

## M0130 – 10. PRAKTIKUM : M0130pr10 (*Boxova–Jenkinsonova metodologie*)

### A. Generování kauzálních a invertibilních ARMA procesů.

1. Uvažujme nejprve **autoregresní** model  $p$ -tého řádu:  $Y_t - \varphi_1 Y_{t-1} - \dots - \varphi_p Y_{t-p} = \varepsilon_t$ , který ekvivalentně můžeme psát takto  $Y_t = \varphi_1 Y_{t-1} + \dots + \varphi_p Y_{t-p} + \varepsilon_t$ .

Zvolme konkrétní hodnoty  $p = 2$ ,  $\varphi_1 = 0.9$  a  $\varphi_2 = -0.2$ .

Zadáme-li model takto přímo, nevíme, zda jde o kauzální či nekauzální proces. To zjistíme, až budeme znát kořeny polynomu  $1 - \varphi_1 z - \dots - \varphi_p z^p$ . K tomu použijeme funkci `polyroot()`, která předpokládá koeficienty polynomu řádu  $p$  v pořadí  $a_0, a_1, \dots, a_p$ . Pokud budou kořeny vně jednotkové kružnice, AR proces bude kauzální.

Nejprve najdeme kořeny polynomu  $1 - \varphi_1 z - \varphi_2 z^2$  a zjistíme modulus kořenů.

```
> ARpar <- c(0.9, -0.2)
> print(ARroots <- polyroot(c(1, -ARpar)))
```

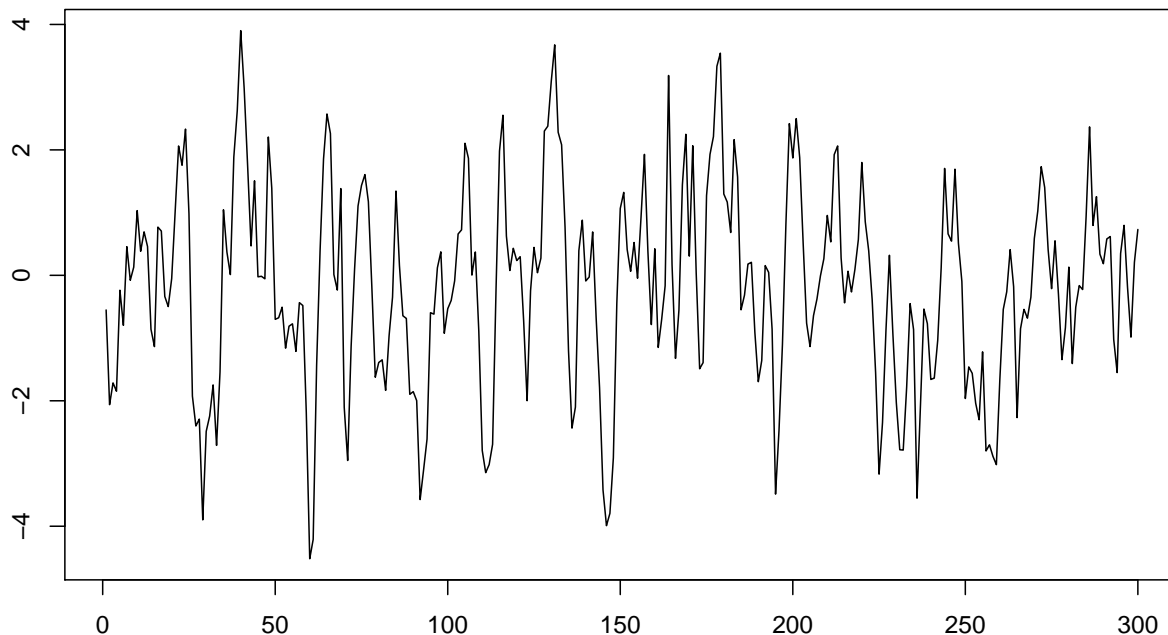
```
[1] 2.0+0i 2.5-0i
```

```
> print(Mod(ARroots))
```

```
[1] 2.0 2.5
```

Vidíme, že kořeny leží vně jednotkové kružnice, takže jde o kauzální AR proces. Pomocí funkce `arima.sim()` provedeme simulaci pro 300 hodnot a data vykreslíme.

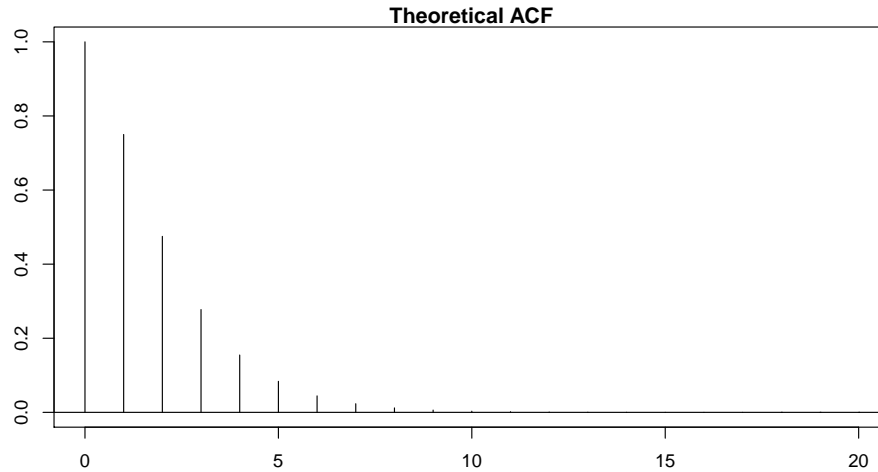
```
> ar.sim <- arima.sim(model = list(ar = ARpar), n = 300)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(ar.sim)
```



Obrázek 1: Simulovaná data AR procesu  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$

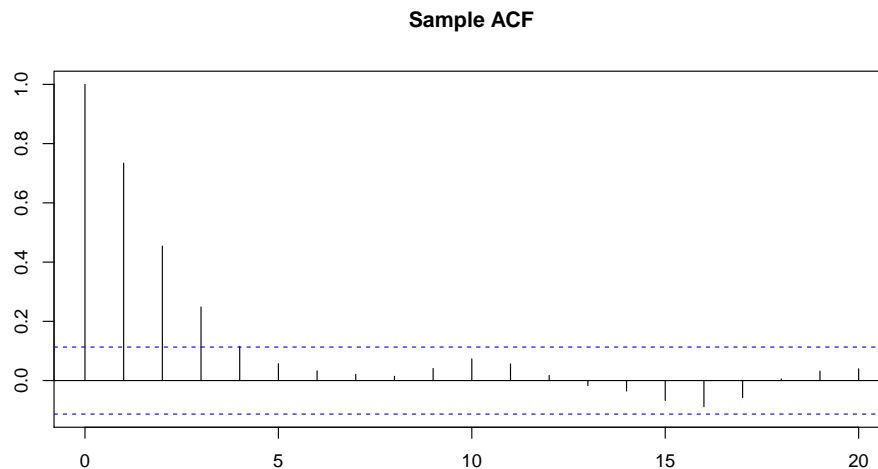
Pomocí funkce `ARMAacf()` můžeme vypočítat teoretickou autokorelační funkci ACF  $\rho(k)$  a teoretickou parciální autokorelační funkci PACF  $\alpha(k)$ . Výběrové protějšky těchto funkcí získáme pomocí funkcí `acf()`, popř. `pacf()`.

```
> tARacf <- ARMAacf(ar = ARpar, lag.max = 20)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(0:(length(tARacf) - 1), tARacf, type = "h", main = "Theoretical ACF")
> abline(h = 0)
```



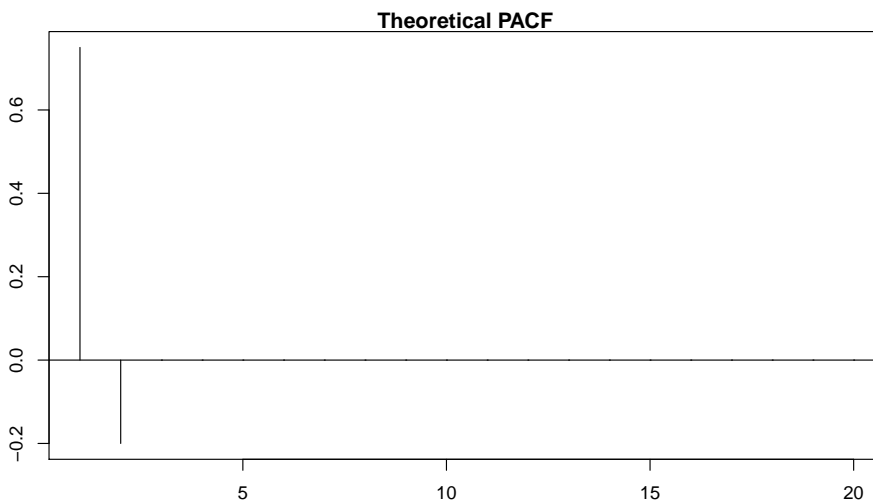
Obrázek 2: Teoretická ACF pro AR proces  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$

```
> vARacf <- acf(ar.sim, lag.max = 20, plot = FALSE)
> par(mfrow = c(1, 1), mar = c(2, 2, 3, 0) + 0.05)
> plot(vARacf, main = "Sample ACF")
```



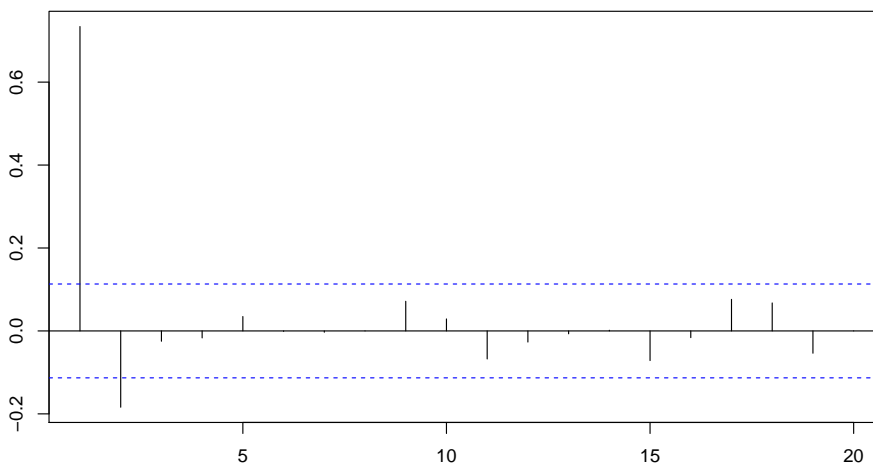
Obrázek 3: Výběrová ACF pro simulovaná data AR procesu  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$

```
> tARpacf <- ARMAacf(ar = ARpar, lag.max = 20, pacf = TRUE)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(1:length(tARpacf), tARpacf, type = "h", main = "Theoretical PACF")
> abline(h = 0)
```

Obrázek 4: Teoretická PACF pro AR proces  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$ 

```
> vARpacf <- pacf(ar.sim, lag.max = 20, plot = FALSE)
> par(mfrow = c(1, 1), mar = c(2, 2, 3, 0) + 0.05)
> plot(vARpacf, main = "Sample ACF")
```

Sample ACF

Obrázek 5: Výběrová PACF pro simulovaná data AR procesu  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$ 

Vedle grafického srovnání teoretických hodnot s odhady provedeme i numerické srovnání.

```
> print(data.frame(teor.ACF = tARacf[2:21], vyb.ACF = vARacf$acf[2:21,
  1, 1], teor.PACF = tARpacf, vyb.PACF = vARpacf$acf),
  digits = 6)
```

	teor.ACF	vyb.ACF	teor.PACF	vyb.PACF
1	7.50000e-01	0.73409109	7.50000e-01	7.34091e-01
2	4.75000e-01	0.45405586	-2.00000e-01	-1.83977e-01
3	2.77500e-01	0.24852575	1.16939e-16	-2.48247e-02
4	1.54750e-01	0.11541681	-2.29079e-17	-1.68247e-02
5	8.37750e-02	0.05714349	2.66066e-17	3.47389e-02
6	4.44475e-02	0.03303746	-6.22478e-18	-1.03889e-03
7	2.32478e-02	0.02098011	-1.28866e-17	-3.10573e-03
8	1.20335e-02	0.01425358	9.57914e-18	1.14509e-04
9	6.18058e-03	0.04090936	-6.11255e-18	7.14957e-02

```
10 3.15582e-03 0.07312471 3.09863e-18 2.89538e-02
11 1.60413e-03 0.05607936 -7.85119e-19 -6.75885e-02
12 8.12549e-04 0.01747044 2.89050e-19 -2.67941e-02
13 4.10469e-04 -0.01707631 -5.07968e-21 -6.96726e-03
14 2.06912e-04 -0.03531715 -6.96911e-20 1.67020e-03
15 1.04127e-04 -0.06734500 2.56505e-21 -7.14611e-02
16 5.23320e-05 -0.08828971 8.38367e-21 -1.61198e-02
17 2.62734e-05 -0.05755135 1.94584e-21 7.61432e-02
18 1.31796e-05 0.00536288 -2.10183e-21 6.75086e-02
19 6.60700e-06 0.03194232 -2.09671e-21 -5.37498e-02
20 3.31037e-06 0.03978868 2.17844e-21 5.61151e-05
```

Pomocí funkce `arima()` provedeme odhad parametrů  $AR(2)$  modelu.

```
> ar.fit <- arima(ar.sim, order = c(2, 0, 0))
> print(ar.fit)
```

Call:

```
arima(x = ar.sim, order = c(2, 0, 0))
```

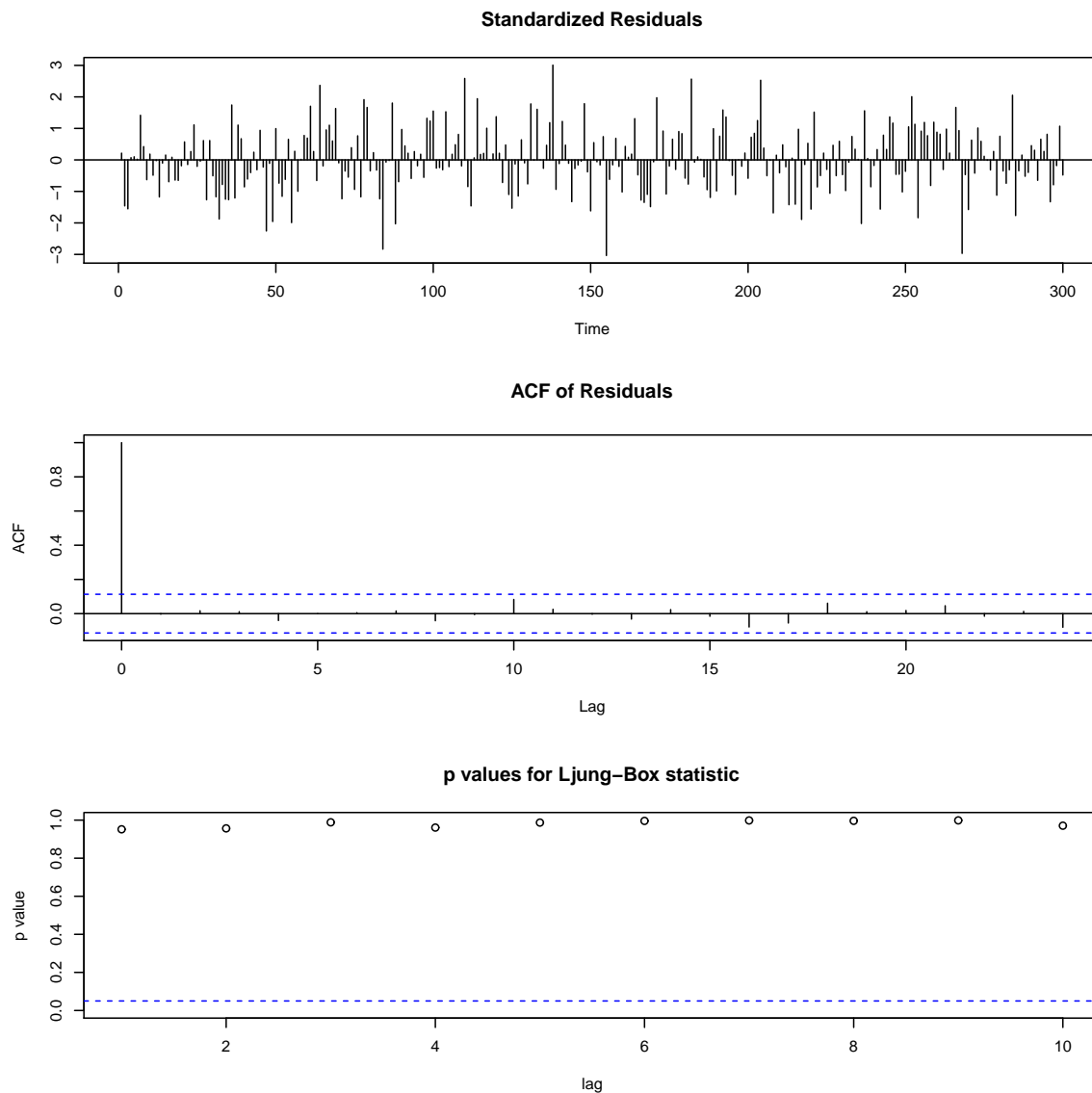
Coefficients:

```
      ar1      ar2 intercept
 0.8684 -0.1853   -0.1944
s.e. 0.0567 0.0568   0.1690
```

```
sigma^2 estimated as 0.8693: log likelihood = -405.1, aic = 818.2
```

Ve statistické knihovně máme ještě k dispozici funkci `tsdiag()`, kterou použije na odhadnutý AR model.

```
> tsdiag(ar.fit)
```



Obrázek 6: Diagnostické grafy pro odhadnutý MA(2) model pro simulovaná data AR procesu  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$

Protože chceme provést predikci pomocí příkazu `predict()` a výsledky budeme chtít zobrazit, nachystáme si nejprve funkci `PlotPredictARIMA`.

```
> PlotPredictARIMA <- function(x.ts, model.fit, n.ahead = 10) {
  arima.pred <- predict(model.fit, n.ahead = n.ahead)
  ylim <- c(min(x.ts, arima.pred$pred - 1.96 * arima.pred$se),
            max(x.ts, arima.pred$pred + 1.96 * arima.pred$se))
  opar <- par(mar = c(4, 4, 2, 2), las = 1)
  xlim <- c(range(time(x.ts))[1], range(time(arima.pred$pred))[2])
  par(mfrow = c(1, 1), mar = c(2, 2, 0, 0) + 0.05)
  plot(x.ts, ylim = ylim, type = "n", xlim = xlim)
  USR <- par("usr")
  rect(USR[1], USR[3], range(time(x.ts))[2], USR[4],
        border = NA, col = "lemonchiffon")
  rect(range(time(x.ts))[2], USR[3], USR[2], USR[4],
```

```

    border = NA, col = "lavender")
  rect(USR[1], USR[3], USR[2], USR[4], border = "black")
  polygon(c(time(arima.pred$pred), rev(time(arima.pred$pred))),
    c(arima.pred$pred - 1.96 * arima.pred$se, rev(arima.pred$pred +
      1.96 * arima.pred$se)), col = "orange", lty = 2,
    border = NA)
  lines(x.ts, lwd = 1)
  lines(arima.pred$pred, lwd = 1, col = "white")
}

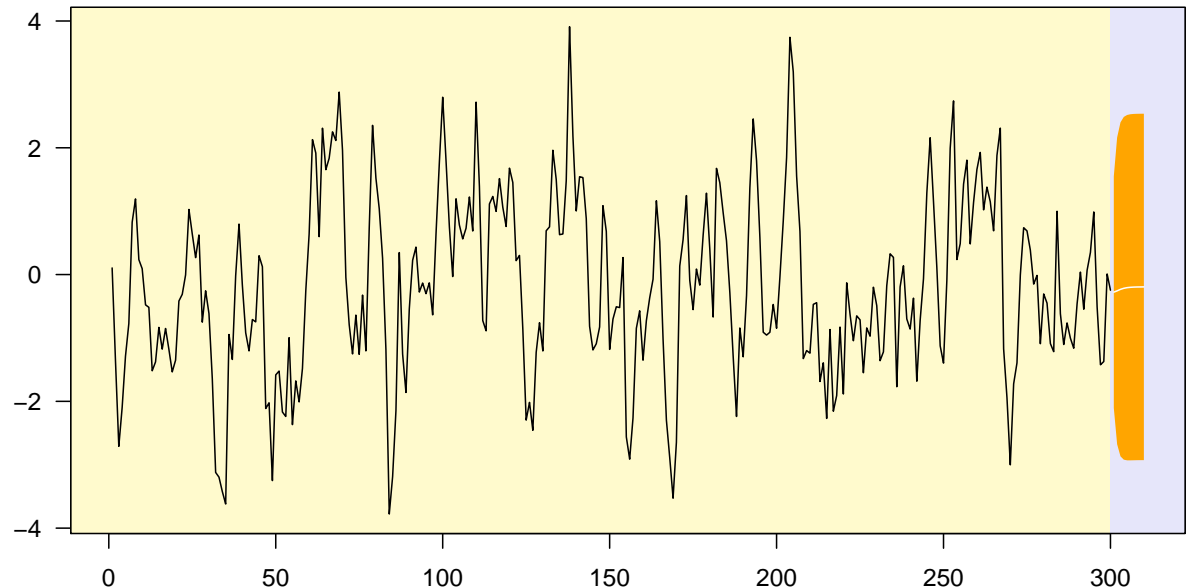
```

Když už máme funkci nachystanou, použijeme ji pro predikci o 10 kroků dopředu.

```

> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> PlotPredictARIMA(ar.sim, ar.fit, n.ahead = 10)

```



Obrázek 7: Predikce na základě odhadnutého AR(2) modelu pro simulovaná data AR procesu  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$

V knihovně `forecast` máme k dispozici funkci `auto.arima()`, kterou použijeme na naše simulovaná data.

```

> library(forecast)

'tseries' version: 0.10-25

'tseries' is a package for time series analysis and
computational finance.

See 'library(help="tseries")' for details.

```

This is forecast 2.16

```

> ar.fitF <- auto.arima(ar.sim, trace = TRUE)

ARIMA(2,0,2) with non-zero mean : 820.7214
ARIMA(0,0,0) with non-zero mean : 1056.964
ARIMA(1,0,0) with non-zero mean : 827.6271
ARIMA(0,0,1) with non-zero mean : 885.546
ARIMA(1,0,2) with non-zero mean : 821.2754
ARIMA(3,0,2) with non-zero mean : 822.2589

```

```

ARIMA(2,0,1) with non-zero mean : 818.8484
ARIMA(2,0,1) with zero mean      : 818.0474
ARIMA(1,0,1) with zero mean      : 819.3603
ARIMA(3,0,1) with zero mean      : 819.3125
ARIMA(2,0,0) with zero mean      : 817.106
ARIMA(2,0,0) with non-zero mean  : 817.9442
ARIMA(1,0,0) with zero mean      : 826.5675
ARIMA(3,0,0) with zero mean      : 817.313

```

```
Best model: ARIMA(2,0,0) with zero mean
```

```
> print(ar.fitF)
```

```
Series: ar.sim
ARIMA(2,0,0) with zero mean
```

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

```
Coefficients:
```

```

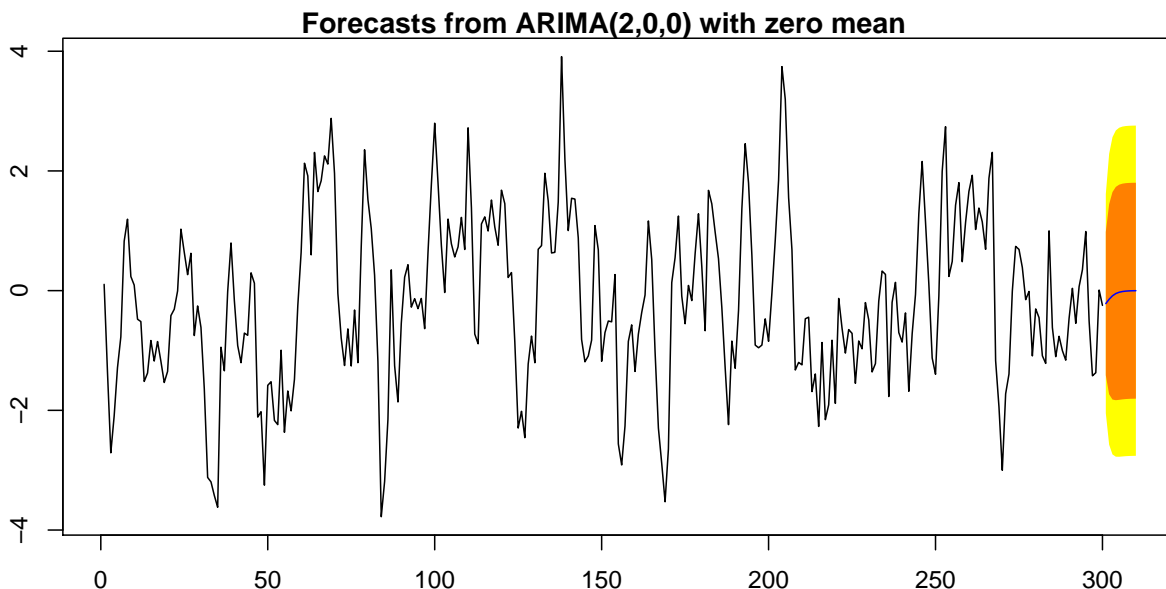
      ar1      ar2
 0.8721 -0.1821
s.e. 0.0567 0.0568

```

```
sigma^2 estimated as 0.873: log likelihood = -405.75
AIC = 817.49  AICc = 817.57  BIC = 828.6
```

V knihovně `forecast` máme ještě k dispozici funkci `forecast()`, kterou použije na odhadnutý AR model.

```
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(forecast(ar.fitF), n = 10)
```



Obrázek 8: Predikce na základě odhadnutého ARIMA(2,0,0) modelu pro simulovaná data AR procesu  $Y_t = 0.9Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t$

- Uvažujme nyní MA model  $q$ -tého řádu:  $Y_t = \varepsilon_t + \theta_1\varepsilon_{t-1} + \dots + \theta_q\varepsilon_{t-q}$ .

Zvolme konkrétní hodnoty  $q = 2$ ,  $\theta_1 = 0.75$  a  $\theta_2 = 0.25$ .

Zadáme-li model takto přímo, nevíme, zda jde o kauzální či nekauzální proces. To zjistíme, až budeme znát kořeny polynomu  $1 + \theta_1 z + \dots + \theta_q z^q$ . K tomu použijeme funkci `polyroot()`, která předpokládá koeficienty polynomu řádu  $p$  v pořadí  $a_0, a_1, \dots, a_p$ . Pokud budou kořeny vně jednotkové kružnice, MA proces bude invertibilní.

Nejprve najdeme kořeny polynomu  $1 + \theta_1 z + \theta_2 z^2$  a zjistíme modulus kořenů.

```
> MApar <- c(0.75, 0.25)
> print(MAroots <- polyroot(c(1, MApar)))
```

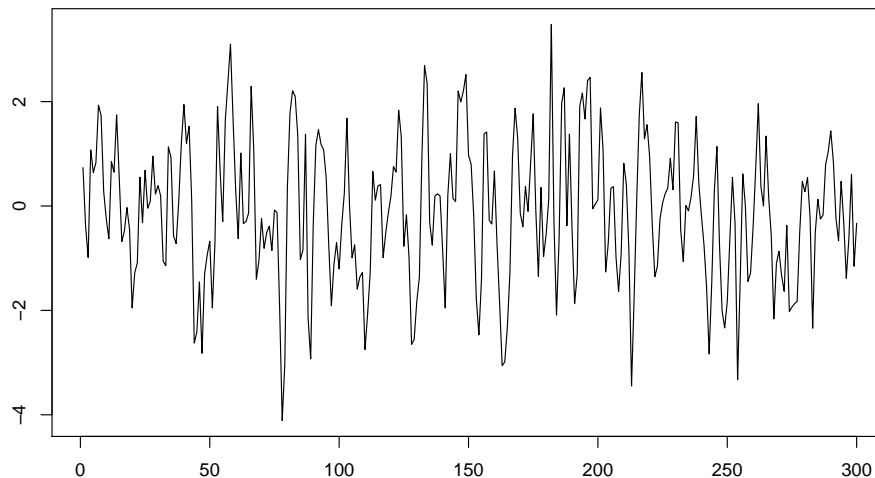
```
[1] -1.5+1.322876i -1.5-1.322876i
```

```
> print(Mod(MAroots))
```

```
[1] 2 2
```

Vidíme, že kořeny leží vně jednotkové kružnice, takže jde o invertibilní MA proces. Pomocí funkce `arima.sim()` provedeme simulaci pro 300 hodnot a data vykreslíme.

```
> ma.sim <- arima.sim(model = list(ma = MApar), n = 300)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(ma.sim)
```

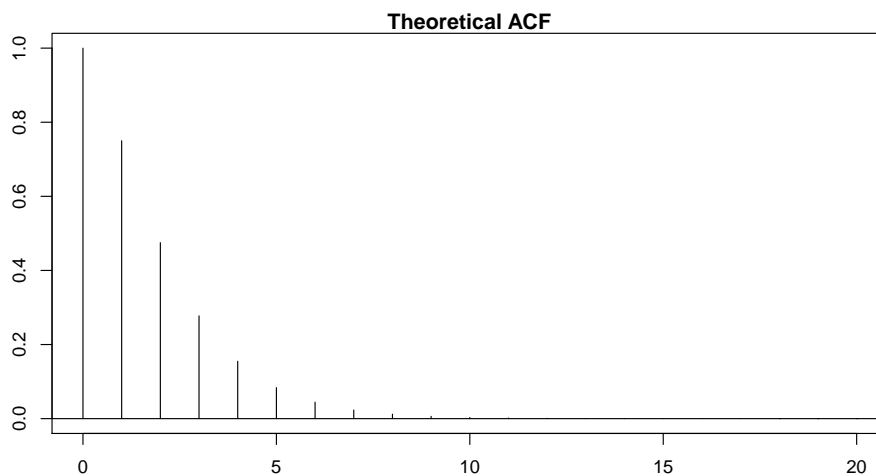


Obrázek 9: Simulovaná data MA procesu  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

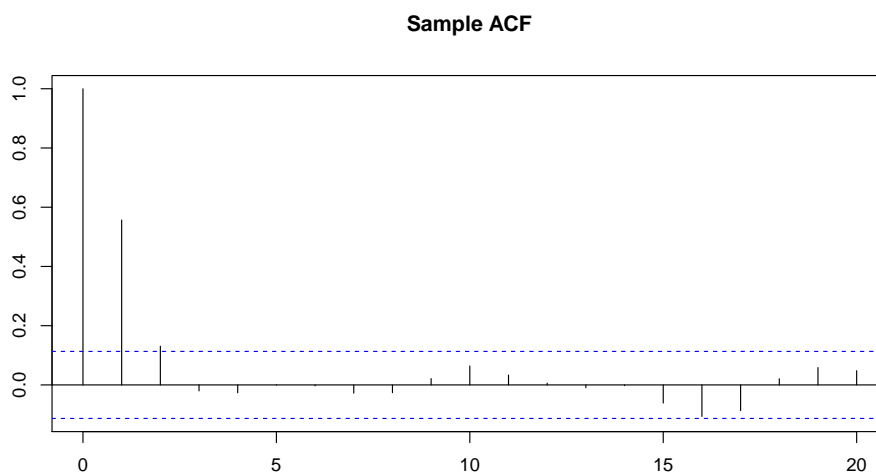
Pomocí funkce `ARMAacf()` můžeme vypočítat teoretickou autokorelační funkci ACF  $\rho(k)$  a teoretickou parciální autokorelační funkci PACF  $\alpha(k)$ . Výběrové protějšky těchto funkcí získáme pomocí funkcí `acf()`, popř. `pacf()`.

```
> tMAacf <- ARMAacf(ma = MApar, lag.max = 20)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(0:(length(tMAacf) - 1), tARacf, type = "h", main = "Theoretical ACF")
> abline(h = 0)
```

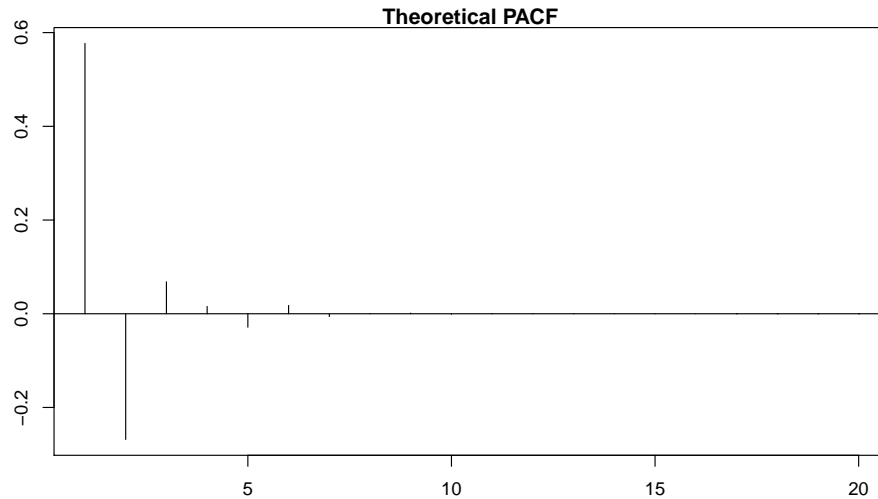


Obrázek 10: Teoretická ACF pro MA proces  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$ 

```
> vMAacf <- acf(ma.sim, lag.max = 20, plot = FALSE)
> par(mfrow = c(1, 1), mar = c(2, 2, 3, 0) + 0.05)
> plot(vMAacf, main = "Sample ACF")
```

Obrázek 11: Výběrová ACF pro simulovaná data MA procesu  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$ 

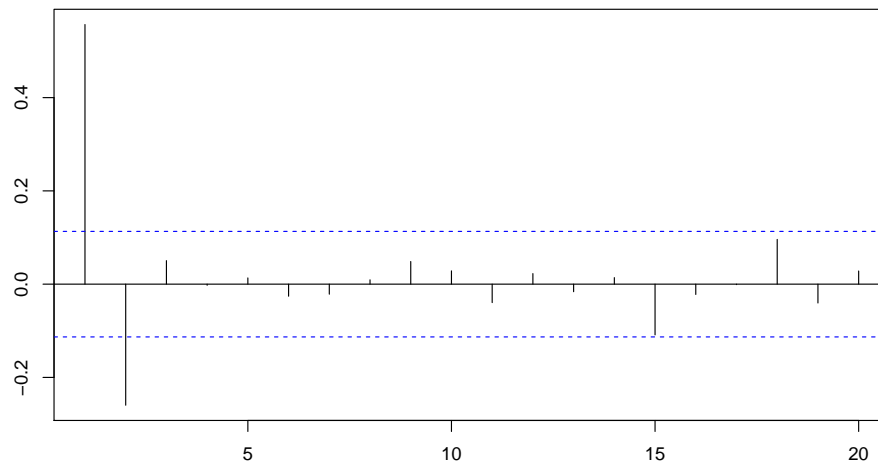
```
> tMApacf <- ARMAacf(ma = MApar, lag.max = 20, pacf = TRUE)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(1:length(tMApacf), tMApacf, type = "h", main = "Theoretical PACF")
> abline(h = 0)
```



Obrázek 12: Teoretická PACF pro MA proces  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

```
> vMApacf <- pacf(ma.sim, lag.max = 20, plot = FALSE)
> par(mfrow = c(1, 1), mar = c(2, 2, 3, 0) + 0.05)
> plot(vMApacf, main = "Sample ACF")
```

Sample ACF



Obrázek 13: Výběrová PACF pro simulovaná data MA procesu  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

Vedle grafického srovnání teoretických hodnot s odhady provedeme i numerické srovnání.

```
> print(data.frame(teor.ACF = tMAacf[2:21], vyb.ACF = vMAacf$acf[2:21],
  1, 1], teor.PACF = tMApacf, vyb.PACF = vMApacf$acf),
  digits = 6)
```

	teor.ACF	vyb.ACF	teor.PACF	vyb.PACF
1	0.576923	0.55676256	5.76923e-01	0.55676256
2	0.153846	0.13084133	-2.68293e-01	-0.25962204
3	0.000000	-0.02032322	6.81818e-02	0.05044899
4	0.000000	-0.02661347	1.56028e-02	-0.00246779
5	0.000000	0.00103477	-2.87138e-02	0.01337238
6	0.000000	-0.00323460	1.76361e-02	-0.02585695
7	0.000000	-0.02778686	-6.05108e-03	-0.02152588
8	0.000000	-0.02624942	1.29458e-04	0.00930449
9	0.000000	0.02138326	1.41567e-03	0.04857000

```

10 0.000000  0.06399888 -1.09412e-03  0.02849594
11 0.000000  0.03327434  4.66671e-04 -0.03959058
12 0.000000  0.00597268 -7.64737e-05  0.02266222
13 0.000000 -0.00938026 -5.93124e-05 -0.01619658
14 0.000000 -0.00254846  6.36028e-05  0.01404376
15 0.000000 -0.06121969 -3.28740e-05 -0.10875593
16 0.000000 -0.10636755  8.75478e-06 -0.02207376
17 0.000000 -0.08653195  1.65241e-06 -0.00101212
18 0.000000  0.02080609 -3.42800e-06  0.09595374
19 0.000000  0.05860587  2.15790e-06 -0.04040562
20 0.000000  0.04792110 -7.61424e-07  0.02821960

```

Pomocí funkce `arima()` provedeme odhad parametrů  $MA(2)$  modelu.

```

> ma.fit <- arima(ar.sim, order = c(0, 0, 2))
> print(ma.fit)

```

```

Series: ar.sim
ARIMA(0,0,2) with non-zero mean

Call: arima(x = ar.sim, order = c(0, 0, 2))

Coefficients:
      ma1      ma2  intercept
 0.7993  0.3665   -0.1967
s.e.  0.0519  0.0488    0.1206

sigma^2 estimated as 0.9342:  log likelihood = -415.83
AIC = 839.65   AICc = 839.79   BIC = 854.47

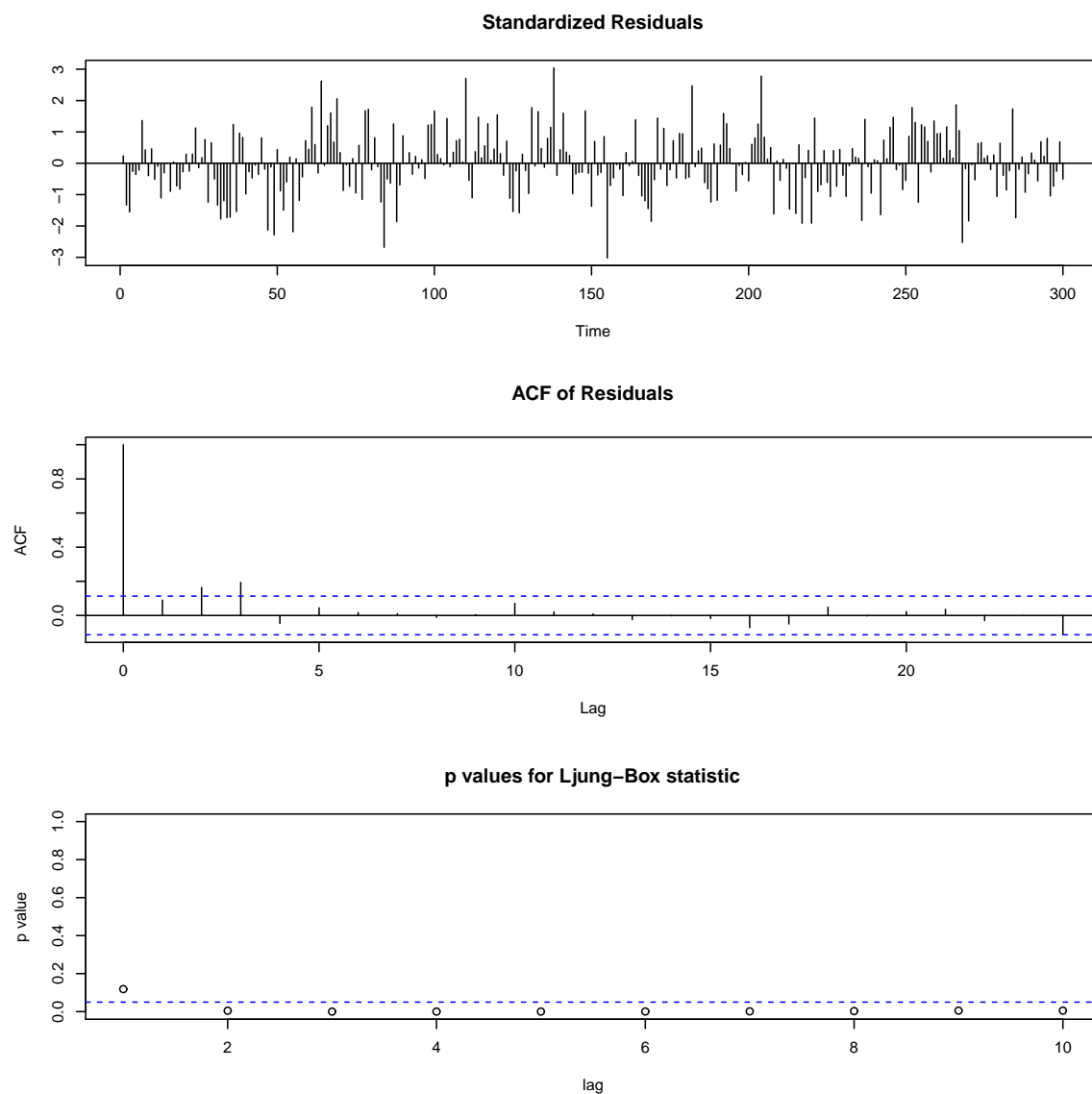
```

Ve statistické knihovně máme ještě k dispozici funkci `tsdiag()`, kterou použije na odhadnutý MA model.

```

> tsdiag(ma.fit)

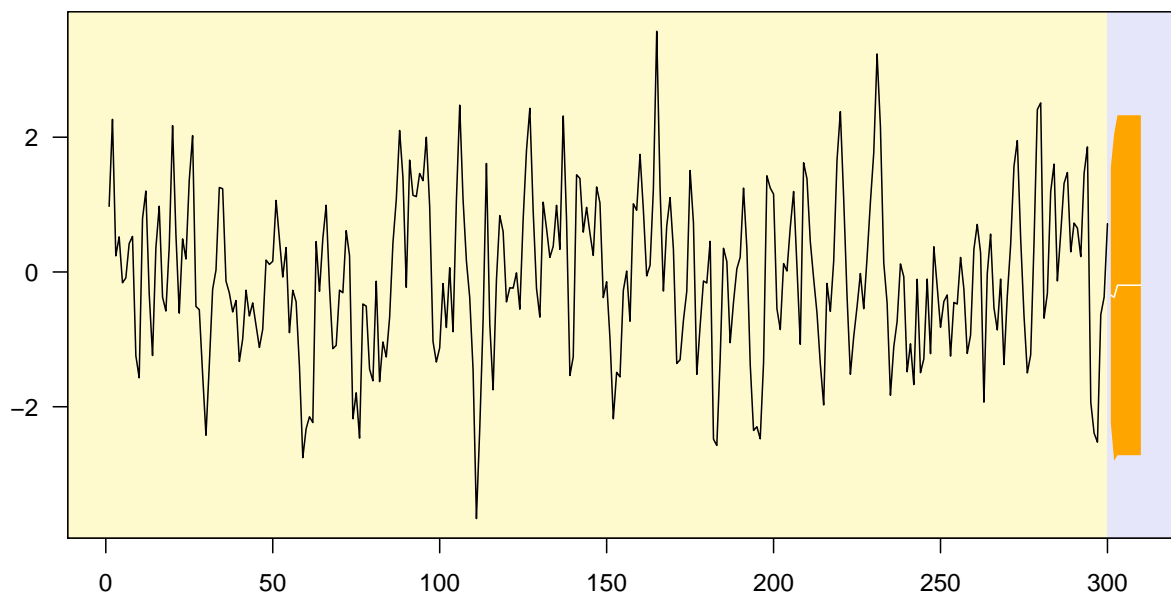
```



Obrázek 14: Diagnostické grafy pro odhadnutý MA(2) model pro simulovaná data MA procesu  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

Pro predikci o 10 kroků dopředu použijeme funkci `PlotPredictARIMA`.

```
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> PlotPredictARIMA(ma.sim, ma.fit, n.ahead = 10)
```



Obrázek 15: Predikce na základě odhadnutého AR(2) modelu pro simulovaná data MA procesu  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

V knihovně `forecast` máme k dispozici funkci `auto.arima()`, kterou použijeme na naše simulovaná data.

```
> ma.fitF <- auto.arima(ma.sim, trace = TRUE)

ARIMA(2,0,2) with non-zero mean : 829.5307
ARIMA(0,0,0) with non-zero mean : 958.9702
ARIMA(1,0,0) with non-zero mean : 849.3271
ARIMA(0,0,1) with non-zero mean : 839.6663
ARIMA(1,0,2) with non-zero mean : 834.5418
ARIMA(3,0,2) with non-zero mean : 831.7523
ARIMA(2,0,1) with non-zero mean : 828.3928
ARIMA(2,0,1) with zero mean      : 827.0926
ARIMA(1,0,1) with zero mean     : 831.0657
ARIMA(3,0,1) with zero mean     : 828.4951
ARIMA(2,0,0) with zero mean     : 826.9416
ARIMA(2,0,0) with non-zero mean : 828.1437
ARIMA(1,0,0) with zero mean     : 847.7004
ARIMA(3,0,0) with zero mean     : 827.8045

Best model: ARIMA(2,0,0) with zero mean

> print(ma.fitF)

Series: ma.sim
ARIMA(2,0,0) with zero mean

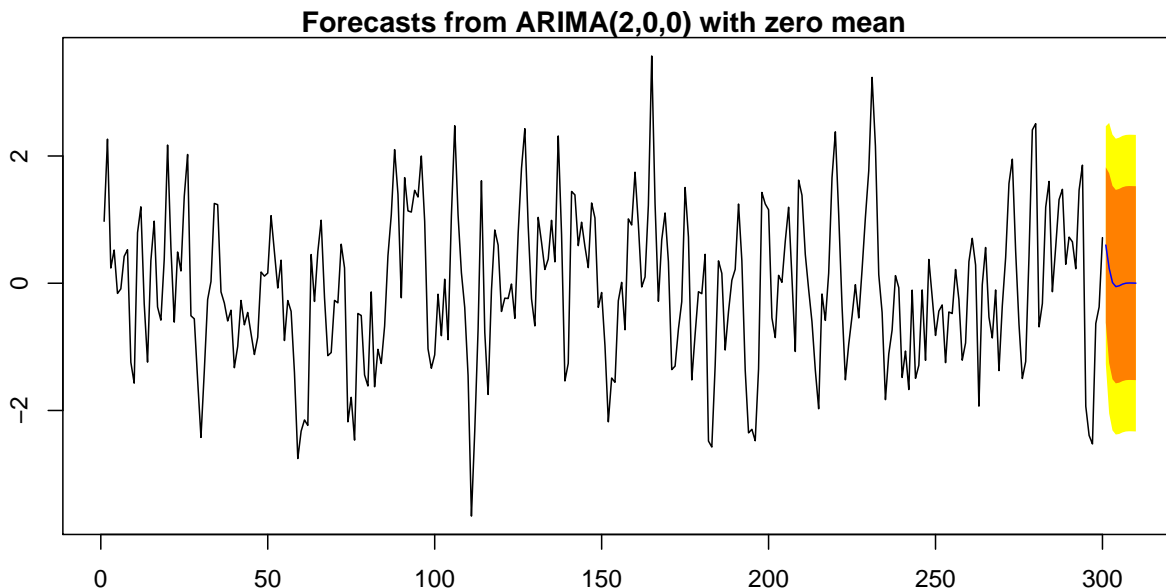
Call: auto.arima(x = ma.sim, trace = TRUE)

Coefficients:
      ar1      ar2
  0.7031 -0.2585
s.e. 0.0557 0.0560

sigma^2 estimated as 0.9077: log likelihood = -411.41
AIC = 828.82  AICc = 828.9  BIC = 839.93
```

Opět použijeme funkci `forecast()` pro predikci o 10 kroků dopředu.

```
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(forecast(ma.fitF), n = 10)
```



Obrázek 16: Predikce na základě odhadnutého ARIMA(2,0,0) modelu pro simulovaná data MA procesu  $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

3. Na závěr ještě prozkoumejme modely se sezónní složkou, například

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

kde

$$\varepsilon_t \sim WN(0, \sigma^2), \quad \sigma^2 = 1.75.$$

Pokud model rozepíšeme, dostaneme

$$Y_t = Y_{t-1} + Y_{t-12} + Y_{t-13} + \varepsilon_t - 0.5\varepsilon_{t-1} - 0.7\varepsilon_{t-12} + 0.35\varepsilon_{t-13}$$

Nejprve si vytvoříme vlastní funkci pro simulování SARIMA modelů.

```
> my.sarima.sim <- function(n = 20, period = 12, sd = 1,
  model, seasonal) {
  x <- arima.sim(model, n * period, sd = sd)
  x <- x[1:(n * period)]
  for (i in 1:period) {
    xx <- arima.sim(seasonal, n, sd = sd)
    xx <- xx[1:n]
    x[i + period * 0:(n - 1)] <- x[i + period * 0:(n -
      1)] + xx
  }
  x <- ts(x, frequency = period)
  x
}
```

Kromě toho si nachystáme velmi užitečnou funkci pro rychlé vykreslení simulovaných dat spolu s ACF, PACF funkcemi, také odhad spektra.

```

> eda.ts <- function(x, bands = FALSE) {
  op <- par(no.readonly = TRUE)
  par(mar = c(0, 0, 0, 0), oma = c(1, 4, 2, 1))
  p.min <- 0.05
  k <- 15
  p <- rep(NA, k)
  for (i in 1:k) {
    p[i] <- Box.test(x, i, type = "Ljung-Box")$p.value
  }
  if (max(p) > p.min) {
    par(mfrow = c(5, 1))
  }
  else {
    par(mfrow = c(4, 1))
  }
  if (!is.ts(x))
    x <- ts(x)
  plot(x, axes = FALSE)
  axis(2)
  axis(3)
  box(lwd = 2)
  if (bands) {
    a <- time(x)
    i1 <- floor(min(a))
    i2 <- ceiling(max(a))
    y1 <- par("usr")[3]
    y2 <- par("usr")[4]
    if (par("ylog")) {
      y1 <- 10^y1
      y2 <- 10^y2
    }
    for (i in seq(from = i1, to = i2 - 1, by = 2)) {
      polygon(c(i, i + 1, i + 1, i), c(y1, y1,
        y2, y2), col = "grey", border = NA)
    }
    lines(x)
  }
  acf(x, axes = FALSE)
  axis(2, las = 2)
  box(lwd = 2)
  mtext("ACF", side = 2, line = 2.5)
  pacf(x, axes = FALSE)
  axis(2, las = 2)
  box(lwd = 2)
  mtext("PACF", side = 2, line = 2.5)
  spectrum(x, col = par("fg"), log = "dB", main = "",
    axes = FALSE)
  axis(2, las = 2)
  box(lwd = 2)
  mtext("Spectrum", side = 2, line = 2.5)
  abline(v = 1, lty = 2, lwd = 2)
  abline(v = 2:10, lty = 3)
  abline(v = 1/2:5, lty = 3)
  if (max(p) > p.min) {
    main <- plot(p, type = "h", ylim = c(0, 1), lwd = 3,
      main = "", axes = F)
    axis(2, las = 2)
  }
}

```

```

    box(lwd = 2)
    mtext("Ljung-Box p-value", side = 2, line = 2.5)
    abline(h = c(0, 0.05), lty = 3)
  }
  par(op)
}

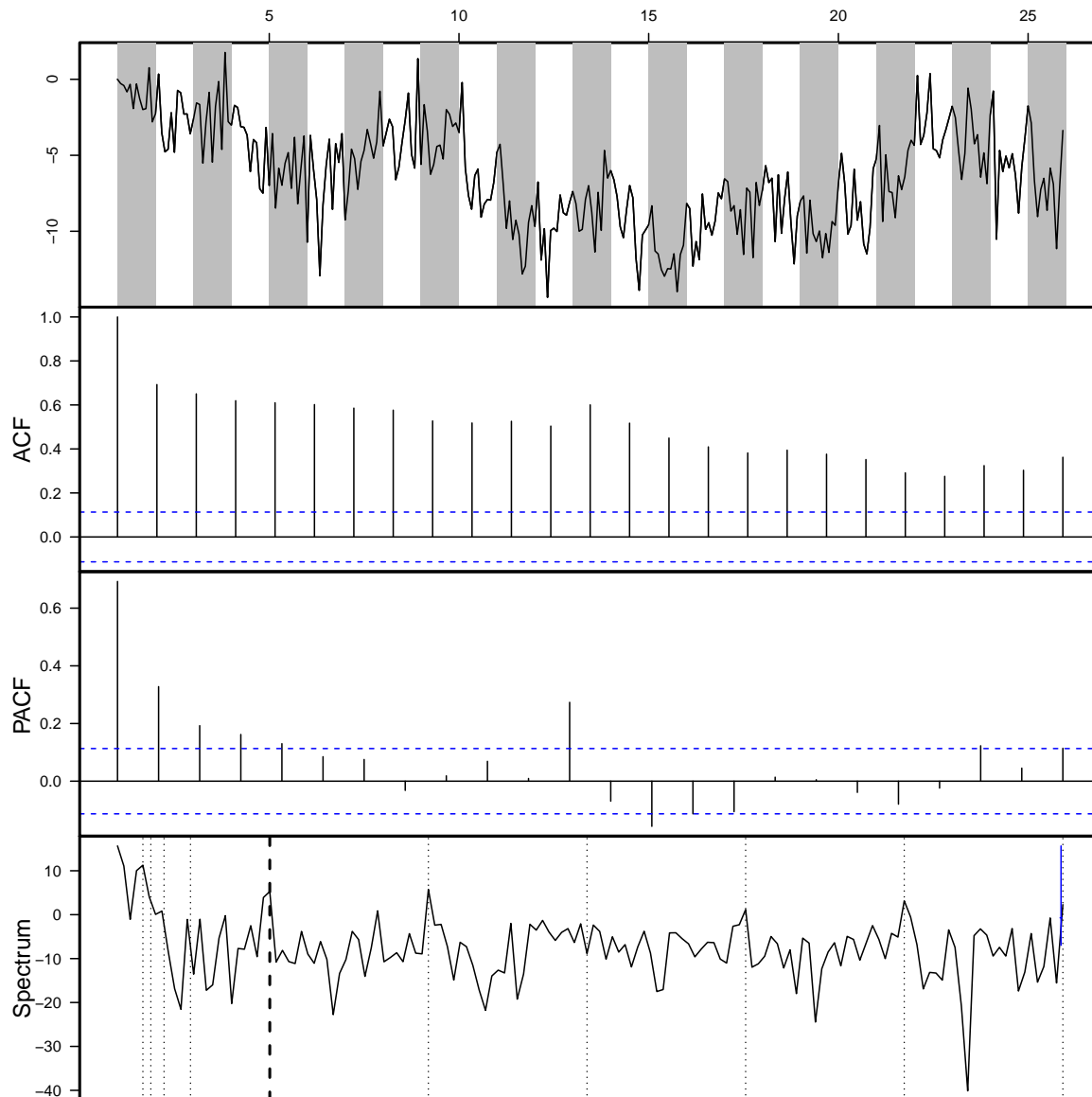
```

A nyní obě dvě nové funkce použijeme pro náš model

```

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

```



Obrázek 17: Simulovaná data pro  $SARIMA(0, 1, 1) \times (0, 1, 1)_{12} : (1-B)(1-B^{12})Y_t = (1-0.5B)(1-0.7B^{12})\varepsilon_t$

Provedeme 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))
```

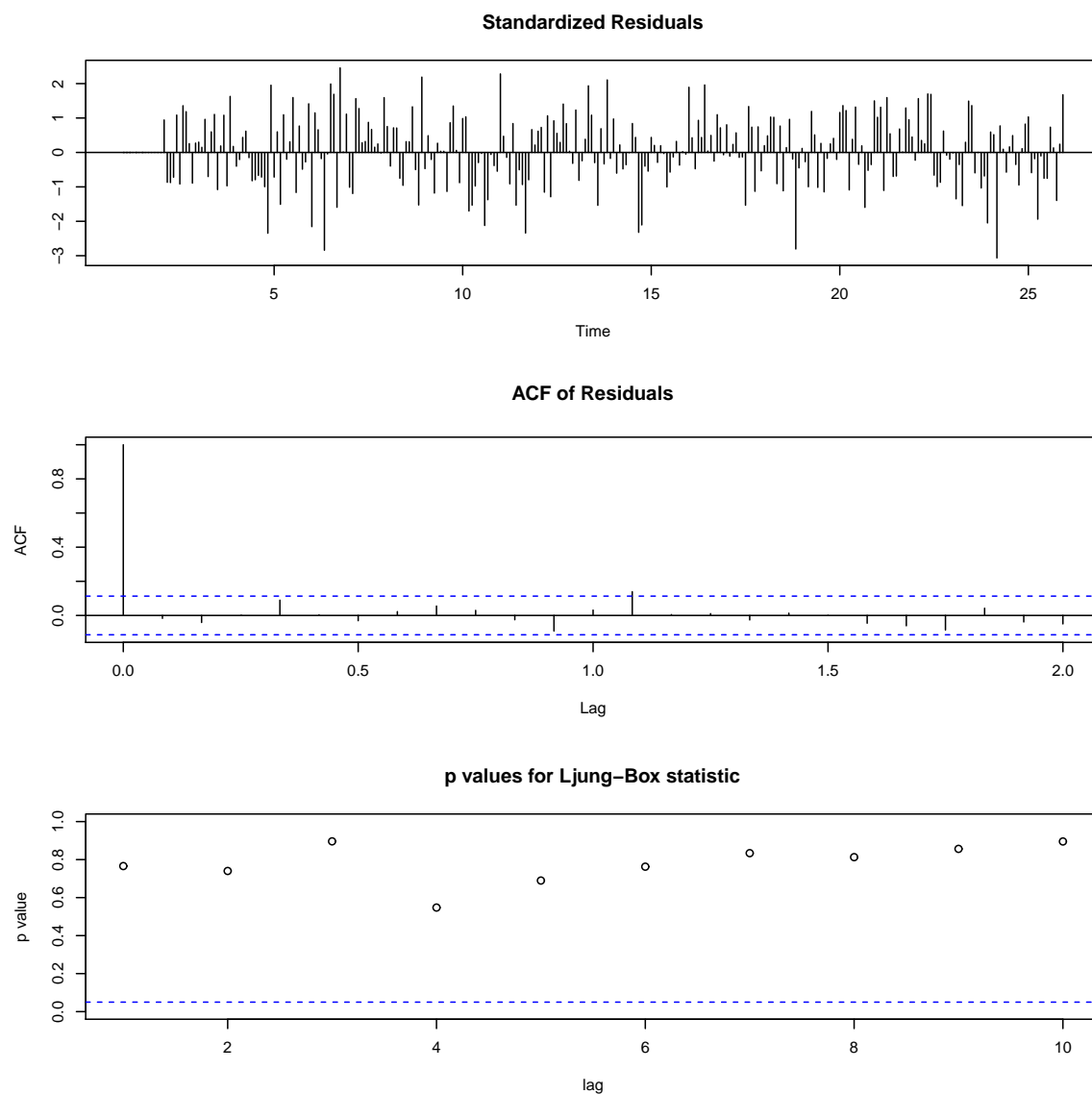
```
Coefficients:
```

```
      ma1      sma1
-0.7085  -0.8287
s.e.    0.0388   0.0430
```

```
sigma^2 estimated as 3.735: log likelihood = -603.67
AIC = 1213.34  AICc = 1213.42  BIC = 1224.32
```

Ve statistické knihovně máme ještě k dispozici funkci `tsdiag()`, kterou použije na odhadnutý SARIMA model.

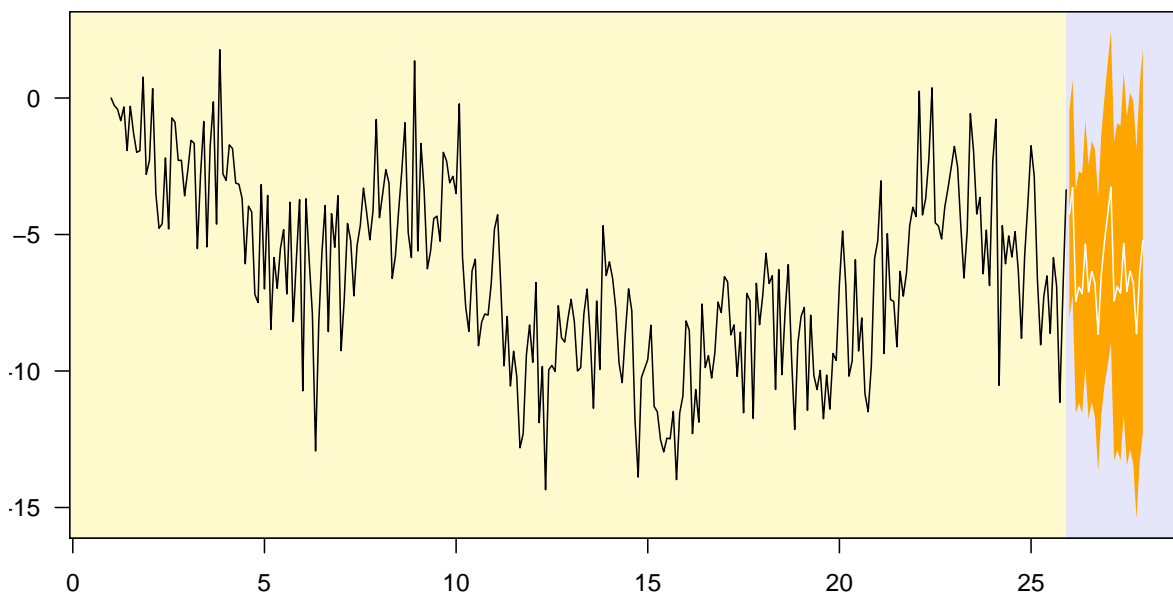
```
> tsdiag(sarima.fit)
```



Obrázek 18: 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 o 2 roky dopředu, a to pomocí příkazu `predict()`. 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 19: Predikce pomocí `predict()` 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 použijeme na naše simulovaná data.

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

```
ARIMA(2,1,2)(1,0,1)[12] with drift      : 1269.63
ARIMA(0,1,0) with drift                 : 1425.096
ARIMA(1,1,0)(1,0,0)[12] with drift      : 1338.143
ARIMA(0,1,1)(0,0,1)[12] with drift      : 1286.208
ARIMA(2,1,2)(0,0,1)[12] with drift      : 1293.502
ARIMA(2,1,2)(2,0,1)[12] with drift      : 1272.867
ARIMA(2,1,2)(1,0,0)[12] with drift      : 1289.397
ARIMA(2,1,2)(1,0,2)[12] with drift      : 1271.278
ARIMA(2,1,2) with drift                 : 1319.074
ARIMA(2,1,2)(2,0,2)[12] with drift      : 1270.713
ARIMA(1,1,2)(1,0,1)[12] with drift      : 1267.634
ARIMA(1,1,1)(1,0,1)[12] with drift      : 1272.444
ARIMA(1,1,3)(1,0,1)[12] with drift      : 1269.597
ARIMA(0,1,1)(1,0,1)[12] with drift      : 1266.325
ARIMA(0,1,1)(1,0,1)[12]                : 1264.451
ARIMA(0,1,1)(0,0,1)[12]                : 1284.568
ARIMA(0,1,1)(2,0,1)[12]                : 1263.100
ARIMA(0,1,1)(2,0,0)[12]                : 1278.173
ARIMA(0,1,1)(2,0,2)[12]                : 1260.893
ARIMA(1,1,1)(2,0,2)[12]                : 1264.184
ARIMA(0,1,0)(2,0,2)[12]                : 1380.871
ARIMA(0,1,2)(2,0,2)[12]                : 1262.876
ARIMA(1,1,2)(2,0,2)[12]                : 1266.007
ARIMA(0,1,1)(2,0,2)[12] with drift      : 1262.986
ARIMA(0,1,1)(1,0,2)[12]                : 1266.334
```

```
Best model: ARIMA(0,1,1)(2,0,2)[12]
```

```
> print(sarima.fitF)
```

```
Series: Sarima.sim
ARIMA(0,1,1)(2,0,2)[12]
```

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

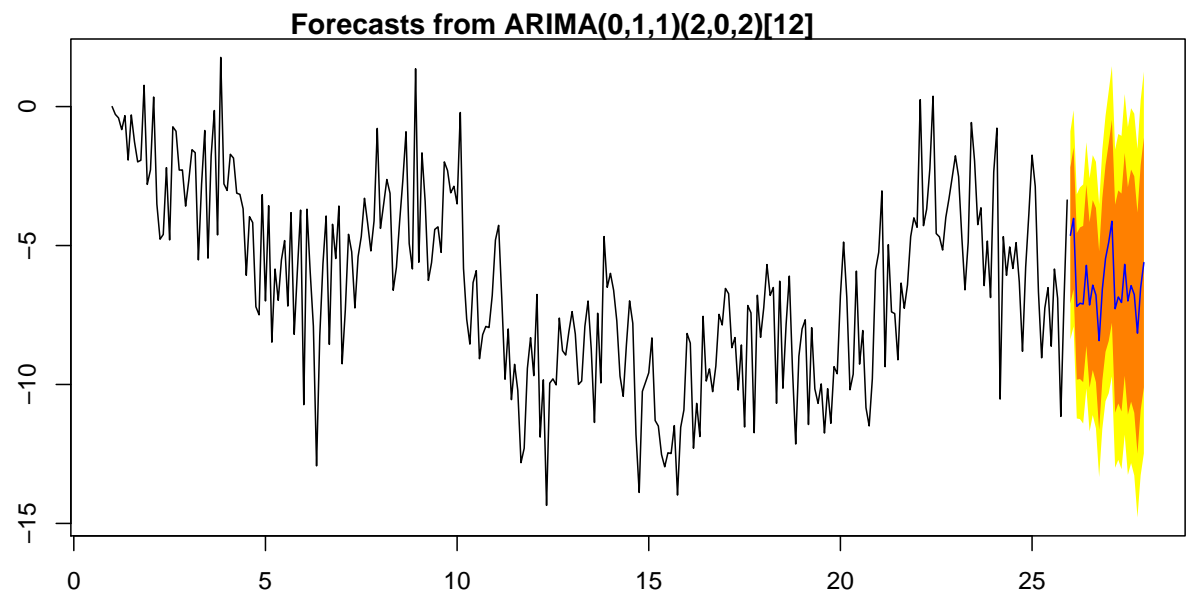
```
Coefficients:
```

	ma1	sar1	sar2	sma1	sma2
	-0.7171	0.7585	0.1972	-0.5636	-0.2213
s.e.	0.0387	0.5001	0.4766	0.4937	0.3901

```
sigma^2 estimated as 3.650: log likelihood = -621.75
AIC = 1255.5   AICc = 1255.78   BIC = 1277.7
```

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

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



Obrázek 20: Predikce na základě odhadnutého modelu  $SARIMA(0, 1, 1) \times (0, 1, 1)_{12}$  :  
 $(1 - B)(1 - B^{12})Y_t = (1 - 0.5B)(1 - 0.7B^{12})\varepsilon_t$

## B. Úkol:

Pro svoje data

- vyzkoušejte na rezidua model ARMA, například s využitím funkce `auto.arima`
- vyzkoušejte na původní data model ARIMA, popřípadě SARIMA, například s využitím funkce `auto.arima`