
Московский государственный университет имени М. В. Ломоносова
Факультет Вычислительной математики и кибернетики

Никонов Максим Викторович
316 группа

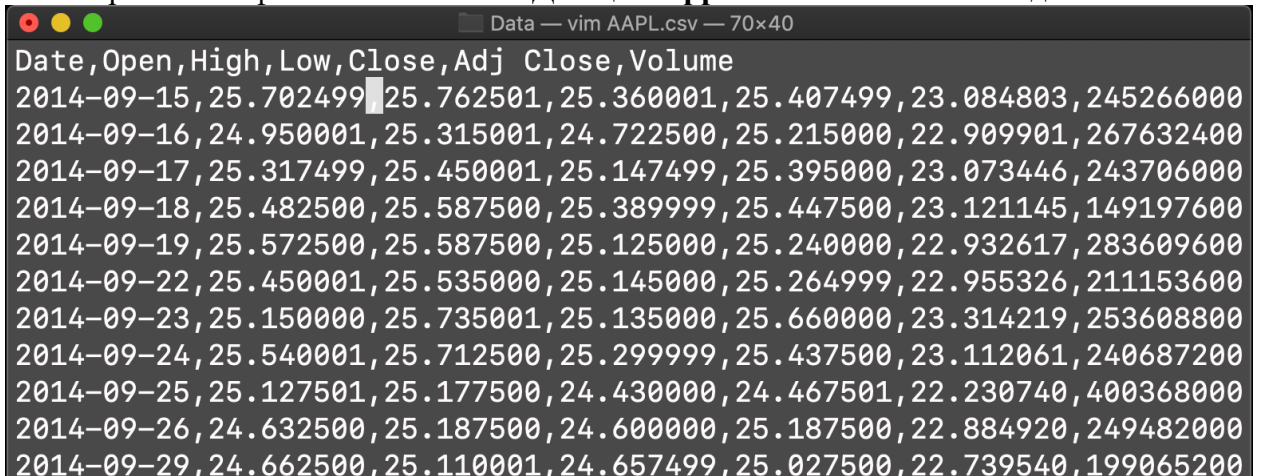
2020

Задание 1: Используя команды `read.table()` и `read.csv()`, а также `file.choose()`, загрузить данные из TXT и CSV-файлов и создать из некоторого их подмножества список и таблицу.

```
head <- read.csv("/Users/Nikon/Desktop/CMC MSU/MC/5  
sem/Data/AAPL.csv",  
               header = TRUE)
```

Читаем в переменную `head` данные из файла `AAPL.csv`. В кавычках путь к файлу. Параметр `header` учитывает заголовок в файле. По умолчанию = `TRUE`

Посмотрим часть файла `AAPL.csv` – БД акций Apple INC за 2014-2020 год:



```
Data — vim AAPL.csv — 70x40  
Date,Open,High,Low,Close,Adj Close,Volume  
2014-09-15,25.702499,25.762501,25.360001,25.407499,23.084803,245266000  
2014-09-16,24.950001,25.315001,24.722500,25.215000,22.909901,267632400  
2014-09-17,25.317499,25.450001,25.147499,25.395000,23.073446,243706000  
2014-09-18,25.482500,25.587500,25.389999,25.447500,23.121145,149197600  
2014-09-19,25.572500,25.587500,25.125000,25.240000,22.932617,283609600  
2014-09-22,25.450001,25.535000,25.145000,25.264999,22.955326,211153600  
2014-09-23,25.150000,25.735001,25.135000,25.660000,23.314219,253608800  
2014-09-24,25.540001,25.712500,25.299999,25.437500,23.112061,240687200  
2014-09-25,25.127501,25.177500,24.430000,24.467501,22.230740,400368000  
2014-09-26,24.632500,25.187500,24.600000,25.187500,22.884920,249482000  
2014-09-29,24.662500,25.110001,24.657499,25.027500,22.739540,199065200
```

В первой строке указываются имена столбцов через запятую, что потом отразится на БД

Возможность: Существует возможность загрузки данных не из локальной директории, посредством интернета. Я выбрал вариант расположения БД на машине, т.к буду работать с этой БД весь семестр, поэтому анализировать вне зависимости от наличия интернета тоже было бы благоприятно.

```
num <- read.table(...table.txt)
```

В переменную `num` считывает таблицу на столько же строк, сколько содержит `AAPL.csv`, содержит числа от 1 до `n`

Посмотрим часть файла table.txt:

```
Data — vim table.txt — 70x40
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

Решение взять столбец – для того, чтобы можно было проиндексировать без непосредственной стандартной индексации, в дальнейшем БД будет создаваться с table.txt, можно было бы и обойтись без этого.

Отличия read.csv и read.table из наблюдений:

функция	формат	парсинг	популярность
read.csv	csv	,	да
read.table	txt	пробел	нет

```
head <- cbind(head,num)
data <- head[-c(2,3,4,6)]
```

Используя уже известные функции, склеиваю данные из table.txt и AAPL.csv. В итоговые данные беру все, кроме столбцов 2,3,4,6

Используемые packages. Нет

Тестирование. Не требует

Неразрешенные вопросы. Нет

Новые функции. read.csv, choose.file(), read.table

Статус компиляции. ОК. Данные из протокола:

```
> ^M
> ^M
bash-3.2$ exit^M
exit^M
```

Задание 2: Нарисовать график и гистограмму для полученных данных

В начале программы идет стандартный макросс для создания БД. Далее рассматривать и возвращаться к этому мы не будем.

```
head <- read.csv("/Users/Nikon/Desktop/CMC MSU/MC/5 sem/Data/AAPL.csv",
  header = TRUE)
num <- read.table("/Users/Nikon/Desktop/CMC MSU/MC/5 sem/Data/table.txt")
head <- cbind(head, num)
data <- head[-c(2,3,4,6)]
```

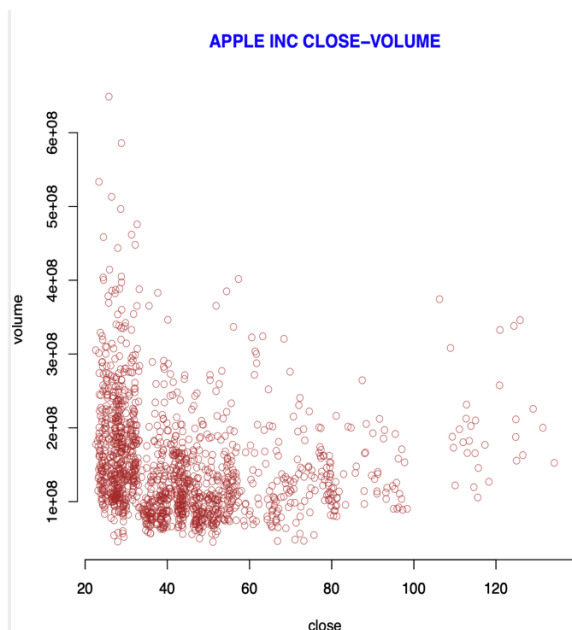
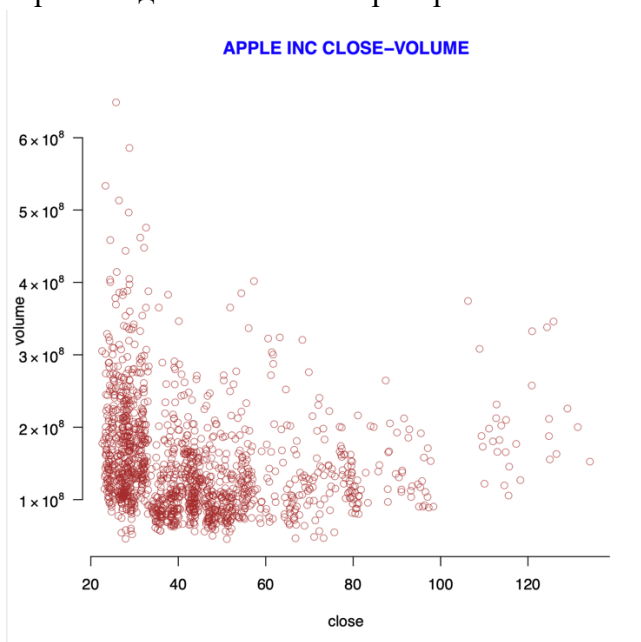
```
data$Date <- as.POSIXct(data$Date)
format(data$Date,"%Y")
```

Переформатирование столбца даты в формат даты, по стандарту – числовой.

```
plot(data$Close,data$Volume,
  main = "APPLE INC CLOSE-VOLUME", xlab = "close", ylab =
  "volume",
  col.main = "blue",type = "p", lwd = 0.5, col = "brown",
  axes = FALSE, frame.plot = FALSE)
axis(1,at = seq(20,140,20))
axis(2,at = my.at, col.axis = "black", las = 1,
  labels = as.expression(lapply(e.y,function(E) bquote(. (E) %*%
  10.^8))))
```

Функция plot имеет много параметров, поясним неочевидные: axes = FALSE позволяет самому задать форматы осей, отключая формат осей по умолчанию.

axis формирует оси (1 параметр – ось (1,2), остальные параметры отвечают за задание формата), например для отображения чисел не в экспоненциальном виде – последние 2 строки кода выше. Посмотрим различия:



Во 2 примере не используем axis и в plot параметр axes = TRUE (по умолчанию)

Рассмотрим подробно каждый построенный график:

```
plot(data$Date, data$Close,
      main = "APPLE INC
TIME-CLOSE", xlab = "date",
      ylab = "close",
      col.main = "blue", type
= "l", lwd = 0.5, col =
"black",
      panel.first =
lines(stats::lowess(data$Date,
data$Close), lty = "dashed"),
      axes = TRUE,
      frame.plot = TRUE)
```

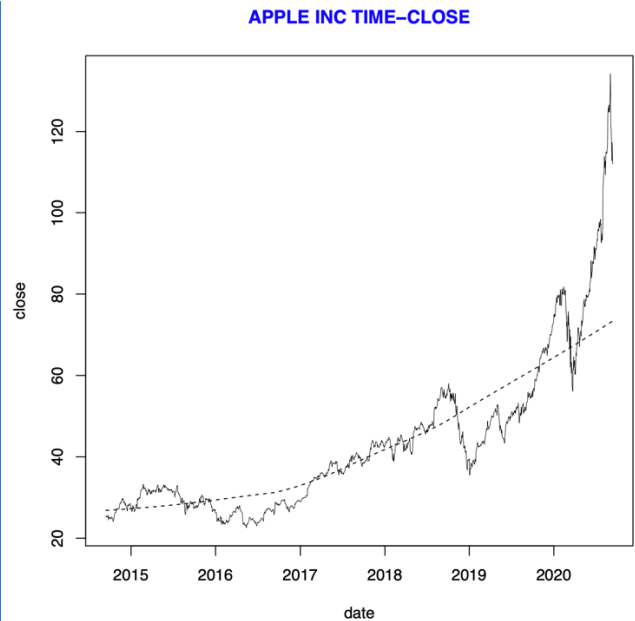


График зависимости конечной цены от времени (аппроксимируя виден рост цены)

```
plot(data$Date, data$Volume,
      main = "APPLE INC TIME-
VOLUME", xlab = "date", ylab =
"volume",
      col.main = "blue", type
= "l", lwd = 0.5, col =
"black",
      axes = FALSE,
      frame.plot = TRUE)
e.y = c(1:6); my.at <- e.y*10^8
axis(1, at = data$Date,
format(data$Date, "%Y"))
axis(2, at = my.at, col.axis =
"black", las = 1,
      labels =
as.expression(lapply(e.y, functi
on(E) bquote.(E) %*% 10.^8))))
```

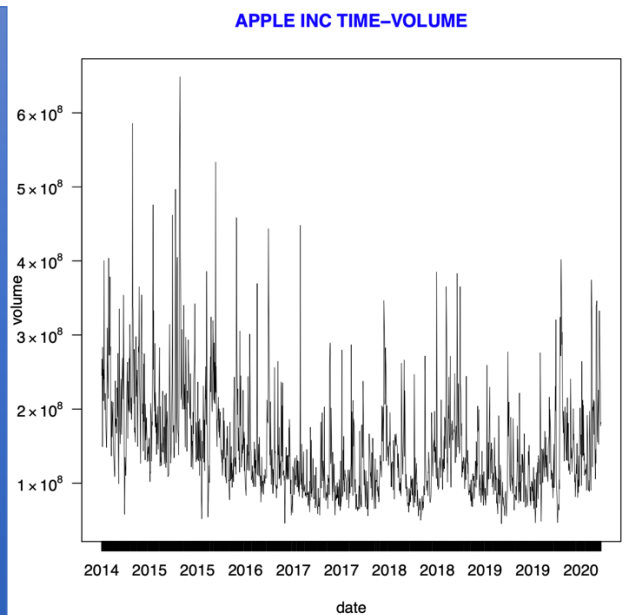


График зависимости объёма продаж от времени (пики спровоцированы не только ценой, но и другими факторами)

```

plot(data$Close,data$Volume,
      main = "APPLE INC
CLOSE-VOLUME", xlab = "close",
ylab = "volume",
      col.main = "blue",type
= "p", lwd = 0.5, col =
"brown",
      axes = FALSE,
frame.plot = FALSE)
axis(1,at = seq(20,140,20))
axis(2,at = my.at, col.axis =
"black", las = 1,
      labels =
as.expression(lapply(e.y,function(E) bquote(. (E) %*% 10.^8))))

```

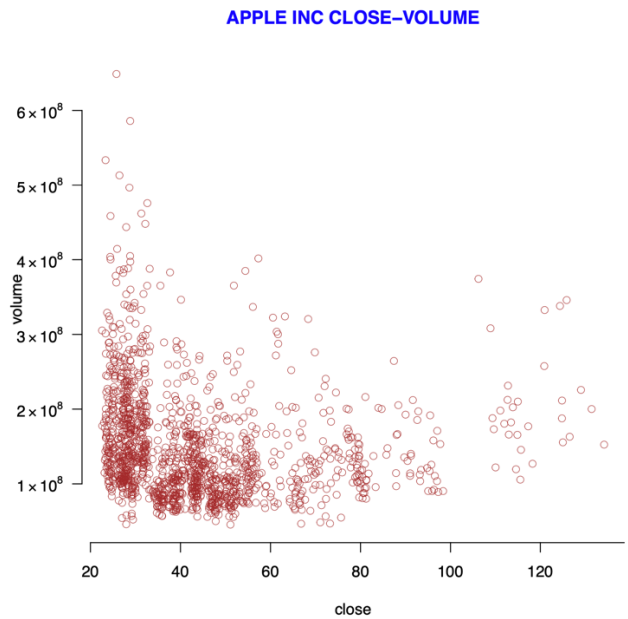


График зависимости объёма продаж от конечной цены (плотность точек уменьшается при отдалении от начала координат)

```

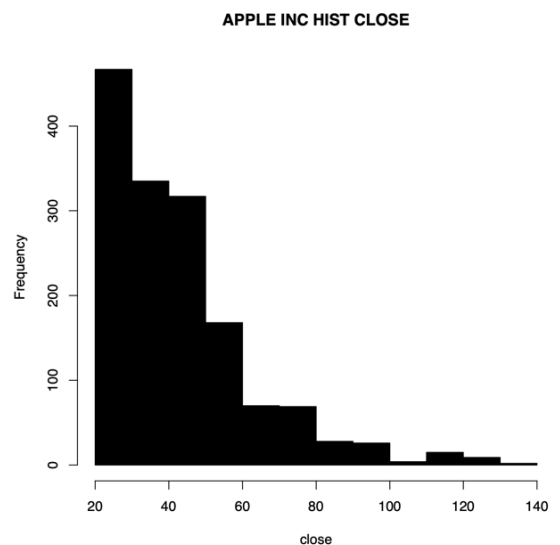
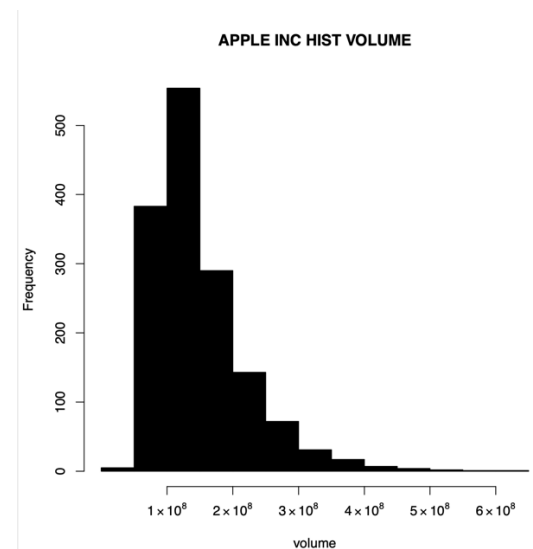
hist(data$Volume, main =
"APPLE INC HIST VOLUME", col =
"black",
      axes = FALSE, xlab =
"volume")
axis(2,at = seq(0,500,100))
axis(1,at = my.at, col.axis =
"black", las=1,
      labels =
as.expression(lapply(e.y,function(E) bquote(. (E) %*%
10.^8))))
hist(data$Close, main = "APPLE
INC

```

```

HIST CLOSE", col = "black",
xlab = "close")

```



Гистограмма распределения конечной цены и объёма продаж

Используемые packages. Нет

Тестирование. Не требует

Неразрешенные вопросы. Отображение форматы даты в формате %Y

Новые функции. plot, axis, hist, as.POSIXct

Статус компиляции. ОК. Данные из протокола:

```
> ^M
bash-3.2$ exit^M
exit^M
```

Script done on Tue Oct 6 16:40:01 2020

Аппроксимация или приближение — научный метод, состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми. Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов.

Гистограмма — это способ сводного представления набора категориальных данных. Сведения на **гистограмме** отображаются с помощью нескольких столбцов одной ширины, каждый из которых представляет определенную категорию.

Задание 3: С помощью условного оператора if-else и циклов for, while, repeat проверить условия неотрицательности данных, а также попадания в некоторый диапазон для всех элементов списка и таблицы.

Рассмотрим синтаксис for и if:

for (x in data)	if (op)
body	body

Решение задачи – в условии проверка на число, проверка на диапазон (0;200)

```
for (elem in data){
  if (elem > 0 && is.numeric(elem) && elem<200){
    print(elem)
  }
}
```

Цикл **for** схож по синтаксису с аналогичным циклом в Python3.

Поддерживает операторные скобки { но можно без них (в точности с Python3)

Необходимый критерий data – перечислимый тип

Условный оператор схож с аналогичным оператором в Python3, операции И/ИЛИ имеют высокий приоритет.

Условный оператор else, циклы while, repeat имеют уже знакомый нам синтаксис и дополнительного ознакомления не требуют.

Используемые packages. Нет

Тестирование. Не требует

Неразрешенные вопросы. Нет

Новые функции: for, if (цикл и оператор)

Статус компиляции. ОК. Данные из протокола:

```
> ^M
bash-3.2$ exit^M
exit^M
```

Script done on Tue Oct 6 16:30:26 2020