Estimating time-varying networks from cryptocurrency dataset

For this notebook, we use packages

```
library(quantmod)
library(cmdstanr)
library(igraph)
library(posterior)
```

The functions used for edge selection and performance evaluation can be found in *btvn_functions.R*.

```
source("btvn_functions.R")
```

Loading the data and creating two datasets: Normalization by month and Normalization for the whole period

```
crypto = c("BTC",
"WBTC",
"WSTETH",
"ETH",
"STETH",
"WETH",
"BNB",
"BCH",
"SOL",
"LTC",
"AVAX",
"LINK",
"LEO",
"TON11419",
"NEAR",
"DOT",
"USDC",
"DAI",
"USDT",
"XRP")

crypto = paste0(crypto, "-USD")

Data2 = c()

for(cr in crypto){

  A = list(Symbols = cr,
           auto.assign = FALSE,
           from = "2022-01-01",
           to = "2024-06-30")

  a = do.call("getSymbols", A)

  Data2 = cbind(Data2, a[, 4]) # Closing prices
}

#monthly
crypto_data_list2 <- create_list_by_month2(Data2,c(2022,2023,2024),
                 list(c("01","02","03","04","05","06","07","08","09",10,11,12),
                      c("01","02","03","04","05","06","07","08","09",10,11,12),
                      c("01","02","03","04","05","06")), scale = T)
#whole period
crypto_data_list3 <- create_list_by_month2(scale(Data2),c(2022,2023,2024),
                 list(c("01","02","03","04","05","06","07","08","09",10,11,12),
                      c("01","02","03","04","05","06","07","08","09",10,11,12),
                      c("01","02","03","04","05","06")), scale = F)

otokset2 <- c()
for (i in 1:30) {
  otokset2[i] <- dim(crypto_data_list2[[i]])[[1]]
}
```

```
data_crypto_n2.2.2 <- list( T=30, N=otokset2, M=20,
                          elet1=as.matrix(crypto_data_list2[[1]]),
                          elet2=as.matrix(crypto_data_list2[[2]]),
                          elet3=as.matrix(crypto_data_list2[[3]]),
                          elet4=as.matrix(crypto_data_list2[[4]]),
                          elet5=as.matrix(crypto_data_list2[[5]]),
                          elet6=as.matrix(crypto_data_list2[[6]]),
                          elet7=as.matrix(crypto_data_list2[[7]]),
                          elet8=as.matrix(crypto_data_list2[[8]]),
                          elet9=as.matrix(crypto_data_list2[[9]]),
                          elet10=as.matrix(crypto_data_list2[[10]]),
                          elet11=as.matrix(crypto_data_list2[[11]]),
                          elet12=as.matrix(crypto_data_list2[[12]]),
                          elet13=as.matrix(crypto_data_list2[[13]]),
                          elet14=as.matrix(crypto_data_list2[[14]]),
                          elet15=as.matrix(crypto_data_list2[[15]]),
                          elet16=as.matrix(crypto_data_list2[[16]]),
                          elet17=as.matrix(crypto_data_list2[[17]]),
                          elet18=as.matrix(crypto_data_list2[[18]]),
                          elet19=as.matrix(crypto_data_list2[[19]]),
                          elet20=as.matrix(crypto_data_list2[[20]]),
                          elet21=as.matrix(crypto_data_list2[[21]]),
                          elet22=as.matrix(crypto_data_list2[[22]]),
                          elet23=as.matrix(crypto_data_list2[[23]]),
                          elet24=as.matrix(crypto_data_list2[[24]]),
                          elet25=as.matrix(crypto_data_list2[[25]]),
                          elet26=as.matrix(crypto_data_list2[[26]]),
                          elet27=as.matrix(crypto_data_list2[[27]]),
                          elet28=as.matrix(crypto_data_list2[[28]]),
                          elet29=as.matrix(crypto_data_list2[[29]]),
                          elet30=as.matrix(crypto_data_list2[[30]]),
                          theta=0.011)

data_crypto_n2.2 <- list( T=30, N=otokset2, M=20,
                        elet1=as.matrix(crypto_data_list3[[1]]),
                        elet2=as.matrix(crypto_data_list3[[2]]),
                        elet3=as.matrix(crypto_data_list3[[3]]),
                        elet4=as.matrix(crypto_data_list3[[4]]),
                        elet5=as.matrix(crypto_data_list3[[5]]),
                        elet6=as.matrix(crypto_data_list3[[6]]),
                        elet7=as.matrix(crypto_data_list3[[7]]),
                        elet8=as.matrix(crypto_data_list3[[8]]),
                        elet9=as.matrix(crypto_data_list3[[9]]),
                        elet10=as.matrix(crypto_data_list3[[10]]),
                        elet11=as.matrix(crypto_data_list3[[11]]),
                        elet12=as.matrix(crypto_data_list3[[12]]),
                        elet13=as.matrix(crypto_data_list3[[13]]),
                        elet14=as.matrix(crypto_data_list3[[14]]),
                        elet15=as.matrix(crypto_data_list3[[15]]),
                        elet16=as.matrix(crypto_data_list3[[16]]),
                        elet17=as.matrix(crypto_data_list3[[17]]),
                        elet18=as.matrix(crypto_data_list3[[18]]),
                        elet19=as.matrix(crypto_data_list3[[19]]),
                        elet20=as.matrix(crypto_data_list3[[20]]),
                        elet21=as.matrix(crypto_data_list3[[21]]),
                        elet22=as.matrix(crypto_data_list3[[22]]),
                        elet23=as.matrix(crypto_data_list3[[23]]),
                        elet24=as.matrix(crypto_data_list3[[24]]),
                        elet25=as.matrix(crypto_data_list3[[25]]),
                        elet26=as.matrix(crypto_data_list3[[26]]),
                        elet27=as.matrix(crypto_data_list3[[27]]),
                        elet28=as.matrix(crypto_data_list3[[28]]),
                        elet29=as.matrix(crypto_data_list3[[29]]),
                        elet30=as.matrix(crypto_data_list3[[30]]),
                        theta=0.011)
```

Model compilation

```
crypto_main_m <- "btvn_b_crypto.stan"
mod_crypto_main_m <- cmdstan_model(crypto_main_m,
                                   cpp_options = list(stan_threads = TRUE),
                                   force=TRUE, stanc_options = list("O1"))
```

Estimating model (normalization by whole period)

```
fit_crypto_main_m2.2 <- mod_crypto_main_m$sample(
  data = data_crypto_n2.2,
  seed = 12345,
  chains = 1,
  parallel_chains = 1,
  threads_per_chain = 4,
  iter_warmup = 500,
  iter_sampling = 1500,
  fixed_param = FALSE,
  adapt_delta = 0.99,
  max_treedepth = 10
)
```

Estimation took 1883.3 seconds.

Saving the draws and constructing the network

```r
draws_crypto_main2.2.2 <- as_draws_rvars(fit_crypto_main2.2$draws())
crypto_verkko_main_m2.2 <- rho_res_mat_to_adj_list(
                           rho_res_mat = fit_crypto_main_m2.2, M = 20,
                           n_time_points = 30,credible_level = 0.99)
```

Plotting individual graphs (time-varying hub structure)

```r
par(mfrow=c(5,6))
for (i in 1:30) {
  graph_plot(crypto_verkko_main_m2.2[[i]])
}
```

Constructing the weighted graph

```r
crypto_m_cum2.2 <- crypto_verkko_main_m2.2[[1]]
for (i in 2:30) {
  crypto_m_cum2.2  <- crypto_m_cum2.2 + crypto_verkko_main_m2.2[[i]]
}

crypto_m_cum_g2.2 <- crypto_m_cum2.2
crypto_m_cum_g2.2[crypto_m_cum_g2.2<2] <- 0
crypto_m_cum_g2.2[crypto_m_cum_g2.2!=0] <- 1
```

The same as above but for the monthly normalized data

```r
fit_crypto_main_m2.2.2 <- mod_crypto_main_m$sample(
  data = data_crypto_n2.2.2,
  seed = 12345,
  chains = 1,
  parallel_chains = 1,
  threads_per_chain = 4,
  iter_warmup = 500,
  iter_sampling = 1500,
  fixed_param = FALSE,
  adapt_delta = 0.99,
  max_treedepth = 10
)
```

Estimation took 3435.8 seconds.

The same post-processing as before

```r
draws_crypto_main2.2.2 <- as_draws_rvars(fit_crypto_main2.2.2$draws())

crypto_verkko_main_m2.2.2 <- rho_res_mat_to_adj_list(
                             rho_res_mat = fit_crypto_main_m2.2.2, M = 20,
                             n_time_points = 30,credible_level = 0.99)
```

Visualization and community detection:

The nodes are named. Coordinates for plotting are taken from the monthly estimate.

```r
network2 <- graph_from_adjacency_matrix(crypto_verkko_main_m2.2.2[[1]],
                                          mode = "undirected",diag = F)
vertex_attr(network2,"name") <- c("BTC",
                                    "WBTC",
                                    "WSTETH",
                                    "ETH",
                                    "STETH",
                                    "WETH",
                                    "BNB",
                                    "BCH",
                                    "SOL",
                                    "LTC",
                                    "AVAX",
                                    "LINK",
                                    "LEO",
                                    "TON11419",
                                    "NEAR",
                                    "DOT",
                                    "USDC",
                                    "DAI",
                                    "USDT",
                                    "XRP")
set.seed(12345)
coords2.2<-layout.fruchterman.reingold(network2)
```

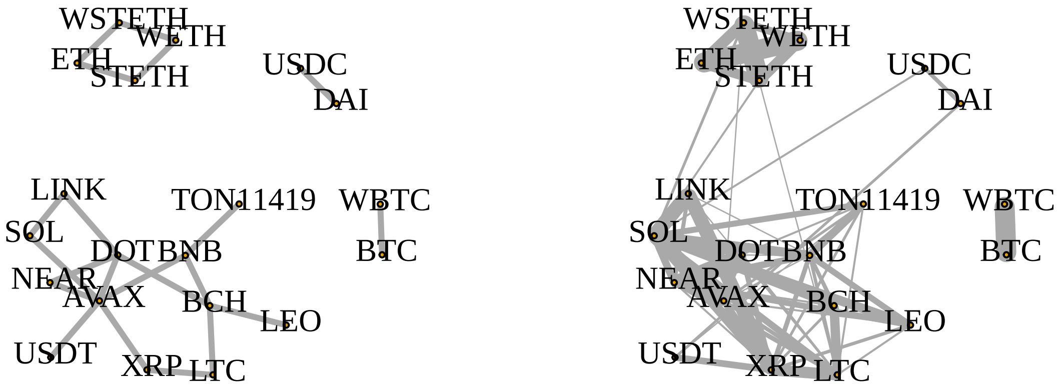
```
network2.2 <- graph_from_adjacency_matrix(crypto_m_cum_g2.2/3,
                 mode = "undirected",diag = F,weighted = TRUE)
vertex_attr(network2.2,"name") <- c("BTC",
                                 "WBTC",
                                 "WSTETH",
                                 "ETH",
                                 "STETH",
                                 "WETH",
                                 "BNB",
                                 "BCH",
                                 "SOL",
                                 "LTC",
                                 "AVAX",
                                 "LINK",
                                 "LEO",
                                 "TON11419",
                                 "NEAR",
                                 "DOT",
                                 "USDC",
                                 "DAI",
                                 "USDT",
                                 "XRP")
```

The resulting graphs can the be plotted (Figure 4 of the manuscript)

```
par(mfrow=c(1,2))

plot(network2,layout=coords2.2,main="", vertex.label.color="black",
     edge.width=3,vertex.label.dist=0.5,vertex.size=3,)

plot(network2.2,layout=coords2.2,main="", vertex.label.color="black",
     edge.width=E(network2.2)$weight,vertex.label.dist=0.5,vertex.size=3,)
```



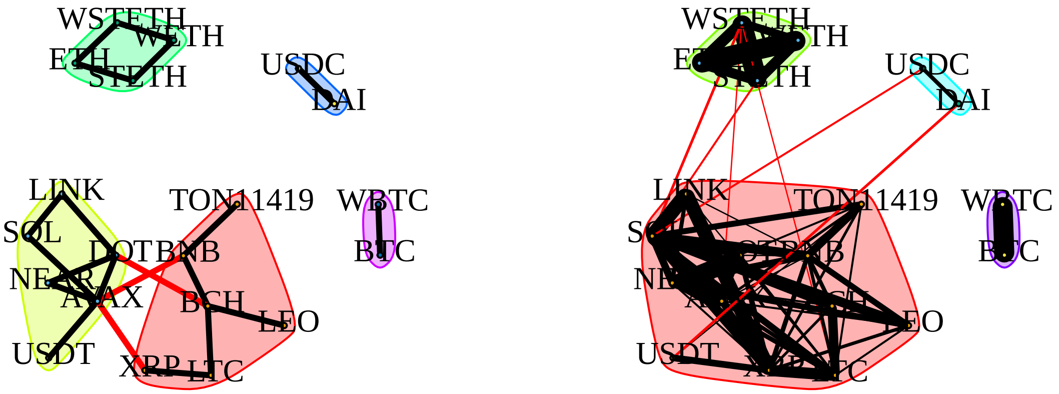
Then, communities are identified using walktrap method

```
coords2.2c <- igraph::cluster_walktrap(network2)
coords2.2.2c <- igraph::cluster_walktrap(network2.2)
```

and can be plotted (Figure 5 of the manuscript)

```
par(mfrow=c(1,2))
plot(coords2.2c,network2,layout=coords2.2,main="", vertex.label.color="black",
     edge.width=3,vertex.label.dist=0.5,vertex.size=3)

plot(coords2.2.2c,network2.2,layout=coords2.2,main="",
     vertex.label.color="black", edge.width=E(network2.2)$weight,
     vertex.label.dist=0.5,vertex.size=3)
```



The found communities can then be compared

```
igraph::compare(igraph::cluster_walktrap(network2),
                igraph::cluster_walktrap(network2.2),
                method = "rand")
```

```
## [1] 0.8105263
```