

# Programowanie obiektowe w R5 na przykładzie wyceny nieruchomości

Teresa Ponikowska

SER VIII

2015-03-19

- Magister matematyki UW
- R uniwersytecko:
  - pierwszy kurs R
  - praca magisterska
  - pierwszy mini-projekt biznesowy
- R zawodowo
  - R jako *Glue Language*
  - utrzymywanie rozwiązań na bazie R
  - budowanie modeli decyzyjnych
  - przygotowywanie materiałów szkoleniowych i szkolenia z R
- Freelancer
  - projekty optymalizacyjne MathProg/CBC
  - automatyzacja raportów

# Plan prezentacji

- ❶ Zagadnienie wyceny nieruchomości
- ❷ Podejście obiektowe w R
  - S4
  - R5
- ❸ Rserve
- ❹ Inne wdrożenia R5/R5+Rserve

# Zagadnienie wyceny nieruchomości

## Cel projektu

Stworzenie narzędzia do wyceny nieruchomości gruntowych, lokali mieszkaniowych.

## Główne założenia

- Realizacja przy użyciu R
- Działanie w trybie on-line w celu wdrożenia u Klientów Klienta
- Możliwość rozbudowy o nowe typy nieruchomości (budynki, obszary leśne)
- Testowanie różnych modeli matematycznych - drzewa regresyjne, sieci neuronowe, itp.

# Zagadnienie wyceny nieruchomości

## Cel projektu

Stworzenie narzędzia do wyceny nieruchomości gruntowych, lokali mieszkaniowych.

## Główne założenia

- Realizacja przy użyciu R
- Działanie w trybie on-line w celu wdrożenia u Klientów Klienta
- Możliwość rozbudowy o nowe typy nieruchomości (budynki, obszary leśne)
- Testowanie różnych modeli matematycznych - drzewa regresyjne, sieci neuronowe, itp.

## Decyzje projektowe

- Implementacja w paradygmacie obiektywym
  - Dziedziczenie wspierające rozbudowę modeli o nowe typy nieruchomości
  - Możliwość zaprojektowania wielu modeli do wyceny przy ustalonym interfejsie
- Wykorzystanie Rserve: szybki dostęp webowy:
  - Etap 1: budowanie modeli
  - Etap 2: wycena na podstawie gotowych modeli
- S4 vs. **R5**

# Zagadnienie wyceny nieruchomości - ...

## Dlaczego R5?

- **Subiektywna decyzja:** nowa technologia
- Składniowo zbliżony do implementacji z Pythona/C++
- Odwoływanie się poprzez referencje

```
object <- changeSlot(object, spec)  # S4:  
object$changeSlot(spec)             # R5:
```

## Podstawowy przegląd

- `.self`, `"$"`, `"<<-"` - obiekt, dostęp do pól i metod, przypisania wewnętrzne (`"@"`)
- `callSuper(...)` - wywołanie metody z superklasy (`callNextMethod()`)
- `$lock(...)` - blokowanie pól klasy
- `getRefClass(...)` - dostęp do generatora klasy

<http://www.inside-r.org/r-doc/methods/ReferenceClasses>

<http://adv-r.had.co.nz/R5.html>

[stackoverflow+R5+?](http://stackoverflow.com/questions/10484000/r5)

# S4 vs. R5 - deklaracja i metody klas

## R5

```
setRefClass(  
  Class = "character",    # nazwa klasy  
  fields = "list",        # pola klasy  
  contains = "character", # superklasy/VIRTUAL  
  methods = list(         # metody klasy  
    initialize = function(...){},  
    finalize = function(...){}  
))
```

## S4

```
setClass(  
  Class = "character",    # nazwa klasy  
  slots = "list",         # pola klasy  
  prototype = "list",     # wartosci domyslne,  
  contains = "character" # superklasy/VIRTUAL  
)  
setMethod("metoda", signature="nazwa_klasy",  
  definition = function(.Object, ...){... return(.Object)})  
)
```



# S4 vs. R5 - deklaracja i metody klas - ...

## Szablon i inicjalizator obiektu grNieruchomosci w R5

```
grNieruchomosci = setRefClass(  
  Class = "grNieruchomosci",  
  fields = list(  
    id = "character",  
    atrybuty = "list",  
    dane = "data.frame"  
  ),  
  contains = "VIRTUAL",  
  methods = list(  
    "initialize" = function() {},  
    "pobierzDane" = function() {},  
    "sprawdzDane" = function() {},  
    "filtrujWartosciOdstajace" = function() {},  
    "raportuj" = function() {}  
  )  
)
```

# S4 vs. R5 - deklaracja i metody klas - ...

## Szablon obiektu grNieruchomosci w S4

```
grNieruchomosci = setClass(  
  Class = "grNieruchomosci",  
  slots = list(  
    id = "character",  
    atrybuty = "list",  
    dane = "data.frame"  
  ),  
  contains = "VIRTUAL"  
)
```

## Metody grNieruchomosci w S4

```
setMethod("initialize", signature="grNieruchomosci",  
  definition = function(.Object, ...){}  
)  
setMethod("pobierzDane", signature="grNieruchomosci",  
  definition = function(.Object, ...){}  
)  
setMethod("sprawdzDane", signature="grNieruchomosci",  
  definition = function(.Object, ...){}  
)  
setMethod("filtrujWartoscOdstajace", signature="grNieruchomosci",  
  definition = function(.Object, ...){}  
)
```

# S4 vs. R5 - podklasy

## grunty w R5

```
grunty = setRefClass(Class="grunty", contains="grNieruchomosci",
  methods = list(
    "pobierzDane" = function(){
      dane <- baza$pobierz("select ...where typ_nier = 'GRU'")
    }
  )
)
```

## lokale w R5

```
lokale = setRefClass(Class="lokale", contains="grNieruchomosci",
  methods = list(
    "pobierzDane" = function(){
      dane <- baza$pobierz("select ...where typ_nier = 'LOK.MIE'")
    }
  )
)
```

# S4 vs. R5 - podklasy

## grunty w S4

```
grunty=setClass(Class="grunty", contains="grNieruchomosci")
  setMethod("pobierzDane", signature = "grunty",
    definition = function(obj){
      obj@dane <- baza@pobierz("select ...where typ_nier = 'GRU'")
      return(obj)
    }
  )
```

## lokale w S4

```
lokale=setClass(Class="lokale", contains = "grNieruchomosci")
  setMethod("pobierzDane", signature = "lokale",
    definition = function(obj){
      obj@dane <- baza@pobierz("select ...where typ_nier = 'LOK.MIE'")
      return(obj)
    }
  )
```

# S4 vs. R5 - modele wyceny - ...

## Szablon modelu do prognozy w R5

```
modelProg = setRefClass(  
  Class = "modelProg",  
  contains = "VIRTUAL",  
  fields = list(  
    "id" = "character",  
    "status" = "character",  
    "nieruchomosci" = "grNieruchomosci"  
    #"model" = "ANY"  
  ),  
  methods = list(  
    initialize = function(flaga, nier){  
      if (flaga == 0){  
        nieruchomosci <- nier  
        nieruchomosci$wczytajDane()  
        nieruchomosci$filtrujObserwacjeOdstajace()  
        .self$uczModel()  
        .self$zapiszModel()  
      }else{  
        .self$wczytajModel()  
      }  
    },  
  
    wczytajModel = function() {},  
    zapiszModel = function() {},  
    #uczModel = function() {},  
    wycenNier = function() {}  
  )  
)
```

# S4 vs. R5 - modele wyceny - ...

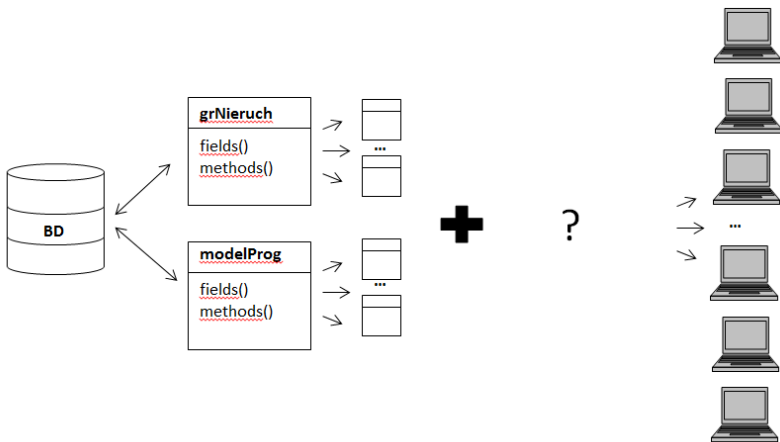
## Model drzewa regresyjnego

```
myRpart <- setRefClass(  
  Class = "myRpart", contains="modelProg",  
  fields = list(model = "rpart"),  
  methods = list(  
    uczModel = function(){  
      model <-<- rpart(cena~., data = nieruchomosci$dane)  
    }  
  ) )
```

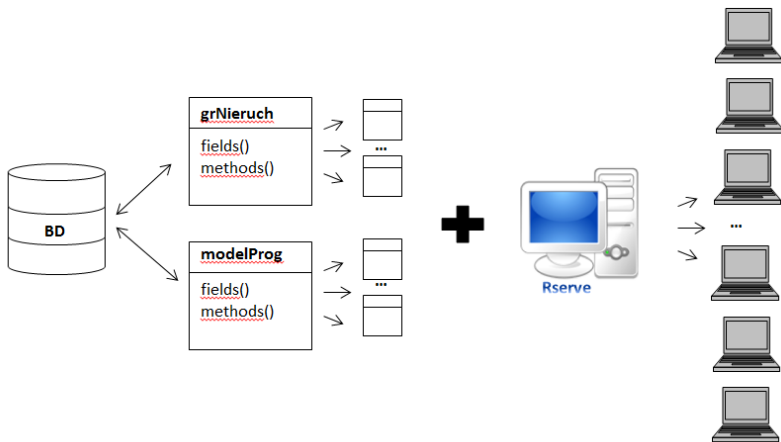
## Model sieci neuronowych

```
myNnet <- setRefClass(  
  Class = "myNnet", contains="modelProg",  
  fields = list(model = "nnet"),  
  methods = list(  
    uczModel = function(){  
      model <-<- rpart(cen~ ., data = nieruchomosci$dane)  
    }  
  ) )
```

# Zagadnienie wyceny nieruchomości - ...



# Zagadnienie wyceny nieruchomości - ...





## O Rserve

- Pakiet: Rserve
- Rodzaj serwera TCP/IP
- Możliwość zdalnego łączenia się z serwerem
- Implementacja Klienta w różnych językach: R, C/C++, Java, ...
- Dostępna opcja uwierzytelniania
- Strona projektu: <https://rforge.net/Rserve/index.html>

## Dlaczego Rserve?

### Czas!

- brak narzutu na rozruch R
- brak etapu wczytywania źródeł i bibliotek
- czas jednego scorowania:  $\sim 15$  ms, a nie kilka s

## Uruchomienie Rserve

- z poziomu R:

```
Rserve(  
  port = port nasłuchiwania Rserve,  
  args = dodatkowe argumenty  
)
```

- z poziomu CMD

```
"C:\Program Files\R\R-3.0.1\bin\i386\R" CMD Rserve --RS-port port [arg]
```

## Dodatkowe argumenty

- `--RS-workdir <path>`: katalog roboczy serwera
- `--R-source <file>`: wczytanie pliku przy starcie serwera
- `--RS-enable-remote`: zezwolenie na połączenie zdalne
- `--RS-conf <file>`: dodatkowy plik konfiguracyjny

## Uruchomienie Rserve

- z poziomu R:

```
Rserve(  
  port = port nasłuchiwania Rserve,  
  args = dodatkowe argumenty  
)
```

- z poziomu CMD

```
"C:\Program Files\R\R-3.0.1\bin\i386\R" CMD Rserve --RS-port port [arg]
```

## Dodatkowe argumenty

- RS-workdir <path>: katalog roboczy serwera
- R-source <file>: wczytanie pliku przy starcie serwera
- RS-enable-remote: zezwolenie na połączenie zdalne
- RS-conf <file>: dodatkowy plik konfiguracyjny

## Klient R

- Pakiet RScclient
- Połączenie z serwerem: `RS.connect(host=NULL, port=...)`
- Wykonanie operacji na serwerze: `RS.eval(rsc=..., x=...)`
- Zamykanie połączenia: `RS.close(rsc=...)`

## Klient R - przykład

```
conn <- RS.connect(port=8111)
wycena <- RS.eval(conn,
  MODELE[[1]]$wycenNier(
    typ = "lokal",
    "woj" = 14,
    "reg" = 71,
    "N_pok" = 3,
    "pow" = 60,
    "balkon" = TRUE
  )
)
print(wycena)
```

## Klient R

- Pakiet RScclient
- Połączenie z serwerem: `RS.connect(host=NULL, port=...)`
- Wykonanie operacji na serwerze: `RS.eval(rsc=..., x=...)`
- Zamykanie połączenia: `RS.close(rsc=...)`

## Klient R - przykład

```
conn <- RS.connect(port=8111)
wycena <- RS.eval(conn,
  MODELE[[1]]$wycenNier(
    typ = "lokal",
    "woj" = 14,
    "reg" = 71,
    "N.pok" = 3,
    "pow" = 60,
    "balkon" = TRUE
  )
)
print(wycena)
```

# Rserve - klient

## ● Serwer:

```
startRserve.bat
C:\Users\Lenovo\Desktop\SER-prezentacja_5.0\R_serwe>start
C:\Users\Lenovo\Desktop\SER-prezentacja_5.0\R_serwe>"C:\P
bin\i386\R" CMD Rserve --RS-port 8111 --RS-enable-remote

Wczytuje plik konfiguracyjny - ...
Ładowanie wymaganego pakietu: zoo

Dołączanie pakietu: 'zoo'

Następujące obiekty zostały zakryte from 'package:base':
    as.Date, as.Date.numeric

Ładowanie wymaganego pakietu: timeDate
This is forecast 5.8

Wczytuje plik konfiguracyjny - OK
Wczytuje klasy nieruchomości - ...
Wczytuje klasy nieruchomości - OK!
Wczytuje klasy modeli - ...
Wczytuje klasy modeli - OK!
Tworze instancje wycenijace - ...
- binaria modelu pomyslnie wczytane!
- binaria modelu pomyslnie wczytane!
Tworze instancje wycenijace - OK
Rserve: Ok, ready to answer queries.
- wyceniam nieruchomosc typu: lokal - ...
- wycena zakonczona: 5780zl/m2
```

## ● Klient:

```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function Project: (None)

testRserve.r
1 library(RSclient)
2 conn <- RS.connect(port=8111)
3 wycena <- RS.eval(conn,
4   MODELE[[1]]$wycenNier(
5     list(
6       typ = "lokal",
7       "woj" = 14,
8       "reg" = 71,
9       "N pok" = 3,
10      "pow" = 60,
11      "balikon" = TRUE
12    )
13  )
14 )
15 print(wycena)
16 RS.close(conn)
17

1:1 [Top Level] R Script

Console C:\Users\Lenovo\Desktop\SER-prezentacja_5.0\R_serwe/
>
>
>
> source("testRserve.r")
[1] 5780
>
```

## Podsumowanie

- Zaproponowane podejście (R5+Rserve) spełniły oczekiwania Klienta
- Paradygmat obietkowy w R: R5

## Inne projekty

- Prognozowanie w branży kosmetycznej R5
- Algorytmy rekomendacyjne w branży internetowej R5+Rserve
- Workflow optymalizacyjny z GLPK/CBC R5

## Podsumowanie

- Zaproponowane podejście (R5+Rserve) spełniły oczekiwania Klienta
- Paradygmat obietkowy w R: R5

## Inne projekty

- Prognozowanie w branży kosmetycznej R5
- Algorytmy rekomendacyjne w branży internetowej R5+Rserve
- Workflow optymalizacyjny z GLPK/CBC R5



Dziękuję za uwagę!