Time Series FFT analysis

Hyom

```
library(readr)
energy_data <- read_csv([redacted]/energydata_co
```

```
## Rows: 19735 Columns: 29
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (28): Appliances, lights, T1, RH_1, T2, RH_2, T3, RH_3, T4, RH_4, T5, R...
## dttm  (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# View(energy_data)
energy_data$date <- strptime(as.character(energy_data$date),format="%Y-%m-%d %H:%M:%S")
energy_data$date <- as.POSIXct(energy_data$date,tz = "UTC")
```

```
library(knitr)

data <- data.frame(
  Variables = c("date time", "Appliances", "lights", "T1", "RH_1", "T2", "RH_2", "T3", "RH_3", "T4", "RH
  Description = c("Year-month-day hour:minute:second", "Energy use in Wh", "Energy use of light fixtures
)

variable <- kable(data)
variable
```

Variables	Description
date time	Year-month-day hour:minute:second
Appliances	Energy use in Wh
lights	Energy use of light fixtures in the house in Wh
T1	Temperature in kitchen area, in Celsius
RH_1	Humidity in kitchen area, in %
T2	Temperature in living room area, in Celsius
RH_2	Humidity in living room area, in %
T3	Temperature in laundry room area
RH_3	Humidity in laundry room area, in %
T4	Temperature in office room, in Celsius
RH_4	Humidity in office room, in %
T5	Temperature in bathroom, in Celsius
RH_5	Humidity in bathroom, in %
T6	Temperature outside the building (north side), in Celsius

Variables	Description
RH_6	Humidity outside the building (north side), in %
T7	Temperature in ironing room, in Celsius
RH_7	Humidity in ironing room, in %
T8	Temperature in teenager room 2, in Celsius
RH_8	Humidity in teenager room 2, in %
T9	Temperature in parents room, in Celsius
RH_9	Humidity in parents room, in %
To	Temperature outside (from Chièvres weather station), in Celsius
Pressure	Pressure (from Chièvres weather station), in mm Hg
RH_out	Humidity outside (from Chièvres weather station), in %
Windspeed	Windspeed (from Chièvres weather station), in m/s
Visibility	Visibility (from Chièvres weather station), in km
Tdewpoint	Tdewpoint (from Chièvres weather station), °C
rv1	Random variable 1, nondimensional
rv2	Random variable 2, nondimensional

EDA

```
any(is.na(energy_data$Appliances))
```

```
## [1] FALSE
```

```
any(is.na(energy_data))
```

```
## [1] TRUE
```

```
energy_data[43,]
```

```
## # A tibble: 1 x 29
##   date   Appliances lights    T1  RH_1    T2  RH_2    T3  RH_3
##   <dttm>      <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 NA             40     10  21.4  44.6  20.8  43.9  20.2  45.7
## # i 20 more variables: T4 <dbl>, RH_4 <dbl>, T5 <dbl>, RH_5 <dbl>, T6 <dbl>,
## #   RH_6 <dbl>, T7 <dbl>, RH_7 <dbl>, T8 <dbl>, RH_8 <dbl>, T9 <dbl>,
## #   RH_9 <dbl>, T_out <dbl>, Press_mm_hg <dbl>, RH_out <dbl>, Windspeed <dbl>,
## #   Visibility <dbl>, Tdewpoint <dbl>, rv1 <dbl>, rv2 <dbl>
```

```
which(is.na(energy_data))
```

```
##   [1]    43   187   331   475   619   763   907  1051  1195  1339  1483  1627
##  [13]  1771  1915  2059  2203  2347  2491  2635  2779  2923  3067  3211  3355
##  [25]  3499  3643  3787  3931  4075  4219  4363  4507  4651  4795  4939  5083
##  [37]  5227  5371  5515  5659  5803  5947  6091  6235  6379  6523  6667  6811
##  [49]  6955  7099  7243  7387  7531  7675  7819  7963  8107  8251  8395  8539
##  [61]  8683  8827  8971  9115  9259  9403  9547  9691  9835  9979 10123 10267
##  [73] 10411 10555 10699 10843 10987 11131 11275 11419 11563 11707 11851 11995
##  [85] 12139 12283 12427 12571 12715 12859 13003 13147 13291 13435 13579 13723
```

```
##  [97] 13867 14011 14155 14299 14443 14587 14731 14875 15019 15163 15307 15451
## [109] 15595 15739 15883 16027 16171 16315 16459 16603 16747 16891 17035 17179
## [121] 17323 17467 17611 17755 17899 18043 18187 18331 18475 18619 18763 18907
## [133] 19051 19195 19339 19483 19627
```

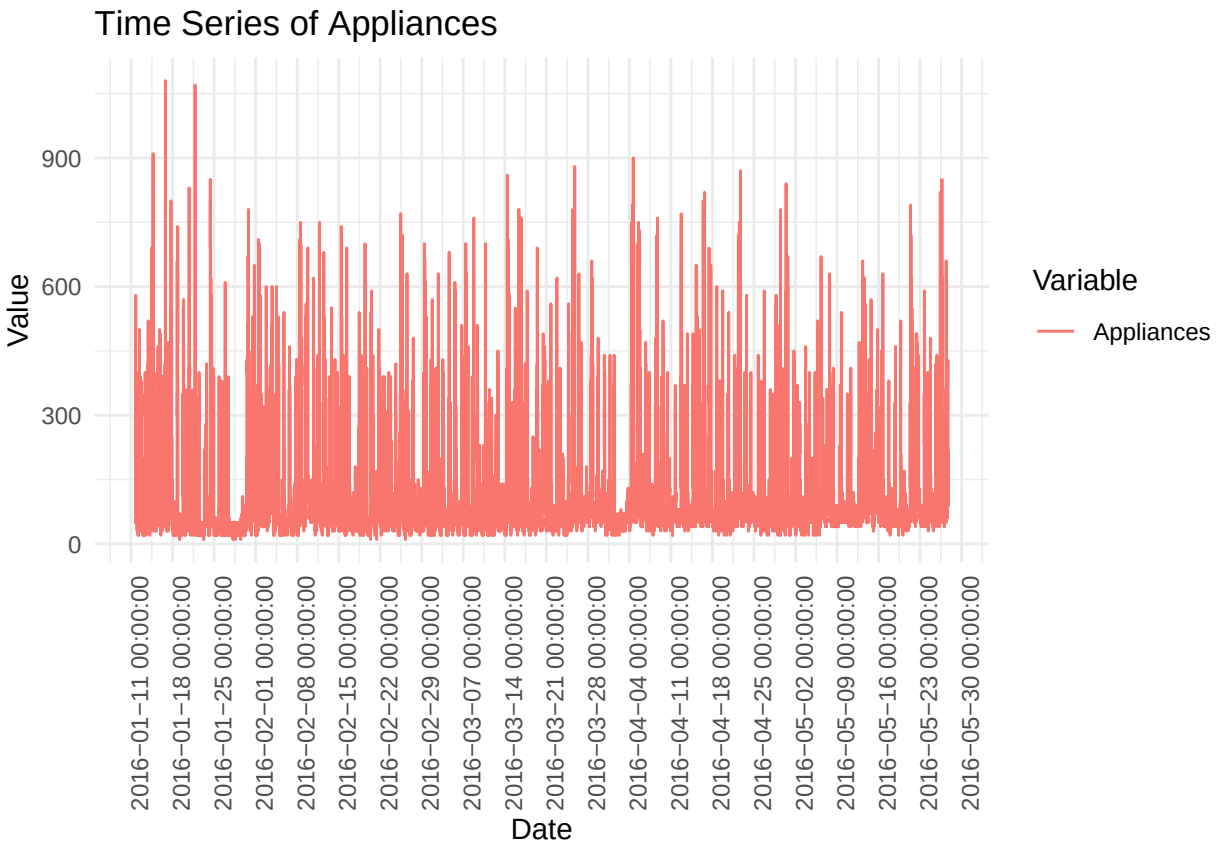
```
# fill in missing values
for(i in 2:(nrow(energy_data)-1)) {
  if(is.na(energy_data$Appliances[i])) {
    energy_data$Appliances[i] <- mean(c(energy_data$Appliances[i-1], energy_data$Appliances[i+1]), na.rm
  }
}
```

```
library(reshape2)
```

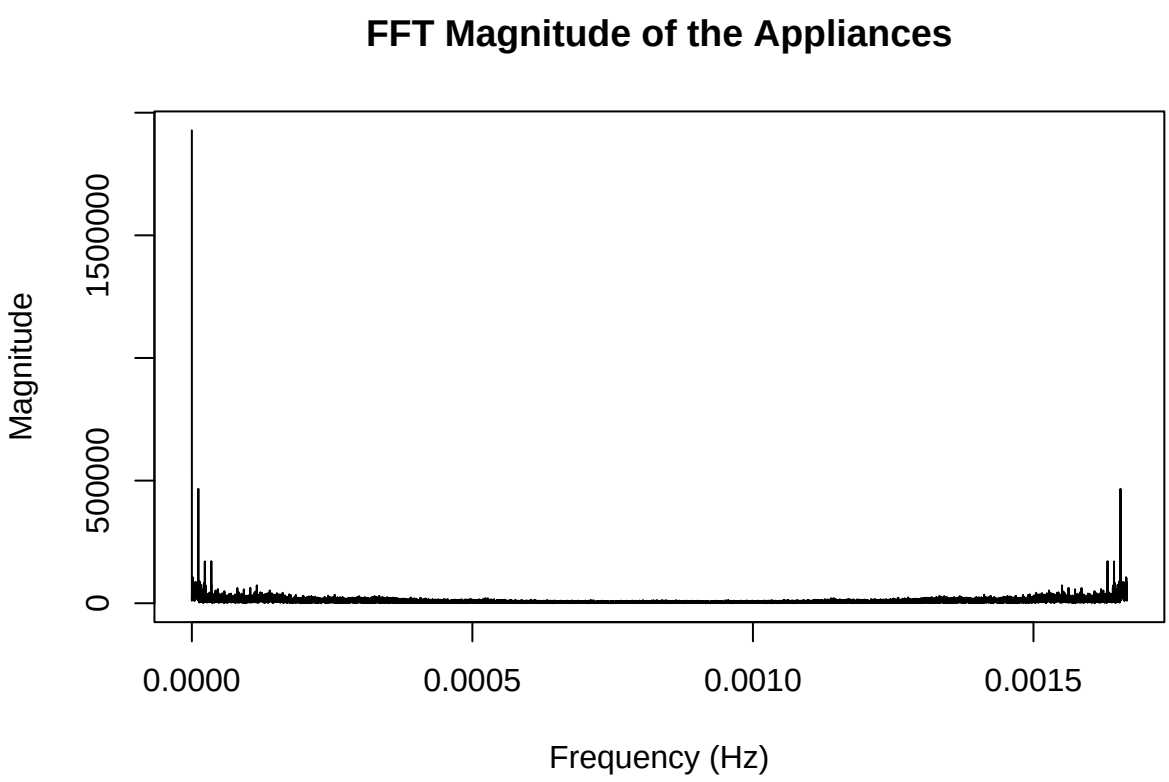
```
## Warning: package 'reshape2' was built under R version 4.3.3
```

```
library(ggplot2)
library(astsa)

energy_data$date <- as.POSIXct(energy_data$date, format="%Y-%m-%d %H:%M:%S", tz="UTC")

energy_data_long <- melt(energy_data, id.vars = "date", measure.vars = c("Appliances"))

ggplot(energy_data_long, aes(x = date, y = value, color = variable)) +
  geom_line() +
  scale_x_datetime(date_labels = "%Y-%m-%d %H:%M:%S", date_breaks = "1 week") +
  labs(x = "Date", y = "Value", color = "Variable") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Time Series of Appliances")
```

```
## Warning: Removed 137 rows containing missing values (`geom_line()`).
```

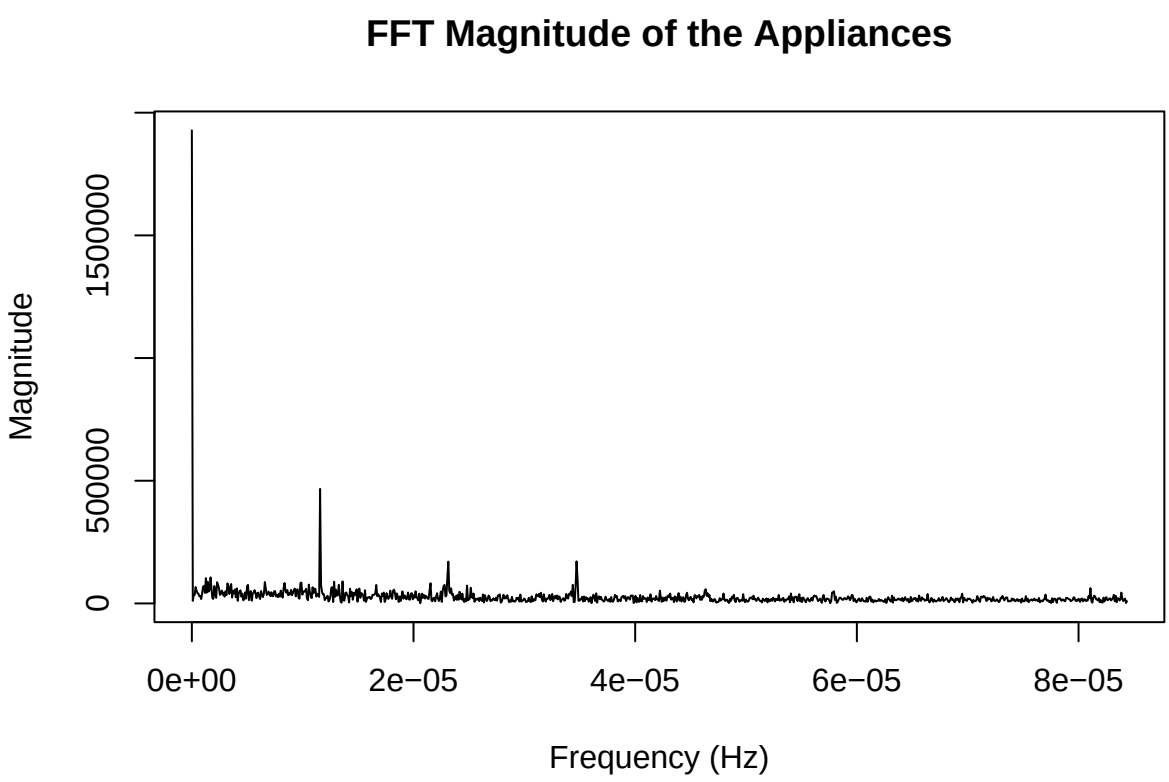



```r
energy_data$date <- as.POSIXct(energy_data$date, format="%Y-%m-%d %H:%M:%S", tz="UTC")

fft_result <- fft(energy_data[[2]])

N <- length(energy_data[[2]])
fs <- 6/3600

frequencies <- (0:(N-1)) * fs / N

magnitudes <- Mod(fft_result)
plot(frequencies, magnitudes, type = 'l', main = "FFT Magnitude of the Appliances", xlab = "Frequency (
```

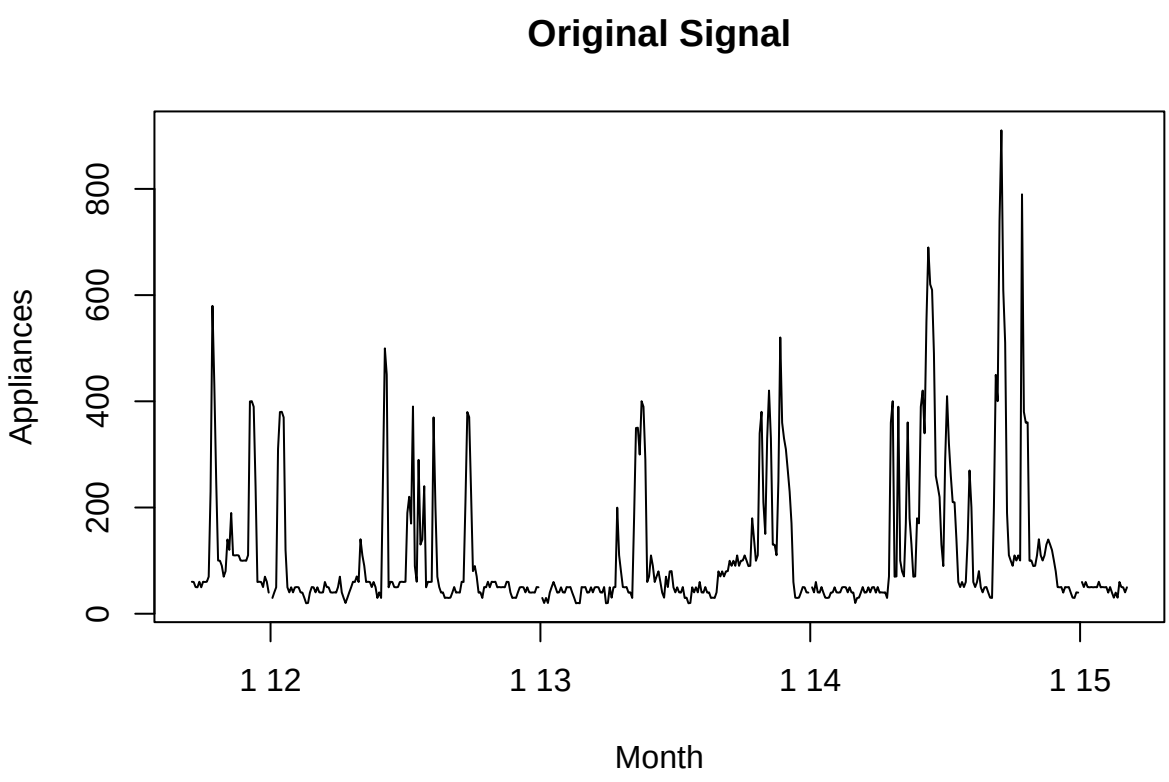


FFT Filtering

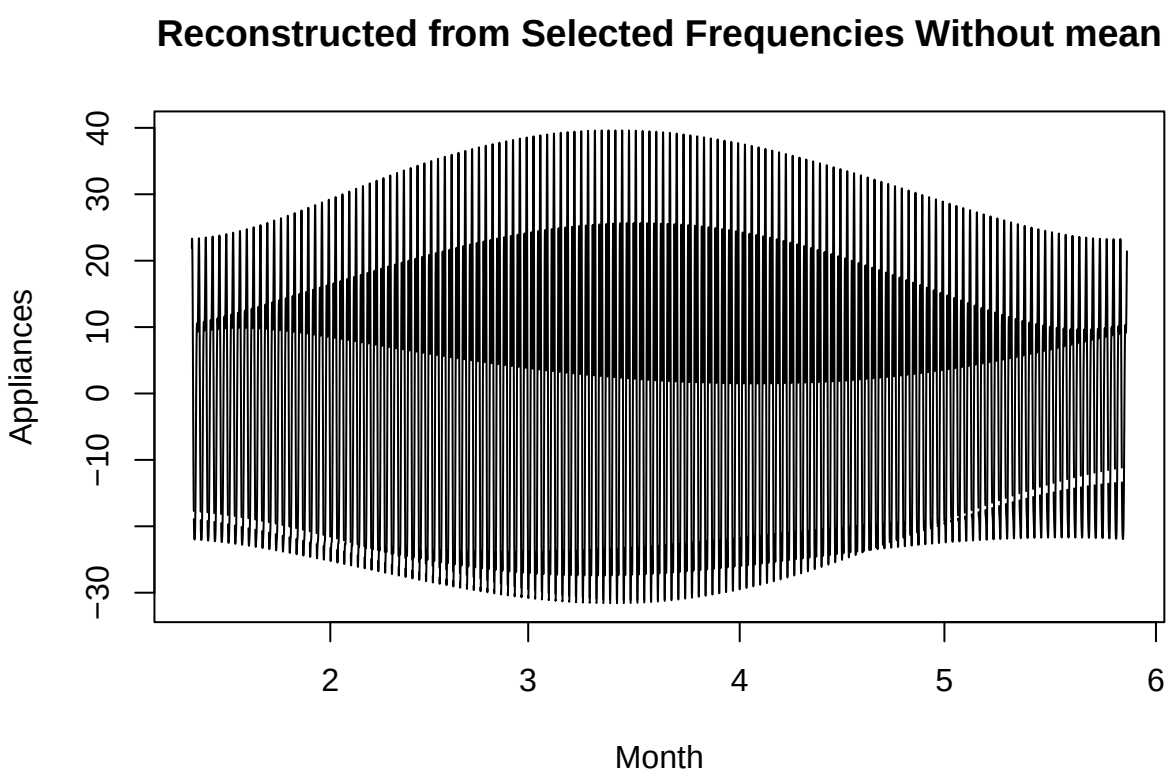
```
fs <- 1 / (600)  #sampling rate , 10 minutes

n <- length(energy_data[[2]])

frequencies <- (0:(n/2-1)) * fs / n

magnitudes <- Mod(fft_result)
magnitudes <- magnitudes[1:(n/2)]

plot(frequencies[0:1000], magnitudes[0:1000], type = 'l', main = "FFT Magnitude of the Appliances", xlab
```



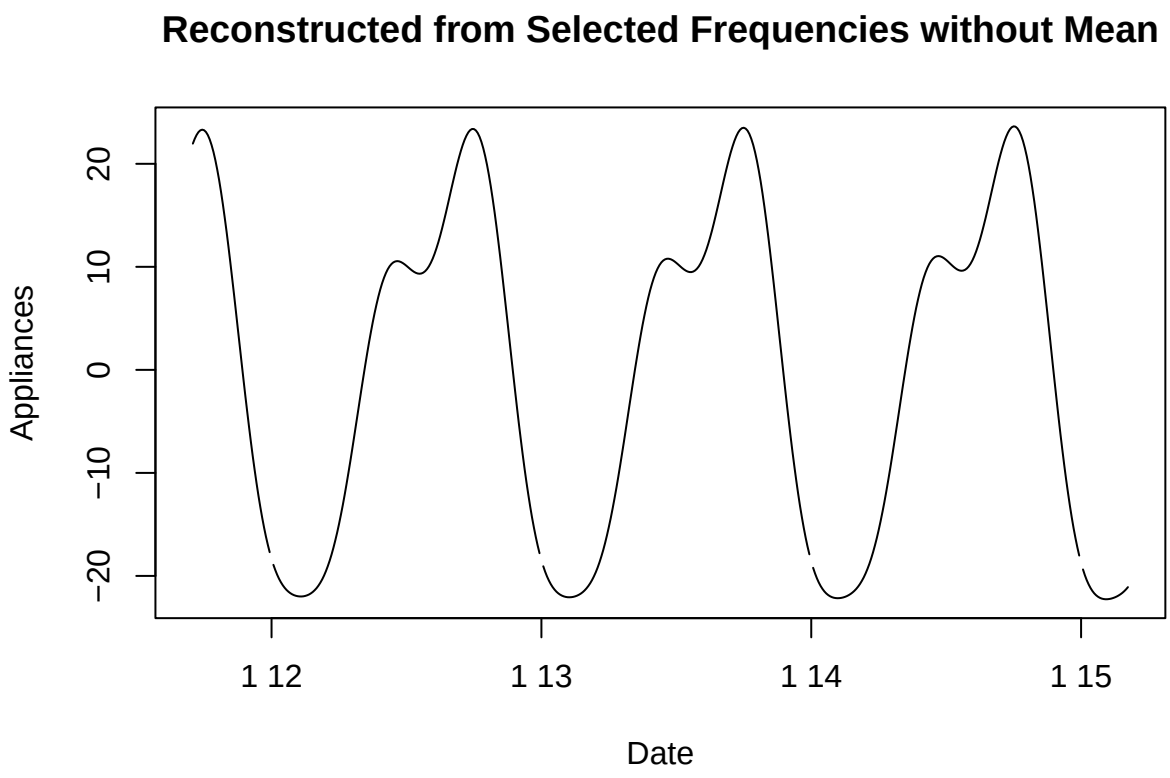

```r
data <- data.frame(index=1:length(frequencies), frequencies=frequencies, magnitudes=magnitudes)
data_sorted <- data[order(-data$magnitudes),]

#filtering top 4
top_frequencies <- data_sorted

fft_filtered <- fft_result
fft_filtered[] <- 0
fft_filtered[top_frequencies$index] <- fft_result[top_frequencies$index]

center_index <- (length(fft_result) + 1) / 2
conjugate_indices <- ifelse(top_frequencies$index < center_index, length(fft_result) - top_frequencies$
conjugate_indices <- conjugate_indices[conjugate_indices != center_index]
fft_filtered[conjugate_indices] <- fft_result[conjugate_indices]

reconstructed_signal <- Re(fft(fft_filtered, inverse = TRUE)) / length(fft_result)

plot(energy_data$date[1:500], reconstructed_signal[1:500], type='l', main="Original Signal", xlab="Mont
```




```r
data <- data.frame(index=1:length(frequencies), frequencies=frequencies, magnitudes=magnitudes)
data_sorted <- data[order(-data$magnitudes),]

#filtering top 4
top_frequencies <- data_sorted[2:4, ]

fft_filtered <- fft_result
fft_filtered[] <- 0
fft_filtered[top_frequencies$index] <- fft_result[top_frequencies$index]

center_index <- (length(fft_result) + 1) / 2
conjugate_indices <- ifelse(top_frequencies$index < center_index, length(fft_result) - top_frequencies$
conjugate_indices <- conjugate_indices[conjugate_indices != center_index]
fft_filtered[conjugate_indices] <- fft_result[conjugate_indices]

reconstructed_signal <- Re(fft(fft_filtered, inverse = TRUE)) / length(fft_result)

plot(energy_data$date, reconstructed_signal, type='l', main="Reconstructed from Selected Frequencies Wi
```



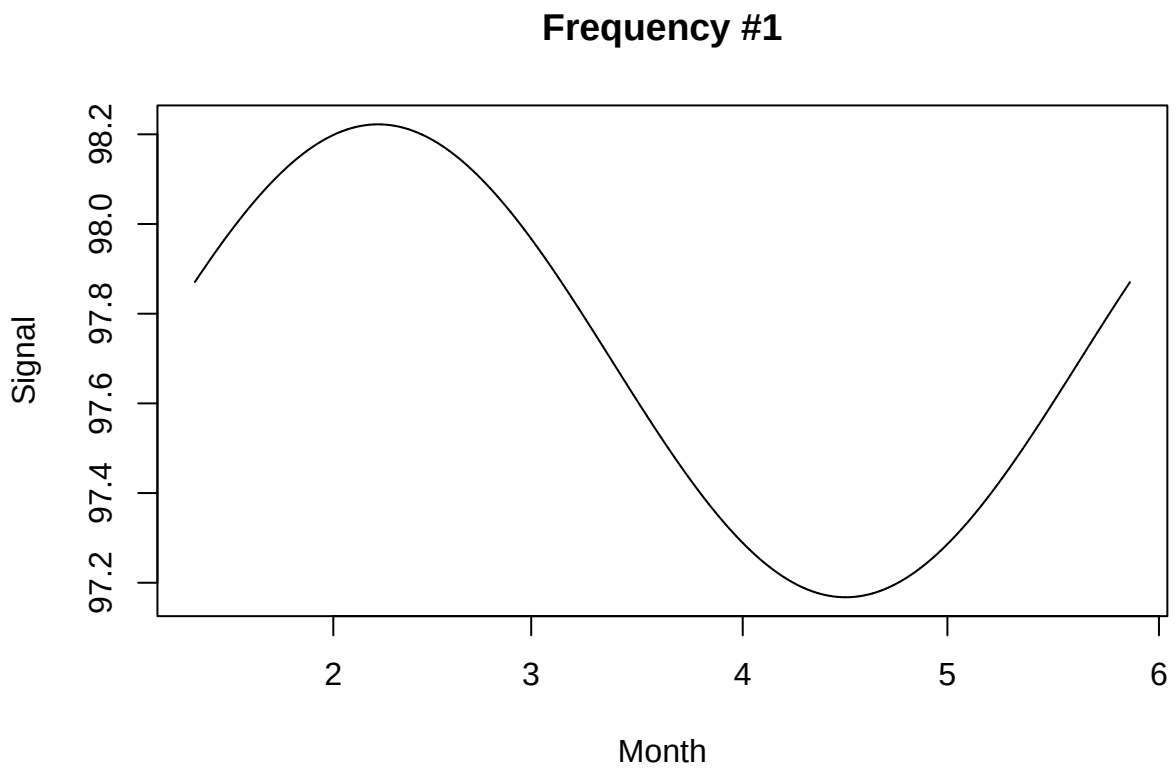

```r
data <- data.frame(index=1:length(frequencies), frequencies=frequencies, magnitudes=magnitudes)
data_sorted <- data[order(-data$magnitudes),]

top_frequencies <- data_sorted[2:4,]

fft_filtered <- fft_result
fft_filtered[] <- 0
fft_filtered[top_frequencies$index] <- fft_result[top_frequencies$index]

center_index <- (length(fft_result) + 1) / 2
conjugate_indices <- ifelse(top_frequencies$index < center_index, length(fft_result) - top_frequencies$i
conjugate_indices <- conjugate_indices[conjugate_indices != center_index]
fft_filtered[conjugate_indices] <- fft_result[conjugate_indices]

reconstructed_signal <- Re(fft(fft_filtered, inverse = TRUE)) / length(fft_result)

plot(energy_data$date[0:500], reconstructed_signal[0:500], type='l', main="Reconstructed from Selected F
```



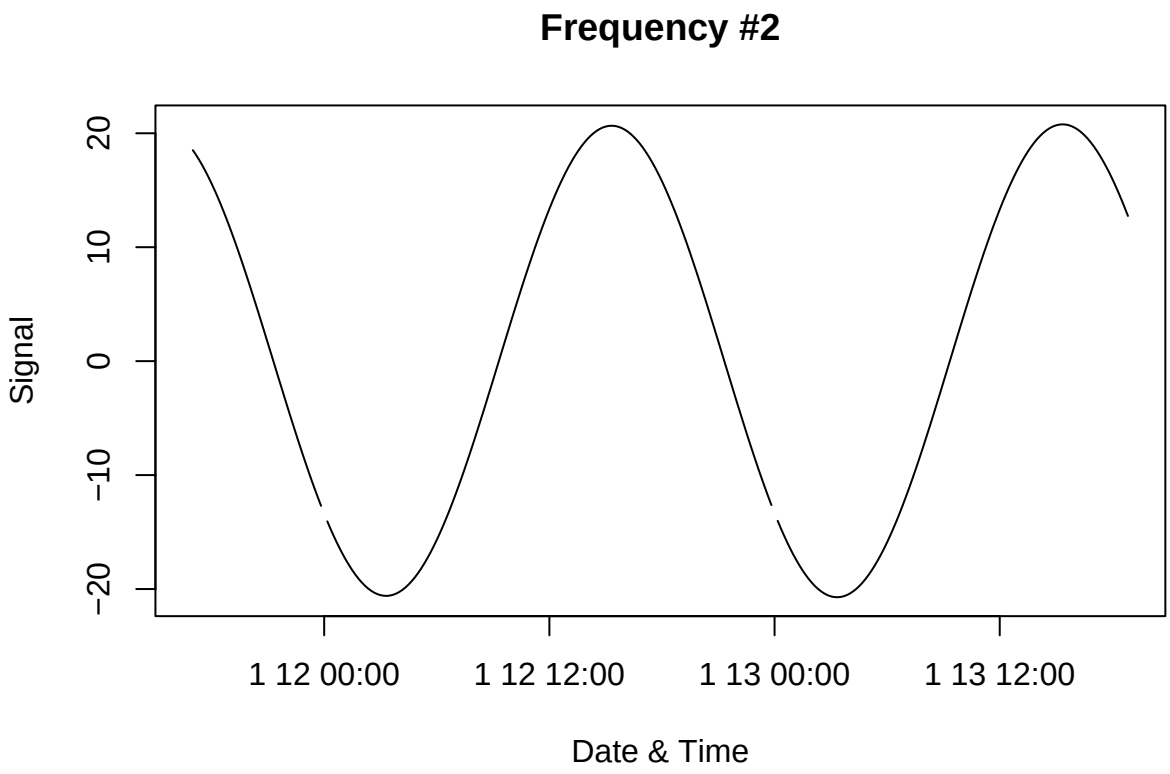

```
top_frequencies
```

```
##     index  frequencies magnitudes
## 138   138 1.156997e-05   466585.5
## 412   412 3.470991e-05   171567.9
## 275   275 2.313994e-05   170487.8
```

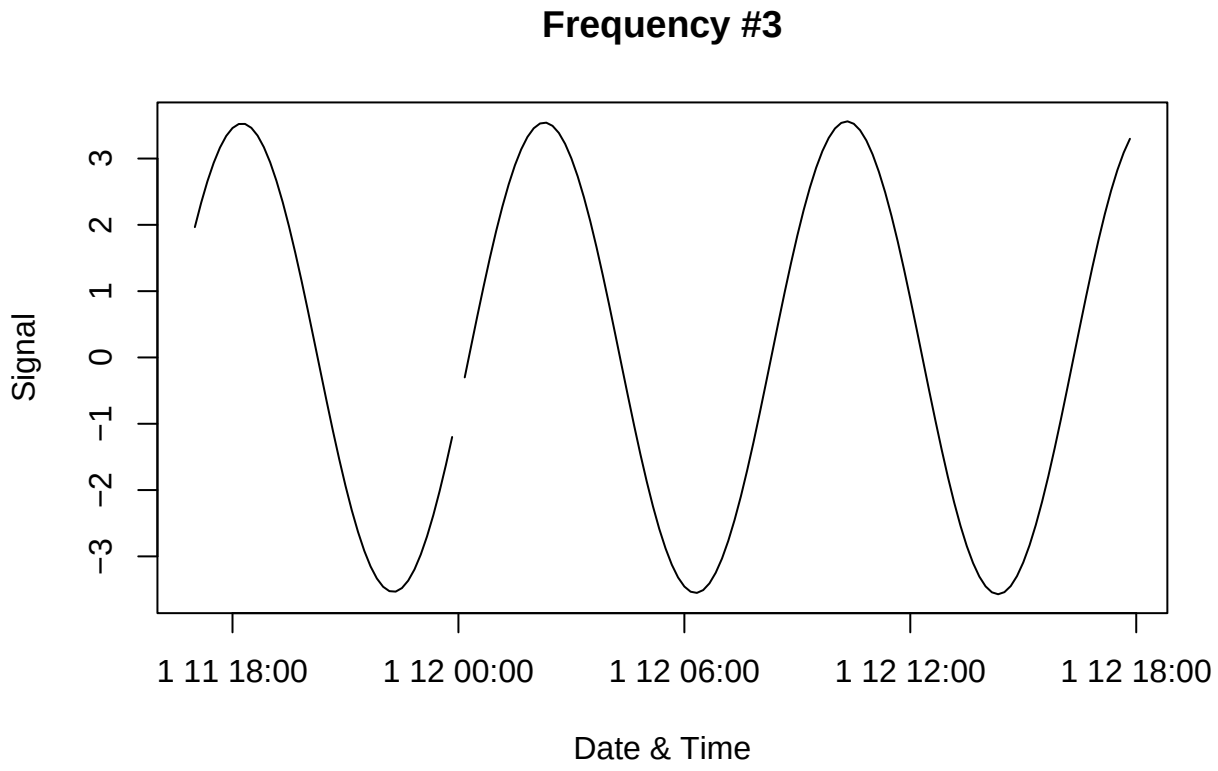
```
top_frequencies <- data_sorted[1, ]

fft_filtered <- fft_result
fft_filtered[] <- 0
fft_filtered[top_frequencies$index] <- fft_result[top_frequencies$index]

center_index <- (length(fft_result) + 1) / 2
conjugate_indices <- ifelse(top_frequencies$index < center_index, length(fft_result) - top_frequencies$i
conjugate_indices <- conjugate_indices[conjugate_indices != center_index]
fft_filtered[conjugate_indices] <- fft_result[conjugate_indices]

reconstructed_signal <- Re(fft(fft_filtered, inverse = TRUE)) / length(fft_result)

plot(energy_data$date, reconstructed_signal, type='l', main="Frequency #1", xlab="Month", ylab="Signal"
```



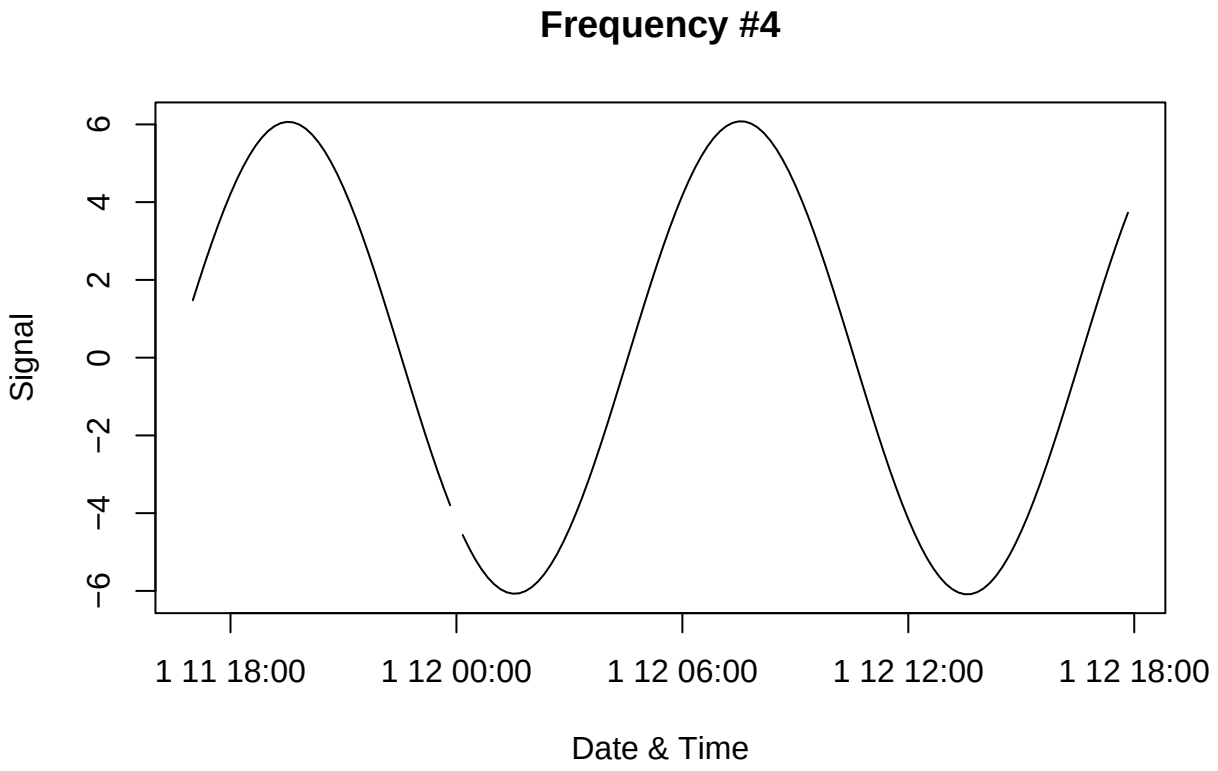

```
top_frequencies <- data_sorted[2, ]

fft_filtered <- fft_result
fft_filtered[] <- 0
fft_filtered[top_frequencies$index] <- fft_result[top_frequencies$index]

center_index <- (length(fft_result) + 1) / 2
conjugate_indices <- ifelse(top_frequencies$index < center_index, length(fft_result) - top_frequencies$
conjugate_indices <- conjugate_indices[conjugate_indices != center_index]
fft_filtered[conjugate_indices] <- fft_result[conjugate_indices]

reconstructed_signal <- Re(fft(fft_filtered, inverse = TRUE)) / length(fft_result)

plot(energy_data$date[0:300], reconstructed_signal[0:300], type='l', main="Frequency #2", xlab="Date & T
```



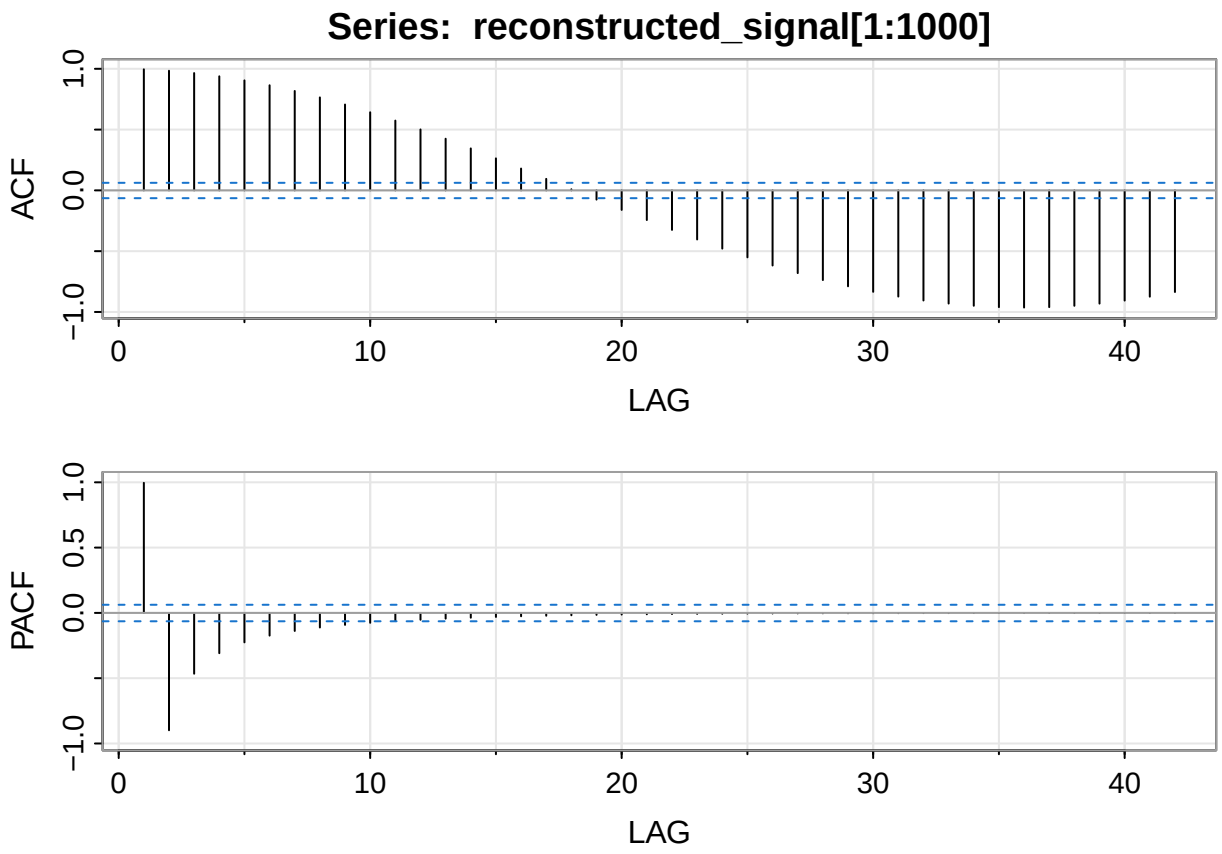

```
top_frequencies <- data_sorted[3, ]

fft_filtered <- fft_result
fft_filtered[] <- 0
fft_filtered[top_frequencies$index] <- fft_result[top_frequencies$index]

center_index <- (length(fft_result) + 1) / 2
conjugate_indices <- ifelse(top_frequencies$index < center_index, length(fft_result) - top_frequencies$i
conjugate_indices <- conjugate_indices[conjugate_indices != center_index]
fft_filtered[conjugate_indices] <- fft_result[conjugate_indices]

reconstructed_signal <- Re(fft(fft_filtered, inverse = TRUE)) / length(fft_result)

plot(energy_data$date[0:150], reconstructed_signal[0:150], type='l', main="Frequency #3", xlab="Date & T
```




```
top_frequencies <- data_sorted[4, ]

fft_filtered <- fft_result
fft_filtered[] <- 0
fft_filtered[top_frequencies$index] <- fft_result[top_frequencies$index]

center_index <- (length(fft_result) + 1) / 2
conjugate_indices <- ifelse(top_frequencies$index < center_index, length(fft_result) - top_frequencies$
conjugate_indices <- conjugate_indices[conjugate_indices != center_index]
fft_filtered[conjugate_indices] <- fft_result[conjugate_indices]

reconstructed_signal <- Re(fft(fft_filtered, inverse = TRUE)) / length(fft_result)

plot(energy_data$date[0:150], reconstructed_signal[0:150], type='l', main="Frequency #4", xlab="Date & '
```

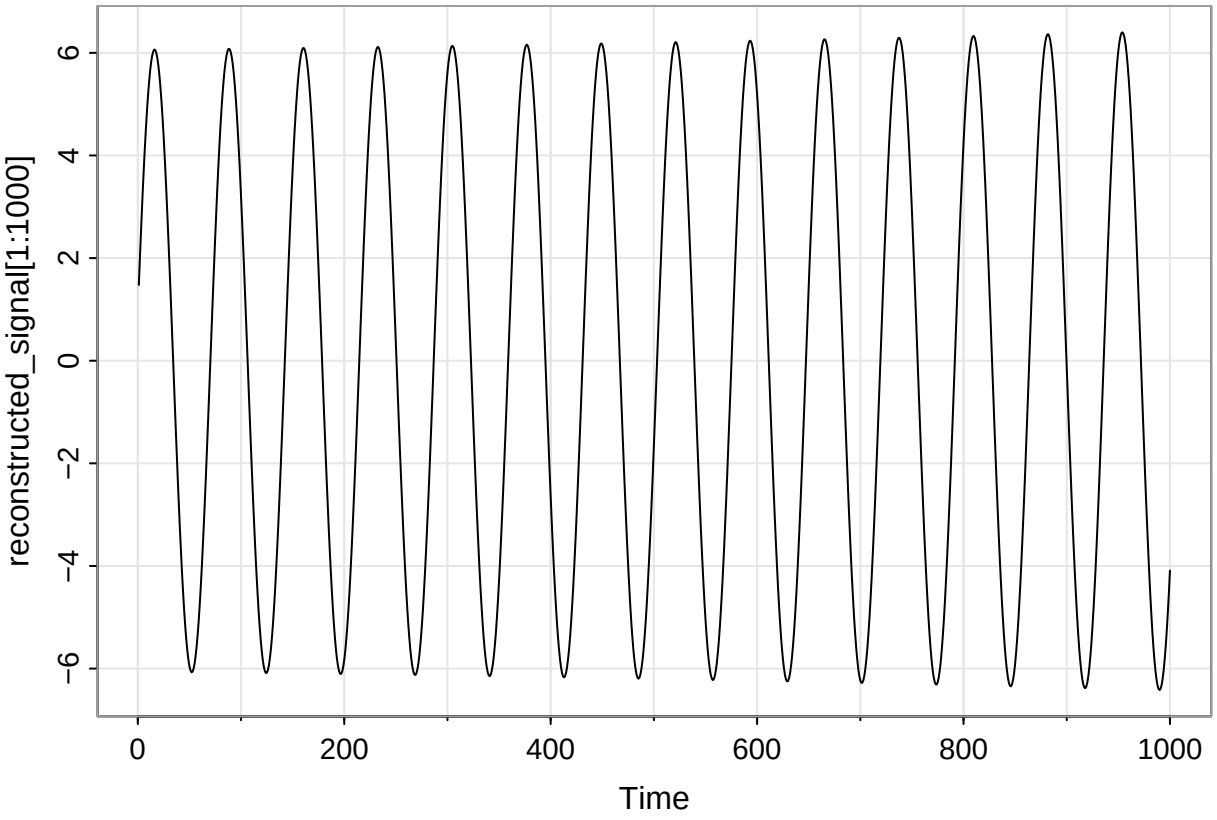



```
acf2(reconstructed_signal[1:1000])
```

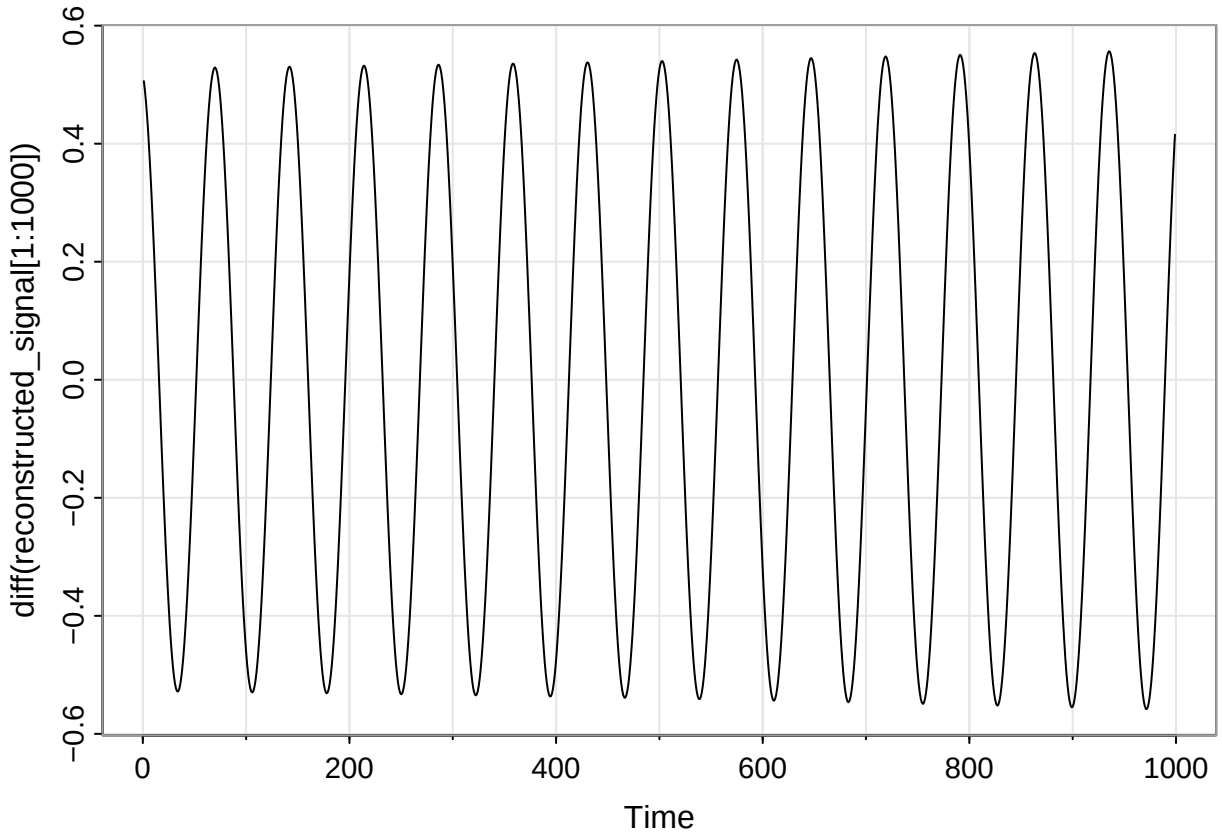



```
##      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF     1  0.98  0.96  0.94  0.90  0.86  0.82  0.76  0.71  0.64  0.57  0.50
## PACF    1 -0.90 -0.47 -0.31 -0.23 -0.17 -0.14 -0.11 -0.09 -0.08 -0.06 -0.05
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.42  0.35  0.26  0.18  0.09  0.01 -0.08 -0.16 -0.24 -0.33 -0.40 -0.48
## PACF -0.04 -0.04 -0.03 -0.03 -0.02 -0.02 -0.02 -0.01 -0.01 -0.01 -0.01 -0.01
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.55 -0.62 -0.68 -0.74 -0.79 -0.83 -0.87 -0.91 -0.93 -0.95 -0.96 -0.96
## PACF -0.01  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
##      [,37] [,38] [,39] [,40] [,41] [,42]
## ACF  -0.96 -0.95 -0.93 -0.91 -0.87 -0.84
## PACF  0.00  0.00  0.00  0.00  0.00  0.00
```

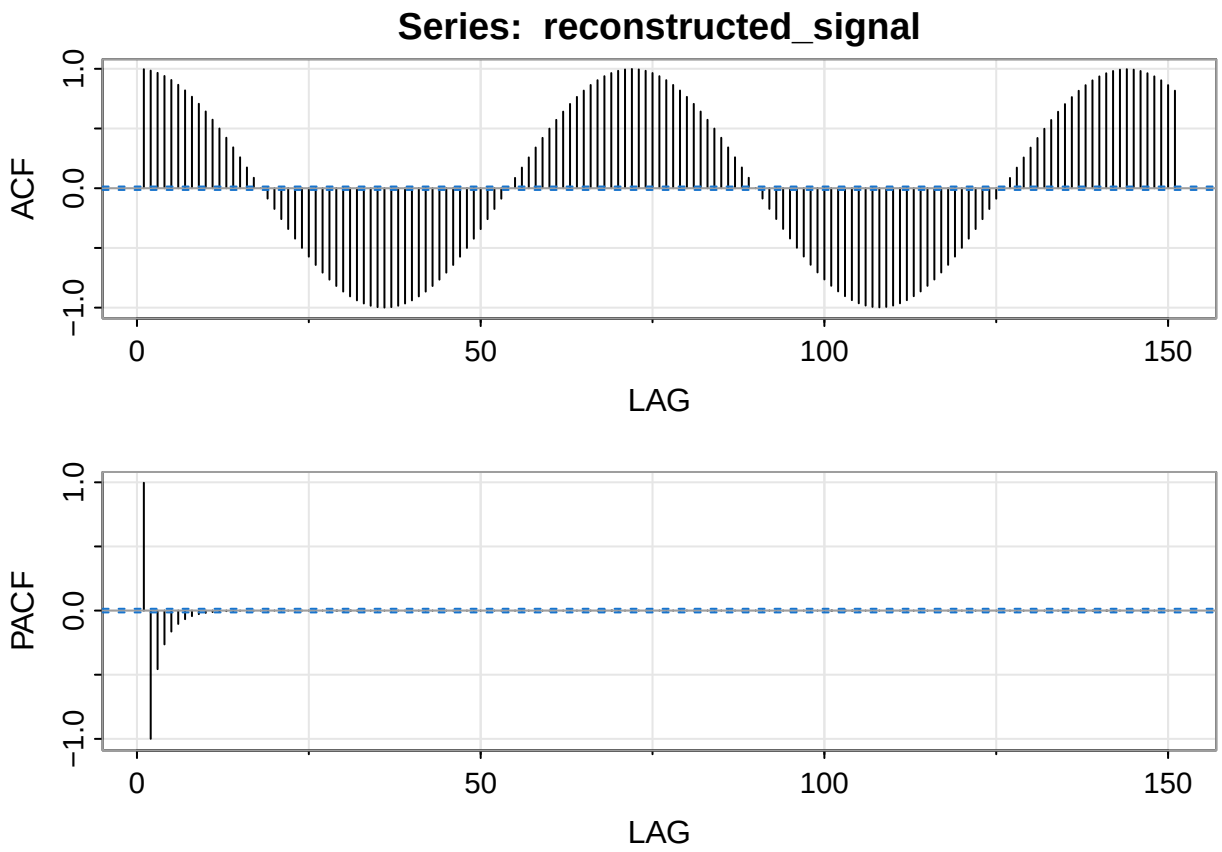
```
tsplot(reconstructed_signal[1:1000])
```




```
tsplot(diff(reconstructed_signal[1:1000]))
```



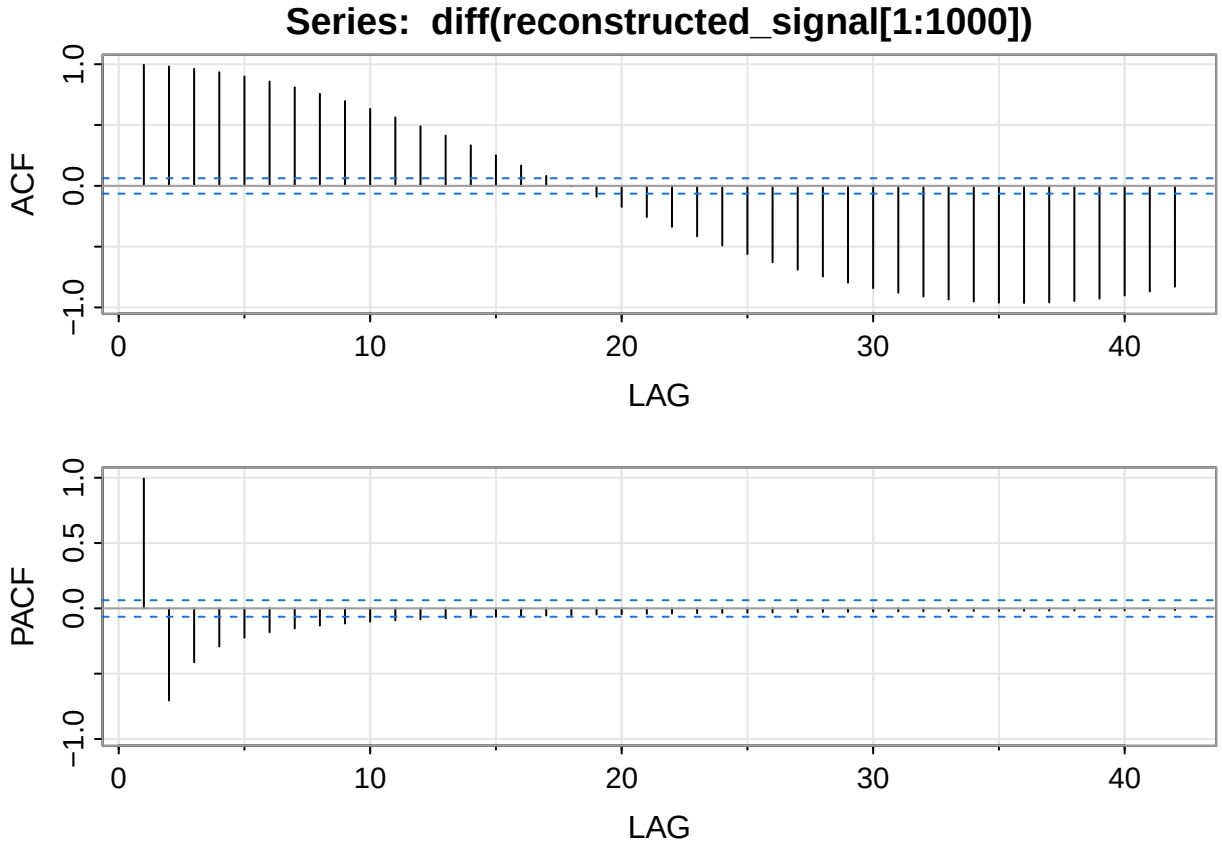

```
acf2(reconstructed_signal)
```




```
##      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF     1  0.98  0.97  0.94  0.91  0.87  0.82  0.77  0.71  0.64  0.57  0.50
## PACF    1 -1.00 -0.46 -0.26 -0.16 -0.10 -0.07 -0.04 -0.03 -0.02 -0.01 -0.01
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.42  0.34  0.26  0.17  0.09     0 -0.09 -0.17 -0.26 -0.34 -0.42  -0.5
## PACF  0.00  0.00  0.00  0.00  0.00     0  0.00  0.00  0.00  0.00  0.00   0.0
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.57 -0.64 -0.71 -0.77 -0.82 -0.87 -0.91 -0.94 -0.97 -0.98    -1    -1
## PACF  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00     0     0
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF     -1 -0.98 -0.97 -0.94 -0.91 -0.87 -0.82 -0.77 -0.71 -0.64 -0.57  -0.5
## PACF     0  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00   0.0
##      [,49] [,50] [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60]
## ACF  -0.42 -0.34 -0.26 -0.17 -0.09     0  0.09  0.17  0.26  0.34  0.42   0.5
## PACF  0.00  0.00  0.00  0.00  0.00     0  0.00  0.00  0.00  0.00  0.00   0.0
##      [,61] [,62] [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72]
## ACF   0.57  0.64  0.71  0.76  0.82  0.86   0.9  0.94  0.96  0.98  0.99     1
## PACF  0.00  0.00  0.00  0.00  0.00  0.00   0.0  0.00  0.00  0.00  0.00     0
##      [,73] [,74] [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84]
## ACF   0.99  0.98  0.96  0.94   0.9  0.86  0.82  0.76  0.71  0.64  0.57   0.5
## PACF  0.00  0.00  0.00  0.00   0.0  0.00  0.00  0.00  0.00  0.00  0.00   0.0
##      [,85] [,86] [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96]
## ACF   0.42  0.34  0.26  0.17  0.09     0 -0.09 -0.17 -0.26 -0.34 -0.42  -0.5
## PACF  0.00  0.00  0.00  0.00  0.00     0  0.00  0.00  0.00  0.00  0.00   0.0
##      [,97] [,98] [,99] [,100] [,101] [,102] [,103] [,104] [,105] [,106] [,107]
## ACF  -0.57 -0.64 -0.71  -0.76  -0.82  -0.86   -0.9  -0.94  -0.96  -0.98  -0.99
```

```
## PACF  0.00  0.00  0.00   0.00   0.00   0.00    0.0   0.00   0.00   0.00   0.00
##      [,108] [,109] [,110] [,111] [,112] [,113] [,114] [,115] [,116] [,117]
## ACF      -1  -0.99  -0.98  -0.96  -0.94   -0.9  -0.86  -0.82  -0.76  -0.71
## PACF      0   0.00   0.00   0.00   0.00    0.0   0.00   0.00   0.00   0.00
##      [,118] [,119] [,120] [,121] [,122] [,123] [,124] [,125] [,126] [,127]
## ACF   -0.64  -0.57   -0.5  -0.42  -0.34  -0.26  -0.17  -0.09      0   0.09
## PACF   0.00   0.00    0.0   0.00   0.00   0.00   0.00   0.00      0   0.00
##      [,128] [,129] [,130] [,131] [,132] [,133] [,134] [,135] [,136] [,137]
## ACF    0.17   0.26   0.34   0.42    0.5   0.57   0.64    0.7   0.76   0.82
## PACF   0.00   0.00   0.00   0.00    0.0   0.00   0.00    0.0   0.00   0.00
##      [,138] [,139] [,140] [,141] [,142] [,143] [,144] [,145] [,146] [,147]
## ACF    0.86    0.9   0.94   0.96   0.98   0.99      1   0.99   0.98   0.96
## PACF   0.00    0.0   0.00   0.00   0.00   0.00      0   0.00   0.00   0.00
##      [,148] [,149] [,150] [,151]
## ACF    0.94    0.9   0.86   0.82
## PACF   0.00    0.0   0.00   0.00
```

```
acf2(diff(reconstructed_signal[1:1000]))
```




```
##      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  0.99  0.98  0.96  0.93  0.90  0.86  0.81  0.76  0.70  0.63  0.56  0.49
## PACF 0.99 -0.71 -0.41 -0.29 -0.22 -0.18 -0.15 -0.13 -0.12 -0.10 -0.09 -0.08
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.41  0.33  0.25  0.17  0.08  0.00 -0.09 -0.17 -0.26 -0.34 -0.42 -0.49
## PACF -0.08 -0.07 -0.06 -0.06 -0.05 -0.05 -0.05 -0.04 -0.04 -0.04 -0.04 -0.03
```

```
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.56 -0.63 -0.69 -0.75 -0.80 -0.84 -0.88 -0.91 -0.93 -0.95 -0.96 -0.96
## PACF -0.03 -0.03 -0.03 -0.03 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02
##      [,37] [,38] [,39] [,40] [,41] [,42]
## ACF  -0.96 -0.95 -0.93 -0.90 -0.87 -0.83
## PACF -0.02 -0.02 -0.02 -0.02 -0.01 -0.01
```

```
#sarima(reconstructed_signal[1:1000],no.constant = TRUE, p=3, d=0, q=0)
#sarima.for(reconstructed_signal, n.ahead = 10, p=3, d=0, q=0)
#sarima.for(reconstructed_signal[1:1000], n.ahead = 10, p=2, d=0, q=0, P=1, D=0, Q=0, S=50)



library(fpp2)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## -- Attaching packages --------------------------------------------- fpp2 2.5 --
```

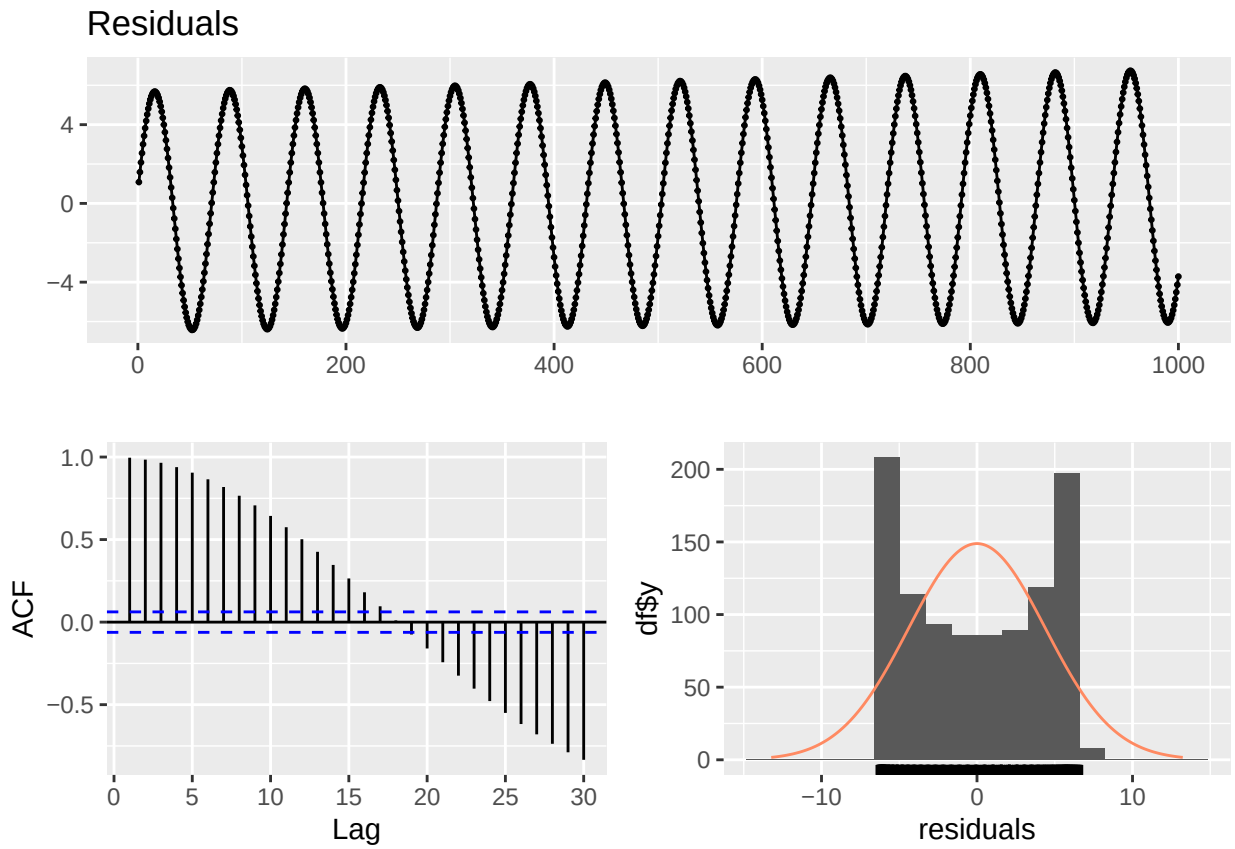
```
## v forecast  8.21.1     v expsmooth 2.3
## v fma       2.5
```

```
##
```

```
##
## Attaching package: 'fpp2'
```

```
## The following object is masked from 'package:astsa':
##
##     oil
```

```
model = lm(reconstructed_signal[1:1000] ~ time(reconstructed_signal[1:1000]))
checkresiduals(model, test="LB")
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 7562.4, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```