```
## Step 1: Estimating minimum spanning trees (MSTs).

# Load igraph package.
library(igraph)

# Load a data file with 73 variables where the first column contains dates while the rest of the columns contain FX spot prices.
load("Data.RData")

# Create a data variable excluding the date column.
Data <- Data[-c(1)]

# Convert spot prices into log prices as a matrix.
Log_data <- as.matrix(log(Data))

# Calculate price changes.
Rets <- diff(Log_data)

# Estimate correlation coefficients from log returns.
Corr_mat <- cor(Rets)

# Label rows and columns.
colnames(Corr_mat) <-
c("EURGBP","EURAUD","EURNZD","EURUSD","EURCAD","EURCHF","EURJPY","EURCNH",
"GBPEUR","GBPAUD","GBPNZD","GBPUSD","GBPCAD","GBPCHF","GBPJPY","GBPCNH",
"AUDEUR","AUDGBP","AUDNZD","AUDUSD","AUDCAD","AUDCHF","AUDJPY","AUDCNH",
"NZDEUR","NZDGBP","NZDAUD","NZDUSD","NZDCAD","NZDCHF","NZDJPY","NZDCNH",
"USDEUR","USDGBP","USDAUD","USDNZD","USDCAD","USDCHF","USDJPY","USDCNH",
"CADEUR","CADGBP","CADAUD","CADNZD","CADUSD","CADCHF","CADJPY","CADCNH",
"CHFEUR","CHFGBP","CHFAUD","CHFNZD","CHFUSD","CHFCAD","CHFJPY","CHFCNH",
"JPYEUR","JPYGBP","JPYAUD","JPYNZD","JPYUSD","JPYCAD","JPYCHF","JPYCNH",
"CNHEUR","CNHGBP","CNHAUD","CNHNZD","CNHUSD","CNHCAD","CNHCHF","CNHJPY")

rownames(Corr_mat) <-
c("EURGBP","EURAUD","EURNZD","EURUSD","EURCAD","EURCHF","EURJPY","EURCNH",
"GBPEUR","GBPAUD","GBPNZD","GBPUSD","GBPCAD","GBPCHF","GBPJPY","GBPCNH",
"AUDEUR","AUDGBP","AUDNZD","AUDUSD","AUDCAD","AUDCHF","AUDJPY","AUDCNH",
"NZDEUR","NZDGBP","NZDAUD","NZDUSD","NZDCAD","NZDCHF","NZDJPY","NZDCNH",
"USDEUR","USDGBP","USDAUD","USDNZD","USDCAD","USDCHF","USDJPY","USDCNH",
"CADEUR","CADGBP","CADAUD","CADNZD","CADUSD","CADCHF","CADJPY","CADCNH",
"CHFEUR","CHFGBP","CHFAUD","CHFNZD","CHFUSD","CHFCAD","CHFJPY","CHFCNH",
"JPYEUR","JPYGBP","JPYAUD","JPYNZD","JPYUSD","JPYCAD","JPYCHF","JPYCNH",
"CNHEUR","CNHGBP","CNHAUD","CNHNZD","CNHUSD","CNHCAD","CNHCHF","CNHJPY")

# Convert correlation matrix into Euclidean distance matrix.
Dis_mat <- as.matrix(sqrt(2-2*Corr_mat))

# Convert the distance matrix as an adjacency matrix.
G <- graph.adjacency(Dis_mat,mode=c("undirected"),weighted=TRUE)

# Estimate minimum spanning tree (MST) using Prim's algorithm.
MST <- minimum.spanning.tree(G,algorithm="prim")

## Step 2: Visualising results.

# Set layout algorithm.
Lay <- layout.kamada.kawai(MST)
```

```
# Set node colours based on base currency.
# EUR nodes.
V(MST)$color[1:8] <- rgb(0,128,0,255,max=255)
# GBP nodes.
V(MST)$color[9:16] <- rgb(15,0,128,255,max=255)
# AUD, NZD nodes.
V(MST)$color[17:32] <- rgb(192,192,192,255,max=255)
#USD nodes.
V(MST)$color[33:40] <- rgb(255,1,255,255,max=255)
# CAD nodes.
V(MST)$color[41:48] <- rgb(192,192,192,255,max=255)
#CHF, JPY nodes.
V(MST)$color[49:64] <- rgb(255,255,0,255,max=255)
#CNH nodes.
V(MST)$color[65:72] <- rgb(255,1,255,255,max=255)

# Plot MST.
plot(MST,layout=lay,vertex.size=4,vertex.shape="circle", asp=FALSE,
vertex.label.cex=0.3, vertex.label.dist=0.25, vertex.label.degree=-pi/2,
vertex.label.color="black",vertex.color=V(MST)$color,
edge.width=0.5,edge.color=rgb(0,0,0,255,max=255))
```