Занятие 6

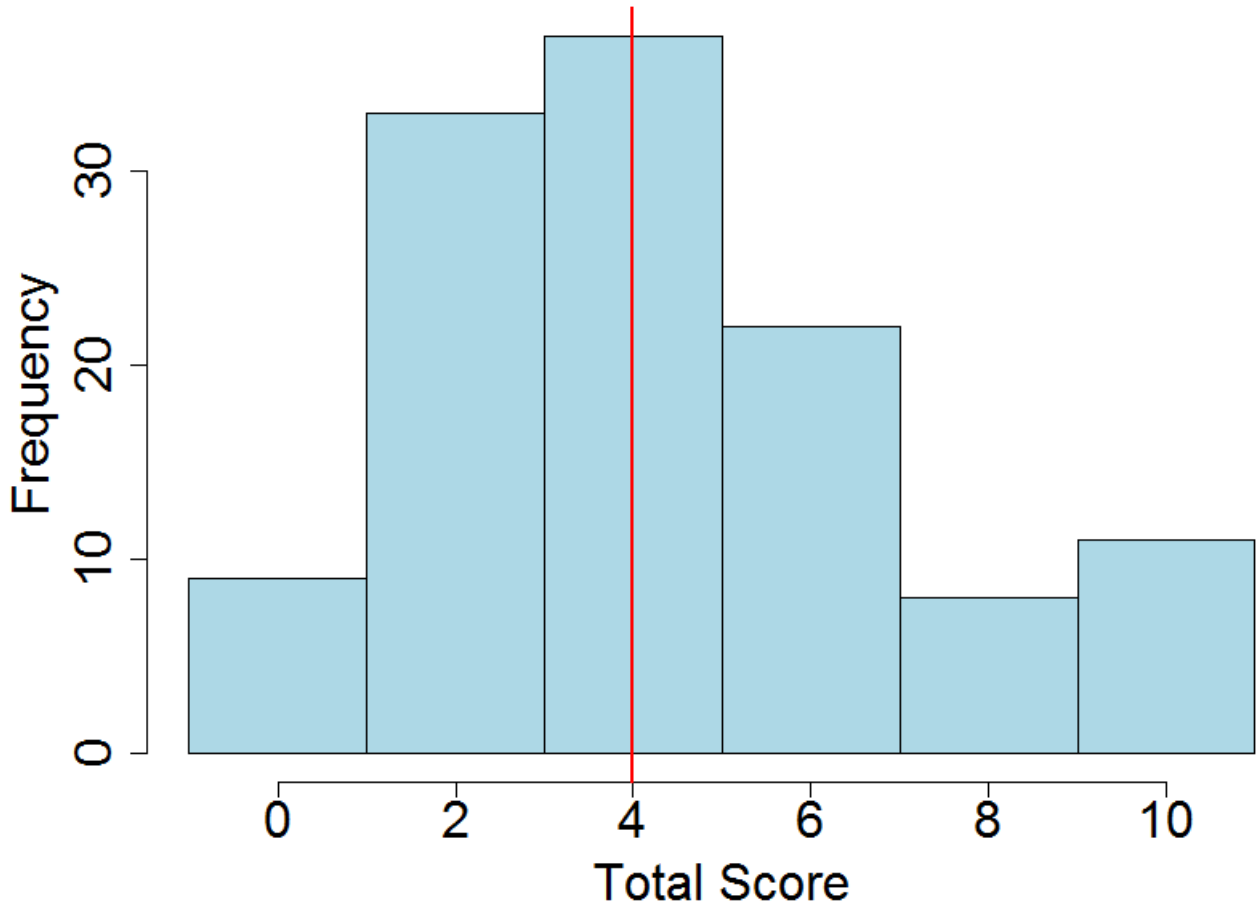
ФББ, 20 марта 2013 г.

План

- Разбор контрольной работы
- Еще раз про циклы и функции
- Еще раз про элементарную статистику
- Множественное тестирование

Разбор КР

Распределение оценок

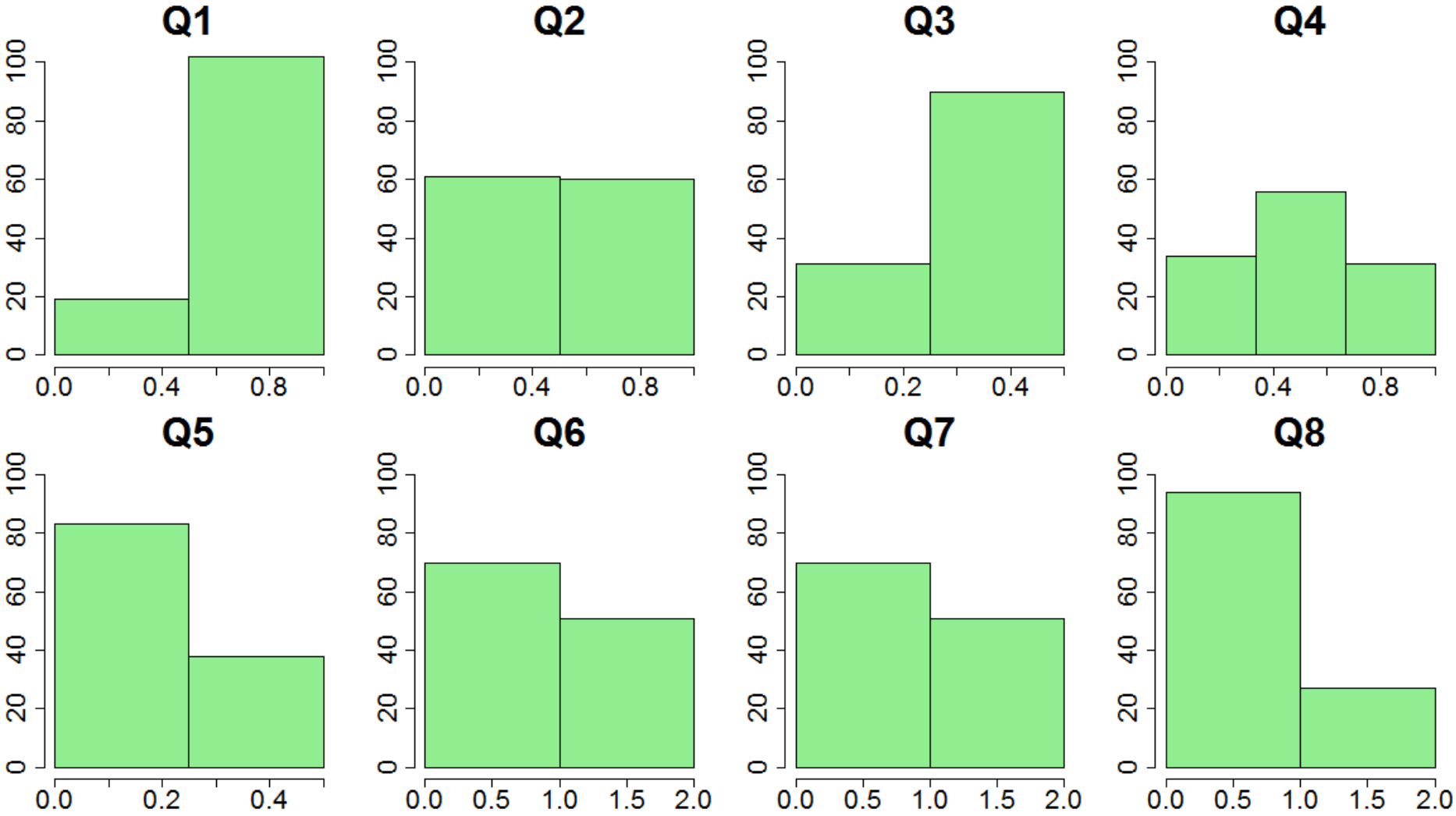


Всего работу написал 121 человек

Вытекающие последствия

- Разбор статистики еще раз
- Сообщение правильных ответов на вопросы домашних работ только после их закрытия

По задачам



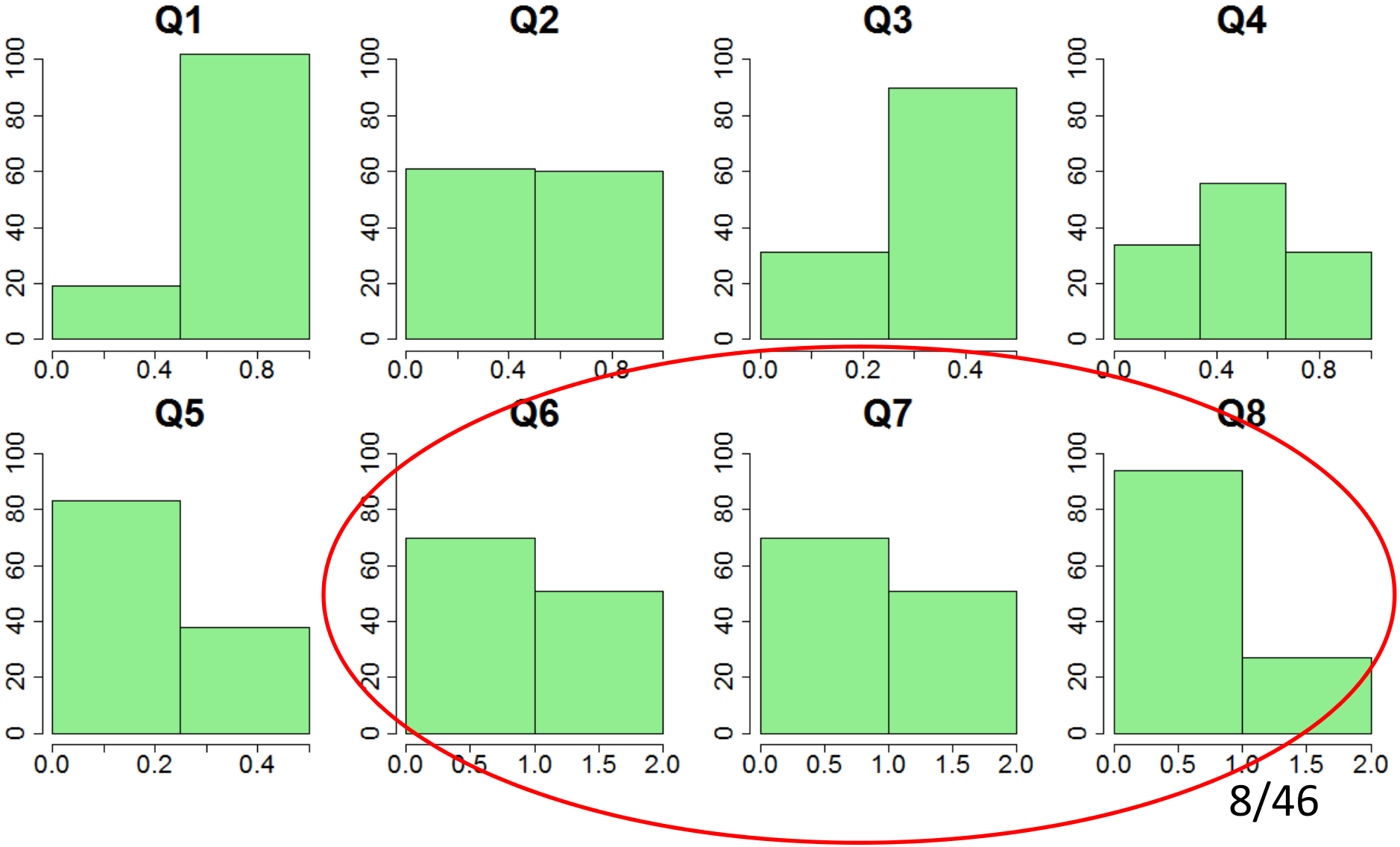
«Простые» задачи

- Задача №1 (графики): `plot`
- Задача №2 (boxplot): `data.frame,(cut),boxplot`
  - **cut:** разбивает значения вектора на интервалы и создает фактор, определяющий принадлежность каждого значения к определенному интервалу

```
> v <- 1:10
> cut(v,3)
[1] (0.991,4] (0.991,4] (0.991,4] (4,7] (4,7] (4,7] (4,7] (7,10]
[9] (7,10] (7,10]
Levels: (0.991,4] (4,7] (7,10]
```

- Задача №3 (значения, повторяющиеся N раз): `table`

По задачам

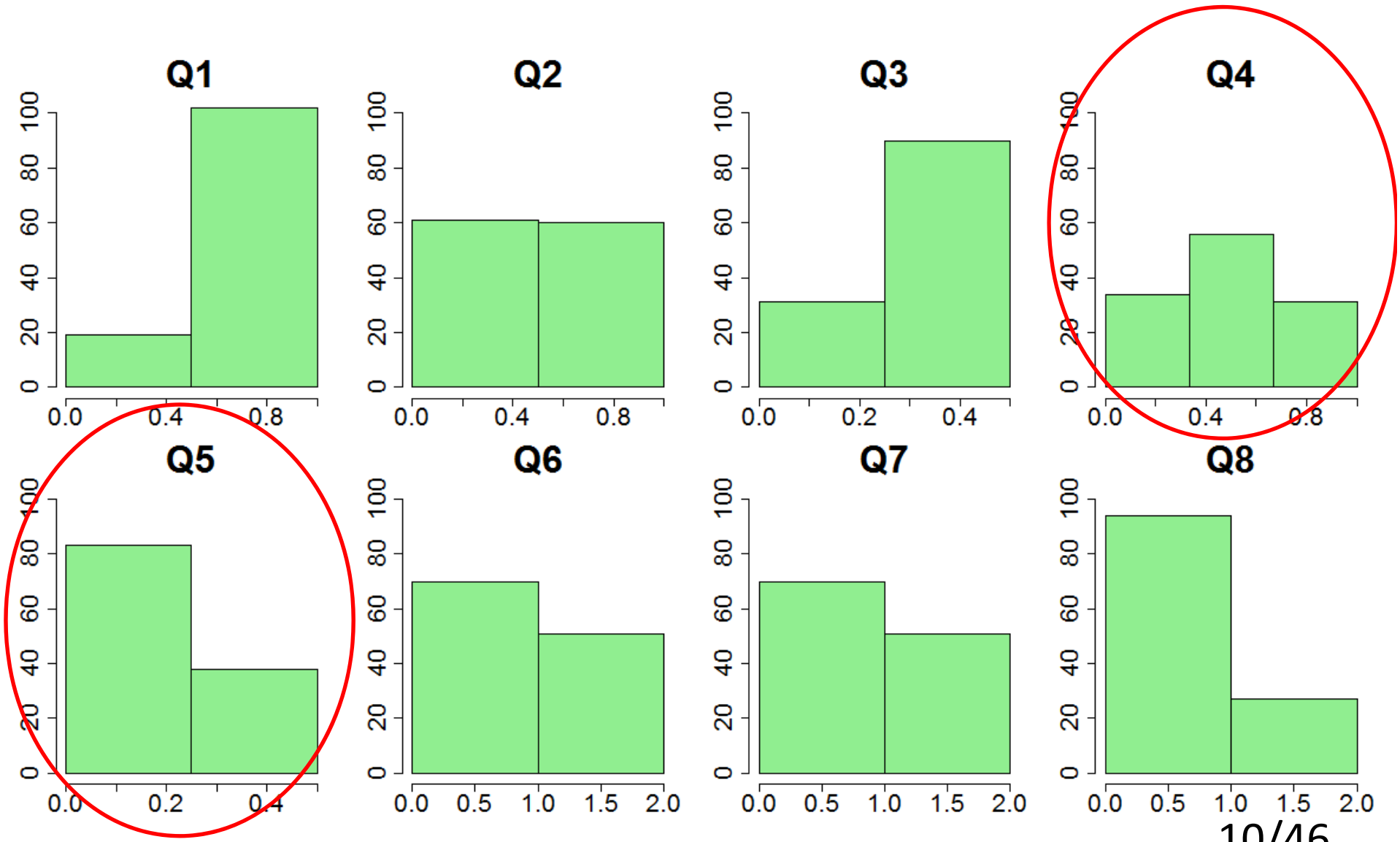


«Сложные» задачи: 6-8

(применение стат. тестов + merge + apply)

- Мало кто попробовал решать (а зря!)
- Те, кто решал, в основном не ошибались
- Техническая трудность (судя по вопросам на КР): проблема с чтением «нетипичных» файлов (например, с комментариями) -> на текстовый файл полезно смотреть глазами прежде чем загружать в R
- Типичная ошибка: неправильное решение о принятии/отвержении H_0 при правильно посчитанном p-value

По задачам



«Проблемные» задачи: 4, 5

- Задача № 4 (при данном P-value выбрать верны утверждения о принятии/отвержении гипотез)
  - У многих полное непонимание, что же делать с полученным p-value:
    - непонимание, что такое p-value?
    - незнание нулевой гипотезы?
    - непонимание, как связать p-value и уровень значимости?
  - Путаница с альтернативными гипотезами: two.sided, greater, less

Альтернативные гипотезы

- В каждом тесте могут быть разные определения, что значит «two.sided», «greater», «less» => хорошо читайте описание теста, если вы не знаете
- На примере непарного теста Стьюдента:
  - two.sided (default): средние значения выборок отличаются (но неизвестно, в какую сторону!)
  - greater: среднее значение первой выборки <u>больше</u> среднего значения второй выборки
  - less: среднее значение первой выборки <u>меньше</u> среднего значения второй выборки
- То, что средние отличаются (two.sided) при данном уровне значимости, не значит, что мы можем делать вывод, о том, что среднее значение первой выборки, например, больше среднего значения второй выборки (потому что оно может оказаться меньше)!

Простые правила для p-value

1) Нулевая гипотеза как правило соответствует случайному, «неинтересному» («по умолчанию») устройству данных:

- выборки/средние/распределения не различаются
- признаки независимы

2) P-value – вероятность наблюдаемого при нулевой гипотезе («мера веры в нулевую гипотезу») => если **«достаточно маленькая»**, то нулевая гипотеза маловероятна => мы ее отвергаем в пользу заранее заданной альтернативы (two.sided, greater, less)

3) Уровень значимости – заранее заданный кем-то (вами же, нами, научным сообществом) порог на p-value: если p-value меньше уровня значимости, значит p-value **«достаточно маленькое»** => отвергаем H_0 при данном уровне значимости

«Проблемные» задачи: 4, 5

- Задача № 5 (какому из двух тестов верить?)
  - Многие правильно выбрали, какому тесту верить, **но** неправильно ответили на вопрос о различие выборок!
  - Основной довод при выборе применимых (!) тестов: лучше применять более мощный тест (то есть тот, который скорее «распознает сигнал», если он есть)
  - Параметрические тесты (t-тест) как правило мощнее непараметрических (U-критерий Манна-Уитни, T-критерий Вилкоксона)

Циклы и функции

Цикл “for”

```
> for(i in 1:4){print(i)}     # на каждой итерации цикла
[1] 1                         # в переменную i кладется
[1] 2                         # следующий элемент вектора
[1] 3                         # 1:4
[1] 4
> v <- c("a","b","c")
> for(i in 1:length(v)){print(v[i])}
[1] "a"           # на каждой итерации цикла в переменную i
[1] "b"           # следующий элемент вектора 1:length(v)
[1] "c"
> seq_along(v)                # удобный способ “пробежаться”
[1] 1 2                       # по индексам вектора/списка
> for(i in seq_along(v)){print(v[i])}
[1] "a"
[1] "b"
[1] "c"
> for(ch in v){print(ch)}
[1] "a"
[1] "b"
[1] "c"
```

Цикл “for”

Ровно то же самое со списками:

```
> l <- list("a", 35, "b")

> for(i in seq_along(l)){print(l[[i]])}
[1] "a"
[1] 35
[1] "b"

> for(e in l){print(e)}
[1] "a"
[1] 35
[1] "b"
```

sapply

```
> lst <- list("one", 35, "two")
```

```
v=c()
for (i in lst)
{
  el <- i
  if (i=="one") {el<-1}
  else if (i=="two"){el<-2}
  else if (i=="three"){el<-3}
  v <- c(v, el)
}
> v
[1] 1 35 2
```

```
v <- sapply (lst, function(i)
{
  if (i=="one") {return(1)}
  else if (i=="two"){return(2)}
  else if (i=="three"){return(3)}
  return(i)
})
```

Функции

Зачем они нужны:

- Для многоразового использования определенного куска кода (то есть «сохранения» некого «действия», которое нужно делать не один раз – например, посчитать среднее значение для вектора чисел)
- Для передачи функции (то есть описания действий) в качестве аргумента другой функции: apply, sapply, aggregate…

Функции

Функцию можно описать и положить в переменную:

```
myFunction <- function(a1, a2){...}
```

название функции (может не быть)

аргументы функции (может не быть)

тело функции

И потом вызвать ее по имени, передав ей значения аргументов:

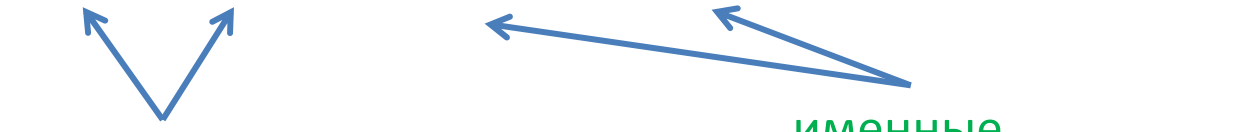
```
myFunction(3,1)
```

Или передать в качестве аргумента другой функции

Аргументы функции

Функцию можно описать и положить в переменную:

```
function(a1, a2, a3=NULL, a4=0){...}
```




```
t.test(x, y = NULL, alternative = c("two.sided",
       "less", "greater"), mu = 0, paired = FALSE,
       var.equal = FALSE, conf.level = 0.95, ...)
```

Чтобы посмотреть аргументы функции:

```
> args(t.test)
function (x, ...)
NULL
```

Функции

Функции могут быть:

- Встроенные (от есть из основных пакетов R) – они доступны всегда
- Из других пакетов R – чтобы они были доступны, нужно загрузить соответствующий пакет
- Пользовательские (то есть ваши) – их нужно описать и положить в переменную (функция будет доступна под именем этой переменной)

Функции

```
> head(grades, 3)
      student write math science   soc
1  student70    52   41      47    57
2 student121    59   53      63    61
3  student86    33   54      58    31
# функция, считающая P-value теста Шапиро для каждой строчки
> shapiro <-function(x){shapiro.test(x[2:length(x)])$p.value}
> apply(grades[1,],1, shapiro)
Ошибка: is.numeric(x) is not TRUE

> apply(grades[1,],1, function(x){x}) # хороший способ понять, в
        1                             # в чем проблема -
student "student70"                   # распечатать
write   "52"
math    "41"
science "47"
socst   "57“
# немного изменим аргумент и саму функцию:
> shapiro <- function(x){shapiro.test(x)$p.value}
> apply(grades[1,2:length(grades)],1, shapiro)
        1
0.9767631
```

Кое-что о вероятности и статистике

Вероятность

Лаплас: “*Вероятность – это здравый смысл и немного математики*”

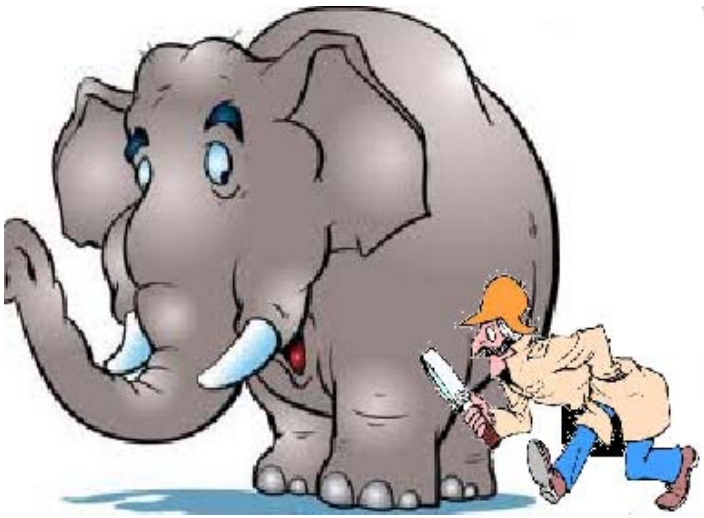
История: страховое дело, азартные игры

Вероятность: какие ставки (в деньгах!) Вы готовы делать на то или иное событие

Аксиоматическая теория: необходима для доказательств и определения области применения

Статистика

- Есть представление о “***генеральной совокупности***” - это то, что как-бы есть, но мы ее не видим
- Есть ***наблюдение*** — это то, что мы видим
- Мы хотим по *наблюдениям* понять свойства *генеральной совокупности*



Статистика: проверка гипотез

- **Нулевая гипотеза**: мир устроен плохо (случайно) и все, что мы наблюдаем не имеет никакого смысла.
- ***P-value*** — вероятность того, что данные пришли из плохого (случайного) мира.

Статистика (число)

- Есть набор данных (*выборка*)
- По выборке вычисляем ЧИСЛО (статистику S)
- Если мы знаем, что мир устроен случайно, то мы можем представить распределение вероятностей для этого ***числа***
- Если ***число*** очень необычное, то одно из двух:
  - Мир устроен не так, как мы предполагали
  - Нам сильно повезло (не повезло)

Проверка гипотез

Нулевая гипотеза H_0: мир устроен не интересно (плохо, случайно)

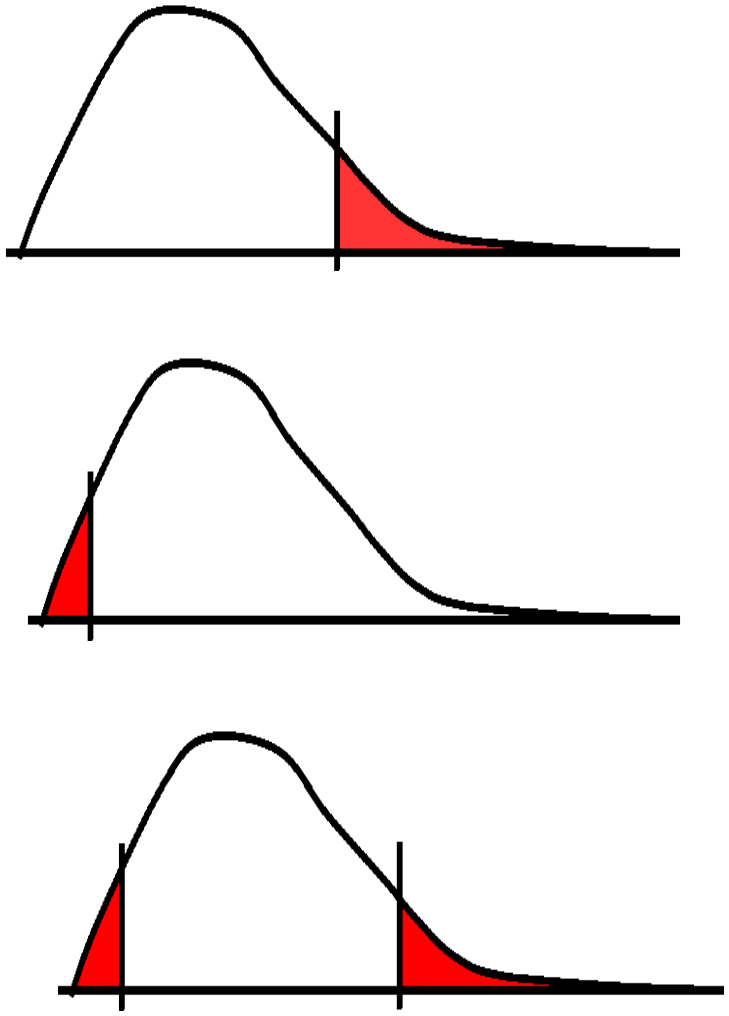
Для проверки нулевой гипотезы считаем Число (статистику) *S*.

В случае, если гипотеза H_0 верна, мы знаем распределение статистики $f(s), s \epsilon H_0$

Для наблюдения *S* мы можем определить место нашего наблюдения среди всех аналогичных наблюдений из H_0

p-value

- вероятность того, что значение статистики S необычно большое (правый односторонний тест)
- вероятность того, что значение S маленькое (левый тест)
- Вероятность того, что S необычное



Пример

Нулевая гипотеза H_0: популяция имеет нормальное распределение с $E(x)=e=2.718281828$.

Выборка $X=\{x_1,\ldots,x_n\}$. Среднее значение $m=E(X)$ разумеется не равно e. **Но:**

Если мы рассмотрим разнообразные выборки $\{X^*\}$ из H_0, мы получим множество средних значений $\{E(X^*)\}$, тоже не равных e.

Спасибо Уильяму Госсету (Стьденту), мы знаем распределение величины:

$$t=\frac{|\bar{x}-m|}{s/\sqrt{n}},\quad s=\frac{\sum(x_i-\bar{x})^2}{n-1}$$

$$p-value=\Pr(\tau>t\mid H_0)$$

p-value

P-value – вероятность увидеть то, что видим (или еще хуже) при условии, что значения пришли из H_0.

$$p - value = \Pr(\tau > t \mid H_0)$$

p-value <<1 (маленькое)	С вероятностью 1 – *p -value* H_0 ***отвергается*** и альтернативная гипотеза H_A принимается
p-value ≈1 (большое)	с вероятностью p-*value* H_0 ***принимается*** и альтернативная гипотеза H_A отвергается

p-value

p-value – ***НЕ*** есть вероятность гипотезы! Это вероятность ***наблюдения***!

$$\Pr(H_0 \mid \tau > t) = \frac{\Pr(\tau > t \mid H_0) \cdot \Pr(H_0)}{\Pr(\tau > t_0)}$$

$$\Pr(\tau > t_0) = \Pr(\tau > t \mid H_0)\Pr(H_0) + \Pr(\tau > t \mid H_A)(1 - \Pr(H_0))$$

Это единственное, что нам известно!

p-value и мощность теста

	На самом деле H_0	На самом деле H_A
Приняли H_0	Правильно приняли	**Ошибки II рода**
Отвергли H_0, Приняли H_A	**Ошибки I рода**	Правильно отвергли

$$1 - \Pr(error\ type\ II) = мощность\ критерия$$

Про функцию распределения

Интегральная функция распределения

$$F(x) = \Pr(\xi \le x) = \int_{-\infty}^{x} f(t)dt$$

Интегральная функция распределения это просто функция. Если в нее подставить случайную величину, то мы получим новую случайную величину

$$\zeta = F(\xi)$$

Если ξ – это та случайная величина, для которой F является интегральной функцией распределения, то:

$\zeta = F_{\xi}(\xi)\ ;\ \ f_{\zeta}(x) = 1$ Распределена равномерно на [0,1]

Вывод: *p-value* распределены ***равномерно*** для выборок из H_0.

Множественное тестирование

Вы бросили 6 костей и набрали 36. Вероятность события $p \approx 2 \cdot 10^{-5}$. Событие удивительное.

Вы бросили 6 костей 10 тыс. раз и набрали в одном из случаев 36. Удивительно ли это?

Общая постановка проблемы: есть много наблюдений (чисел). Хотелось бы среди них отобрать те, которые пришли не из модели H_0 – т.е. «не случайны»

Множественное тестирование. Поправка Бонферрони.

Мы просто умножаем вероятность события на число испытаний. Если после такой процедуры ***Число*** останется достаточно маленьким, то мы продолжаем удивляться.

$$Bp = p \cdot N = 2 \cdot 10^{-5} \cdot 10^{4} = 0.2$$

Не очень-то и удивительно

Контроль частоты ошибок

Есть набор наблюдений (чисел) $\{x_1, \ldots . x_n\}$. Им можно поставить в соответствие вероятности (например, $p_i = \Pr(\xi \geq x_i)$) Вероятности можно упорядочить:

$$0 \leq p_1 \leq p_2 \leq p_3 \leq p_4 \leq \ldots \ldots \leq p_N$$

Контроль частоты ошибок: хочется назвать номер n, такой что все эксперименты с $i \leq n$ нас устраивают (пришли не из фона-шума), а остальные – нет.

Множественное тестирование. Поправка Бонферрони.

Бонферрони: все p умножить на N.

Зададим число α : вероятность того, что хотя бы один результат из хороших получен случайно, не превосходит α .

$$\max i : N \cdot p_i \leq \alpha$$

1) Чиним ошибку 1 рода, получаем 2 рода – слишком строгий отбор
2) Независимость

Множественное тестирование. Доля ложных открытий (FDR)

Тест был применён к куче единичных испытаний.
«Открытие» – наблюдение (одно из выборки) не соответствует фоновой (нулевой) модели

	Test passed	Test failed
True	TP	FN
False	FP	TN

$$p = E\left(\frac{FP}{FP + TN}\right)$$

$$FDR = E\left(\frac{FP}{FP + TP}\right)$$

Для оценки p-value было достаточно знать нулевую модель (она же шум). Для FDR – ещё и модель сигнала.

Benjamini , Hochberg

$$0 \le p_1 \le p_2 \le p_3 \le p_3 \le p_4 \le \ldots\ldots \le p_N$$

Мы хотим контролировать FDR на уровне α.

$$\max i : \left(\frac{Np_i}{i} \right) \le \alpha$$

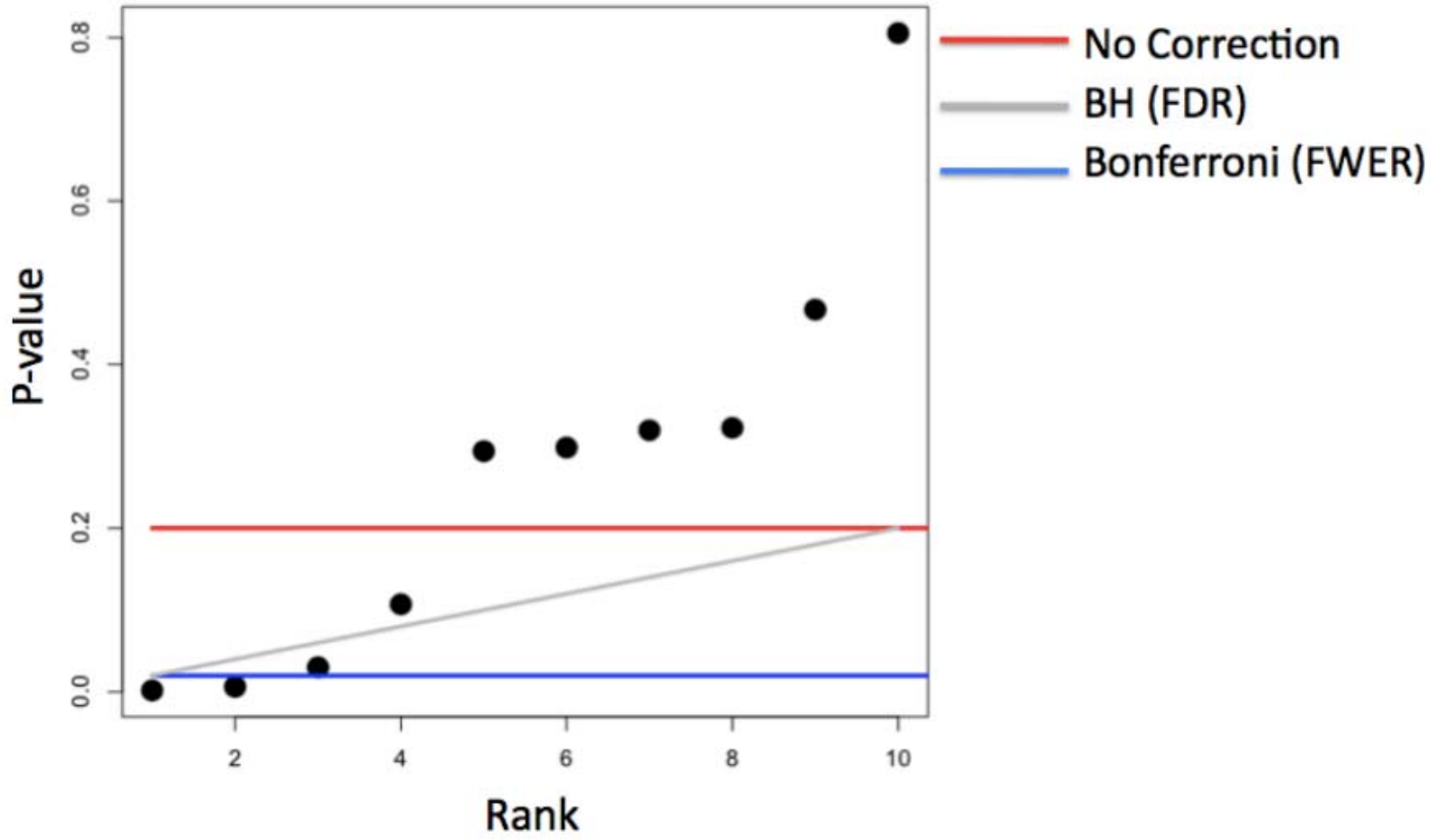
В качестве оценки FP+TP фигурирует i. В остальном, эта оценка устроена так же, как Бонферрони, и тоже предполагает независимость испытаний.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B (Methodological) 57 (1), 289-300.

Множественное тестирование

Тип контролируемой ошибки (error rate)	Метод коррекции	Коррекция порога на p-value, чтобы error < α (при m тестах)	Коррекция p-value (p), чтобы error < α (при m тестах)
False positive rate: $p = E\left(\frac{FP}{FP+TN}\right)$	Без коррекции	нет	нет
Family wise error rate (FWER): *FWER = P(FP ≥ 1)*	Поправка Бонферрони (Bonferroni)	Считаем значимыми $p-value(i) < \frac{\alpha}{m}$	Сравниваем с α $p_{FWER} = \max(m \times p, 1)$ для каждого p (p-value)
False Discovery Rate: $FDR = E\left(\frac{FP}{FP+TP}\right)$	Поправка Benjamini-Hochberg	Считаем значимыми $p-value(i) < \alpha \times \frac{i}{m}$ (***i*** – ранк p-value)	Сравниваем с α $p(i)_{BH} < \frac{p(i) \times m}{i}$ (***i*** – ранк p-value)

Пример для 10 p-value



Пример: в выборке нет TP

```
> set.seed(12345)
> pValues <- rep(NA,1000)
> for(i in 1:1000)
+ {
+   x <- rnorm(20)
+   pValues[i] <- t.test(x)$p.value
+ }

# Control false positive rate
> sum(pValues < 0.05) # ≈ 0.05*1000
[1] 51

# Control FWER
> sum(p.adjust(pValues, method="bonferroni") < 0.05)
[1] 0

# Control FDR
> sum(p.adjust(pValues, method="BH") < 0.05)
[1] 0
```

Пример: в выборке 50% TP

```
# Генирируем 500 выборок с mean = 0 и
# 500 выборок с mean = 1.5 -> применяем t-test
# тестирования среднего 0


> set.seed(12345)
> pValues <- rep(NA,1000)
> for(i in 1:500){pValues[i] <- t.test(rnorm(20))$p.value}
> for(i in 501:1000){pValues[i] <- t.test(rnorm(20,
+ mean=1.5))$p.value}
# сохраняем првильные ответы
> trueStatus <- rep(c("zero", "not zero"), each=500)
```

Пример: в выборке 50% TP

```
> trueStatus <- rep(c("zero", "not zero"), each=500)
# Control false positive rate
> table(pValues < 0.05, trueStatus)
       trueStatus
        not zero zero
  FALSE        0  478
  TRUE       500   22
# Control FWER
> table(p.adjust(pValues, method="bonferroni")<0.05,
trueStatus)
       trueStatus
        not zero zero
  FALSE       59  500
  TRUE       441    0
# Control FDR
> table(p.adjust(pValues, method="BH")<0.05,trueStatus)
trueStatus
        not zero zero
  FALSE        0  487
  TRUE       500   13
```