

Смешанные линейные модели для счетных данных

Линейные модели...

Марина Варфоломеева, Вадим Хайтов
Осень 2022

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 зависит призывный крик совят?

27 семейств сов в западной Швейцарии наблюдали с июня по август 1997.



Young Barn Owls in Tree Nest by Hunter Desportes on Flickr

В день наблюдений совятам либо давали дополнительную подкормку (сытые), либо забирали остатки пищи из гнезда (голодные).

Оба варианта манипуляций использовали в каждом из гнезд в случайном порядке.

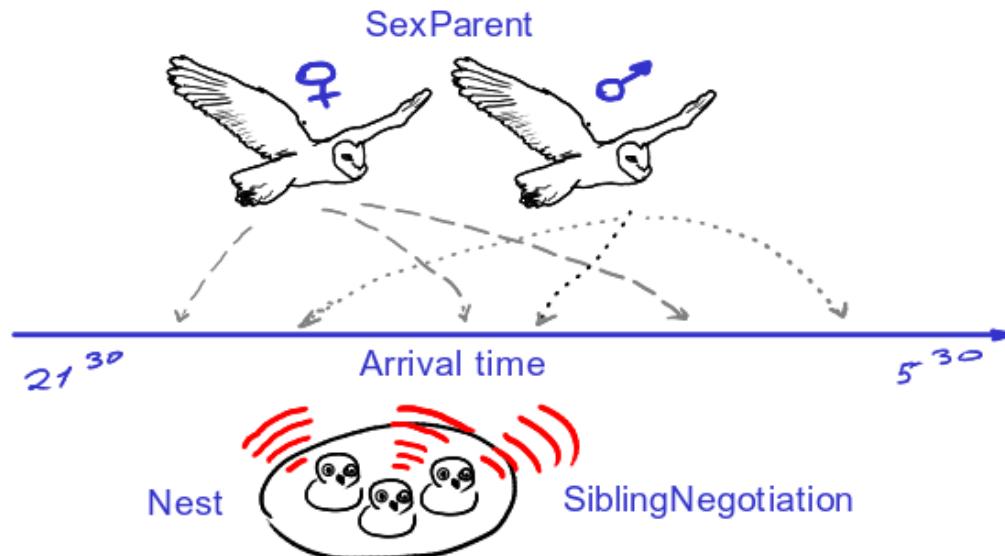
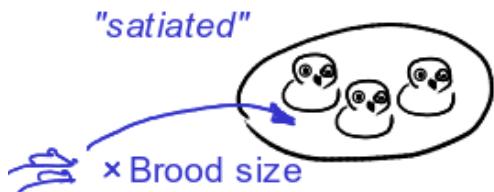
С 21:30 до 5:30 утра записывали звуки и видео.

Данные из [Roulin & Bersier 2007](#), пример из кн. Zuur et al., 2007

Roulin, A. and Bersier, L.F., 2007. Nestling barn owls beg more intensely in the presence of their mother than in the presence of their father. *Animal Behaviour*, 74(4), pp.1099-1106.

От чего зависит призывный крик совят?

FoodTreatment (at 9:00)



- **SiblingNegotiation** — число звуков в течение 15 минут до прибытия родителя
- **FoodTreatment** — сытые или голодные
- **SexParent** — пол родителя
- **ArrivalTime** — время прибытия родителя
- **Nest** — гнездо

Знакомство с данными

```
Owls <- read.delim("data/Roulin_Bersier_2007_Owls.csv")
str(Owls)

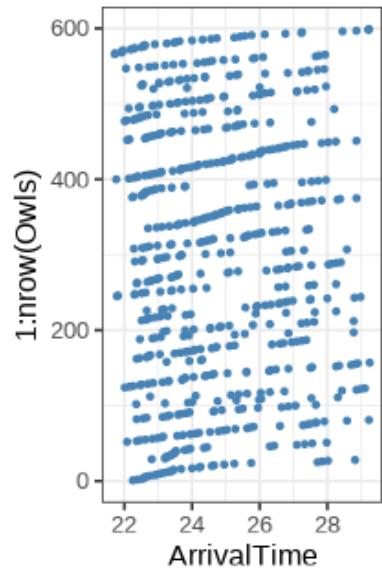
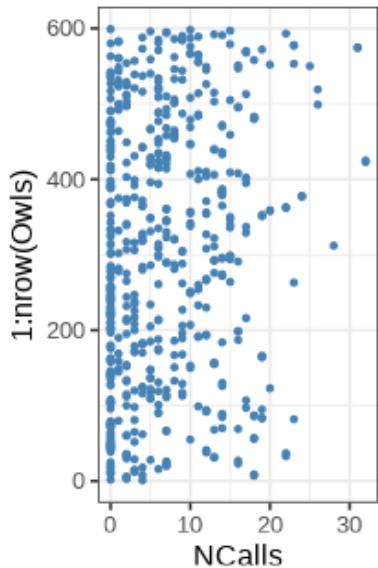
'data.frame': 599 obs. of 8 variables:
 $ Nest           : chr  "AutavauxTV" "AutavauxTV" "AutavauxTV" "AutavauxTV" ...
 $ FoodTreatment  : chr  "Deprived" "Satiated" "Deprived" "Deprived" ...
 $ SexParent      : chr  "Male" "Male" "Male" "Male" ...
 $ ArrivalTime    : num  22.2 22.4 22.5 22.6 22.6 ...
 $ SiblingNegotiation: int  4 0 2 2 2 18 4 18 0 ...
 $ BroodSize       : int  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 1.61 ...
```

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

```
[1] 0
```

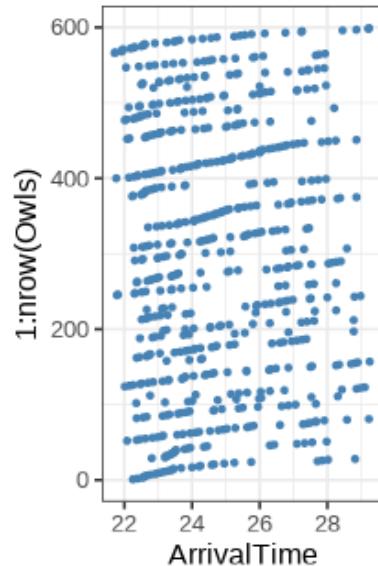
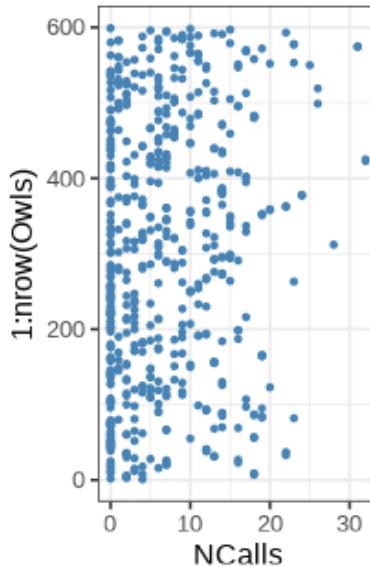
Есть ли выбросы?

```
library(ggplot2); library(cowplot); theme_set(theme_bw(base_size = 16))  
gg_dot <- ggplot(Owls, aes(y = 1:nrow(Owls))) +  
  geom_point(colour = "steelblue")  
  
plot_grid(gg_dot + aes(x = NCalls),  
          gg_dot + aes(x = ArrivalTime), nrow = 1)
```



Есть ли выбросы?

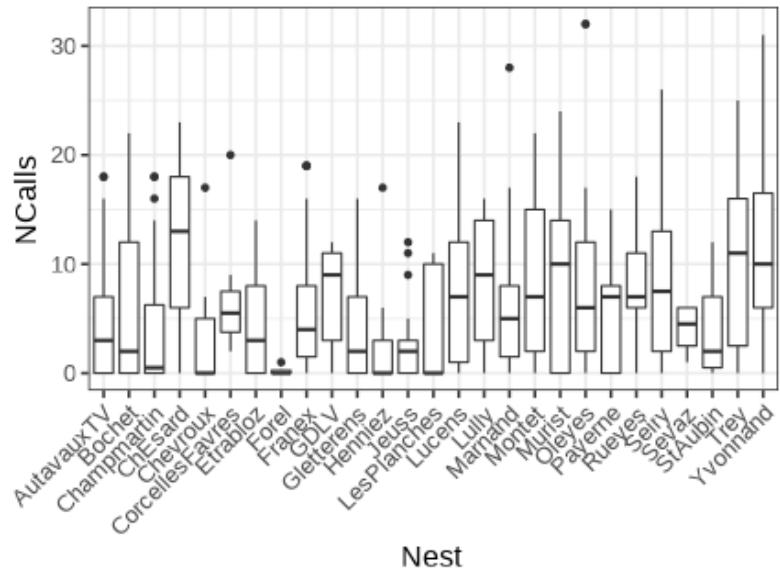
```
library(ggplot2); library(cowplot); theme_set(theme_bw(base_size = 16))  
gg_dot <- ggplot(Owls, aes(y = 1:nrow(Owls))) +  
  geom_point(colour = "steelblue")  
  
plot_grid(gg_dot + aes(x = NCalls),  
          gg_dot + aes(x = ArrivalTime), nrow = 1)
```



- Выбросов нет

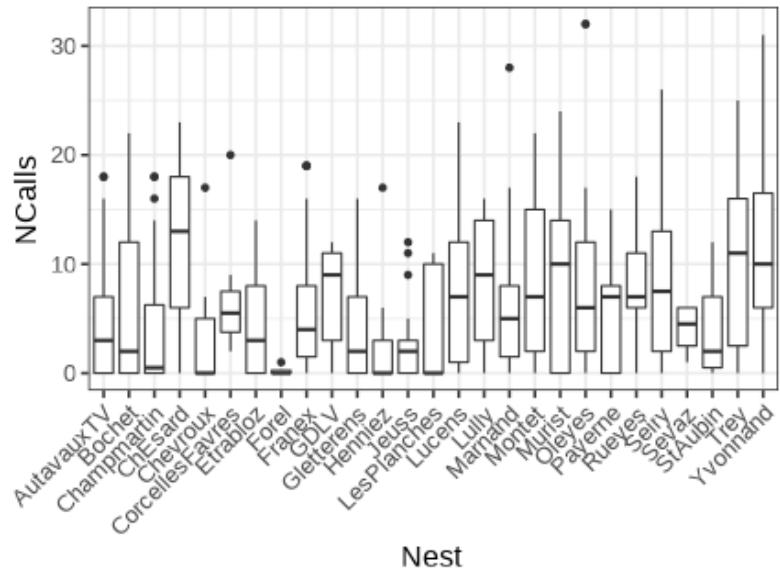
Различаются ли гнезда?

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

Сколько наблюдений в каждом гнезде?

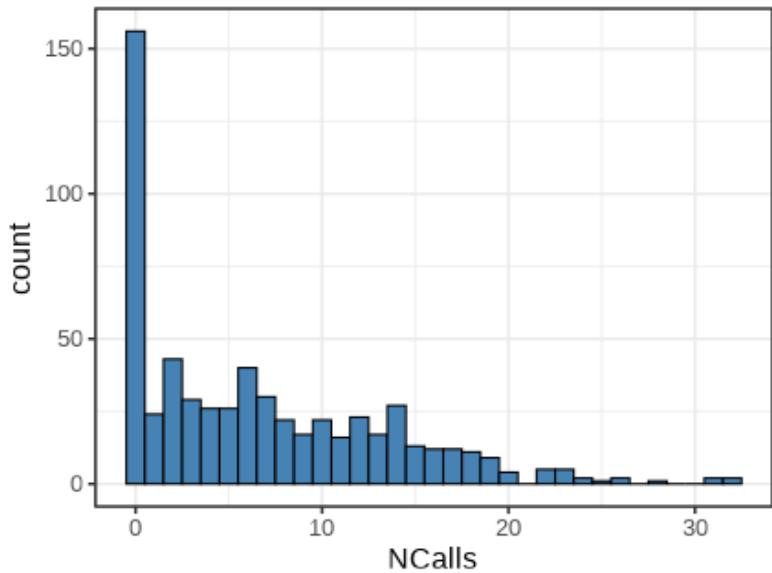
```
table(Owls$Nest)
```

| | | | | | |
|------------|--------|-----------|-------------|-------------|----------|
| Autavaux | TV | Bochet | Champmartin | ChEsard | Chevroux |
| 28 | | 23 | 30 | 20 | 10 |
| Corcelles | Favres | Etrablotz | Forel | Franex | GDLV |
| 12 | | 34 | 4 | 26 | 10 |
| Gletterens | | Henniez | Jeuss | LesPlanches | Lucens |
| 15 | | 13 | 19 | 17 | 29 |
| Lully | | Marnand | Montet | Murist | Oleyes |
| 17 | | 27 | 41 | 24 | 52 |
| Payerne | | Rueyes | Seiry | Sevaz | StAubin |
| 25 | | 17 | 26 | 4 | 23 |
| Trey | | Yvonnand | | | |
| 19 | | 34 | | | |

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

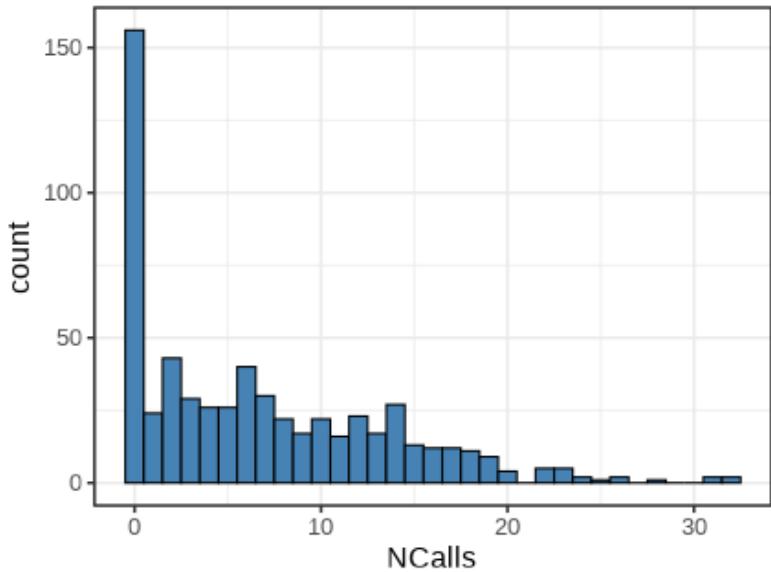
Отклик – счетная переменная

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



Отклик – счетная переменная

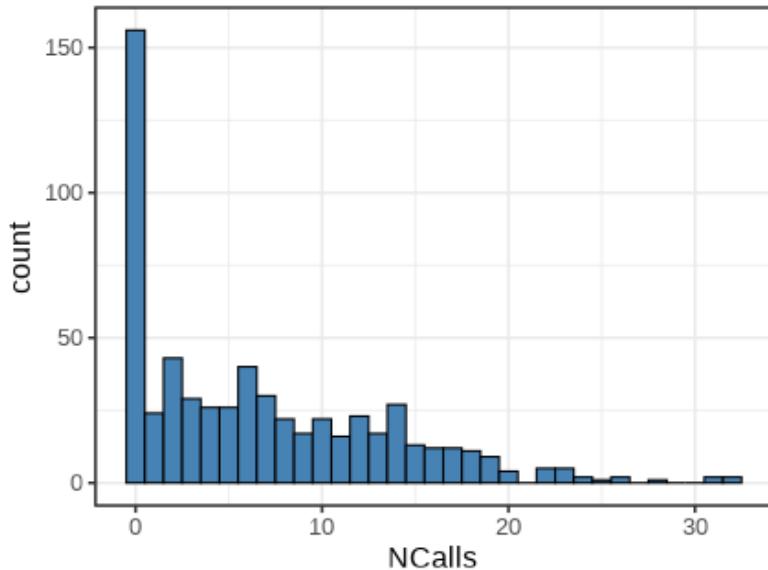
```
ggplot(Owls, aes(x = NCalls)) +  
  geom_histogram(binwidth = 1, fill = "steelblue", colour = "black")
```



Напоминает скорее распределение Пуассона, чем отрицательное биномиальное (т.к. нет длинного правого хвоста)

Отклик – счетная переменная

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



Напоминает скорее распределение Пуассона, чем отрицательное биномиальное (т.к. нет длинного правого хвоста)

```
mean(Owls$NCalls == 0) # доля нулей
```

```
[1] 0.2604
```

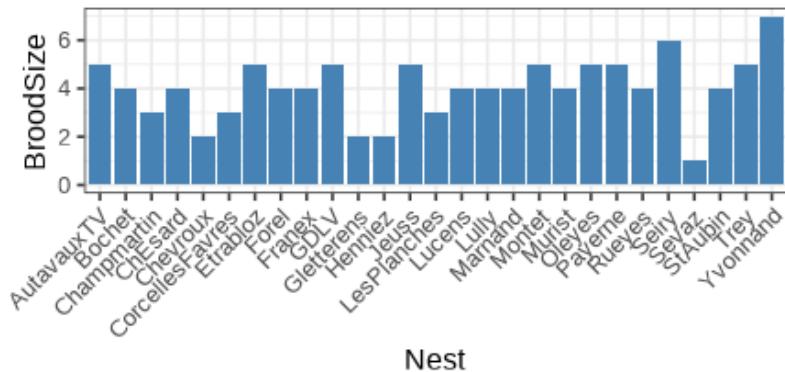
Какого размера выводки в гнездах?

Это нужно учесть, потому что чем больше выводок, тем больше птенцов будут разговаривать, тем больше будет значение отклика `Owls$NCalls`.

```
range(Owls$BroodSize)
```

```
[1] 1 7
```

```
ggplot(Owls, aes(x = Nest, y = BroodSize)) +  
  stat_summary(geom = "bar", fun.y = mean, fill = "steelblue") +  
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



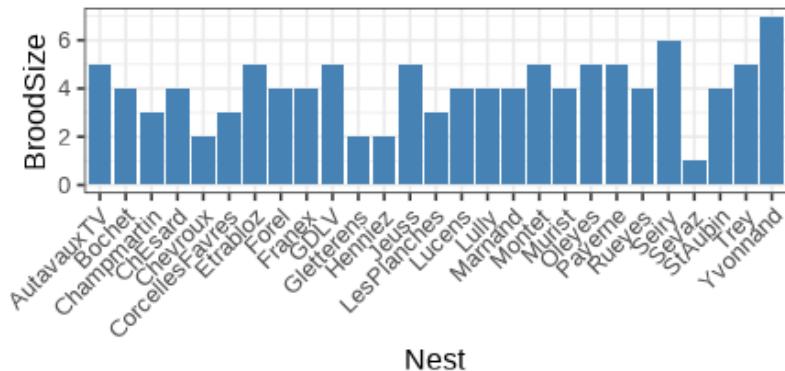
Какого размера выводки в гнездах?

Это нужно учесть, потому что чем больше выводок, тем больше птенцов будут разговаривать, тем больше будет значение отклика `Owls$NCalls`.

```
range(Owls$BroodSize)
```

```
[1] 1 7
```

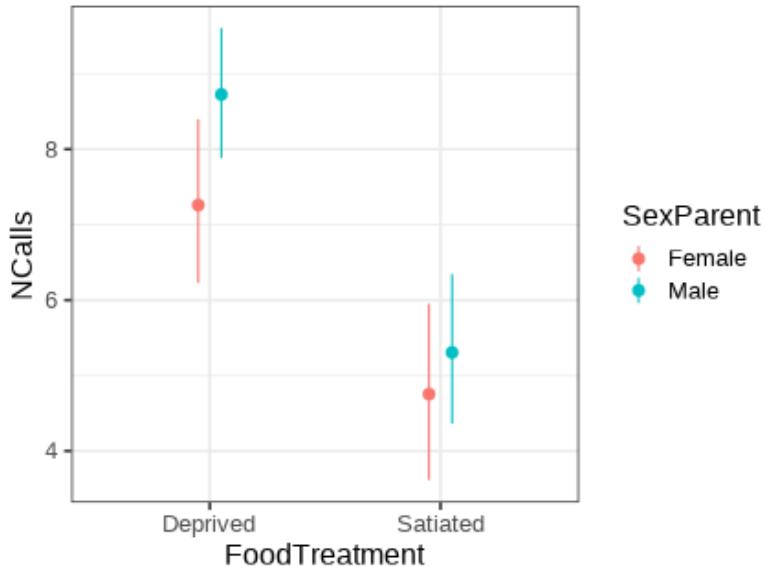
```
ggplot(Owls, aes(x = Nest, y = BroodSize)) +  
  stat_summary(geom = "bar", fun.y = mean, fill = "steelblue") +  
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



- Выводки разные. В пуассоновской `glmer()` это можно откорректировать при помощи `offset` (это предиктор с фиксированным угловым коэффициентом = 1). Сделаем `offset(logBroodSize)`.

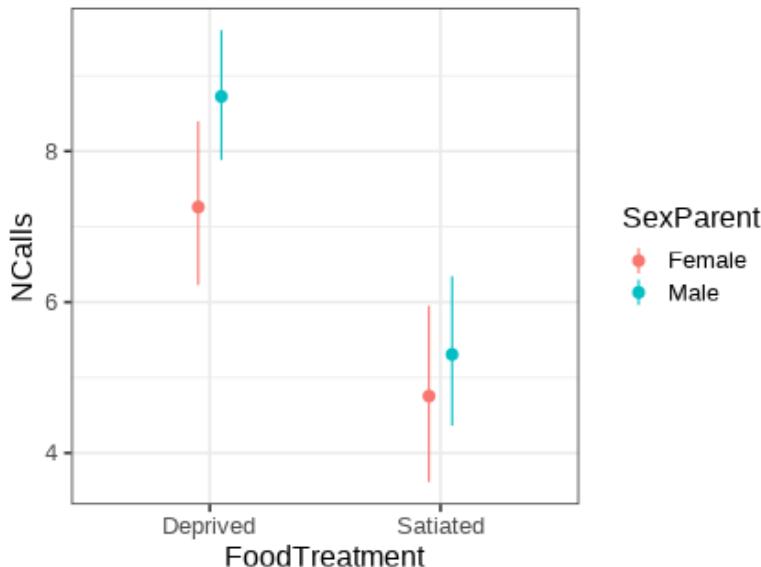
Может быть есть взаимодействие?

```
ggplot(Owls) +  
  stat_summary(aes(x = FoodTreatment, y = NCalls, colour = SexParent),  
               fun.data = "mean_cl_boot", position = position_dodge(width = 0.2))
```



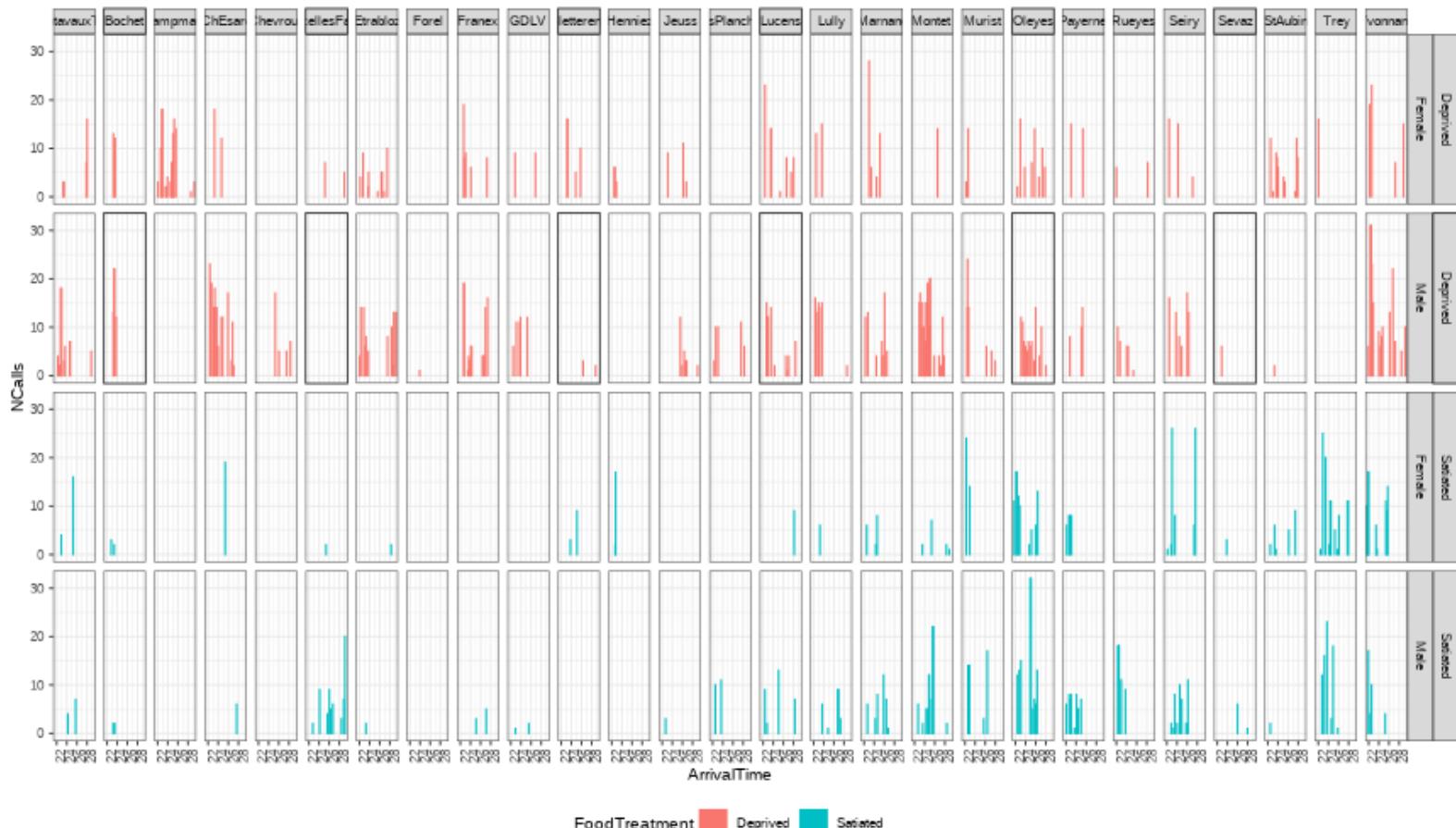
Может быть есть взаимодействие?

```
ggplot(Owls) +  
  stat_summary(aes(x = FoodTreatment, y = NCalls, colour = SexParent),  
               fun.data = "mean_cl_boot", position = position_dodge(width = 0.2))
```

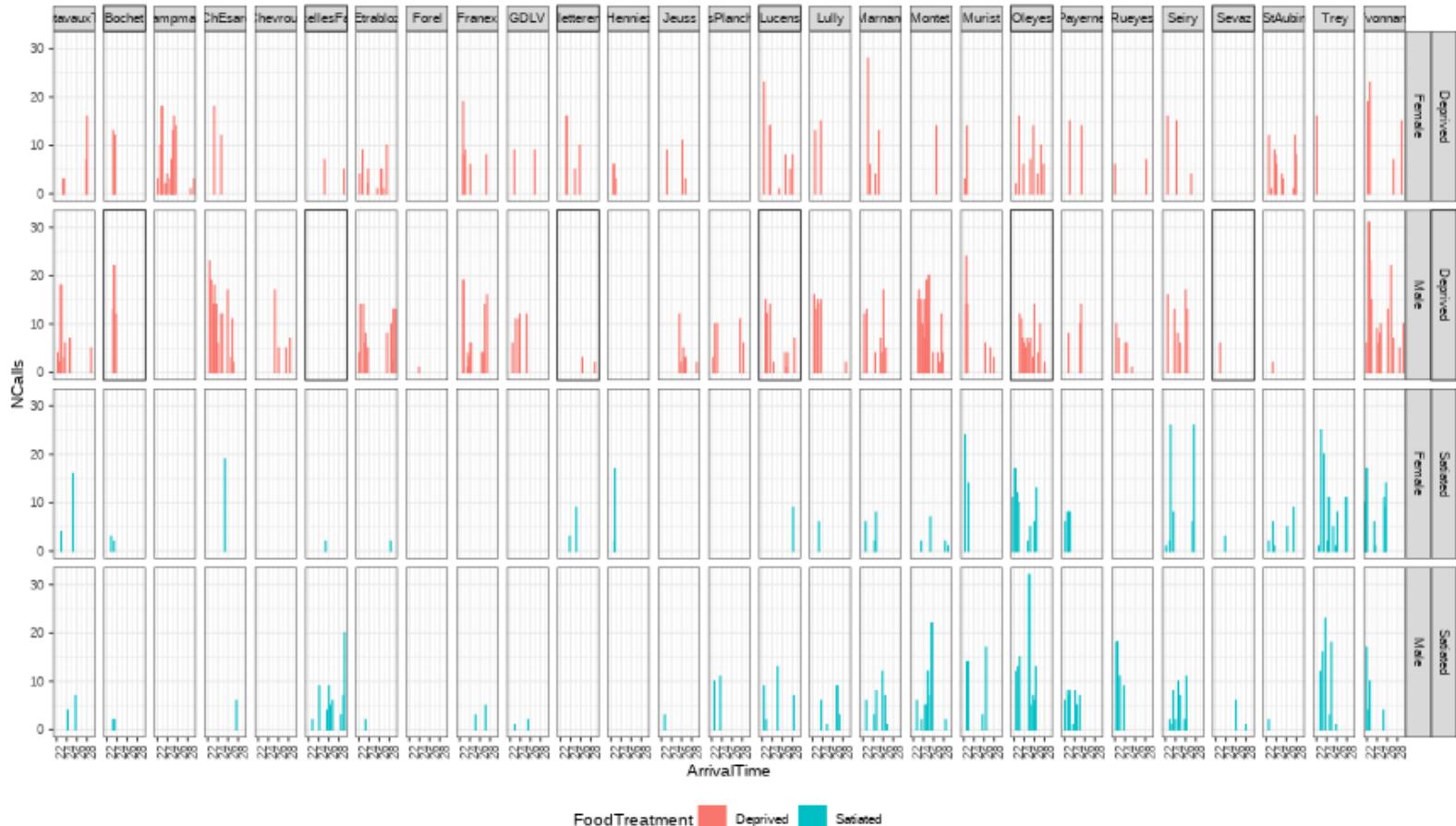


- Похоже, что может быть взаимодействие, но не понятно значимое ли.

Когда орут птенцы?



Когда орут птенцы?



- Птенцы больше орут, если голодали прошлой ночью.
- И, возможно, орут перед прилетом самцов (?)

Код для графика

```
ggplot(Owls, aes(x = ArrivalTime, y = NCalls,
                  colour = FoodTreatment, fill = FoodTreatment)) +
  geom_bar(stat = "identity") +
  facet_grid(FoodTreatment + SexParent ~ Nest) +
  theme_bw(base_size = 10) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90, hjust = 1))
```

Коллинеарность

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

| | SexParent | FoodTreatment | ArrivalTime |
|--|-----------|---------------|-------------|
| | 1.0036 | 1.0044 | 1.0024 |

Коллинеарность

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

| | SexParent | FoodTreatment | ArrivalTime |
|--|-----------|---------------|-------------|
| | 1.0036 | 1.0044 | 1.0024 |

- OK

Смешанная линейная модель с пуассоновским распределением остатков

Линейная модель с пуассоновским распределением остатков

$NCalls \sim Poisson(\mu_{ij})$ — отклик подчиняется распределению Пуассона с параметром μ

$$E(NCalls_{ij}) = \mu_{ij}, \text{var}(NCalls_{ij}) = \mu_{ij}$$

$\ln(\mu_{ij}) = \eta_{ij}$ — функция связи — логарифм

$$\begin{aligned} \eta_{ij} = & \beta_0 + \beta_1 SexParentM_{ij} + \beta_2 FoodTreatmentS_{ij} + \beta_3 ArrivalTime_{ij} + \\ & + \beta_4 SexParentM_{ij}FoodTreatmentS_{ij} + \beta_5 SexParentM_{ij}ArrivalTime_{ij} + \\ & + \log(BroodSize_i) + a_i \end{aligned}$$

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

Подберем линейную модель с пуассоновским распределением остатков

```
library(lme4)
```

```
Loading required package: Matrix
```

```
Attaching package: 'Matrix'
```

```
The following objects are masked from 'package:tidyR':
```

```
expand, pack, unpack
```

```
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| = 0.0059883 (tol = 0.002, component 1)
```

```
Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unidentifiable  
- Rescale variables?
```

Смешанная модель с распределением Пуассона не сходится. Один из возможных вариантов выхода — стандартизация предикторов.

Стандартизуем непрерывные предикторы

У нас только один непрерывный предиктор

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

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

Эта модель сходится

Задание 1

Проверьте модель М1 на избыточность дисперсии вручную.

Подсказка:

Показатель сверхдисперсии — это соотношение суммы квадратов Пирсоновских остатков и числа степеней свободы.

Поскольку сумма квадратов Пирсоновских остатков подчиняется хи-квадрат распределению, можно его использовать для проверки статистической значимости отклонений.

Избыточность дисперсии (Overdispersion)

Для начала разберемся, как это считать вручную

```
R_M1 <- resid(M1, type = "pearson") # Пирсоновские остатки  
N <- nrow(Owls) # Объем выборки  
p <- length(fixef(M1)) + 1# Число параметров (не забудьте сл. эффект!)
```

```
df <- (N - p) # число степеней свободы  
overdispersion <- sum(R_M1^2) /df # во сколько раз var(y) > E(y)  
overdispersion
```

```
[1] 5.461
```

```
pchisq(sum(R_M1^2), df = df, lower.tail = FALSE)
```

```
[1] 0
```

Избыточность дисперсии (Overdispersion)

Для начала разберемся, как это считать вручную

```
R_M1 <- resid(M1, type = "pearson") # Пирсоновские остатки  
N <- nrow(Owls) # Объем выборки  
p <- length(fixef(M1)) + 1# Число параметров (не забудьте сл. эффект!)  
df <- (N - p) # число степеней свободы  
overdispersion <- sum(R_M1^2) / df # во сколько раз var(y) > E(y)  
overdispersion
```

```
[1] 5.461
```

```
pchisq(sum(R_M1^2), df = df, lower.tail = FALSE)
```

```
[1] 0
```

- Избыточность дисперсии.

Избыточность дисперсии при помощи готовых функций

```
# ## Проверка на сверхдисперсию
# Функция для проверки наличия сверхдисперсии в модели (автор Ben Bolker)
# http://bbolker.github.io/mixedmodels-misc/glmmFAQ.html
# Код модифицирован, чтобы учесть дополнительный параметр в NegBin GLMM, подобранных MASS::glm.nb()
overdisp_fun <- function(model) {
  rdf <- df.residual(model) # Число степеней свободы N - p
  if (any(class(model) == 'negbin')) rdf <- rdf - 1 ## учитываем k в NegBin GLMM
  rp <- residuals(model,type='pearson') # Пирсоновские остатки
  Pearson.chisq <- sum(rp^2) # Сумма квадратов остатков, подчиняется Хи-квадрат распределению
  prat <- Pearson.chisq/rdf # Отношение суммы квадратов остатков к числу степеней свободы
  pval <- pchisq(Pearson.chisq, df=rdf, lower.tail=FALSE) # Уровень значимости
  c(chisq=Pearson.chisq, ratio=prat, rdf=rdf, p=pval) # Вывод результатов
}

overdisp_fun(M1)
```

| chisq | ratio | rdf | p |
|----------|-------|---------|-------|
| 3232.737 | 5.461 | 592.000 | 0.000 |

```
library(performance)
check_overdispersion(M1)

# Overdispersion test

  dispersion ratio =      5.461
Pearson's Chi-Squared = 3232.737
                  p-value =  < 0.001
```

Почему здесь могла быть избыточность дисперсии?

Почему здесь могла быть избыточность дисперсии?

- Отскакивающие значения → убрать
- Пропущены ковариаты или взаимодействия предикторов → добавить
- Наличие внутригрупповых корреляций (нарушение независимости выборок) → 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
- Просто большая дисперсия? → NB

График остатков

```
M1_diag <- data.frame(Owls,
                       .fitted = predict(M1, type = "response"),
                       .pears_resid = residuals(M1, type = "pearson"))

gg_resid <- ggplot(M1_diag, aes(x = .fitted, y = .pears_resid,
                                 colour = FoodTreatment)) +
  geom_point() +
  facet_grid(SexParent ~ FoodTreatment)
gg_resid
```

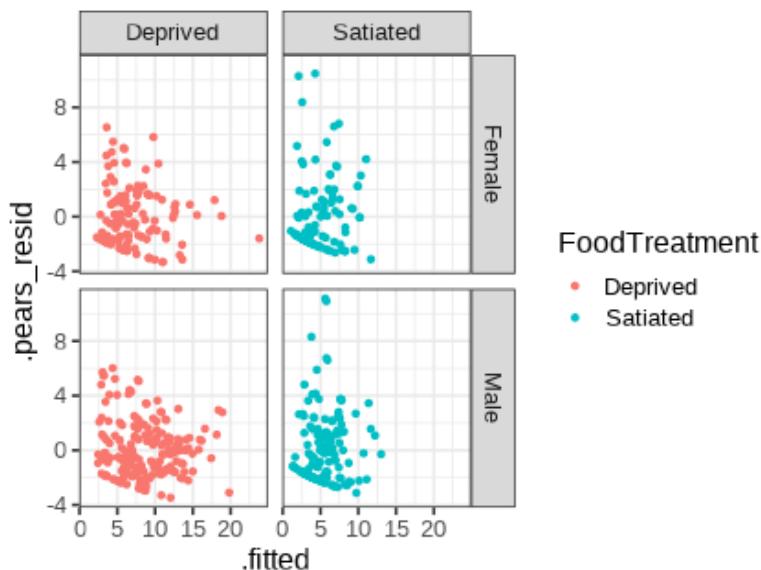
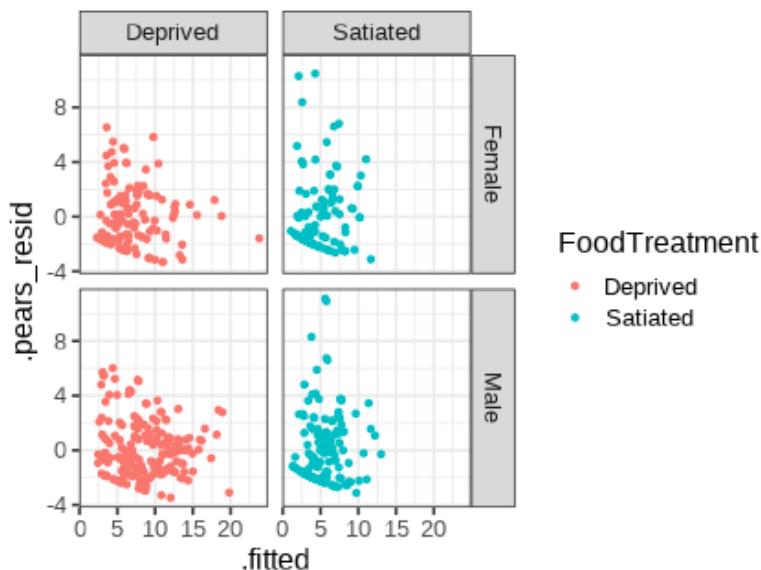


График остатков

```
M1_diag <- data.frame(Owls,
                       .fitted = predict(M1, type = "response"),
                       .pears_resid = residuals(M1, type = "pearson"))

gg_resid <- ggplot(M1_diag, aes(x = .fitted, y = .pears_resid,
                                 colour = FoodTreatment)) +
  geom_point() +
  facet_grid(SexParent ~ FoodTreatment)
gg_resid
```

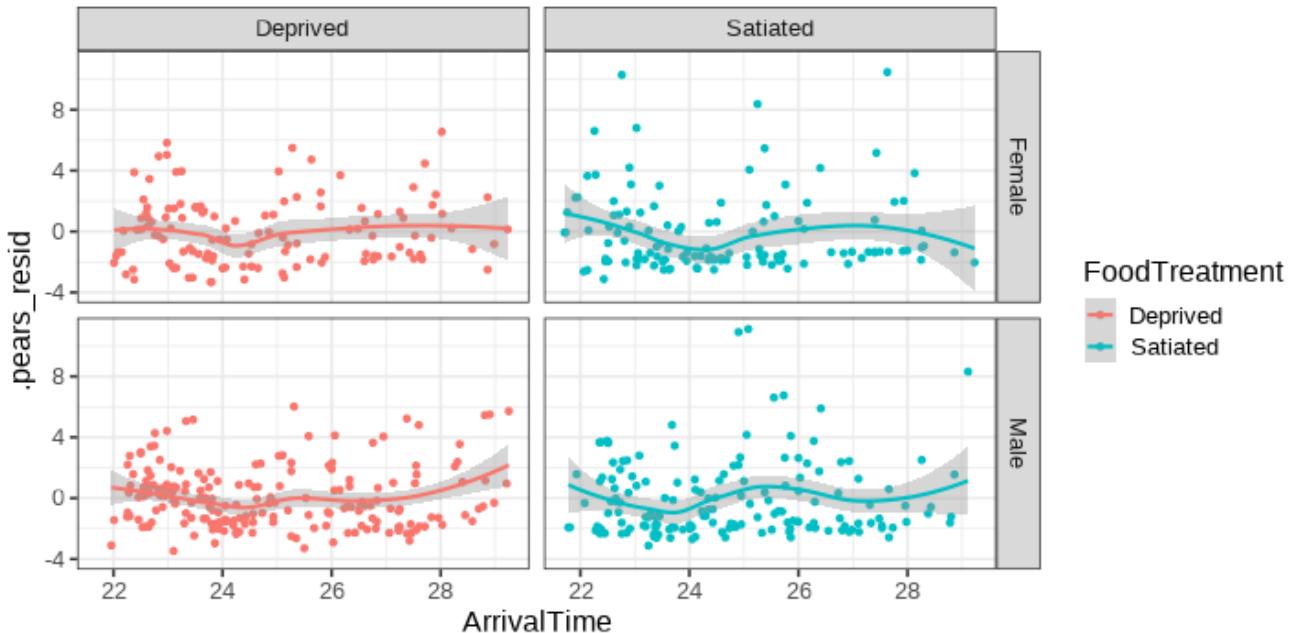


- Есть большие остатки

Есть ли еще какие-то паттерны в остатках?

В этом датасете есть переменная `ArrivalTime`, отражающая время. На графике зависимости остатков от такой переменной можно поискать нелинейные паттерны.

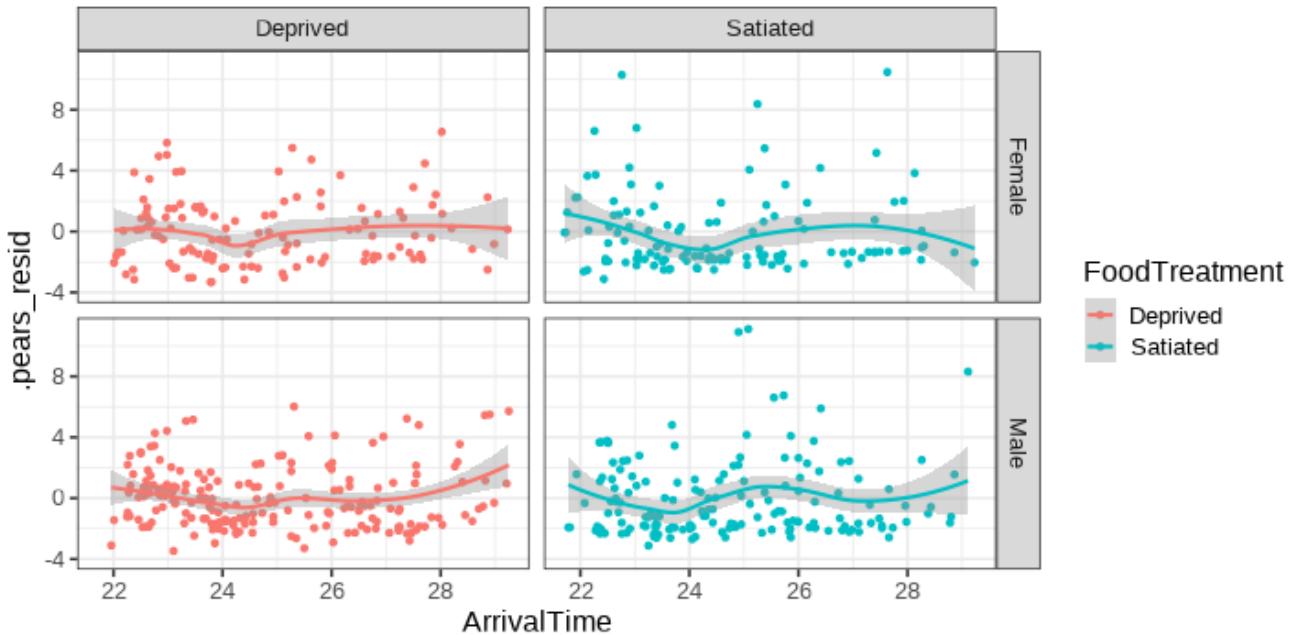
```
gg_resid %>% aes(x = ArrivalTime) + geom_smooth(method = "loess")
```



Есть ли еще какие-то паттерны в остатках?

В этом датасете есть переменная `ArrivalTime`, отражающая время. На графике зависимости остатков от такой переменной можно поискать нелинейные паттерны.

```
gg_resid %>% aes(x = ArrivalTime) + geom_smooth(method = "loess")
```



- Есть намек на нелинейность. Возможно, нужен GAMM

Проверяем, есть ли нелинейный паттерн в остатках

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

```
Family: gaussian
Link function: identity

Formula:
.pears_resid ~ 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 0.0000049 ***
---
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
```

Проверяем, есть ли нелинейный паттерн в остатках

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

```
Family: gaussian
Link function: identity

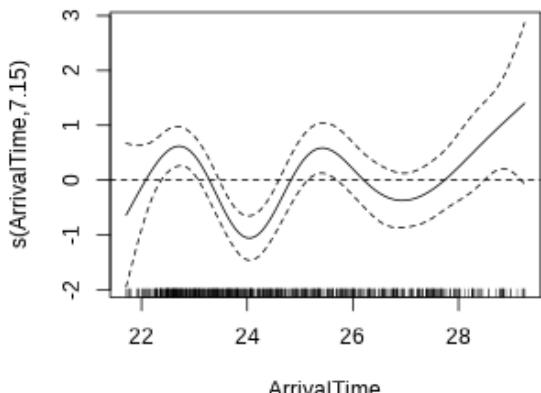
Formula:
.pears_resid ~ s(ArrivalTime)

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 0.00000049 ***
---
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
```

```
# сплайн на графике
plot(nonlin1)
# горизонтальная линия
abline(h = 0, lty = 2)
```



Проверяем, есть ли нелинейный паттерн в остатках

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

```
Family: gaussian
Link function: identity

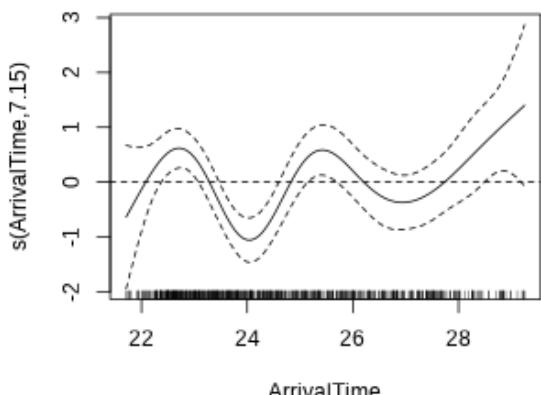
Formula:
.pears_resid ~ s(ArrivalTime)

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 0.00000049 ***
---
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
```

```
# сплайн на графике
plot(nonlin1)
# горизонтальная линия
abline(h = 0, lty = 2)
```



- Совершенно точно нужен GAMM. Но продолжим с GLMM

Смешанная линейная модель с отрицательным биномиальным распределением остатков

У нас была сверхдисперсия. Пробуем NB GLMM

$NCalls_{ij} \sim NegBin(\mu_{ij}, k)$ — отклик подчиняется отрицательному биномиальному распределению с параметрами μ и k

$$E(NCalls_{ij}) = \mu_{ij}, \text{var}(NCalls_{ij}) = \mu_{ij} + \mu_{ij}^2/k$$

$\ln(\mu_{ij}) = \eta_{ij}$ — функция связи — логарифм

$$\begin{aligned}\eta_{ij} = & \beta_0 + \beta_1 SexParentM_{ij} + \beta_2 FoodTreatmentS_{ij} + \beta_3 ArrivalTime_{ij} + \\& + \beta_4 SexParentM_{ij}FoodTreatmentS_{ij} + \beta_5 SexParentM_{ij}ArrivalTime_{ij} + \\& + \log(BroodSize_i) + a_i\end{aligned}$$

- $a_i \sim N(0, \sigma_{Nest}^2)$ — случайный эффект гнезда (intercept)
- 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)
M2.1 <- update(M1, family = negative.binomial(theta=th))
bind_rows(fixef(M2), fixef(M2.1))

# A tibble: 2 × 6
` (Intercept)` SexParentMale FoodTreatmentSatiated ArrivalTime_std SexParent...¹ SexPa...²
<dbl>           <dbl>           <dbl>           <dbl>           <dbl>           <dbl>
1      0.656       -0.00996        -0.771        -0.249        0.178        0.0498
2      0.659       -0.0130         -0.775        -0.248        0.177        0.0484
# ... with abbreviated variable names `¹`SexParentMale:FoodTreatmentSatiated``,
#   `²`SexParentMale:ArrivalTime_std`
```

Задание 2

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 (Не забудьте сл.эффект и k)  
df <- (N - p) # число степеней свободы  
overdispersion <- sum(R_M2^2) /df # во сколько раз var(y) > E(y)  
overdispersion
```

```
[1] 0.851
```

```
pchisq(sum(R_M2^2), df = df, lower.tail = FALSE)
```

```
[1] 0.9963
```

Избыточность дисперсии (Overdispersion)

```
R_M2 <- resid(M2, type = "pearson") # Пирсоновские остатки  
N <- nrow(Owls) # Объем выборки  
p <- length(fixef(M2)) + 1 + 1 # Число параметров (Не забудьте сл.эффект и k)  
df <- (N - p) # число степеней свободы  
overdispersion <- sum(R_M2^2) /df # во сколько раз var(y) > E(y)  
overdispersion
```

```
[1] 0.851
```

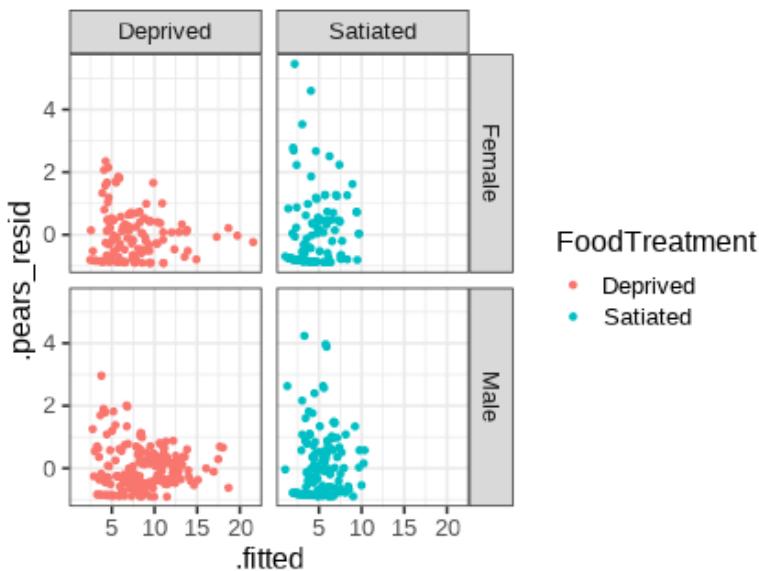
```
pchisq(sum(R_M2^2), df = df, lower.tail = FALSE)
```

```
[1] 0.9963
```

- Хорошо.

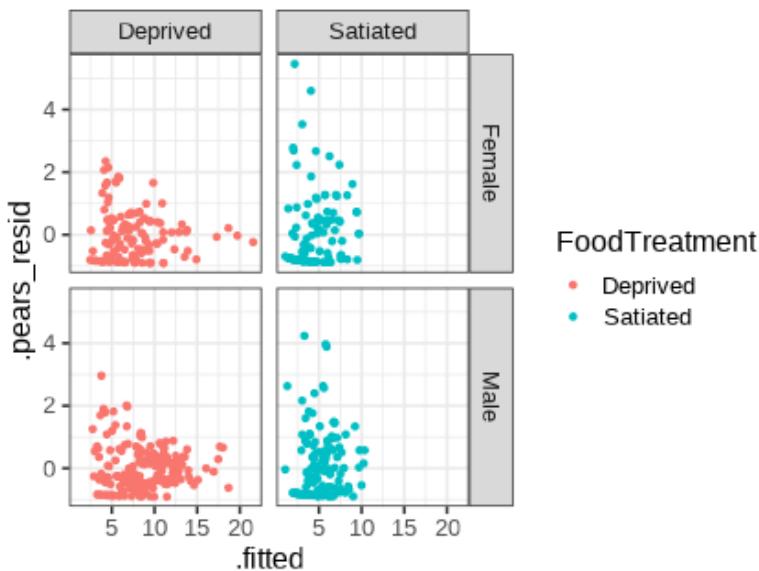
Диагностика отр. биномиальной модели

```
M2_diag <- data.frame(Owls,
                        .fitted = predict(M2, type = "response"),
                        .pears_resid = residuals(M2, type = "pearson"))
gg_resid <- ggplot(M2_diag, aes(x = .fitted, y = .pears_resid,
                                 colour = FoodTreatment)) +
  geom_point() +
  facet_grid(SexParent ~ FoodTreatment)
gg_resid
```



Диагностика отр. биномиальной модели

```
M2_diag <- data.frame(Owls,
                        .fitted = predict(M2, type = "response"),
                        .pears_resid = residuals(M2, type = "pearson"))
gg_resid <- ggplot(M2_diag, aes(x = .fitted, y = .pears_resid,
                                 colour = FoodTreatment)) +
  geom_point() +
  facet_grid(SexParent ~ FoodTreatment)
gg_resid
```

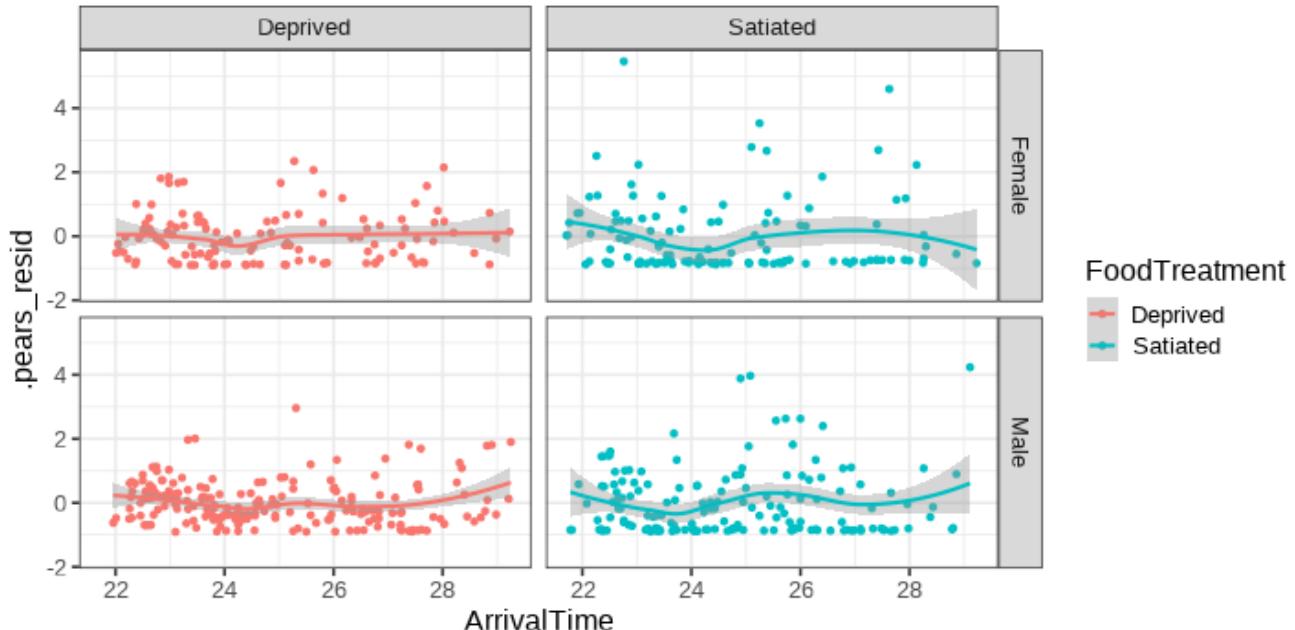


- Есть большие остатки.

Есть ли еще какие-то паттерны в остатках?

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

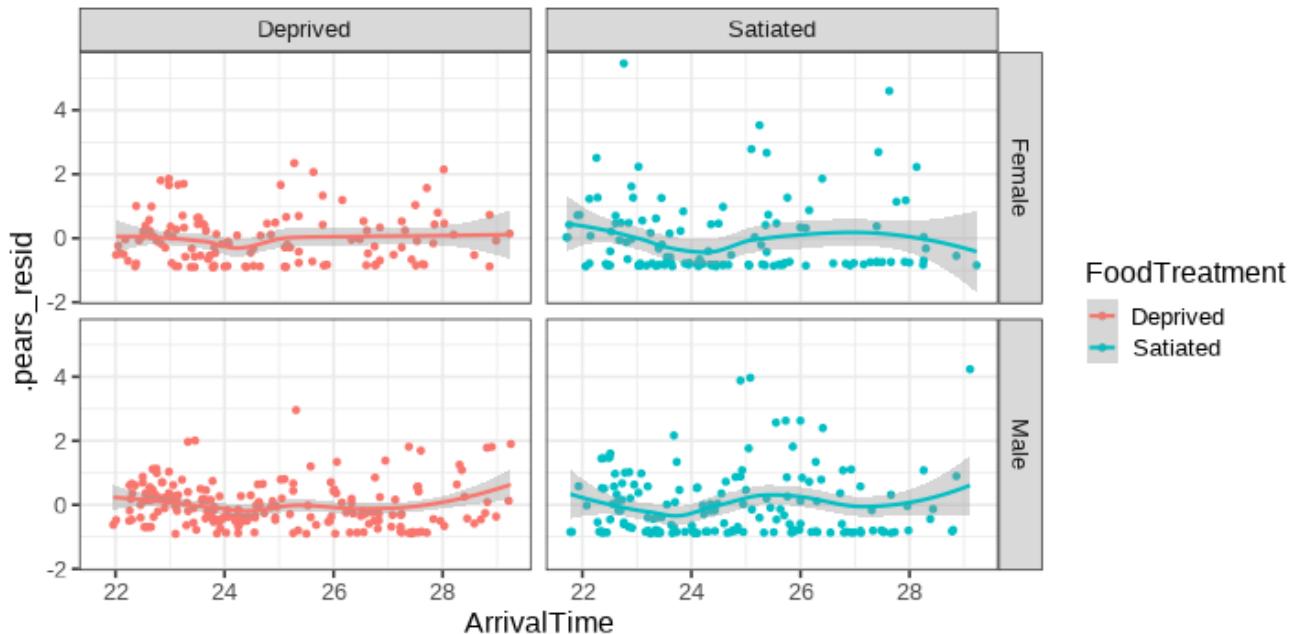
```
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(method = 'loess')
```



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(method = 'loess')
```



- Подозрительно. Возможно, нужен GAMM

Проверяем, есть ли нелинейные паттерны

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

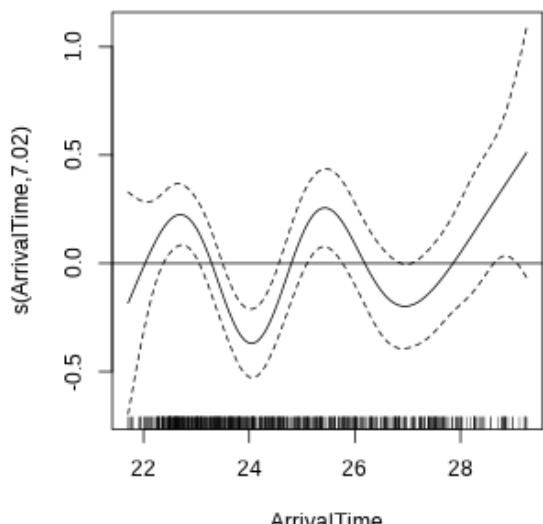
```
Family: gaussian  
Link function: identity  
  
Formula:  
.pears_resid ~ s(ArrivalTime)  
  
Parametric coefficients:  
             Estimate Std. Error t value Pr(>|t|)  
(Intercept) -0.00121    0.03642   -0.03    0.97  
  
Approximate significance of smooth terms:  
              edf Ref.df     F p-value  
s(ArrivalTime) 7.02    8.1 4.55 0.000021 ***  
---  
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
```

Проверяем, есть ли нелинейные паттерны

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

```
Family: gaussian  
Link function: identity  
  
Formula:  
.pears_resid ~ 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 0.000021 ***  
---  
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
```

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

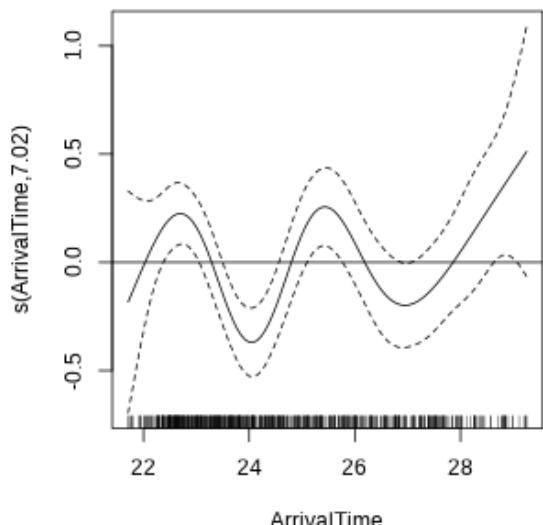


Проверяем, есть ли нелинейные паттерны

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

```
Family: gaussian  
Link function: identity  
  
Formula:  
.pears_resid ~ 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 0.000021 ***  
---  
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
```

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



- Совершенно точно нужен ГАММ

Подбор оптимальной модели

При желании модель можно упростить

```
summary(M2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) [  
glmerMod]
```

```
Family: Negative Binomial(0.8848)  ( 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
```

```
Fixed effects:
```

| | Estimate | Std. Error | z value | Pr(> z) |
|---------------|----------|------------|---------|---------------|
| (Intercept) | 0.65564 | 0.12935 | 5.07 | 0.0000004 *** |
| SexParentMale | -0.00996 | 0.14211 | -0.07 | 0.94415 |

Задание 3

Попробуйте упростить модель M2

Можно ли что-то выкинуть

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

Single term deletions

Model:

```
NCalls ~ SexParent * FoodTreatment + SexParent * ArrivalTime_std +  
       offset(logBroodSize) + (1 | Nest)
```

| | npar | 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)
```

| | npar | 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)
      npar   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)
      npar   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)
      npar  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 0.00000651761 ***
---
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)
      npar  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 0.00000651761 ***
---
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)
          npar   AIC    LRT      Pr(Chi)
<none>           3475
FoodTreatment     1 3513 39.9 0.00000000027 ***
ArrivalTime_std   1 3493 20.1 0.00000746458 ***
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Финальная модель

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

Single term deletions

Model:
NCalls ~ FoodTreatment + ArrivalTime_std + (1 | Nest) + offset(logBroodSize)
          npar   AIC    LRT      Pr(Chi)
<none>       3475
FoodTreatment     1 3513 39.9 0.00000000027 ***
ArrivalTime_std   1 3493 20.1 0.00000746458 ***
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- это финальная модель

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 |

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  
df <- (N - p) # число степеней свободы  
overdispersion <- sum(R_M5^2) / df # во сколько раз var(y) > E(y)  
overdispersion
```

```
[1] 0.8445
```

```
pchisq(sum(R_M5^2), df = df, lower.tail = FALSE)
```

```
[1] 0.9976
```

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  
df <- (N - p) # число степеней свободы  
overdispersion <- sum(R_M5^2) / df # во сколько раз var(y) > E(y)  
overdispersion
```

```
[1] 0.8445
```

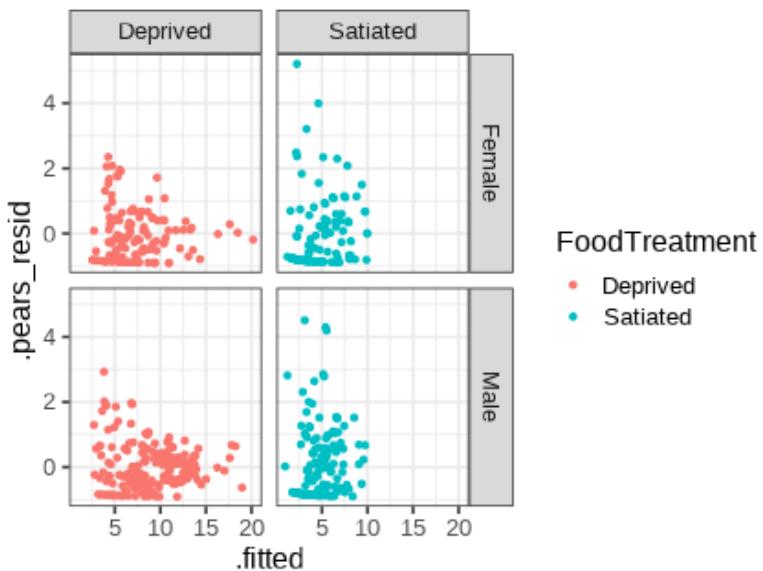
```
pchisq(sum(R_M5^2), df = df, lower.tail = FALSE)
```

```
[1] 0.9976
```

- Хорошо.

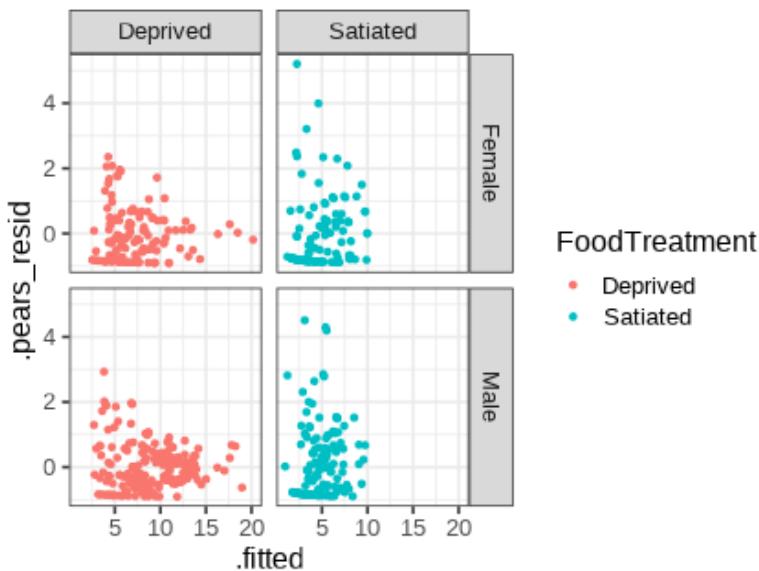
Диагностика отр. биномиальной модели

```
M5_diag <- data.frame(Owls,
                       .fitted <- predict(M5, type = "response"),
                       .pears_resid <- residuals(M5, type = "pearson"))
gg_resid <- ggplot(M5_diag, aes(x = .fitted, y = .pears_resid,
                                 colour = FoodTreatment)) +
  geom_point() +
  facet_grid(SexParent ~ FoodTreatment)
gg_resid
```



Диагностика отр. биномиальной модели

```
M5_diag <- data.frame(Owls,
                       .fitted <- predict(M5, type = "response"),
                       .pears_resid <- residuals(M5, type = "pearson"))
gg_resid <- ggplot(M5_diag, aes(x = .fitted, y = .pears_resid,
                                 colour = FoodTreatment)) +
  geom_point() +
  facet_grid(SexParent ~ FoodTreatment)
gg_resid
```

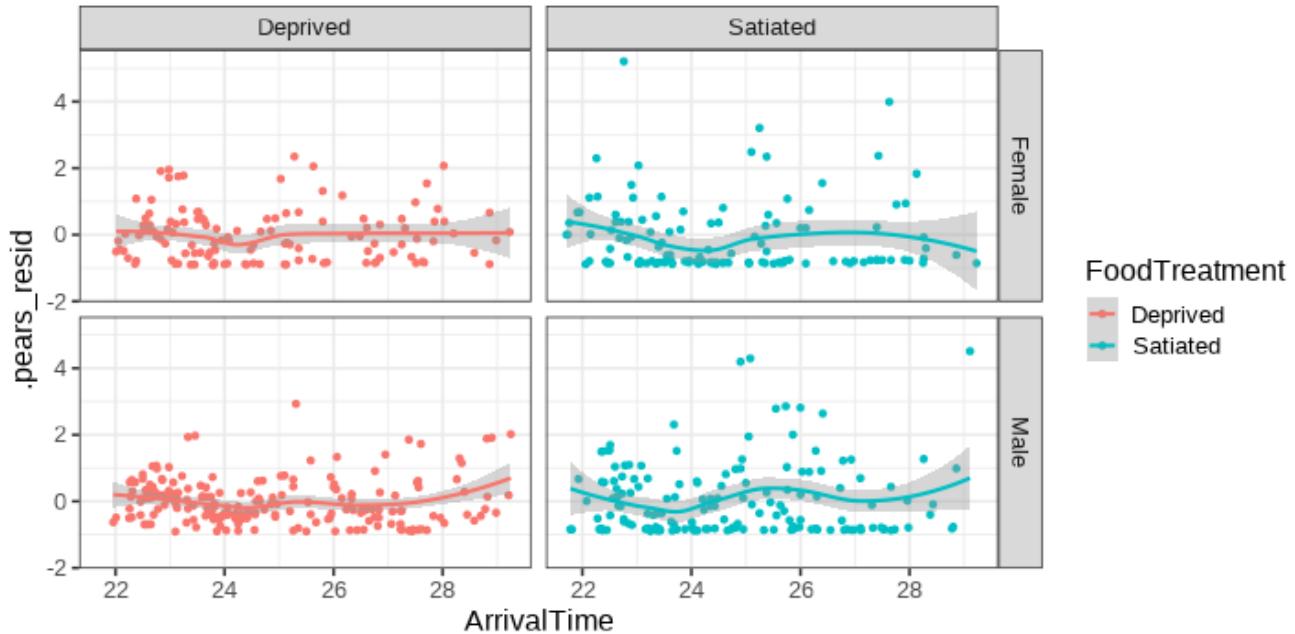


- Есть большие остатки

Есть ли еще какие-то паттерны в остатках?

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

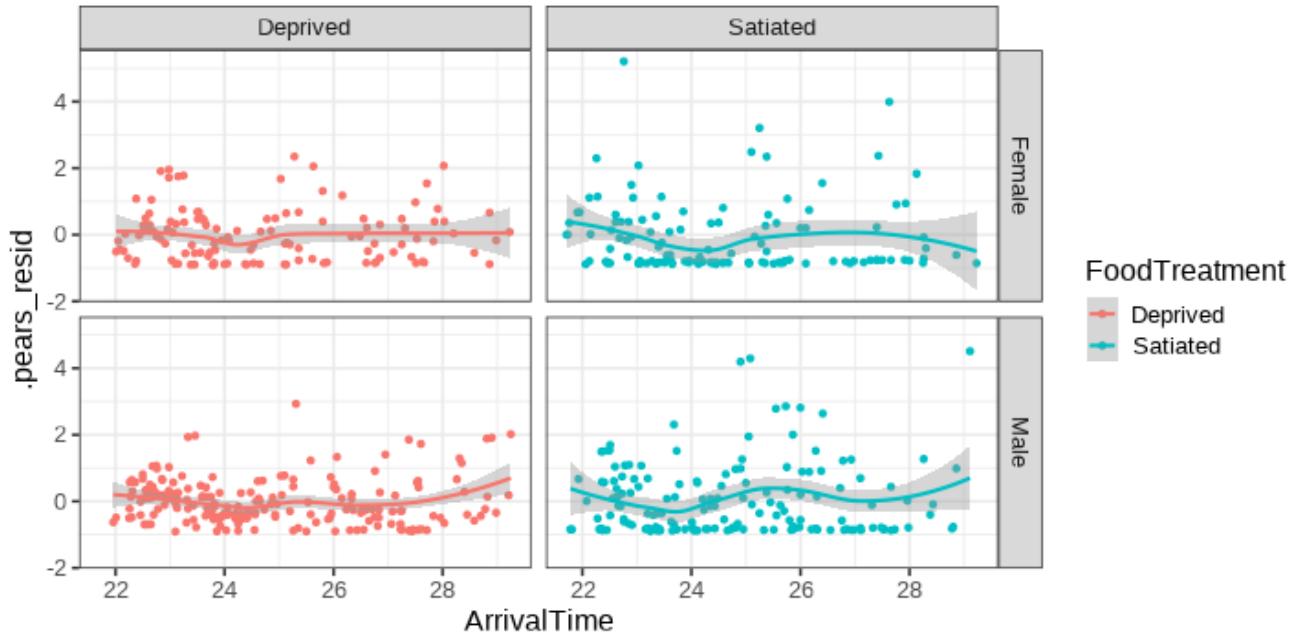
```
gg_resid %+% aes(x = ArrivalTime) + geom_smooth(method = "loess")
```



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(method = "loess")
```



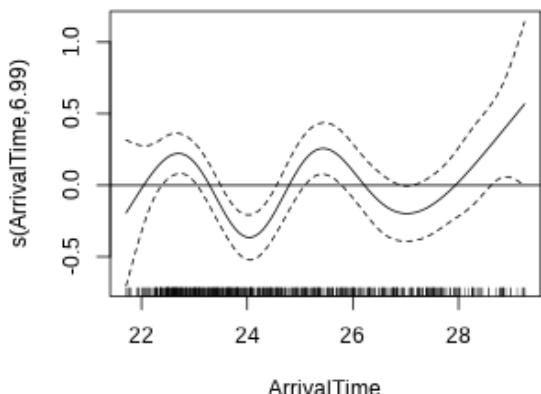
- Подозрительно. Возможно, нужен GAMM

Проверяем, есть ли нелинейные паттерны

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

```
Family: gaussian  
Link function: identity  
  
Formula:  
.pears_resid ~ 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 0.000018 ***  
---  
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
```

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

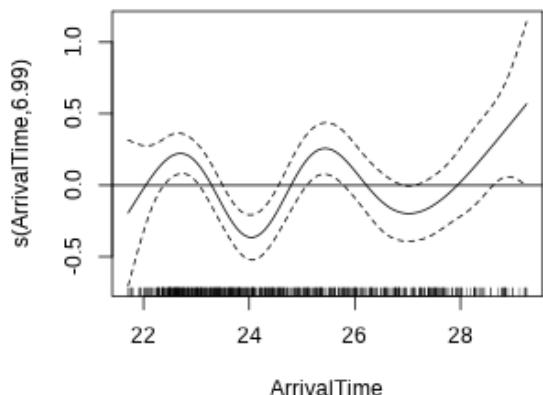


Проверяем, есть ли нелинейные паттерны

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

```
Family: gaussian  
Link function: identity  
  
Formula:  
.pears_resid ~ 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 0.000018 ***  
---  
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
```

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



- Совершенно точно нужен GAMM
- Но мы продолжим в целях обучения

Представление результатов

Финальная GLMM, которую мы получили, выглядит так

- $NCalls_{ij} \sim NegBin(\mu_{ij}, k)$ — 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
- $$\begin{aligned}\eta_{ij} = & \beta_0 + \beta_1 SexParentM_{ij} + \beta_2 FoodTreatmentS_{ij} + \beta_3 ArrivalTime_{ij} + \\ & + \log(BroodSize_i) + a_i\end{aligned}$$
- $a_i \sim N(0, \sigma_{Nest}^2)$ — случайный эффект гнезда (intercept)
- i — гнездо
- j — наблюдение

Готовим данные для графика модели

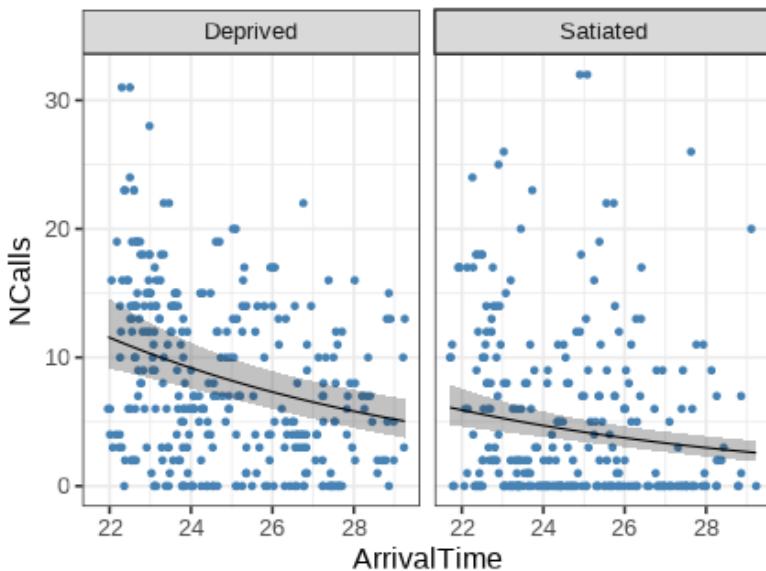
```
library(dplyr)
NewData <- Owls %>% group_by(FoodTreatment) %>%
  do(data.frame(ArrivalTime_std = seq(min(.\$ArrivalTime_std),
                                       max(.\$ArrivalTime_std),
                                       length = 100)))
NewData$ArrivalTime <- NewData$ArrivalTime_std * sd(Owls$ArrivalTime) +
  mean(Owls$ArrivalTime)
```

Предсказания и ошибки

```
# Модельная матрица
X <- model.matrix(~ FoodTreatment + ArrivalTime_std, data = NewData)
# К предсказанным значениям нужно прибавить оффсет.
# Мы будем делать предсказания для среднего размера выводка.
# В масштабе функции связи
NewData$fit_eta <- X %*% fixef(M5) + log(mean(Owls$BroodSize))
NewData$SE_eta <- sqrt(diag(X %*% vcov(M5) %*% t(X)))
# В масштабе отклика
NewData$fit_mu <- exp(NewData$fit_eta)
NewData$lwr <- exp(NewData$fit_eta - 2 * NewData$SE_eta)
NewData$upr <- exp(NewData$fit_eta + 2 * NewData$SE_eta)
```

График предсказанных значений

```
ggplot() +  
  geom_point(data = Owls, aes(x = ArrivalTime, y = NCalls), colour = "steelblue") +  
  geom_ribbon(data = NewData, aes(x = ArrivalTime, ymax = upr, ymin = lwr),  
              alpha = 0.3) +  
  geom_line(data = NewData, aes(x = ArrivalTime, y = fit_mu,  
                                group = FoodTreatment)) +  
  facet_wrap(~ FoodTreatment)
```



Take-home messages

- В случае счетных зависимых переменных (неотрицательных целочисленных величин) применяются модели, основанные на распределении Пуассона или отрицательном 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 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).