Смешаные линейные модели для счетных данных

Линейные модели...

Марина Варфоломеева, Вадим Хайтов

Вы узнаете

- Как анализировать данные, в которых зависимая переменная - счетная величина, и есть случайные факторы

Вы сможете

- Построить линейные модели с пуассоновским и отрицательным биномиальным распределением отклика
- Сможете проверить смешанные модели на избыточность дисперсии
- Научитесь проверять наличие нелинейных паттернов в остатках

Смешанные модели для счетных данных

От чего завист призывный крик совят?

Данные из Roulin & Bersier 2007, пример из кн. Zuur et al., 2007

фото с http://www.mobilmusic.ru/wallpaper.php?id=1343149)

Будем моделировать зависимость числа криков совят (SiblingNegotiation) от многих факторов

FoodTreatment - тритмент (сытые или голодные)

SexParent - пол родителя

FoodTreatment x SexParent

ArrivalTime - время прибытия родителя

ArrivalTime x SexParent

Nest - гнездо

Загружаем пакеты и данные

```
# Загружаем нужные пакеты
library(downloader) # для загрузки файлов из интернета
library(ggplot2)
theme_set(theme_bw(base_size = 14) + theme(legend.key = element_blank()))
update_geom_defaults("point", list(shape = 19))
library(gridExtra)

# Загружаем данные из интернета и сохраняем в файл, если его еще нет
url <- "https://raw.githubusercontent.com/varmara/linmodr/master/16-glmm-pois
filename <- "data/Owls_Roulin_Bersier_2007.csv"
if (!file.exists(filename)) download(url, filename)
```

Знакомство с данными

```
Owls <- read.delim("data/Owls_Roulin_Bersier_2007.csv")
str(Owls)
```

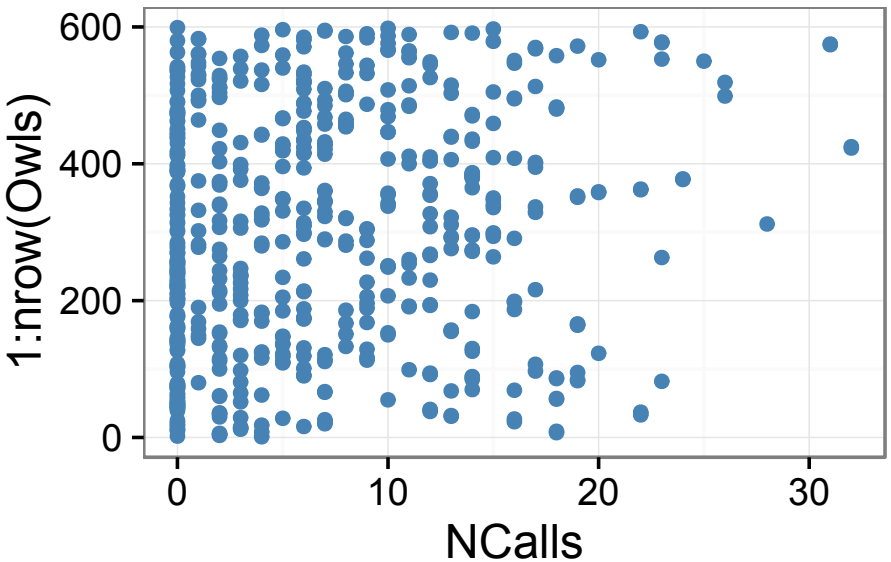
```
# 'data.frame': 599 obs. of  8 variables:
#  $ Nest              : Factor w/ 27 levels "AutavauxTV","Bochet",..: 1 1 1
#  $ FoodTreatment     : Factor w/ 2 levels "Deprived","Satiated": 1 2 1 1 1
#  $ SexParent         : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 1
#  $ ArrivalTime       : num  22.2 22.4 22.5 22.6 22.6 ...
#  $ SiblingNegotiation: int  4 0 2 2 2 2 18 4 18 0 ...
#  $ BroodSize         : int  5 5 5 5 5 5 5 5 5 5 ...
#  $ NegPerChick       : num  0.8 0 0.4 0.4 0.4 0.4 3.6 0.8 3.6 0 ...
#  $ logBroodSize      : num  1.61 1.61 1.61 1.61 1.61 ...
```

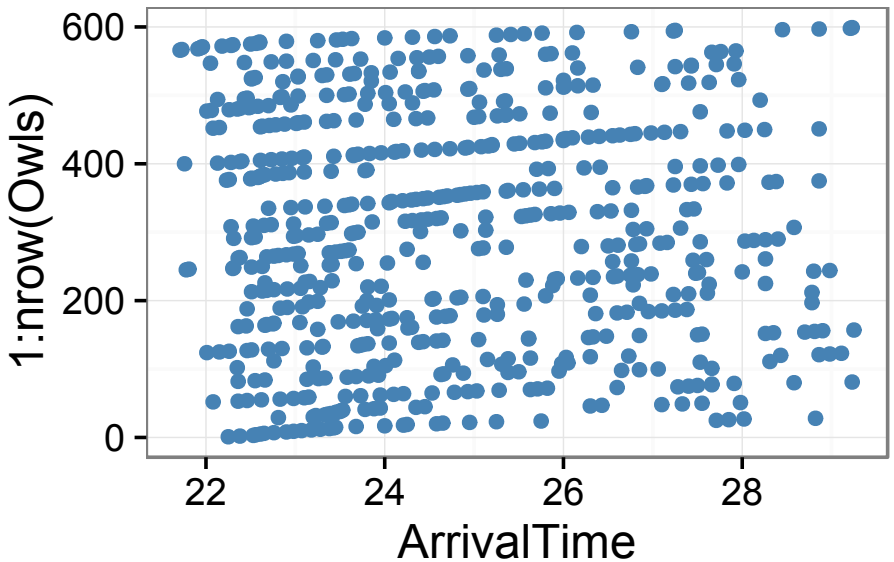
```
#' SiblingNegotiation - число криков совят
#- заменим на более короткое название
Owls$NCalls <- Owls$SiblingNegotiation
# Число пропущенных значений
sum(!complete.cases(Owls))
```

```
# [1] 0
```

Есть ли выбросы?

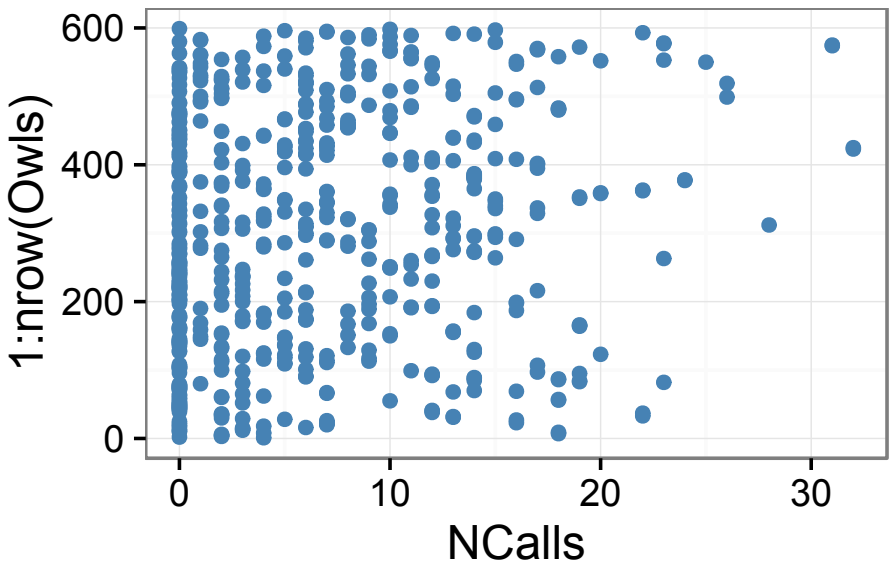
```
# Есть ли наблюдения-выбросы? строим dot-plot
dotplot <- ggplot(Owls, aes(y = 1:nrow(Owls))) + geom_point(colour = "steelbl

grid.arrange(dotplot + aes(x = NCalls),
  dotplot + aes(x = ArrivalTime), nrow = 1)
```

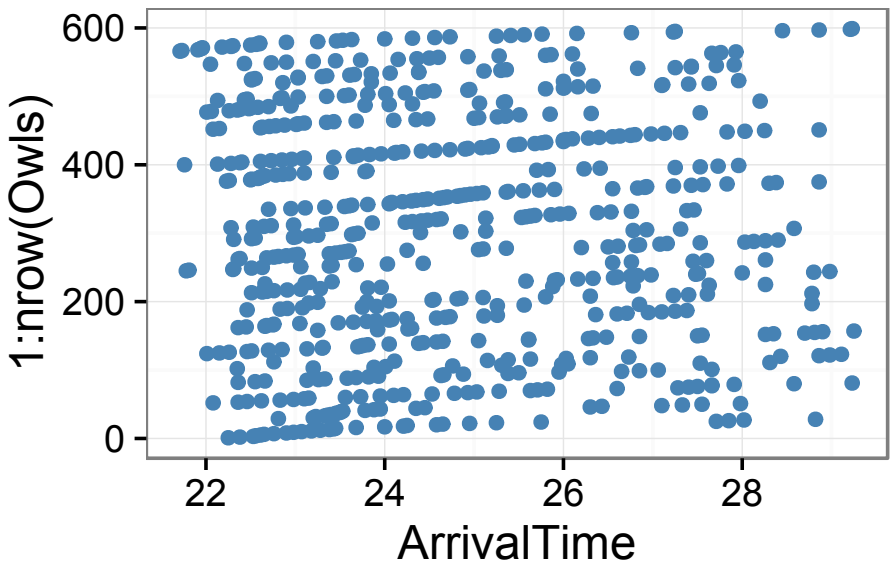




Есть ли выбросы?

```
# Есть ли наблюдения-выбросы? строим dot-plot
dotplot <- ggplot(Owls, aes(y = 1:nrow(Owls))) + geom_point(colour = "steelblu

grid.arrange(dotplot + aes(x = NCalls),
  dotplot + aes(x = ArrivalTime), nrow = 1)
```

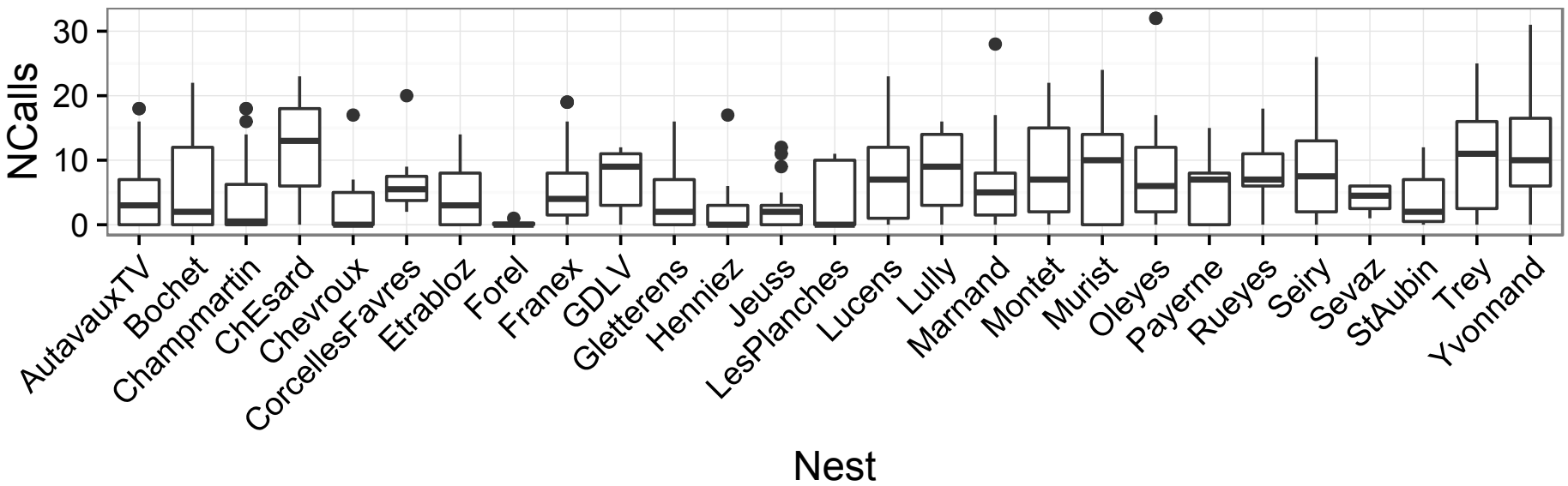




Выбросов нет

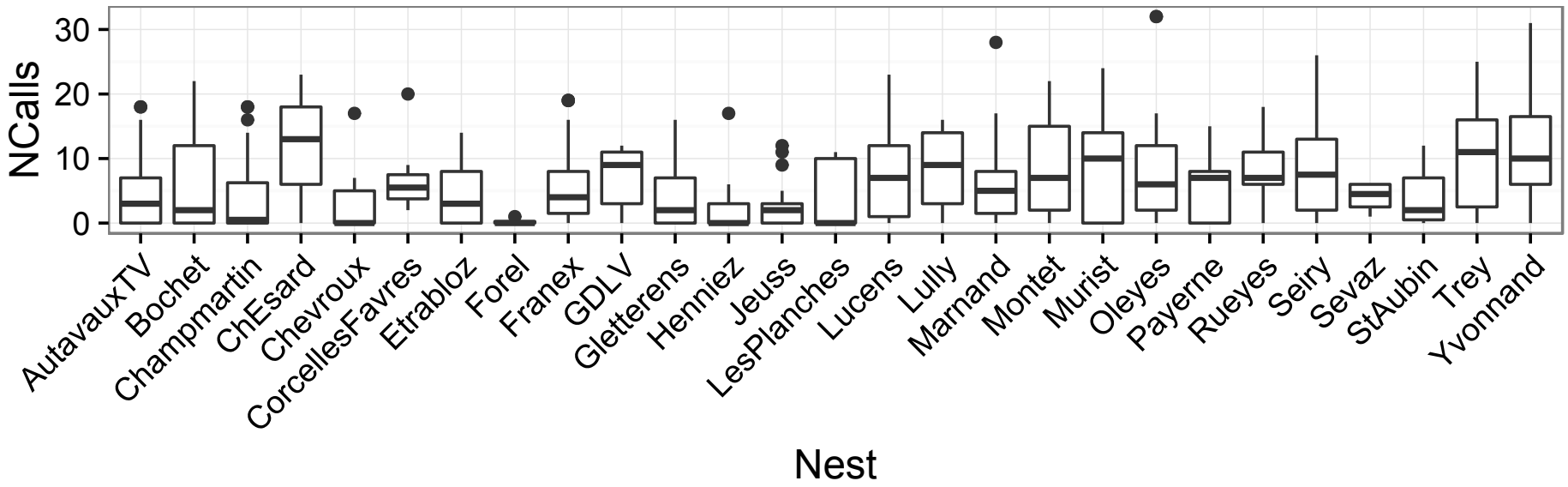
Различаются ли гнезда?

```
ggplot(Owls, aes(x = Nest, y = NCalls)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Различаются ли гнезда?

```
ggplot(Owls, aes(x = Nest, y = NCalls)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Гнезд много, они различаются. Можно и нужно учесть как случайный эффект

Сколько наблюдений в каждом гнезде?

```
table(Owls$Nest)
```

```
#
#      AutavauxTV           Bochet     Champmartin         ChEsard
#              28               23              30              20
#        Chevroux CorcellesFavres        Etrabloz           Forel
#              10               12              34               4
#          Franex            GDLV      Gletterens         Henniez
#              26               10              15              13
#           Jeuss     LesPlanches          Lucens           Lully
#              19               17              29              17
#         Marnand           Montet          Murist          Oleyes
#              27               41              24              52
#         Payerne           Rueyes           Seiry           Sevaz
#              25               17              26               4
#         StAubin            Trey        Yvonnand
#              23               19              34
```

Сколько наблюдений в каждом гнезде?

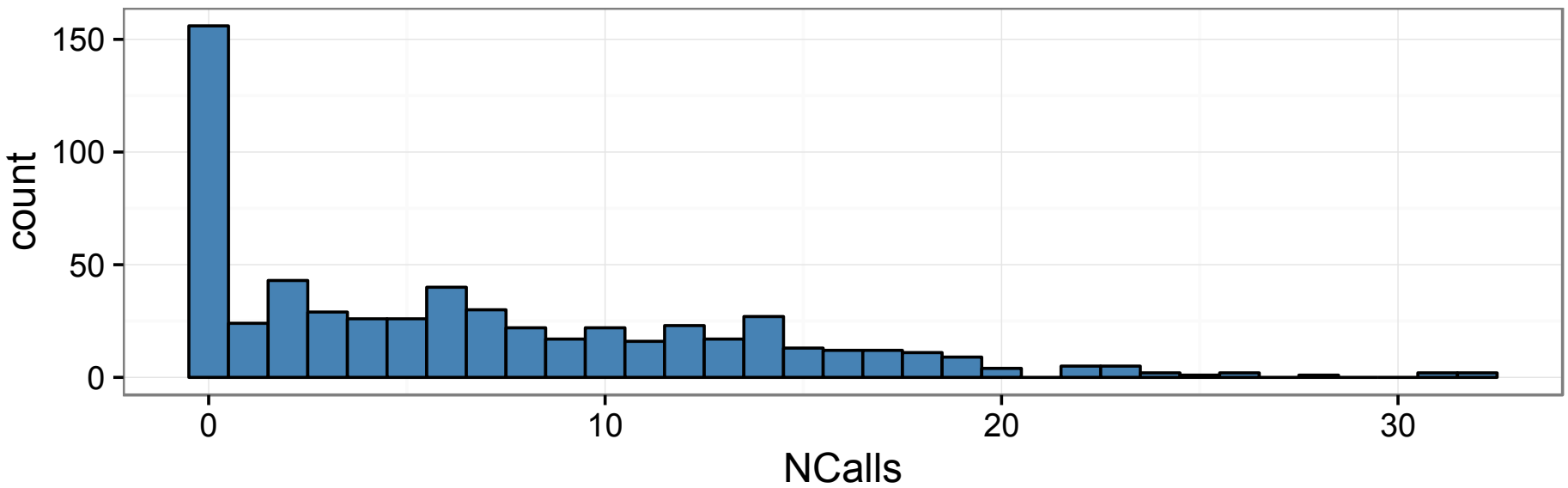
```
table(Owls$Nest)
```

```
#
#      AutavauxTV           Bochet     Champmartin         ChEsard
#              28               23              30              20
#        Chevroux CorcellesFavres        Etrabloz           Forel
#              10               12              34               4
#          Franex            GDLV      Gletterens         Henniez
#              26               10              15              13
#           Jeuss     LesPlanches          Lucens           Lully
#              19               17              29              17
#         Marnand           Montet          Murist          Oleyes
#              27               41              24              52
#         Payerne           Rueyes           Seiry           Sevaz
#              25               17              26               4
#         StAubin            Trey        Yvonnand
#              23               19              34
```

Хорошо, что наблюдений в каждом гнезде много. Только в двух по четыре - не очень.

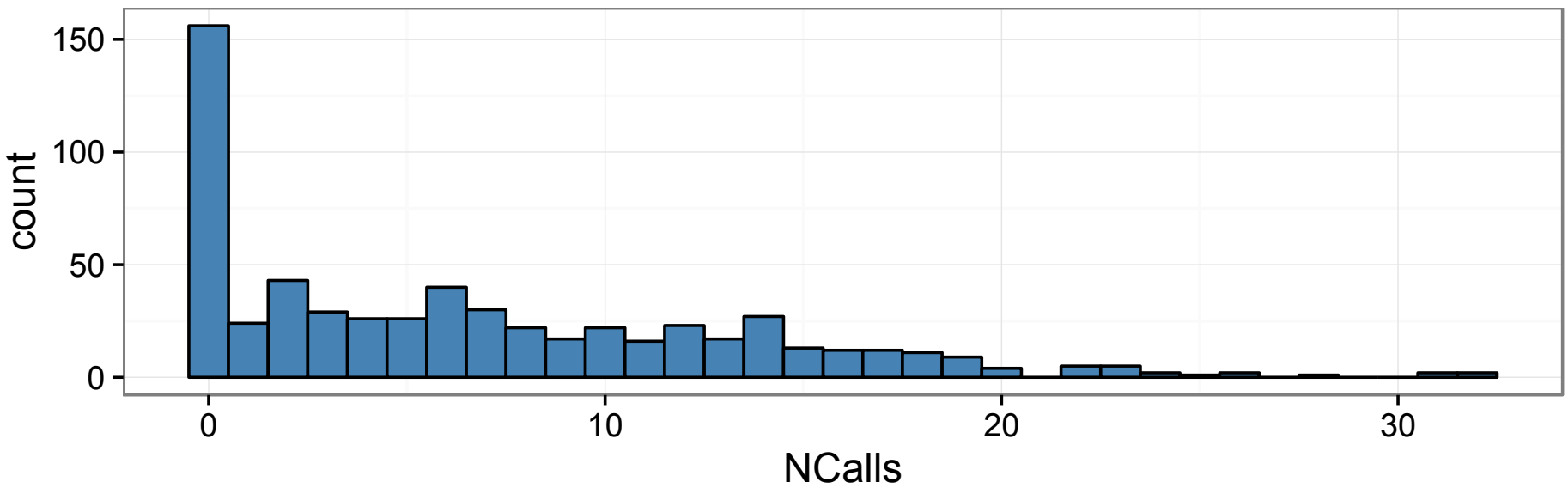
Как распределен отклик?

```
ggplot(Owls, aes(x = NCalls)) +
  geom_histogram(binwidth = 1, fill = "steelblue", colour = "black")
```



Как распределен отклик?

```
ggplot(Owls, aes(x = NCalls)) +
  geom_histogram(binwidth = 1, fill = "steelblue", colour = "black")
```



Напоминает распределение Пуассона

Сколько нулей?

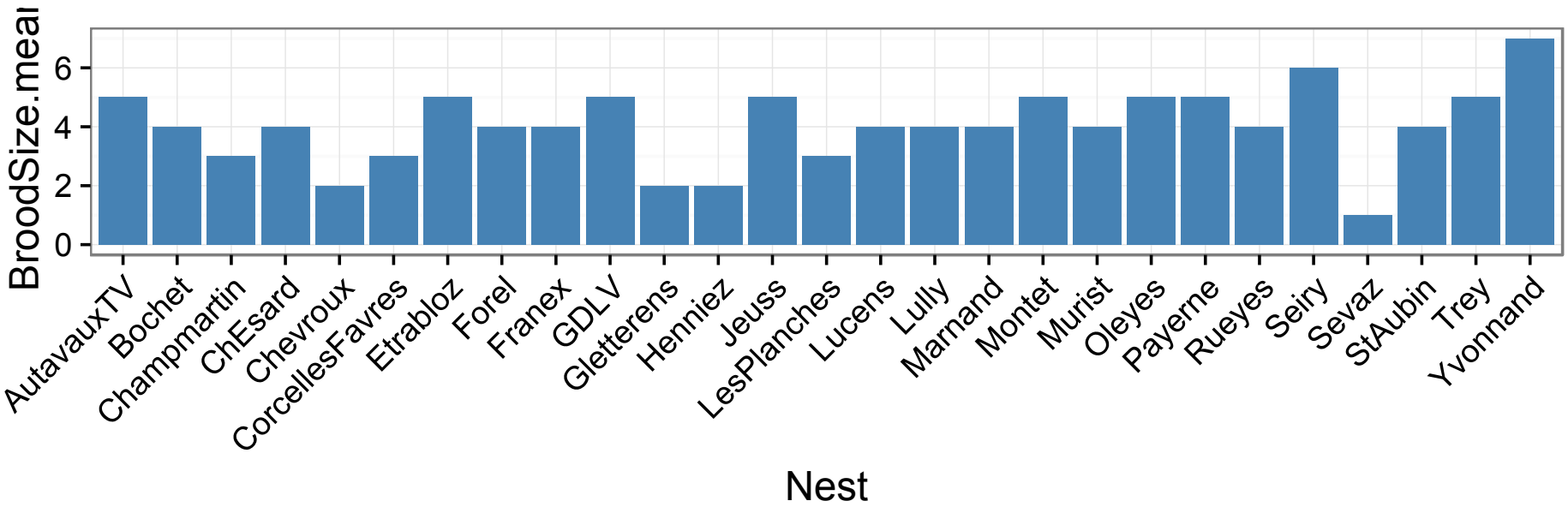
```
sum(Owls$NCalls == 0)/nrow(Owls)
```

```
# [1] 0.26
```

26% нулей

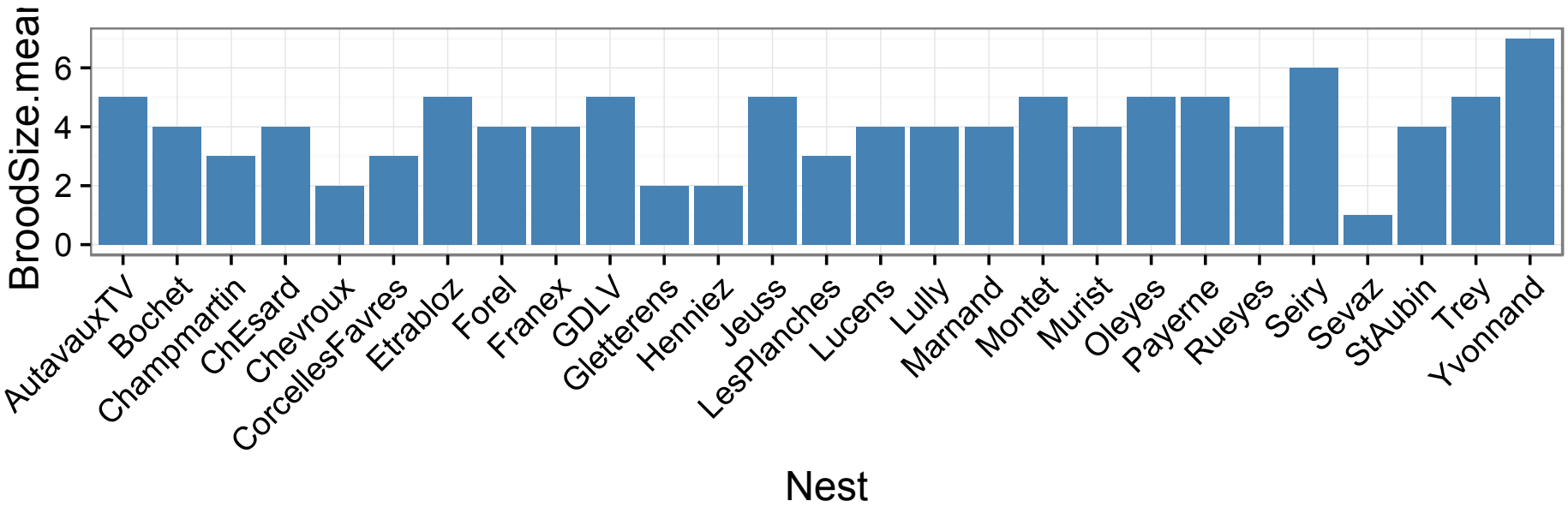
Какого размера выводки в гнездах?

Это нужно учесть, потому что чем больше выводок, тем больше птенцов будут разговаривать.



Какого размера выводки в гнездах?

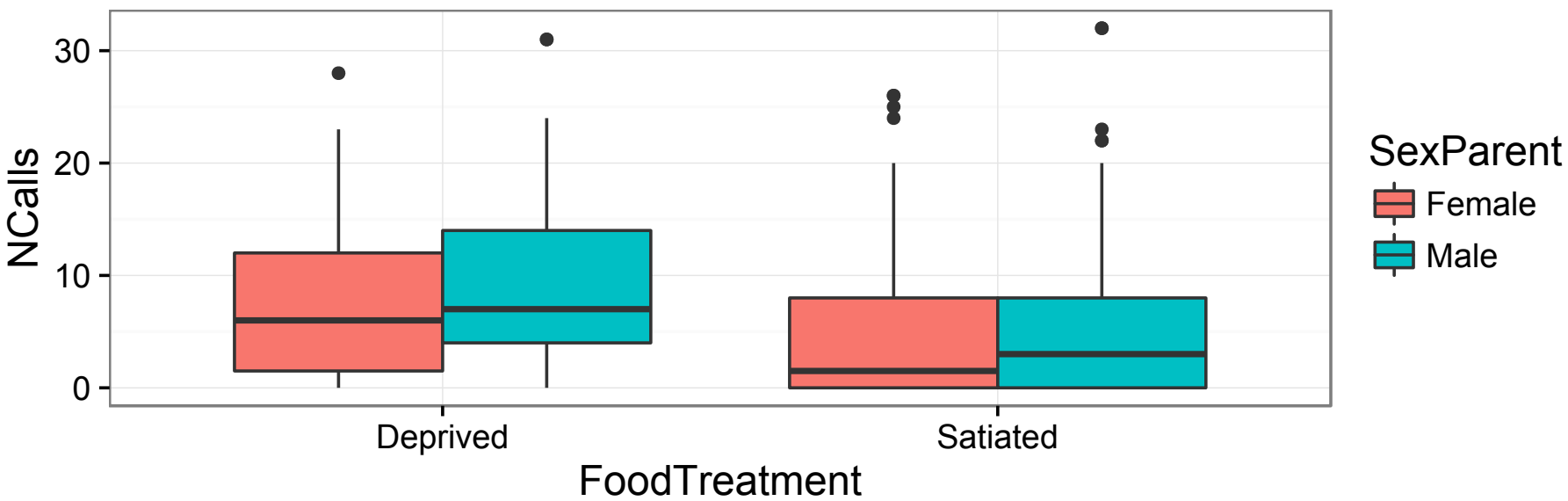
Это нужно учесть, потому что чем больше выводок, тем больше птенцов будут разговаривать.



Выводки разные. В пуассоновской glmer() это можно откорректировать при помощи offset. Сделаем offset(logBroodSize)

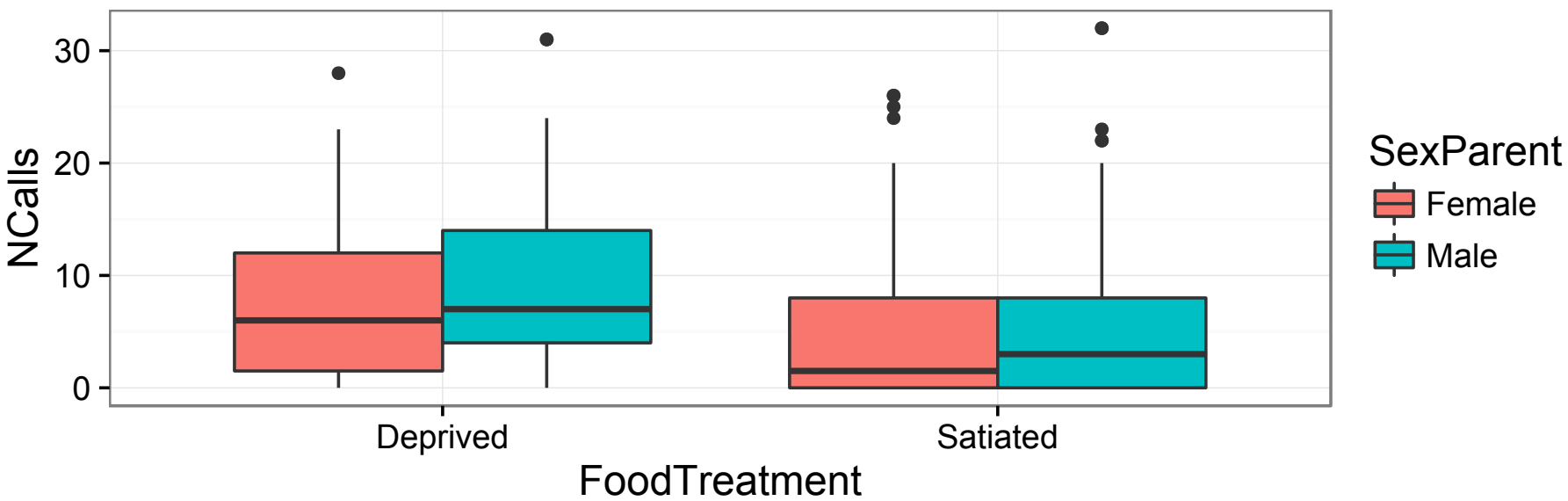
Боксплоты для дискретных факторов

```
ggplot(Owls, aes(y = NCalls, x = FoodTreatment, fill = SexParent)) +
  geom_boxplot()
```



Боксплоты для дискретных факторов

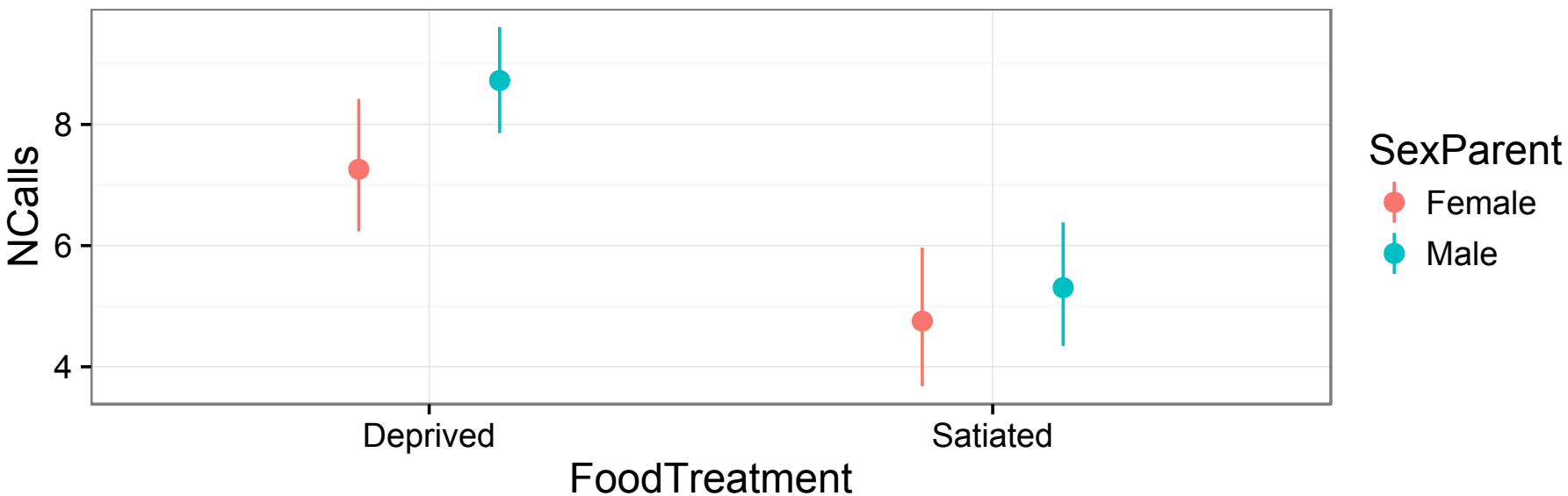
```
ggplot(Owls, aes(y = NCalls, x = FoodTreatment, fill = SexParent)) +
  geom_boxplot()
```



Подозрительно, возможно, есть взаимодействие.

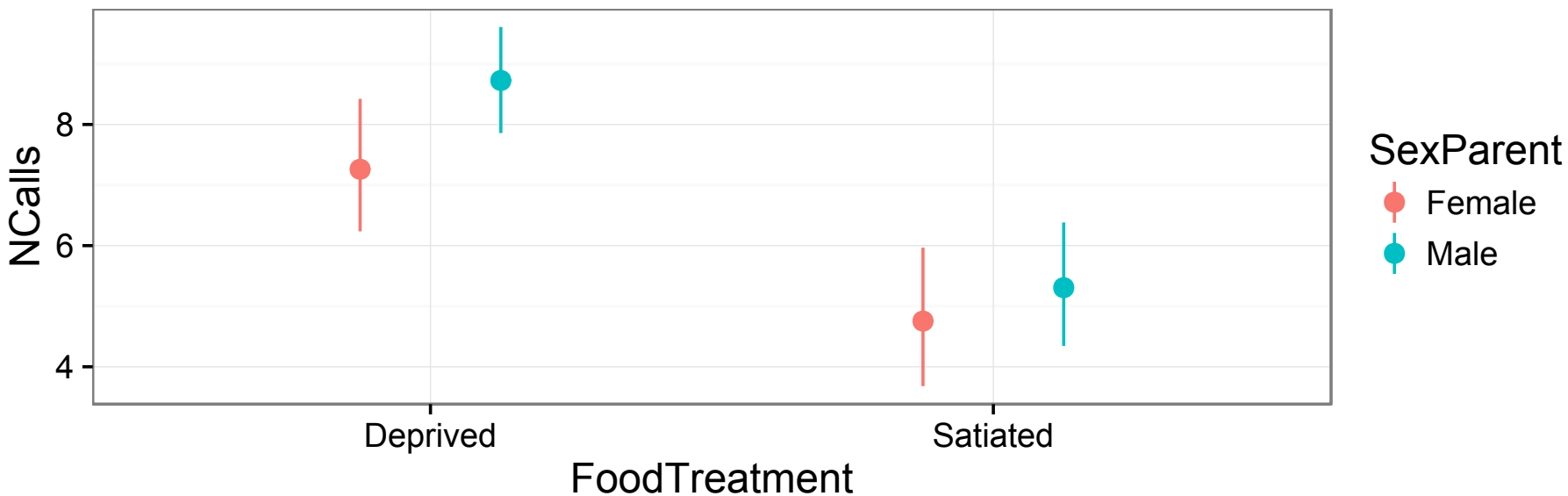
Может быть есть взаимодействие?

```
ggplot(Owls) +
  stat_summary(aes(x = FoodTreatment, y = NCalls, colour = SexParent),
               fun.data = "mean_cl_boot",
               position = position_dodge(width = 0.5))
```



Может быть есть взаимодействие?

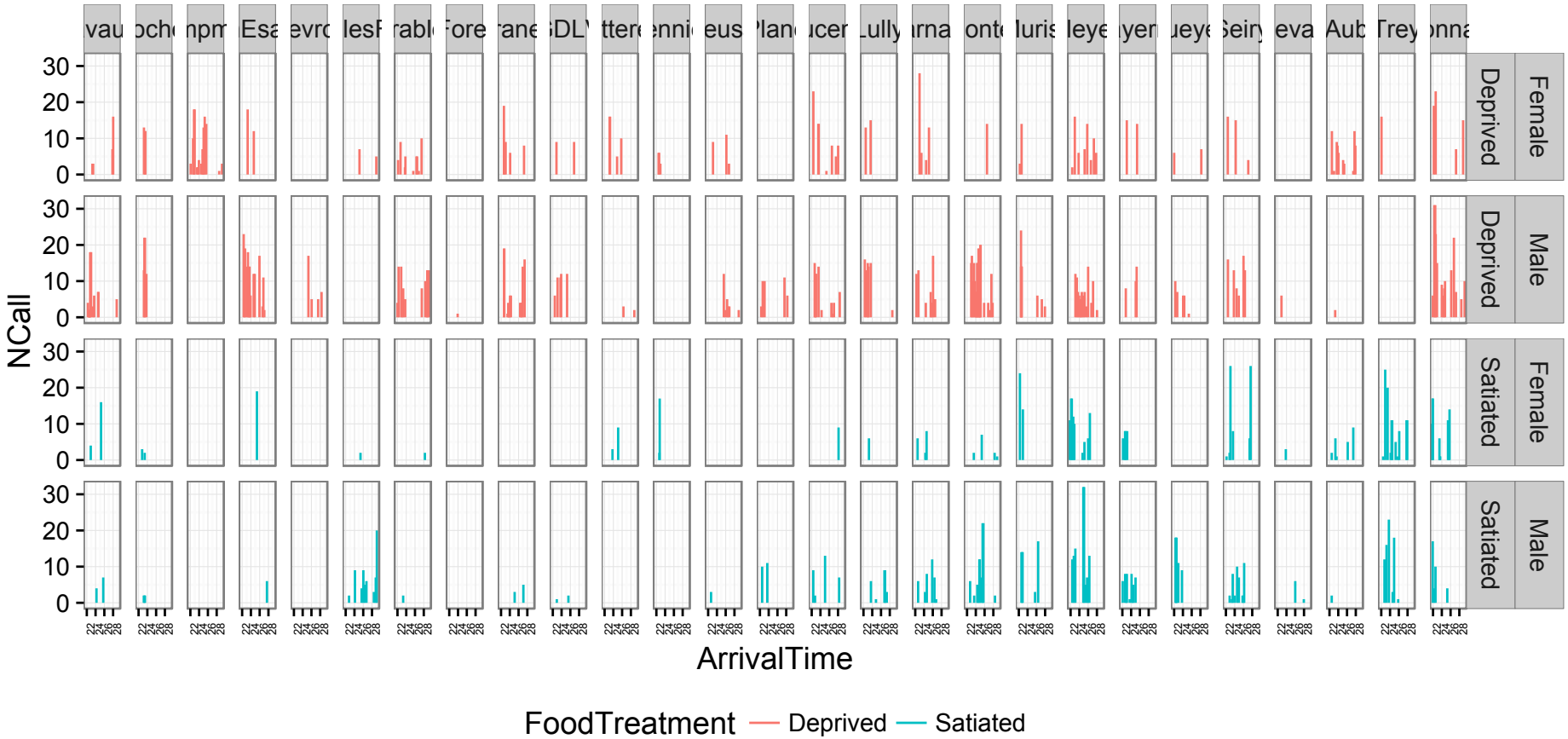
```
ggplot(Owls) +
  stat_summary(aes(x = FoodTreatment, y = NCalls, colour = SexParent),
               fun.data = "mean_cl_boot",
               position = position_dodge(width = 0.5))
```



Похоже, что может быть взаимодействие

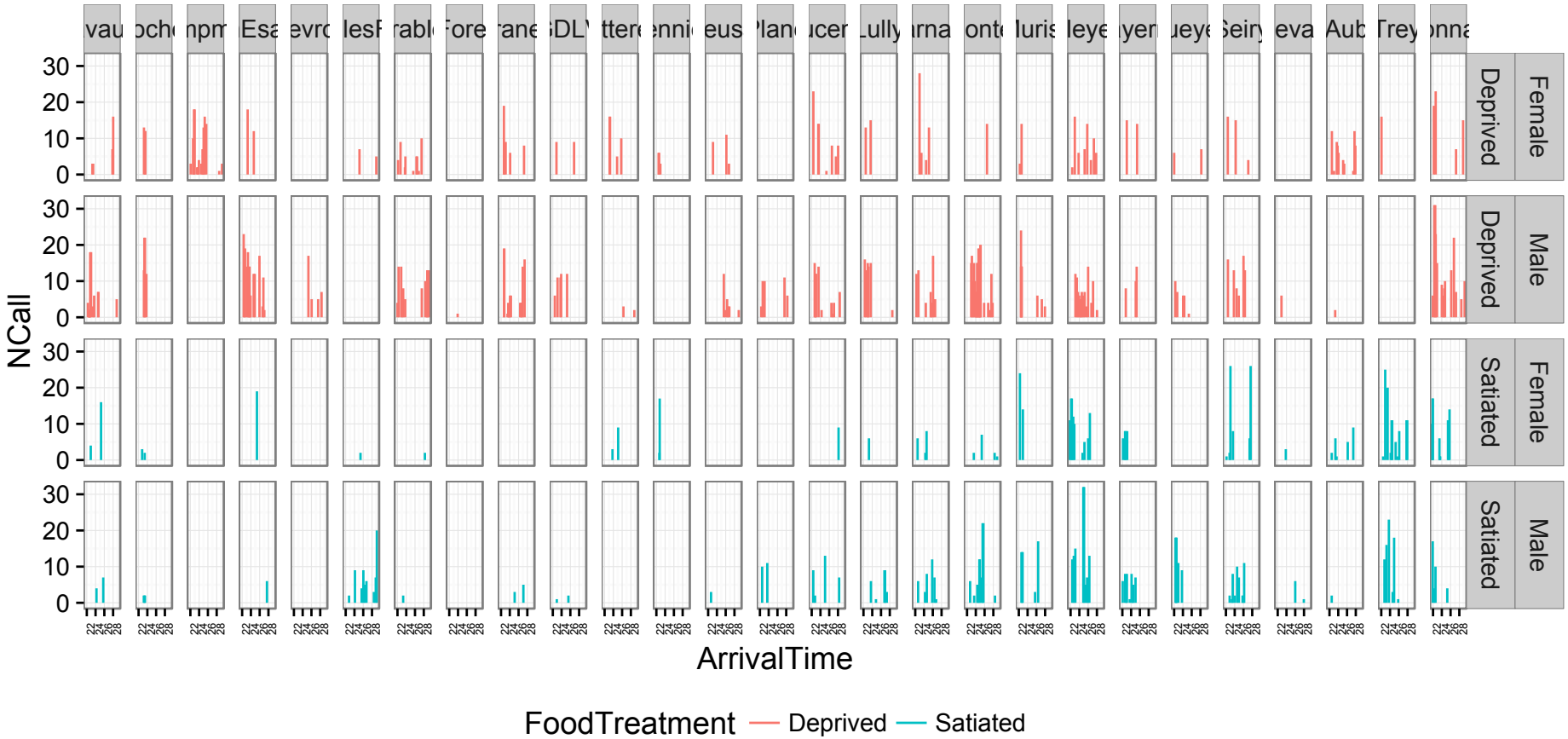
Задание: Постройте такой график

Нарисуем все данные для будущей модели



Задание: Постройте такой график

Нарисуем все данные для будущей модели



Птенцы больше орут, если голодали прошлой ночью. И возможно, орут у самцов

Код для графика

```
ggplot(Owls) + geom_segment(aes(x = ArrivalTime, y = 0,
                                xend = ArrivalTime,
                                yend = NCalls, colour = FoodTreatment)) +
  facet_grid(FoodTreatment + SexParent ~ Nest) +
  ylab("NCall") +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90, hjust = 1, size = 5))
```

Колинеарность

```
M0 <- lm(NCalls ~ SexParent + FoodTreatment + ArrivalTime, data = Owls)
library(car)
vif(M0)
```

```
#     SexParent FoodTreatment   ArrivalTime
#        1.0036        1.0044        1.0024
```

Колинеарность

```
M0 <- lm(NCalls ~ SexParent + FoodTreatment + ArrivalTime, data = Owls)
library(car)
vif(M0)
```

```
#     SexParent FoodTreatment   ArrivalTime
#        1.0036        1.0044         1.0024
```

ок

Линейная модель с пуассоновским распределением остатков

$$NCalls_{ij} = e^{\beta_0+\beta_1 SexP_M+\beta_2 FoodT_S+\beta_3 ArrivalT+\beta_4 SexP_M:FoodT_S+\beta_5 SexPM:ArrivalT+log(BroodSize)+a_i+\epsilon_{ij}}$$

$NCalls \sim \boldsymbol{P}(\mu_{ij})$ — отклик подчиняется распределению Пуассона с параметром μ $E(NCalls_{ij}) = \mu_{ij}$
$var(NCalls_{ij}) = \mu_{ij}$ $ln(\mu_{ij}) = \eta_{ij}$ — функция связи — логарифм

$$\eta_{ij} = \beta_0 + \beta_1 SexParent_M +$$
$$+\beta_2 FoodTreatment_S + \beta_3 ArrivalTime +$$
$$+\beta_4 SexParent_M : FoodTreatment_S +$$
$$+\beta_5 SexParentM : ArrivalTime +$$
$$+log(BroodSize) + a_i + \epsilon_{ij}$$

$a_i \sim N(0, \sigma^2_{Nest})$ — случайный эффект гнезда (intercept)
$\epsilon_{ij} \sim N(0, \sigma^2)$ — остатки модели
i — гнездо
j — наблюдение

Подберем линейную модель с пуассоновским распределением остатков

```
library(lme4)
M1 <- glmer(NCalls ~ SexParent * FoodTreatment +
              SexParent * ArrivalTime +
              offset(logBroodSize) +
              (1 | Nest),
            family = "poisson", data = Owls)
```

```
# Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
# $checkConv, : Model failed to converge with max|grad| = 0.00549642
# (tol = 0.001, component 1)
```

```
# Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv
#  - Rescale variables?
```

Смешанная модель с распределением пуассона не сходится. Один из возможных вариантов выхода - стандартизация предикторов

Стандартизируем непрерывные предикторы

У нас только один непрерывный предиктор

```
Owls$ArrivalTime_std <- (Owls$ArrivalTime - mean(Owls$ArrivalTime)) /
  sd(Owls$ArrivalTime)
```

Модель со стандартизированным предиктором

```
M1 <- glmer(NCalls ~ SexParent * FoodTreatment +
              SexParent * ArrivalTime_std +
              offset(logBroodSize) +
              (1 | Nest),
            family = "poisson", data = Owls)
```

Эта модель сходится

Задание:

Проверьте модель M1 на избыточность дисперсии

Избыточность дисперсии (Overdispersion)

```
R_M1 <- resid(M1, type = "pearson") # Пирсоновские остатки
N <- nrow(Owls) # Объем выборки
p <- length(fixef(M1)) + 1   # Число параметров в модели (сигма для гнезда)
df <- (N - p) # число степенейсвободы
fi <- sum(R_M1^2) /df
#Величина fi показывает во сколько раз в среднем sigma > mu для данной модели
fi
```

```
# [1] 5.46
```

Избыточность дисперсии (Overdispersion)

```
R_M1 <- resid(M1, type = "pearson") # Пирсоновские остатки
N <- nrow(Owls) # Объем выборки
p <- length(fixef(M1)) + 1   # Число параметров в модели (сигма для гнезда)
df <- (N - p) # число степенейсвободы
fi <- sum(R_M1^2) /df
#Величина fi показывает во сколько раз в среднем sigma > mu для данной модели
fi
```

```
# [1] 5.46
```

Явная избыточность дисперсии.

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

-Отскакивающие значения –> **убрать**

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

-Отскакивающие значения –> **убрать**

-Пропущены ковариаты или взаимодействия предикторов –> **добавить**

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

-Отскакивающие значения –> **убрать**

-Пропущены ковариаты или взаимодействия предикторов –> **добавить**

- Наличие внутригрупповых корреляций (нарушение независимости выборок) –> **включить другие случайные эффекты**

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

-Отскакивающие значения –> **убрать**

-Пропущены ковариаты или взаимодействия предикторов –> **добавить**

- Наличие внутригрупповых корреляций (нарушение независимости выборок) –> **включить другие случайные эффекты**

- Нелинейная взаимосвязь между ковариатами и зависимой переменной –> **GAMM**

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

-Отскакивающие значения –> **убрать**

-Пропущены ковариаты или взаимодействия предикторов –> **добавить**

- Наличие внутригрупповых корреляций (нарушение независимости выборок) –> **включить другие случайные эффекты**

- Нелинейная взаимосвязь между ковариатами и зависимой переменной –> **GAMM**

- Неверно подобрана связывающая функция –> **заменить**

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

-Отскакивающие значения –> **убрать**

-Пропущены ковариаты или взаимодействия предикторов –> **добавить**

- Наличие внутригрупповых корреляций (нарушение независимости выборок) –> **включить другие случайные эффекты**

- Нелинейная взаимосвязь между ковариатами и зависимой переменной –> **GAMM**

- Неверно подобрана связывающая функция –> **заменить**

- Количество нулей больше, чем предсказывает распределение Пуассона (Zero inflation) –> **ZIP Model** (Zero inflation Poisson Model)

Почему здесь могла быть избыточность дисперсии? И что с ней делать?

-Отскакивающие значения –> **убрать**

-Пропущены ковариаты или взаимодействия предикторов –> **добавить**

- Наличие внутригрупповых корреляций (нарушение независимости выборок) –> **включить другие случайные эффекты**

- Нелинейная взаимосвязь между ковариатами и зависимой переменной –> **GAMM**

- Неверно подобрана связывающая функция –> **заменить**

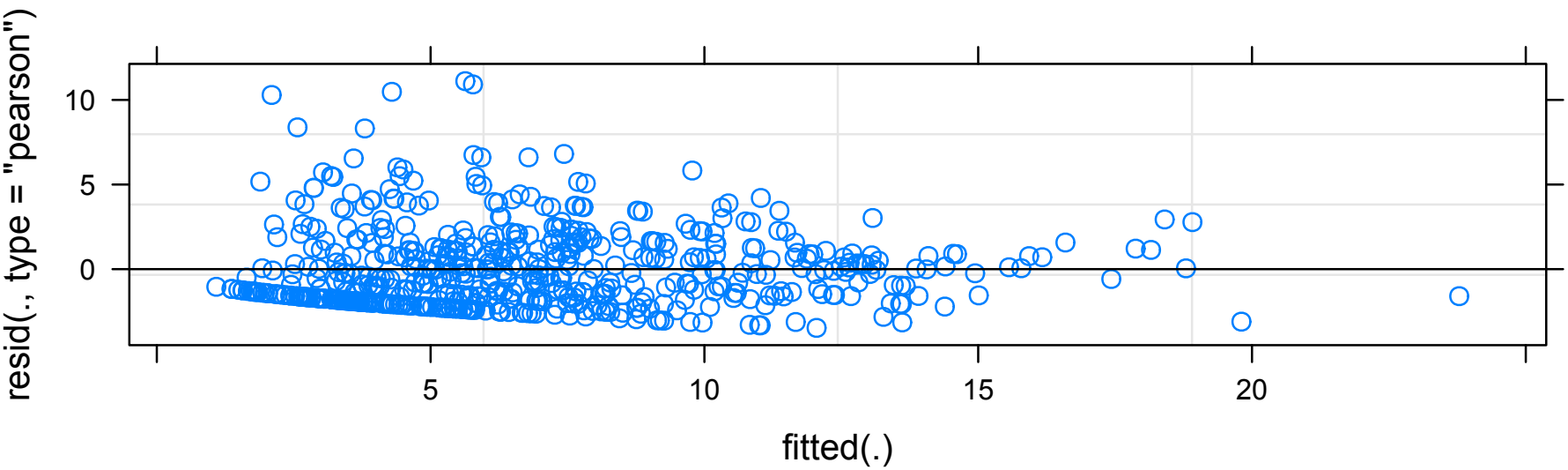
- Количество нулей больше, чем предсказывает распределение Пуассона (Zero inflation) –> **ZIP Model** (Zero inflation Poisson Model)

- Просто большая дисперсия? –> **Модель, основанная на NB распределении**

Диагностика модели

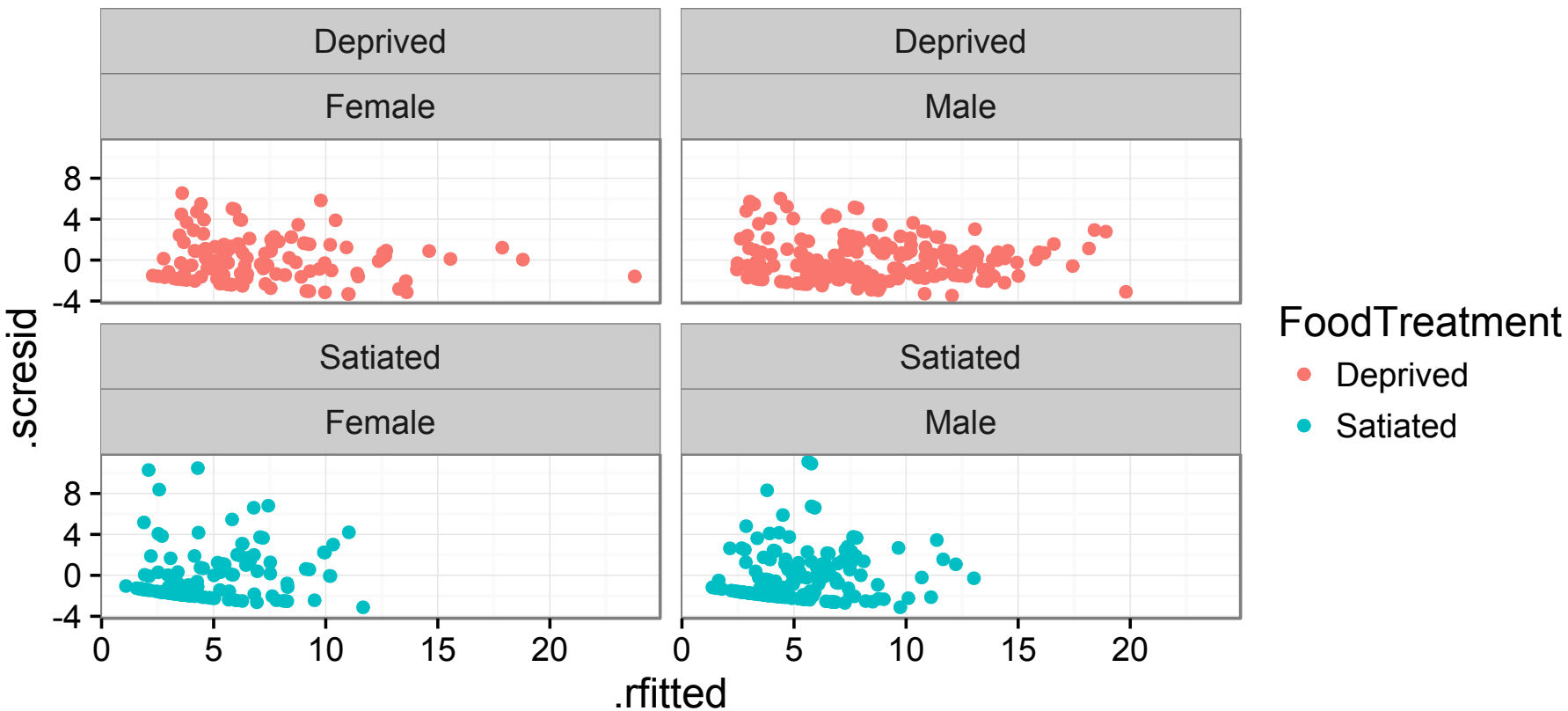
График остатков

```
plot(M1)
```



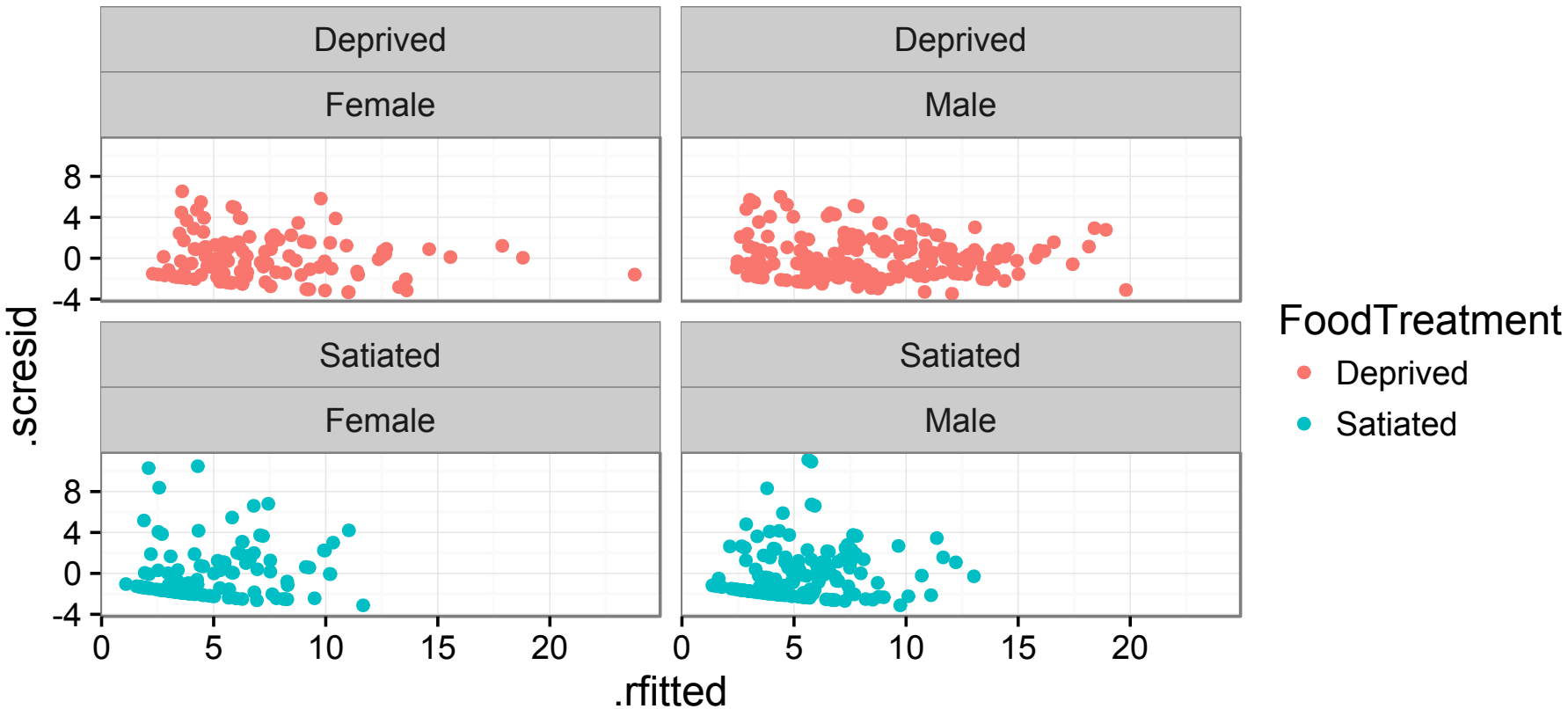
Более удобный график остатков

```
M1_diag <- fortify(M1)
M1_diag$.rfitted <- predict(M1, type = "response")

gg_resid <- ggplot(M1_diag, aes(x = .rfitted, y = .scresid, colour = FoodTrea
gg_resid
```



Более удобный график остатков

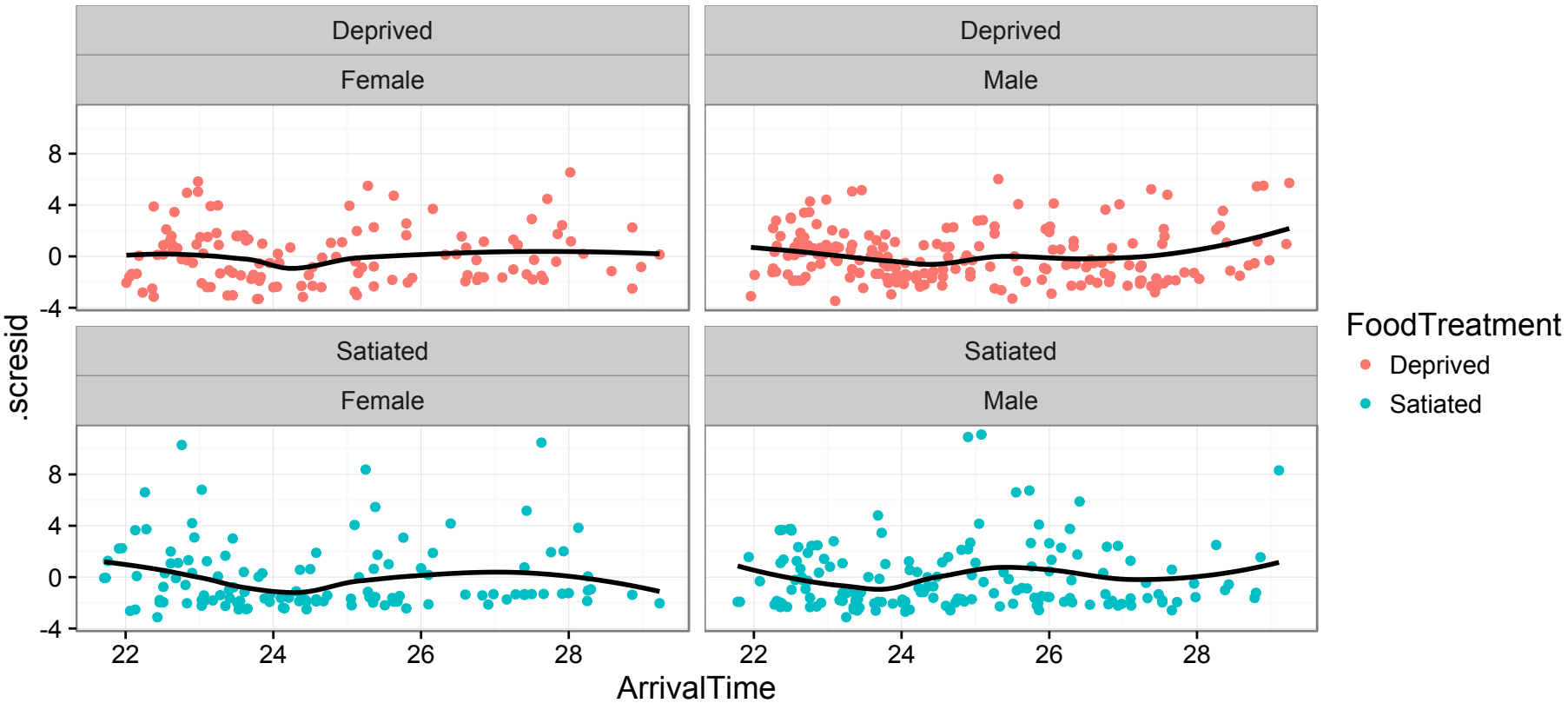
```
M1_diag <- fortify(M1)
M1_diag$.rfitted <- predict(M1, type = "response")

gg_resid <- ggplot(M1_diag, aes(x = .rfitted, y = .scresid, colour = FoodTrea
gg_resid
```



Есть большие остатки

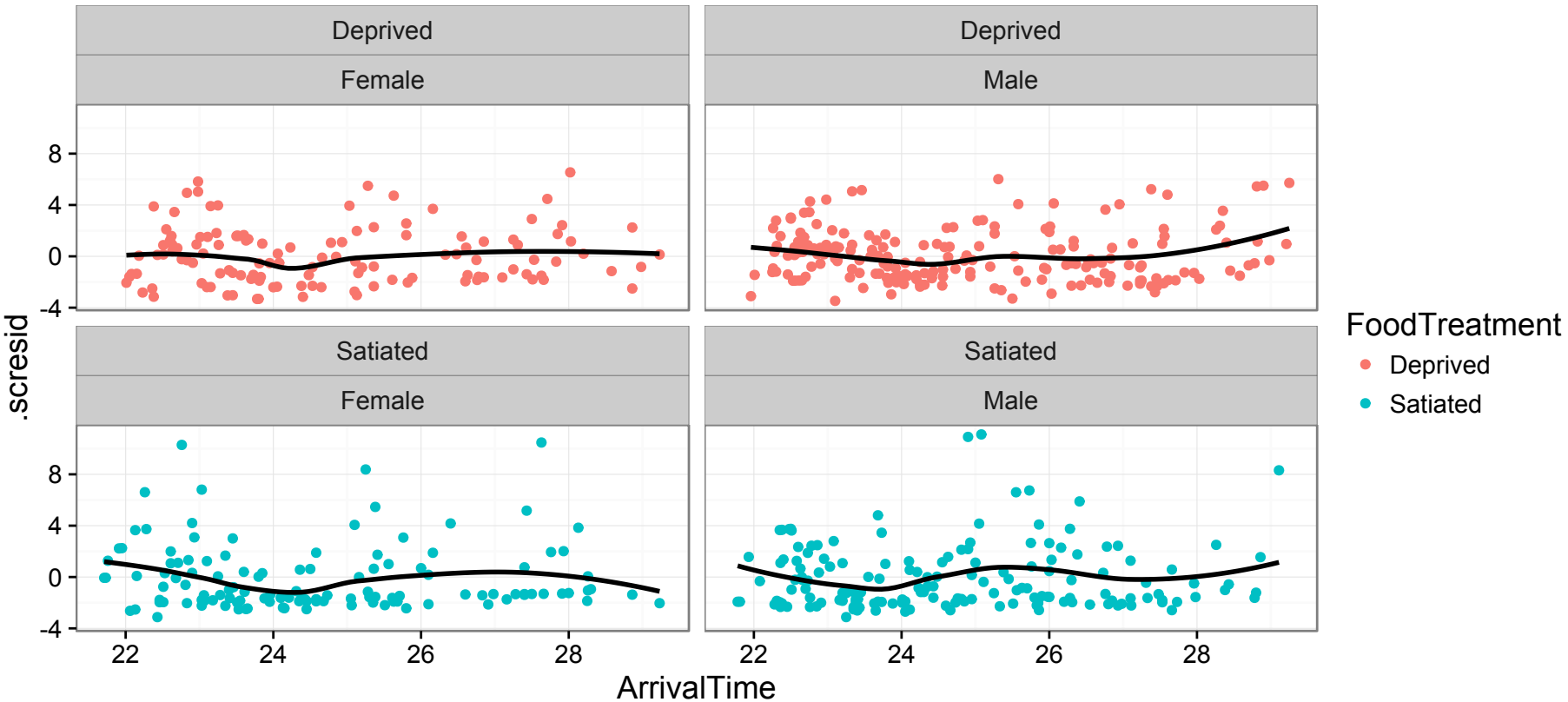
Есть ли еще какие-то паттерны в остатках?

```
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(se = FALSE, color = "black")
```



Есть ли еще какие-то паттерны в остатках?

```
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(se = FALSE, color = "black")
```



Есть намек на нелинейность. Возможно, нужен GAMM

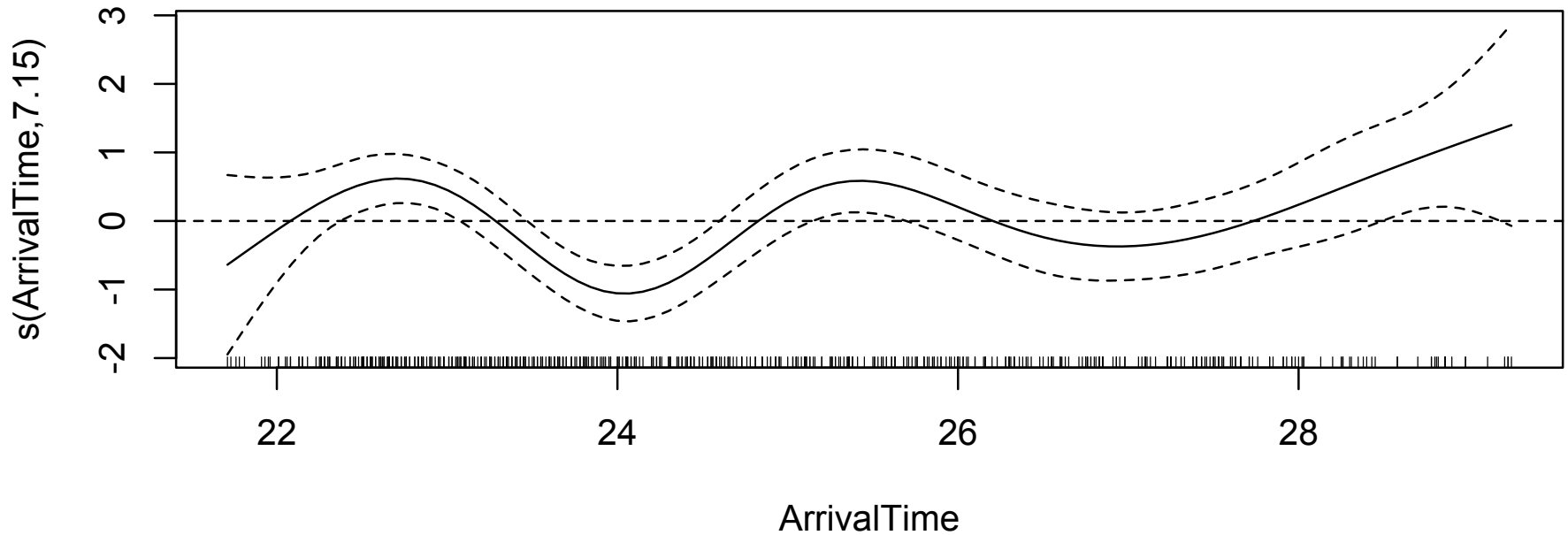
Проверяем, есть ли нелинейный паттерн в остатках

```r
library(mgcv)
nonlin1 <- gam(.scresid ~ s(ArrivalTime), data = M1_diag)
summary(nonlin1)
```

```
#
# Family: gaussian
# Link function: identity
#
# Formula:
# .scresid ~ s(ArrivalTime)
#
# Parametric coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  -0.0111     0.0920   -0.12      0.9
#
# Approximate significance of smooth terms:
#                 edf Ref.df    F   p-value
# s(ArrivalTime) 7.15    8.2 5.04 0.0000039 ***
# ---
# Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
#
# R-sq.(adj) =  0.0618   Deviance explained =  7.3%
# GCV = 5.1414  Scale est. = 5.0715    n = 599
```

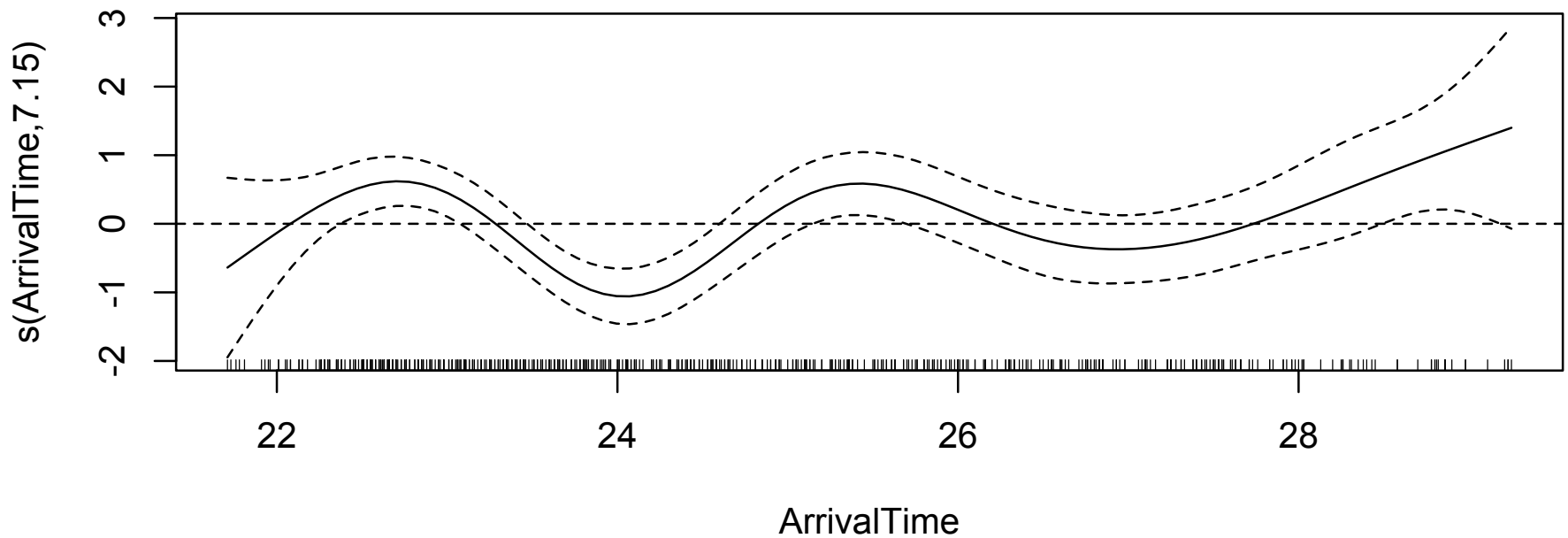
Выявляется нелинейный паттерн

```
plot(nonlin1)
abline(h = 0, lty = 2)
```



Выявляется нелинейный паттерн

```
plot(nonlin1)
abline(h = 0, lty = 2)
```



Совершенно точно нужен GAMM. Но продолжим с GLMM

У нас была сверхдисперсия. Пробуем NB GLMM

$$NCalls_{ij} = e^{\beta_0+\beta_1 SexP_M+\beta_2 FoodT_S+\beta_3 ArrivalT+\beta_4 SexP_M:FoodT_S+\beta_5 SexPM:ArrivalT+log(BroodSize)+a_i+\epsilon_{ij}}$$

- $NCalls_{ij} \sim \boldsymbol{NB}(\mu_{ij}, k)$ — отклик подчиняется отрицательному биномиальному распределению с параметрами μ и k
- $E(NCalls_{ij}) = \mu_{ij}$
- $var(NCalls_{ij}) = \mu_{ij} + \mu_{ij}^2/k$
- $ln(\mu_{ij}) = \eta_{ij}$ — функция связи — логарифм

$$\eta_{ij} = \beta_0 + \beta_1 SexParent_M +$$

$$+\beta_2 FoodTreatment_S + \beta_3 ArrivalTime +$$

$$+\beta_4 SexParent_M : FoodTreatment_S +$$

$$+\beta_5 SexParentM : ArrivalTime + log(BroodSiz$$

$$+a_i + \epsilon_{ij}$$

- $a_i \sim N(0, \sigma^2_{Nest})$ — случайный эффект гнезда (intercept)
- $\epsilon_{ij} \sim N(0, \sigma^2)$ — остатки модели
- i — гнездо
- j — наблюдение

Подберем NB GLMM

```
M2 <- glmer.nb(NCalls ~ SexParent * FoodTreatment +
                 SexParent * ArrivalTime_std +
                 offset(logBroodSize) +
                 (1 | Nest),
               data = Owls)
# # Если эта модель вдруг не сходится, есть обходной маневр.
#Можно попробовать заранее определить k  при помощи внутренней функции.
#В lme4 параметр k называется theta
# th <- lme4:::est_theta(M1)
# M2 <- update(M1, family = negative.binomial(theta=th))
```

Задание:

Проверьте модель с отрицательным биномиальным распределением отклика

- на избыточность дисперсии
- наличие паттернов в остатках
- нелинейность паттернов в остатках

Избыточность дисперсии (Overdispersion)

```
R_M2 <- resid(M2, type = "pearson") # Пирсоновские остатки
N <- nrow(Owls) # Объем выборки
p <- length(fixef(M2)) + 1 + 1  # Число параметров в модели (тета и сигма для
df <- (N - p) # число степенейсвободы
fi <- sum(R_M2^2) /df  #Величина fi показывает во сколько раз в среднем sigma
fi
```

```
# [1] 0.851
```

Избыточность дисперсии (Overdispersion)

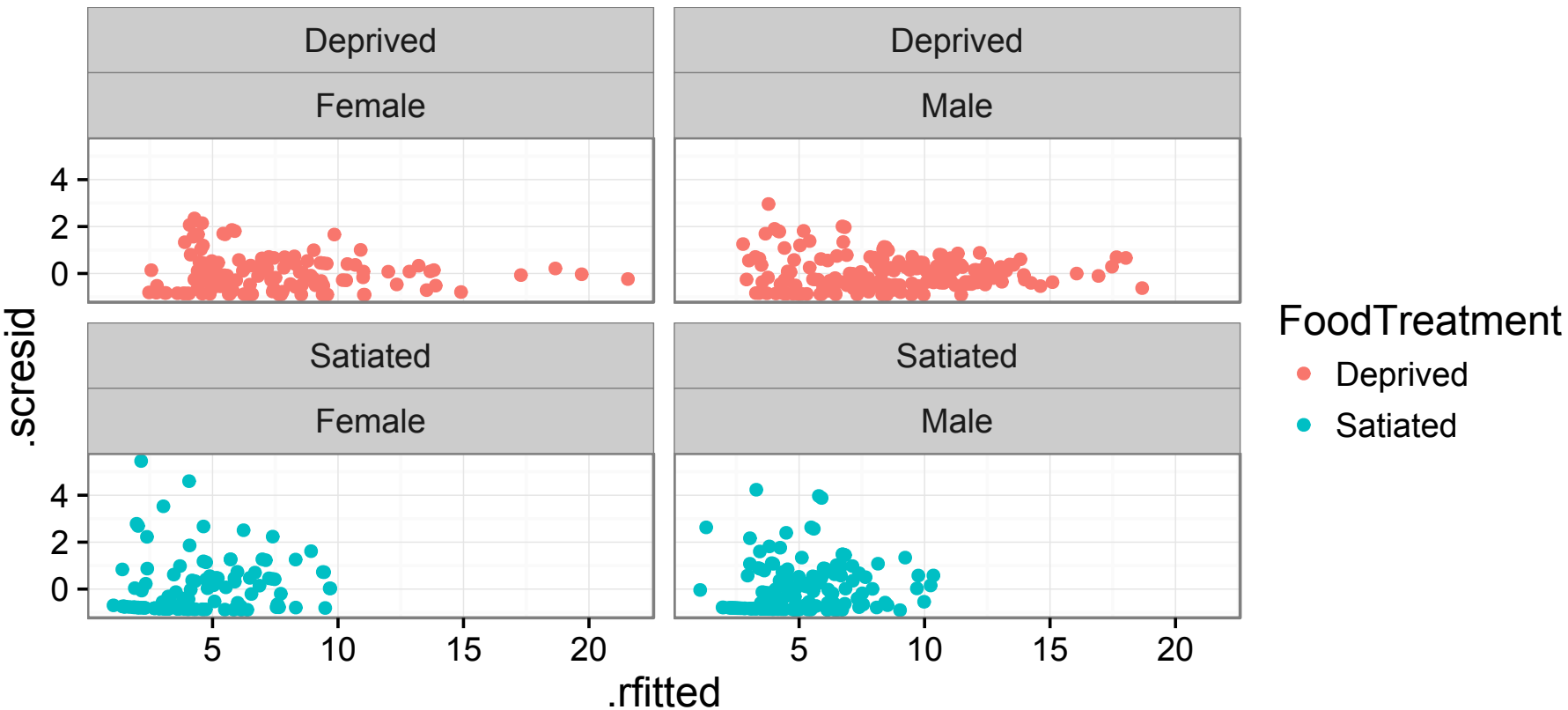
```
R_M2 <- resid(M2, type = "pearson") # Пирсоновские остатки
N <- nrow(Owls) # Объем выборки
p <- length(fixef(M2)) + 1 + 1  # Число параметров в модели (тета и сигма для
df <- (N - p) # число степенейсвободы
fi <- sum(R_M2^2) /df  #Величина fi показывает во сколько раз в среднем sigma
fi
```

```
# [1] 0.851
```

Сносно

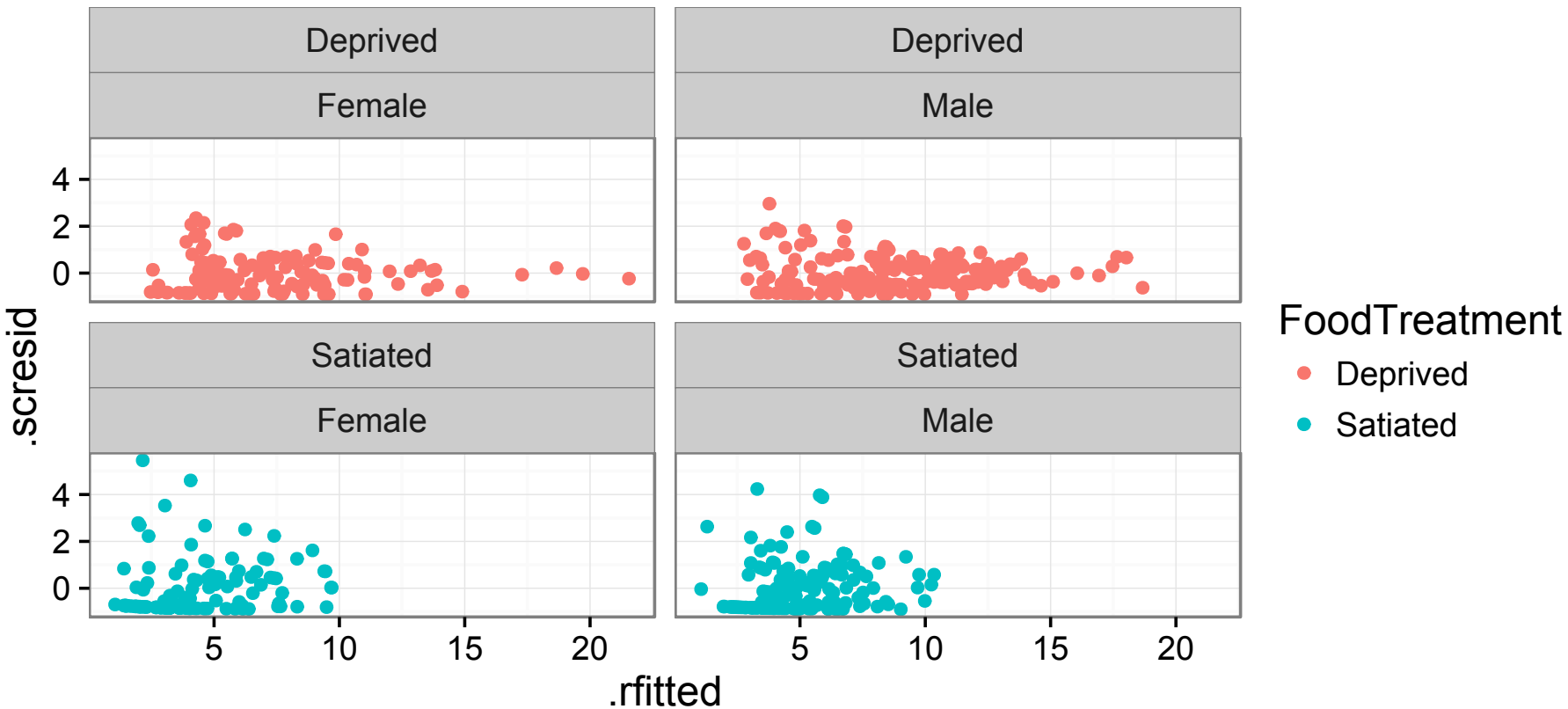
Диагностика модели с отр. биномиальным рспределением остатков

```
M2_diag <- fortify(M2)
M2_diag$.rfitted <- predict(M2, type = "response")
gg_resid <- ggplot(M2_diag, aes(x = .rfitted, y = .scresid, colour = FoodTrea
gg_resid
```



Диагностика модели с отр. биномиальным рспределением остатков

```
M2_diag <- fortify(M2)
M2_diag$.rfitted <- predict(M2, type = "response")
gg_resid <- ggplot(M2_diag, aes(x = .rfitted, y = .scresid, colour = FoodTrea
gg_resid
```

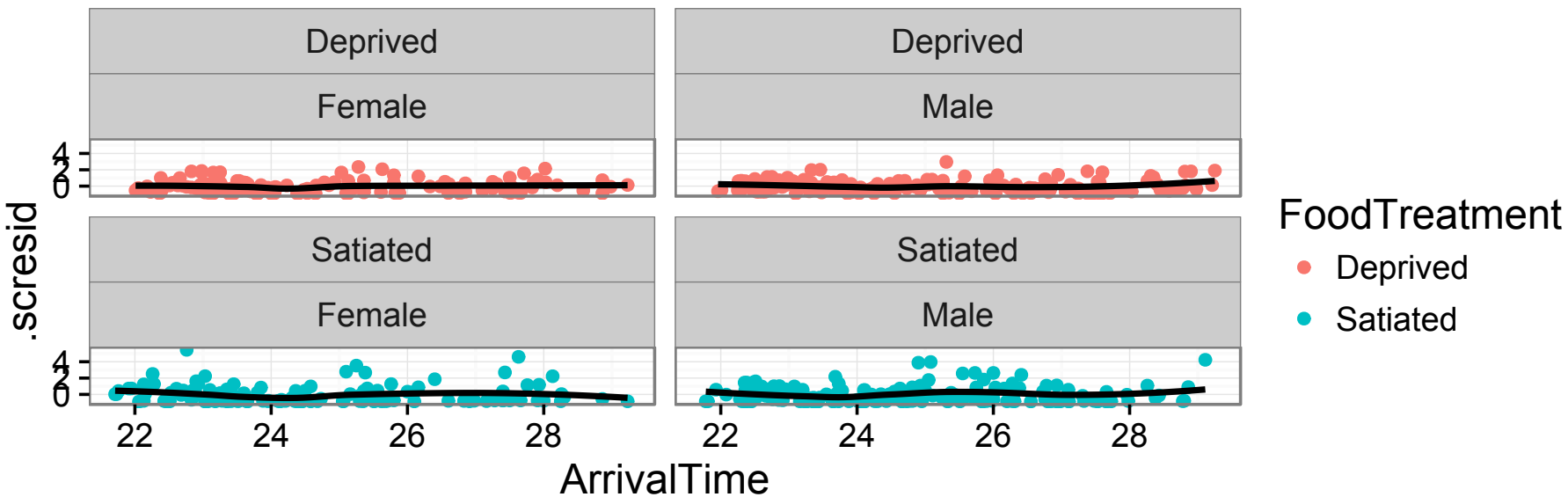


Есть большие остатки

Есть ли еще какие-то паттерны в остатках?

Может быть паттерны в остатках исчезли от того, что мы использовали другую GLMM?

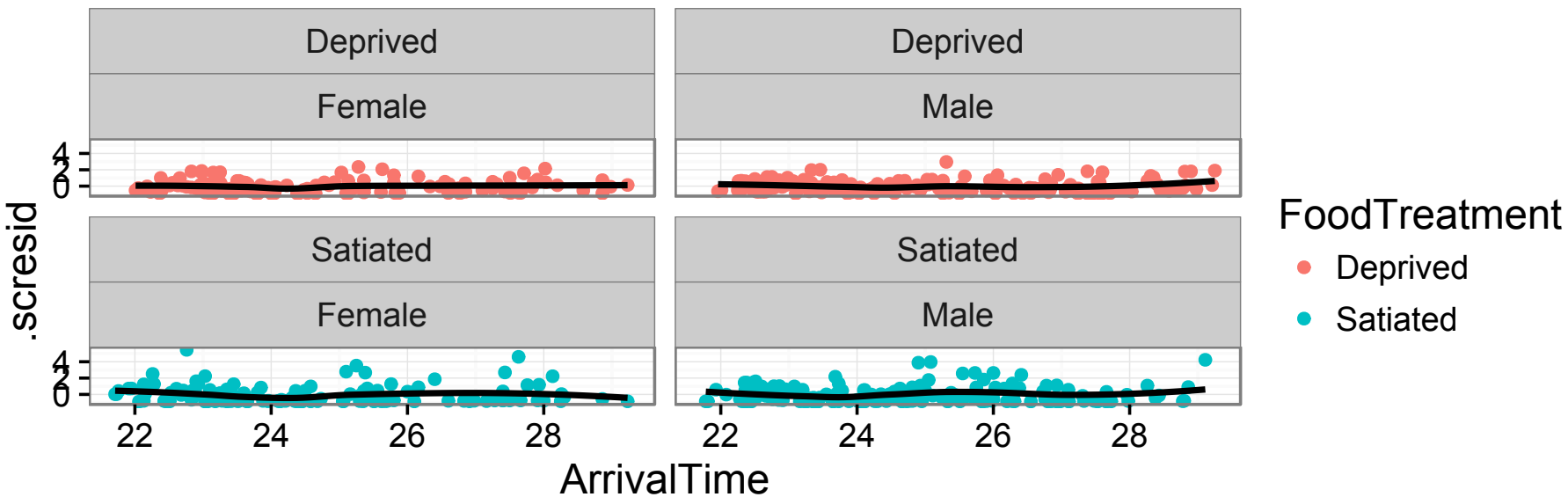
```
# gg_resid %+% aes(x = ArrivalTime)
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(se = FALSE, color = "black")
```



Есть ли еще какие-то паттерны в остатках?

Может быть паттерны в остатках исчезли от того, что мы использовали другую GLMM?

```
# gg_resid %+% aes(x = ArrivalTime)
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(se = FALSE, color = "black")
```



Подозрительно. Возможно, нужен GAMM

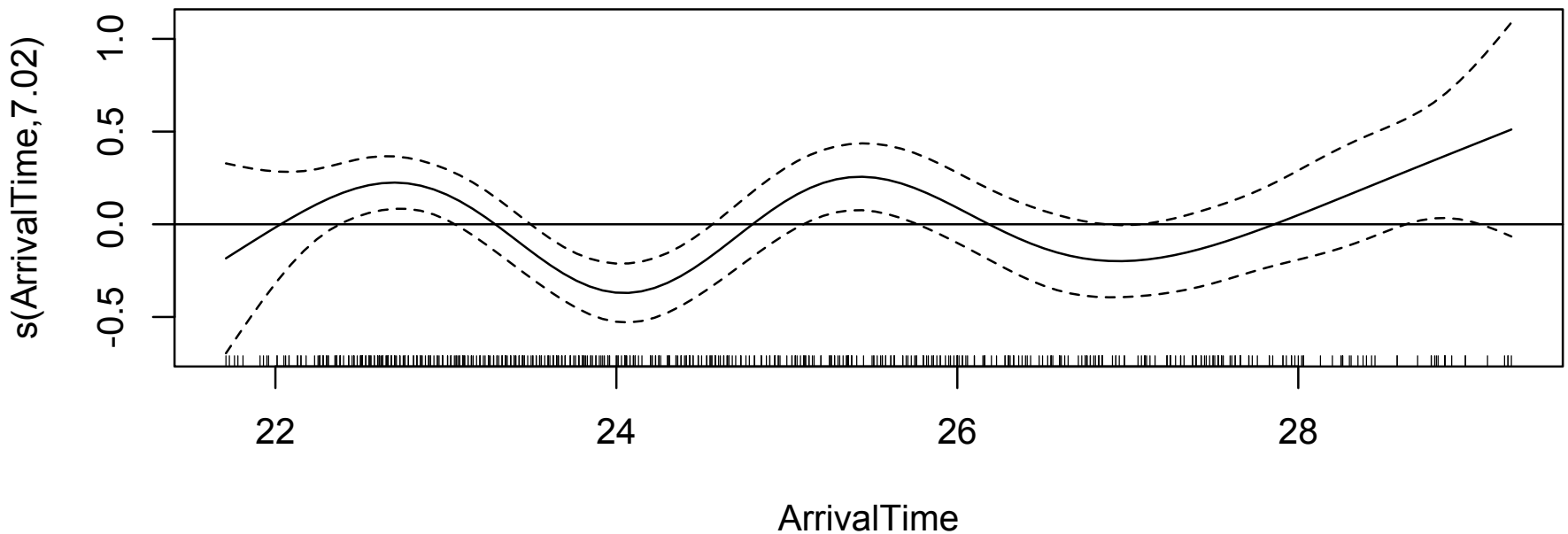
Проверяем, есть ли нелинейные паттерны

```
nonlin2 <- gam(.scresid ~ s(ArrivalTime), data = M2_diag)
summary(nonlin2)
```

```
#
# Family: gaussian
# Link function: identity
#
# Formula:
# .scresid ~ s(ArrivalTime)
#
# Parametric coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -0.00121    0.03642   -0.03     0.97
#
# Approximate significance of smooth terms:
#                 edf Ref.df    F p-value
# s(ArrivalTime) 7.02    8.1 4.55 0.00002 ***
# ---
# Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
#
# R-sq.(adj) =  0.0552   Deviance explained = 6.63%
# GCV = 0.80535  Scale est. = 0.79456   n = 599
```

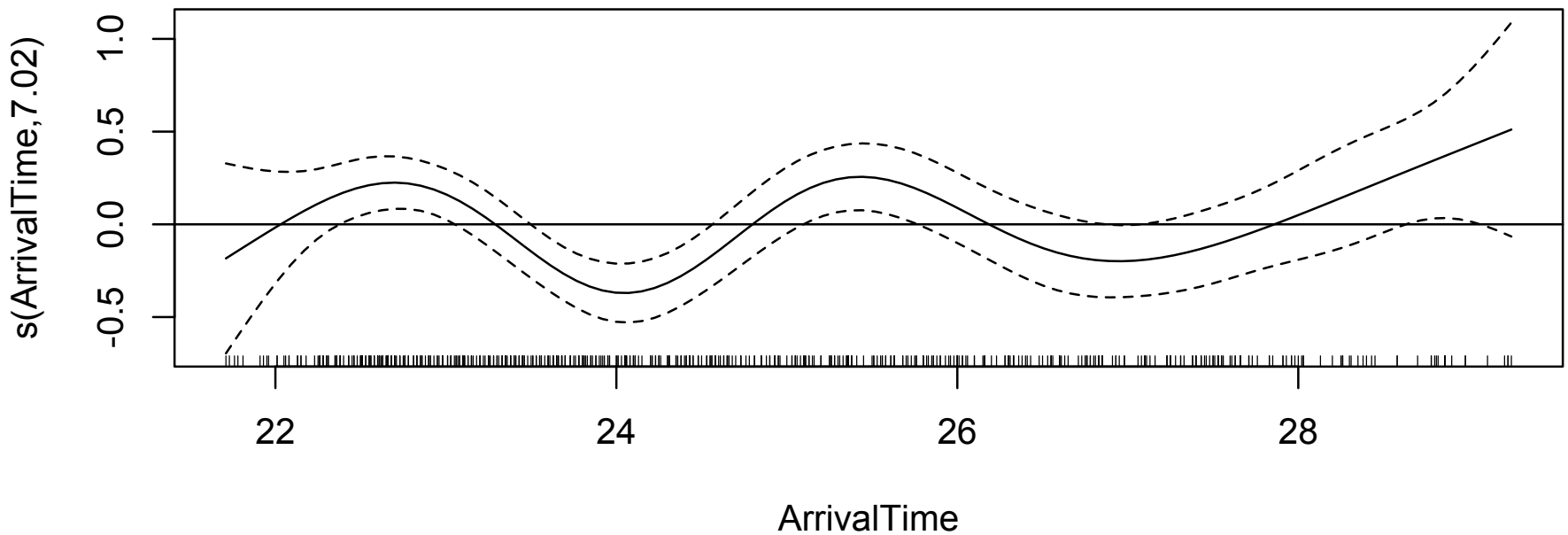
Нелинейный паттерн в остатках остался

```
plot(nonlin2)
abline(h = 0)
```



Нелинейный паттерн в остатках остался

```
plot(nonlin2)
abline(h = 0)
```



Совершенно точно нужен GAMM

Подбор оптимальной модели

Все ли достоверно?

```
summary(M2)
```

```
# Generalized linear mixed model fit by maximum likelihood (Laplace
#   Approximation) [glmerMod]
#  Family: Negative Binomial(0.885)  ( log )
# Formula:
# NCalls ~ SexParent * FoodTreatment + SexParent * ArrivalTime_std +
#     offset(logBroodSize) + (1 | Nest)
#    Data: Owls
#
#      AIC      BIC   logLik deviance df.resid
#     3479     3514    -1732     3463      591
#
# Scaled residuals:
#    Min     1Q Median     3Q    Max
# -0.906 -0.779 -0.202  0.437  5.458
#
# Random effects:
#  Groups Name        Variance Std.Dev.
#  Nest   (Intercept) 0.109    0.33
# Number of obs: 599, groups:  Nest, 27
#
```

Можно ли что-то выкинуть

```
drop1(M2, test = "Chi")
```

```
# Single term deletions
#
# Model:
# NCalls ~ SexParent * FoodTreatment + SexParent * ArrivalTime_std +
#     offset(logBroodSize) + (1 | Nest)
#                            Df  AIC   LRT Pr(Chi)
# <none>                        3479
# SexParent:FoodTreatment     1 3478 0.783    0.38
# SexParent:ArrivalTime_std   1 3478 0.272    0.60
```

Можно ли что-то выкинуть

```
drop1(M2, test = "Chi")
```

```
# Single term deletions
#
# Model:
# NCalls ~ SexParent * FoodTreatment + SexParent * ArrivalTime_std +
#     offset(logBroodSize) + (1 | Nest)
#                            Df  AIC   LRT Pr(Chi)
# <none>                        3479
# SexParent:FoodTreatment     1 3478 0.783    0.38
# SexParent:ArrivalTime_std   1 3478 0.272    0.60
```

Если выкинуть взаимодействия, модель не станет хуже

Выкидываем одно взаимодействие

```
M3 <- update(M2, .~.-SexParent:ArrivalTime_std)
drop1(M3, test = "Chisq")
```

```
# Single term deletions
#
# Model:
# NCalls ~ SexParent + FoodTreatment + ArrivalTime_std + (1 | Nest) +
#     SexParent:FoodTreatment + offset(logBroodSize)
#                         Df  AIC   LRT   Pr(Chi)
# <none>                     3478
# ArrivalTime_std          1 3496 20.47 0.0000061 ***
# SexParent:FoodTreatment  1 3476  0.75      0.39
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Выкидываем одно взаимодействие

```
M3 <- update(M2, .~.-SexParent:ArrivalTime_std)
drop1(M3, test = "Chisq")
```

```
# Single term deletions
#
# Model:
# NCalls ~ SexParent + FoodTreatment + ArrivalTime_std + (1 | Nest) +
#     SexParent:FoodTreatment + offset(logBroodSize)
#                         Df  AIC   LRT   Pr(Chi)
# <none>                     3478
# ArrivalTime_std          1 3496 20.47 0.0000061 ***
# SexParent:FoodTreatment  1 3476  0.75      0.39
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

теперь можно выкинуть второе

Выкидываем второе взаимодействие

```
M4 <- update(M3, .~.-SexParent:FoodTreatment)
drop1(M4, test = "Chisq")
```

```
# Single term deletions
#
# Model:
# NCalls ~ SexParent + FoodTreatment + ArrivalTime_std + (1 | Nest) +
#     offset(logBroodSize)
#                 Df  AIC  LRT       Pr(Chi)
# <none>             3476
# SexParent        1 3475  0.4           0.5
# FoodTreatment    1 3513 39.0 0.00000000043 ***
# ArrivalTime_std  1 3495 20.3 0.00000651759 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Выкидываем второе взаимодействие

```
M4 <- update(M3, .~.-SexParent:FoodTreatment)
drop1(M4, test = "Chisq")
```

```
# Single term deletions
#
# Model:
# NCalls ~ SexParent + FoodTreatment + ArrivalTime_std + (1 | Nest) +
#     offset(logBroodSize)
#                 Df  AIC  LRT       Pr(Chi)
# <none>             3476
# SexParent        1 3475  0.4           0.5
# FoodTreatment    1 3513 39.0 0.00000000043 ***
# ArrivalTime_std  1 3495 20.3 0.00000651759 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

теперь можно выкинуть пол родителя

Финальная модель

```
M5 <- update(M4, .~.-SexParent)
drop1(M5, test = "Chisq")
```

```
# Single term deletions
#
# Model:
# NCalls ~ FoodTreatment + ArrivalTime_std + (1 | Nest) + offset(logBroodSize
#                 Df  AIC  LRT       Pr(Chi)
# <none>             3475
# FoodTreatment    1 3513 39.9 0.00000000027 ***
# ArrivalTime_std  1 3493 20.1 0.00000746455 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Второй способ подбора оптимальной модели - AIC

```
AIC(M2, M3, M4, M5)
```

```
#    df  AIC
# M2  8 3479
# M3  7 3478
# M4  6 3476
# M5  5 3475
```

Второй способ подбора оптимальной модели - AIC

```
AIC(M2, M3, M4, M5)
```

```
#    df  AIC
# M2  8 3479
# M3  7 3478
# M4  6 3476
# M5  5 3475
```

Выбираем модель с минимальным AIC

Модель изменилась. Нужно повторить диагностику

Избыточность дисперсии (Overdispersion)

```
R_M5 <- resid(M5, type = "pearson") # Пирсоновские остатки
N <- nrow(Owls) # Объем выборки
p <- length(fixef(M5)) + 1 + 1  # Число параметров в модели (тета и сигма для
df <- (N - p) # число степенейсвободы
fi <- sum(R_M5^2) /df  #Величина fi показывает во сколько раз в среднем sigma
fi
```

```
# [1] 0.844
```

Модель изменилась. Нужно повторить диагностику

Избыточность дисперсии (Overdispersion)

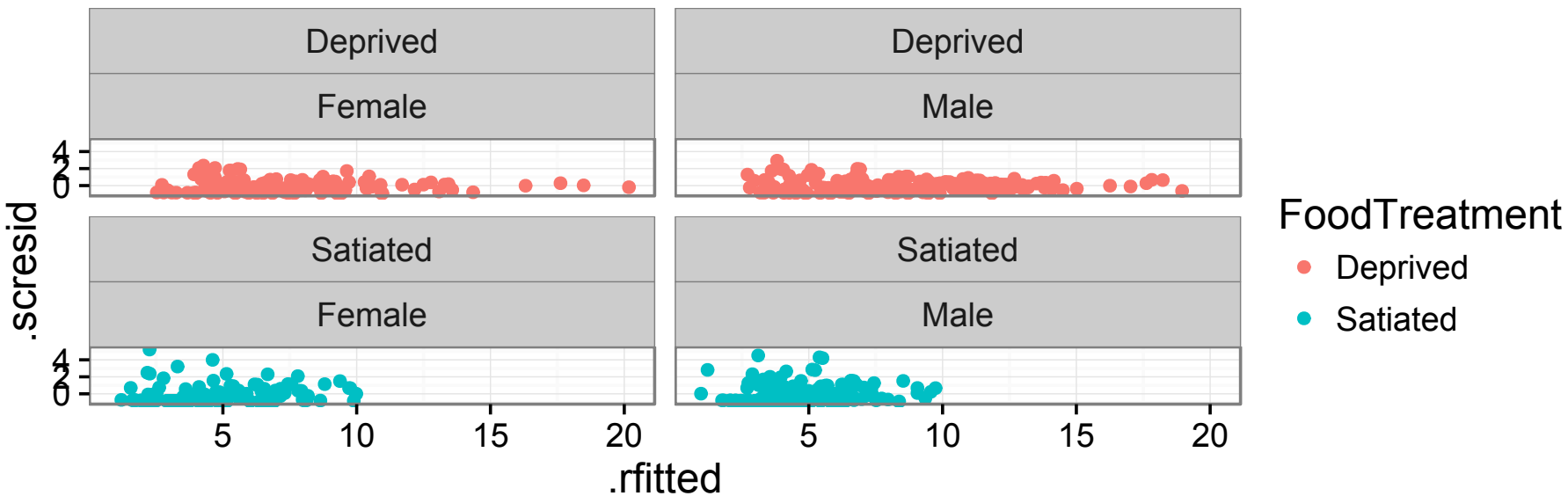
```
R_M5 <- resid(M5, type = "pearson") # Пирсоновские остатки
N <- nrow(Owls) # Объем выборки
p <- length(fixef(M5)) + 1 + 1  # Число параметров в модели (тета и сигма для
df <- (N - p) # число степенейсвободы
fi <- sum(R_M5^2) /df  #Величина fi показывает во сколько раз в среднем sigma
fi
```

```
# [1] 0.844
```

Сносно

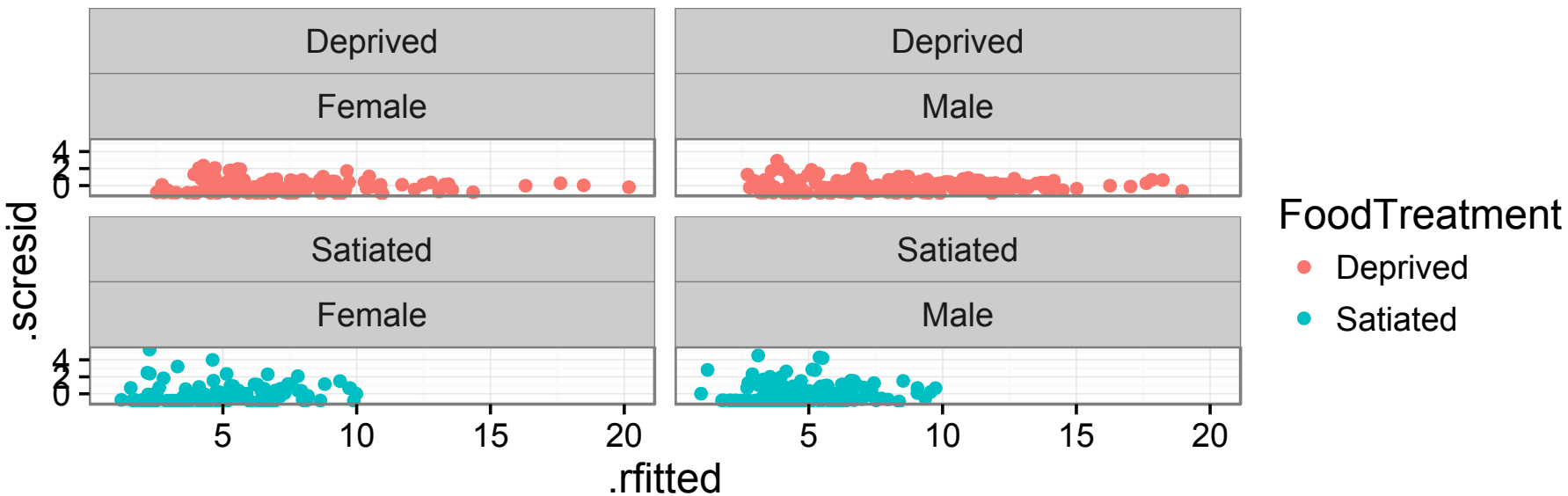
Диагностика отр. биномиальной модели

```
M5_diag <- fortify(M5)
M5_diag$.rfitted <- predict(M5, type = "response")
gg_resid <- ggplot(M5_diag, aes(x = .rfitted, y = .scresid, colour = FoodTrea
gg_resid
```



Диагностика отр. биномиальной модели

```
M5_diag <- fortify(M5)
M5_diag$.rfitted <- predict(M5, type = "response")
gg_resid <- ggplot(M5_diag, aes(x = .rfitted, y = .scresid, colour = FoodTrea
gg_resid
```

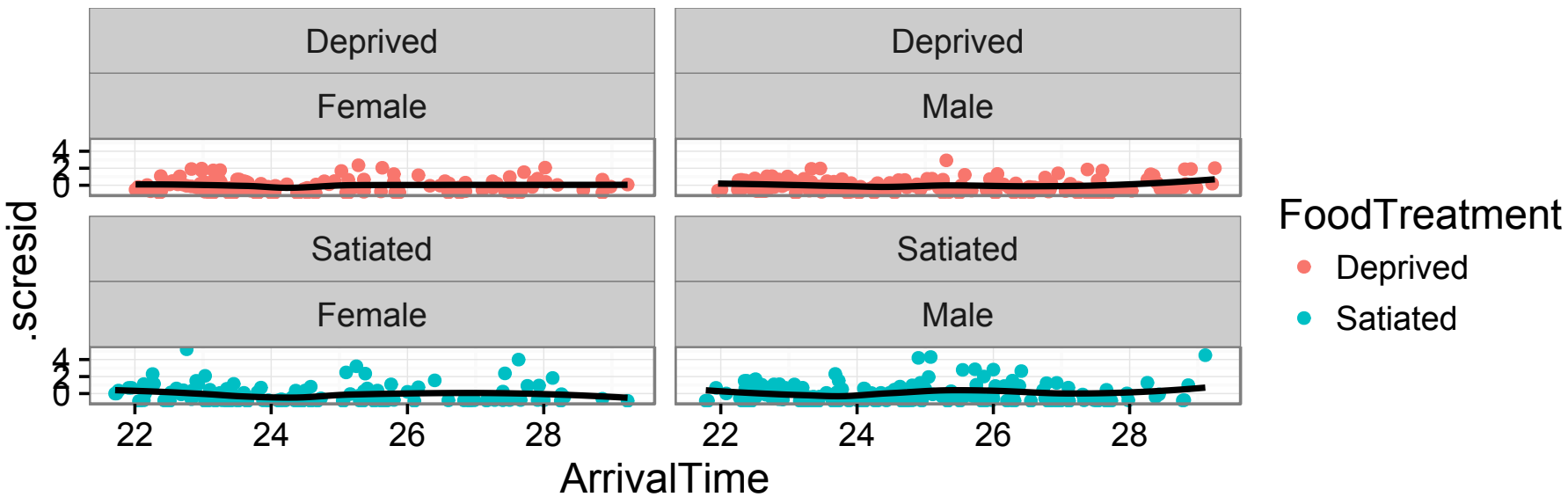


Есть большие остатки

Есть ли еще какие-то паттерны в остатках?

Может быть паттерны в остатках исчезли от того, что мы использовали другую GLMM?

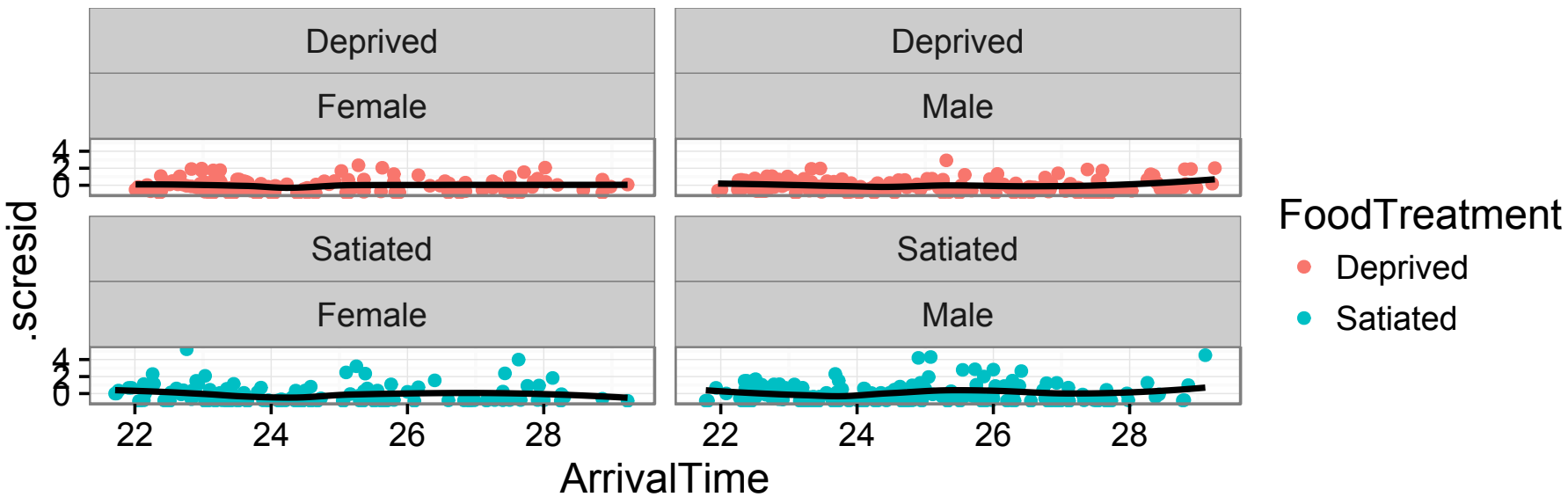
```
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(se = FALSE, color = "black")
```



Есть ли еще какие-то паттерны в остатках?

Может быть паттерны в остатках исчезли от того, что мы использовали другую GLMM?

```
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(se = FALSE, color = "black")
```



Подозрительно. Возможно, нужен GAMM

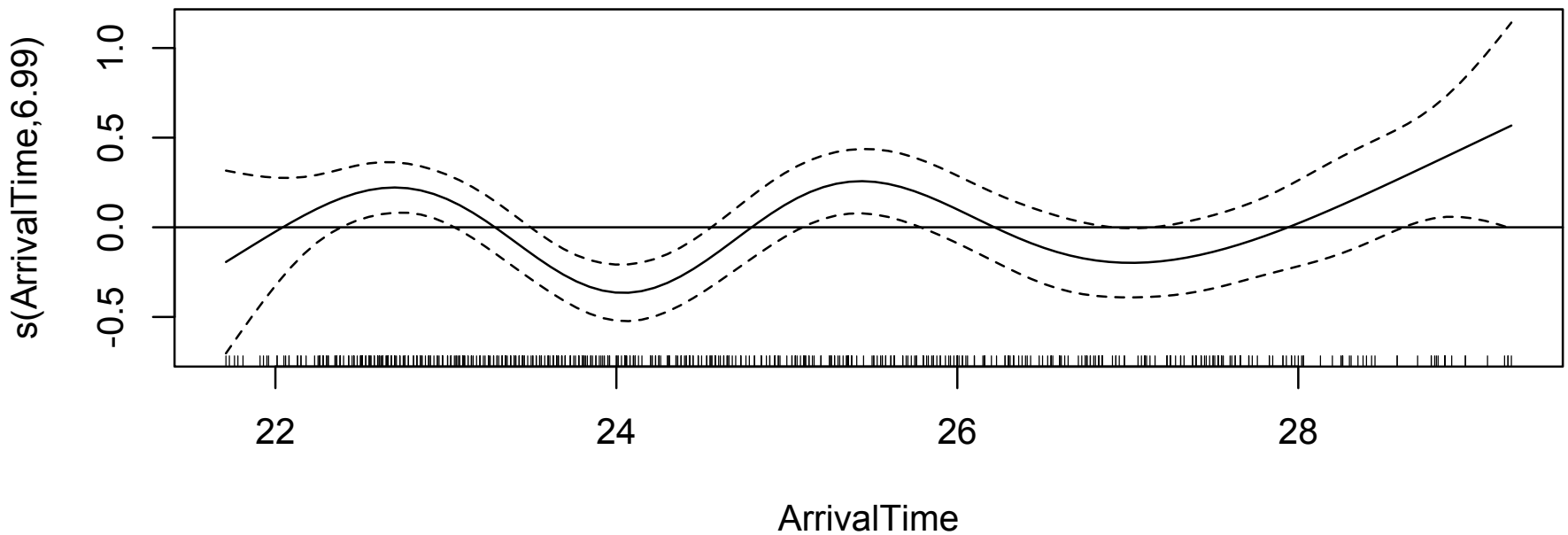
Проверяем, есть ли нелинейные паттерны

```
nonlin5 <- gam(.scresid ~ s(ArrivalTime), data = M5_diag)
summary(nonlin5)
```

```
#
# Family: gaussian
# Link function: identity
#
# Formula:
# .scresid ~ s(ArrivalTime)
#
# Parametric coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -0.00119    0.03636   -0.03     0.97
#
# Approximate significance of smooth terms:
#                 edf Ref.df    F  p-value
# s(ArrivalTime) 6.99   8.07 4.61 0.000017 ***
# ---
# Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
#
# R-sq.(adj) =  0.0559   Deviance explained = 6.69%
# GCV = 0.80266  Scale est. = 0.79195   n = 599
```

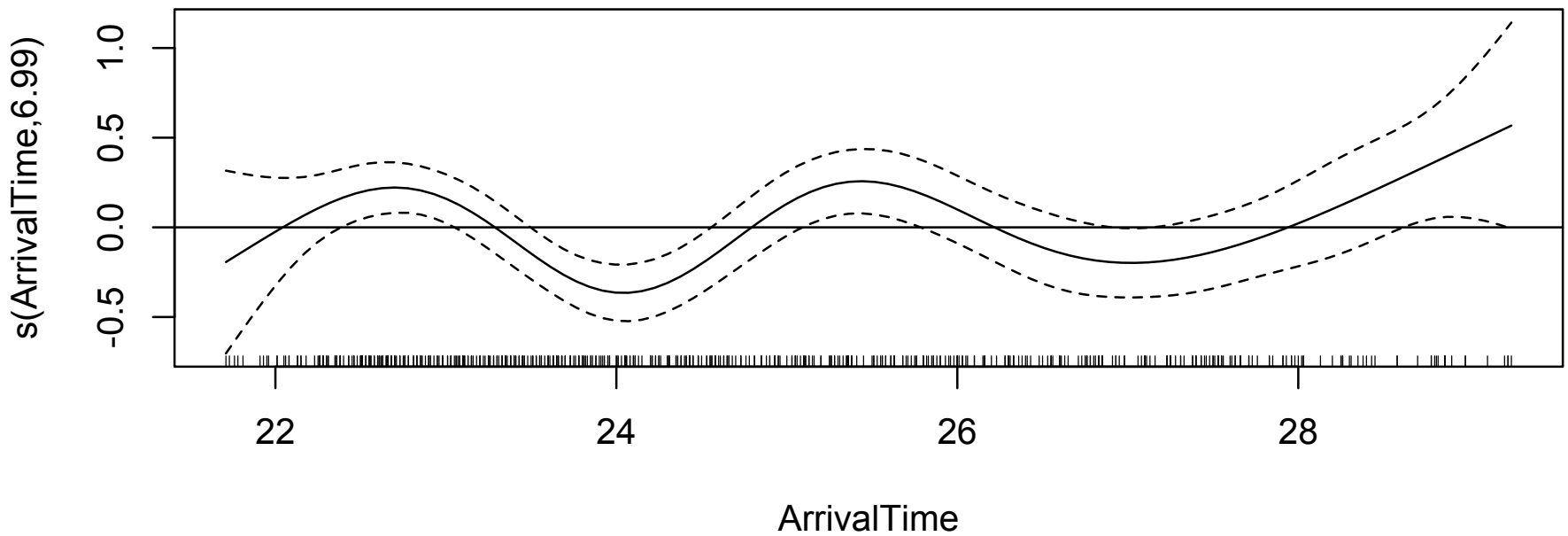
Нелинейный паттерн никуда не делся...

```
plot(nonlin5)
abline(h = 0)
```



Нелинейный паттерн никуда не делся...

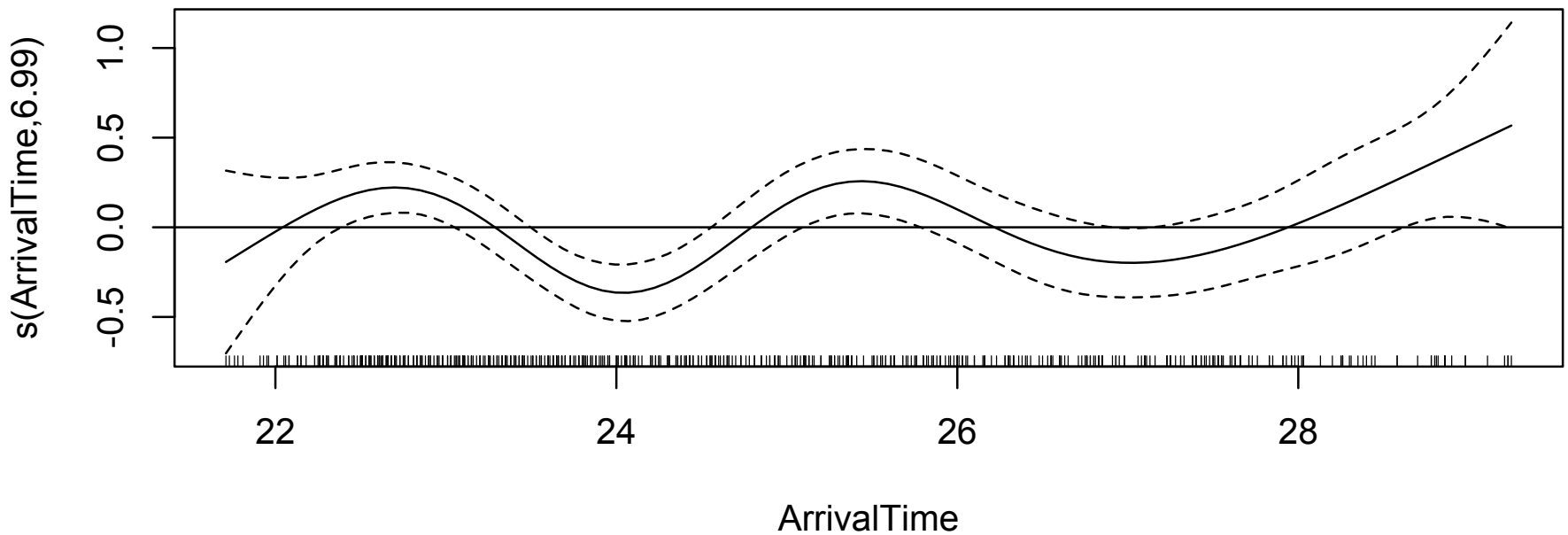
```
plot(nonlin5)
abline(h = 0)
```



Совершенно точно нужен GAMM

Нелинейный паттерн никуда не делся...

```
plot(nonlin5)
abline(h = 0)
```



Совершенно точно нужен GAMM Но мы продолжим

Финальная GLMM, которую мы получили, выглядит так

$$NCalls_{ij} = e^{\beta_0 + \beta_1 SexP_M + \beta_2 FoodT_S + \beta_3 ArrivalT + log(BroodSize) + a_i + \epsilon_{ij}}$$

- $NCalls_{ij} \sim \boldsymbol{NB}(\mu_{ij}, k)$ — отклик подчиняется отрицательному биномиальному распределению с параметрами μ и k
- $E(NCalls_{ij}) = \mu_{ij}$
- $var(NCalls_{ij}) = \mu_{ij} + \mu_{ij}^2/k$
- $ln(\mu_{ij}) = \eta_{ij}$ — функция связи — логарифм

$$\eta_{ij} = \beta_0 + \beta_1 SexParent_M + \beta_2 FoodTreatment_S + \beta_3 ArrivalTime + log(BroodSize) +$$

$$+ a_i + \epsilon_{ij}$$

- $a_i \sim N(0, \sigma^2_{Nest})$ — случайный эффект гнезда (intercept)
- $\epsilon_{ij} \sim N(0, \sigma^2)$ — остатки модели
- i — гнездо
- j — наблюдение

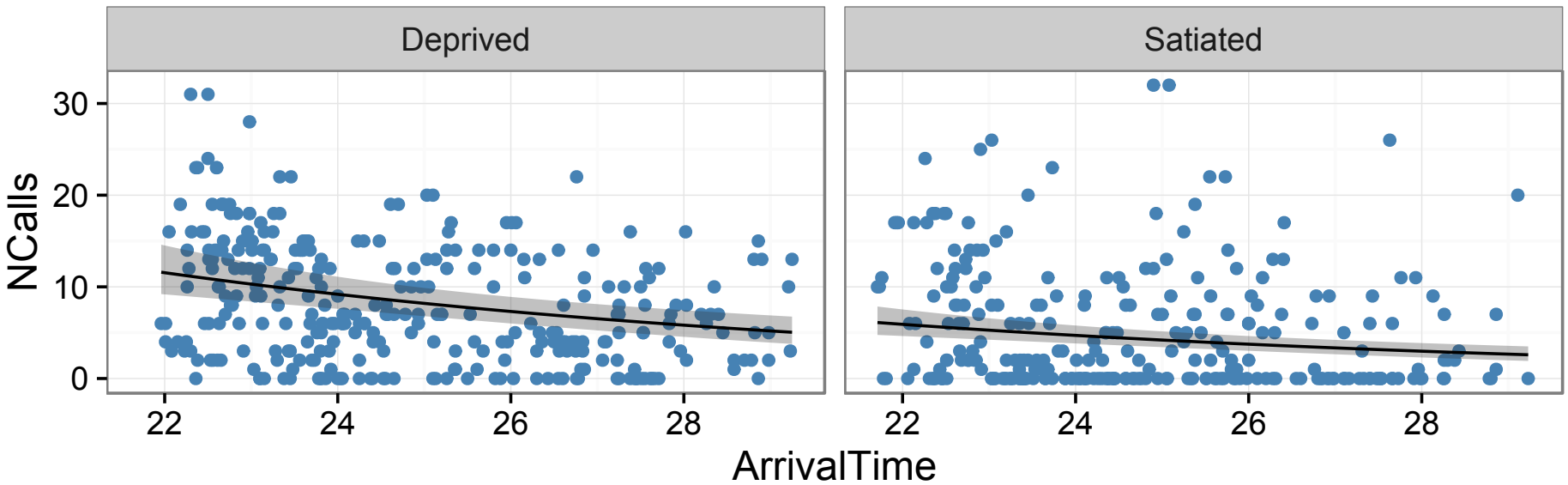
Готовим данные для графика модели

```
library(plyr)
NewData <- ddply(Owls, .variables = .(FoodTreatment),
                 summarise,
                 ArrivalTime_std = seq(min(ArrivalTime_std),
                                       max(ArrivalTime_std), length = 100))

NewData$ArrivalTime <- NewData$ArrivalTime_std * sd(Owls$ArrivalTime) +
  mean(Owls$ArrivalTime)
```

Предсказания и ошибки

```
# Модельная матрица
X <- model.matrix(~ FoodTreatment + ArrivalTime_std, data = NewData)
# К предсказанным значениям нужно прибавить оффсет. Мы будем делать предсказа
NewData$Pred <- X %*% fixef(M5) + log(mean(Owls$BroodSize))
# Стандартные ошибки предсказаний
NewData$SE <- sqrt(diag(X %*% vcov(M5) %*% t(X)))
```

График предсказанных значений



Код для графика предсказанных значений

```
ggplot() +
  geom_point(data = Owls,
             aes(x = ArrivalTime, y = NCalls), colour = "steelblue") +
  geom_ribbon(data = NewData, aes(x = ArrivalTime, ymax = exp(Pred + 1.96 * SE
  geom_line(data = NewData, aes(x = ArrivalTime, y = exp(Pred), group = FoodT
  facet_wrap(~FoodTreatment)
```

Take home messages

- В случае счетных зависимых перменных (неотрицательных целочисленных величин) применяются модели, основанные на распределении Пуассона или отрицаетльном биномиальном распределении.
- При проверке на избыточность дисперсии таких смешанных линейных моделей, нужно учитывать дополнительные параметры: дисперсию связанную со случайными факторами, и параметр тета для отрицательного биномиального распределения
- Нелинейные паттерны в остатках иногда могут быть причиной избыточности (или недостатка) дисперсии.

Дополнительные ресурсы

- Crawley, M.J. (2007). The R Book (Wiley).
- Zuur, A.F., Ieno, E.N., Walker, N., Saveliev, A.A., and Smith, G.M. (2009). Mixed Effects Models and Extensions in Ecology With R (Springer).