

Дисперсионный анализ, часть 1

Математические методы в зоологии с использованием R

Марина Варфоломеева

- 1 Множественные сравнения**
- 2 Дисперсионный анализ**
- 3 Post hoc тесты**
- 4 Графическое представление результатов пост-хок теста**

Знакомимся дисперсионным анализом

Вы сможете

- Объяснить, в чем опасность множественных сравнений, и как с ними можно бороться
- Рассказать, как в дисперсионном анализе моделируются значения зависимой переменной
- Перечислить и проверить условия применимости дисперсионного анализа
- Интерпретировать и описать результаты, записанные в таблице дисперсионного анализа
- Провести множественные попарные сравнения при помощи post hoc теста Тьюки, представить и описать их результаты

Множественные сравнения

Пример: сон у млекопитающих

- TotalSleep - общая продолжительность сна. В нашем анализе это будет зависимая переменная
- Danger - уровень опасности среды для вида, пять градаций (1 - 5)

Скачиваем данные с сайта

Не забудьте войти в вашу директорию для матметодов при помощи `setwd()`

```
library(downloader)
```

```
# в рабочем каталоге создаем суб-директорию для данных
```

```
if(!dir.exists("data")) dir.create("data")
```

```
# скачиваем файл в xlsx, либо в текстовом формате
```

```
if (!file.exists("data/sleep.xlsx")) {
```

```
  download(
```

```
    url = "https://varmara.github.io/mathmethr/data/sleep.xlsx",
```

```
    destfile = "data/sleep.xlsx")
```

```
}
```

```
if (!file.exists("data/sleep.csv")) {
```

```
  download(
```

```
    url = "https://varmara.github.io/mathmethr/data/sleep.xls",
```

```
    destfile = "data/sleep.csv")
```

```
}
```

Читаем данные из файла одним из способов

Чтение из xlsx

```
library(readxl)
sleep <- read_excel(path = "data/sleep.xlsx", sheet = 1)
```

Чтение из csv

```
sleep <- read.table("data/sleep.csv", header = TRUE, sep = "\t")
```

Все ли правильно открылось?

```
str(sleep) # Структура данных
```

```
# 'data.frame': 62 obs. of 11 variables:
# $ Species      : Factor w/ 62 levels "Africanelephant",...: 1 2 3 4 5 6 7 8 9
# $ BodyWt       : num  6654 1 3.38 0.92 2547 ...
# $ BrainWt      : num  5712 6.6 44.5 5.7 4603 ...
# $ NonDreaming  : num  NA 6.3 NA NA 2.1 9.1 15.8 5.2 10.9 8.3 ...
# $ Dreaming     : num  NA 2 NA NA 1.8 0.7 3.9 1 3.6 1.4 ...
# $ TotalSleep  : num  3.3 8.3 12.5 16.5 3.9 9.8 19.7 6.2 14.5 9.7 ...
# $ LifeSpan    : num  38.6 4.5 14 NA 69 27 19 30.4 28 50 ...
# $ Gestation   : num  645 42 60 25 624 180 35 392 63 230 ...
# $ Predation   : int   3 3 1 5 3 4 1 4 1 1 ...
# $ Exposure    : int   5 1 1 2 5 4 1 5 2 1 ...
# $ Danger      : int   3 3 1 3 4 4 1 4 1 1 ...
```

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

```
#           Species BodyWt BrainWt NonDreaming Dreaming
# 1 Africanelephant  6654  5712.0          NA          NA
# 2 Africangiantpouchedrat      1      6.6      6.3      2
#   TotalSleep LifeSpan Gestation Predation Exposure Danger
```


Знакомимся с данными

Есть ли пропущенные значения?

```
sapply(sleep, function(x) sum(is.na(x)))
```

```
#      Species      BodyWt      BrainWt NonDreaming      Dreaming
#           0             0             0           14           12
# TotalSleep    LifeSpan    Gestation    Predation    Exposure
#           4             4             4             0             0
#      Danger
#           0
```

К счастью про уровень опасности (Danger) информация есть для всех объектов.

Но есть пропущенные значения продолжительности сна (TotalSleep).

Каков объем выборки?

В одной из переменных, которые нам интересны, есть пропущенные значения. Это нужно учесть при расчете объема выборки.

Удалим из датафрейма `sleep` строки, где `TotalSleep` принимает значение `NA`.

```
flt <- ! is.na(sleep$TotalSleep)
sl <- sleep[flt, ]
```

Дальше будем работать с датафреймом `sl`. В нем нет пропущенных значений в интересующих нас переменных

```
nrow(sl)
```

```
# [1] 58
```

Каков объем выборки в каждой группе?

```
tapply(X = sl$TotalSleep,
       INDEX = sl$Danger,
       FUN = length)
```

#	очень низкий	низкий	средний	высокий	очень высокий
#	18	14	10	9	7

Задание

Постройте график зависимости общей продолжительности сна (`TotalSleep`) от уровня опасности среды (`Danger`). Используйте `geom_boxplot`.

Раскрасьте график в зависимости от уровня опасности среды (используйте эстетики `fill` или `colour`)

Придумайте, каким образом посчитать, в какой группе животных общая продолжительность сна больше?

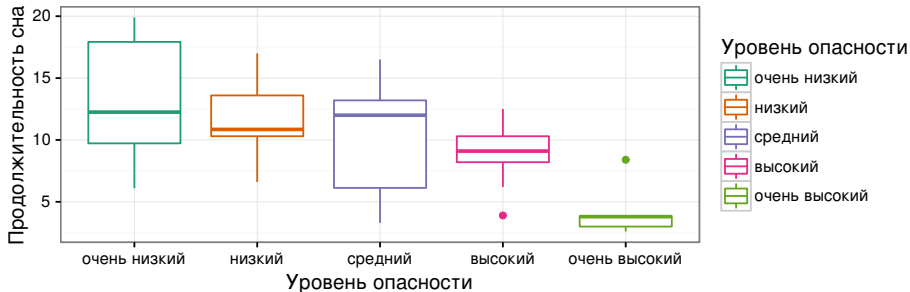
Дополнительное задание:

Попробуйте сменить палитру раскраски, используя `scale_colour_brewer` (варианты можно посмотреть в справке в подразделе примеров или в интернете [Colors \(ggplot2\): раздел RColorBrewer palette chart](#))

Решение

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

gg_sl <- ggplot(data = sl, aes(x = Danger, y = TotalSleep, colour = Danger)) +
  labs(x = "Уровень опасности", y = "Продолжительность сна") +
  scale_colour_brewer("Уровень опасности", palette = "Dark2")
gg_sl + geom_boxplot()
```



Множественные сравнения

Мы могли бы сравнить среднюю продолжительность сна в разных группах при помощи t-критерия.

- 5 групп
- 10 сравнений

Если для каждого сравнения вероятность ошибки первого рода будет $\alpha_{per\ comparison} = 0.05$, то для группы — ?

Множественные сравнения

Мы могли бы сравнить среднюю продолжительность сна в разных группах при помощи t-критерия.

- 5 групп
- 10 сравнений

Если для каждого сравнения вероятность ошибки первого рода будет $\alpha_{per\ comparison} = 0.05$, то для группы — ?

$$\alpha_{family\ wise} = 0.05 * 10$$

В половине случаев мы рискуем найти различия там где их нет!!!

Поправка Бонферрони

Если нужно много сравнений можно снизить $\alpha_{per\ comparison}$

$$\alpha_{per\ comparison} = \frac{\alpha_{family\ wise}}{n}$$

Поправка Бонферрони

Если нужно много сравнений можно снизить $\alpha_{per\ comparison}$

$$\alpha_{per\ comparison} = \frac{\alpha_{family\ wise}}{n}$$

Например, если хотим зафиксировать $\alpha_{family\ wise} = 0.05$

С поправкой Бонферрони $\alpha_{per\ comparison} = 0.05/10 = 0.005$

Очень жесткий критерий!

Дисперсионный анализ

Модель дисперсионного анализа

$$y_{ij} = \mu + a_i + \epsilon_{ij}$$

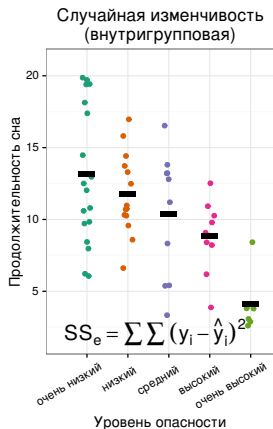
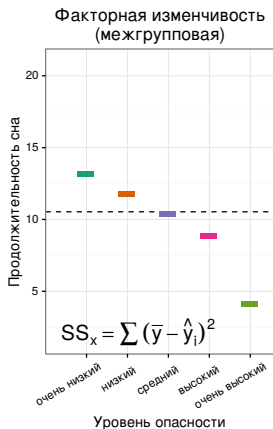
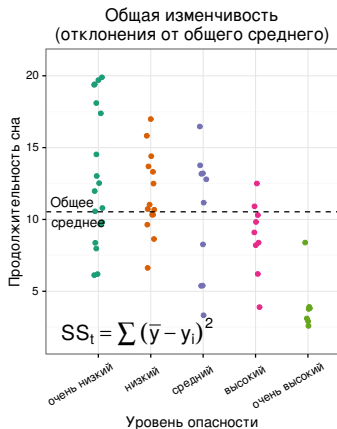
Из чего складываются средние значения в группах по фактору?

Группа	Общее среднее	Эффект	Случайная изменчивость
очень низкий	μ	a_1	ϵ_{1j}
низкий	μ	a_2	ϵ_{2j}
...
очень высокий	μ	a_5	ϵ_{5j}

#	Danger	TotalSleep
# 1	средний	3.3
# 2	средний	8.3
# 3	очень низкий	12.5
# 4	средний	16.5

Структура общей изменчивости

Общая изменчивость (SS_t) = Факторная (SS_x) + Случайная (SS_e)



Если выборки из одной совокупности, то Факторная изменчивость =
Случайная изменчивость

Таблица дисперсионного анализа

Источник изменчивости	Суммы квадратов отклонений, SS	Число степеней свободы, df	Средний квадрат отклонений(дисперсия), MS	F
-----------------------	--------------------------------	----------------------------	---	---

Название фактора

$$SS_x = \sum (y_i - \bar{y})^2 \quad df_x = a - 1 \quad MS_x = \frac{SS_x}{df_x} \quad F_{df_r, df_e} = \frac{MS_x}{MS_e}$$

Случайная

$$SS_e = \sum (y_i - \bar{y})^2 \quad df_e = N - a \quad MS_e = \frac{SS_e}{df_e}$$

Дисперсионный анализ в R

Используем Anova из пакета car, хотя есть и другие функции. Зачем? Когда факторов будет больше одного, эта функция сможет правильно оценить достоверность каждого из них независимо от других.

Anova(результат_функции_lm) - дисперсионный анализ

```
library(car)
mod <- lm(TotalSleep ~ Danger, data=sl)
sl_anova <- Anova(mod)
sl_anova
```

```
# Anova Table (Type II tests)
#
# Response: TotalSleep
#           Sum Sq Df F value    Pr(>F)
# Danger      457.26  4   8.0523 0.0000378 ***
# Residuals  752.41 53
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Общая продолжительность сна различается у видов животных, которые в разной степени подвержены опасностям в течение жизни ($F_{4,53} = 8.05$,

Вопрос:

Назовите условия применимости дисперсионного анализа

- Подсказка: дисперсионный анализ - линейная модель, как и регрессия

Ответ:

Условия применимости дисперсионного анализа:

- Случайность и независимость групп и наблюдений внутри групп
- Нормальное распределение остатков
- Гомогенность дисперсий остатков

Другие ограничения:

- Лучше работает, если размеры групп примерно одинаковы (т.наз. сбалансированный дисперсионный комплекс)
- Устойчив к отклонениям от нормального распределения (при равных объемах групп или при больших выборках)

Задание: Проверьте условия применимости

Проверьте условия применимости дисперсионного анализа используя графики остатков

Решение

1. Данные для проверки условий применимости на графиках остатков

Данные для анализа остатков

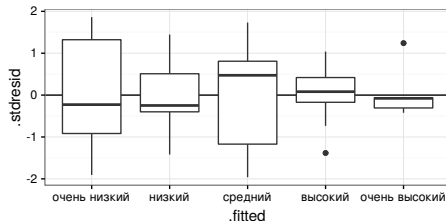
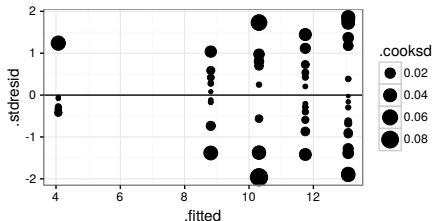
```
sl_diag <- fortify(mod)
head(sl_diag)
```

#	TotalSleep	Danger	.hat	.sigma	.cooksd	.fitted
# 1	3.3	средний	0.10000000	3.663258	0.0854675133	10.310000
# 2	8.3	средний	0.10000000	3.792511	0.0070267928	10.310000
# 3	12.5	очень низкий	0.05555556	3.802964	0.0002985783	13.083333
# 4	16.5	средний	0.10000000	3.694691	0.0666417404	10.310000
# 5	3.9	высокий	0.11111111	3.734657	0.0477828562	8.811111
# 6	9.8	высокий	0.11111111	3.801093	0.0019373477	8.811111

#	.resid	.stdresid
# 1	-7.0100000	-1.9611318
# 2	-2.0100000	-0.5623217
# 3	-0.5833333	-0.1593084
# 4	6.1900000	1.7317270
# 5	-4.9111111	-1.3825029
# 6	0.9888889	0.2783773

2. Выбросы, гомогенность дисперсий

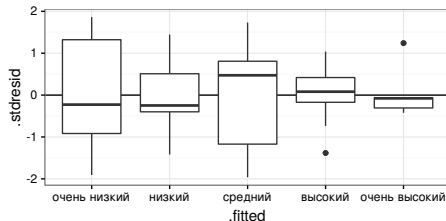
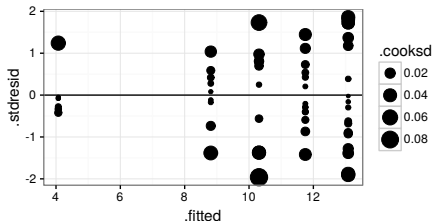
```
gg_res <- ggplot(sl_diag, aes(x = .fitted, y = .stdresid)) +
  geom_hline(yintercept = 0)
gg_1 <- gg_res + geom_point(aes(size = .cooksd))
gg_2 <- gg_res + geom_boxplot(aes(x = Danger))
library(gridExtra)
grid.arrange(gg_1, gg_2, ncol = 2)
```



- Остатки в пределах двух стандартных отклонений, расстояния Кука маленькие - можно продолжать.

2. Выбросы, гомогенность дисперсий

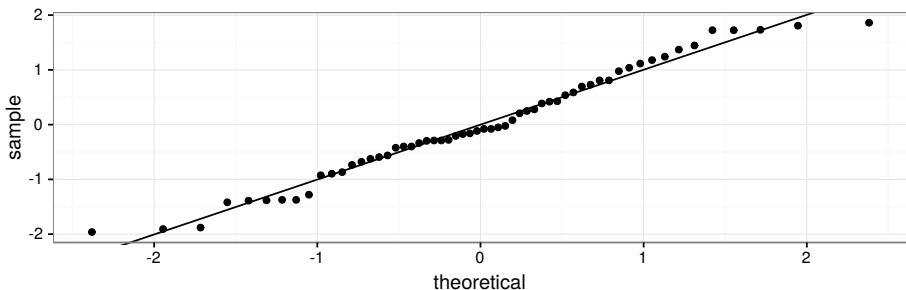
```
gg_res <- ggplot(sl_diag, aes(x = .fitted, y = .stdresid)) +
  geom_hline(yintercept = 0)
gg_1 <- gg_res + geom_point(aes(size = .cooksd))
gg_2 <- gg_res + geom_boxplot(aes(x = Danger))
library(gridExtra)
grid.arrange(gg_1, gg_2, ncol = 2)
```



- Остатки в пределах двух стандартных отклонений, расстояния Кука маленькие - можно продолжать.
- Подозрительно маленькая дисперсия продолжительности сна в группе с очень высоким уровнем опасности.

3. Нормальность распределения

```
ggplot(sl_diag) + geom_point(stat = "qq", aes(sample = .stdresid)) +  
  geom_abline(intercept = 0, slope = sd(sl_diag$.stdresid))
```

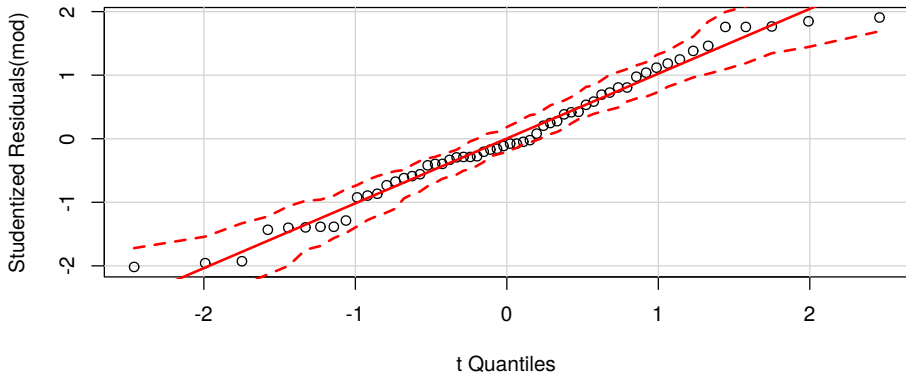


- Распределение практически нормальное

Немного более удобный квантильный график для проверки нормальности распределения

qqPlot() из пакета car

`qqPlot(mod)`



- Нет отклонений от нормального распределения

Post hoc тесты

Post-hoc тесты

Дисперсионный анализ показывает, есть ли влияние фактора (= различаются ли средние значения зависимой переменной между группами)

Пост-хок тесты показывают, какие именно из возможных пар средних значений различаются.

Свойства post-hoc тестов для попарных сравнений средних

- Применяются, если влияние фактора значимо
- Делают поправку для снижения вероятности ошибки I рода α , (но не слишком, чтобы не снизилась мощность, чтобы не возросла β)
- Учитывают величину различий между средними
- Учитывают количество сравниваемых пар
- Различаются по степени консервативности (Тьюки - разумный компромисс)
- Работают лучше при равных объемах групп, при гомогенности дисперсий

Пост-хок тест Тьюки в R

- `glht()` - "general linear hypotheses testing"
- `linfct` - аргумент, задающий гипотезу для тестирования
- `mcp()` - функция, чтобы задавать множественные сравнения (обычные пост-хоки)
- `Danger = "Tukey"` - тест Тьюки по фактору `Danger`

```
library(multcomp)
sl_pht <- glht(mod, linfct = mcp(Danger = "Tukey"))
```

Результаты попарных сравнений (тест Тьюки)

```
summary(sl_pht)
```

```
#
#      Simultaneous Tests for General Linear Hypotheses
#
# Multiple Comparisons of Means: Tukey Contrasts
#
#
# Fit: lm(formula = TotalSleep ~ Danger, data = sl)
#
# Linear Hypotheses:
#
#              Estimate Std. Error t value Pr(>|t|)
# низкий - очень низкий == 0      -1.333      1.343  -0.993   0.8553
# средний - очень низкий == 0     -2.773      1.486  -1.866   0.3441
# высокий - очень низкий == 0     -4.272      1.538  -2.777   0.0550 .
# очень высокий - очень низкий == 0 -9.012      1.678  -5.370 <0.001 ***
# средний - низкий == 0           -1.440      1.560  -0.923   0.8850
# высокий - низкий == 0           -2.939      1.610  -1.826   0.3662
# очень высокий - низкий == 0     -7.679      1.744  -4.402 <0.001 ***
# высокий - средний == 0          -1.499      1.731  -0.866   0.9066
# очень высокий - средний == 0    -6.239      1.857  -3.360   0.0118 *
# очень высокий - высокий == 0    -4.740      1.800  -2.633   0.1052
```

Описываем результаты пост-хок теста

- Продолжительность сна у видов, подвергающихся очень высокому уровню опасности в течение жизни, значительно меньше, чем у тех, кто живет при среднем, низком и очень низком уровне опасности (тест Тьюки, $p < 0.05$).

Графическое представление результатов пост-хок теста

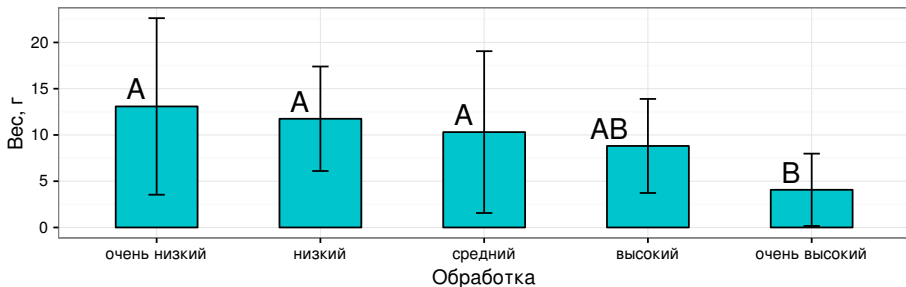
Посчитаем описательную статистику по группам

```
library(dplyr) # есть удобные функции для описания данных
sl_summary <- sl %>% # берем датафрейм sl
  group_by(Danger) %>% # делим на группы по Danger
  # по каждой группе считаем разное
  summarise(
    .n = sum(!is.na(TotalSleep)),
    .mean = mean(TotalSleep, na.rm = TRUE),
    .sd = sd(TotalSleep, na.rm = TRUE),
    .upper_cl = .mean + 1.98*.sd,
    .lower_cl = .mean - 1.98*.sd)
sl_summary
```

```
# # A tibble: 5 × 6
#       Danger      .n      .mean      .sd .upper_cl .lower_cl
#       <fctr> <int>      <dbl>      <dbl>      <dbl>      <dbl>
# 1  очень низкий    18 13.083333  4.819721 22.626380  3.5402867
# 2    низкий      14 11.750000  2.852731 17.398408  6.1015917
# 3   средний     10 10.310000  4.417000 19.055660  1.5643400
# 4   высокий      9  8.811111  2.567316 13.894397  3.7278256
# 5 очень высокий   7  4.071429  1.974601  7.981138  0.1617193
```

Этот график можно использовать для представления результатов

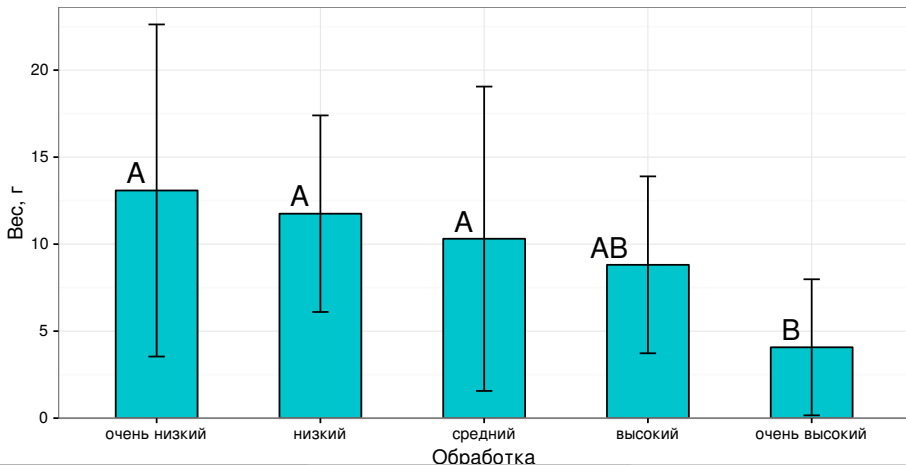
```
gg_means <- ggplot(sl_summary, aes(x = Danger, y = .mean)) +
  geom_bar(stat = "identity", fill = "turquoise3", colour = "black", width = 1) +
  geom_errorbar(aes(ymin = .lower_cl, ymax = .upper_cl), width = 0.1) +
  labs(x = "Обработка", y = "Вес, г") +
  geom_text(aes(label = c("A", "A", "A", "AB", "B"), vjust = -0.3, hjust = 1.1))
gg_means
```



- Достоверно различающиеся по пост-хок тесту группы обозначим разными буквами.

Можно “опустить” прямоугольники на ось x

```
upperlimit <- max(sl_summary$.upper_cl + 1)
gg_means +
  scale_y_continuous(expand = c(0,0),
    limit = c(0, upperlimit))
```



Сохраняем таблицу дисперсионного анализа в файл

```
# 1) в csv
write.csv(sl_anova, file = "medley_res.csv")

# 2) в xls илиxlsx с помощью XLConnect
library(XLConnect)
writeWorksheetToFile(data = sl_anova, file = "medley_res.xls",
                     sheet = "anova_table")

# или

# 3) отправляем в буфер обмена (только Windows) для вставки в Word-Excel
write.table(file = "clipboard", x = sl_anova, sep = "\t")
```

Take home messages

- При множественных попарных сравнениях увеличивается вероятность ошибки первого рода. Поправка Бонферрони - способ точно рассчитать, насколько нужно снизить уровень значимости для каждого из сравнений

Take home messages

- При множественных попарных сравнениях увеличивается вероятность ошибки первого рода. Поправка Бонферрони - способ точно рассчитать, насколько нужно снизить уровень значимости для каждого из сравнений
- При помощи дисперсионного анализа можно проверить гипотезу о равенстве средних значений

Take home messages

- При множественных попарных сравнениях увеличивается вероятность ошибки первого рода. Поправка Бонферрони - способ точно рассчитать, насколько нужно снизить уровень значимости для каждого из сравнений
- При помощи дисперсионного анализа можно проверить гипотезу о равенстве средних значений
- Условия применимости (должны выполняться, чтобы тестировать гипотезы)

Take home messages

- При множественных попарных сравнениях увеличивается вероятность ошибки первого рода. Поправка Бонферрони - способ точно рассчитать, насколько нужно снизить уровень значимости для каждого из сравнений
- При помощи дисперсионного анализа можно проверить гипотезу о равенстве средних значений
- Условия применимости (должны выполняться, чтобы тестировать гипотезы)
- Случайность и независимость групп и наблюдений внутри групп

Take home messages

- При множественных попарных сравнениях увеличивается вероятность ошибки первого рода. Поправка Бонферрони - способ точно рассчитать, насколько нужно снизить уровень значимости для каждого из сравнений
- При помощи дисперсионного анализа можно проверить гипотезу о равенстве средних значений
- Условия применимости (должны выполняться, чтобы тестировать гипотезы)
- Случайность и независимость групп и наблюдений внутри групп
- Нормальное распределение

Take home messages

- При множественных попарных сравнениях увеличивается вероятность ошибки первого рода. Поправка Бонферрони - способ точно рассчитать, насколько нужно снизить уровень значимости для каждого из сравнений
- При помощи дисперсионного анализа можно проверить гипотезу о равенстве средних значений
- Условия применимости (должны выполняться, чтобы тестировать гипотезы)
- Случайность и независимость групп и наблюдений внутри групп
- Нормальное распределение
- Гомогенность дисперсий

Take home messages

- При множественных попарных сравнениях увеличивается вероятность ошибки первого рода. Поправка Бонферрони - способ точно рассчитать, насколько нужно снизить уровень значимости для каждого из сравнений
- При помощи дисперсионного анализа можно проверить гипотезу о равенстве средних значений
- Условия применимости (должны выполняться, чтобы тестировать гипотезы)
- Случайность и независимость групп и наблюдений внутри групп
- Нормальное распределение
- Гомогенность дисперсий
- Post hoc тесты - это попарные сравнения после дисперсионного анализа, которые позволяют сказать, какие именно средние различаются

Дополнительные ресурсы

- Quinn, Keough, 2002, pp. 173-207
- Logan, 2010, pp. 254 - 282
- [Open Intro to Statistics](#), pp.236-246
- Sokal, Rohlf, 1995, pp. 179-260
- Zar, 2010, pp. 189-207