

# Групиране и обхождане на данни

## Статистическа обработка на данни с R

Пламен Петров и Тодор Балабанов

Център за обучение  
Институт по информационни и комуникационни технологии  
Българската академия на науките

*p.petrov@iit.bas.bg todorb@iinf.bas.bg*

25.V.2020

## Acknowledgments

These teaching materials are funded by Velbazhd Software LLC and it is partially supported by the Bulgarian Ministry of Education and Science (contract D01–205/23.11.2018) under the National Scientific Program “Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)”, approved by DCM # 577/17.08.2018.

Теми	Фамилията функции apply	Пакетът plyr	Пакетът data.table	Бързи операции с пакета dplyr	Пакетът purrr
●	○ ○ ○ ○ ○	○ ○○ ○ ○○	○ ○○ ○ ○	○ ○ ○○○ ○○○ ○○○ ○○○	○ ○○○○

# Съдържание

- 1 Фамилията функции apply
  - apply
  - lapply и sapply
  - mapply
  - Агрегация
- 2 Пакетът plyr
  - ddply
  - ddply
  - Помощни функции и бързодействие
- 3 Пакетът data.table
  - Разширяване на възможностите
  - Ключове
  - Агрегация
- 4 Бързи операции с пакета dplyr
  - Потоци и таблици
  - Извличане по колони
  - Филтриране
  - Модификация, обобщение, групиране и подреждане
  - Специфични изчисления и връзка с база данни
- 5 Пакетът purrr
  - Фамилията функции map
- 6 Заключение
  - Дискусия

# Фамилията функции apply



## Вектори и списъци

### Сума на обекти в списък

```
l1 <- list(m2=matrix(1:9,3), l2=1:5, m3=matrix(1:4,2), n1=2)
lapply(l1, sum)
sapply(l1, sum)
```

## Множество от списъци

### Проверка за идентичност на елементите

```
l3 <- list(m4=matrix(1:25,5), m5=matrix(1:16,2), l4=1:5)
l5 <- list(m6=matrix(1:25,5), m7=matrix(1:16,8), l6=15:1)
mapply(identical, l3, l5)
mapply(f1<-function(x,y){NROW(x)+NROW(y)}, l3, l5)
```

# Агрегация

## Групиране на данни

```
data(diamonds, package="ggplot2")
aggregate(price~cut, diamonds, mean)
aggregate(price~cut+color, diamonds, mean)
aggregate(cbind(price,carat)~cut, diamonds, mean)
aggregate(cbind(price,carat)~cut+color, diamonds, mean)
```



# Пакетът plyr

## Подготовка на данните

### Бейзболна статистика

```
library( plyr )
baseball$sf[baseball$year < 1954] <- 0
baseball$hbp[ is.na(baseball$hbp) ] <- 0
baseball <- baseball[baseball$ab>=50,]
baseball$OBP <- with(baseball, (h+bb+hbp)/(ab+bb+hbp+sf))
```

## Обработка на числител и делител

Пресмятане на OBP за цялата кариера на играча

```
career <- ddply(baseball, .variables="id",
.fun=function(data)c(OBP=with(data,sum(h+bb+hbp) /
sum(ab+bb+hbp+sf))))
career <- career[ order(career$OBP, decreasing=TRUE), ]
head(career, n=3)
```

## Групова обработка

Сума на всеки елемент в списък

```
l1 <- list(m2=matrix(1:9,3), l2=1:5, m3=matrix(1:4,2), n1=2)
llply(l1, sum)
identical(lapply(l1,sum), llply(l1,sum))
laply(l1, sum)
```

## Усложняване на агрегацията

### Повече от една агрегатна функция

```
library(ggplot2)
aggregate(price~cut, diamonds, each(mean, median))
```

## Намалена консумация на памет

### Бързодействие при използване на референции

```
system.time(dplyr(baseball, "id", nrow))
reference <- idata.frame( baseball )
system.time(dplyr(reference, "id", nrow))
```

# Пакетът data.table

## Реализация с вътрешно индексване

### Създаване на data.table

```
df <- data.frame(x1=10:1, x2=letters[11:20], x3=LETTERS[1:10],
x4=rep(c(" One", " Two", " Three"), length.out=10))
dt <- data.table(x1=10:1, x2=letters[11:20], x3=LETTERS[1:10],
x4=rep(c(" One", " Two", " Three"), length.out=10))
diamonds <- data.table( diamonds )
```





# Индексиране

## Операции с таблици

```
tables()
setkey(dt, x4)
key( dt )
setkey(diamonds, cut, color)
```

## Бързодействие заради индексирането

### Агрегатни функции

```
aggregate(price~cut, diamonds, mean)
diamonds[, list(price=mean(price)), by=cut]
diamonds[, list(price=mean(price)), by=list(cut,color)]
diamonds[, list(price=mean(price),carat=sum(carat)),
by=list(cut,color)]
```

# Бързи операции с пакета dplyr

# Верига от изчисления

## Поточни операции

```
library( ggplot2 )
library( magrittr )
dim( head(diamonds,n=4) )
diamonds %>% head(4) %>% dim
```

## Избор по редове по аналогия с релационните бази данни

### Избор на редове

```

diamonds %>% select(carat, price)
diamonds %>% select(c(carat, price))
diamonds %>% select_('carat', 'price')
names <- c('carat', 'price')
diamonds %>% select_(.dots=names)
diamonds %>% select( one_of('carat', 'price') )
names <- c('carat', 'price')
diamonds %>% select( one_of(names) )
select(diamonds, 1, 7)
diamonds %>% select(1, 7)

```

Теми	Фамилията функции apply	Пакетът plyr	Пакетът data.table	Бързи операции с пакета dplyr	Пакетът purrr
○	○ ○ ○ ○ ○	○ ○○ ○ ○○	○ ○○ ○ ○	○ ○ ○●○ ○○○ ○○○ ○○○ ○○○	○ ○○○○

Извличане по колони

## Допълнителни възможности при търсене

### Търсене по частично съвпадение

```
diamonds %>% select( starts_with('c') )
diamonds %>% select( ends_with('e') )
diamonds %>% select( contains('l') )
diamonds %>% select( matches('r.+t') )
```







## Отсяване на информация

### Филтриране на редове

```

diamonds %>% filter_(~cut == 'Ideal')
diamonds %>% filter_(~cut == 'Ideal')
cut <- 'Ideal'
diamonds %>% filter_(~cut == cut)
col <- 'cut'
cut <- 'Ideal'
diamonds %>% filter_(sprintf("%s == '%s'", col, cut))

```

## Използване на индексирането

### Избор по индекси

```
diamonds %>% slice(1:5)
diamonds %>% slice(c(1:5, 8, 15:20))
diamonds %>% slice(-1)
```



## Двупосочна поточна операции

### Отразяване на модификациите

```
diamonds2 <- diamonds
diamonds2 %<>% select(carat, price) %>%
mutate(ratio=price/carat, square=ratio*ratio)
head(diamonds2, n=3)
```

## Изчисления по няколко агрегатни функции

### Обобщаваща информация

```
summarize(diamonds, sd(price))
diamonds %>% summarize(sd(price))
diamonds %>% summarize(AveragePrice=mean(price),
MedianPrice=median(price), AverageCarat=mean(carat))
```



## Потребителски функции върху данни

### Специфични изчисления

```
bottom <- function(x, N=5) { x %>% arrange(carat) %>%
head(N) }
diamonds %>% group_by(cut) %>% do(bottom(., N=3))
```



# Системи за управление на бази от данни

## Работа с база данни

```
setwd( "~ / Desktop" )
download.file(" https://github.com/TodorBalabanov/Statistical-
Data-Processing-with-R/blob/master/data/diamonds.db?
raw=true", destfile=" ./diamonds.db", mode=" wb" )
source <- src_sqlite(" diamonds.db")
table <- tbl(source, " diamonds")
```

## Интерфейс към данни в СУБД

### Пресмятания с данни в база данни

```
table %>% group_by(cut) %>%
dplyr::summarize(AveragePrice=mean(price),
AverageCarat=mean(carat))
```

# Пакетът purrr

## Обработка елемент по елемент

### Прилагане на функцията map

```
library(purrr)
l1 <- list(m2=matrix(1:9,3), l2=1:5, m3=matrix(1:4,2), n1=2)
l1 %>% map( sum )
l1 %>% map(function(x) sum(x, na.rm=TRUE))
l1 %>% map(sum, na.rm=TRUE)
```

## Фамилия функции map

Функция	Тип на върнатата стойност
map	list
map_int	integer
map_dbl	numeric
map_chr	character
map_lgl	logical
map_df	data.frame

## Контекстна зависимост

### Извиквания на map според типа на върнатата стойност

```
l1 %>% map_int(NROW)
l1 %>% map_dbl(mean)
l1 %>% map_chr(class)
l1 %>% map_lgl(function(x) NROW(x) < 3)
list(3,4,1,5) %>% map( function(x){ data.frame(A=1:x,B=x:1) } )
list(3,4,1,5) %>% map_df( function(x){
data.frame(A=1:x,B=x:1) } )
l1 %>% map_if(is.matrix, function(x) x*2)
l1 %>% map_if(is.matrix, ~.x*2)
```



# Заклучение



Теми	Фамилията	функции <code>apply</code>	Пакетът <code>plyr</code>	Пакетът <code>data.table</code>	Бързи операции с пакета <code>dplyr</code>	Пакетът <code>purrr</code>
○	○	○	○	○	○	○
	○	○○	○○	○○	○○○	○○○○
	○	○	○	○	○○○	
	○	○○	○		○○○	
					○○○	

Дискусия

## Въпроси и отговори

Благодаря за вниманието!