State Space models for Pairs Trading

Import the library and set the path

```
# Set working directory to project root
library(here)
setwd(here::here())

# Core tidyverse and time series tools
library(dplyr)
library(tidyr)
library(purrr)
library(ggplot2)

# Time series and financial data
library(xts)
library(zoo)
library(TTR)
library(quantmod)

# Kalman filter and Partial CI
library(KFAS)
library(partialCI)

# I/O
library(readxl)
library(writexl)
```

Generate the dataset needed

```
source("src/stock_list.R")
source("src/generate_dataset.R")
begin_date <- as.Date("2010-01-01")
end_date <- as.Date("2024-05-01")
output_file <- "data/cleaned_etfs.csv"
```

```
generate_dataset(stock_namelist, begin_date, end_date, output_file)
```

Load the dataset created

```
output_file <- "data/cleaned_etfs.csv"
df <- read.csv(output_file)
df$Date <- as.Date(df[, 1])
data_xts <- xts(df[, -1], order.by = df$Date)
```

Define the estimation period you want to choose and the rolling window

Note that it takes 3-4 hours to run

```
source("src/func_partial_ci.R")
# Define the ticker you want to fit
stock_tickers <- colnames(data_xts)
```

```
# Creates the combination to estimate
stock_pairs <- combn(stock_tickers, 2, simplify = FALSE)

# Fitting parameters
estimation_years <- 3
rolling_step_months <- 6
save_dir <- "results/fit"
```

```
# Execute the function given defined parameters
run_partial_ci_backtest(stock_pairs, data_xts, estimation_years, rolling_step_months, save_dir)
```

Filter for the fitted parameters that you want to consider in the backtest

```
source("src/filtering_func.R")

results_folder <- "results/fit"
save_dir <- "results/pairs"

# Filter parameters
rho_min <- 0.9
rho_max <- 0.98
rsq_min <- 0.9
loglik_max <- 0

for (year in 2013:2024) {
  for (half in c("H1", "H2")) {
    process_period(year, half,
                   results_folder = results_folder,
                   rho_min = rho_min,
                   rho_max = rho_max,
                   rsq_min = rsq_min,
                   loglik_max = loglik_max,
                   save_dir = save_dir)
  }
}
```

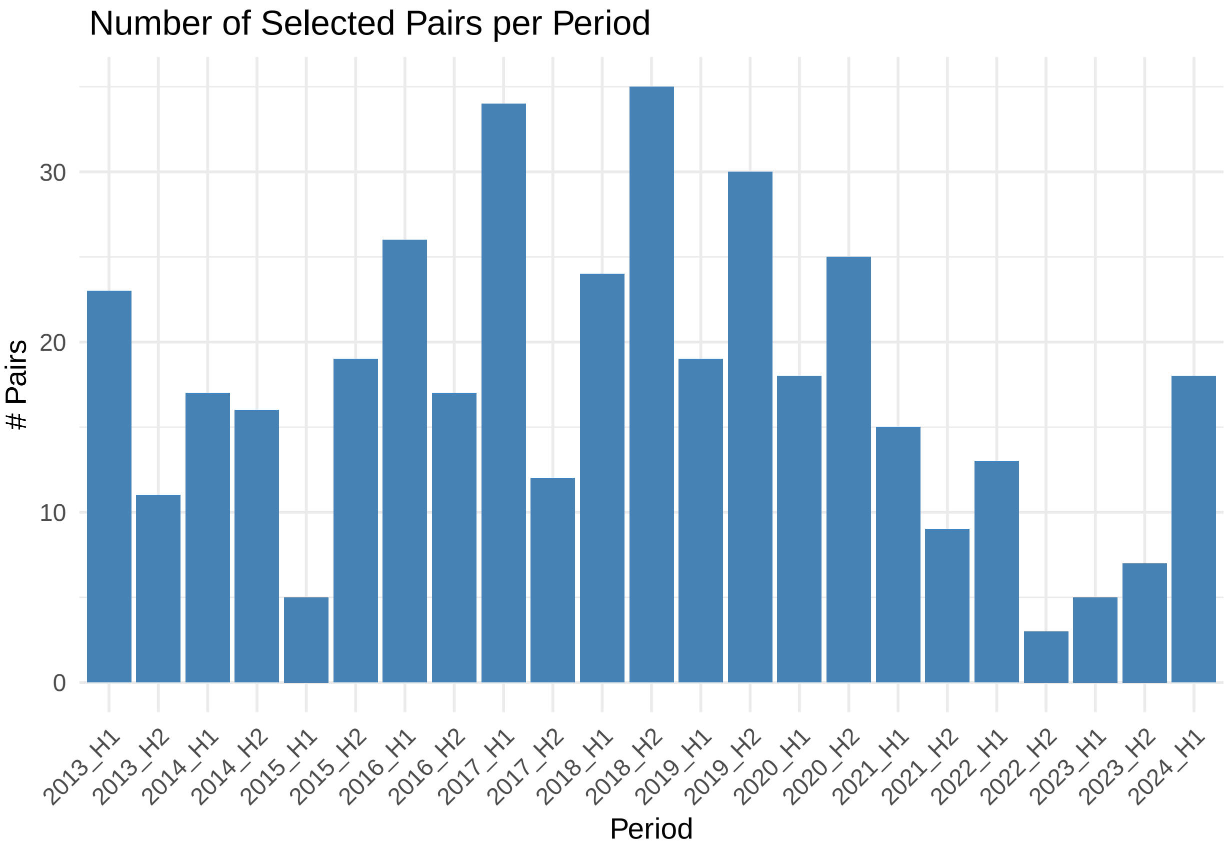
```
##  Saved filtered pairs for 2013_H1 to results/pairs/pairs_2013_H1.RData
##  Saved filtered pairs for 2013_H2 to results/pairs/pairs_2013_H2.RData
##  Saved filtered pairs for 2014_H1 to results/pairs/pairs_2014_H1.RData
##  Saved filtered pairs for 2014_H2 to results/pairs/pairs_2014_H2.RData
##  Saved filtered pairs for 2015_H1 to results/pairs/pairs_2015_H1.RData
##  Saved filtered pairs for 2015_H2 to results/pairs/pairs_2015_H2.RData
##  Saved filtered pairs for 2016_H1 to results/pairs/pairs_2016_H1.RData
##  Saved filtered pairs for 2016_H2 to results/pairs/pairs_2016_H2.RData
##  Saved filtered pairs for 2017_H1 to results/pairs/pairs_2017_H1.RData
##  Saved filtered pairs for 2017_H2 to results/pairs/pairs_2017_H2.RData
##  Saved filtered pairs for 2018_H1 to results/pairs/pairs_2018_H1.RData
##  Saved filtered pairs for 2018_H2 to results/pairs/pairs_2018_H2.RData
##  Saved filtered pairs for 2019_H1 to results/pairs/pairs_2019_H1.RData
##  Saved filtered pairs for 2019_H2 to results/pairs/pairs_2019_H2.RData
##  Saved filtered pairs for 2020_H1 to results/pairs/pairs_2020_H1.RData
##  Saved filtered pairs for 2020_H2 to results/pairs/pairs_2020_H2.RData
##  Saved filtered pairs for 2021_H1 to results/pairs/pairs_2021_H1.RData
##  Saved filtered pairs for 2021_H2 to results/pairs/pairs_2021_H2.RData
##  Saved filtered pairs for 2022_H1 to results/pairs/pairs_2022_H1.RData
##  Saved filtered pairs for 2022_H2 to results/pairs/pairs_2022_H2.RData
##  Saved filtered pairs for 2023_H1 to results/pairs/pairs_2023_H1.RData
##  Saved filtered pairs for 2023_H2 to results/pairs/pairs_2023_H2.RData
##  Saved filtered pairs for 2024_H1 to results/pairs/pairs_2024_H1.RData
##  File not found: results/fit/res_2024_H2.RData
```

Merge all the selected pairs in a dataframe

```r
pairs_dir <- "results/pairs/"

#  list all the pairs
pair_files <- list.files(pairs_dir, pattern = "^pairs_.*\\.RData$", full.names = TRUE)

all_pairs <- list()

for (file in pair_files) {
  temp_env <- new.env()
  load(file, envir = temp_env)
  var_name <- ls(temp_env)[grepl("^pairs_", ls(temp_env))]
  pairs <- get(var_name, envir = temp_env)
  all_pairs[[gsub("pairs_|\\.RData", "", basename(file))]] <- pairs
}

# Rbind all the pairs in a unique dataset
pairs_df <- do.call(rbind, lapply(names(all_pairs), function(period) {
  do.call(rbind, lapply(all_pairs[[period]], function(pair) {
    data.frame(period = period, stock_a = pair[1], stock_b = pair[2])
  }))
}))
```

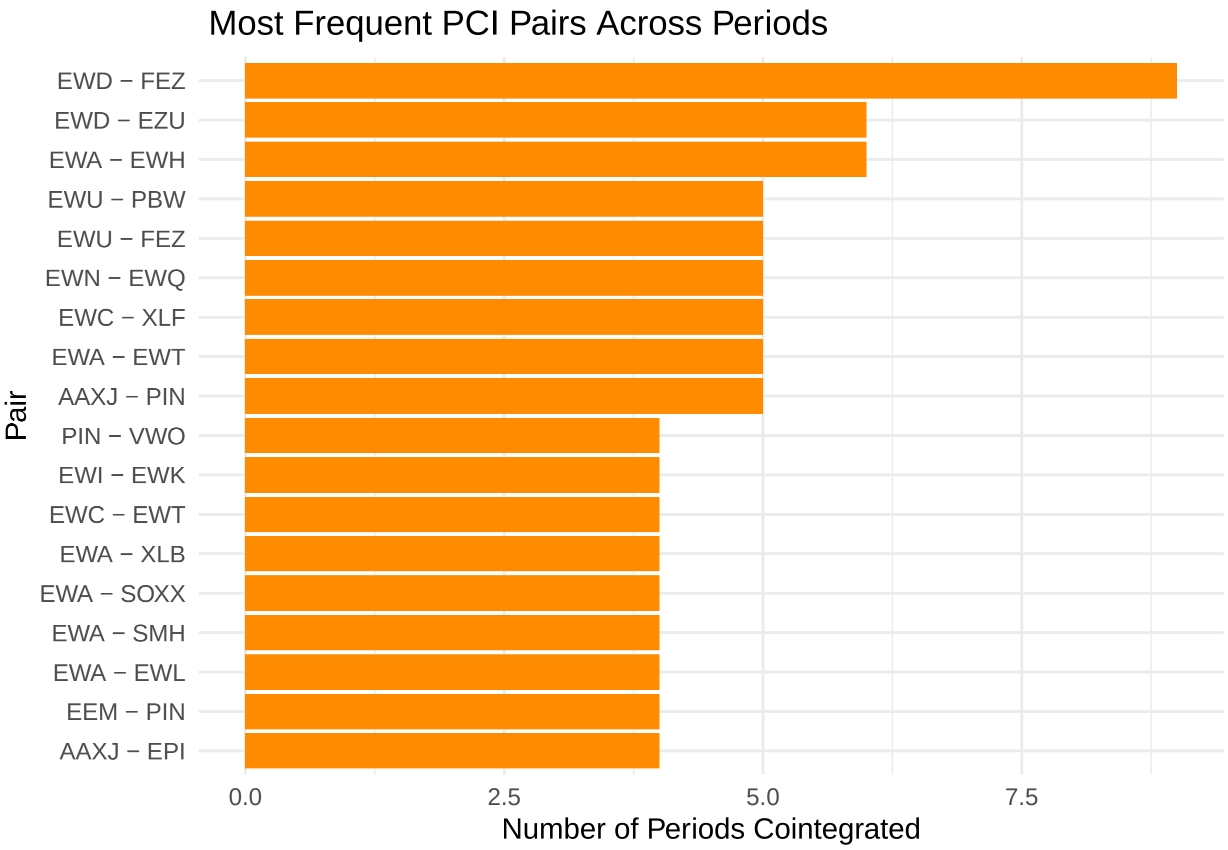
Statistics of the selected pairs

```r
# Count the number of selected pairs for each period
pair_counts <- pairs_df %>%
  group_by(period) %>%
  summarise(num_pairs = n())

ggplot(pair_counts, aes(x = period, y = num_pairs)) +
  geom_col(fill = "steelblue") +
  theme_minimal() +
  labs(title = " Number of Selected Pairs per Period",
       x = "Period",
       y = "# Pairs") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

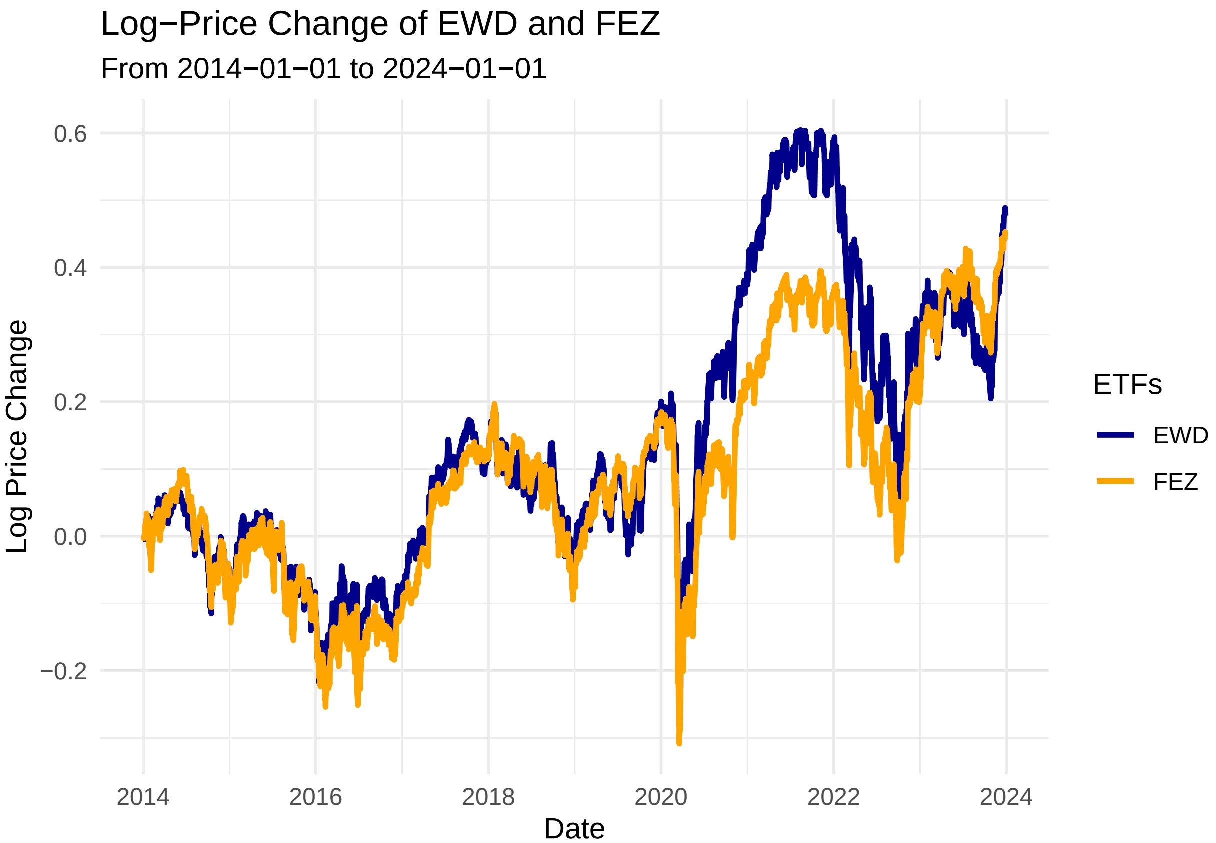


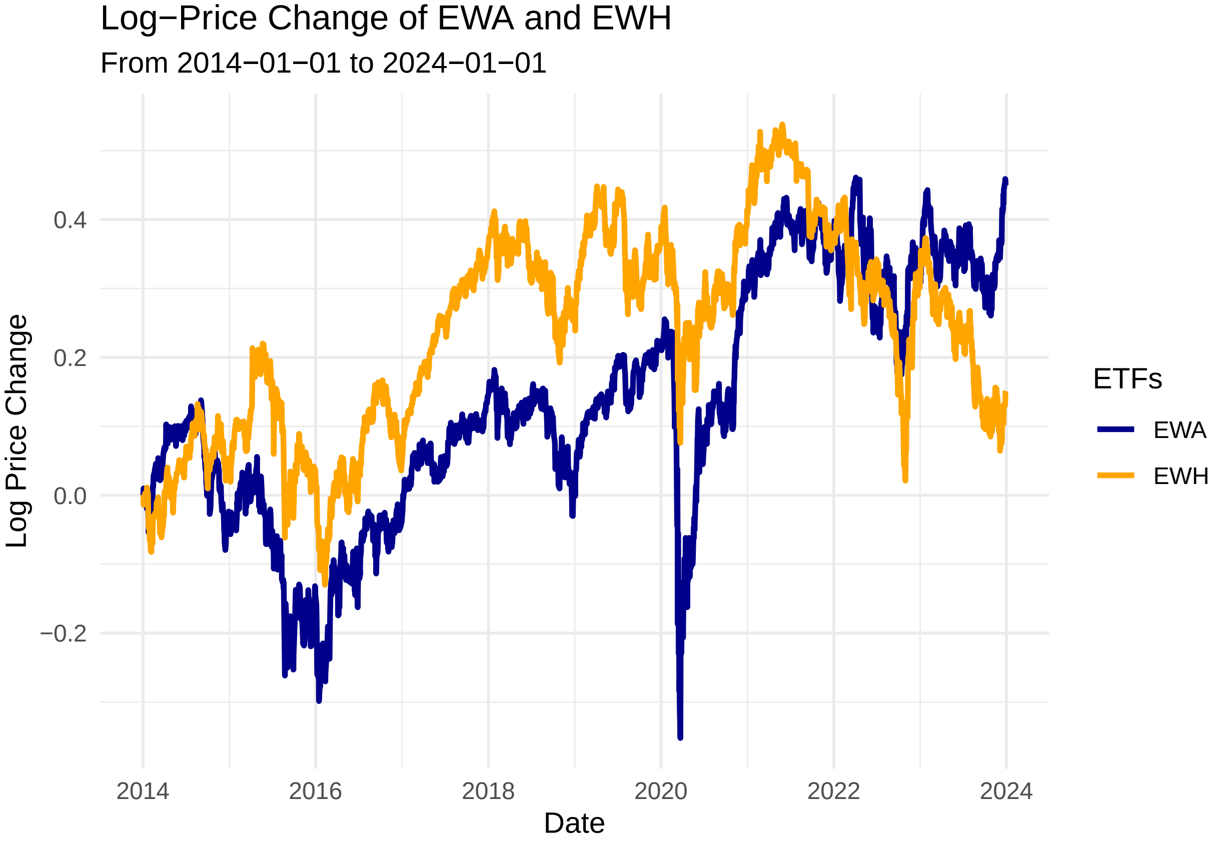
Statistics of the pairs

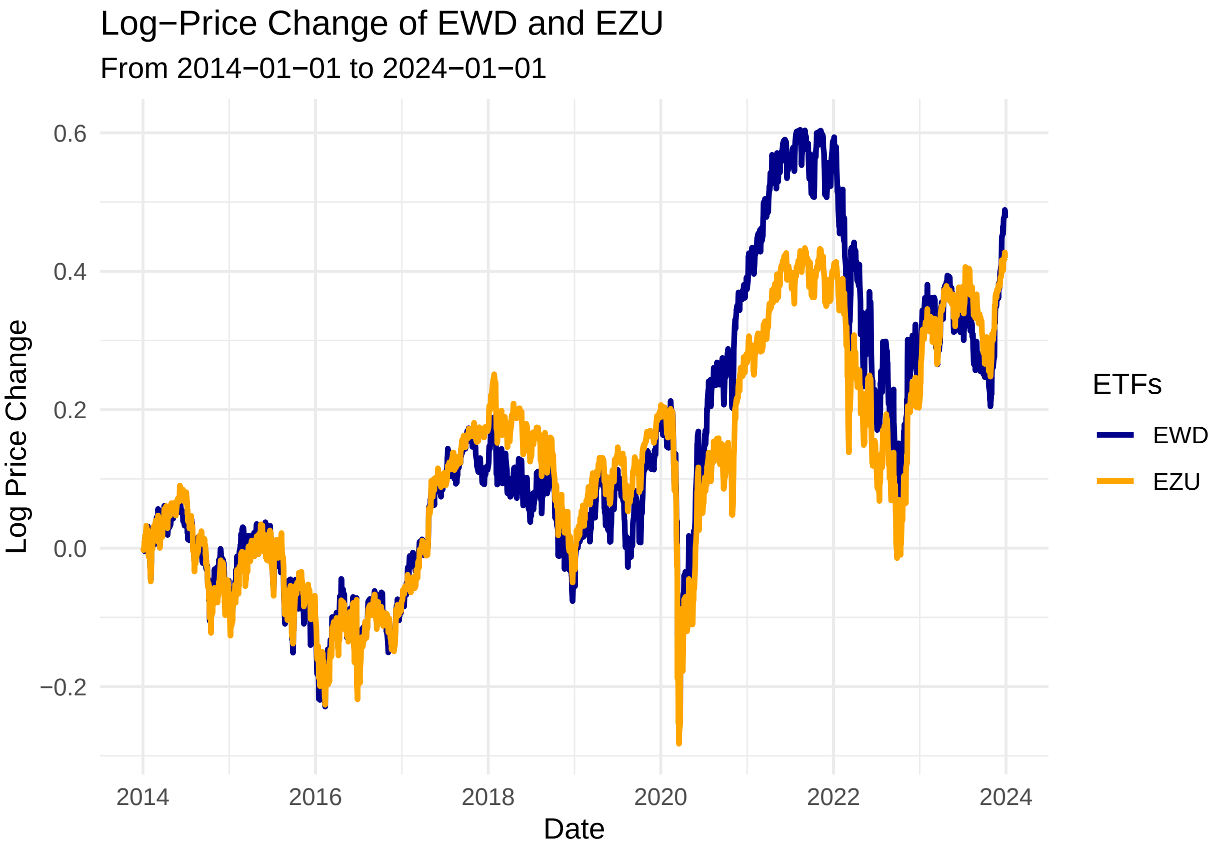
```
#  Uniqie pair name
pairs_df <- pairs_df %>%
  mutate(pair = paste(pmin(stock_a, stock_b), pmax(stock_a, stock_b), sep = " - "))

#  only the top 10
top_pairs <- pairs_df %>%
  count(pair, sort = TRUE) %>%
  top_n(10, n)

ggplot(top_pairs, aes(x = reorder(pair, n), y = n)) +
  geom_col(fill = "darkorange") +
  coord_flip() +
  theme_minimal() +
  labs(title = " Most Frequent PCI Pairs Across Periods",
       x = "Pair",
       y = "Number of Periods Cointegrated")
```



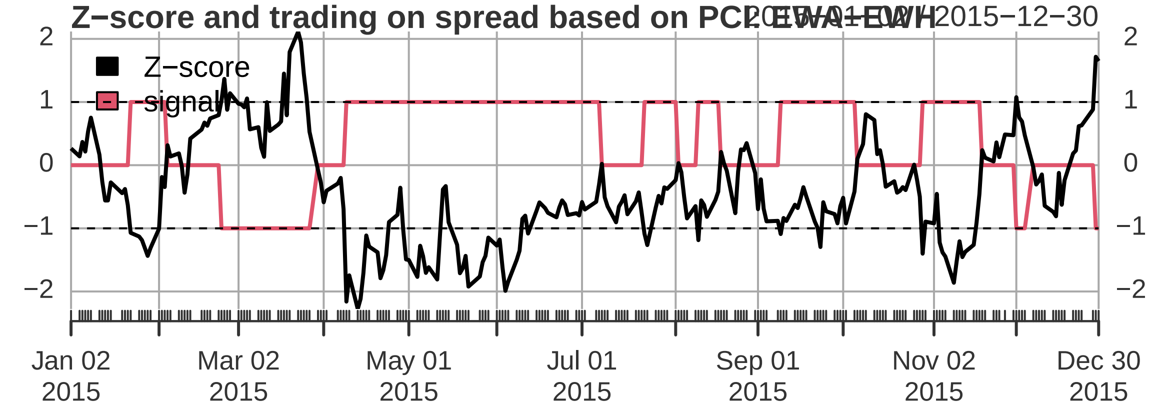
```r
source("src/plots_func.R")
pair_counts <- pairs_df %>%
  group_by(period) %>%
  summarise(num_pairs = n())
#  Uniqie pair name
pairs_df <- pairs_df %>%
  mutate(pair = paste(pmin(stock_a, stock_b), pmax(stock_a, stock_b), sep = " - "))

#  plot only the top 5
top_pairs <- pairs_df %>%
  count(pair, sort = TRUE) %>%
  top_n(3, n)

walk(top_pairs$pair, function(p) {
  tickers <- unlist(strsplit(p, " - "))
  print(plot_pair_log_price_change(data_xts, tickers[1], tickers[2],
                                   start_date = "2014-01-01", end_date = "2024-01-01"))
})
```

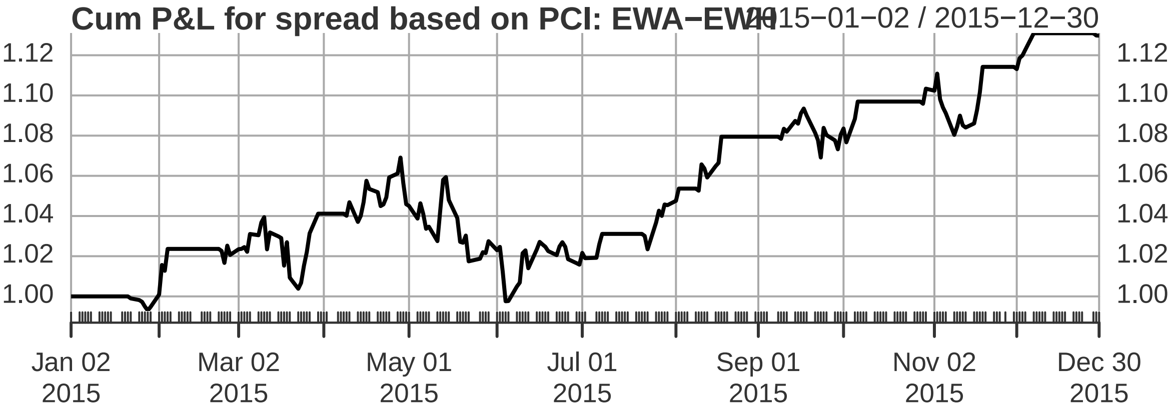

Log−Price Change of EWD and FEZ
From 2014−01−01 to 2024−01−01
Log Price Change
0.6
0.4
0.2
0.0
−0.2
2014
2016
2018
2020
2022
2024
Date
ETFs
EWD
FEZ


Log−Price Change of EWA and EWH
From 2014−01−01 to 2024−01−01
Log Price Change
0.4
0.2
0.0
−0.2
2014
2016
2018
2020
2022
2024
Date
ETFs
EWA
EWH



```r
source("/Users/carlocascini/Desktop/pairs-trading/src/Backtest_func.R")

result <- run_pairs_trading_strategy(
  Y = data_xts,
  tickers = c("EWA", "EWH"),
  test_start = "2015-01-01",
  test_end = "2015-12-30",
  training_years = estimation_years,
  transaction_cost = 0.001,
  threshold = 1,
  risk_free_rate = 0.02,
  plot = TRUE
)
```





```
# View performance
print(result$performance)
```

```
##   Strategy Train.Start  Train.End Test.Start   Test.End Total.Return....
## 1      PCI  2011-12-31 2014-12-31 2015-01-01 2015-12-30            12.98
## 2      KFB  2011-12-31 2014-12-31 2015-01-01 2015-12-30            12.40
##   Annualized.Return.... Sharpe.Ratio Annualized.SD....
## 1                 13.03         1.23              8.35
## 2                 12.45         1.16              8.39
```