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

Никонов Максим Викторович
316 группа

2020

Задание 1: Написать функцию, которая выполняет загрузку данных и их рисование (аналогично заданию с прошлого семинара)

Синтаксис функции :

```
function <- name(param)
{
  body
  return (smt)
}
```

Разбиение на функции проводятся только по логическим блокам, разбиение на блоки иным образом недопустимо.

Логические блоки программы 2 с прошлого семинара:

- Инициализация БД
- Печать таблицы и имен столбцов
- Построение Графиков (требуется дополнительного разбиения на логические блоки)
- Построение Гистограмм (не требует разбиения на отдельные логические блоки в частном порядке)

Рассмотрим функции:

```
create_table <- function(data2)
{
  num <-
  read.table("/Users/Nikon/Desktop/CMC
MSU/MC/5 sem/Data/table.txt")
  data2 <- cbind(data2, num)
  data_all <- data2[-
c(2,3,4,6)]
  return (data_all)
}
```

Инициализация БД с прошлого семинара

```
print_names_and_data <- function(data)
{
    print(data)
    print(names(data))
}
```

Печать БД и имен столбцов

```
plot_data_time_close <- function(data)
{
    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)
}
```

Построение графика цены от времени

```
plot_data_time_volume <- function(data)
{
    plot(data$Date, data$Volume,
        main = "APPLE INC TIME-VOLUME", xlab = "date", ylab =
"volume",
        col = "black", type = "l", lwd = 0.5,
        axes = FALSE, frame.plot = TRUE)
    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, function(E) bquote{.(E) %*%
10.^8})))
}
```

Построение графика объема продаж от времени

```
plot_data_close_volume <- function(data)
{
    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})))
}
```

Построение графика объема продаж от цены

```
hists_data <- function(data)
{
    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")
}
```

Построение ДВУХ гистограмм

```
print_names_and_data(data)
plot_data_time_close(data)
plot_data_time_volume(data)
plot_data_close_volume(data)
hists_data(data)
```

Вызов функций

Используемые packages. Нет

Тестирование. Не требует

Неразрешенные вопросы. Повтор. Отображение форматы даты в формате %Y

Новые функции: function

Статус компиляции. ОК. Данные из протокола:

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

Script done on Tue Oct 6 16:42:49 2020

Задание 2: Продемонстрировать визуализацию данных на одном листе для различных комбинаций настроек функции plot (использовать максимальный набор настраиваемых параметров)

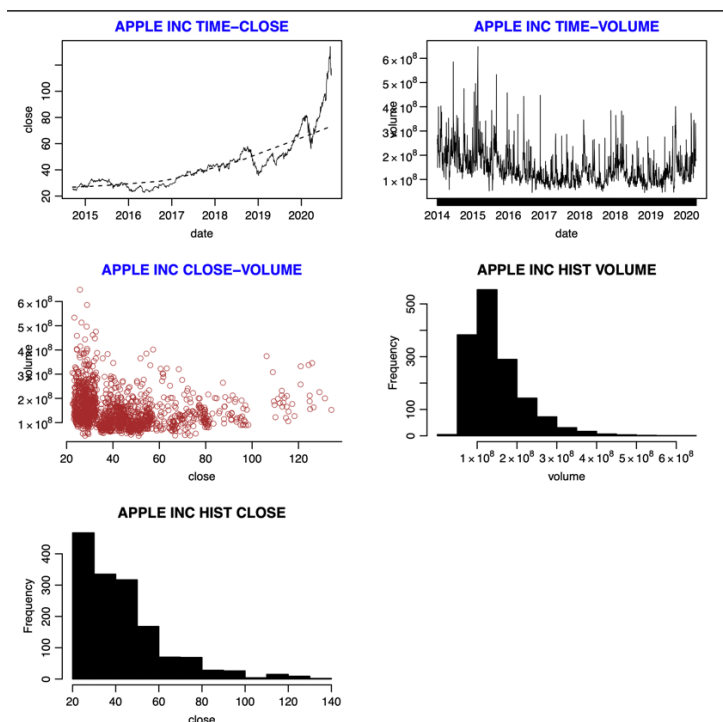
Перед построением графиков и гистограмм необходимо прописать строку:

```
op <- par(mfrow = c(3,2), mar = par("mar")-c(1,0,2,0), mgp = c(2, .7, 0))
```

Далее после функций plot и hist прописываем :

par(op)

В следствии имеем объединение графиков на одном листе :



Используемые packages. Нет

Тестирование. Не требует

Неразрешенные вопросы. Повтор. Отображение формата даты в формате %Y

Новые функции: par

Статус компиляции. ОК. Данные из протокола:

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

Script done on Tue Oct 6 16:44:19 2020

Задание 3: Реализовать векторную обработку данных с помощью функций apply / lapply / sapply / vapply / mapply / rapply / tapply.

apply(data, 2, mean)	Среднее значение по каждому столбцу	
apply(data, 2, function(x) length(x[x<0]))	Количество отрицательных значений	
apply(data, 2, function(x) is.matrix(x))	Каждый столбец матрица?	
sapply(data[c(1,500,1000),2], function(x) x^2)	Возводит вектор в квадрат и возвращает	Вектор
lapply(data[c(1,500,1000),2], function(x) x^2)		Список

Остальные функции имеют схожее логическое направление и демонстрации не требуют. Остается на самостоятельное изучение вне данного отчета

Используемые packages. Нет

Тестирование. Не требует

Неразрешенные вопросы. Нет

Новые функции: apply, sapply, lapply, vapply, mapply, rapply, tapply

Статус компиляции. ОК. Данные из протокола:

```
[1] "Close" "Volume" ^M
> apply(data, 2, mean) ^M
Close Volume ^M
4.344185e+01 1.486636e+08 ^M
> apply(data, 2, function(x) length(x[x<0])) ^M
Close Volume ^M
0 0 ^M
> apply(data, 2, function(x) is.matrix(x)) ^M
Close Volume ^M
FALSE FALSE ^M
> apply(data, 2, is.vector) ^M
Close Volume ^M
TRUE TRUE ^M
> sapply(data[c(1,500,1000),2], function(x) x^2) ^M
[1] 6.015541e+16 2.871574e+16 3.005383e+16 ^M
> lapply(data[c(1,500,1000),2], function(x) x^2) ^M
[[1]] ^M
[1] 6.015541e+16 ^M
^M
[[2]] ^M
[1] 2.871574e+16 ^M
^M
[[3]] ^M
[1] 3.005383e+16 ^M
^M
> #возвращает список ^M
> ^M
bash-3.2$ exit ^M
exit ^M
```

Script done on Tue Oct 6 16:45:09 2020