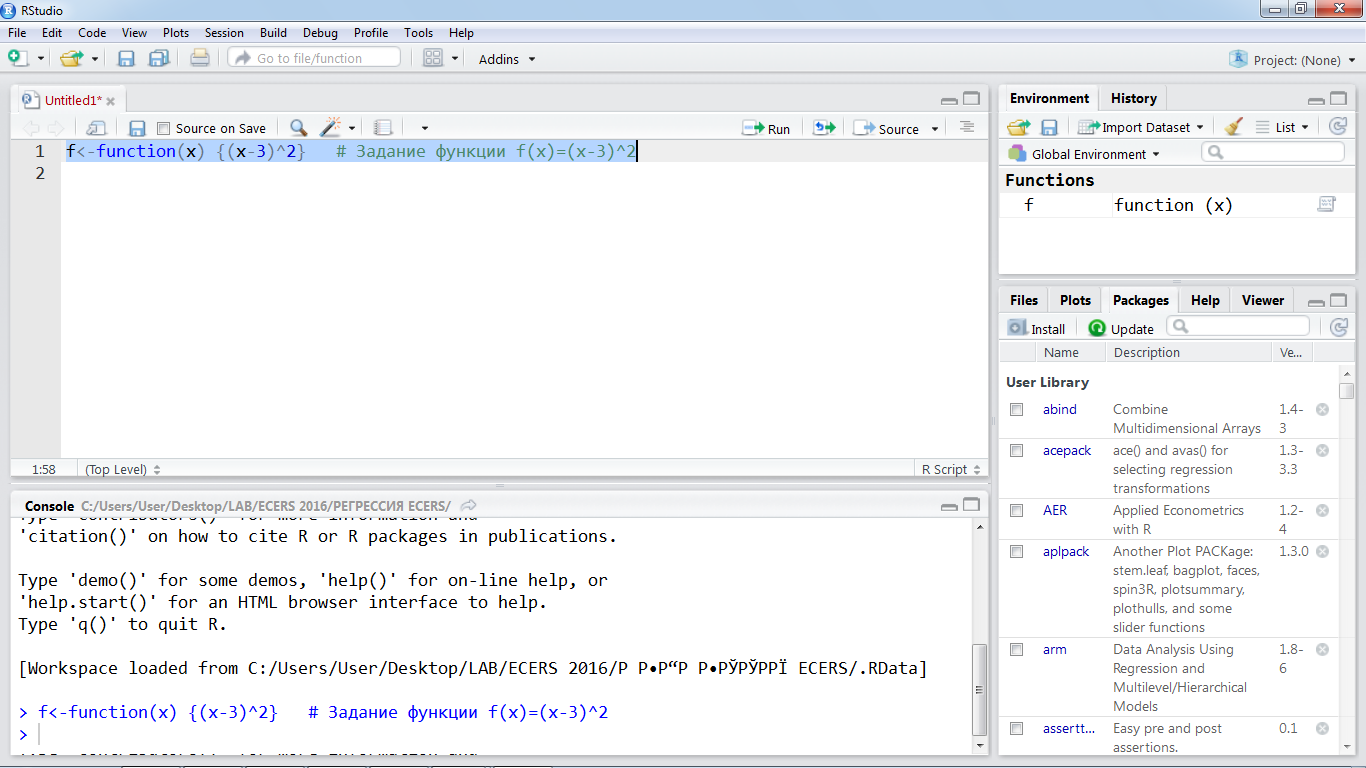
**Графика в R**

**Задание математических функций и построение графиков**

Запустим RStudio, создадим новый лист программы (например, по сочетанию Ctrl+Shift+N) и наберем или скопируем следующий код:

f <-function(x) {(x-3)^2} # Задание функции f(x)=(x-3)^2

Что произойдет, если мы отправим этот код на компиляцию (Ctrl+Enter)?



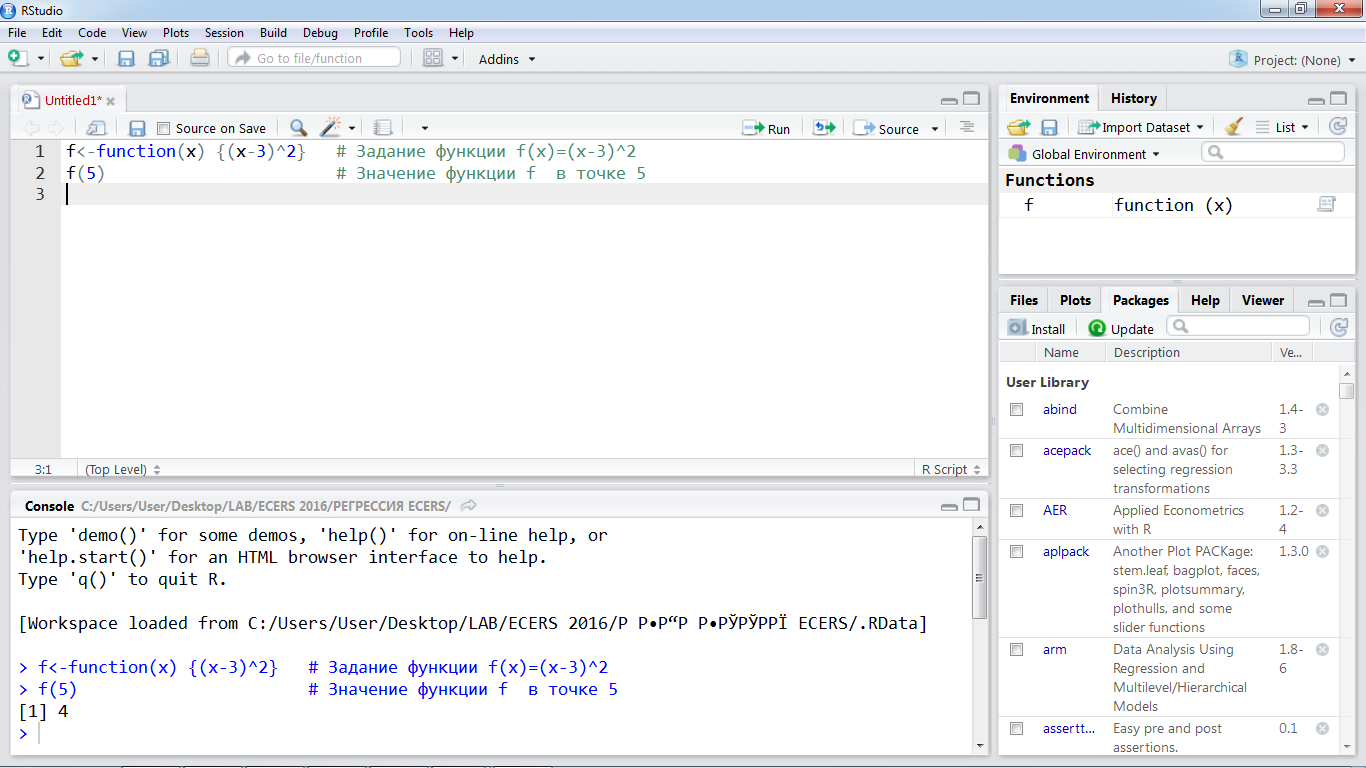
На первый взгляд R никак не отреагировал на введенный текст, однако в действительности, в памяти R создалась функция под именем , значения которой, согласно нашему коду в фигурных скобках {(x-3)^2}, зависят от одной переменной следующим образом:

.

Обратите внимание на правое верхнее окно RStudio: в нем появилась только что созданная функция .

Это означает, что теперь R будет понимать обращение к этой функции для конкретных чисел. Например, вычислим значение этой функции в точке , набрав:

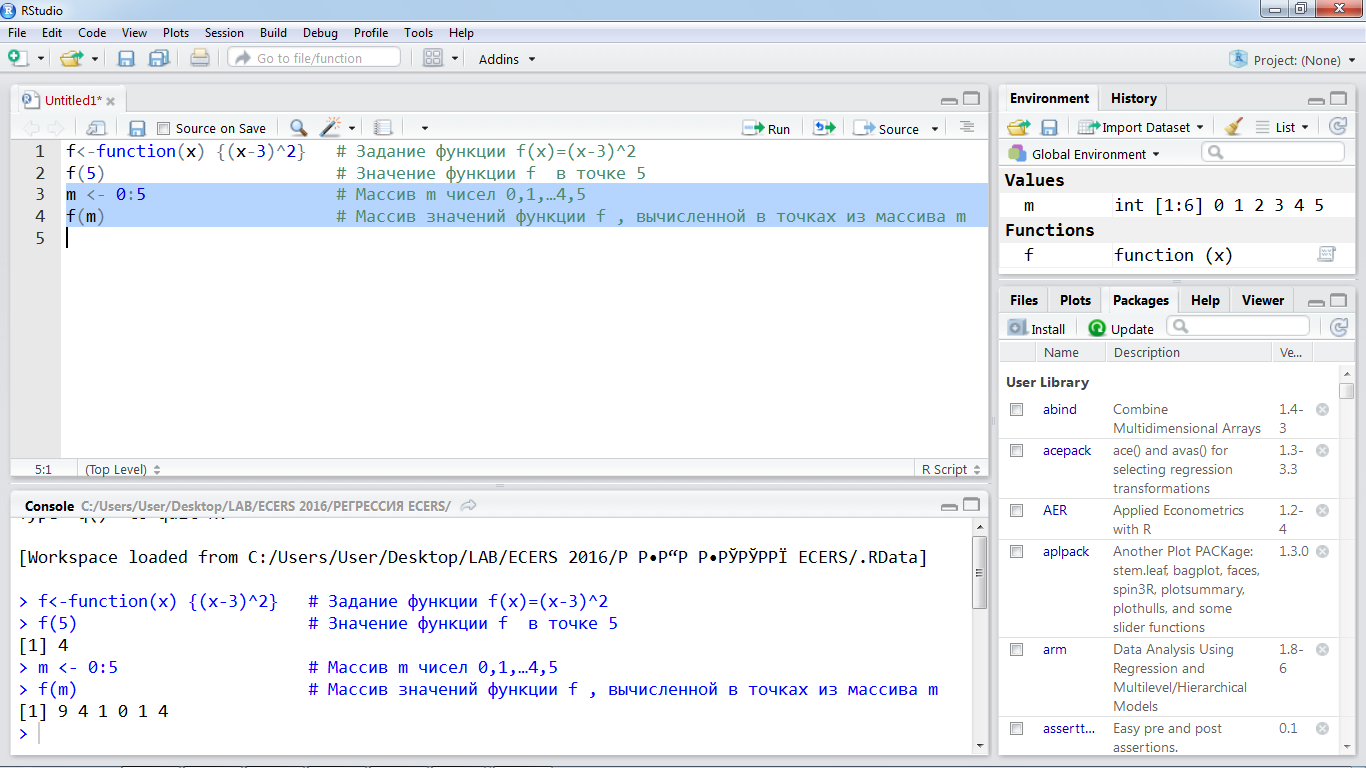
f(5) # Значение функции f в точке 5



В качестве аргумента введенной функции можно подставлять и целые массивы, тогда функция будет вычислена в каждой точке массива:

m <- 0:5 # Массив m целых чисел 0,1,…4,5

f(m) # Массив значений функции f , вычисленной в точках из массива m



Как видим, строка внизу консоли R перечисляет все значения функции. Таким образом, можно легко задавать/генерировать конечные последовательности значений функции для заданных последовательностях значений аргументов.

**Задание 1.** Объявить в R функцию

,

где и – параметры (по умолчанию считать параметры равными единицам) и вычислить значения , и .

**Решение.** Наберем следующий код в окне программы:

g <-function(x , a = 1, b = 1) {(x^2-a)/(x-b)} # Задание функции g(x,a,b)

Если теперь в следующей строчке набрать

g (2) # Вычисление функции g(x=2,a=1,b=1)

мы получим вычисленное значение функции для , при этом опущенные нами параметры и будут приравнены к значениям по умолчанию, т.е. 1, объявленным ранее в аргументах function(x , a = 1, b = 1).

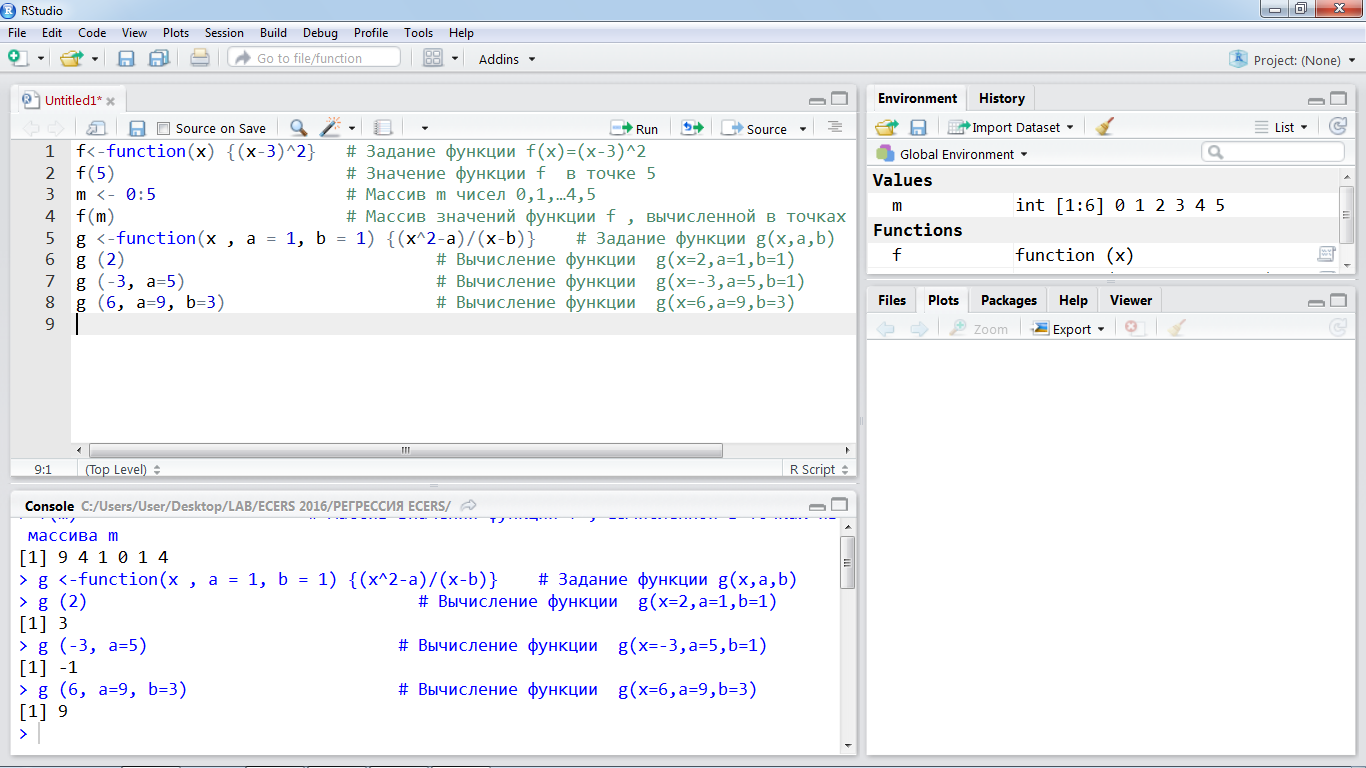
Вызов функции в формате:

g (-3, a=5) # Вычисление функции g(x=-3,a=5,b=1)

даст нам значение функции для . При этом параметр , а опущенное значение параметра будет приравнено к значению по умолчанию, т.е. 1.

Естественно, возможно и полное обращение к нашей функции, с указанием всех ее аргументов:

g (6, a=9, b=3) # Вычисление функции g(x=6,a=9,b=3)



*Замечание 1.* При вызове функции мы опускали название переменной, но не было бы ошибкой записать . Вообще, в R можно опускать названия аргументов функции, если только помнить в каком порядке они перечисляются. Так, например, команда будет эквивалента вызову функции .

*Замечание 2.* Попробуйте объяснить результат NaN («невычисляемо», «не число»), если ввести команду:

g (3, 9, 3) # Вычисление функции g(x=6,a=9,b=3)

**Построение графиков функций**

Используем возможности, описанные в предыдущем пункте, для построения графиков функций.

**Задание 2.** Построить график функции на отрезке

**Решение.** Данную функцию мы уже ранее объявили в R. Зададим последовательность аргументов , пробегающих отрезок с достаточно малым шагом, например, . Это можно сделать с помощью команды «seq»:

x <- seq(-5, 8, 0.05) # Последовательность чисел от -5 до 8 с шагом 0.05, записанная в x

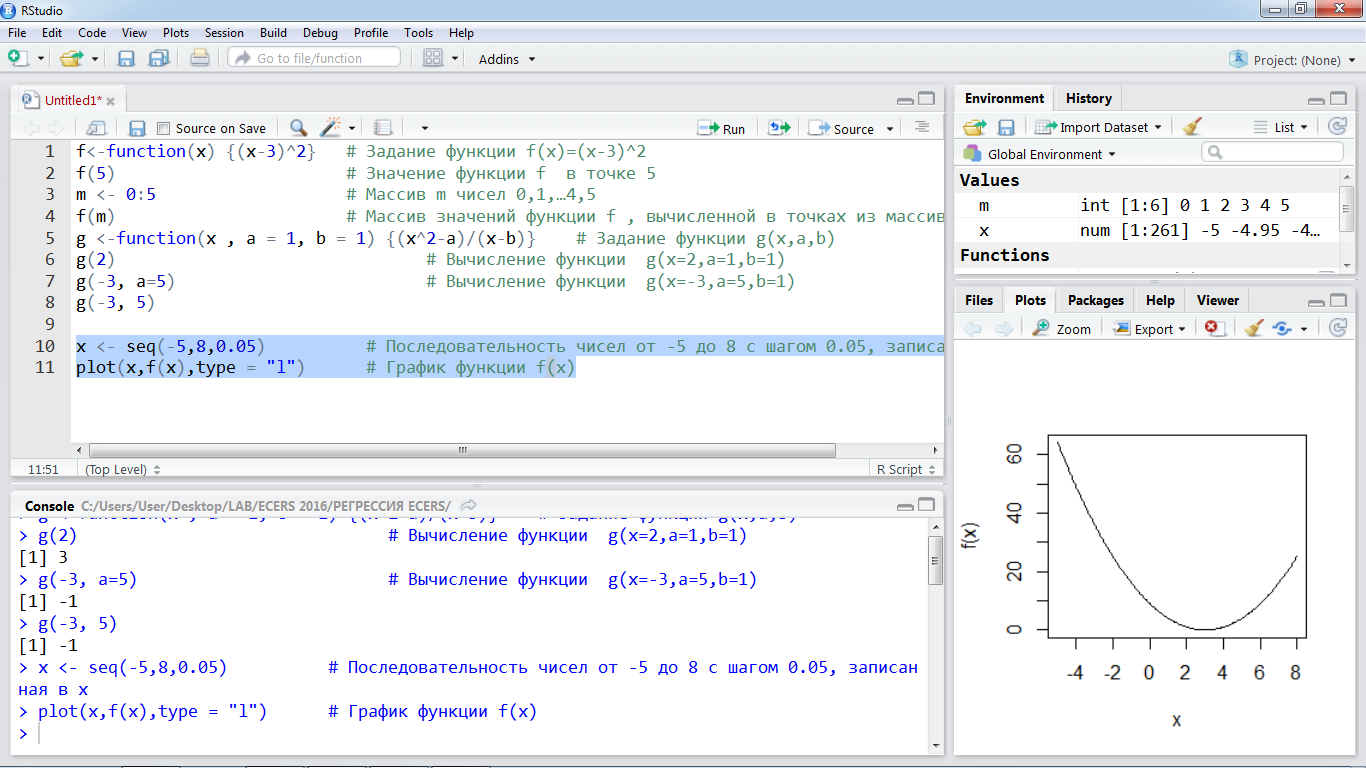
или в расширенном виде

x <- seq(from = -5, to = 8, by = 0.05) # То же самое

Теперь остается вызвать стандартную функцию построения графика:

plot(x,f(x), type = "*l*") # График функции f(x)

Здесь в качестве значения параметра типа кривой «type» выбрано значение "" от слова line – соединение точек прямыми линиями.



Функция seq имеет еще несколько полезных параметров. Например, часто бывает удобно задать последовательность от *a* до *b* не с помощью шага, а посредством указания общего количества точек в последовательности. Следующая ниже строчка кода задает последовательность из 5 чисел, эквидистантно пробегающим диапазон от 21 до 36:

x <- seq(from = 21, to = 36, length.out = 5) # Последовательность из пяти чисел от 21 до 36

> x <- seq(from = 21, to = 36, length.out = 5) # Последовательность из пяти чисел от 21 до 36

> x

[1] 21.00 24.75 28.50 32.25 36.00

**Задание 3.** Построить графики функции на отрезке при различных значениях параметров:

**Решение.** Функция нами была уже объявлена. Зададим, как и раньше, последовательность аргументов

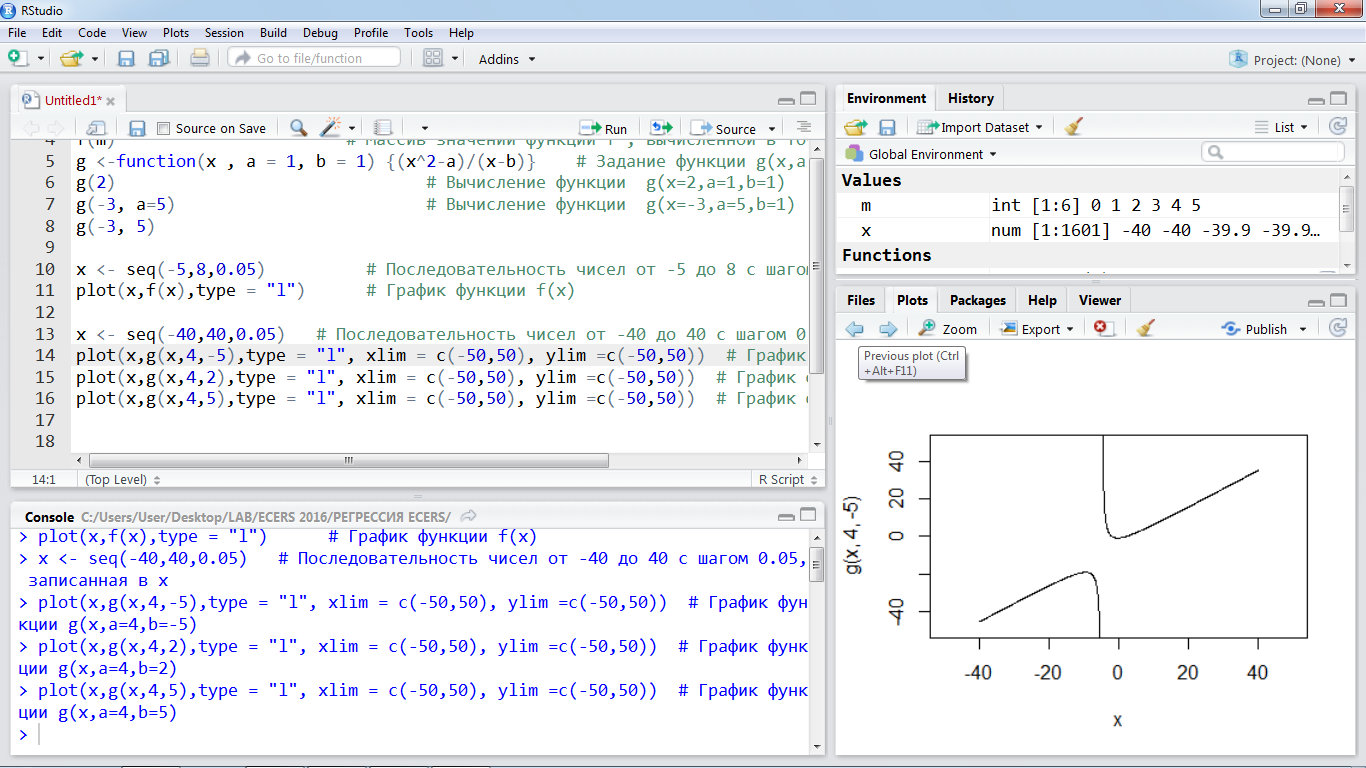
x <- seq(-40, 40, 0.05) # Последовательность чисел от -40 до 40 с шагом 0.05, записанная в x

и поочередно вызовем построение графиков соответствующих функций:

plot(x,g(x,4,-5),type = "*l*", xlim = c(-50,50), ylim =c(-50,50)) # График функции g(x,a=4,b=-5)

plot(x,g(x,4,2),type = "*l*", xlim = c(-50,50), ylim =c(-50,50)) # График функции g(x,a=4,b=2)

plot(x,g(x,4,5),type = "*l*", xlim = c(-50,50), ylim =c(-50,50)) # График функции g(x,a=4,b=5)



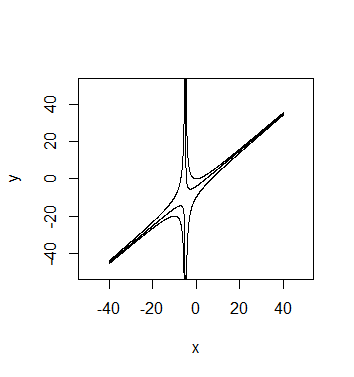
Для того, чтобы вернуться к предыдущему или последующему графику используйте в левой верхней части окна графика стрелки вперед и назад. Также, полезно нажать кнопку Zoom, находящуюся чуть правее, для увеличения картинки.

Кстати, часто бывает необходимым разместить несколько графиков на одном рисунке. Достигается это заменой последующих plot на lines. Например:

plot(x,g(x,0,-5),type = "*l*", xlim = c(-50,50), ylim =c(-50,50) ,ylab = "y") # График g(x, 0, -5)

lines(x,g(x,20,-5),type = "*l*", xlim = c(-50,50), ylim =c(-50,50)) # Добавлен график g(x,20, -5)

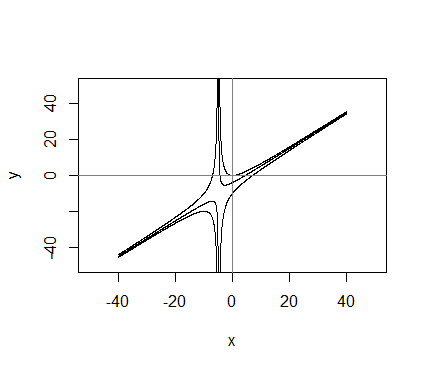
lines(x,g(x,50,-5),type = "*l*", xlim = c(-50,50), ylim =c(-50,50)) # Добавлен график g(x,50, -5)



Как можно было заметить мы использовали дополнительные параметры в команде plot: xlim и ylim, устанавливающие границы изменения и на графике от до .

При желании, можно разместить на рисунке привычные оси координат *ox* и *oy*, дополнительно используя следующую команду:

abline(h = 0, v = 0, col = "gray50") # Нанесение на график линий ox и oy



Полный перечень аргументов данной команды plot и их точное описание доступно по команде вызова справки:

?plot # Вызов справки по команде plot

или нажатием клавиши F1 при поставленном курсоре перед первой буквой интересующей нас команды. Попробуйте найти графический параметр *tck* для функции plot, который автоматически рисует координатную сетку? Если задать его значение *tck = 1*.

**Задания для самостоятельной работы**

1. Построить график функции на отрезке . *( – функция, возвращающая знак числа , т.е. +1 для положительных и –1 для отрицательных, в нуле – ноль).* Указание: используйте встроенную в R функцию sign.
2. Подготовить 10 разных графиков с различными вариантами оформления. Использовать все изученные параметры. В качестве наборов данных использовать различные математические функции. Подобрать функции интересные с точки зрения графического представления. На одном графике размещать несколько функций.