

Analiza szeregu czasowego

Magda Kozajda

2024-01-09

```
## Registered S3 method overwritten by 'quantmod':  
##   method           from  
##   as.zoo.data.frame zoo  
  
##  
## Dołączanie pakietu: 'astsa'  
  
## Następujący obiekt został zakryty z 'package:forecast':  
##  
##   gas  
  
##  
## Dołączanie pakietu: 'lubridate'  
  
## Następujące obiekty zostały zakryte z 'package:base':  
##  
##   date, intersect, setdiff, union  
  
##  
## Dołączanie pakietu: 'dplyr'  
  
## Następujące obiekty zostały zakryte z 'package:stats':  
##  
##   filter, lag  
  
## Następujące obiekty zostały zakryte z 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

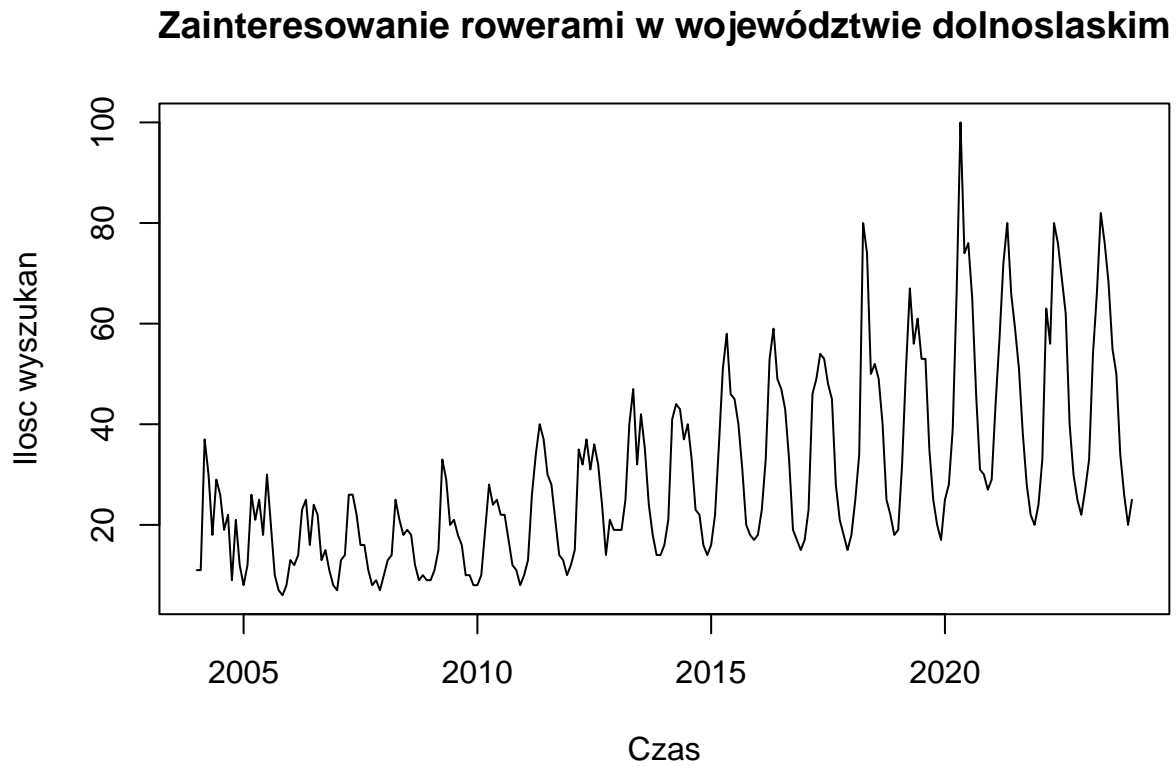
Wczytywanie danych do szeregu

```
data = read.csv('/Users/magda/Desktop/szeregi_czasowe2/rowery.csv', sep = ';')  
  
df = as.data.frame(data)  
time_series = ts(df$Number, frequency=12, start = c(2004,1))
```

Intuicja o danych

```
# Szereg czasowy
```

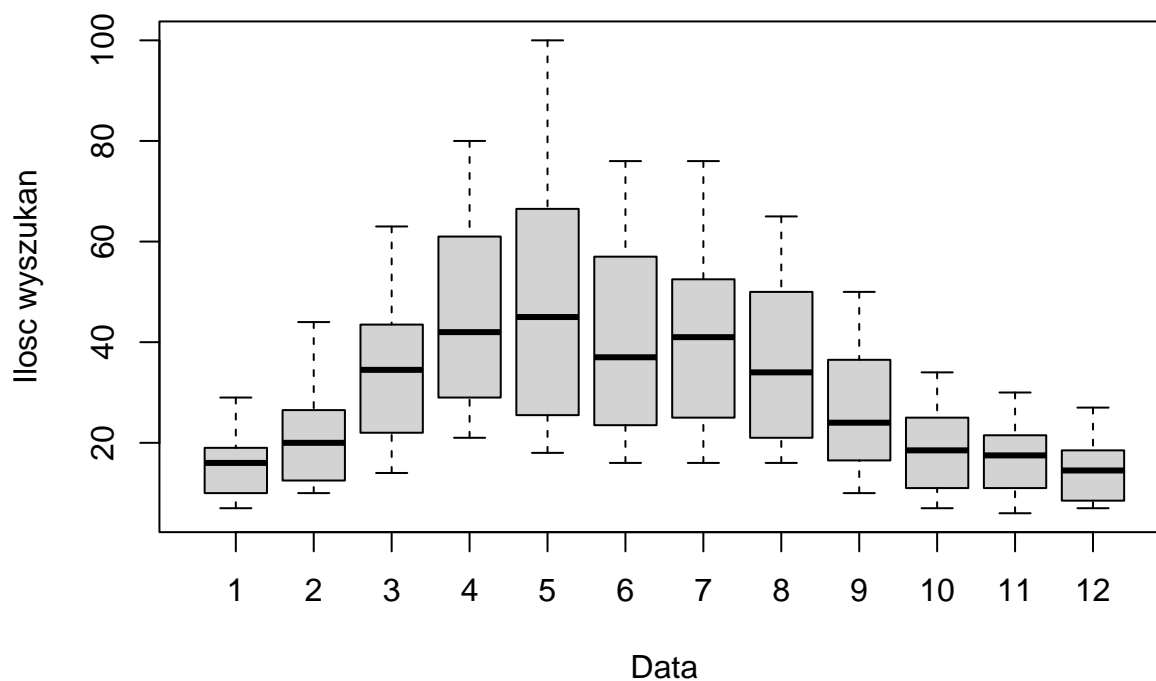
```
plot(time_series, main = 'Zainteresowanie rowerami w województwie dolnośląskim', xlab = 'Czas', ylab = 'Ilość wyszukiwań')
```



```
# Wartości dla kolejnych miesięcy
```

```
boxplot(time_series~cycle(time_series), xlab="Data", ylab = "Ilość wyszukiwań", main = "Boxplot popularności rowerów w województwie dolnośląskim")
```

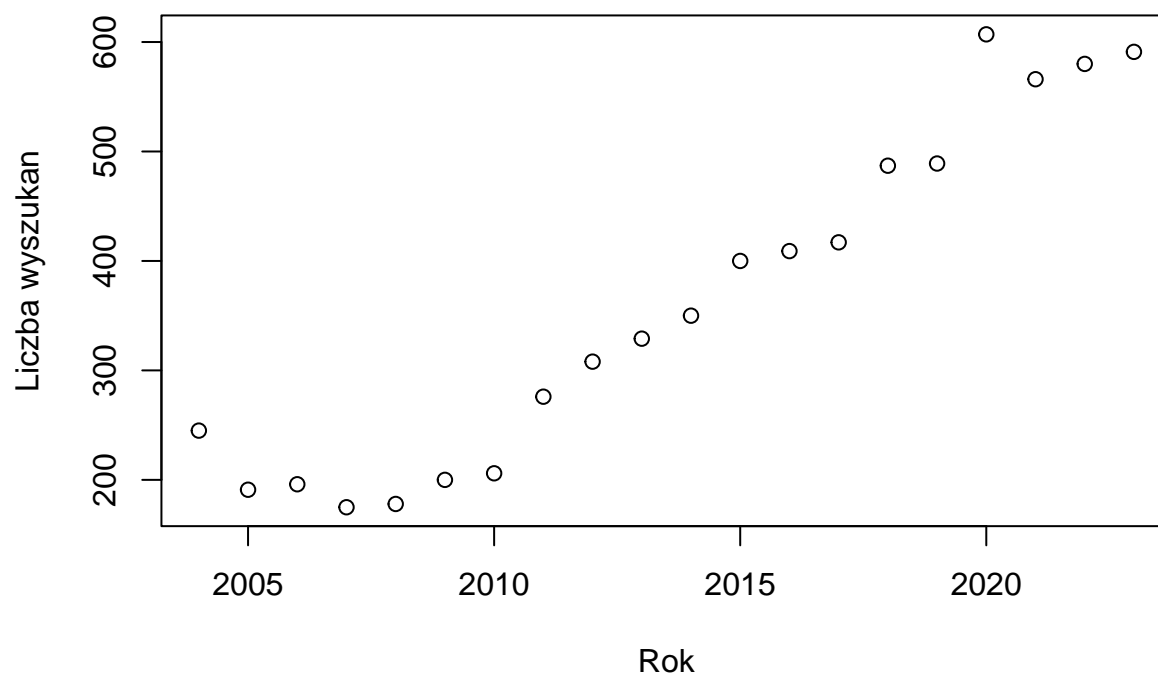
Boxplot popularności rowerów w zależności od miesiąca



```
# Suma dla poszczególnych lat
years = data.frame(substring(df$Date,1,4), df$Number)
colnames(years) = c('Year', 'Number')
year_sum = years %>%
  group_by(Year) %>%
  summarise(Suma = sum(Number))
year_sum = year_sum %>%
  filter(!row_number() %in% c(21))

plot(year_sum$Year, year_sum$Suma, main = 'Łączna liczba wyszukiwań dla poszczególnych lat',
      xlab = 'Rok', ylab='Liczba wyszukiwań')
```

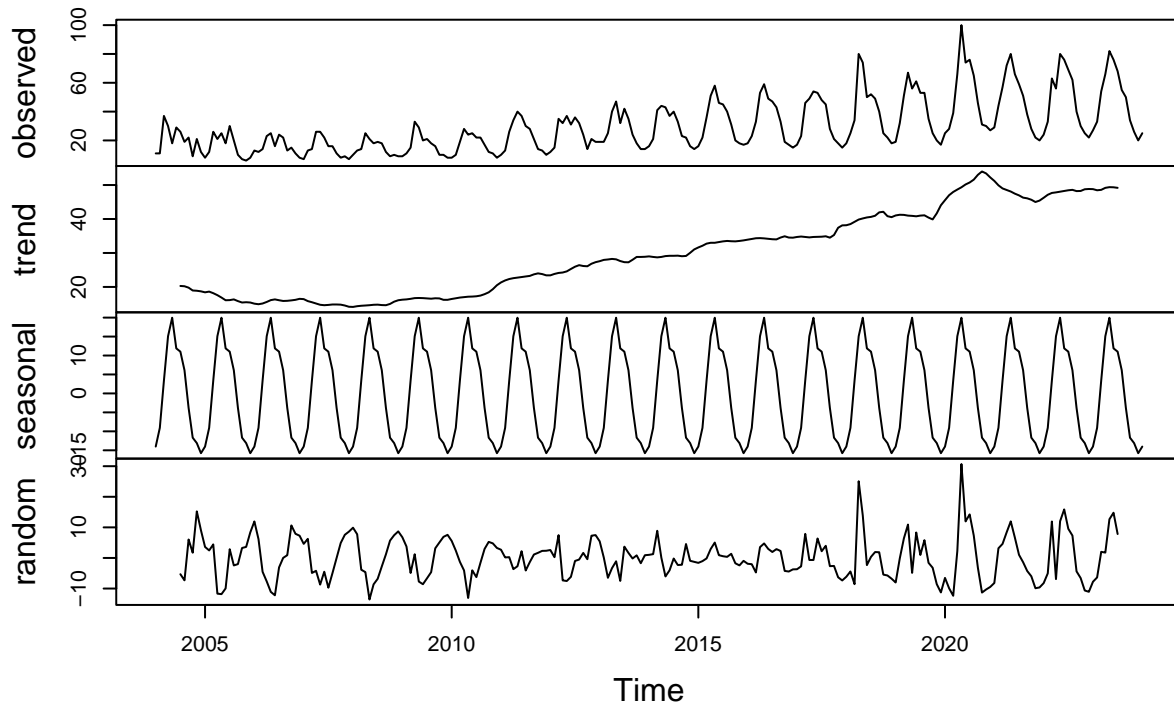
Laczna liczba wyszukan dla poszczególnych lat



Dekompozycja

```
dec = decompose(time_series)
plot(dec)
```

Decomposition of additive time series

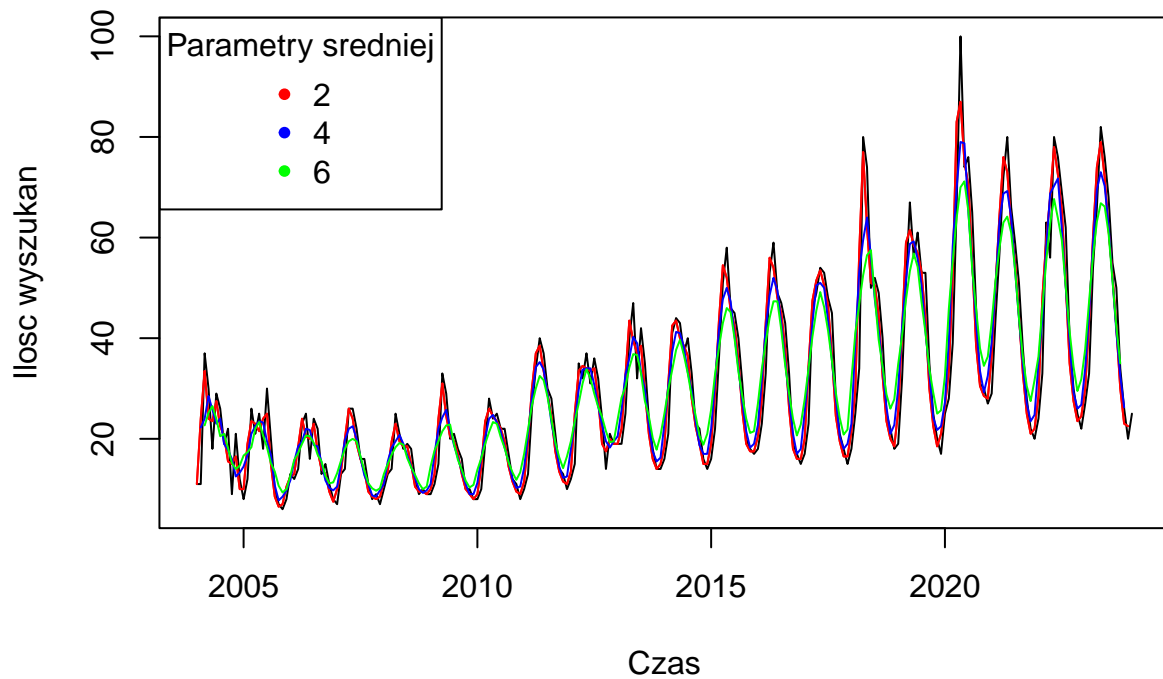


Wygładzanie szeregu

```
# wygładzanie - metoda średniej ruchomej
ts_filter = stats::filter(time_series, rep(1/2,2))

plot(time_series, main = 'Wygładzanie szeregu metodą średniej ruchomej', ylab = 'Ilość wyszukiń', xlab = 'Time')
lines(ts_filter, col = 'red')
lines(stats::filter(time_series, rep(1/2,2)), type='l', col = 'red')
lines(stats::filter(time_series, rep(1/4,4)), type='l', col = 'blue')
lines(stats::filter(time_series, rep(1/6,6)), type='l', col = 'green')
legend("topleft", legend = c(2, 4, 6), col = c('red', 'blue', 'green'), pch=20,
      title = 'Parametry średniej')
```

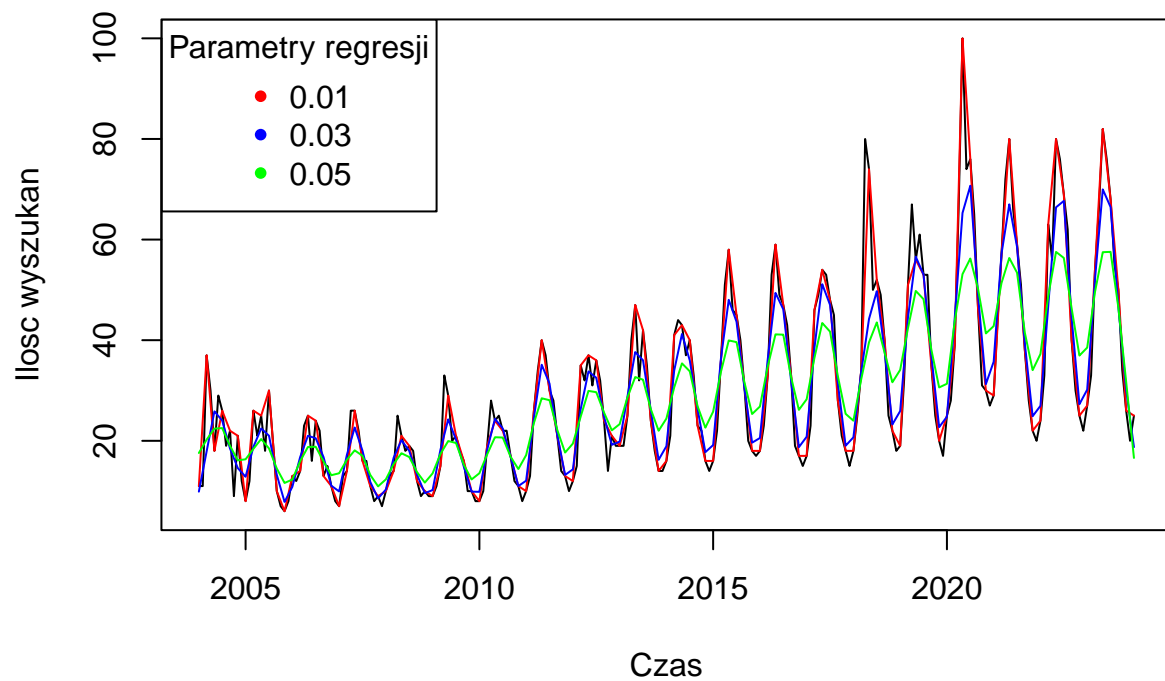
Wygladzanie szeregu metoda sredniej ruchomej



```
# wygladzanie - regresja nielokalna  
ts_lowess = lowess(time_series, f=0.01)
```

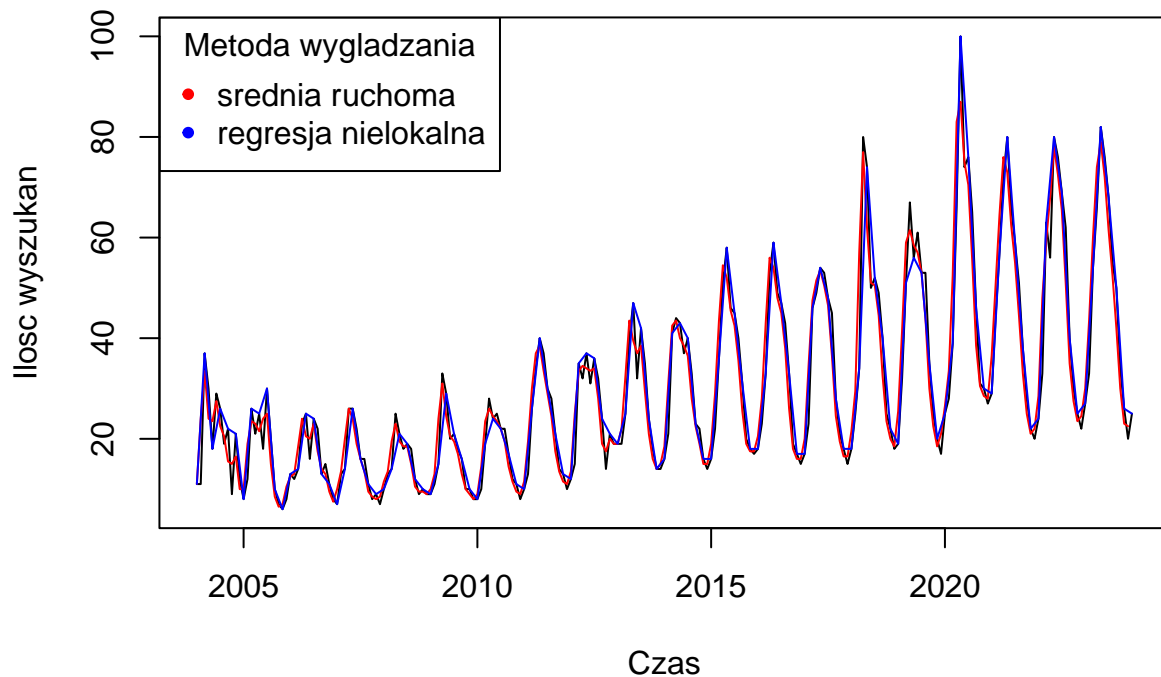
```
plot(time_series, main = 'Wygladzanie szeregu metoda regresji nielokalnej', ylab = 'Ilość wyszukan', xlab = 'Czas')  
lines(lowess(time_series, f=0.01), type='l', col = 'red')  
lines(lowess(time_series, f=0.03), type='l', col = 'blue')  
lines(lowess(time_series, f=0.05), type='l', col = 'green')  
legend("topleft", legend = c(0.01, 0.03, 0.05), col = c('red', 'blue', 'green'), pch=20,  
      title = 'Parametry regresji')
```

Wygladzanie szeregu metoda regresji nielokalnej



```
# porównanie obu metod
plot(time_series, main = 'Wygladzanie szeregu', ylab = 'Ilość wyszukan', xlab = 'Czas')
lines(ts_filter, col = 'red')
lines(ts_lowess, col = 'blue')
legend("topleft", legend = c('średnia ruchoma', 'regresja nielokalna'), col = c('red', 'blue'), pch=20, ...)
```

Wygladzanie szeregu



Trend

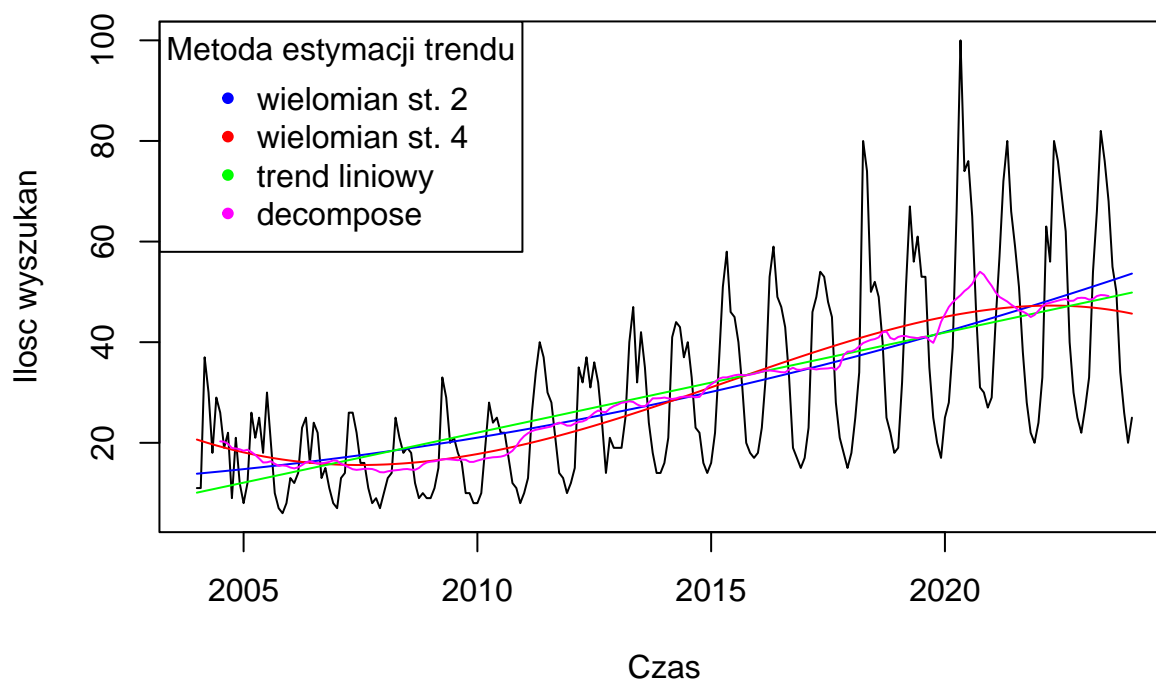
```
# trend - decompose
ts_decompose_trend = dec$trend

# model trendu liniowego
ts_linear_trend = tslm(time_series ~ trend)
ts_linear_trend_vals = fitted.values(ts_linear_trend)

# model trendu wielomianowego
poly2 = tslm(time_series ~ trend + poly(trend,2))
poly4 = tslm(time_series ~ trend + poly(trend,4))
poly2_vals = fitted.values(poly2)
poly4_vals = fitted.values(poly4)

# wykres dla trendow
plot(time_series, main = 'Analiza trendu', ylab = 'Ilość wyszukiwań', xlab = 'Czas')
lines(poly2_vals, col = 'blue')
lines(poly4_vals,col = 'red')
lines(ts_linear_trend_vals, col = 'green')
lines(ts_decompose_trend, col = 'magenta')
legend("topleft",
      legend = c('wielomian st. 2', 'wielomian st. 4', 'trend liniowy', 'decompose'),
      col = c('blue', 'red', 'green', 'magenta'),pch=20, title = 'Metoda estymacji trendu')
```


Analiza trendu



```
# porownanie trendow
slinear = summary(ts_linear_trend)
spoly2 = summary(poly2)
spoly4 = summary(poly4)

rsquares = c(slinear$adj.r.squared, spoly2$adj.r.squared, spoly4$adj.r.squared)
sigmas = c(slinear$sigma, spoly2$sigma, spoly4$sigma)
names = c('trend liniowy', 'trend wielomianowy st. 2', 'trend wielomianowy st. 4')
trends = data.frame(names, sigmas, rsquares)
colnames(trends) = c('Metoda estymacji', 'RSE', 'R kwadrat')
kable(trends, caption = 'Statystyki dla różnych metod estymacji trendu')
```

Table 1: Statystyki dla różnych metod estymacji trendu

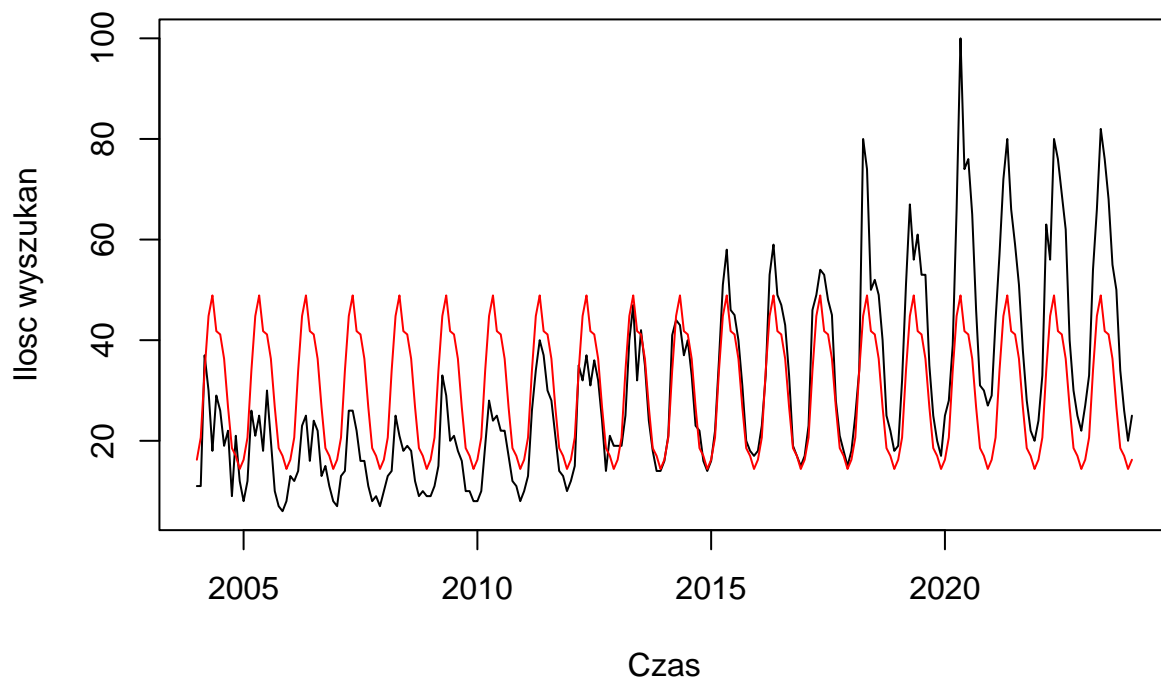
Metoda estymacji	RSE	R kwadrat
trend liniowy	14.67058	0.3812449
trend wielomianowy st. 2	14.60083	0.3871146
trend wielomianowy st. 4	14.37376	0.4060295

Sezonowość

```
# sezonowość linowo
ts_linear_season = tslm(time_series ~ season)
ts_linear_season_vals = fitted.values(ts_linear_season)

# wykres dla sezonowości
plot(time_series, main = 'Analiza sezonowości', ylab = 'Ilość wyszukiń', xlab = 'Czas')
lines(ts_linear_season_vals, col = 'red')
```

Analiza sezonowosci

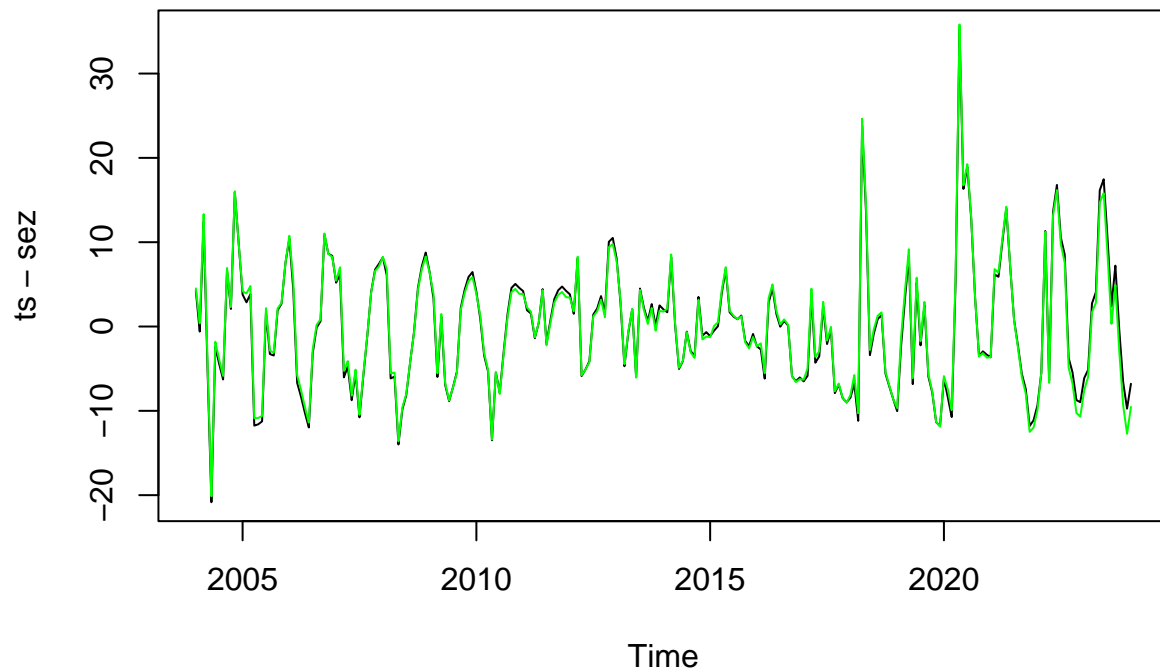


Usuwanie trendu i sezonowości

```
#usuwamy trend i sezonowosc
ts = time_series - poly4_vals
sez = fitted.values(tslm(ts ~ season))

#usuwamy sezonowosc i trend
ts2 = time_series - fitted.values(tslm(time_series ~ season))
trend2 = fitted.values(tslm(ts2 ~ trend + poly(trend,4)))

plot(ts-sez)
lines(ts2-trend2, col='green')
```



#uniasek: nie ma znaczenia co usuwamy pierwsze

Dekompozycja

```
# metody usuwania trendu:
trend_poly = tslm(time_series ~ trend + poly(trend,4))
trend_poly_vals = fitted.values(poly4)

trend_diff = diff(time_series)

trend_dec = decompose(time_series)$trend

# metody usuwania sezonowosci
season_diff = diff(time_series, 12)

season_dec = decompose(time_series)$season # albo fitted.values(tslm(time_series ~ season))

# dalej: usuwamy najpierw trend i od nowych danych usuwamy nowa sezonowosc

# szereg z trendem poly4
dane1 = time_series - trend_poly_vals
ts1 = dane1 - fitted.values(tslm(dane1 ~ season))

# szereg z diff
```

```

dane2 = diff(time_series)
ts2 = diff(dane2, 12)

# szereg z decompose
dane3 = time_series - trend_dec
ts3 = dane3 - decompose(dane3)$season

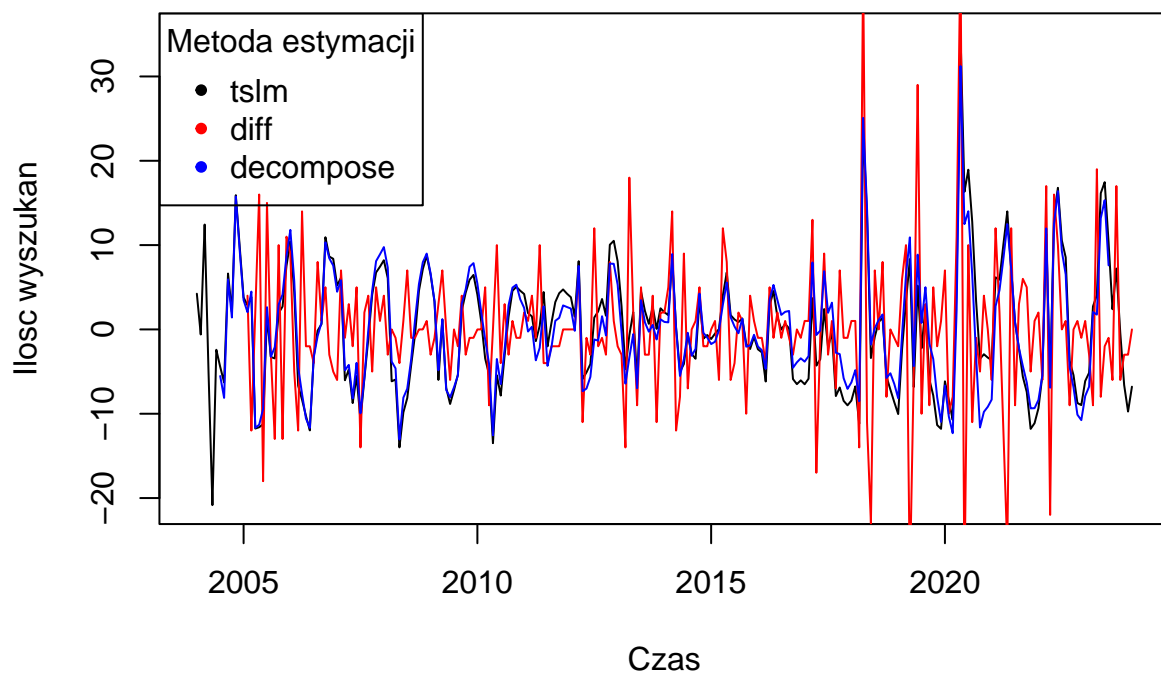
# srednia ruchoma
trend_filter = fitted.values(tslm(ts_filter ~ trend + poly(trend,4)))
dane4 = ts_filter - trend_filter
ts4 = dane4 - fitted.values(tslm(dane4 ~ season))

# regresja nielokalna
pom5 = ts(ts_lowess$y, start=c(2004, 1), frequency=12)
trend_lowess = fitted.values(tslm(pom5 ~ trend + poly(trend,4)))
dane5 = pom5 - trend_lowess
ts5 = dane5 - fitted.values(tslm(dane5 ~ season))

# wykresy dla 5 modeli:
plot(ts1, main = 'Szereg czasowy po usunięciu trendu i sezonowości', ylab = 'Ilość wyszukiń', xlab = 'Czas')
lines(ts2, col = 'red')
lines(ts3, col='blue')
legend("topleft",
      legend = c('tslm', 'diff', 'decompose'),
      col = c('black', 'red', 'blue'), pch=20, title = 'Metoda estymacji')

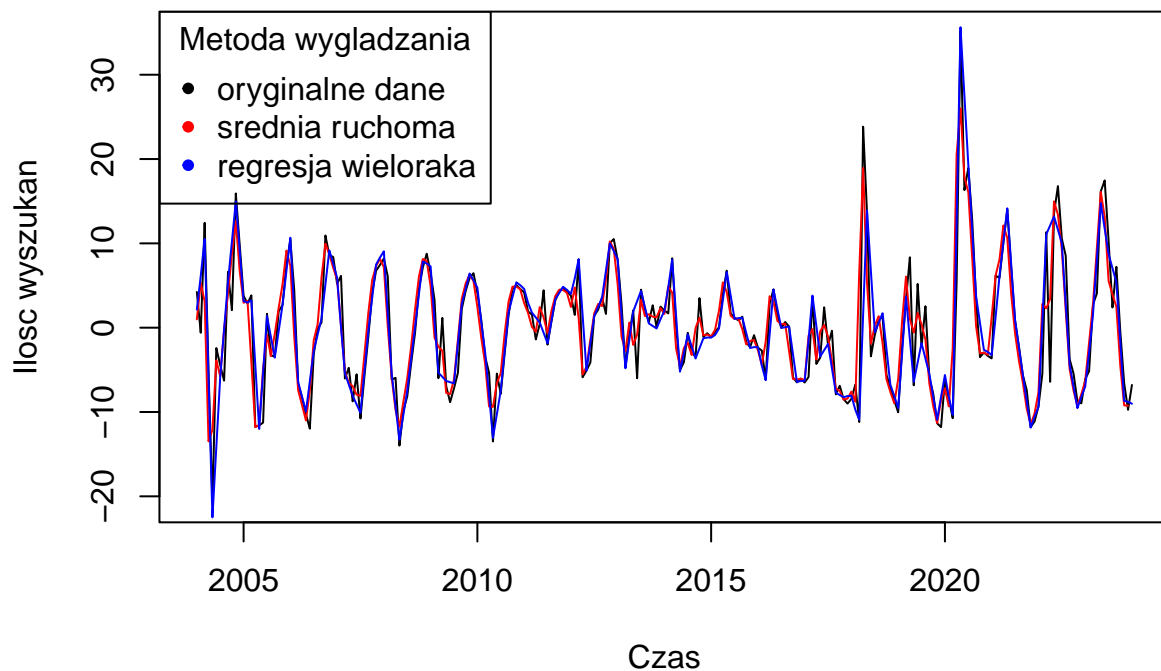
```

Szereg czasowy po usunięciu trendu i sezonowości



```
plot(ts1, main = 'Szereg czasowy po usunięciu trendu i sezonowości', ylab = 'Ilość wyszukikań', xlab = 'Czas')
lines(ts4, col = 'red')
lines(ts5, col='blue')
legend("topleft",
      legend = c('oryginalne dane', 'średnia ruchoma', 'regresja wieloraka'),
      col = c('black', 'red','blue'),pch=20, title = 'Metoda wygładzania')
```

Szereg czasowy po usunięciu trendu i sezonowości



jak wybrać który jest najlepszy?????

Dobór modelu

Wskaźniki AIC, BIC, AICC

```
# statystyki aic, bic, aicc
stats = function(ts){
  model = auto.arima(ts)
  s = c(model$aic, model$bic, model$aicc)
  return(s)}

names = c('AIC', 'BIC', 'AICC')
df = data.frame(names, stats(ts1), stats(ts2), stats(ts3), stats(ts4), stats(ts5))
```

```
colnames(df) = c('Statystyka', 'tslm', 'diff', 'decompose', 'filter', 'lowess')
kable(df, caption= 'Wskaźniki dla różnych modeli')
```

Table 2: Wskaźniki dla różnych modeli

Statystyka	tslm	diff	decompose	filter	lowess
AIC	1450.000	1466.086	1328.156	1140.042	1080.710
BIC	1463.735	1496.950	1341.675	1160.618	1097.879
AICC	1450.179	1466.912	1328.345	1140.422	1080.979

ARIMA

```
# wybrać dwa modele i zrobić dla nich analize acf i pacf
```