

# Wavelets for behavioral oscillations

Bartłomiej KroczeK

2022-01-10

## Contents

Dane surowe (high freq) . . . . .	1
Teoria . . . . .	2
Wykresy . . . . .	2

Na naszym ostatnim spotkaniu Tomek pokazywał wykresy z falek z których wynikało, że jest taki przedział CSI w którym wydaje się że jest sprzężenie (i to istotne statystycznie).

```
# load data
low_data_dir <- "../data/low_resolution"
high_data_dir <- "../data/high_resolution"
low_files <- dir(low_data_dir, patter = "*.csv$")
high_files <- dir(high_data_dir, patter = "*.csv$")

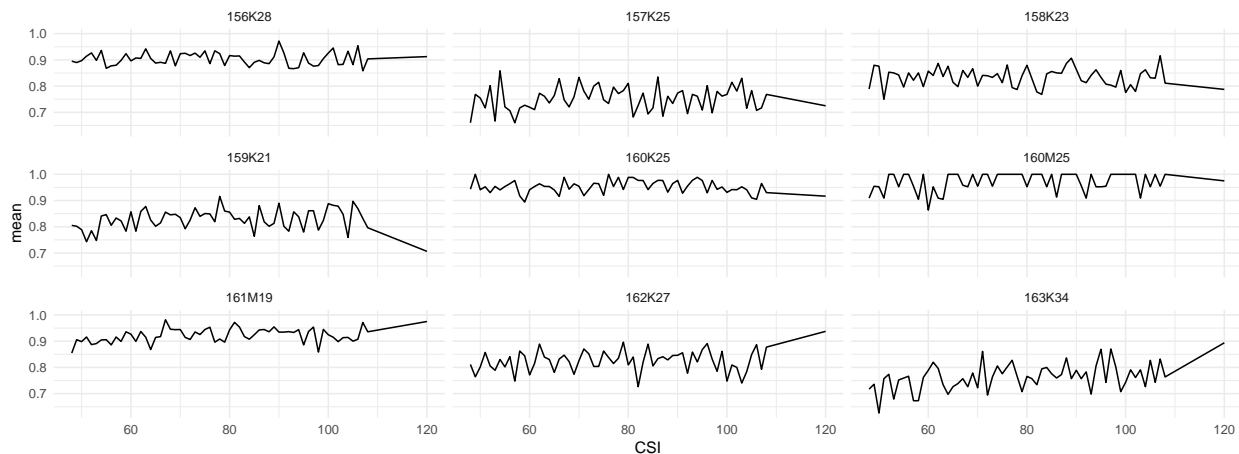
low_data <- low_files %>%
  map(~ read_csv(file.path(low_data_dir, .), show_col_types = F)) %>%
  reduce(rbind)

high_data <- high_files %>%
  map(~ read_csv(file.path(high_data_dir, .), show_col_types = F)) |>
  reduce(rbind)

# high res data gathered in 4 sessions encoded PARTID + S1-S4. S1-S4 removed
high_data <- high_data |>
  mutate(PART_ID = str_sub(PART_ID, 1, nchar(PART_ID)-2))
```

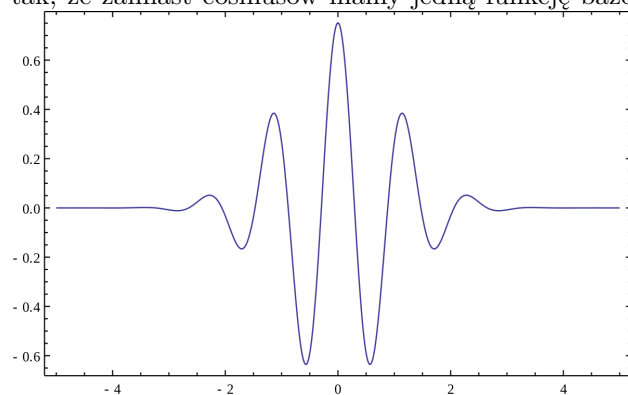
## Dane surowe (high freq)

```
high_data %>%
  group_by(PART_ID, CSI) %>%
  summarise(mean = mean(Corr)) %>%
  ggplot(mapping = aes(x = CSI, y = mean)) +
  geom_line() +
  facet_wrap(~PART_ID)
```



## Teoria

Transformata fouriera działa tak, że bierzemy sobie sygnał i rozbijamy go na sumę cosinusów o różnej częstotliwości i fazie. Wiemy też, że transformata fouriera działa w obie strony - na zbiorze cosinusów możemy zrobić transformatę odwrotną inverse-FFT i dostajemy sygnał oryginalny. Z kolei z falkami jest tak, że zamiast cosinusów mamy jedną funkcję bazową, najpopularniejszym wyborem jest falka morleta;



(Formalnie to trzeba by było pamiętać, że tam jest komponent rzeczywisty i urojony, ale dla wyvodu to nie ma znaczenia)

No i transformata falkowa to jest takie zwierze, że bierze tyle takich falek morleta ile potrzebuje i każdą z nich rozciąga/ściska a następnie przesuwa po osi x. Czyli tak jak wynikiem Fouriera jest zbiór współczynników jak przeskalować (jaką amplitudę nadać) i jak przesunąć w fazie (ale o tym na potrzeby wyvodu zapominamy, bo to część urojona) kolejne cosinusy, to wynikiem falek jest recepta jak pogniatać i przeciągać po osi x ileś tam kopii funkcji bazowej.

No i tutaj się pojawia sanity-check w postaci rekonstrukcji sygnału. Jest to nic innego jak zrobienie odwrotnej transformaty falkowej na tym zbiorze pogniecionych i przesuwanych falek w celu uzyskania oryginalnego sygnału.

I to właśnie widać na wykresach poniżej.

## Wykresy

Przepraszam, że zostawiam brzydko opisane ośki. Biblioteka która rysuje te falki średnio ze mną współpracuje i uznałem, że użeranie się z nią nie ma sensu.

Dla każdego badanego mamy trzy wykresy, kolejno:

### 1. Widmo czasowo-częstotliwościowe

- oś X to kolejne punkty danych, czyli ten nasz przedział 300-900 ms czy jakoś tak.
- oś Y - długość okresu czyli 1/Hz skala logarytmiczna 0.25 to 4 Hz 0.125 to 8 Hz 0.0625 to 16 Hz 0.03125 to 32 Hz
- Białą obwolutą zaznaczone są obszary istotne statystycznie

### 2. Falki

Poszczególne falki dopasowane do sygnału. Warte uwagi jest to, że (1) jest ich relatywnie niewiele (2) są duże obszary pustki, co oznacza, że aproksymujemy zerem.

### 3. Wynik rekonstrukcji falek nałożony na oryginalny sygnał.

Widać, że te rekonstrukcje są kiepskie, co oznacza, że dopasowanie falek jest równie złe.

**Insight:** Właściwie wszystkie wykresy mają obszary istotne statystycznie, zgodnie z wynikami Tomka. Są one jednak rozmieszczone dość chaotycznie.

**Insight:** Ta istotność statystyczna wychodzi być może z porównywania obszarów gdzie amplituda dopasowanych falek jest w miarę duża, do obszarów, w których nie dopasowano niczego. Z porównania wykresów widm do rekonstrukcji widać, że istotnie jest tam, gdzie jest peak falki.

**Insight:** Te “wyspy istotności statystycznej” są bardzo niestabilne numerycznie. Wystarczy lekko zmienić skalę osi X (nawet na wielokrotność) i wynik jest zupełnie inny. Gdyby te efekty były realne, to zmiana parametru próbkowania nie powinna wiele zmieniać, co najwyżej zakłamywać odczyt lokalizacji z wykresu, tutaj natomiast po zmianie parametru, “wyspy” pojawiają się zupełnie gdzie indziej. To również sugeruje, że istotność może wychodzić z numerycznych błędów przy porównywaniu do zera.

```
part_ids <- high_data |> select(PART_ID) |> unique()
part_ids <- part_ids$PART_ID

# params
detrend = FALSE
sampling = 120
low_freq_band = 4
upper_freq_band = 32

for (pid in part_ids) {
  w <- high_data |>
    filter(Trial_type == "experiment", PART_ID == pid) |>
    group_by(CSI) |>
    summarise(mean = mean(Corr)) |>
    analyze.wavelet(
      "mean",
      loess.span = detrend,
      dt = 1 / sampling,
      lowerPeriod = 1 / upper_freq_band,
      upperPeriod = 1 / low_freq_band,
      make.pval = TRUE,
      n.sim = 10
    )

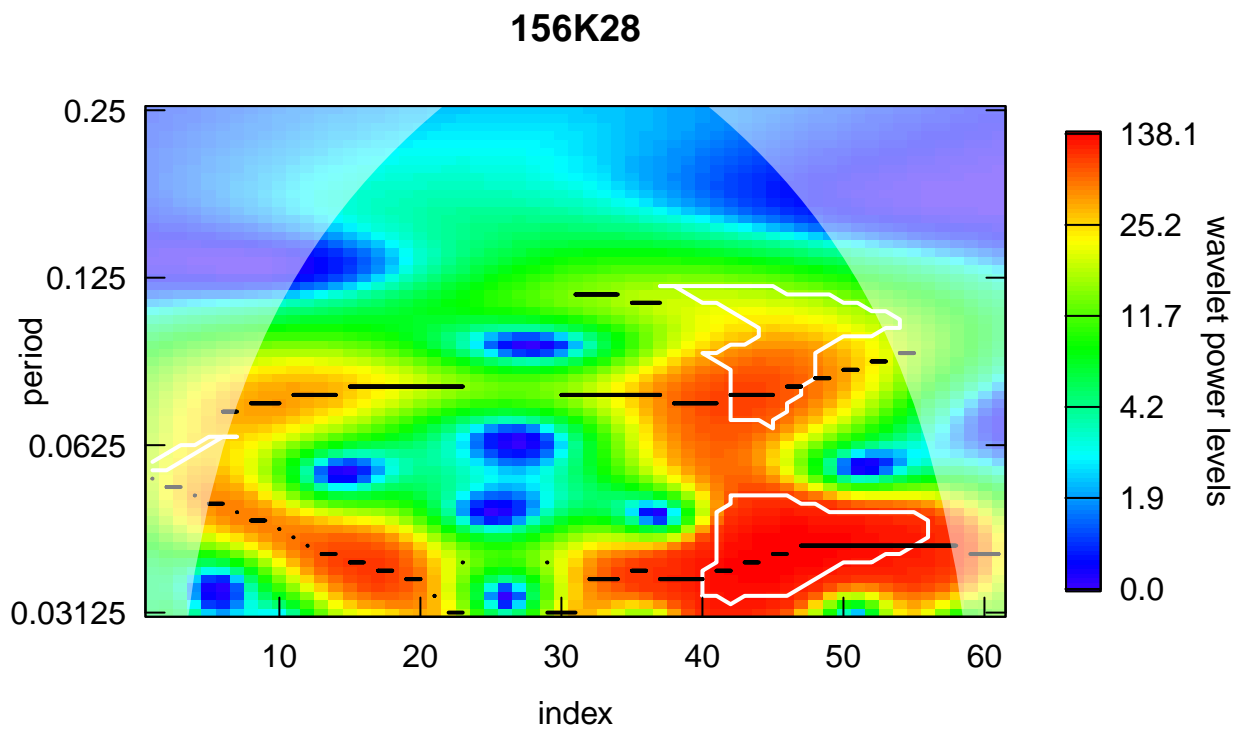
  wt.image(
    w,
    color.key = "quantile",
    main = pid,
    n.levels = 100,
```

```

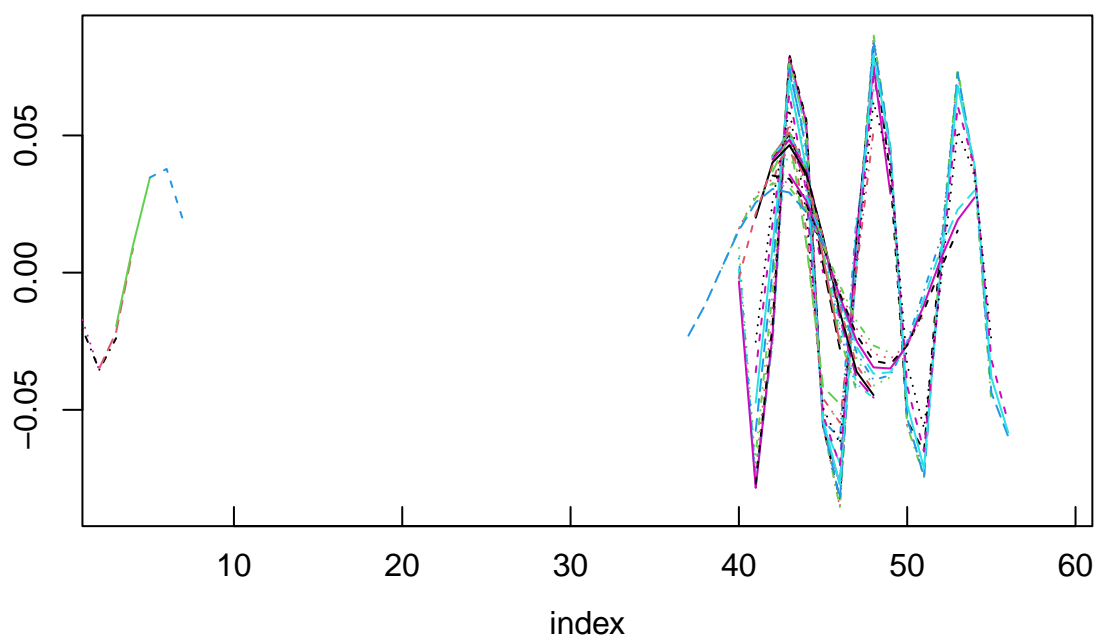
    legend.params = list(lab = "wavelet power levels", mar = 4.7)
  )

  reconstruct(
    w,
    plot.waves = TRUE,
    lwd = c(1, 2),
    legend.coords = "bottomleft",
    main = pid
  )
}

```

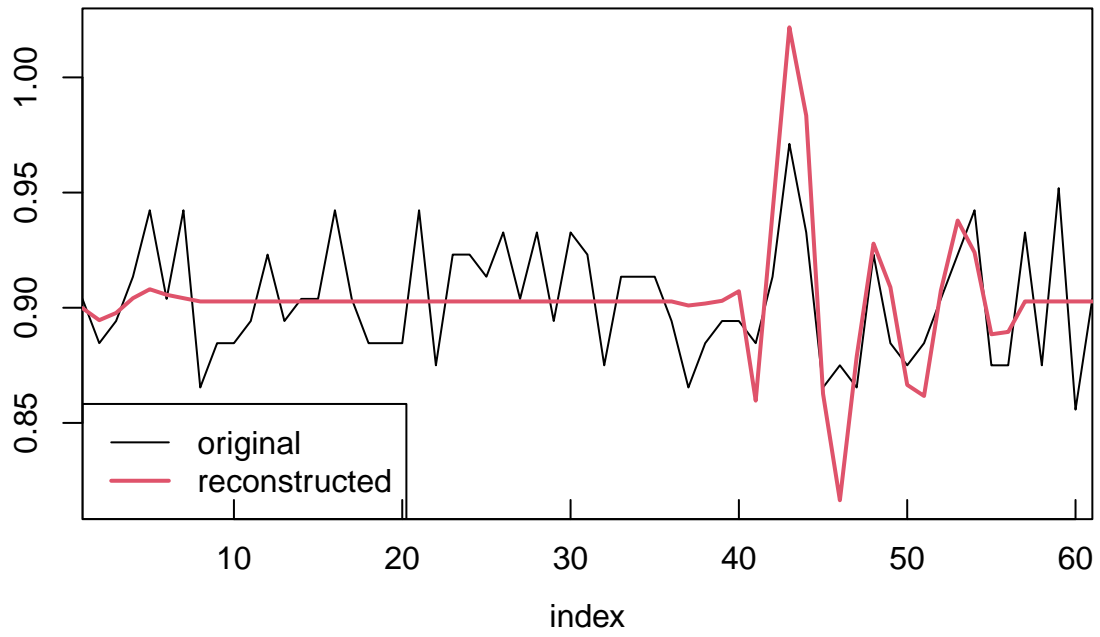


**156K28**



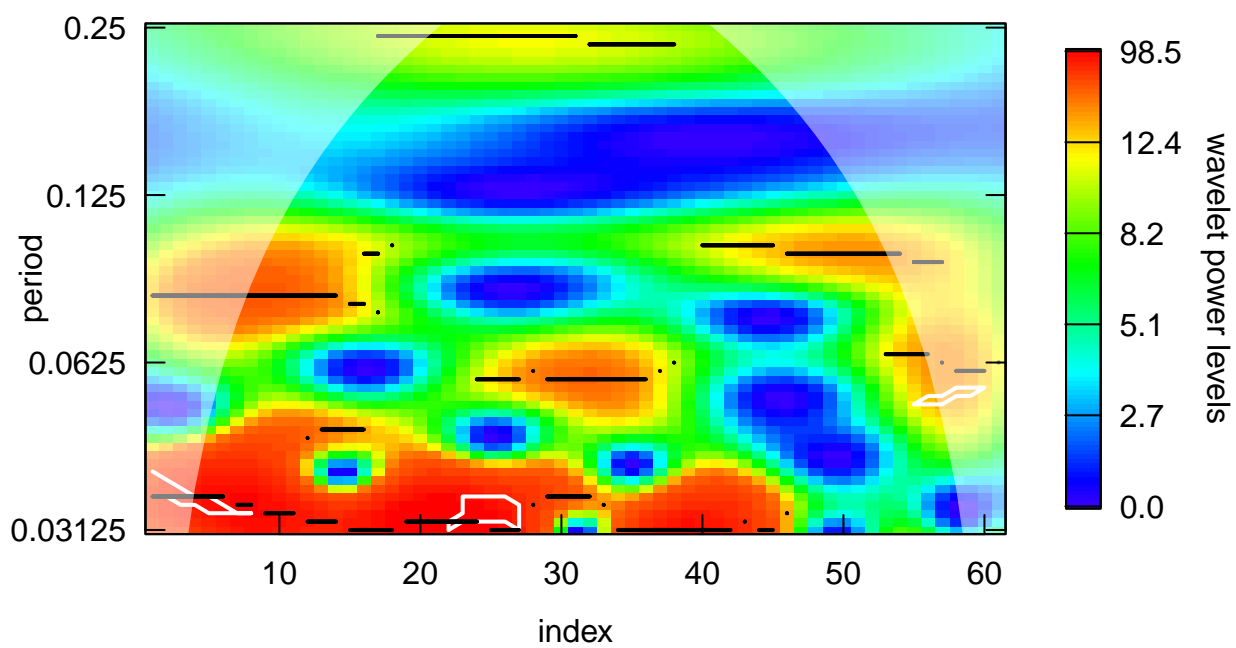
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

## 156K28

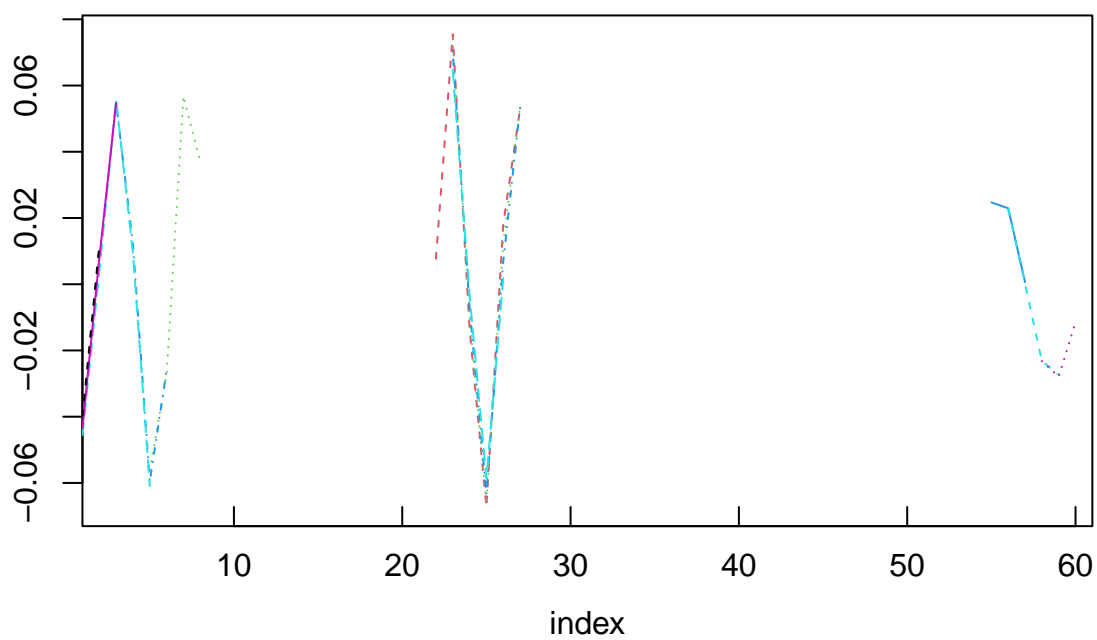


num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

157K25



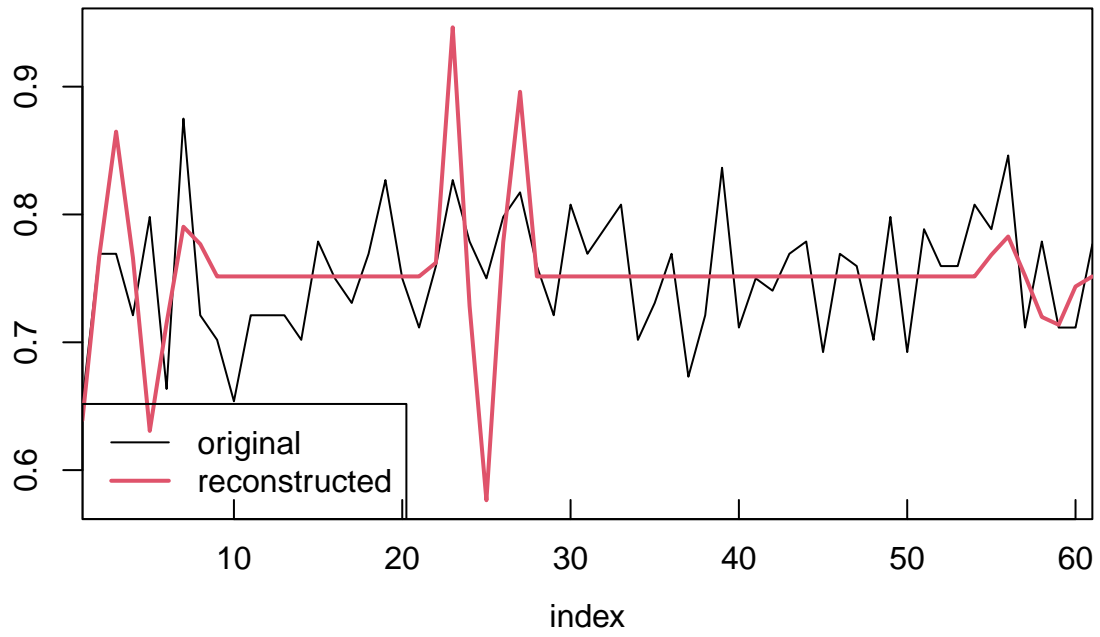
## 157K25



num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

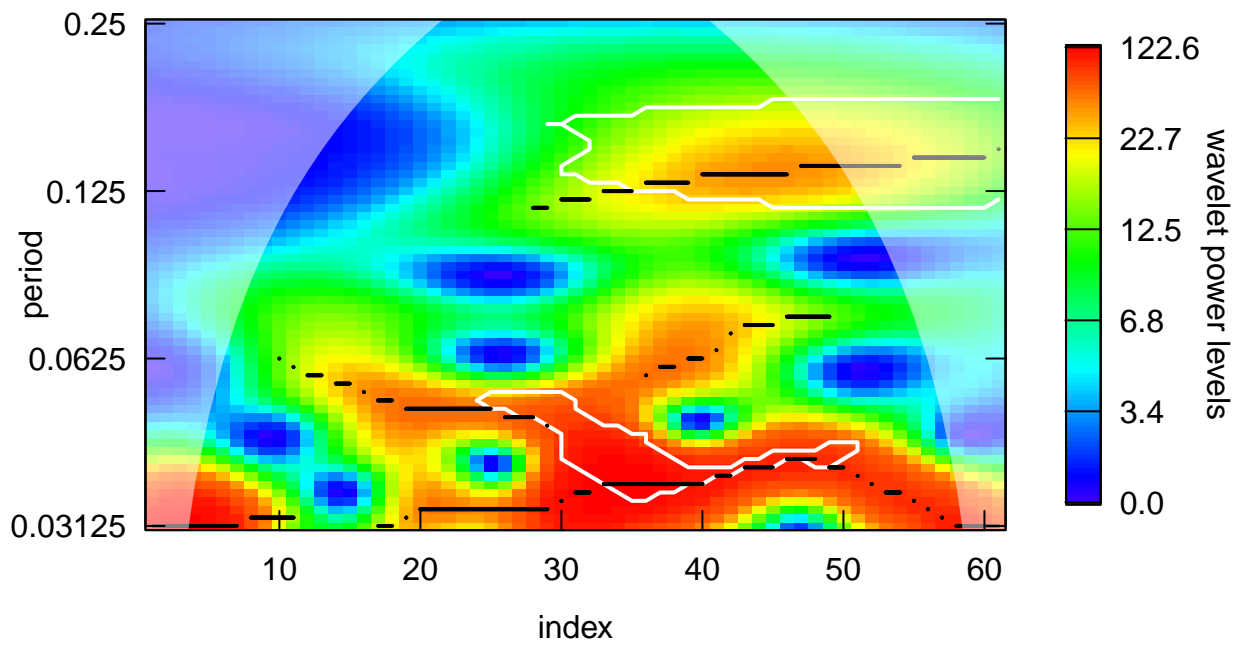


## 157K25

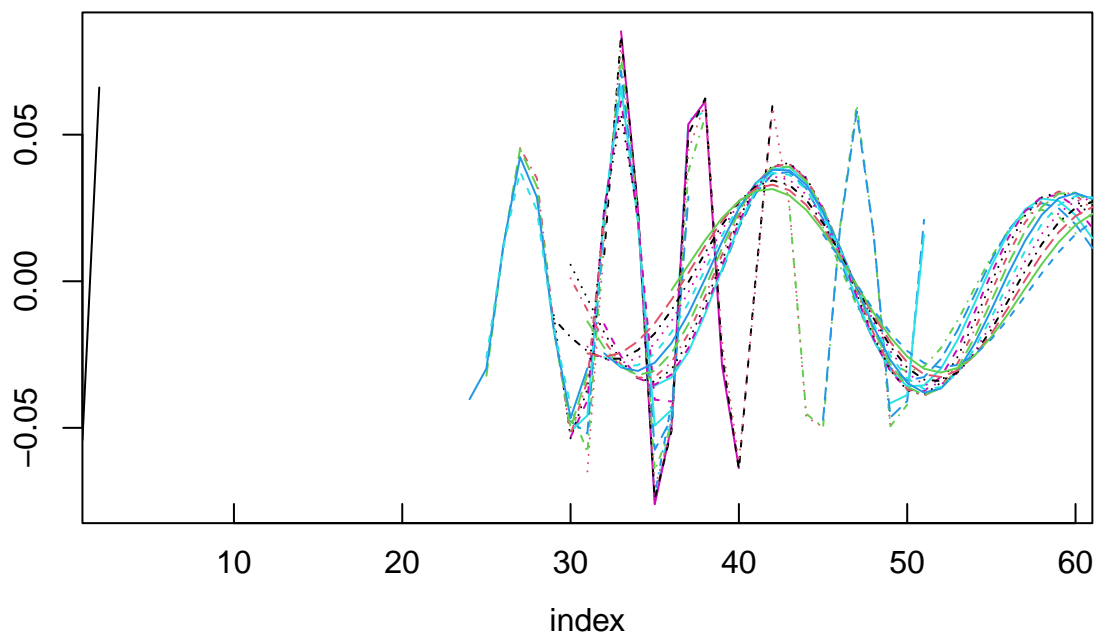


num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

158K23

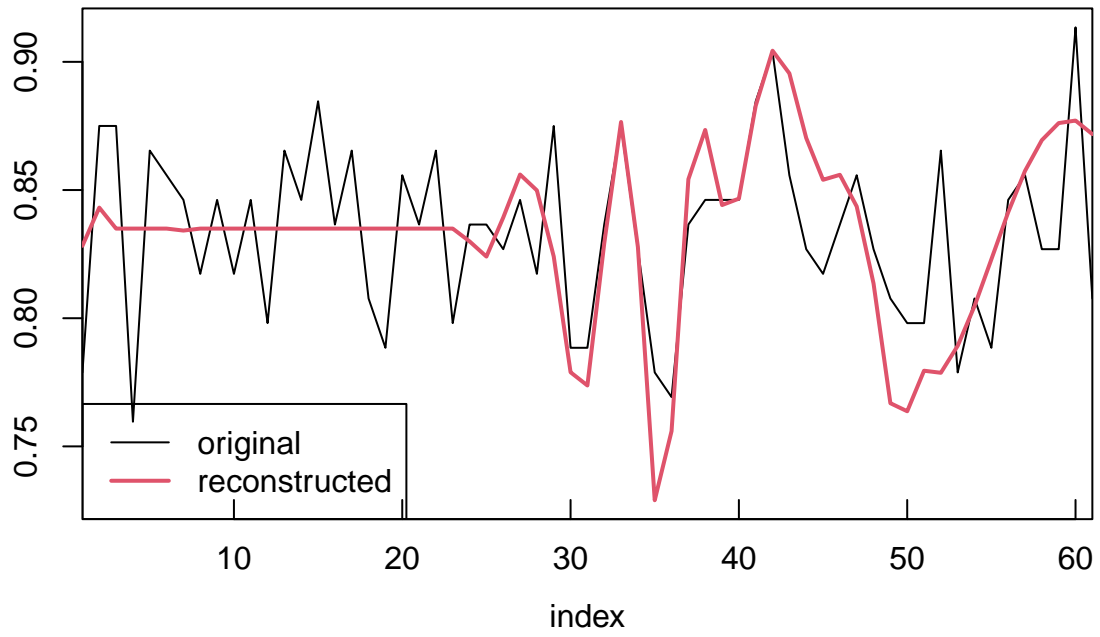


**158K23**



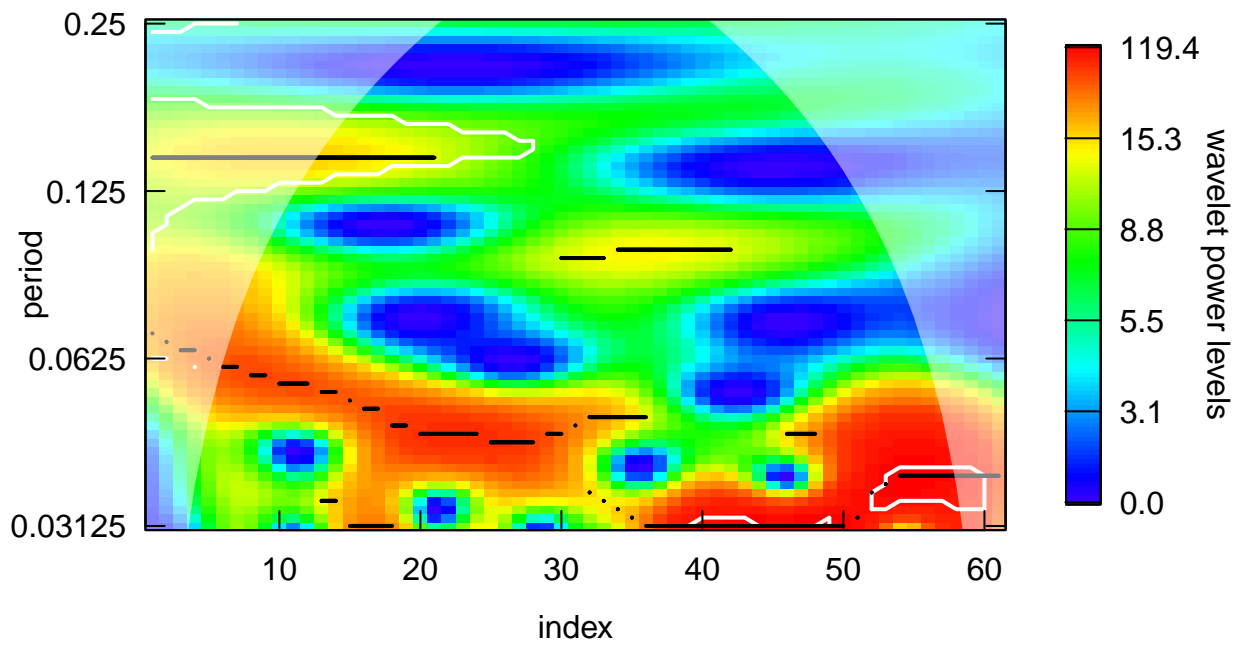
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 4

### 158K23

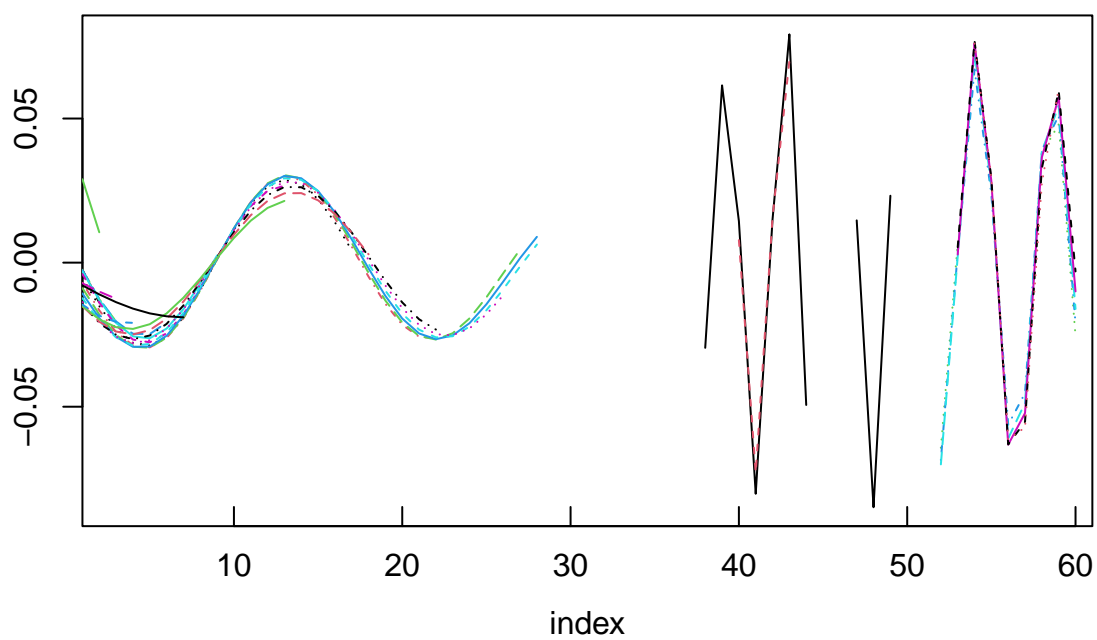


num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

159K21

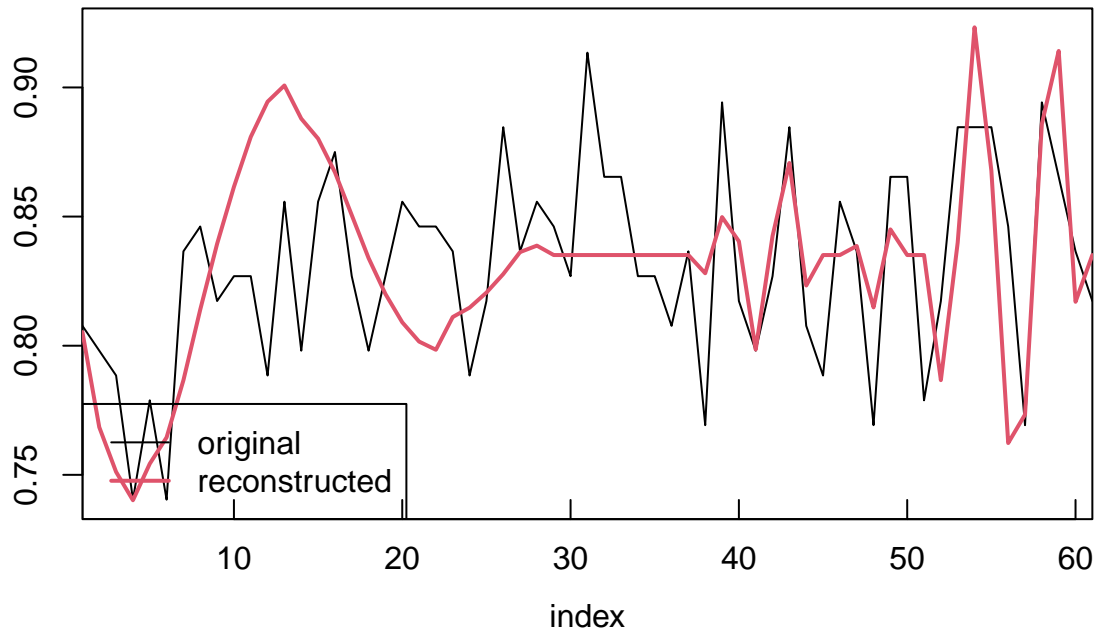


159K21



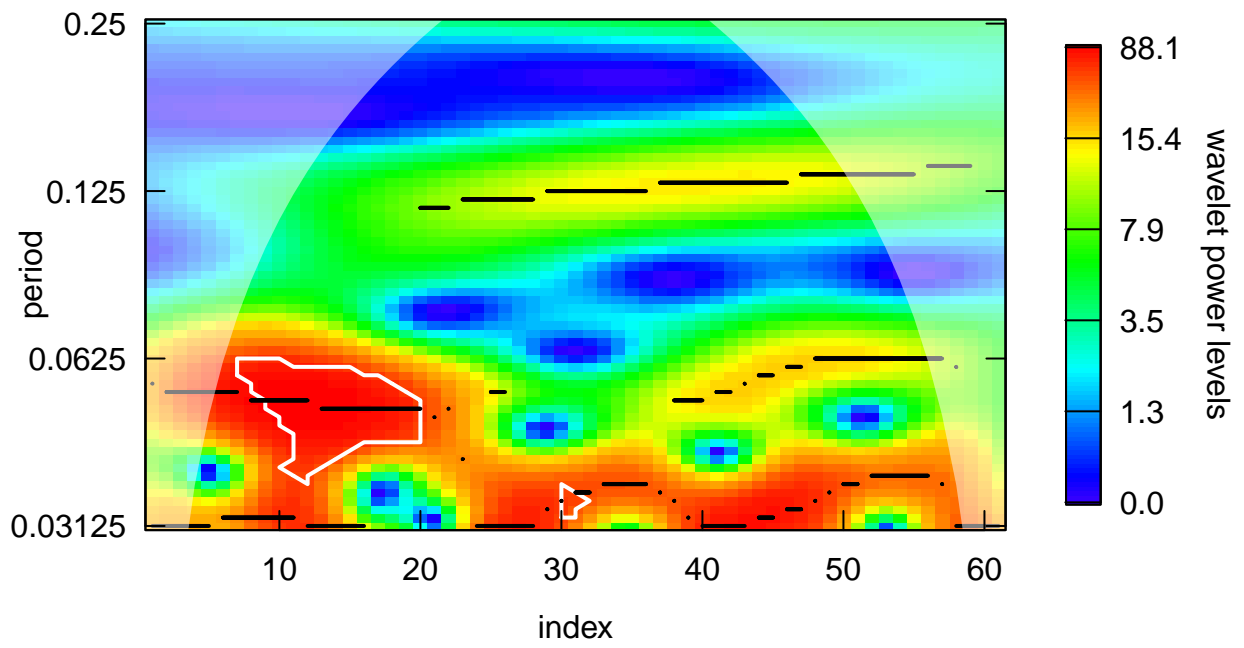
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 2

**159K21**



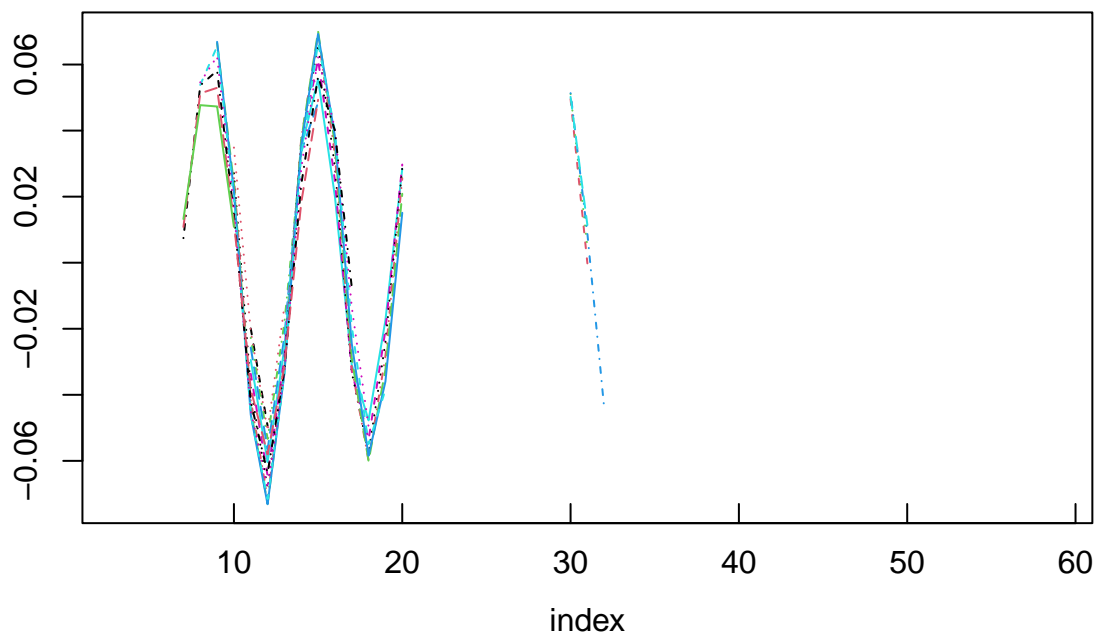
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

# 160K25



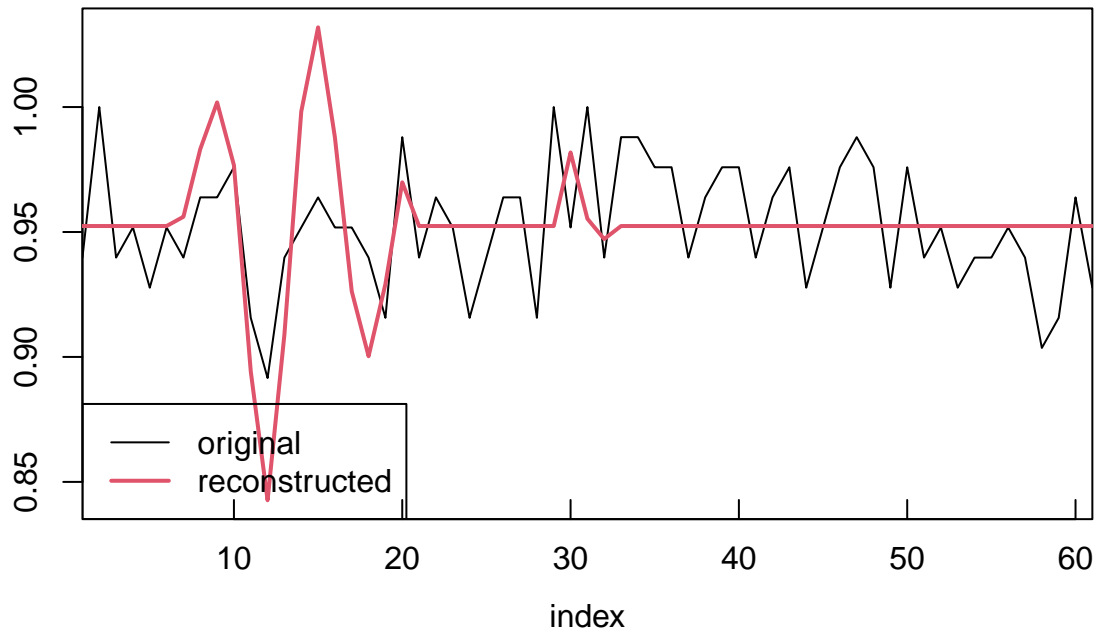


## 160K25



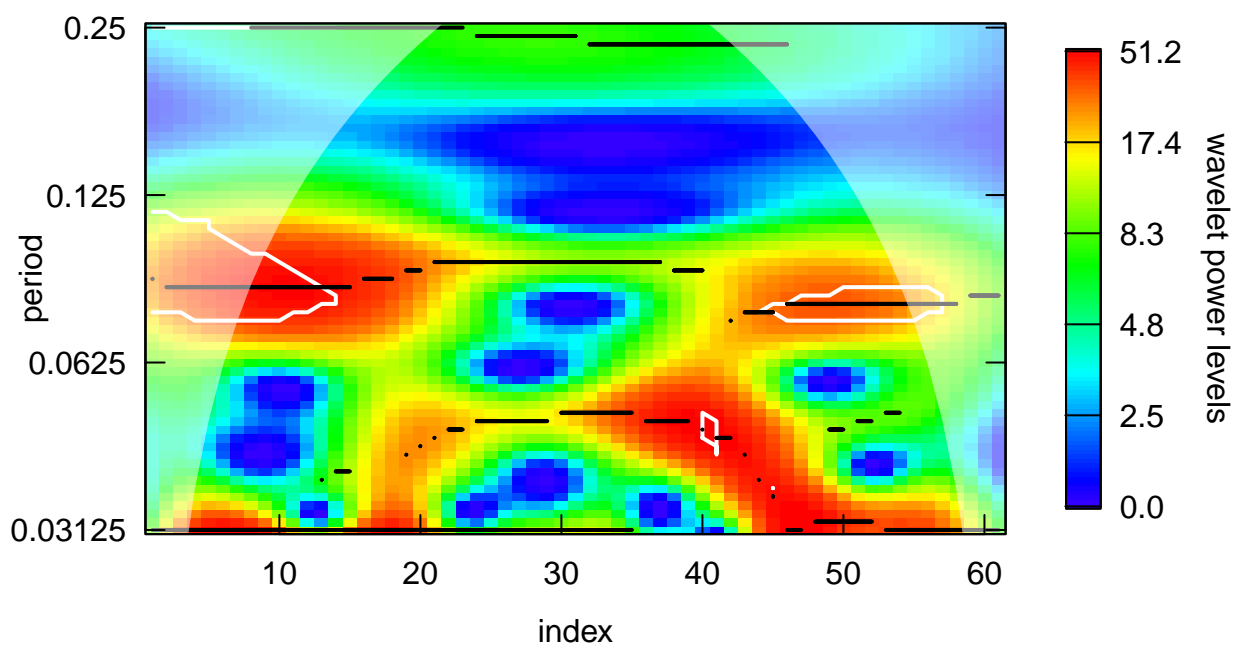
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

## 160K25

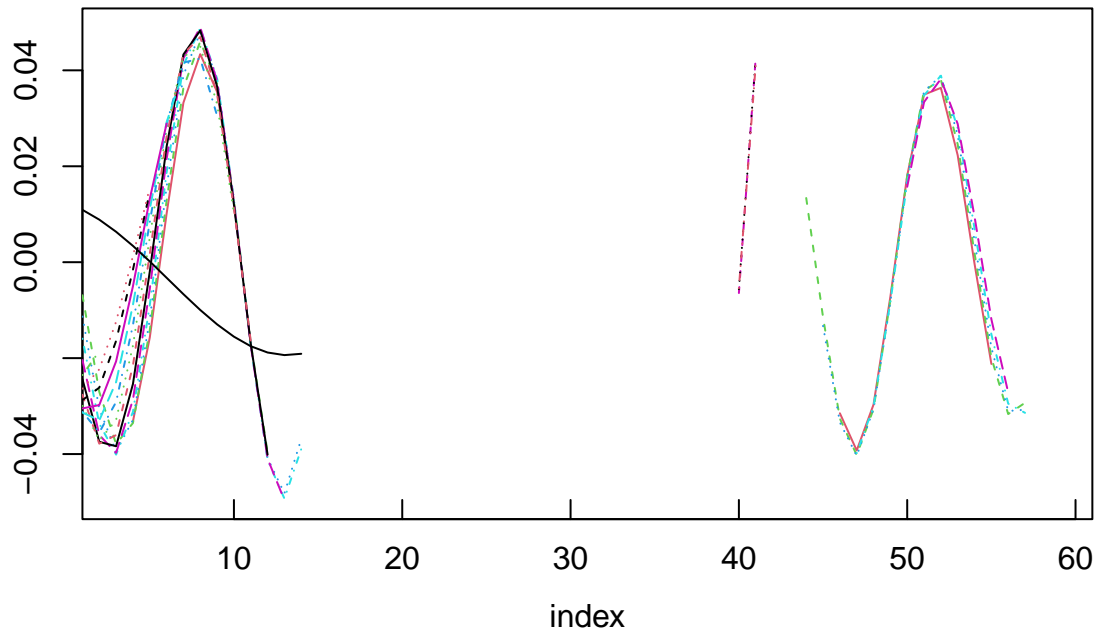


num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

# 160M25

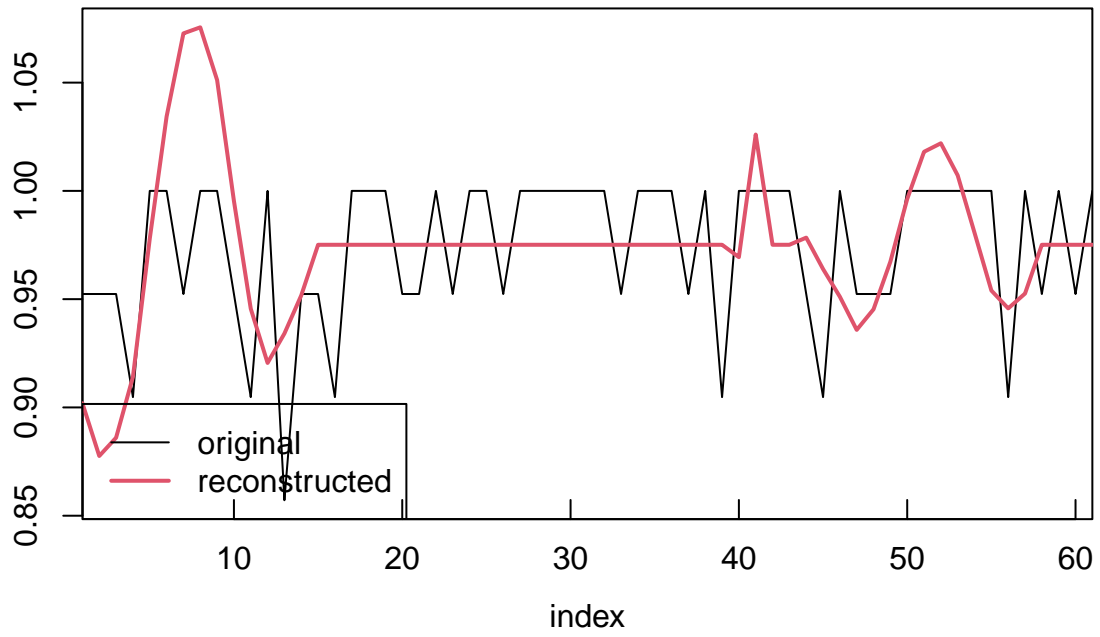


## 160M25



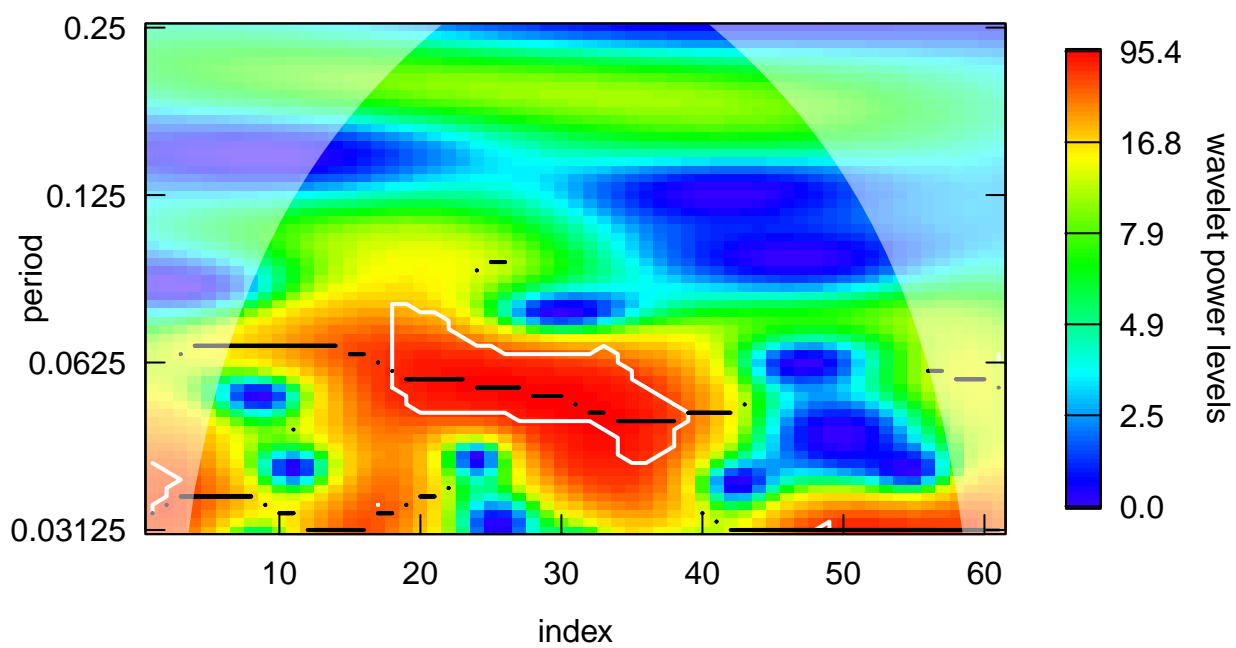
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 2

## 160M25

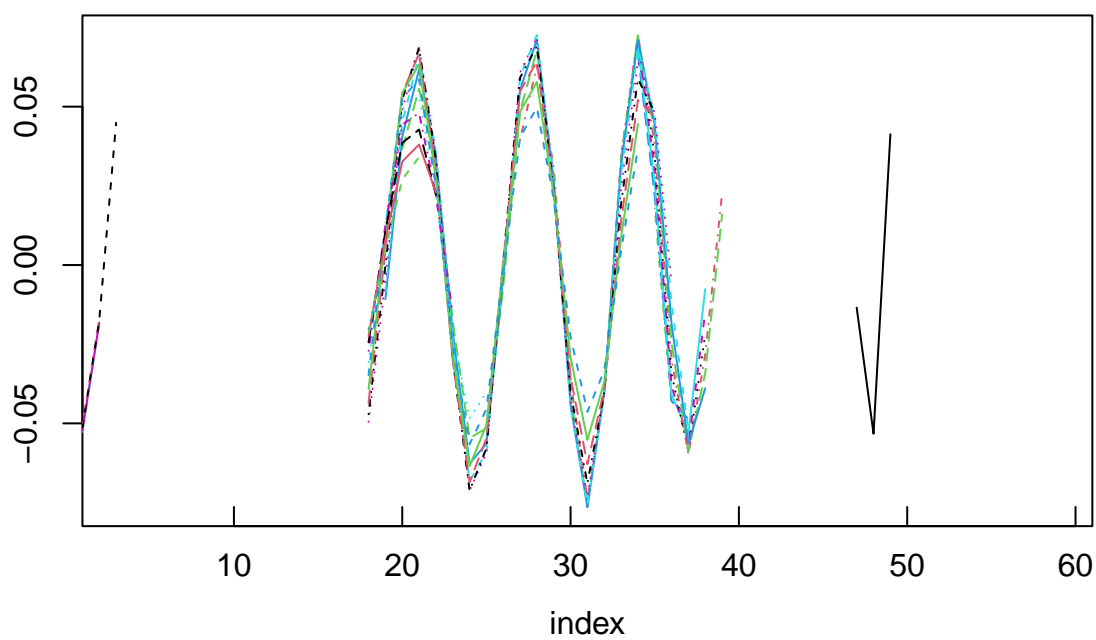


num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

# 161M19

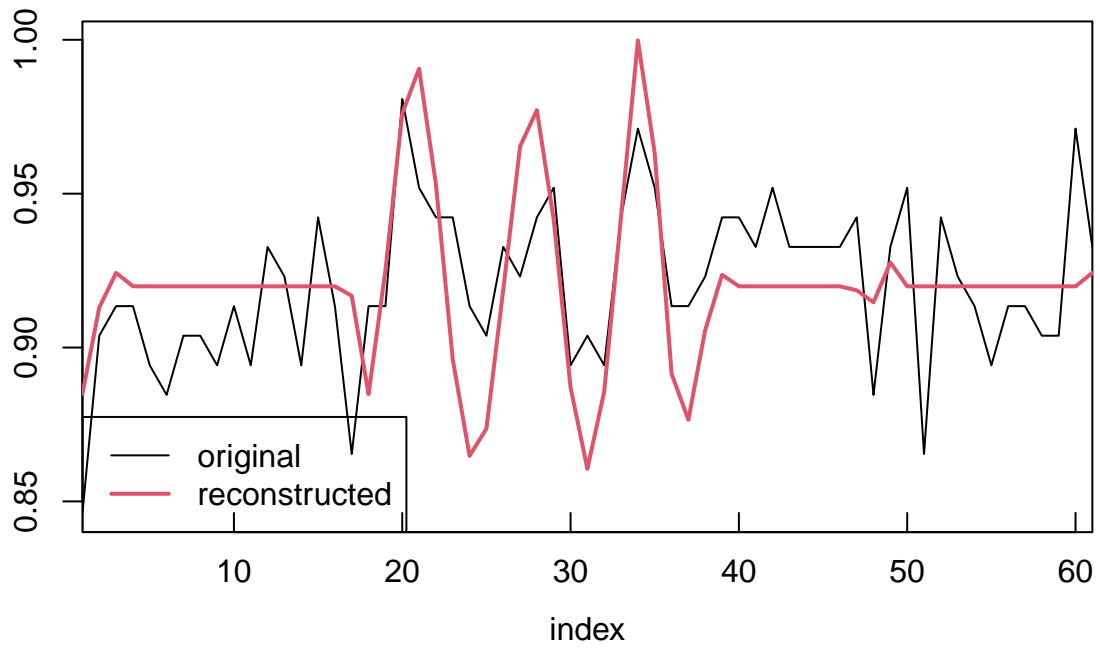


# 161M19



num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

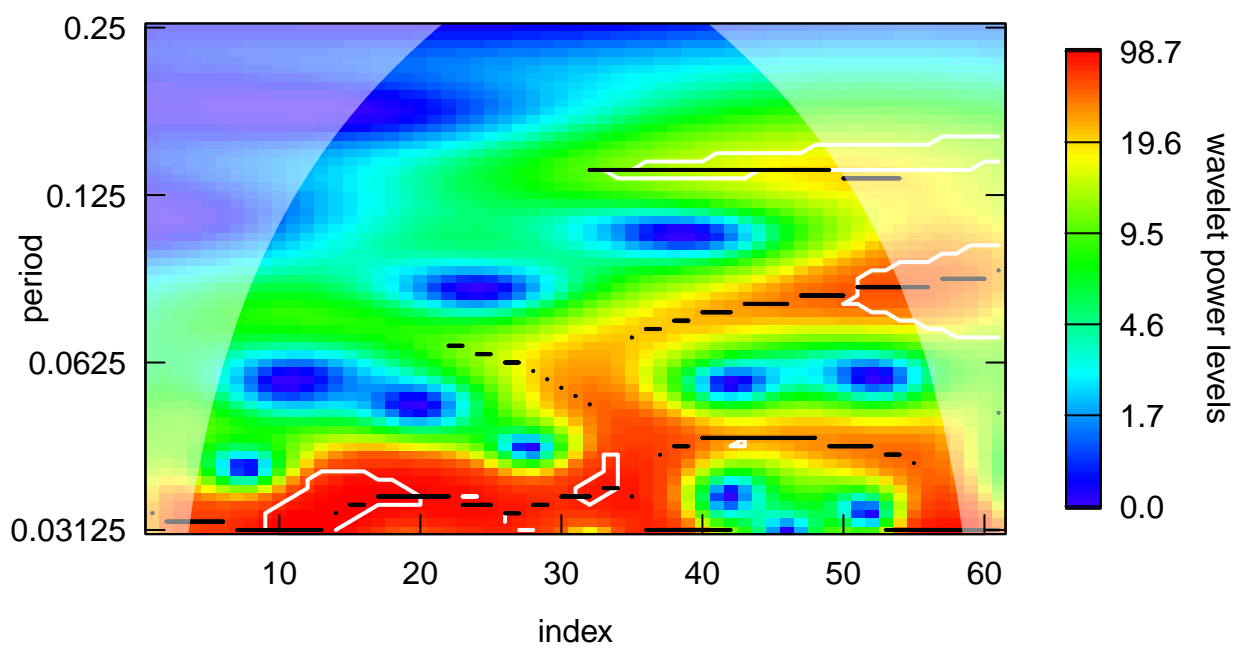
# 161M19



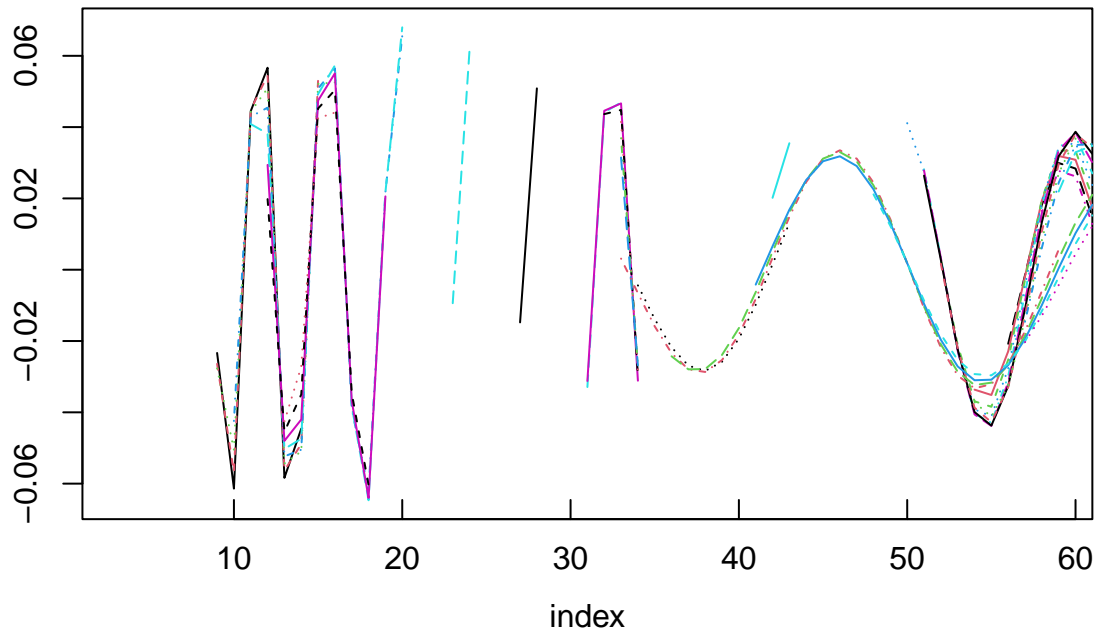
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3



162K27

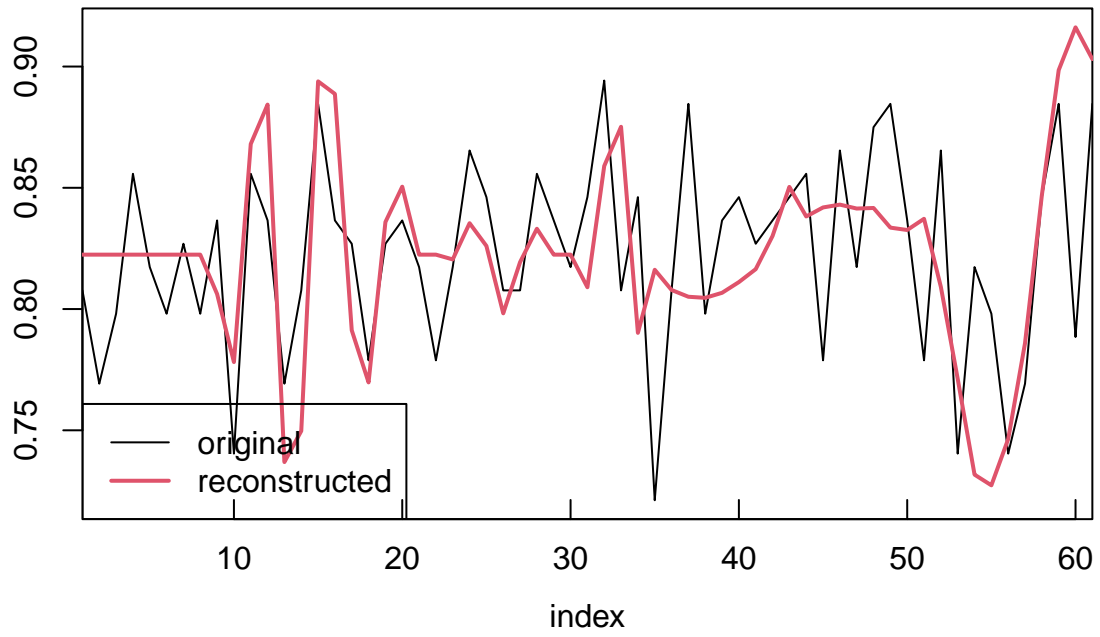


## 162K27



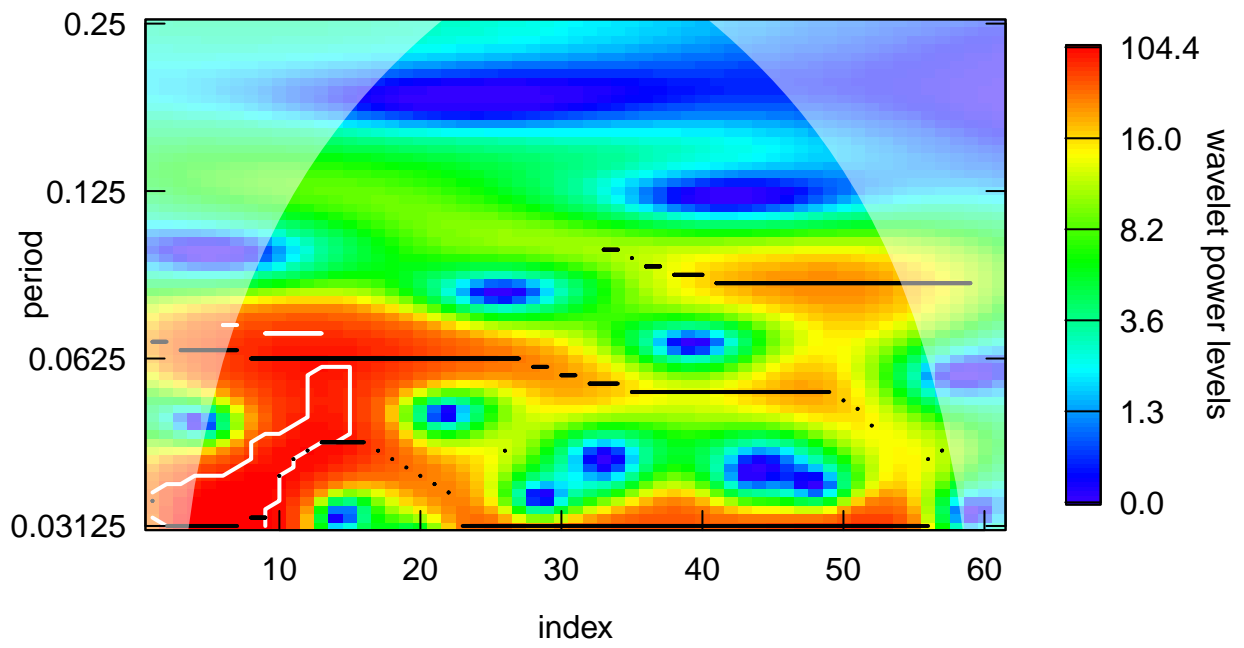
num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 4

## 162K27

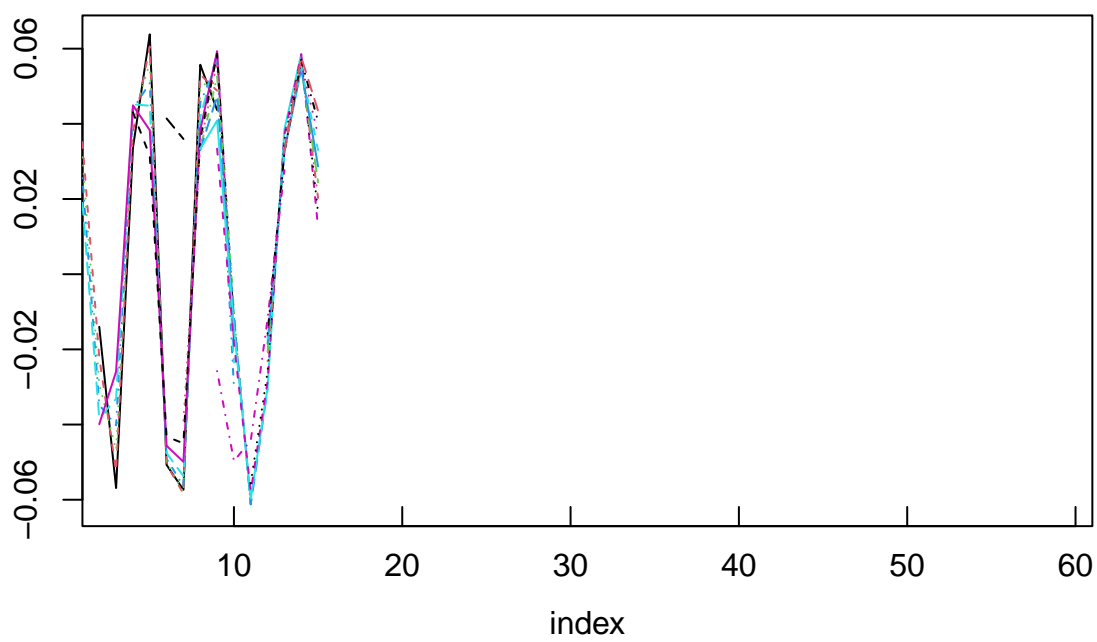


num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

163K34

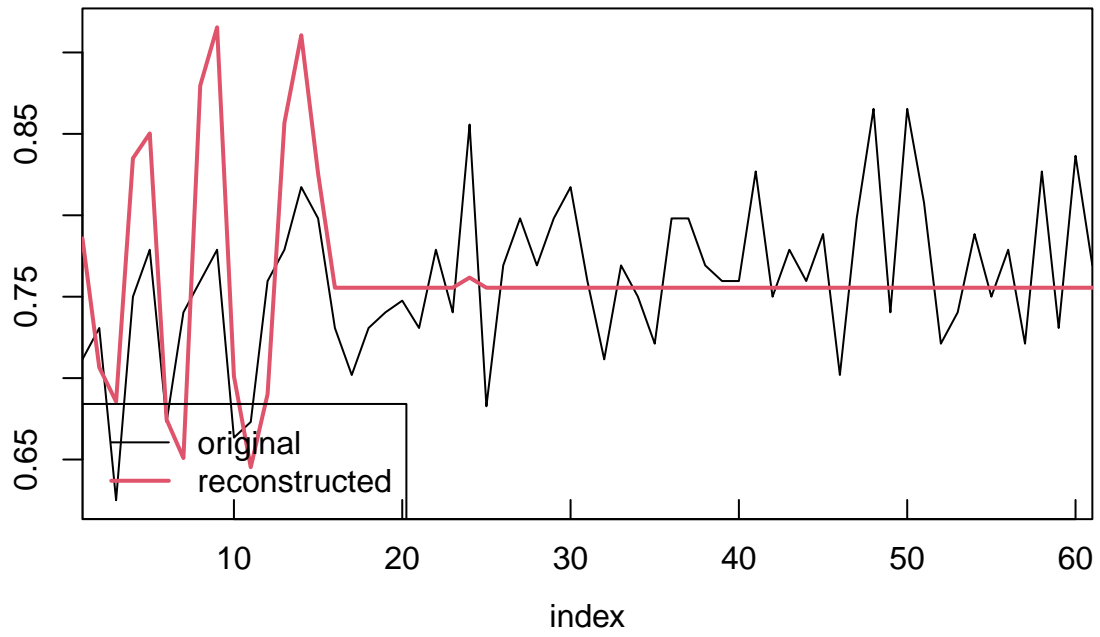


**163K34**



num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3

**163K34**



num power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: 3