

Домашнее задание 4

Файл с выполненным заданием необходимо загрузить на Dropbox до дедлайна, указанного на сайте.

Домашние задания, сданные после срока, оцениваются с использованием понижающих коэффициентов: опоздание в пределах часа – штраф 10% от полученной оценки, в пределах суток – штраф 20%, в пределах недели – штраф 50%. Домашние задания, сданные через неделю после указанного срока и позже, не принимаются и не оцениваются.

Если при проверке работ установлен факт нарушения академической этики, студент получает оценку «0» за данную работу. Работа студента, предоставившего свою работу для списывания, также аннулируется.

1. Скачайте с сайта файл `hw4-template.Rmd`.
2. Впишите в этот файл решения задач в ячейки с кодом между строками `###BEGIN YOUR CODE` и `###END YOUR CODE`.
3. Выполните действия для запуска автоматических тестов (см. ниже), свяжите Rmd-файл в html-файл и проверьте, что тесты пройдены. Загрузите итоговый Rmd-файл на Dropbox.

Автоматические тесты

Большинство задач в домашнем задании по R предполагают автоматическое тестирование решений. Один тест представляет собой блок кода, который возвращает значение TRUE, если тест пройден, и значение FALSE, если тест не пройден. Если какие-то тесты не пройдены, выводится ошибка вида `[something] не TRUE`. Перед каждым тестом приводятся пояснения, которые помогают понять, что именно не так с решением.

1. Если у вас **успешно** установились библиотеки `devtools` и `testrmd`:

```
install.packages("devtools")
devtools::install_github("ropenscilabs/testrmd")
```

можете убрать опцию `eval=FALSE` в ячейке ниже, связывать Rmd-файл в html и смотреть на выполнение тестов. При связывании текущего Rmd-файла в готовом html-файле отображаются включения с информацией о пройденных тестах. Если какие-то тесты не пройдены, рядом с блоком с решением задачи появляется красная кнопка, а в начале файла появляется предупреждение вида `Warning! This document contains N failing tests`. Можно кликнуть на красную кнопку и посмотреть, какие тесты не пройдены и почему.

```
testrmd::init()
```

2. Если библиотеки `devtools` и `testrmd` **не установились**, запускайте строки в ячейке с тестами как обычный код и проверяйте, пройден ли тест. Если после запуска ячейки с тестом не выводится сообщение об ошибке, тест пройден.

Задача 1

Два события A и B являются независимыми, если выполняется условие $P(A \cap B) = P(A) \times P(B)$. Напишите код, который запрашивает у пользователя с клавиатуры вероятности событий A и B и вероятность их пересечения (одна вероятность — один ввод) и проверяет, являются ли события A и B независимыми, то есть сохраняет и выводит на экран логическое значение TRUE или FALSE.

Пример 1

```
Enter P(A): 0.4
Enter P(B): 0.5
```

```
Enter P(A and B): 0.2
[1] TRUE
```

Пример 2

```
Enter P(A): 0.3
Enter P(B): 0.1
Enter P(A and B): 0.2
[1] FALSE
```

Задача 2

Напишите код, который запрашивает у пользователя с клавиатуры значения дискретной случайной величины через пробел, потом запрашивает соответствующие значения вероятности через пробел, и выводит на экран математическое ожидание и дисперсию этой величины.

Пример

```
Enter values: -1 0 3
Enter probabilities: 0.4 0.1 0.5
E(X) = 1.1 and D(X) = 3.69
```

Напоминание. Математическое ожидание — ожидаемое среднее значение случайной величины — вычисляется следующим образом:

$$E(X) = \sum_{i=1}^n x_i p_i = x_1 p_1 + x_2 p_2 + \dots + x_n p_n.$$

Другими словами, перемножаем значения и соответствующие им вероятности, а потом все суммируем).

Дисперсия — показатель разброса значений случайной величины относительно ожидаемого среднего значения — вычисляется следующим образом:

$$D(X) = E(X^2) - [E(X)]^2.$$

###Задача 3

Число действительных бюллетеней на избирательных участках некоторой территориальной избирательной комиссии задается вектором `valid_votes`, а число недействительных бюллетеней — вектором `invalid_votes`. Общее число зарегистрированных избирателей задается вектором `voters`.

```
valid_votes <- c(25, 8, 12, 18, 25, 32, 10, 17, 22, 17)
invalid_votes <- c(1, 0, 4, 3, 9, 2, 0, 5, 1, 0)
voters <- c(50, 75, 62, 54, 98, 55, 72, 80, 44, 48)
```

Создайте вектор `turnout`, который содержит значения явки на избирательных участках, выраженные в процентах и округленные до *второго* знака после запятой.

Напоминание. Явка определяется как доля суммы действительных и недействительных бюллетеней от общего числа зарегистрированных избирателей.

Задача 4

Вектор `ages` представляет собой набор значений возраста респондентов мужского пола, принимавших участие в опросе.

```
ages <- c(23, 18, 19, 33, 36, 27, 68, 62, 78, 45, 42, 38, 55, 16, 14, 17, 92)
```

1. Выберите из вектора `ages` значения, которые соответствуют трудоспособному возрасту (в России для мужчин — с 16 до 59 лет включительно), и сохраните их в вектор `work_age`. Сколько людей такого возраста среди наших респондентов? Сохраните ответ на вопрос в переменную `N1`.
2. Выберите из вектора `ages` значения, которые соответствуют возрасту, младше трудоспособного (в России — менее 16 лет), и сохраните их в вектор `young_age`. Сколько людей такого возраста среди наших респондентов? Сохраните ответ на вопрос в переменную `N1`.

Задача 5

Политолог Мебейн (Walter R. Mebane) считает, что большая доля избирательных участков со значениями явки, заканчивающихся на 0 или 5, свидетельствует о фальсификациях результатов выборов. Аргументирует он это чисто психологическими причинами: если значения явки сочиняют люди, то они более склонны записывать круглые числа и числа, кратные 5.

Перед вами вектор значений явки на избирательных участках в районе F страны Флатландии:

```
Fturnout <- c(100, 124, 121, 130, 150, 155, 144, 132, 189, 145, 125, 110, 118, 129, 127)
```

Сохраните в вектор `s` индексы избирательных участков, где явка, согласно Мебейну, выглядит подозрительной.

Задача 6

Известно, что в таблице содержатся показатели по Ивановской области за 8 лет (с 2008 по 2015 включительно). В Ивановской области 21 муниципальный район. Создайте три вектора, которые могут быть использованы в качестве столбцов `region` (название области), `district_id` (номер района) и `year` (год):

- а) считая, что сначала идут значения показателей по одному району за 8 лет, потом — по второму за 8 лет и так далее;
- б) считая, что сначала идут значения показателей по одному году по всем 21 районам, потом — по второму по 21 районам и так далее.

Задача 7

Создайте матрицу `M`, которая содержит данные по четырем городам (город, численность населения, площадь):

```
Москва, 12 615 279 человек, 2561.5 кв. км  
Санкт-Петербург, 5 383 890 человек, 1439 кв. км  
Омск, 1 164 815 человек, 566.9 кв. км  
Екатеринбург, 1 483 119 человек, 468 кв. км
```

Обратите внимание: в матрице обязательно должны присутствовать названия городов, но при этом числовые данные должны иметь тип `numeric`.

Задача 8

Напишите код, который принимает на вход числовую матрицу `Nums` (дана в коде для примера) и сохраняет в переменную `gmax` индекс строки, имеющей наибольшую сумму элементов. Если таких строк несколько, то переменная `gmax` будет представлять собой вектор и хранить несколько индексов.

Ваш код должен корректно работать для матрицы любой размерности. В задаче нельзя использовать условные конструкции и циклы.

Подсказка. Для нахождения суммы по каждой строке матрицы в R есть готовая функция `rowSums()`.

Задача 9

Дан список `Q` с характеристиками мячей для игры в квиддич. Напишите код, который добавит в этот список данные по снитчу:

- name: snitch,
- points: 150,
- who_is_after: seeker.