

Федеральное государственное образовательное бюджетное учреждение  
высшего профессионального образования  
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ  
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»  
(Финансовый университет)

---

Департамент анализа данных, принятия решений  
и финансовых технологий

Дисциплина «Программирование в среде R»

П.Б. Лукьянов

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ  
КОНТРОЛЬНОЙ РАБОТЫ № 1

**Расчет показателей эффективности торговой сети**  
**Часть 1**

Для студентов, обучающихся по направлению подготовки  
«Прикладная информатика»  
(программа подготовки бакалавра)

Москва 2020

Целью выполнения контрольной работы является знакомство с бизнес-процессами реализации товара в торговой сети и написание программы по расчету показателей эффективности работы ее магазинов. В процессе написания программы студент приобретает практические навыки использования языка R для решения прикладной экономической задачи.

### **Постановка задачи**

Задача взята из реальной жизни и максимально упрощена.

**Дано.** Имеется торговая сеть из десяти магазинов. Во все магазины поставляется и продается только один скоропортящийся товар (например, натуральное молоко, или выберите свой товар). В конце каждого дня магазин заказывает на ферме некоторое количество контейнеров с товаром для продажи на следующий день. В течение дня товар частично или полностью продается. Нераспроданный товар списывается и утилизируется.

В конце недели служба доставки формирует отчеты по объемам поставок товара в магазины. Каждый магазин в конце недели готовит отчет о количестве проданного товара. Отчеты представляют собой текстовые файлы из двух столбцов: первый столбец – номер дня недели (1-7), второй столбец – объем поставки товара по дням или объем продаж по дням. Значения в строках отделяются друг от друга пробелом или знаком табуляции.

Магазины закупают товар по оптовой цене 5500 руб. за контейнер. Выручка магазина от продажи одного контейнера составляет 8000 руб., потери от списания и утилизации не проданного контейнера составляют 400 руб.

С целью упростить постановку задачи и программирование, не учитываются постоянные затраты (оплата труда, аренда площадей и т.д.), налоги и отчисления.

**Требуется.** Требуется рассчитать итоговую выручку и прибыль за неделю и выручку и прибыль по дням для каждого магазина, а также

определить лучший и худший магазины по следующим итоговым показателям:

- выручка за неделю
- прибыль за неделю
- объем реализации товара
- объем списания товара
- равномерность продаж в течение недели
- максимальный объем продаж за день
- минимальный объем продаж за день

Также требуется определить дни максимальных и минимальных продаж по всем магазинам, подсчитать суммарные и средние выручку и прибыль сети, потери из-за списания, определить усредненный показатель равномерности продаж. Для удобства анализа показателей расчетные значения нужно представить в табличном виде. Один из вариантов представления показан на рис. 1.

	Показатели за неделю					Показатели за день					
	Выручка, руб	Прибыль, руб	Реализация, конт.	Списание, конт.	Равномерность продаж	Продажи макс	День	Продажи мин	День	Списание макс	День
Магазин 1											
Магазин 2											
Магазин 3											
Магазин 4											
Магазин 5											
Магазин 6											
Магазин 7											
Магазин 8											
Магазин 9											
Магазин 10											
Итого											
Среднее											

Рис. 1. Итоговая таблица показателей работы магазинов сети

**Решение задачи** разбивается на несколько этапов:

1. Подготовка исходных данных по поставкам и продажам
2. Копирование файлов с исходными данными в папку выполнения расчета
3. Считывание исходных данных из файлов в переменные среды R
4. Создание пустой таблицы для итоговых значений
5. Выполнение расчетов по каждому магазину, заполнение итоговой таблицы
6. Сохранение таблицы с результатами в файл
7. Вызов программы Excel и выполнение графического анализа рассчитанных показателей.

### **Шаг 1. Подготовка исходных данных по поставкам и продажам**

Придумать название своей торговой сети. На локальном или сетевом диске создать папку с фамилией и именем студента. В этой папке создать папку с названием своей торговой сети. В папке сети создать десять папок с названиями **Магазин1**, ..., **Магазин10** для хранения в этих папках исходных данных, создать одну папку **Анализ** для хранения результатов. В каждой папке (Магазин1, ..., Магазин10) создать по два текстовых файла in.txt и out.txt с данными о поставках и продажах соответственно. Файлы заполнить произвольными целочисленными значениями (см. рис. 1).

День	Поставка	День	Продажа
1	52	1	43
2	43	2	43
3	40	3	36
4	48	4	43
5	40	5	40
6	53	6	51
7	55	7	46

Рис. 1. Пример файлов in.txt, out.txt с исходными данными

При выполнении Шага 1 пользоваться средствами операционной системы и любым текстовым редактором, файл сохранять в кодировке UTF-8. Язык R не использовать.

Важно, чтобы при выполнении шага 1 названия некоторых папок были на русском языке. В реальных условиях на компьютерах в организациях, как правило, установлены русскоязычные операционные системы, и при использовании различных программ автоматизации данные сбрасываются в папки на русском языке. Также важно, чтобы заголовки таблиц в файлах `in.txt` и `out.txt` были на русском языке. Наличие национальных символов в именах папок и в заголовках таблиц позволит лучше изучить использование кодировок при обработке таких файлов.

## **Шаг 2. Копирование файлов с исходными данными в папку выполнения расчета**

При выполнении Шага 2 язык R не использовать.

Скопировать файлы с данными из папок магазинов в папку Анализ. При копировании имена файлов `in.txt`, `out.txt` из папок Магазин1 и т.д. заменять на имена `store1_in.txt`, `store1_out.txt` и т.д. в папке Анализ. Для копирования файлов перейти в режим командной строки операционной системы и использовать соответствующие команды.

Для упрощения работы с командами операционной системы группу команд сохранить в отдельный текстовый файл с расширением `bat` (`batch` – папка) и использовать вызов этого `bat`-файла один раз для выполнения всех записанных в нем команд (для ОС Windows). Правила создания `bat`-файлов изучить самостоятельно. Для корректной обработки пути к файлам с использованием русских букв сохранять `bat`-файл в кодировке 866 (OEM-русская).

Запуск бат-файла можно выполнить и в оболочке PowerShell (для ОС Windows), для чего использовать следующий синтаксис:

start example.bat

### Шаг 3. Считывание исходных данных в переменные среды R

Все действия далее выполнять в скрипте на языке R.

В среде R перейти в папку Result, где находятся все файлы с исходными данными. Из среды R убедиться в наличии этих файлов, используя функцию `dir()`. В языке имеется несколько функций для работы с ОС, которые нужно использовать при написании скрипта:

`getwd()` – возвращает текущую рабочую папку (get working directory – узнать рабочую директорию)

`setwd()` – выполняет переход в заданную папку (set working directory – перейти в рабочую директорию)

`dir()` – выводит содержимое текущей папки (директории)

Для выполнения расчетов данные из текстовых файлов скопировать в переменные R. В языке R для считывания таблиц с данными из текстовых файлов предназначена функция **`read.table()`**. При вызове функции нужно определить некоторые параметры, управляющие считыванием данных из файла (см. табл. 1).

Таблица 1. Некоторые параметры функции `read.table()`

Имя	Значение
file	Строка с именем файла с данными. В строке может быть указан абсолютный или относительный путь к файлу, например, <code>file="c:/lessons.R/dixy/result/store1_in.txt"</code> , или только имя файла, например, <code>file="store1_out.txt"</code> . Если указывается только имя файла, до вызова функции <code>read.table()</code> должен быть выполнен переход в папку с нужным файлом с помощью <code>setwd()</code> . В строке с именем можно указывать URL-ссылку на файл, если он размещен в Internet.
header	В таблице данных первой строкой может идти строка с заголовками (именами столбцов). В этом случае ее нужно считать так, чтобы имена столбцов файла стали именами

	столбцов таблицы R. Если заголовки есть, нужно выставить <code>header</code> в <code>TRUE</code> , иначе в <code>FALSE</code> . Значение по умолчанию <code>FALSE</code> .
<code>row.names</code>	В файле данных может быть столбец, в котором содержатся имена строк таблицы. Чтобы при считывании данных скопировать имена строк из файла в имена строк таблицы R нужно указать номер этого столбца, например, <code>row.names = 1</code> . Имена строк должны быть уникальными.
<code>sep</code>	от <code>separator</code> – разделитель. Параметр определяет, как отделяются друг от друга значения в строках. По умолчанию считается, что разделитель – пробел или табуляция. Если разделитель – точка с запятой, пишут <code>sep = “;”</code>
<code>dec</code>	От <code>decimal</code> – десятичный. Параметр определяет, какой символ используется для отделения целой части от десятичной. По умолчанию <code>dec=”.”</code>
<code>nrows</code>	Целое число, которое задает, сколько строк должно быть считано из таблицы
<code>skip</code>	Целое число, которое указывает, сколько строк с начала таблицы нужно пропустить перед началом считывания данных
<code>encoding</code>	Строка, задающая кодировку, в соответствии с которой будут преобразовываться коды символов из считываемого файла в символы таблицы.

Пример. Выполним копирование данных из таблиц:

```
in1 <- read.table(store1_in.txt, head = TRUE)
out1 <- read.table(store1_out.txt, head = TRUE)
....
in10 <- read.table(store10_in.txt, head = TRUE)
out10 <- read.table(store10_out.txt, head = TRUE)
```

Теперь исходные таблицы хранятся в переменных `in1`, `out1` и т.д. Как проверить, что находится в переменных `in1`, `out1`? Какого типа эти переменные? К какому классу принадлежат?

## Проверить и в случае необходимости исправить заголовки таблиц R.

В случае, если строковые поля или заголовки скопированной таблицы в R отображаются не корректно, при считывании таблицы нужно принудительно задать кодировку, в соответствии с которой будет обрабатываться строковые поля копируемой таблицы. В функции `read.table()` нужно задать значение параметра `encoding`, соответствующее той кодировке, в которой был сохранен файл с данными. Рекомендуется использовать кодировку UTF-8.

Если заголовки столбцов нуждаются в корректировке, существует несколько способов заменить имя столбца. Используются функции `names()` или `colnames()`:

```
colnames(in1)[1] <- "День недели" или  
names(in1)[1] <- "День недели"
```

Что не правильно в таком способе изменения имени столбца? Необходимо явно задавать номер столбца, но этот номер в сложной таблице может быть не известен. Рекомендуется использовать более сложную конструкцию:

```
names(in1)[names(in1) == "День"] <- "День недели" или  
colnames(in1)[which(names(in1) == "День")] <- "День недели" или  
colnames(in1)[which(colnames(in1) == "День")] <- "День недели"
```

Логика в этих примерах следующая: функции `names(in1)` или `colnames(in1)` считывают все имена столбцов в `in1` в массив. По заданному условию на соответствие имени столбца нужному значению мы находим нужный столбец и присваиваем ему новое имя.

## Шаг 4. Создание пустой итоговой таблицы

Для сохранения рассчитанных значений нужно подготовить пустую сводную таблицу с показателями всех магазинов, аналогичную таблице из рис. 1. Для создания таблицы используется функция **`data.frame()`**, которая



склеивает друг с другом вектора одинакового размера, интерпретируя их как столбцы. Следовательно, до вызова `data.frame()` нужно подготовить соответствующие столбцы данных. Первый столбец – переменная со значениями выручки по всем магазинам плюс 2 значения для суммы и среднего. Создадим первый столбец:

```
rev <- rep(0, 12)
```

Второй столбец создадим, используя значение размера вектора `rev`:

```
profit <- rep(0, length(rev))
```

Объединим два столбца в таблицу, сразу пропишем заголовки столбцам таблицы:

```
res.tab <- data.frame(Выручка = rev, “Прибыль” = profit)
```

Разберитесь, в каком случае кавычки можно не ставить, а в каком случае они необходимы.

Количество столбцов и строк в таблице можно узнать, используя функции `ncol()` и `nrow()`.

Создадим вектор для третьего столбца (переменную `sale`) следующим образом:

```
sale <- rep(0, nrow(res.tab))
```

К столбцу таблицы можно обращаться через конструкцию `ИмяТаблицы$ИмяСтолбца`. Добавим новый столбец к таблице следующим образом:

```
res.tab$Реализация <- sale
```

`res.tab$Реализация` создает для `res.tab` новый столбец с именем “Реализация”, знак `$` используется в качестве разделителя.

Как обратиться к конкретному элементу столбца? Через его индекс:

```
res.tab$Реализация[10] <- 155
```

Аналогичным образом саму таблицу можно рассматривать как матрицу или двумерный массив и обращаться к элементам матрицы через

задание соответствующих индексов, где первый индекс – номер строки, второй – номер столбца.

Присвоим значение 17 шестой строке, второму столбцу:

```
res.tab[6,2] <- 17
```

Чтобы обратиться ко всей четвертой строке, напишем:

```
res.tab[4, ] <- 44
```

Для задания значения всему второму столбцу делаем аналогично:

```
res.tab[, 2] <- 59
```

Ко столбцу можно обратиться проще:

```
res.tab[2] <- 62
```

Добавить новый столбец можно следующим образом

```
res.tab[3]<-0
```

но у такого столбца названия не будет. Как его задать?

Ниже – еще один способ создать столбец:

```
res.tab[ , “Реализация”] <- sale
```

или проще:

```
res.tab[ “Реализация”] <- sale
```

Для оставшихся столбцов таблицы не обязательно сначала создавать вектора, чтобы потом добавлять их таблице. Нужно объявить новый столбец и сразу его проинициализировать:

```
res.tab$“Списание, конт.” <- 0
```

Самостоятельно заполнить таблицу недостающими столбцами.

Самостоятельно задать имена строкам таблицы. Разобрать и проверить способы доступа к строкам таблицы через имена строк. Как добавить новую строку в таблицу?

Кроме вышеприведенных способов добавлять столбцы и строки, есть другой способ, заключающийся в использовании функций `cbind()` для добавления столбцов и `rbind()` для добавления строк. Самостоятельно

рассмотреть способы добавления строк и столбцов в `res.tab` с помощью данных функций.

Самостоятельно рассмотреть способы удаления лишних строк и столбцов.

### **О показателе «Равномерность продаж»**

Разберем смысл показателя «Равномерность продаж». Для чего требуется определять равномерность продаж магазина в течение некоторого периода?

Руководитель магазина заинтересован в информации о динамике продаж товаров по дням. Эти данные помогают планировать объемы поставок товара для будущей реализации и позволяют максимизировать будущие продажи, увеличить выручку и прибыль. Также показатель равномерности продаж товара позволяет сокращать потери от списания нереализованной продукции.

Каким численным значением оценить равномерность продаж?

В математике и статистике равномерность набора данных или, другими словами, меру разброса данных оценивают по значению среднеквадратичного отклонения (см. формулу на рис. 2), которое является квадратным корнем из дисперсии. Дисперсия, в свою очередь, является рассчитываемой величиной и характеризует меру разброса некоторого набора данных от их среднего значения.

$$S = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$$

Рис. 2. Формула для расчета среднеквадратического отклонения

В языке R имеется специальная функция `sd()` для расчета среднеквадратического отклонения. Параметр этой функции – вектор,

состоящий из набора данных, по которым нужно определить отклонение. В нашем случае это вектор объема продаж по дням. Нужно переименовать в таблице столбец «Равномерность продаж» в столбец с именем «sd» и заполнять столбец значениями среднеквадратичного отклонения.