Ćwiczenia 7

2023-12-11

Zadanie 1

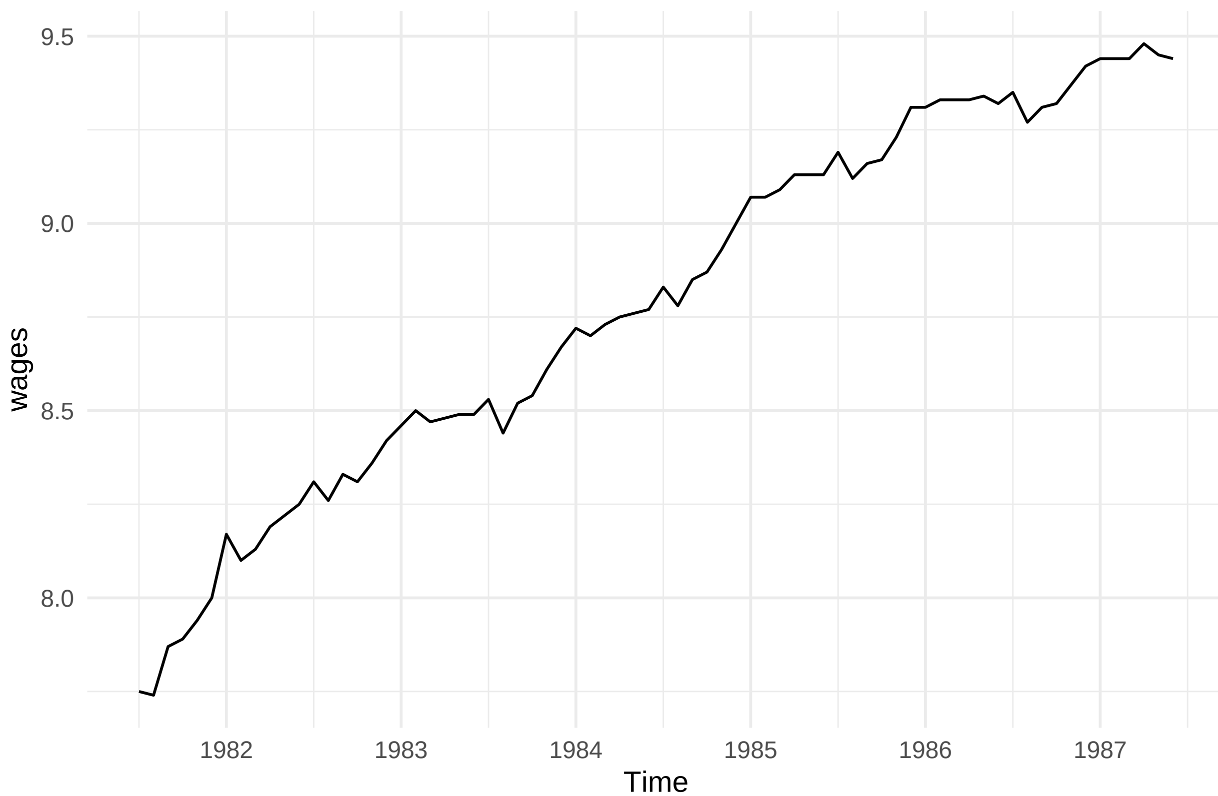
```
print(c(
  "auto"  = difftime(as.Date("2000-12-31"), as.Date("1901-01-01")) |> as.numeric(),
  "secs"  = difftime(as.Date("2000-12-31"), as.Date("1901-01-01"), units = "secs") |> as.numeric(),
  "mins"  = difftime(as.Date("2000-12-31"), as.Date("1901-01-01"), units = "mins") |> as.numeric(),
  "hours" = difftime(as.Date("2000-12-31"), as.Date("1901-01-01"), units = "hours") |> as.numeric(),
  "days"  = difftime(as.Date("2000-12-31"), as.Date("1901-01-01"), units = "days") |> as.numeric(),
  "weeks" = difftime(as.Date("2000-12-31"), as.Date("1901-01-01"), units = "weeks") |> as.numeric()
))
```

```
##         auto         secs         mins        hours         days        weeks
## 3.652400e+04 3.155674e+09 5.259456e+07 8.765760e+05 3.652400e+04 5.217714e+03
```

Zadanie 2

```
data(wages, package = "TSA")
```

```
autoplot(wages) +
  theme_minimal()
```



```r
df <- data.frame(
  y = wages |> as.numeric(),
  time = time(wages) |> as.numeric()
)
model_lin <- lm(y ~ time, data = df)
model_quad <- lm(y ~ time + I(time ^ 2), data = df)
print(summary(model_lin))
```

```
##
## Call:
## lm(formula = y ~ time, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23828 -0.04981  0.01942  0.05845  0.13136
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.490e+02  1.115e+01  -49.24   <2e-16 ***
## time         2.811e-01  5.618e-03   50.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08257 on 70 degrees of freedom
## Multiple R-squared:  0.9728, Adjusted R-squared:  0.9724
## F-statistic:  2503 on 1 and 70 DF,  p-value: < 2.2e-16
```

```r
print(summary(model_quad))
```

```
##
## Call:
## lm(formula = y ~ time + I(time^2), data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.148318 -0.041440  0.001563  0.050089  0.139839
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.495e+04  1.019e+04  -8.336 4.87e-12 ***
## time         8.534e+01  1.027e+01   8.309 5.44e-12 ***
## I(time^2)   -2.143e-02  2.588e-03  -8.282 6.10e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05889 on 69 degrees of freedom
## Multiple R-squared:  0.9864, Adjusted R-squared:  0.986
## F-statistic:  2494 on 2 and 69 DF,  p-value: < 2.2e-16
```

```r
print(AIC(model_lin, model_quad))
```

```
##            df       AIC
## model_lin   3 -150.8585
## model_quad  4 -198.5489
```

```r
print(BIC(model_lin, model_quad))
```

```
##            df       BIC
## model_lin   3 -144.0285
## model_quad  4 -189.4423
```

```r
print(shapiro.test(resid(model_lin)))
```

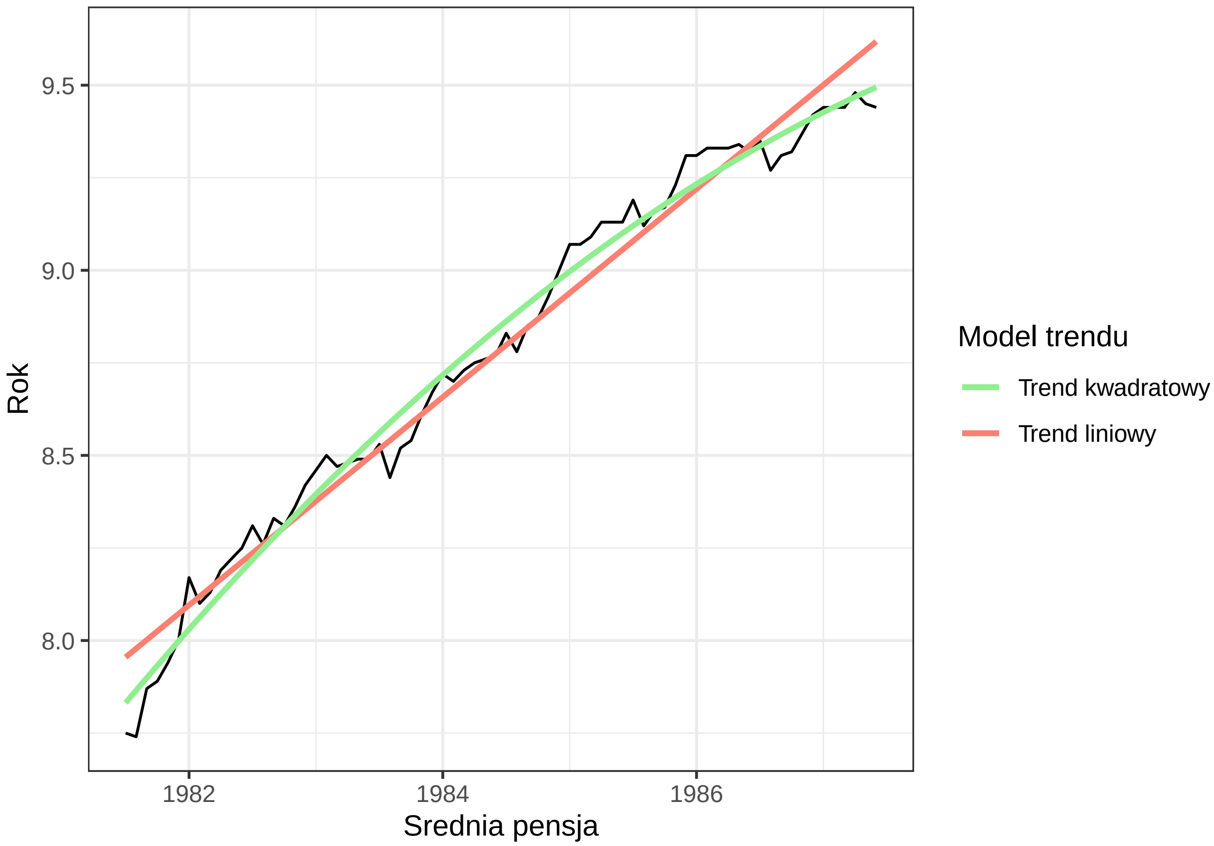
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model_lin)
## W = 0.94775, p-value = 0.00474
```

```r
print(shapiro.test(resid(model_quad)))
```

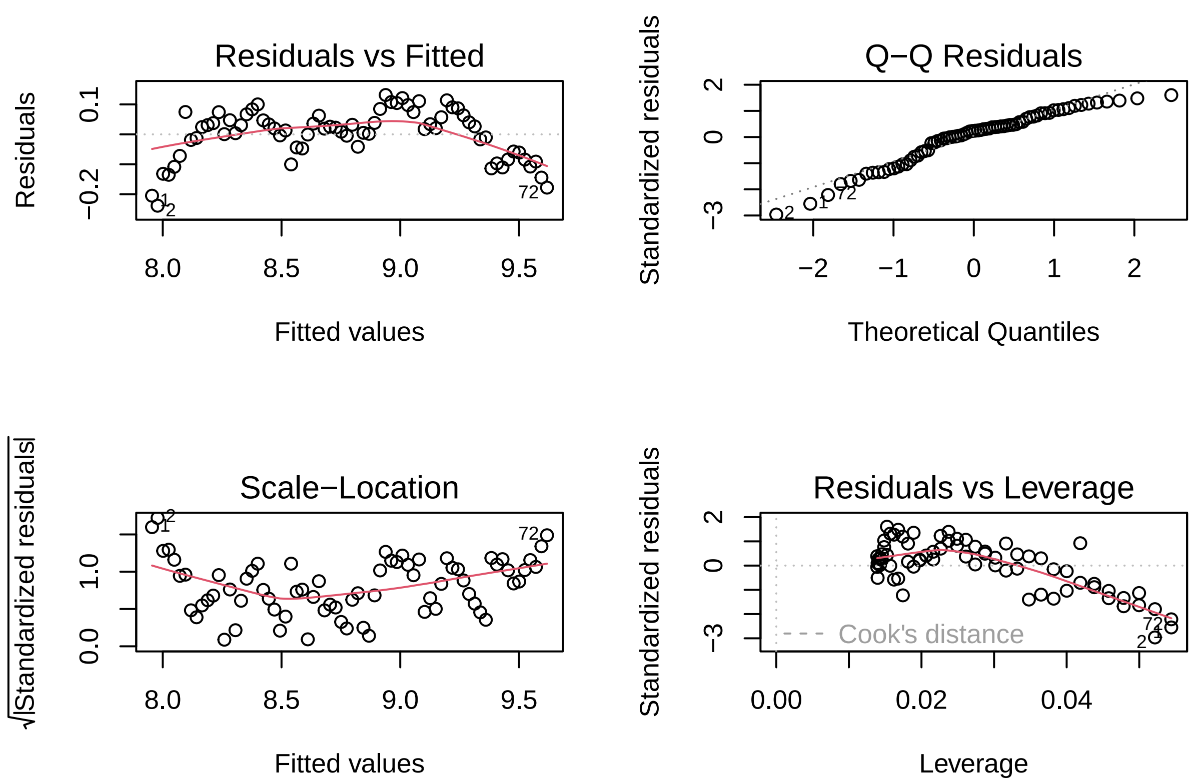
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model_quad)
## W = 0.98856, p-value = 0.7622
```

```r
df |>
  ggplot(aes(y = y, x = time)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE,
              aes(col = "Trend liniowy")) +
  geom_smooth(method = "lm", se = FALSE,
              formula = y ~ x + I(x ^ 2),
              aes(col = "Trend kwadratowy")) +
  scale_color_manual(values = c("Trend liniowy" = "salmon",
                                "Trend kwadratowy" = "lightgreen")) +
  labs(color = "Model trendu",
       x = "Średnia pensja",
       y = "Rok") +
  theme_bw()
```

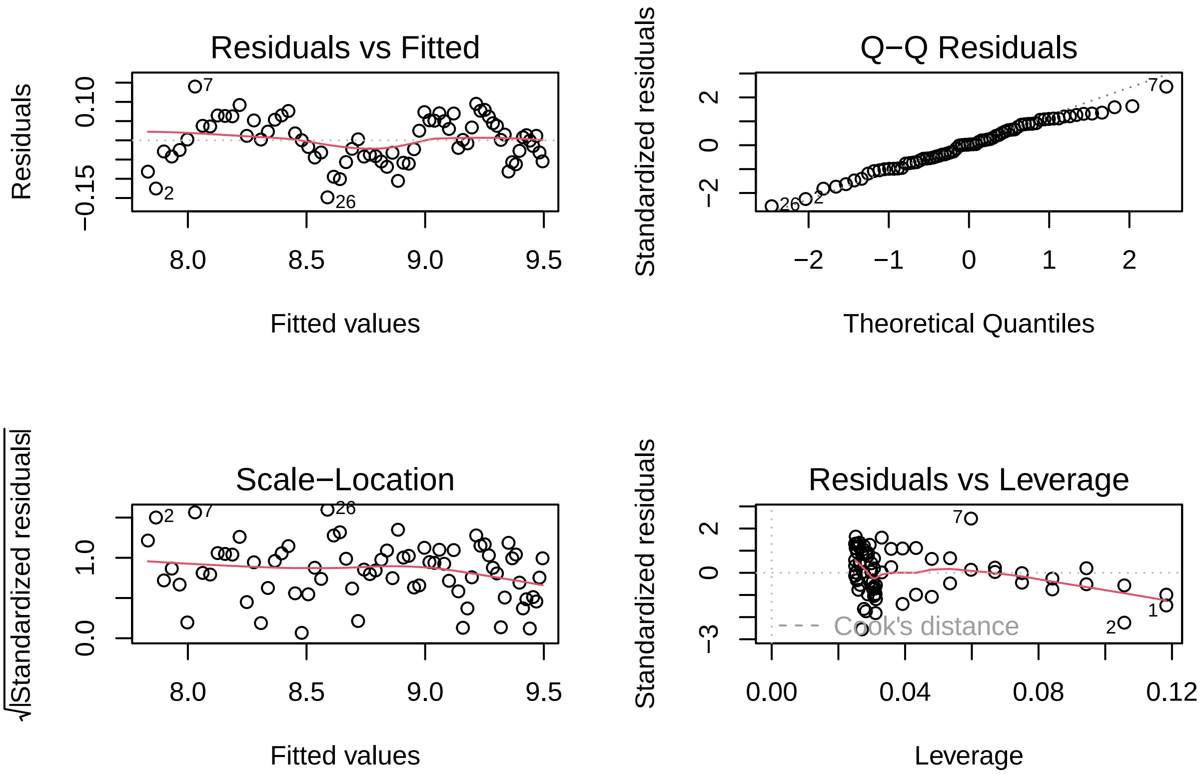
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
par(mfrow = c(2, 2))
plot(model_lin, which = c(1:3, 5))
```



```r
par(mfrow = c(2, 2))
plot(model_quad, which = c(1:3, 5))
```



Zadanie 3

```r
df <- readr::read_csv("https://drizzt.home.amu.edu.pl/images/DADA_AIPD/female.txt",
    col_names = FALSE)$X1 |> ts(start = c(1961, 7), frequency = 12)
```

```
## Rows: 300 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): X1
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#autoplot(forecast(df, h = 17))
model_ma <- arima(df, order = c(0, 0, 17))

autoplot(df) +
  geom_line(data = stats::filter(df, filter = rep(1, 17) / 17),
            colour = "red")  +
  geom_line(data = stats::filter(df, filter = rep(1, 12) / 12),
            colour = "green")
```

```
## Warning: Removed 16 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 11 rows containing missing values (`geom_line()`).
```

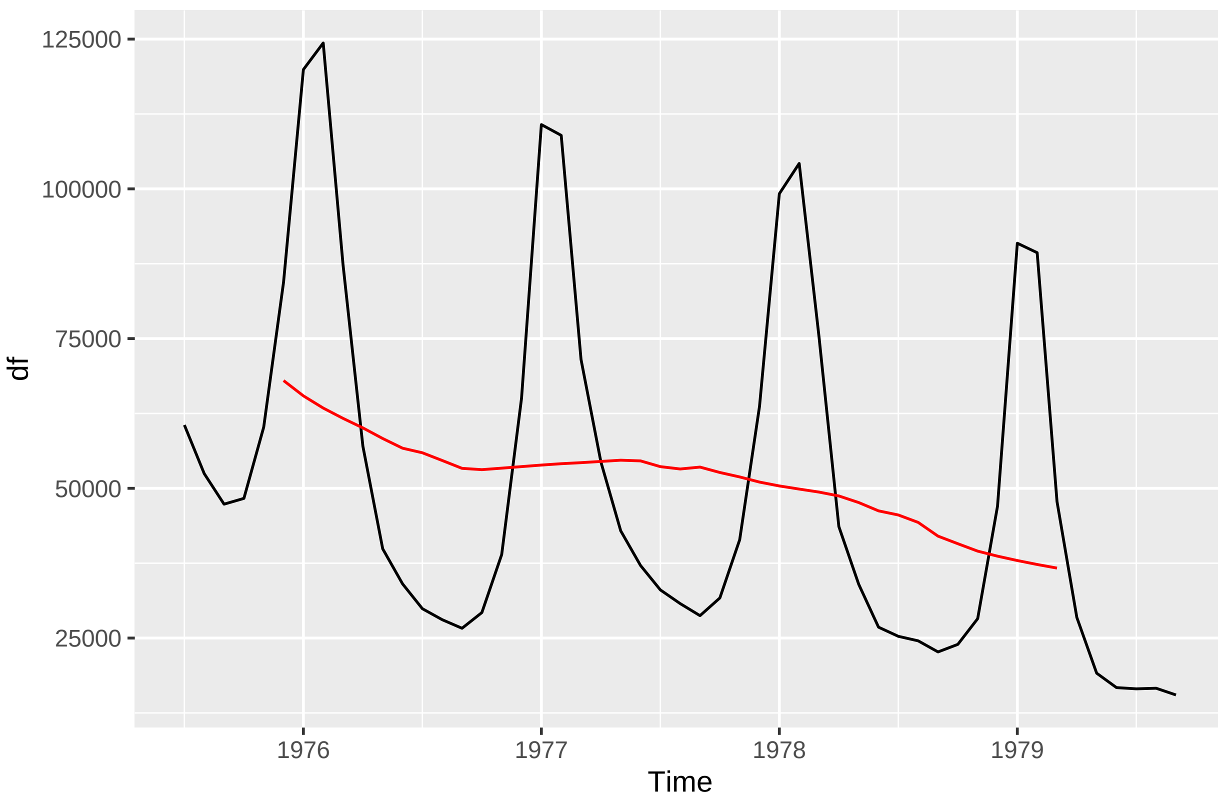
Zadanie 4

```
df <- readr::read_csv("https://drizzt.home.amu.edu.pl/images/DADA_AIPD/unemp.txt",
    col_names = FALSE)$X1 |> ts(start = c(1975, 7), frequency = 12)
```

```
## Rows: 51 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): X1
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
autoplot(df) +
  geom_line(data = stats::filter(df, filter = rep(1, 12) / 12),
            colour = "red")
```

```
## Warning: Removed 11 rows containing missing values (`geom_line()`).
```




```
tseries::adf.test(df, k = 12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df
## Dickey-Fuller = -1.1149, Lag order = 12, p-value = 0.9117
## alternative hypothesis: stationary
```

```
HoltWinters(df)
```

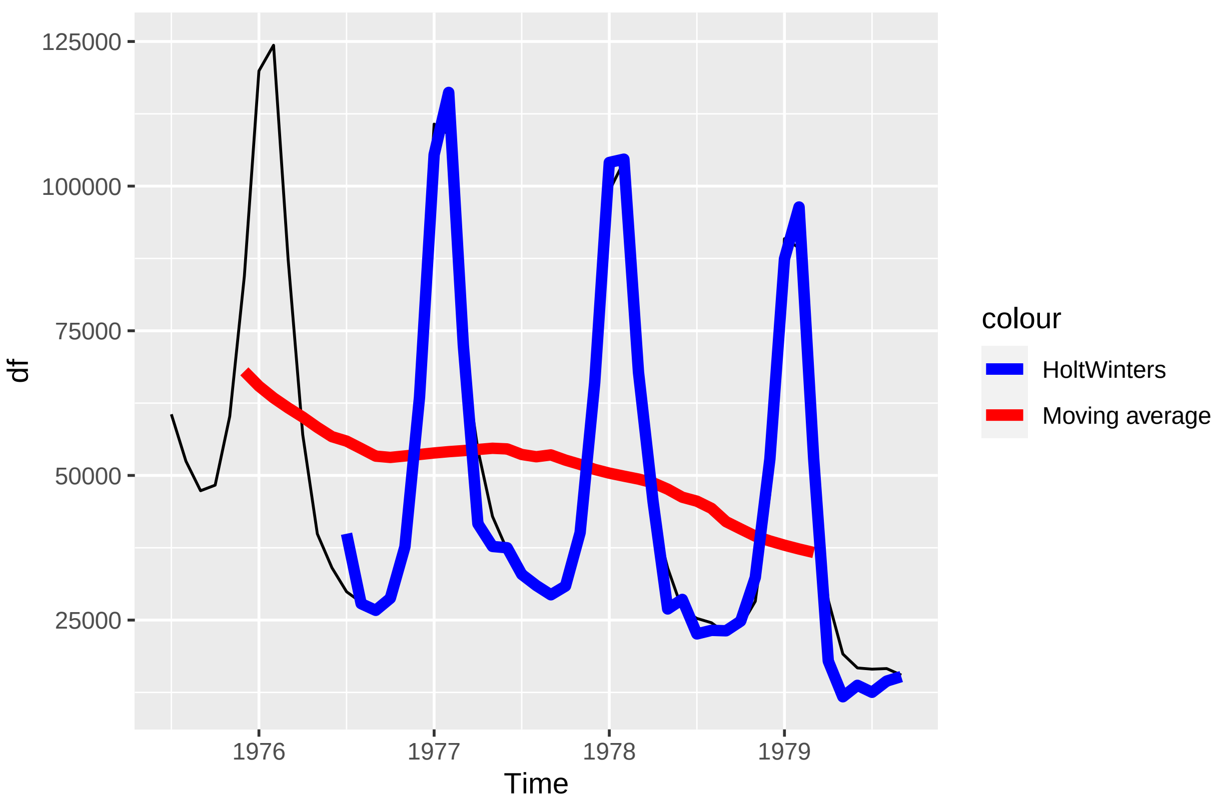
```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = df)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0
##  gamma: 0
##
## Coefficients:
##           [,1]
## a    41488.295
## b    -1253.852
## s1  -22597.670
## s2  -12931.378
## s3   12909.372
## s4   54569.997
## s5   61298.163
```

```
## s6   26136.997
## s7   -2479.337
## s8  -17930.587
## s9  -22086.462
## s10 -25044.045
## s11 -25855.753
## s12 -25989.295
```

```r
autoplot(df) +
  geom_line(data = stats::filter(df, filter = rep(1, 12) / 12),
            aes(col = "Moving average"),
            linewidth = 2) +
  geom_line(data = forecast(HoltWinters(df))$fitted,
            aes(col = "HoltWinters"),
            linewidth = 2) +
  scale_color_manual(values = c("HoltWinters" = "blue",
                                "Moving average" = "red"))
```

```
## Warning: Removed 11 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```



Zadanie 5

- Badanie stacjonarności

Model ARMA(2, 1) / ARIMA(2, 0, 1)

```r
polyroot(c(1, -1, 1/4)) |> abs()
```

```
## [1] 2 2
```

stacjonarny.

Model AR(2) / ARMA(2, 0) / ARIMA(2, 0, 0)

```r
polyroot(c(1, -2, -1)) |> abs()
```

```
## [1] 0.4142136 2.4142136
```

nie jest stacjonarny.

Model ARMA(2, 2) / ARIMA(2, 0, 2)

```r
polyroot(c(1, -.5, .5)) |> abs()
```

```
## [1] 1.414214 1.414214
```

stacjonarny.

Zadanie 6

```r
polyroot(c(1, -3/2, 1/2)) |> abs()
```

```
## [1] 1 2
```

Szereg nie jest stacjonarny.

```r
polyroot(c(1, -5/6, 1/6)) |> abs()
```

```
## [1] 2 3
```

Szereg jest stacjonarny.

```r
polyroot(c(1, -2/3, 5/3)) |> abs()
```

```
## [1] 0.7745967 0.7745967
```

Szereg nie jest stacjonarny

Zadanie 7

```r
data(robot, package = "TSA")
m100 <- arima(robot, order = c(1, 0, 0))
m011 <- arima(robot, order = c(0, 1, 1))
AIC(m100, m011)
```

```
## Warning in AIC.default(m100, m011): models are not all fitted to the same
## number of observations
```

```
##      df       AIC
## m100  3 -2945.078
## m011  2 -2957.901
```

```r
BIC(m100, m011)
```

```
## Warning in BIC.default(m100, m011): models are not all fitted to the same
## number of observations
```

```
##      df       BIC
## m100  3 -2933.735
## m011  2 -2950.346
```

```r
m_auto <- auto.arima(robot)
```

```r
forecast(m_auto, h = 5)
```

```
##     Point Forecast        Lo 80       Hi 80        Lo 95       Hi 95
## 325    0.001659461 -0.001494673 0.004813596 -0.003164371 0.006483294
## 326    0.001421454 -0.001789826 0.004632733 -0.003489775 0.006332682
## 327    0.001421454 -0.001804733 0.004647640 -0.003512573 0.006355480
## 328    0.001421454 -0.001819571 0.004662478 -0.003535266 0.006378173
## 329    0.001421454 -0.001834341 0.004677249 -0.003557855 0.006400763
```

```r
forecast(m011, h = 5)
```

```
##     Point Forecast        Lo 80       Hi 80        Lo 95       Hi 95
## 325    0.001742672 -0.001414569 0.004899913 -0.003085912 0.006571256
## 326    0.001742672 -0.001440618 0.004925962 -0.003125750 0.006611094
## 327    0.001742672 -0.001466455 0.004951799 -0.003165265 0.006650608
## 328    0.001742672 -0.001492086 0.004977430 -0.003204464 0.006689808
## 329    0.001742672 -0.001517516 0.005002859 -0.003243355 0.006728699
```

```r
forecast(m100, h = 5)
```

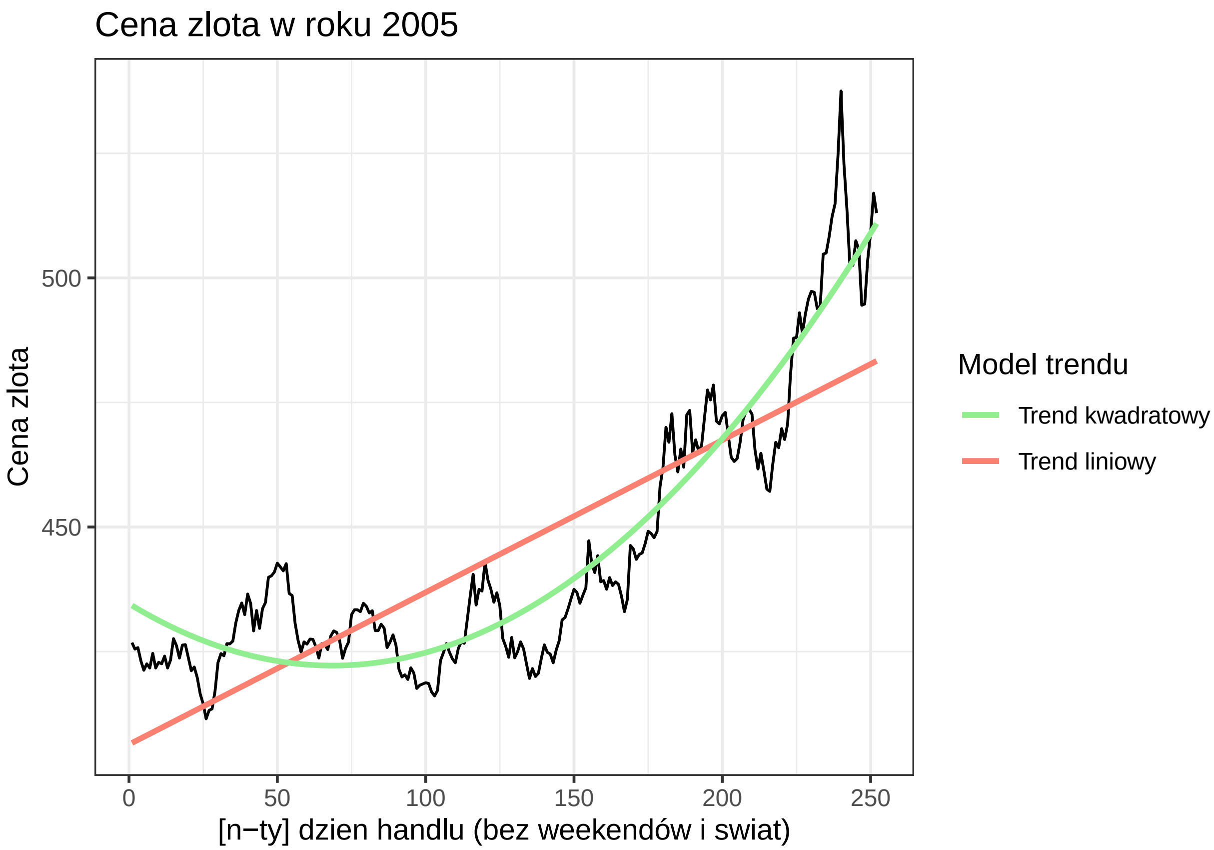
```
##     Point Forecast        Lo 80       Hi 80        Lo 95       Hi 95
## 325    0.002114451 -0.001148405 0.005377307 -0.002875657 0.007104559
## 326    0.001657797 -0.001755739 0.005071333 -0.003562755 0.006878349
## 327    0.001517423 -0.001910009 0.004944854 -0.003724381 0.006759227
## 328    0.001474272 -0.001954469 0.004903014 -0.003769535 0.006718080
## 329    0.001461008 -0.001967857 0.004889873 -0.003782989 0.006705005
```

Zadanie 8

Wizualizacja samego szeregu:

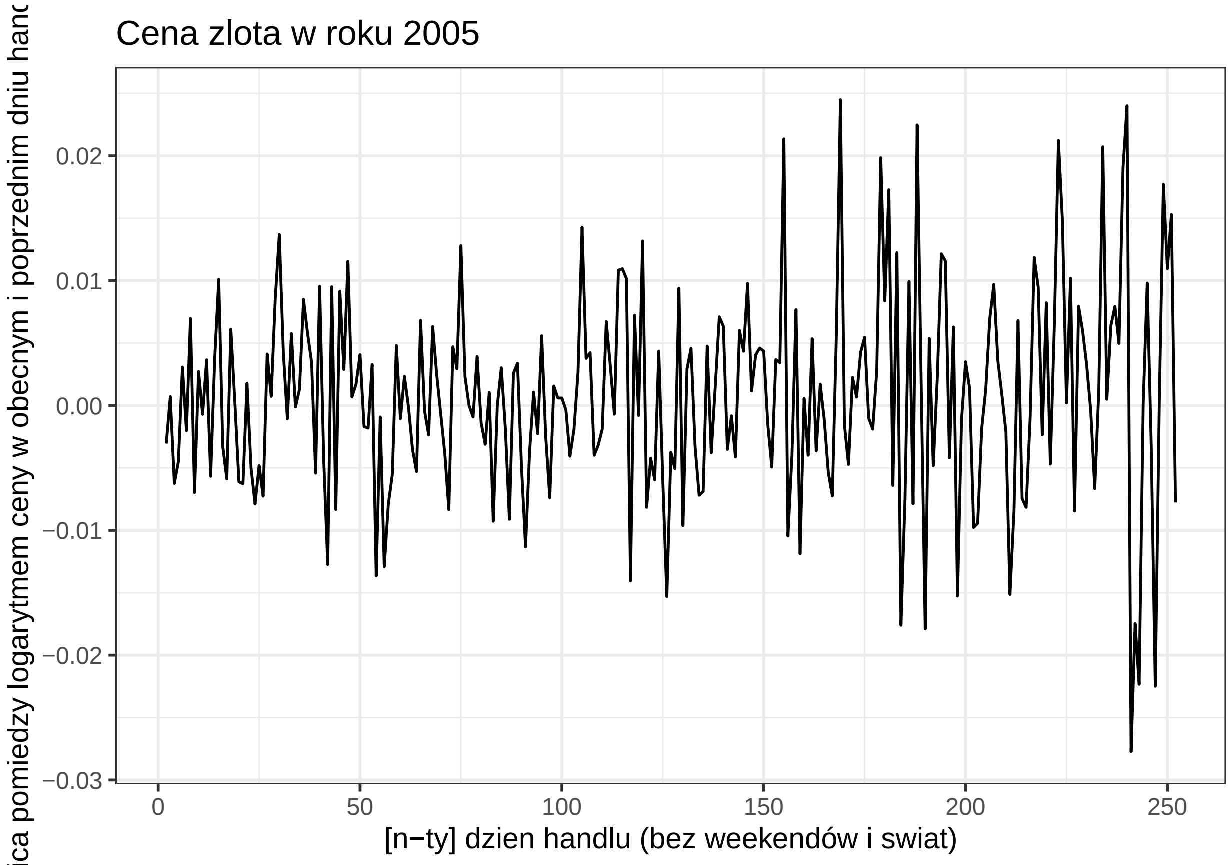
```r
data(gold, package = "TSA")
data.frame(y = as.numeric(gold),
           time = time(gold) |> as.numeric()) |>
  ggplot(aes(y = y, x = time)) +
  geom_line() +
  geom_smooth(method = "lm", se = FALSE,
              aes(col = "Trend liniowy")) +
  geom_smooth(method = "lm", se = FALSE,
              formula = y ~ x + I(x ^ 2),
              aes(col = "Trend kwadratowy")) +
  scale_color_manual(values = c("Trend liniowy" = "salmon",
                                "Trend kwadratowy" = "lightgreen")) +
  labs(color = "Model trendu",
       x = "[n-ty] dzień handlu (bez weekendów i świąt)",
       y = "Cena złota") +
  ggtitle("Cena złota w roku 2005") +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Wizualizacja różnic logarytmów:

```
xx <- diff(log(gold))
data.frame(y = xx |> as.numeric(),
           time = time(xx) |> as.numeric()) |>
  ggplot(aes(y = y, x = time)) +
  geom_line() +
  labs(x = "[n-ty] dzień handlu (bez weekendów i świąt)",
       y = "Różnica pomiędzy logarytmem ceny w obecnym i poprzednim dniu handlowym") +
  ggtitle("Cena złota w roku 2005") +
  theme_bw()
```



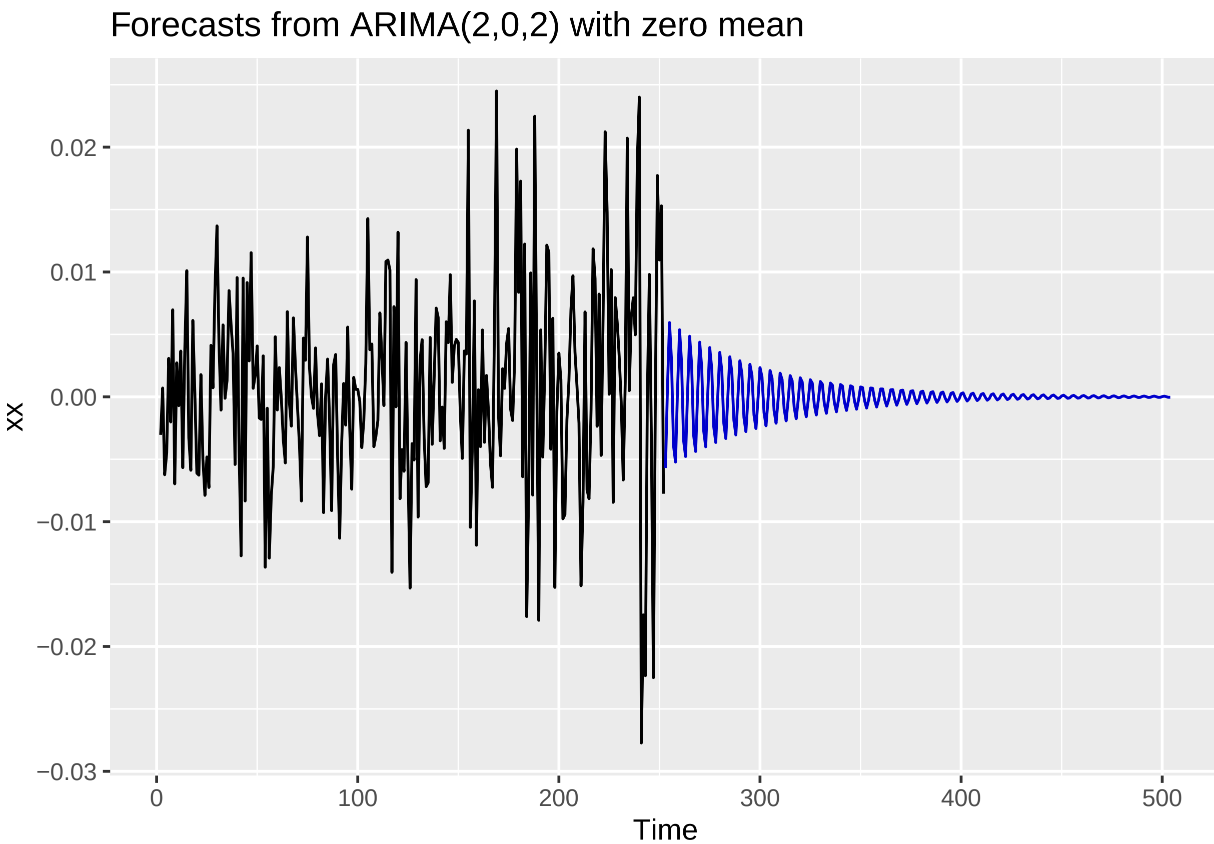
Model ARIMA:

```r
m_auto <- auto.arima(y = xx)
m_auto
```

```
## Series: xx
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.6139  -0.9618  -0.5364  0.9303
## s.e.  0.0317   0.0322   0.0439  0.0424
##
## sigma^2 = 6.375e-05:  log likelihood = 857.81
## AIC=-1705.63   AICc=-1705.38   BIC=-1688
```

Wizualizacja modelu z predykcją na następny rok:

```r
autoplot(forecast(m_auto, h = 252, level = 0))
```


Forecasts from ARIMA(2,0,2) with zero mean
0.02
0.01
0.00
−0.01
−0.02
−0.03
xx
0
100
200
300
400
500
Time