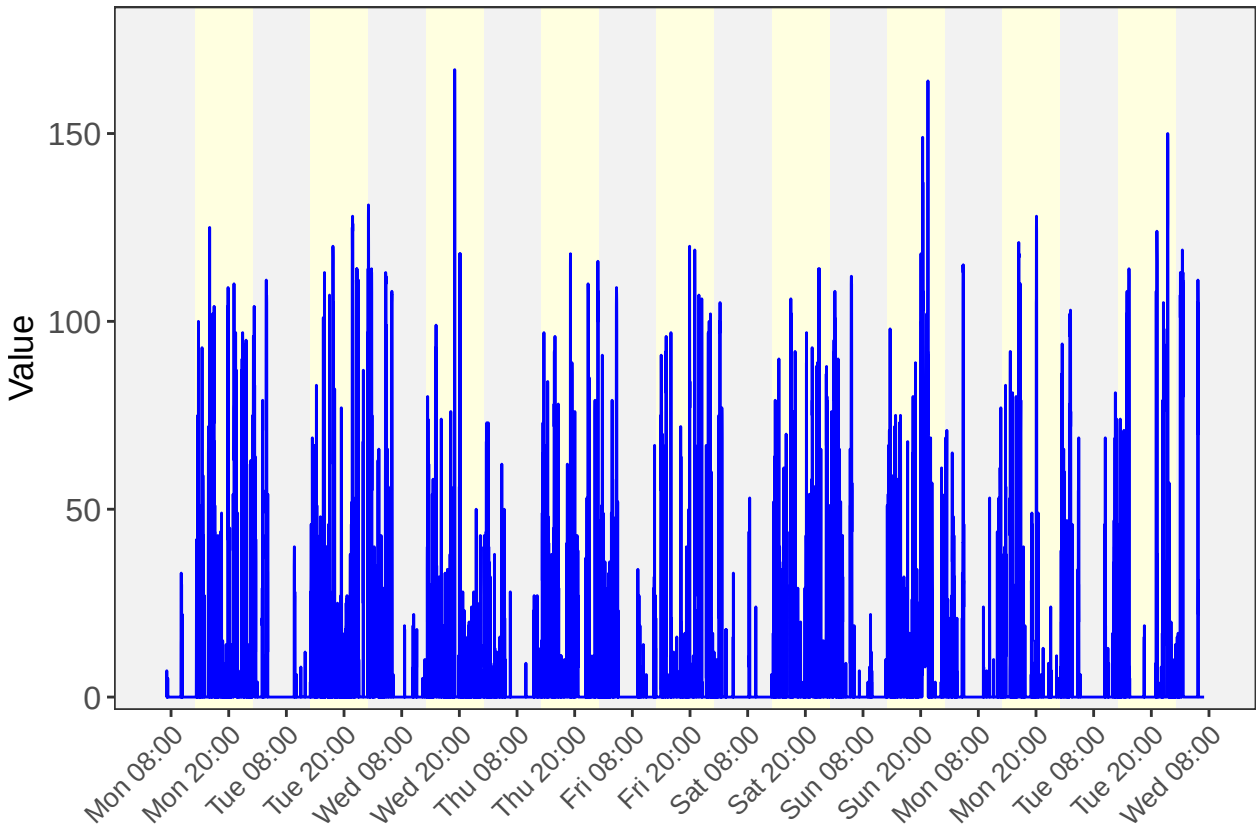
Compare `laggedCor` with classic `ccf`

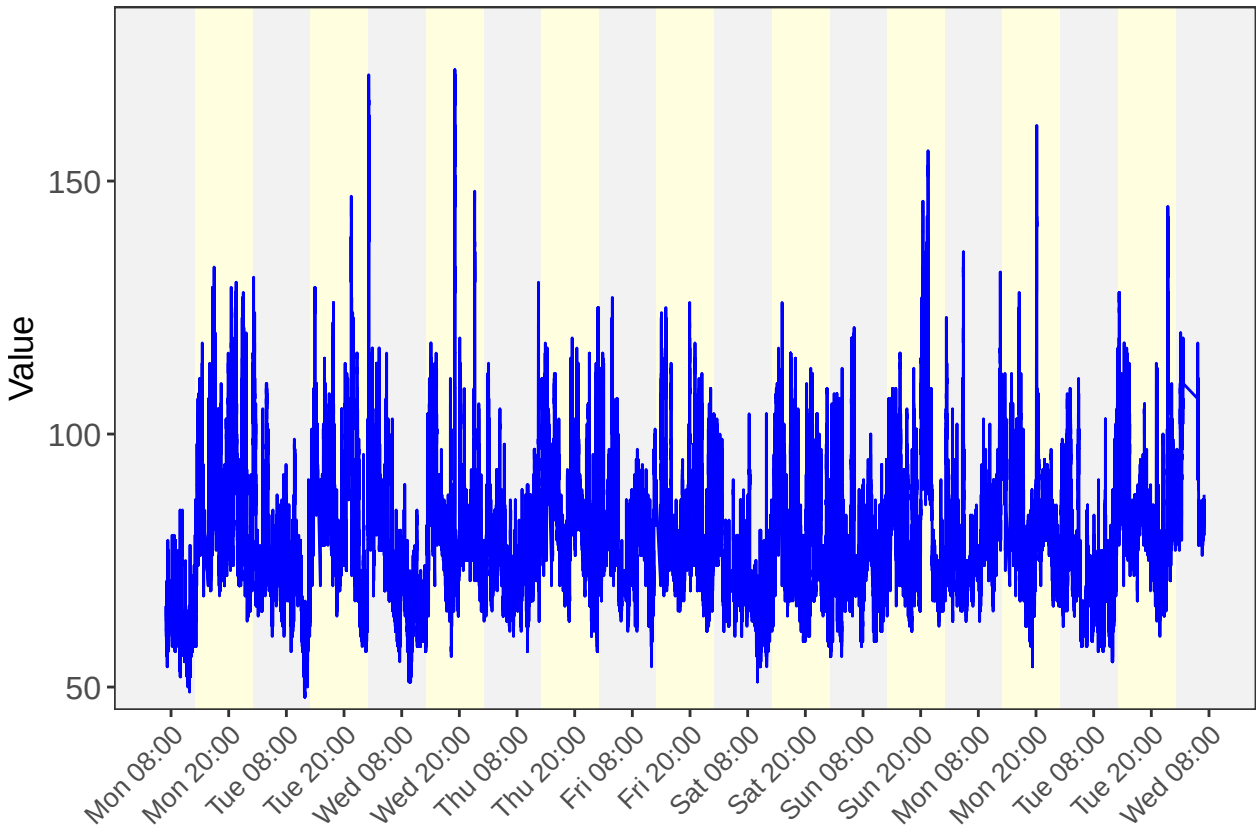
```
library(laggedcor)
library(funtimes)
library(xts)
library(lubridate)
library(zoo)
```

```
data("step_data")
data("heart_data")
```

```
time_plot(step_data$step, step_data$time)
```




```
time_plot(heart_data$heart, heart_data$time)
```




```r
# Function to align and resample time series
align_timeseries <- function(ts1, ts2,
                             time1, time2,
                             target_freq = "1 min") {
    # Convert to xts objects
    xts1 <- xts(ts1, order.by = time1)
    xts2 <- xts(ts2, order.by = time2)

    # Find common time range
    start_time <- max(min(time1), min(time2))
    end_time <- min(max(time1), max(time2))

    # Create regular time sequence at target frequency
    regular_times <- seq(from = start_time,
                         to = end_time,
                         by = target_freq)

    # Resample both series to regular grid
    # Using approximation for irregular series
    aligned1 <- approx(x = time1, y = ts1,
                       xout = regular_times,
                       method = "linear")$y

    aligned2 <- approx(x = time2, y = ts2,
                       xout = regular_times,
                       method = "linear")$y
```

```
    return(list(
        time = regular_times,
        ts1 = aligned1,
        ts2 = aligned2,
        freq = target_freq
    ))
}
```

```
# Function to analyze sampling rates
analyze_sampling <- function(time1, time2) {
    # Calculate intervals
    diff1 <- diff(time1)
    diff2 <- diff(time2)

    # Summary statistics
    summary1 <- summary(as.numeric(diff1, units = "secs"))
    summary2 <- summary(as.numeric(diff2, units = "secs"))

    # Print results
    cat("Series 1 sampling intervals (seconds):\n")
    print(summary1)
    cat("\nSeries 2 sampling intervals (seconds):\n")
    print(summary2)

    # Suggest target frequency
    suggest_freq <- max(median(diff1), median(diff2))
    cat("\nSuggested resampling frequency:", suggest_freq, "seconds\n")

    return(suggest_freq)
}
```

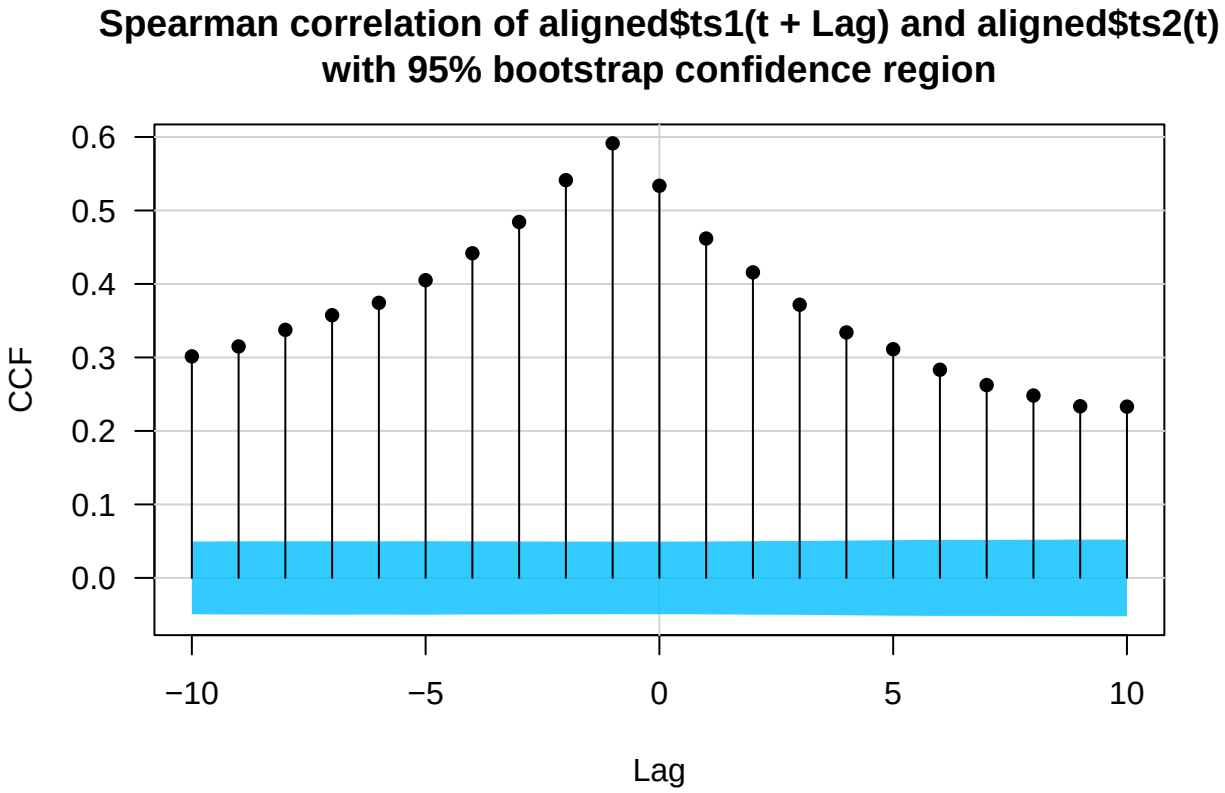
```
process_and_analyze <- function(ts1,
                                time1,
                                ts2,
                                time2,
                                plot_stat = "Spearman",
                                target_freq = NULL,
                                time_tol = 60) {
  # Analyze sampling rates
  suggested_freq <- analyze_sampling(time1, time2)

  # Use suggested frequency if none provided
  if (is.null(target_freq)) {
    target_freq <- paste(round(suggested_freq), "sec")
  }

  # Align time series
  aligned <- align_timeseries(
    ts1 = ts1,
    time1 = time1,
    ts2 = ts2,
    time2 = time2,
    target_freq = target_freq
```

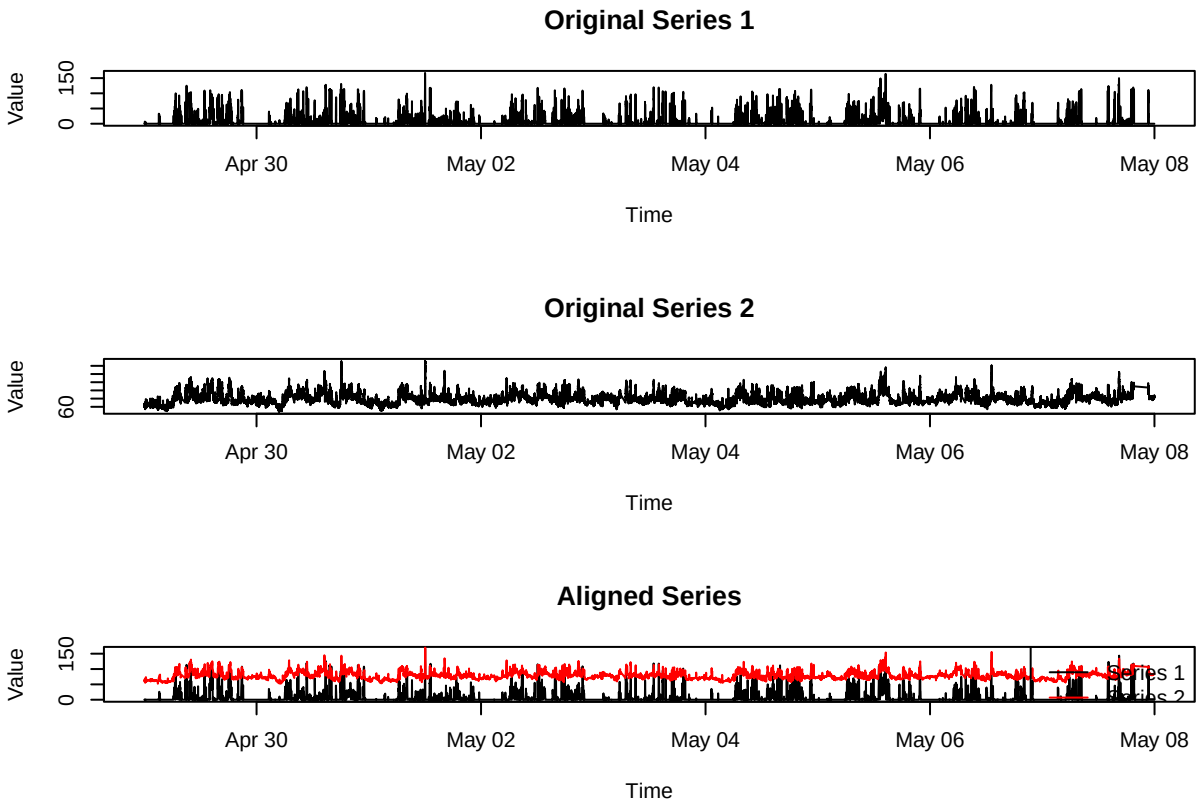
```
)

# Calculate CCF
ccf_result <- ccf_boot(aligned$ts1,
                       aligned$ts2,
                       lag.max = time_tol,
                       plot = plot_stat)

# Plot original and resampled data
par(mfrow = c(3, 1))

# Original series 1
plot(
  time1,
  ts1,
  type = "l",
  main = "Original Series 1",
  xlab = "Time",
  ylab = "Value"
)

# Original series 2
plot(
  time2,
  ts2,
  type = "l",
  main = "Original Series 2",
  xlab = "Time",
  ylab = "Value"
)

# Aligned series
plot(
  aligned$time,
  aligned$ts1,
  type = "l",
  main = "Aligned Series",
  xlab = "Time",
  ylab = "Value"
)
lines(aligned$time, aligned$ts2, col = "red")
legend(
  "topright",
  c("Series 1", "Series 2"),
  col = c("black", "red"),
  lty = 1
)

# Plot CCF
par(mfrow = c(1, 1))
plot(ccf_result, main = "Cross-correlation")

return(
```

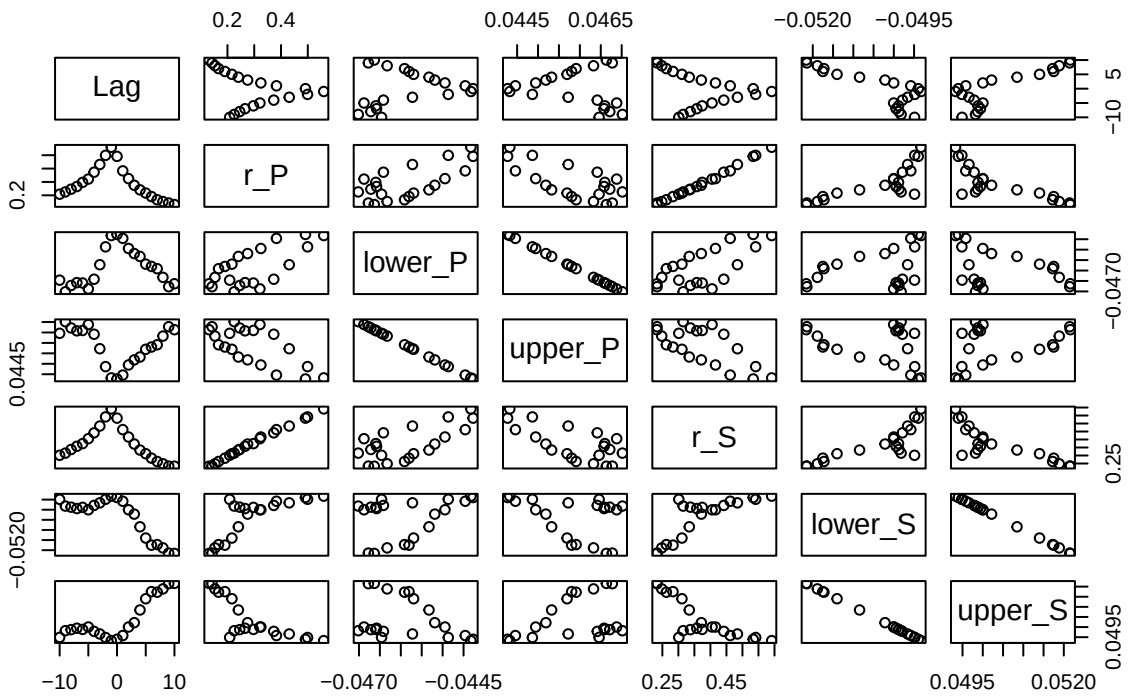
```
    list(
      aligned_data = aligned,
      ccf = ccf_result,
      original_freq1 = median(diff(time1)),
      original_freq2 = median(diff(time2)),
      target_freq = target_freq
    )
  )
}
```

```
start_time <- Sys.time()
res <- process_and_analyze(
  ts1 = step_data$step,
  time1 = step_data$time,
  ts2 = heart_data$heart,
  time2 = heart_data$time,
  time_tol = 0.2 * 60 - 2,
  target_freq = "120 sec"
)
```

```
## Series 1 sampling intervals (seconds):
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      60      60      60      60      60      60
##
## Series 2 sampling intervals (seconds):
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     1.000     5.000     5.000     8.456    10.000 10805.000
##
## Suggested resampling frequency: 60 seconds
```


Spearman correlation of aligned$ts1(t + Lag) and aligned$ts2(t)
with 95% bootstrap confidence region
CCF
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−10
−5
0
5
10
Lag


Original Series 1
Value
0
150
Apr 30
May 02
May 04
May 06
May 08
Time
Original Series 2
Value
60
Apr 30
May 02
May 04
May 06
May 08
Time
Aligned Series
Value
0
150
Series 1
Apr 30
May 02
May 04
May 06
May 08
Time




```
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 9.589478 secs
```

```
max_lag <- which.max(abs(res$ccf$r_S))
detected_lag <- res$ccf$Lag[max_lag]
detected_lag
```

```
## [1] -1
```

```
start_time <- Sys.time()
lagged_res <- calculate_lagged_correlation(
  x = step_data$step,
  y = heart_data$heart,
  time1 = step_data$time,
  time2 = heart_data$time,
  step = 2/60,
  time_tol = 0.2,
  threads = 16
)
```

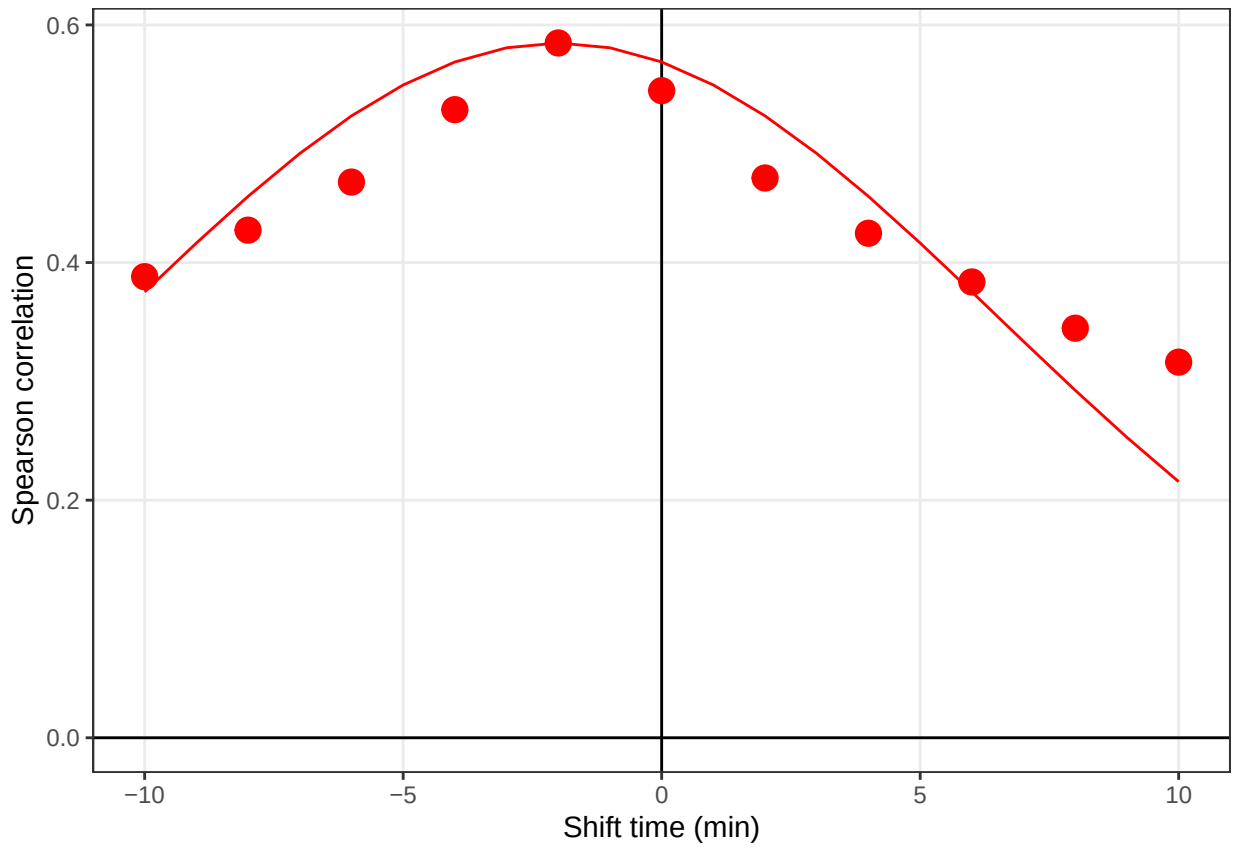
```
##   |                                                                      |
```

```r
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 6.613212 mins
```

```r
evaluate_lagged_cor(object = lagged_res, plot = TRUE)
```

```
## $score
## [1] 0.9973992
##
## $plot
```




```r
extract_max_cor(lagged_res)
```

```
##   (-3,-1]
## 0.5848359
```