

Дисперсионный анализ, часть 2

Математические методы в зоологии с использованием R

Марина Варфоломеева

- 1 **Пример: Возраст и способы запомниания**
- 2 **Двухфакторный дисперсионный анализ**
- 3 **Взаимодействие факторов**
- 4 **Несбалансированные данные**
- 5 **Многофакторный дисперсионный анализ в R**
- 6 **Фиксированные и случайные факторы**

Двухфакторный дисперсионный анализ

Вы сможете

- Проводить двухфакторный дисперсионный анализ и интерпретировать его результаты с учетом взаимодействия факторов
- Отличать фиксированные и случайные факторы и выбирать подходящую модель дисперсионного анализа

Пример: Возраст и способы запомниания

Пример: Возраст и способы запомниания

Какие способы запоминания информации лучше работают для молодых и для пожилых? (Eysenck, 1974)

Факторы:

- Age - Возраст:
 - Younger - 50 молодых
 - Older - 50 пожилых (55-65 лет)
- Process - тип активности:
 - Counting - посчитать число букв
 - Rhyming - придумать рифму к слову
 - Adjective - придумать прилагательное
 - Imagery - представить образ
 - Intentional - запомнить слово

Зависимая переменная - Words - сколько вспомнили слов

Открываем данные

```
memory <- read.table(file = "data/eysenck.csv", header = TRUE, sep = "\t")
# Все ли правильно открылось?
str(memory) # Структура данных
```

```
# 'data.frame': 100 obs. of 3 variables:
# $ Age      : Factor w/ 2 levels "Older","Younger": 2 2 2 2 2 2 2 2 2 2 2 ...
# $ Process: Factor w/ 5 levels "Adjective","Counting",...: 2 2 2 2 2 2 2 2 2 2 2 ...
# $ Words   : num  8 6 4 6 7 6 5 7 9 7 ...
```

```
head(memory, 2) # Первые несколько строк файла
```

```
#      Age Process Words
# 1 Younger Counting    8
# 2 Younger Counting    6
```

Знакомимся с данными

```
# Есть ли пропущенные значения
# (особенно, в переменных, которые нас интересуют)?
colSums(is.na(memory))
```

```
#      Age Process  Words
#      0        0      0
```

```
# Каков объем выборки?
nrow(memory) # всего
```

```
# [1] 100
```

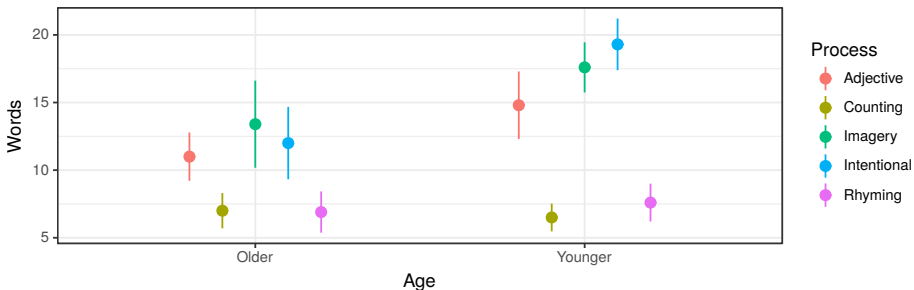
```
table(memory$Age, memory$Process) # в группах
```

```
#
#      Adjective Counting Imagery Intentional Rhyming
# Older           10       10       10           10       10
# Younger          10       10       10           10       10
```

Задание 1

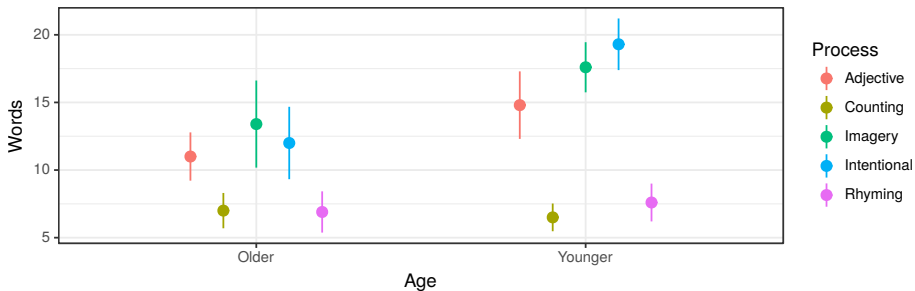
Дополните код, чтобы построить график, на котором приведено среднее число слов (Words) для каждого возраста (Age) и способа запоминания (Process).

```
library()
theme_set()
ggplot(data = , aes()) +
  stat_summary(geom = 'point', fun.data = ,
               position = position_dodge(width = 0.5))
```



Решение 1

```
library(ggplot2)
theme_set(theme_bw())
ggplot(data = memory, aes(x = Age, y = Words, colour = Process)) +
  stat_summary(geom = 'pointrange', fun.data = mean_cl_normal,
              position = position_dodge(width = 0.5))
```



“Некрасивый” порядок уровней на графике

На этом графике некрасивый порядок уровней: средние для разных способов запоминания `memory$Process` расположены, как кажется, хаотично.

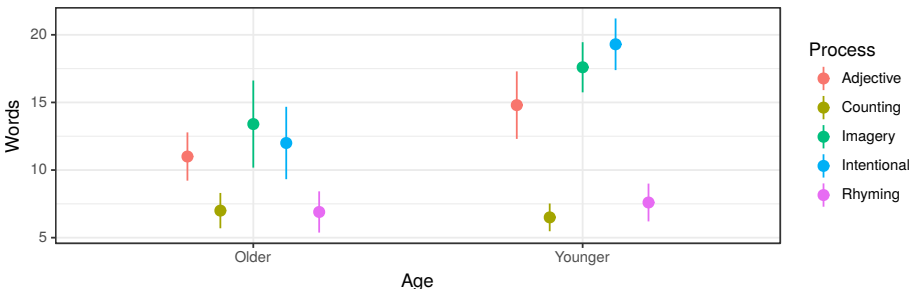
Порядок групп на графике определяется порядком уровней фактора

```
# “старый” порядок уровней
```

```
levels(memory$Process)
```

```
# [1] "Adjective"    "Counting"    "Imagery"     "Intentional"
```

```
# [5] "Rhyming"
```



Изменим порядок уровней

Давайте изменим порядок уровней в факторе `memory$Process` так, чтобы он соответствовал возрастанию средних значений `memory$Words`

```
# "старый" порядок уровней  
levels(memory$Process)
```

```
# [1] "Adjective"    "Counting"     "Imagery"      "Intentional"  
# [5] "Rhyming"
```

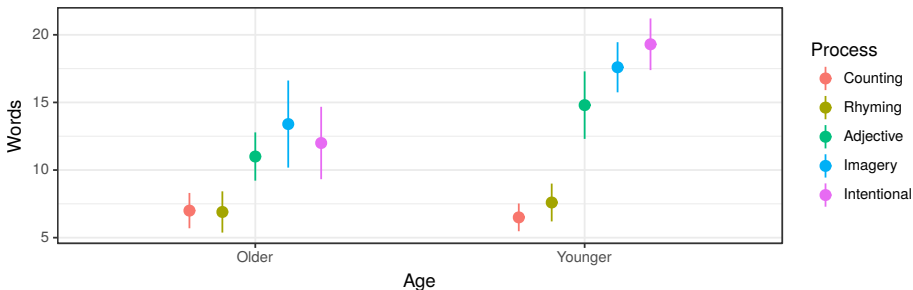
```
# переставляем уровни в порядке следования средних значений memory$Words  
memory$Process <- reorder(x = memory$Process, X = memory$Words, FUN = mean)  
# "новый" порядок уровней стал таким  
levels(memory$Process)
```

```
# [1] "Counting"     "Rhyming"      "Adjective"    "Imagery"  
# [5] "Intentional"
```

График с “правильным” порядком уровней

С новым порядком уровней нам легче визуально сравнивать друг с другом число запомненных слов при разных способах запоминания.

```
ggplot(data = memory, aes(x = Age, y = Words, colour = Process)) +  
  stat_summary(geom = 'pointrange', fun.data = mean_cl_normal,  
    position = position_dodge(width = 0.5))
```



Двухфакторный дисперсионный анализ

Двухфакторный дисперсионный анализ

Значения зависимой переменной складываются из нескольких частей:

- общего среднего μ
- отклонений α_i , связанных с фактором А (в примере две возрастные группы)
- отклонений β_j , связанных с фактором В (в примере это пять способов запоминания)
- отклонений $(\alpha\beta)_{ij}$, связанных с группировкой по обоим факторам (в примере это 10 групп по возрасту и способу запоминания)
- "случайных" отклонений, не связанных с учтенными факторами ϵ_{ijk} .

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

Общая сумма квадратов SS_t складывается из нескольких составляющих:

- изменчивости связанной со всеми факторами $SS_x = SS_a + SS_b + SS_{ab}$
- случайной изменчивости SS_e

$$SS_t = SS_a + SS_b + SS_{ab} + SS_e$$

Таблица традиционного дисперсионного анализа

Источник изменчивости	SS	df	MS	F
Название фактора A	$SS_A = an \sum_i (\bar{A}_i - \bar{y})^2$	$df_A = a - 1$	$MS_A = \frac{SS_A}{df_A}$	$F_{df_A df_e} = \frac{MS_A}{MS_e}$
Название фактора B	$SS_B = bn \sum_j (\bar{B}_j - \bar{y})^2$	$df_B = b - 1$	$MS_B = \frac{SS_B}{df_B}$	$F_{df_B df_e} = \frac{MS_B}{MS_e}$
Взаимодействие факторов A и B	$SS_{AB} = SS_t - SS_A - SS_B - SS_e$	$df_{AB} = (a - 1)(b - 1)$	$MS_{AB} = \frac{SS_{AB}}{df_{AB}}$	$F_{df_{AB} df_e} = \frac{MS_{AB}}{MS_e}$
Случайная	$SS_e = \sum_i \sum_j \sum_k (y_{ijk} - \bar{y}_{ij})^2$	$df_e = (n - 1)ab$	$MS_e = \frac{SS_e}{df_e}$	
Общая	$SS_t = \sum_i \sum_j \sum_k (y_{ijk} - \bar{y})^2$	$df_t = N - 1$		

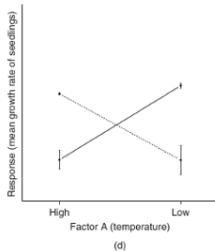
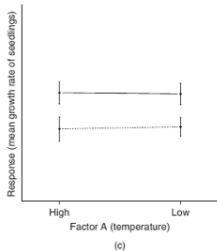
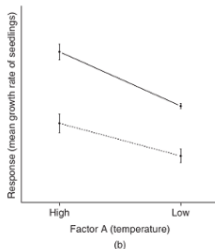
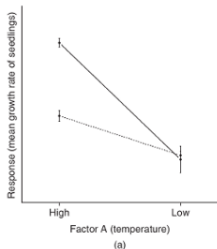
- i, \dots, a — уровни фактора A, j, \dots, b — уровни фактора B, k, \dots, n — индекс наблюдения в группе, N — общее число наблюдений
- \bar{y} — общее среднее значение, \bar{A}_i — средние в группах по фактору A, \bar{B}_j — в группах по фактору B

Взаимодействие факторов

Взаимодействие факторов

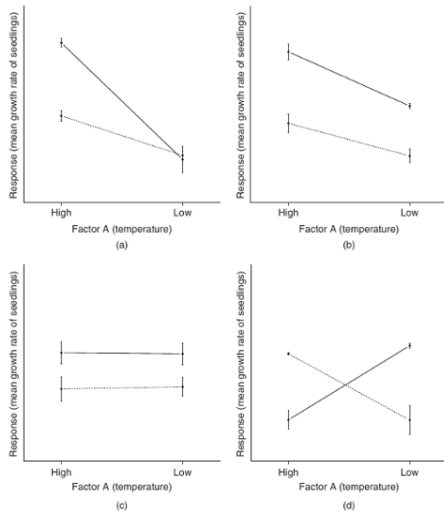
Взаимодействие факторов — когда эффект фактора В разный в зависимости от уровней фактора А и наоборот

На каких рисунках есть взаимодействие факторов?



Взаимодействие факторов

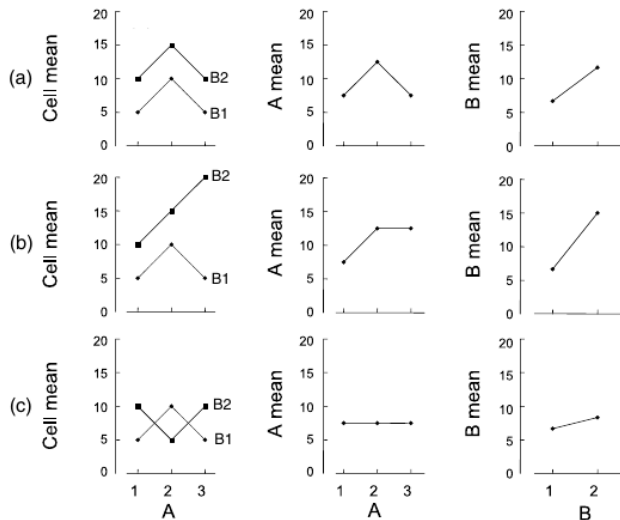
Взаимодействие факторов — когда эффект фактора В разный в зависимости от уровней фактора А и наоборот



На каких рисунках есть взаимодействие факторов?

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

Взаимодействие факторов может маскировать главные эффекты



Если есть значимое взаимодействие

Несбалансированные данные

Сбалансированность данных

A / B	B1	B2	B3
A1	n_{11}	n_{12}	n_{13}
A2	n_{21}	n_{22}	n_{23}

Сбалансированные данные

Одинаковое число наблюдений во всех группах $n_{11} = n_{12} = \dots = n_{ij}$

Несбалансированные данные

Неодинаковое число наблюдений в группах

Суммы квадратов в многофакторном дисперсионном анализе со взаимодействием

Если данные сбалансированы, то ...

взаимодействие и эффекты факторов независимы, их суммы квадратов и соответствующие тесты можно посчитать в одном анализе и его результат не будет зависеть от того, в каком порядке мы рассматриваем факторы.

Если данные несбалансированы, то ...

взаимодействие и эффекты факторов уже не являются полностью независимыми, суммы квадратов для факторов не равны общей сумме квадратов. Если делать все как обычно, результат анализа будет зависеть от порядка включения факторов в модель. (Для вычислений используется регрессионный подход к дисперсионному анализу)

Суммы квадратов III типа

Если данные не сбалансированы и анализ учитывает взаимодействие факторов, то чтобы найти “правильные” суммы квадратов нужно на самом деле выполнить анализ несколько раз:

- последним учитывается фактор A: $Y = B + AB + \mathbf{A}$
- последним учитывается фактор B: $Y = A + AB + \mathbf{B}$
- последним учитывается взаимодействие AB: $A + B + \mathbf{AB}$

Те суммы квадратов, которые рассчитаны **для последнего фактора в каждой модели** будут использоваться в тестах.

Это называется III тип расчета сумм квадратов (= суммы квадратов III типа). Некоторые авторы рекомендуют использовать именно III тип сумм квадратов для несбалансированных данных (Maxwell & Delaney 1990, Milliken, Johnson 1984, Searle 1993, Yandell 1997, Glantz, Slinker 2000).

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 $\sigma_\epsilon^2 > 0$ 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 $\sigma_\epsilon^2 > 0$ 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

Многофакторный дисперсионный анализ в R

Задаем модель со взаимодействием в R

Взаимодействие обозначается : — двоеточием

Если есть факторы A и B, то их взаимодействие A:B

Для такой модели $y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$

Формула модели со взаимодействием:

$Y \sim A + B + A:B$

Сокращенная запись такой же модели обозначает, что модель включает все главные эффекты и их взаимодействия:

$Y \sim A*B$

Внимание: при использовании III типа сумм квадратов, нужно при подборе линейной модели **обязательно указывать тип кодирования для факторов**. В данном случае — `contrasts = list(Age = contr.sum, Process = contr.sum)`

Задание 2

Дополните этот код, чтобы подобрать линейную модель со взаимодействием факторов, в которой используется нужный тип кодирования для факторов:

```
contrasts = list(Age = contr.sum, Process = contr.sum)
```

```
# Линейная модель дисперсионного анализа со взаимодействием факторов  
mem_mod <- lm(formula = , data = ,  
contrasts = list(Age = contr.sum, Process = contr.sum))
```

Решение

```
# Линейная модель дисперсионного анализа со взаимодействием факторов  
mem_mod <- lm(formula = Words ~ Age * Process, data = memory,  
contrasts = list(Age = contr.sum, Process = contr.sum))
```

Задание 3

Проверьте условия применимости дисперсионного анализа

- Есть ли гомогенность дисперсий?
- Не видно ли паттернов в остатках?
- Нормальное ли у остатков распределение?

```
# Данные для анализа остатков
mem_diag <- fortify()
# График расстояния Кука
ggplot(data = , aes(x = 1:nrow( ), y = )) +
  geom_
# График остатков от предсказанных значений
ggplot(data = , aes()) +
  geom_
# Квантильный график остатков
qqPlot
```

Решение:

```
# Данные для анализа остатков
```

```
mem_diag <- fortify(mem_mod)
```

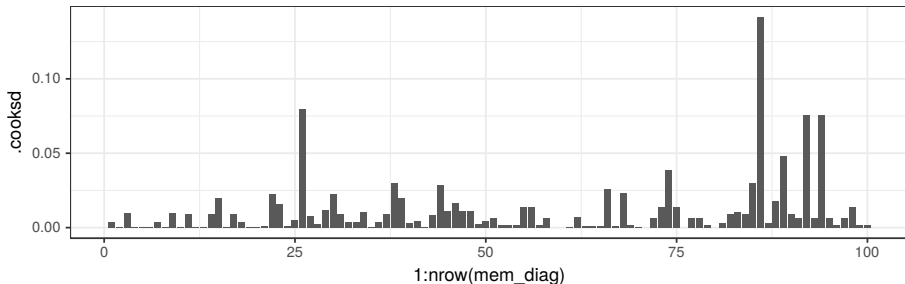
```
head(mem_diag)
```

```
#   Words      Age Process .hat   .sigma   .cooksd .fitted .resid
# 1      8 Younger Counting 0.1 2.843878 0.003461166      6.5    1.5
# 2      6 Younger Counting 0.1 2.848264 0.000384574      6.5   -0.5
# 3      4 Younger Counting 0.1 2.835084 0.009614349      6.5   -2.5
# 4      6 Younger Counting 0.1 2.848264 0.000384574      6.5   -0.5
# 5      7 Younger Counting 0.1 2.848264 0.000384574      6.5    0.5
# 6      6 Younger Counting 0.1 2.848264 0.000384574      6.5   -0.5
#   .stdresid
# 1  0.5581263
# 2 -0.1860421
# 3 -0.9302104
# 4 -0.1860421
# 5  0.1860421
# 6 -0.1860421
```

Решение:

График расстояния Кука

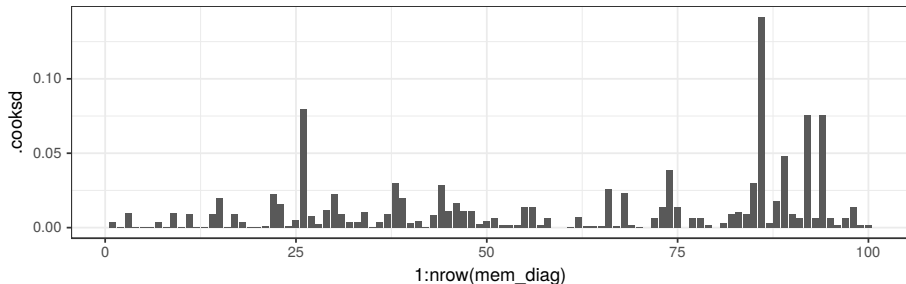
```
ggplot(data = mem_diag, aes(x = 1:nrow(mem_diag), y = .cooks)) +  
  geom_bar(stat = "identity")
```



Решение:

График расстояния Кука

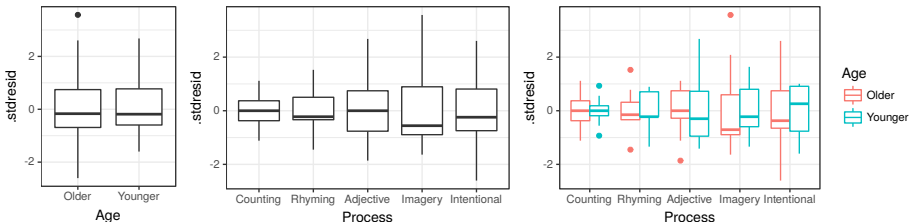
```
ggplot(data = mem_diag, aes(x = 1:nrow(mem_diag), y = .cooks)) +  
  geom_bar(stat = "identity")
```



- Влиятельных наблюдений нет

Решение:

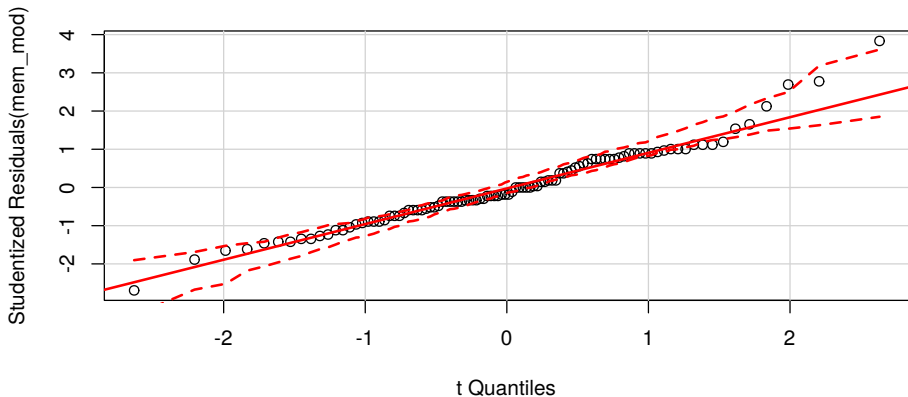
```
ggplot(data = mem_diag, aes(x = Age, y = .stdresid)) +  
  geom_boxplot()  
ggplot(data = mem_diag, aes(x = Process, y = .stdresid)) +  
  geom_boxplot()  
ggplot(data = mem_diag, aes(x = Process, y = .stdresid, colour = Age)) +  
  geom_boxplot()
```



- маленький разброс остатков в группах Counting и Rhyming у обоих возрастов

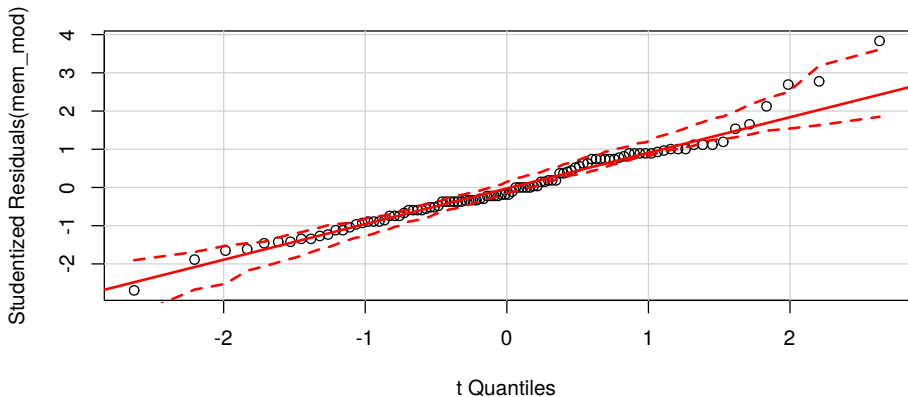
Решение:

```
qqPlot(mem_mod)
```



Решение:

```
qqPlot(mem_mod)
```



- Нет значительных отклонений от нормального распределения

Результаты дисперсионного анализа

```
library(car)
```

```
Anova(mem_mod, type = 3)
```

```
# Anova Table (Type III tests)
```

```
#
```

```
# Response: Words
```

#	Sum Sq	Df	F value	Pr(>F)	
# (Intercept)	13479.2	1	1679.5361	< 2.2e-16	***
# Age	240.3	1	29.9356	0.0000003981	***
# Process	1514.9	4	47.1911	< 2.2e-16	***
# Age:Process	190.3	4	5.9279	0.0002793	***
# Residuals	722.3	90			

```
# ---
```

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

Результаты дисперсионного анализа

```
library(car)
Anova(mem_mod, type = 3)
```

```
# Anova Table (Type III tests)
#
# Response: Words
#
#           Sum Sq Df    F value    Pr(>F)
# (Intercept) 13479.2  1 1679.5361    < 2.2e-16 ***
# Age          240.3  1   29.9356 0.0000003981 ***
# Process      1514.9  4   47.1911    < 2.2e-16 ***
# Age:Process   190.3  4    5.9279 0.0002793 ***
# Residuals    722.3 90
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Взаимодействие достоверно, факторы отдельно можно не тестировать, так как взаимодействие может все равно изменять их эффект до узнаваемости.
- Нужно делать пост хок тест по взаимодействию факторов.

Вычислительный трюк для пост хок теста по взаимодействию факторов

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

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

Задание 4

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

```
# Создаем переменную-взаимодействие
memory$AgeProc <- interaction(memory$Age, memory$)
# Подбираем линейную модель без свободного члена
cell_means <- (Words ~ AgeProc - 1, data = )
# Делаем пост хок тест для этой модели
library(multcomp)
memory_tukey <- glht(model = ,
                     linfct = mcp())
summary(memory_tukey)
```


Решение

```
# Создаем переменную-взаимодействие
memory$AgeProc <- interaction(memory$Age, memory$Process)
# Подбираем линейную модель без свободного члена
cell_means <- lm(Words ~ AgeProc - 1, data = memory)
# Делаем пост хок тест для этой модели
library(multcomp)
memory_tukey <- glht(model = cell_means,
                     linfct = mcp(AgeProc = "Tukey"))
summary(memory_tukey)
```

Смотрим на результаты пост хок теста

В виде таблицы результаты нечитабельны Лучше построить график.

```
#
# Simultaneous Tests for General Linear Hypotheses
#
# Multiple Comparisons of Means: Tukey Contrasts
#
#
# Fit: lm(formula = Words ~ AgeProc - 1, data = memory)
#
# Linear Hypotheses:
#
# Estimate Std. Error t value Pr(>|t|)
# Younger.Counting ~ Older.Counting == 0 -0.500 1.267 -0.395 1.0000
# Older.Rhyming ~ Older.Counting == 0 -0.100 1.267 -0.079 1.0000
# Younger.Rhyming ~ Older.Counting == 0 0.600 1.267 0.474 1.0000
# Older.Adjective ~ Older.Counting == 0 4.000 1.267 3.157 0.0634 .
# Younger.Adjective ~ Older.Counting == 0 7.800 1.267 6.157 <0.01 ***
# Older.Imagery ~ Older.Counting == 0 6.400 1.267 5.052 <0.01 ***
# Younger.Imagery ~ Older.Counting == 0 10.600 1.267 8.367 <0.01 ***
# Older.Intentional ~ Older.Counting == 0 5.000 1.267 3.947 <0.01 **
# Younger.Intentional ~ Older.Counting == 0 12.300 1.267 9.709 <0.01 ***
# Older.Rhyming ~ Younger.Counting == 0 0.400 1.267 0.316 1.0000
# Younger.Rhyming ~ Younger.Counting == 0 1.100 1.267 0.868 0.9970
# Older.Adjective ~ Younger.Counting == 0 4.500 1.267 3.552 0.0207 *
# Younger.Adjective ~ Younger.Counting == 0 8.300 1.267 6.551 <0.01 ***
# Older.Imagery ~ Younger.Counting == 0 6.900 1.267 5.446 <0.01 ***
# Younger.Imagery ~ Younger.Counting == 0 11.100 1.267 8.761 <0.01 ***
# Older.Intentional ~ Younger.Counting == 0 5.500 1.267 4.341 <0.01 **
# Younger.Intentional ~ Younger.Counting == 0 12.800 1.267 10.103 <0.01 ***
# Younger.Rhyming ~ Older.Rhyming == 0 0.700 1.267 0.553 0.9999
# Older.Adjective ~ Older.Rhyming == 0 4.100 1.267 3.236 0.0509 .
# Younger.Adjective ~ Older.Rhyming == 0 7.900 1.267 6.236 <0.01 ***
# Older.Imagery ~ Older.Rhyming == 0 6.500 1.267 5.131 <0.01 ***
# Younger.Imagery ~ Older.Rhyming == 0 10.700 1.267 8.446 <0.01 ***
# Older.Intentional ~ Older.Rhyming == 0 5.100 1.267 4.025 <0.01 **
# Younger.Intentional ~ Older.Rhyming == 0 12.400 1.267 9.787 <0.01 ***
# Older.Adjective ~ Younger.Rhyming == 0 3.400 1.267 2.684 0.1958
# Younger.Adjective ~ Younger.Rhyming == 0 7.200 1.267 5.683 <0.01 ***
# Older.Imagery ~ Younger.Rhyming == 0 5.800 1.267 4.578 <0.01 ***
# Younger.Imagery ~ Younger.Rhyming == 0 10.000 1.267 7.893 <0.01 ***
# Older.Intentional ~ Younger.Rhyming == 0 4.400 1.267 3.473 0.0259 *
# Younger.Intentional ~ Younger.Rhyming == 0 11.700 1.267 9.235 <0.01 ***
```

Данные для графиков

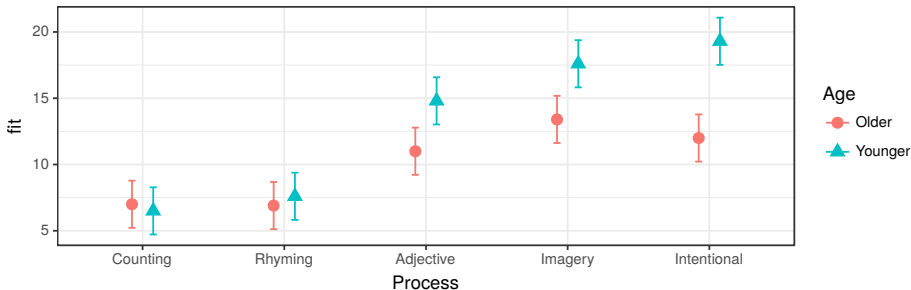
```
process <- levels(memory$Process)
fprocess <- factor(process, levels = process)
MyData <- expand.grid(Age = levels(memory$Age),
                     Process = fprocess)
MyData <- data.frame(MyData,
                     predict(mem_mod, newdata = MyData,
                             interval = "confidence"))
head(MyData)
```

#	Age	Process	fit	lwr	upr
# 1	Older	Counting	7.0	5.220228	8.779772
# 2	Younger	Counting	6.5	4.720228	8.279772
# 3	Older	Rhyming	6.9	5.120228	8.679772
# 4	Younger	Rhyming	7.6	5.820228	9.379772
# 5	Older	Adjective	11.0	9.220228	12.779772
# 6	Younger	Adjective	14.8	13.020228	16.579772

Задание 5

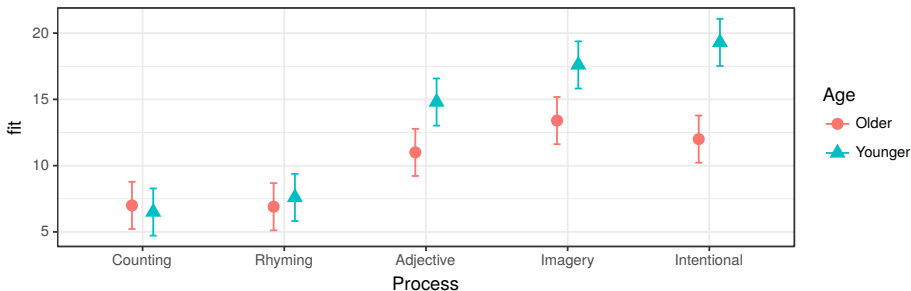
Видоизмените код, чтобы построить такой график результатов

```
pos <- position_dodge(width = 0.3)
gg_pointp <- ggplot() +
  geom_ (aes(), size = 3, position = pos) +
  geom_ (aes(ymin = , ymax = ), width = 0.1, position = pos)
gg_pointp
```



Решение

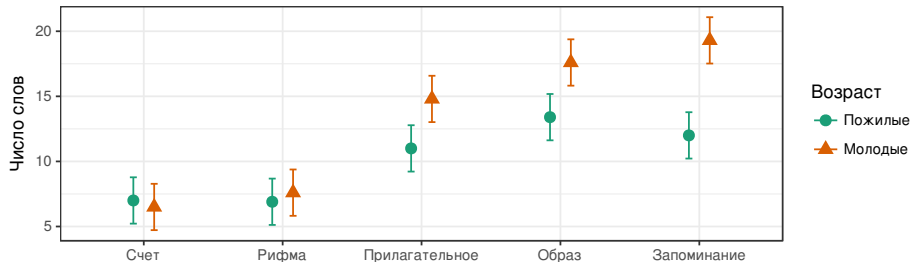
```
pos <- position_dodge(width = 0.3)
gg_pointp <- ggplot(data = MyData, aes(x = Process, y = fit, colour = Age)) +
  geom_point(aes(shape = Age), size = 3, position = pos) +
  geom_errorbar(aes(ymin = lwr, ymax = upr), width = 0.1, position = pos)
gg_pointp
```



Приводим график в приличный вид

```
gg_final <- gg_pointp +
  scale_colour_brewer(name = "Возраст", palette = "Dark2",
    labels = c("Пожилые", "Молодые")) +
  scale_shape_discrete(name = "Возраст",
    labels = c("Пожилые", "Молодые")) +
  scale_x_discrete(name = "Процесс",
    labels = c("Счет", "Рифма", "Прилагательное",
      "Образ", "Запоминание")) +
  labs(y = "Число слов")
```

```
gg_final
```



Фиксированные и случайные факторы

Фиксированные и случайные факторы

Свойства	Фиксированные факторы	Случайные факторы
Уровни фактора	фиксированные, заранее определенные и потенциально воспроизводимые уровни	случайная выборка из всех возможных уровней
Используются для тестирования гипотез	о средних значениях отклика между уровнями фактора $H_0 : \mu_1 = \mu_2 = \dots = \mu_i = \mu$	о дисперсии отклика между уровнями фактора $H_0 : \sigma_{rand.fact.}^2 = 0$
Выводы можно экстраполировать	только на уровни из анализа	на все возможные уровни
Число уровней фактора	Осторожно! Если уровней фактора слишком много, то нужно подбирать слишком много коэффициентов — должно быть много данных	Важно! Для точной оценки σ нужно много уровней фактора — не менее 5

Задание: Примеры фиксированных и случайных факторов

Опишите ситуации, когда эти факторы будут фиксированными, а когда случайными

- Несколько произвольно выбранных градаций плотности моллюсков в полевом эксперименте, где плотностью манипулировали.
- Фактор размер червяка (маленький, средний, большой) в выборке червей.
- Деление губы Чупа на зоны с разной степенью распреснения.

Задание: Примеры фиксированных и случайных факторов

Опишите ситуации, когда эти факторы будут фиксированными, а когда случайными

- Несколько произвольно выбранных градаций плотности моллюсков в полевом эксперименте, где плотностью манипулировали.
- Фактор размер червяка (маленький, средний, большой) в выборке червей.
- Деление губы Чупа на зоны с разной степенью распреснения.
- Приведите другие примеры того, как тип фактора будет зависеть от проверяемых гипотез

Внимание: сегодня говорили только про фиксированные факторы.

Если есть случайные факторы - смешанные модели. О них в магистратуре.

Пакеты nlme и lme4

Книги:

- Pinheiro, J., Bates, D., 2000. Mixed-Effects Models in S and S-PLUS. Springer.
- Zuur, A.F., Ieno, E.N., Walker, N., Saveliev, A.A., Smith, G.M., 2009. Mixed Effects Models and Extensions in Ecology With R. Springer.

Take home messages

- Многофакторный дисперсионный анализ позволяет оценить взаимодействие факторов. Если оно значимо, то лучше воздержаться от интерпретации их индивидуальных эффектов
- В случае, если численности групп неравны (несбалансированные данные), лучше использовать III тип сумм квадратов
- В зависимости от типа факторов (фиксированные или случайные) по разному формулируются гипотезы и рассчитывается F-критерий.

Дополнительные ресурсы

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