

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

---

Департамент Математики

Дисциплина «Программирование в среде R»

П.Б. Лукьянов

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ  
КОНТРОЛЬНОЙ РАБОТЫ № 1

**Расчет показателей эффективности торговой сети**  
**Часть 1. Постановка задачи, подготовка и обработка данных**

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

Москва 2021

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 приобретаются 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  $N$  ( $N=10$ ) магазинов. Во все магазины поставляется и продается один скоропортящийся товар (например, натуральное молоко, или выберите свой товар). В конце каждого дня магазин заказывает у Поставщика некоторое количество контейнеров с товаром для продажи на следующий день. В течение дня товар частично или полностью продается. Нераспроданный товар списывается и утилизируется, магазин оплачивает утилизацию списанного товара.

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

Магазины закупают товар по оптовой цене  $P_{\text{supply}} = 5500$  руб. за контейнер. Цена, по которой магазин продает товар, составляет  $P_{\text{sale}} = 8000$  руб. за контейнер. Потери от списания и утилизации не проданного контейнера составляют  $P_{\text{util}} = 400$  руб.

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

**Требуется.** Для каждого магазина требуется рассчитать динамику продаж и прибыли по дням периода, а также вычислить итоговую выручку и прибыль; определить лучший и худший магазины по следующим итоговым показателям:

- выручка за период Т
- прибыль за период Т
- 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 представить в табличном виде. Один из вариантов таблицы 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
8. Создание собственной функции генерации исходных данных для одного товара.
9. Создание собственной функции генерации исходных данных для нескольких товаров.
10. Изменение основной программы для расчетов по нескольким товарам.
11. Проверка работы программы на сторонних тестовых данных
12. Графический анализ средствами R
13. Имитационное моделирование планируемых экономических показателей на последующие периоды

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

Придумать название своей торговой сети, например **Микси**. 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 **Магазин 1**, ..., **Магазин 10** для хранения в этих папках исходных данных, создать одну папку **Анализ** для хранения результатов. В каждой папке (Магазин 1, ..., Магазин 10) создать по два

текстовых файла **Микси.in** и **Микси.out** с данными о поставках и продажах соответственно. Файлы заполнить произвольными целочисленными значениями (см. рис. 1).

Важно, чтобы при выполнении Шага 1 названия 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 и файлы на русском языке.

Также важно, чтобы заголовки таблиц в файлах данных были на русском языке. Наличие национальных символов в именах папок и в заголовках таблиц позволит лучше изучить использование кодировок при обработке таких файлов.

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 с исходными данными

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

Альтернативный вариант выполнения Шага 1 = написать функцию `generate.data()` на языке R для генерации файлов поставки и продажи (см. Шаг 8). Проверять, что величина продаж в любой из дней не превышает величину поставки товара.

## Шаг 2. Копирование файлов с данными в папку расчета

Скопировать файлы с данными из папок магазинов в папку **Анализ**. При копировании файлов в папку Анализ имена файлов Микси.in, Микси.out из папок Магазин 1, Магазин 2 и т.д. заменять на имена Магазин1\_Микси.in, Магазин1\_Микси.out и т.д. Для копирования файлов перейти в режим командной строки операционной системы и выполнить соответствующие консольные команды.

Для автоматизации работы по копированию файлов поставок и продаж команды копирования сохранить в отдельный текстовый файл с расширением bat (для ОС Windows) и использовать вызов этого bat-файла один раз для выполнения всех записанных в нем команд. Запуск бат-файла можно выполнить и в оболочке PowerShell, для чего использовать следующий синтаксис:

```
start example.bat
```

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

Все действия далее выполняются в скрипте R.

В среде R перейти в папку Анализ, где находятся файлы с исходными данными. Из среды R увидеть эти файлы, используя функцию `dir()`. При написании скрипта использовать следующие функции:

`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 параметры, управляющие считыванием данных из файла (см. Табл. 1).

Таблица 1. Основные 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, file="c:/lessons.R/Микси/Анализ/Магазин1_Микси.in", или только имя файла, например, file="Магазин1_Микси.in". Если указывается только имя файла, до вызова функции read.table() должен быть выполнен переход в папку с нужным файлом с помощью setwd(). 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 header в TRUE, иначе в FALSE. Значение по умолчанию FALSE.
row.names	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, row.names = 1. Имена строк должны быть уникальными.
sep	от separator – 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 sep = “;”
dec	Параметр определяет, какой символ используется для отделения целой части от десятичной. По умолчанию dec=“.”
nrows	Целое число, которое задает, сколько строк должно быть считано из таблицы
skip	Целое число, которое указывает, сколько строк с начала таблицы нужно пропустить перед началом считывания данных

encoding	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(Магазин1_Микси.in, head = TRUE)
out1 <- read.table(Магазин1_Микси.out, head = TRUE)
....
in10 <- read.table(Магазин10_ Микси.in, head = TRUE)
out10 <- read.table(Магазин10_ Микси.out, 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**

При считывании таблицы нужно принудительно задавать кодировку, в соответствии с которой будут обрабатываться строковые поля копируемой таблицы. В функции read.table() задать значение параметра encoding, соответствующее той кодировке, в которой был сохранен файл с данными. Рекомендуется использовать кодировку UTF-8.

Переименовать заголовок столбца «День» на «День недели». Использовать функции names() или colnames():

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

Такой способ не слишком надежен, т.к. необходимо явно задавать номер столбца, а в большой таблице этот номер 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 `in1` в массив. По заданному условию на соответствие имени столбца нужному значению находится нужный столбец, ему присваивается новое имя.

В случае, если заголовки скопированной таблицы в R отображаются не корректно, для исправления заголовков также использовать функции `names()` и `colnames()`.

#### Шаг 4. Создание пустой итоговой таблицы

Для сохранения рассчитанных значений 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
```

К столбцу можно обратиться проще:

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

Добавить новый столбец можно следующим образом

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

У вновь созданного 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 таблицу. 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.

Как удалить строку? Как удалить столбец? Самостоятельно изучить 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?

Руководителя магазина интересует динамика продаж товаров по дням. Эти данные нужны для планирования объемов поставок товара для будущей реализации и позволяют оптимизировать 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 среднеквадратичного отклонения

В R имеется специальная функция `sd()` для расчета среднеквадратичного 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.

## Шаг 5. Выполнение расчетов

Определение прибыли. Прибыль  $Pr$  (profit) определяется по формуле:

$$Pr = TR - TC \quad (1)$$

где  $TR$  (total revenue) – выручка от продажи товара

$TC$  (total cost) – затраты

$$TR = Q_{\text{sale}} * P_{\text{sale}} \quad (2)$$

где  $Q_{\text{sale}}$  (quantity sale) – количество проданных контейнеров

$P_{\text{sale}}$  (price) – розничная цена контейнера

$$TC = Q_{\text{supply}} * P_{\text{supply}} + Q_{\text{util}} * P_{\text{util}}$$

где  $Q_{\text{supply}}$  – количество закупленных магазином контейнеров

$P_{\text{supply}}$  – цена закупки контейнера магазином

Q\_util – количество непроданных контейнеров

P\_util – цена утилизации контейнера

Для выполнения расчетов использовать следующие функции R:

sum() – суммирует значения аргументов

mean() – находит среднее значение аргументов

min() – возвращает минимум из массива переданных значений

max() – возвращает максимум из массива переданных значений

which.min() – возвращает индекс элемента массива с наименьшим значением

which.max() – возвращает индекс элемента массива с наибольшим значением

## Шаг 6. Сохранение таблицы с результатами в файл

После выполнения расчетов таблицу с результатами нужно сохранить в файл для дальнейшего анализа. Сохранить таблицу в текстовом файле, в качестве разделителя значений использовать точку с запятой. Задать файлу специальное расширение “csv” (comma-separated values – значения, разделенные запятыми). Подготовленный таким образом файл может быть открыт в виде таблицы программой Excel или аналогичной.

Для записи таблиц и матриц в текстовый файл предназначена функция write.table(). При вызове функции нужно задать некоторые параметры, управляющие записью (см. табл. 2).

Таблица 2. Некоторые параметры функции write.table()

Имя	Значение
x	Переменная языка R с таблицей данных (тип dataframe)
file	Строка с именем файла. В строке может быть указан абсолютный или относительный путь к файлу, например, file="c:/lessons.R/Микси/Анализ/resultTab.csv", или только имя файла. В строке с именем можно указывать URL-ссылку на файл, если файл должен быть размещен в Internet. Если имя

	файла не задано (file=''), таблица будет выведена в консоль. По умолчанию file=''
col.names	По умолчанию col.names=TRUE. Это означает, что в файл будет записана первая строчка с именами столбцов. В обратном случае (col.names=FALSE) имена записаны не будут.
row.names	По умолчанию row.names=TRUE. Это означает, что в таблицу будут записан первый столбец с именами строк. Если предполагается дальнейшее использование файла в программе Excel или других табличных редакторах, то наличие в файле имен строк приведет смещению заголовков столбцов, поэтому нужно задавать row.names=FALSE.
sep	Разделитель значений в строках таблицы. По умолчанию sep = ' '.
dec	Параметр определяет, какой символ используется для отделения целой части от десятичной. По умолчанию dec=".". В Excel десятичным разделителем является запятая, поэтому для открытия таблицы в Excel использовать dec = ','.

Файл с результатами создать в папке Анализ. После создания файла проверить его наличие средствами операционной системы и с помощью функции R file.exists(), открыть файл в любом редакторе, убедиться в правильности созданной таблицы.

## Шаг 7. Вызов программы Excel

В конце программы, после сохранения результатов, перейти в режим командной строки и запустить программу Excel с одновременным открытием файла с результатами. Написать bat-файл для быстрого запуска Excel и открытия файла данных. По данным из файла средствами Excel построить несколько графиков. Объяснить полученные результаты. Варианты графиков выбрать самостоятельно. Подготовить разумные подписи для осей и заголовков. Из подписей должно быть ясно, что это за график.

## Шаг 8. Генерация исходных данных на языке R

Для проверки различных вариантов хозяйственной деятельности магазина требуются различные наборы данных. Для решения этой задачи написать функцию **generate.data()** для генерации исходных данных по поставкам и продажам и использовать вызовы этой функции для получения данных по всем магазинам.

Функция должна создавать текстовый файл с двумя столбцами: день (месяца, недели, декады) и количество товара, полученное или проданное магазином, в единицах измерения товара. Количество поставляемого товара в разные дни может отличаться, поэтому при генерации значений поставок и продаж должен использоваться генератор случайных чисел.

У функции `generate.data()` должны быть следующие параметры:

- Имя файла без расширения
- Место размещения файла (может указываться абсолютный или относительный путь к файлу)
- Тип данных: поставка или продажа
- Минимальное значение поставки или продажи за день
- Максимальное значение поставки или продажи за день
- Количество дней, для которых создаются данные

Для каждого параметра задать значение по умолчанию, чтобы функцию можно было вызывать без указания параметров.

В зависимости от типа формируемых данных для создания файла с данными использовать расширения, заданные с помощью глобальных констант. Глобальные константы объявить в самом начале программы, до вызова функции:

```
EXT_SUPPLY <-'in'
```

```
EXT_SALE <-'out'
```

В случае, если путь к файлу не равен ‘’, проверять существование папки назначения с помощью функции `dir.exists()`. Если папка не

существует, пытаться создать ее с помощью функции `dir.create()`, после чего еще раз проверить существование папки.

Если задан тип данных «Продажа», искать в указанной папке файл с тем же именем, но с расширением `EXT_SUPPLY`, проверить его наличие с помощью функции `file.exists()`.

Если файл существует, считать его и при генерации данных по продажам товара учитывать значения поставок (продажи не могут превышать поставки). Если файла поставок нет, создать его, используя переданные параметры и расширение файла `EXT_SUPPLY`.

### **Шаг 9. Генерация исходных данных для нескольких товаров**

После того, как последовательность Шагов 1-8 отлажена, и программа выдает предсказуемые результаты по данным для одного товара, требуется увеличить количество товаров в файлах исходных данных.

На практике это означает, что в качестве входных параметров для функции `generate.data()` должно передаваться произвольное количество товаров, для каждого товара должен задаваться свой интервал поставок и продаж. Для передачи данных по товарам использовать структуру `list()`. Также через структуру `list()` передавать цены закупки, продажи, утилизации каждого товара.

Для удобства при работе с данными дополнительно формировать файл с ценами поставки, продажи, утилизации по каждому товару. На этапе обработки данных считывать цены из этого файла.

Для генерации файла с ценами написать функцию `generate.price()` и вызывать ее при генерации исходных данных из функции `generate.data()`.

### **Шаг 10. Выполнение расчетов для нескольких товаров**

По сформированным данным выполнить несколько видов анализа:

- по отдельному товару выбранного магазина



- по отдельному товару во всех магазинах
- по всем товарам выбранного магазина
- по всем товарам всей торговой сети с расчетом итоговых экономических показателей.

### **Шаг 11. Проверка работы программы на тестовых данных**

На этом шаге требуется получить от Преподавателя набор файлов с данными для анализа и выполнить расчет экономических показателей по этим данным.