Praca z danymi w R

Bartłomiej Tartanus

Macierze i ramki danych

Macierze

```
m <- matrix(1:6, nrow = 2, ncol = 3)
> m
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Czynniki

```
> f <- factor(rep(c("mało","średnio", "dużo"), times=3:1))
> f
[1] mało    mało    mało    średnio średnio dużo
Levels: dużo mało średnio
> str(f)
 Factor w/ 3 levels "dużo","mało",..: 2 2 2 3 3 1
```

Ramki danych

```
> df <- data.frame(typ=c("A","B","C"),
wartosc=1:3, zapas=50)
> df
  typ wartosc zapas
1   A       1    50
2   B       2    50
3   C       3    50
```

Odczyt i zapis danych

Odczyt

- read.table, read.csv - dane tabelaryczne
- readLines - dane tekstowe
- source - pliki źródłowe z kodem R
- dget - odczyt obiektów zapisanych w kodzie
- load - odczytywanie workspace'a
- unserialize - deserializacja obiektów z R zapisanych w postaci binarnej

Zapis

- write.table, write.csv - dane tabelaryczne
- writeLines - dane tekstowe
- dump - zapis wielu obiektów
- dput - zapis obiektu w języku R
- save - zapis workspace'a
- serialize - serializacja obiektów z R do postaci binarnej

JSON

Pakiet `rjson`

```
install.packages("rjson")
library("rjson")
json_data <-
fromJSON(file="/home/plik.json")
writeLines(toJSON(df),"/home/plik.json")
```

XLSX

Pakiet `openxlsx`

```
install.packages("openxlsx")
library("openxlsx")
xlsx <- read.xlsx("/home/plik.xlsx",sheet=1)
```

Bazy danych SQL

Pakiet RMySQL

```
install.packages("RMySQL")
library(RMySQL)
mydb <- dbConnect(MySQL(), user='admin',
password='1234',dbname='baza',host='local')
dbListTables(mydb)

rs <- dbSendQuery(mydb, "select * from tab")
```

Bazy danych NoSQL

Pakiet `mongolite`

```
library(mongolite)
m <- mongo(collection = "nycflights")
m$insert(flights)

m$count('{"month":1, "day":1}')
jan1 <- m$find('{"month":1, "day":1}')
```

Przetwarzanie danych z `dplyr`

Zaczynamy

```
require(dplyr)

#opakowanie ramki danych
df <- tbl_df(mtcars)
```

Funkcje na jednej ramce danych

- `filter()` / `slice()`
- `arrange()`
- `select()` / `rename()`
- `distinct()`
- `mutate()` / `transmute()`
- `summarise()` / `count()`
- `sample_n()` / `sample_frac()`

Grupowanie rekordów

Funkcja `group_by()` rozdziela wiersze na grupy względem wartości podanych kolumn

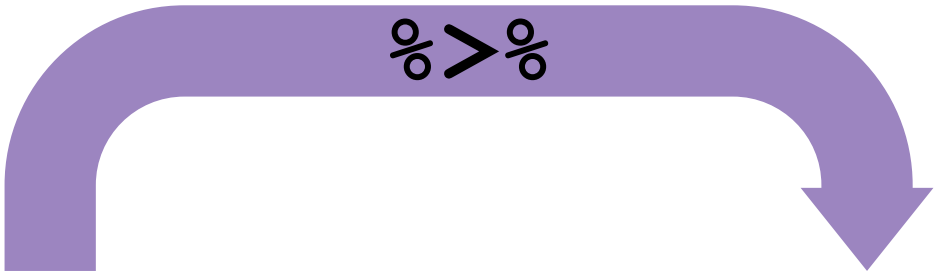
Wiele operacji w łańcuchach

Wszystkie funkcję mają część wspólną - każda przyjmuje jako pierwszy parametr ramkę danych a także zwraca ramkę danych.

W celu ułatwienia łączenia kolejnych operacji można używać operatora %>%
(przykłady w kodzie)

Wiele operacji w łańcuchach

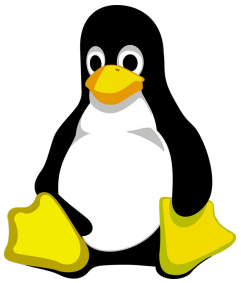
```
select(auta, car, mpg, hp)
auta %>% select(car, mpg, hp)
```



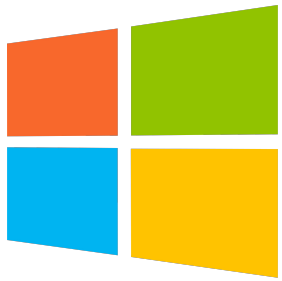

```
auta        select(    , car,mpg,hp)
```

Wiele operacji w łańcuchach

%>% za trudne do napisania?



Ctrl Shift M



Cmd Shift M

Funkcje na dwóch ramkach danych

- `inner_join()`
- `left_join()`
- `right_join()`
- `full_join()`
- `union()`
- `intersect()`
- `setdiff()`

Cheat Sheet

Różne ściągawki odnośnie używania przydatnych pakietów:

http://www.rstudio.com/resources/cheatsheets