AR(p) modely: CVIČENIA

Beáta Stehlíková

2-PMS-10 Časové rady

Fakulta matematiky, fyziky a informatiky, UK v Bratislave

Cvičenie 1: Overovanie stacionarity AR procesov I.

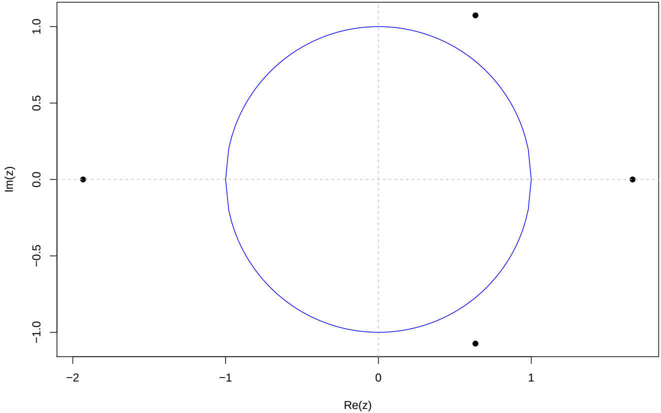
Zistite, či sú nasledovné AR procesy stacionárne:

- $x_t = 5 + 1.3x_{t-1} - 0.4x_{t-2} + u_t$
- $x_t = 15 + 0.3x_{t-1} - 0.5x_{t-2} - 0.1x_{t-3} + 0.2x_{t-4} + u_t$
- $(1 - 0.8L - 0.3L^2 - 0.2L^3)x_t = 10 + u_t$

Cvičenie 2: Grafické znázornenie koreňov

- Cieľom je znázorniť korene v komplexnej rovine spolu s jednotkovým kruhom (podobne, ako to bolo na prednáškových slajdoch) - pre nejaký AR proces.
- Užitočné funkcie a parametre:
  - ak dostne `plot` ako vstup komplexné čísla, nakreslí na x-ovú os ich reálnu a na y-ovú os imaginárnu časť
  - parameter `asp = 1` zabezpečí rovnakú mierku na obidvoch osiach
  - funkcia `curve` nakreslí zadanú krivku pre zadaný rozsah nezávislej premennej, s parameterom `add = TRUE` ju dokreslí do pôvodného obrázku
  - osi môžeme kresliť pomocou `abline` (horizontálnu s parametrom `h`, vertikálnu s parametrom `v`)

Ukážka pre $x_t = 0.9x_{t-1} - 0.4x_{t-2} - 0.2x_{t-3} + 0.2x_{t-4} + u_t$:



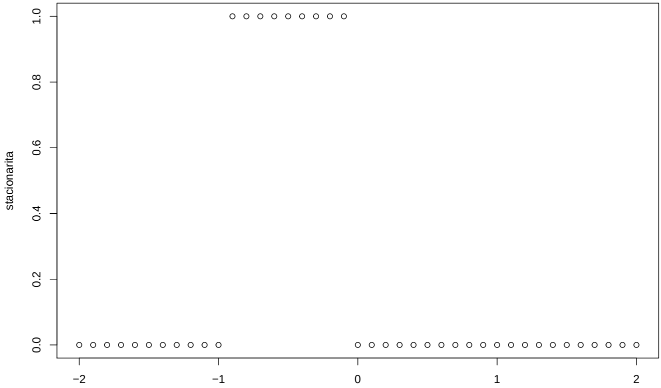
Cvičenie 3: Overovanie stacionarity AR procesov II.

- Ak riešime úlohy typu *"pre aké hodnoty parametra je daný proces stacionárny"* analyticky, je dobré mať numerickú kontrolu.
- **Príklad 1:** Pre aké hodnoty parametra k je stacionárny proces $x_t = x_{t-1} + kx_{t-2} + u_t$?
- **Numerická simulácia:**
  - Vytvoríme vektor hodnôt `kVektor` a pre každú jeho zložku zistíme, či je získaný proces stacionárny.
  - Dopíšte funkciu, ktorá pre zadanú hodnotu `k` vráti hodnotu `TRUE` alebo `FALSE` podľa toho, či je proces stacionárny.

```
kVektor <- seq(from = -3, to = 3, by = 0.1)
stacionarita <- sapply(kVektor, function(k) .... )
```

- Tieto hodnoty `TRUE` alebo `FALSE` sa potom dajú napríklad kresliť do grafu ako hodnoty 1 a 0:

```
plot(kVektor, stacionarita)
```



Alebo si vypíšeme:

```
kVektor[stacionarita]  # kVektor[stacionarita == TRUE]
```

```
## [1] -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1
```

```
kVektor[!stacionarita] # kVektor[stacionarita == FALSE]
```

```
##  [1] -2.0 -1.9 -1.8 -1.7 -1.6 -1.5 -1.4 -1.3 -1.2 -1.1 -
## [16]  0.4  0.5  0.6  0.7  0.8  0.9  1.0  1.1  1.2  1.3
## [31]  1.9  2.0
```

Spravte numerickú simuláciu k príkladu:

- **Príklad 2:** Pre aké hodnoty parametra k je stacionárny proces $x_t = x_{t-1} + kx_{t-2} - x_{t-3} + u_t$?

Cvičenie 4: Woldova reprezentácia procesu

- Funkcia `ARMAtoMA`, napríklad pre proces $x_t = 0.5 + 0.3x_{t-1} - 0.6x_{t-2} + u_t$ spravíme:

```
ARMAtoMA(ar = c(0.5, -0.6), lag.max = 20)
```

```
##  [1]  0.500000000 -0.350000000 -0.475000000 -0.02750000
##  [6]  0.152125000 -0.086687500 -0.134618750 -0.01529687
## [11]  0.045739531 -0.021003922 -0.037945680 -0.00637048
## [16]  0.013613374 -0.004942612 -0.010639330 -0.00235409
```

Cvičenie. Uvažujme procesy:

$$\begin{aligned} x_t &= 0.5 + 0.3x_{t-1} - 0.6x_{t-2} + u_t \\ y_t &= 0.5 + 0.5x_{t-1} + 0.2x_{t-2} + u_t \end{aligned}$$

- Ukážte, že sú stacionárne. Všimnite si, či sú korene reálne alebo komplexné.
- Zobrazte koeficienty Woldovej reprezentácie. Ako ich priebeh súvisí s koreňmi? Vyskúšajte aj na príkladoch iných procesov.

Cvičenie 5: Woldova reprezentácia pre odhadnuté modely

Dáta z predchádzajúceho cvičenia:

- Z balíka `priceR` použijeme `retrieve_inflation_data` na získanie dát z *World Bank API*
- Vytvorili sme z týchto dát časový rad `x` s hodnotami inflácie a hľadali sme AR model bez konštanty pre diferencie.

```
library(priceR)
data <- retrieve_inflation_data("US")
```

```
## Validating iso2Code for US
## Generating URL to request all 297 results
## Retrieving inflation data for US
## Generating URL to request all 63 results
```


x
Time
0
2
4
6
8
10
12
14
1960
1970
1980
1990
2000
2010
2020

Zadanie 1.

- Nájdeme koeficienty Woldovej reprezentácie pre procesy dané modelmi rádov 1 až 5.
- Najskôr si zapíšeme vektory koeficientov týchto modelov (budeme ich ešte potrebovať a nechceme ich odhadovať znovu).
  - Analogické ako pri použití funkcie `sapply`
  - Ale teraz sa nedajú zapísať výsledky do matice, lebo vektory koeficientov budú mať rôzne dĺžky
  - Použijeme preto funkciu `lapply` a výstupom z nej bude `list` (zoznam)

```
koefAR <- lapply(1:5, function(p) ... )
  # function(p) ...
  # vstup: rad procesu p pre pre diferencie bez konst.
  # vystup: vektor ar koeficientov
```

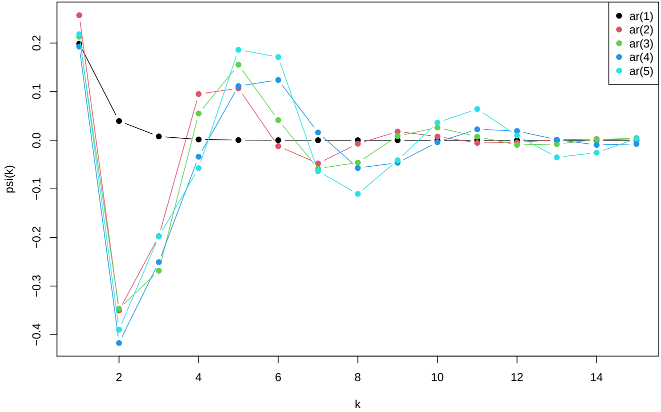
```
## [[1]]
##       ar1
## 0.1987695
##
## [[2]]
##        ar1        ar2
##  0.2574264 -0.4166409
##
## [[3]]
##        ar1        ar2        ar3
##  0.2130452 -0.3919758 -0.1111052
##
## [[4]]
##         ar1         ar2         ar3         ar4
##  0.19254716 -0.45430245 -0.08302077 -0.15911670
##
## [[5]]
```

- Teraz použijeme jednotlivé zložky zoznamu ako vstup do funkcie, ktorá pre zadaný vektor AR koeficientov vypočíta koeficienty Woldovej reprezentácie (vypočítajme prvých 15).
- Tu už bude výstupom matica (rozmery sú dané počtom modelov a počtom Woldovych koeficientov), preto môžeme použiť funkciu `sapply`

```
koefWold <- sapply(koefAR, function(ARvektor) ... )
  # vstup funkcie: vektor AR koef. modelu
  # vystup: prvych 15 hodnot Woldovej reprezetacie
```

```
##                [,1]          [,2]         [,3]         [,
##  [1,] 1.987695e-01  0.2574263627  0.213045191  0.1925471
##  [2,] 3.950932e-02 -0.3503726009 -0.346587550 -0.4172280
##  [3,] 7.853250e-03 -0.1974495042 -0.268452589 -0.2508314
##  [4,] 1.560987e-03  0.0951508597  0.054990968 -0.0338512
##  [5,] 3.102766e-04  0.1067598854  0.155450166  0.1114365
##  [6,] 6.167353e-05 -0.0121609340  0.041389265  0.1240476
##  [7,] 1.225882e-05 -0.0476110833 -0.058224703  0.0159809
##  [8,] 2.436680e-06 -0.0071896051 -0.045899408 -0.0571433
##  [9,] 4.843376e-07  0.0179859323  0.008445463 -0.0462929
## [10,] 9.627156e-08  0.0076255369  0.026259791 -0.0040180
## [11,] 1.913585e-08 -0.0055306614  0.007383769  0.0224585
## [12,] 3.803624e-09 -0.0046008489 -0.009658461  0.0190854
## [13,] 7.560446e-10  0.0011199201 -0.007869547  0.0011714
## [14,] 1.502786e-10  0.0022051989  0.001288939 -0.0096702
## [15,] 2.987081e-11  0.0001010718  0.004432380 -0.0075521
```

- Pri kreslení sa môže zísť funkcia `matplot` (spoločná x-ová os `1:15`, na y-ovej stĺpce zadanej matice)



- Porovnajte Woldove reprezentácie.
- Z minulého cvičenia: ktoré z týchto modelov mali dobré rezíduá?

Cvičenie 6: Autokorelačná funkcia procesu

- Odvodenie na prednáške, teraz spravíme numerický výpočet
- Funkcia `ARMAacf`, napríklad pre proces
  $x_t = 0.5 + 0.3x_{t-1} - 0.6x_{t-2} + u_t$ spravíme:

```r
ARMAacf(ar = c(0.3, -0.6), lag.max = 20)
```

```
##            0            1            2            3
##  1.000000000  0.187500000 -0.543750000 -0.275625000  0.2
##            6            7            8            9
## -0.074604375 -0.165447562 -0.004871644  0.097807044  0.0
##           12           13           14           15
## -0.034060469  0.019184677  0.026191684 -0.003653301 -0.0
##           18           19           20
##  0.009231205  0.004480153 -0.004194677
```

Cvičenie. Uvažujme znovu procesy:

$$
\begin{aligned}
x_t &= 0.5 + 0.3x_{t-1} - 0.6x_{t-2} + u_t \\
y_t &= 0.5 + 0.5x_{t-1} + 0.2x_{t-2} + u_t
\end{aligned}
$$

- Zobrazte pre každý z nich priebeh autokorelačnej funkcie, vynechajte pritom nultý lag, ktorý sa vždy rovná nule.
- Ako súvisí priebeh ACF s koreňmi (reálne/komplexné)? Ako súvisí s priebehom koeficientov Woldovej reprezentácie? Vyskúšajte aj na príkladoch iných procesov.

Cvičenie 7: ACF pre odhadnuté modely

Zadanie 2 pre dáta o inflácii:

- Uvažujme modely, ktorých koeficienty máme uložené v zozname `koefAR`.
- Pomocou `sapply` vypočítajte pre každý z nich hodnoty ACF, zobrazte ich graficky a porovnajte.
- Porovnajte aj s výstupom z cvičenia, v ktorom sme porovnávali koeficienty Woldovej reprezentácie.

Cvičenie 8: Stacionarita AR(2) procesu

Vytvorte funkciu, ktorá pre zadaný vektor parametrov (α_1, α_2) vráti `TRUE`, resp. `FALSE` podľa toho, či je AR(2) proces s týmito koeficientami stacionárny alebo nie.

```
stacAR2 <- function(alpha12) {
  # vstup: dvojzlozkovy vektor koeficientov
  # vytup: TRUE/FALSE podla stacionarity
  ....
}
```

Otestujte:

```r
stacAR2(c(1.2, -0.5))
```

```
## [1] TRUE
```

```r
stacAR2(c(1.2, 0.5))
```

```
## [1] FALSE
```

```r
stacAR2(c(0.5, 0)) # ar(1), ale stacionarny
```

```
## [1] TRUE
```

```r
stacAR2(c(0, 0))    # konstanta + biely sum -> stacionarny
```

```
## [1] TRUE
```

- Spravíme sieť bodov (α_1, α_2) a zakreslíme, ktoré sú stacionárne a ktoré nie sú.
- Na vytvorenie dvojíc bodov použijeme náhodne generované čísla z rovnomerného rozdelenia na štvorci $(-2, 2) \times (-2, 2)$.

```r
set.seed(123)
alfa1 <- runif(1000, min = -2, max = 2)
alfa2 <- runif(1000, min = -2, max = 2)
df <- data.frame(alfa1, alfa2)
head(df)
```

```
##        alfa1       alfa2
## 1 -0.8496899 -0.90550906
## 2  1.1532205  0.37546773
## 3 -0.3640923 -1.35926075
## 4  1.5320696  1.41372096
## 5  1.7618691  1.39095663
```

- Teraz pridáme do *data framu* tretí stĺpec, v ktorom bude výsledok overovania stacionarity.
- Namiesto `for`-cyklu použijeme funkciu `apply`. Tá umožňuje aplikovať zadanú funkciu na riadky alebo stĺpce objektu (matica, data frame).

Ukážka použitia `apply`

```
set.seed(123)
dfUkazka <- data.frame(a = sample.int(5),
                       b = sample.int(5),
                       c = sample.int(5))
dfUkazka # nejake cisla
```

```
##   a b c
## 1 3 3 2
## 2 2 1 3
## 3 5 2 1
## 4 4 5 4
## 5 1 4 5
```

```
f <- function(x) (x[1] + x[2]) * x[3] # nejaka funkcia
```

```r
# POUZITIE APPLY
apply(dfUkazka,    # vstupne data pre funkciu
      MARGIN = 1, # po riadkoch
      FUN = f)     # pouzita funkcia
```

```
## [1] 12  9  7 36 25
```

```r
# priradenie ako novy stlpec
dfUkazka$vysledok <- apply(dfUkazka, 1, f)
dfUkazka
```

```
##   a b c vysledok
## 1 3 3 2       12
## 2 2 1 3        9
## 3 5 2 1        7
## 4 4 5 4       36
## 5 1 4 5       25
```

```r
# DALSIE POUZITIE APPLY: teraz nie je vstupom cely riadok
g <- function(x) x[2]/x[1] + x[1] # zase nejaka funkcia
dfUkazka$vysledok2 <- apply(dfUkazka[ , 3:4], 1, g)
dfUkazka
```

```
##   a b c vysledok vysledok2
## 1 3 3 2       12         8
## 2 2 1 3        9         6
## 3 5 2 1        7         8
## 4 4 5 4       36        13
## 5 1 4 5       25        10
```

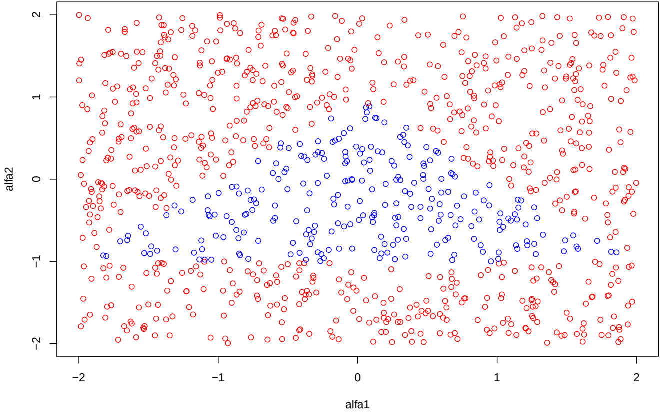
Vrátime sa k overovaniu stacionarity:

- Pomocou `apply` a vytvorenej funkcie `stacAR2` pridajte do *data framu* `df` stĺpec s informáciou o stacionarite procesu s danymi parametrami.

```
# na kontrolu
head(df)
```

```
##        alfa1       alfa2 stacionarita
## 1 -0.8496899 -0.90550906         TRUE
## 2  1.1532205  0.37546773        FALSE
## 3 -0.3640923 -1.35926075        FALSE
## 4  1.5320696  1.41372096        FALSE
## 5  1.7618691  1.39095663        FALSE
## 6 -1.8177740 -0.08845275        FALSE
```

- Vykreslite body farebne odlíšené podľa toho, či je proces stacionárny alebo nie.



Cvičenie 9: Hľadanie AR modelu pre zadané dáta

Všetky nasledujúce dáta majú spoločné zadanie:

- nájsť vhodný autoregresný model pre zadané dáta
- vysvetliť, prečo sú rezíduá vyhovujúce - skomentovať ACF rezíduí aj Ljung-Boxov test, pričom treba presne povedať, aká hypotéza sa testuje a či sa v tomto prípade zamieta alebo nie (a prečo sme s tým výsledkom spokojní)
- overiť stacionaritu získaného modelu (napísať polynóm, ktorého korene overujete, aké absolútne hodnoty koreňov vyšli a prečo sme s tým spokojní)
- spraviť predikcie

V prípade modelovania diferencií sa treba na základe prítomnosti, resp. neprítomnosti trendu treba rozhodnúť, či do modelu zahrnúť konštantu alebo nie.

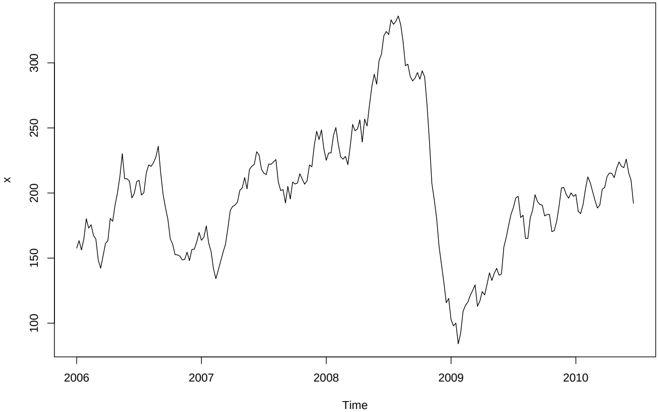
Dáta I. Ceny benzínu

Dáta z knižnice `astsa`, z popisu v helpe: *New York Harbor conventional regular gasoline weekly spot price FOB (in cents per gallon) from 2000 to mid-2010*. Zoberieme dáta od roku 2006.

```
library(astsa)
x <- window(gas, start = c(2006, 1))
```

Nájdite model pre premennú x tak, že budete jej diferencie modelovať AR procesom.

```r
plot(x)
```



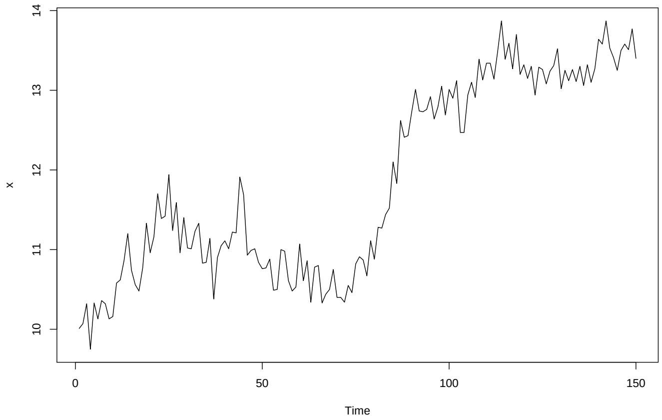
Dáta II. Indikátor od Boxa a Jenkinsa

Dáta z knižnice `astsa`, z popisu v helpe: *Leading indicator, 150 months; taken from Box and Jenkins (1970).*

```r
library(astsa)
x <- lead
```

Nájdite model pre premennú x tak, že budete jej diferencie modelovať AR procesom.

```
plot(x)
```



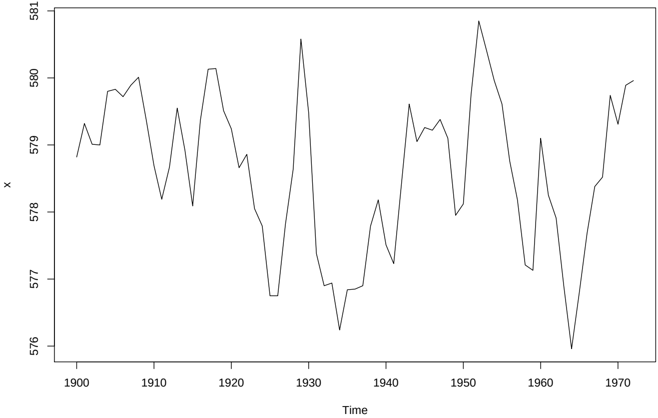
Dáta III. Hladina Hurónskeho jazera

Dáta z knižnice `datasets`, z popisu v helpe: *Annual measurements of the level, in feet, of Lake Huron 1875–1972.* Použijeme dáta od roku 1900.

```
library(datasets)
x <- window(LakeHuron, start = 1900)
```

Dáta nediferencujeme, AR model hľadáme priamo pre tieto dáta.

```
plot(x)
```



581
580
579
578
577
576
x
1900
1910
1920
1930
1940
1950
1960
1970
Time