Analiza szeregów czasowych - notowania giełdowe firm McDonald's oraz Starbucks

Alicja Hołowiecka, Matylda Jankowska, Marcin Dziadosz

23 12 2019

Contents

Figure 1: Jedzenie z McDonald's

Wstęp

W tym raporcie przeanalizujemy dwa szeregi czasowe: notowania firm McDonald's oraz Starbucks z okresu dwóch lat (od początku 2018 do końca 2019). Na potrzeby oceny w raporcie pojawia się nie tylko sama analiza, ale też wszystkie polecenia w języku `R`, jakich używaliśmy w jej celu.

McDonald's

Opis firmy

McDonald's to największa na świecie sieć restauracji szybkiej obsługi. Obejmuje ona ponad 30 tys. restauracji, każdego dnia obsługujących ponad 46 mln osób w 119 krajach. Wartość marki McDonald's szacuje się na 24,7 mld dolarów.

Wczytanie danych i rysunki

Na początek wczytujemy bibliotekę `tseries`, która będzie nam potrzebna do wykonania analizy szeregu czasowego.

```
library(tseries)
```

Dane pobieramy z `yahoo finance` za pomocą funkcji `get.hist.quote` i zamieniamy na typ numeryczny.

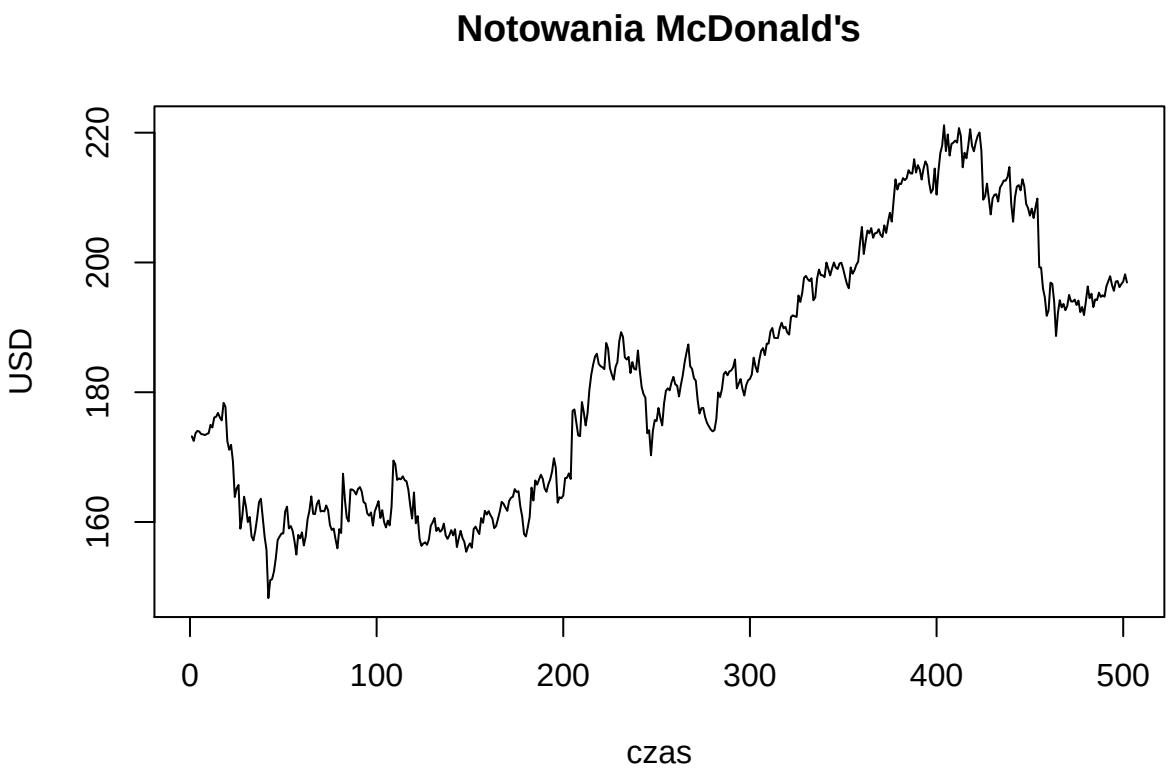
```
mcd<- get.hist.quote(instrument = "MCD", provider = "yahoo",
                         quote = "Close", start = "2018-01-01", end = "2019-12-31")
```

```
## time series starts 2018-01-02
## time series ends   2019-12-30
```

```
mcd <- as.numeric(mcd)
```

Wykonamy rysunek przedstawiający notowania firmy McDonald's od 01-01-2018 do 31-12-2019

```
plot(mcd, type = "l", xlab = "czas", ylab = "USD", main = "Notowania McDonald's")
```



Na rysunku w ciągu tych dwóch lat wyraźnie widać trend rosnący.

Dopasowanie wielomianu

Spróbujemy do danych dopasować wielomian stopnia 1, 2 i 3.

```
t <- 1:length(mcd)
```

Model liniowy

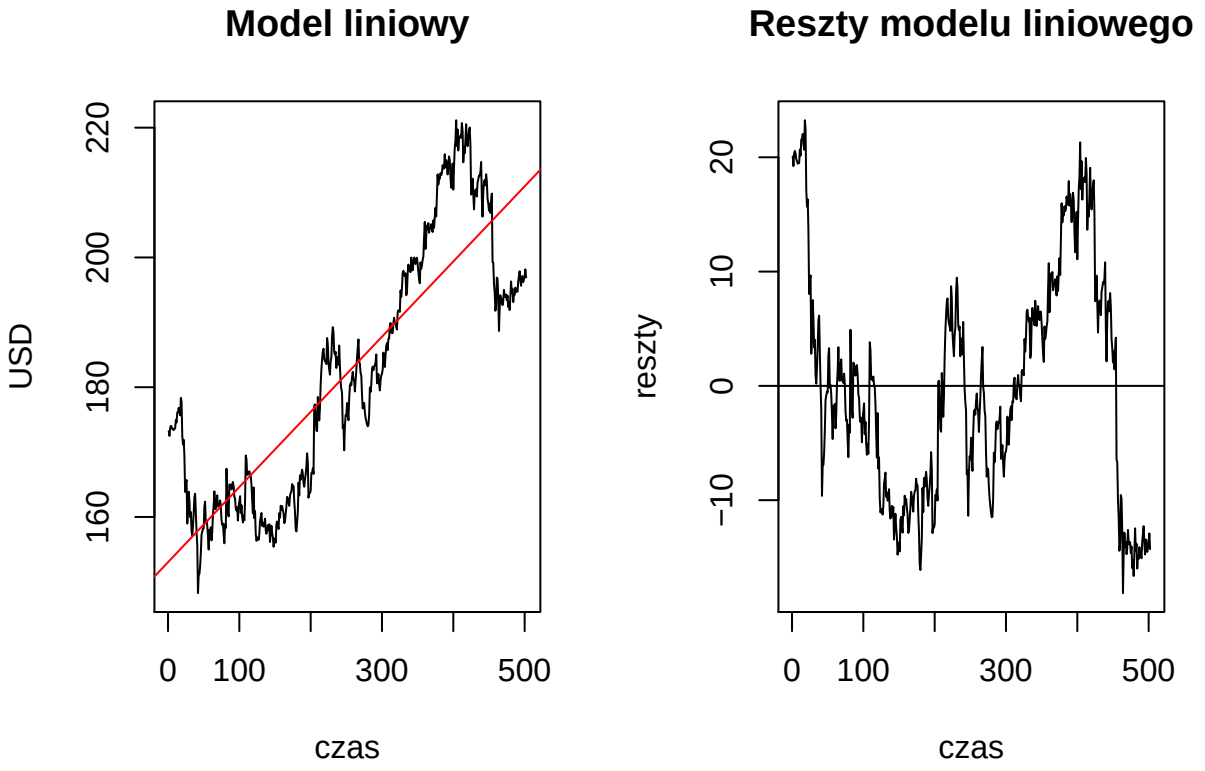
```
mod1 <- lm(mcd~t)
summary(mod1)
```

```
##
## Call:
## lm(formula = mcd ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.1318  -8.6506  -0.5902   6.3288  23.2450
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.530e+02  8.855e-01  172.82   <2e-16 ***
## t           1.159e-01  3.051e-03   37.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.905 on 500 degrees of freedom
## Multiple R-squared:  0.7426, Adjusted R-squared:  0.7421
## F-statistic:  1443 on 1 and 500 DF,  p-value: < 2.2e-16
```

Zarówno wyraz wolny, jak i współczynnik kierunkowy są istotne statystycznie. R^2 wynosi około 74%.

```
par(mfrow = c(1, 2))
plot(mcd, type = "l", main = "Model liniowy", xlab = "czas", ylab = "USD")
abline(mod1, col = "red")
plot(mod1$residuals, type = "l", main = "Reszty modelu liniowego", xlab = "czas", ylab = "reszty")
abline(h=0)
```




```
par(mfrow = c(1, 1))
```

Model kwadratowy

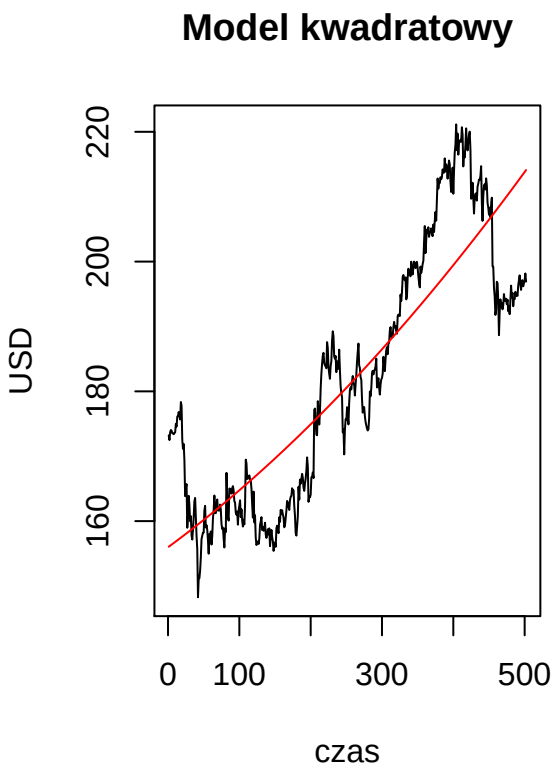
Teraz stworzymy model kwadratowy.

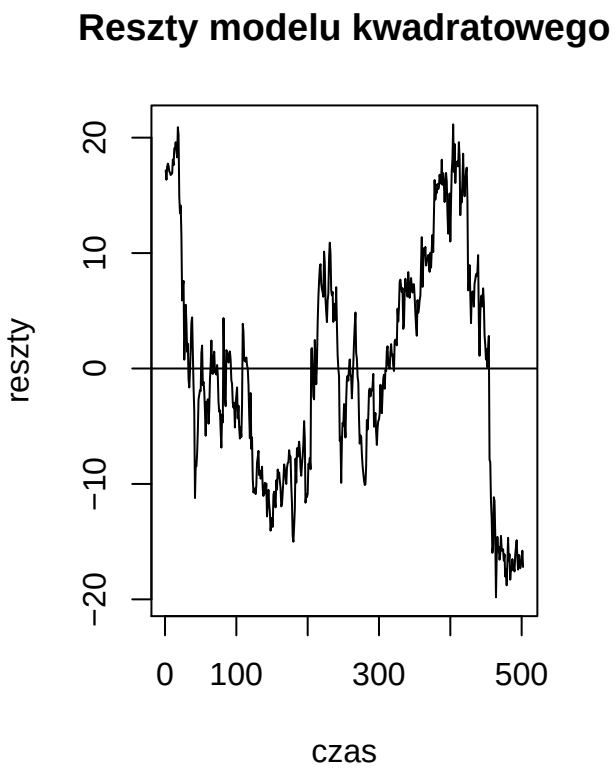
```
mod2 <- lm(mcd~t+I(t^2))
summary(mod2)
```

```
##
## Call:
## lm(formula = mcd ~ t + I(t^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8159  -7.7496  -0.3429   6.7492  21.1529
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.560e+02  1.321e+00 118.060  < 2e-16 ***
## t           8.080e-02  1.213e-02   6.661 7.19e-11 ***
## I(t^2)      6.972e-05  2.335e-05   2.985  0.00297 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.828 on 499 degrees of freedom
## Multiple R-squared:  0.7471, Adjusted R-squared:  0.7461
## F-statistic: 737.1 on 2 and 499 DF,  p-value: < 2.2e-16
```

Wszystkie współczynniki są istotne statystycznie. R^2 wynosi około 75%, a więc zmieniło się bardzo nieznacznie.

```
par(mfrow = c(1, 2))
plot(mcd, type = "l", main = "Model kwadratowy", xlab = "czas", ylab = "USD")
lines(t, mod2$fitted.values, col = "red", )
plot(mod2$residuals, type = "l", main = "Reszty modelu kwadratowego", xlab = "czas", ylab = "reszty")
abline(h = 0)
```






```
par(mfrow = c(1, 1))
```

Model kwadratowy zachowuje się bardzo podobnie jak model liniowy.

Model sześcienny

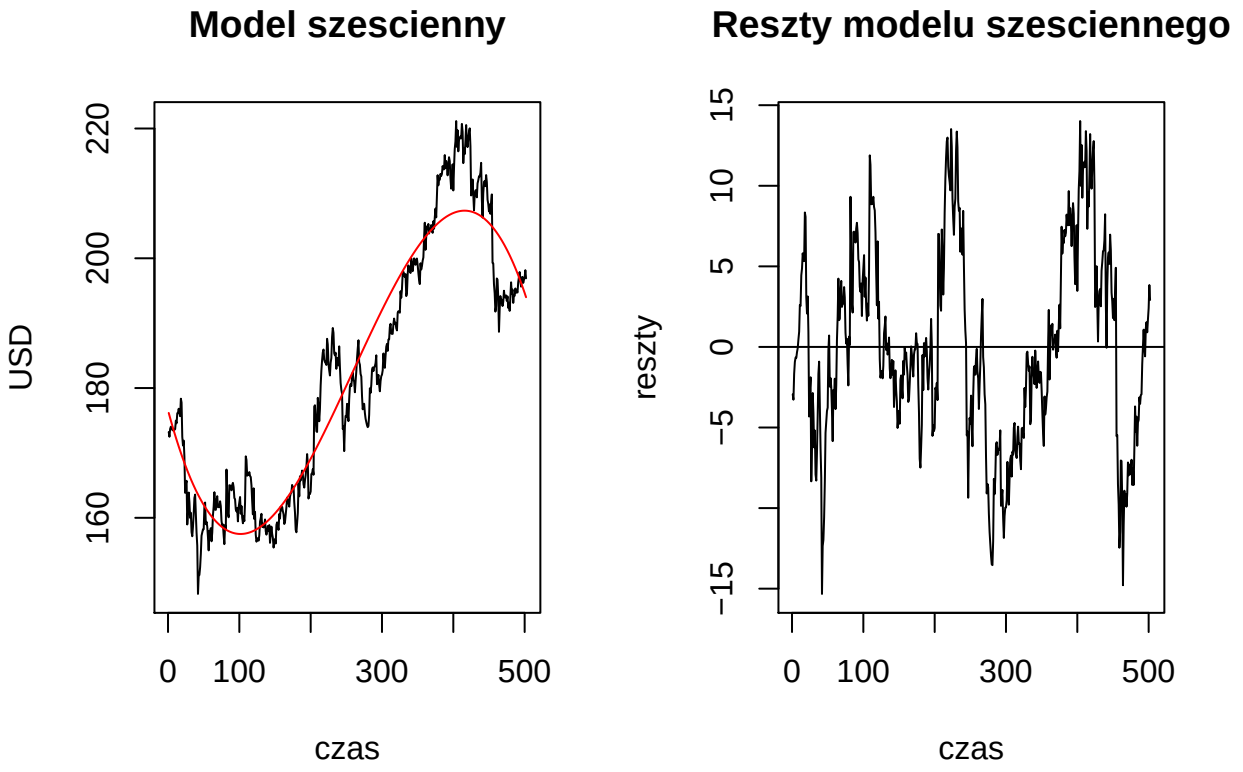
```
mod3 <- lm(mcd~t+I(t^2)+I(t^3))
summary(mod3)
```

```
##
## Call:
## lm(formula = mcd ~ t + I(t^2) + I(t^3))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.3266  -4.3160  -0.4541   4.2902  14.0168
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.766e+02  1.096e+00  161.08   <2e-16 ***
## t           -4.081e-01  1.885e-02  -21.65   <2e-16 ***
## I(t^2)       2.497e-03  8.705e-05   28.69   <2e-16 ***
## I(t^3)      -3.217e-06  1.138e-07  -28.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 6.094 on 498 degrees of freedom
## Multiple R-squared:  0.903,  Adjusted R-squared:  0.9024
## F-statistic:  1545 on 3 and 498 DF,  p-value: < 2.2e-16
```

W modelu sześciennym wszystkie współczynniki są istotne statystycznie. R^2 wynosi 90%, a więc znacząco się poprawił w stosunku do poprzednich dwóch modeli.

```
par(mfrow = c(1, 2))
plot(mcd, type = "l", main = "Model sześcienny", xlab = "czas", ylab = "USD")
lines(t, mod3$fitted.values, col = "red")
plot(mod3$residuals, type = "l", main = "Reszty modelu sześciennego", xlab = "czas", ylab = "reszty")
abline(h= 0)
```



Widać, że reszty modelu mają mniejszy rozrzut niż poprzednio - teraz mamy skalę od -15 do 15, a wcześniej było od -20 do 20.

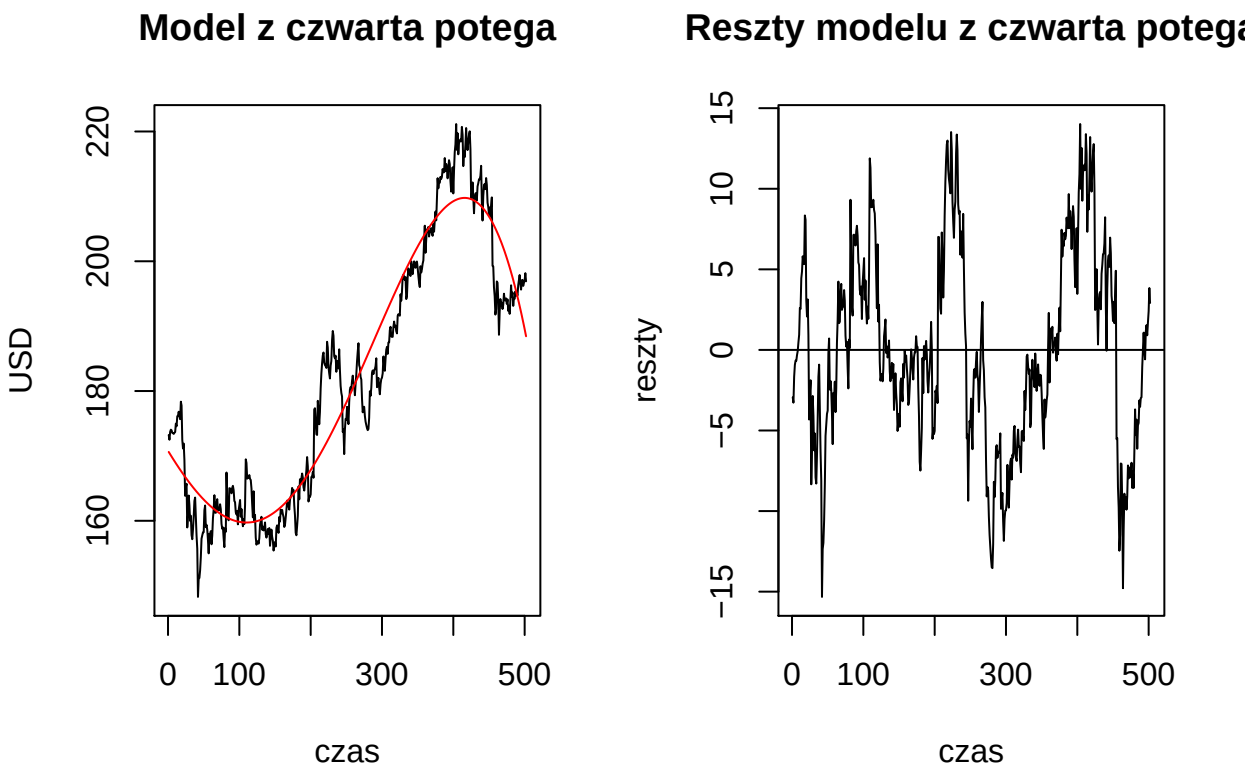
Model z czwartą potęgą

```
mod4 <- lm(mcd~t+I(t^2)+I(t^3)+I(t^4))
summary(mod4)
```

```
##
## Call:
## lm(formula = mcd ~ t + I(t^2) + I(t^3) + I(t^4))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.9730  -3.9885  -0.5363   4.0235  15.3758
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.708e+02  1.309e+00 130.435  < 2e-16 ***
## t           -1.804e-01  3.599e-02  -5.012 7.52e-07 ***
## I(t^2)       4.647e-04  2.906e-04   1.599 0.110408
## I(t^3)       3.064e-06  8.676e-07   3.532 0.000451 ***
## I(t^4)      -6.244e-09  8.557e-10  -7.297 1.17e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.798 on 497 degrees of freedom
## Multiple R-squared:  0.9124, Adjusted R-squared:  0.9116
## F-statistic:  1293 on 4 and 497 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(1, 2))
plot(mcd, type = "l", main = "Model z czwartą potęgą", xlab = "czas", ylab = "USD")
lines(t, mod4$fitted.values, col = "red")
plot(mod3$residuals, type = "l", main = "Reszty modelu z czwartą potęgą", xlab = "czas", ylab = "reszty
abline(h= 0)
```



W modelu z czwartą potęgą współczynnik przy t^3 jest nieistotny statystycznie, ale nie możemy go usunąć, ponieważ efekt wyższego rzędu (t^4) jest istotny. R^2 wynosi około 91%, więc niewiele się różni od modelu sześciennego. Reszty także znajdują się w podobnym przedziale jak w poprzednim modelu. Model z t^4 niewiele się różni od tego z t^3, dlatego do dalszych badań wykorzystamy model sześcienny.

Testy jednorodności wariancji reszt

Aby zbadać czy reszty w modelu są homoskedastyczne posłużymy się kilkoma popularnymi testami.

Test Breuscha-Pagana

H_0 : jednorodność wariancji reszt. H_1 : wariancja reszt zależy od zmiennych objaśniających w modelu

```
library(lmtest)
pv1 <- bptest(mod3)$p.value
```

P-value wynosi 0.0000044826, należałoby zatem odrzucić hipotezę o jednorodności wariancji reszt.

Test Goldfelda-Quandta

Weryfikacja hipotezy polega na podziale danych na dwie grupy i sprawdzeniu, czy w obu wariancja ma taką samą wartość.

```
pv2 <- gqtest(mod3, order.by = ~fitted(mod3))$p.value
```

P-value wynosi 0.678, zatem nie ma podstaw do odrzucenia hipotezy o równości wariancji.

Test Harrisona-McCabe'a

Sprawdza hipotezę podobną do tej, którą weryfikuje test Goldfelda-Quandta; jednak w tym przypadku porównuje pierwszą połowę wartości do całości danych.

```
hmctest(mod3, order.by = ~fitted(mod3))
```

```
##
##  Harrison-McCabe test
##
## data:  mod3
## HMC = 0.39433, p-value < 2.2e-16
```

```
pv3 <- hmctest(mod3, order.by = ~fitted(mod3))$p.value
```

P-value wynosi jest praktycznie równe 0, należy przyjąć hipotezę alternatywną, czyli wariancja reszt modelu ulega zmianie.

Biorąc pod uwagę uzyskane wyniki, należy przyjąć, że reszty z modelu trzeciego stopnia są heteroskedastyczne.

Ruchoma średnia

Wykorzystamy metody ruchomych średnich, aby wygładzić szereg i zaobserwować ogólne trendy. Metoda średniej ruchomej ma na celu zmniejszenie rozrzutu razy $m+1$.

W metodzie średniej ruchomej estymator części deterministycznej ma postać

$$\hat{f}(t) = \frac{1}{m+1} \sum_{k=0}^{m} x_{t-k}$$

Do wykonania wygładzonych wykresów napisaliśmy funkcję `ruchoma`, której argumentami są `x` - szereg czasowy, `m` - paramter metody średniej ruchomej, `kolor` - kolor, na jaki dorysujemy wygładzoną linię na wykresie.

```r
ruchoma <- function(x, m, kolor){
  t <- length(x)
  f <- NULL
  for(i in (m+1):t){
    f[i] <- mean(x[(i-m):i])
  }
  plot(x, type = "l")
  lines((m+1):t, f[(m+1):t], lwd = 2, col = kolor)
}
```

Narysujemy wykresy dla kilku parametrów `m`.

Dla `m` = 3:

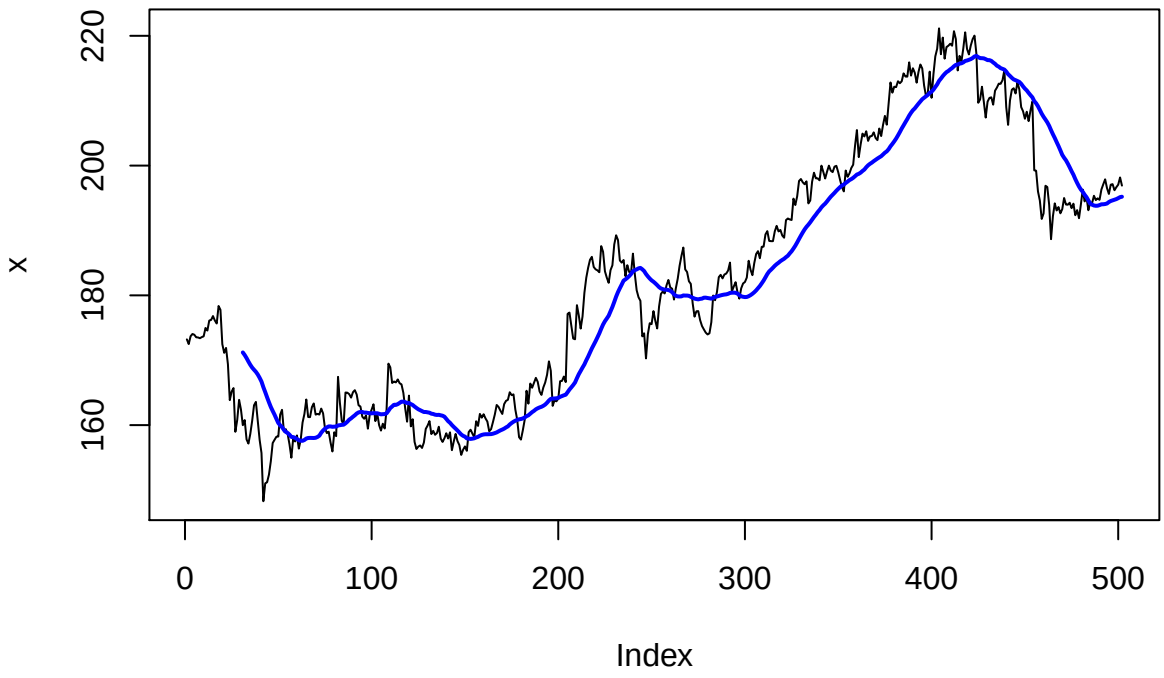
```r
ruchoma(mcd, 3, "red")
```

Dla `m` = 10:

```r
ruchoma(mcd, 10, "green")
```

Dla m = 30:

```
ruchoma(mcd, 30, "blue")
```



Jak widać, im większy parametr `m` przyjmiemy, tym bardziej wygładzony wykres uzyskujemy, ale też mniej dokładny.

Metoda wykładniczych wag ruchomej średniej

W metodzie ruchomej średniej obserwacje starsze i nowsze mają taką samą wagę, dlatego ta metoda jest mało dokładna. Skorzystamy teraz z dokładniejszej metody wykładniczych wag ruchomej średniej.

W tej metodzie estymator części deterministycznej ma postać:

$$\hat{f}(t) = \frac{1-\eta}{1-\eta^t} \sum_{k=0}^{t-1} \eta^k x_{t-k}$$

gdzie $\eta \in (0,1)$

Skorzystamy z postaci rekurencyjnej:

$$\hat{f}(t) = \frac{1-\eta}{1-\eta^t} \left[x_t + \eta \frac{1-\eta^{t-1}}{1-\eta} \hat{f}(t-1) \right]$$

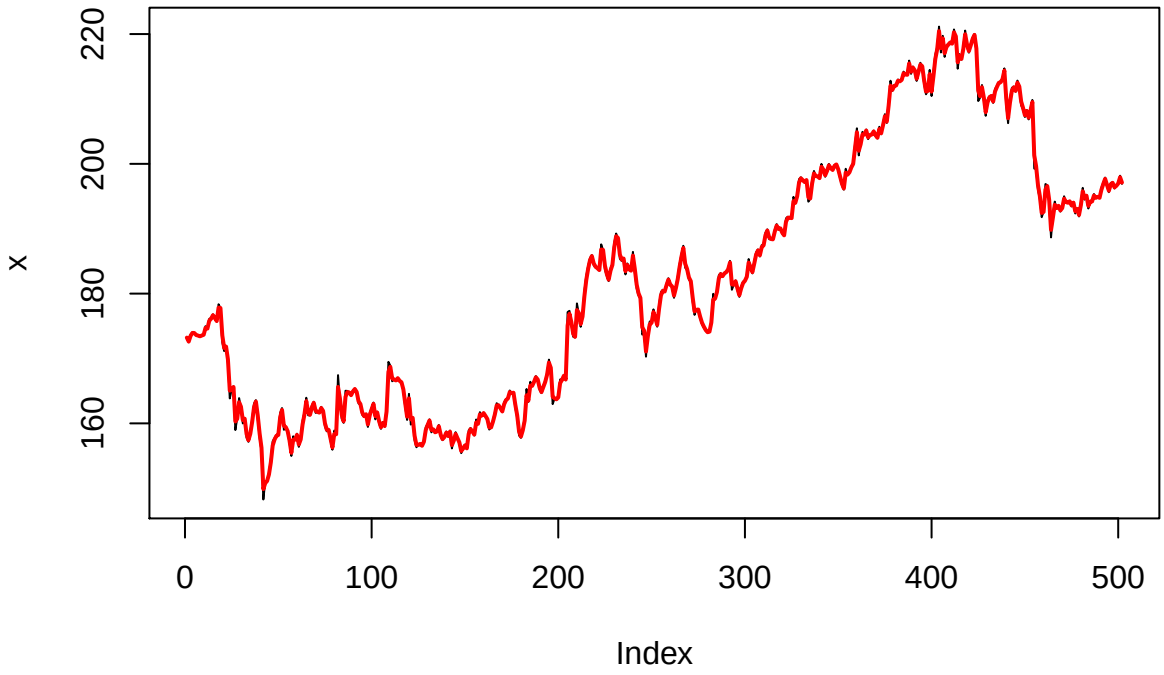
```
wykladnicza <- function(x, mi, kolor){
  f <- NULL
  f[1] <- x[1]

  for (i in 2:length(x)){
```

```
    f[i] <- (1-mi)/(1-mi^i)*(x[i]+mi*(1-mi^(i-1))/(1-mi)*f[i-1])
  }
  plot(x, type = "l")
  lines(1:length(x), f, lwd = 2, col = kolor)
}
```

Dla $\eta = 0.2$

```
wykladnicza(mcd, 0.2, "red")
```

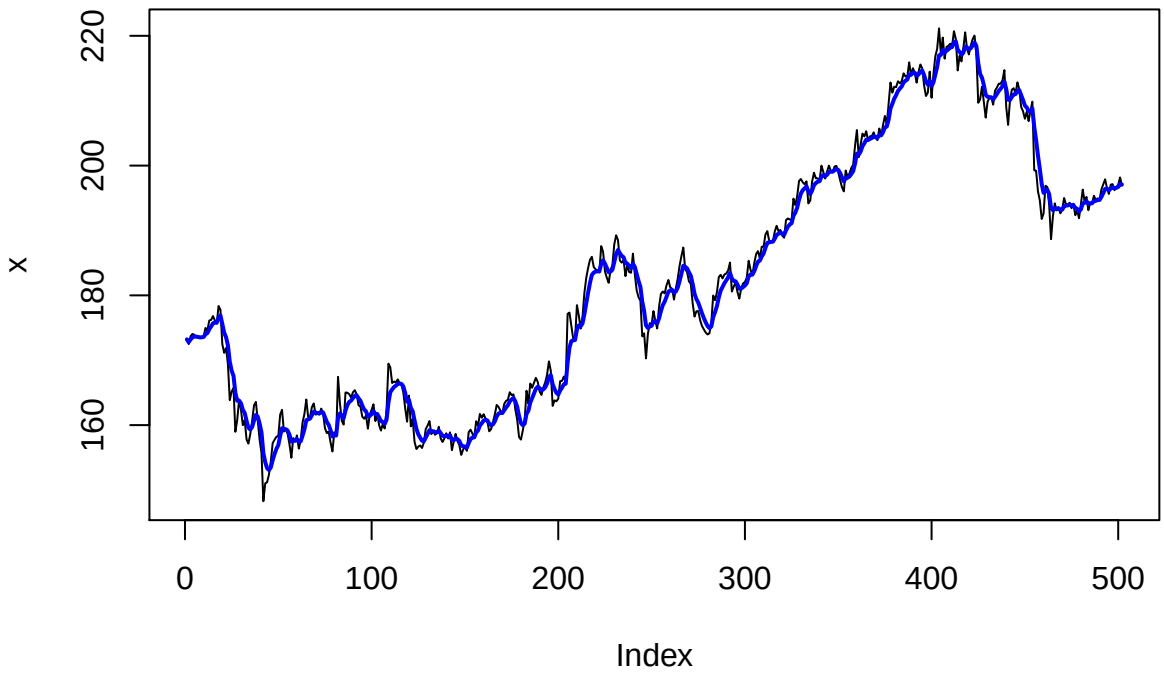


Dla $\eta = 0.5$

```
wykladnicza(mcd, 0.5, "green")
```

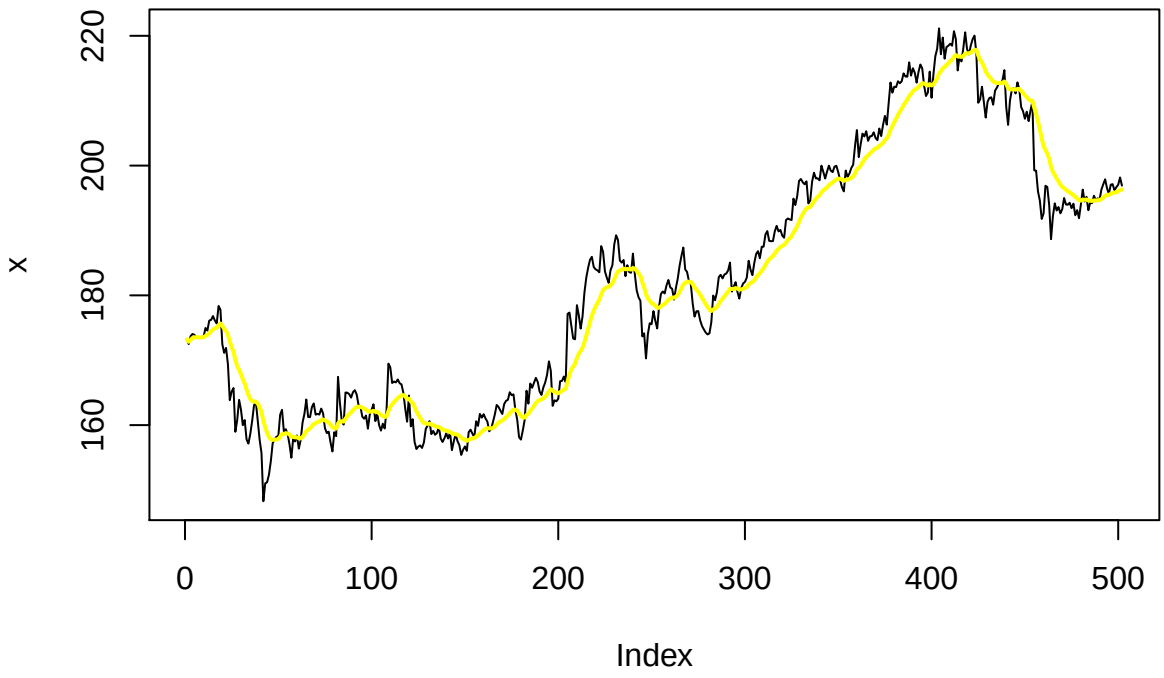
Dla $\eta = 0.7$

```
wykladnicza(mcd, 0.7, "blue")
```



Dla $\eta = 0.9$

```
wykladnicza(mcd, 0.9, "yellow")
```



Podobnie jak w przypadku prostej metody średniej ruchomej - im większy parametr η, tym bardziej wygładzony wykres, ale i mniejsza dokładność.

Testy na resztach modelu

Do danych dobraliśmy wcześniej model wielomianowy trzeciego stopnia. Teraz sprawdzimy, czy reszty tego modelu spełniają założenia:

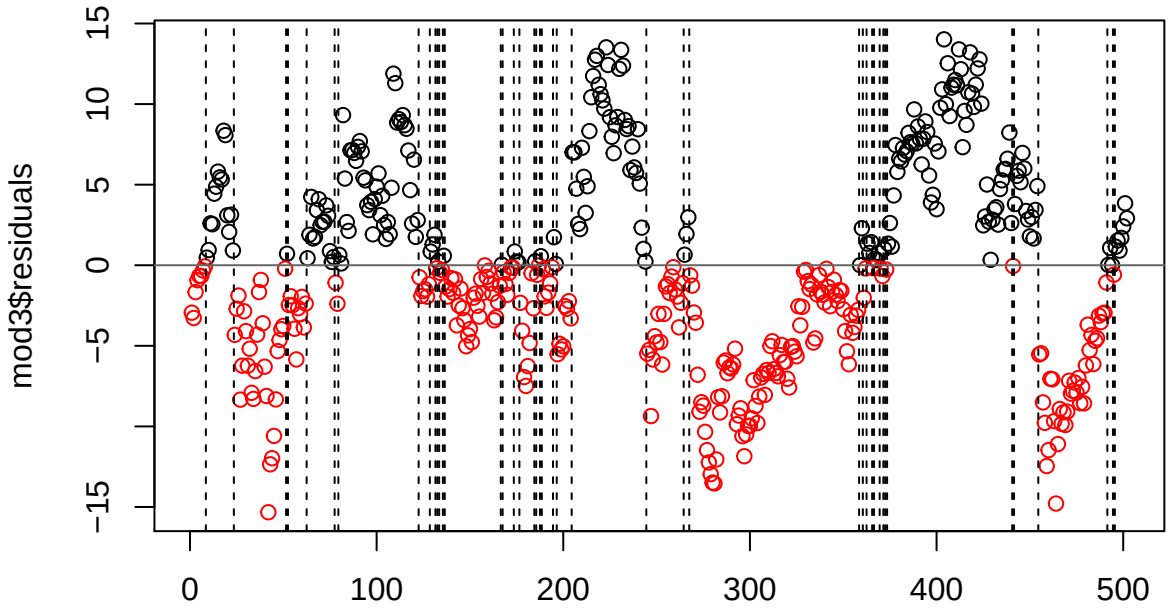
- losowość
- jednorodność wariancji

-normalność

- brak autokorelacji

Losowość

```
library(randtests)
runs.test(mod3$residuals, threshold = 0, plot = T)
```

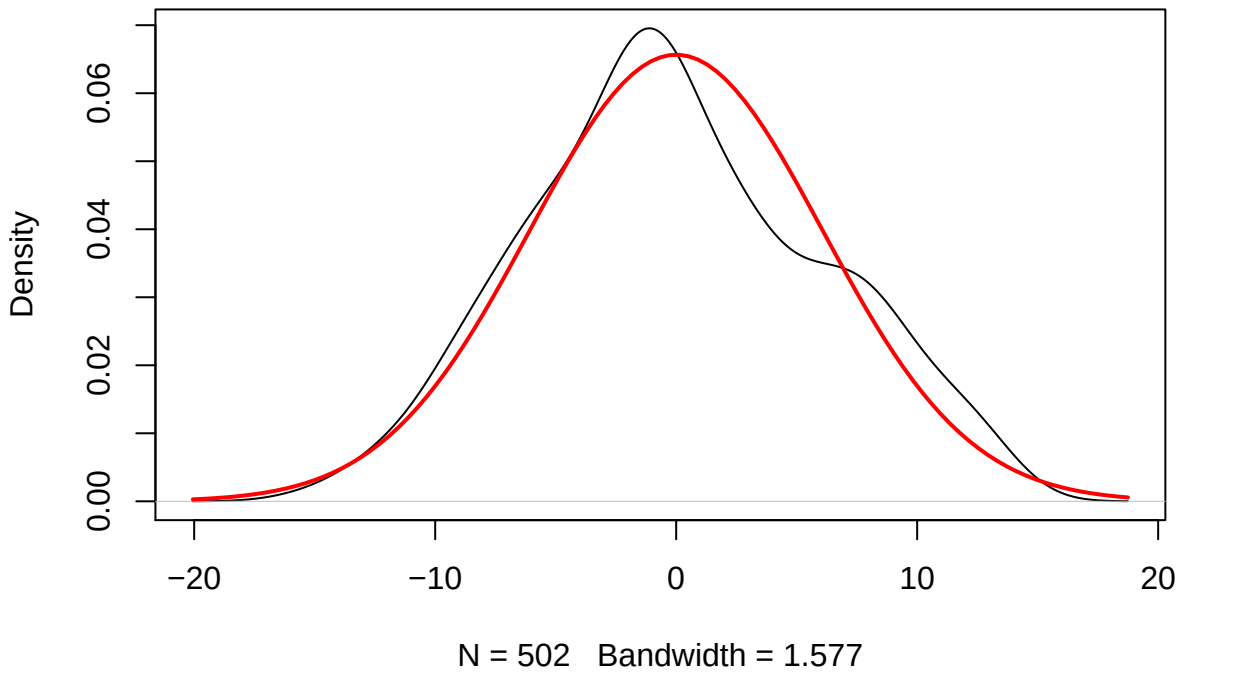



```
##
##  Runs Test
##
## data:  mod3$residuals
## statistic = -18.556, runs = 44, n1 = 229, n2 = 273, n = 502,
## p-value < 2.2e-16
## alternative hypothesis: nonrandomness
```

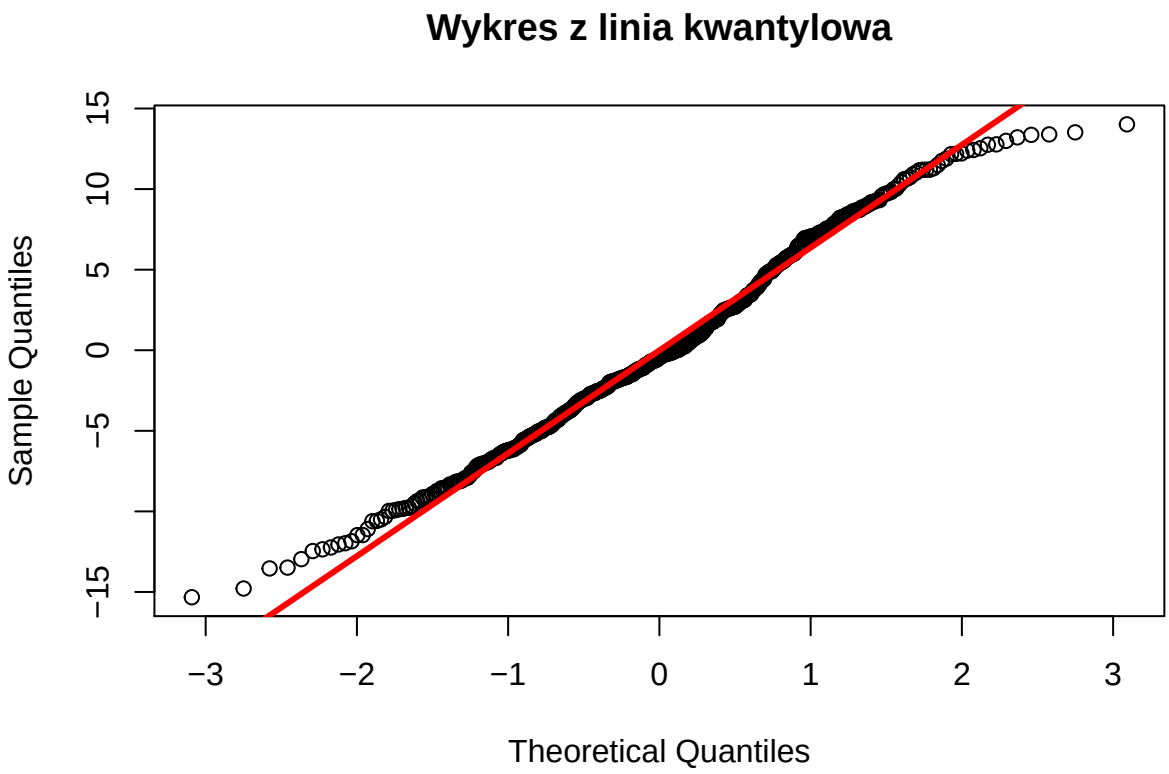
P-value bliskie zero, odrzucamy hipotezę o losowości reszt.

Normalność

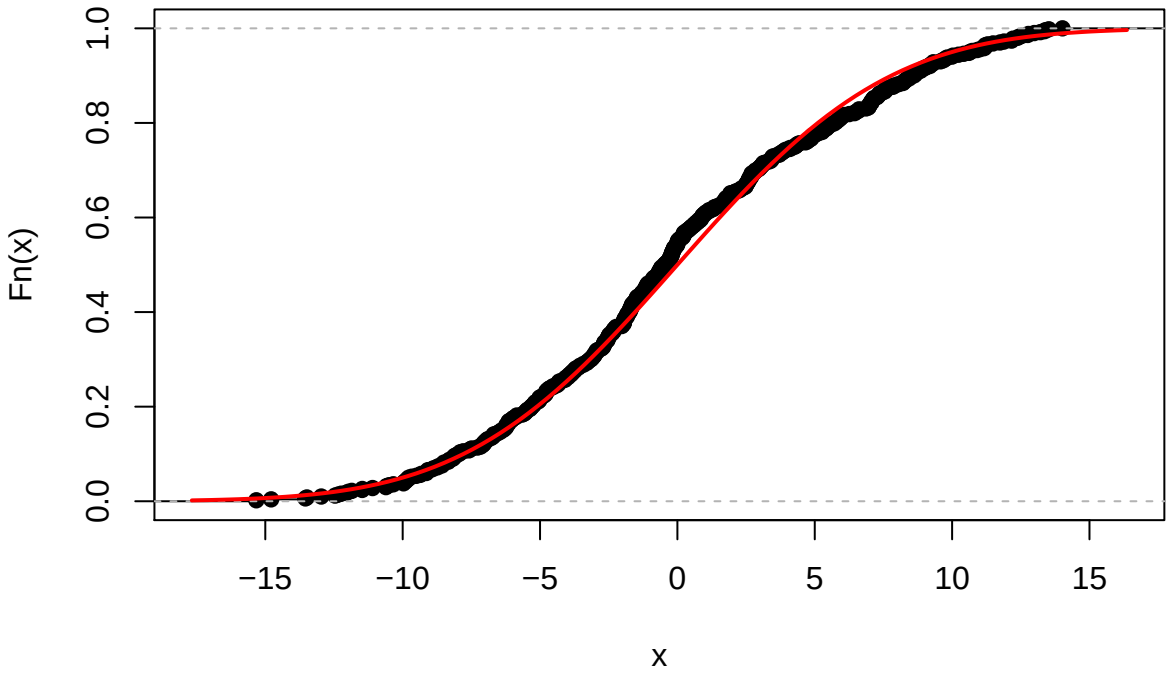
```
plot(density(mod3$residuals), main = "Wykres gęstości rozkładu reszt w porównaniu z rozkładem normalnym
curve(dnorm(x, 0, sd(mod3$residuals)), add = T, col = 2, lwd = 2)
```




```r
qqnorm(mod3$residuals, main = "Wykres z linią kwantylową")
qqline(mod3$residuals, col=2, lwd = 3)
```




```
plot(ecdf(mod3$residuals), main = "Dystrybuanta empiryczna w porównaniu z rozkładem normalnym")
curve(pnorm(x, 0, sd(mod3$residuals)), add = T, col = 2, lwd =2)
```




```
library(nortest)
ks.test(x = mod3$residuals, y = "pnorm", mean = 0, sd = sd(mod3$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  mod3$residuals
## D = 0.051812, p-value = 0.135
## alternative hypothesis: two-sided
```

```
lillie.test(mod3$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  mod3$residuals
## D = 0.051812, p-value = 0.002624
```

```
shapiro.test(mod3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod3$residuals
## W = 0.98969, p-value = 0.001356
```

```
ad.test(mod3$residuals)
```

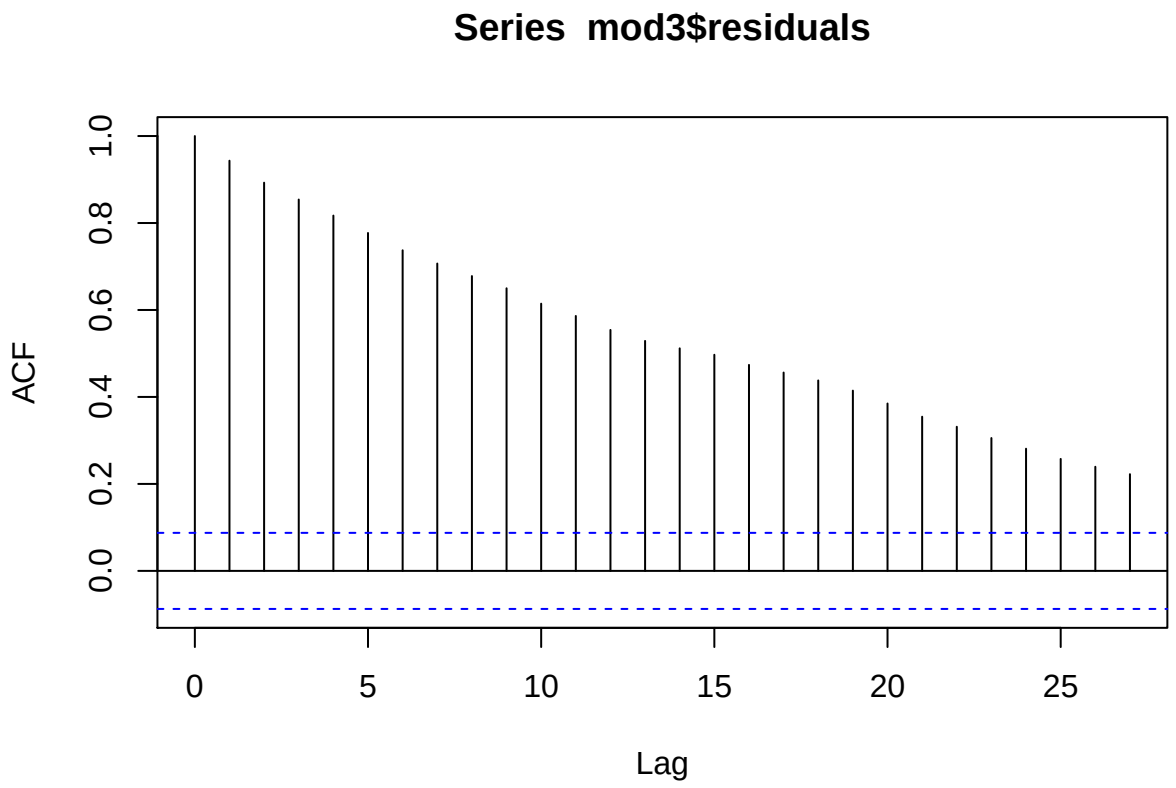
```
##
```

```
##  Anderson-Darling normality test
##
## data:  mod3$residuals
## A = 1.3485, p-value = 0.001691
```

Z testów Kołmogorowa-Lillieforsa, Shapiro-Wilka oraz Andersona-Darlinga wynika, że musimy odrzucić hipotezę o normalności rozkładu reszt (dla testu Kołmogorowa-Smirnova nie było podtsaw do odrzucenia, p-value około 0.14). Jeżeli chodzi o wykresy, to brak normalności najbardziej widać na wykresie gęstości. Na drugim wykresie (z linią kwantylową) reszty najbardziej odstają od rozkładu normalnego na początku i na końcu. Dystrybuanta empiryczna jest zbliżona do dystrybuanty rozkładu normalnego.

Autokorelacja

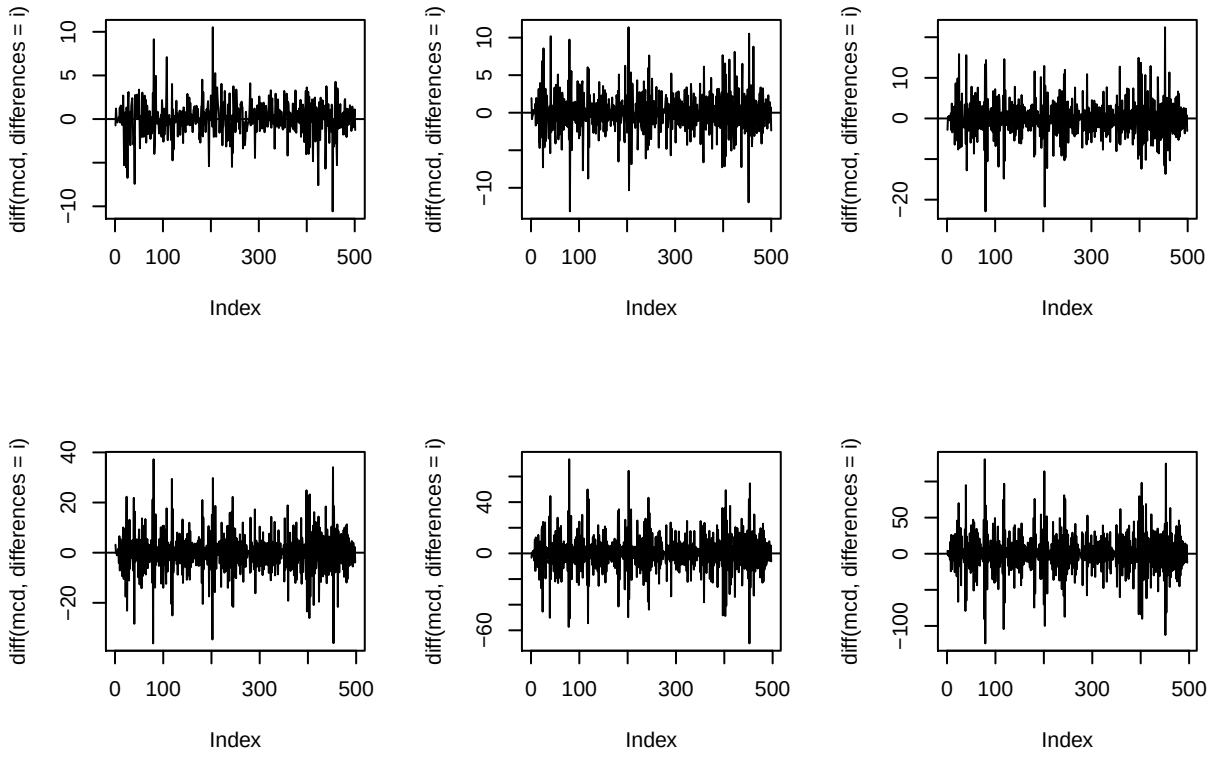
```
acf(mod3$residuals)
```



Dla opóźnień do rzędu 25 obserwacje nie mieszczą się w niebieskich przerywanych liniach - wnioskujemy, że pojawia się autokorelacja.

Metoda różnicowa

```
par(mfrow = c(2, 3))
for(i in 1:6){
plot(diff(mcd, differences = i), type = "l")
abline(h = 0)}
```




```
par(mfrow=c(1,1))
```

Stacjonarność

```
adf.test(mcd) #niest
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  mcd
## Dickey-Fuller = -2.2351, Lag order = 7, p-value = 0.4788
## alternative hypothesis: stationary
```

```
kpss.test(mcd) #niest
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  mcd
## KPSS Level = 7.0755, Truncation lag parameter = 5, p-value = 0.01
```

```
kpss.test(mcd, null = "Trend") #niest
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  mcd
```

```
## KPSS Trend = 0.66563, Truncation lag parameter = 5, p-value = 0.01
```

```
adf.test(diff(mcd, differences = 1)) #stacj
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(mcd, differences = 1)
## Dickey-Fuller = -8.574, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(mcd, differences = 1)) #stacj
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(mcd, differences = 1)
## KPSS Level = 0.12182, Truncation lag parameter = 5, p-value = 0.1
```

```
kpss.test(diff(mcd, differences = 1), null = "Trend") #stacj
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  diff(mcd, differences = 1)
## KPSS Trend = 0.1159, Truncation lag parameter = 5, p-value = 0.1
```

Szereg jest niestacjonarny, i niestacjonarny względem trendu. Po zróżnicowaniu 1 raz, jest zarówno stacjonarny, jak i TS (Trend Stationary).

```
library(forecast)
auto.arima(mcd)
```

```
## Series: mcd
## ARIMA(0,1,0)
##
## sigma^2 estimated as 4.176:  log likelihood=-1068.94
## AIC=2139.87   AICc=2139.88   BIC=2144.09
```

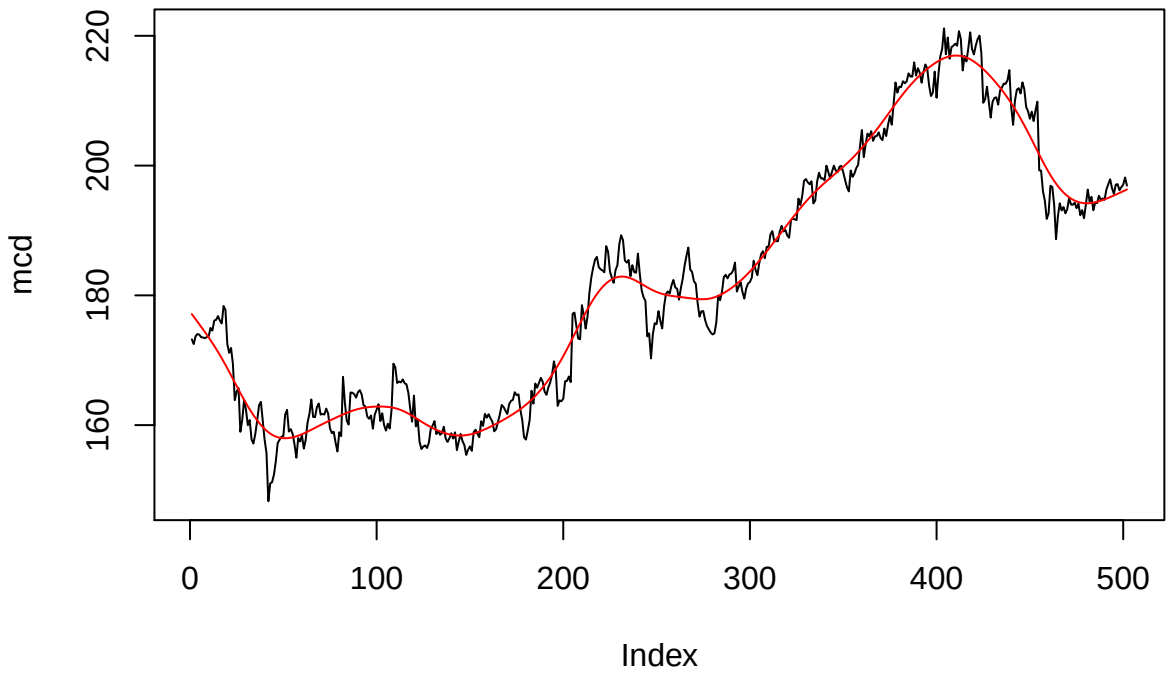
Inne rzeczy

Trend

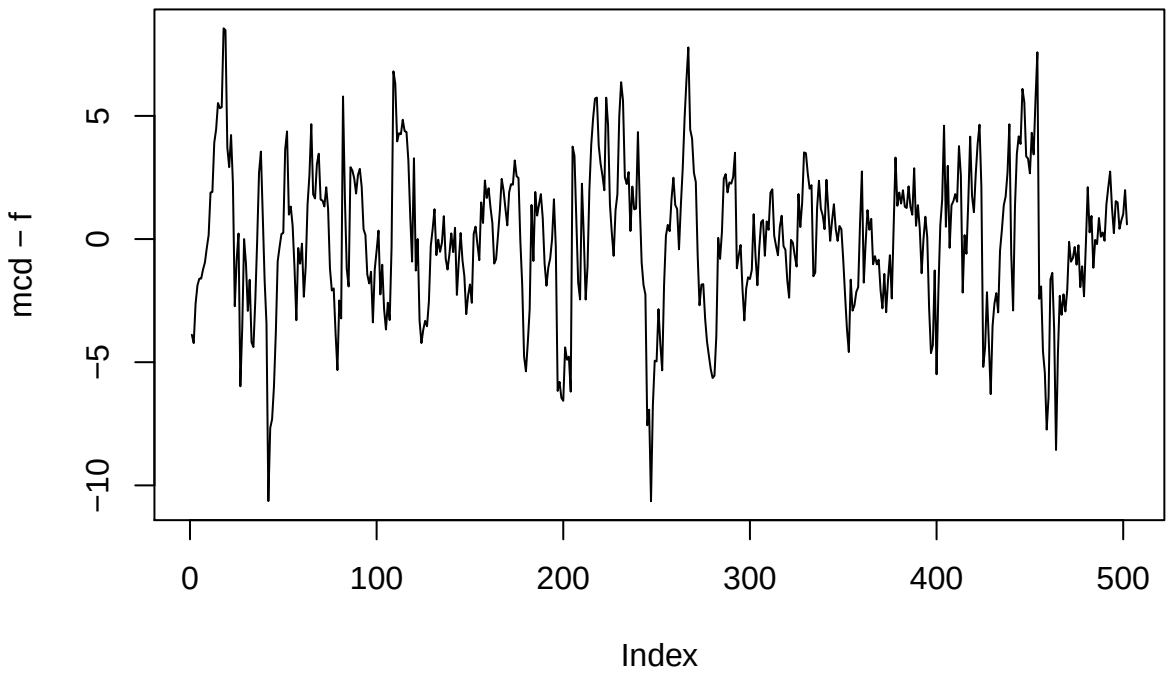
W środowisku R dostępne są także funkcje dotyczące filtrowania szeregów czasowych. Jest to takie przekształcenie danych które doprowadza do oczyszczenia szeregu czasowego z wahań periodycznych. W środowisku R dostępnych jest kilka takich filtrów. Jeden z bardzie popularnych to filtr Hodrick-Prescotta zaimplementowany w pakiecie FRAPO::trdhp. Stosując filtr HP należy pamiętać o odpowiednim doborze parametru λ. Hodrick oraz Prescott zalecają, aby wartość współczynnika λ była równa 400, 1600 i 14400 odpowiednio dla danych rocznych, kwartalnych i miesięcznych.

(P. Biecek Na przełaj przez Data Mining)

```
library(FRAPO)
f <- FRAPO::trdhp(mcd, 14400)
plot(mcd, type = "l")
lines(f, col = 2)
```

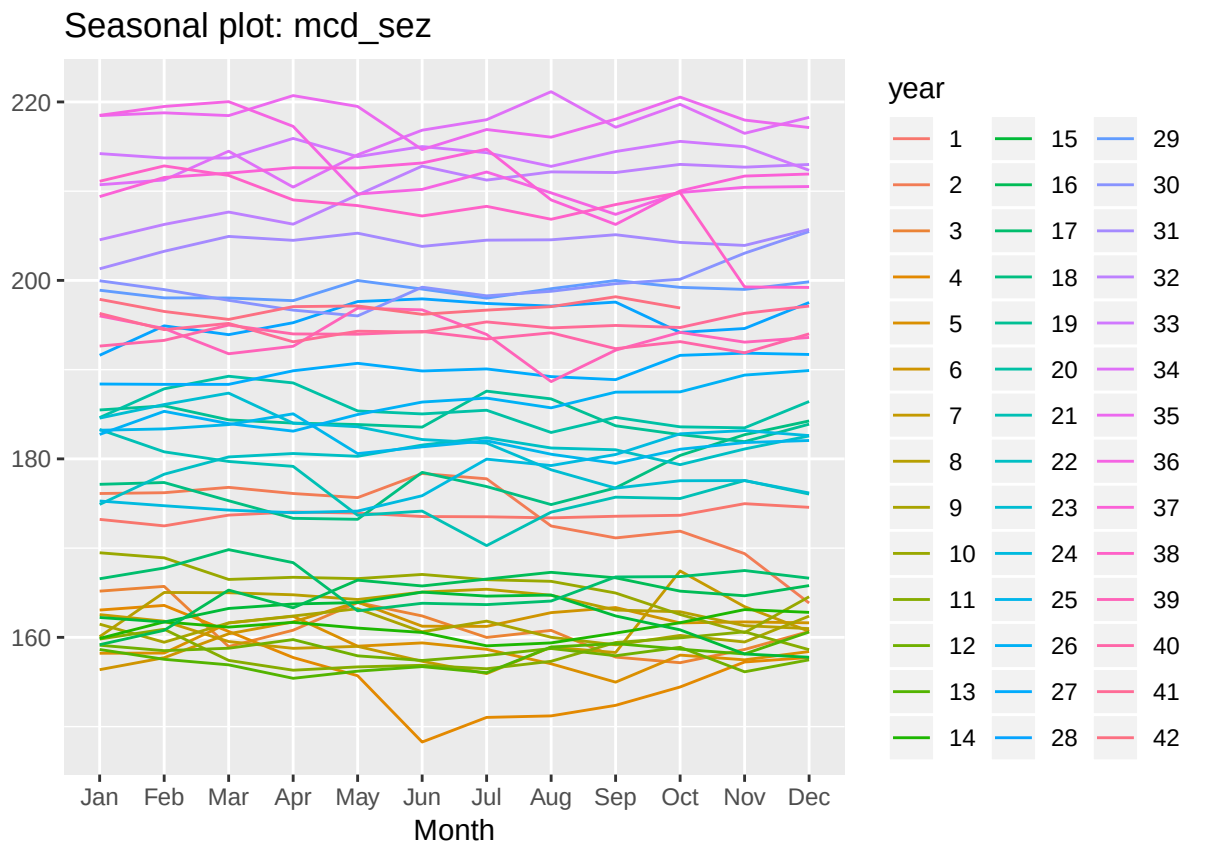



```
plot(mcd - f, type = "l")
```

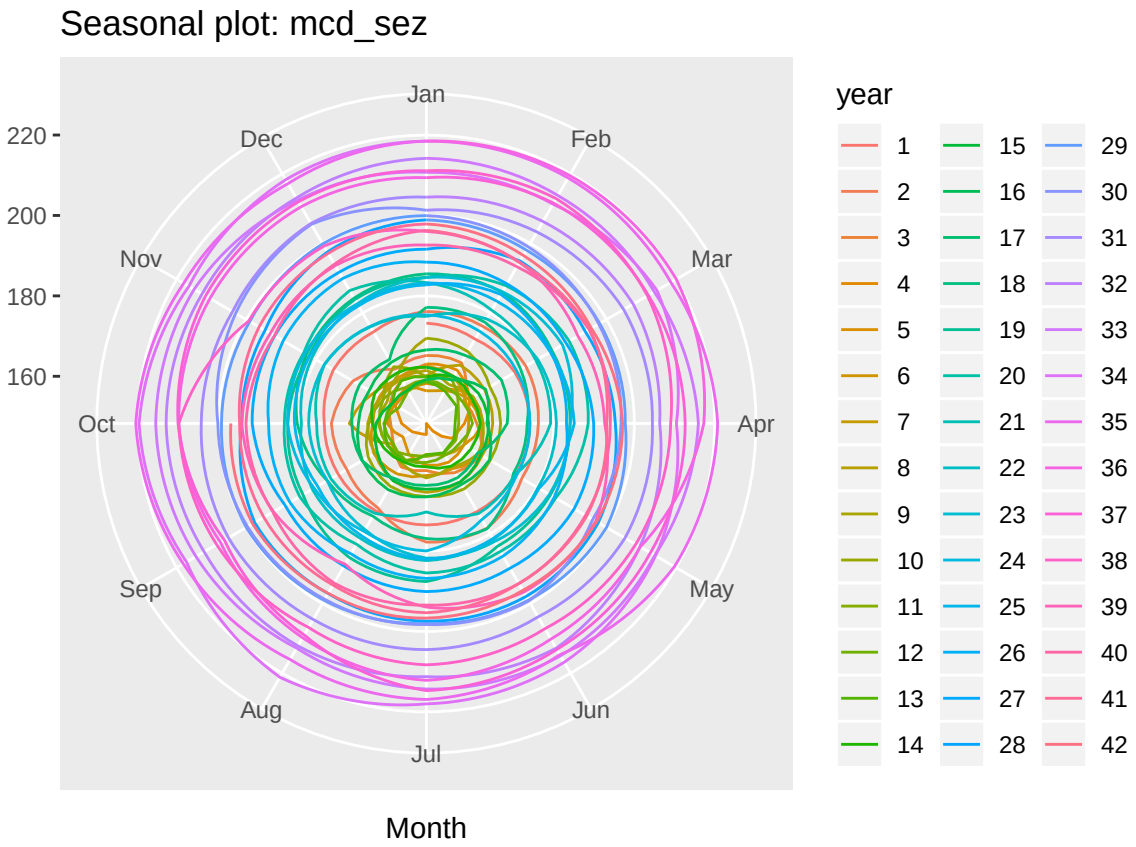


Sezonowość

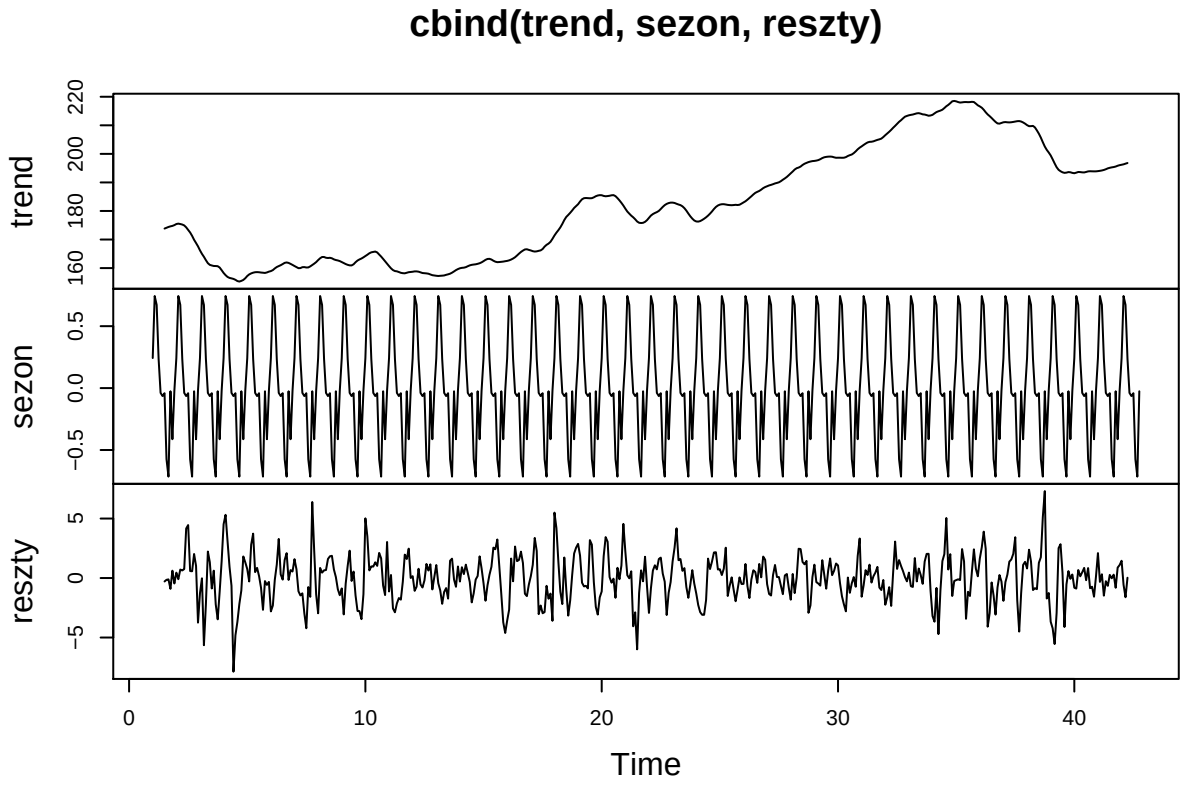
```
mcd_sez <- ts(mcd, frequency = 12)
mcd_dek <- decompose(mcd_sez)
forecast:: ggseasonplot(mcd_sez)
```



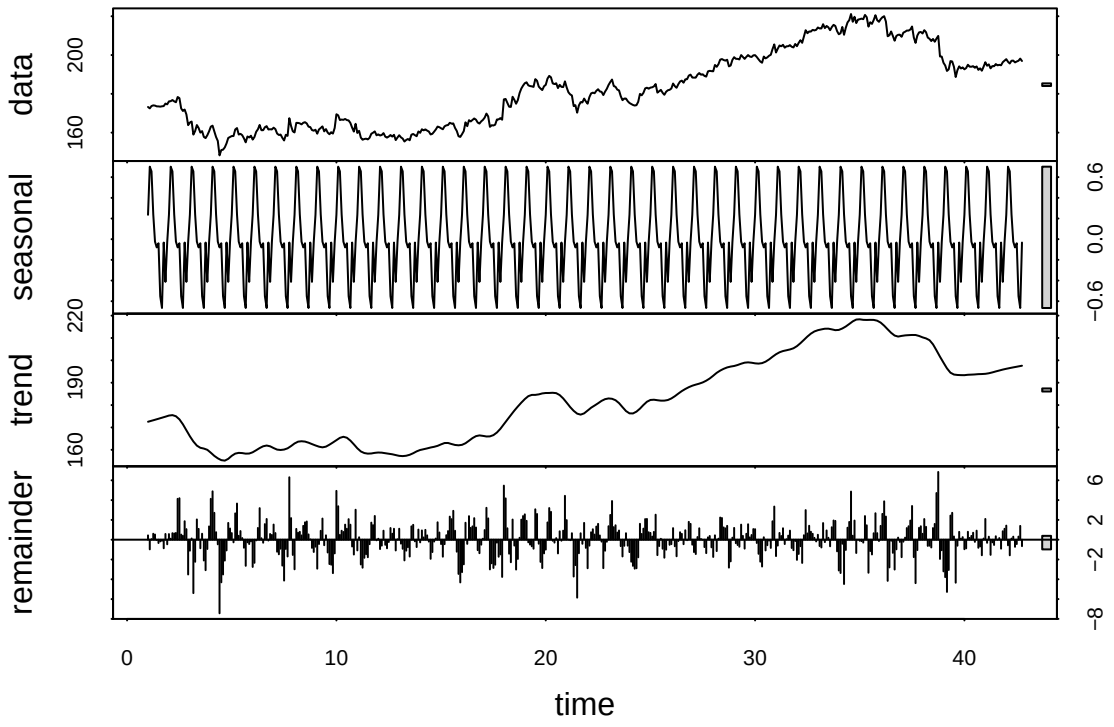

```
forecast::ggseasonplot(mcd_sez, polar = T)
```




```
trend <- mcd_dek$trend
sezon <- mcd_dek$seasonal
reszty <- mcd_dek$random
plot(cbind(trend, sezon, reszty))
```



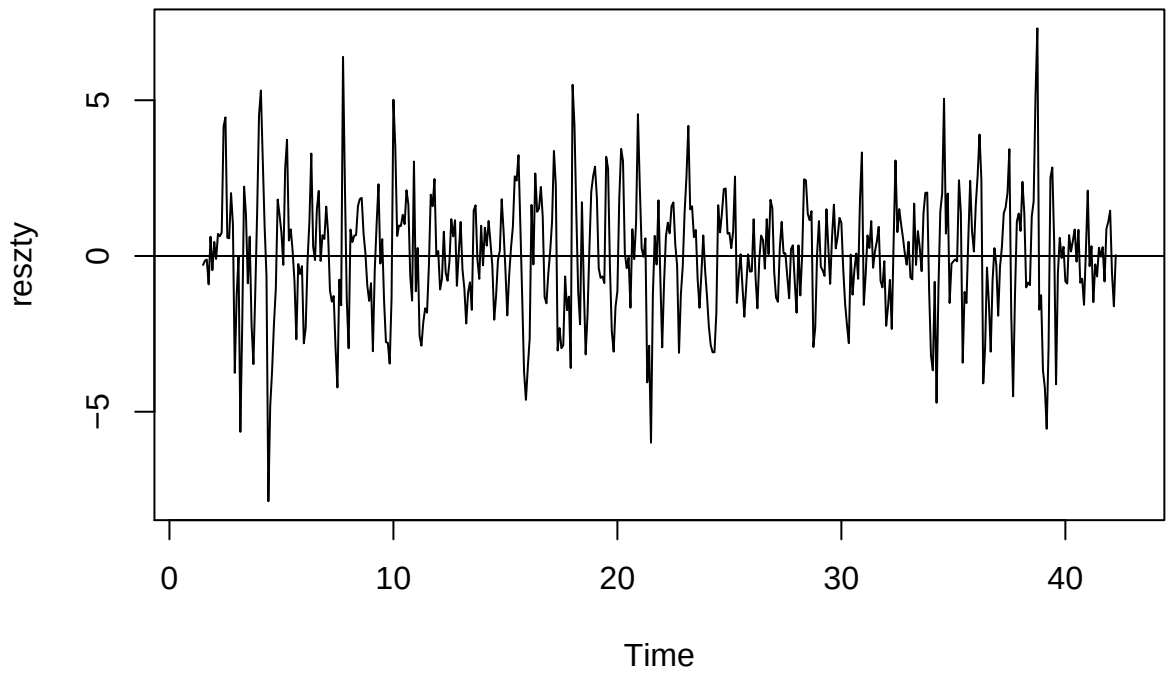

```
plot(stl(mcd_sez, "periodic"))
```



Chyba McDonalds jest jakiś super sezonowy, bo wszystko idealnie wygląda na wykresach... ;p

To skoro ten model wygląda najlepiej, to chyba dla jego reszt trzeba by testować... ?

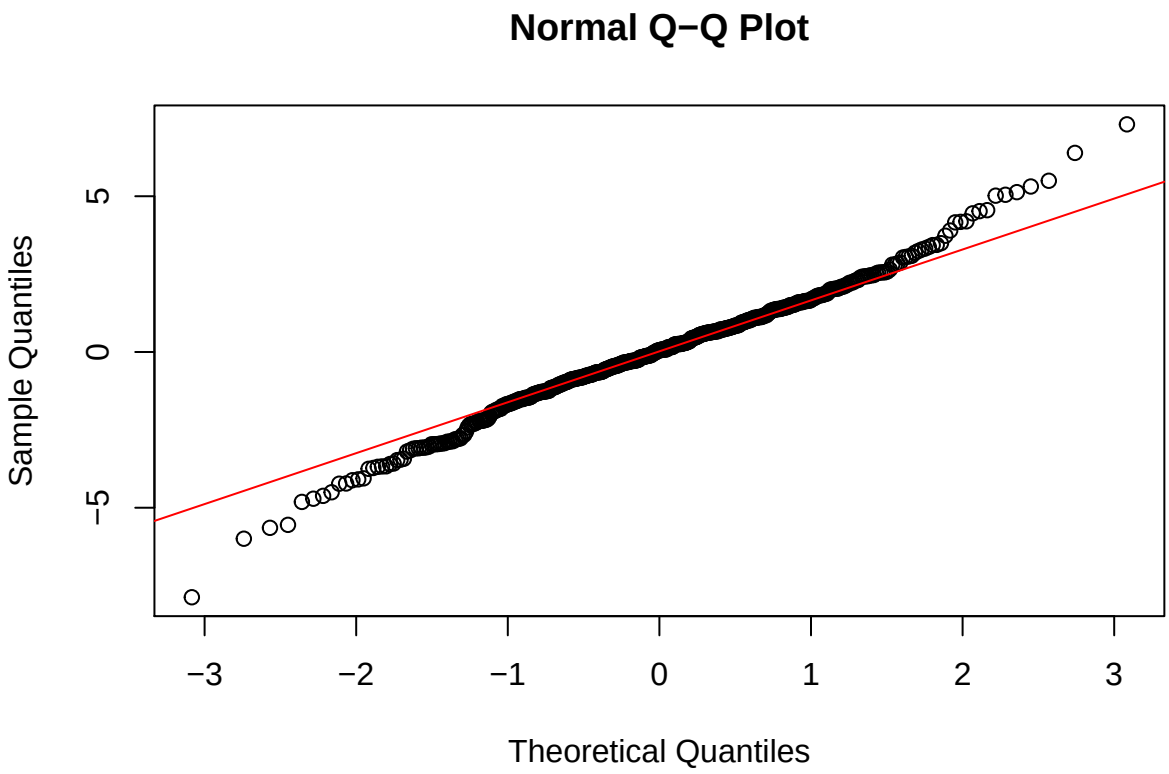
```
plot(reszty)
abline(h=0)
```




```
shapiro.test(reszty) #ohohoo malutkie p value
```

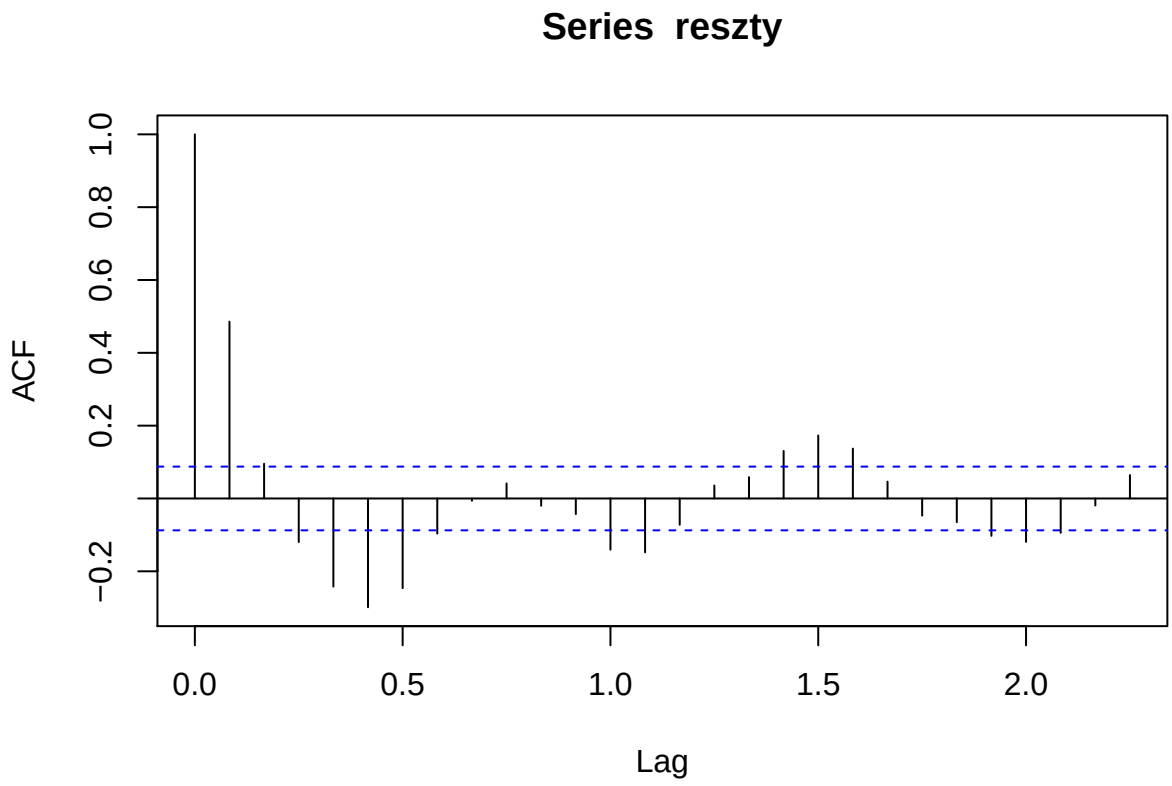
```
##
##  Shapiro-Wilk normality test
##
## data:  reszty
## W = 0.98851, p-value = 0.0006816
```

```
library(lmtest)
qqnorm(reszty)
qqline(reszty, col = 2)
```




```
#nie ma normalności!!
```

```
acf(reszty, na.action = na.pass) #dla dalszych z grubsza się mieszczą w pasku
```



Arima

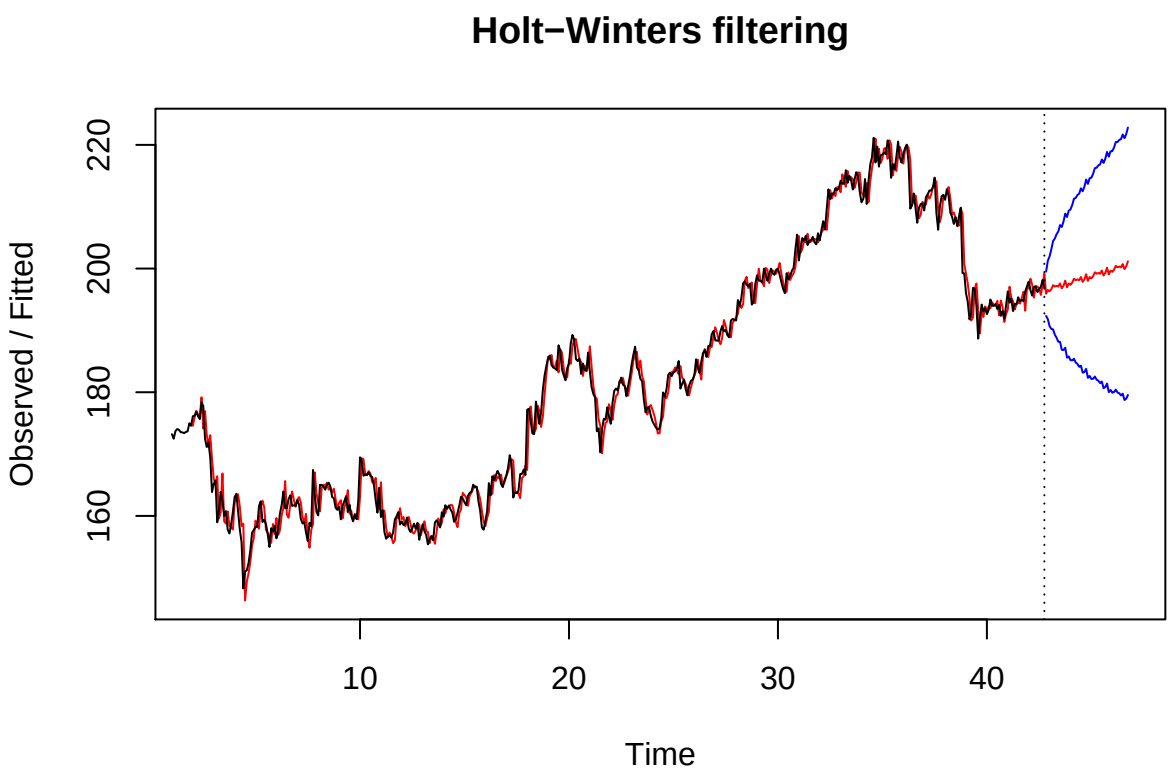
Dobra, ogólnie tutaj nie wiem jak to zrobić, czy nasz w końcu ma tę sezonowość? Bo jeśli ma, to SARIMA podobno, a jeśli nie to ARIMA. Plus, ten szereg chyba jest niestacjonarny, nie? To też przecież musi być stacjonarny i ja nie wiem już nic w końcu :((()))

```
Arima(reszty, order = c(12,0,2)) #tu AIC jakoś 1842, jak sie leci z p do gory, to coraz lepiej, ale tez
```

```
## Series: reszty
## ARIMA(12,0,2) with non-zero mean
##
## Coefficients:
##          ar1     ar2      ar3     ar4      ar5      ar6     ar7     ar8
##       0.5512  0.5878  -0.4389  0.0194  -0.0822  -0.0345  0.1207  0.0038
## s.e.  0.1224  0.1712   0.0915  0.0622   0.0615   0.0616  0.0615  0.0626
##          ar9     ar10    ar11     ar12      ma1      ma2     mean
##       0.0114  -0.1060  0.0122  -0.0057  -0.1144  -0.8486  -0.0016
## s.e.  0.0620   0.0616  0.0684   0.0566   0.1142   0.1163   0.0081
##
## sigma^2 estimated as 2.418:  log likelihood=-905.29
## AIC=1842.59   AICc=1843.74   BIC=1909.7
```

Holt - Winters

```
mcd_hw <- HoltWinters(mcd_sez, seasonal = "additive")
pred <- predict(mcd_hw, n.ahead = 4*12, prediction.interval = T, level = 0.9)
plot(mcd_hw, pred)
```



Predykcja za pomocą metody Holta-Wintersa. Testując dla różnej liczby okresów naprzód, widzimy, że przedział ufności drastycznie się rozszerza im większe `n.ahead`.

Starbucks

Opis firmy

Starbucks Corporation – największa na świecie sieć kawiarni. Została założona 30 marca 1971 w Seattle w stanie Waszyngton.

Kawa ze Starbucks jest znana z tego, że pojawiła się jako błąd w jednym z odcinków Gry o tron.

Kawiarnia jest znana z tego, że każdy kubek jest podpisany imieniem zamawiającego. Urocze!

Figure 2: Kawa ze Starbucks

Figure 3: Starbucks jest tak popularny, że piją go nawet w fantasy, które dzieje się w średniowieczu!

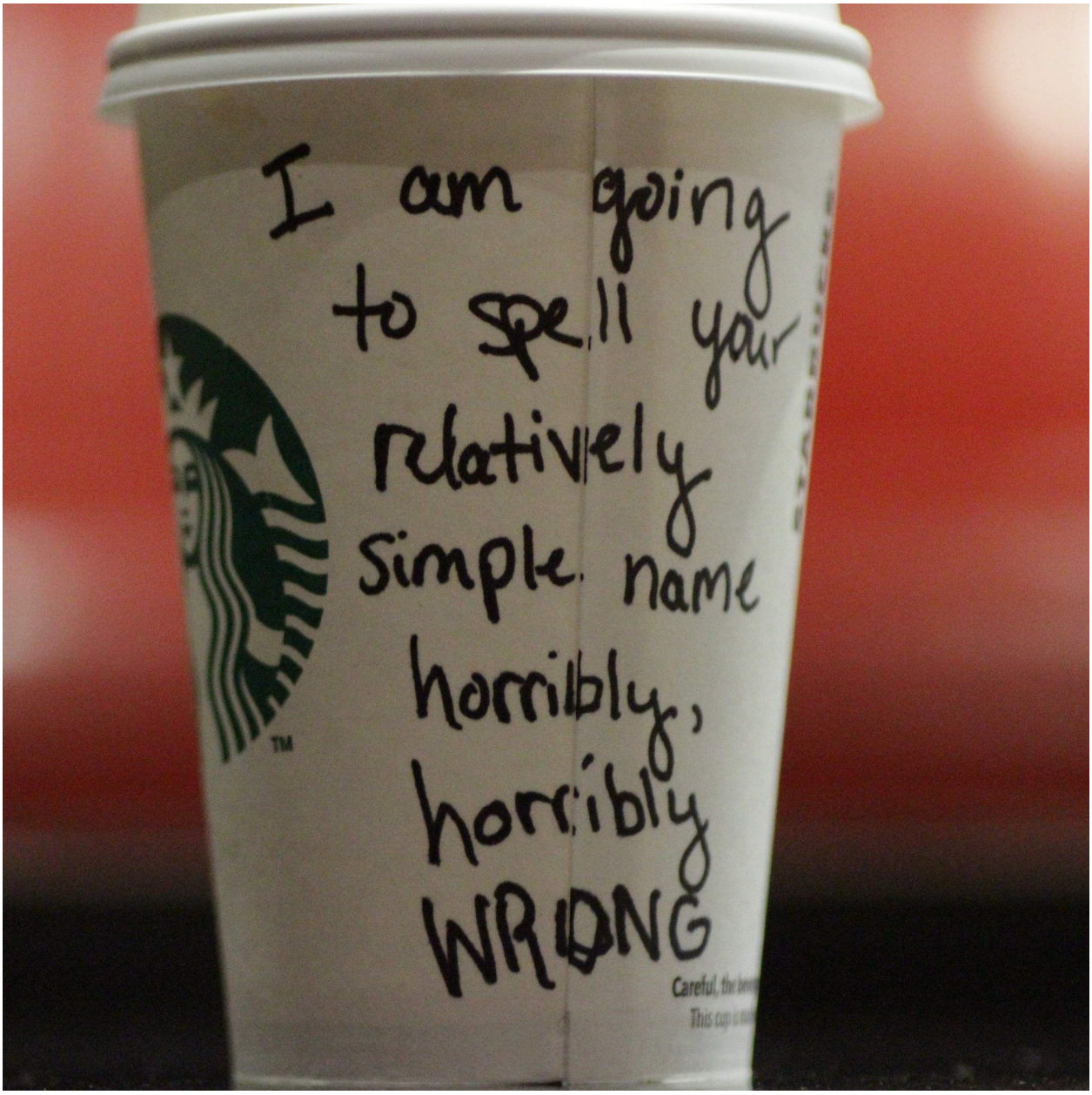


Figure 4: hehe

Wczytanie danych

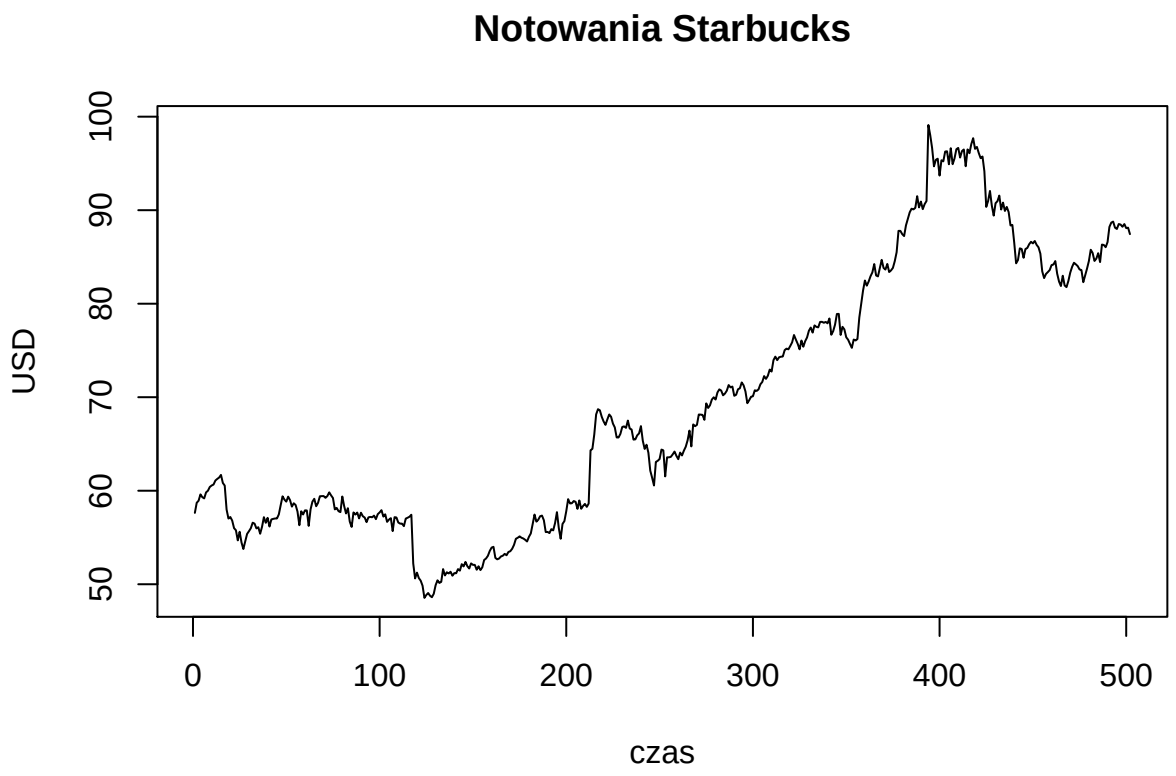
```
sbux<- get.hist.quote(instrument = "SBUX", provider = "yahoo",
                          quote = "Close", start = "2018-01-01", end = "2019-12-31")
```

```
## time series starts 2018-01-02
## time series ends   2019-12-30
```

```
sbux<- as.numeric(sbux)
```

Rysunek

```
plot(sbux, type = "l", xlab = "czas", ylab = "USD", main = "Notowania Starbucks")
```



Widać trend rosnący.

Dopasowanie wielomianu

```
t <- 1:length(sbux)
```

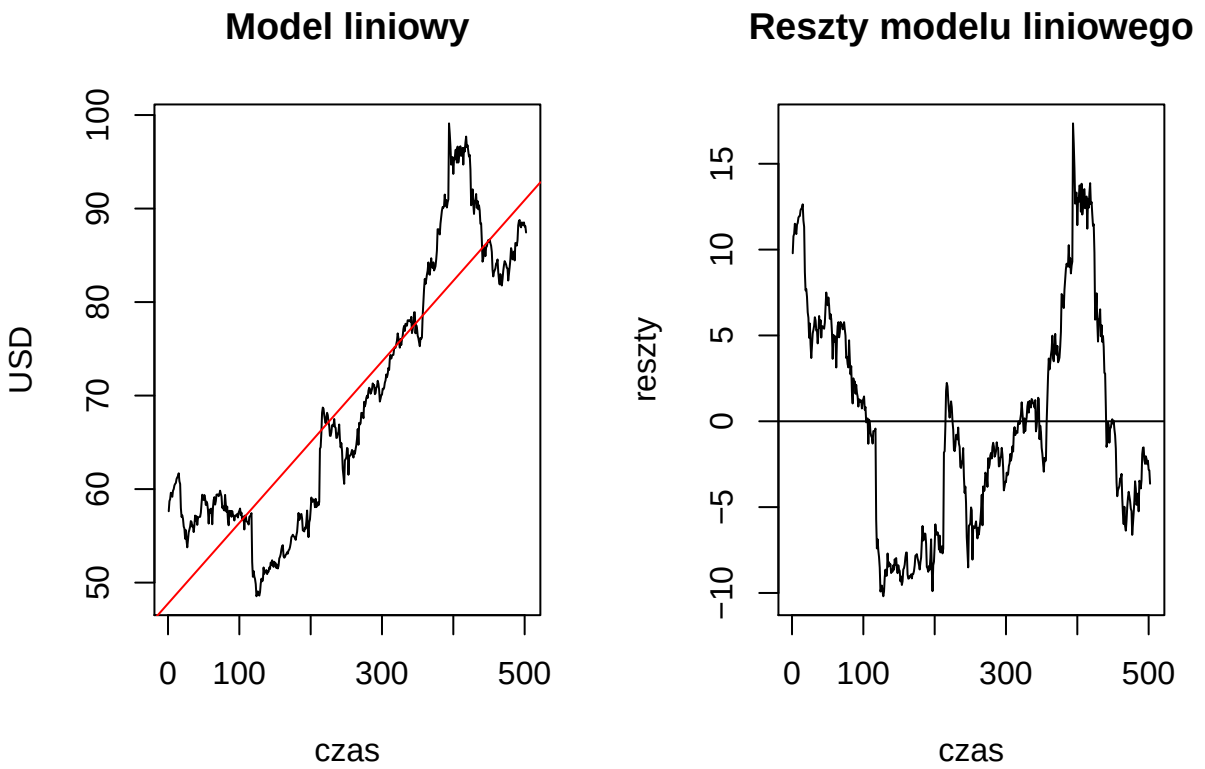
Model liniowy

```
mod1 <- lm(sbux~t)
summary(mod1)
```

```
##
## Call:
## lm(formula = sbux ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.1924  -5.1586  -0.6162   4.8674  17.3536
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47.756892   0.569068   83.92   <2e-16 ***
## t            0.086293   0.001961   44.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.366 on 500 degrees of freedom
## Multiple R-squared:  0.7949, Adjusted R-squared:  0.7944
## F-statistic:  1937 on 1 and 500 DF,  p-value: < 2.2e-16
```

Wszystkie współczynniki są istotne statystycznie. R^2 wynosi około 79%.

```
par(mfrow = c(1, 2))
plot(sbux, type = "l", main = "Model liniowy", xlab = "czas", ylab = "USD")
abline(mod1, col = "red")
plot(mod1$residuals, type = "l", main = "Reszty modelu liniowego", xlab = "czas", ylab = "reszty")
abline(h=0)
```



```r
par(mfrow = c(1, 1))
```

Na wykresie widać, że reszty mają rozrzut mniej więcej od -10 do 15. Model nie jest zbyt dokładny - na początku przeszacowuje wartości, potem zdecydowanie niedoszacowuje, na koniec znowu przeszacowuje.

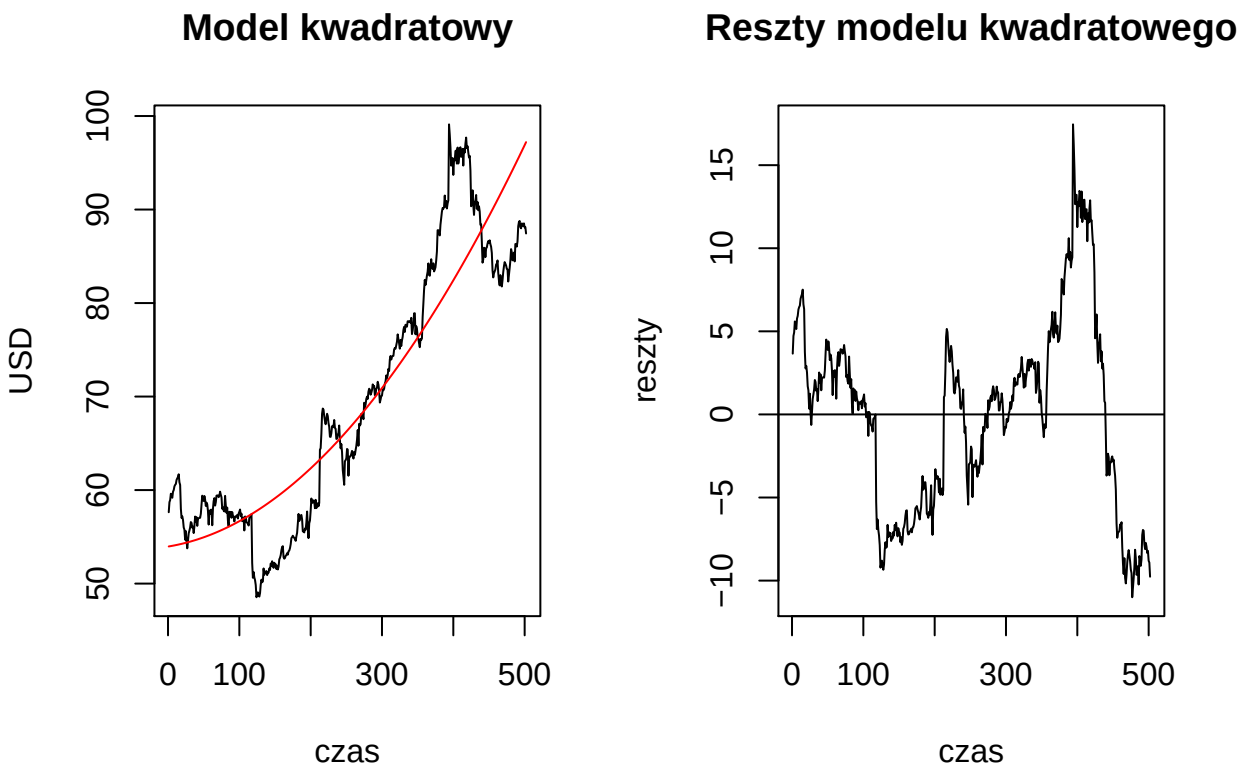
Model kwadratowy

```r
mod2 <- lm(sbux~t+I(t^2))
summary(mod2)
```

```
##
## Call:
## lm(formula = sbux ~ t + I(t^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0010  -4.7965   0.7868   3.2228  17.4555
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.396e+01  7.717e-01  69.924   <2e-16 ***
## t           1.245e-02  7.085e-03   1.757   0.0795 .
## I(t^2)      1.468e-04  1.364e-05  10.763   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.74 on 499 degrees of freedom
## Multiple R-squared:  0.8335, Adjusted R-squared:  0.8328
## F-statistic:  1249 on 2 and 499 DF,  p-value: < 2.2e-16
```

Współczynnik przy t jest nieistotny statystycznie (p-value około 0.08, więc decyzja niejednoznaczna), ale R^2 poprawiło się - wynosi teraz około 83%.

```r
par(mfrow = c(1, 2))
plot(sbux, type = "l", main = "Model kwadratowy", xlab = "czas", ylab = "USD")
lines(t, mod2$fitted.values, col = "red")
plot(mod2$residuals, type = "l", main = "Reszty modelu kwadratowego", xlab = "czas", ylab = "reszty")
abline(h = 0)
```




```
par(mfrow = c(1, 1))
```

Wykres reszt jest bardzo podobny jak w przypadku modelu liniowego.

Model sześcienny

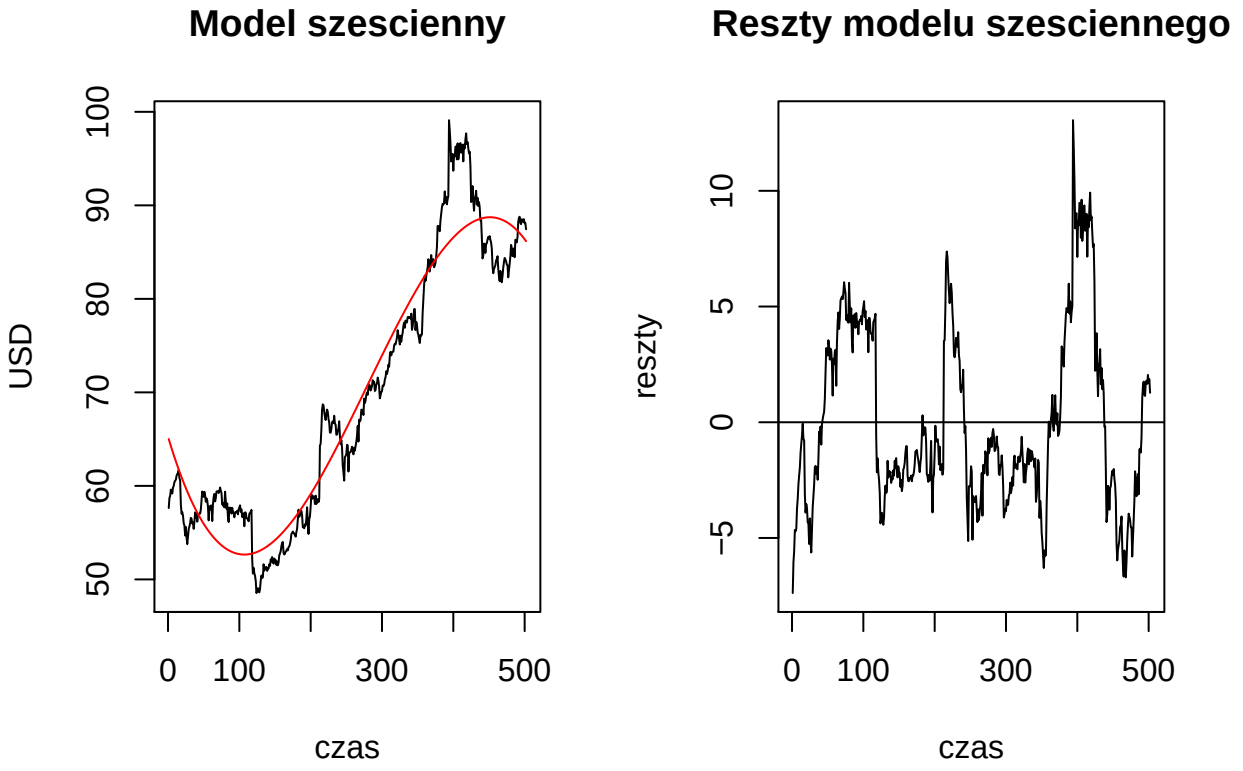
```
mod3 <- lm(sbux~t+I(t^2)+I(t^3))
summary(mod3)
```

```
##
## Call:
## lm(formula = sbux ~ t + I(t^2) + I(t^3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.380 -2.608 -1.257  3.041 13.053
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.526e+01  6.978e-01   93.53   <2e-16 ***
## t           -2.559e-01  1.200e-02  -21.32   <2e-16 ***
## I(t^2)       1.479e-03  5.542e-05   26.69   <2e-16 ***
## I(t^3)      -1.766e-06  7.243e-08  -24.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.879 on 498 degrees of freedom
## Multiple R-squared:  0.9241, Adjusted R-squared:  0.9237
## F-statistic:  2021 on 3 and 498 DF,  p-value: < 2.2e-16
```

Wszystkie współczynniki są istotne statystycznie, a R^2 znów wzrosło - wynosi około 92% (znaczna poprawa).

```
par(mfrow = c(1, 2))
plot(sbux, type = "l", main = "Model sześcienny", xlab = "czas", ylab = "USD")
lines(t, mod3$fitted.values, col = "red")
plot(mod3$residuals, type = "l", main = "Reszty modelu sześciennego", xlab = "czas", ylab = "reszty")
abline(h= 0)
```

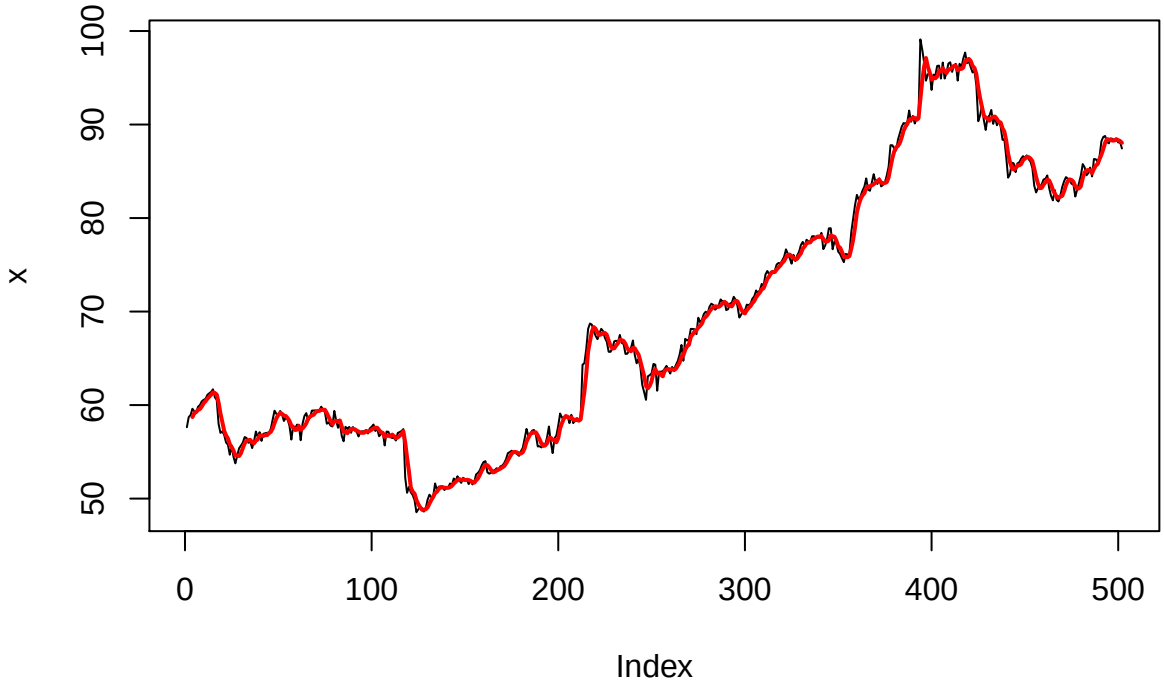



```
par(mfrow=c(1,1))
```

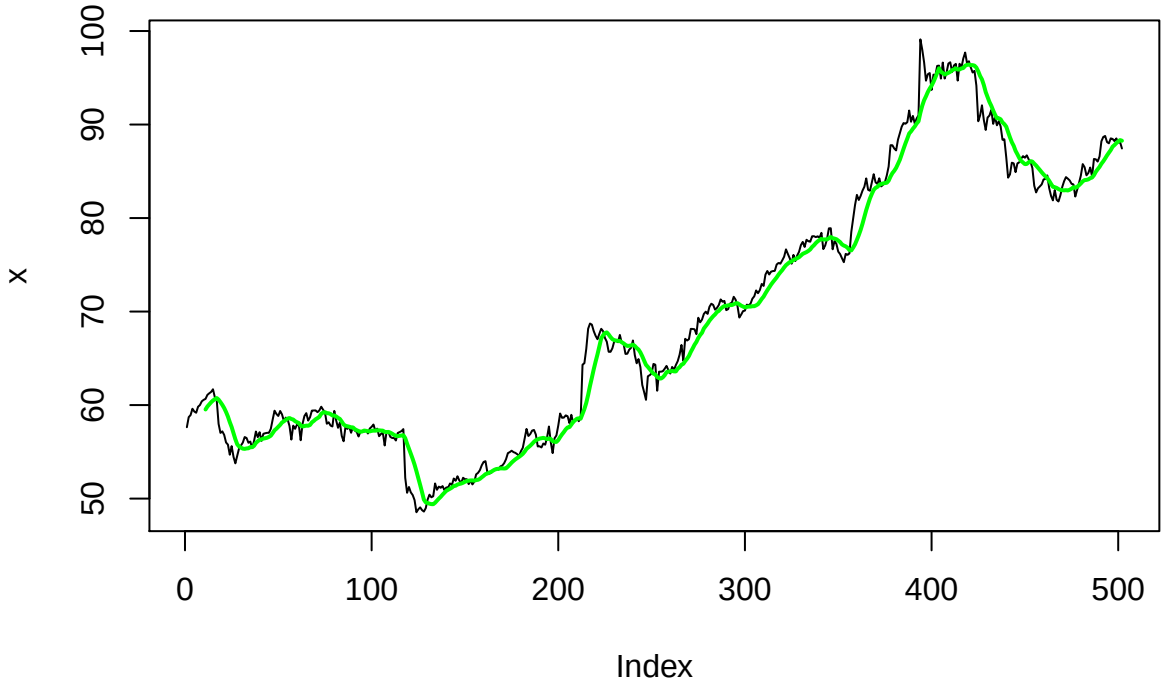
Reszty modelu sześciennego mają mniejszy rozrzut niż w poprzednich przypadkach (od około -5 do 10).

Ruchoma średnia

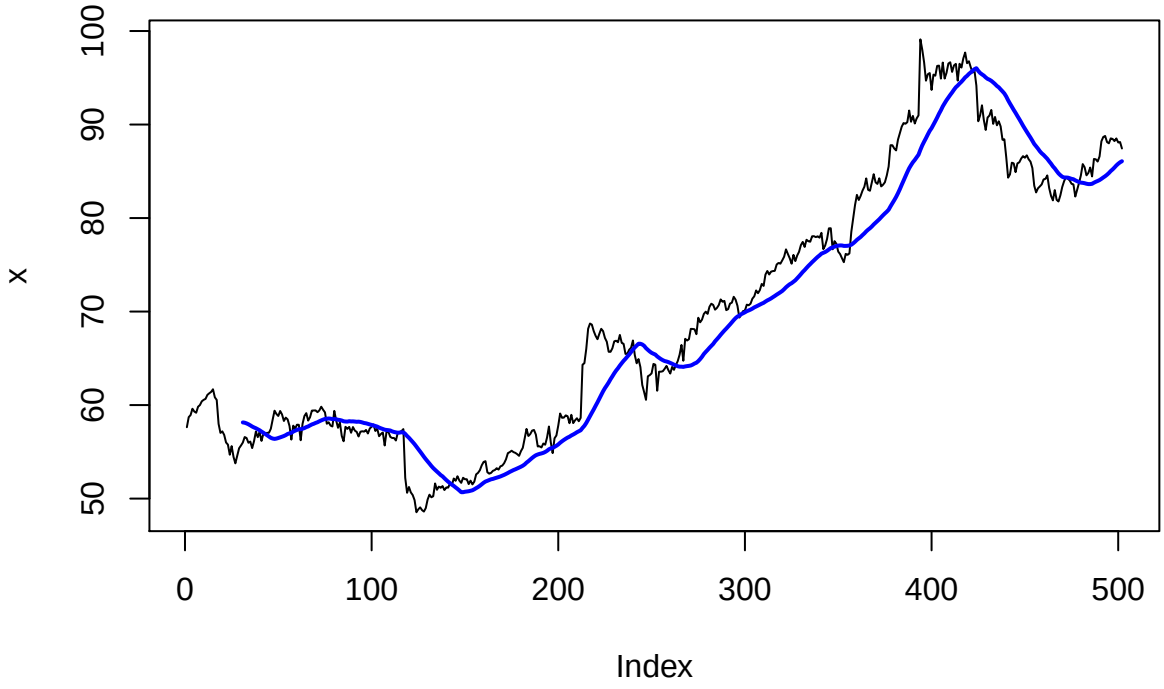
```
ruchoma(sbux, 3, "red")
```




```
ruchoma(sbux, 10, "green")
```

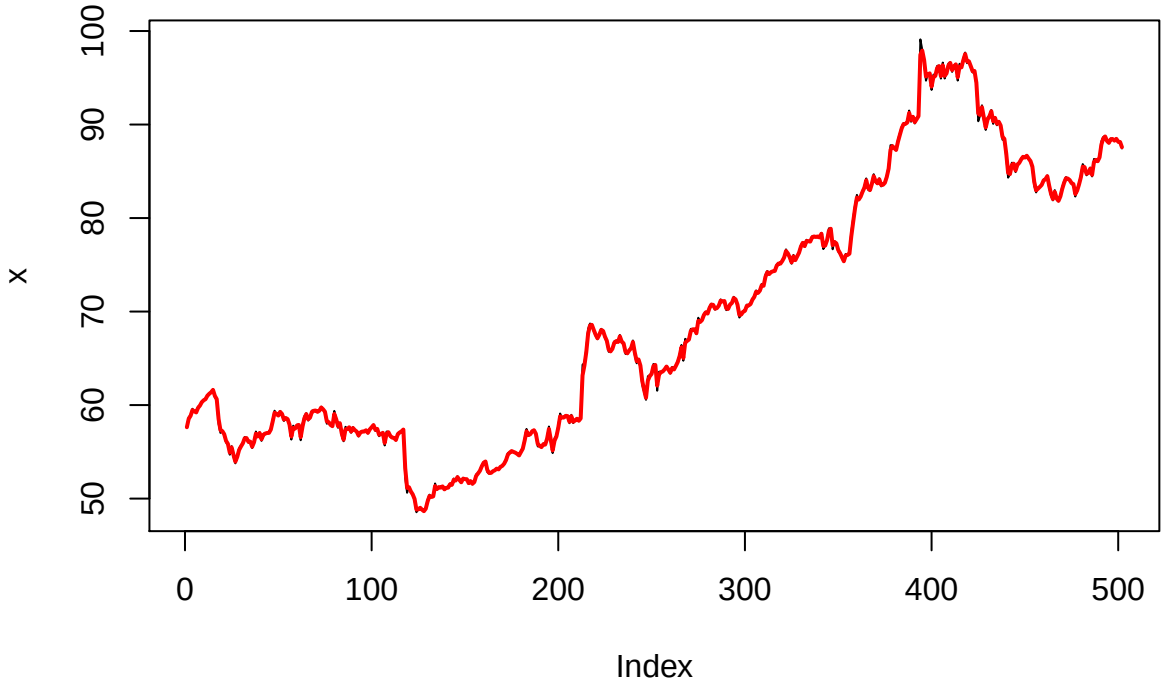



```
ruchoma(sbux, 30, "blue")
```

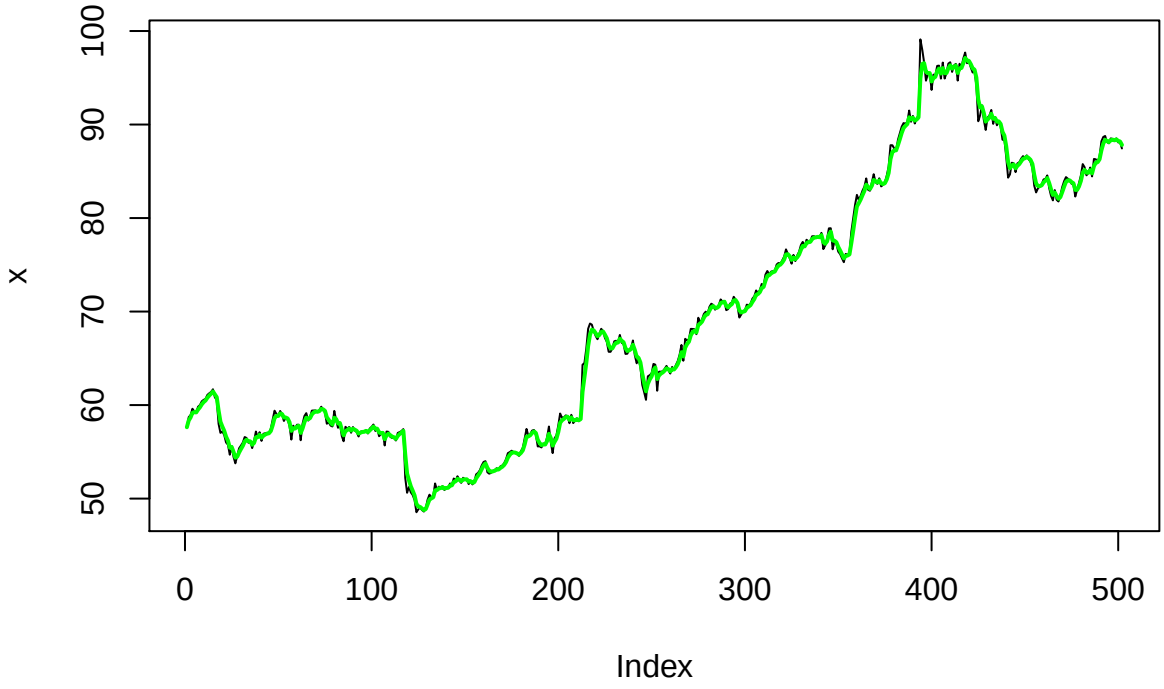


Metoda wykładniczych wag ruchomej średniej

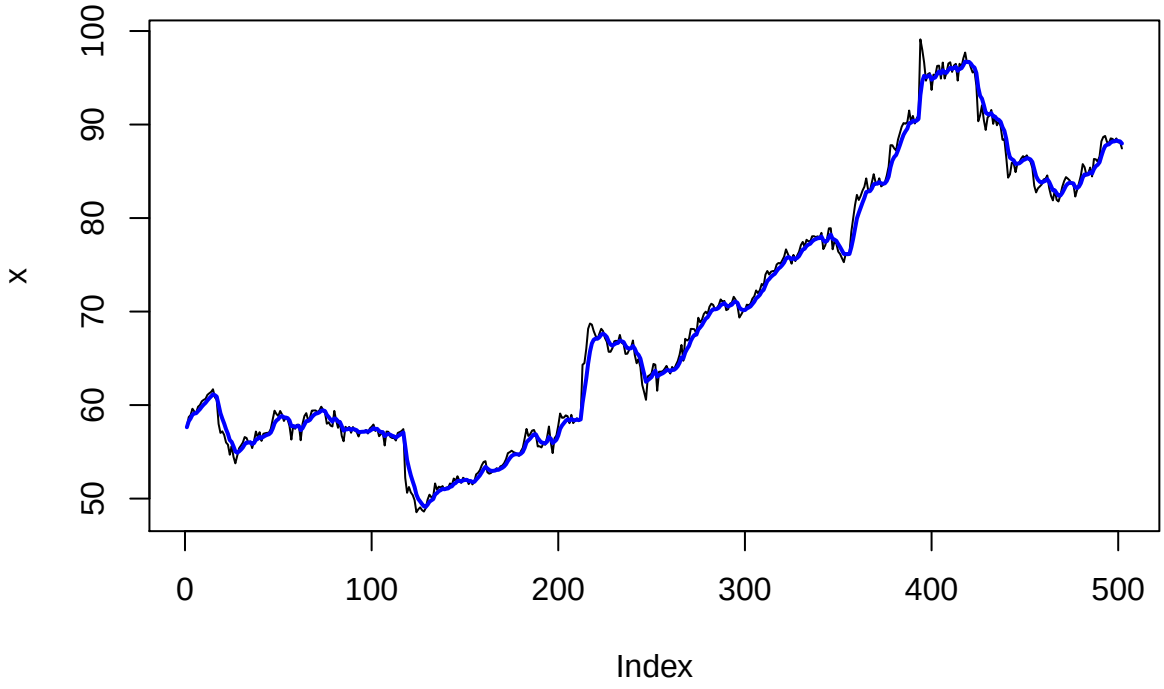
```
wykladnicza(sbux, 0.2, "red")
```



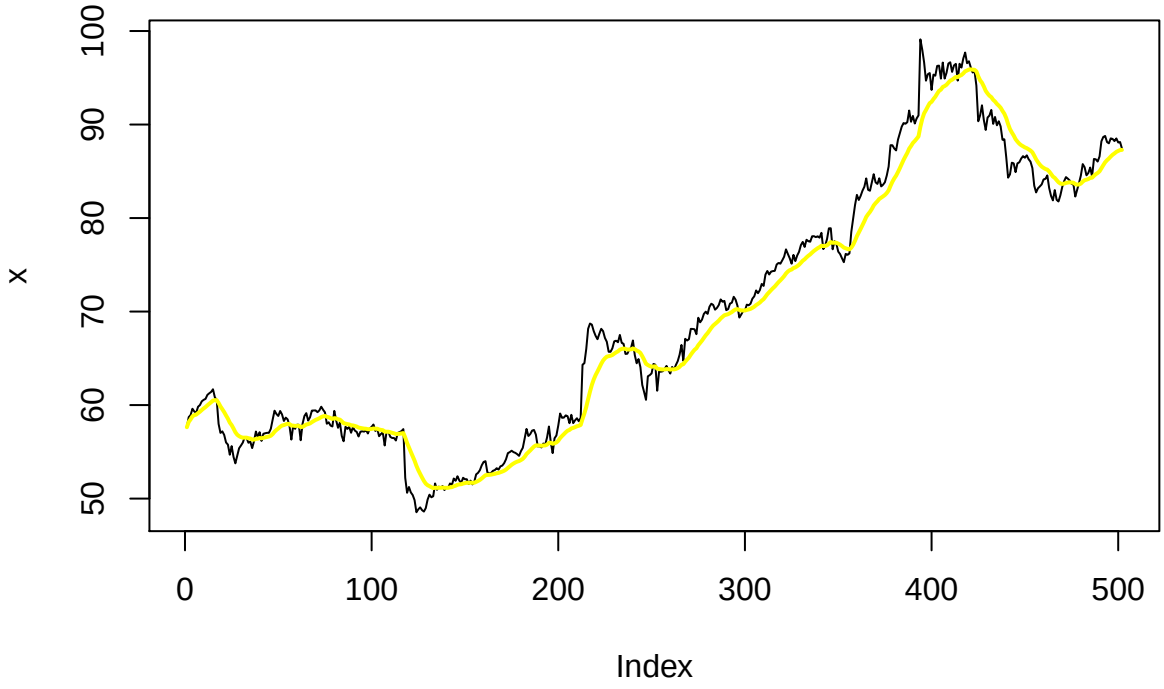

```
wykladnicza(sbux, 0.5, "green")
```




```
wykladnicza(sbux, 0.7, "blue")
```

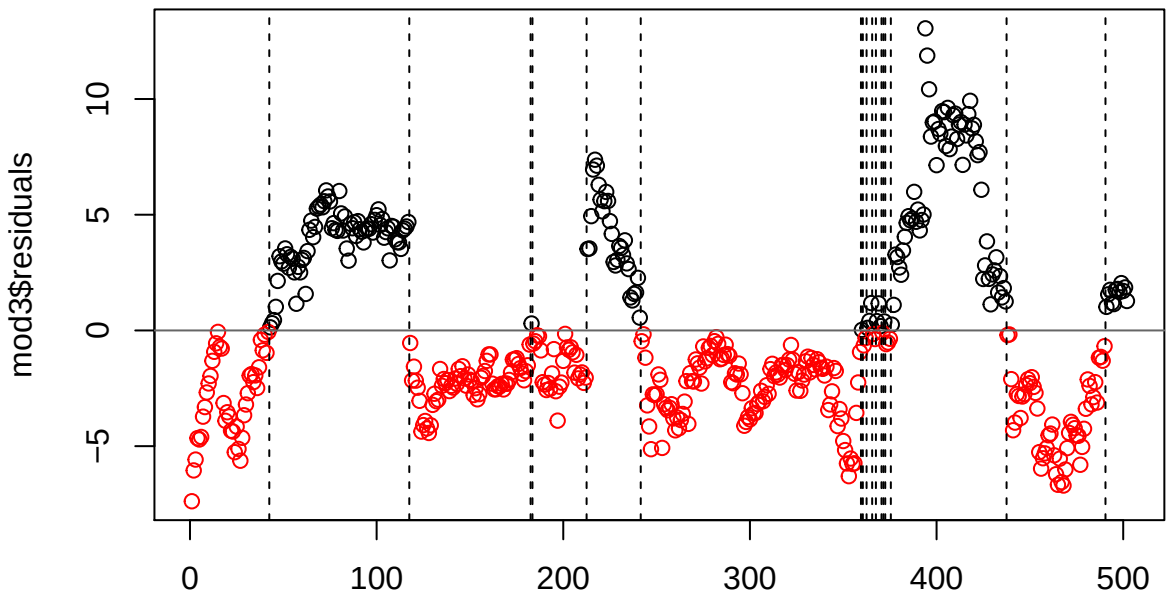



```
wykladnicza(sbux, 0.9, "yellow")
```



Testy na resztach modelu sześciennego

```
runs.test(mod3$residuals, threshold = 0, plot = T)
```

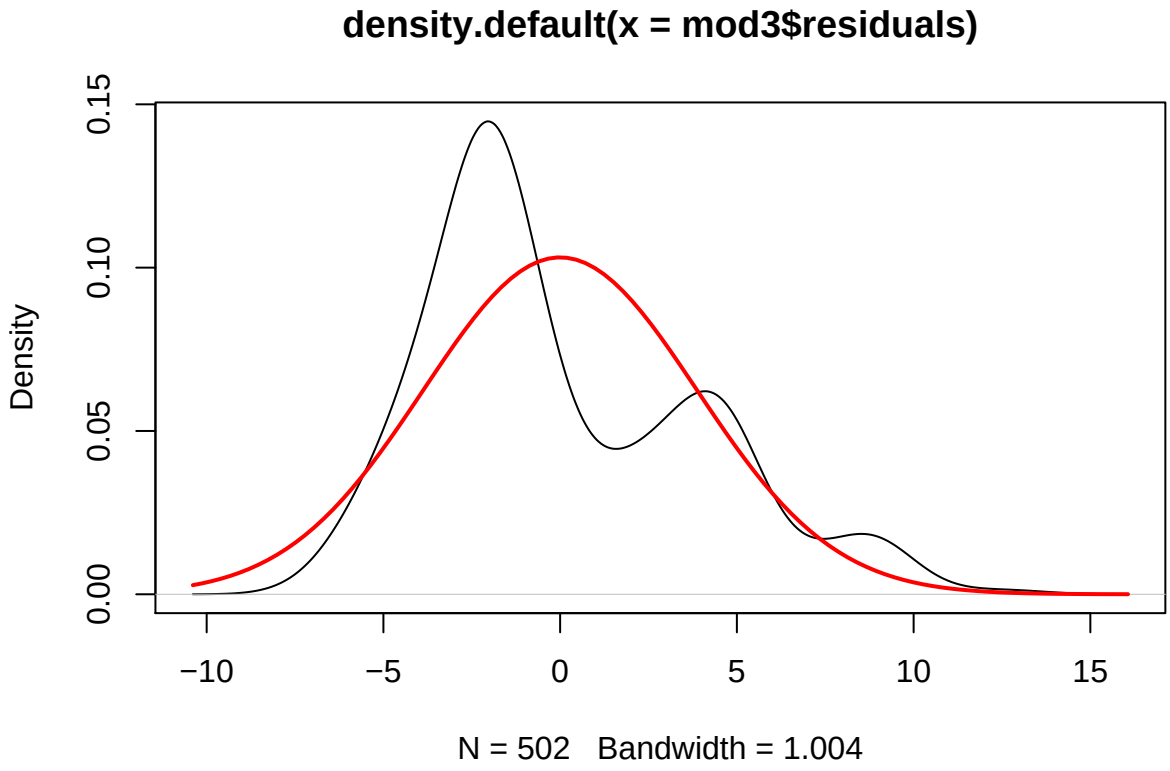



```
##
##  Runs Test
##
## data:  mod3$residuals
## statistic = -20.806, runs = 18, n1 = 187, n2 = 315, n = 502,
## p-value < 2.2e-16
## alternative hypothesis: nonrandomness
```

P-value jest bliskie 0, odrzucamy hipotezę zerową o losowości reszt

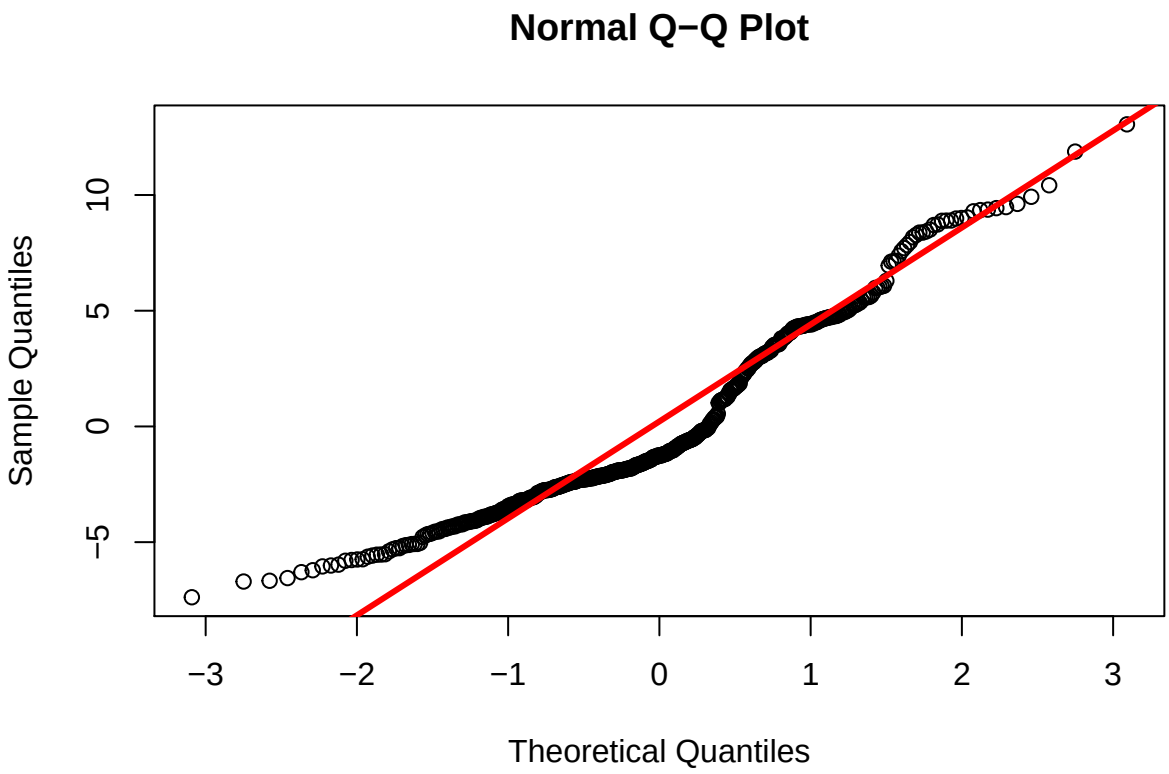
Wykresy normalności

```
plot(density(mod3$residuals))
curve(dnorm(x, 0, sd(mod3$residuals)), add = T, col = 2, lwd = 2)
```



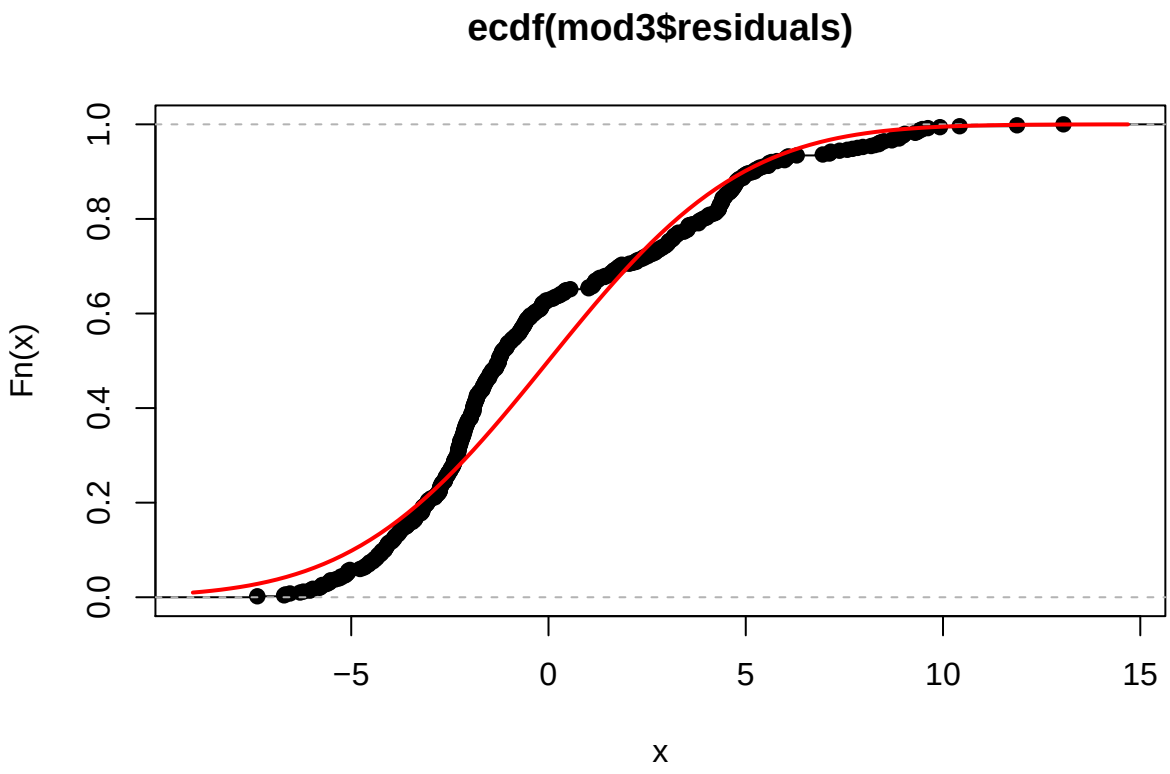
Wykres gęstości empirycznej znacząco różni się od gęstości rozkładu normalnego. Bardzo znacząco.

```
qqnorm(mod3$residuals)
qqline(mod3$residuals, col=2, lwd = 3)
```



Tutaj także wyraźne odchyłki.

```
plot(ecdf(mod3$residuals))
curve(pnorm(x, 0, sd(mod3$residuals)), add = T, col = 2, lwd =2)
```



Nawet na dystrybuancie widać, że rozkład normalny wygląda inaczej.

```
ks.test(x = mod3$residuals, y = "pnorm", mean = 0, sd = sd(mod3$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  mod3$residuals
## D = 0.14346, p-value = 2.127e-09
## alternative hypothesis: two-sided
```

```
lillie.test(mod3$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  mod3$residuals
## D = 0.14346, p-value < 2.2e-16
```

```
shapiro.test(mod3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod3$residuals
## W = 0.93719, p-value = 1.043e-13
```

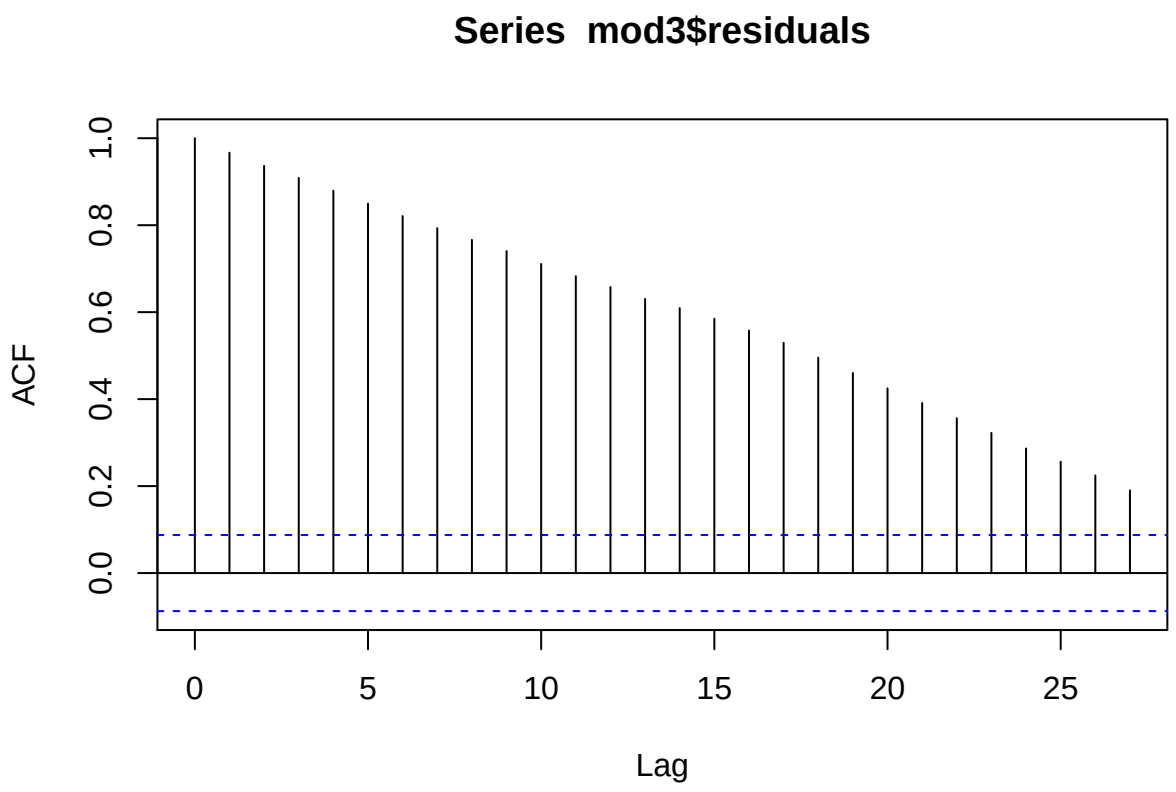
```
ad.test(mod3$residuals)
```

```
##
##  Anderson-Darling normality test
##
## data:  mod3$residuals
## A = 12.438, p-value < 2.2e-16
```

W każdym z testów p-value jest bardzo bliskie zero, stanowczo odrzucamy hipotezę o rozkładzie normalnym.

Badanie autokorelacji

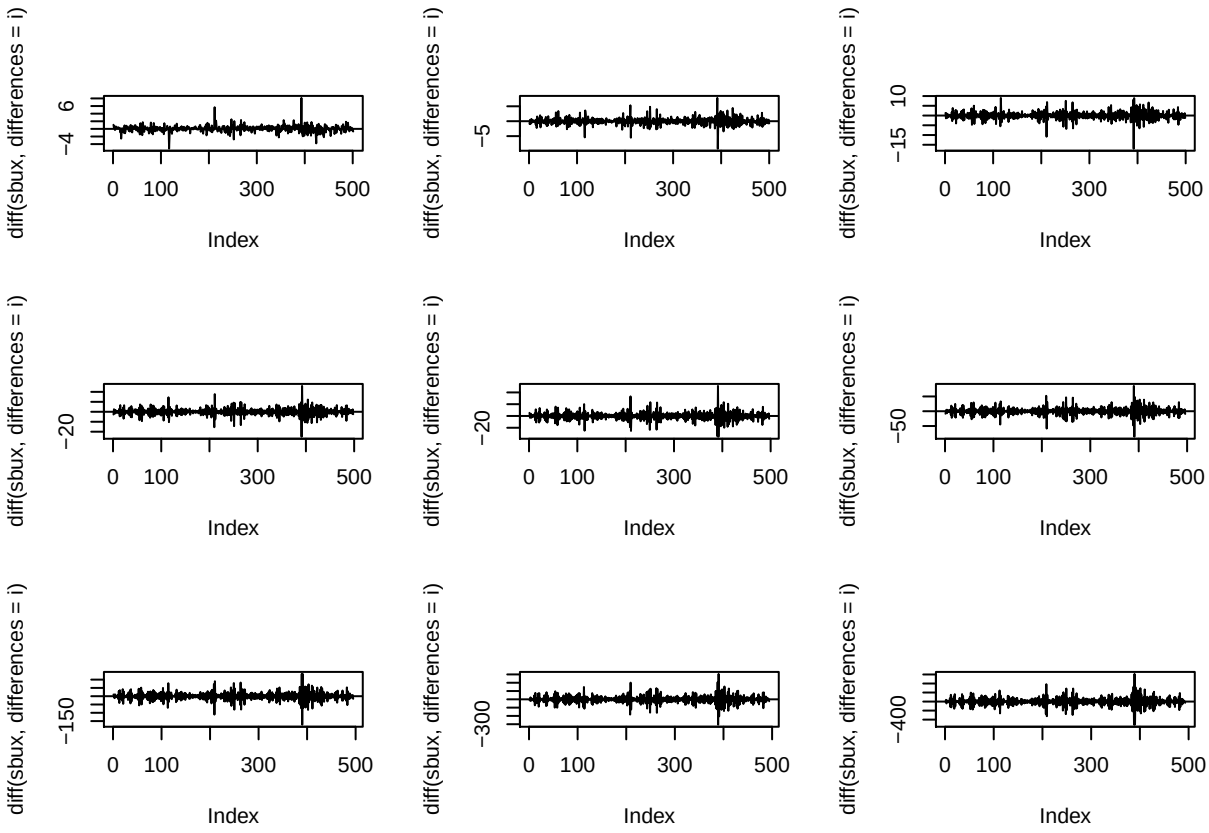
```
acf(mod3$residuals)
```



Słupki nie mieszczą się w niebieskim "pasku", zatem prawdopodobnie ma miejsce autokorelacja.

Metoda różnicowa

```
par(mfrow = c(3, 3))
for(i in 1:9){
plot(diff(sbux, differences = i), type = "l")
abline(h = 0)}
```




```
par(mfrow=c(1,1))
```

Z wykresów widać, że największa stabilizacja jest przy różnicowaniu rzędu 2 lub 3, potem rozrzut zaczyna się znacząco zwiększać.

Stacjonarność

```
adf.test(sbux) #niest
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sbux
## Dickey-Fuller = -2.1662, Lag order = 7, p-value = 0.5079
## alternative hypothesis: stationary
```

```
kpss.test(sbux) #niest
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  sbux
## KPSS Level = 7.4101, Truncation lag parameter = 5, p-value = 0.01
```

```
kpss.test(sbux, null = "Trend") #niest
```

```
##
```

```
##  KPSS Test for Trend Stationarity
##
## data:  sbux
## KPSS Trend = 1.0249, Truncation lag parameter = 5, p-value = 0.01
```

Szereg nie jest ani stacjonarny ani TS.

```
adf.test(diff(sbux, differences = 1)) #st
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(sbux, differences = 1)
## Dickey-Fuller = -7.8439, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(sbux, differences = 1)) #st
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(sbux, differences = 1)
## KPSS Level = 0.15287, Truncation lag parameter = 5, p-value = 0.1
```

```
kpss.test(diff(sbux, differences = 1), null = "Trend") #st
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  diff(sbux, differences = 1)
## KPSS Trend = 0.11762, Truncation lag parameter = 5, p-value = 0.1
```

Po zróżnicowaniu rzędu 1 szereg jest zarówno stacjonarny jak i TS.

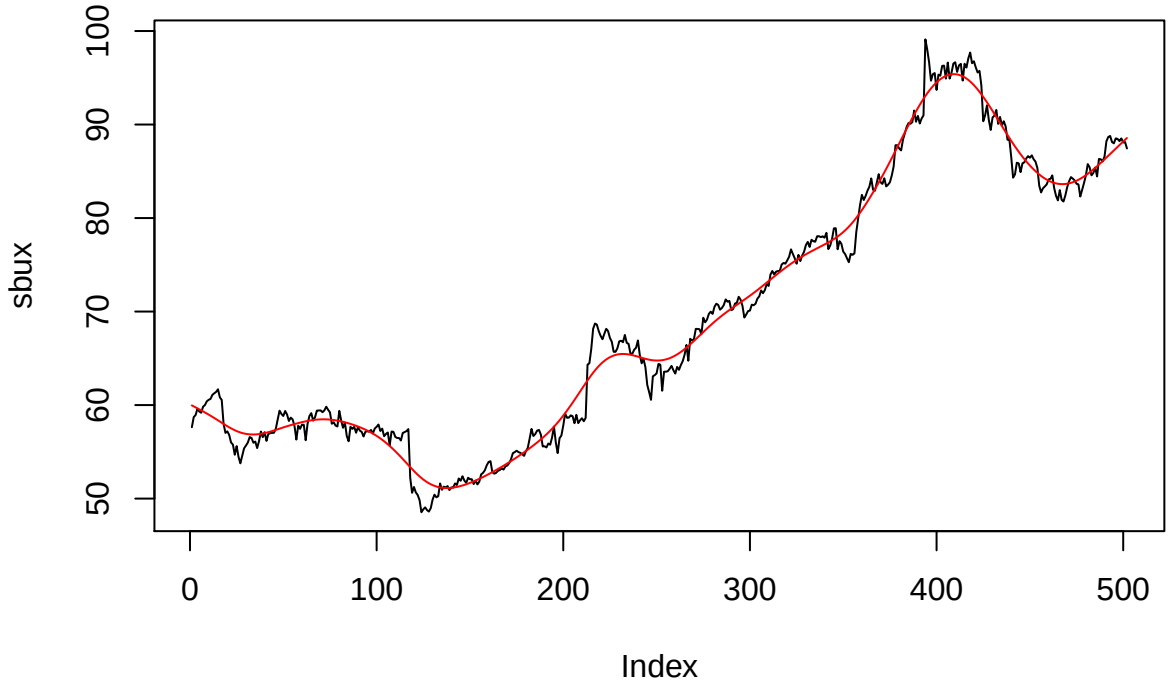
```
auto.arima(sbux)
```

```
## Series: sbux
## ARIMA(0,1,0)
##
## sigma^2 estimated as 0.8884:  log likelihood=-681.25
## AIC=1364.51   AICc=1364.52   BIC=1368.73
```

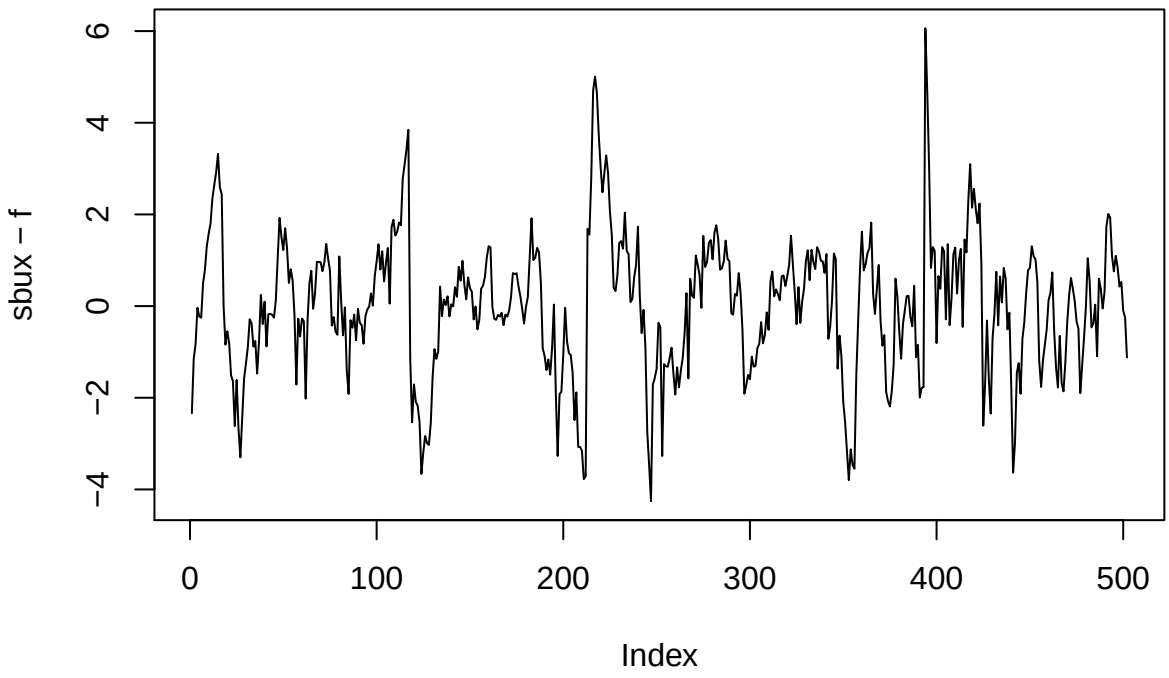
Inne rzeczy

Rysunek bez trendu.

```
f <- FRAPO::trdhp(sbux, 14400)
plot(sbux, type = "l")
lines(f, col = 2)
```

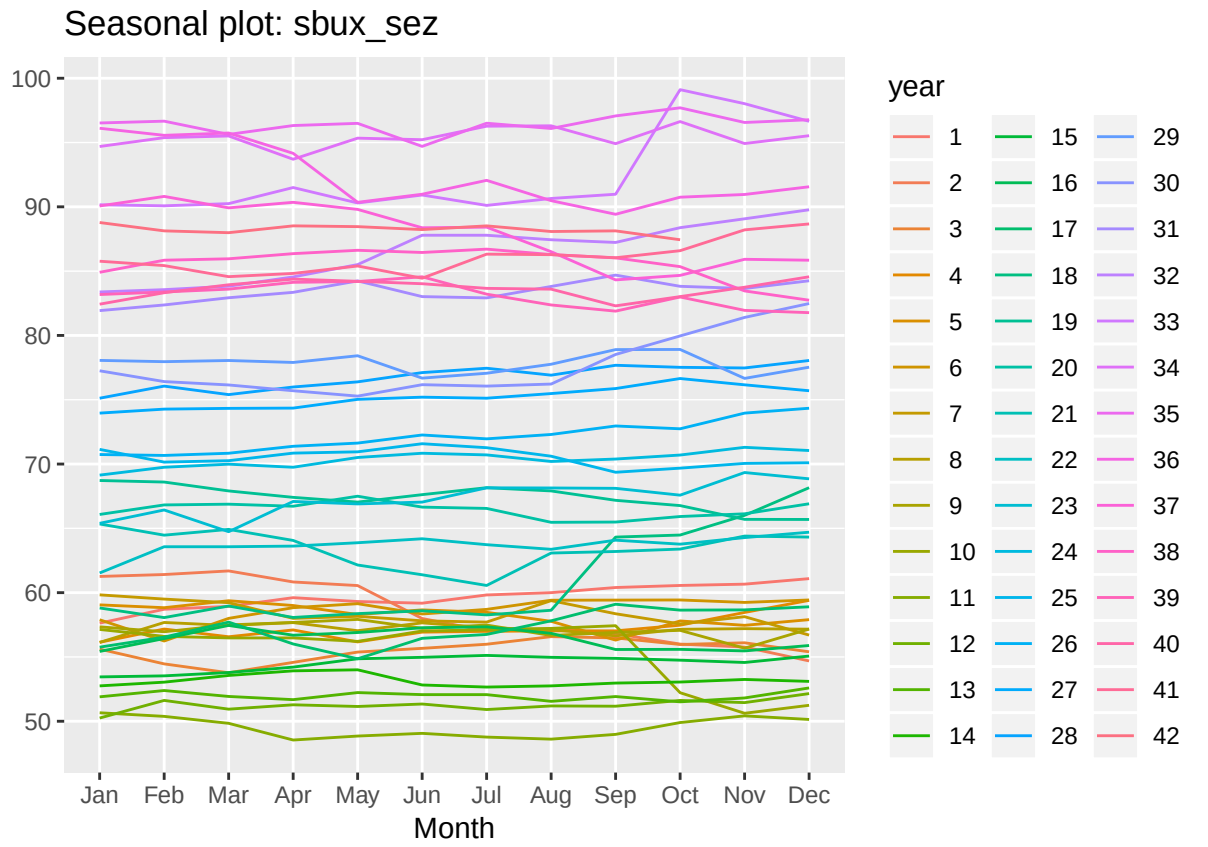



```
plot(sbux - f, type = "l")
```

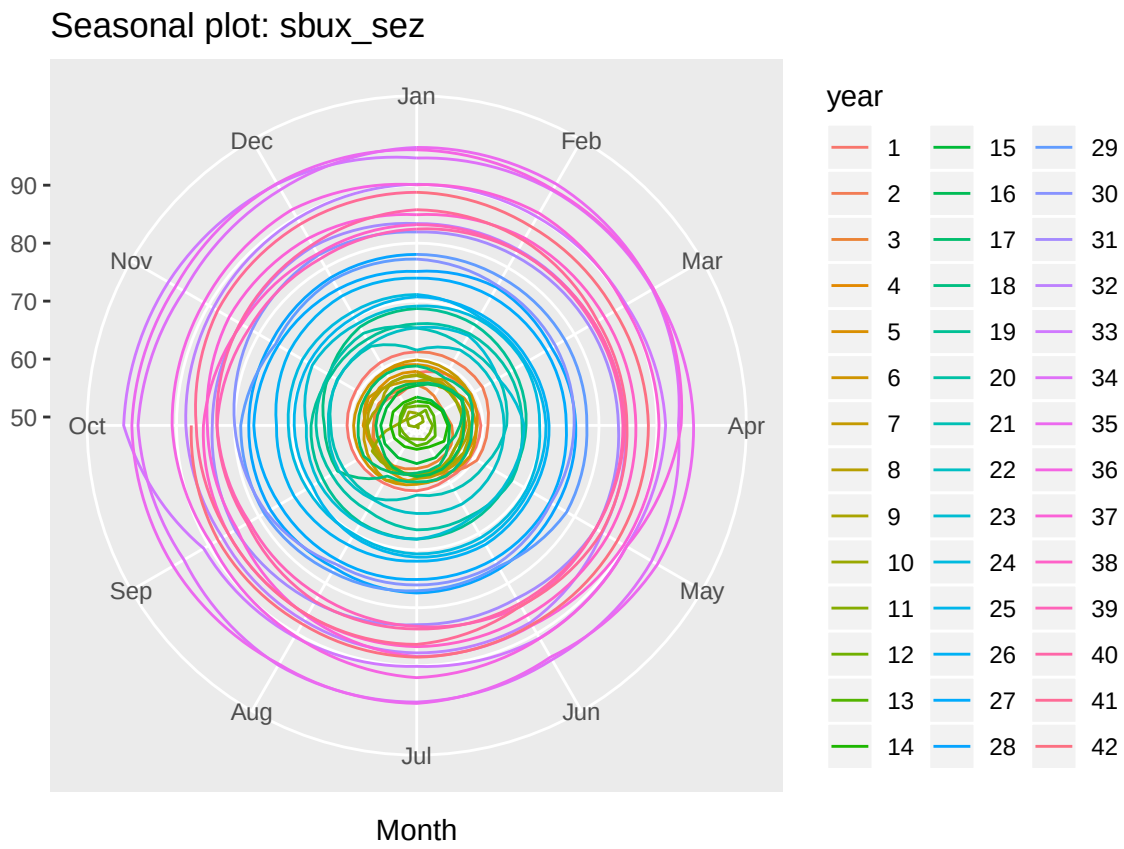


Sezonowość

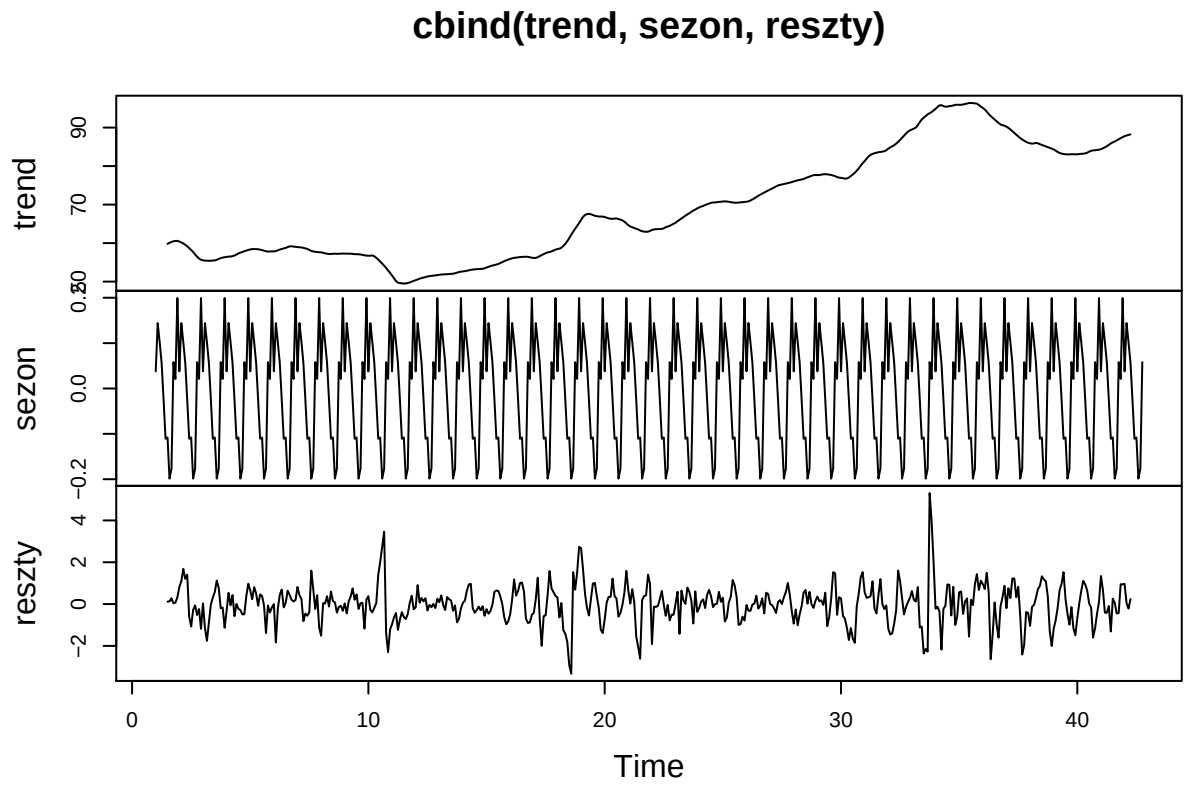
```
sbux_sez <- ts(sbux, frequency = 12)
sbux_dek <- decompose(sbux_sez)
forecast:: ggseasonplot(sbux_sez)
```



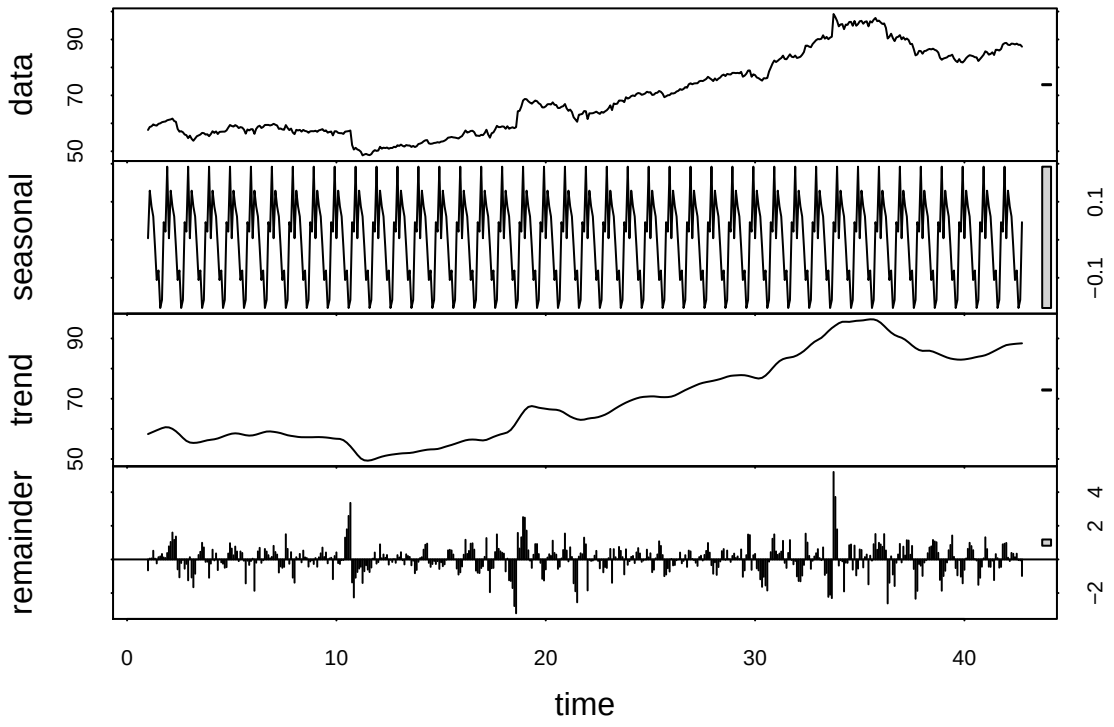

```
forecast::ggseasonplot(sbux_sez, polar = T)
```



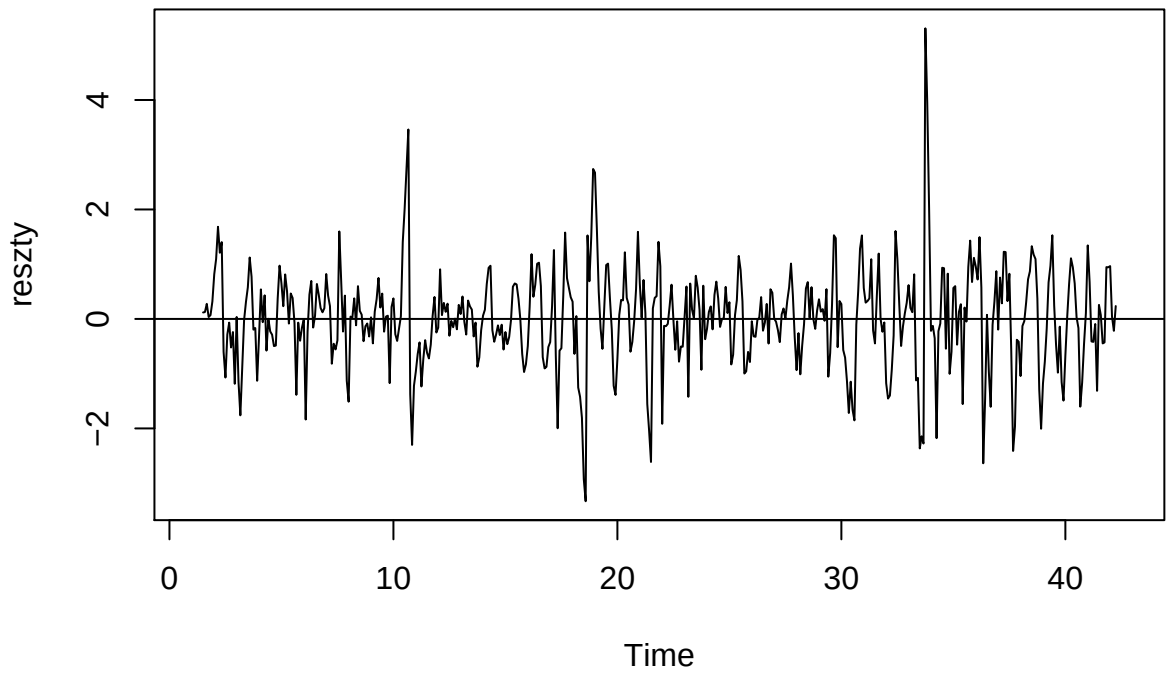

```
trend <- sbux_dek$trend
sezon <- sbux_dek$seasonal
reszty <- sbux_dek$random
plot(cbind(trend, sezon, reszty))
```




```
plot(stl(sbux_sez, "periodic"))
```



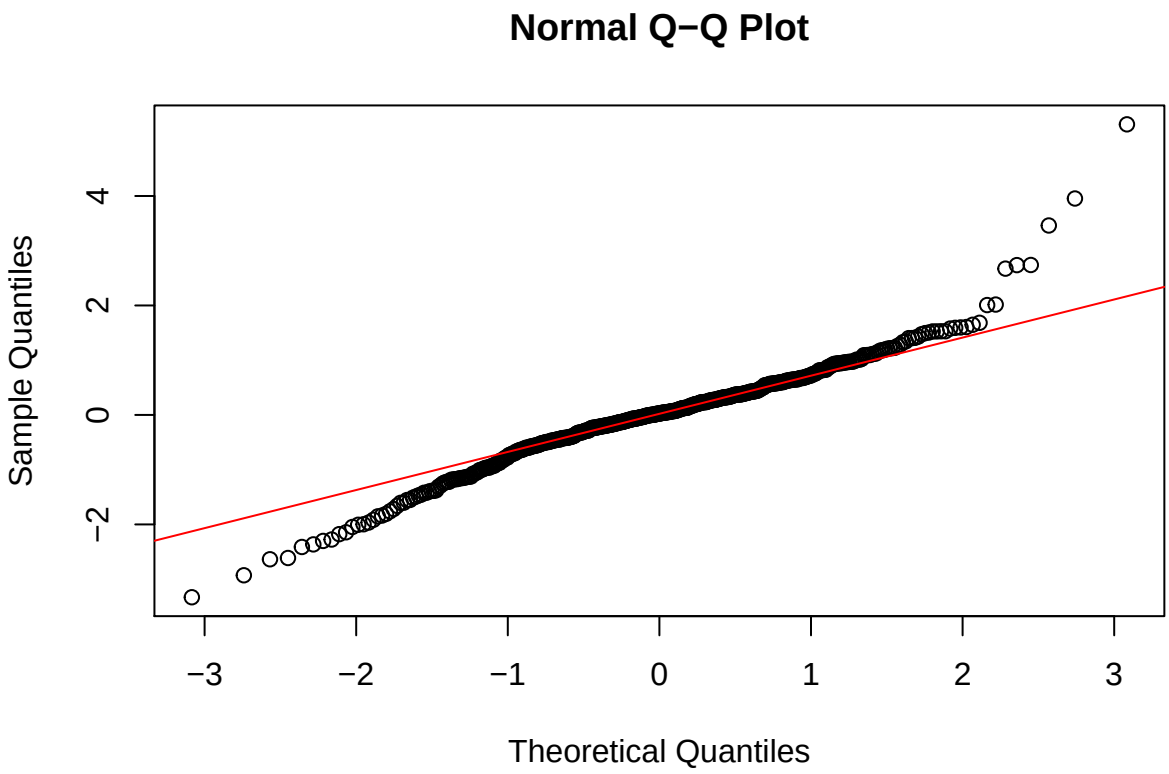

```
plot(reszty)
abline(h=0)
```




```
shapiro.test(reszty) #ohohoo malutkie p value
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reszty
## W = 0.95604, p-value = 6.553e-11
```

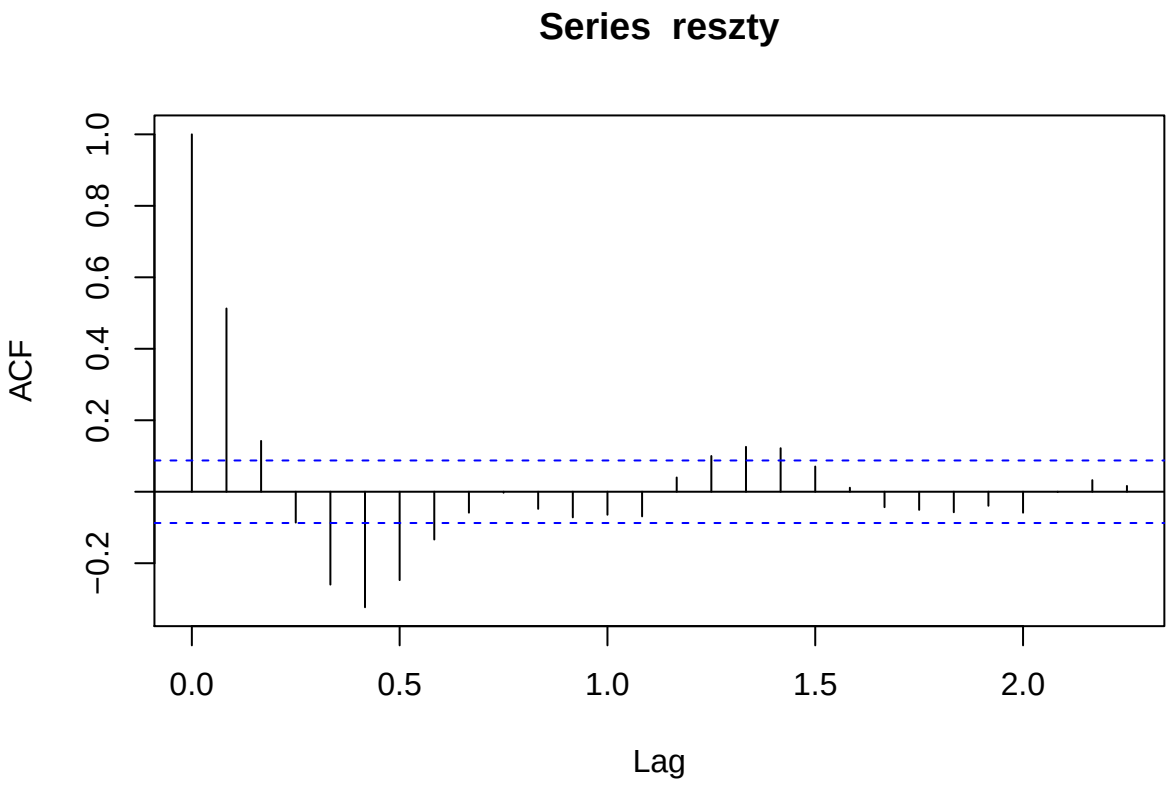
```
qqnorm(reszty)
qqline(reszty, col = 2)
```




```
#nie ma normalności!!
```

Stanowczo nie ma normalności reszt.

```
acf(reszty, na.action = na.pass) #dla dalszych z grubsza się mieszczą w pasku
```



Z wykresu wynika, że raczej nie mamy do czynienia z autokorelacją.