# geral de análises estatísticas

Prof. Dra. Juliana Hipólito

15/07/2019

## tipos de variáveis

## discretas x contínuas

**Variável aleatória discreta: número ou a quantidade observada na unidade experimental ou tentativa.**

**- Representada por números inteiros (0, 1, 2, 3, 4...);**

**- Não pode conter números negativos;**

**- Número finito de possibilidades;**

**- Podemos achar a probabilidade de cada evento.**

**Variável aleatória contínua: usualmente medidas contínuas como peso, altura, distância, pH, biomassa, etc.**

**Representada por números não inteiros (1,3; - 1,54; - 1,7); Pode conter números negativos;**

**Número infinito de possibilidades; Probabilidade de cada evento é zero.**

## Distribuição binomial —-

## É a distribuição de probabilidade discreta do número de sucessos em uma sequência de n tentativas tal que: i) as tentativas são independentes; ii) cada tentativa resulta apenas em duas possibilidades, sucesso ou fracasso; e iii) a probabilidade de cada tentativa, p, permanece constante.

help(Binomial)

# Problema: Há uma probabilidade de 0,30 de um girino, ao forragear em um corpo d'água, ser predado por uma larva de odonata. Determine as probabilidades de que, dentre seis girinos que estão forrageando no corpo d'água, 0, 1, 2, 3, 5 ou 6 sejam predados.

# # # # #dbinom(x, size, prob)

dbinom (2, size = 6, prob = 0.3)

#para descobrirmos qual a probabilidade de dois visitantes chegarem a flores

dbinom (3, size = 6, prob = 0.3)

pbinom (2, size = 6, prob = 0.3) #probabilidade acumulativa: probabilidade de dois ou menos girinos (0, 1) serem predados, precisamos digitar o seguinte comando

pbinom (5, size = 6, prob = 0.3) qbinom(p, size, prob) #Inverso da função de probabilidade acumulativa - Um exemplo contrário ao comando anterior é utilizado quando um valor de probabilidade é fornecido e o programa retorna o valor de X associado a ele. Para isso utiliza-se o seguinte comando:

qbinom(0.74, size = 6, prob = 0.3) #Qual o valor de X (número de girinos predados) associado à probabilidade de 0,74

plot(pbinom(seq(0,6, by =1), size = 6, prob = 0.3),type ="h", xlab = "Número de girinos predados", ylab = "Probabilidade", main = "Função de probabilidade acumulada")

## Distribuição de Poisson —-

**Na teoria da probabilidade e na estatística, a distribuição de Poisson é uma distribuição de probabilidade discreta. Expressa a probabilidade de uma série de eventos ocorrem em um período fixo de tempo, área, volume, quadrante, etc. Esta distribuição segue as mesmas premissas da distribuição binomial: i) as tentativas são independentes; ii) a variável aleatória é o número de eventos em cada amostra; e iii) a probabilidade é constante em cada intervalo.**

**Suponha que um pesquisador registrou o número de visitas à flor de uma planta durante um período de 15 minutos. O número médio de borboletas que visitam no período de 15 minutos é 10 (λ). Determine a probabilidade de que cinco borboletas visitem a flor em 15 minutos. A probabilidade de uma borboleta visitar é a mesma para quaisquer dois períodos de tempo de igual comprimento.**

```
help(Poisson)

#dpois(x, lambda)

dpois (5, lambda = 10) #probabilidade de que cinco borboletas visitem uma flor

dpois (8, lambda = 10)

plot(ppois(seq(1,10, by =1), lambda = 10),type =“h”, xlab = “Número visitas”, ylab = “Probabilidade”, main = “Função de probabilidade acumulada”)
```

**Normal —-**

**A distribuição normal é uma das mais importantes distribuições com probabilidades contínuas. Conhecida também como Distribuição de Gauss ou Gaussiana. Esta distribuição é inteiramente descrita por parâmetros de média (μ) e desvio padrão (σ), ou seja, conhecendo-se estes parâmetros consegue-se determinar qualquer probabilidade em uma distribuição Normal.**

**Análises —-**

**Podemos usar modelos lineares generalizados (GLM) quando a variância não é constante, e/ou quando os erros não são normalmente distribuídos. Muitos tipos de dados têm erros não normais. No passado, as únicas maneiras capazes de lidar com esse problema eram a transformação da variável resposta ou a adoção de métodos não paramétricos. Em GLM, assumimos que cada resultado da variável dependente Y seja gerado a partir de uma variedade de diferentes tipos de distribuições que lidam com esse problema:**

**Poisson – úteis para dados de contagem**

**Binomial – úteis para dados com proporções**

**Gamma – úteis para dados mostrando um coeficiente constante de variância #Exponencial – úteis com dados de análises de sobrevivência**

**Os passos finais do processo de modelagem são constituídos pela estimativa dos parâmetros a partir dos dados e teste dos modelos uns contra os outros. Estimar os parâmetros dos modelos significa achar os parâmetros que fazem o modelo se ajustar melhor aos dados coletados. Nosso goodness-of-fit será baseado na probabilidade (likelihood) - a probabilidade de se encontrar nossos dados dado um modelo particular. Queremos a estimativa da máxima verossimilhança (maximum likelihood estimate) dos parâmetros – aqueles valores dos parâmetros que fazem os dados observados mais prováveis de terem acontecido. Uma vez que as observações são independentes, a junção das probabilidades dos dados totais é o produto das probabilidades de cada observação individual. Por conveniência matemática, sempre maximizamos o logaritimo das probabilidades (log-likelihood) ao invés da probabilidade direto.**

**Likelihood Ratio Test**

**Os modelos GLM são ajustados aos dados pelo método de máxima verossimilhança, proporcionando não apenas estimativas dos coeficientes de regressão, mas também estimando erros padrões dos coeficientes.**

**Akaike Information Criterion (AIC) - Critério de Informação de Akaike —-**

**O critério de Akaike é uma ferramenta para seleção de modelos, pois oferece uma medida relativa do goodness-of-fit (qualidade do ajuste) de um modelo estatístico. AIC não fornece um teste de um modelo no sentido usual de testar uma hipótese nula, ou seja, ele não pode dizer nada sobre o quão bem o modelo ajusta os dados em um sentido absoluto.**

**Dado um conjunto de modelos candidatos, o modelo preferido é aquele com o valor mínimo de AIC. O valor de AIC não só recompensa goodness- of-fit, mas inclui também uma penalização que é uma função crescente do número de parâmetros estimados. Esta penalidade desencoraja overfitting (aumentando o número de parâmetros livres no modelo melhora a qualidade do ajuste, independentemente do número de parâmetros livres no processo de geração de dados).**

**AIC x AICc**

**Dez orientações para Seleção de Modelo:**

**1) Cada modelo deve representar uma hipótese (interessante) específica a ser testada.**

**2) Mantenha os sub-grupos de modelos candidatos curtos. É desaconselhável considerar tantos modelos quanto o número de dados que você tem.**

**3) Verificar a adequação do modelo: use o seu modelo global (modelo mais complexo) ou modelos subglobais para determinar se as hipóteses são válidas. Se nenhum dos modelos se ajustar aos dados, critérios de informação indicarão apenas o mais parcimonioso dos modelos mais pobres.**

**4) Evitar a dragagem de dados (e.g., procura de padrões após uma rodada inicial de análise).**

**5) Evite modelos overfitted.**

**6) Tenha cuidado com os valores faltantes (NA). Lembre-se de que valores faltantes somente para algumas variáveis alteram o tamanho do conjunto de dados e amostras dependendo de qual variável é incluída em um dado modelo. É sugirido remover casos omissos antes de iniciar a seleção de modelos.**

**7) Use a mesma variável resposta para todos os modelos candidatos. É inadequado executar alguns modelos com variável resposta transformados e outros com a variável não transformada. A solução é usar uma função de ligação diferente para alguns modelos (e.g., identity vs. log link).**

**8) Quando se trata de modelos com overdispersion, utilize o mesmo valor de c-hat para todos os modelos em um conjunto de modelos candidatos. Para modelos binomiais com trials > 1 ou com Poisson GLM, deve-se estimar o c-hat do modelo mais complexo (modelo global). Se c hat > 1, deve-se usar o mesmo valor para cada modelo do conjunto de modelos candidatos e inclui- lo na contagem dos parâmetros (K). Da mesma forma, para binomial negativa, você deve estimar o parâmetro de dispersão do modelo global e usar o mesmo valor em todos os modelos.**

**9) Burnham e Anderson (2002) recomendam evitar misturar a abordagem da teoria da informação e noções de significância (ou seja, os valores P). É melhor fornecer estimativas e uma medida de sua precisão (erro padrão, intervalos de confiança).**

**10) Determinar o ranking das modelos é apenas o primeiro passo. A soma do Peso Akaike é 1 para o modelo de todo o conjunto e pode ser interpretado como o peso das evidências em favor de um determinado modelo. Modelos com grandes valores do Peso Akaike têm forte apoio. Taxas de evidências, valores de importância, e intervalo de confianca para o melhor modelo são outras medidas que auxiliam na interpretação. Nos casos em que o melhor modelo do ranking tem um Peso Akaike > 0,9, pode-se inferir que este modelo é o mais parcimonioso. Quando muitos modelos são classificados por valores altos (ou seja, o delta (Q) AIC (c) < 2 ou 4), deve- se considerar a média dos parâmetors dos modelos de interesse que aparecem no topo. A média dos modelos consiste em fazer inferências com base no conjunto de modelos candidatos, em vez de basear as conclusões em um único "melhor" modelo. É uma maneira elegante de fazer inferências com base nas informações contidas no conjunto inteiro de modelos.**

```
data(RoadKills)

## Carregando dados - Os dados consistem do número de mortes de anfíbios em uma rodovia em 52 sítios em Portugal

#Teoria: Ecologia de Paisagem

#Variável dependente: Número de anfíbios mortos

#Questão: Quais variáveis da paisagem melhor explicam a mortalidade de anfíbios?

# RK <- RoadKills

## Renomeando para facilitar

M1 <- glm (TOT.N ~ OPEN.L + MONT.S + SQ.POLIC + SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD + SQ.DWATCOUR + D.PARK, family = poisson, data=RK)

M2 <- glm (TOT.N ~ OPEN.L + MONT.S + SQ.POLIC + SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD + D.PARK, family = poisson, data=RK)

M3 <- glm (TOT.N ~ MONT.S + SQ.POLIC + SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD + D.PARK, family = poisson, data=RK)

M4 <- glm (TOT.N ~ L.WAT.C + SQ.LPROAD + D.PARK, family = poisson, data=RK)
```

AIC <- ICtab (M1, M2, M3, M4, type = c("AIC"), weights = TRUE, delta = TRUE, sort = TRUE)

AIC #Contudo, quando o número de amostras dividido pelo número de paramêtros for < 40 é recomendado utilizar um AIC corrigido (AICc) para pequenas amostras. Na verdade, como em grandes amostras o valor de AICc tende ao valor de AIC sem correção, é recomendado sempre utilizar AICc. #

AICc <- ICtab(M1, M2, M3, M4, type = c("AICc"), weights = TRUE, delta = TRUE, sort = TRUE, nobs = 52) AICc

## outra maneira de calcular

step(M1)

require(MuMIn) função (dredge)

## OVERDISPERSION —-

## Antes de analisar os resultados e realizar as análises de seleção você precisa checar se os seus dados possuem overdispersion. A overdispersion significa que a variância é maior do que a média.

M1 <- glm (TOT.N ~ OPEN.L + MONT.S + SQ.POLIC + SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD + SQ.DWATCOUR + D.PARK, family = poisson, data=RK) summary(M1)

## mudamos a distribuição?

M4 <- glm(TOT.N ~ OPEN.L + MONT.S + SQ.POLIC+ SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD+ SQ.DWATCOUR + D.PARK, family = quasipoisson, data = RK) summary(M4)

**Veja que o parâmetro de dispersão f é estimado em 5,93. Isto significa que todos os erros padrões foram multiplicados por 2,43 (a raiz quadrada de 5,93), e como resultado, a maioria dos parâmetros não são mais significativos. Não escreva na sua dissertação ou artigo que usou uma distribuição Quasi-Poisson. Quasi-Poisson não é uma distribuição. Basta dizer que você fez GLM com distribuição Poisson, detectou overdispersion, e corrigiu os erros padrões usando um modelo Quasi-GLM, onde a variância é dada por f × μ, onde μ é a média e f é o parâmetro de dispersão.**

**Em Quasi-Poisson não é possível calcular o valor de AIC. Por isso, é necessário calcular um valor de QUASI-AIC**

dd1 <- dredge (M4, rank = "QAICc", chat = summary(M4)$dispersion)

MQP1 <- get.models (dd1, 1:4) model.avg(MQP1)

**Os usuários devem ter em mente os riscos que correm usando tal "abordagem impensada" de avaliação de todos os modelos possíveis. Embora este procedimento seja útil em certos casos e justificado, ele pode resultar na escolha de um "melhor" modelo espúrio.**

**"Deixar o computador descobrir" é uma estratégia pobre e geralmente reflete o fato de que o pesquisador não se preocupou em pensar claramente sobre o problema de interesse e sua configuração científica (Burnham e Anderson, 2002).**

## Dados hierárquicos

## arquivo "gusanos.csv"

gusa<- (gusanos.csv)

**No âmbito de um projecto de melhoramento genético e de aclimatação de linhas genéticas do bicho-da-seda, um grupo de investigadores do Departamento de Zoologia Agrícola avaliou a produção de seda de duas linhas do bicho-da-seda recentemente admitidas no país. Os bichos-da-seda das linhagens chinesa e japonesa cresceram sob condições controladas de luz, fotoperíodo, umidade e alimentação. Os vermes foram colocados em caixas separadas por linhas (30 vermes por caixa em 3 caixas por linha)**

## 2.a- Pergunta de interesse:

**Quais características dos vermes podem prever a produção de seda?**

**2.b - Resposta variável:**

**Peso da casca (estimador da produção de seda).**

**A casca é a parte exterior do casulo e é calculada como "casulo - pupa = casca"**

**2.c- variáveis preditoras:**

**- linha: linha genética de vermes de seda**

**sexo**

**- peso da larva (gramas)**

**- peso do casulo (gramas)**

**- peso de pupa (gramas)**

**# Frequentemente, os dados com os quais trabalhamos violam a suposição de independência.**

**Isto é, os dados têm uma relação espacial, temporal ou filogenética.**

**Então, eles têm uma estrutura hierárquica (multinível ou aninhada).**

**3 - Identificar hierarquias ====**

**3.a- Quais hierarquias podemos identificar no estudo de caso?**

**minhocas contidos em caixas**

**Isto é, dois níveis**

**Neste caso, a variável de resposta e as variáveis preditoras são determinadas no nível de minhocas**

**No entanto, em modelos mais complexos, podemos encontrar variáveis medidas em diferentes níveis.**

**Por exemplo, enquanto o peso é medido no nível da minhoca**

**(temos dados por minhoca), a intensidade da luz que chega pode ser medida no nível de caixa (temos um dado por caixa).**

## 4 - modelos ====

mod_base <-lm (cut_weight ~ linha + sexo, dados = gusa)

# neste caso, “mod_base” é o nome do objeto que contém o modelo

# Começamos com uma função que ajusta modelos lineares (lm)

# Os argumentos da função são “answer” ~ (indica “depende de”) “predictor1” + “predictor2” … etc, data = “tabela de dados”

# Nesse caso, o peso do córtex depende da linha e do sexo.

## Como as interações do preditor são indicadas?

mod_base <-lm (cut_weight ~ linha + sex + linha: sex, dados = gusa)

# “:” indica interação

# Este último modelo é equivalente a mod_base <-lm (weight_roof ~ line * sex, data = gusa)

# O asterisco considera os efeitos simples e as interações entre as variáveis.

## 5 - Modelos de efeitos mistos ====

## Até agora definimos um modelo linear de efeitos fixos (lm).

## Isto é, ignoramos a estrutura de dependência dada pelas caixas que contêm os worms.

## Neste caso, a estrutura de dependência é dada pelas caixas que contêm os worms.

## # Podemos incluir a estrutura de dependência (worms dentro de caixas) no modelo através do uso de modelos de efeitos mistos.

## Existem pelo menos três funções que permitem trabalhar com modelos de efeitos mistos.

## 5.a - Pacote lme4

biblioteca ("lme4") # O pacote lme4 contém a função "lmer" para ajustar modelos # com estruturas de dependência.

mod_lmer <- lmer (weight_roof ~ line * sex + (1 | caixa), data = gusa) # "box" deve ser fator. # Se usarmos números para identificá-los, podemos especificar "(1 | factor (gusa $ caja))"

## Neste caso, a estrutura de dependência ou, fator aleatório, é indicado entre parênteses como (1 | caixa)

## Esse argumento indica que o modelo ajustará um

## Interceptar (ordenada para a origem) intercepta geral e diferente para cada caixa.

## 5.b - Pack nlme

biblioteca ("nlme") # No pacote nlme existem, entre outras, duas funções que permitem ajustar modelos com estruturas de dependência: lme e gls

mod_lme <-lme (weight_roof ~ line * sex, random = ~ 1 | box, data = gusa) # Neste caso, a estrutura de dependência é indicada com o argumento "random = ~ 1 | box"

mod_gls <-gls (peso_corteza ~ linha * sex, correlação = corCompSymm (form = ~ 1 | box), data = gusa) # Neste caso, a estrutura de dependência é indicada com o argumento "correlation = corCompSymm (form = ~ 1 | box)"

**As três formas especificam modelos análogos**

**5.c - Diferenças entre lme4 e nlme**

**Apesar de serem muito parecidos eles têm algumas diferenças:**

**• lme4 pode ser mais eficiente no uso da memória do que o nlme.**

**• lme4 permite incluir fatores aleatórios cruzados,**

**Enquanto isso não for possível no nlme.**

**• lme4 permite ajustar modelos de efeitos mistos lineares generalizados (GLMM), através da função glimmer.**

**Ou seja, os dados podem ser ajustados com distribuições de erros diferentes do normal, como binomial ou poisson.**

**Enquanto o nlme permite apenas a distribuição normal.**

**• nlme permite incluir funções para modelar a heterocedasticidade e**

**Correlação (temporal, espacial e filogenética) dos resíduos.**

**Estas funções não podem ser incluídas no lme4.**

**• o nlme é mais flexível que o lme4 para compor estruturas complexas de variações-covariâncias.**

## • o nlme está melhor documentado que o lme4 (até agora).

## 6 - Validação do modelo ====

## 6.a - Resíduos

E_lmer <-resid (mod_lmer, escalado = TRUE) # In lmer pedimos os resíduos “em escala” para que # Considere a estrutura de dependência (o fator aleatório). # O equivalente para o modelo lme seria:

E_lme <-resid (mod_lme, type = “normalized”)

# Com type = “normalized”, informamos para calcular os resíduos padronizados

# Esse desperdício é o que devemos usar para validar os modelos de efeitos mistos.

# 6.b –

#### Ajustado F_lmer <-fitted (mod_lmer)

# 6.c - Gráficos #### layout (matriz (1: 3, 1,3)) # a função “layout” permite dimensionar a quantidade de gráficos e distribuição na janela de gráficos gráfico (x = F_lmer, y = E_lmer, xlab = “ajustado”, ylab = “resíduos normalizados”) abline (0,0, col = “rede”, lwd = 3) # IMPORTANTE! Nós sempre queremos um gráfico que não mostre um padrão ou uma tendência. # Em contraste, neste caso, observamos três grandes grupos # com diferenças aparentes na dispersão dos resíduos ao longo dos valores ajustados.

boxplot (E_lmer ~ gusa $ linea, main = “linha genética”) boxplot (E_lmer ~ gusa $ sex, main = “gender”) # Observamos que a linha Eoro e as fêmeas possuem maior variância.

## 7 - Compare os três modelos ===

resumo (mod_lme) resumo (mod_lmer) resumo (mod_gls)

## 7.a - resumo da função lme

resumo (mod_lme)

**Modelo linear de efeitos mistos ajustado por REML**

**Data: gusa**

**AIC BIC logLik**

**-669.2444 -650.5006 340.6222**

**Existem diferentes índices de adequação ou probabilidade do modelo.**

**Posteriormente, usaremos o critério AIC para selecionar modelos.**

**Efeitos aleatórios:**

**Fórmula: ~ 1 | caixinha**

**(Interceptar) residual**

**StdDev: 0.005260823 0.03024337**

**Observamos o desvio do fator aleatório (caixas) e os resíduos do modelo.**

**variance = deviation ^ 2**

**IMPORTANTE: quando usamos a função lmer, o modelo nos dará tanto o desvio quanto a variância**

**Efeitos fixos: peso_corteza ~ linea * sexo**

**Valor Std.Error DF valor-t valor-p**

**(Interceptar) 0.22235107 0.005453594 164 40.77148 0.0000**

**lineaEoro -0.05210470 0.008119775 4 -6.41701 0,0030**

**sexM -0.00604792 0.006465802 164 -0.93537 0.3510**

**lineaEoro: sexoM 0.02831680 0.009366297 164 3.02327 0.0029**

**Observe o modelo gera para os efeitos fixos (simples e interações) a estimação do parâmetro.**

**A primeira linha refere-se à interceptação do modelo. Ou seja, o modelo estima que a ordem de origem seja 0,22.**

**ATENÇÃO:**

**Neste caso, o modelo estima que o peso da casca da pupa feminina da linha CE é de 0,22.**

**O peso da casca das fêmeas da linha Eoro é 0,052 menor que as da linha CE (0,22-0,052 = 1,68)**

**O peso da casca dos machos EC é 0,006 menor que as fêmeas da mesma linha (0,222-0,0064 = 0,2156)**

**y, o peso da casca dos machos Eoro é 0,028 maior que as fêmeas CE (0,22 + 0,028 = 0,248)**

**Em seguida, as colunas indicam o erro padrão, os graus de liberdade (DF), o valor da estatística T e o valor de probabilidade associado.**

**ATENÇÃO: os testes de T comparam os níveis dos fatores em relação à ordem de origem. Mais tarde, vamos ver outro tipo de análise.**

**Correlação:**

**(Intr) lineEr sexM**

**lineaEoro -0.672**

**sexoM -0.587 0.394**

**lineaEoro: sexoM 0.405 -0.628 -0.690**

**Essa tabela de correlação nos mostraria multicolinearidade.**

**Residuais padronizados dentro do grupo:**

**Min Q1 Med Q3 Max**

**-4.24827208 -0.63482939 0.01851832 0.58041960 3.50352726**

Número de observações: 172 Número de grupos: 6

## 7.b - resumo da função lmer

resumo (mod_lmer) # Modelo misto linear ajustado por REML [‘lmerMod’] # Formula: weight_cortex ~ line * sex + (1 | caixa) # Data: gusa # # Critério REML na convergência: -681,2

**Indica a configuração do modelo.**

**Posteriormente, usaremos o critério AIC para selecionar modelos.**

**Resíduos escalonados:**

**Min 1T Median 3Q Max**

**-4.2483 -0,6348 0,0185 0,5804 3,5035**

**Efeitos aleatórios:**

**Grupos Nome Variação Std.Dev.**

**box (Intercept) 2.768e-05 0.005261**

**Residual 9.147e-04 0.030243**

**Número de obs: 172, grupos: caixa, 6**

**É equivalente à informação apenas para o modelo lme**

**com uma ordem ligeiramente diferente.**

**Observamos a variância e o desvio do fator aleatório (caixas) e os resíduos do modelo.**

**variance = deviation ^ 2**

**IMPORTANTE: quando usamos a função lme, o modelo só nos dará o desvio**

**Efeitos fixos:**

**Estimate Std. Error t value**

**(Interceptar) 0,222351 0,005454 40,77**

**lineaEoro -0.052105 0.008120 -6.42**

**sexoM -0.006048 0.006466 -0.94**

**lineaEoro: sexoM 0.028317 0.009366 3.02**

**Correlação de efeitos fixos:**

**(Intr) lineEr sexM**

**lineaEoro -0.672**

**sexoM -0.587 0.394**

**lineaEr: sxM 0.405 -0.628 -0.690**

**7.c - resumo da função gls**

resumo (mod_gls) # Quadrados mínimos generalizados ajustados por REML # Modelo: peso_corteza ~ linea * sex # Data: gusa # AIC BIC logLik # -669.2444 -650.5006 340.6222

**O modelo ajustado pelos mínimos quadrados gerou valores de ajuste iguais ao modelo misto ajustado com a função nlme.**

**Compare:**

**Modelo linear de efeitos mistos ajustado por REML**

**Data: gusa**

**AIC BIC logLik**

**-669.2444 -650.5006 340.6222**

**Estrutura de Correlação: simetria composta**

**Fórmula: ~ 1 | caixinha**

**Estimativa (s) de parâmetro:**

**Rho**

**0.02936948**

**Quando usamos essa estrutura de correlação, o parâmetro “rho”**

**é o coeficiente de correlação entre dois resíduos da mesma caixa (neste caso).**

**Tanto a interceptação aleatória usada em “mod_lme” quanto a**

## estrutura de correlação usada em “mod_gls” assume que o

## Correlação entre qualquer par de observações dentro de uma caixa é a mesma.

## Eles são semelhantes?

## Coeficientes:

## Valor Std.Error valor-t valor-p

## (Intercepção) 0.22235107 0.005453585 40.77154 0.0000

## lineaEoro -0.05210470 0.008119763 -6.41702 0.

## Abelhas —-

data(Bees) #Como variável dependente temos densidade de esporos medido em cada colméia. A variável independente Infection quantifica o grau de infecção, com valores 0, 1, 2 e 3. Embora mixed effects modelling podem lidar com um certo grau de dados desbalanceados, neste caso, é melhor converter a variável Infection em 0 (sem infecção) e 1 (infectado) porque existem poucas observações com valores 2 e 3. #Transformar a variável Infection em presença e ausência: Bees$Infection01 < -Bees$Infection Bees$Infection01[Bees$Infection01 > 0] <- 1 Bees$fInfection01 < -factor(Bees$Infection01)

## Transformar colméia em fator e logaritimizar esporos:

Bees$fHive < -factor(Bees$Hive) Bees$LSpobee < -log10(Bees$Spobee + 1)

## Plotar os dados por colméia:

op <- par(mfrow = c(1, 2), mar = c(3, 4, 1, 1)) dotchart(Bees$Spobee, groups = Bees$fHive) dotchart(Bees$LSpobee, groups = Bees$fHive) par(op)

## Começaremos com uma regressão linear e plotaremos os resíduos por colmeia:

M1 <- lm (LSpobee ~ fInfection01 * BeesN, data = Bees) E1 <- rstandard(M1) plot (E1 ~ Bees$fHive, ylab = "Resíduos", xlab = "Colméias") abline (0, 0)

## Veja que algumas colméias apresentam os três resíduos acima do esperado, enquanto outras possuem três resíduos abaixo do esperado. Temos a opção de colocar colméia como random effect.

## Selecionando random effect

M1 <- lme(LSpobee ~ fInfection01 * BeesN, random = ~ 1 | fHive, method = "REML", data = Bees) M2 <- lme(LSpobee ~ fInfection01 * BeesN, random = ~ 1 + BeesN | fHive, method = "REML", data = Bees) M3 <- lme (LSpobee ~ fInfection01 * BeesN, random = ~ 1 + fInfection01 | fHive, method = "REML") anova(M1,M2) anova(M1,M3)

plot (M1, col = 1) boxplot (LSpobee ~ fInfection01, data = Bees, varwidth = TRUE) #por infecção #Veja que há diferença na variação entre as categorias. Inserimos um comando para dizer que as variâncias para infecção são diferentes. #varIdent = permite modelar diferentes variâncias para variáveis categóricas.

M1 <- lme (LSpobee ~ fInfection01 * BeesN, random = ~ 1 | fHive, method = "REML", data = Bees) M4 <- lme (LSpobee ~ fInfection01 * BeesN, random = ~ 1 | fHive, method = "REML", data = Bees, weights = varIdent (form = ~ 1 | fInfection01)) anova (M1,M4)

## Selecionando estrutura fixa:

M7full<- lme (LSpobee ~ fInfection01 * BeesN, random = ~ 1|fHive, weights = varIdent(form = ~ 1 | fInfection01), method = "ML", data = Bees)

M7sub <- update(M7full, .~. -fInfection01 : BeesN )

anova (M7full,M7sub)

M8full <- lme (LSpobee ~ fInfection01 + BeesN, random = ~ 1|fHive, method = “ML”, data = Bees, weights = varIdent(form =~ 1 | fInfection01))

M8sub1 <- update (M8full, .~. -fInfection01 )

M8sub2 <- update (M8full, .~. -BeesN )

anova(M8full,M8sub1)

anova(M8full,M8sub2)

M9full<-lme(LSpobee ~ fInfection01, random = ~ 1|fHive, method=“ML”, data = Bees, weights = varIdent(form =~ 1 | fInfection01))

M9sub1<-update(M9full, .~. -fInfection01 )

anova(M9full,M9sub1)

## Modelo final:

Mfinal <- lme (LSpobee ~ fInfection01, random =~ 1|fHive, data = Bees, weights = varIdent (form = ~ 1 | fInfection01), method = “REML”)

plot(Mfinal)