M0130 – 8. PRAKTIKUM : **M0130pr08** (*Modelování sezónnosti*)

A. Model I: Metoda malého trendu

Uvažujme regresní model ve tvaru:

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

Přeindexujme $Y_1, \dots, Y_n$ na Y_{jk}, $\quad j = 1, \dots, r \quad$ $r \dots$ počet sezón
$\qquad\qquad\quad \varepsilon_1, \dots, \varepsilon_n \quad \varepsilon_{jk}$, $\quad k = 1, \dots, d \quad$ $d \dots$ 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

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

Blokově lze psát

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

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. Proto 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

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

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

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

Příklad 1: **Návštěvnost v Moravském krasu**

Načteme datový soubor pomocí příkazu `read.table()` s parametrem `header=TRUE`, neboť v prvním řádku jsou názvy proměnných. Příkazem `str()` vypíšeme strukturu právě načteného datového rámce.

```
> fileDat <- paste(data.library, "MorKrasNavstevnost.txt", sep = "")
> MorKras <- read.table(fileDat, header = TRUE)
> str(MorKras)
```

```
'data.frame':        34 obs. of  6 variables:
 $ Rok              : int  1999 1999 1999 1999 1999 1999 1999 1999 1999 1999 ...
 $ Mesic            : Factor w/ 12 levels "březen","červen",..: 6 11 1 4 5 2 3 10 12 9 ...
 $ Punkevni         : int  2002 1694 4927 14186 29207 32158 39025 38629 19381 13625 ...
 $ SloupskoSosuvske: int  0 240 314 1498 4136 5343 12306 11725 2993 1629 ...
 $ Katerinska       : int  0 292 508 2141 7510 8822 18225 19681 3669 2212 ...
 $ Balcarka         : int  0 198 231 1265 4310 7445 10353 11243 1752 1400 ...
```

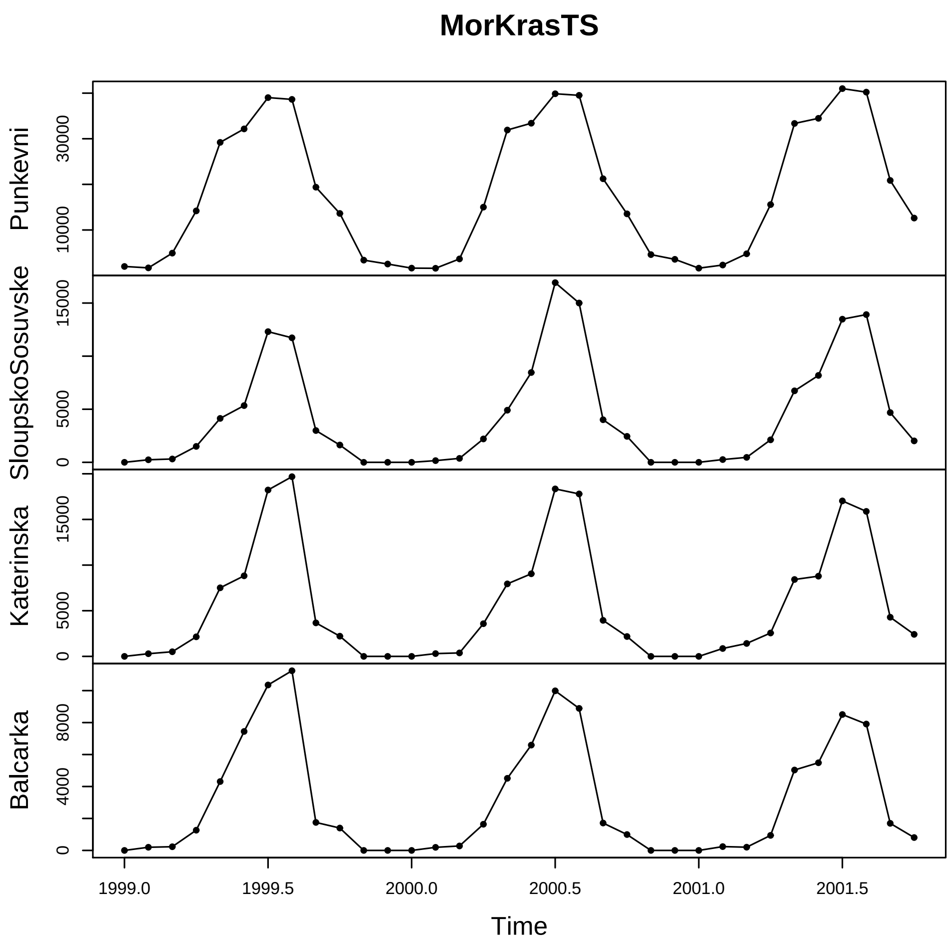
Pomocí funkce `ts()` vytvoříme vícerozměrnou časovou řadu. Příkazem `str()` se podíváme na její strukturu.

```
> MorKrasTS <- ts(MorKras[, -(1:2)], start = 1999, frequency = 12)
> str(MorKrasTS)
```

```
int [1:34, 1:4] 2002 1694 4927 14186 29207 32158 39025 38629 19381 13625 ...
- attr(*, "dimnames")=List of 2
 ..$ : NULL
 ..$ : chr [1:4] "Punkevni" "SloupskoSosuvske" "Katerinska" "Balcarka"
- attr(*, "tsp")= num [1:3] 1999 2002 12
- attr(*, "class")= chr [1:2] "mts" "ts"
```

Časové řady vykreslíme pomocí příkazu `plot()`.

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



Obrázek 1: *Počty návštěvníků ve vybraných jeskyních Moravského Krasu v letech 1999–2001*

V prostředí R se design matice plánu pro kategoriální proměnné řídí pomocí kontrastů.

Chceme–li použít model

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

s dodatečnou podmínkou

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

budeme muset správně nastavit kontrasty.

Protože na roční úrovně (parametry $m_1, \ldots, m_r$) nejsou 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`, je již v kontrastech předem nachystaná.

Vrátíme se k načteným datům. Ze čtyř časových řad si vybereme první a vytvoříme datový rámec s kategoriálními proměnnými `grY` a `grS`, který dále použijeme při definici regresního modelu.

```
> xts <- MorKrasTS[, 1]
> d <- frequency(xts)
> n <- length(xts)
> m <- ceiling(n/d)
> shift <- start(xts)[2] - 1
> r <- m * d - n - shift
> 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':        34 obs. of  3 variables:
 $ x  : int  2002 1694 4927 14186 29207 32158 39025 38629 19381 13625 ...
 $ grS: Factor w/ 12 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ grY: Factor w/ 3 levels "1999","2000",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Pomocí funkce `lm()` odhadneme regresní model. Nesmíme zapomenout potlačit první sloupec jedniček a 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
-1628.21  -252.20   -47.25   320.60  1280.83

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
grY1999  16730.2      229.6  72.870  < 2e-16 ***
grY2000  17462.6      229.6  76.060  < 2e-16 ***
grY2001  17967.1      261.8  68.637  < 2e-16 ***
grS1    -15629.3      441.6 -35.391  < 2e-16 ***
grS2    -15513.6      441.6 -35.129  < 2e-16 ***
grS3    -12933.0      441.6 -29.285  < 2e-16 ***
grS4     -2467.6      441.6  -5.588 1.81e-05 ***
grS5     14105.0      441.6  31.939  < 2e-16 ***
grS6     15960.4      441.6  36.140  < 2e-16 ***
grS7     22575.4      441.6  51.119  < 2e-16 ***
grS8     22070.4      441.6  49.976  < 2e-16 ***
grS9      3104.4      441.6   7.029 8.09e-07 ***
grS10    -4128.3      441.6  -9.348 9.71e-09 ***
grS11   -13099.9      538.4 -24.330 2.49e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 795.3 on 20 degrees of freedom
Multiple R-squared: 0.9993,       Adjusted R-squared: 0.9988
F-statistic:  2074 on 14 and 20 DF,  p-value: < 2.2e-16
```

Vidíme, že všechny koeficienty jsou 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 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        3 1.1295e+10 3765109314  5952.4 < 2.2e-16 ***
grS       11 7.0702e+09  642748531  1016.1 < 2.2e-16 ***
Residuals 20 1.2651e+07     632534
---
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.

Abychom dostali koeficienty

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

použijeme funkci `predict()` s volbou `type="terms"`. Protože neuvádíme parametr `newdata`, počítají se predikce z dat, která vstupovala do modelu. Matici `M1.terms` si podrobně prohlédneme.

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

```
 num [1:34, 1:2] 16730 16730 16730 16730 16730 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:34] "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 16730.17 -15629.292  1100.875   1100.875
2 16730.17 -15513.625  1216.542   1216.542
3 16730.17 -12932.958  3797.208   3797.208
4 16730.17  -2467.625 14262.542  14262.542
5 16730.17  14105.042 30835.208  30835.208
6 16730.17  15960.375 32690.542  32690.542
```

Ze všech hodnot matice `M1.terms` vybereme právě ty, keré se týkají koeficientů m_j a s_k.

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

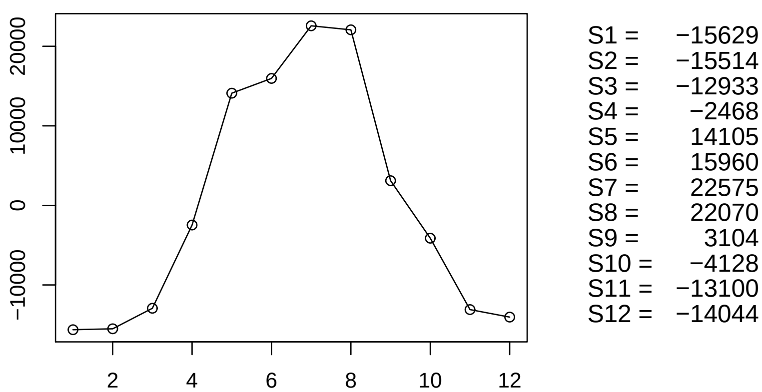
```
       1       13       25
16730.17 17462.58 17967.12
```

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

```
         1          2          3          4          5          6          7          8
-15629.292 -15513.625 -12932.958  -2467.625  14105.042  15960.375  22575.375  22070.375
         9         10         11         12
  3104.375  -4128.292 -13099.875 -14043.875
```

Sezónní kolísání graficky znázorníme.

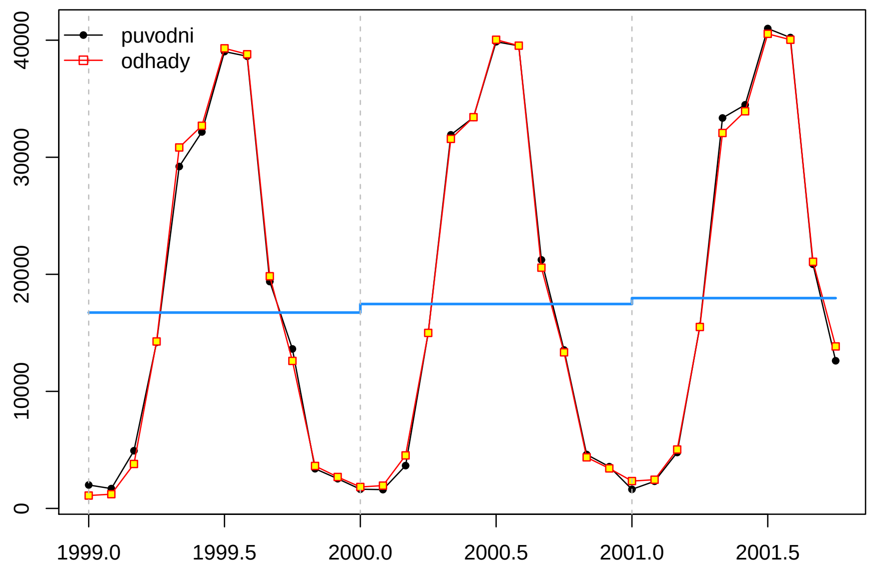
```
> SK <- M1.sk
> nf <- layout(matrix(c(1, 2), nrow = 1), width = c(2.75, 1.25))
> txtS <- paste("S", 1:length(SK), " =", sep = "")
> par(mar = c(2, 2, 0, 0) + 0.05)
> plot(1:length(SK), SK, type = "o")
> plot(c(0, 1), c(0, length(SK)), type = "n", axes = FALSE)
> text(rep(0, length(SK)), rev(1:length(SK)), 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 *Počty návštěvníků v Punkevních jeskyních v Moravského Krasu v letech 1999–2001*

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 *Počty návštěvníků v Punkevních jeskyních v Moravského Krasu v letech 1999–2001*

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

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

Regresní model dostaneme obdobným způsobem jako u modelu M_I, pouze změníme kontrasty pro kategoriální proměnnou `grY` a nevynecháme konstantní člen.

```
> 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
-1628.21  -252.20   -47.25   320.60  1280.83

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  17386.62     139.02 125.063  < 2e-16 ***
grY1          -656.46     192.09  -3.417  0.00273 **
grY2            75.96     192.09   0.395  0.69671
```

```
grS1        -15629.29     441.62 -35.391  < 2e-16 ***
grS2        -15513.62     441.62 -35.129  < 2e-16 ***
grS3        -12932.96     441.62 -29.285  < 2e-16 ***
grS4         -2467.63     441.62  -5.588 1.81e-05 ***
grS5         14105.04     441.62  31.939  < 2e-16 ***
grS6         15960.38     441.62  36.140  < 2e-16 ***
grS7         22575.38     441.62  51.119  < 2e-16 ***
grS8         22070.38     441.62  49.976  < 2e-16 ***
grS9          3104.37     441.62   7.029 8.09e-07 ***
grS10        -4128.29     441.62  -9.348 9.71e-09 ***
grS11       -13099.88     538.43 -24.330 2.49e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 795.3 on 20 degrees of freedom
Multiple R-squared: 0.9982,       Adjusted R-squared: 0.9971
F-statistic: 871.2 on 13 and 20 DF,  p-value: < 2.2e-16
```

```
> anova(M1m)
```

```
Analysis of Variance Table

Response: x
          Df     Sum Sq   Mean Sq  F value    Pr(>F)
grY        2   93946520  46973260   74.262 5.542e-10 ***
grS       11 7070233840 642748531 1016.149 < 2.2e-16 ***
Residuals 20   12650673    632534
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Z výpisu výsledků vidíme, že souhrnné charakteristiky (statistika F, koeficient determinace a upravený koeficient determinace) jsou úplně stejné. Tato situace však musela nastat, neboť jsme provedli pouze reparametrizaci.

Parametry $m_1, \ldots, m_r$ v tomto případě však 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 [1:34, 1:2] -622 -622 -622 -622 -622 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:34] "1" "2" "3" "4" ...
  ..$ : chr [1:2] "grY" "grS"
 - attr(*, "constant")= num 18151
```

Ukážeme, že tentokrát (protože model obsahuje `(Intercept)`) fitované hodnoty jsou součtem dvou sloupců matice `M1m.terms` označených `grY` a `grS` a konstanty, kterou získáme volbou `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 -622.3113 -16427.637         1100.875    1100.875
2 -622.3113 -16311.971         1216.542    1216.542
3 -622.3113 -13731.304         3797.208    3797.208
4 -622.3113  -3265.971        14262.542   14262.542
5 -622.3113  13306.696        30835.208   30835.208
6 -622.3113  15162.029        32690.542   32690.542
```

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

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

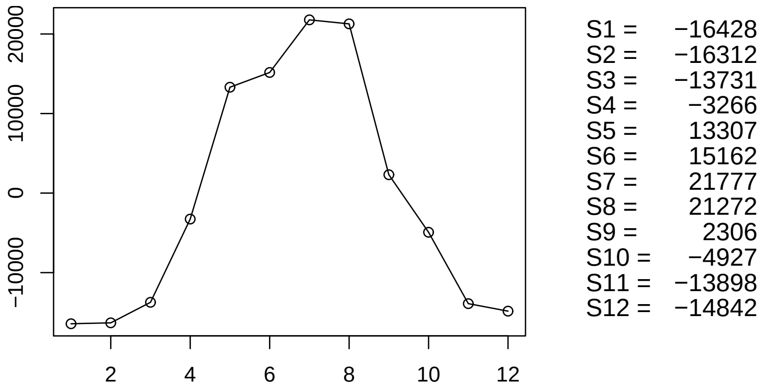
```
         1          2          3          4          5          6          7          8
-16427.637 -16311.971 -13731.304  -3265.971  13306.696  15162.029  21777.029  21272.029
         9         10         11         12
  2306.029  -4926.637 -13898.221 -14842.221
```

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

```
       1       13       25
17528.51 18260.93 18765.47
```

Sezónní kolísání s_k vykresleme opět do grafu.

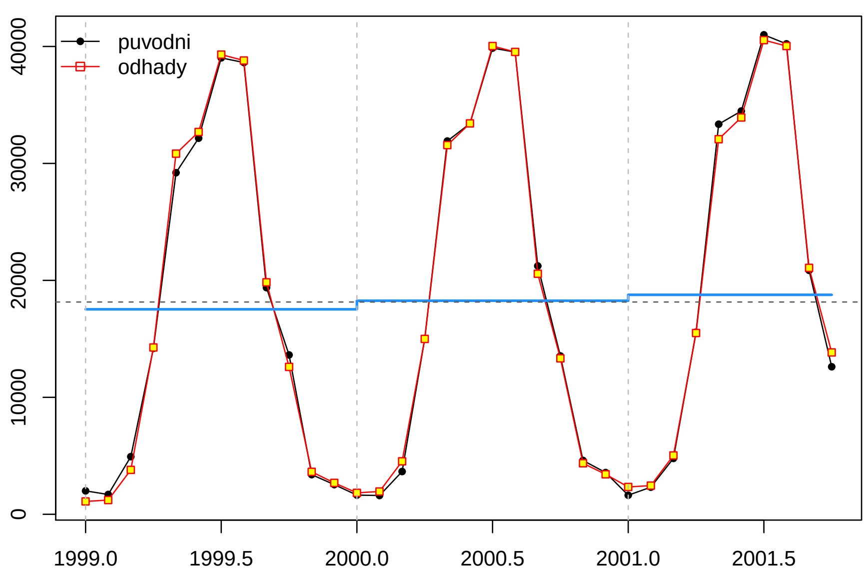
```
> SK <- M1m.sk
> nf <- layout(matrix(c(1, 2), nrow = 1), width = c(2.75, 1.25))
> txtS <- paste("S", 1:length(SK), " =", sep = "")
> par(mar = c(2, 2, 0, 0) + 0.05)
> plot(1:length(SK), SK, type = "o")
> plot(c(0, 1), c(0, length(SK)), type = "n", axes = FALSE)
> text(rep(0, length(SK)), rev(1:length(SK)), 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 *Počty návštěvníků v Punkevních jeskyních v Moravského Krasu v letech 1999–2001*

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 *Počty návštěvníků v Punkevních jeskyních v Moravského Krasu v letech 1999–2001*

Přesvědčeme se na závěr, že obě dvě metody dávají úplně stejné odhady.

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

```
       [,1]      [,2]
1  1100.875  1100.875
2  1216.542  1216.542
3  3797.208  3797.208
4 14262.542 14262.542
5 30835.208 30835.208
6 32690.542 32690.542
```

Abychom mohli obě dvě metody použít i na jiná data, vytvoříme dvě nové funkce, nejdříve funkci `SzSmallTrend()` a následně `SzSmallTrendModif()`. Doplníme ještě jednu pomocnou funkci `FindPositionForLegend()`, která hledá nejvhodnější pozici pro uvedení legendy grafu.

```
> SzSmallTrend <- function(xts, figure = TRUE) {
     if (class(xts) != "ts")
         stop("xts must be a time series (ts)")
     d <- frequency(xts)
     n <- length(xts)
     m <- ceiling(n/d)
     shift <- start(xts)[2] - 1
     r <- m * d - n - shift
     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)
     M11 <- lm(x ~ grY + grS - 1, data, contrasts = list(grY = diag(1,
         length(levels(data$grY))), grS = contr.sum))
     M11.terms <- predict(M11, type = "terms")
     M11.sk <- M11.terms[1:d, 2]
     M11.mj <- M11.terms[seq(1, n, d), 1]
     if (figure) {
         tt <- as.vector(time(xts))
         xx <- as.vector(xts)
         fit <- fitted(M11)
         tr <- rep(M11.mj, each = d, length.out = n)
         ylim <- range(range(xx), range(fit), tr)
         par(mfrow = c(1, 1), mar = c(2, 2, 2, 0) + 0.5)
         plot(tt, xx, type = "o", pch = 20, col = "black", ylim = ylim)
         lines(tt, fit, type = "o", pch = 22, col = "red", bg = "yellow",
             cex = 0.85)
         lines(tt, tr, type = "s", col = "dodgerblue", lwd = 2)
         abline(v = as.numeric(as.character(levels(data$grY))), col = "gray",
             lty = 2)
         legend(FindPositionForLegend(xx), bty = "n", legend = c("original",
             "estimate"), lty = c(1, 1), pch = c(20, 22), bg = c("black",
             "yellow"), col = c("black", "red"))
     }
     return(list(model = M11))
 }
> SzSmallTrendModif <- function(xts, figure = TRUE) {
     if (class(xts) != "ts")
         stop("xts must be a time series (ts)")
     d <- frequency(xts)
     n <- length(xts)
```

```
    m <- ceiling(n/d)
    shift <- start(xts)[2] - 1
    r <- m * d - n - shift
    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)
    M1m <- lm(x ~ grY + grS, data, contrasts = list(grY = contr.sum,
        grS = contr.sum))
    M1m.terms <- predict(M1m, type = "terms")
    M1m.sk <- M1m.terms[1:d, 2]
    M1m.mj <- attr(M1m.terms, "constant") + M1m.terms[seq(1, n, d),
        1]
    if (figure) {
        tt <- as.vector(time(xts))
        xx <- as.vector(xts)
        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 = "gray")
        legend(FindPositionForLegend(xx), bty = "n", legend = c("original",
            "estimate"), lty = c(1, 1), pch = c(20, 22), bg = c("black",
            "yellow"), col = c("black", "red"))
    }
    return(list(model = M1m, mu = coef(M1m)[1], mj = M1m.mj, sk = M1m.sk))
}
> FindPositionForLegend <- function(x) {
    if (min(dim(as.matrix(x))) > 1)
        stop("x is not a vector")
    n <- length(x)
    m <- trunc(n/3)
    MaxT <- c(max(x[1:m]), max(x[(m + 1):(n - m)]), max(x[(n - m + 1):n]))
    MinB <- c(min(x[1:m]), min(x[(m + 1):(n - m)]), min(x[(n - m + 1):n]))
    indT <- which.min(MaxT)
    indB <- which.max(MinB)
    if (max(abs(diff(MaxT[c(1:3, 1)]))) > max(abs(diff(MinB[c(1:3, 1)]))))
        PositionLegend <- c("topleft", "top", "topright")[indT]
    else PositionLegend <- c("bottomleft", "bottom", "bottomright")[indB]
    return(PositionLegend)
}
```

B. Model II: polynomický trend za celé období spolu se sezónností

Uvažujme regresní model ve tvaru $Y_t = Tr_t + Sz_t + \varepsilon_t, \quad \varepsilon_t \sim WN(0, \sigma^2)$.

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

Je-li trend polynomický za celé období $Tr_t = \beta_0 + \beta_1 t + \dots + \beta_m t^m$
a k-tá sezónní složka je rovna $Sz_k = s_k$,

pak regresní model můžeme napsat ve tvaru:

$$\boxed{M_{II}}: \; Y_{jk} = \beta_0 + \beta_1 t + \dots + \beta_m t^m + s_k + \varepsilon_{jk}, \quad \begin{array}{ll} j = 1, \dots, r & k = 1, \dots, d \\ t = (j-1)d + k & \end{array}$$

Matice plánu je pak tvaru

$$\mathbf{X}^* = \left(\begin{array}{cccc|ccccc}
1 & 1 & \cdots & 1^m & 1 & 0 & 0 & \cdots & 0 \\
1 & 2 & \cdots & 2^m & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
1 & d & \cdots & d^m & 0 & 0 & \cdots & 0 & 1 \\
\hline
1 & d+1 & \cdots & (d+1)^m & 1 & 0 & 0 & \cdots & 0 \\
1 & d+2 & \cdots & (d+2)^m & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
1 & 2d & \cdots & (2d)^m & 0 & 0 & \cdots & 0 & 1 \\
\hline
\vdots & \vdots & \vdots & \vdots & 1 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & 0 & \cdots & 0 & 1 \\
\hline
1 & (r-1)d+1 & \cdots & [(r-1)d+1]^m & 1 & 0 & 0 & \cdots & 0 \\
1 & (r-1)d+2 & \cdots & [(r-1)d+2]^m & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
1 & rd & \cdots & (rd)^m & 0 & 0 & \cdots & 0 & 1
\end{array}\right)$$

Matice plánu však není plné hodnosti, neboť když sečteme posledních d sloupců, dostaneme vektor samých jedniček. Proto použijeme tvz. *metodu horního rohu* a položíme první sezónu rovnu nule

$$s_1 = 0.$$

Této reparametrizaci na matici plné hodnosti v prosředí R odpovídá implicitně nastavený kontrast `contr.treatment`.

Za těchto předpokladů lze model maticově napsat ve tvaru

$$
\begin{pmatrix} Y_{11} \\ \vdots \\ \vdots \\ \vdots \\ Y_{1d} \\ \hline Y_{21} \\ \vdots \\ \vdots \\ \vdots \\ Y_{2d} \\ \hline \vdots \\ \vdots \\ \vdots \\ \vdots \\ \hline Y_{r1} \\ \vdots \\ \vdots \\ \vdots \\ Y_{rd} \end{pmatrix}
=
\underbrace{\left(\begin{array}{cccc|cccc}
1 & 1 & \cdots & 1^m & 0 & 0 & \cdots & 0 \\
1 & 2 & \cdots & 2^m & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
1 & d & \cdots & d^m & 0 & \cdots & 0 & 1 \\ \hline
1 & d+1 & \cdots & (d+1)^m & 0 & 0 & \cdots & 0 \\
1 & d+2 & \cdots & (d+2)^m & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
1 & 2d & \cdots & (2d)^m & 0 & \cdots & 0 & 1 \\ \hline
\vdots & \vdots & \vdots & \vdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & \cdots & 0 & 1 \\ \hline
1 & (r-1)d+1 & \cdots & [(r-1)d+1]^m & 0 & 0 & \cdots & 0 \\
1 & (r-1)d+2 & \cdots & [(r-1)d+2]^m & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\
1 & rd & \cdots & (rd)^m & 0 & \cdots & 0 & 1
\end{array}\right)}_{\text{označme } \mathbf{X}_{M_{II}}=\mathbf{X}}
\underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \\ \hline s_2 \\ \vdots \\ s_d \end{pmatrix}}_{\text{ozn.}\boldsymbol{\beta}}
+
\begin{pmatrix} \varepsilon_{11} \\ \vdots \\ \vdots \\ \vdots \\ \varepsilon_{1d} \\ \hline \varepsilon_{21} \\ \vdots \\ \vdots \\ \vdots \\ \varepsilon_{2d} \\ \hline \vdots \\ \vdots \\ \vdots \\ \vdots \\ \hline \varepsilon_{r1} \\ \vdots \\ \vdots \\ \vdots \\ \varepsilon_{rd} \end{pmatrix}
$$

Odhad *vektoru neznámých parametrů* metodou nejmenších čtverců je dán vztahem

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

Příklad 2: Návštěvnost v hromadných ubytovacích zařízeních v ČR v letech 2000–2010

V datovém souboru `NavstevnostHromadUbytZar2000_2010Q.dat` jsou obsažena čtvrtletní data, která se týkají návštěvnosti v hromadných ubytovacích zařízeních v ČR v letech 2000–2010. Hromadným ubytovacím zařízením se rozumí zařízení s minimálně pěti pokoji nebo deseti lůžky.

Čtvrtletní data v jednotlivých sloupcích obsahují

1. sloupec – počet hostů
2. sloupec – počet přenocování

Datový soubor načteme pomocí příkazu `read.table()`. Vzhledem k tomu, že obsahuje v prvním řádku názvy proměnných, v příkazu `read.table` nesmíme zapomenout nastavit `header=TRUE`. Příkazem `str()` vypíšeme strukturu datového rámce.

```
> fileDat <- paste(data.library, "NavstevnostHromadUbytZar2000_2010Q.dat",
    sep = "")
> navstevnost <- read.table(fileDat, header = TRUE)
```

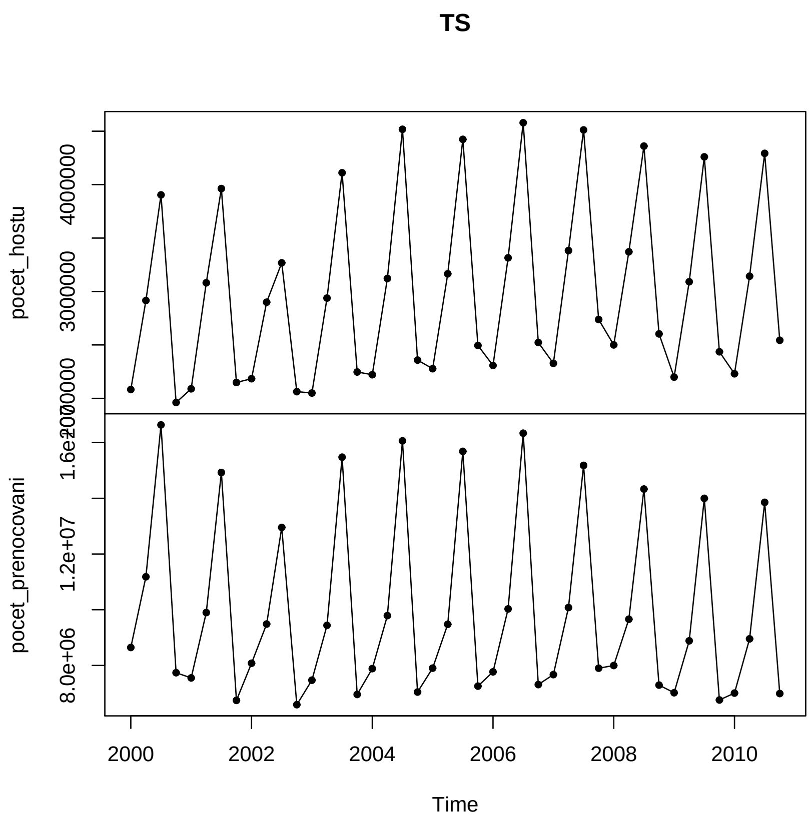
Vedle načteného datového rámce vytvoříme i vícerozměrnou časovou řadu, pomocí příkazu `str()` se podíváme na její strukturu.

```
> TS <- ts(navstevnost, start = 2000, frequency = 4)
> str(TS)
```

```
 int [1:44, 1:2] 2082827 2915857 3903838 1961250 2089561 3081402 3963317 2148905 2183879 2899998 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:2] "pocet_hostu" "pocet_prenocovani"
 - attr(*, "tsp")= num [1:3] 2000 2011 4
 - attr(*, "class")= chr [1:2] "mts" "ts"
```

Časové řady vykreslíme pomocí příkazu `plot()`.

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



Obrázek 6: *Návštěvnost v hromadných ubytovacích zařízeních v ČR v letech 2000–2010*

Uvažujme nejprve regresní model plné hodnosti (s $m = 1$)

$$\boxed{M_{II}}:\ Y_{jk} = \beta_0 + \beta_1 t + \cdots + \beta_m t^m + s_k + \varepsilon_{jk},\quad j = 1, \ldots, r \qquad k = 2, \ldots, d$$
$$t = (j-1)d+k$$

ve kterém se předpokládá, že $s_1 = 0$.

Ze dvou časových řad vybereme například první. Protože data nabývají hodnot v řádu miliónů, budeme je dále uvažovat v tisících. Z takto upravených hodnot vytvoříme nový datový rámec s kategoriálními proměnnými pro rok i sezónu, i když bychom v našem případě vystačili s jedinou kategoriální proměnnou, a to sezónou. Ale bude praktické mít datový rámec nachystaný i pro využití metod malého trendu.

```
> xts <- TS[, 1]/1000
> d <- frequency(xts)
> n <- length(xts)
> m <- ceiling(n/d)
> shift <- start(xts)[2] - 1
> r <- m * d - n - shift
> Season <- factor(as.integer(gl(d, 1, n)) - shift)
> Year <- factor(floor(time(xts)))
> Time <- (1:n) - mean(1:n)
> Data <- data.frame(x = as.vector(xts), grS = Season, grY = Year, Time = Time)
> str(Data)
```

```
'data.frame':       44 obs. of  4 variables:
 $ x   : num  2083 2916 3904 1961 2090 ...
 $ grS : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 4 1 2 ...
 $ grY : Factor w/ 11 levels "2000","2001",..: 1 1 1 1 2 2 2 2 3 3 ...
 $ Time: num  -21.5 -20.5 -19.5 -18.5 -17.5 -16.5 -15.5 -14.5 -13.5 -12.5 ...
```

V prostředí R použijeme pro odhad parametrů modelu M_{II} funkci `lm()`, ve které jsou pro kategoriální proměnné implicitně nastaveny kontrasty `contr.treatment`, takže je nemusíme uvádět.

```
> summary(MP1 <- lm(x ~ Time + grS, Data))
```

```
Call:
lm(formula = x ~ Time + grS, data = Data)

Residuals:
    Min      1Q  Median      3Q     Max 
-791.15  -91.76   25.92  101.63  365.18 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2242.113     60.124  37.292  < 2e-16 ***
Time          11.668      2.372   4.918 1.62e-05 ***
grS2         893.463     84.912  10.522 5.93e-13 ***
grS3        1951.342     85.011  22.954  < 2e-16 ***
grS4         114.688     85.176   1.346    0.186    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 199.1 on 39 degrees of freedom
Multiple R-squared: 0.9475,       Adjusted R-squared: 0.9421 
F-statistic: 175.8 on 4 and 39 DF,  p-value: < 2.2e-16
```

```
> anova(MP1)
```

```
Analysis of Variance Table

Response: x
          Df   Sum Sq Mean Sq F value    Pr(>F)
```

```
Time       1  1154275 1154275  29.131 3.552e-06 ***
grS        3 26716293 8905431 224.748 < 2.2e-16 ***
Residuals 39  1545337   39624
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

K vykreslení časové řady po jednotlivých složkách je třeba je nejprve získat, což se děje příkazem `predict()`.

```
> MP1.terms <- predict(MP1, type = "terms")
> str(MP1.terms)
```

```
 num [1:44, 1:2] -251 -239 -228 -216 -204 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:44] "1" "2" "3" "4" ...
  ..$ : chr [1:2] "Time" "grS"
 - attr(*, "constant")= num 2982
```

Také tentokrát, protože model obsahuje `(Intercept)`, fitované hodnoty jsou součtem dvou sloupců matice `MP1.terms` označených `Time` a `grS` a konstanty, kterou získáme volbou `attr(MP1.terms,"constant")`.

```
> pom <- cbind(MP1.terms, MP1.terms[, 1] + MP1.terms[, 2] + attr(MP1.terms,
     "constant"), fitted(MP1))
> colnames(pom) <- c(colnames(MP1.terms), "Time+grS+constant", "fitted(MP1)")
> print(pom[1:6, ])
```

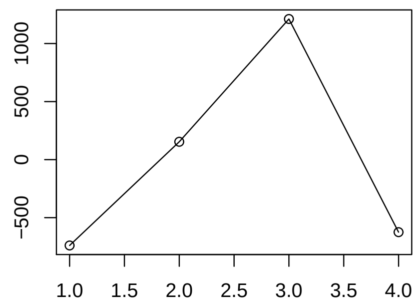
```
       Time       grS Time+grS+constant fitted(MP1)
1 -250.8655 -739.8733          1991.248    1991.248
2 -239.1973  153.5897          2896.379    2896.379
3 -227.5291 1211.4689          3965.926    3965.926
4 -215.8610 -625.1852          2140.940    2140.940
5 -204.1928 -739.8733          2037.920    2037.920
6 -192.5247  153.5897          2943.051    2943.051
```

Sezónní složky s_k vykreslíme do grafu.

```
> (MP1.sk <- MP1.terms[1:d, 2])
```

```
        1         2         3         4
-739.8733  153.5897 1211.4689 -625.1852
```

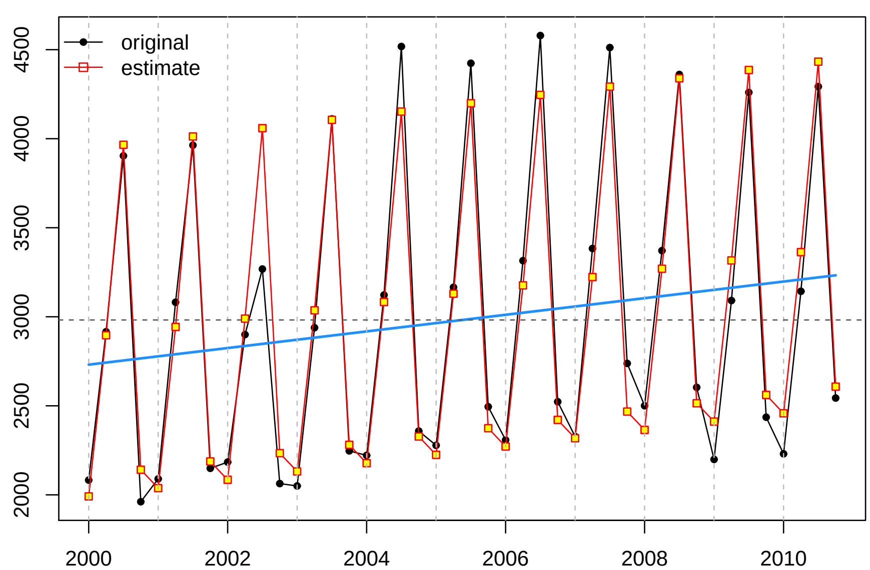
```
> SK <- MP1.sk
> nf <- layout(matrix(c(1, 2), nrow = 1), width = c(2.75, 1.25))
> txtS <- paste("S", 1:length(SK), " =", sep = "")
> par(mar = c(2, 2, 0, 0) + 0.05)
> plot(1:length(SK), SK, type = "o")
> plot(c(0, 1), c(0, length(SK)), type = "n", axes = FALSE)
> text(rep(0, length(SK)), rev(1:length(SK)), 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 7: Sezónní složky získané modelem M_{II} pro data *Návštěvnost (v tisících) v hromadných ubytovacích zařízeních v ČR v letech 2000–2010 (počet hostů)*

Do jediného grafu se budeme snažit zakreslit fitované hodnoty a trendovou komponentu modelu.

```
> tt <- as.vector(time(xts))
> xx <- as.vector(xts)
> fit <- fitted(MP1)
> tr <- attr(MP1.terms, "constant") + MP1.terms[, 1]
> 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, col = "dodgerblue", lwd = 2)
> abline(v = as.numeric(as.character(levels(Data$grY))), col = "gray",
     lty = 2)
> abline(h = attr(MP1.terms, "constant"), lty = 2, col = "gray35")
> legend(par("usr")[1], par("usr")[4], bty = "n", legend = c("original",
     "estimate"), lty = c(1, 1), pch = c(20, 22), bg = c("black", "yellow"),
     col = c("black", "red"))
```



Obrázek 8: Dekompozice pomocí modelu M_{II} pro data *Návštěvnost (v tisících) v hromadných ubytovacích zařízeních v ČR v letech 2000–2010 (počet hostů)*

Mnohem vhodnějším modelem vzhledem k interpretaci sezónních složek bude jistě model, ve kterém dodatečnou podmínku $s_1 = 0$ nahradíme součtovou podmínku

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

a budeme uvažovat modifikovaný model ve tvaru (s $m = 1$)

$$\boxed{M_{II}^{modif}}: \; Y_{jk} = \beta_0 + \beta_1 t + \cdots + \beta_m t^m + s_k + \varepsilon_{jk}, \quad \begin{array}{l} j = 1, \ldots, r \qquad k = 1, \ldots, d-1 \\ t = (j-1)d+k \end{array}$$

V prostředí R použijeme pro odhad parametrů tohoto modelu funkci `lm()`, ve které vhodně nastavíme kontrasty.

```
> Time <- (1:n) - mean(1:n)
> summary(MP1m <- lm(x ~ Time + grS, Data, contrasts = list(grS = contr.sum)))


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

Residuals:
    Min      1Q  Median      3Q     Max 
-791.15  -91.76   25.92  101.63  365.18 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2981.986     30.009  99.369  < 2e-16 ***
Time          11.668      2.372   4.918 1.62e-05 ***
grS1        -739.873     52.099 -14.201  < 2e-16 ***
grS2         153.590     51.991   2.954  0.00529 ** 
grS3        1211.469     51.991  23.302  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 199.1 on 39 degrees of freedom
Multiple R-squared: 0.9475,	Adjusted R-squared: 0.9421 
F-statistic: 175.8 on 4 and 39 DF,  p-value: < 2.2e-16 


> anova(MP1m)


Analysis of Variance Table

Response: x
          Df   Sum Sq Mean Sq F value    Pr(>F)    
Time       1  1154275 1154275  29.131 3.552e-06 ***
grS        3 26716293 8905431 224.748 < 2.2e-16 ***
Residuals 39  1545337   39624                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Opět, abychom byli schopni vykreslit dekompozici časové řady po jednotlivých složkách, použijeme příkaz `predict()` k jejich získání.

```
> MP1m.terms <- predict(MP1m, type = "terms")
> str(MP1m.terms)
```

```
 num [1:44, 1:2] -251 -239 -228 -216 -204 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:44] "1" "2" "3" "4" ...
  ..$ : chr [1:2] "Time" "grS"
 - attr(*, "constant")= num 2982
```

Také tentokrát, protože model obsahuje `(Intercept)`, fitované hodnoty jsou součtem dvou sloupců matice `MP1m.terms` označených `Time` a `grS` a konstanty, kterou získáme volbou `attr(MP1m.terms,"constant")`.

```
> pom <- cbind(MP1m.terms, MP1.terms[, 1] + MP1m.terms[, 2] + attr(MP1m.terms,
     "constant"), fitted(MP1m))
> colnames(pom) <- c(colnames(MP1m.terms), "Time+grS+constant", "fitted(MP1m)")
> print(pom[1:6, ])
```

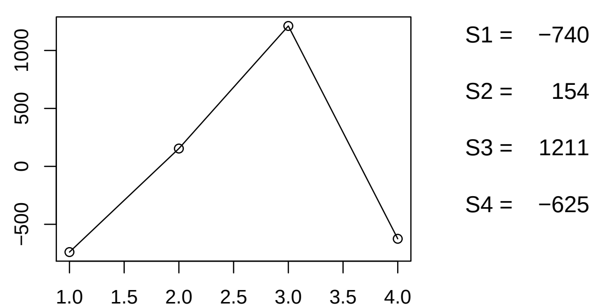
```
       Time       grS Time+grS+constant fitted(MP1m)
1 -250.8655 -739.8733          1991.248     1991.248
2 -239.1973  153.5897          2896.379     2896.379
3 -227.5291 1211.4689          3965.926     3965.926
4 -215.8610 -625.1852          2140.940     2140.940
5 -204.1928 -739.8733          2037.920     2037.920
6 -192.5247  153.5897          2943.051     2943.051
```

Sezónní kolísání s_k vykresleme opět do grafu.

```
> (MP1m.sk <- MP1m.terms[1:d, 2])
```

```
        1         2         3         4
-739.8733  153.5897 1211.4689 -625.1852
```

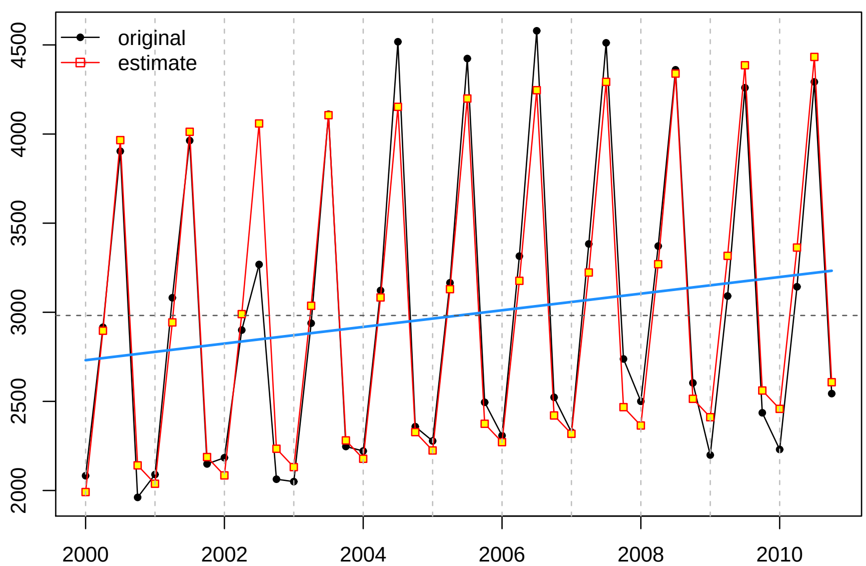
```
> SK <- MP1m.sk
> nf <- layout(matrix(c(1, 2), nrow = 1), width = c(2.75, 1.25))
> txtS <- paste("S", 1:length(SK), " =", sep = "")
> par(mar = c(2, 2, 0, 0) + 0.05)
> plot(1:length(SK), SK, type = "o")
> plot(c(0, 1), c(0, length(SK)), type = "n", axes = FALSE)
> text(rep(0, length(SK)), rev(1:length(SK)), 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 9: Sezónní složky získané modelem M_{II}^{modif} pro data *Návštěvnost (v tisících) v hromadných ubytovacích zařízeních v ČR v letech 2000–2010 (počet hostů)*

Do jediného grafu se zakreslíme fitované hodnoty a trendovou komponentu modelu.

```
> tt <- as.vector(time(xts))
> xx <- as.vector(xts)
> fit <- fitted(MP1m)
> tr <- attr(MP1m.terms, "constant") + MP1m.terms[, 1]
> 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, col = "dodgerblue", lwd = 2)
> abline(v = as.numeric(as.character(levels(Data$grY))), col = "gray",
    lty = 2)
> abline(h = attr(MP1m.terms, "constant"), lty = 2, col = "gray35")
> legend(par("usr")[1], par("usr")[4], bty = "n", legend = c("original",
    "estimate"), lty = c(1, 1), pch = c(20, 22), bg = c("black", "yellow"),
    col = c("black", "red"))
```



Obrázek 10: Dekompozice pomocí modelu M_{II}^{modif} pro data *Návštěvnost (v tisících) v hromadných ubytovacích zařízeních v ČR v letech 2000–2010 (počet hostů)*

Nyní ještě vytvoříme funkci `SzPolTrend()`, která pro zadaný stupeň polynomu provede dekompozici časové řady podle modelu M_{II}^{modif}.

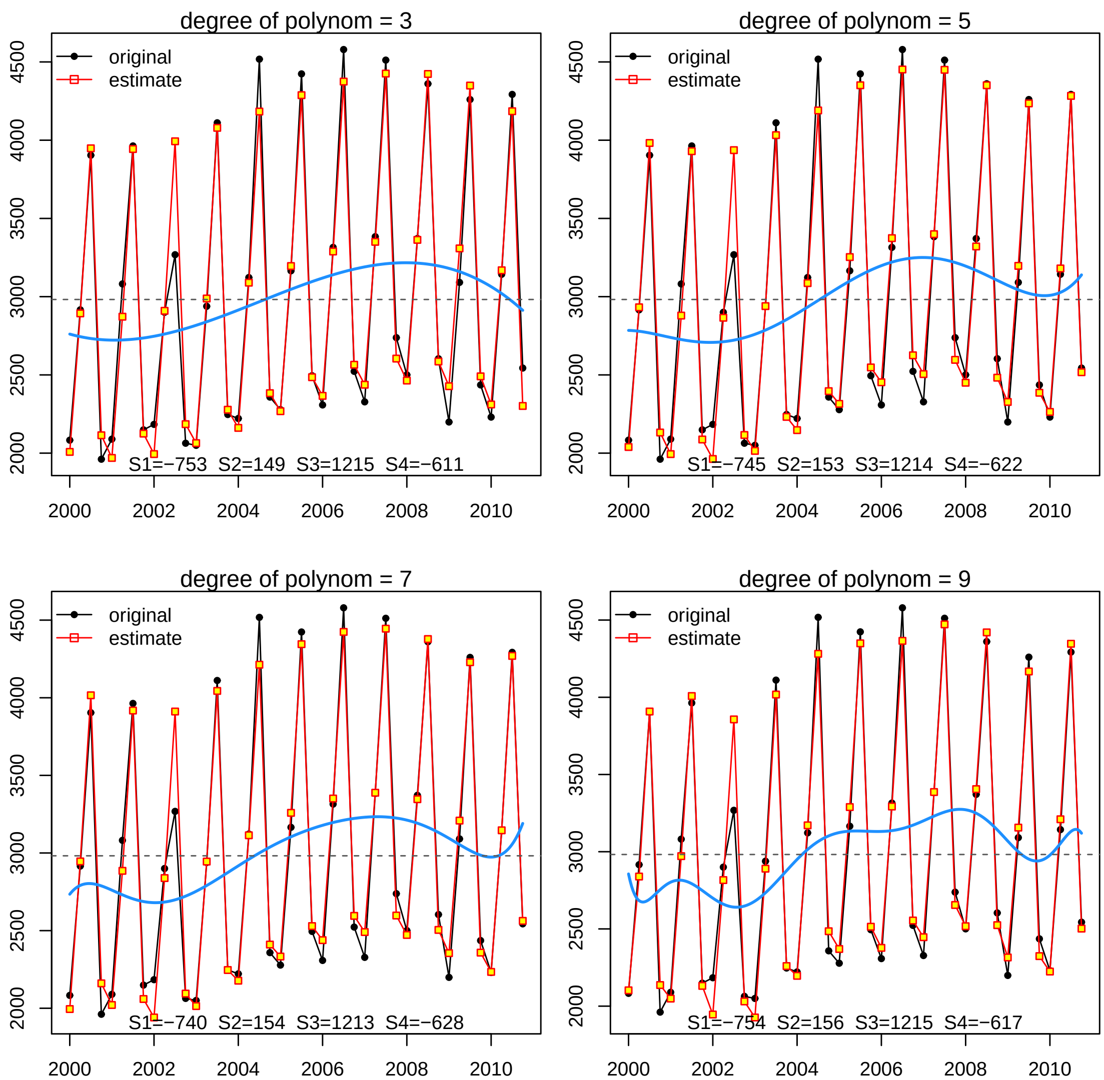
```
> SzPolTrend <- function(xts, Dgr = 1, figure = TRUE) {
    if (class(xts) != "ts")
        stop("xts must be a time series (ts)")
    nn <- 300
    d <- frequency(xts)
    n <- length(xts)
    m <- ceiling(n/d)
    shift <- start(xts)[2] - 1
    r <- m * d - n - shift
    grS <- factor(as.integer(gl(d, 1, n)) - shift)
    grY <- factor(floor(time(xts)))
    Time <- (1:n) - mean(1:n)
    Data <- data.frame(x = as.vector(xts), grS = grS, Time = Time)
```

```
    T.terms <- "Time"
    if (Dgr > 1)
        T.terms <- c(T.terms, paste("I(Time^", 2:Dgr, ")", sep = ""))
    strF <- paste("x~", paste(T.terms, collapse = "+"), "+grS", sep = "")
    pomf <- as.formula(strF)
    M <- lm(pomf, Data, contrasts = list(grS = contr.sum))
    m.terms.grS <- predict(M, type = "terms", terms = "grS")
    m.sk <- m.terms.grS[1:d]
    if (figure) {
        tt <- as.vector(time(xts))
        xx <- as.vector(xts)
        TTime <- Time
        TT <- tt
        if (n < nn) {
            TTime <- seq(Time[1], Time[n], length.out = nn)
            TT <- seq(tt[1], tt[n], length.out = nn)
        }
        newdata <- data.frame(Time = TTime, grS = rep(grS[1:d], length = length(TTime)))
        m.terms.Time <- predict(M, newdata = newdata, type = "terms",
            terms = T.terms)
        Constant <- attr(m.terms.Time, "constant")
        tr <- Constant + apply(m.terms.Time, 1, sum)
        fit <- fitted(M)
        ylim <- range(range(xx), range(fit), tr)
        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, col = "dodgerblue", lwd = 2)
        abline(v = as.numeric(as.character(levels(Data$grY))), col = "gray",
            lty = 2)
        abline(h = Constant, lty = 2, col = "gray35")
        legend(FindPositionForLegend(xx), bty = "n", legend = c("original",
            "estimate"), lty = c(1, 1), pch = c(20, 22), bg = c("black",
            "yellow"), col = c("black", "red"))
    }
    return(list(model = M, sk = m.sk))
}
```

Funkci vyzkoušíme u časové řady uložené v proměnné `xts`, a to pro různé stupně polynomu (například 3, 5, 7, 9).

Výsledky necháme vykreslit do grafu. Kromě toho do každého grafu doplníme odhady sezónních složek, abychom mohli posoudit, jak velký vliv má volba stupně polynomu na odhady sezónních složek.

```
> par(mfrow = c(2, 2), mar = c(2, 2, 2, 0) + 0.5)
> for (dgr in c(3, 5, 7, 9)) {
    vysl <- SzPolTrend(xts, Dgr = dgr)
    txtS <- paste(paste("S", 1:length(vysl$sk), "=", round(vysl$sk),
        sep = ""), collapse = "  ")
    txtDgr <- paste("degree of polynom =", dgr)
    mtext(txtDgr)
    mtext(txtS, side = 1, line = -1, cex = 0.85)
}
```



Obrázek 11: Dekompozice pomocí M_{II}^{modif} pro různé stupně polynomu pro data *Návštěvnost (v tisících) v hromadných ubytovacích zařízeních v ČR v letech 2000–2010 (počet hostů)*

C. Funkce `decompose()` a `stl()` v prostředí R

Pro dekompozici časové řady jsou v prostředí R k dispozici dvě funkce `decompose()` a `stl()`.

(a) Funkce `decompose()` – *Classical Seasonal Decomposition by Moving Averages*

```
> dec1 <- decompose(xts)
> str(dec1)
```

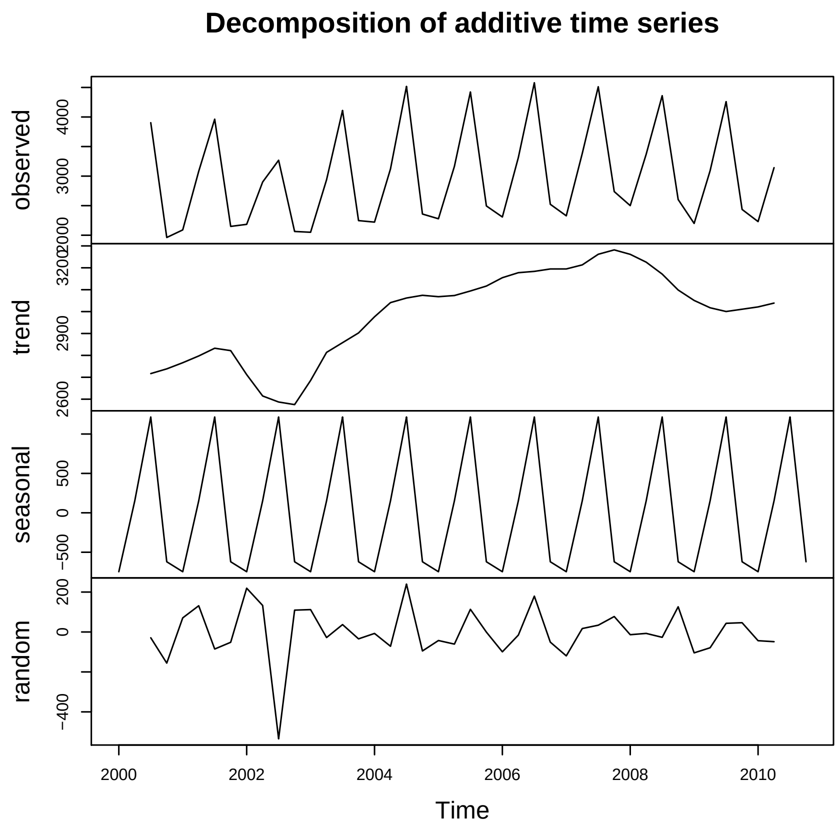
```
List of 5
 $ seasonal: Time-Series [1:44] from 2000 to 2011: -748 153 1216 -621 -748 ...
 $ trend   : Time-Series [1:44] from 2000 to 2011: NA NA 2717 2738 2766 ...
```

```
 $ random   : Time-Series [1:44] from 2000 to 2011: NA NA -29 -155.8 70.6 ...
 $ figure   : num [1:4] -748 153 1216 -621
 $ type     : chr "additive"
 - attr(*, "class")= chr "decomposed.ts"
```

```
> dec1$seasonal[1:d]
```

```
[1] -747.5080  152.6937 1216.0461 -621.2319
```

```
> plot(dec1)
```



Obrázek 12: Dekompozice pomocí funkce `decompose()` pro data *Návštěvnost (v tisících) v hromadných ubytovacích zařízeních v ČR v letech 2000–2010 (počet hostů)*

(b) Funkce `stl()` – *Seasonal Decomposition of Time Series by Loess*

```
> dec2 <- stl(xts, "periodic")
> str(dec2)
```

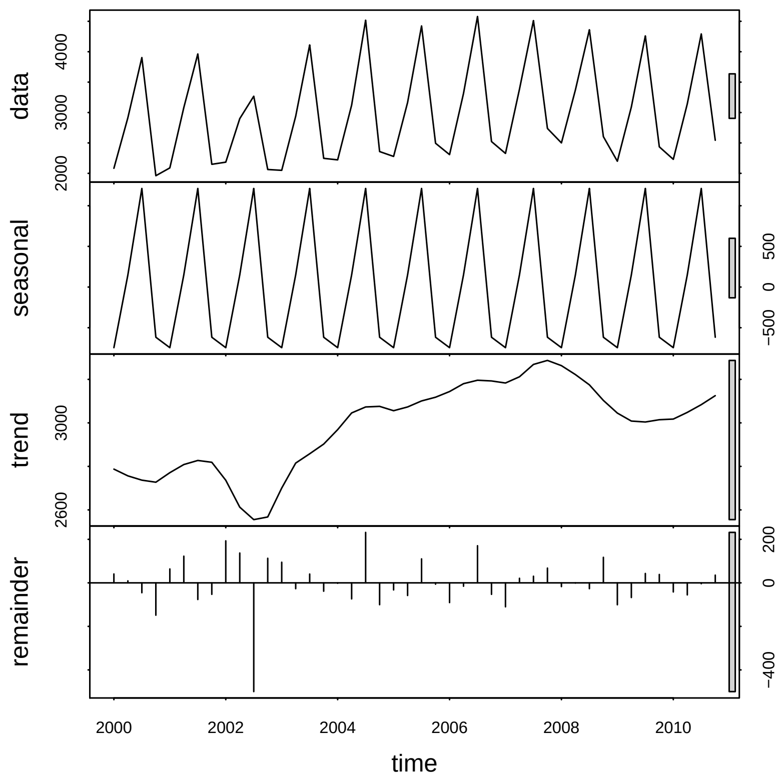
```
List of 8
 $ time.series: mts [1:44, 1:3] -745 150 1213 -617 -745 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:3] "seasonal" "trend" "remainder"
  ..- attr(*, "tsp")= num [1:3] 2000 2011 4
  ..- attr(*, "class")= chr [1:2] "mts" "ts"
 $ weights    : num [1:44] 1 1 1 1 1 1 1 1 1 1 ...
 $ call       : language stl(x = xts, s.window = "periodic")
 $ win        : Named num [1:3] 441 7 5
  ..- attr(*, "names")= chr [1:3] "s" "t" "l"
```

```
 $ deg        : Named int [1:3] 0 1 1
  ..- attr(*, "names")= chr [1:3] "s" "t" "l"
 $ jump       : Named num [1:3] 45 1 1
  ..- attr(*, "names")= chr [1:3] "s" "t" "l"
 $ inner      : int 2
 $ outer      : int 0
 - attr(*, "class")= chr "stl"
```

```
> dec2$time.series[1:d, 1]
```

```
[1] -745.4963  150.1728 1212.5610 -617.2366
```

```
> plot(dec2)
```



Obrázek 13: Dekompozice pomocí funkce `stl()` pro data *Návštěvnost (v tisících) v hromadných ubytovacích zařízeních v ČR v letech 2000–2010 (počet hostů)*

D. Úkoly:

(a) Na právě načtenou časovou řadu vyzkoušejte s využitím funckí `SzSmallTrend()` a `SzSmallTrendModif()` obě dvě metody malého trendu.

(b) Totéž vyzkoušejte na vhodné sezónní časové řadě, kterou si sami najdete.

(c) Na vaší časové řadě také odhadněte model M_{II}^{modif} pomocí funkce `SzPolTrend()`.

(d) Prostudujte a aplikujte funkce `decompose()` a `stl()` na vaši sezónní časovou řadu.