### Исследование методов прогнозирования с помощью среды R. Базовые графические возможности R.

Как правило, создание графика начинается с функции высокого уровня, которая определяет его общую структуру: размерность (1D, 2D, 3D), масштабы осей, названия и др. Наиболее часто используемые графические функции высокого уровня – plot(), hist(), boxplot(), scatterplot() и pairs(). С использованием богатого набора функций низкого уровня к построенному графику могут быть добавлены дополнительные элементы: текст, линии, легенда и проч. (примерами таких функций являются lines(), points(), text() и axis()).

Особенностями деталей графического изображения управляет набор параметров, как правило, общий для большинства высокоуровневых и низкоуровневых функций, который определяет цвет, типы и размер символов или маркеров, толщину и характер линий, штриховку, рамку графика и проч.

##### Функция plot() и ее параметры

plot() – основная функция, используемая для построения графиков в R. Поведение этой функции высокого уровня определяется классом объектов, указываемых в качестве ее аргументов. Соответственно, с помощью plot() можно создать очень большой набор разнотипных графиков.

В качестве примера используем данные по скорости выведения из организма человека индометацина – одного из наиболее активных противовоспалительных препаратов. В эксперименте приняли участие шесть испытуемых. Результаты этого исследования входят в базовый набор данных R и доступны по команде

data(Indometh)

Применив команду

names(Indometh)

[1] "Subject" "time" "conc"

видим, что в состав таблицы Indometh входят переменные Subject (испытуемый), time (время с момента введения препарата) и conc (концентрация препарата в крови).

Чтобы облегчить дальнейшую работу, прикрепим таблицу Indometh к поисковому пути R:

attach(Indometh)

Благодаря этой команде, теперь мы можем напрямую обращаться к переменным таблицы Indometh (т.е. использовать их имена непосредственно, например, time вместо Indometh$time).

Зависимость концентрации индометацина в крови от времени можно легко изобразить при помощи следующей команды:

plot(time, conc)



Предположим, что перед нами стоит задача отобразить на графике не все исходные данные, а только средние значения концентрации индометацина для каждой временной точки. Рассчитать средние значения (или любые другие количественные величины) для отдельных групп данных позволяет функция tapply():

(means <- tapply(conc, time, mean))

0.25 0.5 0.75 1 1.25 2 3

2.07666667 1.32166667 0.91833333 0.68333333 0.55666667 0.33166667 0.19833333

4 5 6 8

0.13666667 0.12500000 0.09000000 0.07166667

Можно создать числовой вектор со значениями времени учета концентрации препарата:

indo.times <- as.numeric(names(means))

# строим график типа “точки с линиями”:

plot(indo.times, means, type="b")



**Управляющие параметры функции plot()**

Функция plot() имеет большое количество управляющих параметров, которые позволяют осуществить тонкую настройку внешнего вида графика. Ниже рассмотрены некоторые из них.

1. Параметры xlab и ylab

Параметры xlab и ylab служат для изменения названий осей X и Y соответственно:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация")

2. Параметр type

Параметр type позволяет изменять внешний вид точек на графике. Он принимает одно из следующихзначений:

° "p" – точки (points; используется по умолчанию)

° "l" – линии (lines)

° "b" – изображаются и точки, и линии (both points and lines)

° "o" – точки изображаются поверх линий (points over lines)

° "h" – гистограмма (histogram)

° "s" – ступенчатая кривая (steps)

° "n" – данные не отображаются (no points)

3. Параметры xlim и ylim

Эти два параметра контролируют размах значений на каждой из осей графика. По умолчанию они оба принимают значение NULL – в этом случае размах выбирается программой автоматически. Для отмены автоматических настроек соответствующему параметру необходимо присвоить значение в виде числового вектора, содержащего минимальное и максимальное значения, которые должны отображаться на оси. Например:

plot(indo.times, means, xlab="Время", ylab="Концентрация", xlim=c(0, 15))

plot(indo.times, means, xlab="Время", ylab="Концентрация", ylim=c(0, 5))

4. Параметры axes и ann

Эти два параметра контролируют отображение осей и их названий соответственно.

Каждый из них может принимать одно из двух возможных значений – TRUE или FALSE:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", axes = TRUE, ann = TRUE)

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", axes = FALSE, ann = TRUE)

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", axes = TRUE, ann = FALSE)

5. Параметр log

При помощи аргумента log можно перевести одну или обе оси графика на логарифмическую шкалу, например:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", log = "x")

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", log = "y")

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", log = "xy")

6. Параметр main

Аргумент main служит для создания заголовка графика. По умолчанию название размещается в верхней части рисунка:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o")

Далее будут рассмотрены графические параметры, контролирующие внешний вид графиков, например, тип, размер и цвет символов и линий, тип и размер шрифта в названиях графика и его осей, использование математических символов в названиях, размещение легенды, и т.п. Они применяются в качестве аргументов не только при вызове plot(), но и многих других функций.

Управление общими параметрами – аргументами графических функций

1. Тип символа

Как видно из приведенного выше рисунка, отдельные измерения по умолчанию изображаются в виде кружков. Изменить тип символов, используемых для отображения наблюдений, позволяет аргумент pch (plotting character – символ изображения). В стандартных случаях этот аргумент принимает численные значения от 1 до 25. Например, при pch = 2 символы превратятся из кружков в незакрашенные треугольники:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", pch = 2)

Таблица 25-ти стандартных маркеров и соответствующие им численные коды представлена ниже:



Набор стандартных маркеров может быть значительно расширен в случае, когда аргумент pch используется в комбинации с другим аргументом – font, задающим шрифт символов. Параметр pch может при этом принимать любое целое число от 1 до 128 и от 160 до 254. Например, при font = 5 маркеру в виде "сердечка" соответствует код 169:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", pch = 169, font = 5)

В качестве маркеров можно также использовать обычные печатные символы, например, буквы:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", pch = "A")

2. Размер маркера

Размер маркеров задается при помощи аргумента cex (character extension – размер символа), который по умолчанию равен 1. Уменьшение или увеличение этого параметра приводит к соответствующим пропорциональным изменениям размера символов.

При необходимости мы можем также изменить ширину линии обводки символа. Для этого служит параметр lwd (line width – ширина линии).

3. Цвет маркера

Цвет любого графического объекта может быть задан несколькими способами:

по названию цвета: например, col = "red" (красный), col = "green" (зеленый), или col = "black" (черный). Всего в R имеется 675 стандартных цветов. Их названия доступны по команде colors();

путем непосредственного указания красного, зеленого и синего компонентов RGB спектра, например: "#RRGGBB";

по численному коду, например: col = 2 (красный), col = 3 (зеленый), или col = 1 (черный).

Цвет маркеров задается при помощи аргумента col (color – цвет). Подобрать его можно следующим образом:

n <- 20

plot(1:n, pch=CIRCLE<-16, cex=1:n, col=1:n)

text(1:n)



Имеются также отдельные параметры для настройки цвета других элементов графика (например, заголовка col.main, названий осей col.lab, меток осей col.axes и др.).

Ниже приведены несколько примеров, иллюстрирующих эффекты параметра col.

# символы в виде треугольников синего цвета pch = 2, col = "blue"

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", pch = 2, cex = 1.2, col = "blue")

# символы в виде ромбиков (pch = 5) красного цвета (col = 2)

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", pch = 5, cex = 1.2, col = 2)

# заголовок графика синего цвета (col.main = "blue")

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", col.main = "blue", cex = 1.2)

# названия осей выполнены красным цветом (col.lab = "red")

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", col.main = "blue", col.lab = "red", cex = 1.2)

При работе с символами 21-25 (см. таблицу выше) можно использовать аргумент bg (background – фон) для указания цвета, которым они должны быть закрашены, например:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "Скорость выведения индометацина", type = "o", pch = 21, cex = 1.2, bg = "red", lwd=2, col.main = "blue")



4. Ширина линии

Ширина линии задается при помощи аргумента lwd (от line width) функции plot(). Аргумент принимает положительные числовые значения, показывающие, во сколько раз ширина линии должна быть больше относительно ширины, заданной по умолчанию. Ширина линии (по умолчанию равна 1) является безразмерной величиной, поскольку на разных графических устройствах линии с одинаковыми параметрами могут выглядеть по-разному. Ниже приведены примеры трех графиков с разными значениями параметра lwd:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "lwd = 2", type = "l", lwd = 2)

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "lwd = 5", type = "l", lwd = 5)

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "lwd = 10", type = "l", lwd = 10)

5. Концы и места соединения линий

Аргумент lend (от line end – окончание линии) функции plot() позволяет настроить внешний вид концов линии. Этот аргумент принимает значения 0 (по умолчанию), 1 или 2, что соответствует округлым, усеченным квадратным и квадратным концам соответственно. Места соединения линий также могут выглядеть по-разному, что определяется аргументом ljoin (от line – линия, и join – место соединения). Аргумент ljoin принимает значения 0 (по умолчанию), 1 или 2, что соответствует округлому, остроугольному и усеченному соединениям соответственно.

6. Тип линии

Тип линии настраивается при помощи аргумента lty (от line – линия, и type – тип) функции plot(). Существует шесть предустановленных типов линий, которые задаются числами от 1 до 6 соответственно.

При необходимости можно создать пользовательские типы линий. В таких случаях в качестве значения аргумента lty выступает текстовая последовательность из четырех цифр. Эти числа (от 1 до 9) определяют размер четырех элементов, составляющих повторяющийся паттерн "штрих - пробел - штрих - пробел". Например, при lty = "4241" линия будет состоять из повторяющегося паттерна, в котором имеется штрих длиной 4 единицы, пробел длиной 2 единицы, опять штрих длиной 4 единицы, и пробел в 1 единицу. Примеры пользовательских типов линий приведены ниже:



Цвет линий задается при помощи аргумента col (от color – цвет). Использование параметра col в отношении линий ничем не отличается от его использования в отношении графических символов.

7. Рамка графика

Для настройки внешнего вида рамки графика служит аргумент bty (от box – коробка, и type – тип) функции plot(). Этот аргумент принимает одно из следущих шести текстовых значений:

"O" "L" "7" "C" "U" "["

Рамка будет принимать вид в соответствии с формой указанного символа (допускается использование также строчных букв o, l, c, и u). Ниже приведен пример использования различных перечисленных опций:

plot(indo.times, means, xlab = "Время", ylab = "Концентрация", main = "", type = "l", lwd = 10, lty = 2, col = 6, bty= "L")



##### Гистограммы, функции ядерной плотности

Гистограмма является важным инструментом статистики, позволяющим наглядно представить распределение значений анализируемой переменной. В системе R для построения гистограмм служит функция hist(). Ее основным аргументом выступает имя анализируемой переменной.

В качестве примера создадим нормально распределенную совокупность X из 100 наблюдений со средним значением 15 и стандартным отклонением 5:

X <- rnorm(n = 100, mean = 15, sd = 5)

Для создания переменной X использована функция rnorm() (от random –случайный, и norm – нормальный). Используя генератор случайных чисел, эта функция формирует нормально распределенные совокупности с заданными размером (n), средним значением (mean) и стандартным отклонением (sd).

Изобразить значения переменной X в виде гистограммы очень просто:

hist(X)



Как видно из приведенного рисунка, функция hist() автоматически выбирает количество столбцов для отображения на графике, а также создает названия осей и заголовок графика.

В приведенном выше примере программа автоматически разбила значения переменной X на 6 классов. Для более детальной прорисовки можно увеличить дробность деления данных на классы (т.е. использовать меньший классовый промежуток). Сделать это позволяет аргумент breaks (разломы) функции hist(). При необходимости столбцы гистограммы можно залить желаемым цветом. Для этого следует воспользоваться аргументом col.

hist(X, breaks = 20, col = "lightblue")



Как видим, результатом выполнения предыдущей команды стала гистограмма с двадцатью столбцами, позволяющая более детально проанализировать распределение значений переменной X.

По умолчанию функция hist() отображает по оси ординат частоты встречаемости для каждого класса значений X. Такое поведение функции можно изменить, придав аргументу freq (от frequency – частота) значение FALSE. В этом случае ось ординат будет отражать плотность вероятности каждого класса так, что суммарная площадь под гистограммой составит 1:

hist(X, breaks = 20, freq = FALSE)

В ряде случаев, в частности при небольшом числе наблюдений, гистограммы могут давать неверное представление о свойствах совокупности, например, из-за небольшого числа редко расположенных столбцов:

X <- rnorm(n = 50, mean = 15, sd = 5)

hist(X, breaks = 20, freq = FALSE, col = "lightblue")

Вместо гистограммы (или параллельно с ней) в таких случаях рекомендуется воспользоваться кривой плотности вероятности. Оценка плотности вероятности выполняется при помощи функции density(), которую можно применить в качестве аргумента функции plot() для графического изображения результата:

plot(density(X))

Гладкость получаемой кривой регулируется при помощи аргумента bw (от bandwidth – ширина окна), например:

plot(density(X, bw = 0.8))

Для полноты картины гистограмму можно совместить с кривой плотности вероятности. При этом сначала необходимо построить саму гистограмму, а затем добавить к ней кривую плотности при помощи функции lines():

hist(X, breaks = 20, freq = FALSE, col = "lightblue", xlab = "Переменная X", ylab = "Плотность вероятности", main = "Гистограмма, совмещенная с кривой плотности")

lines(density(X), col = "red", lwd = 2)

lines(density(X, bw = 0.8), col = "blue", lwd = 2)



Наконец, для того, чтобы оценить взаимодействие между двумя переменными, представляет интерес построить график поверхности ядерной плотности распределения двухмерной случайной величины z = f(x ,y). Это можно сделать с использованием функции kde2d() из популярного пакета MASS. В качестве примера покажем диаграмму совместного распределения показателей time и conc в эксперименте со скоростью выведения индометацина:

data(Indometh) ; attach(Indometh)

library(MASS)

f <- kde2d(time, conc)

image(f,xlab="Время выведения",ylab="Концентрация индометацина")

contour(f, add=TRUE)

Здесь функция image() создает окрашенную прямоугольную сетку, цвет которой зависит от значения переменной f (в нашем случае это – плотность вероятности), а функция contour() – добавляет на график изолинии.



Кроме этого, R поддерживает создание следующих диаграмм: диаграммы размахов, круговые и столбиковые диаграммы, диаграммы Кливленда и одномерные диаграммы рассеяния, категоризованные графики и другие.

##### Задание

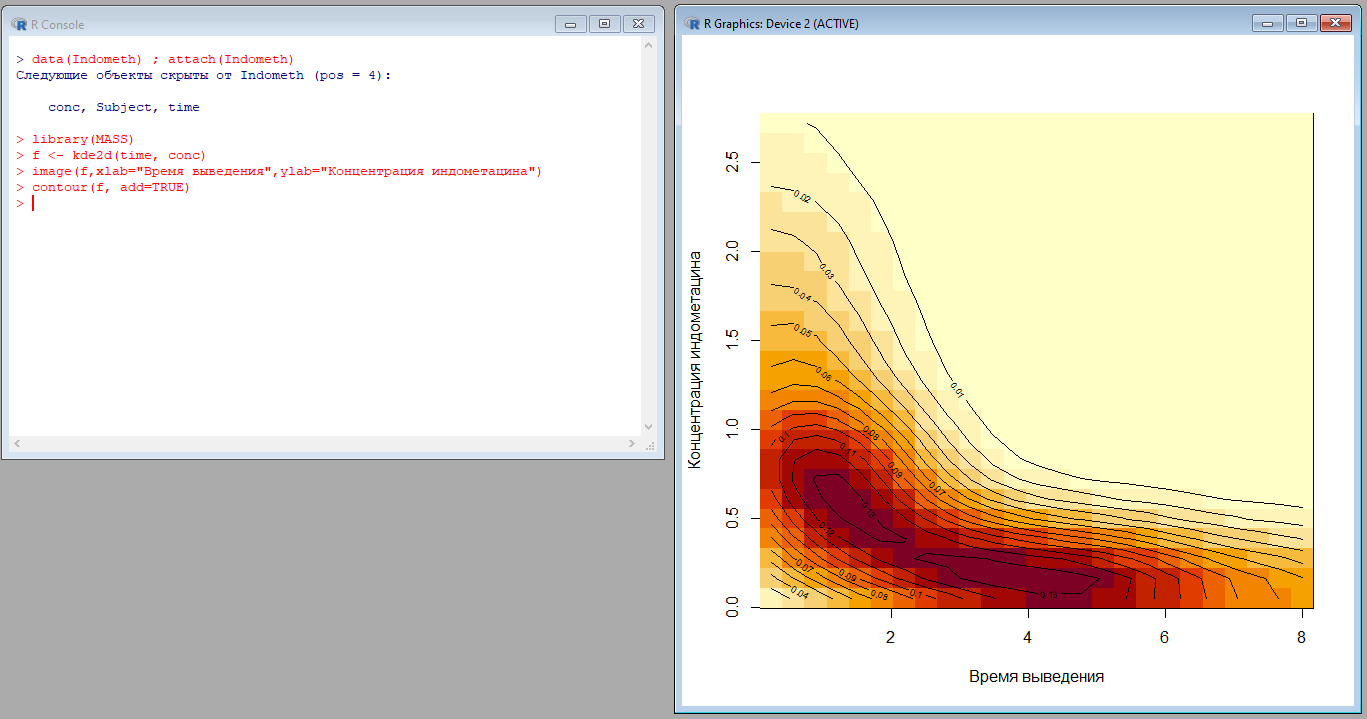
1) Выполнить приведенные выше примеры для скорости выведения индометацина и нормального распределения. В качестве параметров нормального распределения взять следующие: количество наблюдений – 200 среднее значение – 20, стандартное отклонение – 5;  


Рисунок 1 – Пример для скорости выведения индометацина.

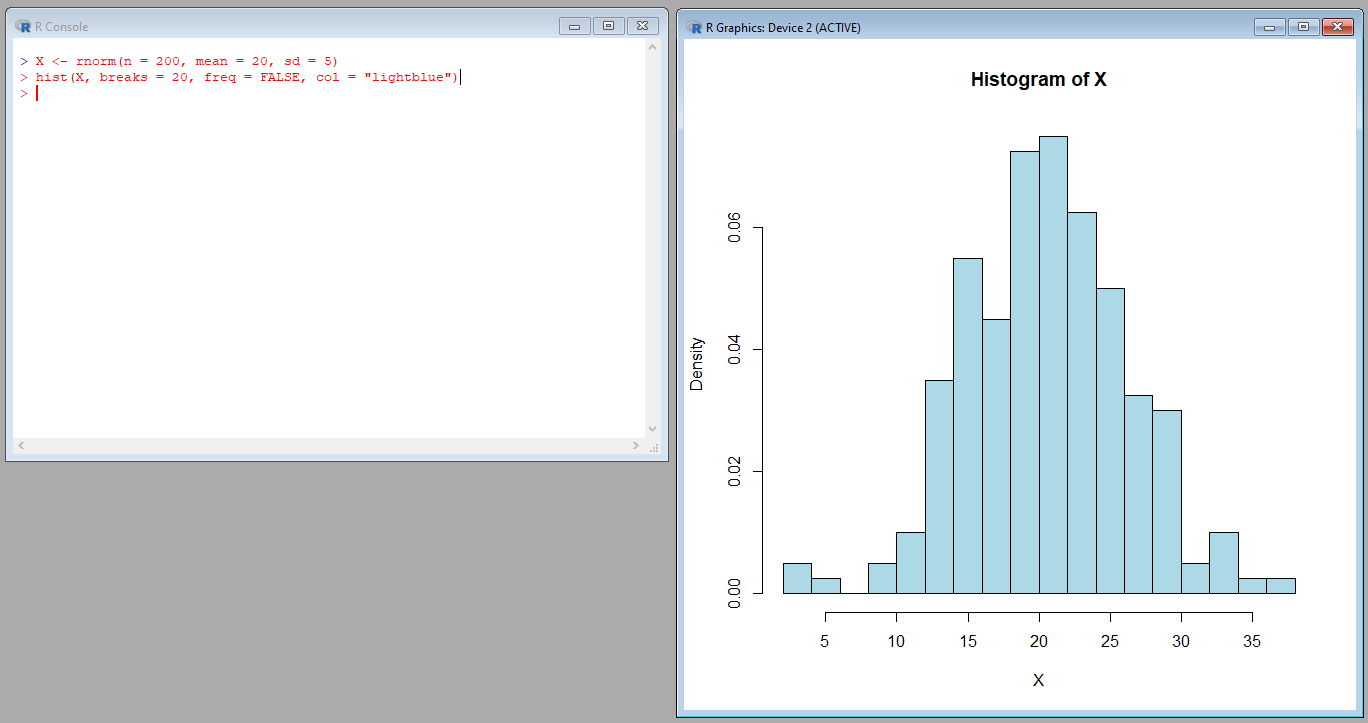


Рисунок 2 – Пример нормального распределения.

2) Проанализировать свои данные. Выбрать данные для построения графиков с использованием функции plot(), hist(). Построить 5-6 графиков с различным оформлением либо различными исходными данными.

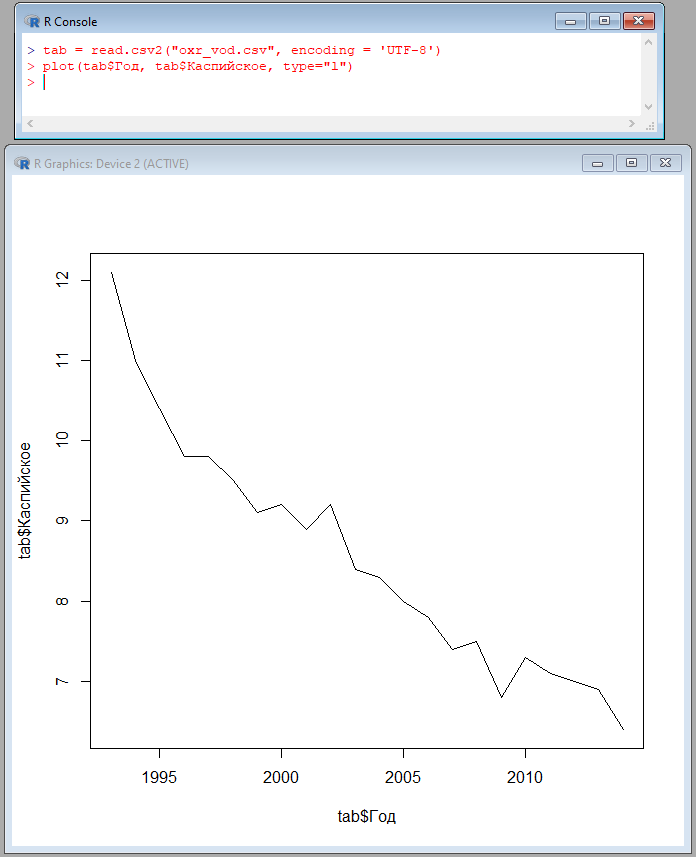


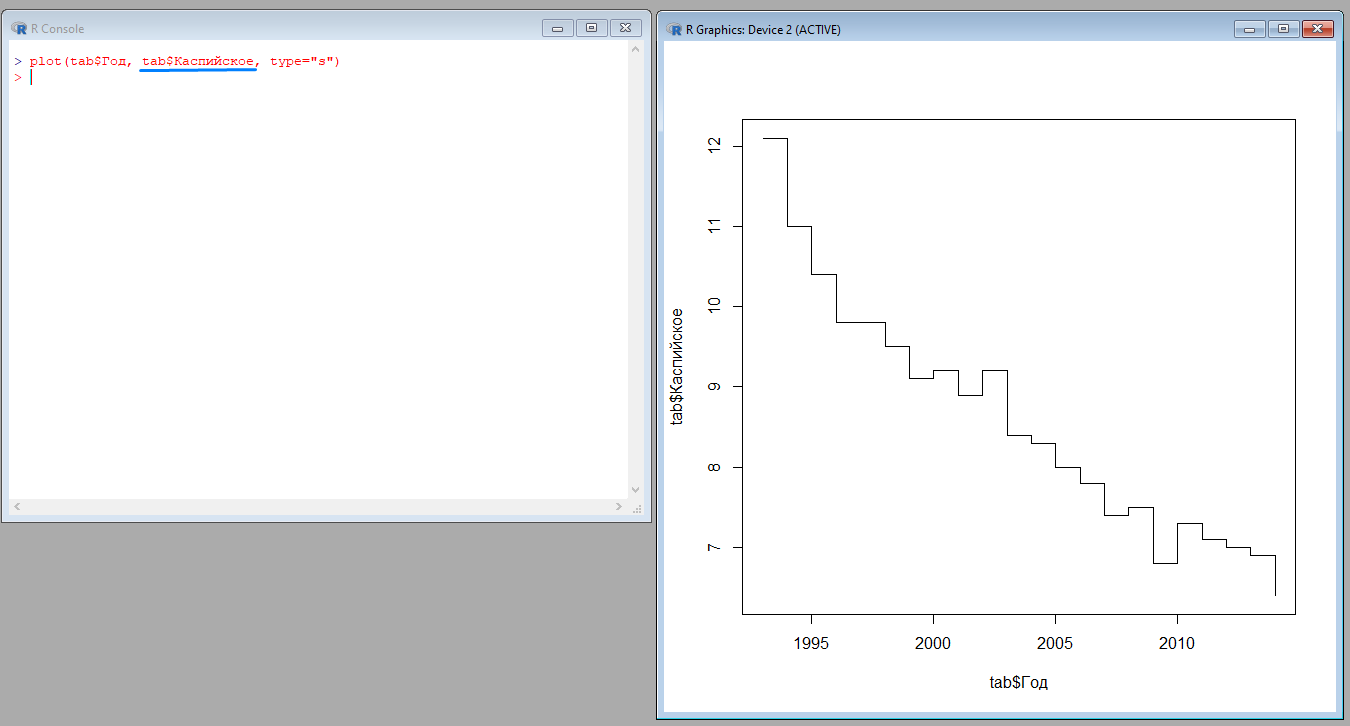
Рисунок 3 – Объем сброса загрязненных сточных вод по Каспийскому морю (млрд куб. м).

Рисунок 4 – Объем сброса загрязненных сточных вод по Каспийскому морю (млрд куб. м).

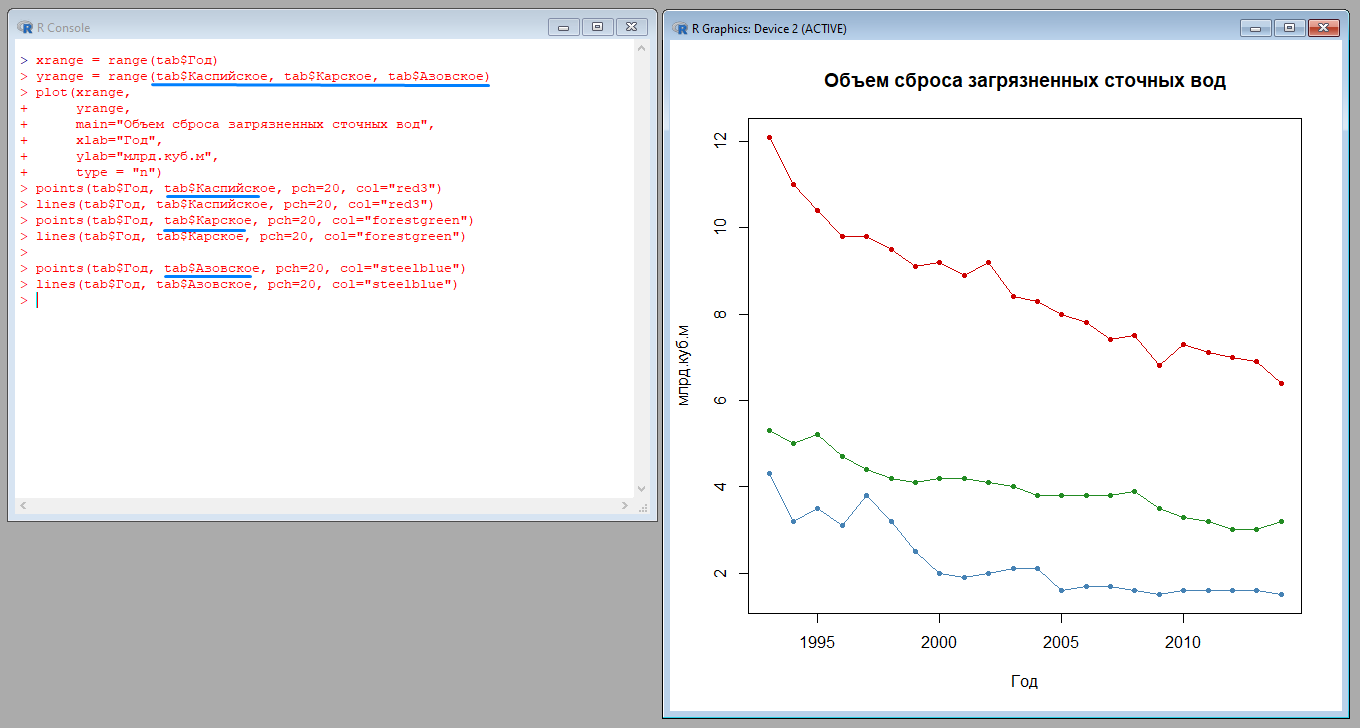


Рисунок 5 – Объем сброса загрязненных сточных вод по морям Каспийское, Карское, Азовское (млрд куб. м).

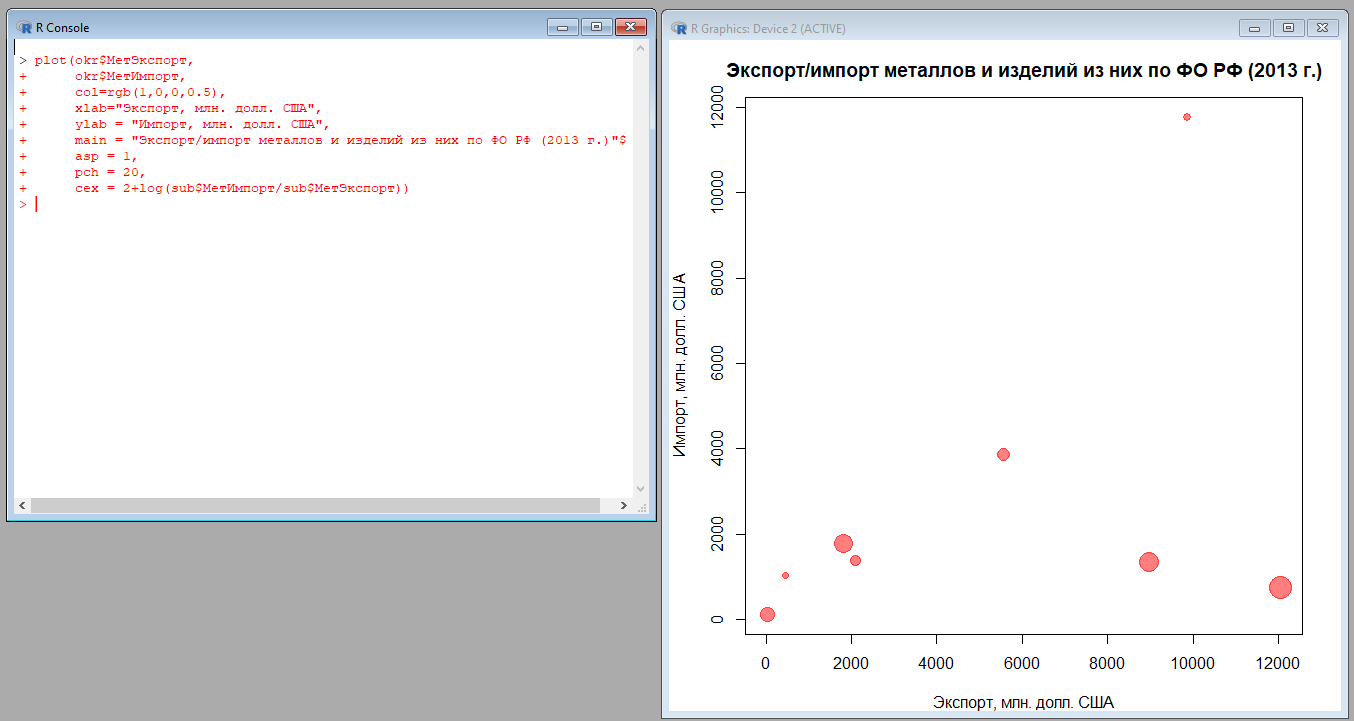


Рисунок 6 – Экспорт/импорт металлов и изделий из них по ФО РФ (2013 г.).

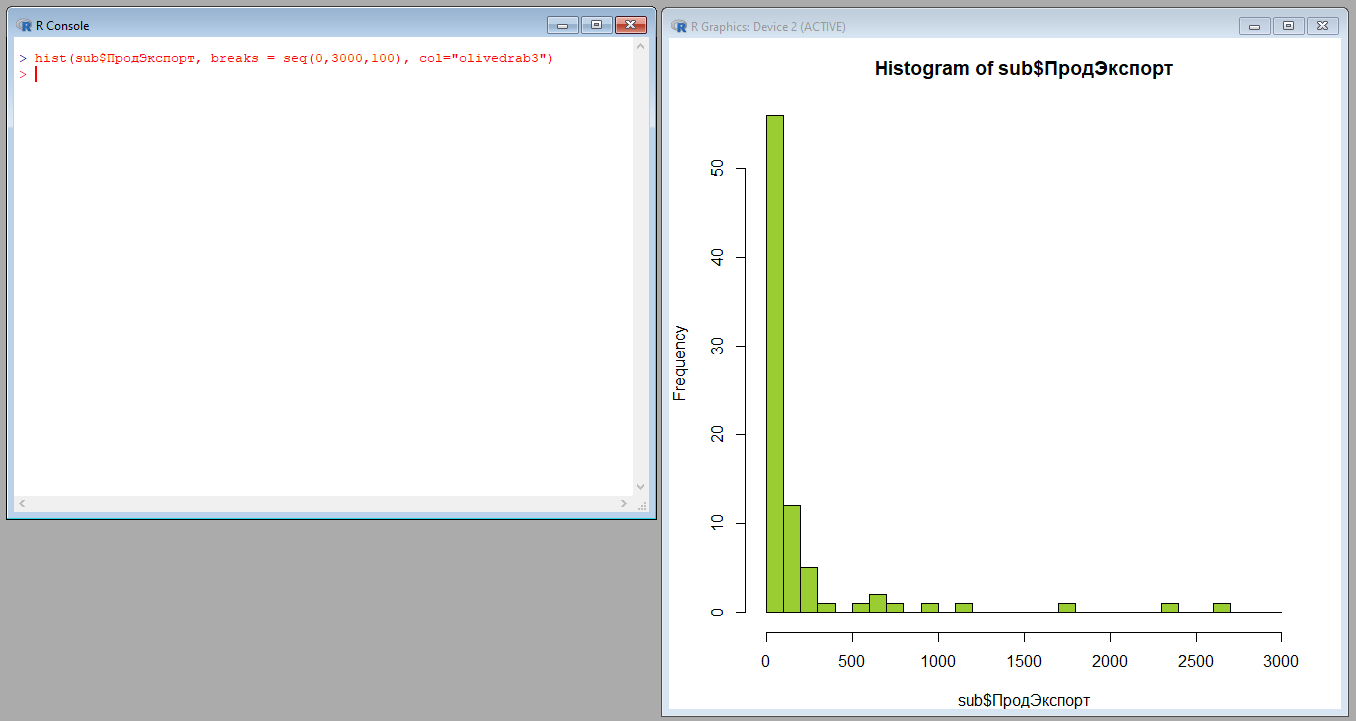


Рисунок 7.

3) Выполнить предложенный скрипт из предыдущей практической работы Example1.R. Изменить параметры функции plot(), например, изменить маркеры, цвет и т.п. Результат представить в отчете.

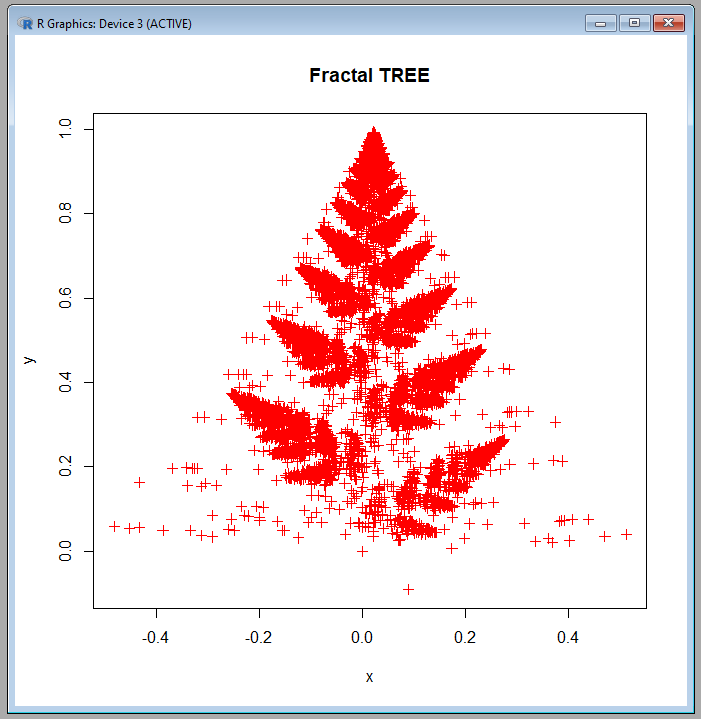


Рисунок 7 – Скрипт Example1.R с измененным цветом.

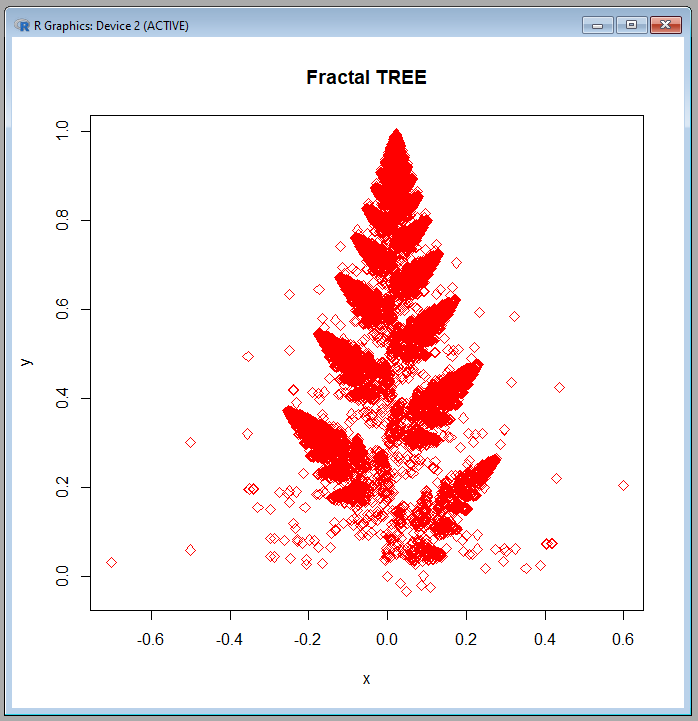


Рисунок 7 – Скрипт Example1.R с измененным pch.

##### Содержание отчёта

1. Фрагмент данных, использованных для выполнения работы.

2. Полученные результаты по каждому пункту (или «скриншоты»).

3. Выводы о проделанной работе.