Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

# ОТЧЁТ

по лабораторной работе №1

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ЯЗЫКА R ДЛЯ СТАТИСТИЧЕСКОГО

АНАЛИЗА ДАННЫХ

Выполнил:

ст. гр. ИС/б-21-2-о

Мовенко К. М.

Проверила:

Сырых О. А.

Севастополь

2024

# ЦЕЛЬ РАБОТЫ

Изучить основные особенности языка R. Исследовать возможности языка R для работы с графикой.

# ЗАДАНИЕ

* 1. Установить R на ПК;
  2. Установить RStudio – инсталлятор скачать с официального сайта проекта;
  3. Ознакомиться с кратким руководством пользователя RStudio;
  4. Исследовать команду demo(), полученные результаты вставить в отчёт;
  5. Исследовать основные функции и команды языка R, представленные в данной лабораторной работе, полученные результаты вставить в отчёт;
  6. Ответить на контрольные вопросы;

# ХОД РАБОТЫ

После установки всех необходимых компонентов был начат процесс исследования команд языка R. Первой была рассмотрена команда demo().

Команда demo() предназначена для запуска демонстрационных программ. С помощью ввода команды demo(); без параметров были выведены все доступные демонстрации (рисунок 1).

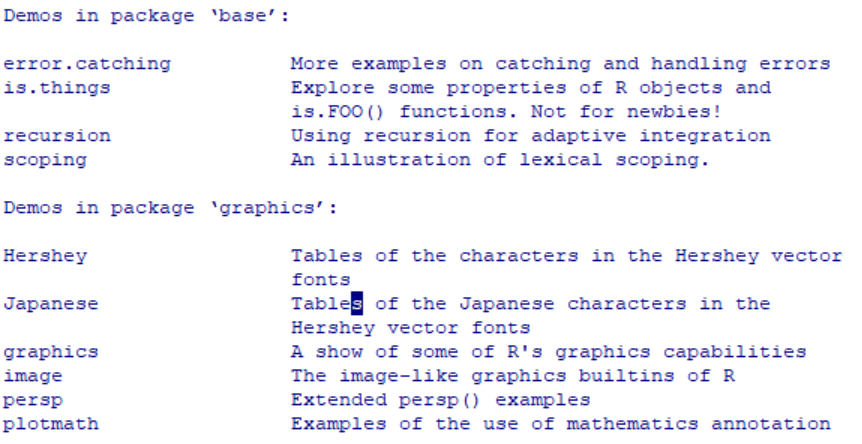


Рисунок 1 – Часть доступных демонстраций языка R

Например, в демонстрации demo(graphics) содержатся различные способы визуализации информации, поддерживаемые языком R, круговые диаграммы (рисунок 2), графики (рисунок 3) и пр.

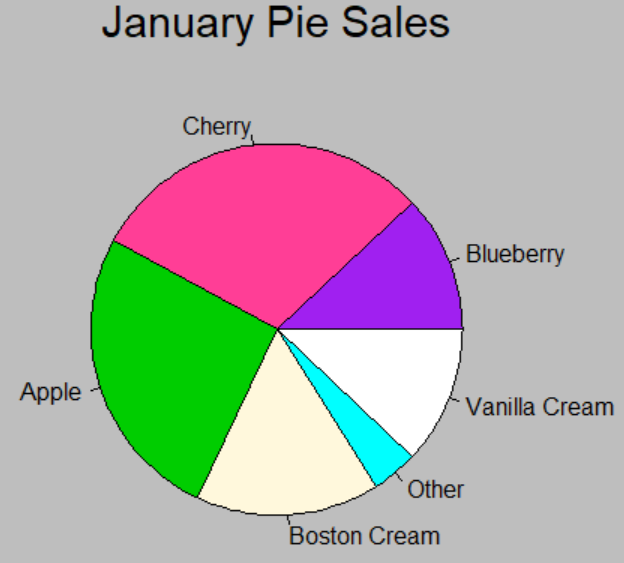


Рисунок 2 – Демонстрация применения круговой диаграммы

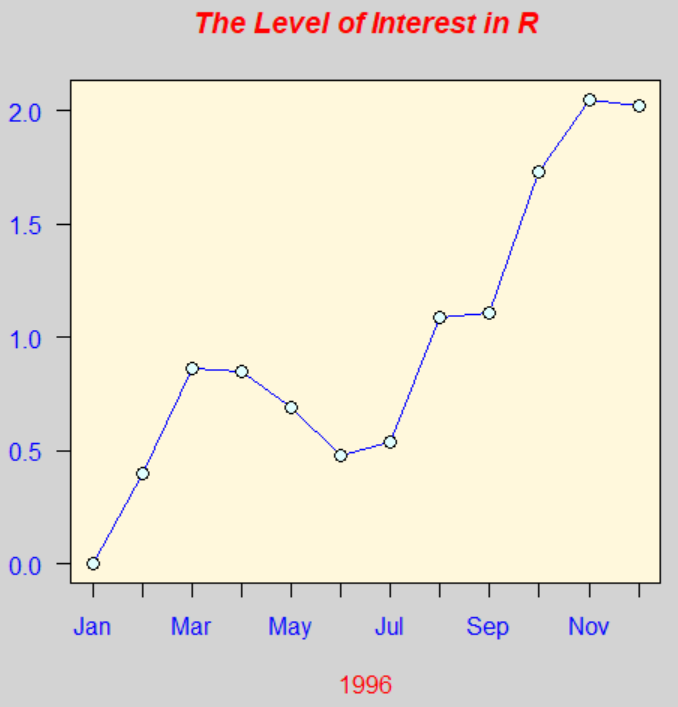


Рисунок 3 – Демонстрация применения линейного графика

В демонстрации demo(image) содержатся контурные графики с использованием цветовых шкал (рисунки 4-5).

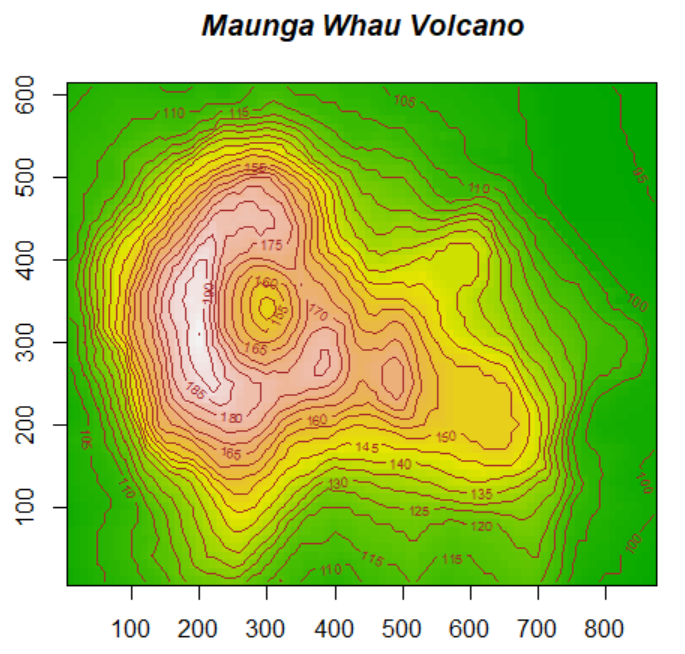


Рисунок 4 – Демонстрация demo(images)

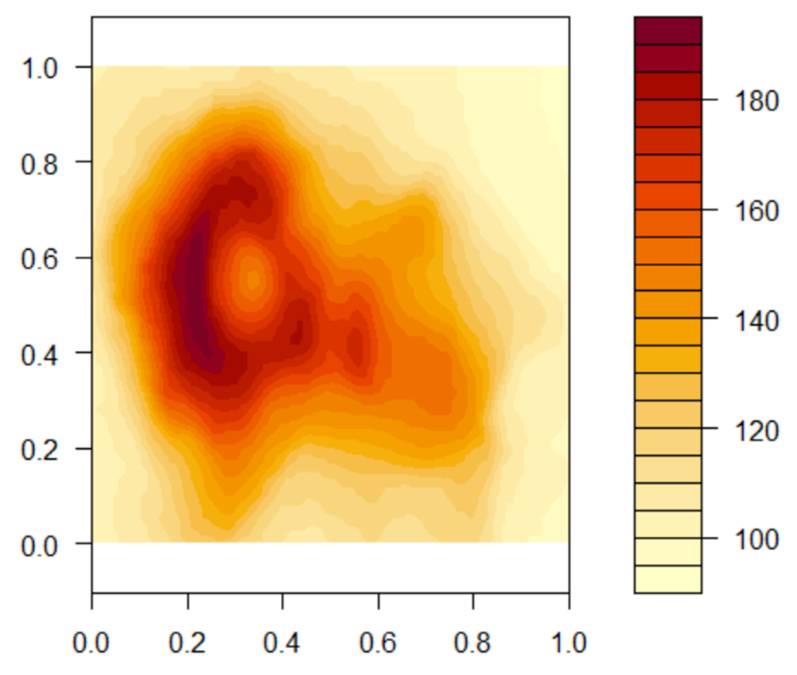


Рисунок 5 – Демонстрация demo(images)

В демонстрации demo(recursion) был открыт график с примером вычисления рекурсивной функции (рисунок 6). Также demo поддерживает демонстрации списка возможных символов, алфавитов, цветов и прочих компонентов языка R.

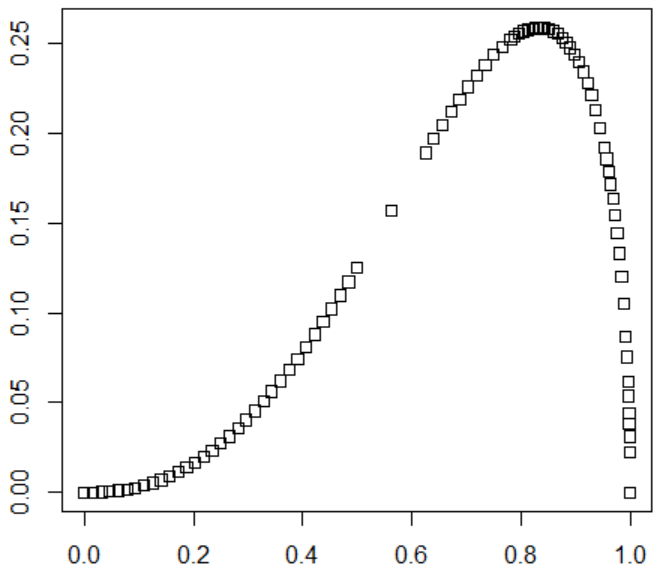


Рисунок 6 – Демонстрация рекурсивной функции

Далее были исследованы операторы присваивания (<-, ->, =) и способы их применения при арифметических операциях над переменными (рисунок 7).

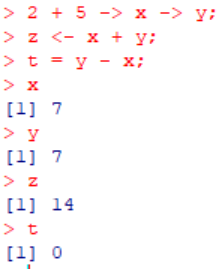


Рисунок 7 – Применение операторов присваивания

R поддерживает стандартную арифметику и позволяет использовать все возможности стандартного калькулятора. Так с помощью скобок можно менять порядок операций в выражении (рисунок 8).

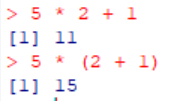


Рисунок 8 – Влияние скобок на результат операции

Для округления чисел существует команда round (рисунок 9). Она позволяет округлять числа до целых и десятичных разрядов, подробнее расписано в контрольных вопросах.

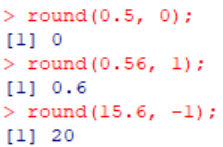


Рисунок 9 – Использование функции round

Отделить одну команду от другой помогает символ “;” или перевод на новую строку (рисунок 10).

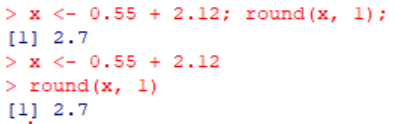


Рисунок 10 – Разделение команд символами

Помимо численных и строковых типов данных R поддерживает векторы – массивы однотипных объектов. Создать вектор можно с помощью функции c() прямым перечислением, регулярной последовательности («:» или seq()) или повторения rep() (рисунок 11).

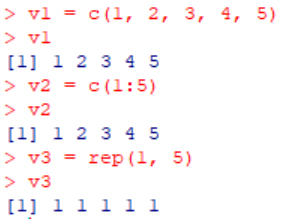


Рисунок 11 – Задание векторов

Над векторами можно проводить как простые арифметические операции (поэлементно), так и векторные, например, скалярное произведение (рисунок 12).

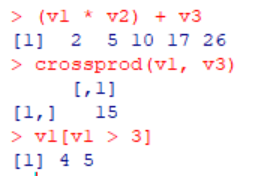


Рисунок 12 – Операции над векторами

У векторов есть свойства – длина (length), среднее значение (mean), дисперсия (var), значения которых можно получать с помощью специальных команд (рисунок 13).

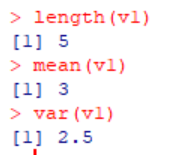


Рисунок 13 – Свойства вектора

Также в R можно создавать матрицы. Заполнять их можно по строкам или столбцам (рисунок 14), а также неопределёнными значениями.

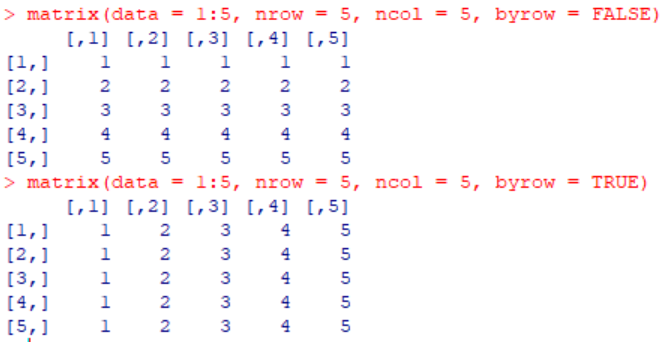


Рисунок 14 – Создание матриц

Создавать матрицы также можно путём объединения векторов по строкам (rbind) или столбцам (cbind) (рисунок 15).

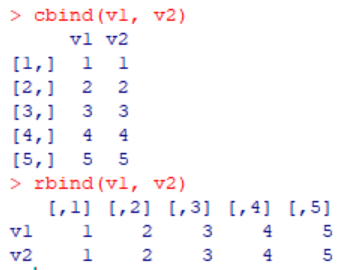


Рисунок 15 – Объединение векторов

С помощью специальных функций можно получить информацию о типе и размерности вектора или матрицы (рисунок 16).

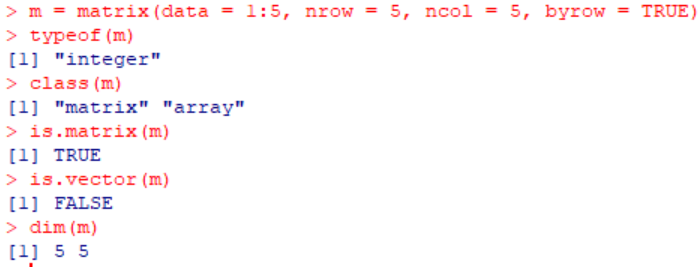


Рисунок 16 – Информация о матрице

Важной опцией языка R является построение графиков, осуществляемое командой plot(). В её аргументах можно задавать тип графика, значения, отображаемые на графике, цвет и подписи для осей.

x <- c(1, 2, 3, 4, 5)

y <- c(2, 4, 6, 8, 10)

plot(x, y, type = "l", main = "Пример графика", xlab = "X", ylab = "Y")



Рисунок 17 – Построение линейного графика

plot(sin, -pi, 2\*pi)

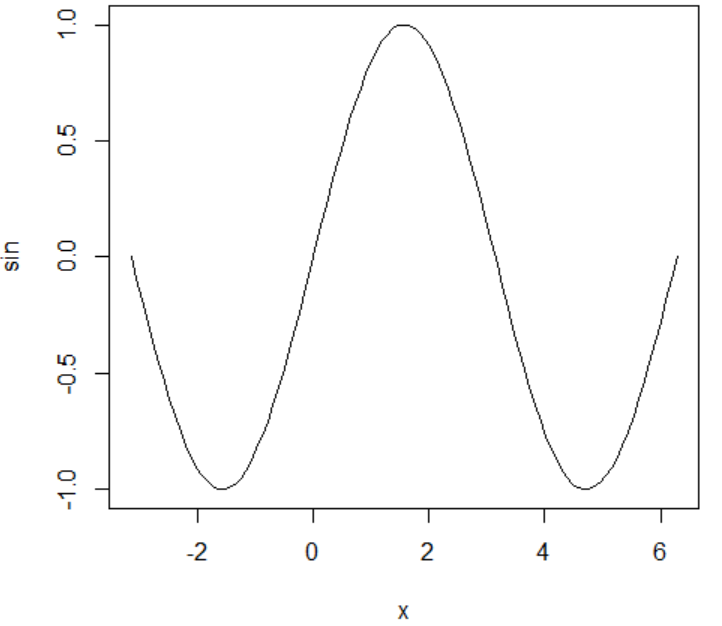


Рисунок 18 – Построение графика функции на интервале

# КОНТРОЛЬНЫЕ ВОПРОСЫ

* 1. Особенности языка R;

R одновременно является и языком, и программным обеспечением; его основные особенности:

* возможность эффективной обработки данных и удобные средства для сохранения результатов;
* может работать с различными парадигмами программирования;
* набор операторов для обработки массивов, матриц и других сложных конструкций;
* интерпретируемость (программа, написанная на нём, готова к исполнению сразу же, без необходимости компиляции в исполняемый файл);
* большая, последовательная, интегрированная коллекция инструментов для проведения статистического анализа;
* многочисленные графические средства для визуализации результатов анализа;
  1. Команда для получения подробной информации о функции в R;

Для получения подробной информации (справки) о любой функции необходимо выполнить команду help(<имя функции>) либо ?<имя функции>.

Если есть необходимость открыть справку через браузер, можно выполнить команду help.start().

Результат, выводимый командой help, содержит словесное описание функции, список её аргументов, примеры её использования и детали реализации.

* 1. Структура и особенности команды round() в R;

Команда round принимает на вход два числовых аргумента – x и digits.

Первый – вещественное число, которое необходимо округлить до некоторого значения. Второй – целое число, обозначающее количество цифр после запятой, до которых происходит округление. Если аргумент digits отрицательный, округление происходит до целых разрядов.

Полученные значения можно сохранять в переменные или просто выводить на экран. При округлении 0.5 округление происходит в меньшую сторону.

* 1. Команды для работы с векторами в R;

Помимо указанных команд для создания векторов (c(…), seq(…), rep(x, …)), получения их свойств (length, mean, var), а также арифметико-логических операций для векторов доступны:

* sum(…, na.rm=FALSE) – сумма элементов вектора;
* min(…, na.rm=FALSE) – минимальный элемент вектора;
* max(…, na.rm=FALSE) – максимальный элемент вектора;
* sort(x, decreasing, …) – сортировка вектора, если decreasing=FALSE − по возрастанию;
* rev(x) – изменить порядок элементов наоборот;

na.rm – параметр, указывающий нужно ли исключать из вектора пропущенные значения и NaN.

* 1. Команды для работы с матрицами в R;

Помимо упомянутых в работе команд (matrix, dim и др.) для матриц есть:

* rowSums(x, na.rm=FALSE, dims=1) – возвращает суммы строк;
* colSums(x, na.rm=FALSE, dims=1) – возвращает суммы столбцов;
* rowMeans(x, na.rm=FALSE, dims=1) – среднее значение элементов каждой строки;
* colMeans(x, na.rm=FALSE, dims=1) – среднее значение элементов каждого столбца;
* t(x) – возвращает транспонированную матрицу;
* apply(X, MARGIN, FUN, …) – применяет функцию к каждой строке (MARGIN=1) или столбцу (MARGIN=2);
  1. Работа с графикой в R;

Помимо обычных графиков (plot) в R можно создавать:

* barplot(height, …) – создает столбчатые диаграммы;
* hist(x, …) – создает гистограммы;
* boxplot(x, …) – "ящик с усами" для отображения распределения данных;
* pie(x, …) – создает круговые диаграммы;

# ВЫВОД

В ходе ознакомления с особенностями языка R было определено, что он предназначен для обработки и анализа больших объёмов структурированных данных. Также он обладает стандартным набором функций для проведения арифметических и логических операций, представления данных в различных типах и структурах, визуализации статистической информации.

Основные операторы и функции были выписаны и проанализированы в ходе работы. Полученные знания позволяют приступить к более углублённому изучения аналитических функций языка R.