

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

Alicja Hołowiecka, Matylda Jankowska, Marcin Dziadosz

23 12 2019

Spis treści

Matylda!! (to do, to do, to do to do to do to doooooooo)	2
Wstęp	2
Wczytanie bibliotek	2
McDonald's	3
Opis firmy	3
Wczytanie danych i rysunki	3
Wydzielanie trendu	3
Ruchoma średnia	4
Metoda wykładniczych wag ruchomej średniej	5
Dopasowanie wielomianu	6
Metoda różnicowa	6
Dopasowanie modelu liniowego	7
Dopasowanie wielomianem drugiego stopnia	8
Dopasowanie wielomianem trzeciego stopnia	10
Dopasowanie wielomianem czwartego stopnia	13
Testy na resztach modelu	13
Jednorodność wariancji	13
Normalność	15
Autokorelacja	15
Stacjonarność	18
ARIMA	19
Testy na resztach modelu ARIMA	20
Prognozowanie	22
Metoda dryftu	22
Przewidywania z modelu	23
Podsumowanie analizy notowań McDonald's	24
Starbucks	24
Opis firmy	24
Wczytanie danych i rysunki	24
Wydzielanie trendu	25
Ruchoma średnia	25
Metoda wykładniczych wag ruchomej średniej	26
Dopasowanie wielomianu	27
Metoda różnicowa	27
Dopasowanie modelu liniowego	28
Dopasowanie wielomianu drugiego stopnia	30
Dopasowanie wielomianu trzeciego stopnia	31
Dopasowanie wielomianu czwartego stopnia	34
Testy na resztach modelu	34
Stacjonarność	39
Podsumowanie analizy notowań Starbucks	40

Matylda!! (to do, to do, to do to do to do to doooooooo)

- estetycznie można trochę poprawić, np. jak są te testy kpss i inne, to na razie robiłam tak, żeby wyświetlało ten domyślny output, może zamiast tego zrób tak jak Ty robiłaś w testach, że zapisać p-value do zmiennej i opisać? i popatrz jak wyglądają wykresy, szczególnie w pdf, może coś nie wygląda tak dobrze jak się nam wydawało, np tytuły się nie mieszczą, albo coś jest za bardzo ściśnięte, np wiem że te wykresy dla wielomianu 4 stopnia z nieznanymi dla mnie przyczyn strasznie odwalają w pdf
- jeżeli nadal coś się nie zgadza z tą kolejnością co Kozłowski proponował, to możesz poprzestawiać
- co do opisów firm - ze Starbucksa trzeba wywalić zdjęcia bo dupnie wyglądają w pdf. I nie wiem czy ewentualnie z neta nie skopiować ze dwa zdania więcej do każdego, bo jakoś blado. spadliśmy już do 44 stron także jest coraz lepiej :) a jak wywalimy rysunki to jeszcze kolejne 3-4 strony pójdą
- sprawdź, jak wyglądają tabelki w pdfie - czy na pewno każda się wyświetla dokładnie gdzie ma być. Jeśli nie, to np dla tabelki robionych poprzez `kable_styling` można dodawać parametry: `position = 'center'`, `latex_options = "HOLD_position"` i powinny się trzymać swojego miejsca
- za Chiny ludowe nie wiem jak robić jakieś prognozy czy cokolwiek... bo ten holt winters to chyba tylko dla sezonowych idzie. zobacz czy coś się da, a jak nie to usuwaj to gadownictwo.
- co do ARIMY - opisałam tak, jak na laborkach robiliśmy, zobacz czy z tego modelu da się wyciągnąć reszty i robić na nich testy. jeśli się da, to zrób te testy
- myślałam żeby na koniec każdej z tych firm zrobić takie podsumowanie - że np dla McDonalda z modeli wielomianowych najlepszy był taki i taki, jego reszty spełniają/nie spełniają cośtam, oprócz tego dopasowaliśmy model ARIMA taki śmaki, jego reszty cośtam cośtam... jakieś takie ładne sratatata na zakończenie
- pracuj przede wszystkim nad McDonaldem, potem Marcin spisie do Starbucks. Chociaż tam też oczywiście możesz coś zacząć żeby mu ułatwić.

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.

Wczytanie bibliotek

Na początek wczytamy wszystkie potrzebne biblioteki. Biblioteka `tseries` przyda nam się m. in. do wczytania danych oraz wykonania testów na stacjonarność szeregu. Z pakietu `randtests` skorzystamy przy testach na losowość reszt. W bibliotece `nortest` znajduje się wiele testów na normalność. Dzięki paczce `lmtest` utworzymy modele wielomianowe różnego stopnia i zbadamy ich dopasowanie. W bibliotece `forecast` są m. in. funkcje dotyczące modeli ARIMA. Pakiety `stargazer`, `tables` i `kableExtra` pozwolą nam estetycznie wyświetlać tabele.

```
library(tseries)
library(randtests)
library(nortest)
library(lmtest)
library(forecast)
library(stargazer)
library(kableExtra)
library(tidyverse)
library(tables)
```

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

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")
```

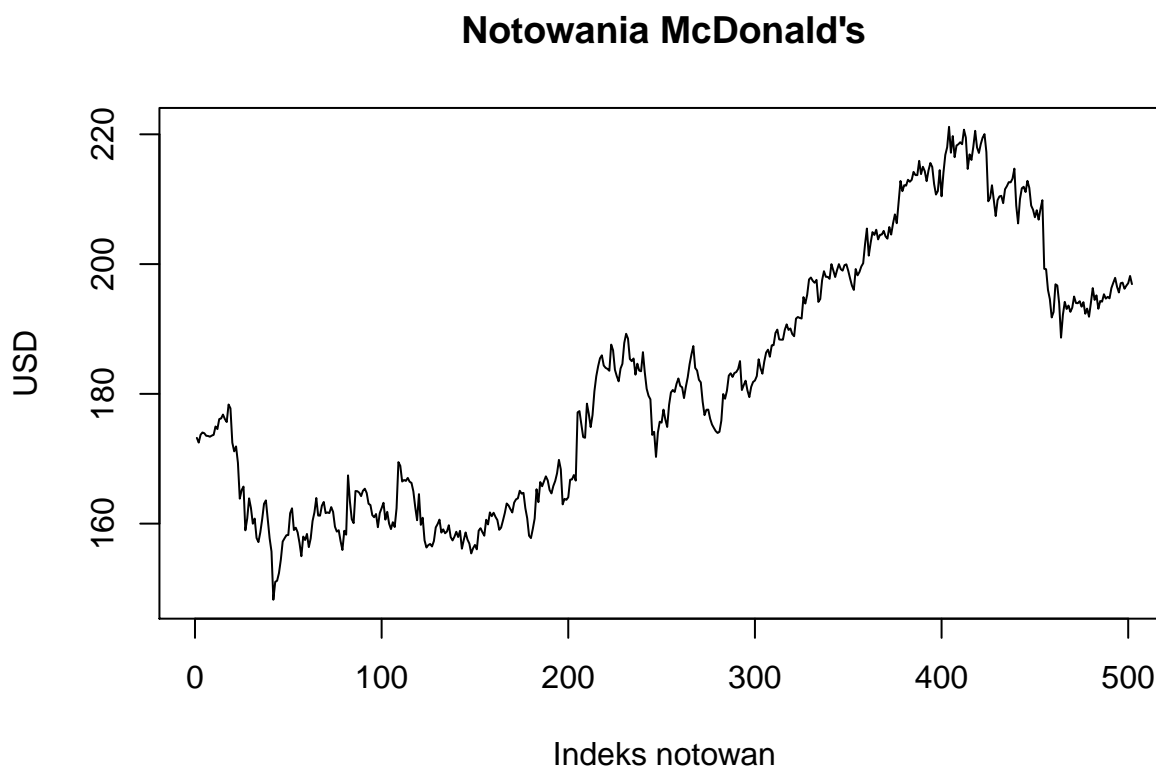
```
## time series starts 2018-01-02
```

```
## 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 = "Indeks notowań", ylab = "USD", main = "Notowania McDonald's")
```



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

Wydzielanie trendu

Spróbujemy wydzielić część deterministyczną. Do tego celu posłużymy się metodami średnich ruchomych (prostą i wykładniczą).

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

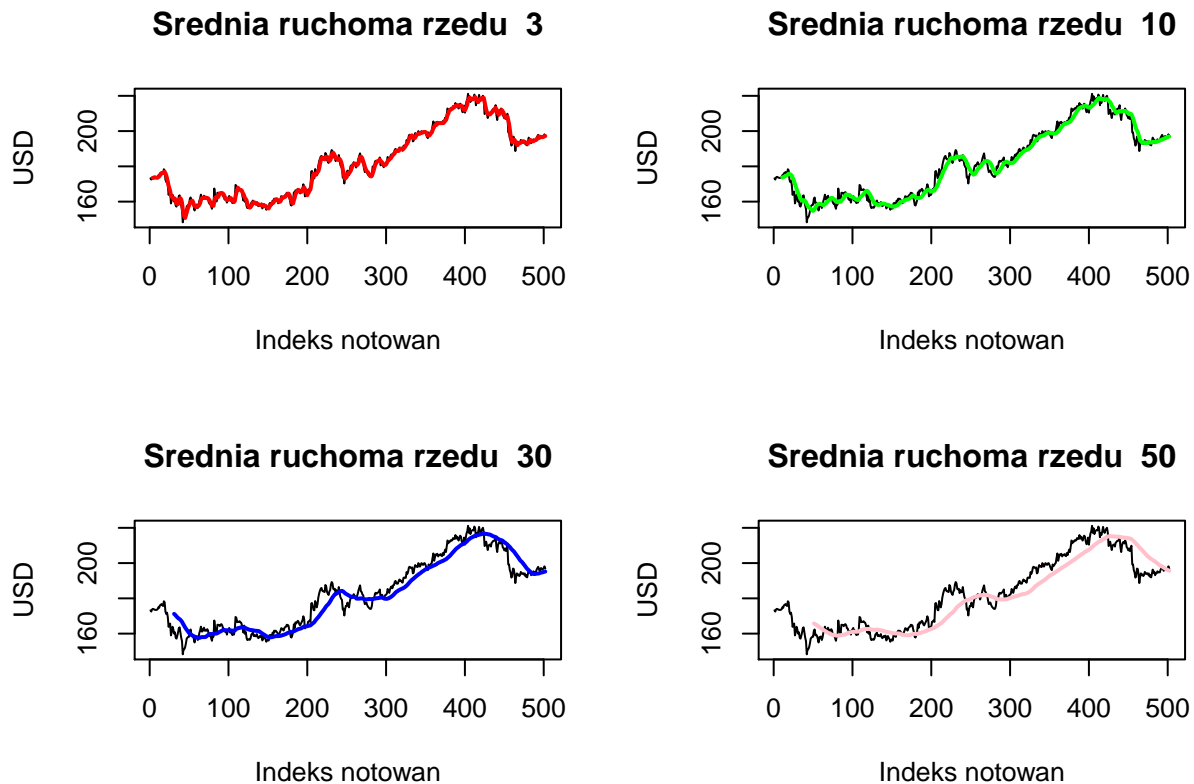
$$\hat{f}(t) = \frac{1}{m+1} \sum_{k=0}^m x_{t-k}$$

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.

```
ruchoma <- function(x, m, kolor){  
  t <- length(x)  
  f <- NULL  
  for(i in (m+1):t){  
    f[i] <- mean(x[(i-m):i])  
  }  
  tytul = paste("Średnia ruchoma rzędu ", m)  
  plot(x, type = "l", main = tytul, xlab = "Indeks notowań", ylab = "USD")  
  lines((m+1):t, f[(m+1):t], lwd = 2, col = kolor)  
}
```

Narysujemy wykresy dla kilku parametrów `m`.

```
par(mfrow = c(2, 2))  
ruchoma(mcd, 3, "red")  
ruchoma(mcd, 10, "green")  
ruchoma(mcd, 30, "blue")  
ruchoma(mcd, 50, "pink")
```



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

Jak widać, im większy parametr m przyjmujemy, 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:

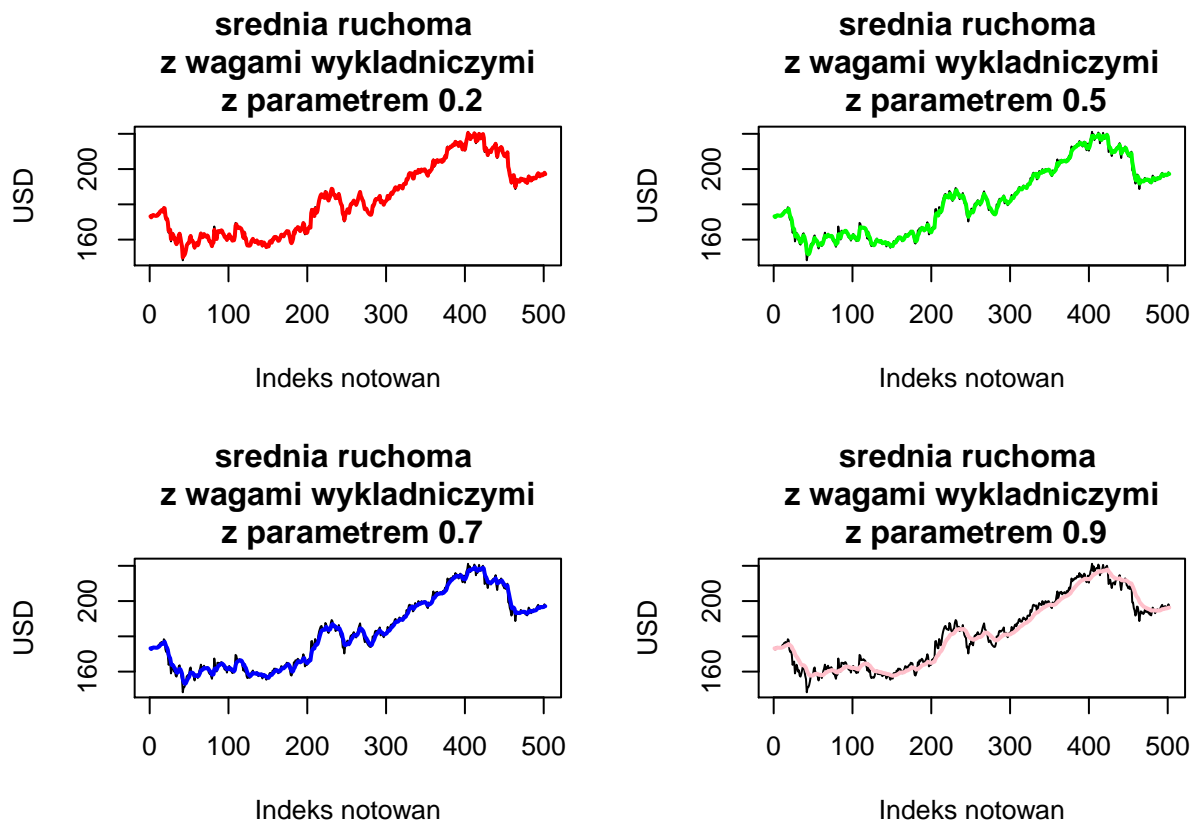
$$\hat{f}(t) = \frac{1 - \eta}{1 - \eta^t} \left[x_t + \eta \frac{1 - \eta^{t-1}}{1 - \eta} \hat{f}(t-1) \right]$$

```
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])
}
tytul = paste("średnia ruchoma \n z wagami wykładniczymi \n z parametrem", mi)
plot(x, type = "l", main = tytul, xlab = "Indeks notowań", ylab = "USD")
lines(1:length(x), f, lwd = 2, col = kolor)
}
```

```
par(mfrow = c(2, 2))
wykladnicza(mcd, 0.2, "red")
wykladnicza(mcd, 0.5, "green")
wykladnicza(mcd, 0.7, "blue")
wykladnicza(mcd, 0.9, "pink")
```



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

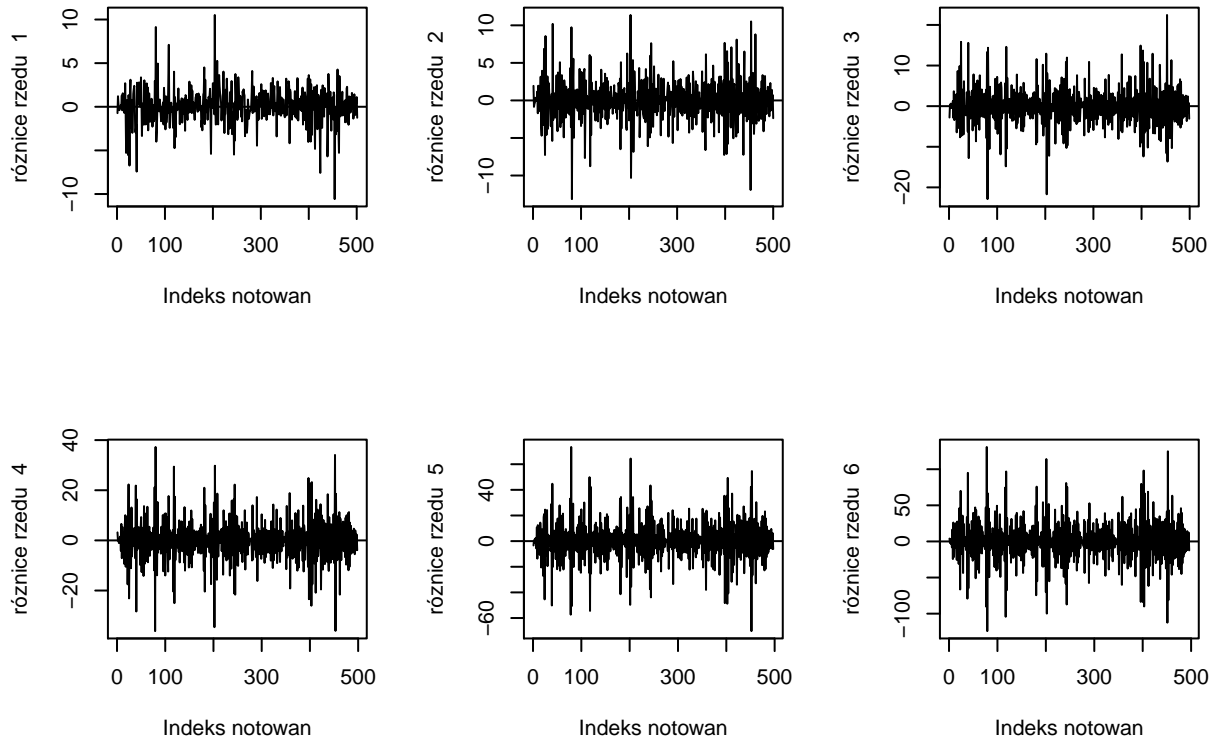
Podobnie jak w przypadku prostej metody średniej ruchomej - im większy parametr η , tym bardziej wygładzony wykres, ale i mniejsza dokładność. Jednakże, dokładność jest i tak większa niż w przypadku prostej metody ruchomej średniej.

Dopasowanie wielomianu

Metoda różnicowa Za pomocą metody różnicowej sprawdzimy, jaki stopień wielomianu byłby najbardziej odpowiedni.

```
par(mfrow = c(2, 3))
for(i in 1:6){
  ylab <- paste("różnice rzędu ", i)
```

```
plot(diff(mcd, differences = i), type = "l", xlab = "Indeks notowań", ylab = ylab)
abline(h = 0)}
```



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

Z wykresów można wysnuć wniosek, że największą stabilizację osiągamy przy różnicowaniu rzędu pierwszego lub drugiego. Wraz ze wzrostem rzędu różnicowania, wahania ulegają znacznemu rozszerzeniu.

Spróbujemy do danych dopasować wielomian stopnia 1, 2, 3 i 4.

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

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

```
stargazer(mod1, header = F)
```

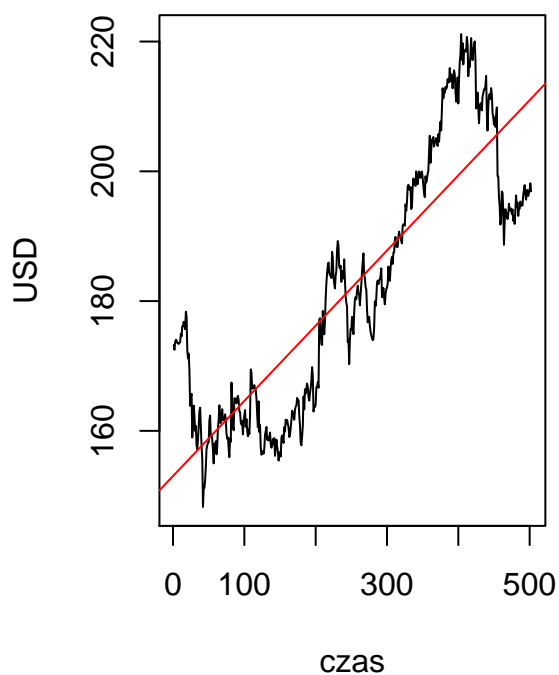
Dopasowanie modelu liniowego Zarówno wyraz wolny, jak i współczynnik kierunkowy są istotne statystycznie. R^2 wynosi około 74%.

```
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)
```

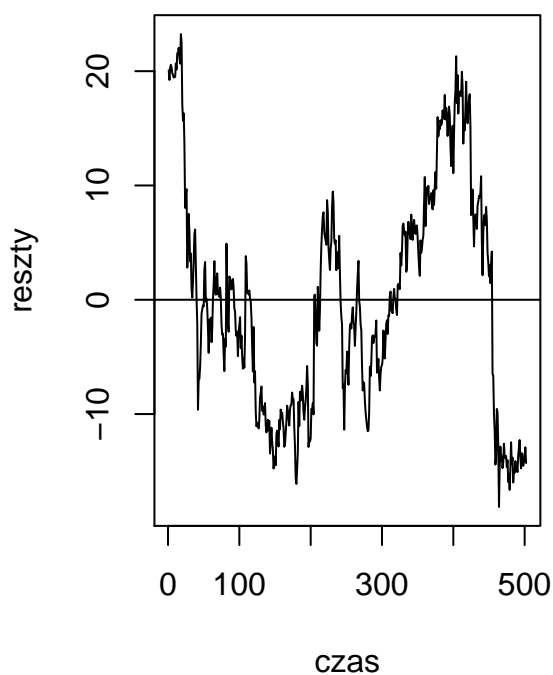
Tablica 1:

<i>Dependent variable:</i>	
mcd	
t	0.116*** (0.003)
Constant	153.029*** (0.885)
Observations	502
R ²	0.743
Adjusted R ²	0.742
Residual Std. Error	9.905 (df = 500)
F Statistic	1,442.546*** (df = 1; 500)
<i>Note:</i> *p<0.1; **p<0.05; ***p<0.01	

Model liniowy



Reszty modelu liniowego



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

Dopasowanie wielomianem drugiego stopnia Teraz stworzymy model wielomianowy drugiego stopnia.

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



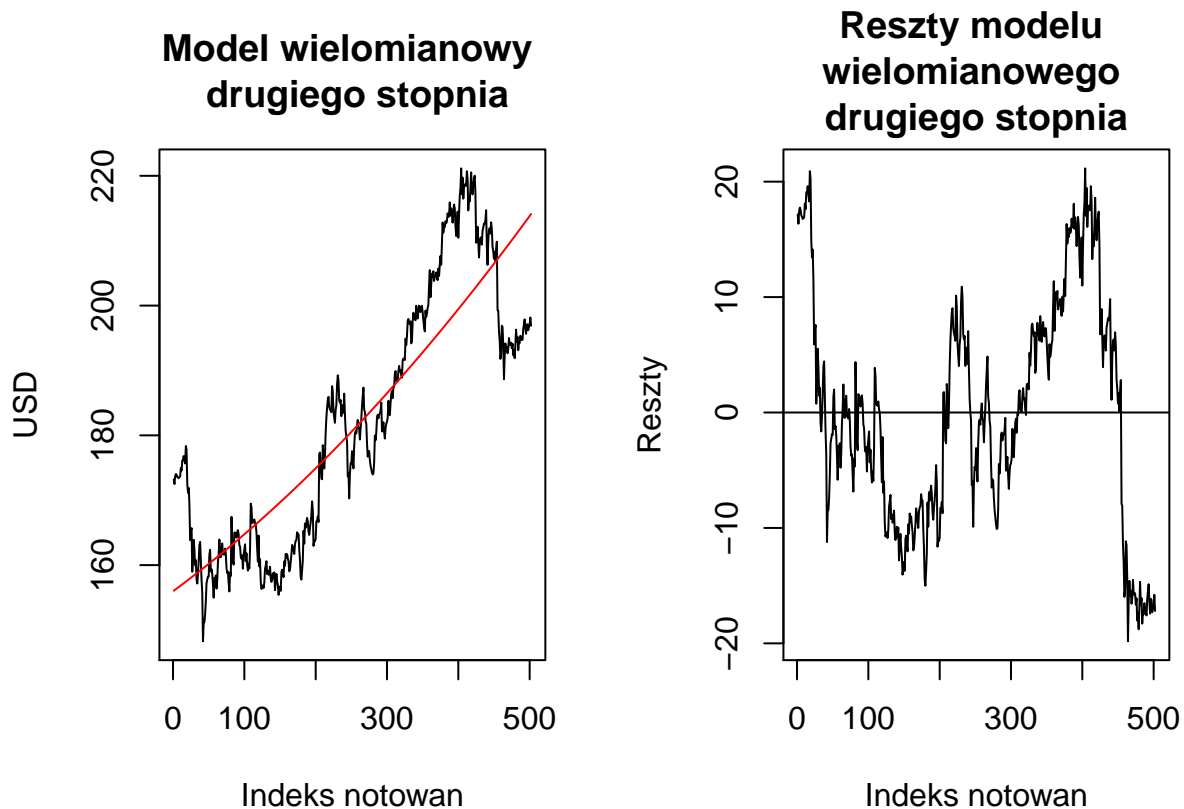
```
stargazer(mod2, header = F)
```

Tablica 2:

	<i>Dependent variable:</i>
	mcd
t	0.081*** (0.012)
I(t ²)	0.0001*** (0.00002)
Constant	155.975*** (1.321)
Observations	502
R ²	0.747
Adjusted R ²	0.746
Residual Std. Error	9.828 (df = 499)
F Statistic	737.143*** (df = 2; 499)
<i>Note:</i>	*p<0.1; **p<0.05; ***p<0.01

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 wielomianowy \n drugiego stopnia",
     xlab = "Indeks notowań", ylab = "USD")
lines(t, mod2$fitted.values, col = "red")
plot(mod2$residuals, type = "l",
     main = "Reszty modelu \nwielomianowego \ndrugiego stopnia",
     xlab = "Indeks notowań", ylab = "Reszty")
abline(h = 0)
```



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

Model kwadratowy zachowuje się bardzo podobnie jak model liniowy.

Dopasowanie wielomianem trzeciego stopnia Jako kolejny zbudowany zostanie model wielomianowy trzeciego stopnia.

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

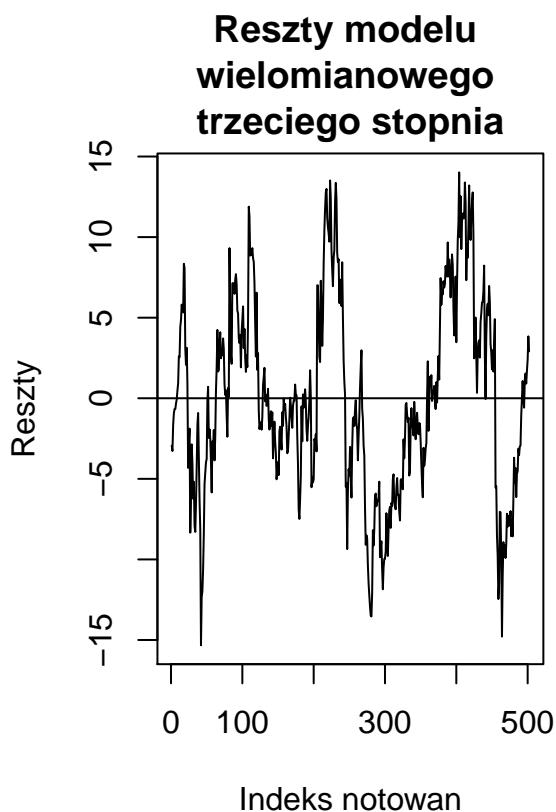
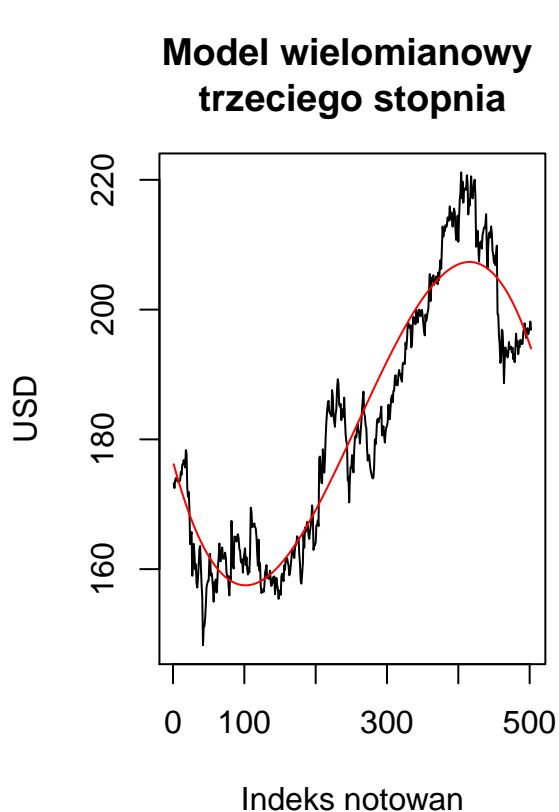
```
stargazer(mod3, header = F)
```

W modelu wielomianowym trzeciego stopnia 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 wielomianowy \ntrzeciego stopnia",
     xlab = "Indeks notowań", ylab = "USD")
lines(t, mod3$fitted.values, col = "red")
plot(mod3$residuals, type = "l",
     main = "Reszty modelu \nwielomianowego \ntrzeciego stopnia",
     xlab = "Indeks notowań", ylab = "Reszty")
abline(h= 0)
```

Tablica 3:

<i>Dependent variable:</i>	
mcd	
t	-0.408*** (0.019)
I(t ²)	0.002*** (0.0001)
I(t ³)	-0.00000*** (0.00000)
Constant	176.570*** (1.096)
Observations	502
R ²	0.903
Adjusted R ²	0.902
Residual Std. Error	6.094 (df = 498)
F Statistic	1,544.632*** (df = 3; 498)
<i>Note:</i> *p<0.1; **p<0.05; ***p<0.01	



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

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

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

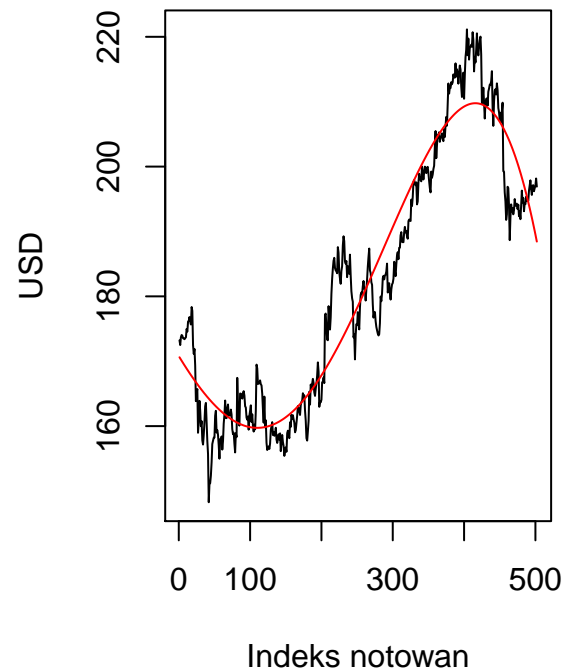
```
stargazer(mod4, header = F)
```

Tablica 4:

	<i>Dependent variable:</i>
	mcd
t	-0.180*** (0.036)
I(t^2)	0.0005 (0.0003)
I(t^3)	0.00000*** (0.00000)
I(t^4)	-0.000*** (0.000)
Constant	170.792*** (1.309)
Observations	502
R ²	0.912
Adjusted R ²	0.912
Residual Std. Error	5.798 (df = 497)
F Statistic	1,293.328*** (df = 4; 497)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
par(mfrow = c(1, 2))
plot(mcd, type = "l", main = "Model wielomianowy \nczwartego stopnia",
     xlab = "Indeks notowań", ylab = "USD")
lines(t, mod4$fitted.values, col = "red")
plot(mod4$residuals, type = "l",
     main = "Reszty modelu \nwielomianowego \nczwartego stopnia",
     xlab = "Indeks notowań", ylab = "Reszty")
abline(h= 0)
```

Model wielomianowy czwartego stopnia



Dopasowanie wielomianem czwartego stopnia

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

W modelu wielomianowym czwartego stopnia 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 wielomianowego 3 stopnia. 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 wielomianowy 3 stopnia.

Testy na resztach modelu

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

- jednorodność wariancji
- normalność
- losowość

Jednorodność wariancji Aby zbadać czy jednorodność reszt posłużymy się kilkoma popularnymi testami.

- Test Breuscha-Pagana

H_0 : jednorodność wariancji reszt.

H_1 : wariancja reszt zależy od zmiennych objaśniających w modelu.

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

P-value wynosi 0.0000044826, zatem Wug testu Breuscha-Pagana należałoby 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.

H_0 : wariancja reszt jest równa w obu grupach.

H_1 : wariancja reszt różni się w obu grupach.

```
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 się zależność wariancji reszt dla całości obserwacji i wybranego kwantyla (w tym przypadku rzędu 0.5).

H_0 : wariancja reszt jest równa w porównywanych grupach.

H_1 : wariancja reszt różni się w porównywanych grupach.

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

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

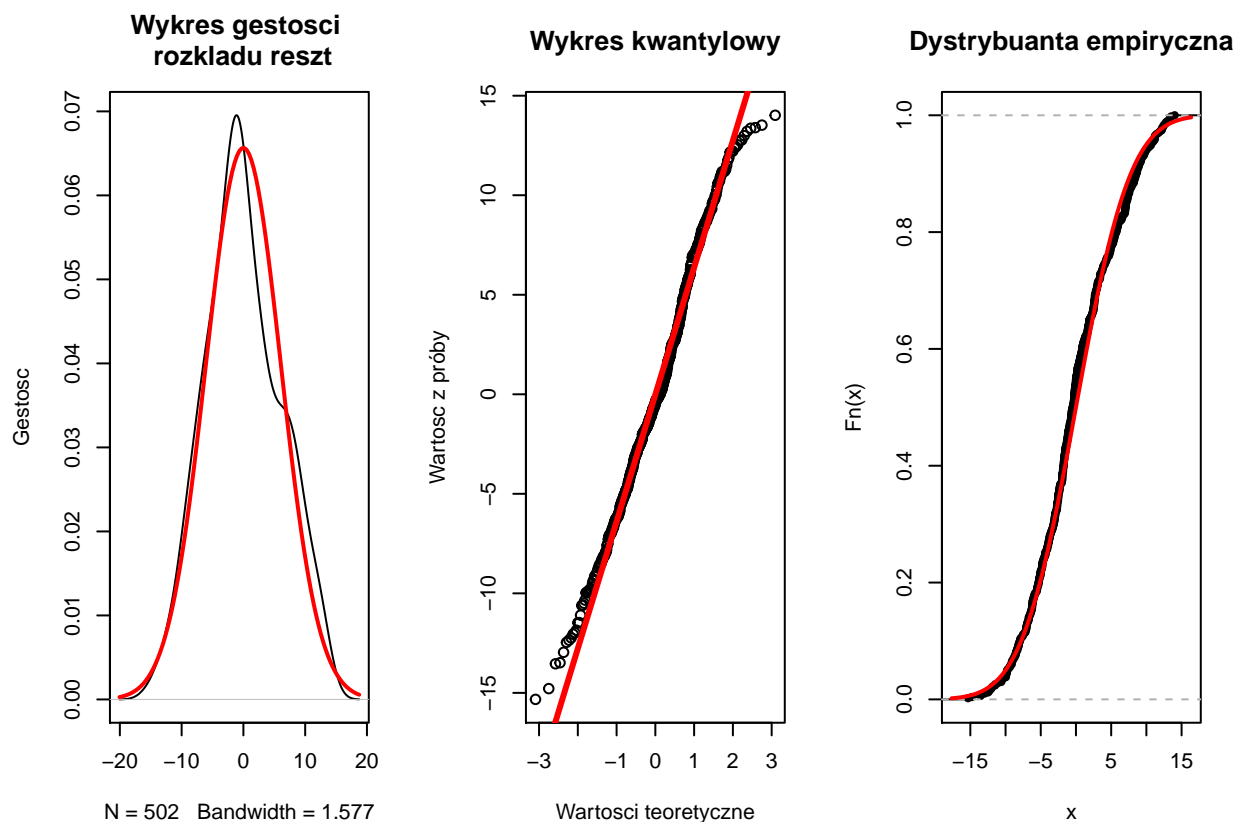
Biorąc pod uwagę uzyskane wyniki, należy przyjąć, że reszty z modelu trzeciego stopnia nie są jednorodne.

```
par(mfrow = c(1, 3))

plot(density(mod3$residuals),
     main = "Wykres gęstości \n rozkładu reszt", ylab = "Gęstość")
curve(dnorm(x, 0, sd(mod3$residuals)), add = T, col = 2, lwd = 2)

qqnorm(mod3$residuals, main = "Wykres kwantylowy", xlab = "Wartości teoretyczne",
       ylab = "Wartość z próby")
qqline(mod3$residuals, col=2, lwd = 3)

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



Normalność

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

```
kstest <- ks.test(x = mod3$residuals, y = "pnorm", mean = 0, sd = sd(mod3$residuals))$p.value
lillie <- lillie.test(mod3$residuals)$p.value
shapiro <- shapiro.test(mod3$residuals)$p.value
adtest <- ad.test(mod3$residuals)$p.value
```

Z testów Kołmogorowa-Lillieforsa, Shapiro-Wilka oraz Andersona-Darlinga wynika, że musimy odrzucić hipotezę o normalności rozkładu reszt (dla testu Kołmogorowa-Smirnova nie było podstaw do odrzucenia, p-value około 0.1350084). 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.

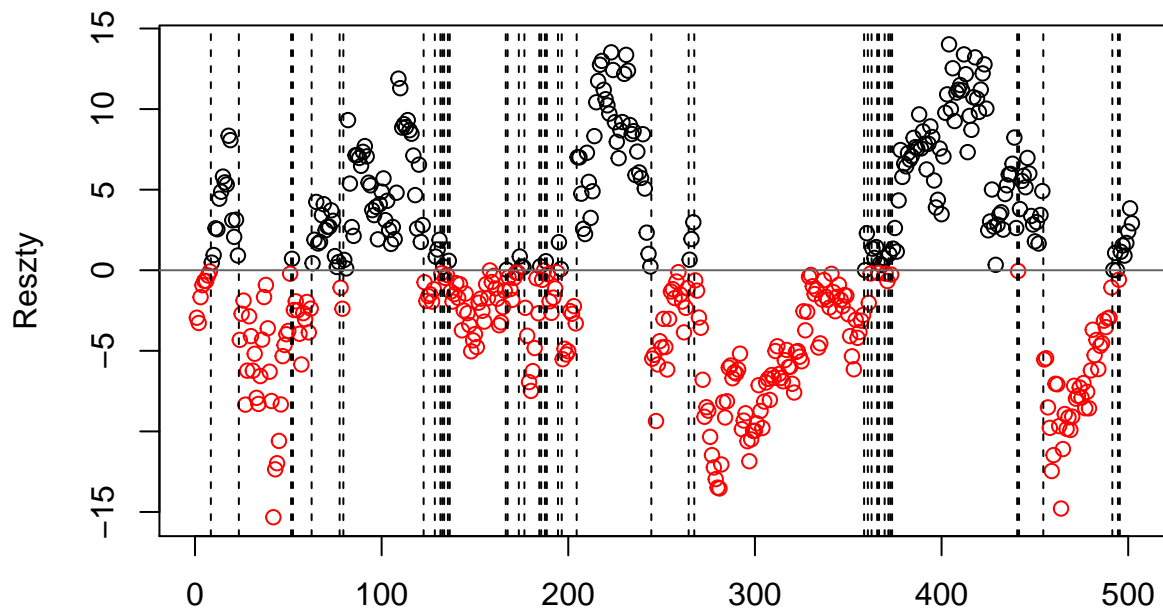
Autokorelacja

- Test serii

H_0 : losowość

H_1 : brak losowości

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



```
##
## Runs Test
##
## data: Reszty
## 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. Także na wykresie widać, że prawdopodobnie istnieje zależność między kolejnymi resztami, prawdopodobnie występuje autokorelacja.

- Test Durbina-Watsona

Weryfikuje hipotezę o niezależności reszt, sprawdzając, czy istotna jest autokorelacja reszt rzędu pierwszego.

```
dwtest <- dwtest(mod3, order.by = ~t)$p.value
```

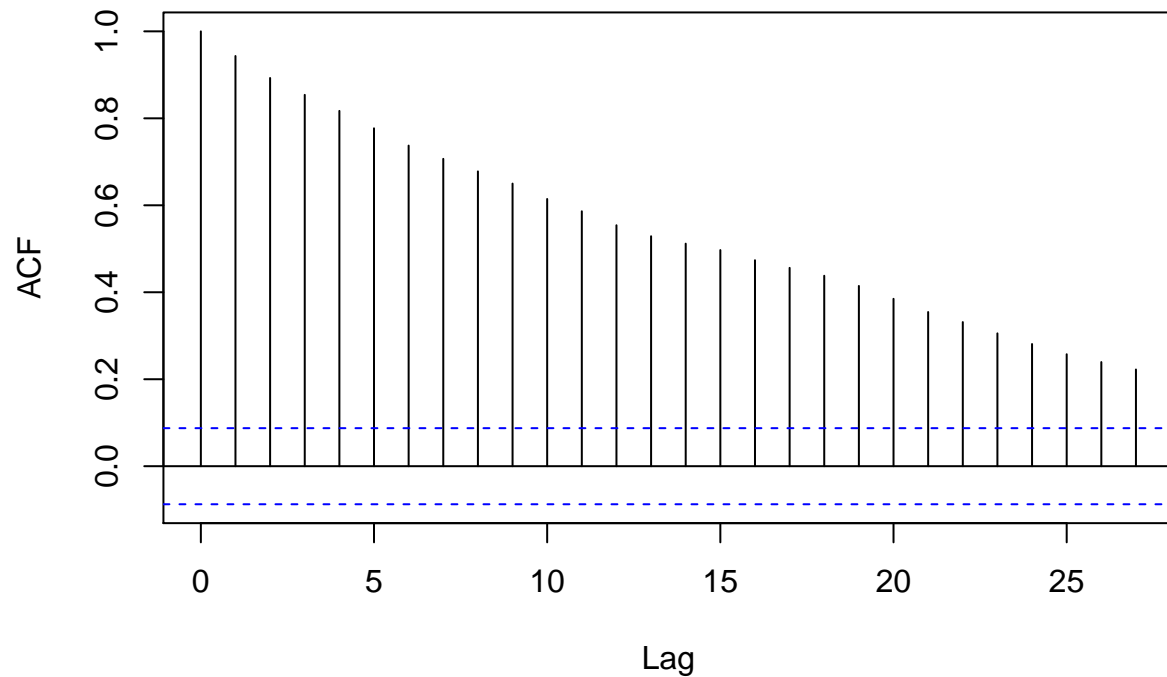
- Test Breuscha-Godfrey

```
bgtest <- bgtest(mod3, order.by = ~t, order = 3)$p.value
```

- Wykres ACF (funkcji autokorelacji)

```
acf(mod3$residuals, main = "Wykres funkcji autokorelacji")
```


Wykres funkcji autokorelacji

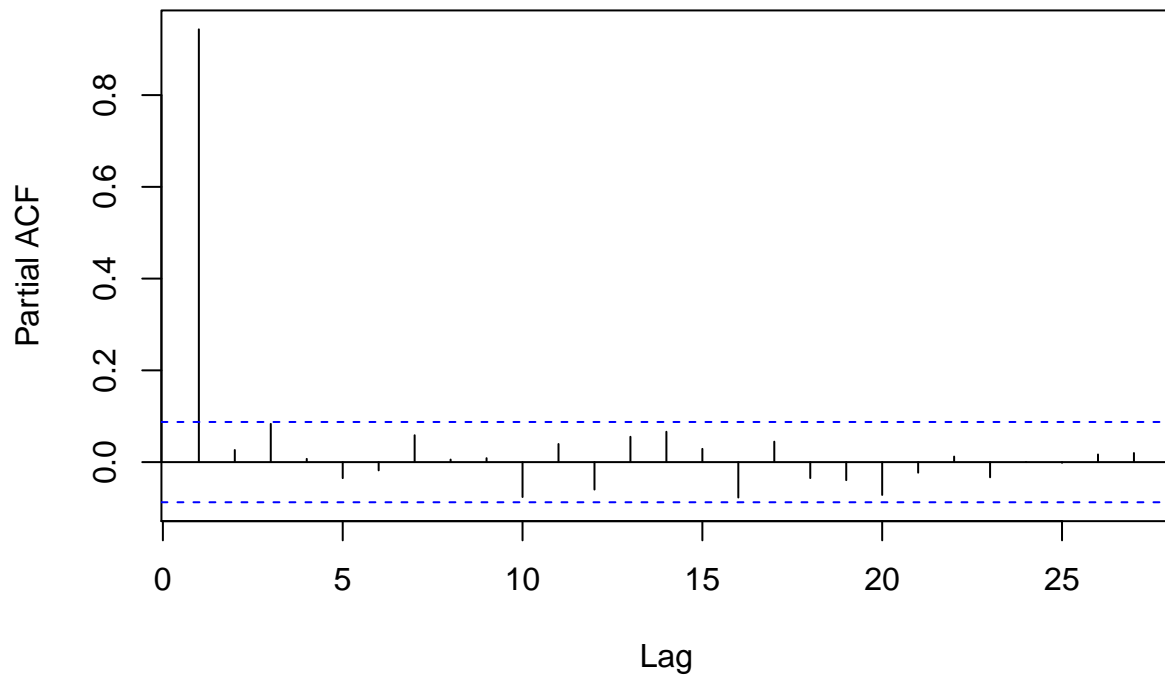


Oba testy wskazują, że istnieje seryjna korelacja błędów, p-value w obu przypadkach jest praktycznie równe zero. Również z wykresu funkcji `acf` możemy wyciągnąć te same wnioski. Dla opóźnień do rzędu 25 obserwacji nie mieszczą się w niebieskich przerywanych liniach - wnioskujemy, że pojawia się autokorelacja.

- Wykres PACF (częstkowej funkcji autokorelacji)

```
pacf(mod3$residuals, main = "Częstkowa funkcja autokorelacji")
```

Czastkowa funkcja autokorelacji



Wykres PACF jako jedyny nie wykrywa autokorelacji.

- Test Ljunga-Boxa

H_0 : niezależność

H_1 : brak niezależności

```
lbox <- Box.test(mod3$residuals, type = "Ljung-Box")$p.value
```

P-value jest bardzo bliskie zero, odrzucamy hipotezę zerową, reszty nie są niezależne.

Ostatecznie stwierdzamy, że zachodzi autokorelacja reszt.

Stacjonarność

Zbadamy, czy szereg jest stacjonarny albo TS (trend stationary).

Skorzystamy z dwóch testów :

- `adf` (Dickey-Fullera)

H_0 : niestacjonarność

H_1 : stacjonarność

- `kpss` (Kwiatkowskiego-Phillipsa-Schmidta-Shina)

H_0 : stacjonarność

H_1 : niestacjonarność

```
adf <- adf.test(mcd)$p.value
kpss <- kpss.test(mcd)$p.value
kpsst <- kpss.test(mcd, null = "Trend")$p.value
```

Po wykonaniu testów otrzymujemy wniosek, że szereg nie jest stacjonarny, ani stacjonarny wobec trendu.

```
adf2 <- adf.test(diff(mcd, differences = 1))$p.value
kpss2 <- kpss.test(diff(mcd, differences = 1))$p.value
kpsst2 <- kpss.test(diff(mcd, differences = 1), null = "Trend")$p.value
```

Po zróżnicowaniu 1 raz, szereg jest zarówno stacjonarny, jak i TS.

```
n <- ndiffs(mcd)
```

Liczba różnicowań uzyskana za pomocą funkcji `ndiffs` wynosi 1, co zgadza się z wcześniejszymi wnioskami.

ARIMA

Nasz szereg jest niestacjonarny, więc spróbujemy do niego dopasować model ARIMA.

```
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
```

Według funkcji `auto.arima`, najlepszy model dla badanego szeregu czasowego to ARIMA(0, 1, 0), co oznacza, że nie ma składnika ani AR (Auto-Regressive), ani MA (Moving Average), a jedynie należy ten szereg zróżnicować jeden raz.

```
mcd_arima <- arima(mcd, c(0,1,0))
```

Kryterium Akaike dla modelu ARIMA(0,1,0) wynosi 2139.8741327, a wariancja 4.1759287

Na wszelki wypadek sprawdzimy dopasowanie kilku innych modeli ARIMA(p , r , q). Przyjmujemy $r = 1$, jako że wcześniej otrzymaliśmy, że szereg należy zróżnicować jeden raz. Parametry p i q będą się zmieniać w pętłach od 0 do 3. Chcemy znaleźć najlepszy model ze względu na kryterium Akaike (tj. szukamy jak najmniejszego `aic`).

```
akaike <- NULL
for (p in 0:3){
  akaike1 <- NULL
  for (q in 0:3){
    akaike1 <- c(akaike1, arima(mcd, c(p,1,q))$aic)
  }
  akaike <- rbind(akaike, akaike1)
}
akaike <- as.data.frame(akaike)
colnames(akaike) <- c("q=0", "q=1", "q=2", "q=3")
rownames(akaike) <- c("p=0", "p=1", "p=2", "p=3")

kable(akaike) %>%
  kable_styling(full_width = T)
```

	q=0	q=1	q=2	q=3
p=0	2139.874	2140.328	2138.043	2140.027
p=1	2140.623	2139.634	2140.032	2141.989
p=2	2137.953	2139.850	2140.285	2141.628
p=3	2139.795	2141.713	2141.873	2141.595

Z powyższej tabeli widać, że faktycznie najlepszym modelem jest ARIMA(0, 1, 0).

W takim razie nasz szereg można by zapisać jako:

$$\Delta \varepsilon_t = \varepsilon_t$$

gdzie $\varepsilon_t \sim N(0, 4.176)$.

Testy na resztach modelu ARIMA Dla reszt otrzymanego modelu ARIMA można przeprowadzić testy na jednorodność wariancji oraz normalność rozkładu. Tak jak poprzednio, posłużymy się testami Breuscha-Pagana, Goldfelda-Quandt i Harrisona-McCabe'a dla zbadania jednorodności reszt oraz Kołmogorowa-Smirnova, Lillieforsa-Smirnova, Shapiro-Wilka i Andersona-Darlinga dla zbadania normalności.

NIE WIEM CZEMU, ALE TESTY NA JEDNORODNOŚĆ NIE CHCĄ MI PÓJŚĆ, NIE MAM JUŻ SIŁY OGARNIAĆ CO MU NIE PASUJE.

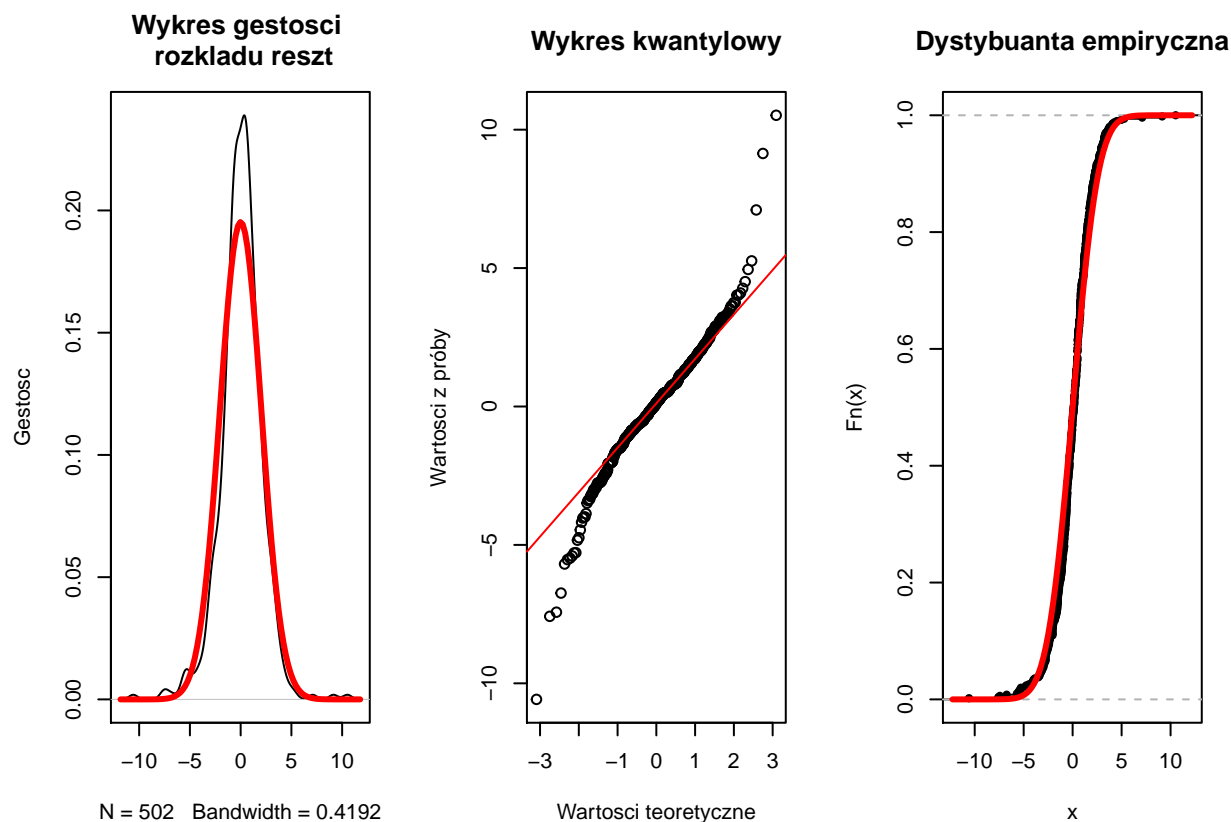
```
#library(car)
#leveneTest(as.numeric(mcd_arima$fitted))
#bptest(mcd_arima)
#ggtest(mcd_arima)
#hmctest(mcd_arima)
```

- Normalność

```
par(mfrow = c(1,3))
plot(density(mcd_arima$residuals), main = "Wykres gęstości \n rozkładu reszt",
     ylab = "Gęstość")
curve(dnorm(x,0, sd = sd(mcd_arima$residuals)), col = 2, add = T, lwd = 3)

qqnorm(mcd_arima$residuals, main = "Wykres kwantylowy", xlab = "Wartości teoretyczne",
       ylab = "Wartości z próby")
qqline(mcd_arima$residuals, col = 2)

plot(ecdf(mcd_arima$residuals), main = "Dystybuanta empiryczna")
curve(pnorm(x,0, sd = sd(mcd_arima$residuals)), add = T, lwd = 3, col = 2)
```



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

Z wykresu gęstości rozkładu reszt można wnioskować, że prawdopodobnie ten rozkład charakteryzuje się dużą kurtozą, co powoduje jego “wyciągnięcie” w górę, natomiast z wykresu kwantylowego jasno wynika, że rozkład nie może być normalny.

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

```
##
## One-sample Kolmogorov-Smirnov test
##
## data: mcd_arima$residuals
## D = 0.076145, p-value = 0.005928
## alternative hypothesis: two-sided
```

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

```
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: mcd_arima$residuals
## D = 0.068073, p-value = 8.431e-06
```

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

```
##
## Shapiro-Wilk normality test
##
## data: mcd_arima$residuals
```

```
## W = 0.95335, p-value = 1.7e-11
```

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

```
##
```

```
## Anderson-Darling normality test
```

```
##
```

```
## data: mcd_arima$residuals
```

```
## A = 3.8694, p-value = 1.189e-09
```

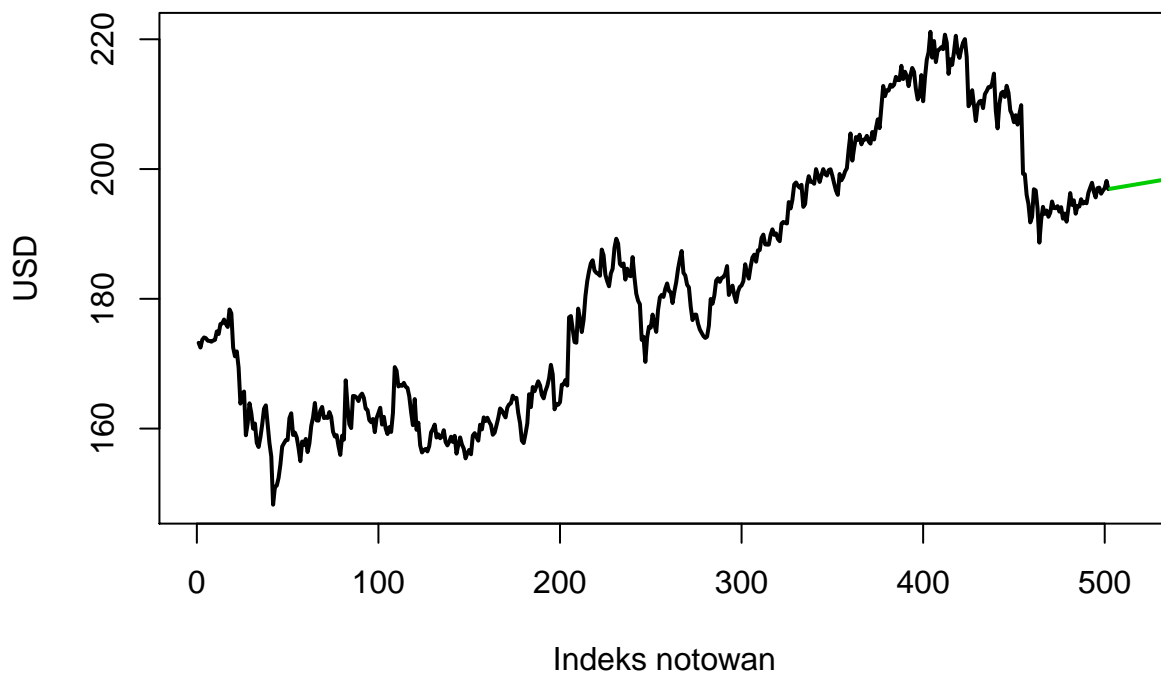
Wszystkie testy jednoznacznie każą odrzucić hipotezę o normalności rozkładu reszt modelu ARIMA, co potwierdza wnioski, które nasuwają się patrząc na wykresy.

Prognozowanie

Metoda dryftu Ponieważ badany szereg czasowy nie wykazuje sezonowości, nie możemy zatem przeprowadzić predykcji metodą Holta-Wintersa. Zamiast tego użyjemy metody dryftu, będącej odmianą metody naiwnej. Dzięki niej można zaobserwować trend.

```
progniza <- rwf(mcd, drift = T, h = 30)
progniza <- as.data.frame(progniza)[,1]
plot(mcd, type = "l", main = "Prognoza z całości danych na najbliższe 30 notowań",
     xlim = c(0,515), xlab = "Indeks notowań", ylab = "USD", lwd = 2)
t1 <- 503:(503+length(progniza)-1)
lines(t1, progniza, col = 3, lwd = 2)
```

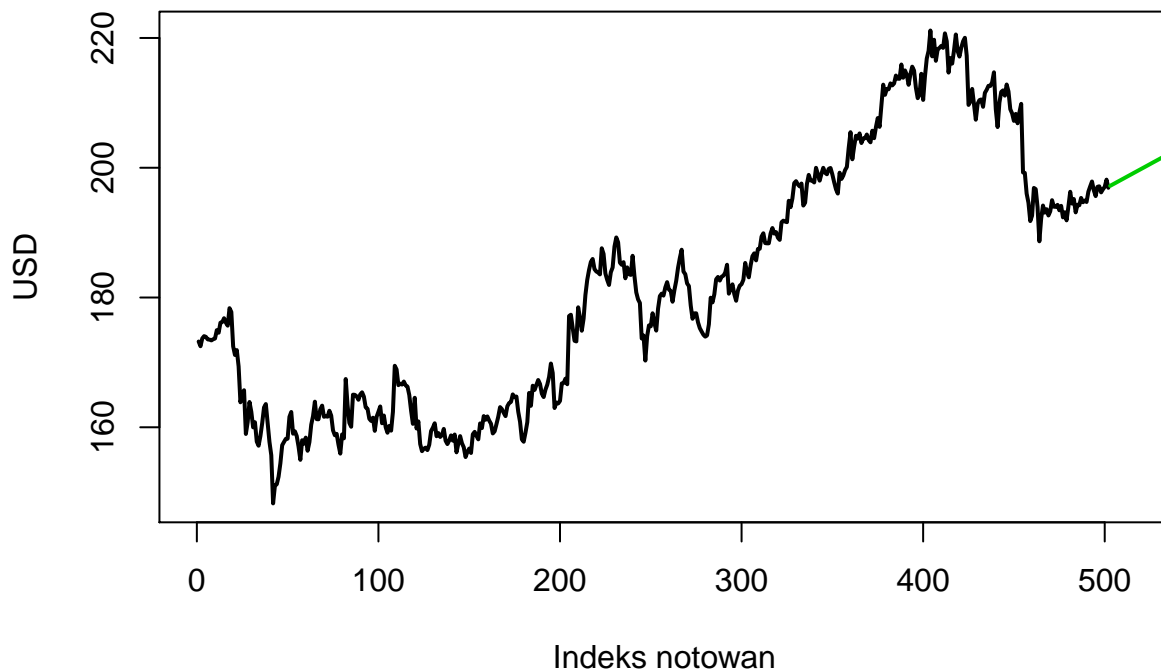
Prognoza z całości danych na najbliższe 30 notowań



```
progniza <- rwf(mcd[480:500], drift = T, h = 30)
progniza <- as.data.frame(progniza)[,1]
```

```
plot(mcd, type = "l", main = "Prognoza z ostatnich 20 notowań na najbliższe 30",
     xlim = c(0,515), xlab = "Indeks notowań", ylab = "USD", lwd = 2)
t1 <- 503:(503+length(prognoza)-1)
lines(t1, prognoza, col = 3, lwd = 2)
```

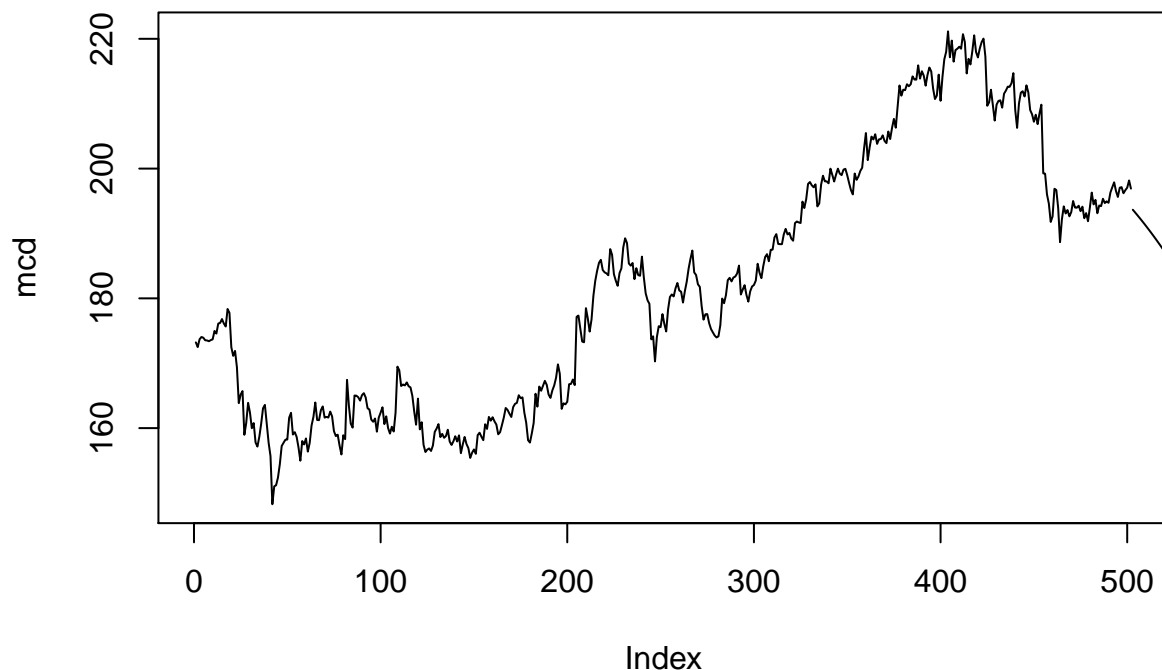
Prognoza z ostatnich 20 notowań na najbliższe 30



Z racji braku sezonowości, prognoza nie uwzględnia wahań, a jedynie ogólnie występujący trend. Z tego względu jest dość niedokładna i bardziej wiarygodne wyniki można uzyskać prognozując na krótsze odcinki czasu. Widać to z porównania powyższych wykresów - nachylenie linii prognozy, biorącej pod uwagę wszystkie wartości notowań jest mniejsze i nie uwzględnia istotnych zmian, które zaszły pomiędzy pierwszym a ostatnim notowaniem.

Przewidywania z modelu Próbuje przewidywać z modelu wielomianowego trzeciego stopnia.

```
wsp <- mod3$coefficients
funkcja <- function(t){
  wsp[1]+wsp[2]*t+wsp[3]*t^2+wsp[4]*t^3
}
t2 <- 503:520
plot(mcd, type = "l")
lines(t2, funkcja(t2))
```



Podsumowanie analizy notowań McDonald's

Zrobiliśmy analizę. Nie umiemy jej zinterpretować. Jest fajnie.

Starbucks

Opis firmy

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

Wczytanie danych i rysunki

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

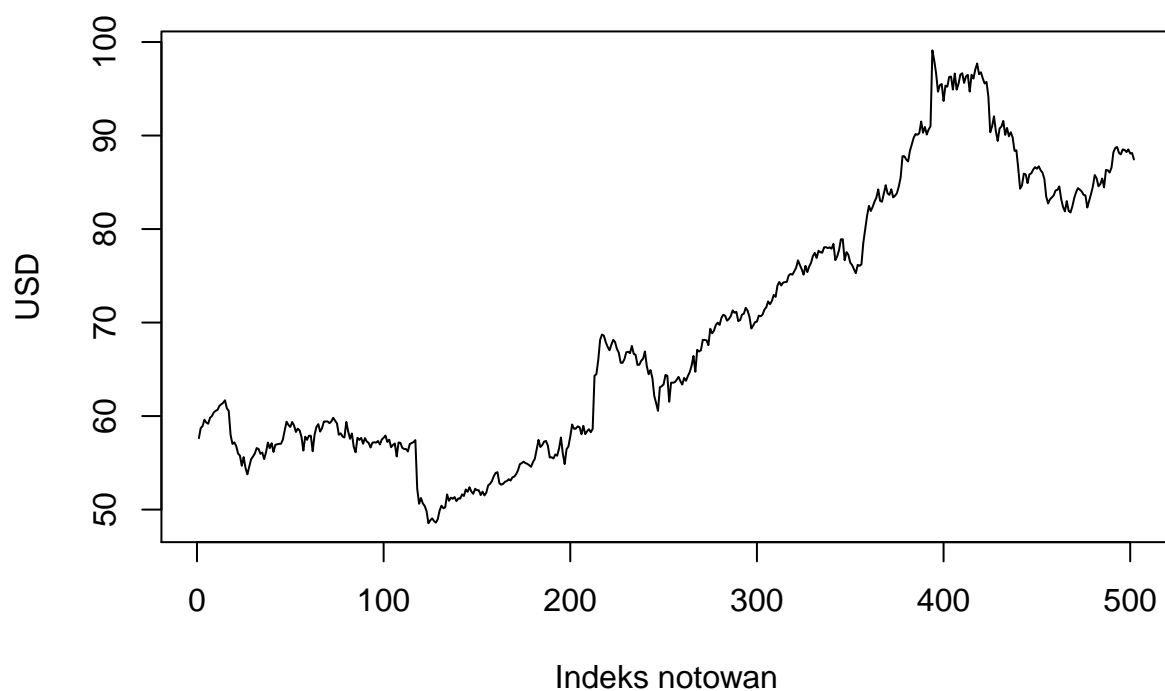
```
## time series starts 2018-01-02
## time series ends   2019-12-30
```

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

Wykonamy rysunek przedstawiający notowania firmy Starbucks z ostatnich dwóch lat.

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


Notowania Starbucks



Na rysunku możemy zaobserwować trend rosnący.

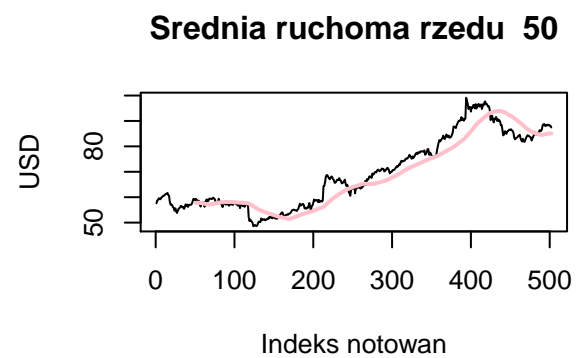
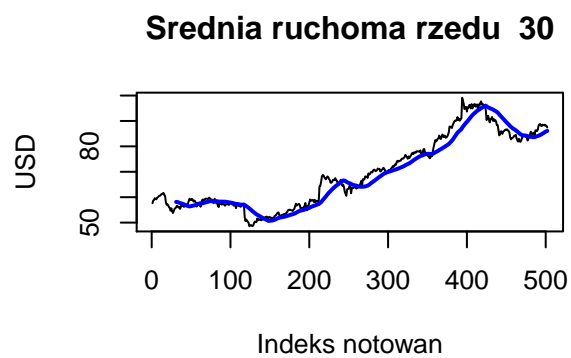
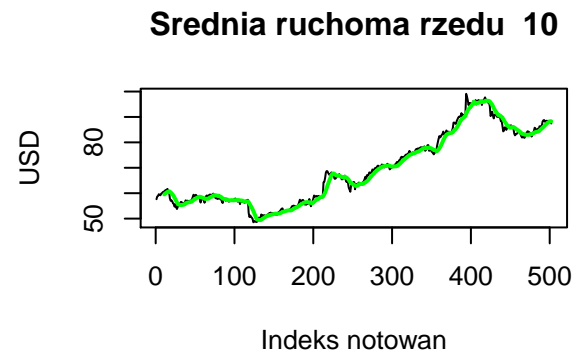
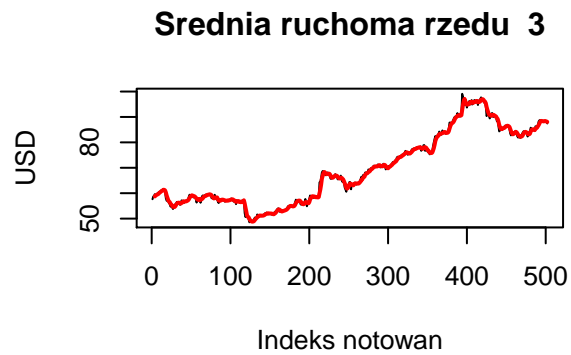
Wydzielanie trendu

Wydzielamy część deterministyczną. Zastosujemy metody średnich ruchomych (prostą i wykładniczą).

Ruchoma średnia

Skorzystamy z napisanej wcześniej funkcji `ruchoma`. Narysujemy wykresy dla kilku parametrów `m`.

```
par(mfrow=c(2,2))
ruchoma(sbox, 3, "red")
ruchoma(sbox, 10, "green")
ruchoma(sbox, 30, "blue")
ruchoma(sbox, 50, "pink")
```

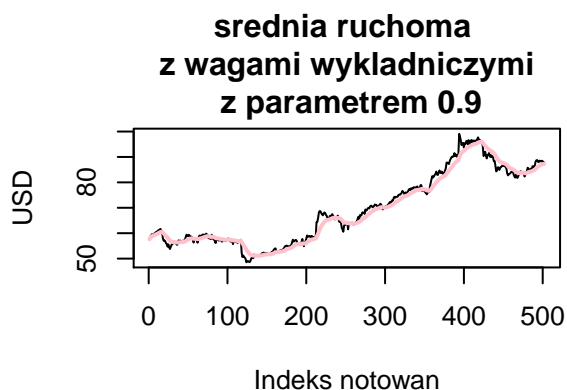
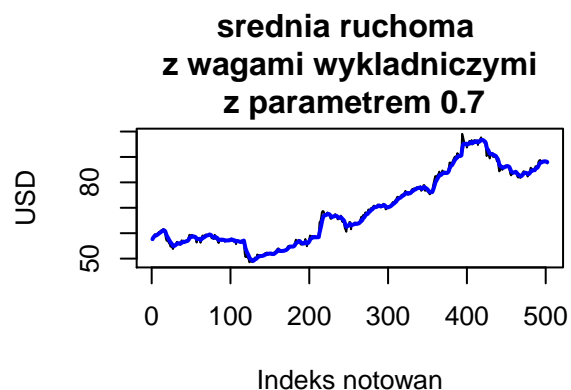
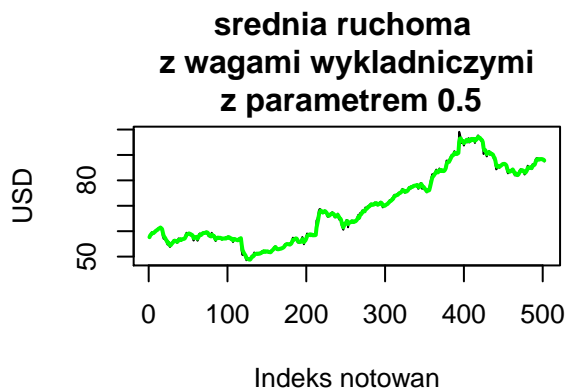
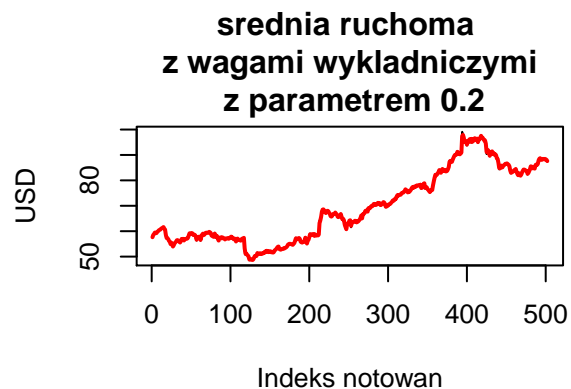


```
par(mfrow=c(1,1))
```

Wraz ze wzrostem parametru m otrzymujemy bardziej wygładzony wykres, ale niestety mniej dokładny.

Metoda wykładniczych wag ruchomej średniej

```
par(mfrow=c(2,2))
wykladnicza(sbox, 0.2, "red")
wykladnicza(sbox, 0.5, "green")
wykladnicza(sbox, 0.7, "blue")
wykladnicza(sbox, 0.9, "pink")
```



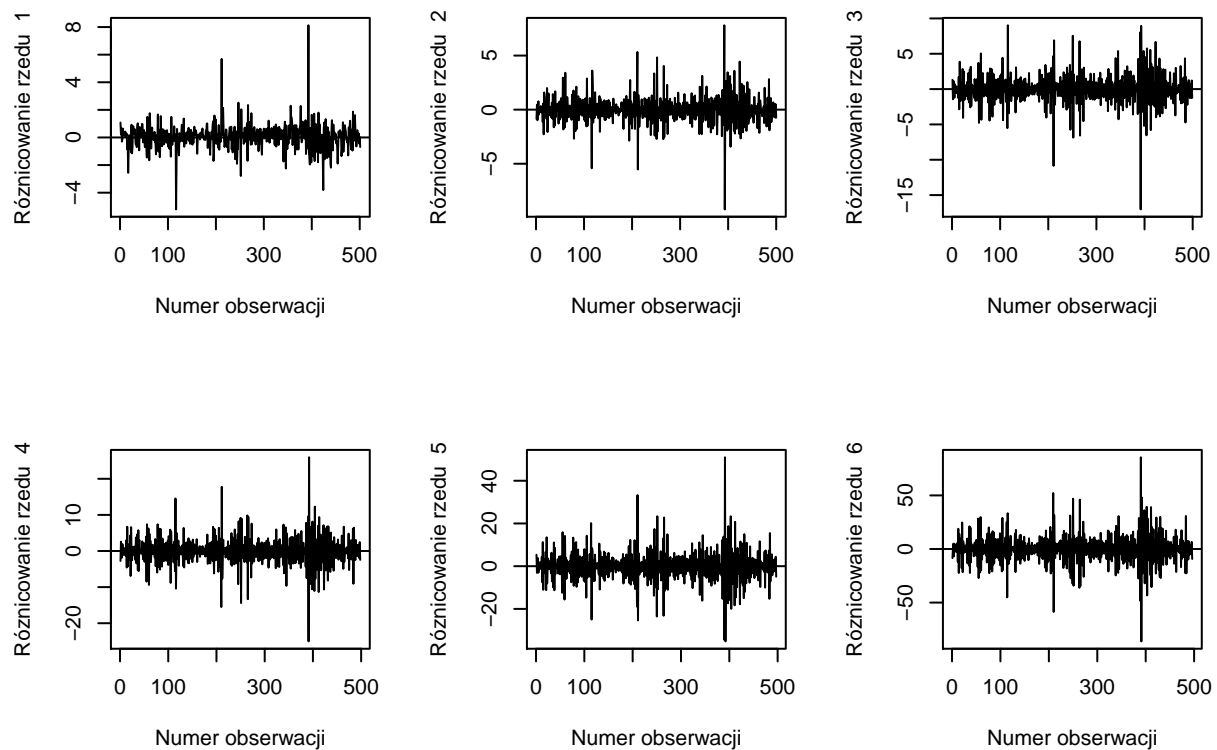
```
par(mfrow=c(1,1))
```

Wraz ze wzrostem parametru η uzyskujemy bardziej wygładzony wykres, ale mniejszą dokładność. Jest ona jednak większa niż w przypadku prostej metody ruchomej średniej.

Dopasowanie wielomianu

Metoda różnicowa

```
par(mfrow = c(2, 3))
for(i in 1:6){
  plot(diff(sbox, differences = i), type = "l",
       xlab = "Numer obserwacji", ylab = paste("Różnicowanie rzędu ", i))
  abline(h = 0)}
```



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

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

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

```
mod1 <- lm(sbox~t)
```

```
stargazer(mod1, header = F)
```

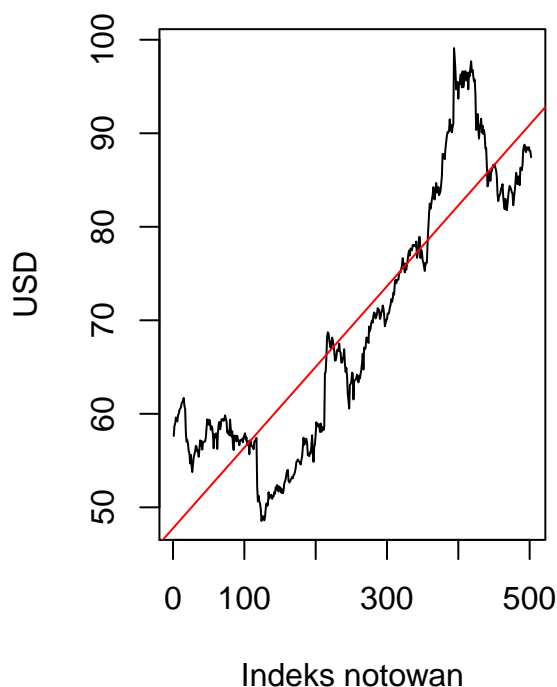
Dopasowanie modelu liniowego Wszystkie współczynniki są istotne statystycznie. R^2 wynosi około 79%.

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

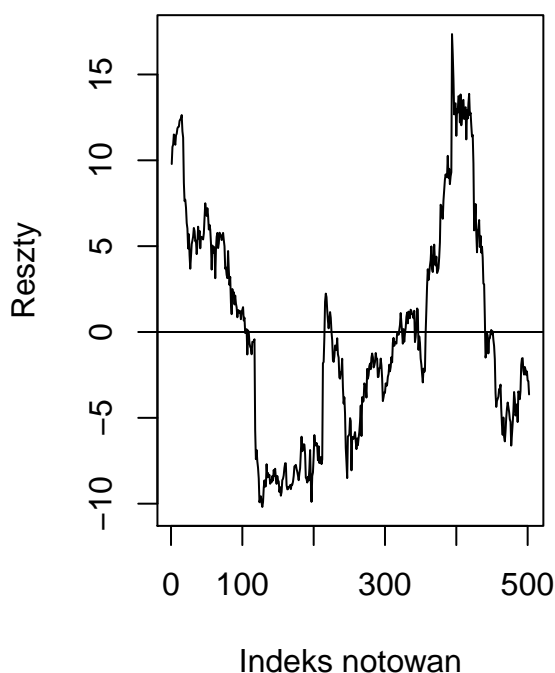
Tablica 5:

<i>Dependent variable:</i>	
sbux	
t	0.086*** (0.002)
Constant	47.757*** (0.569)
Observations	502
R ²	0.795
Adjusted R ²	0.794
Residual Std. Error	6.366 (df = 500)
F Statistic	1,937.353*** (df = 1; 500)
<i>Note:</i> *p<0.1; **p<0.05; ***p<0.01	

Model liniowy



Reszty modelu liniowego



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

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

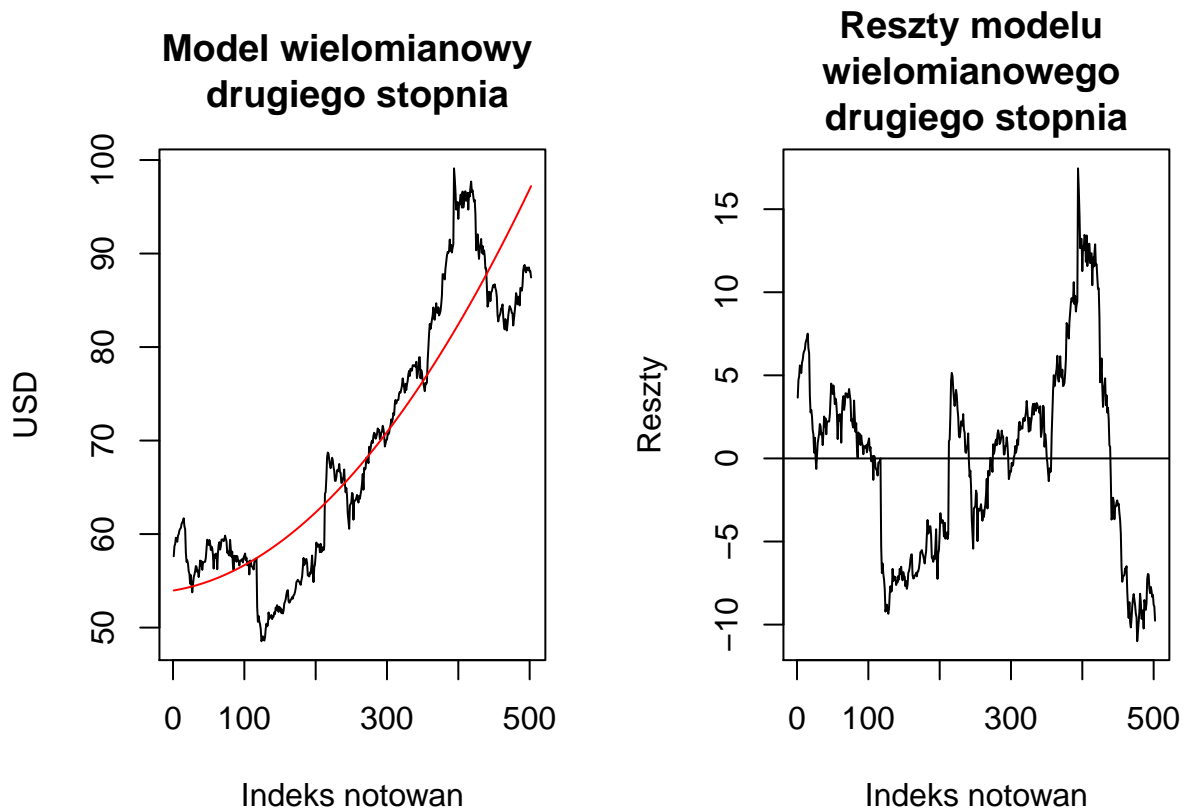
```
stargazer(mod2, header = F)
```

Tablica 6:

	<i>Dependent variable:</i>
	sbux
t	0.012* (0.007)
I(t^2)	0.0001*** (0.00001)
Constant	53.960*** (0.772)
Observations	502
R ²	0.834
Adjusted R ²	0.833
Residual Std. Error	5.740 (df = 499)
F Statistic	1,249.086*** (df = 2; 499)
<i>Note:</i>	*p<0.1; **p<0.05; ***p<0.01

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

```
par(mfrow = c(1, 2))
plot(sbx, type = "l", main = "Model wielomianowy \n drugiego stopnia",
     xlab = "Indeks notowań", ylab = "USD")
lines(t, mod2$fitted.values, col = "red")
plot(mod2$residuals, type = "l", main = "Reszty modelu \nwielomianowego \ndrugiego stopnia",
     xlab = "Indeks notowań", ylab = "Reszty")
abline(h = 0)
```



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

Wykres reszt jest bardzo podobny jak w przypadku modelu liniowego.

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

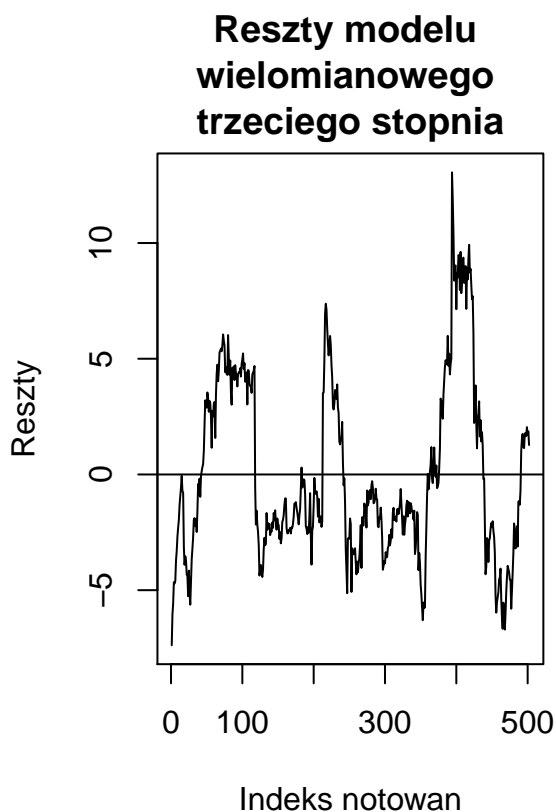
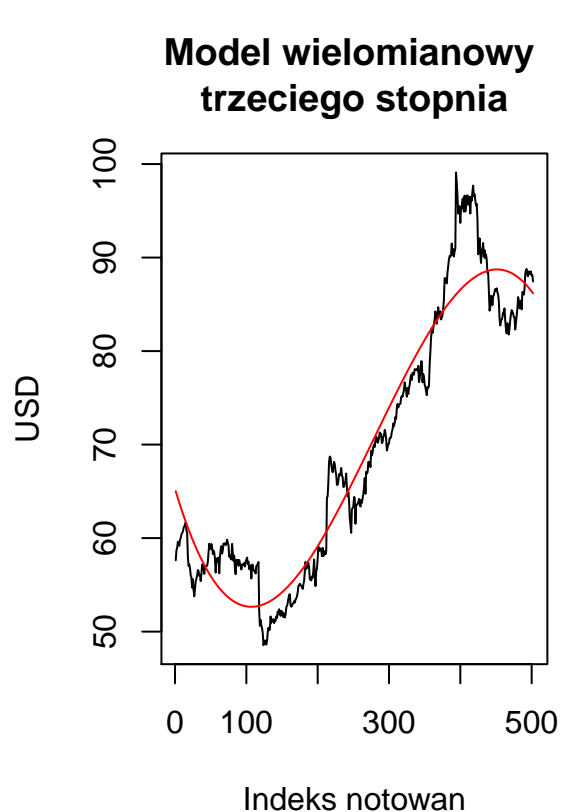
```
stargazer(mod3, header = F)
```

Dopasowanie wielomianu trzeciego stopnia Wszystkie współczynniki są istotne statystycznie, a R^2 znów wzrosło - wynosi około 92% (znaczną poprawę).

```
par(mfrow = c(1,2))
plot(sbox, type = "l", main = "Model wielomianowy \ntrzeciego stopnia",
     xlab = "Indeks notowań", ylab = "USD")
lines(t, mod3$fitted.values, col = "red")
plot(mod3$residuals, type = "l", main = "Reszty modelu \nwielomianowego \ntrzeciego stopnia",
     xlab = "Indeks notowań", ylab = "Reszty")
abline(h= 0)
```

Tablica 7:

<i>Dependent variable:</i>	
sbux	
t	-0.256*** (0.012)
I(t ²)	0.001*** (0.0001)
I(t ³)	-0.00000*** (0.00000)
Constant	65.265*** (0.698)
Observations	502
R ²	0.924
Adjusted R ²	0.924
Residual Std. Error	3.879 (df = 498)
F Statistic	2,021.491*** (df = 3; 498)
<i>Note:</i> *p<0.1; **p<0.05; ***p<0.01	




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

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

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

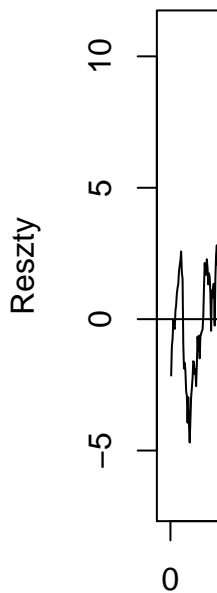
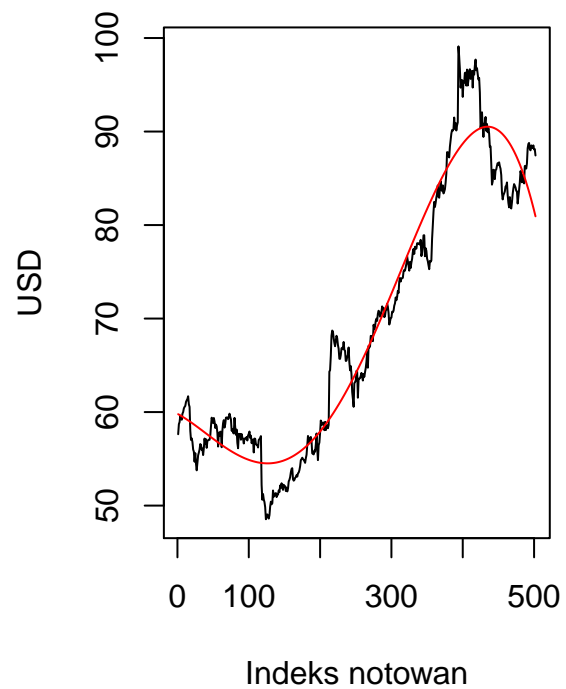
```
stargazer(mod4, header = F)
```

Tablica 8:

	<i>Dependent variable:</i>
	sbox
t	-0.041* (0.021)
I(t^2)	-0.0004** (0.0002)
I(t^3)	0.00000*** (0.00000)
I(t^4)	-0.000*** (0.000)
Constant	59.817*** (0.778)
Observations	502
R ²	0.940
Adjusted R ²	0.940
Residual Std. Error	3.447 (df = 497)
F Statistic	1,954.177*** (df = 4; 497)
<i>Note:</i> *p<0.1; **p<0.05; ***p<0.01	

```
par(mfrow = c(1,2))
plot(sbox, type = "l", main = "Model wielomianowy \nczwartego stopnia",
     xlab = "Indeks notowań", ylab = "USD")
lines(t, mod4$fitted.values, col = "red")
plot(mod4$residuals, type = "l",
     main = "Reszty modelu \nwielomianowego \nczwartego stopnia",
     xlab = "Indeks notowań", ylab = "Reszty")
abline(h= 0)
```

Model wielomianowy czwartego stopnia

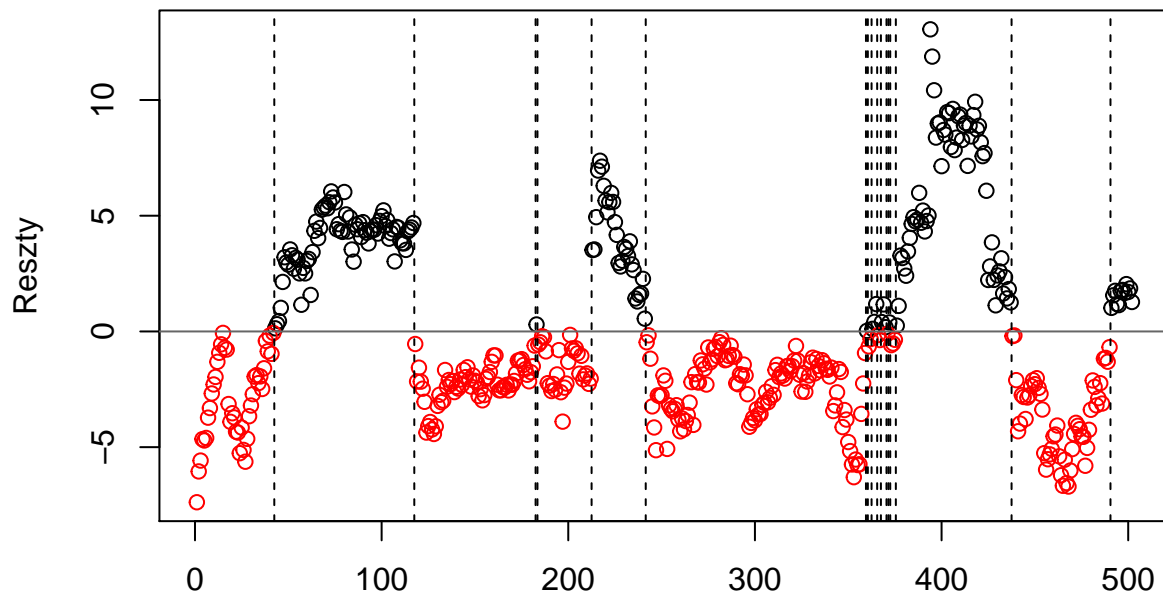


Dopasowanie wielomianu czwartego stopnia

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

Testy na resztach modelu

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



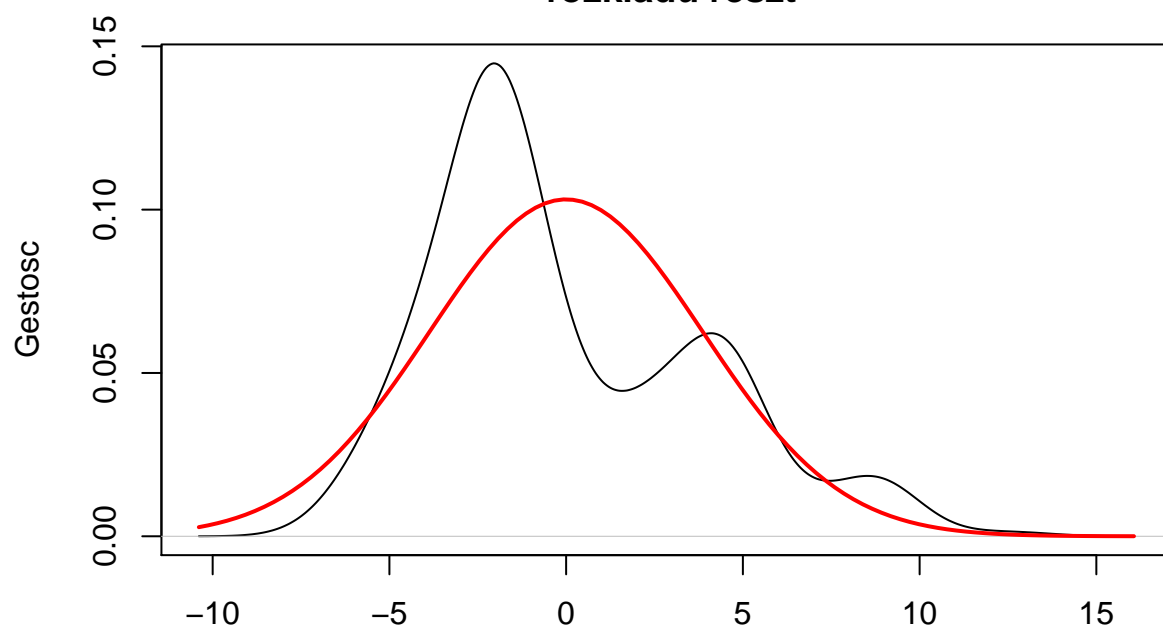
```
##
## Runs Test
##
## data: Reszty
## statistic = -20.806, runs = 18, n1 = 187, n2 = 315, n = 502, p-value <
## 2.2e-16
## alternative hypothesis: nonrandomness
```

P-value jest bliskie 0, odrzucamy hipotezę zerową o losowości reszt. Wykres także wskazuje na skorelowanie reszt.

Wykresy normalności

```
plot(density(mod3$residuals), main = "Wykres gęstości \n rozkładu reszt",
     ylab = "Gęstość")
curve(dnorm(x, 0, sd(mod3$residuals)), add = T, col = 2, lwd = 2)
```

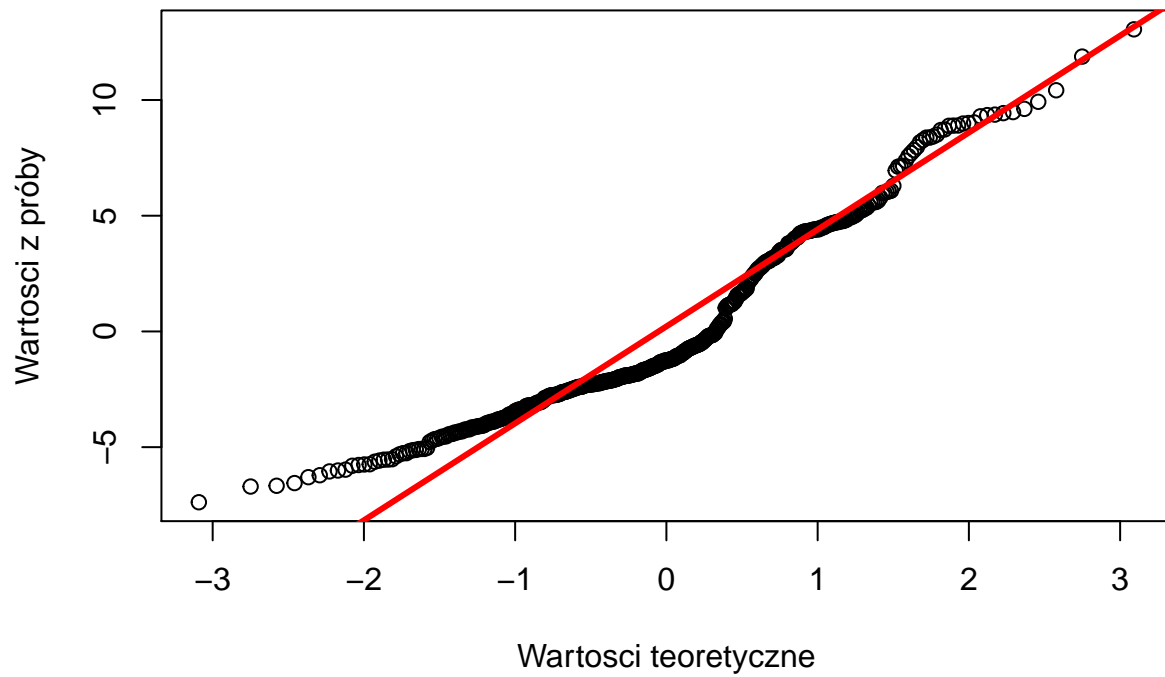
Wykres gestosci rozkladu reszt



N = 502 Bandwidth = 1.004

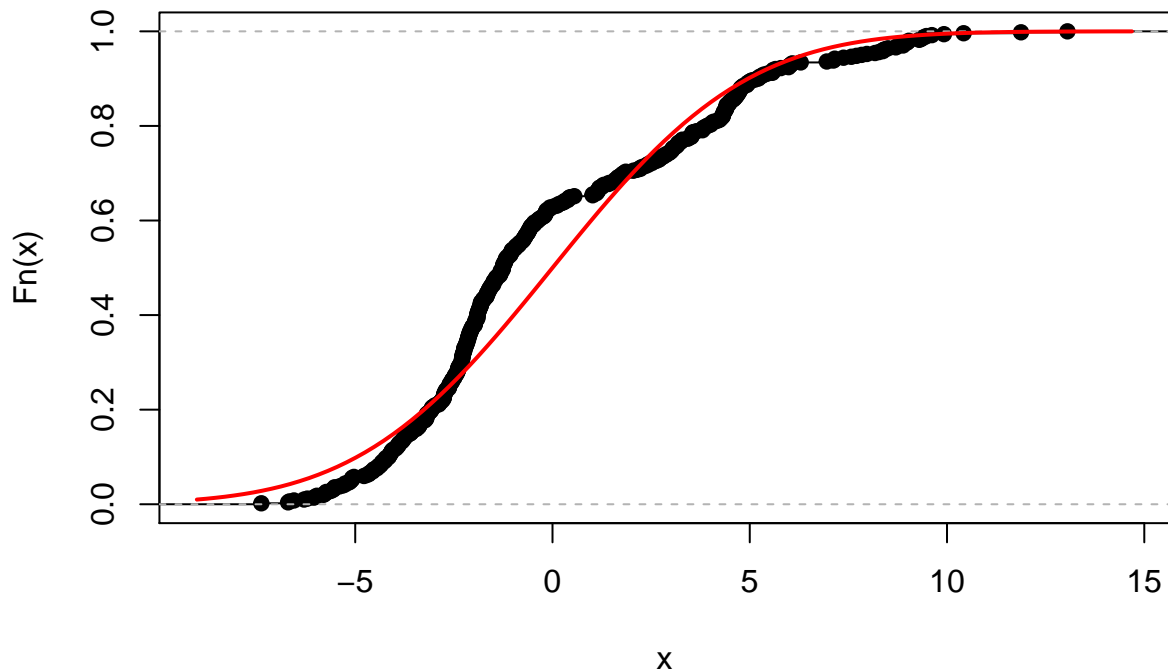
```
qqnorm(mod3$residuals, main = "Wykres kwantylowy",  
       xlab = "Wartości teoretyczne", ylab = "Wartości z próby")  
qqline(mod3$residuals, col=2, lwd = 3)
```

Wykres kwantylowy



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

Dystrybuanta empiryczna



Wykres gęstości rozkładu reszt bardzo znacząco różni się od gęstości rozkładu normalnego. Widać, że rozkład jest co najmniej dwumodalny i ma zbyt silną asymetrię prawostronną. Na wykresie kwantylowym widać znaczące odchyłki od linii kwantylowej. Podobne zjawisko można zaobserwować na wykresie dystrybuanty empirycznej.

```
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
```

```
##
```

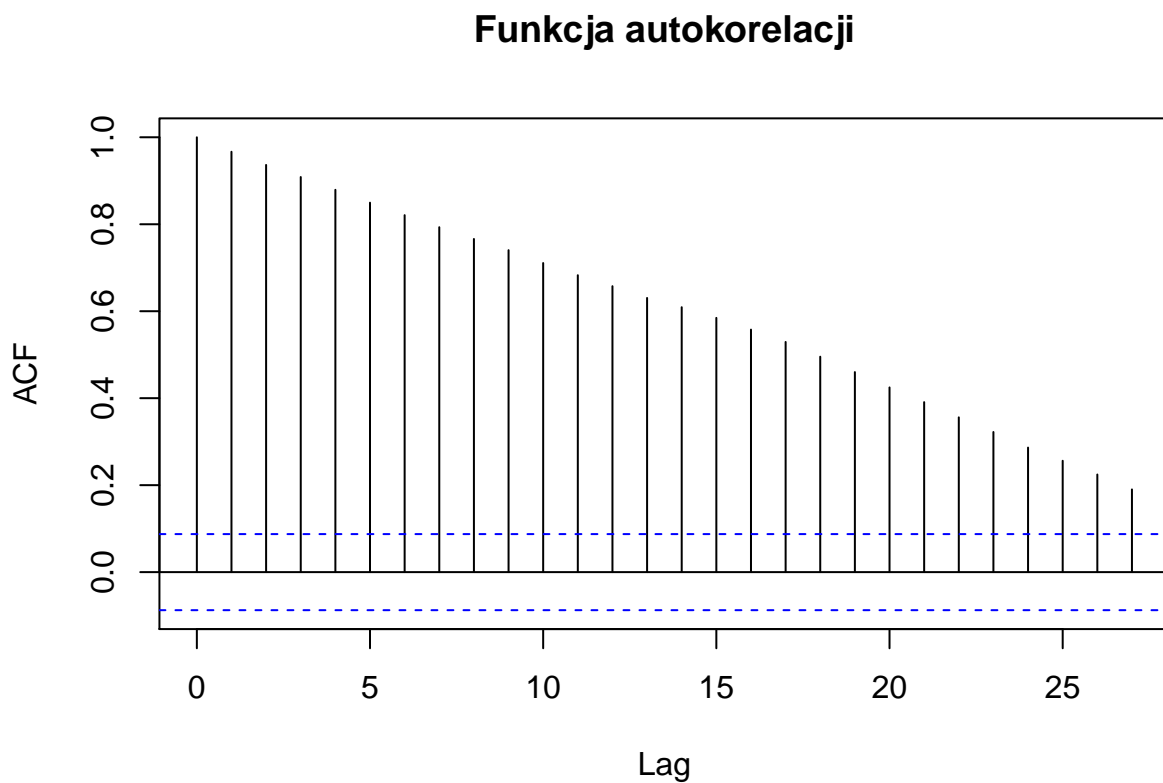
```
## data: mod3$residuals
```

```
## 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, main = "Funkcja autokorelacji")
```



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

Stacjonarność

```
adf.test(sbox)
```

```
##
```

```
## Augmented Dickey-Fuller Test
```

```
##
```

```
## data: sbox
```

```
## Dickey-Fuller = -2.1662, Lag order = 7, p-value = 0.5079
```

```
## alternative hypothesis: stationary
```

```
kpss.test(sbox)
```

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

```
kpss.test(sbox, null = "Trend")
```

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

Szereg nie jest ani stacjonarny ani TS.

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

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

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

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

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

```
##  
## KPSS Test for Trend Stationarity  
##  
## data: diff(sbox, 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(sbox)
```

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

Podsumowanie analizy notowań Starbucks