Лекция 6

Функции в R

```
dist_to_origin <- function(x){
    return ((x[1] * x[1] + x[2] * x[2]) ** 0.5)
}
```

Задали функцию, которая для вектора из двух элементов считает расстояние до точки (0, 0)

```
dist_to_origin <- function(x){
    (x[1] * x[1] + x[2] * x[2]) ** 0.5
}
```

То же самое, возвращается результат последнего вычисления

```
dist_to_origin <- function(x, ...){
    (x[1] * x[1] + x[2] * x[2]) ** 0.5
}
```

То же самое, ... означает - игнорируем этот аргумент

Функции в R

Хотим сделать функцию, которая будет считать расстояние до точки в 2D или в 3D

```
dist_to_origin <- function(x){
    if (length(x) == 2) { # for 2D-case
        (x[1] * x[1] + x[2] * x[2]) ** 0.5
    } else if (length(x) == 3){ # for 3D-case
        (x[1] * x[1] + x[2] * x[2] + x[3] * x[3]) ** 0.5
    } else{
        stop("Can't calculate distance to origin")
    }
}
```

Если не тот размер, то ошибка

Минусы такого подхода

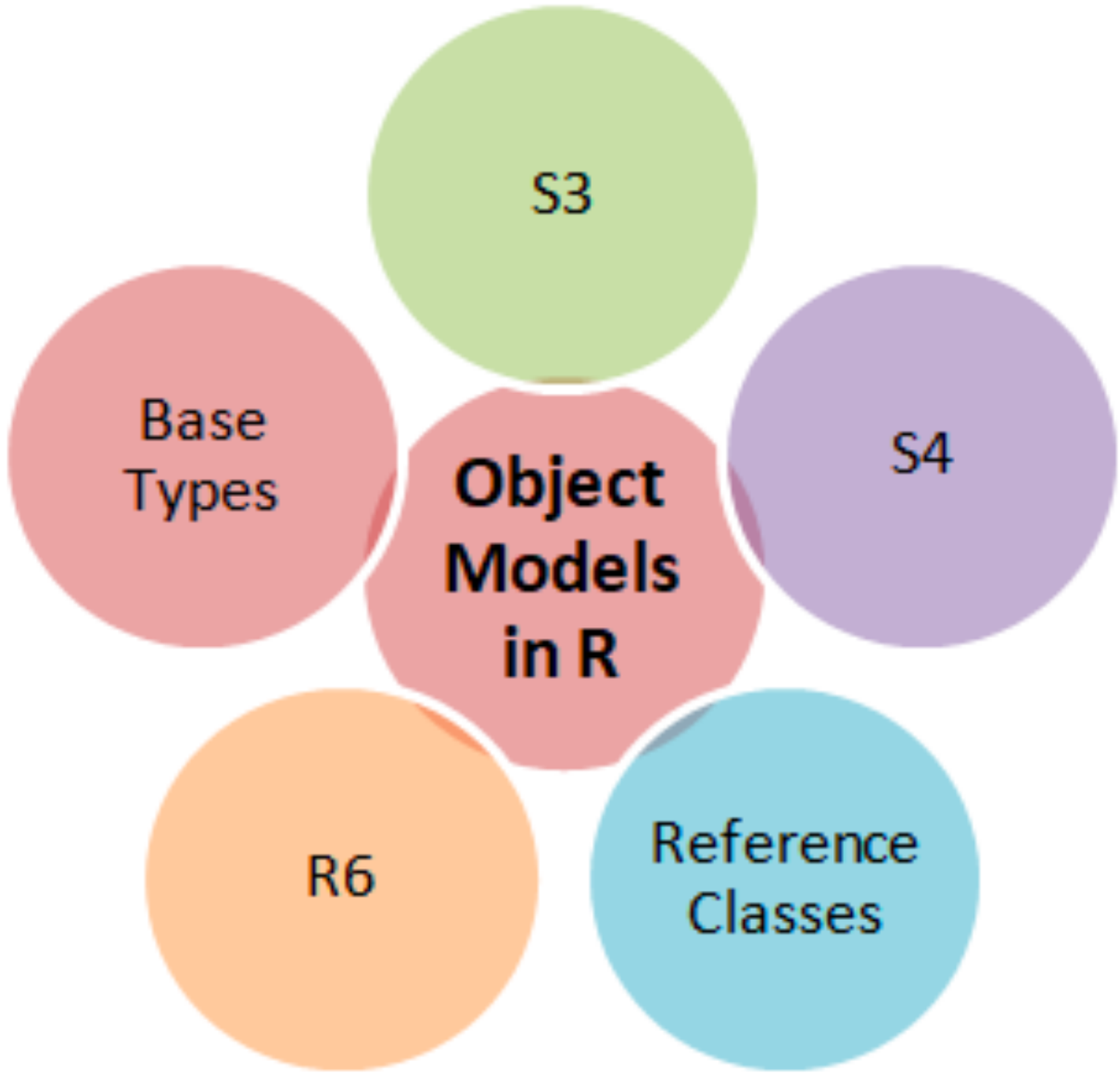
- Можем иметь несколько условий - на каждый набор условий нужно прописывать свое поведение;
- Можем иметь несколько схожих функций - в каждой функции надо писать один и тот же набор условий, можем ошибиться

ООП в R

Изначального ООП в R нет. Не подразумевалось, под него не закладывались какие-либо решения

Но ООП нужно, потому были сделаны надстройки. Так как надстройки получались с изюминкой, появилось 4 типа классов в R (ага, привет Perl)

ООП в R



Base-types

Не классы. Просто встроены в R. Лежат в основе остальных:

1) Векторы

2) Матрицы

3) Списки

4) Факторы

S3-объекты

Самый малофункциональный способ. При этом самый простой и часто используемый

Создание объекта класса

```
x <- c(1, 2)
class(x) <- 'Point'
```

Создание объекта класса

Можно “сделать“ объект представителем нескольких классов

```
x <- c(1, 2)
class(x) <- c("A", "B")
```

Класс A “наследует” от B. Если нет метода для класса A, вызывается метод для класса B.

Методы

```
x <- c(1, 2)
class(x) <- 'Point2D'
y <- c(1, 2, 3)
class(y) <- 'Point3D'
```

Хотим написать функцию, возвращающую расстояние до начала координат для обоих классов

Методы

```
x <- c(1, 2)
class(x) <- 'Point2D'
dist_to_origin.Point2D <- function(x){
    (x[1] ** 2 + x[2] ** 2) ** 0.5
}
```

```
y <- c(1, 2, 3)
class(y) <- 'Point3D'
dist_to_origin.Point3D <- function(x){
    (x[1] ** 2 + x[2] ** 2 + x[3] ** 2) ** 0.5
}
```

Написали отдельно по функции для каждого из классов

Если бы не было нормального решения

```
dist_to_origin <- function(x) {
    if ('Point2D' %in% class(x)){
        dist_to_origin.Point2D(x)
    } else if ('Point3D' %in% class(x)){
        dist_to_origin.Point3D(x)
    } else{
        stop("no applicable method")
    }
}
```

Работает, но это боль.
И почти ничем не отличается от решения без классов.

Методы

Можем написать специальную generic function

```
dist_to_origin <- function(x, ...){
UseMethod("dist_to_origin", x)
}
x <- c(1, 2)
class(x) <- "Point2D"
print(dist_to_origin(x))
```

```
## [1] 2.236068
```

```
y <- c(1, 2, 3)
class(y) <- "Point3D"
print(dist_to_origin(y))
```

```
## [1] 3.741657
```

Работает.
Если дать класс, для которого нет соответствующей функции - dist_to_origin.class_name, то выдастся ошибка

Методы

```
y <- c(1, 2, 3)
class(y) <- "Point2D"
print(dist_to_origin(y))
```

Что выдаст этот код?

Методы

```
y <- c(1, 2, 3)
class(y) <- "Point2D"
print(dist_to_origin(y))
```

```
## [1] 2.236068
```

Никакой проверки, что то, что вы назвали объектом класса, этим классом является


ОДНА ОШИБКА
И ТЫ ОШИБСЯ
risovach.ru

Методы

```
print(x)
```

```
## [1] 1 2
## attr(,"class")
## [1] "Point2D"
```

```
print(y)
```

```
## [1] 1 2 3
## attr(,"class")
## [1] "Point2D"
```

Хотим сделать более красивый вывод

Методы

```
print.Point2D <- function(x) {
    desc <- paste("Point2D\n",
                  "x :", x[1], "\n",
                  "y :", x[2])
    cat(desc) # to print \n as newline
}
print.Point3D <- function(x) {
    desc <- paste("Point3D\n",
                  "x1:", x[1], "\n",
                  "x2:", x[2], "\n",
                  "x3:", x[3])
    cat(desc) # to print \n as newline
}
```

Методы

```
x <- c(1, 2); class(x) <- "Point2D"
y <- c(1, 2, 3); class(y) <- "Point3D"
print(x)
```

```
## Point2D
##   x : 1
##   y : 2
```

```
print(y)
```

```
## Point3D
##   x1: 1
##   x2: 2
##   x3: 3
```

print уже generic-функция

Что в R вы использовали и оно уже S3-объект?

Что в R вы использовали и оно уже S3-объект?

```
df <- starwars
print(class(df))
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
print(typeof(df))
```

```
## [1] "list"
```

```
class(df) <- " "
print(df)
```

Что в R вы использовали и оно уже S3-объект?

```
class(df) <- " "
print(df)
```

```
## $name
##  [1] "Luke Skywalker"          "C-3PO"
##  [3] "R2-D2"                   "Darth Vader"
##  [5] "Leia Organa"             "Owen Lars"
##  [7] "Beru Whitesun lars"      "R5-D4"
##  [9] "Biggs Darklighter"       "Obi-Wan Kenobi"
## [11] "Anakin Skywalker"        "Wilhuff Tarkin"
## [13] "Chewbacca"               "Han Solo"
## [15] "Greedo"                  "Jabba Desilijic Tiure"
## [17] "Wedge Antilles"          "Jek Tono Porkins"
## [19] "Yoda"                    "Palpatine"
## [21] "Boba Fett"               "IG-88"
```

И так далее, печатается как list

А что еще в R вы использовали и оно уже S3-объект?

А что еще в R вы использовали и оно уже S3-объект?

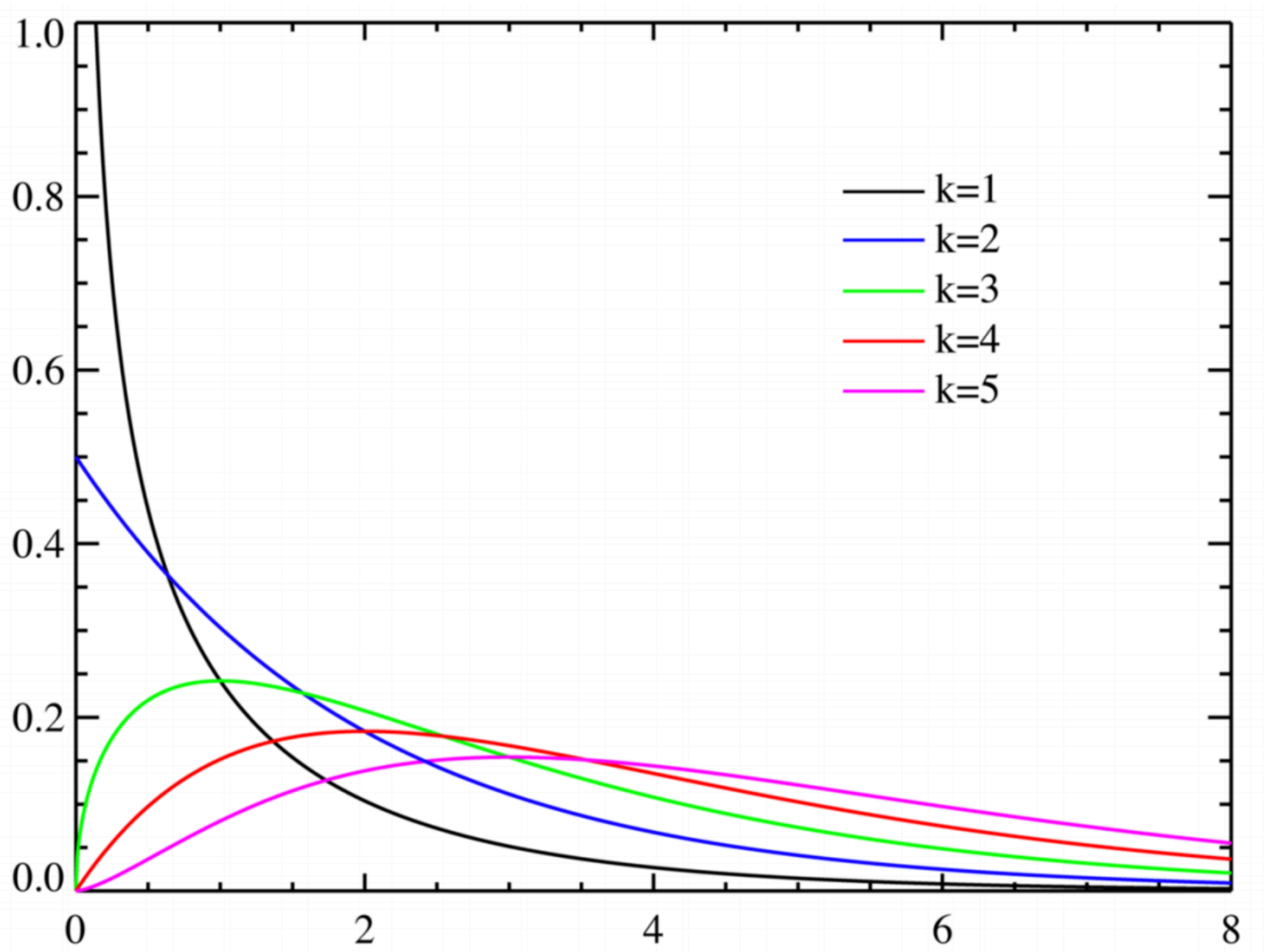
```
human <- starwars %>%
    filter(species == "Human") %>%
    pull(mass)
not_human <- starwars %>%
    filter(species != "Human") %>%
    pull(mass)
k <- t.test(human, not_human)
print(class(k))
```

```
## [1] "htest"
```

```
print(typeof(k))
```

```
## [1] "list"
```

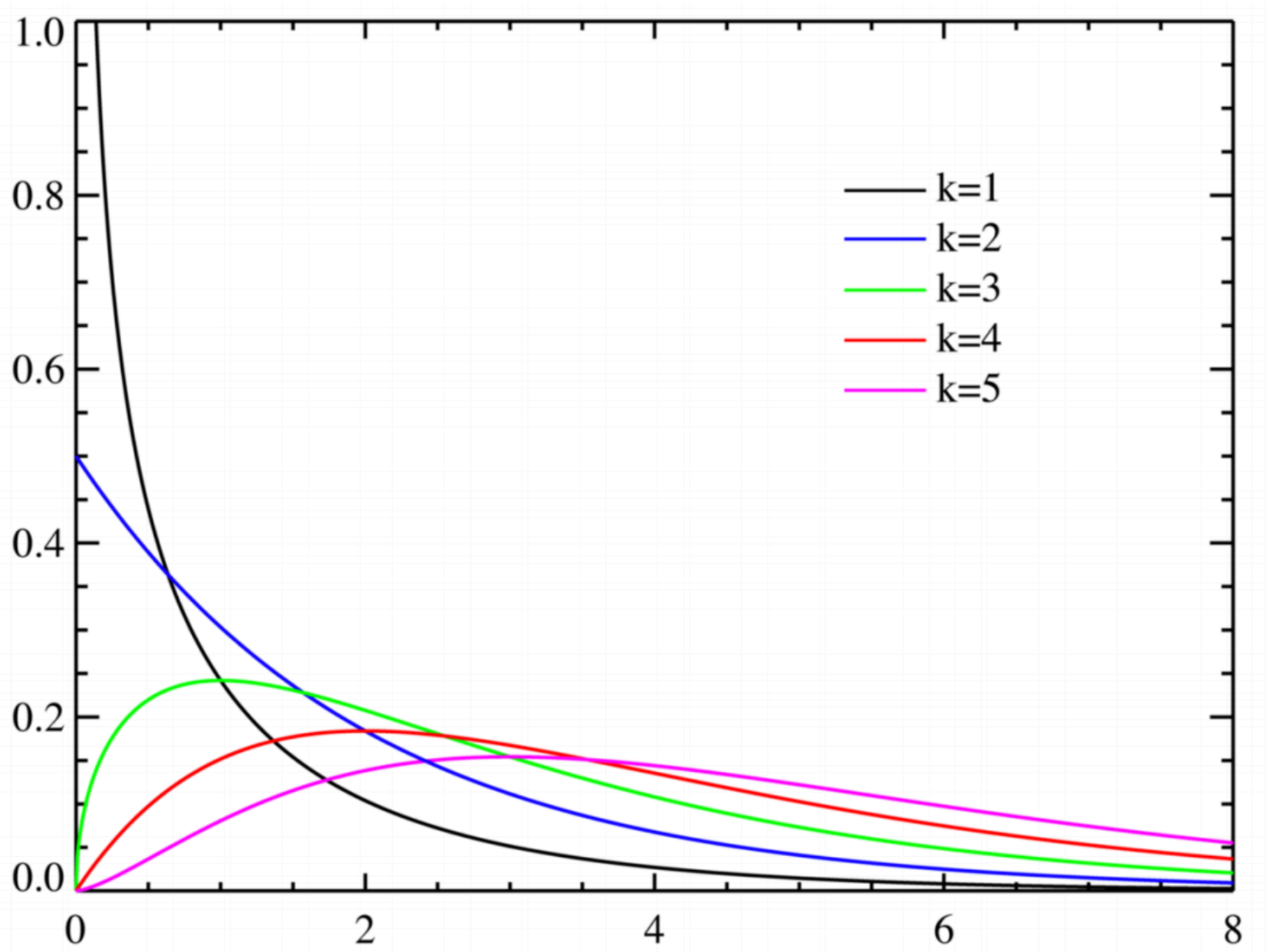
Распределение Хи-квадрат



Сумма k независимых стандартных нормальных случайных величин

Правильно?

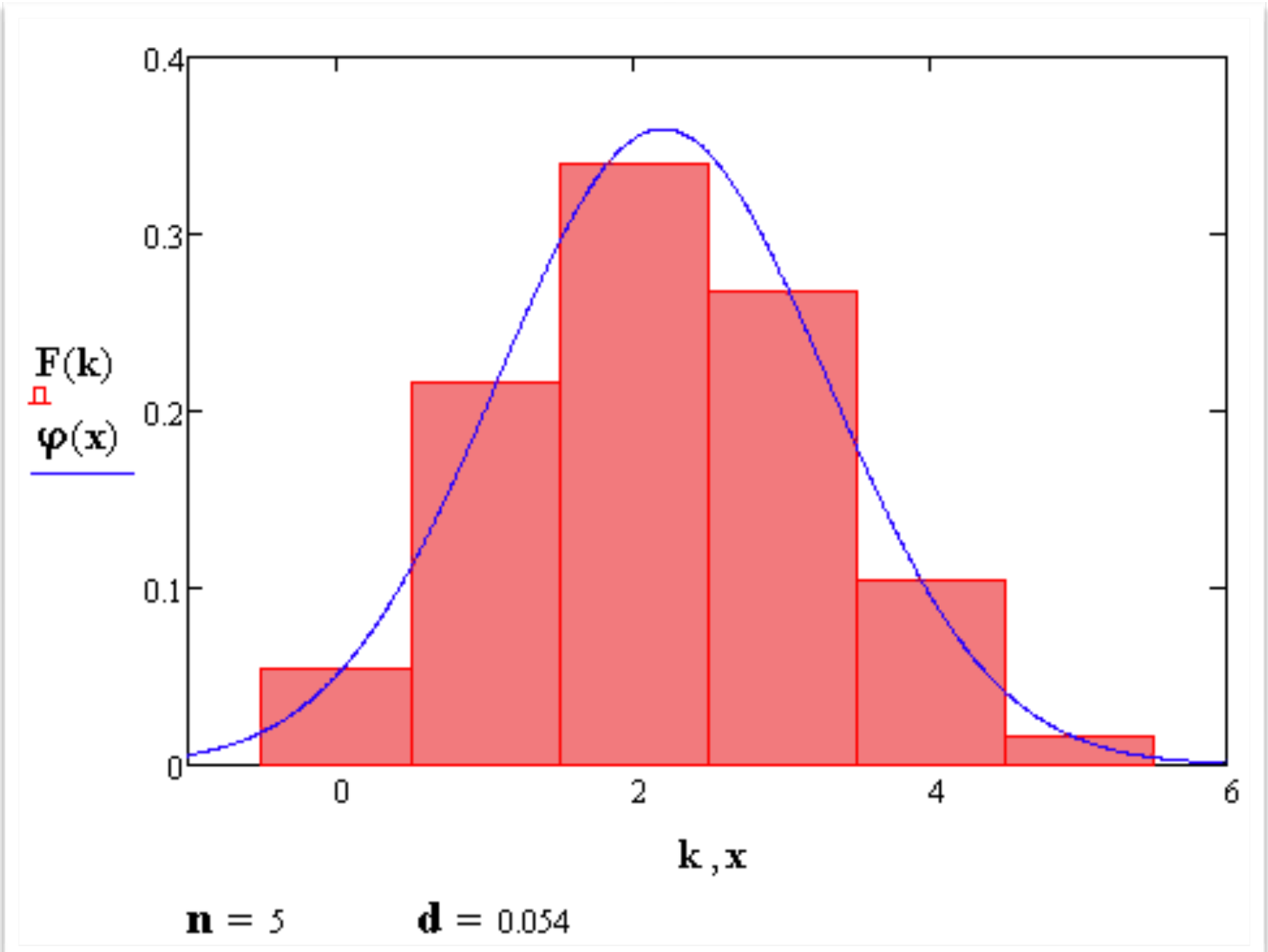
Распределение Хи-квадрат



Сумма квадратов k независимых случайных стандартных нормальных величин

Откуда они берутся в хи-тесте ?

Откуда они берутся в хи-тесте ?



Откуда они берутся в хи-тесте ?

Любой хи-квадрат тест - проверка того, что ваши данные распределены по мультиномиальному закону

Значение	val_0	val_1	val_2	val_3	val_4	…
Вероятность	p0	p1	p2	p3	p4	…

Критерий Хи-квадрат

Тест на Goodness of fit

Насколько ваша модель распределения данной переменной описывает реально наблюдаемые значения

H0: модель верна

H1: Модель неверна

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

$$df = n - 1$$

n - число ячеек

Задача

Программист Петя считает, что количество лайков, которые соберут посты с шутками на тему неприятных особенностей языка, одинаковы. Для теста были выбраны языки C++, Python, Javascript, Java и R. Количество лайков для постов про эти языки составило соответственно:

17, 23, 72, 44, 65

Прав ли Петя? Уровень значимости 0.001, так он не хочет никого в случае чего обидеть незаслуженно.

Задача

Программист Петя считает, что количество лайков, которые соберут посты с шутками на тему неприятных особенностей языка, одинаковы. Для теста были выбраны языки C++, Python, Javascript, Java и R. Количество лайков для постов про эти языки составило соответственно:

17, 23, 72, 44, 65

Прав ли Петя? Уровень значимости 0.001, так он не хочет никого в случае чего обидеть незаслуженно.

	C++	Python	Javascript	Java	R
Число лайков	17	23	72	44	65
Вероятность при условии H0	0.20	0.20	0.20	0.20	0.20

Решение

Гипотеза H0: Все языки получили равное число лайков, распределение лайков равномерное
Гипотеза H1: Языки получили значимо разное число лайков

Если распределение лайков равномерное, то ожидаемое число лайков для каждого языка:

$$E = 221/5 = 44.2$$

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected} = 56.2$$

$$df = n - 1 = 4$$

$$P(\chi^2(4) > 56.2) = 1e - 11 < 0.001$$

На уровне значимости 0.01 мы отвергаем гипотезу H0 о том, что распределение лайков равномерное

Goodness of fit

- Всегда ли число степеней свободы n - 1?

Goodness of fit

- Всегда ли число степеней свободы n - 1?

Нет

Оно зависит от того, сколько условий вы накладываете

Goodness of fit

- Всегда ли число степеней свободы n - 1?

Нет

Оно зависит от того, сколько условий вы накладываете на ваши наблюдения/величин из них считаете непосредственно до теста

В предыдущем случае есть только одно условие - сумма всех наблюдений равна n.

Потому из числа наблюдений (n) мы и вычитаем 1

Задача

Наблюдается число студентов, опаздывающих на 0 минут, минуту, две, три, четыре и 5 минут и более

Значение	0	1	2	3	4	5 и более
Студентов	14	30	33	14	6	3

Проверьте гипотезу о том, что число студентов распределено по Пуассону

Решение

Наблюдается число студентов, опаздывающих на 0 минут, минуту, две, три, четыре и 5 минут (больше не опаздывают)

Значение	0	1	2	3	4	5
Студентов	14	30	33	14	6	3

Если число студентов распределено по Пуассону, то lambda = (14 * 0 + 30 * 1 + 2 * 33 + 3 * 14 + 4 * 6 + 5 * 3) / 100 = 1.77

Можно подсчитать (по формуле или с использованием функции dpoiss R вероятность значения попасть в каждую из ячеек)

Решение

Значение	0	1	2	3	4	5
Студентов	14	30	33	14	6	3
p	0.17	0.30	0.27	0.16	0.07	0.02

Остается подсчитать ожидаемое число студентов

Значение	0	1	2	3	4	5
Студентов	14	30	33	14	6	3
p	0.17	0.30	0.27	0.16	0.07	0.02
Ожидаемое	17	30	27	16	7	2

Решение

Теперь можно подсчитать значение статистики, оно равно 2.76

В этом случае у нас было условие на то, что наблюдений суммарно N = 100 и на то, что lambda нашего Пуассоновского распределения равна 1.77 (мы ее считали из наших наблюдений)

Потому суммарно получаем число степеней свободы равным n - 2 = 6 - 2 = 4.

Получаем, что p-value близко к 1, то есть у нас нет оснований отвергать гипотезу о том, что наблюдения распределены по Пуассону.

Решение

Сколько степеней свободы надо в случае:

- Проверить гипотезу о том, что наши наблюдения распределены по нормальному закону?

- Проверить гипотезу о том, что наши наблюдения распределены по равномерному закону?

- Проверить гипотезу о том, что наши наблюдения распределены по Пуассону с параметром 2?

Решение

Сколько степеней свободы надо в случае:

- Проверить гипотезу о том, что наши наблюдения распределены по нормальному закону?

  n - 1 - 2 = n - 3, считаем среднее и дисперсию
- Проверить гипотезу о том, что наши наблюдения распределены по равномерному закону?

  n - 1, мы это делаем по-умолчанию
- Проверить гипотезу о том, что наши наблюдения распределены по Пуассону с параметром 2?

  n - 1, мы взяли параметр не из наблюдений

Критерий Хи-квадрат

Тест на независимость

Используется как на то, есть ли значимая ассоциация между двумя факторными переменными

H0: факторы независимы

H1: факторы зависимы

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

$$df = (n-1) \cdot (m-1)$$

Где df - число степеней свободы, n - число разных значений первой переменной, m - число разных значений второй

Задача

Для четырех категорий людей - школьников, студентов, программистов (закончивших учебу со стажем < 5 лет и программистов (закончивших учебу) со стажем больше 5 лет имеются данные о их отношении к PHP. Отношение может быть “хороший язык”, “ну а шо поделать” “ненавижу”. Проверить гипотезу о том, что категории независимы. Уровень значимости принять равным 0.01

Отношение/ Категория	Школьники	Студенты	Программист, < 5 лет	Программист, > 5 лет
Хороший язык	40	22	17	12
Ну а шо поделать	15	12	20	35
Ненавижу	35	20	22	10

Решение

Гипотеза H0: Отношение не зависит от категории

Гипотеза H1: Отношение зависит от категории

Если отношение не зависит от категории, то P(хороший язык, категория) = P(хороший язык) * P(категория). То есть вероятность объекта оказаться в ячейке - произведение вероятностей в соответствующих столбце и строке

Отношение /Категория	Школьники	Студенты	Программист, < 5 лет	Программист, > 5 лет	Сумма	Вероятность
Хороший язык	40	22	17	12	91	0.35
Ну а шо поделать	15	12	20	35	82	0.32
Ненавижу	35	20	22	10	87	0.33
Сумма	90	54	59	57	260	
Вероятность	0.35	0.21	0.23	0.22		-

Решение

Тогда ожидаемые нами числа:

Отношение /Категория	Школьники	Студенты	Программист, < 5 лет	Программист, > 5 лет	Сумма	Вероятность
Хороший язык	31,85	19,11	20,93	20,02	91	0,35
Ну а шо поделать	29,12	17,472	19,136	18,304	82	0,32
Ненавижу	30,03	18,018	19,734	18,876	87	0,33
Сумма	90	54	59	57	260	
Вероятность	0,35	0,21	0,23	0,22		-

Решение

Посчитаем значение критерия хи-квадрат

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

$$\chi^2 = 35.8$$

$$df = (4-1) \cdot (3-1) = 6$$

$$P(\chi^2(6) > 33.8) = 0.0004 < 0.01$$

На уровне значимости 0.01 мы отвергаем гипотезу H0 о независимости

chisq.test

Pearson's Chi-squared Test for Count Data

Description

`chisq.test` performs chi-squared contingency table tests and goodness-of-fit tests.

Usage

```
chisq.test(x, y = NULL, correct = TRUE,
           p = rep(1/length(x), length(x)), rescale.p = FALSE,
           simulate.p.value = FALSE, B = 2000)
```

Задача

Программист Петя считает, что количество лайков, которые соберут посты с шутками на тему неприятных особенностей языка, одинаковы. Для теста были выбраны языки C++, Python, Javascript, Java и R. Количество лайков для постов про эти языки составило соответственно:

17, 23, 72, 44, 65

Прав ли Петя? Уровень значимости 0.001, так он не хочет никого в случае чего обидеть незаслуженно.

```
x <-  c(17, 23, 72, 44, 65)

chisq.test(x=x, p = rep(1, length(x)) / length(x))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  x
## X-squared = 54.181, df = 4, p-value = 4.823e-11
```

Задача

Для четырех категорий людей - школьников, студентов, программистов (закончивших учебу) со стажем < 5 лет и программистов (закончивших учебу) со стажем больше 5 лет имеются данные о их отношении к PHP. Отношение может быть “хороший язык”, “ну а шо поделать” “ненавижу”. Проверить гипотезу о том, что категории независимы. Уровень значимости принять равным 0.01

Отношение/ Категория	Школьники	Студенты	Программист, < 5 лет	Программист, > 5 лет
Хороший язык	40	22	17	12
Ну а шо поделать	15	12	20	35
Ненавижу	35	20	22	10

Задача

```
cont_tab <- rbind(c(40, 22, 17, 12),
                  c(15, 12, 20, 35),
                  c(35, 20, 22, 10))

chisq.test(x=cont_tab)
```

```
##
##  Pearson's Chi-squared test
##
## data:  cont_tab
## X-squared = 36.21, df = 6, p-value = 2.51e-06
```

Проблемы с критерием Хи-квадрат

Критерий Хи-квадрат можно применять только тогда, когда ожидаемое число наблюдений в каждой клетке больше 5.
Иначе необходимо использовать точный тест Фишера

Точный тест Фишера

	Исход есть	Исхода нет	Всего
Фактор есть	A	B	A+ B
Фактора нет	C	D	C + D
Всего	A + C	B + D	A + B + C + D

$$p(table) = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}$$

В чем проблема:?

Точный тест Фишера

	Исход есть	Исхода нет	Всего
Фактор есть	A	B	A+ B
Фактора нет	C	D	C + D
Всего	A + C	B + D	A + B + C + D

$$p(table) = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}$$

В чем проблема:?

Мы получили точечную оценку. Для получения p-value нам надо посчитать весь хвост (односторонний тест) или оба хвоста (двусторонний тест)

Точный тест Фишера

Левый хвост, сложить вероятности всех таблиц здесь

Все хорошо

Правый хвост, сложить вероятности всех таблиц здесь

Таблица, перекошенная, как наша, но в другую сторону

Наша таблица

Таблицы с еще более перекошенной в другую сторону связью

	Исход есть	Исход а нет	Всего
Фактор есть	A	B	A+ B
Фактора нет	C	D	C + D
Всего	A + C	B + D	A + B + C + D

	Исход есть	Исход а нет	Всего
Фактор есть	A	B	A+ B
Фактора нет	C	D	C + D
Всего	A + C	B + D	A + B + C + D

Таблицы с еще более перекошенной в нашу сторону связью

Пример

	Юноши	Девушки	Всего
На диете	1	9	10
Без диеты	11	3	14
Всего	12	12	24

Гипотеза H0: Юноши и девушки сидят на диетах одинаково
Гипотеза H1: Девушки сидят на диетах чае

$$p(table) = ?$$

Пример

Для вычисление p-value нам надо посчитать еще все таблицы, которые критичнее нашей, в данном случае она одна..

	Юноши	Девушки	Всего
На диете	0	10	10
Без диеты	12	2	14
Всего	12	12	24

$$p(table_1) = ?$$

$$Pvalue = ?$$

fisher.test

Fisher's Exact Test for Count Data

Description

Performs Fisher's exact test for testing the null of independence of rows and columns in a contingency table with fixed marginals.

Usage

```
fisher.test(x, y = NULL, workspace = 200000, hybrid = FALSE,
            hybridPars = c(expect = 5, percent = 80, Emin = 1),
            control = list(), or = 1, alternative = "two.sided",
            conf.int = TRUE, conf.level = 0.95,
            simulate.p.value = FALSE, B = 2000)
```

fisher.test

```
cont_mat <- rbind(c(1, 9 ), c(11, 3))
fisher.test(cont_mat)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  cont_mat
## p-value = 0.002759
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0006438284 0.4258840381
## sample estimates:
## odds ratio
## 0.03723312
```

Домашнее задание*

Написать функцию chisq.goodness_test(x, distribution=“uniform”)

Данная функция должна осуществлять тест на goodness of fit для вектора x для распределений:

1) uniform (равномерное)

2) pois (Пуассона)

3) norm (Нормальное)

И возвращать объект S3-класса **good_test**, который красиво печатается (выдает информацию тесте, какое распределение использовалось и какой p-value получился).