

Дисперсионный анализ, часть 1

Математические методы в зоологии с использованием R

Марина Варфоломеева

1 Множественные сравнения

2 Дисперсионный анализ

3 Post hoc тесты

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

Известно, что у разных млекопитающих продолжительность сна сильно варьирует. Ленивцы спят, коалы спят, а кому-то достаточно. Условия жизни у всех тоже разные. Давайте проверим, есть ли связь между продолжительностью сна и уровнем опасности среды.

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

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 (особенно, в переменных, которые нас интересуют)?

```
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?

```
table(sl$Danger)
```

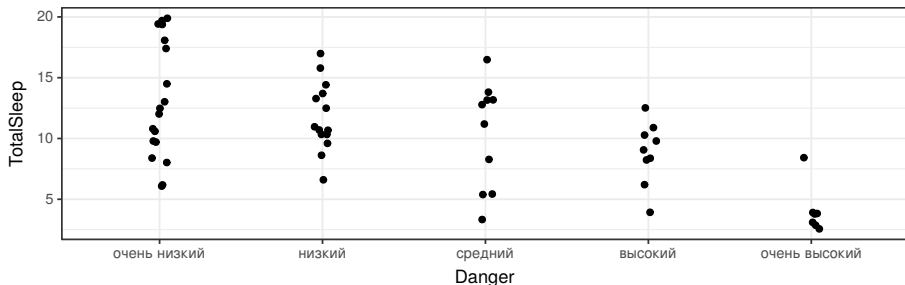
```
#
#  очень низкий      низкий      средний      высокий  очень высокий
#           18           14           10           9           7
```

Задание 1

Дополните код, чтобы построить точечный график зависимости общей продолжительности сна (TotalSleep) от уровня опасности среды (Danger).

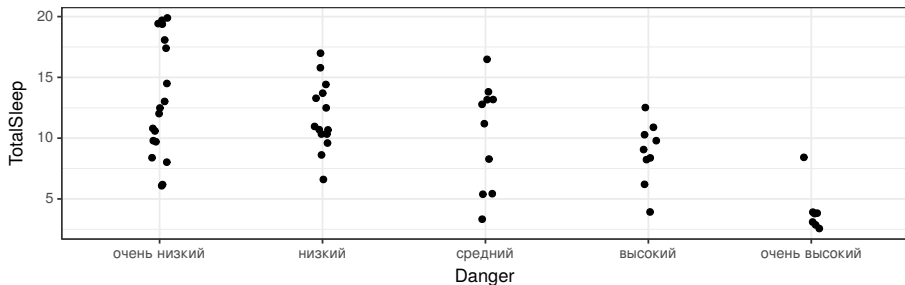
```
library( )
theme_set(theme_bw())

ggplot(data = , aes(x = , y = )) +
  geom_ (position = position_jitter(width = 0.05))
```



Решение 1

```
library(ggplot2)
theme_set(theme_bw())
ggplot(data = sl, aes(x = Danger, y = TotalSleep)) +
  geom_point(position = position_jitter(width = 0.05))
```



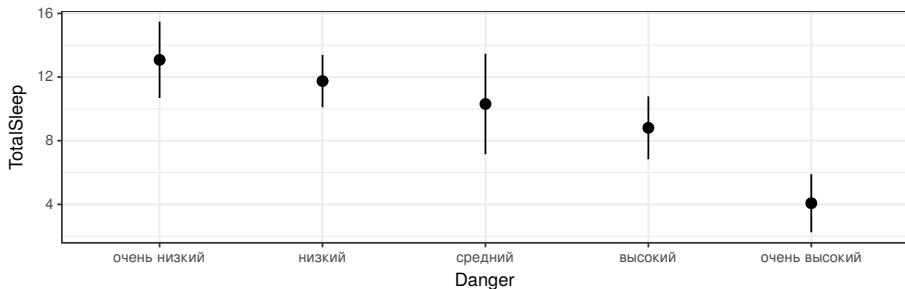
Точечный график — не самый удобный способ представления таких данных.

Задание 2

Точечный график — не самый удобный способ представления таких данных. Лучше было бы изобразить средние значения и их 95% доверительные интервалы.

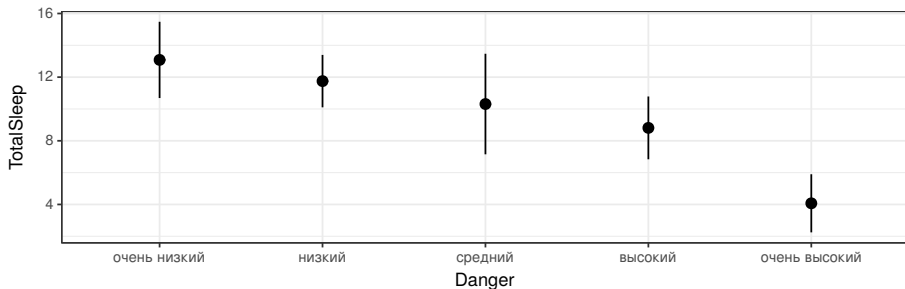
Замените в графике из прошлого задания геом на этот стат:

```
stat_summary(geom = 'pointrange', fun.data = mean_cl_normal)
```



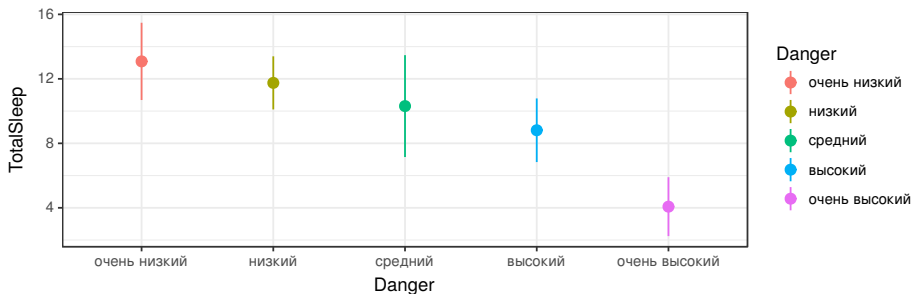
Решение 2

```
ggplot(data = sl, aes(x = Danger, y = TotalSleep)) +  
  stat_summary(geom = 'pointrange', fun.data = mean_cl_normal)
```



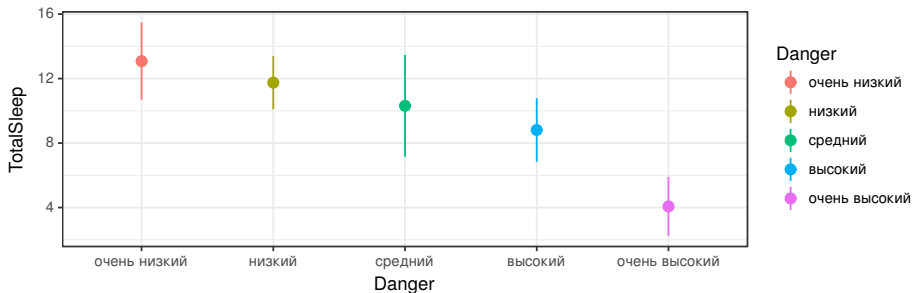
Задание 3

Раскрасьте график в зависимости от уровня опасности среды (используйте эстетику colour)



Решение 3

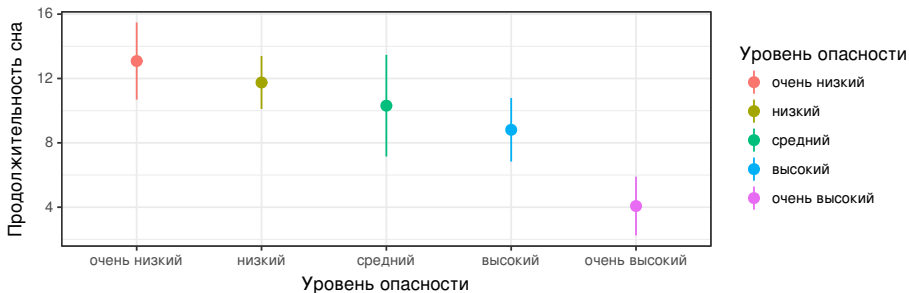
```
ggplot(data = sl, aes(x = Danger, y = TotalSleep, colour = Danger)) +
  stat_summary(geom = 'pointrange', fun.data = mean_cl_normal)
```



Задание 4

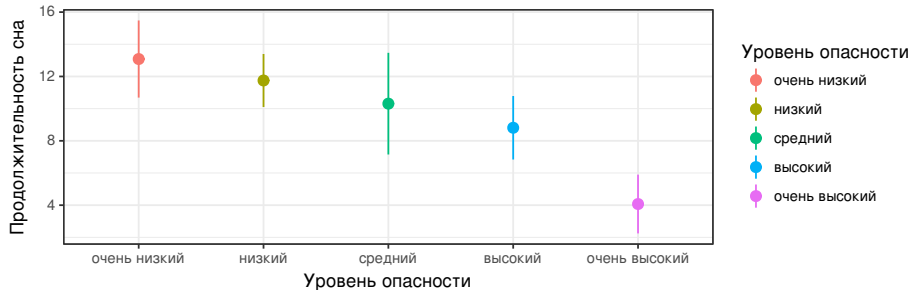
Создайте подписи к осям и легенде, при помощи слоя подписей `labs()`.
Отредактируйте этот код и добавьте его к графику с предыдущего шага

```
labs(x = , y = "Продолжительность сна", colour = )
```



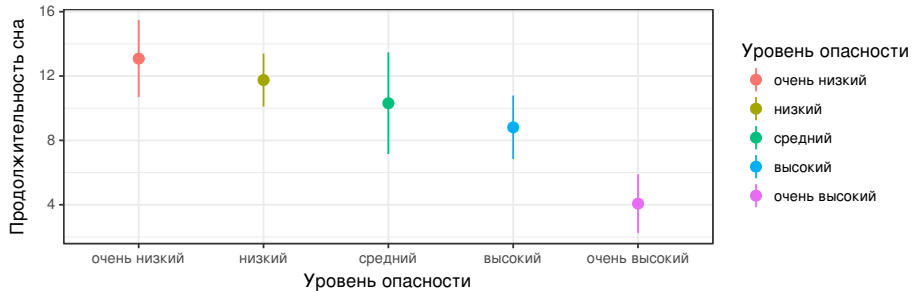
Решение 4

```
ggplot(data = sl, aes(x = Danger, y = TotalSleep, colour = Danger)) +
  stat_summary(geom = 'pointrange', fun.data = mean_cl_normal) +
  labs(x = "Уровень опасности",
       y = "Продолжительность сна",
       colour = "Уровень опасности")
```



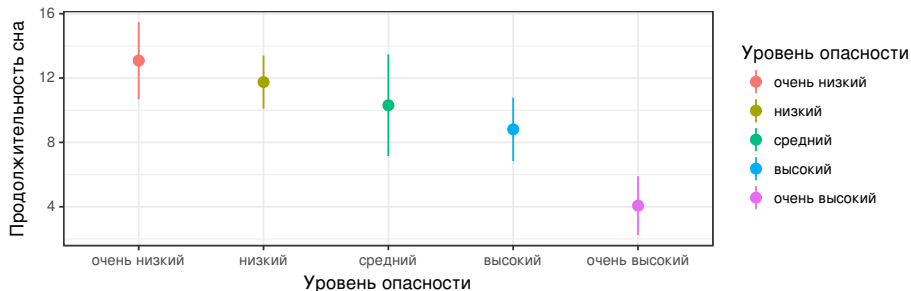
Множественные сравнения

Мы могли бы сравнить среднюю продолжительность сна в разных группах при помощи t-критерия. У нас всего 5 групп. Сколько возможно между ними попарных сравнений?



Множественные сравнения

Мы могли бы сравнить среднюю продолжительность сна в разных группах при помощи t-критерия. У нас всего 5 групп. Сколько возможно между ними попарных сравнений?

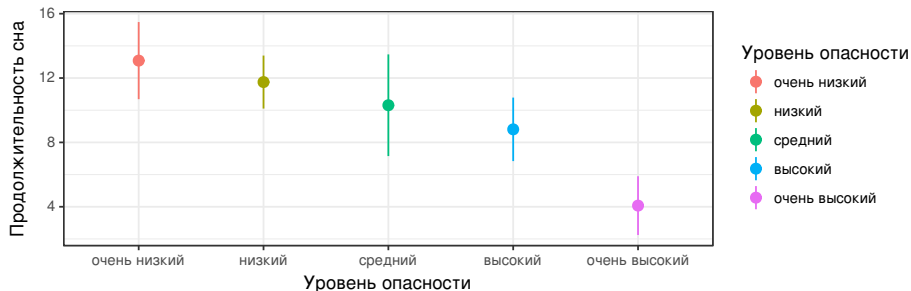


Всего возможно 10 сравнений

Если для каждого сравнения вероятность ошибки первого рода будет $\alpha_{per\ comparison} = 0.05$, то для всей группы из 10 сравнений — ?

Множественные сравнения

Мы могли бы сравнить среднюю продолжительность сна в разных группах при помощи t-критерия. У нас всего 5 групп. Сколько возможно между ними попарных сравнений?



Всего возможно 10 сравнений

Если для каждого сравнения вероятность ошибки первого рода будет $\alpha_{per\ comparison} = 0.05$, то для всей группы из 10 сравнений — ?

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

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

Поправка Бонферрони

Если все-таки приходится делать много сравнений, нужно снизить $\alpha_{per\ comparison}$ до обычного уровня

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

Поправка Бонферрони

Если все-таки приходится делать много сравнений, нужно снизить $\alpha_{per\ comparison}$ до обычного уровня

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

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

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

Это очень жесткая поправка! Мы рискуем не найти достоверных различий, даже там, где они есть...

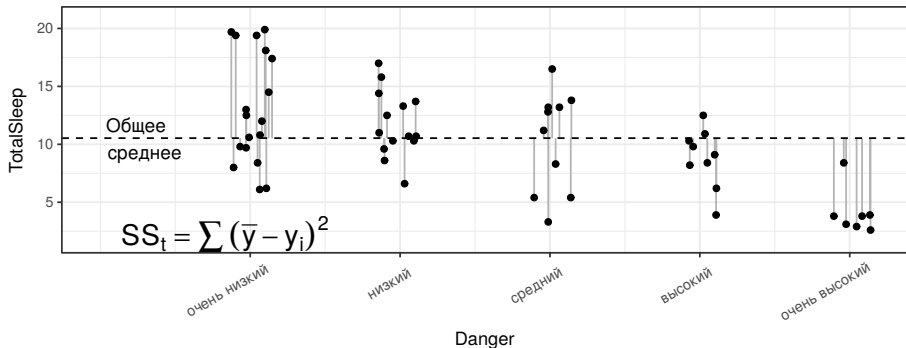
Но есть выход. Вместо множества попарных сравнений можно использовать один тест — дисперсионный анализ (analysis of variation, ANOVA).

Дисперсионный анализ

Общая изменчивость

Общая изменчивость SS_t — это сумма квадратов отклонений наблюдаемых значений y_i от общего среднего \bar{y}

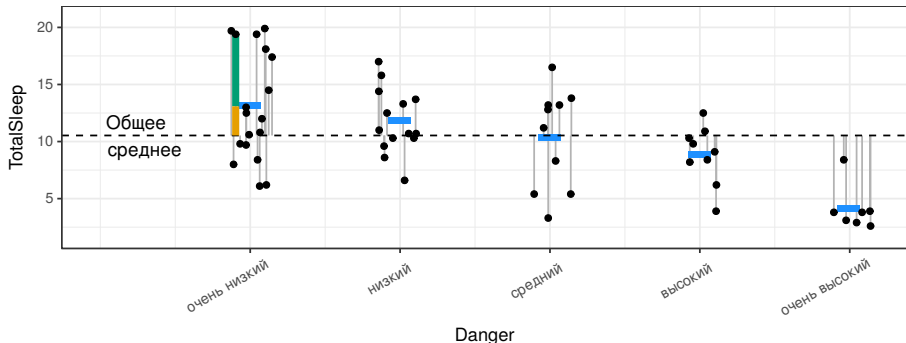
Общая изменчивость
(отклонения от общего среднего)



Отклонения от общего среднего

Отклонения от общего среднего складываются из двух составляющих:

- Внутригрупповые отклонения — отклонения наблюдаемых значений от внутригрупповых средних
- Межгрупповые отклонения — отклонения внутригрупповых средних от общего среднего ("эффекты" групп)

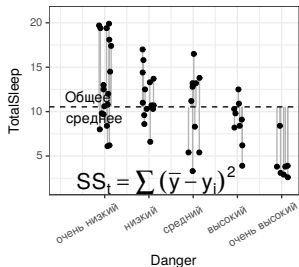


Структура общей изменчивости

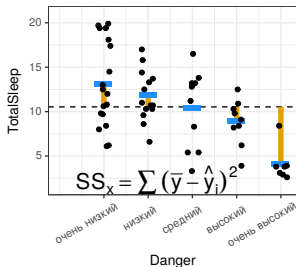
Общая изменчивость SS_t складывается из изменчивости связанной с фактором SS_x и случайной изменчивости SS_e

$$SS_t = SS_x + SS_e$$

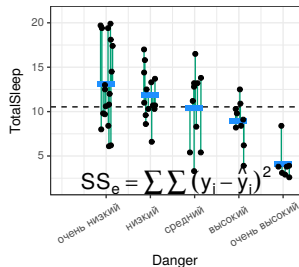
Общая изменчивость
(отклонения от общего среднего)



Факторная изменчивость
(межгрупповая)



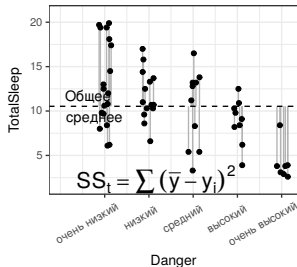
Случайная изменчивость
(внутригрупповая)



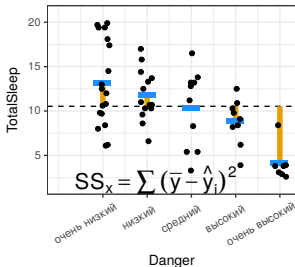
Средние квадраты отклонений

Если поделить суммы квадратов отклонений (SS) на их число степеней свободы, то получатся средние квадраты отклонений (MS) — дисперсии

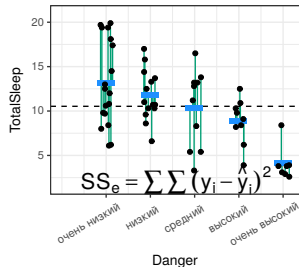
Общая изменчивость
(отклонения от общего среднего)



Факторная изменчивость
(межгрупповая)



Случайная изменчивость
(внутригрупповая)



MS_t полная дисперсия

MS_x факторная дисперсия

MS_e остаточная дисперсия

$$MS_t = \frac{SS_t}{df_t}$$

$$SS_t = \sum (\bar{y} - y_i)^2$$

$$df_t = n - 1$$

$$MS_x = \frac{SS_x}{df_x}$$

$$SS_x = \sum (\hat{y} - \bar{y})^2$$

$$df_x = 1$$

$$MS_e = \frac{SS_e}{df_e}$$

$$SS_e = \sum (\hat{y} - y_i)^2$$

$$df_e = n - 2$$

Если выборки из одной совокупности, то

Если выборки из одной совокупности, то наблюдения из разных групп будут отличаться друг от друга не больше, чем наблюдения из одной группы, т.е. факторная дисперсия будет близка к случайной дисперсии $MS_x \sim MS_e$. Их равенство можно проверить при помощи F-критерия

$$F_{df_x, df_e} = \frac{MS_x}{MS_e}$$

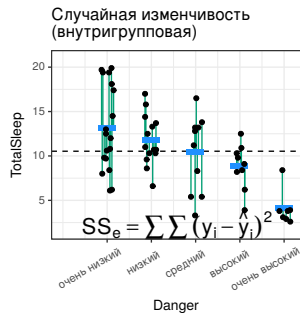
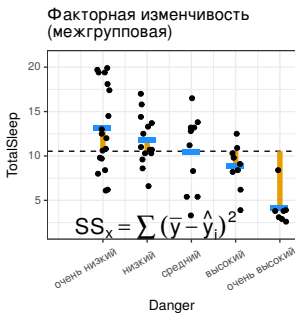
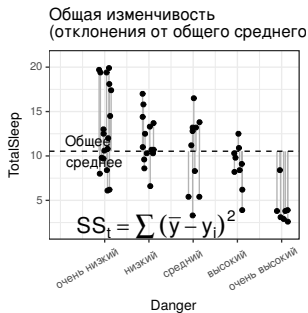


Таблица дисперсионного анализа

Источник изменчивости	SS	df	MS	F
Название фактора	$SS_x = \sum (\bar{y} - \hat{y}_i)^2$	$df_x = a - 1$	$MS_x = \frac{SS_x}{df_x}$	$F_{df_x, df_e} = \frac{MS_x}{MS_e}$
Случайная	$SS_e = \sum (y_i - \hat{y}_i)^2$	$df_e = N - a$	$MS_e = \frac{SS_e}{df_e}$	
Общая	$SS_t = \sum (\bar{y} - y_i)^2$	$df_t = N - 1$		

Гипотезы: $H_0 : MS_x = MS_e$, $H_A : MS_x \neq MS_e$

Минимальное упоминание результатов в тексте должно содержать F_{df_x, df_e} и p .

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 значимость каждого из них независимо от других.

Anova(результат_функции_lm, type = 3) - дисперсионный анализ

```
library(car)
sl_mod <- lm(TotalSleep ~ Danger, data = sl)
sl_anova <- Anova(sl_mod, type = 3)
sl_anova
```

```
# Anova Table (Type III tests)
#
# Response: TotalSleep
#               Sum Sq Df  F value    Pr(>F)
# (Intercept) 3081.12   1 217.0348 < 2.2e-16 ***
# 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 ($F_{4,53} = 8.05$, $p < 0.01$).

Описание результатов дисперсионного анализа

Результаты дисперсионного анализа можно представить в виде таблицы

- Общая продолжительность сна различается у видов животных, которые в разной степени подвержены опасностям в течение жизни (Табл. 1).

Table 1: Результаты дисперсионного анализа продолжительности сна млекопитающих в зависимости от уровня опасностей, которым они подвергаются в течении жизни. SS — суммы квадратов отклонений, df — число степеней свободы, F — значение F-критерия, P — доверительная вероятность.

	SS	df	F	P
Уровень опасности	457.3	4	8.1	< 0.01
Остаточная	752.4	53		

Вопрос:

Назовите условия применимости дисперсионного анализа

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

Вопрос:

Назовите условия применимости дисперсионного анализа

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

Вопрос:

Назовите условия применимости дисперсионного анализа

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

Условия применимости дисперсионного анализа:

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

Вопрос:

Назовите условия применимости дисперсионного анализа

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

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 для модели `sl_mod`, Дополните код, чтобы построить графики остатков

```
# Данные для анализа остатков
sl_diag <- fortify()
head(sl_diag)

# График расстояния Кука
ggplot(data = , aes(x = 1:nrow(  ), y = )) +
  geom_bar(stat = "identity")

# График остатков от предсказанных значений
ggplot(data = , aes(x = , y = .stdresid)) +
  geom_

# Квантильный график остатков
qqPlot
```

Решение:

Данные для анализа остатков

sl_diag <- fortify(sl_mod)

head(sl_diag)

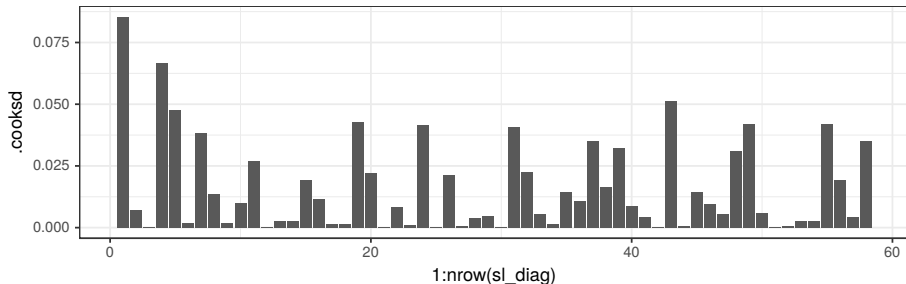
#	TotalSleep	Danger	.hat	.sigma	.cooksd	.fitted
# 1	3.3	средний	0.10000000	3.663258	0.0854675133	10.310000
# 2	8.3	средний	0.10000000	3.792511	0.0070267928	10.310000
# 3	12.5	очень низкий	0.05555556	3.802964	0.0002985783	13.083333
# 4	16.5	средний	0.10000000	3.694691	0.0666417404	10.310000
# 5	3.9	высокий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

Решение:

График расстояния Кука

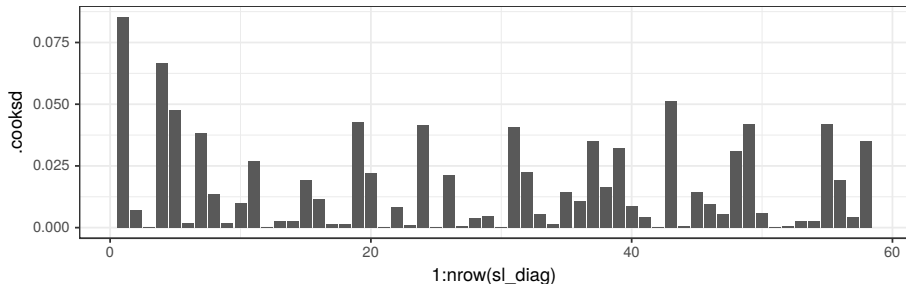
```
ggplot(data = sl_diag, aes(x = 1:nrow(sl_diag), y = .cooksd)) +  
  geom_bar(stat = "identity")
```



Решение:

График расстояния Кука

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

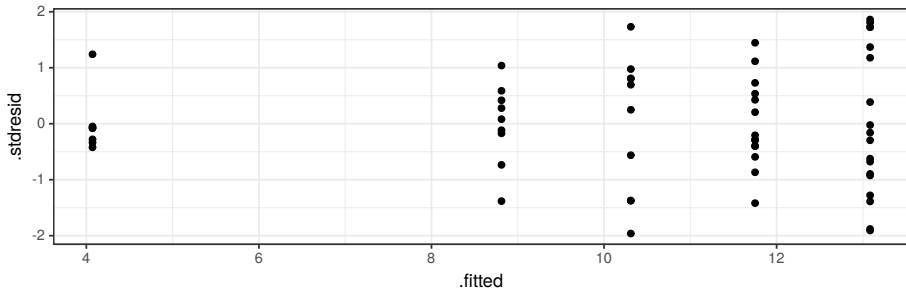


- Влиятельных наблюдений нет

Решение:

График остатков от предсказанных значений

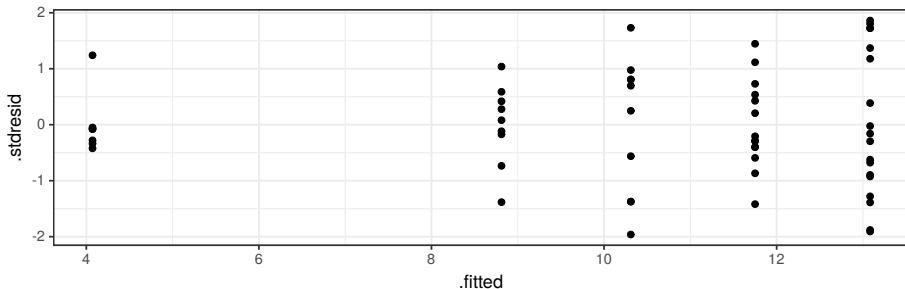
```
ggplot(data = sl_diag, aes(x = .fitted, y = .stdresid)) +  
  geom_point()
```



Решение:

График остатков от предсказанных значений

```
ggplot(data = sl_diag, aes(x = .fitted, y = .stdresid)) +  
  geom_point()
```



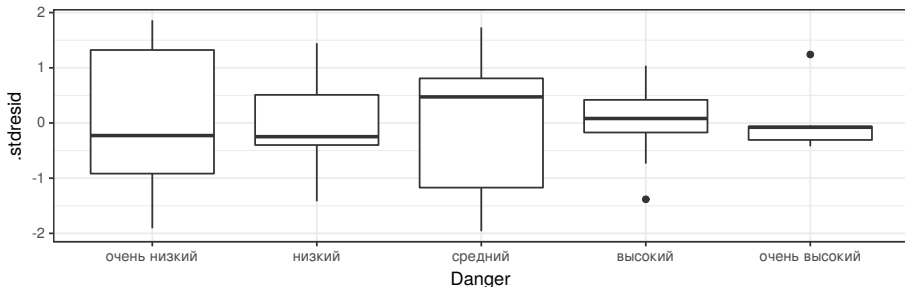
- В одной из групп маленький разброс

Решение:

Если предиктор — дискретная переменная, то вместо обычного точечного графика остатков лучше построить боксплот остатков

График остатков от значений дискретного предиктора

```
ggplot(data = sl_diag, aes(x = Danger, y = .stdresid)) + geom_boxplot()
```

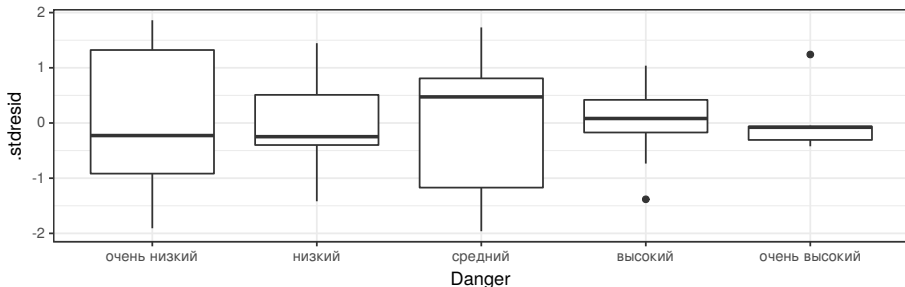


Решение:

Если предиктор — дискретная переменная, то вместо обычного точечного графика остатков лучше построить боксплот остатков

График остатков от значений дискретного предиктора

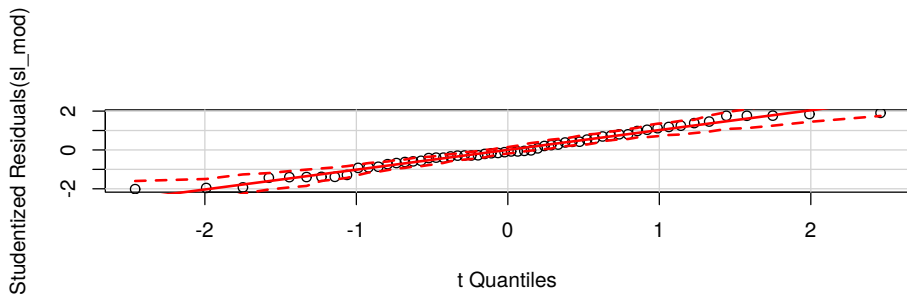
```
ggplot(data = sl_diag, aes(x = Danger, y = .stdresid)) + geom_boxplot()
```



- Остатки в пределах двух стандартных отклонений.
- Подозрительно маленькая дисперсия продолжительности сна в группе с очень высоким уровнем опасности.

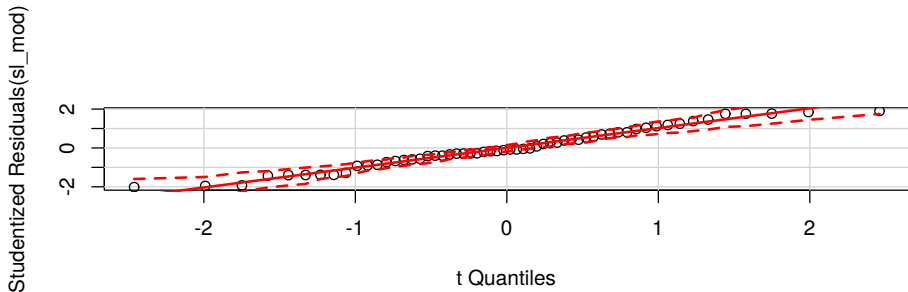
Решение:

```
qqPlot(sl_mod)
```



Решение:

```
qqPlot(sl_mod)
```



- Остатки распределены нормально

Решение: Паттерны в остатках (графики остатков от переменных в модели и вне ее).

На самом деле, нужно еще построить графики остатков от переменных в модели и вне ее — чтобы выяснить, не забыли ли мы включить другие важные предикторы.

Построй те самостоятельно графики, используя код. Какие из переменных хорошо было бы добавить в модель?

```
sl_diag_full <- data.frame(sl_diag, sl)
gg_other <- ggplot(sl_diag_full, aes(y = .stdresid)) + geom_hline(yintercept = 0)
gg_other + geom_point(aes(x = log(BodyWt)))
gg_other + geom_point(aes(x = log(BrainWt)))
gg_other + geom_point(aes(x = NonDreaming))
gg_other + geom_point(aes(x = Dreaming))
gg_other + geom_point(aes(x = log(LifeSpan)))
gg_other + geom_point(aes(x = Gestation))
gg_other + geom_point(aes(x = Predation))
gg_other + geom_point(aes(x = Exposure))
```

Решение: Паттерны в остатках (графики остатков от переменных в модели и вне ее).

На самом деле, нужно еще построить графики остатков от переменных в модели и вне ее — чтобы выяснить, не забыли ли мы включить другие важные предикторы.

Построй те самостоятельно графики, используя код. Какие из переменных хорошо было бы добавить в модель?

```
sl_diag_full <- data.frame(sl_diag, sl)
gg_other <- ggplot(sl_diag_full, aes(y = .stdresid)) + geom_hline(yintercept = 0)
gg_other + geom_point(aes(x = log(BodyWt)))
gg_other + geom_point(aes(x = log(BrainWt)))
gg_other + geom_point(aes(x = NonDreaming))
gg_other + geom_point(aes(x = Dreaming))
gg_other + geom_point(aes(x = log(LifeSpan)))
gg_other + geom_point(aes(x = Gestation))
gg_other + geom_point(aes(x = Predation))
gg_other + geom_point(aes(x = Exposure))
```

- На всех графиках, кроме Predation и Exposure, величина остатков зависит от переменных, не включенных в модель. Правильно было бы их добавить. Но сейчас, в учебных целях, мы продолжим работать с простым однофакторным дисперсионным анализом.

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(sl_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  0.0119 *
# очень высокий - высокий == 0   -4.740      1.899  -2.496  0.1051
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# (Adjusted p-values reported -- single step method)
```

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, $p < 0.05$).

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, $p < 0.05$).

Но лучше еще и нарисовать график.

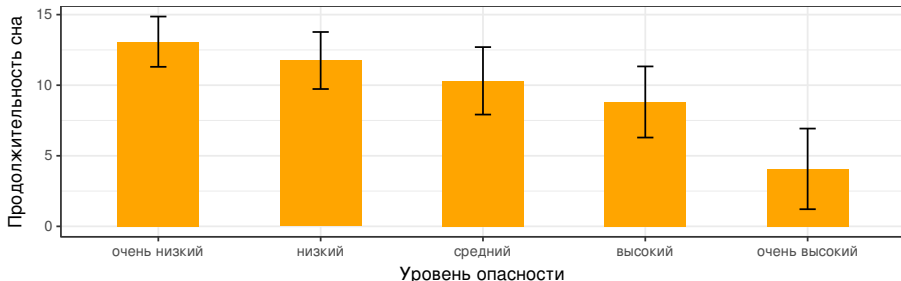
Данные для графика при помощи predict()

```
MyData <- data.frame(Danger = levels(sl$Danger))
MyData$Danger <- factor(MyData$Danger, levels = levels(sl$Danger),
                        labels = levels(sl$Danger))
Predictions <- predict(sl_mod, newdata = MyData,
                      interval = "confidence")
MyData <- data.frame(MyData, Predictions)
MyData
```

#	Danger	fit	lwr	upr
# 1	очень низкий	13.083333	11.302064	14.864603
# 2	низкий	11.750000	9.730230	13.769770
# 3	средний	10.310000	7.920176	12.699824
# 4	высокий	8.811111	6.292015	11.330207
# 5	очень высокий	4.071429	1.215043	6.927815

Задание

Используя данные из датафрейма `MyData`, дополните код и постройте график? где разными столбиками с заливкой оранжевого цвета будут показаны предсказанные средние значения, а усами будут показаны их доверительные интервалы.

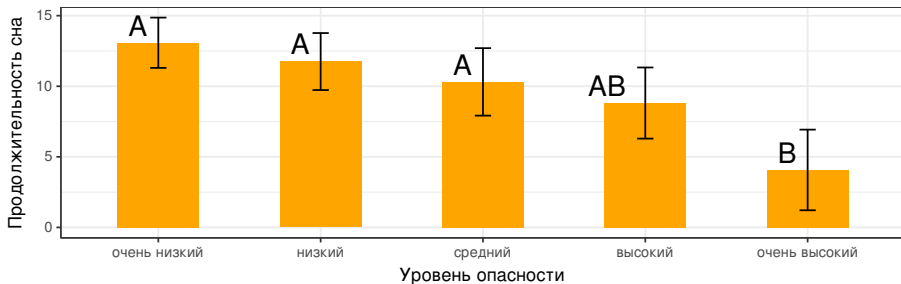


```
gg_means <- ggplot(data = , aes(x = , y = )) +
  geom_bar(stat = "", width = 0.5) +
  geom_errorbar(aes(ymin = , ymax = ), width = 0.1) +
  labs()
gg_means
```


Этот график можно использовать для представления результатов

Достоверно различающиеся по пост-хок тесту группы обозначим разными буквами

```
gg_means +  
  geom_text(aes(label = c("A", "A", "A", "AB", "B"),  
                 vjust = -0.3, hjust = 1.5), size = 6)
```



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](#)
- Sokal, Rohlf, 1995, pp. 179-260
- Zar, 2010, pp. 189-207