M5201 – 10. cvičení: ***Modelování sezónnosti***

1 Metoda malého trendu

Uvažujme regresní model ve tvaru:

$$Y_t = Tr_t + Sz_t + \varepsilon_t \qquad\qquad \varepsilon_t \sim WN(0, \sigma^2).$$

Přeindexujeme	$Y_1, \dots, Y_n$ na	Y_{jk},	$j = 1, \dots, r$	$r \dots$ počet sezón
	$\varepsilon_1, \dots, \varepsilon_n$	ε_{jk},	$k = 1, \dots, d$	$d \dots$ délka sezóny.

Předpokládejme, že trend je konstantní pro j-tou sezónu, tj. $Tr_j = m_j$
a rovněž sezónní hodnota je konstantní pro k-tou sezónní složku, tj. $Sz_k = s_k$.
Regresní model můžeme napsat ve tvaru

$$\boxed{M_I}: \ Y_{jk} = m_j + s_k + \varepsilon_{jk} \qquad \varepsilon_{jk} \sim WN(0, \sigma^2) \qquad j = 1, \dots, r \qquad k = 1, \dots, d.$$

Maticově lze tento model rozepsat takto

$$\begin{pmatrix} Y_{11} \\ \vdots \\ \vdots \\ Y_{1d} \\ \hline Y_{21} \\ \vdots \\ \vdots \\ Y_{2d} \\ \hline \vdots \\ \vdots \\ \vdots \\ \hline Y_{r1} \\ \vdots \\ \vdots \\ Y_{rd} \end{pmatrix} = \left(\begin{array}{ccccccc|cccc}
1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 1 & \ddots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & \cdots & 0 & 1 \\ \hline
0 & 1 & 0 & \cdots & \cdots & \cdots & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 1 & \ddots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
0 & 1 & 0 & \cdots & \cdots & \cdots & 0 & 0 & \cdots & 0 & 1 \\ \hline
0 & 0 & 0 & \cdots & 0 & 1 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 1 & \ddots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & 0 & \cdots & \cdots & 1 & 0 & 0 & \cdots & 0 & 1 \\ \hline
0 & 0 & 0 & \cdots & \cdots & 0 & 1 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 1 & \ddots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & 0 & \cdots & \cdots & 0 & 1 & 0 & \cdots & 0 & 1
\end{array}\right) \begin{pmatrix} m_1 \\ \vdots \\ m_r \\ \hline s_1 \\ \vdots \\ s_d \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} \\ \vdots \\ \vdots \\ \varepsilon_{1d} \\ \hline \varepsilon_{21} \\ \vdots \\ \vdots \\ \varepsilon_{2d} \\ \vdots \\ \vdots \\ \vdots \\ \hline \varepsilon_{r1} \\ \vdots \\ \vdots \\ \varepsilon_{rd} \end{pmatrix}$$

Blokově lze psát

$$\begin{pmatrix} \mathbf{Y}_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{Y}_r \end{pmatrix} = \underbrace{\left(\begin{array}{cccccc|c}
\mathbf{1}_d & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{I}_d \\
\mathbf{0} & \ddots & \ddots & & & \vdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & & \vdots & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \vdots & \vdots \\
\vdots & & & \ddots & \ddots & \mathbf{0} & \vdots \\
\mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{1}_d & \mathbf{I}_d
\end{array}\right)}_{\text{označme } \mathbf{X}_{M_I} = \mathbf{X} = (\mathbf{X}_{(1)} | \mathbf{X}_{(2)})} \begin{pmatrix} m_1 \\ \vdots \\ m_r \\ \hline s_1 \\ \vdots \\ s_d \end{pmatrix} + \begin{pmatrix} \boldsymbol{\varepsilon}_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \boldsymbol{\varepsilon}_r \end{pmatrix}$$

kde $\mathbf{1}_d$ je sloupcový vektor délky d samých jedniček a $\mathbf{I}_d$ je diagonální jednotková matice typu $d \times d$.

Matice plánu $\mathbf{X}_{M_I} = \mathbf{X} = \left(\mathbf{X}_{(1)}|\mathbf{X}_{(2)}\right)$ však není plné hodnosti, neboť když sečteme prvních r sloupců, dostaneme vektor samých jedniček, což je rovno také součtu posledních d sloupců. Aby měl model plnou hodnost, přidejme ještě jednu podmínku, a to

$$s_1 + \cdots + s_d = 0$$

Potom

$$\mathbf{X}'\mathbf{X} = \left(\begin{array}{cccc|cccc} d & 0 & \cdots & 0 & 1 & \cdots & \cdots & 1 \\ 0 & d & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & d & 1 & \cdots & \cdots & 1 \\ \hline 1 & \cdots & \cdots & 1 & r & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & r & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 1 & \cdots & \cdots & 1 & 0 & \cdots & 0 & r \end{array}\right) \quad \text{a} \quad \mathbf{X}'\mathbf{Y} = \left(\begin{array}{c} Y_{1\cdot} \\ \vdots \\ Y_{r\cdot} \\ \hline Y_{\cdot 1} \\ \vdots \\ Y_{\cdot d} \end{array}\right),$$

kde využíváme tzv. tečkové notace

$$Y_{j\boldsymbol{.}} = \sum_{i=1}^{d} Y_{ji} \qquad Y_{\boldsymbol{.}k} = \sum_{i=1}^{r} Y_{ik}.$$

Normální rovnice $\mathbf{X}'\mathbf{X}\beta = \mathbf{X}'\mathbf{Y}$ můžeme přepsat (pro $j = 1, \ldots, r,\ k = 1, \ldots, d$) do tvaru

$$\begin{array}{rl} dm_j + \underbrace{\sum_{i=1}^d s_i}_{=0} = Y_{j\boldsymbol{.}} \ \Rightarrow & \boxed{\begin{array}{l} m_j = \frac{1}{d} Y_{j\boldsymbol{.}} = \overline{Y}_j \\ rs_k = Y_{\boldsymbol{.}k} - \sum_{i=1}^r m_i = \sum_{i=1}^r (Y_{ik} - m_i) \\ s_k = \frac{1}{r}\sum_{i=1}^r (Y_{ik} - m_i) \end{array}} \\ \sum_{i=1}^r m_i + rs_k = Y_{\boldsymbol{.}k} \ \Rightarrow & \\ \Rightarrow & \end{array}$$

Příklad 1

Typický příklad sezónního chování můžeme vysledovat například u návštěvnosti zoologických zahrad, kde je obvykle návštěvnost v zimních měsících poměrně malá, zatímco v létě velká. Proto zkusíme metodu malého trendu aplikovat na data v souboru `zoo_jihlava.txt`, v němž jsou uvedeny měsíční počty návštěvníků jihlavské ZOO v letech 2006-2010. Datový soubor načteme pomocí příkazu `scan()`. Příkazem `str()` vypíšeme strukturu načtených dat.

```
> fileDat <- paste(data.library, "zoo_jihlava.txt", sep = "")
> zoo <- scan(fileDat)
> str(zoo)
```

```
 num [1:60] 665 1309 2093 15624 33531 ...
```

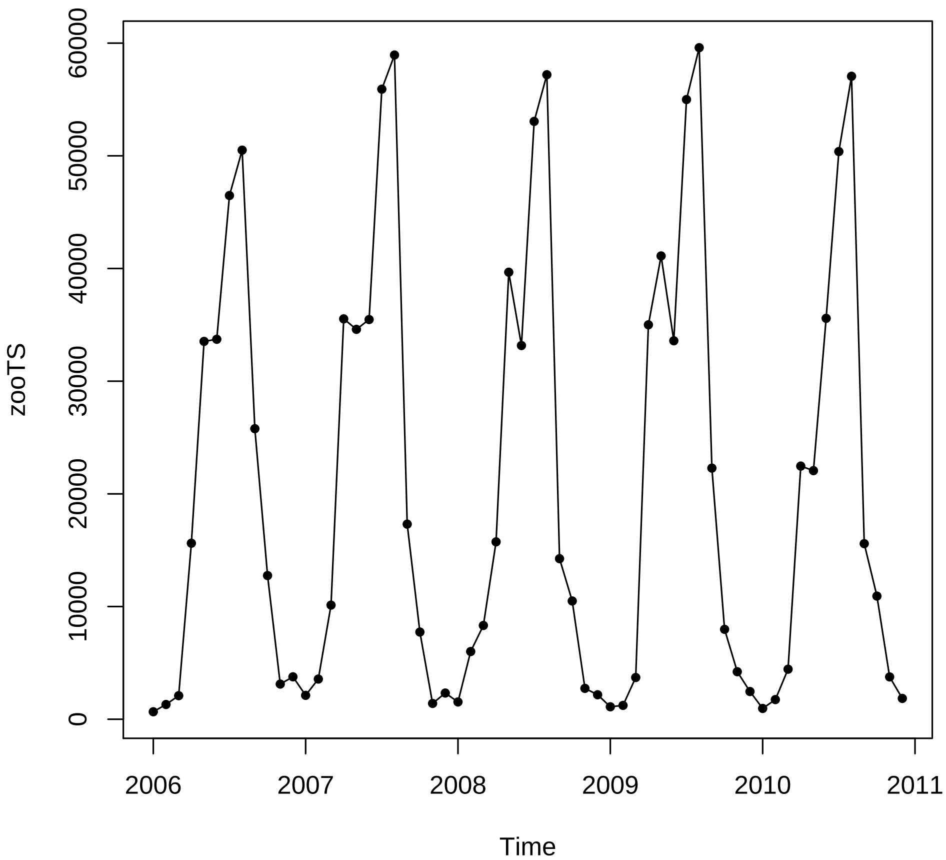
Pomocí funkce `ts()` vytvoříme z vektoru časovou řadu. Příkazem `str()` se podíváme na její strukturu.

```
> zooTS <- ts(zoo, start = 2006, frequency = 12)
> str(zooTS)
```

```
 Time-Series [1:60] from 2006 to 2011: 665 1309 2093 15624 33531 ...
```

Časovou řadu vykreslíme pomocí příkazu `plot()`.

```
> plot(zooTS, type = "o", pch = 20, cex = 3)
```



Obrázek 1: *Návštěvnost v ZOO Jihlava v letech 2006–2010*

Pracujeme-li v prostředí R s regresním modelem pro kategoriální proměnné, je třeba specifikovat přesnou podobu matice plánu pomocí tzv. kontrastů.

Chceme-li v našem příkladu použít model

$$\boxed{M_I}: \quad Y_{jk} = m_j + s_k + \varepsilon_{jk} \qquad \varepsilon_{jk} \sim WN(0, \sigma^2) \qquad j = 1, \ldots, r \qquad k = 1, \ldots, d$$

s dodatečnou podmínkou

$$s_1 + \cdots + s_d = 0,$$

budeme muset odpovídajícím způsobem nastavit kontrasty.

Protože na roční úrovně (parametry $m_1, \ldots, m_r$) nejsou kladeny žádné dodatečné podmínky a matici plánu jsme (dost neobvykle) navrhli tak, že neobsahuje vektor jedniček, tak pro parametry $m_1, \ldots, m_r$ platí

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} m_1 \\ \vdots \\ \vdots \\ m_r \end{pmatrix} = \begin{pmatrix} m_1 \\ \vdots \\ \vdots \\ m_r \end{pmatrix}$$

Vidíme, že matice kontrastů vztahující se k $m_1, \ldots, m_r$ bude jednotková diagonální matice řádu $r \times r$.

Na rozdíl od ročních úrovní m_j jsou sezónní kolísání vázána podmínkou

$$s_1 + \cdots + s_d = 0.$$

Takže jednu složku můžeme vyjádřit pomocí ostatních, nejčastěji jde o poslední, tj.

$$s_d = -s_1 - \cdots - s_{d-1}.$$

Díky tomu můžeme vypustit poslední složku. Celou reparametrizaci lze vyjádřit maticově takto

$$\underbrace{\begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -1 & -1 & \cdots & -1 & -1 \end{pmatrix}}_{\texttt{contr.sum}} \begin{pmatrix} s_1 \\ \vdots \\ s_{d-1} \end{pmatrix} = \begin{pmatrix} s_1 \\ \vdots \\ s_{d-1} \\ s_1 - \cdots - s_{d-1} \end{pmatrix}$$

a matice typu $d \times (d-1)$, kterou jsme označili jako `contr.sum` představuje kontrasty pro měsíční úrovně. Tato matice je již v prostředí R v kontrastech předem nadefinovaná.

Vrátíme se k načteným datům. Vytvoříme novou proměnnou typu data-frame, kam uložíme načtená data spolu s dalšími dvěma kategoriálními proměnnými `grY` (rok pozorování) a `grS` (měsíc pozorování). Takto vytvořený datový rámec budeme používat při práci s regresním modelem.

```
> xts <- zooTS
> d <- frequency(xts)
> n <- length(xts)
> m <- ceiling(n/d)
> shift <- start(xts)[2] - 1
```

K vytvoření vektoru, který by udával rok odpovídající pozorováním, použijeme funkci `gl(n,k,length)`. Tato funkce vytvoří vektor kategoriálních proměnných (faktorů) s n úrovněmi, každou z nich použije k-krát a celková délka vektoru je dána parametrem `length`.

```
> Season <- factor(as.integer(gl(d, 1, n)) - shift)
> Year <- factor(floor(time(xts)))
> data <- data.frame(x = as.vector(xts), grS = Season, grY = Year)
> str(data)
```

```
'data.frame':        60 obs. of  3 variables:
 $ x  : num  665 1309 2093 15624 33531 ...
 $ grS: Factor w/ 12 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ grY: Factor w/ 5 levels "2006","2007",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Pomocí funkce `lm()` odhadneme regresní model. Vysvětlujícími proměnnými budou kategoriální proměnné *rok* (`grY`) a *měsíc* (`grS`). Při zadávání modelu nesmíme zapomenout potlačit v matici plánu první sloupec jedniček, který by se jinak nastavil automaticky, a taky správně nastavit kontrasty.

```
> M1 <- lm(x ~ grY + grS - 1, data, contrasts = list(grY = diag(1, length(levels(data$grY))),
     grS = contr.sum))
> summary(M1)
```

```
Call:
lm(formula = x ~ grY + grS - 1, data = data, contrasts = list(grY = diag(1,
    length(levels(data$grY))), grS = contr.sum))

Residuals:
   Min     1Q Median     3Q    Max
-10486  -1835    397   1649   9116

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
grY2006    19113       1225  15.604  < 2e-16 ***
grY2007    22089       1225  18.034  < 2e-16 ***
grY2008    20363       1225  16.624  < 2e-16 ***
grY2009    22274       1225  18.185  < 2e-16 ***
grY2010    18901       1225  15.431  < 2e-16 ***
grS1      -19271       1817 -10.607 1.05e-13 ***
grS2      -17776       1817  -9.784 1.30e-12 ***
grS3      -14810       1817  -8.152 2.44e-10 ***
grS4        4326       1817   2.381   0.0216 *
grS5       13649       1817   7.513 2.04e-09 ***
grS6       13757       1817   7.572 1.67e-09 ***
grS7       31617       1817  17.403  < 2e-16 ***
grS8       36113       1817  19.878  < 2e-16 ***
grS9       -1504       1817  -0.828   0.4122
grS10     -10569       1817  -5.817 6.26e-07 ***
grS11     -17500       1817  -9.632 2.09e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4243 on 44 degrees of freedom
Multiple R-squared: 0.9835,        Adjusted R-squared: 0.9775
F-statistic: 163.9 on 16 and 44 DF,  p-value: < 2.2e-16
```

Vidíme, že kromě jednoho jsou všechny koeficienty vysoce signifikantní, což znamená že se statisticky významně liší od nuly (testováno pomocí statistik t).

Také p-hodnota u statistiky F naznačuje významnost modelu jako celku.

Jestliže pracujeme s regresním modelem, kde matice plánu nemá první sloupec tvořený jednotkami (ve výpisu chybí proměnná `(Intercept)`), koeficient determinace (`Multiple R-squared`), popř. upravený koeficient determinace (`Adjusted R-squared`) nemá interpretaci. Proto je lepší takovému modelu se spíše vyhnout.

Chceme–li posoudit významnost faktorů **grY** a **grS** ne po jednotlivých složkách (jak to nabízí jednotlivé t–testy) ale jako celek, musíme použít F–testy, jejichž hodnotu získáme použitím funkce **anova()**.

```
> anova(M1)


Analysis of Variance Table

Response: x
         Df     Sum Sq    Mean Sq F value    Pr(>F)
grY       5 2.5454e+10 5090860284  282.77 < 2.2e-16 ***
```

```
grS       11 2.1751e+10 1977338565  109.83 < 2.2e-16 ***
Residuals 44 7.9215e+08   18003488
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Výsledky F testů ukazují, že jak faktor roku (`grY`), tak faktor sezóny (`grS`) jsou statisticky významné (p-hodnoty v řádcích označených `grY` a `grS` jsou menší než hladina významnosti 0.05).

Abychom zjistili odhady koeficientů

$$m_j \qquad (j = 1, \ldots, r) \qquad \text{a} \qquad s_k \qquad (k = 1, \ldots, d)$$

použijeme funkci `predict()` s volbou `type="terms"`. Protože tentokrát neuvádíme parametr `newdata` (datový rámec s hodnotami vysvětlujících proměnných, pro které se má predikce počítat), počítají se predikce z dat, která vstupovala do modelu. Získanou matici `M1.terms` si podrobně prohlédneme.

```
> M1.terms <- predict(M1, type = "terms")
> str(M1.terms)
```

```
 num [1:60, 1:2] 19113 19113 19113 19113 19113 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:60] "1" "2" "3" "4" ...
  ..$ : chr [1:2] "grY" "grS"
 - attr(*, "constant")= num 0
```

Ukážeme, že fitované hodnoty jsou součtem dvou sloupců matice `M1.terms` označených `grY` a `grS` .

```
> pom <- cbind(M1.terms, M1.terms[, 1] + M1.terms[, 2], fitted(M1))
> colnames(pom) <- c(colnames(M1.terms), "grY+grS", "fitted(M1)")
> print(pom[1:6, ])
```

```
       grY        grS    grY+grS fitted(M1)
1 19112.75 -19271.267  -158.5167  -158.5167
2 19112.75 -17775.467  1337.2833  1337.2833
3 19112.75 -14810.067  4302.6833  4302.6833
4 19112.75   4326.333 23439.0833 23439.0833
5 19112.75  13649.133 32761.8833 32761.8833
6 19112.75  13757.333 32870.0833 32870.0833
```

Vidíme, že ve 3. a 4. sloupci matice jsou stejná čísla, tudíž je zřejmé, že hodnoty vyrovnané na základě modelu M_I jsou dány součtem odpovídajících odhadnutých parametrů m_j a s_k.

Ze všech hodnot matice `M1.terms` vypíšeme ty, keré se týkají koeficientů m_j a s_k. Sloupec `grY` udává roční úroveň příslušející každému pozorování, tzn. že se vždy opakují 12-krát po sobě. Koeficienty m_j tudíž získáme tak, že vypíšeme pouze 1., 13., 25. . . . hodnotu z prvního sloupce matice.

```
> (M1.mj <- M1.terms[seq(1, n, d), 1])
```

```
       1        13        25        37        49
19112.75 22088.58 20362.67 22273.58 18900.75
```

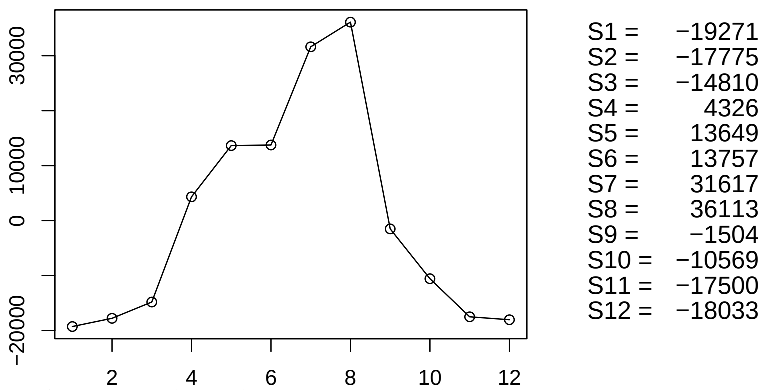
Koeficienty s_k najdeme ve sloupci **grS**, kde jsou měsíční odchylky od průměrné roční úrovně, a opakují se pro pozorování v každém roce. Koeficienty s_k proto dostaneme tak, že vypíšeme prvních 12 hodnot z druhého sloupce.

```
> (M1.sk <- M1.terms[1:d, 2])
```

```
         1          2          3          4          5          6          7          8
-19271.267 -17775.467 -14810.067   4326.333  13649.133  13757.333  31616.933  36112.933
         9         10         11         12
 -1504.067 -10568.667 -17499.667 -18033.467
```

Sezónní kolísání graficky znázorníme.

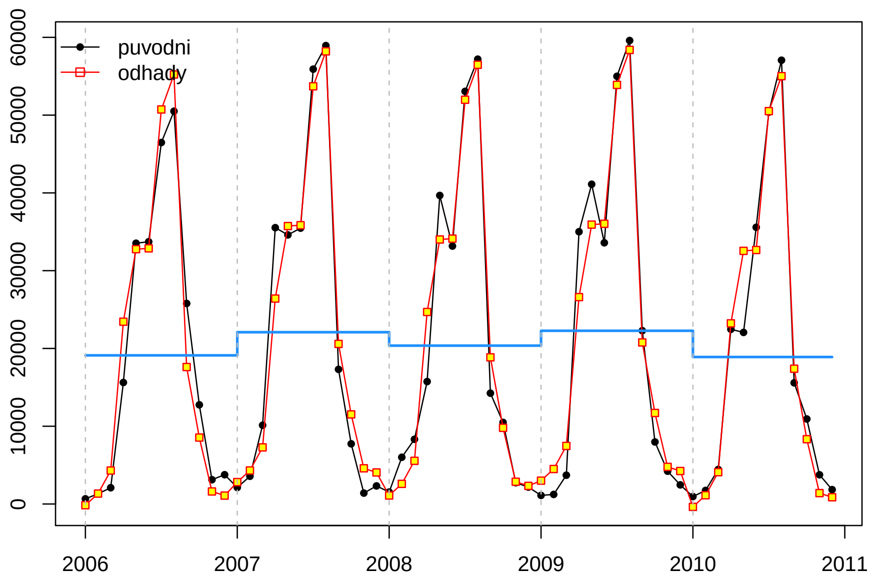
```
> SK <- M1.sk
> nf <- layout(matrix(c(1, 2), nrow = 1), width = c(2.75, 1.25))
> txtS <- paste("S", 1:length(SK), " =", sep = "")
> par(mar = c(2, 2, 0, 0) + 0.05)
> plot(1:length(SK), SK, type = "o")
> plot(c(0, 1), c(0, length(SK)), type = "n", axes = FALSE)
> text(rep(0, length(SK)), rev(1:length(SK)), txtS, adj = c(0, 1), cex = 1.125)
> text(rep(1, length(SK)), rev(1:length(SK)), round(SK), adj = c(1, 1),
    cex = 1.125)
```



Obrázek 2: Sezónní složky získané metodou malého trendu pro data *Návštěvnost v ZOO Jihlava v letech 2006–2010*

Výsledky celé dekompozice vykreslíme do jediného grafu.

```
> tt <- as.vector(time(xts))
> xx <- as.vector(xts)
> fit <- fitted(M1)
> tr <- rep(M1.mj, each = d, length.out = n)
> ylim <- range(range(xx), range(fit), tr)
> par(mfrow = c(1, 1), mar = c(2, 2, 2, 0) + 0.5)
> plot(tt, xx, type = "o", pch = 20, col = "black", ylim = ylim)
> lines(tt, fit, type = "o", pch = 22, col = "red", bg = "yellow", cex = 0.85)
> lines(tt, tr, type = "s", col = "dodgerblue", lwd = 2)
> abline(v = as.numeric(as.character(levels(data$grY))), col = "gray",
    lty = 2)
> legend(par("usr")[1], par("usr")[4], bty = "n", legend = c("puvodni",
    "odhady"), lty = c(1, 1), pch = c(20, 22), bg = c("black", "yellow"),
    col = c("black", "red"))
```



Obrázek 3: Metoda malého trendu pro data *Návštěvnost v ZOO Jihlava v letech 2006–2010*

Pokusme se výchozí model M_I modifikovat tak, aby matice plánu měla v prvním sloupci samé jedničky.

Toho lze dosáhnout, budeme–li uvažovat následující model

$$\boxed{M_I^{modif}}: \; Y_{jk} = \mu + m_j + s_k + \varepsilon_{jk} \qquad \varepsilon_{jk} \sim WN(0, \sigma^2), \; j = 1, \dots, r, \; k = 1, \dots, d$$

Přidáním nového parametru μ (celkový průměr) dostaneme opět model s neúplnou hodnosti. Protože budeme chtít jediné řešení, přidáme dvě doplňující podmínky, a to

$$s_1 + \cdots + s_d = 0 \qquad \text{a} \qquad m_1 + \cdots + m_r = 0.$$

Odhad parametrů regresního modelu dostaneme obdobným způsobem jako u modelu M_I, pouze změníme kontrasty pro kategoriální proměnnou `grY` a nevynecháme konstantní člen. Protože doplňující podmínka pro m_j je analogická podmínce pro s_k, je i tvar matice kontrastů stejný (až na počet řádků a sloupců), v R se jedná o kontrasty `contr.sum`.

```
> M1m <- lm(x ~ grY + grS, data, contrasts = list(grY = contr.sum, grS = contr.sum))
> summary(M1m)


Call:
lm(formula = x ~ grY + grS, data = data, contrasts = list(grY = contr.sum,
    grS = contr.sum))

Residuals:
   Min     1Q Median     3Q    Max
-10486  -1835    397   1649   9116

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  20547.7      547.8  37.511  < 2e-16 ***
grY1         -1434.9     1095.6  -1.310   0.1971
grY2          1540.9     1095.6   1.407   0.1666
```

```
grY3         -185.0     1095.6  -0.169   0.8667
grY4         1725.9     1095.6   1.575   0.1223
grS1       -19271.3     1816.8 -10.607 1.05e-13 ***
grS2       -17775.5     1816.8  -9.784 1.30e-12 ***
grS3       -14810.1     1816.8  -8.152 2.44e-10 ***
grS4         4326.3     1816.8   2.381   0.0216 *
grS5        13649.1     1816.8   7.513 2.04e-09 ***
grS6        13757.3     1816.8   7.572 1.67e-09 ***
grS7        31616.9     1816.8  17.403  < 2e-16 ***
grS8        36112.9     1816.8  19.878  < 2e-16 ***
grS9        -1504.1     1816.8  -0.828   0.4122
grS10      -10568.7     1816.8  -5.817 6.26e-07 ***
grS11      -17499.7     1816.8  -9.632 2.09e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4243 on 44 degrees of freedom
Multiple R-squared: 0.965,      Adjusted R-squared: 0.9531
F-statistic: 80.99 on 15 and 44 DF,  p-value: < 2.2e-16
```

```
> anova(M1m)
```

```
Analysis of Variance Table

Response: x
          Df     Sum Sq    Mean Sq  F value Pr(>F)
grY        4 1.2191e+08   30476274   1.6928 0.1687
grS       11 2.1751e+10 1977338565 109.8309 <2e-16 ***
Residuals 44 7.9215e+08   18003488
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Parametry $m_1, \ldots, m_r$ v tomto případě mají jinou interpretaci. Nejsou roční hladinou, ale ukazují kolísání kolem průměrné hladiny dané parametrem `(Intercept)`, který je odhadem parametru μ v modelu M_I^{modif}.

Pro získání koeficientů

$$m_j \qquad (j = 1, \ldots, r) \qquad \text{a} \qquad s_k \qquad (k = 1, \ldots, d)$$

opět použijeme funkci `predict()` s volbou parametru `type="terms"` a bez volby `newdata`, takže se predikce počítají z dat, která vstupovala do modelu.

```
> M1m.terms <- predict(M1m, type = "terms")
> str(M1m.terms)
```

```
 num [1:60, 1:2] -1435 -1435 -1435 -1435 -1435 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:60] "1" "2" "3" "4" ...
  ..$ : chr [1:2] "grY" "grS"
 - attr(*, "constant")= num 20548
```

Ukážeme, že tentokrát (protože model obsahuje `(Intercept)`) jsou fitované hodnoty součtem dvou sloupců matice `M1m.terms` označených `grY` a `grS` a konstanty, kterou získáme příkazem `attr(M1m.terms,"constant")`.

```
> pom <- cbind(M1m.terms, M1m.terms[, 1] + M1m.terms[, 2] + attr(M1m.terms,
     "constant"), fitted(M1m))
> colnames(pom) <- c(colnames(M1m.terms), "grY+grS+constant", "fitted(M1m)")
> print(pom[1:6, ])
```

```
        grY        grS grY+grS+constant fitted(M1m)
1 -1434.917 -19271.267        -158.5167   -158.5167
2 -1434.917 -17775.467        1337.2833   1337.2833
3 -1434.917 -14810.067        4302.6833   4302.6833
4 -1434.917   4326.333       23439.0833  23439.0833
5 -1434.917  13649.133       32761.8833  32761.8833
6 -1434.917  13757.333       32870.0833  32870.0833
```

Vybereme nyní příslušné koeficienty s_k a $\mu + m_j$

```
> (M1m.sk <- M1m.terms[1:d, 2])
```

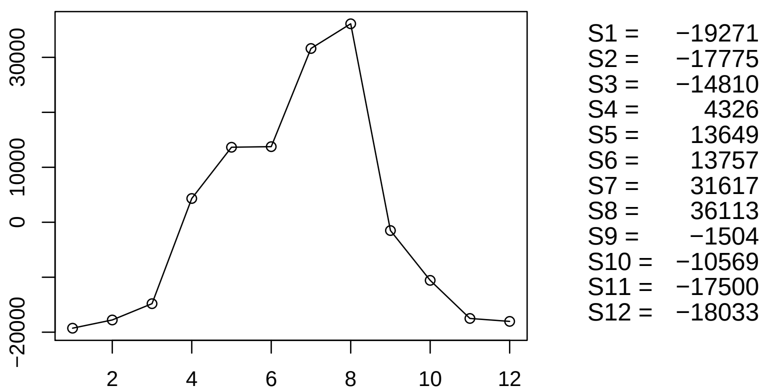
```
         1          2          3          4          5          6          7          8
-19271.267 -17775.467 -14810.067   4326.333  13649.133  13757.333  31616.933  36112.933
         9         10         11         12
 -1504.067 -10568.667 -17499.667 -18033.467
```

```
> (M1m.mj <- attr(M1m.terms, "constant") + M1m.terms[seq(1, n, d), 1])
```

```
       1       13       25       37       49
19112.75 22088.58 20362.67 22273.58 18900.75
```

Sezónní kolísání s_k vykresleme opět do grafu.

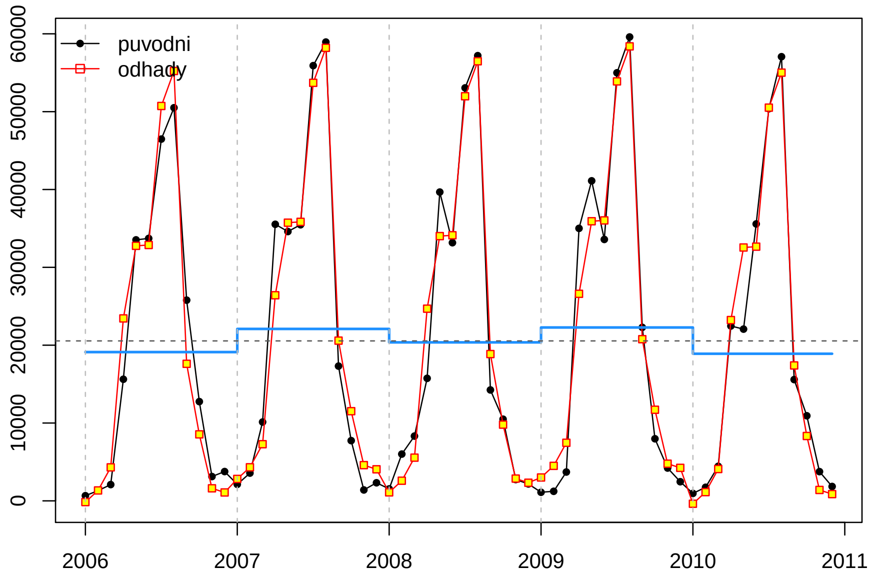
```
> SK <- M1m.sk
> nf <- layout(matrix(c(1, 2), nrow = 1), width = c(2.75, 1.25))
> txtS <- paste("S", 1:length(SK), " =", sep = "")
> par(mar = c(2, 2, 0, 0) + 0.05)
> plot(1:length(SK), SK, type = "o")
> plot(c(0, 1), c(0, length(SK)), type = "n", axes = FALSE)
> text(rep(0, length(SK)), rev(1:length(SK)), txtS, adj = c(0, 1), cex = 1.125)
> text(rep(1, length(SK)), rev(1:length(SK)), round(SK), adj = c(1, 1),
     cex = 1.125)
```



Obrázek 4: Sezónní složky získané modifikovanou metodou malého trendu pro data *Návštěvnost v ZOO Jihlava v letech 2006–2010*

Výsledky celé dekompozice vykreslíme do jediného grafu.

```
> fit <- fitted(M1m)
> tr <- rep(M1m.mj, each = d, length.out = n)
> ylim <- range(range(xx), range(fit), tr)
> par(mfrow = c(1, 1), mar = c(2, 2, 2, 0) + 0.5)
> plot(tt, xx, type = "o", pch = 20, col = "black", ylim = ylim)
> lines(tt, fit, type = "o", pch = 22, col = "red", bg = "yellow", cex = 0.85)
> lines(tt, tr, type = "s", col = "dodgerblue", lwd = 2)
> abline(v = as.numeric(as.character(levels(data$grY))), col = "gray",
     lty = 2)
> abline(h = attr(M1m.terms, "constant"), lty = 2, col = "gray35")
> legend(par("usr")[1], par("usr")[4], bty = "n", legend = c("puvodni",
     "odhady"), lty = c(1, 1), pch = c(20, 22), bg = c("black", "yellow"),
     col = c("black", "red"))
```



Obrázek 5: Modifikovaná metoda malého trendu pro data *Návštěvnost v ZOO Jihlava v letech 2006–2010*

Na závěr se můžeme porovnáním vyrovnaných hodnot pro model M_I a M_I^{modif} přesvědčit, že obě dvě metody dávají úplně stejné odhady.

```
> cbind(fitted(M1), fitted(M1m))[1:6, ]
```

```
        [,1]       [,2]
1  -158.5167  -158.5167
2  1337.2833  1337.2833
3  4302.6833  4302.6833
4 23439.0833 23439.0833
5 32761.8833 32761.8833
6 32870.0833 32870.0833
```

Pozn.: Obě dvě metody použité v Příkladu 1 jsou naprogramovány ve funkcích `SzSmallTrend()` a `SzSmallTrendModif()`, které jsou uloženy v souboru `FunkceM5201.R`. Po načtení příkazem `source` si je můžete prohlédnout, pokud v R zadáte `SzSmallTrend`, resp. `SzSmallTrendModif`.

2 Modelování sezónnosti pomocí SARIMA modelů

Sezónnost je v Box-Jenkinsově metodologii stejně jako trend modelována stochasticky. K tomu se zavádí *sezónní diferenční operátor* o délce $L > 0$:

$$\begin{aligned}
\Delta_L &= Y_t - Y_{t-L} = (1 - B^L)Y_t \\
\Delta_L^2 &= \Delta_L(\Delta_L Y_t) = (1 - B^L)^2 Y_t \\
&\vdots \\
\Delta_L^D &= (1 - B^L)^D Y_t
\end{aligned}$$

Při konstrukci SARIMA modelů se postupuje způsobem, který si přiblížíme na příkladu dat, která vykazují sezónnost s periodou o délce $L = 12$.

1. Vezmeme nejprve pouze lednová měření, tj. řadu $\{S_t^1 = B^{12} Y_t\}$ a zkonstruujeme pro ně model $ARIMA(P_1, D_1, Q_1)$

   $$\pi_1(B^{12})\Delta_{12}^{D_1} Y_t = \Psi_1(B^{12})\eta_t^{(1)} \sim ARIMA(P_1, D_1, Q_1),$$

   kde časový index t odpovídá lednovým obdobím. Přitom

   $$\begin{aligned}
   \pi_1(B^{12}) &= 1 - \pi_{1,1}B^{12} - \cdots - \pi_{1,P_1}B^{12\cdot P_1} \ \textit{(sezónní autoregresní operátor } SAR(P_1)\textit{)} \\
   \Psi_1(B^{12}) &= 1 + \psi_{1,1}B^{12} + \cdots + \psi_{1,Q_1}B^{12\cdot Q_1} \ \textit{(sezónní operátor klouzavých součtů } SMA(Q_1)\textit{)} \\
   \Delta_{12}^{D_1} &= (1 - B^{12})^{D_1} \ \textit{(sezónní diferenční operátor } SI(D_1)\textit{)}
   \end{aligned}$$

2. Podobné modely zkonstruujeme i pro ostatní měsíce:

   $$\begin{aligned}
   \pi_2(B^{12})\Delta_{12}^{D_2} Y_t &= \Psi_2(B^{12})\eta_t^{(2)} \sim ARIMA(P_2, D_2, Q_2) \\
   &\vdots \\
   \pi_{12}(B^{12})\Delta_{12}^{D_{12}} Y_t &= \Psi_{12}(B^{12})\eta_t^{(12)} \sim ARIMA(P_{12}, D_{12}, Q_{12})
   \end{aligned}$$

3. Dále předpokládáme, že modely pro jednotlivé měsíce jsou přibližně stejné, tj.

   $$\begin{aligned}
   P_1 \approx \cdots \approx P_{12} \approx P \\
   Q_1 \approx \cdots \approx Q_{12} \approx Q \\
   D_1 \approx \cdots \approx D_{12} \approx D
   \end{aligned}$$

   $$\begin{aligned}
   \pi_1(B^{12}) \approx \cdots \approx \pi_{12}(B^{12}) \approx \pi(B^{12}) \\
   \Psi_1(B^{12}) \approx \cdots \approx \Psi_{12}(B^{12}) \approx \Psi(B^{12})
   \end{aligned}$$

4. Náhodné veličiny $\eta_t^{(j)}, j = 1, \ldots, 12$ by však v těchto modelech měly být pro různé měsíce mezi sebou korelované, neboť by měl existovat např. vztah mezi lednovými a únorovými hodnotami. Předpokládáme proto, že také řada η_t je popsána modelem $ARIMA(p, d, q)$ tvaru:
$$\Phi(B)\Delta^d \eta_t = \Theta(B)\varepsilon_t \sim ARIMA(p, d, q),$$
kde $\varepsilon_t \sim WN(0, \sigma^2)$ je bílý šum.

5. Oba předchozí model spojíme do jediného tzv. *multiplikativního sezónního modelu řádu* $(p, d, q) \times (P, D, Q)_L$
$$\Phi(B)\pi(B^L)\Delta^d\Delta_L^D Y_t = \Theta(B)\Psi(B^L)\varepsilon_t \sim SARIMA(p, d, q) \times (P, D, Q)_L, L = 12$$

Příklad 2

Budeme zkoumat SARIMA proces zadaný předpisem
$$SARIMA(0, 1, 1) \times (0, 1, 1)_{12} : \ (1 - B)(1 - B^{12})Y_t = (1 - 0.5B)(1 - 0.7B^{12})\varepsilon_t,$$
kde
$$\varepsilon_t \sim WN(0, \sigma^2), \qquad \sigma^2 = 1.75.$$
Pokud model rozepíšeme, dostaneme
$$Y_t = Y_{t-1} + Y_{t-12} + Y_{t-13} + \varepsilon_t - 0.5\varepsilon_{t-1} - 0.7\varepsilon_{t-12} + 0.35\varepsilon_{t-13}$$

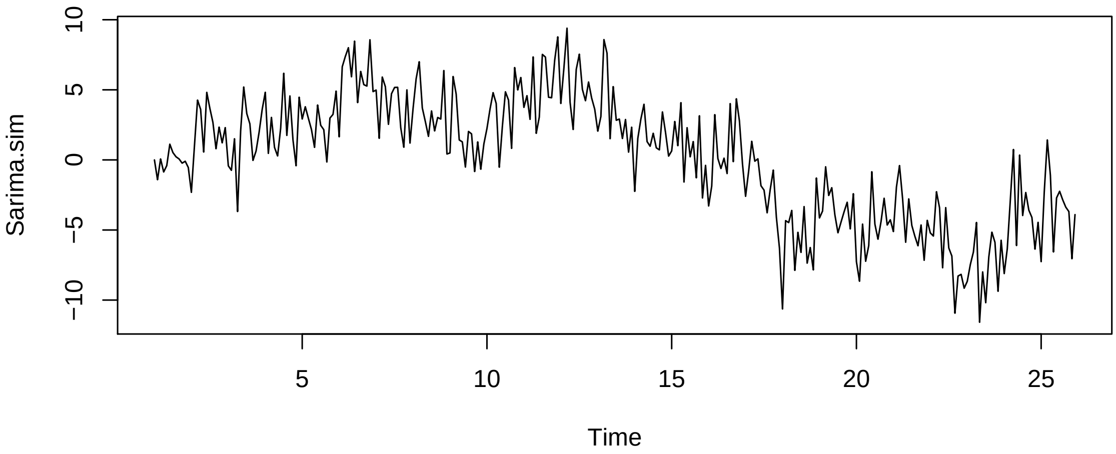
A nyní nasimulujeme časovou řadu s takto zadanými parametry. Použijeme k tomu další funkci ze souboru `FunkceM5201.R`, která se jmenuje `my.sarima.sim`. Použijeme tyto argumenty:

- `n=25` … 25 sezón (let)
- `period=12` … pro každou sezónu chceme 12 pozorování (odpovídá měsícům v roce)
- `sd=sqrt(1.75)` … směrodatná odchylka bílého šumu σ
- `model=list(ma=-0.5,order=c(0,1,1))` … parametry ARIMA procesu pro řadu $\eta_t^{(j)}$
- `seasonal=list(ma=-0.7, order=c(0,1,1))` … parametry ARIMA procesu pro časové řady odpovídající zvlášť jednotlivým měsícům

```
> fileFun <- paste(data.library, "FunkceM5201.R", sep = "")
> source(fileFun)
> Sarima.sim <- my.sarima.sim(n = 25, period = 12, sd = sqrt(1.75), model = list(ma = -0.5,
    order = c(0, 1, 1)), seasonal = list(ma = -0.7, order = c(0, 1,
    1)))
```

Vygenerovanou časovou řadu vykreslíme do grafu.

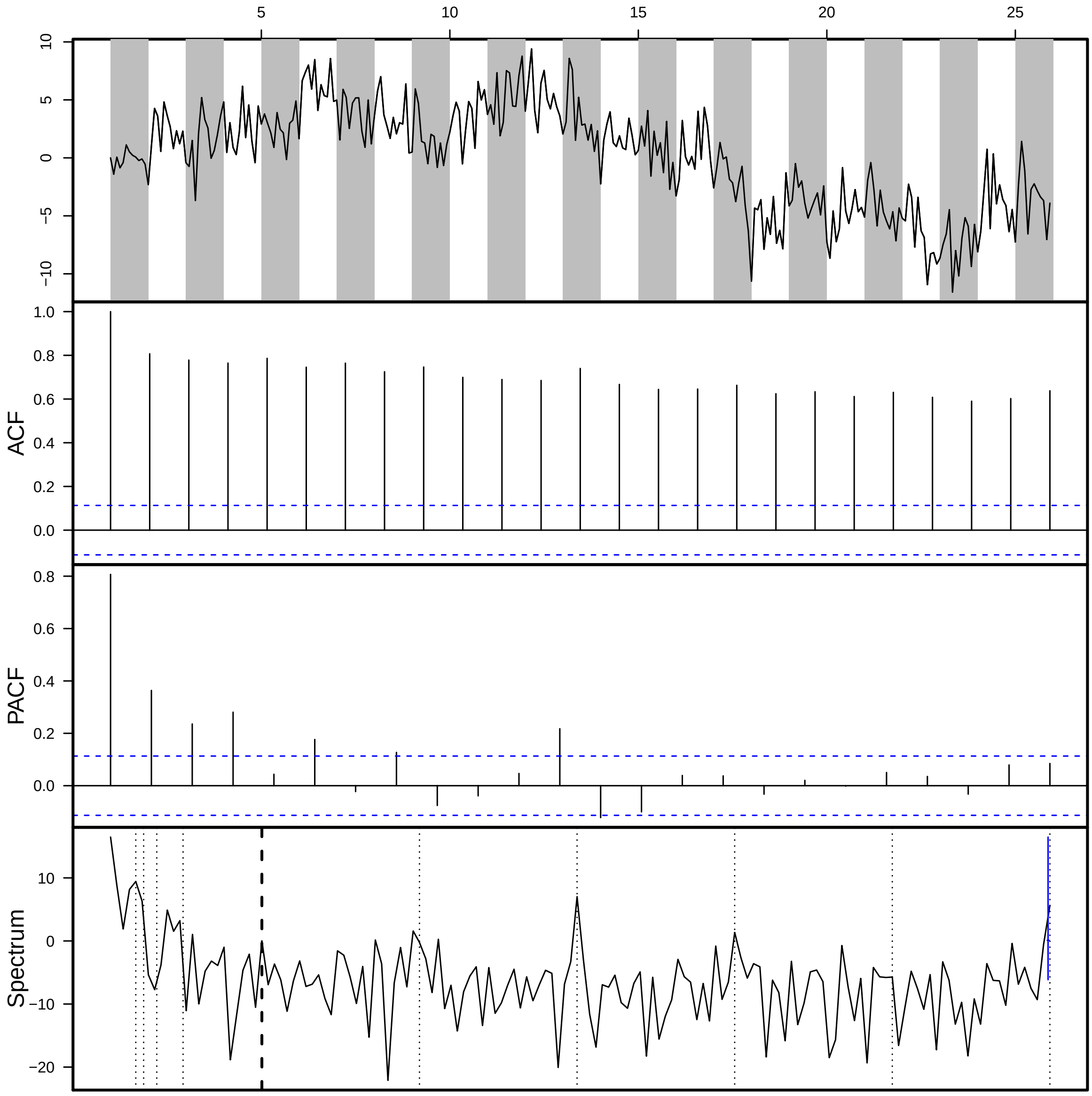
```
> plot(Sarima.sim)
```



Obrázek 6: Simulovaná data pro $SARIMA(0,1,1) \times (0,1,1)_{12}: \ (1-B)(1-B^{12})Y_t = (1-0.5B)(1-0.7B^{12})\varepsilon_t$

V dalším kroku by bylo dobré si prohlédnout výběrovou autokorelační funkci, výběrovou parciální autokorelační funkci a periodogram. Protože máme v souboru `FunkceM5201.R` pro tento případ naprogramovanou funkci `eda.ts`, můžeme všechny grafy vykreslit naráz v jednom kroku.

```
> eda.ts(Sarima.sim, bands = TRUE)
```



Obrázek 7: ACF, PACF a periodogram pro $SARIMA(0,1,1)\times(0,1,1)_{12}$: $(1-B)(1-B^{12})Y_t = (1-0.5B)(1-0.7B^{12})\varepsilon_t$

U autokorelační funkce můžeme najít pro $k = 12$ lokální maximum a podobné chování by se mělo objevit v dalších násobcích čísla 12 (to už ale v grafu nevidíme). I u parciální autokorelační funkce jsou patrné vyšší hodnoty v bodech $k = 12$ a $k = 24$.

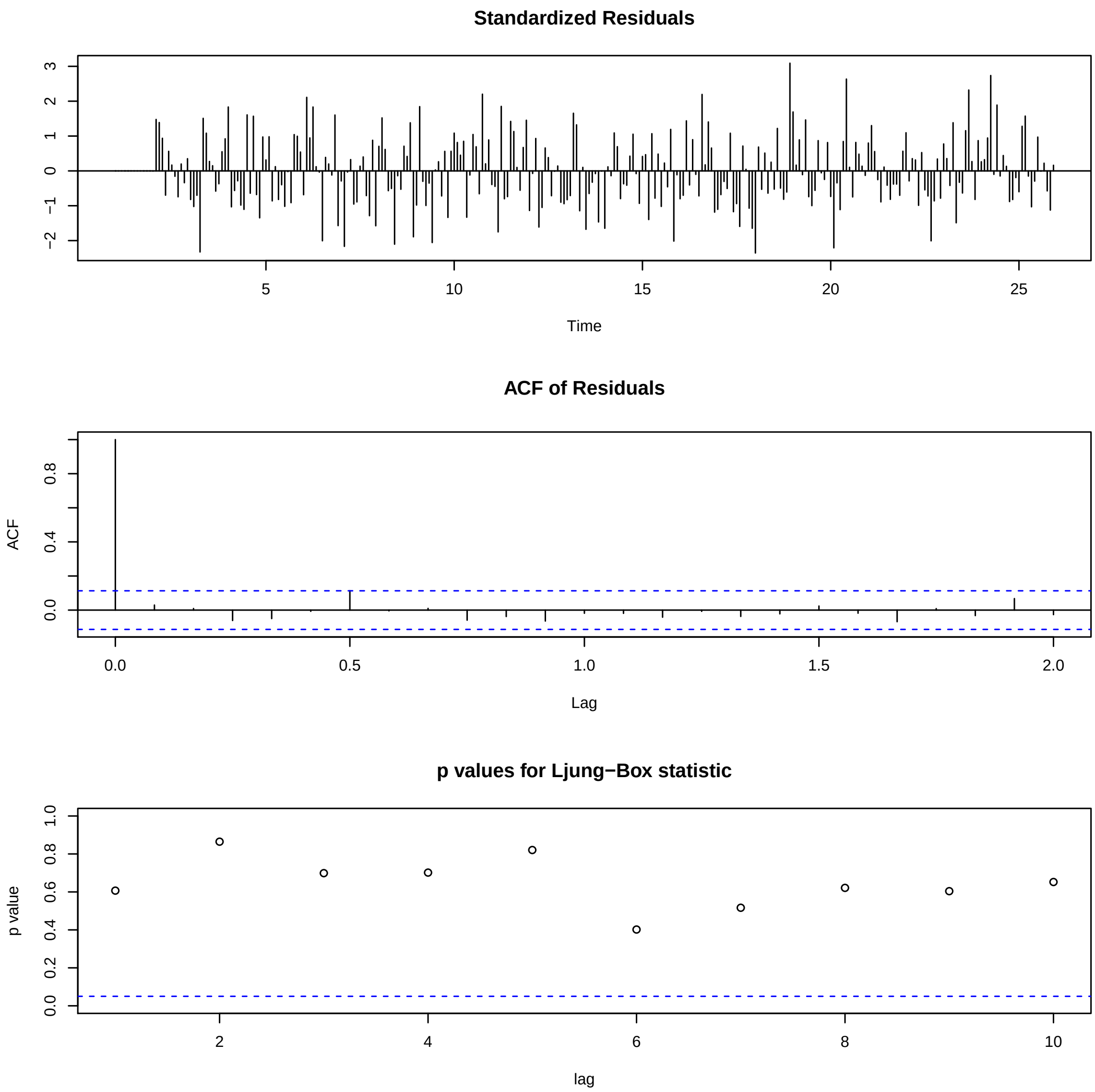
Předpokládejme, že máme k dispozici pouze časovou řadu `Sarima.sim` a nevíme, jakým procesem byla vygenerována. Provedeme pro ni odhad neznámých parametrů nejprve pomocí funkce `arima()`.

```
> sarima.fit <- arima(Sarima.sim, order = c(0, 1, 1), seasonal = list(order = c(0,
     1, 1), period = 12))
> print(sarima.fit)
```

```
Series: Sarima.sim
ARIMA(0,1,1)(0,1,1)[12]

Call: arima(x = Sarima.sim, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),      period = 12))

Coefficients:
          ma1     sma1
      -0.6588  -0.8473
s.e.   0.0473   0.0450

sigma^2 estimated as 4.178:  log likelihood = -620.3
AIC = 1246.6   AICc = 1246.68   BIC = 1257.57
```

Ve statistické knihovně máme ještě k dispozici funkci `tsdiag()`, kterou použijeme na odhadnutý SARIMA model, abychom ověřili, že rezidua již připomínají bílý šum.

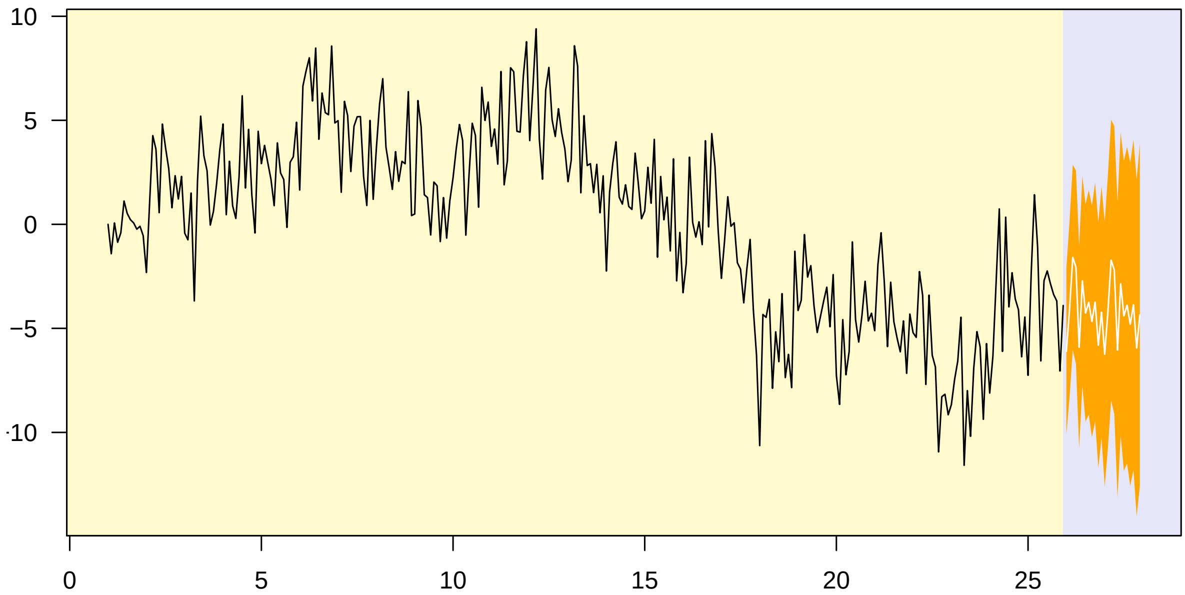
```
> tsdiag(sarima.fit)
```



Obrázek 8: Diagnostické grafy pro odhadnutý model $SARIMA(0,1,1) \times (0,1,1)_{12}$: $(1 - B)(1 - B^{12})Y_t = (1 - 0.5B)(1 - 0.7B^{12})\varepsilon_t$

Provedeme predikci na 2 roky dopředu, a to pomocí příkazu `PlotPredictARIMA()`. Výsledky zakreslíme do grafu.

```
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> PlotPredictARIMA(Sarima.sim, sarima.fit, n.ahead = 24)
```



Obrázek 9: Predikce pro odhadnutý model $SARIMA(0,1,1) \times (0,1,1)_{12}$: $(1-B)(1-B^{12})Y_t = (1-0.5B)(1-0.7B^{12})\varepsilon_t$

V knihovně `forecast` máme k dispozici funkci `auto.arima()`, kterou také můžeme použít při odhadu modelu na naše simulovaná data (u této funkce nemusíme předem znát řád SARIMA procesu).

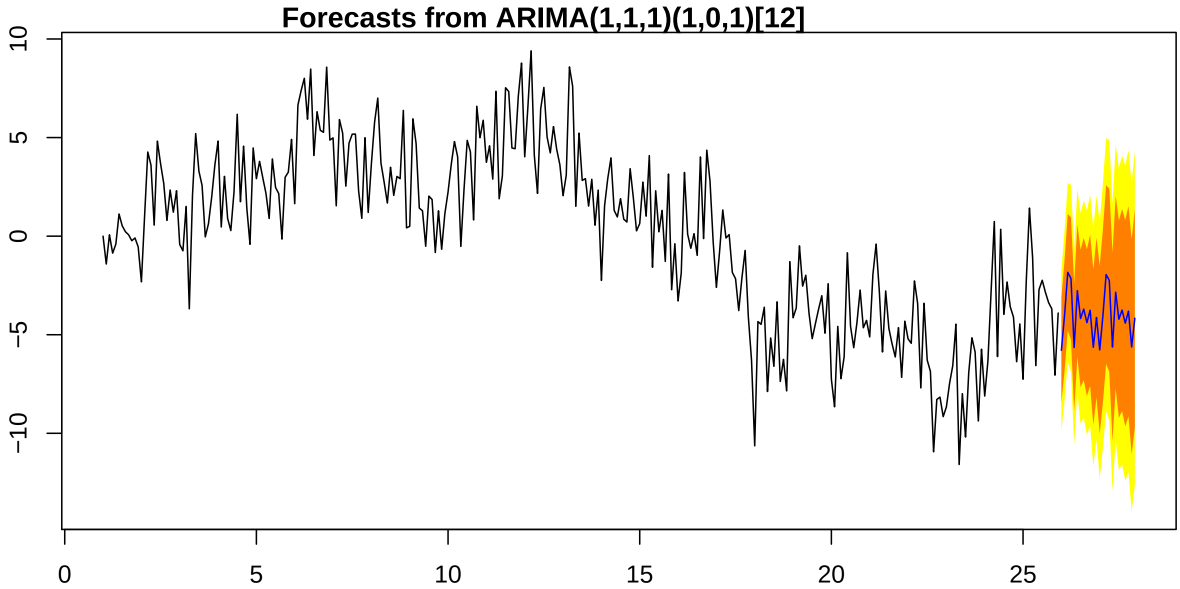
```
> library(forecast)
> sarima.fitF <- auto.arima(Sarima.sim, trace = TRUE)
```

```
 ARIMA(2,1,2)(1,0,1)[12] with drift         : 1304.535
 ARIMA(0,1,0) with drift         : 1452.135
 ARIMA(1,1,0)(1,0,0)[12] with drift         : 1363.831
 ARIMA(0,1,1)(0,0,1)[12] with drift         : 1321.387
 ARIMA(2,1,2)(0,0,1)[12] with drift         : 1320.695
 ARIMA(2,1,2)(2,0,1)[12] with drift         : 1307.369
 ARIMA(2,1,2)(1,0,0)[12] with drift         : 1324.994
 ARIMA(2,1,2)(1,0,2)[12] with drift         : 1306.999
 ARIMA(2,1,2) with drift         : 1331.400
 ARIMA(2,1,2)(2,0,2)[12] with drift         : 1e+20
 ARIMA(1,1,2)(1,0,1)[12] with drift         : 1295.781
 ARIMA(1,1,1)(1,0,1)[12] with drift         : 1295.731
 ARIMA(0,1,0)(1,0,1)[12] with drift         : 1e+20
 ARIMA(1,1,1)(1,0,1)[12]                    : 1293.779
 ARIMA(1,1,1)(0,0,1)[12]                    : 1320.990
 ARIMA(1,1,1)(2,0,1)[12]                    : 1301.367
 ARIMA(1,1,1)(1,0,0)[12]                    : 1320.275
 ARIMA(1,1,1)(1,0,2)[12]                    : 1295.749
 ARIMA(1,1,1)                    : 1336.943
 ARIMA(1,1,1)(2,0,2)[12]                    : 1e+20
 ARIMA(0,1,1)(1,0,1)[12]                    : 1297.169
 ARIMA(2,1,1)(1,0,1)[12]                    : 1303.838
 ARIMA(1,1,0)(1,0,1)[12]                    : 1e+20
 ARIMA(1,1,2)(1,0,1)[12]                    : 1293.807
 ARIMA(0,1,0)(1,0,1)[12]                    : 1e+20
 ARIMA(2,1,2)(1,0,1)[12]                    : 1303.339
```

```
 Best model: ARIMA(1,1,1)(1,0,1)[12]


> print(sarima.fitF)


Series: Sarima.sim
ARIMA(1,1,1)(1,0,1)[12]

Call: auto.arima(x = Sarima.sim, trace = TRUE)

Coefficients:
         ar1      ma1     sar1     sma1
      0.0028  -0.6554  0.9666  -0.8050
s.e.  0.0798   0.0651  0.0356   0.0627

sigma^2 estimated as 4.31:  log likelihood = -642.69
AIC = 1293.78   AICc = 1293.98   BIC = 1312.28
```

Pro predikci o 2 roky dopředu použijeme funkci `forecast()`.

```
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(forecast(sarima.fitF), n = 24)
```



Obrázek 10: Predikce na základě odhadnutého modelu $SARIMA(0,1,1) \times (0,1,1)_{12}: \ (1-B)(1-B^{12})Y_t = (1-0.5B)(1-0.7B^{12})\varepsilon_t$

3 Úkoly:

Načtěte do R datový soubor `rusove_prenocovani.txt`. V něm jsou uvedeny čtvrtletní údaje o počtu přenocování návštěvníků z Ruska v lázeňských zařízeních v ČR (vyjádřeno v počtu nocí strávených v ubytovacích zařízeních). Data se vztahují k období od 1. čtvrtletí roku 2000 do 2. čtvrtletí 2011. Na tato data aplikujte metodu malého trendu.