ФГБОУ ВО

Уфимский государственный авиационный технический университет

Кафедра ВМиК

Отчет

по лабораторной работе №5

Тема: «Введение в систему статистического анализа данных R»

Выполнили: ст. гр. ПРО-411

Букатина В.В. и Кабирова Л.Р.

Проверила: Харисова Э. А.

Уфа 2019

**Цель:** приобрести навыки работы с системой статистического анализа данных R.

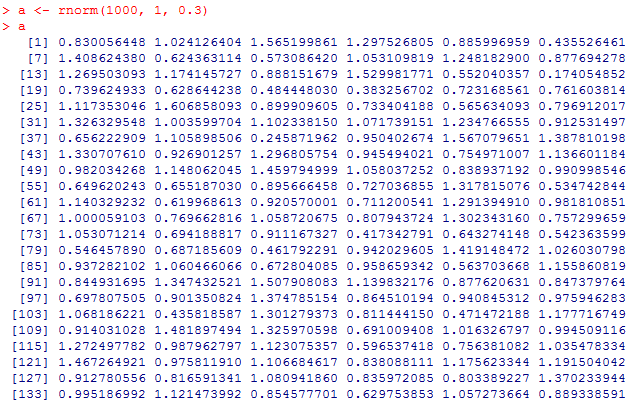
**Задачи:**

1. Сгенерируйте вектор длины N=1000, элементами которого являются реализации нормально распределенной случайной величины с математическим ожиданием, равным 1, и стандартным отклонением, равным 0.3. Подсчитайте статистическое мат. ожидание и стандарную ошибку, не используя встроенные функции и проверьте правильность результата. Подсчитайте .95,.99-квантили. Исследуйте отклонение статистического мат. ожидания от 1 при росте N (N=1000,2000,4000,8000).
2. Создайте фрейм данных из N=20 записей со следующими полями: Nrow — номер записи, Name — имя пользователя, BirthYear — год рождения, EmployYear — год приема на работу, Salary — зарплата, где Nrow изменяется от 1 до N, Name задается произвольно, BithYear распределен равномерно на отрезке [1960,1985], EmployYear распределен равномерно на отрезке [BirthYear+18,2006], Salary для работников младше 1975 г.р. определяется по формуле Salary=(ln(2007-EmployYear)+1)\*8000, для остальных Salary=(log2(2007-EmployYear)+1)\*8000. Подсчитайте число сотрудников с зарплатой, большей 15000. Добавьте в таблицу поле, соответствующее суммарному подоходному налогу (ставка 13%), выплаченному сотрудником за время работы в организации, если его зарплата за каждый код начислялась согласно формулам для Salary, где вместо 2007 следует последовательно подставить каждый год работы сотрудника в организации.
3. Напишите функцию, которая принимает на вход числовой вектор x и число разбиений интервала k (по умолчанию равное числу элементов вектора, разделенному на 10) и выполняет следующее: находит минимальное и максимальное значение элементов вектора xmin и xmax, разделяет полученный отрезок [xmin;xmax] на k равных интервалов и подсчитывает число элементов вектора, принадлежащих каждому интервалу. Далее должен строиться график, где по оси абсцисс — середины интервалов, по оси ординат — число элементов вектора, принадлежащих интервалу, разделенное на общее число точек. Проведите эксперимент на данной функции, где x — вектор длины 5000, сгенерированный из экспоненциально распределенной случайной величины, k=500. Приближение какого графика мы получаем в итоге при большом числе точек и числе разбиений?
4. Спроектируйте и реализуйте метод наименьших квадратов.

**Ход работы:**

1)Для генерации вектора длины N=1000, элементами которого являются реализации нормально распределенной случайной величины с математическим ожиданием, равным 1, и стандартным отклонением, равным 0.3, будем использовать функцию rnorm(1000, 1, 0.3).

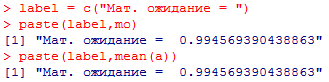
Функция rnorm(n) создает вектор указанной длины n со случайными элементами, распределенными по нормальному закону с математическим ожиданием 0 и дисперсией 1.



*Рисунок 1 - результат выполнения функции rnorm(1000, 1, 0.3)*

Математическое ожидание — среднее значение случайной величины при стремлении количества выборок или количества её измерений (иногда говорят — количества испытаний) к бесконечности. Для вектора (последовательности чисел) математическое ожидание считается как среднее арифметическое по формуле:

Для подсчета статистического мат. ожидания напишем простую формулу mo = sum(a)/length(a). Функция sum(a) — сумма элементов вектора a, а length(a) — длина вектора a. Затем сравним ее результат с результатом выполнения встроенной формулы mean (mean(a) — среднее значение элементов вектора a (оценка математического ожидания)).

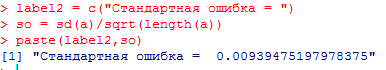


*Рисунок 2 - подсчет и вывод статистического мат. ожидания*

Результаты совпали, значит, подсчет выполнен верно. Можно переходить к следующему шагу.

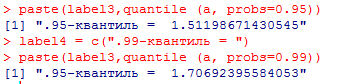
Стандартная ошибка среднего в математической статистике — величина, характеризующая стандартное отклонение выборочного среднего, рассчитанное по выборке размера n из генеральной совокупности. Стандартная ошибка вычисляется по формуле:

Для подсчета стандартной ошибки также напишем формулу so = sd(a)/sqrt(length(a)). Функция sd(a) — стандартное отклонения, а sqrt(a) — квадратный корень из a. Результат ее выполнения представлен на рис. 3.



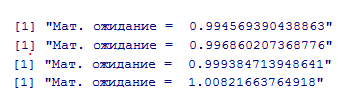
*Рисунок 3 - подсчет и вывод стандартной ошибки*

Далее подсчитаем .95 и .99-квантили. Для этого воспользуемся функцией quantile, которая возвращает значения квантилей для выборки, в качестве атрибута доверительной вероятности probs зададим 0.95 и 0.99 соответственно. Квантиль в [математической статистике](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0) — значение, которое заданная [случайная величина](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D0%B0%D1%8F_%D0%B2%D0%B5%D0%BB%D0%B8%D1%87%D0%B8%D0%BD%D0%B0) не превышает с фиксированной [вероятностью](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C).



*Рисунок 4 - подсчет .95 и .99-квантилей*

Теперь исследуем отклонение статистического мат. ожидания от 1 при росте N (N=1000,2000,4000,8000). Вычислим у этих распределений мат. ожидание по уже использующейся выше формуле sum(a)/length(a).

**

*Рисунок 5 - исследование отклонения статистического мат. ожидания*

Итак, полученные результаты свидетельствуют о том, что чем больше длина вектора, тем ближе к 1 его математическое ожидание.

**Листинг выполнения задания 1:**

a <- rnorm(1000, 1, 0.3)

mo = sum(a)/length(a)

so = sd(a)/sqrt(length(a))

quantile (a, probs = 0.95)

quantile (a, probs = 0.99)

a <- rnorm(2000, 0.6, 0.3)

sum(a)/length(a)

a <- rnorm(4000, 0.6, 0.3)

sum(a)/length(a)

a <- rnorm(8000, 0.6, 0.3)

sum(a)/length(a)

2) Для того, чтобы создать фрейм, необходимо сначала сформировать поля для него. Первое поле Nrow — номер записи, принимает значение от 1 до 20. То есть, Nrow <- c(1:20).

Далее Name — имя пользователя, зададим имена заглавными буквами английского алфавита, то есть Name <- c('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T').

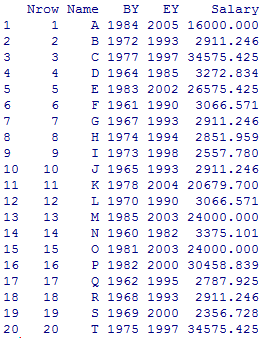
Следующее поле BithYear, его значения распределены равномерно на отрезке [1960,1985]. Сформируем его так: BY <- sample(20, 1960,1985). Функция sample используется для генерации целых чисел в диапазоне [min, max].

Далее идет поле EmployYear, его значения распределены равномерно на отрезке [BirthYear+18,2006]. Формируем его аналогично - EY <- sample(20, BY+18, 2006).

Теперь поле Salary: для работников младше 1975 г.р. определяется по формуле Salary=(ln(2007-EmployYear)+1)\*8000, для остальных Salary=(log2(2007-EmployYear)+1)\*8000. То есть, Salary <- ifelse(BY<1975, (log(2007-EY)+1)\*800, (log2(2007-EY)+1)\*8000). Команда ifelse() воспринимает первый аргумент как условие, второй аргумент возвращается если условие верно, а третий аргумент - если нет.

Добавим все эти поля в фрейм frm и выведем на экран следующим образом: frm <- data.frame(Nrow, Name, BY, EY, Salary), frm.

Функция data.frame создает фрейм данных, содержащий векторы, заданные ранее.



*Рисунок 6 - создание фрейма frm с 5 полями (Nrow, Name, BY, EY, Salary)*

Подсчитаем число сотрудников с зарплатой, большей 15000. Для этого добавим еще одну колонку Result, содержимым которой будут 0 и 1 (если зарплата работника будет больше 15000, то значение будет 1, а иначе 0). Сделаем это следующим образом:

Result <- ifelse(Salary>15000, 1, 0)

frm <- data.frame(Nrow, Name, BY, EY, Salary, Result)

frm

paste("Количество работников с заработной платой более 15 тысяч - ", sum(Result))

Функция paste осуществляет конкатенацию строк.

**

*Рисунок 7 - вывод количества сотрудников с зарплатой более 15000*

Теперь добавим в таблицу поле, соответствующее суммарной зарплате, выплаченной сотруднику за время работы в организации. Сделаем это следующим образом:

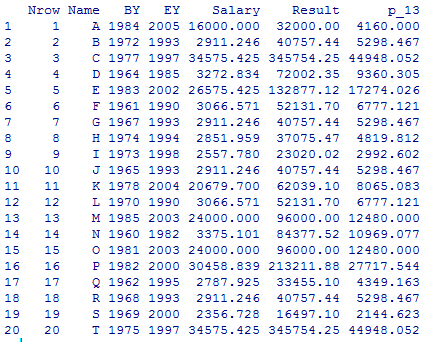
Result <- (2007 - EY) \* Salary

Далее добавим в таблицу поле, соответствующее суммарному подоходному налогу (ставка 13%), выплаченному сотрудником за время работы в организации, если его зарплата за каждый код начислялась согласно формулам для Salary, где вместо 2007 следует последовательно подставить каждый год работы сотрудника в организации. Сделаем это следующим образом:

p\_13 <- 0.13 \* (2007 - EY) \* Salary

frm <- data.frame(Nrow, Name, BY, EY, Salary, Result, p\_13)

frm



*Рисунок 8 - подсчет суммарной заработной платы сотрудника за все годы работы и*

*подсчет суммарного подоходного налога*

Итак, было подсчитано число сотрудников с зарплатой более 15 тыс., суммарный подоходный налог и суммарная заработная плата сотрудника за все годы работы на предприятии.

**Листинг выполнения задания 2:**

Nrow <- c(1:20)

Name <- c('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T')

BY <- sample(1960:1985,20)

EY <- floor(runif(20, min=BY+18, max=2006))

Salary <- ifelse(BY<1975, (log(2007-EY)+1)\*800, (log2(2007-EY)+1)\*8000)

Result <- ifelse(Salary>15000, 1, 0)

paste("Количество работников с заработной платой более 15 тысяч - ", sum(Result))

Result <- (2007 - EY) \* Salary

p\_13 <- 0.13 \* (2007 - EY) \* Salary

frm <- data.frame(Nrow, Name, BY, EY, Salary, Result, p\_13)

frm

3) Сначала найдем минимальное и максимальное значение элементов вектора xmin и xmax с помощью функций min и max (min(v) — минимальный элемент, max(v) — максимальный элемент):

xmin = min(x)

xmax = max(x)

Теперь сформируем отрезок [xmin;xmax] следующим образом: result = x[c(which.xmin:which.xmax)] (то есть новый вектор result будет содержать элементы из вектора x начиная с xmin до xmax).

Далее разделим полученный отрезок на k равных интервалов. Для этого напишем свою функцию chunk = function(x,n) split(x, factor(sort(rank(x)%%n))). Здесь используется ряд функций, опишем их:

* split() разбивает строку по шаблону. Его формат: split (<Исходная строка>[, <Лимит>]);
* factor() - преобразует числовой вектор в фактор в R;
* sort() на месте сортирует элементы массива и возвращает отсортированный массив;
* rank() возвращает вектор с "рангом" каждого значения;
* %% - модуль (остаток от деления).

Вызовем эту функцию так - chunk(result,k). Дальнейшие вычисления, а именно подсчет числа элементов вектора, принадлежащих каждому интервалу (summa), середин интервалов (boo) и чисел элементов вектора, принадлежащих интервалу, разделенное на общее число точек (foo), будем проводить в цикле. Напишем следующий код:

for(i in 0:(k-1)) {

summa = sum(chunk(result,k)$`i`)

boo = (max(chunk(result,k)$`i`)+min(chunk(result,k)$`i`))/2

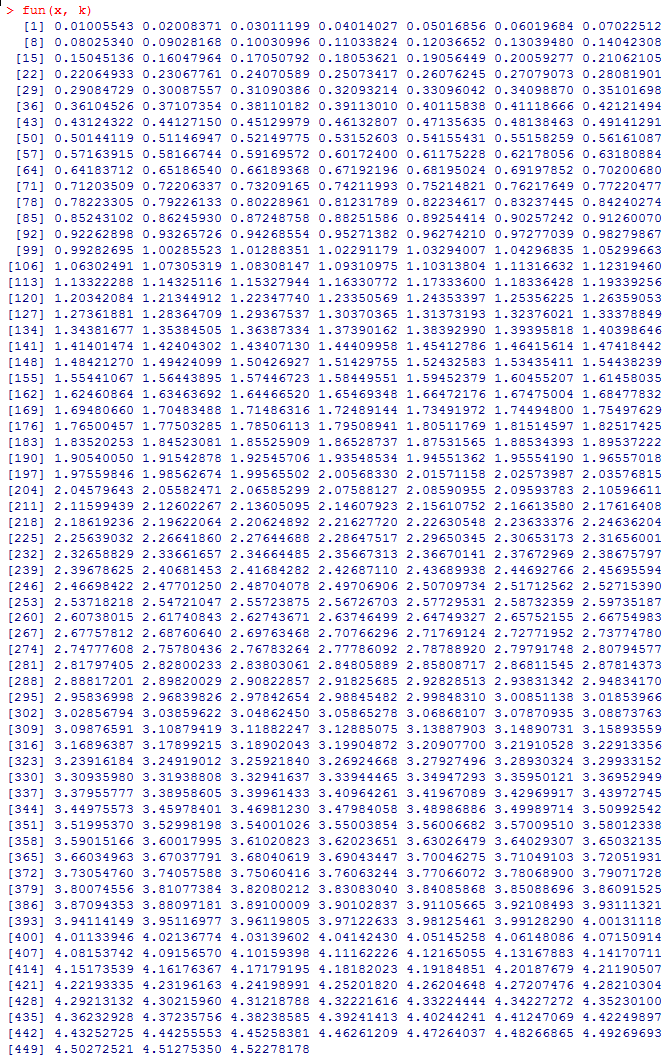
foo = (max(chunk(result,k)$`i`)-min(chunk(result,k)$`i`))/sum(x)

}

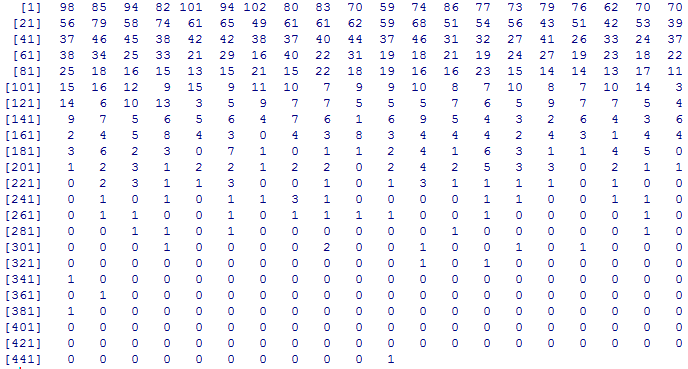
Функции, использующиеся в данном куске кода, были описаны ранее.

И наконец построим график, где по оси абсцисс — середины интервалов, по оси ординат — число элементов вектора, принадлежащих интервалу, разделенное на общее число точек. Для изображения графиков функций в R есть функция plot(x, y). Здесь x — вектор значений абсцисс и y — вектор значений ординат. Таким образом, это реализуется функцией plot(boo, foo).

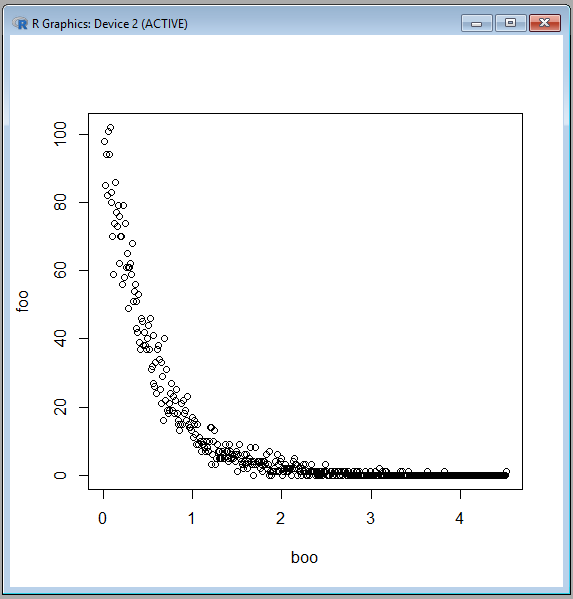
Теперь проверим работоспособность нашей функции на экспоненциально распределенной случайной величине. Для этого используем функцию rexp() (генерация экспоненциального распределения) и зададим количество элементов 5000, то есть rexp(5000).



*Рисунок 9 - подсчет середины интервалов экспоненциально распределенной случайной величины из 5000 элементов*

**

*Рисунок 10 - подсчет чисел элементов вектора, принадлежащих интервалу, разделенное на общее число точек*



*Рисунок 11 - результат построения графика функцией*

Итак, при большом числе точек и числе разбиений мы получаем в итоге приближение графика к показательному (экспоненциальному) виду.

**Листинг выполнения задания 3:**

fun <- function(x, k)

{

xmin = min(x)

xmax = max(x)

result = x[c(which.xmin:which.xmax)]

chunk = function(x,n) split(x, factor(sort(rank(x)%%n)))

chunk(result,k)

for(i in 0:(k-1)) {

summa = sum(chunk(result,k)$`i`)

boo = (max(chunk(result,k)$`i`)+min(chunk(result,k)$`i`))/2

boo

foo = (max(chunk(result,k)$`i`)-min(chunk(result,k)$`i`))/sum(x)

foo

}

plot(boo, foo)

}

4) Метод наименьших квадратов (МНК) — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных. Он может использоваться для «решения» переопределенных систем уравнений (когда количество уравнений превышает количество неизвестных), для поиска решения в случае обычных (не переопределенных) нелинейных систем уравнений, для аппроксимации точечных значений некоторой функции. МНК является одним из базовых методов регрессионного анализа для оценки неизвестных параметров регрессионных моделей по выборочным данным.

Задача заключается в нахождении коэффициентов линейной зависимости, при которых функция двух переменных *а* и *b* принимает наименьшее значение. То есть, при данных *а* и *b* сумма квадратов отклонений экспериментальных данных от найденной прямой будет наименьшей. В этом вся суть метода наименьших квадратов.

Таким образом, решение примера сводится к нахождению экстремума функции двух переменных.

В случае, если система уравнений имеет решение, то наименьшее значение суммы квадратов будет равно нулю и могут быть найдены точные решения системы уравнений аналитически или, например, различными численными методами оптимизации. Если система переопределена, то есть, говоря нестрого, количество независимых уравнений больше количества искомых переменных, то система не имеет точного решения и метод наименьших квадратов позволяет найти некоторый «оптимальный» вектор *x* в смысле максимальной близости векторов *y* и или максимальной близости вектора отклонений к нулю (близость понимается в смысле евклидова расстояния).

Экспериментальные данные о значениях переменных х и у представляют собой годовую доходность обыкновенных акций ПАО Сбербанк:

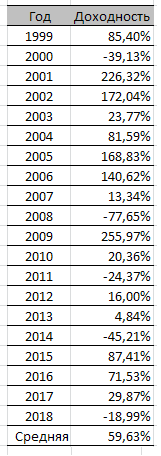


Рисунок 12 - годовая доходность обыкновенных акций ПАО Сбербанк

Добавим их в R следующим образом:

*x <- c(1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018)*

*y <- c(85.40,-39.13,226.32,172.04,23.77,81.59,168.83,140.62,13.34,-77.65,255.97,20.36,-24.37,16.00,4.84,-45.21,87.41,71.53,29.87,-18.99)*

Затем построим график зависимости y от x с помощью функции plot (y,x).

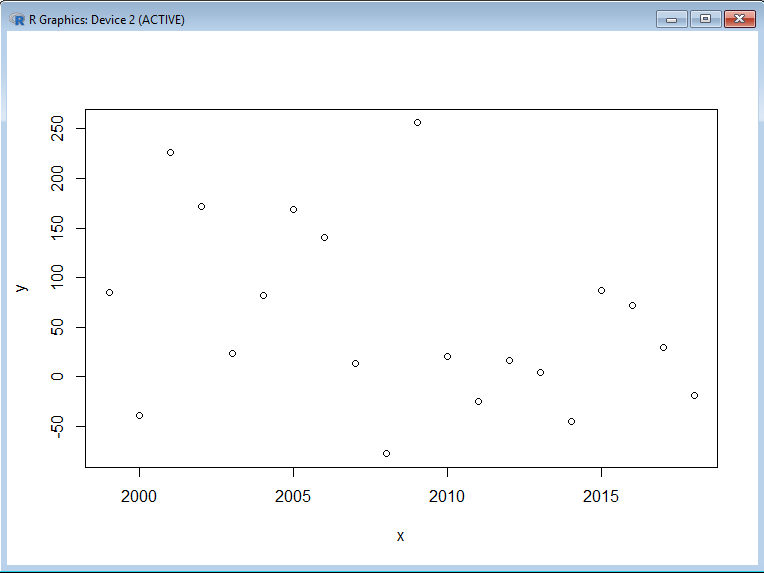
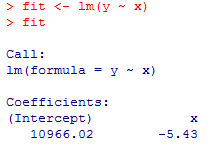


Рисунок 13 - график годовой доходности обыкновенных акций ПАО Сбербанк

Вычислим параметры регрессионного уравнения с помощью функции fit. То есть, fit <- lm(y~x).



*Рисунок 14* - *вычисление параметров регрессионного уравнения*

Следовательно, регрессионное уравнение принимает вид:

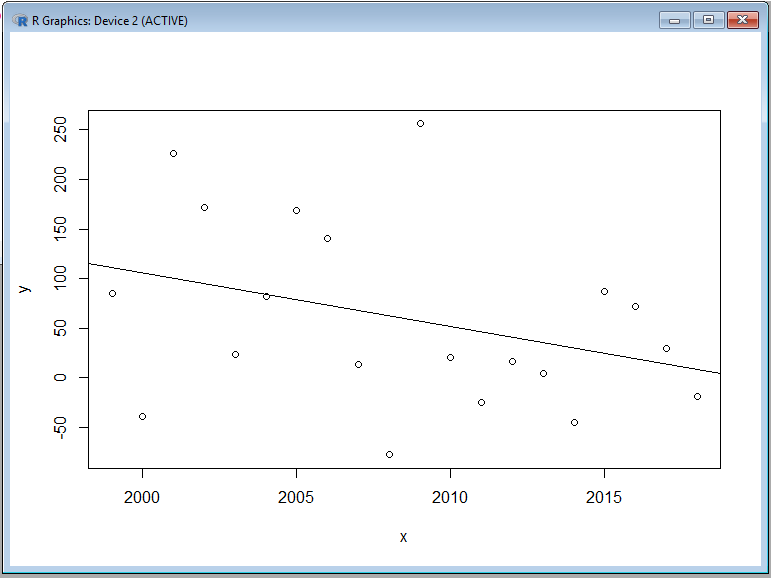


Рисунок 15 - график статистики ожидаемой годовой доходности обыкновенных акций ПАО Сбербанк и её прямая аппроксимации, найденная МНК

Итак, мы спроектировали и реализовали метод наименьших квадратов в R, а также произвели эксперимент на собственных данных.

**Листинг выполнения задания 4:**

*x <- c(1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018)*

*y <- c(85.40,-39.13,226.32,172.04,23.77,81.59,168.83,140.62,13.34,-77.65,255.97,20.36,-24.37,16.00,4.84,-45.21,87.41,71.53,29.87,-18.99)*

plot(x, y) #график зависимости y от x

fit <- lm(y ~ x) #нахождение коэффициентов регрессионного уравнения

fit

abline(fit) #рисование прямой аппроксимации

**Вывод по лабораторной работе:**

При выполнении лабораторной работы были приобрести навыки работы с системой статистического анализа данных R. Также было выполнено задание для самостоятельной работы в соответствии с руководством по выполнению лабораторной работы и оформлен отчет о выполнении лабораторной работы в соответствии с требованиями к его оформлению.