

Моделирование структуры дисперсии в смешанных моделях

Линейные модели...

Вадим Хайтов, Марина Варфоломеева



“Эволюция” регрессии

Простая регрессионная модель

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

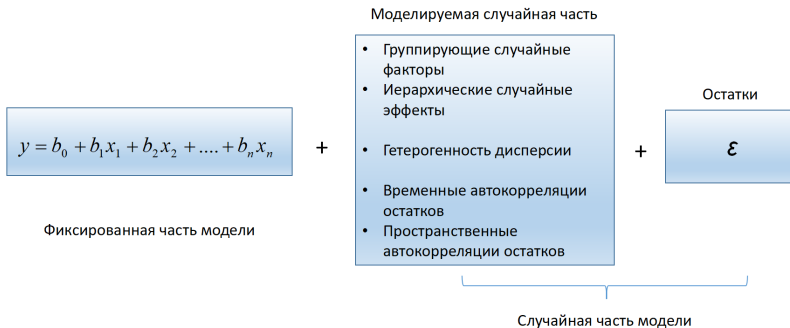
Фиксированная часть модели: $\mathbf{X}\beta$

Случайная часть модели: ε

В моделях, основанных на нормальном распределении $\varepsilon \sim N(0, \sigma^2)$

Важно! Остатки независимы и одинаково распределены со средним 0 и дисперсией σ^2 , одинаковой для всех уровней y_i . То есть остатки - это шум, в котором нет каких-то паттернов.

Смешанные модели



Смешанная линейная модель с группирующими факторами

$$\mathbf{Y}_i = \mathbf{X}_i \beta + \mathbf{Z}_i \mathbf{b}_i + \varepsilon_i$$

$$\varepsilon_i \sim N(0, \Sigma_i)$$

$$\mathbf{b}_i \sim N(0, \mathbf{D})$$

Расширенная смешанная линейная модель

$$\mathbf{Y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{b}_i + \varepsilon_i$$

$$\varepsilon_i \sim N(0, \sigma^2\Lambda_i)$$

Поведение остатков в пределах групп, связанных со случайными факторами, модифицируется (моделируется) матрицей Λ

$$\mathbf{b}_i \sim N(0, \mathbf{D})$$

Ковариата дисперсии (Variance covariate)

Расширенная модель может включать еще один компонент

$$\varepsilon \sim N(0, \sigma^2 \times f(VC))$$

VC - ковариата дисперсии

$f(VC)$ - функция, вводящая поправку, стабилизирующую дисперсию

В зависимости от формы функции $f(VC)$ мы получим разную структуру дисперсии в модели

Пример – сексуальная активность мух



Зависит ли продолжительность жизни самцов от их сексуальной активности?



www.shutterstock.com • 625417247

<https://www.shutterstock.com/ru/image-photo/fruit-flies-drosophila-red-eyes-625417247>

Вопрос исследования:

Зависит ли продолжительность жизни самца от его половой активности?

Зависимая переменная

-longevity Продолжительность жизни самца (количество дней)

Предикторы

-activity— дискретный фактор, характеризующий условия активности самцов.

-thorax — длина груди, непрерывная величина (мм)

Дизайн эксперимента



В фокусе исследования переменная activity однако известно, что крупные самцы живут дольше мелких. В качестве ковариаты взят размер самца thorax

Читаем данные

```
library(faraway)
data(fruitfly)
fly <- fruitfly # Переименуем датасет для краткости
str(fly)
```

```
# 'data.frame': 124 obs. of 3 variables:
# $ thorax : num 0.68 0.68 0.72 0.72 0.76 0.76 0.76 0.76 0.76 0.8 ...
# $ longevity: int 37 49 46 63 39 46 56 63 65 56 ...
# $ activity : Factor w/ 5 levels "isolated","one",...: 4 4 4 4 4 4 4 4 4 4 ...
```

Проверяем данные

Есть ли пропущенные значения?

```
colSums(is.na(fly))
```

```
#   thorax longevity  activity
```

```
#         0         0         0
```

Сколько измерений по каждой из градаций?

```
table(fly$activity)
```

```
#
```

```
# isolated      one      low      many      high
```

```
#          25       25       25       24       25
```

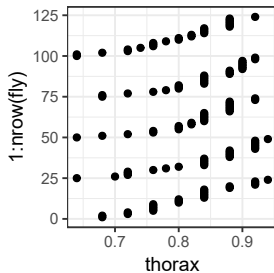
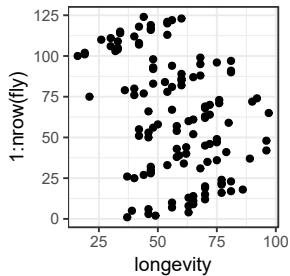
Нет ли выбросов: пишем код

```
library(ggplot2)
theme_set(theme_bw())

gg_dot <- ggplot(fly, aes(y = 1:nrow(fly))) +
  geom_point()
Pl1 <- gg_dot + aes(x = longevity)
Pl2 <- gg_dot + aes(x = thorax)
```

Нет ли выбросов: строим диаграммы Кливленда

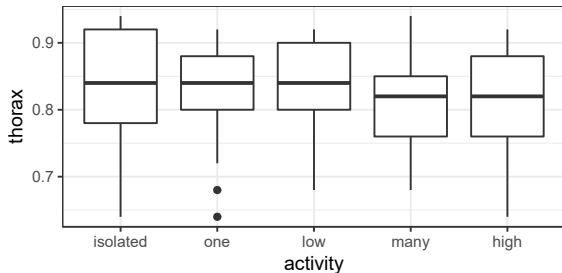
```
library(cowplot)
plot_grid(P11, P12)
```



Выбросов нет

Нет ли коллинеарности

```
ggplot(fly, aes(x = activity, y = thorax)) + geom_boxplot()
```



Коллинеарности предикторов нет

Гипотеза: Продолжительность жизни зависит от половой активности

Модель:

$$Longivity_i = \beta_0 + \beta_1 Thorax_i + \beta_2 I_{isolated} + \beta_3 I_{one} + \beta_4 I_{many} + \beta_5 I_{low} + Interactions + \varepsilon_i$$

$$\varepsilon_i \sim N(0, \sigma^2)$$

Код для подгонки модели

```
mod_formula <- longevity ~ thorax*activity
M1 <- lm(mod_formula, data = fruitfly)
```

```
library(car)
Anova(M1)
```

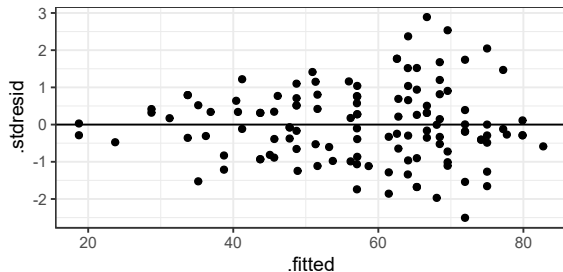
```
# Anova Table (Type II tests)
#
# Response: longevity
#
```

	Sum Sq	Df	F value	Pr(>F)
# thorax	12368.4	1	107.774	< 2.2e-16 ***
# activity	9634.6	4	20.988	5.503e-13 ***
# thorax:activity	24.3	4	0.053	0.9947
# Residuals	13083.0	114		

```
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Диагностика модели

```
M1_diagn <- fortify(M1)
ggplot(M1_diagn, aes(x = .fitted, y = .stdresid)) + geom_point() + geom_hline
```



Мы не можем доверять результатам оценки, так как присутствуют явные признаки гетероскедастичности

Generalized Least Squares

Обобщенный метод наименьших квадратов (Generalized Least Squares)

Суть обычного метода наименьших квадратов OLS:

Ищем вектор \mathbf{b} при котором $\Sigma \mathbf{e}^2 = \min$

Суть GLS:

Ищем вектор \mathbf{b} при котором $\Sigma(\mathbf{e}' \times \mathbf{W}) = \min$

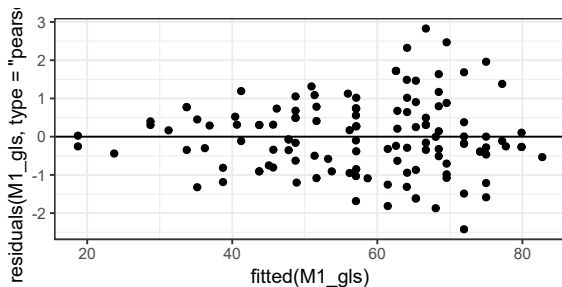
Матрица \mathbf{W} - весовая матрица

Если $\mathbf{W} = \mathbf{I}$, то GLS = OLS.

GLS модель и ее диагностика

```
library(nlme)
M1_gls <- gls(mod_formula, data = fruitfly)

Pl_resid_M1_gls <- qqplot(x = fitted(M1_gls), y = residuals(M1_gls, type = "pearson"),
Pl_resid_M1_gls
```



Особенности функции `glS()`

Если ничего не менять, функция `glS()` дает результаты полностью идентичные результатам функции `lm()`.

Для оценки параметров по умолчанию используется Restricted Maximum Likelihood (REML). Этот метод дает более точные оценки случайных факторов, чем обычный ML.

Внимание! Модели, подобранные с помощью REML, можно сравнивать только если у них одинаковая фиксированная часть!

Основная идея: Дисперсия закономерно изменяется в ответ на влияние некоторой ковариаты.

Задача: подобрать функцию, которая свяжет величину дисперсии с ковариатой дисперсии так, чтобы ML была бы максимальной.

Для подбора оптимальной структуры дисперсии мы будем работать со случайной частью модели, поэтому вместо ML оценки производятся с помощью REML.

Дисперсия зависит от непрерывной ковариаты

Фиксированная структура дисперсии: `varFixed()`

Дисперсия изменяется пропорционально значениям ковариаты дисперсии

$$\varepsilon_i \sim N(0, \sigma^2 \times VC_i)$$

Предположим, что дисперсия меняется пропорционально размеру груди мух (thorax).

```
M2_gls <- gls(mod_formula, data = fly, weights = varFixed( ~ thorax))
```

Вопрос: Как выяснить, стала ли модель лучше?

Можем сравнить две модели при помощи AIC

```
AIC(M1_gls, M2_gls)
```

```
#           df      AIC  
# M1_gls  11 892.2724  
# M2_gls  11 889.7385
```

Можем сравнить две модели при помощи AIC

```
AIC(M1_gls, M2_gls)
```

```
#           df      AIC  
# M1_gls  11 892.2724  
# M2_gls  11 889.7385
```

Стало лучше! Но может есть и другие зависимости?

Степенная зависимость дисперсии от ковариаты: `varPower()`

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times |VC|^{2\delta})$$

Параметр δ неизвестен и требует оценки

Если $\delta = 0$, то структура дисперсии будет аналогична структуре дисперсии в “обычной” регрессионной модели, где $\varepsilon \sim N(0, \sigma^2)$

Важно! Если значения ковариаты дисперсии могут принимать значение равное нулю, то такая форма структуры дисперсии не определена и использоваться не может.

```
M3_gls <- gls(mod_formula, data = fly, weights = varPower(form = ~ thorax))
```

Что произошло в результате работы функции `varPower()`?

```
summary(M3_gls)
```

Часть вывода `summary(M3_gls)`

Variance function:

Structure: Power of variance covariate

Formula: ~thorax

Parameter estimates:

power

1.987254

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times |VC|^{2\delta})$$

Оценка параметра δ

```
M3_gls$modelStruct
```

```
# varStruct parameters:
```

```
# power
```

```
# 1.987254
```

Степенная зависимость дисперсии от ковариаты для разных уровней дискретного фактора

```
M4_gls <- gls(mod_formula, data = fly,  
              weights = varPower(form = ~ thorax|activity))
```

Подобранные параметры

```
M4_gls$modelStruct
```

```
# varStruct parameters:  
#      many isolated      one      low      high  
# 1.8615166 1.6814263 0.7859999 1.4189907 3.3338618
```

Экспоненциальная зависимость дисперсии от ковариаты: `varExp()`

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times e^{2\delta \times VC_i})$$

Эта форма структуры дисперсии может применяться для случаев, когда $VC = 0$

Если $\delta = 0$, то структура дисперсии будет аналогична структуре дисперсии в “обычной” регрессионной модели, то есть $\varepsilon_{ij} \sim N(0, \sigma^2)$

```
M5_gls <- gls(mod_formula, data = fly,  
              weights = varExp(form = ~ thorax))  
M6_gls <- gls(mod_formula, data = fly,  
              weights = varExp(form = ~ thorax|activity))
```

Подобранные параметры

```
M5_gls$modelStruct
```

```
# varStruct parameters:  
#      expon  
# 2.443051
```

```
M6_gls$modelStruct
```

```
# varStruct parameters:  
#      many isolated      one      low      high  
# 1.659506 1.962826 2.101393 1.933775 1.441147
```


Усложненная степенная зависимость дисперсии от ковариаты

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times (\delta_1 + |VC|^{2\delta_2})^2)$$

То есть подбирается не только показатель степени $\delta_2 = 0$, но еще и константа $\delta_1 = 0$

При $\delta_1 = 0$ и $\delta_2 = 0$ выражение $\varepsilon_{ij} \sim N(0, \sigma^2 \times (0 + |VC|^0)$ будет эквивалентно $\varepsilon_{ij} \sim N(0, \sigma^2)$

```
M7_gls <- gls(mod_formula, data = fly,
              weights = varConstPower(form = ~ thorax))
M8_gls <- gls(mod_formula, data = fly,
              weights = varConstPower(form = ~ thorax|activity))
```

Что произошло в результате работы функции `varConstPower()`?

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times (\delta_1 + |VC|^{2\delta_2})^2)$$

```
M7_gls$modelStruct
```

```
# varStruct parameters:  
#      const      power  
# -15.854472  1.987251
```

```
M8_gls$modelStruct
```

```
# varStruct parameters:  
#   const.many const.isolated  const.one  const.low  const.high  power  
#   -17.19168748  -0.03472110   0.04990109   0.16842986  -  
0.96134849  -0.56205148  
# power.isolated  power.one  power.low  power.high  
#    3.89001634   2.70691989  8.26979392  3.08712351
```

Дисперсия зависит от дискретного фактора

Разные дисперсии для разных уровней категориальных предикторов: `varIdent()`

$$\varepsilon_{ij} \sim N(0, \sigma_j^2)$$

При построении моделей с такой структурой дисперсии подбирается $k - 1$ новых параметров, где k — количество уровней категориального предиктора.

```
M9_gls <- gls(mod_formula, data = fly,  
              weights = varIdent(form = ~1|activity))
```

Что произошло в результате работы функции `varIdent()`?

```
summary(M9_gls)
```

Часть вывода `summary(M9_gls)`

Variance function:`

Structure: Different standard deviations per stratum

Formula: ~1 | activity

Parameter estimates:

	many	isolated	one	low	high
	1.0000000	1.4269619	1.5332811	1.3764655	0.8608559

$$\varepsilon_{ij} \sim N(0, \sigma_j^2)$$

Т.е. в выводе `summary()` присутствуют оценки σ_j^2

Комбинированная структура дисперсии: varComb()

```
M10_gls <- gls(mod_formula, data = fly,
               weights = varComb(varIdent(form = ~ 1|activity),
                                varFixed(~ thorax)))

M11_gls <- gls(mod_formula, data = fly,
               weights = varComb(varIdent(form = ~ 1|activity),
                                varPower(form = ~ thorax)))

M12_gls <- gls(mod_formula, data = fly,
               weights = varComb(varIdent(form = ~ 1| activity),
                                varExp(form = ~ thorax)))

M13_gls <- gls(mod_formula, data = fly,
               weights = varComb(varIdent(form = ~ 1|activity),
                                varConstPower(form = ~ thorax)))
```

Моделирование гетерогенности дисперсий - финальная модель

Находим финальную модель

```
AICs <- AIC(M1_gls, M2_gls, M3_gls,  
            M4_gls, M5_gls, M6_gls,  
            M7_gls, M8_gls, M9_gls,  
            M10_gls, M12_gls, M13_gls)
```

AICs

#		df	AIC
#	M1_gls	11	892.2724
#	M2_gls	11	889.7385
#	M3_gls	12	888.3362
#	M4_gls	16	889.2829
#	M5_gls	12	888.5790
#	M6_gls	16	888.9346
#	M7_gls	13	890.3362
#	M8_gls	21	896.5485
#	M9_gls	15	889.8115
#	M10_gls	15	888.1942
#	M12_gls	16	888.9848
#	M13_gls	17	890.7429

Финальная модель

```
AICs[AICs$AIC == min(AICs$AIC), ]
```

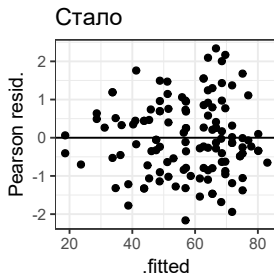
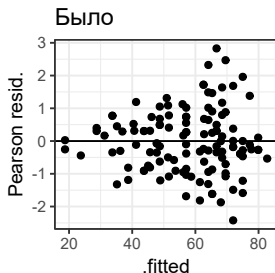
```
#           df      AIC  
# M10_gls 15 888.1942
```

```
summary(M10_gls)$call
```

```
# gls(model = mod_formula, data = fly, weights = varComb(varIdent(form = ~1 |  
#           activity), varFixed(~thorax)))
```

Диагностика финальной модели

```
Pl_resid_M1_gls <- Pl_resid_M1_gls + ggtitle("Было") +  
  labs(x = ".fitted", y = "Pearson resid.")  
Pl_resid_M10_gls <- qplot(x = fitted(M10_gls),  
                           y = residuals(M10_gls, type = "pearson")) +  
  geom_hline(yintercept = 0) +  
  ggtitle("Стало") + labs(x = ".fitted", y = "Pearson resid.")  
  
library(cowplot)  
plot_grid(Pl_resid_M1_gls, Pl_resid_M10_gls)
```



Задание: упростите модель

Задание: упростите модель

Для упрощения финальной модели надо изменять фиксированную часть, REML не годится!

```
M10_gls_ML <- update(M10_gls, method = "ML")
drop1(M10_gls_ML, test = "Chi")
```

```
# Single term deletions
```

```
#
```

```
# Model:
```

```
# longevity ~ thorax * activity
```

```
#           Df      AIC      LRT Pr(>Chi)
```

```
# <none>          946.03
```

```
# thorax:activity  4 938.57 0.54333  0.9691
```

Больше ничего упростить нельзя

```
M10_gls_ML2 <- update(M10_gls_ML, .~.-thorax:activity)
drop1(M10_gls_ML2, test = "Chi" )
```

```
# Single term deletions
#
# Model:
# longevity ~ thorax + activity
#           Df      AIC      LRT Pr(>Chi)
# <none>          938.57
# thorax      1 1033.25 96.673 < 2.2e-16 ***
# activity    4 1000.93 70.354 1.911e-14 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Финальная модель и подготовка визуализации

```
M10_final <- update(M10_gls_ML2, method = "REML")

library(dplyr)
new_data <- fly %>% group_by(activity) %>%
  do(data.frame(thorax = seq(min(.$thorax), max(.$thorax), length.out = 100))

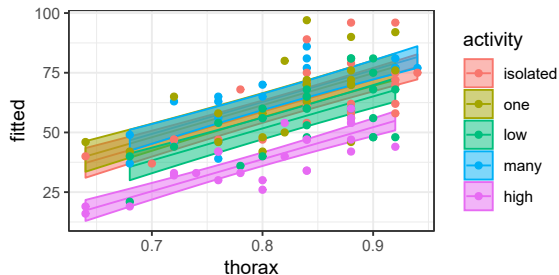
X <- model.matrix(~ thorax + activity, data = new_data)
b <- coef(M10_final)

new_data$fitted <- X%*%b

new_data$SE <- sqrt(diag(X %*% vcov(M10_final) %*% t(X)))
```

Визуализация финальной модели

```
ggplot(new_data, aes(x = thorax, y = fitted, color = activity)) +  
  geom_line() +  
  geom_ribbon(aes(ymin = fitted - 2 * SE,  
                 ymax = fitted + 2 * SE,  
                 fill = activity), alpha = 0.5) +  
  geom_point(data = fly, aes(x = thorax, y = longevity))
```



Моделирование структуры дисперсии при наличии случайных факторов

Рост крыс при разной диете

```
data("BodyWeight")  
bw <- as.data.frame(BodyWeight)  
head(bw, 14)
```

#	weight	Time	Rat	Diet
# 1	240	1	1	1
# 2	250	8	1	1
# 3	255	15	1	1
# 4	260	22	1	1
# 5	262	29	1	1
# 6	258	36	1	1
# 7	266	43	1	1
# 8	266	44	1	1
# 9	265	50	1	1
# 10	272	57	1	1
# 11	278	64	1	1
# 12	225	1	2	1
# 13	230	8	2	1
# 14	230	15	2	1

Три группы крыс, содержались при разных условиях кормления 64 дня. Каждую крысу взвешивали с определенной периодичностью.

Всего было изучено 16 особей

Задача:

Построить модель, которая дала бы ответ на вопрос, изменяется ли характер роста крыс в зависимости от типа диеты?

пример из книги Pinheiro and Bates, 2000

оригинальное исследование Hand and Crowder, 1996

Решение: Неправильная модель

```
M1 <- gls(weight ~ Time*Diet, data = bw)
```

Вопрс: Почему такая модель неправильная?

Решение: Неправильная модель

```
M1 <- gls(weight ~ Time*Diet, data = bw)
```

Вопрс: Почему такая модель неправильная?

Важно! Строить простую линейную модель в данном случае *некорректно!*

- ▶ Дизайн эксперимента изначально включает случайный фактор Rat. Здесь мы имеем дело с повторными наблюдениями одного и того же объекта.
- ▶ Однако мы рассмотрим M1 для демонстрации того, что происходит, если не учитывать этой особенности экспериментального дизайна.

Anova(M1)

```
# Analysis of Deviance Table (Type II tests)
#
# Response: weight
#           Df      Chisq Pr(>Chisq)
# Time       1    19.5541  9.779e-06 ***
# Diet       2 2228.7639 < 2.2e-16 ***
# Time:Diet   2     3.5865    0.1664
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Решение: Модель со случайными факторами

Задание: напишите код для модели, включающей случайные факторы.

Решение: Модель со случайными факторами

Задание: напишите код для модели, включающей случайные факторы.

```
M2 <- lme(weight ~ Time*Diet, data = bw, random = ~1|Rat)
```

```
M3 <- lme(weight ~ Time*Diet, data = bw, random = ~1 + Time|Rat)
```

Решение: Модель со случайными факторами

Задание: напишите код для модели, включающей случайные факторы.

```
M2 <- lme(weight ~ Time*Diet, data = bw, random = ~1|Rat)
```

```
M3 <- lme(weight ~ Time*Diet, data = bw, random = ~1 + Time|Rat)
```

Какую из моделей выбрать?

Решение: Модель со случайными факторами

Задание: напишите код для модели, включающей случайные факторы.

```
M2 <- lme(weight ~ Time*Diet, data = bw, random = ~1|Rat)
M3 <- lme(weight ~ Time*Diet, data = bw, random = ~1 + Time|Rat)
```

Какую из моделей выбрать?

AIC(M2, M3)

```
#      df      AIC
# M2   8 1248.245
# M3  10 1171.720
```

Решение: Пытаемся ответить на вопрос исследования

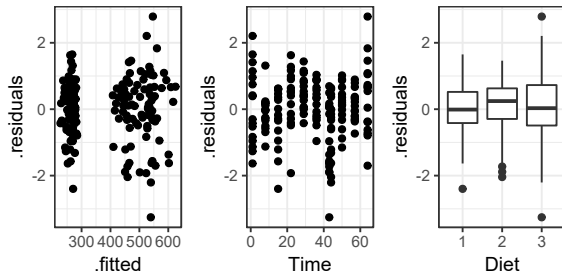
Anova(M3)

```
# Analysis of Deviance Table (Type II tests)
#
# Response: weight
#           Chisq Df Pr(>Chisq)
# Time      82.592  1 < 2.2e-16 ***
# Diet     170.701  2 < 2.2e-16 ***
# Time:Diet  15.149  2  0.0005135 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Наличие взаимодействия говорит о том, что экспериментальное воздействие повлияло на характер роста крыс.

Но! можем ли мы доверять этим результатам?

Диагностика модели



Есть некоторые признаки гетерогенности дисперсии.

Моделируем структуру дисперсии

```
M3_1 <- update(M3, weights = varIdent(form = ~ 1|Diet))
M3_2 <- update(M3, weights = varPower(form = ~Time))
M3_3 <- update(M3, weights = varPower(form = ~Time|Diet))
# M3_4 <- update(M3, weights = varConstPower(form = ~Time))
M3_5 <- update(M3, weights = varExp(form = ~Time))
M3_6 <- update(M3, weights = varExp(form = ~Time|Diet))
M3_7 <- update(M3, weights = varComb(varExp(form = ~Time),
                                     varIdent(form = ~1|Diet)))
M3_8 <- update(M3, weights = varComb(varPower(form = ~Time),
                                     varIdent(form = ~1|Diet)))
```

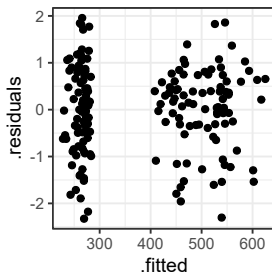
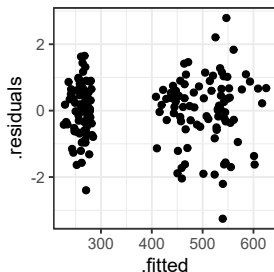
Выбираем лучшую модель

```
AIC(M3, M3_1, M3_2, M3_3, M3_5, M3_6, M3_7, M3_8)
```

#		df	AIC
#	M3	10	1171.720
#	M3_1	12	1163.961
#	M3_2	11	1172.534
#	M3_3	13	1157.534
#	M3_5	11	1173.609
#	M3_6	13	1154.622
#	M3_7	13	1164.982
#	M3_8	13	1162.166

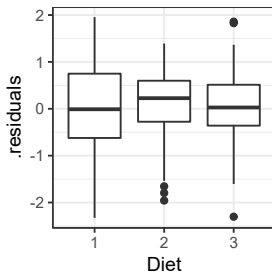
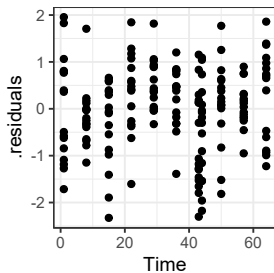
Диагностика модели

```
M3_6_diagn <- data.frame(.fitted = fitted(M3_6),  
                          .residuals = residuals(M3_6, type = "pearson"),  
                          Diet = bw$Diet,  
                          Time = bw$Time)  
  
Pl4 <- ggplot(M3_6_diagn, aes(x=.fitted, y=.residuals) ) + geom_point()  
Pl5 <- ggplot(M3_6_diagn, aes(x=Time, y=.residuals) ) + geom_point()  
Pl6 <- ggplot(M3_6_diagn, aes(x=Diet, y=.residuals) ) + geom_boxplot()  
grid.arrange(Pl1, Pl4, nrow = 1)
```



Диагностика модели

```
grid.arrange(Pl5, Pl6, nrow = 1)
```



Отвечаем на вопрос

Anova(M3_6)

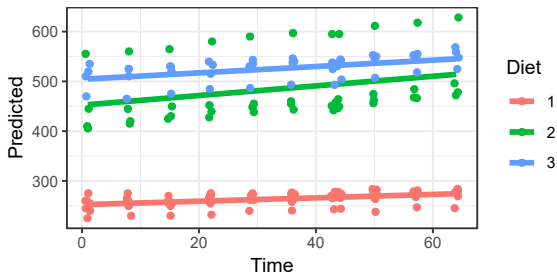
```
# Analysis of Deviance Table (Type II tests)
#
# Response: weight
#           Chisq Df Pr(>Chisq)
# Time      83.169  1 < 2.2e-16 ***
# Diet     169.262  2 < 2.2e-16 ***
# Time:Diet  17.301  2  0.0001751 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Взаимодействие факторов осталось!



Смотрим на предсказания модели

```
MyData <- expand.grid(Time = unique(bw$Time), Diet = factor(1:3))  
  
MyData$Predicted <- predict(M3_6, newdata = MyData, level = 0)  
  
ggplot(MyData, aes(x = Time, y = Predicted, color = Diet)) +  
  geom_line( size = 1.5) +  
  geom_point(data = bw, aes(x = Time, y = weight),  
            position = position_jitter())
```



Углы наклона в разных группах различаются!

При наличии признаков гетероскедастичности можно пойти тремя путями

1. Произвести преобразование зависимой переменной
2. Включить в модель элемент, описывающий связь дисперсии с ковариатой дисперсии
3. Если природа данных позволяет, то построить модель, основанную на распределении Пуассона или отрицательном биномиальном распределении.

- ▶ Zuur, A.F. et al. 2009. Mixed effects models and extensions in ecology with R. - Statistics for biology and health. Springer, New York, NY.
- ▶ Pinheiro J, Bates D (2000) Mixed effects models in S and S-Plus. Springer-Verlag, New York, USA