

M5201 – 6. CVIČENÍ:
Generování náhodných procesů
(bílý šum, IID proces; AR, MA, ARMA procesy),
odhad jejich ACF a PACF

1 Bílý šum a IID proces

Náhodný proces $\{\varepsilon_t, t \in T\}$ je **bílý šum** (značíme $\varepsilon_t \sim WN(0, \sigma^2)$), jestliže ε_t jsou nekorelované náhodné veličiny s nulovou střední hodnotou a konstantním rozptylem, tj. platí

$$E\varepsilon_t = 0, \quad D\varepsilon_t = \sigma^2, \quad C(\varepsilon_t, \varepsilon_s) = 0, \quad s \neq t.$$

Bílý šum, v němž jsou náhodné veličiny ε_t nejen nekorelované, ale také nezávislé, se nazývá **IID proces** a značí se $\varepsilon_t \sim IID(0, \sigma^2)$.

Autokorelační funkce stacionárního náhodného procesu $\{X_t, t \in T\}$ je definována vztahem

$$\varrho(t) = \frac{\gamma(t)}{\gamma(0)},$$

kde $\gamma(t)$ je autokovarianční funkce a $\gamma(0) = D(X_t)$.

Parciální autokorelační funkce stacionárního náhodného procesu $\{X_t, t \in T\} \in L(\omega, \mathcal{A}, P)$ je definována vztahem

$$\begin{aligned} \alpha(1) &= R(X_t, X_{t+1}), \\ \alpha(k) &= R(X_t - \hat{X}_t, X_{t-k} - \hat{X}_{t-k}) \quad \text{pro } |k| > 1, \end{aligned}$$

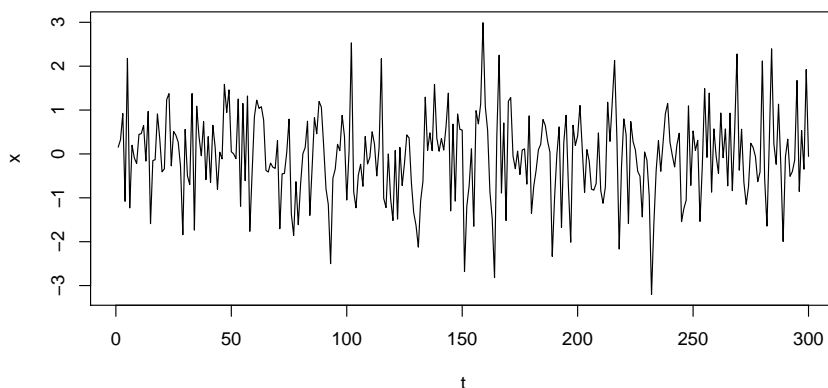
kde \hat{X}_t , resp. \hat{X}_{t-k} jsou nejlepší lineární predikce X_t , resp. X_{t-k} pomocí $X_{t-k+1}, \dots, X_{t-1}$.

PŘÍKLAD 1

Mějme náhodný proces $\{\varepsilon_t, t \in T\}$, kde všechny náhodné veličiny ε_t mají normální rozdělení $N(0, \sigma^2)$ a jsou navzájem nezávislé. Je zřejmé, že takový proces představuje bílý šum (jedná se dokonce o IID proces).

Hodnotu rozptylu zvolme $\sigma^2 = 1$ a vygenerujeme 300 hodnot tohoto procesu. Vygenerovanou řadu vykreslíme.

```
> x <- rnorm(300, mean = 0, sd = 1)
> plot(x, type = "l", xlab = "t")
```

Obrázek 1: Simulovaná data bílého šumu $\varepsilon_t \sim N(0, 1)$

Nyní prozkoumáme autokorelační a parciální autokorelační funkci bílého šumu. Vydeme při tom ze znalosti autokovarianční funkce pro bílý šum

$$\gamma(k) = \begin{cases} \sigma^2 & k = 0 \\ 0 & \text{jinak} \end{cases}.$$

Odtud ihned vidíme, že

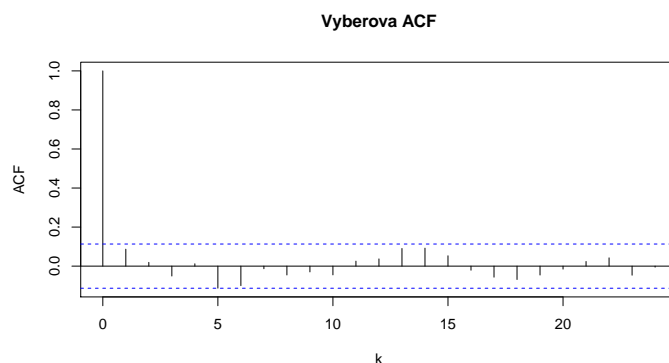
$$\varrho(k) = \begin{cases} 1 & k = 0 \\ 0 & \text{jinak} \end{cases}$$

$$\alpha(k) = 0 \quad k = 1, 2, \dots$$

Podívejme se, jak tomuto faktu odpovídají odhady autokorelační funkce a parciální autokorelační funkce, které získáme na základě vygenerovaných dat.

K výpočtu a vykreslení výběrové autokorelační funkce použijeme v R funkci `acf()` a pro výběrovou parciální autokorelační funkci existuje analogická funkce `pacf()`.

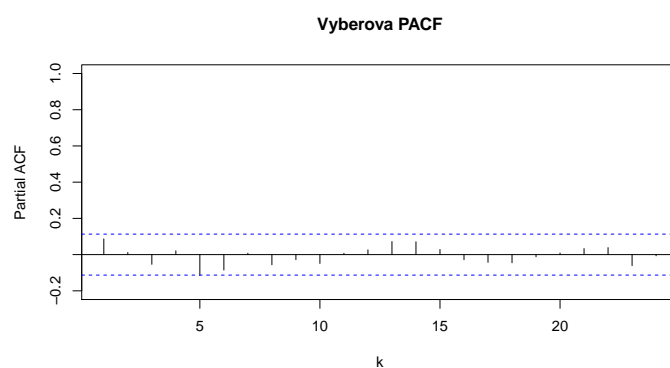
```
> acfWN <- acf(x, main = "Vyberova ACF", xlab = "k")
```

Obrázek 2: Výběrová autokorelační funkce bílého šumu $\varepsilon_t \sim N(0, 1)$

V obrázku odhadnuté autokorelační funkce jsou dvě modré přerušované čáry. Ty představují 95% pás spolehlivosti pro autokorelační funkci bílého šumu. To znamená, že v případě

bílého šumu by hodnoty výběrové autokorelační funkce pro $k > 0$ měly ležet s pravděpodobností 95 % uvnitř tohoto pásu.

```
> pacfWN <- pacf(x, main = "Vyberova PACF", xlab = "k",
  ylim = c(-0.2, 1))
```



Obrázek 3: Výběrová parciální autokorelační funkce bílého šumu $\varepsilon_t \sim N(0, 1)$

Na obrázku opět vidíme, že hodnoty výběrové parciální autokorelační funkce leží uvnitř 95% pásu spolehlivosti zkonstruovaného pro případ bílého šumu, což souhlasí s tím, že naše zkoumaná řada byla simulována jako realizace bílého šumu.

Pro testování nulové hypotézy

$$H_0: \text{časová řada představuje bílý šum}$$

byl odvozen Box-Piercův a Box-Ljungův test. Vyzkoušejme oba tyto testy na simulovaná data bílého šumu.

```
> Box.test(x)
```

Box-Pierce test

```
data: x
X-squared = 2.2612, df = 1, p-value = 0.1327
```

```
> Box.test(x, type = "Ljung-Box")
```

Box-Ljung test

```
data: x
X-squared = 2.2839, df = 1, p-value = 0.1307
```

V obou případech jsme dostali p-hodnoty (p-value) větší než 0.05, a proto testy nezamítají hypotézu, že testovaná data jsou bílým šumem (což, jak víme, skutečně platí).

2 Kauzální AR procesy

Uvažujme **autoregresní model** p -tého řádu. Tento model je definován vztahem:

$$Y_t - \varphi_1 Y_{t-1} - \cdots - \varphi_p Y_{t-p} = \varepsilon_t$$

a ekvivalentně ho můžeme psát takto:

$$Y_t = \varphi_1 Y_{t-1} + \cdots + \varphi_p Y_{t-p} + \varepsilon_t.$$

AR proces $Y_t \sim AR(p)$ se nazývá **kauzální**, jestliže je možné ho zapsat jako proces $MA(\infty)$, tj. $Y_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}$.

Kauzální AR proces je zároveň stacionární. Nutnou a postačující podmínkou kauzality AR procesu je, aby všechny kořeny polynomu $\Phi(z) = 1 - \varphi_1 z - \varphi_2 z^2 - \dots - \varphi_p z^p$ ležely vně jednotkové kružnice.

Pro autokorelační a parciální autokorelační funkci kauzálního $AR(p)$ procesu platí následující:

- $\rho(k)$ klesá pro $k \rightarrow \infty$ exponenciálně k nule
- Je-li $\{Y_t, t \in \mathbb{Z}\}$ centrováný nedegenerovaný kauzální $AR(p)$ proces, pak platí:
 - (1) $\alpha(p) = \varphi_p$
 - (2) $\alpha(p) = 0$ pro $k > p$ (důležitá identifikační vlastnost $AR(p)$ procesu)

PŘÍKLAD 2

Dále budeme pracovat s AR procesem druhého řádu s konkrétními hodnotami $\varphi_1 = 0.9$ a $\varphi_2 = -0.25$, tj. proces zadaný rovnicí

$$Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t.$$

U takto zadaného modelu nevíme, zda se jedná o kauzální nebo nekauzální proces. Abychom to byli schopni zjistit, potřebujeme znát kořeny polynomu $1 - \varphi_1 z - \cdots - \varphi_p z^p$. K tomu použijeme funkci `polyroot()`, která předpokládá, že koeficienty polynomu řádu p jsou zadány v pořadí od nejnižší mocniny k nejvyšší, tj. a_0, a_1, \dots, a_p . Až budeme znát kořeny polynomu $\Phi(z)$, pomocí funkce `Mod()` zjistíme jejich absolutní hodnoty.

```
> ARphi <- c(0.9, -0.25)
> ARroots <- polyroot(c(1, -ARphi))
> Mod(ARroots)
```

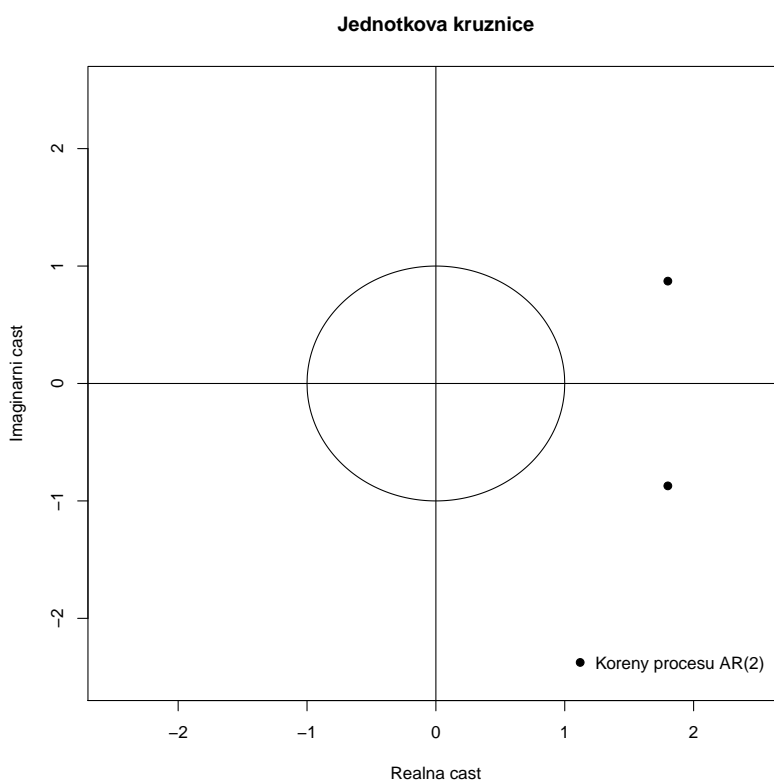
[1] 2 2

Pro větší názornost si ještě vykresleme obrázek, v němž bude znázorněna poloha kořenů polynomu $\Phi(z)$ vůči jednotkové kružnici.

```

> x <- seq(-1, 1, length = 1000)
> y <- sqrt(1 - x^2)
> plot(Re(ARroots), Im(ARroots), type = "p", pch = 19,
      xlab = "Realna cast", ylab = "Imaginarni cast", main = "Jednotkova kruznice",
      ylim = c(-2.5, 2.5), xlim = c(-2.5, 2.5))
> lines(c(x, rev(x)), c(y, -rev(y)))
> abline(h = 0)
> abline(v = 0)
> legend(c(-2.2), legend = "Koreny procesu AR(2)", pch = 19,
      bty = "n")

```



Obrázek 4: Poloha kořenů polynomu $\Phi(z)$ vzhledem k jednotkové kružnici

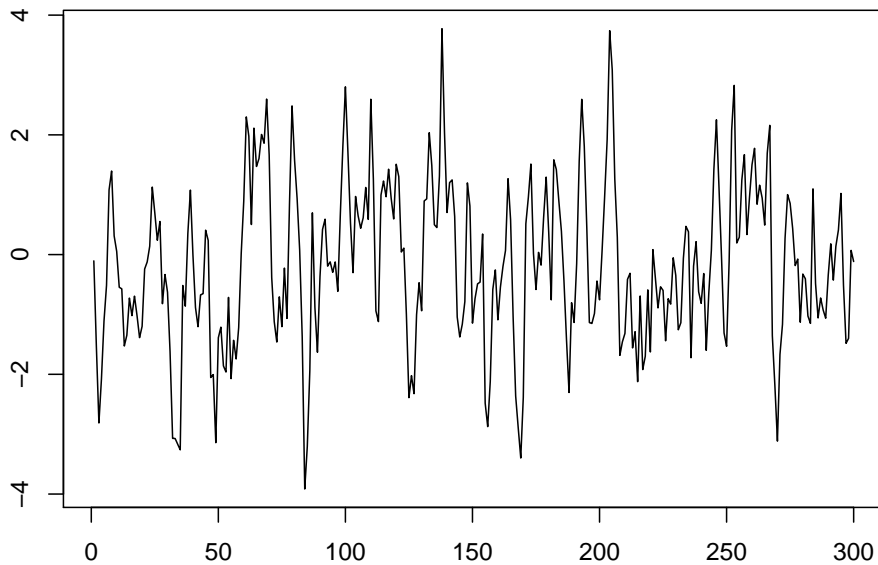
Vidíme, že kořeny leží vně jednotkové kružnice, takže jde o kauzální AR proces.

Nyní pomocí R vygenerujeme 300 hodnot takto zadaného AR procesu. Použijeme k tomu funkci `arima.sim()`, která je určena ke generování ARIMA procesů. Parametry těchto procesů se zadávají do argumentu `model` a musí se jednat o datový objekt typu *seznam* (*list*), který má položky s názvy `ar`, `ma` a `order`. V případě AR procesu stačí zadat seznam s jedinou položkou `ar` obsahující vektor s koeficienty $\varphi_1, \dots, \varphi_p$. Po vygenerování získanou časovou řadu vykreslíme.

```

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

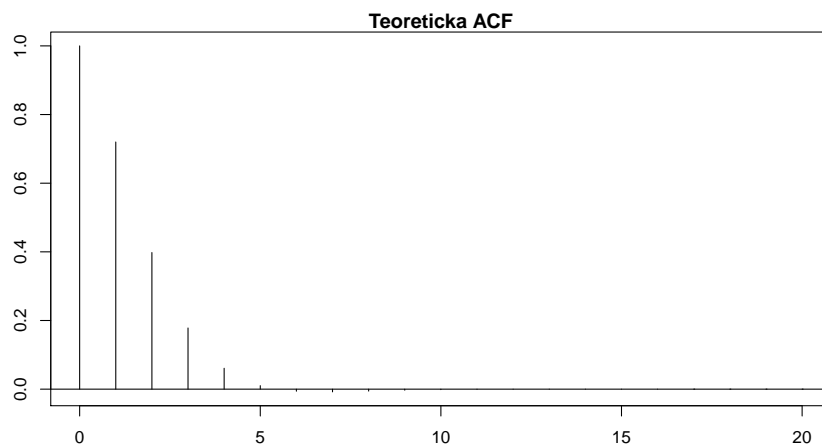
```



Obrázek 5: Simulovaná data AR procesu $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

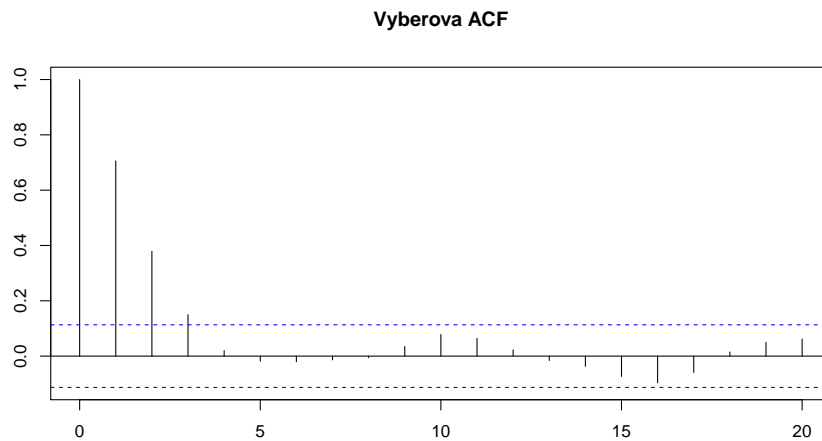
Pomocí funkce `ARMAacf()` můžeme vypočítat teoretickou autokorelační funkci $\rho(k)$ (ACF) a teoretickou parciální autokorelační funkci $\alpha(k)$ (PACF) AR procesu $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$. Výběrové protějšky těchto funkcí vypočtené na základě vygenerovaných hodnot procesu získáme opět pomocí funkcí `acf()`, popř. `pacf()`.

```
> tARacf <- ARMAacf(ar = ARphi, 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 = "Teoreticka ACF")
> abline(h = 0)
```



Obrázek 6: Teoretická ACF pro AR proces $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

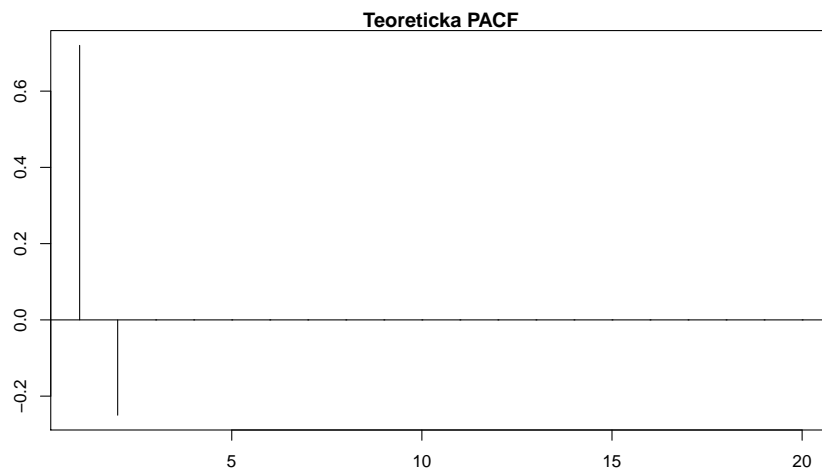
```
> vARacf <- acf(ar.sim, lag.max = 20, plot = FALSE)
> par(mfrow = c(1, 1), mar = c(2, 2, 3, 0) + 0.05)
> plot(vARacf, main = "Vyberova ACF")
```



Obrázek 7: Výběrová ACF pro simulovaná data AR procesu $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

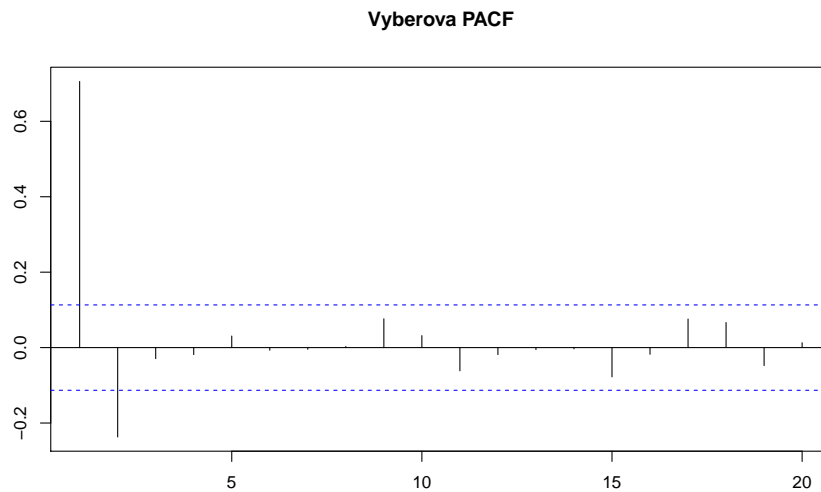
Z grafů je dobře vidět že jak teoretická, tak výběrová autokorelační funkce klesá exponenciálně k nule.

```
> tARpacf <- ARMAacf(ar = ARphi, 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 = "Teoretická PACF")
> abline(h = 0)
```



Obrázek 8: Teoretická PACF pro AR proces $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

```
> 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 = "Vyberova PACF")
```



Obrázek 9: Výběrová PACF pro simulovaná data AR procesu $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

Teoretická parciální autokorelační funkce nabývá nenulových hodnot pouze pro $k = 1$ a $k = 2$ a v souladu s tím je také její výběrový protějšek, kde pro $k > 2$ jsou hodnoty výběrové parciální autokorelační funkce blízké nule a leží uvnitř 95%ního pásu spolehlivosti pro bílý šum.

Pomocí funkce `arima()` můžeme provést odhad parametrů $AR(2)$ modelu. Při použití této funkce je nutné do argumentu `order` zadat řád odhadovaného ARIMA modelu. AR proces v našem příkladu je speciálním případem ARIMA procesu s řádem $c(2, 0, 0)$.

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

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

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

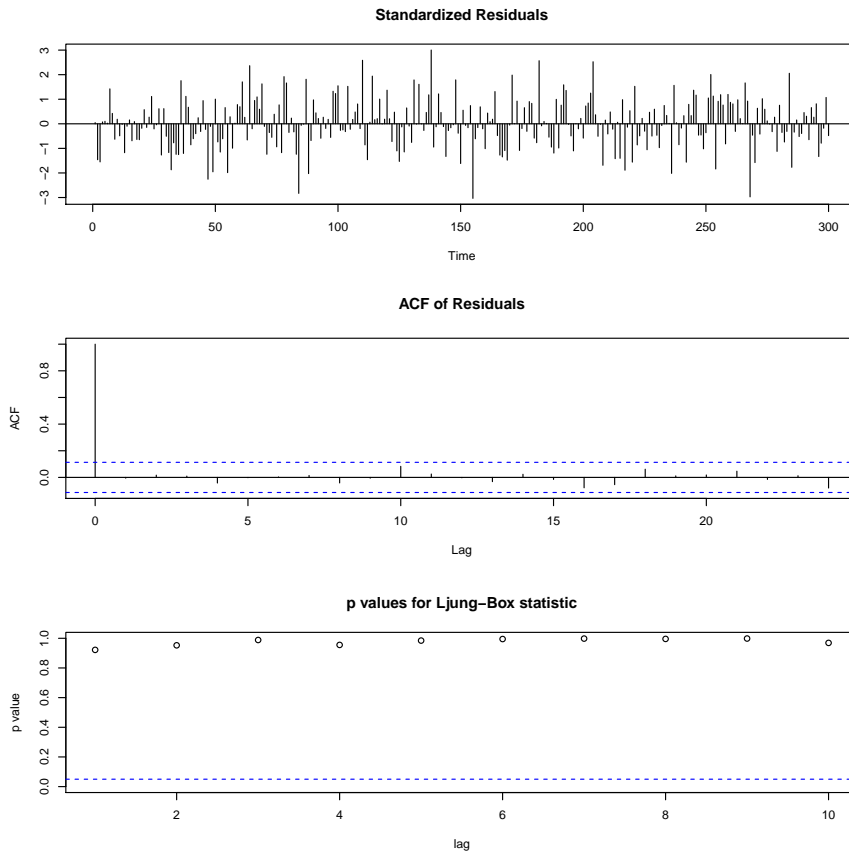
```
Coefficients:
      ar1      ar2  intercept
    0.8714 -0.2373  -0.1692
s.e.  0.0560  0.0561   0.1466
```

```
sigma^2 estimated as 0.8694:  log likelihood = -405.08
AIC = 818.17  AICc = 818.3  BIC = 832.98
```

Jak je vidět z výsledků, odhady se příliš neliší od skutečných hodnot parametrů.

Kvalitu vyrovnání dat pomocí právě odhadnutého modelu můžeme posoudit na základě reziduí. Ve statistické knihovně k tomu máme k dispozici funkci `tsdiag()`, která se aplikuje na odhadnutý AR model.

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

Obrázek 10: Diagnostické grafy pro odhadnutý AR(2) model pro simulovaná data AR procesu $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

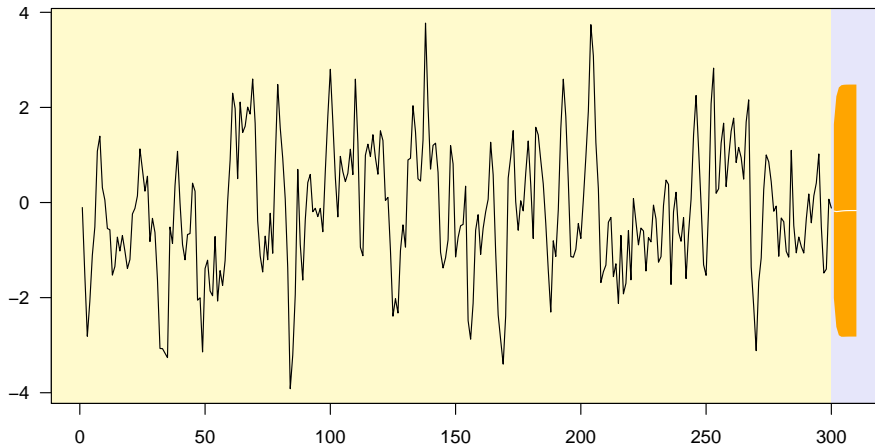
Jestliže byl model zvolen a odhadnut správně, rezidua by měla představovat bílý šum. První graf zobrazuje samotná rezidua. Ta by neměla vykazovat žádnou systematickosti. V druhém grafu je vykreslena výběrová autokorelační funkce reziduí, která by měla mít významně nenulovou hodnotu pouze v bodě $k = 0$. Třetí graf ukazuje p-hodnoty Box-Ljungova testu, zda rezidua (pro různá zpoždění k) představují bílý šum. Je-li model určen správně, měly by všechny hodnoty ležet nad hladinou významnosti testu, která je v grafu znázorněna přerušovanou čarou.

Všechny grafy nasvědčují tomu, že AR proces 2. řádu představuje vhodný model, protože po jeho odstranění z časové řady jsme získali bílý šum. Vzhledem k tomu, že jsme data vygenerovali právě pomocí tohoto AR procesu, je výsledek v souladu s očekáváním.

Nyní, když jsme pro data našli vhodný model, jsme schopni predikovat hodnoty procesu pro několik kroků dopředu. Pro tento účel je v souboru `FunkceM5201.R` vytvořena funkce `PlotPredictARIMA`. Abychom ji mohli používat, musíme tento soubor načíst příkazem `source()`. Pokud by nás zajímal zdrojový kód funkce, stačí jako příkaz zadat do R název funkce `PlotPredictARIMA` a zdrojový kód se vypíše na obrazovku.

Funkci použijeme pro predikci o 10 kroků dopředu. Jako argumenty zadáme zkoumanou časovou řadu (`ar.sim`), odhadnutý model pro tuto řadu (`ar.fit`) a počet predikovaných hodnot (`n.ahead=10`).

```
> fileSkript <- paste(data.library, "FunkceM5201.R", sep = "")
> source(fileSkript)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> PlotPredictARIMA(ar.sim, ar.fit, n.ahead = 10)
```



Obrázek 11: Predikce na základě odhadnutého AR(2) modelu pro simulovaná data AR procesu $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

Když jsme odhadovali parametry AR procesu pomocí funkce `arima`, museli jsme dopředu určit řád odhadovaného modelu. V knihovně `forecast` máme k dispozici funkci `auto.arima()`, kterou také můžeme použít k odhadu modelu pro naše simulovaná data. Tato funkce se od funkce `arima` liší předpokladem, že vstupní data jsou generována obecným ARIMA procesem a tím, že v průběhu výpočtu zároveň zjišťuje vhodný řád modelu a zároveň odhaduje parametry pro tento model. Zadáme-li volbu `trace=TRUE`, na obrazovku se nám vypíše všechny typy ARIMA modelů, které funkce `auto.arima()` brala při výpočtech v úvahu. Vyzkoušejme tedy, jaký model odhadneme pomocí této funkce.

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

```
ARIMA(2,0,2) with non-zero mean : 820.6625
ARIMA(0,0,0) with non-zero mean : 1038.305
ARIMA(1,0,0) with non-zero mean : 834.5473
ARIMA(0,0,1) with non-zero mean : 872.167
ARIMA(1,0,2) with non-zero mean : 821.9867
ARIMA(3,0,2) with non-zero mean : 822.1195
ARIMA(2,0,1) with non-zero mean : 818.8195
ARIMA(2,0,1) with zero mean      : 818.033
ARIMA(1,0,1) with zero mean     : 820.6264
ARIMA(3,0,1) with zero mean     : 819.195
ARIMA(2,0,0) with zero mean     : 817.0522
ARIMA(2,0,0) with non-zero mean : 817.9045
ARIMA(1,0,0) with zero mean     : 833.37
ARIMA(3,0,0) with zero mean     : 817.1965
```

```
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
  0.8750 -0.2343
s.e.  0.0561  0.0561

sigma^2 estimated as 0.8731:  log likelihood = -405.74
AIC = 817.47  AICc = 817.56  BIC = 828.59

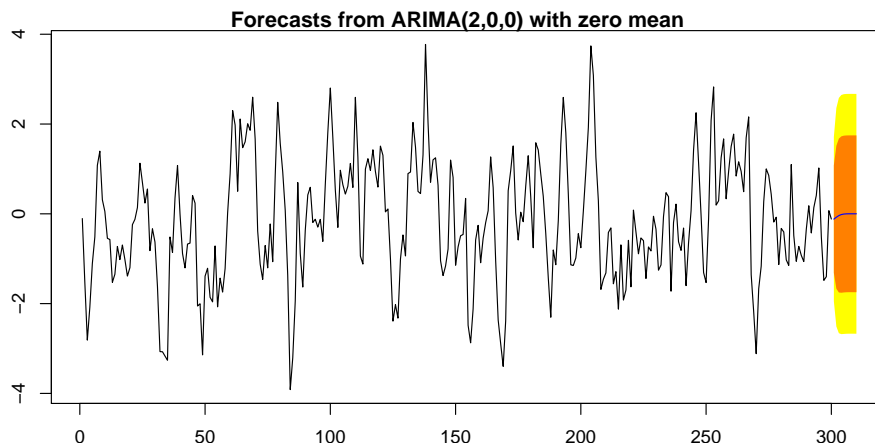
```

V knihovně `forecast` je ještě k dispozici funkce `forecast()`, kterou použijeme při predikci pro právě odhadnutý model.

```

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

```



Obrázek 12: Predikce na základě odhadnutého ARIMA(2,0,0) modelu pro simulovaná data AR procesu $Y_t = 0.9Y_{t-1} - 0.25Y_{t-2} + \varepsilon_t$

3 Invertibilní MA procesy

Uvažujme nyní MA model q -tého řádu:

$$Y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}.$$

MA proces $Y_t \sim MA(q)$ se nazývá **invertibilní**, jestliže je možné ho zapsat jako proces $AR(\infty)$, tj. $\varepsilon_t = \sum_{j=0}^{\infty} \pi_j Y_{t-j}$. $MA(q)$ proces je vždy slabě stacionární.

Nutnou a postačující podmínkou invertibility MA procesu je, aby všechny kořeny polynomu $\Theta(z) = 1 + \theta_1 z + \theta_2 z^2 + \cdots + \theta_q z^q$ ležely vně jednotkové kružnice.

Pro autokorelační a parciální autokorelační funkci invertibilního $MA(q)$ procesu platí následující:

- $\rho(k)$ je nulová pro všechna $k > q$ (důležitá identifikační vlastnost $MA(q)$ procesů)

- Neexistuje takové $k_0 \in \mathbb{N}$, že by pro $k > k_0$ platilo $\alpha(k) = 0$.

PŘÍKLAD 3

V příkladu budeme pracovat s MA procesem druhého řádu s konkrétními hodnotami $\theta_1 = 0.75$ a $\theta_2 = 0.2$, tj. s procesem tvaru

$$Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$$

U takto přímo zadaného modelu nevíme, zda se jedná o invertibilní či neinvertibilní proces. Abychom byli schopni rozhodnout o invertibilitě zadaného procesu, budou nás zajímat kořeny polynomu $1 + \theta_1 z + \dots + \theta_q z^q$. K tomu opět použijeme funkci `polyroot()`, která předpokládá koeficienty polynomu v pořadí a_0, a_1, \dots, a_q . Jestliže kořeny leží vně jednotkové kružnice, MA proces je invertibilní.

```
> Mtheta <- c(0.75, 0.2)
> print(MAroots <- polyroot(c(1, Mtheta)))
```

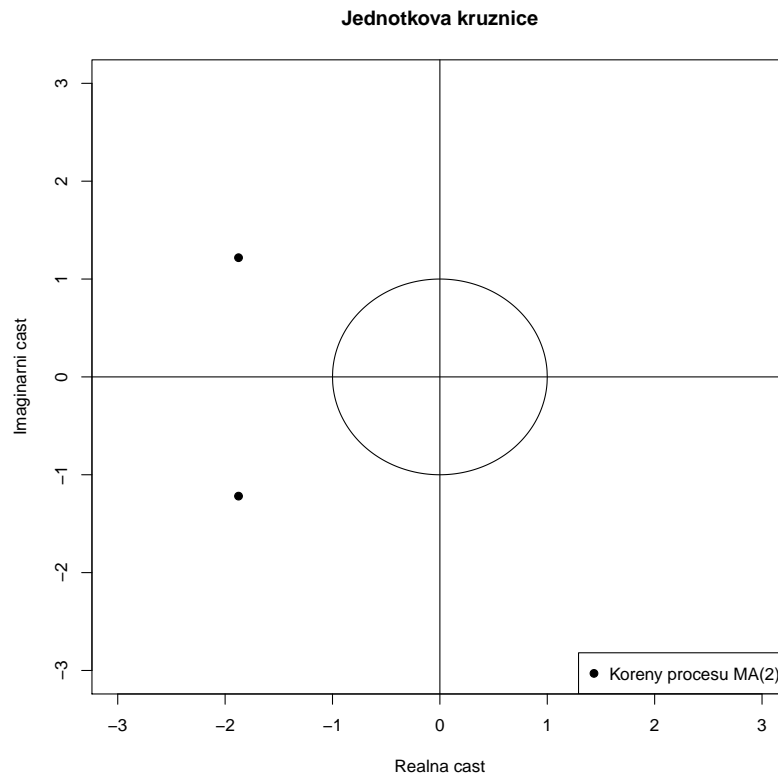
```
[1] -1.875+1.218349i -1.875-1.218349i
```

```
> Mod(MAroots)
```

```
[1] 2.236068 2.236068
```

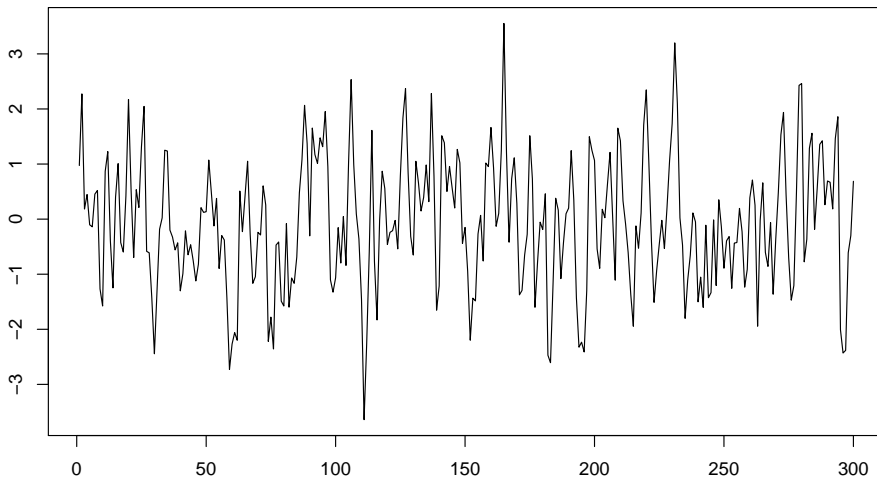
Kořeny opět znázorníme jako body v komplexní rovině, abychom viděli jejich polohu vůči jednotkové kružnici.

```
> x <- seq(-1, 1, length = 1000)
> y <- sqrt(1 - x^2)
> plot(Re(MAroots), Im(MAroots), type = "p", pch = 19,
      xlab = "Realna cast", ylab = "Imaginarni cast", main = "Jednotkova kruznice",
      ylim = c(-3, 3), xlim = c(-3, 3))
> lines(c(x, rev(x)), c(y, -rev(y)))
> abline(h = 0)
> abline(v = 0)
> legend("bottomright", legend = "Koreny procesu MA(2)",
      pch = 19)
```

Obrázek 13: Poloha kořenů polynomu $\Theta(z)$ vzhledem k jednotkové kružnici

Vidíme, že kořeny leží vně jednotkové kružnice, takže jde o invertibilní MA proces. Pomocí funkce `arima.sim()` provedeme simulaci pro 300 hodnot. Jak víme, funkce `arima.sim()` simuluje data obecně pro zadaný ARIMA model. V našem případě, kdy generujeme časovou řadu MA procesu 2. řádu (což je speciální případ ARIMA procesu), zadáme parametry modelu do argumentu `model` jako seznam s jedinou složkou `ma = Mtheta`. Vygenerovaná data vykreslíme.

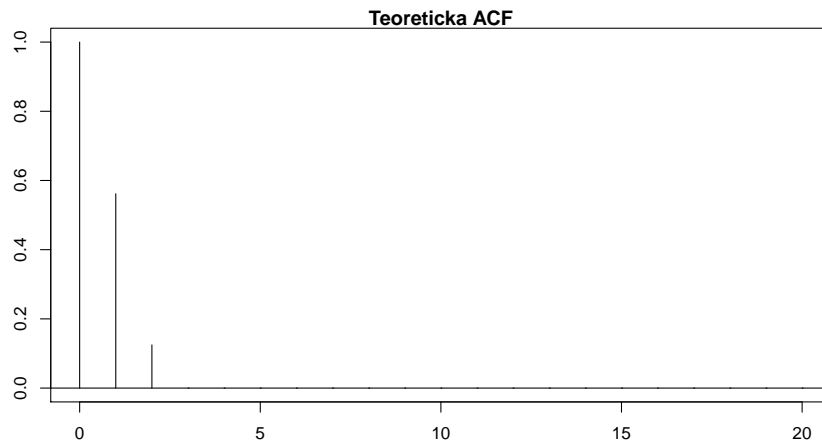
```
> ma.sim <- arima.sim(model = list(ma = Mtheta), n = 300)
> par(mfrow = c(1, 1), mar = c(2, 2, 1, 0) + 0.05)
> plot(ma.sim)
```



Obrázek 14: Simulovaná data MA procesu $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$

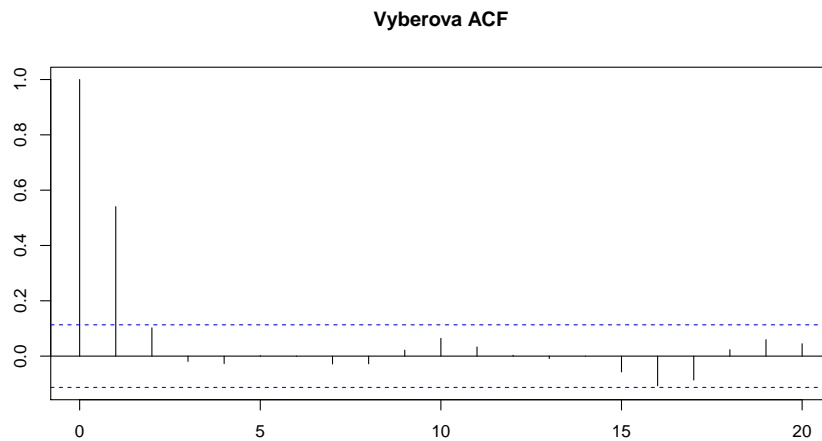
Pomocí funkce `ARMAacf()` můžeme vypočítat teoretickou autokorelační funkci $\rho(k)$ (ACF) a teoretickou parciální autokorelační funkci $\alpha(k)$ (PACF) MA procesu $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$. Výběrové protějšky těchto funkcí získáme pomocí funkcí `acf()`, resp. `pacf()`.

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



Obrázek 15: Teoretická ACF pro MA proces $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$

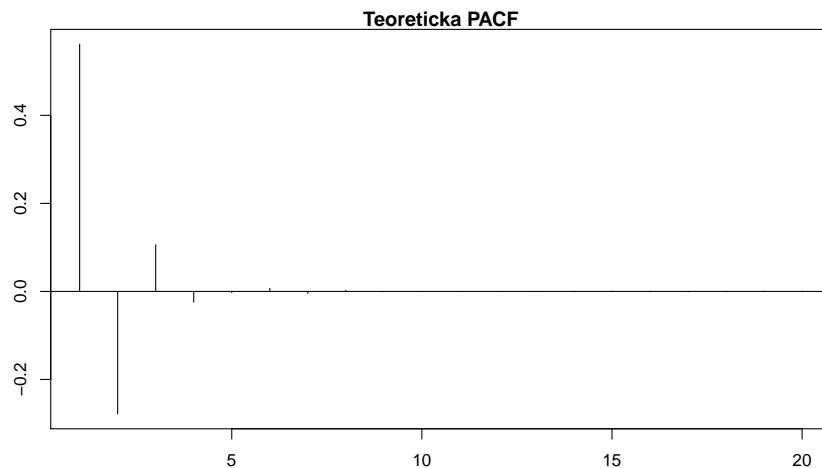
```
> vMAacf <- acf(ma.sim, lag.max = 20, plot = FALSE)
> par(mfrow = c(1, 1), mar = c(2, 2, 3, 0) + 0.05)
> plot(vMAacf, main = "Vyberova ACF")
```



Obrázek 16: Výběrová ACF pro simulovaná data MA procesu $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$

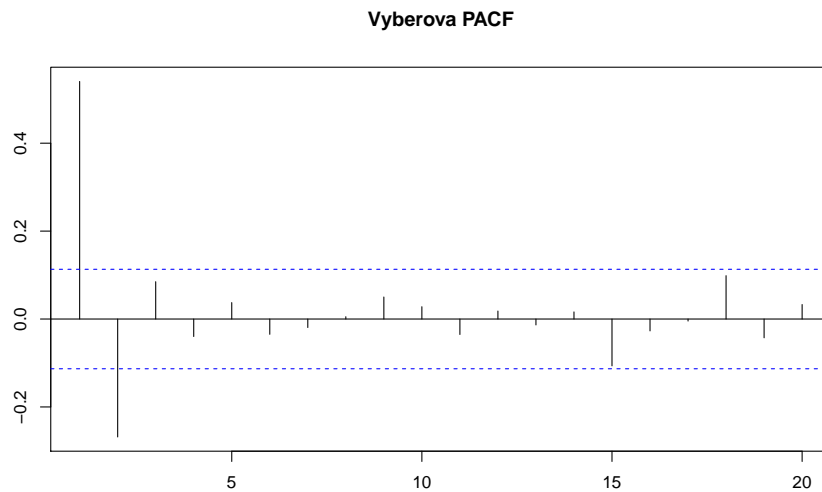
Vidíme, že teoretická autokorelační funkce je nenulová pouze pro $k = 0, 1, 2$. Tomu odpovídá i výběrová autokorelační funkce, která je významně nenulová pouze v těchto bodech a všude jinde její hodnoty leží uvnitř 95%ního pásu spolehlivosti pro bílý šum.

```
> tMApacf <- ARMAacf(ma = Mtheta, 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 = "Teoreticka PACF")
> abline(h = 0)
```



Obrázek 17: Teoretická PACF pro MA proces $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$

```
> 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 = "Vyberova PACF")
```



Obrázek 18: Výběrová PACF pro simulovaná data MA procesu $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$

Vidíme, že teoretická parciální autokorelační funkce klesá v absolutní hodnotě exponenciálně k nule a tomu odpovídá i průběh výběrové parciální autokorelační funkce.

Pomocí funkce `arima()` provedeme odhad parametrů $MA(2)$ modelu na základě simulovaných dat.

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

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

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

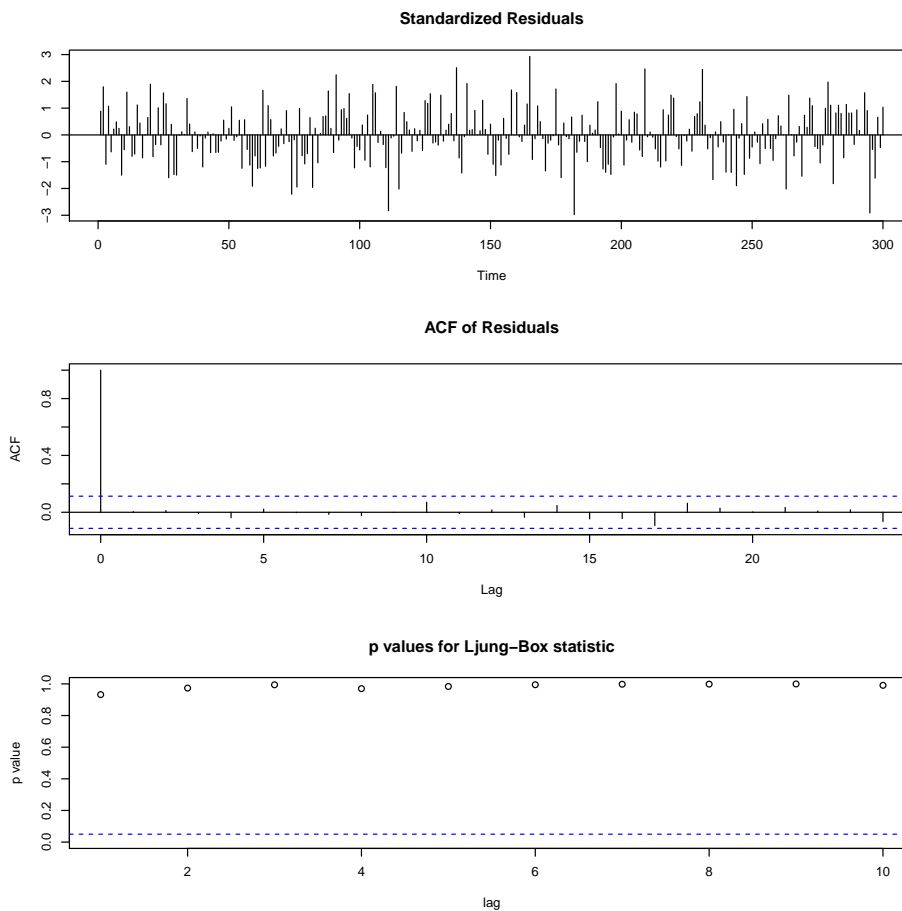
Coefficients:
      ma1      ma2  intercept
 0.7108  0.1472  -0.0703
s.e.  0.0561  0.0575   0.1019

sigma^2 estimated as 0.9057:  log likelihood = -411.09
AIC = 830.19  AICc = 830.32  BIC = 845
```

I v tomto případě jsou odhady blízké skutečným hodnotám parametrů.

Vhodnost použitého modelu posoudíme podle reziduí. Diagnostické grafy získáme opět díky funkci `tsdiag()`, kterou použijeme na odhadnutý MA model.

```
> tsdiag(ma.fit)
```

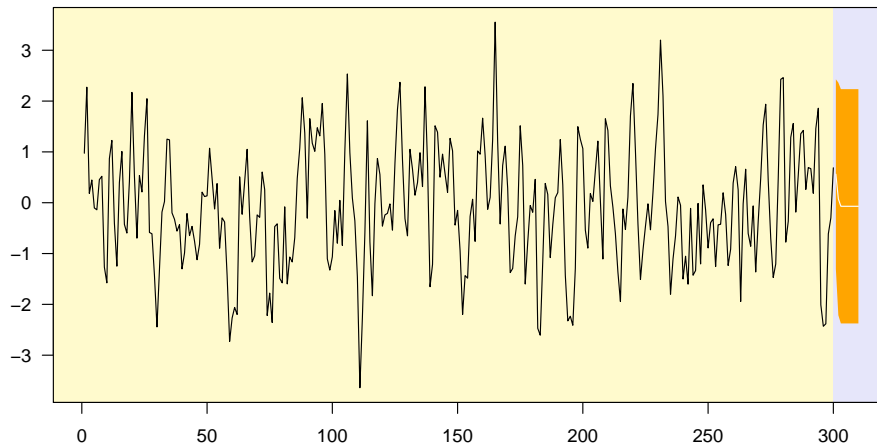



Obrázek 19: Diagnostické grafy pro odhadnutý MA(2) model pro simulovaná data MA procesu $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$

Žádný z diagnostických grafů nenaznačuje, že by náš model byl nevhodný. Residua nevykazují žádnou systematickosti, výběrová autokorelační funkce kromě $k = 0$ leží v 95%ním pásu spolehlivosti pro bílý šum a ani Ljung-Boxův test nezamítá hypotézu o tom, že residua představují bílý šum.

Odhadnutý model použijeme k predikci o 10 kroků dopředu pomocí funkce `PlotPredictARIMA`.

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



Obrázek 20: Predikce na základě odhadnutého MA(2) modelu pro simulovaná data MA procesu $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.2\varepsilon_{t-2}$

Ještě provedeme obecnější odhad modelu, kdy použijeme funkci `auto.arima()`. Již víme, že tato funkce odhaduje zároveň řád ARIMA modelu i jeho parametry.

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

```
ARIMA(2,0,2) with non-zero mean : 829.2493
ARIMA(0,0,0) with non-zero mean : 955.3027
ARIMA(1,0,0) with non-zero mean : 853.376
ARIMA(0,0,1) with non-zero mean : 834.5677
ARIMA(1,0,2) with non-zero mean : 834.5871
ARIMA(3,0,2) with non-zero mean : 831.7525
ARIMA(2,0,1) with non-zero mean : 827.8052
ARIMA(2,0,1) with zero mean      : 826.4774
ARIMA(1,0,1) with zero mean     : 831.0727
ARIMA(3,0,1) with zero mean     : 828.398
ARIMA(2,0,0) with zero mean     : 829.5102
ARIMA(2,0,2) with zero mean     : 827.8355
ARIMA(1,0,0) with zero mean     : 851.7538
ARIMA(3,0,2) with zero mean     : 830.3901
```

```
Best model: ARIMA(2,0,1) with zero mean
```

```
> print(ma.fitF)
```

```
Series: ma.sim
```

```
ARIMA(2,0,1) with zero mean
```

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

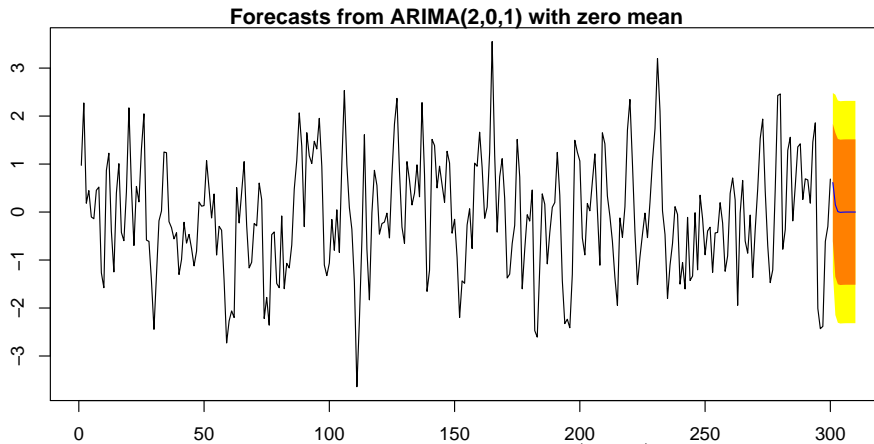
```
Coefficients:
```

```
      ar1      ar2      ma1
      0.3151 -0.0629  0.4022
s.e.  0.2033  0.1343  0.1993
```

```
sigma^2 estimated as 0.9063: log likelihood = -411.2
AIC = 830.39  AICc = 830.53  BIC = 845.21
```

Druhý odhadnutý model použijeme při volání funkce `forecast()`, pomocí níž budeme chtít zjistit 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 21: Predikce na základě odhadnutého $ARIMA(2,0,0)$ modelu pro simulovaná data MA procesu $Y_t = \varepsilon + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

4 Kauzální a invertibilní ARMA procesy

Uvažujme proces $ARMA(p,q)$. Tento model je definován vztahem:

$$Y_t - \varphi_1 Y_{t-1} - \dots - \varphi_p Y_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}.$$

Kauzalita a invertibilita jsou definovány obdobně jako v případě AR a MA procesu:

- ARMA proces $Y_t \sim ARMA(p, q)$ se nazývá **kauzální**, jestliže je možné ho zapsat jako proces $MA(\infty)$, tj. $Y_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}$.
- ARMA proces $Y_t \sim ARMA(p, q)$ se nazývá **invertibilní**, jestliže je možné ho zapsat jako proces $AR(\infty)$, tj. $\varepsilon_t = \sum_{j=0}^{\infty} \pi_j Y_{t-j}$.

Analogicky platí, že nutnou a postačující podmínkou kauzality $ARMA(p,q)$ procesu je, aby všechny kořeny polynomu $\Phi(z) = 1 - \varphi_1 z - \varphi_2 z^2 - \dots - \varphi_p z^p$ ležely vně jednotkové kružnice.

Nutnou a postačující podmínkou invertibility $ARMA(p,q)$ procesu je, aby všechny kořeny polynomu $\Theta(z) = 1 + \theta_1 z + \theta_2 z^2 + \dots + \theta_q z^q$ ležely vně jednotkové kružnice.

Pro autokorelační a parciální autokorelační funkci kauzálního a invertibilního $ARMA(p, q)$ procesu platí následující:

- Neexistuje takové k_0 , že by pro všechna $k > k_0$ platilo $\rho(k) = 0$.
Od $k = q - p$ klesá autokorelační funkce exponenciálně k nule.
- Neexistuje takové k_0 , že by pro všechna $k > k_0$ platilo $\alpha(k) = 0$.
Od $k = p - q$ klesá parciální autokorelační funkce exponenciálně k nule.

PŘÍKLAD 4

Dále budeme pracovat s ARMA(2,2) procesem s konkrétními hodnotami $\varphi_1 = 0.9$, $\varphi_2 = -0.5$, $\theta_1 = 0.75$ a $\theta_2 = 0.25$, tj. s procesem tvaru

$$Y_t = 0.9Y_{t-1} - 0.5Y_{t-2} + \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}.$$

U takto zadaného modelu opět nevíme, zda se jedná o kauzální a invertibilní proces, proto budeme zjišťovat, zda kořeny polynomů $1 - \varphi_1z - \dots - \varphi_pz^p$ a $1 + \theta_1z + \dots + \theta_qz^q$ leží vně jednotkového kruhu. K tomu opět použijeme funkci `polyroot()`, kde zadáváme koeficienty v pořadí od nejnižší po nejvyšší mocninu. Až zjistíme kořeny polynomu, pomocí funkce `Mod()` určíme jejich absolutní hodnoty.

Nejprve najdeme kořeny polynomu $1 - 0.9\varphi_1z + 0.5\varphi_2z^2$ a zjistíme modulus kořenů.

```
> ARphi <- c(0.9, -0.5)
> ARroots <- polyroot(c(1, -ARphi))
> Mod(ARroots)
```

```
[1] 1.414214 1.414214
```

Vidíme, že kořeny leží vně jednotkové kružnice, takže jde o kauzální ARMA proces.

Ted' najdeme kořeny polynomu $1 + 0.75z + 0.25z^2$ a zjistíme jejich absolutní hodnoty.

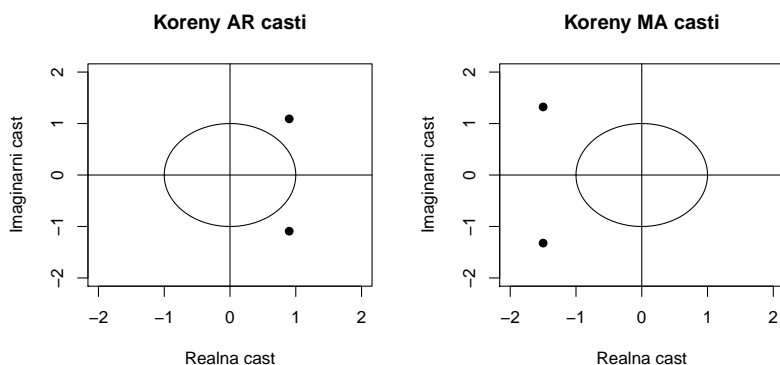
```
> Mtheta <- c(0.75, 0.25)
> MArroots <- polyroot(c(1, Mtheta))
> Mod(MArroots)
```

```
[1] 2 2
```

Vidíme, že kořeny leží vně jednotkové kružnice, takže jde o invertibilní ARMA proces.

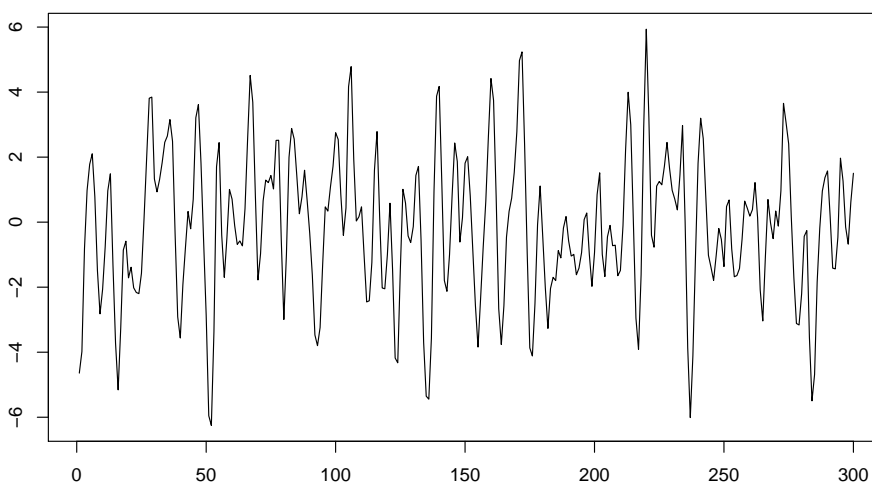
Ještě vykresleme polohu kořenů pro oba polynomy vzhledem k jednotkové kružnici v komplexní rovině.

```
> par(mfrow = c(1, 2))
> plot(Re(ARroots), Im(ARroots), type = "p", pch = 19,
      xlab = "Realna cast", ylab = "Imaginarni cast", main = "Koreny AR casti",
      ylim = c(-2, 2), xlim = c(-2, 2))
> lines(c(x, rev(x)), c(y, -rev(y)))
> abline(h = 0)
> abline(v = 0)
> plot(Re(MArroots), Im(MArroots), type = "p", pch = 19,
      xlab = "Realna cast", ylab = "Imaginarni cast", main = "Koreny MA casti",
      ylim = c(-2, 2), xlim = c(-2, 2))
> lines(c(x, rev(x)), c(y, -rev(y)))
> abline(h = 0)
> abline(v = 0)
```

Obrázek 22: Poloha kořenů polynomů $\Phi(z)$ a $\Theta(z)$ vzhledem k jednotkové kružnici

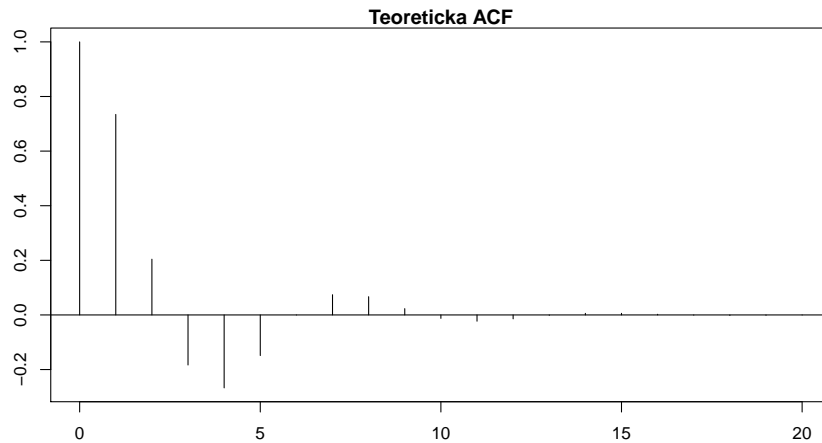
Pomocí funkce `arma.sim()` provedeme simulaci 300 hodnot procesu. Jako argument `model` zadáme seznam se dvěma položkami `ar` (parametry popisující AR část ARMA procesu) a `ma` (parametry popisující MA část ARMA procesu).

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

Obrázek 23: Simulovaná data ARMA procesu $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

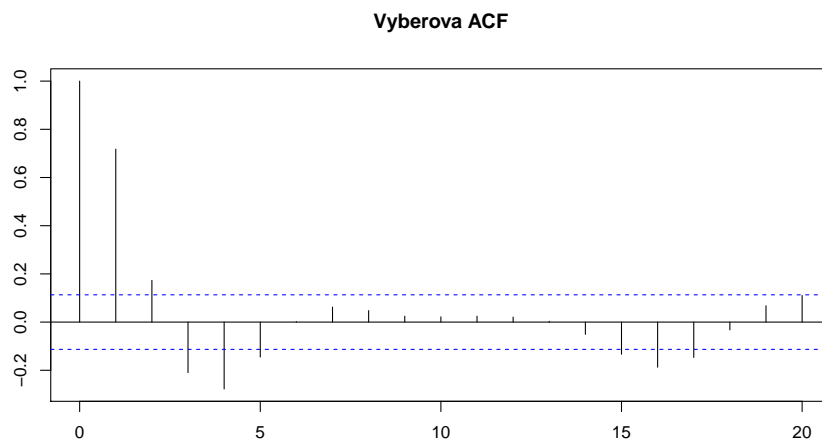
Pomocí funkce `ARMAacf()` vypočítáme teoretickou autokorelační funkci $\rho(k)$ (ACF) a teoretickou parciální autokorelační funkci $\alpha(k)$ (PACF). Výběrové protějšky těchto funkcí získáme opět pomocí funkcí `acf()`, popř. `pacf()`.

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



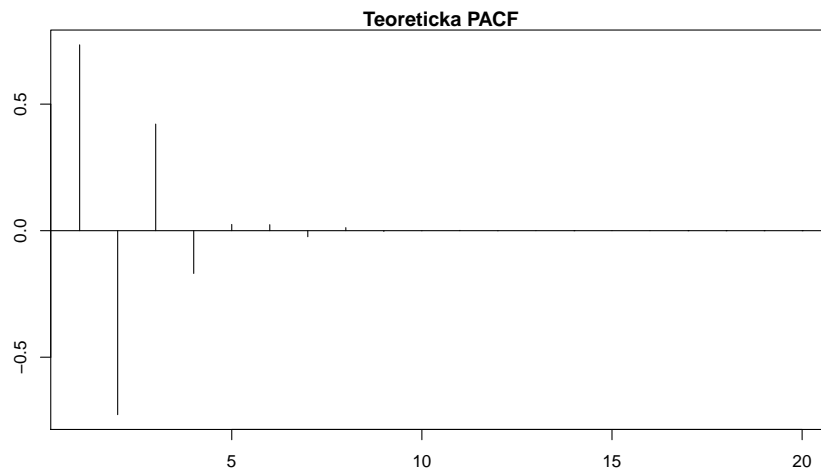
Obrázek 24: Teoretická ACF pro ARMA proces $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

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



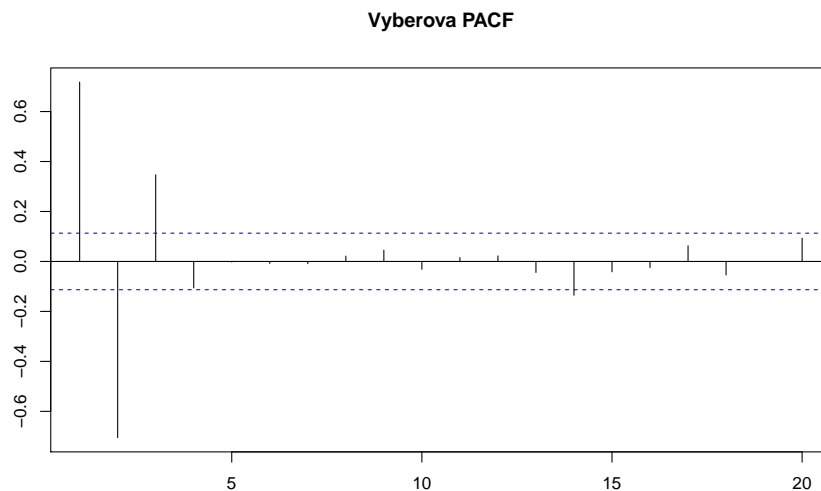
Obrázek 25: Výběrová ACF pro simulovaná data ARMA procesu $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

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



Obrázek 26: Teoretická PACF pro ARMA proces $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

```
> vARMAPacf <- pacf(arma.sim, lag.max = 20, plot = FALSE)
> par(mfrow = c(1, 1), mar = c(2, 2, 3, 0) + 0.05)
> plot(vARMAPacf, main = "Vyberova PACF")
```



Obrázek 27: Výběrová PACF pro simulovaná data ARMA procesu $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

Pomocí funkce `arima()` provedeme odhad parametrů $ARMA(2, 2)$ modelu. Do argumentu `order` musíme zadat řád odhadovaného ARIMA modelu. ARMA proces v našem příkladu je speciálním případem ARIMA procesu s řádem $c(2, 0, 2)$.

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

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

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

```
Coefficients:
```

```

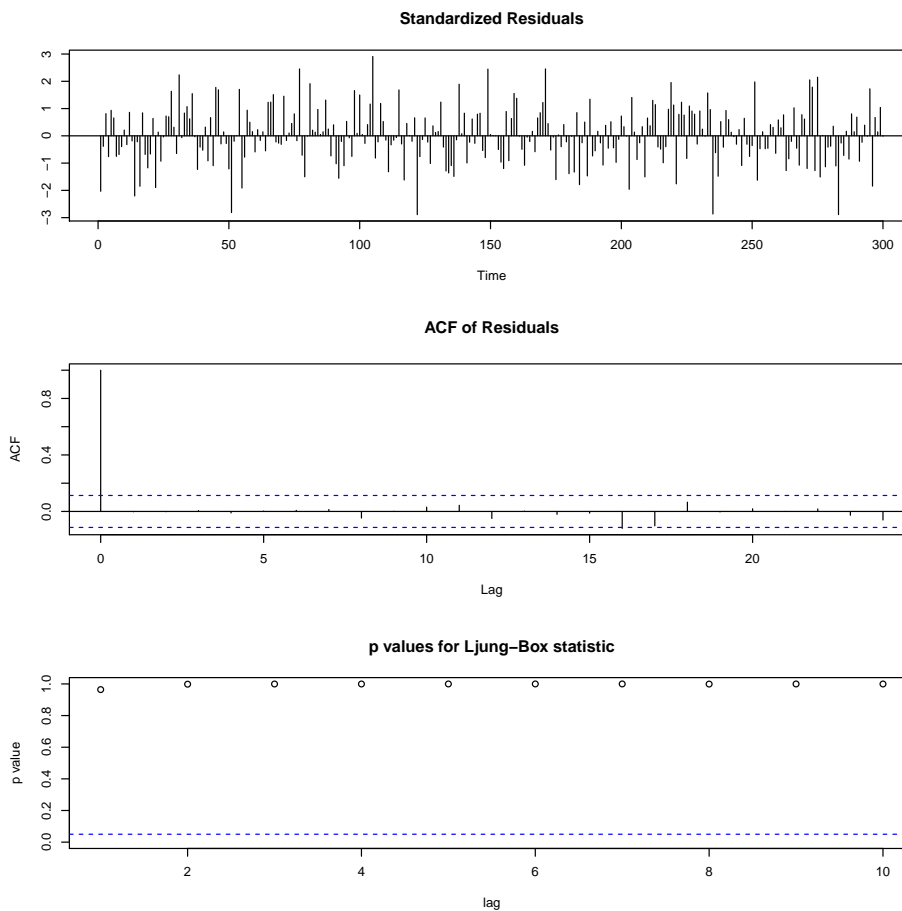
      ar1      ar2      ma1      ma2  intercept
0.9072 -0.5185  0.6644  0.2077   -0.1461
s.e.  0.0944   0.0723  0.1045  0.0923    0.1710

sigma^2 estimated as 0.9377:  log likelihood = -417.41
AIC = 846.82   AICc = 847.11   BIC = 869.04

```

Kvalitu vyrovnání dat pomocí právě odhadnutého modelu posoudíme na základě reziduí s využitím funkce `tsdiag()`, kterou aplikujeme na odhadnutý ARMA(2,2) model.

```
> tsdiag(arma.fit)
```



Obrázek 28: Diagnostické grafy pro odhadnutý ARMA(2,2) model pro simulovaná data ARMA procesu $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

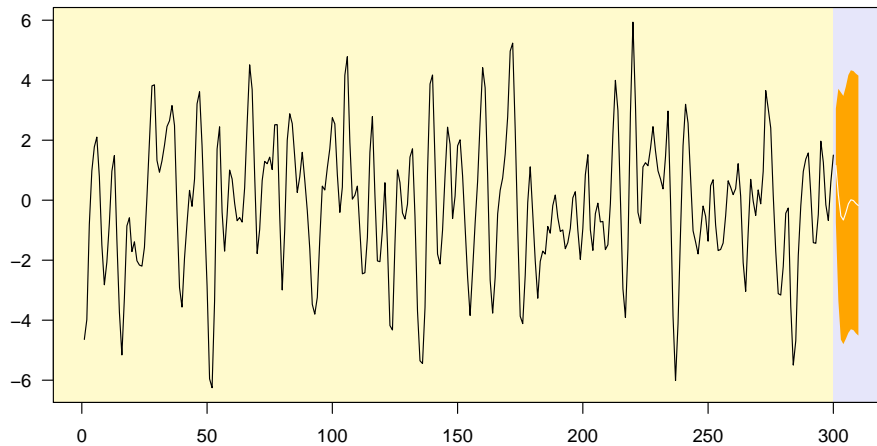
Z grafů opět vidíme, že po odstranění systematické složky z časové řady dostaneme pouze bílý šum.

Na základě odhadnutého modelu provedeme predikci pro 10 budoucích hodnot. K tomu opět použijeme funkci `PlotPredictARIMA`. Jako argumenty zadáme zkoumanou časovou řadu (`arma.sim`), odhadnutý model pro tuto řadu (`arma.fit`) a počet predikovaných hodnot (`n.ahead=10`).

```

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

```

Obrázek 29: Predikce na základě odhadnutého ARMA(2,2) modelu pro simulovaná data ARMA procesu $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

Ještě se podívejme, jak odhadne řád modelu a parametry procesu funkce `auto.arima`.

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

```
ARIMA(2,0,2) with non-zero mean : 841.6855
ARIMA(0,0,0) with non-zero mean : 1332.781
ARIMA(1,0,0) with non-zero mean : 1109.331
ARIMA(0,0,1) with non-zero mean : 1045.133
ARIMA(1,0,2) with non-zero mean : 883.1501
ARIMA(3,0,2) with non-zero mean : 843.9904
ARIMA(2,0,1) with non-zero mean : 845.0074
ARIMA(2,0,3) with non-zero mean : 843.5533
ARIMA(1,0,1) with non-zero mean : 929.7609
ARIMA(3,0,3) with non-zero mean : 845.8741
ARIMA(2,0,2) with zero mean : 840.1418
ARIMA(1,0,2) with zero mean : 881.8383
ARIMA(3,0,2) with zero mean : 842.563
ARIMA(2,0,1) with zero mean : 843.601
ARIMA(2,0,3) with zero mean : 841.9994
ARIMA(1,0,1) with zero mean : 928.0157
ARIMA(3,0,3) with zero mean : 844.4304
```

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

```
> print(arma.fitF)
```

```
Series: arma.sim
```

```
ARIMA(2,0,2) with zero mean
```

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

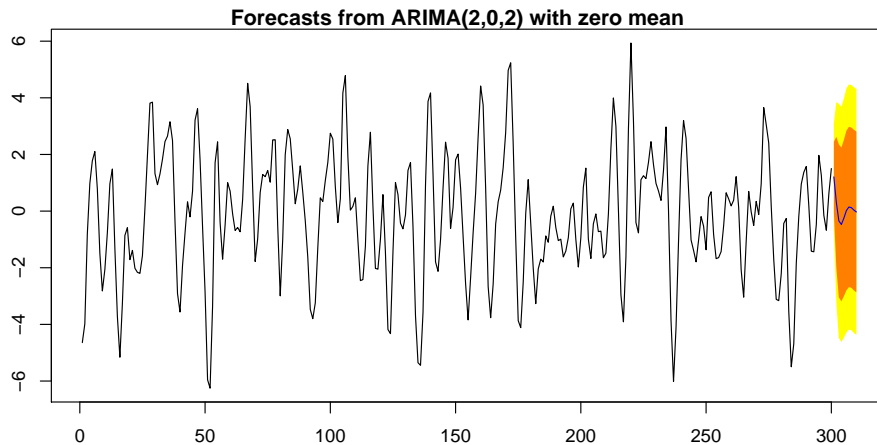
```
Coefficients:
```

```
      ar1      ar2      ma1      ma2
      0.9057 -0.5157  0.6681  0.2104
s.e.  0.0945  0.0726  0.1044  0.0921
```

```
sigma^2 estimated as 0.94: log likelihood = -417.77
AIC = 845.55  AICc = 845.75  BIC = 864.06
```

Na úplný závěr provedeme predikci pro 10 budoucích hodnot na základě posledního modelu, přičemž použijeme funkci `forecast()`.

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



Obrázek 30: Predikce na základě odhadnutého ARIMA(2,0,3) modelu pro simulovaná data ARMA procesu $Y_t - 0.9Y_{t-1} + 0.5Y_{t-2} = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.25\varepsilon_{t-2}$

5 Úkol:

1. Rozhodněte o kauzalitě, resp. invertibilitě následujících procesů:

(a) $Y_t = 0.75Y_{t-1} + 0.5Y_{t-2} + \varepsilon_t$

(b) $Y_t = \varepsilon_t + 0.75\varepsilon_{t-1} + 0.75\varepsilon_{t-2}$

(c) $Y_t = 0.8Y_{t-1} - 0.2Y_{t-2} + \varepsilon_t + 0.6\varepsilon_{t-1} - 0.2\varepsilon_{t-2} + 0.1\varepsilon_{t-3}$

2. Pro procesy z 1. úkolu, u nichž jste ověřili, že jsou kauzální (u AR procesu), invertibilní (u MA procesu) nebo zároveň kauzální i invertibilní (u ARMA procesu) vygenerujte 300 hodnot časové řady, vykreslete autokorelační a parciální autokorelační funkci a odhadněte parametry modelu např. s využitím funkce `arma`.