Дисперсионный анализ, часть 2

Линейные модели…

Марина Варфоломеева, Вадим Хайтов

Кафедра Зоологии беспозвоночных, Биологический факультет, СПбГУ

Многофакторный дисперсионный анализ

- Модель многофакторного дисперсионного анализа
- Взаимодействие факторов
- Несбалансированные данные, типы сумм квадратов
- Многофакторный дисперсионный анализ в R
- Дисперсионный анализ в матричном виде

Вы сможете

- Проводить многофакторный дисперсионный анализ и интерпретировать его результаты с учетом взаимодействия факторов

Данные

Пример: Удобрение и беспозвоночные

Влияет ли добавление азотных и фосфорных удобрений на беспозвоночных?

Небольшие искуственные субстраты экспонировали в течение разного времени в верхней части сублиторали (Hall et al., 2000).

Зависимая переменная:

- `richness` — Число видов

Факторы:

- `time` — срок экспозиции (2, 4 и 6 месяцев)
- `treat` — удобрения (добавляли или нет)

Планировали сделать 5 повторностей для каждого сочетания факторов

Данные из Quinn, Keough, 2002

Знакомимся с данными

```
fert <- read.csv(file='data/hall.csv')
str(fert)

# 'data.frame': 29 obs. of  3 variables:
#  $ TREAT   : Factor w/ 2 levels ”control”,”nutrient”: 1 1 1 1 1 1 1 1 1 1 ...
#  $ TIME    : int  2 2 2 2 2 4 4 4 4 4 ...
#  $ RICHNESS: int  5 7 5 7 5 20 18 20 18 17 ...
```

```
# Для удобства названия переменных маленькими буквами
colnames(fert) <- tolower(colnames(fert))
# Время делаем фактором
fert$time <- factor(fert$time)
levels(fert$time)

# [1] ”2” ”4” ”6”
```

Пропущенные значения

```
colSums(is.na(fert))
```

```
#    treat      time richness
#        0         0         0
```

- Нет пропущенных значений

Объемы выборок в группах

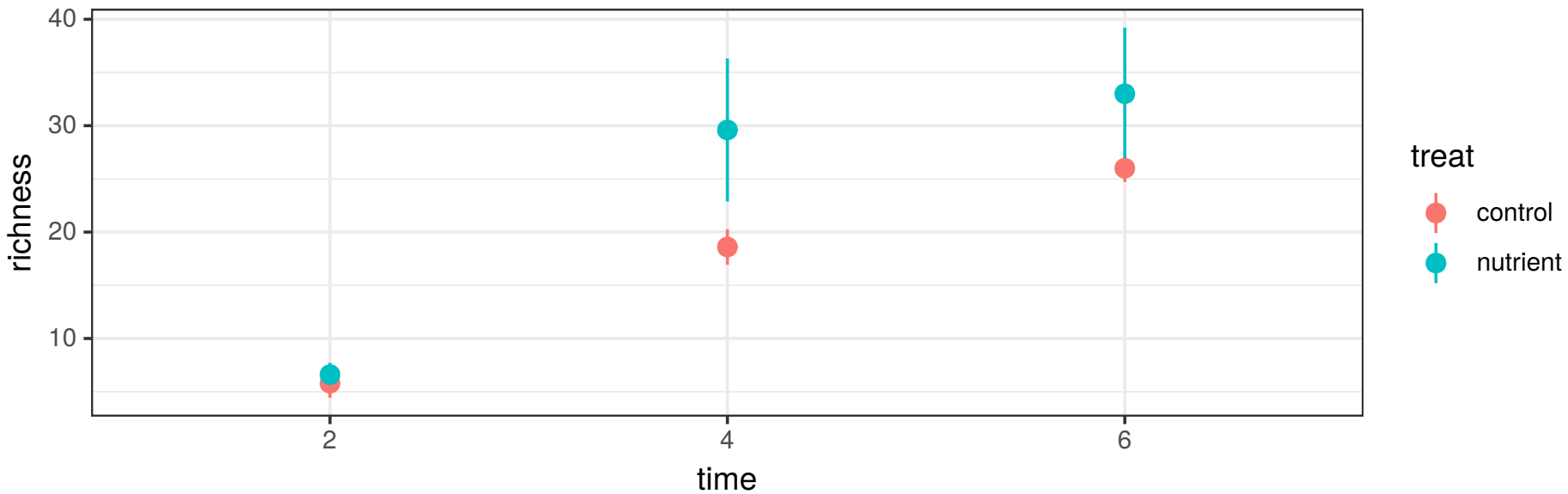
```
table(fert$time, fert$treat)
```

```
#
#     control nutrient
#   2       5        5
#   4       5        5
#   6       4        5
```

- Группы разного размера

Посмотрим на график

```
library(ggplot2)
theme_set(theme_bw())
gg_rich <- ggplot(data = fert, aes(x = time, y = richness, colour = treat)) +
  stat_summary(geom = 'pointrange', fun.data = mean_cl_normal)
gg_rich
```



- Вполне возможно, здесь есть гетерогенность дисперсий.

Преобразовываем данные

Зависимая переменная `richness` – это счетная величина. Она подчиняется распределению Пуассона (и чем больше ее среднее значение, тем больше дисперсия).

Правильно было бы воспользоваться обобщенными линейными моделями с Пуассоновским распределением ошибок вместо нормального.

Но сейчас мы с вами попробуем действовать грубее (пока еще не разобрались, как это делать правильно).

Давайте мы попробуем преобразовать зависимую переменную, чтобы ее распределение стало больше походить на нормальное. Это может помочь, а может и нет.

```
fert$log_rich <- log10(fert$richness + 1)
```

Многофакторный дисперсионный анализ

Многофакторный дисперсионный анализ

Дисперсионный анализ становится многофакторным, если в модели используется несколько дискретных факторов.

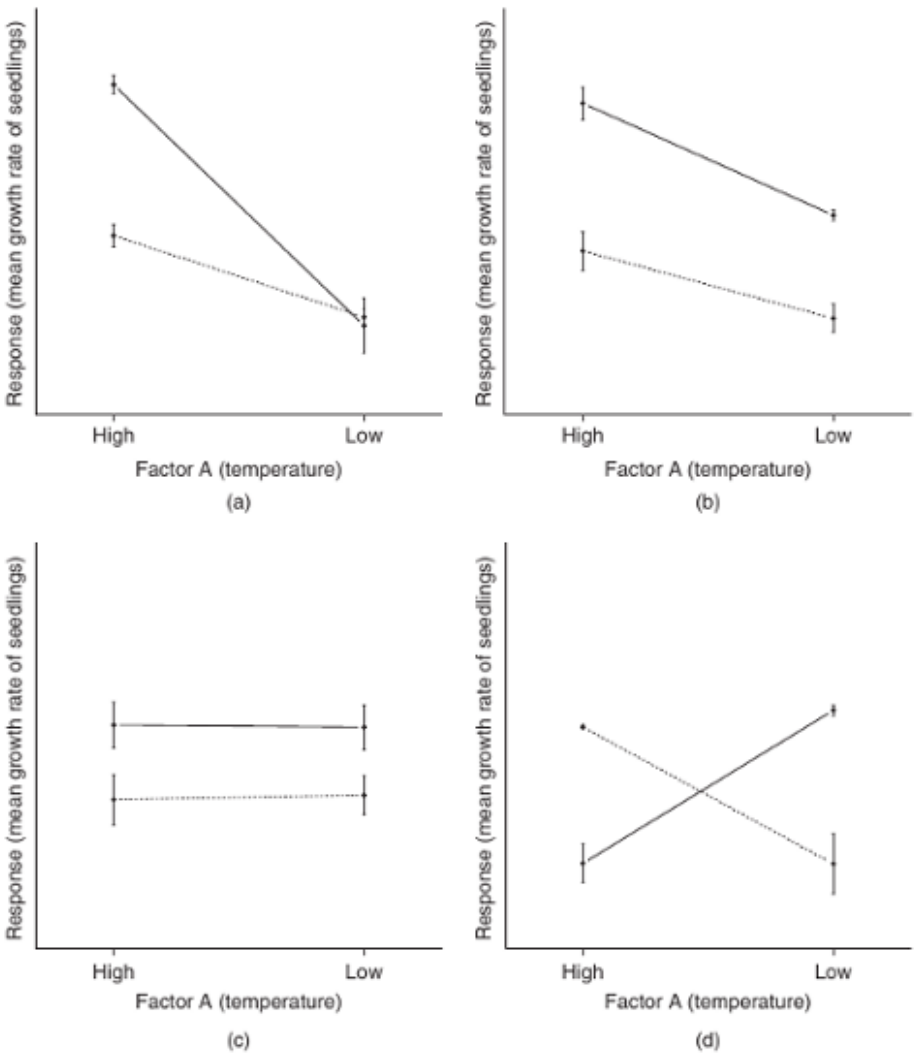
В таком анализе появляется взаимодействие факторов.

Взаимодействие факторов возникает, когда у одного фактора эффект разный в зависимости от уровней другого.

Разберемся с этим на схемах.

Что такое взаимодействие дискретных предикторов

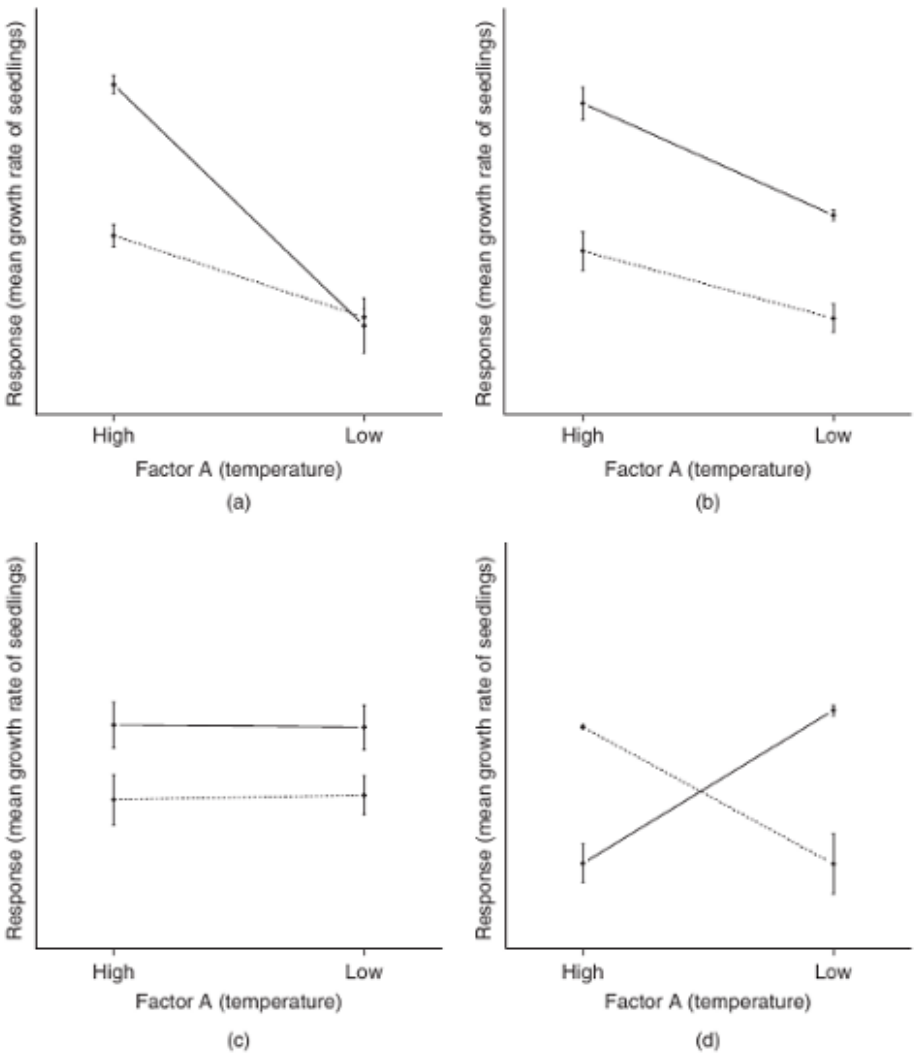
Взаимодействие факторов - когда эффект фактора B разный в зависимости от уровней фактора A и наоборот.



На каких рисунках есть взаимодействие факторов?

Что такое взаимодействие дискретных предикторов

Взаимодействие факторов - когда эффект фактора B разный в зависимости от уровней фактора A и наоборот.



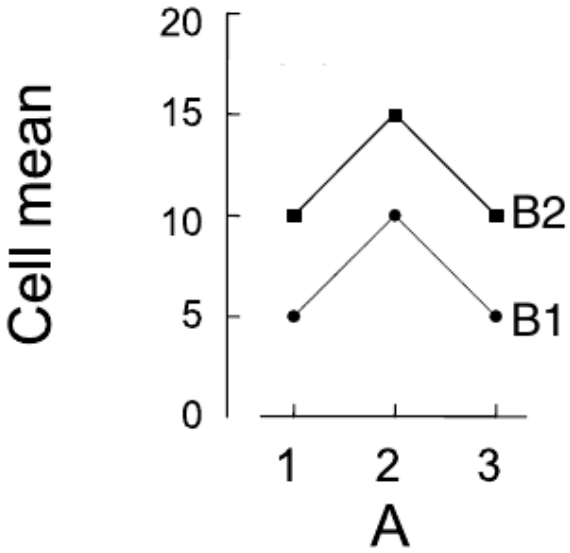
На каких рисунках есть взаимодействие факторов?

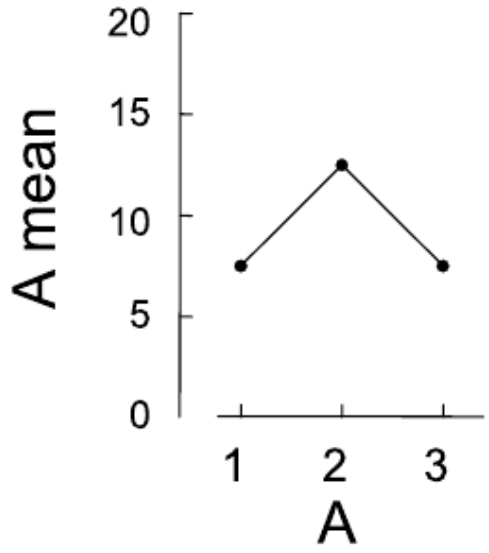
- b, c - нет взаимодействия (эффект фактора B одинаковый для групп по фактору A, линии для разных групп по фактору B на графиках расположены параллельно)
- a, d - есть взаимодействие (эффект фактора B разный для групп по фактору A, на графиках линии для разных групп по фактору B расположены под наклоном).

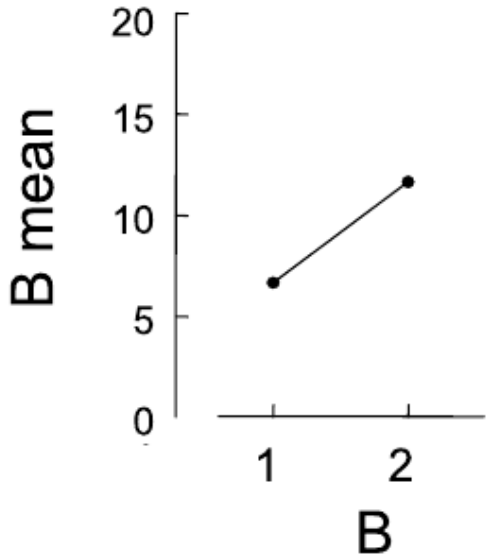
Рисунок из Logan, 2010, fig.12.2

Влияют ли главные эффекты и взаимодействие?

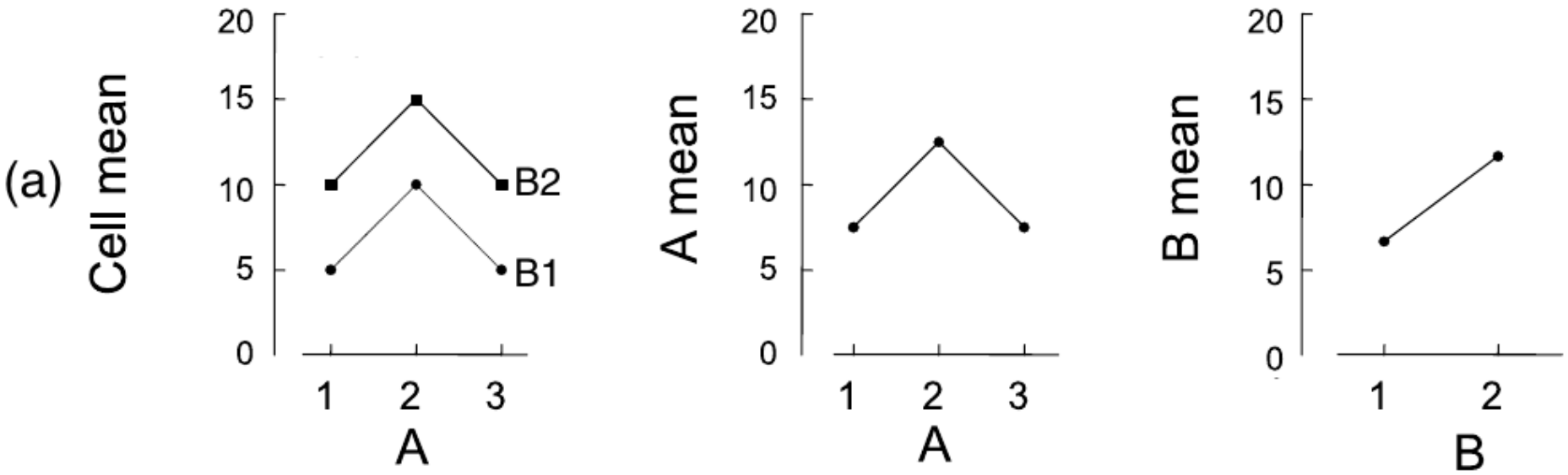
(a)







Влияют ли главные эффекты и взаимодействие?

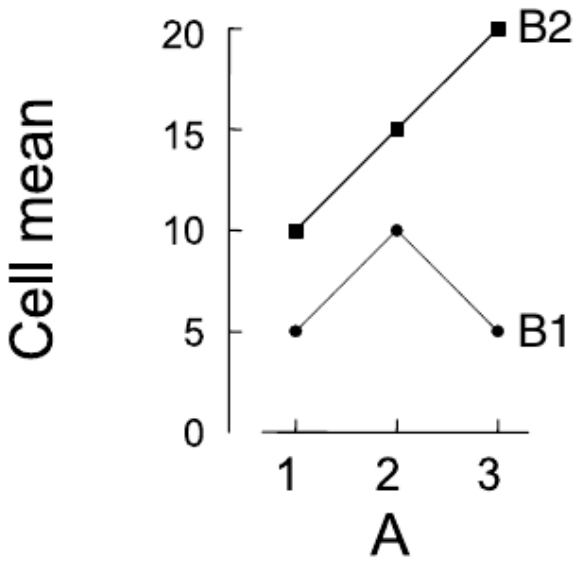


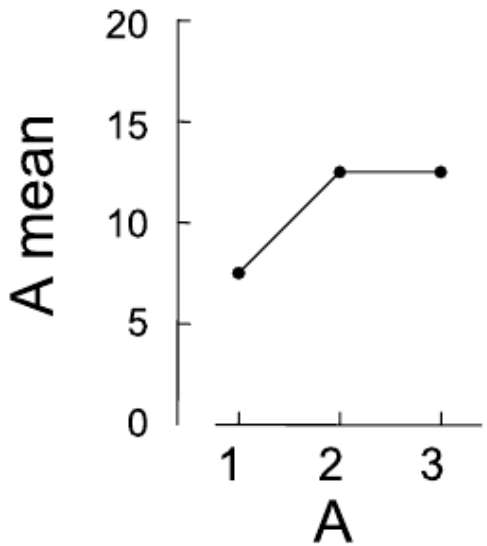
- взаимодействие не значимо, и не мешает интерпретировать эффекты факторов.
  - фактор A влияет
  - фактор B влияет

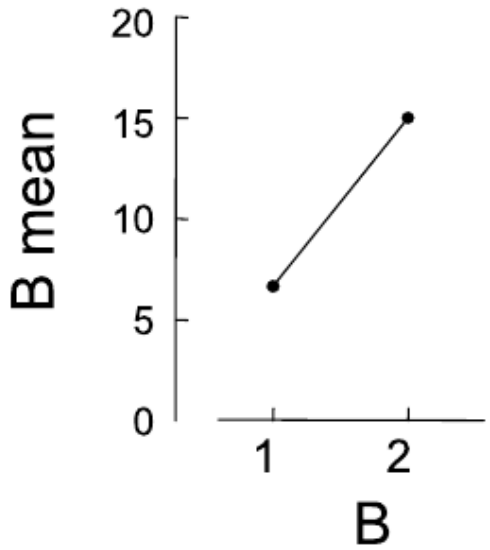
Рисунок из Quinn, Keough, 2002, fig.9.3

Влияют ли главные эффекты и взаимодействие?

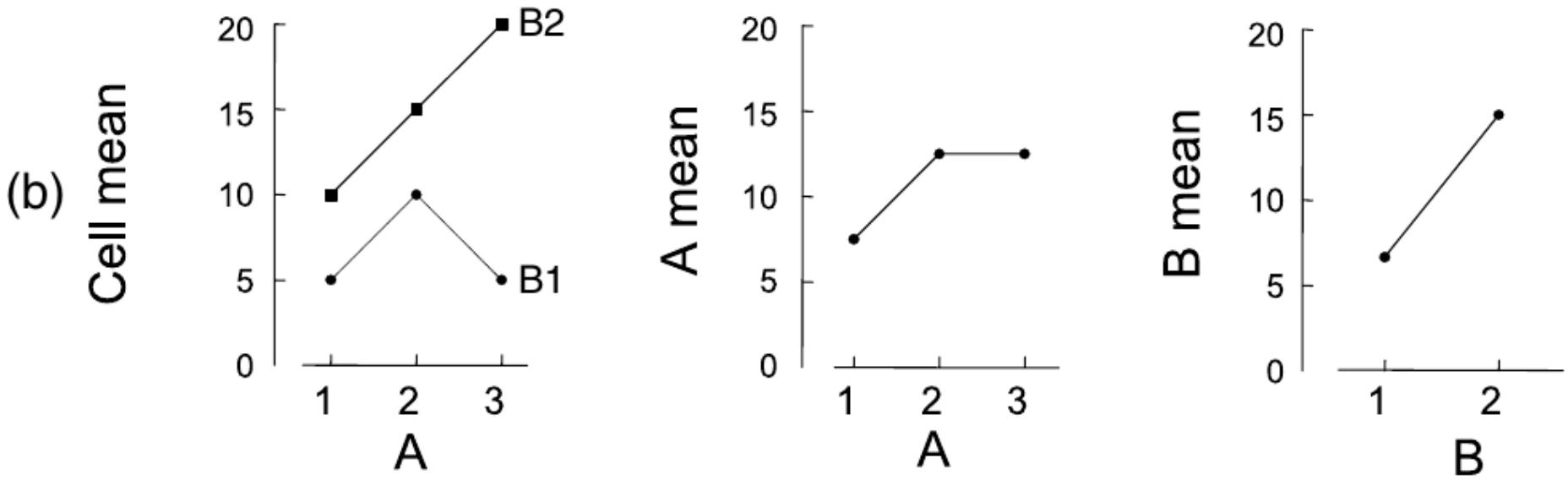
(b)







Влияют ли главные эффекты и взаимодействие?

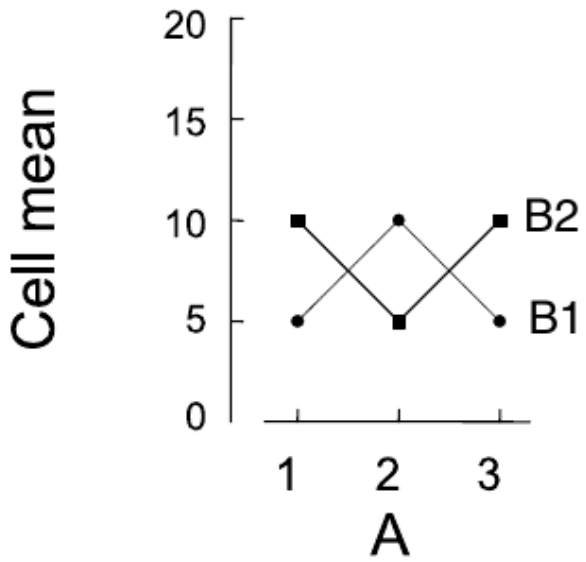


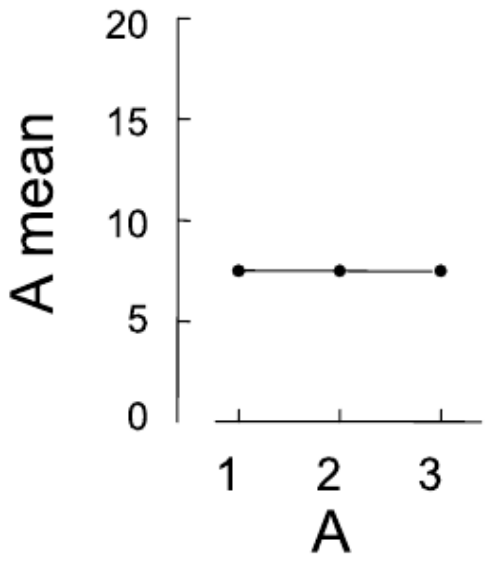
- взаимодействие значимо и мешает интерпретировать влияние факторов отдельно:
  - для B2 зависимая переменная возрастает с изменением уровня A
  - для B1 зависимая переменная возрастает только на A2, но не различается на A1 и A3
- **если смотреть на главные эффекты, можно сделать неправильные выводы (о факторе A)**:
  - фактор A влияет, группы A2 и A3 не отличаются
  - фактор B влияет, в группе B2 зависимая переменная больше, чем в B1

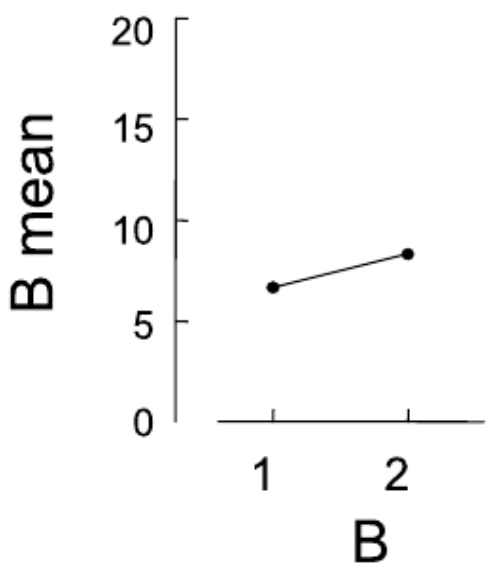
Рисунок из Quinn, Keough, 2002, fig.9.3

Влияют ли главные эффекты и взаимодействие?

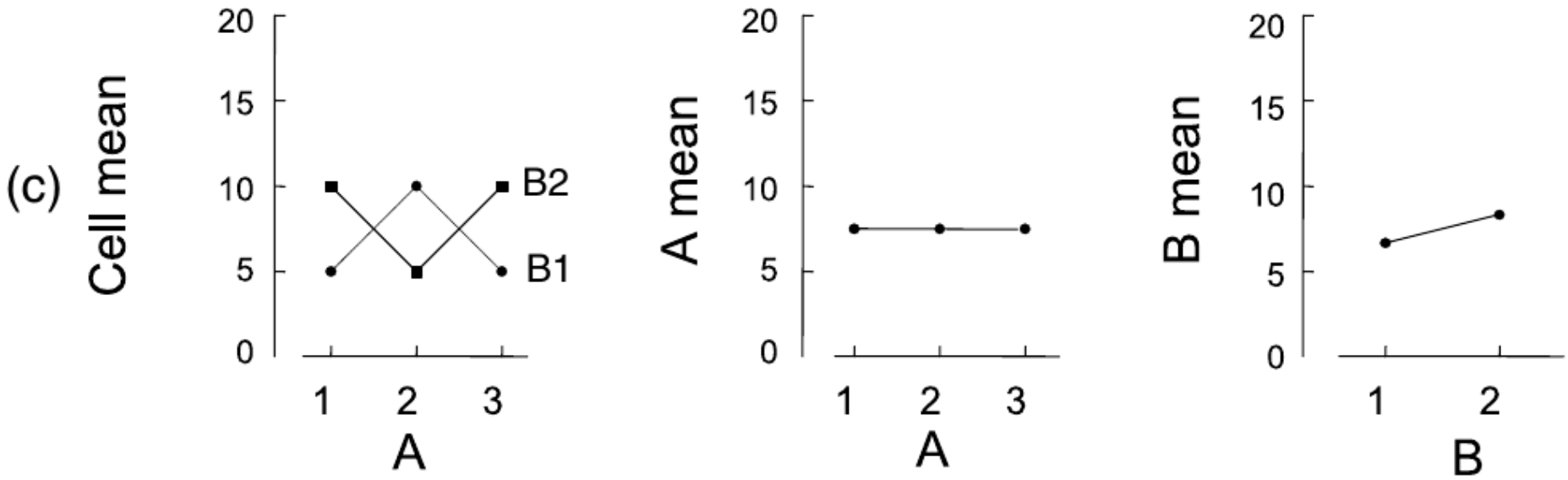
(c)







Влияют ли главные эффекты и взаимодействие?



- взаимодействие значимо и мешает интерпретировать влияние факторов отдельно:
  - на уровне A2 меняется порядок различий уровней фактора B
- **если смотреть на главные эффекты, можно сделать неправильные выводы**:
  - факторы A и B не влияют

Рисунок из Quinn, Keough, 2002, fig.9.3

Взаимодействие факторов может маскировать главные эффекты

Если есть значимое взаимодействие, то - главные эффекты обсуждать не имеет смысла
- пост хок тесты проводятся только для ваимодействия

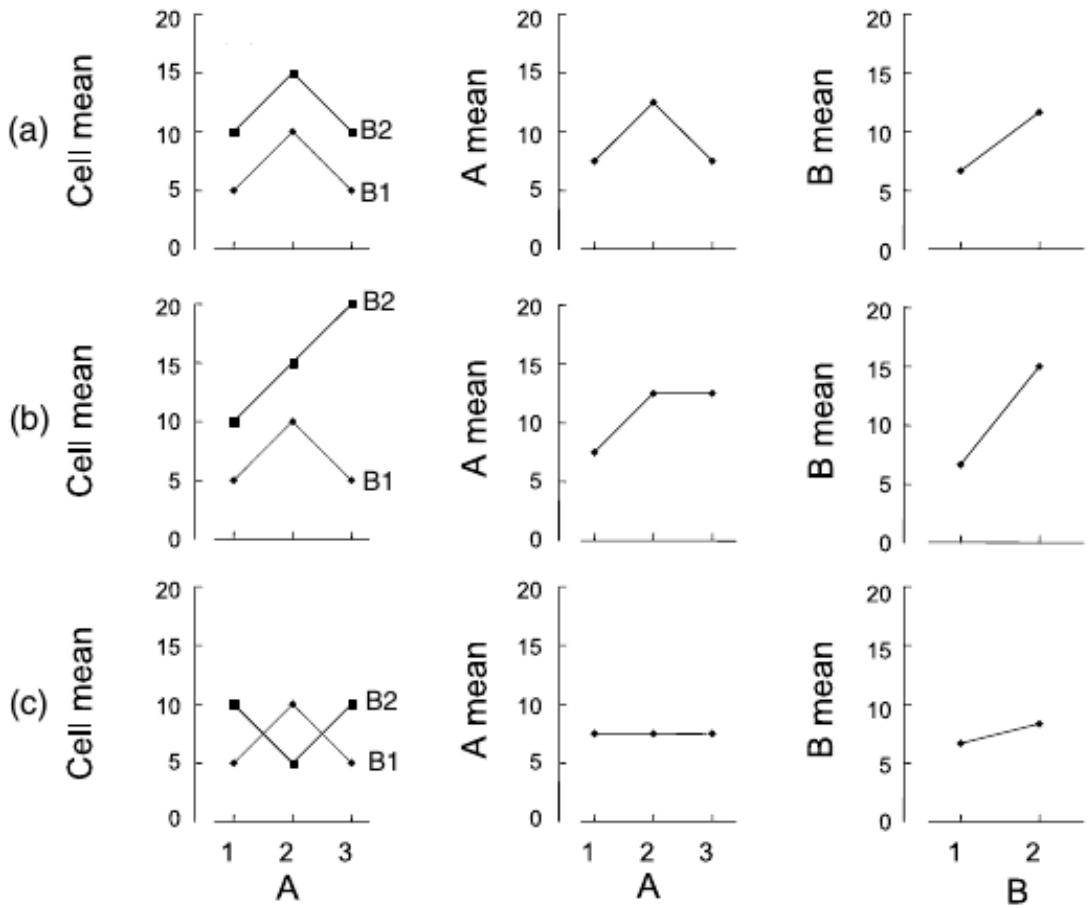


Рисунок из Quinn, Keough, 2002, fig.9.3

Двухфакторный дисперсионный анализ в параметризации индикаторов

Переменные-индикаторы

В нашем примере отклик — видовое богатство, и два дискретных фактора:

- `treat` — 2 уровня (базовый `control`), для кодирования нужна одна переменная.

```
contr.treatment(levels(fert$treat))
```

```
#           nutrient
# control          0
# nutrient         1
```

- `time` — 3 уровня (базовый 2), для кодирования нужно две переменных.

```
contr.treatment(levels(fert$time))
```

```
#   4 6
# 2 0 0
# 4 1 0
# 6 0 1
```

Переменные-индикаторы

Дополнительные переменные понадобятся, чтобы учесть взаимодействие факторов.

Факторы		**Переменные-индикаторы**				
`treat`	`time`	`treatnutrient` x_1	`time4` x_2	`time6` x_3	`treatnutrient:time4` x_4	`treatnutrient:time6` x_5
control	2	0	0	0	0	0
nutrient	2	1	0	0	0	0
control	4	0	1	0	0	0
nutrient	4	1	1	0	1	0
control	6	0	0	1	0	0
nutrient	6	1	0	1	0	1

Уравнение линейной модели в параметризации индикаторов

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{3i} + b_4 x_{4i} + b_5 x_{5i} + e_i$$

- b_0 — значение отклика для контроля через 2 месяца (на базовом уровне обоих факторов)

Отклонения относительно базового уровня обоих факторов:

- b_1 — для удобренных площадок
- b_2 и b_3 — для площадок с экспозицией 4 или 6 мес
- b_4 и b_5 — для удобренных площадок с экспозицией 4 или 6 мес

Подбираем линейную модель в параметризации индикаторов (contr.treatment)

```
mod_treatment <- lm(log_rich ~ treat * time, data = fert)
mod_treatment
```

```
#
# Call:
# lm(formula = log_rich ~ treat * time, data = fert)
#
# Coefficients:
#         (Intercept)        treatnutrient                time4
#             0.82813              0.05040              0.46332
#               time6  treatnutrient:time4  treatnutrient:time6
#             0.60309              0.13814              0.04619
```

Общее уравнение модели

$$\widehat{log\_rich}_i = 0.828 + 0.05 treat_{nutrient\,i} + 0.463 time_{4\,i} + 0.603 time_{6\,i} + \\ + 0.138 treat_{nutrient}\ time_{4\,i} + 0.046 treat_{nutrient}\ time_{6\,i}$$

Двухфакторный дисперсионный анализ в параметризации эффектов

Переменные-эффекты

В нашем примере отклик — видовое богатство, и два дискретных фактора:

- `treat` — 2 уровня (базовый `control`), для кодирования нужна одна переменная.

```
contr.sum(levels(fert$treat))
```

```
#           [,1]
# control     1
# nutrient   -1
```

- `time` — 3 уровня (базовый 2), для кодирования нужно две переменных.

```
contr.sum(levels(fert$time))
```

```
#   [,1] [,2]
# 2    1    0
# 4    0    1
# 6   -1   -1
```

Переменные-эффекты

Дополнительные переменные понадобятся, чтобы учесть взаимодействие факторов.

Факторы		**Переменные-индикаторы**				
`treat`	`time`	`treat1` x_1	`time1` x_2	`time2` x_3	`treat1:time1` x_4	`treat1:time2` x_5
control	2	1	1	0	1	0
nutrient	2	-1	1	0	-1	0
control	4	1	0	1	0	1
nutrient	4	-1	0	1	0	-1
control	6	1	-1	-1	-1	-1
nutrient	6	-1	-1	-1	1	1

Уравнение линейной модели в параметризации эффектов

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + b_3 x_{3i} + b_4 x_{4i} + b_5 x_{5i} + e_i$$

- b_0 — среднее значение отклика по всем данным

Отклонения от общего среднего значений отклика:

- b_1 — в зависимости от тритмента (фактор `treat`)
- b_2 и b_3 — в зависимости от экспозиции (фактор `time`)
- b_4 и b_5 — для тритментов в зависимости от экспозиции (взаимодействие)

Подбираем линейную модель в параметризации эффектов (contr.sum)

```
mod_sum <- lm(log_rich ~ treat * time, data = fert,
              contrasts = list(treat = 'contr.sum', time = 'contr.sum'))
mod_sum
```

```
#
# Call:
# lm(formula = log_rich ~ treat * time, data = fert, contrasts = list(treat = "contr.sum",
#     time = "contr.sum"))
#
# Coefficients:
#  (Intercept)        treat1         time1         time2  treat1:time1
#      1.23952      -0.05592      -0.38619       0.14620       0.03072
# treat1:time2
#     -0.03835
```

Общее уравнение модели

$$\widehat{log\_rich}_i = 1.24 - 0.056 treat_{1\,i} - 0.386 time_{1\,i} + 0.146 time_{2\,i} + \\ + 0.031 treat_{1\,i} time_{1\,i} - 0.038 treat_{1\,i} time_{2\,i}$$

Диагностика линейной модели

Диагностика линейной модели

Нужно проверить, выполняются ли условия применимости
для модели в нужной параметризации

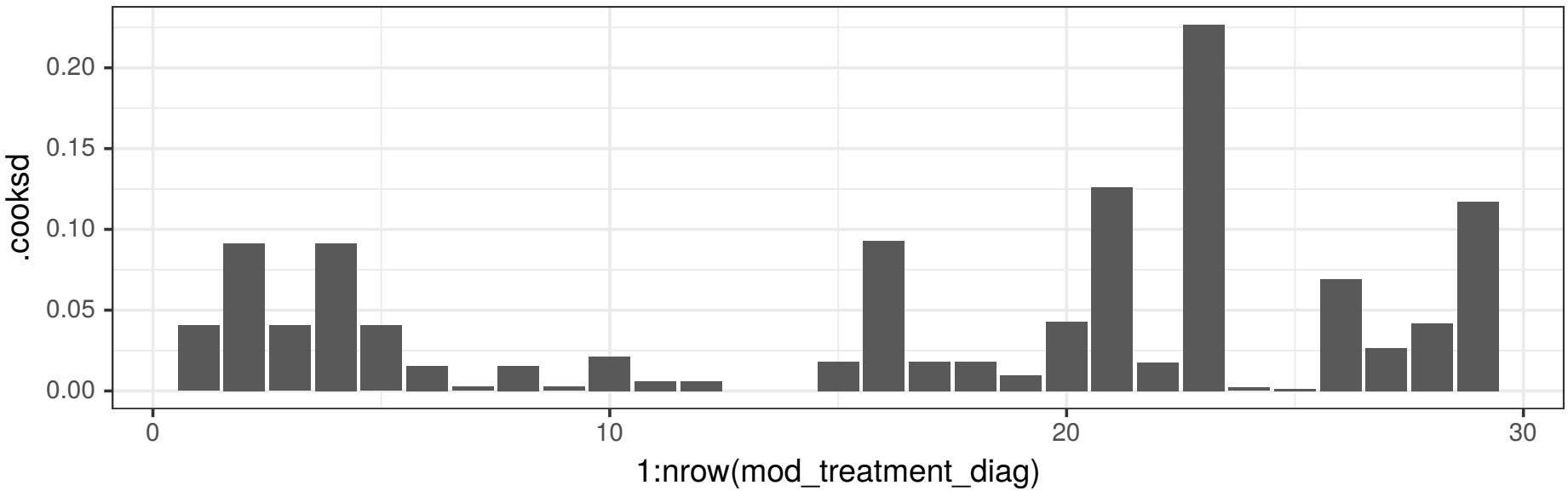
Данные для анализа остатков

```
mod_treatment_diag <- fortify(mod_treatment) # функция из пакета ggplot2
head(mod_treatment_diag, 2)
```

```
#    log_rich   treat time .hat     .sigma    .cooksd
# 1 0.7781513 control    2  0.2 0.05671028 0.04049882
# 2 0.9030900 control    2  0.2 0.05512416 0.09112234
#     .fitted      .resid  .stdresid
# 1 0.8281267 -0.04997549 -0.9858862
# 2 0.8281267  0.07496324  1.4788293
```

График расстояния Кука

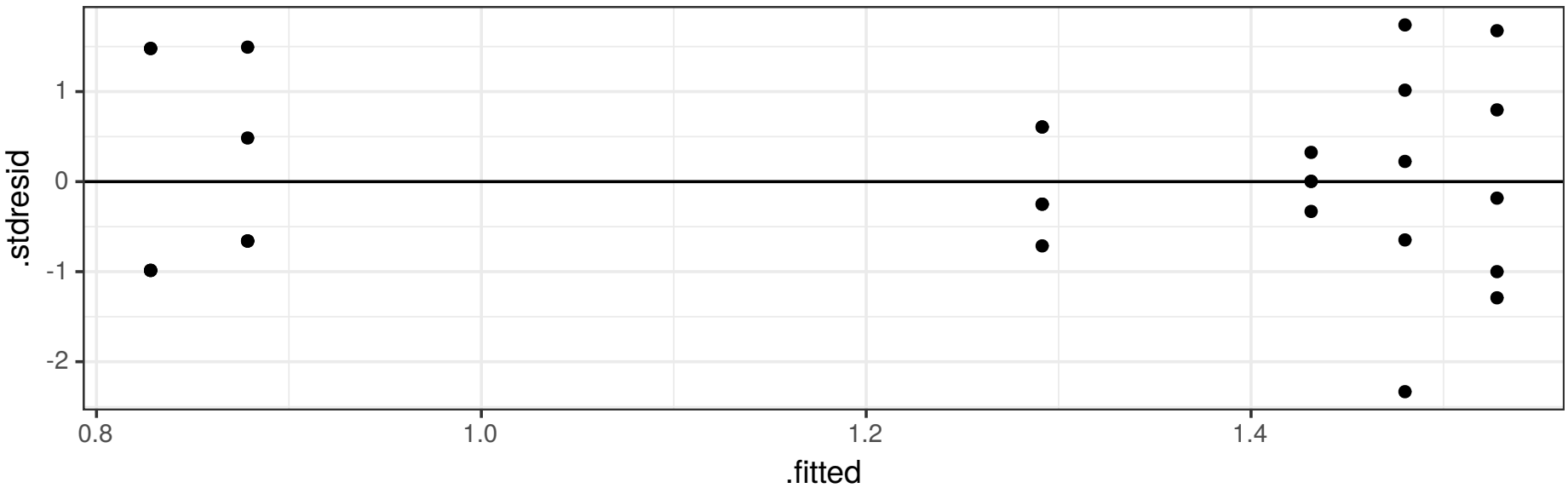
```
ggplot(mod_treatment_diag, aes(x = 1:nrow(mod_treatment_diag), y = .cooksd)) +
  geom_bar(stat = 'identity')
```



- Влиятельных наблюдений нет.

График остатков от предсказанных значений

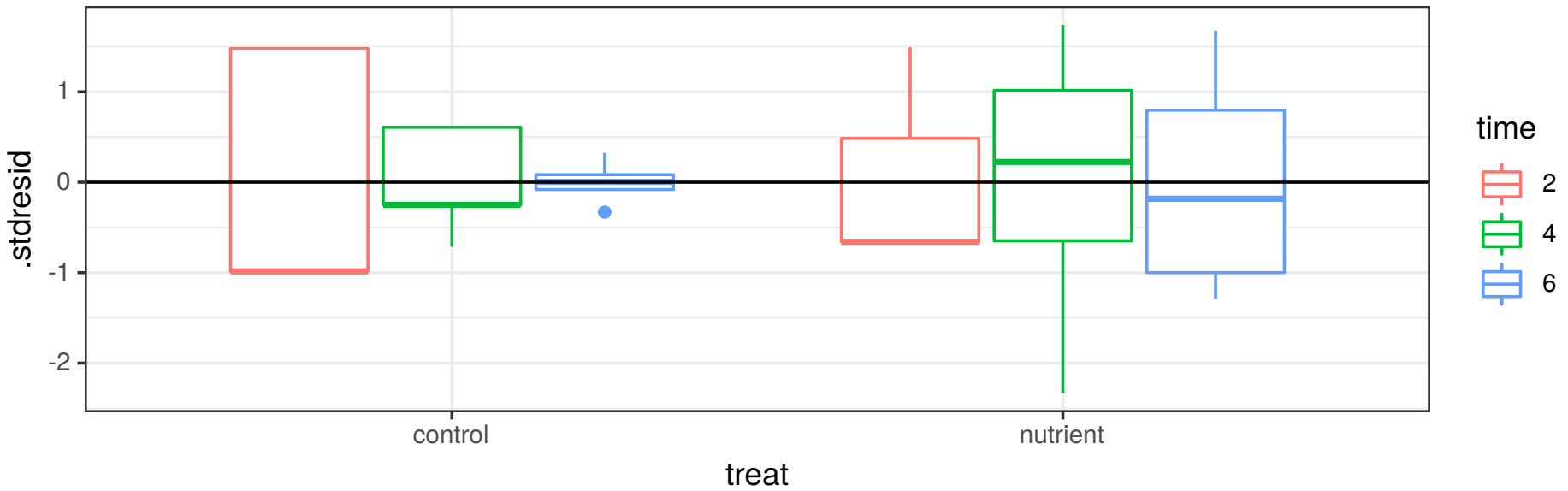
```
gg_resid <- ggplot(data = mod_treatment_diag, aes(x = .fitted, y = .stdresid)) +
  geom_point() + geom_hline(yintercept = 0)
gg_resid
```



- Влиятельных наблюдений нет (все в пределах 3 SD).

График зависимости остатков от предикторов в модели

```
ggplot(data = mod_treatment_diag, aes(x = treat, y = .stdresid, colour = time)) +
  geom_boxplot() + geom_hline(yintercept = 0)
```



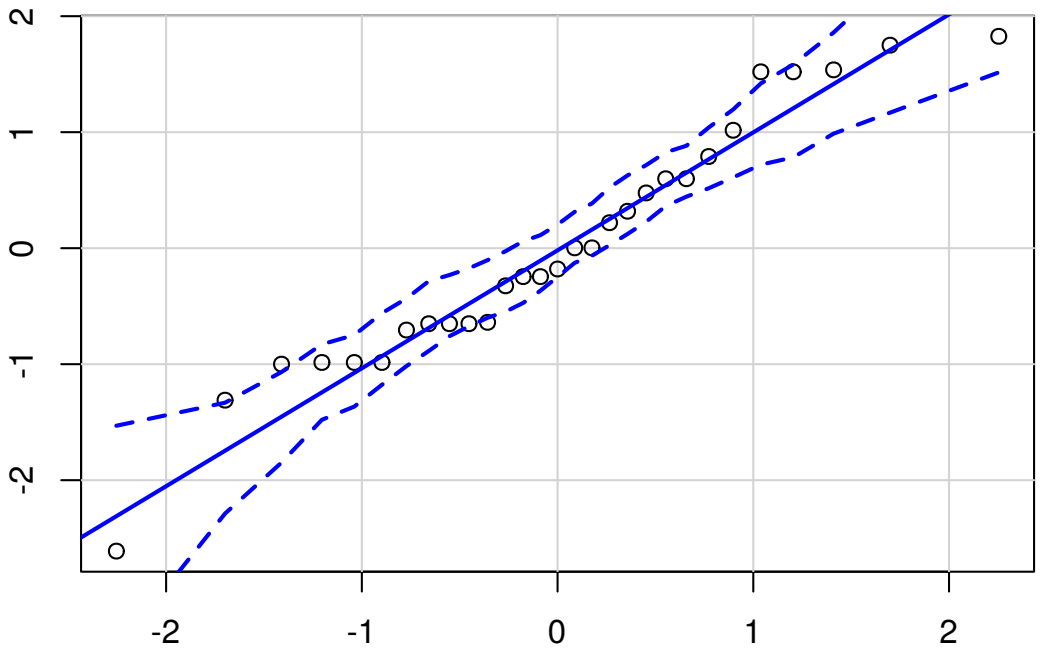
Удобнее смотреть на боксплот.

- Видна гетерогенность дисперсии.

В данном случае это не страшно, т.к. дисперсионный анализ устойчив к ситуации, когда в одной из групп разброс меньше, чем в других (особенно, если данные не слишком несбалансированные) (Underwood, 1997, McGuinness, 2002)

Квантильный график остатков

```
library(car)
qqPlot(mod_treatment, id = FALSE) # функция из пакета car
```



Отклонений от нормального распределения нет.

Несбалансированные данные, типы сумм квадратов

Несбалансированные данные - когда численности в группах по факторам различаются

Например так,

	A1	A2	A3
B1	5	5	5
B2	5	4	5

или так,

	A1	A2	A3
B1	3	8	4
B2	4	7	4

Проблемы из-за несбалансированности данных

- Оценки средних в разных группах с разным уровнем точности (Underwood 1997)
- ANOVA менее устойчив к отклонениям от условий применимости (особенно от гомогенности дисперсий) при разных размерах групп (Quinn Keough 2002, section 8.3)
- Проблемы с расчетом мощности. Если $\sigma^2_\epsilon > 0$ и размеры выборок разные, то $\frac{MS_x}{MS_e}$ не следует F-распределению (Searle et al. 1992).

Проблемы из-за несбалансированности данных

- Оценки средних в разных группах с разным уровнем точности (Underwood 1997)
- ANOVA менее устойчив к отклонениям от условий применимости (особенно от гомогенности дисперсий) при разных размерах групп (Quinn Keough 2002, section 8.3)
- Проблемы с расчетом мощности. Если $\sigma^2_\epsilon > 0$ и размеры выборок разные, то $\frac{MS_x}{MS_e}$ не следует F-распределению (Searle et al. 1992).

- Старайтесь *планировать* группы равной численности!
- Но если не получилось - не страшно:
  - Для фикс. эффектов неравные размеры - проблема при нарушении условий применимости только, если значения доверительной вероятности *p* близки к выбранному критическому уровню значимости α

Суммы квадратов в многофакторном дисперсионном анализе со взаимодействием

Если данные сбалансированы, то …

- взаимодействие и эффекты факторов независимы (в любой параметризации),
- все суммы квадратов и соответствующие тесты можно посчитать в одном анализе,
- результат не зависит от порядка включения факторов в модель.

Если данные несбалансированы, то …

- суммы квадратов для факторов не равны общей сумме квадратов,
- для вычислений используется регрессионный подход (несколько сравнений вложенных моделей),
- результат анализа может зависеть от порядка включения факторов в модель.

Порядок тестирования значимости предикторов в дисперсионном анализе

”Типы сумм квадратов”	I тип	II тип	III тип
Название	Последовательный	Без учета взаимодействий высоких порядков	Иерархический
Порядок расчета SS			

Порядок тестирования значимости предикторов в дисперсионном анализе

”Типы сумм квадратов”	I тип	II тип	III тип
Название	Последовательный	Без учета взаимодействий высоких порядков	Иерархический
Порядок расчета SS	SS(A) SS(B | A) SS(AB | B, A)	SS(A | B) SS(B | A) SS(AB | B, A)	SS(A | B, AB) SS(B | A, AB) SS(AB | B, A)
Величина эффекта зависит от выборки в группе	Да	Да	Нет
Результат зависит от порядка включения факторов в модель	Да	Нет	Нет
Параметризация	Любая	Любая	Только параметризация эффектов
Команда R	aov(), anova()	Anova() (пакет car)	Anova() (пакет car)

Осторожно! Тестируя предикторы в разном порядке, вы тестируете разные гипотезы!

Если несбалансированные данные, выберите подходящий порядок тестирования гипотез

Если данные сбалансированы, то …

- При использовании любого типа сумм квадратов результаты расчетов будут одинаковы.

Если данные несбалансированы, то …

- Результаты зависят от выбранного типа сумм квадратов (т.к. он определяет, какие гипотезы при этом тестируются).

Для несбалансированных данных иногда рекомендуют **суммы квадратов III типа** если есть взаимодействие факторов (Maxwell & Delaney 1990, Milliken, Johnson 1984, Searle 1993, Yandell 1997, Glantz, Slinker 2000). Но при этом **нарушается принцип маргинальности**, поэтому некоторые статистики не любят тех, кто так делает…

Многофакторный дисперсионный анализ в R

Дисперсионный анализ со II типом сумм квадратов

При таком способе, сначала тестируется взаимодействие, затем отдельные факторы в модели без взаимодействия.

```
mod_treatment <- lm(log_rich ~ treat * time, data = fert)
library(car)
Anova(mod_treatment, type = 'II')
```

```
# Anova Table (Type II tests)
#
# Response: log_rich
#             Sum Sq Df  F value     Pr(>F)
# treat      0.09129  1  28.4226 0.00002059 ***
# time       2.21551  2 344.8835  < 2.2e-16 ***
# treat:time 0.02466  2   3.8388    0.03643 *
# Residuals  0.07388 23
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Дисперсионный анализ с III типом сумм квадратов

Опишем процедуру на тот случай, если вдруг вам понадобится воспроизвести в R дисперсионный анализ с III типом сумм квадратов.

При этом способе вначале тестируют взаимодействие, когда все другие факторы есть в модели. Затем тестируют факторы, когда все другие факторы и взаимодействие есть в модели.

Внимание: при использовании III типа сумм квадратов, нужно обязательно указывать тип контрастов для факторов (contrasts=list(фактор_1 = contr.sum, фактор_2=contr.sum)).

```
mod_sum <- lm(log_rich ~ treat * time, data = fert,
              contrasts = list(treat = contr.sum, time = contr.sum))
Anova(mod_sum, type = 3)
```

```
# Anova Table (Type III tests)
#
# Response: log_rich
#              Sum Sq Df    F value     Pr(>F)
# (Intercept) 44.248  1 13776.1092  < 2.2e-16 ***
# treat        0.090  1    28.0402 0.00002249 ***
# time         2.215  2   344.7544  < 2.2e-16 ***
# treat:time   0.025  2     3.8388    0.03643 *
# Residuals    0.074 23
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Почему для расчета III типа сумм квадратов обязательно использовать параметризацию эффектов ?

Для расчета III типа сумм квадратов нужно иметь возможность удалить из модели влияние предиктора, и одновременно оставить в ней взаимодействие (т.е. предикторы и взаимодействие были независимы друг от друга).

В параметризации индикаторных переменных предикторы и взаимодействие коллинеарны, т.е. суммы квадратов III типа будут рассчитаны неправильно.

```
vif(mod_treatment)
```

```
#                GVIF Df GVIF^(1/(2*Df))
# treat      2.896552  1        1.701926
# time       4.344828  2        1.443754
# treat:time 8.517241  2        1.708342
```

В параметризации эффектов переменных предикторы и взаимодействие независимы, значит получатся верные суммы квадратов III типа.

```
vif(mod_sum)
```

```
#                GVIF Df GVIF^(1/(2*Df))
# treat      1.005747  1        1.002869
# time       1.008621  2        1.002148
# treat:time 1.008621  2        1.002148
```

Пост хок тест для взаимодействия факторов

Пост хок тесты в многофакторном дисперсионном анализе

- Поскольку взаимодействие достоверно, факторы отдельно можно не тестировать. Проведем пост хок тест по взаимодействию, чтобы выяснить, какие именно группы различаются
- Если бы взаимодействие было недостоверно, мы бы провели пост хок тест по тем факторам, влияние которых было бы достоверно. Как? См. предыдущую презентацию.

Пост хок тест для взаимодействия факторов

Пост хок тест для взаимодействия факторов делается легче всего “обходным путем”

1. Создаем переменную-взаимодействие
2. Подбираем модель без свободного члена
3. Делаем пост хок тест для этой модели

Задание 1

Дополните этот код, чтобы посчитать пост хок тест Тьюки по взаимодействию факторов

```
# Создаем переменную-взаимодействие
fert$treat_time <- interaction(fert$treat, fert$time)
# Подбираем линейную модель от этой переменной без свободного члена
fit_inter <- lm()
# Делаем пост хок тест для этой модели
library(multcomp)
dat_tukey <- glht(, linfct = mcp( = 'Tukey'))
summary()
```

Решение

```
# Создаем переменную-взаимодействие
fert$treat_time <- interaction(fert$treat, fert$time)
# Подбираем линейную модель без свободного члена
fit_inter <- lm(log_rich ~ treat_time - 1, data = fert)
# Делаем пост хок тест для этой модели
library(multcomp)
dat_tukey <- glht(fit_inter, linfct = mcp(treat_time = 'Tukey'))
summary(dat_tukey)
```

Результаты пост хок теста в виде таблицы почти нечитабельны

```
#
#    Simultaneous Tests for General Linear Hypotheses
#
# Multiple Comparisons of Means: Tukey Contrasts
#
#
# Fit: lm(formula = log_rich ~ treat_time - 1, data = fert)
#
# Linear Hypotheses:
#                              Estimate Std. Error t value Pr(>|t|)
# nutrient.2 - control.2 == 0   0.05040    0.03584   1.406   0.7229
# control.4 - control.2 == 0    0.46332    0.03584  12.926   <0.001 ***
# nutrient.4 - control.2 == 0   0.65186    0.03584  18.186   <0.001 ***
# control.6 - control.2 == 0    0.60309    0.03802  15.863   <0.001 ***
# nutrient.6 - control.2 == 0   0.69968    0.03584  19.520   <0.001 ***
# control.4 - nutrient.2 == 0   0.41292    0.03584  11.520   <0.001 ***
# nutrient.4 - nutrient.2 == 0  0.60146    0.03584  16.780   <0.001 ***
# control.6 - nutrient.2 == 0   0.55269    0.03802  14.537   <0.001 ***
# nutrient.6 - nutrient.2 == 0  0.64928    0.03584  18.114   <0.001 ***
# nutrient.4 - control.4 == 0   0.18854    0.03584   5.260   <0.001 ***
# control.6 - control.4 == 0    0.13977    0.03802   3.676   0.0139 *
# nutrient.6 - control.4 == 0   0.23636    0.03584   6.594   <0.001 ***
# control.6 - nutrient.4 == 0  -0.04877    0.03802  -1.283   0.7909
# nutrient.6 - nutrient.4 == 0  0.04782    0.03584   1.334   0.7634
# nutrient.6 - control.6 == 0   0.09659    0.03802   2.541   0.1533
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# (Adjusted p values reported -- single-step method)
```

Данные для графика при помощи predict()

У нас два дискретных фактора, поэтому вначале используем expand.grid()

```
MyData <- expand.grid(treat = levels(fert$treat),
                      time = levels(fert$time))
MyData <- data.frame(
  MyData,
  predict(mod_treatment, newdata = MyData, interval = 'confidence')
  )
# Обратная трансформация (не забываем про единичку, которую прибавляли)
MyData$richness <- 10^MyData$fit - 1
MyData$LWR <- 10^MyData$lwr - 1
MyData$UPR <- 10^MyData$upr - 1
MyData
```

```
#      treat time       fit       lwr       upr  richness      LWR       UPR
# 1  control    2 0.8281267 0.7756956 0.8805578  5.731731  4.96617  6.595526
# 2 nutrient    2 0.8785253 0.8260942 0.9309564  6.560061  5.70030  7.530145
# 3  control    4 1.2914437 1.2390126 1.3438748 18.563370 16.33854 21.073681
# 4 nutrient    4 1.4799856 1.4275545 1.5324167 29.198519 25.76422 33.073499
# 5  control    6 1.4312147 1.3725950 1.4898345 25.990736 22.58278 29.891178
# 6 nutrient    6 1.5278038 1.4753727 1.5802348 32.713493 28.87945 37.039504
```

Задание 2

Создайте MyData вручную для модели в обычной параметризации:

- предсказанные значения
- стандартные ошибки
- верхнюю и нижнюю границы доверительных интервалов

```
MyData <- expand.grid(treat = levels(fert$treat),
                      time = levels())
X <- model.matrix(~ , data = )
betas <- coef()
MyData$fit <-
MyData$se <-   (X %*% vcov(mod_treatment) %*% t(X))
MyData$lwr <- MyData$ - 1.96 *
MyData$upr <- MyData$ + 1.96 *

# Обратная трансформация
MyData$richness <-
MyData$LWR <-
MyData$UPR <-
MyData
```

```
#      treat time       fit          se       lwr       upr  richness       LWR
# 1  control    2 0.8281267 0.02534547 0.7784496 0.8778039  5.731731  5.004124
# 2 nutrient    2 0.8785253 0.02534547 0.8288482 0.9282024  6.560061  5.742923
# 3  control    4 1.2914437 0.02534547 1.2417665 1.3411208 18.563370 16.448839
# 4 nutrient    4 1.4799856 0.02534547 1.4303085 1.5296628 29.198519 25.934476
# 5  control    6 1.4312147 0.02833709 1.3756740 1.4867554 25.990736 22.750569
# 6 nutrient    6 1.5278038 0.02534547 1.4781266 1.5774809 32.713493 29.069530
```

Решение:

```
MyData <- expand.grid(treat = levels(fert$treat),
                      time = levels(fert$time))
X <- model.matrix(~ treat * time, data = MyData)
betas <- coef(mod_treatment)
MyData$fit <- X %*% betas
MyData$se <- sqrt(diag(X %*% vcov(mod_treatment) %*% t(X)))
MyData$lwr <- MyData$fit - 1.96 * MyData$se
MyData$upr <- MyData$fit + 1.96 * MyData$se
# Обратная трансформация
MyData$richness <- 10^MyData$fit - 1
MyData$LWR <- 10^MyData$lwr - 1
MyData$UPR <- 10^MyData$upr - 1
MyData
```

```
#       treat time       fit         se       lwr       upr  richness       LWR
# 1   control    2 0.8281267 0.02534547 0.7784496 0.8778039  5.731731  5.004124
# 2 nutrient    2 0.8785253 0.02534547 0.8288482 0.9282024  6.560061  5.742923
# 3   control    4 1.2914437 0.02534547 1.2417665 1.3411208 18.563370 16.448839
# 4 nutrient    4 1.4799856 0.02534547 1.4303085 1.5296628 29.198519 25.934476
# 5   control    6 1.4312147 0.02833709 1.3756740 1.4867554 25.990736 22.750569
# 6 nutrient    6 1.5278038 0.02534547 1.4781266 1.5774809 32.713493 29.069530
#         UPR
# 1  6.547513
# 2  7.476224
# 3 20.934148
# 4 32.858114
# 5 29.672942
# 6 36.799049
```

Задание 3

Постройте график результатов, на котором будут изображены предсказанные средние значения видового богатства в зависимости от тритмента и времени экспозиции.

```
pos <- position_dodge(width = 0.2)
gg_linep <- ggplot(data = , aes()) +
  geom_  (position = pos) +
  geom_  (aes(group = ), position = pos) +
  geom_  (position = pos, width = 0.1)
gg_linep
```

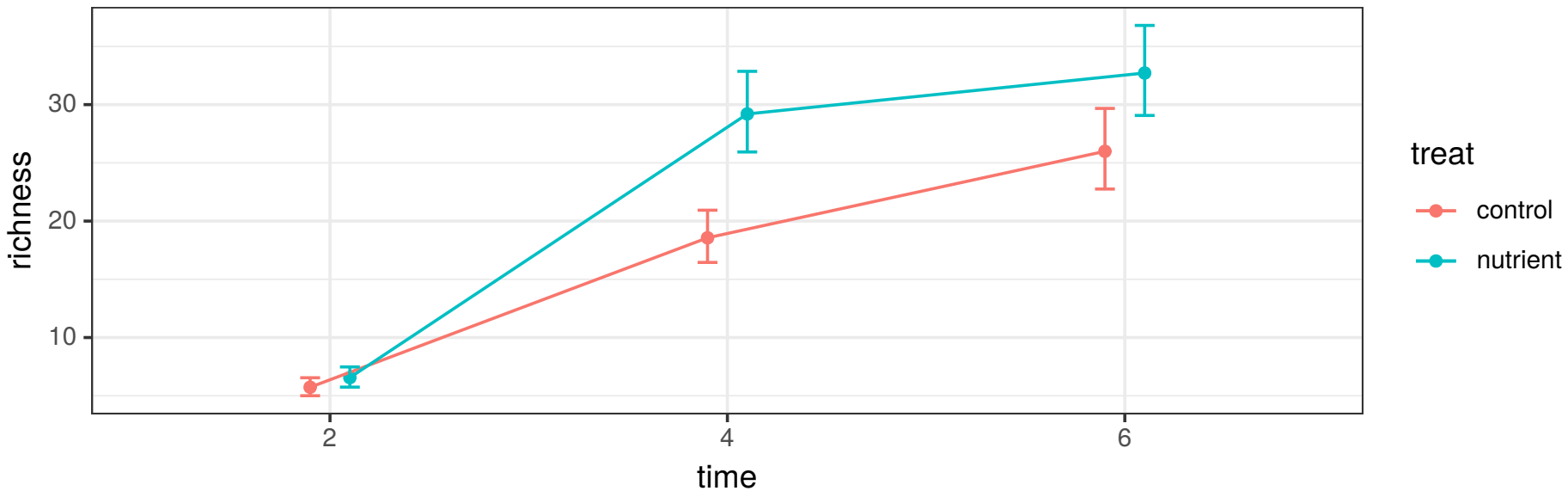
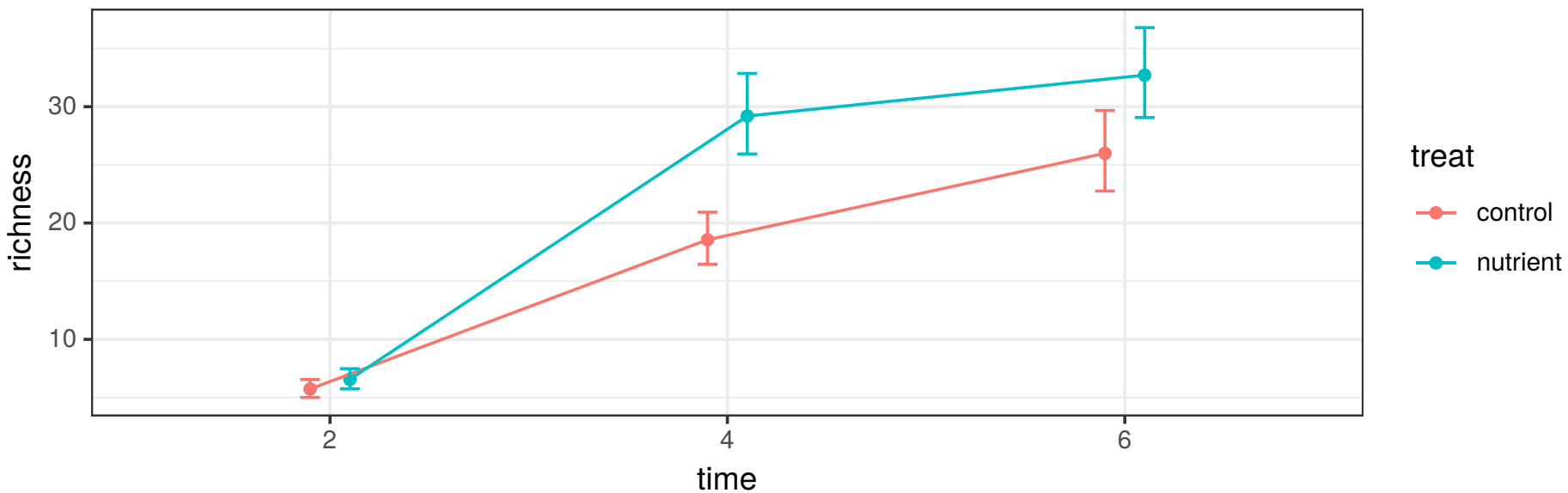


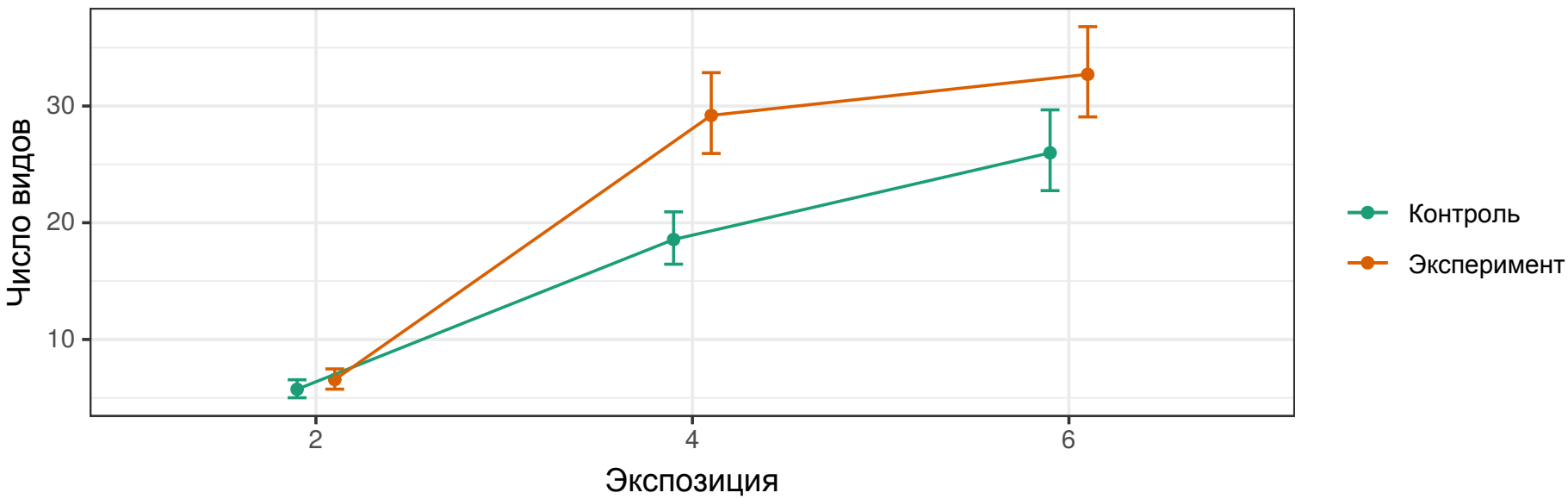
График результатов: Линии с точками

```
pos <- position_dodge(width = 0.2)
gg_linep <- ggplot(data = MyData, aes(x = time, y = richness,
                ymin = LWR,  ymax = UPR, colour = treat)) +
  geom_point(position = pos) +
  geom_line(aes(group = treat), position = pos) +
  geom_errorbar(position = pos, width = 0.1)
gg_linep
```



Приводим график в приличный вид

```
gg_final <- gg_linep + labs(x = 'Экспозиция',  y = 'Число видов') +
  scale_colour_brewer(name = '', palette = 'Dark2',
    labels = c('Контроль', 'Эксперимент'))
gg_final
```



Take home messages

- Многофакторный дисперсионный анализ позволяет оценить взаимодействие факторов. Если оно значимо, то лучше воздержаться от интерпретации их индивидуальных эффектов
- Если численности групп равны, получаются одинаковые результаты вне зависимости от порядка тестирования значимости факторов
- В случае, если численности групп неравны (несбалансированные данные), есть несколько способов тестирования значимости факторов (I, II, III типы сумм квадратов)

Дополнительные ресурсы

- Quinn, Keough, 2002, pp. 221-250
- Logan, 2010, pp. 313-359
- Sokal, Rohlf, 1995, pp. 321-362
- Zar, 2010, pp. 246-266