fun_logit(x, a0 = coefs[1] + coefs[3] * 31.1, a1 = coefs[2] + coefs[4] *
  31.1), add = T)

```

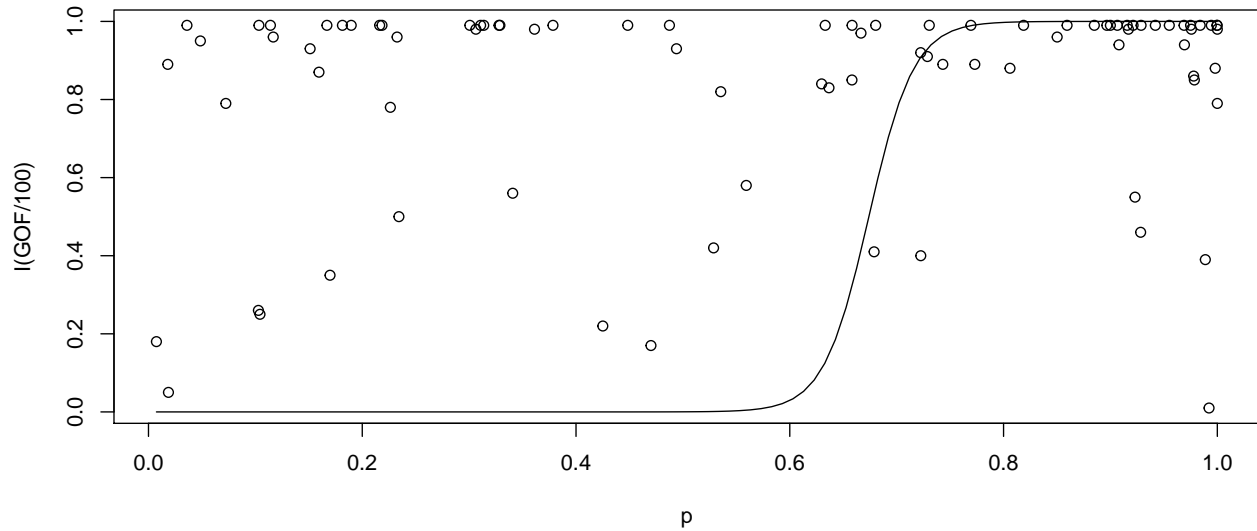


```

plot(I(GOF/100) ~ p, df_temp1, subset = kernel == 157.8)
curve(fun_logit(x, a0 = coefs[1] + coefs[3] * 157.8, a1 = coefs[2] + coefs[4] *

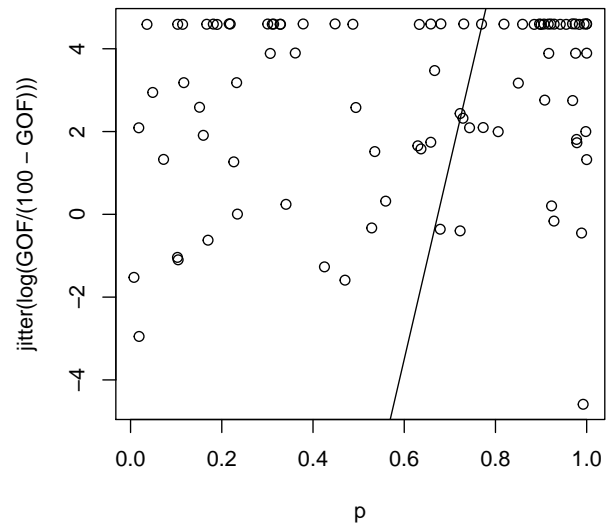
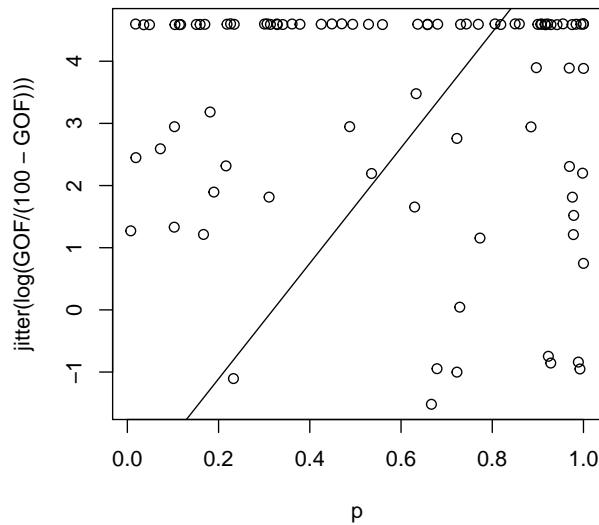
```

```
157.8), add = T)
```



```
par(mfrow = c(1, 2))
plot(jitter(log(GOF/(100 - GOF))) ~ p, df_temp1, subset = kernel == 31.1)
abline(coefs[1] + coefs[3] * 31.1, coefs[2] + coefs[4] * 31.1)

plot(jitter(log(GOF/(100 - GOF))) ~ p, df_temp1, subset = kernel == 157.8)
abline(coefs[1] + coefs[3] * 157.8, coefs[2] + coefs[4] * 157.8)
```



Distribuição dos parâmetros do modelo: avaliando a distribuição dos parâmetros da estrutura aleatória

-> Distribuição dos parâmetros do modelo $GOF \sim \text{kernel} * \text{site}$

Janela de Código 5 Seleção do modelo para avaliar a distribuição dos parâmetros do modelo $GOF \sim \text{kernel} + (1|\text{Site})$

```
l_md <- vector("list", length = 3)
names(l_md) <- c("logit", "probit", "cloglog")
l_md[[1]] <- glm(cbind(GOF, 100 - GOF) ~ kernel.z * Site, family = binomial(link = probit),
```

```

df_temp)
l_md[[2]] <- glm(cbind(GOF, 100 - GOF) ~ kernel.z * Site, family = binomial(link = probit),
df_temp)
l_md[[3]] <- glm(cbind(GOF, 100 - GOF) ~ kernel.z * Site, family = binomial(link = cloglog),
df_temp)
AICctab(l_md, weights = TRUE)

```

```

##          dAICc df weight
## logit      0.0 172 0.5
## probit      0.0 172 0.5
## cloglog    58.6 172 <0.001

```

```

### (~Site) ### l_md <- vector('list', length = 3) names(l_md) <-
### c('logit','probit','cloglog') l_md[[1]] <- glmer(cbind(GOF,100-GOF) ~
### kernel.z + (1 | Site), family = 'binomial', data = df_temp) l_md[[2]] <-
### glmer(cbind(GOF,100-GOF) ~ kernel.z + (1 | Site), family =
### 'binomial'(link=probit), data = df_temp) l_md[[3]] <-
### glmer(cbind(GOF,100-GOF) ~ kernel.z + (1 | Site), family =
### 'binomial'(link=cloglog), data = df_temp) AICctab(l_md, weights = TRUE) (~
### kernel) ### l_md <- vector('list', length = 3) names(l_md) <-
### c('logit','probit','cloglog') l_md[[1]] <- glmer(cbind(GOF,100-GOF) ~ Site
### + (1 | kernel.z), family = 'binomial', data = df_temp) l_md[[2]] <-
### glmer(cbind(GOF,100-GOF) ~ Site + (1 | kernel.z), family =
### 'binomial'(link=probit), data = df_temp) l_md[[3]] <-
### glmer(cbind(GOF,100-GOF) ~ Site + (1 | kernel.z), family =
### 'binomial'(link=cloglog), data = df_temp) AICctab(l_md, weights = TRUE)

```

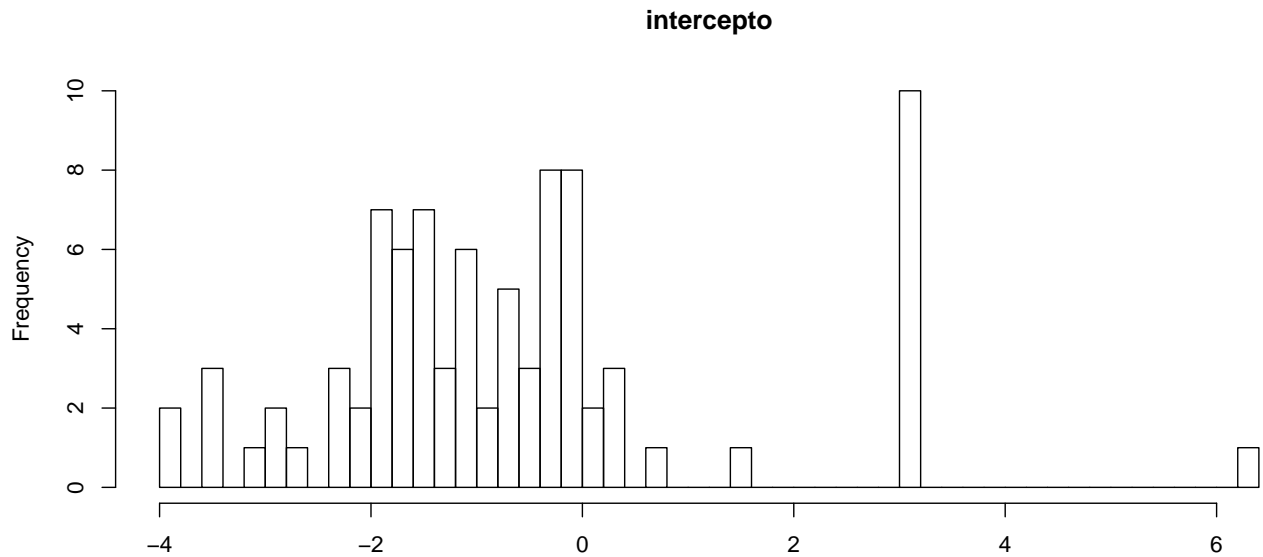
- Houveram problemas de convergência em todos os modelos: i) glm.fit: probabilidades ajustadas numericamente 0 ou 1 ocorreu e ii) glm.fit: algoritmo não convergiu glm.fit: probabilidades ajustadas numericamente 0 ou 1 ocorreu.
- Como a logit e a probit tiveram igual AIC vou utilizar a logito para avaliar a distribuição dos parâmetros.
- em código os gráficos diagnósticos do modelo 1 - não houve bom ajuste

Janela de Código 6 Distribuição dos parâmetros do modelo `glm(cbind(GOF,100-GOF) ~ kernel.z * Site, family = "binomial"(link=probit),df_temp)`

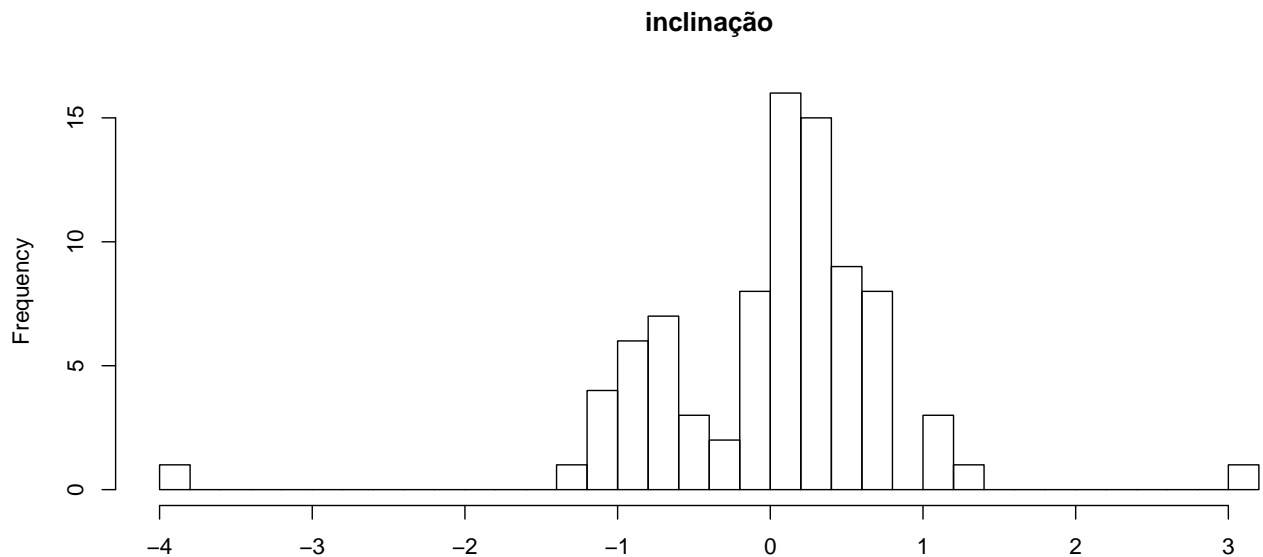
```

# primeiro uso do broom #
tidy(l_md[[1]])[1:87, 2] %>% hist(., breaks = 40, main = "intercepto")

```



```
tidy(l_md[[1]])[88:172, 2] %>% hist(., breaks = 40, main = "inclinação")
```



4)“Fit fixed-effect GLMs both to the full (pooled) data set and within each level of the random factors [28,50]. Estimated parameters should be approximately normally distributed across groups (group-level parameters can have large uncertainties, especially for groups with small sample sizes). Adjust model as necessary (e.g. change link function or add covariates).”

5)“Fit the full GLMM.”

6)“Recheck assumptions for the final model (as in step 3) and check that parameter estimates and confidence intervals are reasonable (gigantic confidence intervals could indicate fitting problems). The magnitude of the standardized residuals should be independent of the fitted values. Assess overdispersion (the sum of the squared Pearson residuals should be χ^2 distributed [66,67]). If necessary, change distributions or estimate a scale parameter. Check that a full model that includes dropped random effects with small standard deviations gives similar results to the final model. If different models lead to substantially different parameter esti-

mates, consider model averaging.”