Test `laggedcor`, `ccf`, and `ccf_boost` on generated data

```r
# installed using renv::install("matchy233/laggedcor@develop")
library(laggedcor)
```

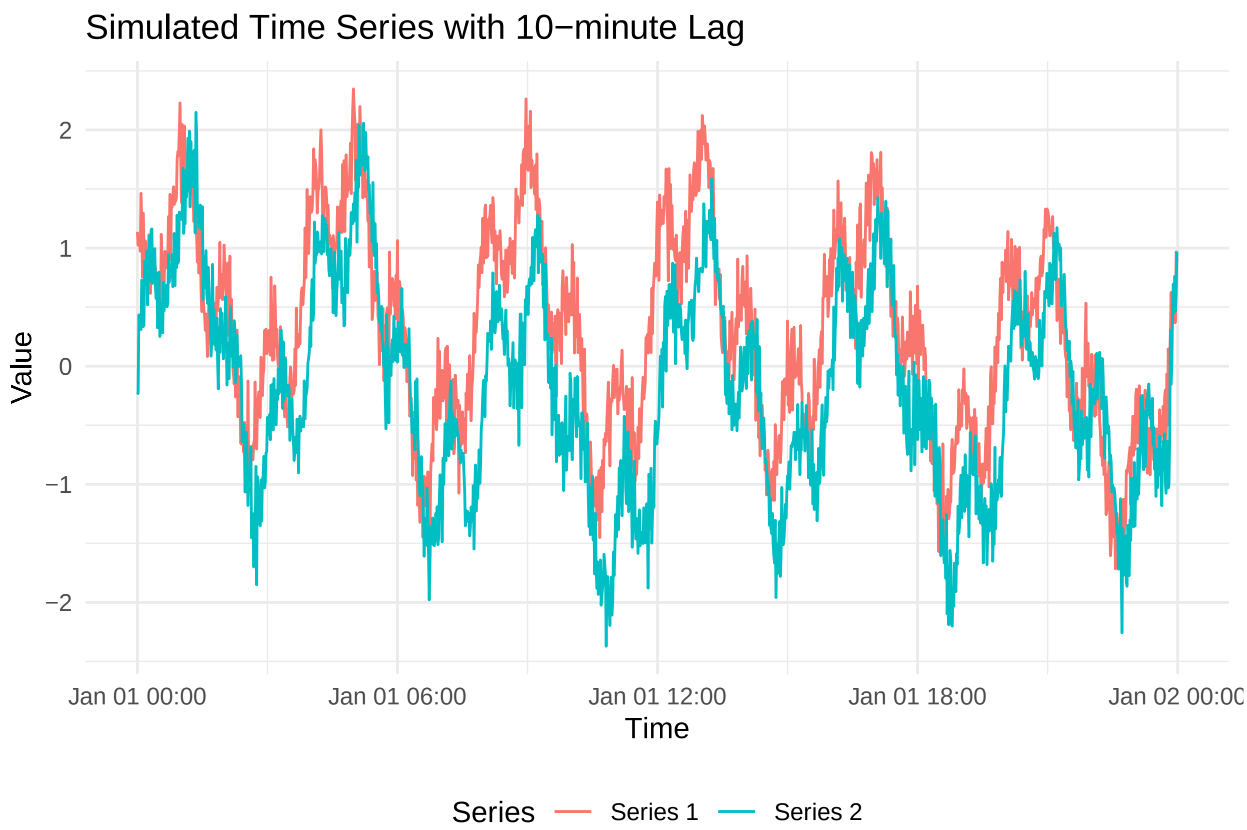
```r
library(lubridate)
library(dplyr)
library(ggplot2)
library(funtimes)
```

```r
# Set random seed for reproducibility
set.seed(123)
```

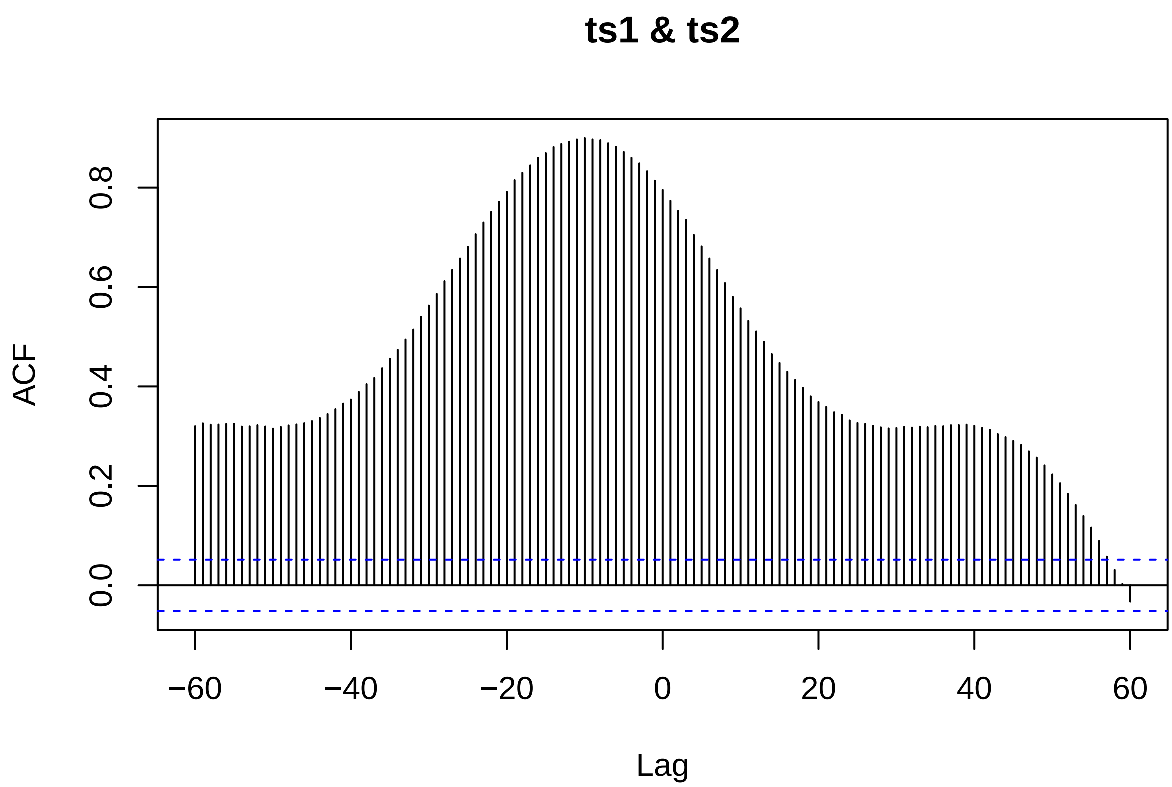
```r
# Generate time series parameters
n_hours <- 24  # 24 hours of data
freq <- 1/60   # One observation per minute
n <- n_hours * 60  # Total number of observations
lag_minutes <- 10  # Lag in minutes

# Generate time sequence
time_seq <- seq(
  from = ymd_hms("2024-01-01 00:00:00"),
  by = sprintf("%d sec", 60),  # One minute intervals
  length.out = n
)

# Generate base signal (combination of sine waves for complexity)
base_signal <- sin(2 * pi * seq_len(n + lag_minutes) / (60 * 4)) +  # 4-hour cycle
               0.5 * sin(2 * pi * seq_len(n + lag_minutes) / (60 * 1)) +  # 1-hour cycle
               0.3 * sin(2 * pi * seq_len(n + lag_minutes) / (60 * 12))   # 12-hour cycle

# Add different types of noise
generate_noise <- function(n, sd = 0.1) {
  # Combine white noise and random walk
  white_noise <- rnorm(n, mean = 0, sd = sd)
  random_walk <- cumsum(rnorm(n, mean = 0, sd = sd/5))
  # Normalize random walk
  random_walk <- random_walk * (sd / sd(random_walk))
  return(white_noise + random_walk)
}

# Create the two time series with lag
ts1 <- base_signal + generate_noise(n + lag_minutes, sd = 0.2)
ts1 <- ts1[(lag_minutes + 1):(n + lag_minutes)]
ts2 <- base_signal[1:(n-lag_minutes)] + generate_noise(n, sd = 0.2)



# Create data frame
df <- data.frame(
  timestamp = time_seq,
  series1 = ts1,
```

```r
  series2 = ts2
)
```

```r
# Plot the time series
ggplot(df, aes(x = timestamp)) +
  geom_line(aes(y = series1, color = "Series 1")) +
  geom_line(aes(y = series2, color = "Series 2")) +
  labs(title = "Simulated Time Series with 10-minute Lag",
       x = "Time",
       y = "Value",
       color = "Series") +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```r
# Calculate CCF
start_time <- Sys.time()
ccf_result <- ccf(ts1, ts2, lag.max = 60, plot = TRUE)  # Check up to 60 minutes lag
```



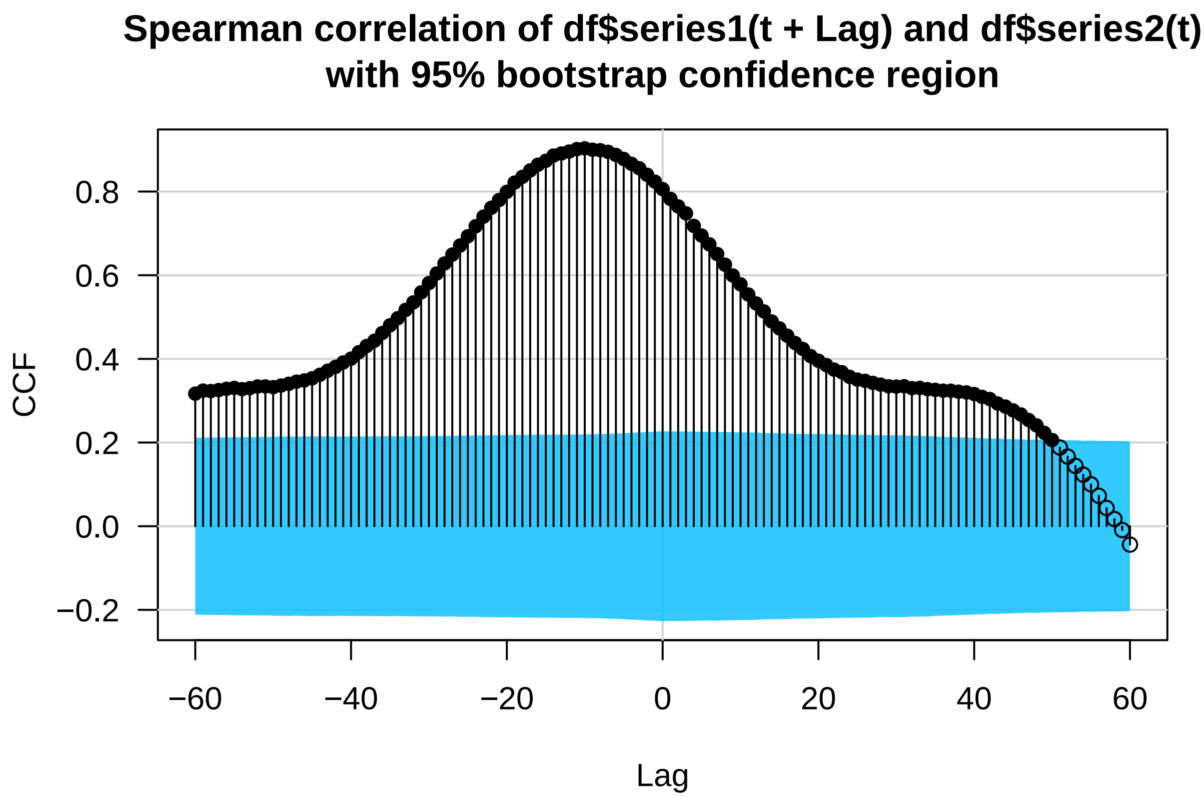

```
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 0.003981352 secs
```

```
max_lag <- which.max(abs(ccf_result$acf))
detected_lag <- ccf_result$lag[max_lag]
detected_lag
```

```
## [1] -10
```

```
start_time <- Sys.time()
ccf_boot_res <- ccf_boot(
  x = df$series1,
  y = df$series2,
  lag.max = 60,
  plot = "Spearman"
)
```




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

```
## Time difference of 3.221652 secs
```

```
max_lag_boot <- which.max(abs(ccf_boot_res$r_S))
detected_lag_boot <- ccf_boot_res$Lag[max_lag]
detected_lag_boot
```

```
## [1] -10
```

```
start_time <- Sys.time()
# Use laggedcor package to detect lag
laggedcor_result <- calculate_lagged_correlation(
  x = df$series1,
  y = df$series2,
  time1 = df$timestamp,
  time2 = df$timestamp,
  time_tol = 1,
  step = 1 / 60,
  min_matched_sample = 10,
  threads = 16
)
```

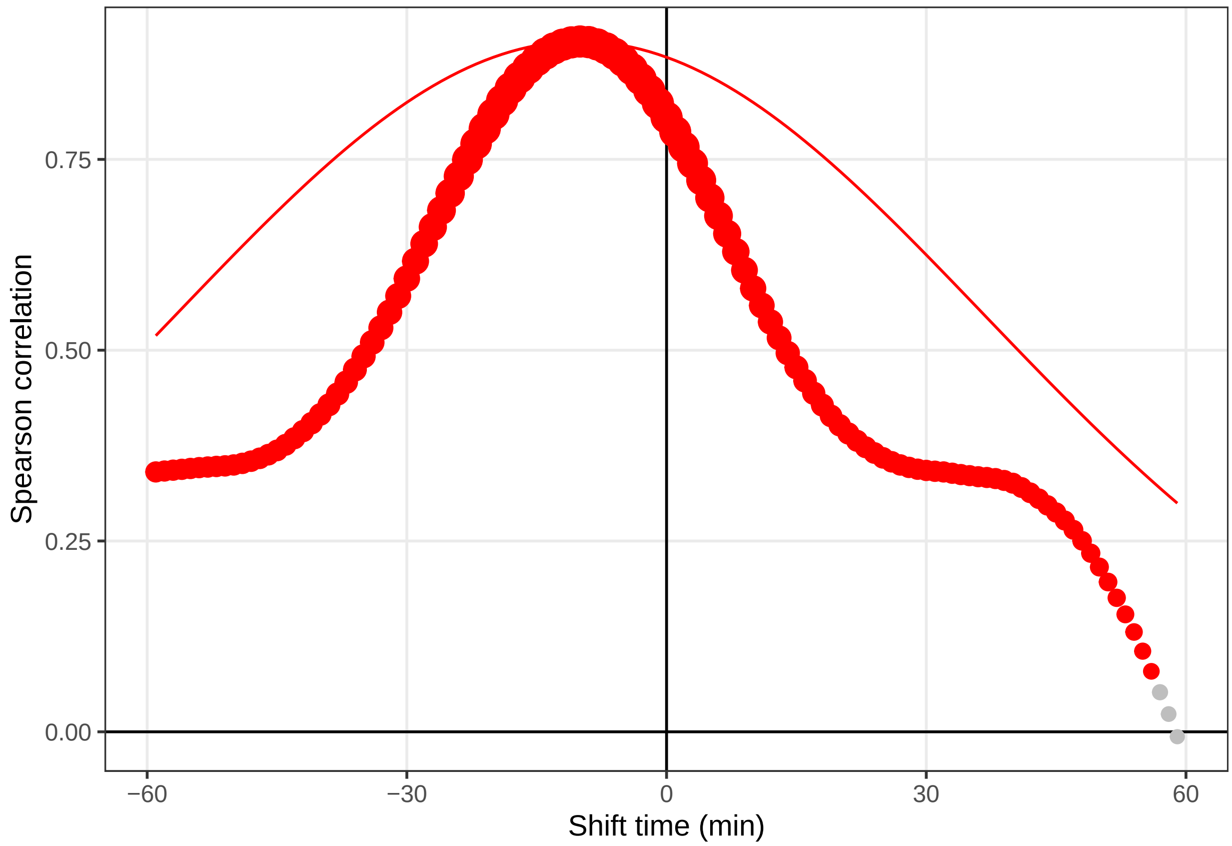
```
##   |                                                                              |
```

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

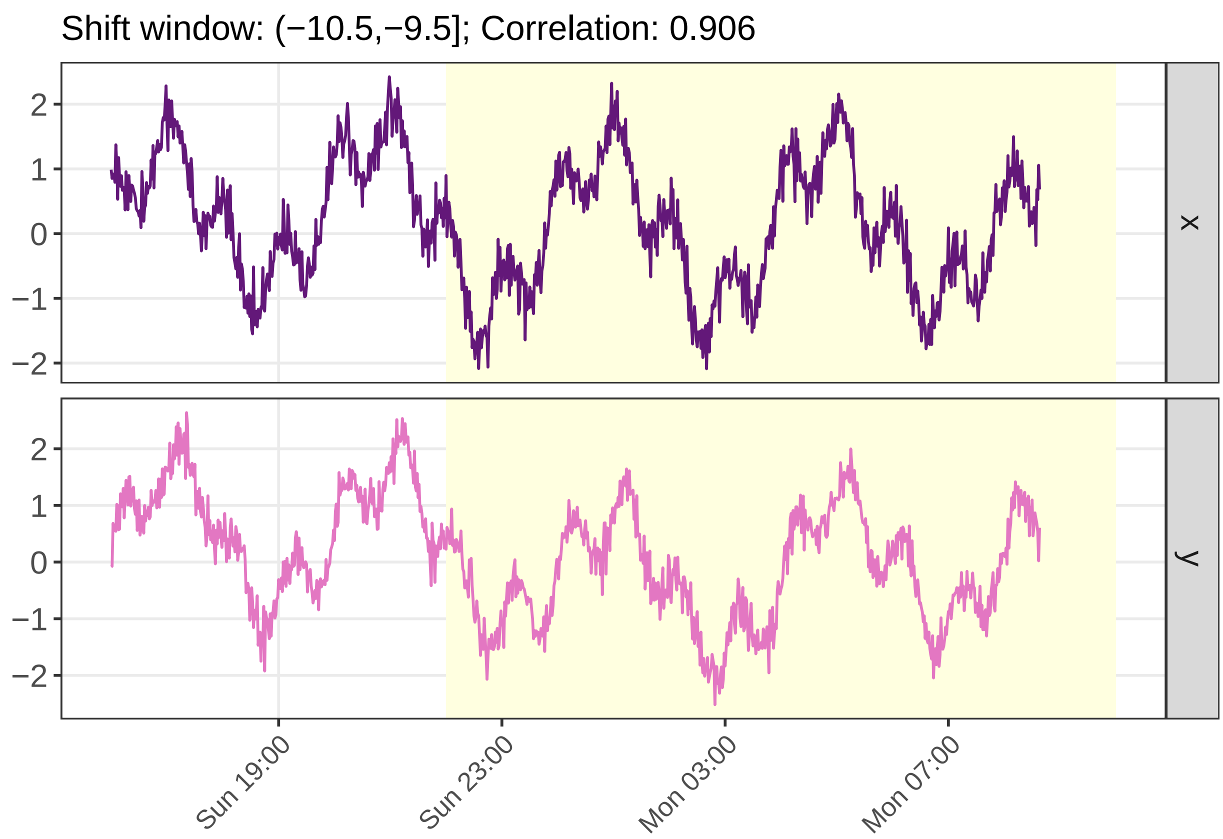
```
## Time difference of 37.90709 secs
```

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

```
## $score
## [1] 0.9972064
##
## $plot
```




```r
lagged_alignment_plot(
  laggedcor_result, which = "max"
)
```




```
shift_time <- extract_shift_time(laggedcor_result, numeric = TRUE)[which.max(extract_all_cor(laggedcor_r
```

- True lag: -10 minutes
- Detected lag via `ccf`: -10 minutes
  - Correlation (Pearson) at detected lag: 0.8995288.
- Detected lag via `ccf_boot`: -10 minutes
  - Correlation (Spearman) at detected lag: 0.9032516
- Detected lag via `laggedcor`: -10 minutes