D’Amelio-Stagnitto

DS

14/1/2018

In this project we analyzed financial data taken from Yahoo for studying the dependency among companies stock we’ve considered, using two kind of correlation methods:

- Pearson
- Kendall

In the following, you’ll find the code and all the comments about how we proceeded.

```
require(tseries, quietly = TRUE)
library(reshape2)
library(igraph)
library('visNetwork')
library('doSNOW')
library('doParallel')
library('foreach')
```

```
options("getSymbols.warning4.0" =FALSE)
options("getSymbols.yahoo.warning" = FALSE)

# The first part of this function takes data from Yahoo.
# In the second part we transform the data in a DataFrame first and in a Time Seri
es then. We change the
# structure of the dataset transforming the Date (which was the index) in a variab
le.

data <- function(x) {
  z <- suppressWarnings(get.hist.quote(instrument= x, start="2003-01-01", end="200
8-01-01",
                 quote = "Close", provider="yahoo", drop=TRUE, compression = "d"))

  df <- data.frame(z)
  df$Date <- time(z)
  rownames(df) <- NULL
  df <- df[,c(ncol(df),1:(ncol(df)-1))]
  colnames(df) <- c("Date","Close")
  return (df)

}

# These are the groups we chose

# Consumer Discretionary
Nike <- data("NKE")
```

```
Hasbro <- data("HAS")
Walt_Disney <- data("DIS")
McDonald <- data("MCD")
Tiffany <- data("TIF")

# Energy
Marathon <- data("MRO")
Apache <- data("APA")
Schlumb <- data("SLB")
Williams <- data("WMB")
Occid_Petr <- data("OXY")

# Financials
Goldman <- data("GS")
American_exp <- data("AXP")
American_bank <- data("BAC")
Morgan <- data("MS")
Metlife <- data("MET")

# Healt Care
Zimmer <- data("ZBH")
Stryker <- data("SYK")
Metler <- data("MDT")
JJ <- data("JNJ")
Humana <- data("HUM")

# Industrials
Robert <- data("RHI")
Grumann <- data("NOC")
Textron <- data("TXT")
Boeing <- data("BA")
Etn <- data("ETN")

# I-T
EA <- data("EA")
Adobe <- data("ADBE")
Microsoft <- data("MSFT")
Oracle <- data("ORCL")
Intel <- data("INTC")

# Materials
Mon <- data("MON")
Eastman <- data("EMN")
Air <- data("APD")
Sealed <- data("SEE")
Sherwin <- data("SHW")

# Utilities
Ameren <- data("AEE")
Duke <- data("DUK")
```

```
CenterPoint <- data("CNP")
Next <- data("NEE")
AES <- data("AES")

# Telecomunication
AT <- data("T")
Verizon <- data("VZ")
Century <- data("CTL")

# Consumer
Colgate <- data("CL")
Coca_cola <- data("KO")
Kellogs <- data("K")
Costco <- data("COST")
Pepsi <- data("PEP")


# This function computes the log to all the variables of our dataset
loga <- function(x){
  vec <- c()
  for (i in 1:nrow(x)){
    a <- log(x$Close[i]/x$Close[i-1])
    vec <- c(vec,a)
  }
  return(vec)
}

log_Nike <- loga(Nike)
log_Hasbro <- loga(Hasbro)
log_Walt_Disney <- loga(Walt_Disney)
log_McDonald <- loga(McDonald)
log_Tiffany <- loga(Tiffany)

log_Marathon <- loga(Marathon)
log_Apache <- loga(Apache)
log_Schlumb <- loga(Schlumb)
log_Williams <- loga(Williams)
log_Occid_Petr <- loga(Occid_Petr)

log_Goldman <- loga(Goldman)
log_American_exp <- loga(American_exp)
log_American_bank <- loga(American_bank)
log_Morgan <- loga(Morgan)
log_Metlife <- loga(Metlife)

log_Zimmer <- loga(Zimmer)
log_Stryker <- loga(Stryker)
log_Metler <- loga(Metler)
log_JJ <- loga(JJ)
log_Humana <- loga(Humana)
```

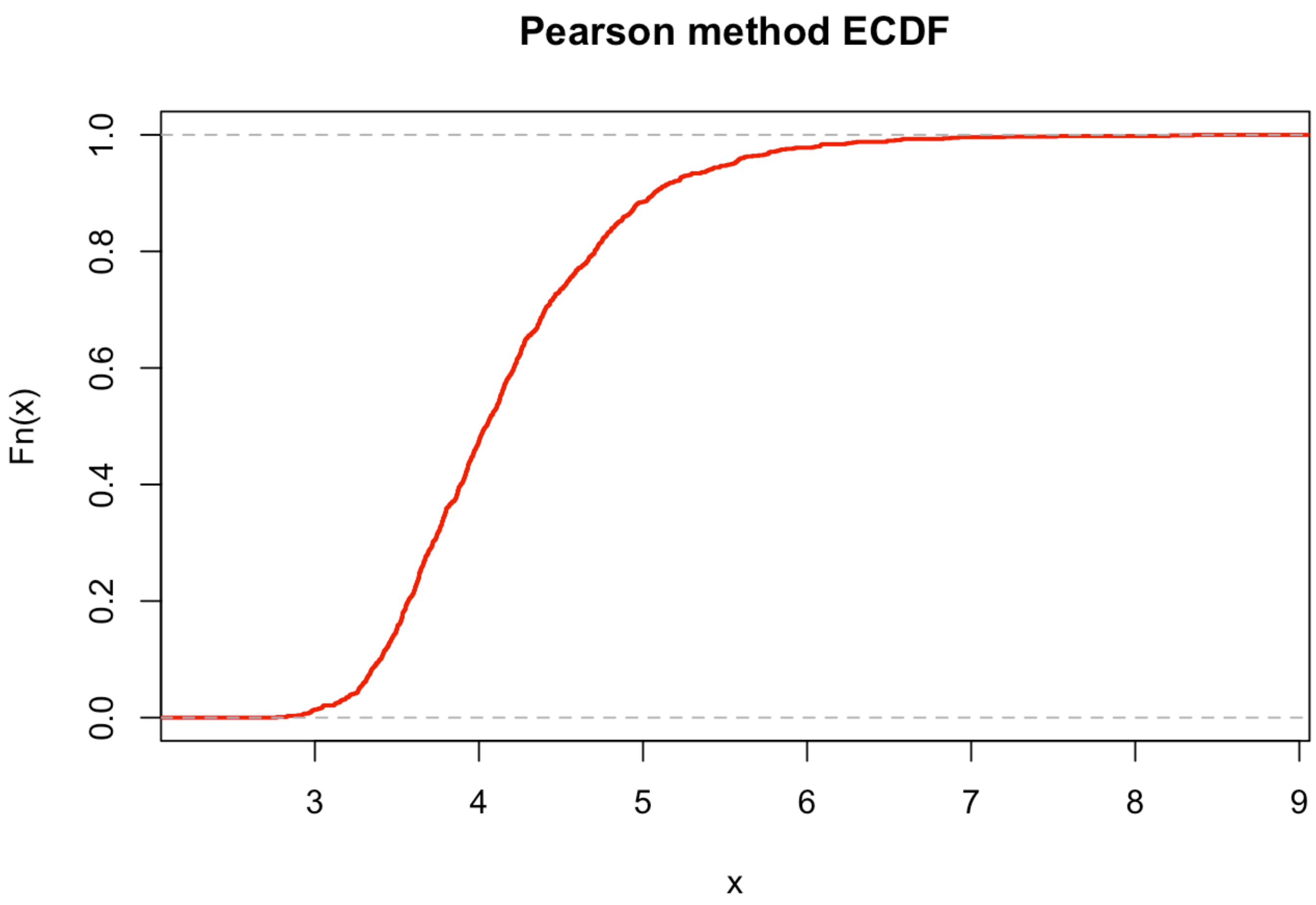
```
log_Robert <- loga(Robert)
log_Grumann <- loga(Grumann)
log_Textron <- loga(Textron)
log_Boeing <- loga(Boeing)
log_Etn <- loga(Etn)

log_EA <- loga(EA)
log_Adobe <- loga(Adobe)
log_Microsoft <- loga(Microsoft)
log_Oracle <- loga(Oracle)
log_Intel <- loga(Intel)

log_Mon <- loga(Mon)
log_Eastman <- loga(Eastman)
log_Air <- loga(Air)
log_Sealed <- loga(Sealed)
log_Sherwin <- loga(Sherwin)

log_Ameren <- loga(Ameren)
log_Duke <- loga(Duke)
log_CenterPoint <- loga(CenterPoint)
log_Next <- loga(Next)
log_AES <- loga(AES)

log_AT <- loga(AT)
log_Verizon <- loga(Verizon)
log_Century <- loga(Century)

log_Colgate <- loga(Colgate)
log_Coca_cola <- loga(Coca_cola)
log_Kellogs <- loga(Kellogs)
log_Costco <- loga(Costco)
log_Pepsi <- loga(Pepsi)



# This is our final Data Frame

X <- data.frame(Pepsi$Date[-1],
              log_Nike,log_Hasbro,log_Walt_Disney,          log_McDonald,log_Tiffa
ny,log_Marathon,log_Apache,log_Schlumb,log_Williams,log_Occid_Petr, log_Goldman,lo
g_American_exp,log_American_bank,log_Morgan,log_Metlife,log_Zimmer,
log_Stryker,log_Metler,log_JJ,log_Humana,log_Robert,log_Grumann,log_Textron,log_Bo
eing,
log_Etn,log_EA,log_Adobe,log_Microsoft,log_Oracle,log_Intel,log_Mon,log_Eastman,lo
g_Air,
log_Sealed,log_Sherwin,log_Ameren,log_Duke,log_CenterPoint,log_Next,log_AES,log_AT
,
log_Verizon,log_Century,log_Colgate,log_Coca_cola,log_Kellogs,log_Costco,log_Pepsi
```

```
)
```

Pearson Correlation Method

```
# Correlation matrix
R_hat <- cor(X[,2:49], method = "pearson")


# Here we compute a bootstrap sampling (by Date) building a list up where each pos
ition refers to a bootstrap matrix.
n <- nrow(X)
B = 1000
R_star <- list()
X1 <- X[,2:49]

for ( b in 1:B ){

  idx <- sample(1:n,replace = T)
  bsamp <- X1[idx,]
  R_star[[b]] <- cor(bsamp, method = "pearson")

}


# At this point we create the "delta boot" through the following steps:

# - First of all we create an empty matrix with the same size of our correlation m
atrix (R-hat)

# - After that, for each bootstrap matrix, we compute the abs in position [i,j] mi
nus R_hat in position [i,j].
#   We append the result in the empty matrix created in the step above

# - Eventually, we take the max values for each filled matrix and we append them i
n a vector

matrice_max1 <- matrix(NA,nrow = 48,ncol = 48)

vettore <- c()
for(k in 1:length(R_star)){
  matrice_max1 <- matrix(NA,nrow = 48,ncol = 48)
  for(i in 1:nrow(R_hat)){
    for(j in 1:ncol(R_hat)){
      matrice_max1[i,j] <- abs(R_star[[k]][i,j] - R_hat[i,j])
    }

  }
  vettore <- c(vettore,max(matrice_max1))
}
```

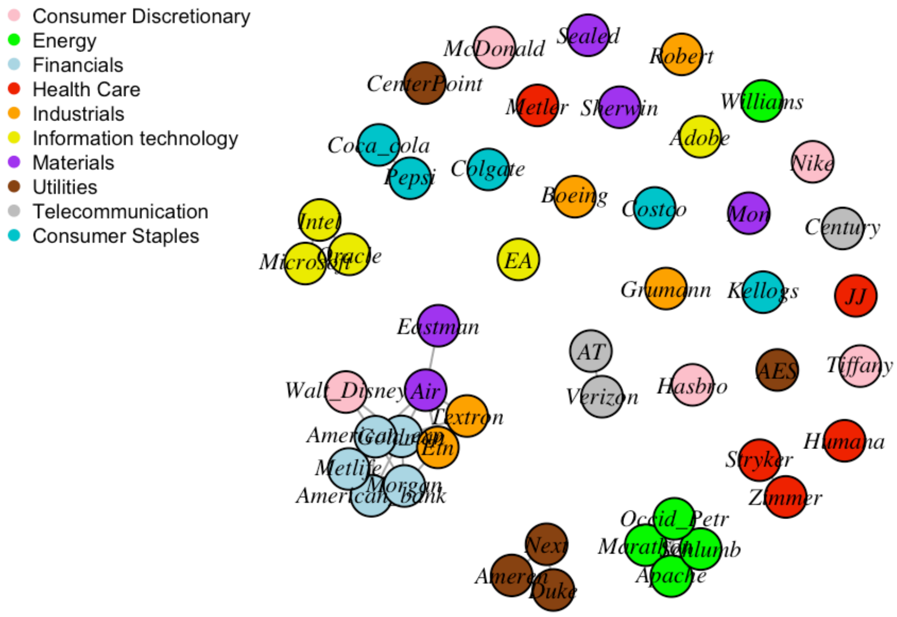
```
# Once the steps above have done, we multiply each max value stored in the vector
by sqrt(n) where n is the number of   rows of the correlation matrix

vettore_radici <- c()
for(i in vettore){
  vettore_radici <- c(vettore_radici,sqrt(n)*i)

}

e.c.d.f <- ecdf(vettore_radici)
plot(e.c.d.f,main='Pearson method ECDF',lwd=2,col="red2")
```




```
quant <- quantile(vettore_radici,probs = 0.99)

# This is the adjacency matrix which will have 1 if two nodes are connected or 0 i
f don't.
adj <- matrix(NA,nrow(R_hat),ncol(R_hat))

# Treshold which we assume both negative and positive values
epsilon <- 0.28

# Here we create the Confidence Interval. If our interval is smaller or greater th
```

```
an our threshold, so we create an edge between the nodes we're considering, no con
nection otherwise.

for ( j in 1:nrow(R_hat)){
  for ( k in 1:ncol(R_hat)){

    I.C. <- round(c(R_hat[j,k] - quant/sqrt(n),R_hat[j,k] + quant/sqrt(n)),2)

    if ( I.C. < -epsilon || I.C. > epsilon && j!=k ){
      adj[j,k] <- 1

    }else{
      adj[j,k] <- 0
    }

  }
}

# Just to fix row names and column names of adj
vect <- c()
for(i in colnames(X[,2:49])){
  v=strsplit(i,"log_")
  for(j in v[[1]]){
    if(j!=""){
      vect <- c(vect,j)
    }
  }
}

rownames(adj) <- vect
colnames(adj) <- vect


# Let's create the graph
G <- graph.adjacency(adj,mode = "undirected")

#colour nodes
V(G)[1:5]$color='pink'
V(G)[6:10]$color='green'
V(G)[11:15]$color='lightblue'
V(G)[16:20]$color='red2'
V(G)[21:25]$color='orange'
V(G)[26:30]$color='yellow2'
V(G)[31:35]$color='purple'
V(G)[36:40]$color='saddlebrown'
V(G)[41:43]$color='grey'
V(G)[44:48]$color='turquoise3'


V(G)$label.cex=.7
```

```
V(G)$label.font=3
V(G)$label.color='black'
plot(G,vertex.size=15)
legend("topleft",
  legend = c("Consumer Discretionary","Energy","Financials","Health Care","Industrials","Information technology","Materials","Utilities","Telecommunication","Consumer Staples"),
  col = c('pink','green','lightblue','red2','orange','yellow2','purple','saddlebrown','grey','turquoise3'),
  pch =20,
  bty = "n",
  pt.cex = 1,
  cex = 0.6,
  text.col = "black",
  horiz = F )
```



As shown from the graph above, most of the companies we analyzed coming from the same sector even if that's not true for all of them.

It's the case of "The Walt Disney Companies", "Etn", "Air" and other but we're going to talk about them later on.

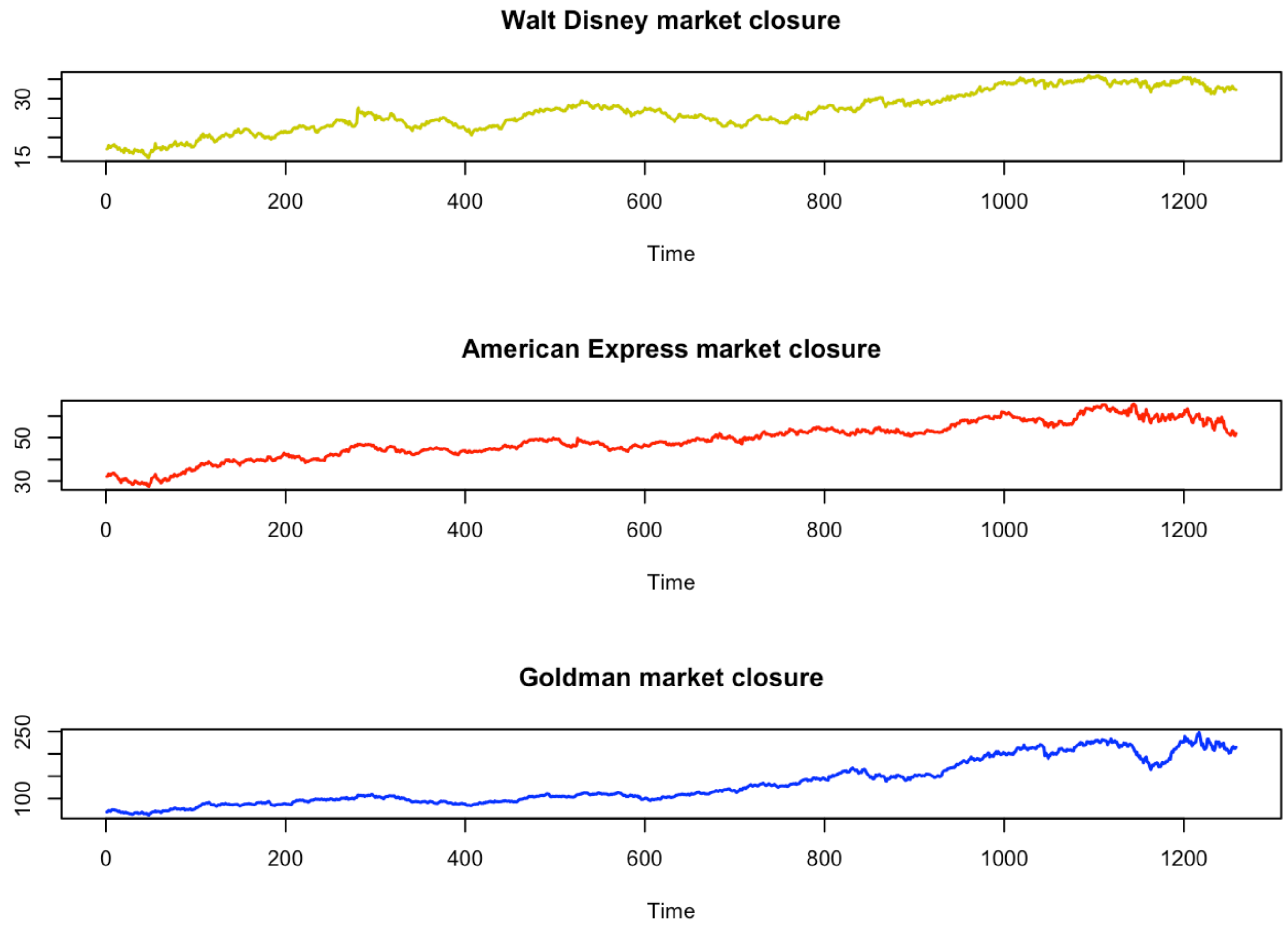
How we can see, the companies are not only connected by each other building up an own group, but also with other groups.

That means we have companies from different sectors whose market closure follows the same way (indipendently from the closure price).

Let's take a look at the **"Walt Disney"** behaviour. It's pretty tricky figuring out which companies are connected togheter (we created an interactive graph which will helps you to understand the linkages more easily see plot below), but with patience and carefully we see that Disney (which belongs to the *Consumer Discretionary* sector) is connected with other companies like **"Goldman"** and **"American Express"** which belong to the *Financial* sector.

We can show how his trend market closure follows the other companies trends:

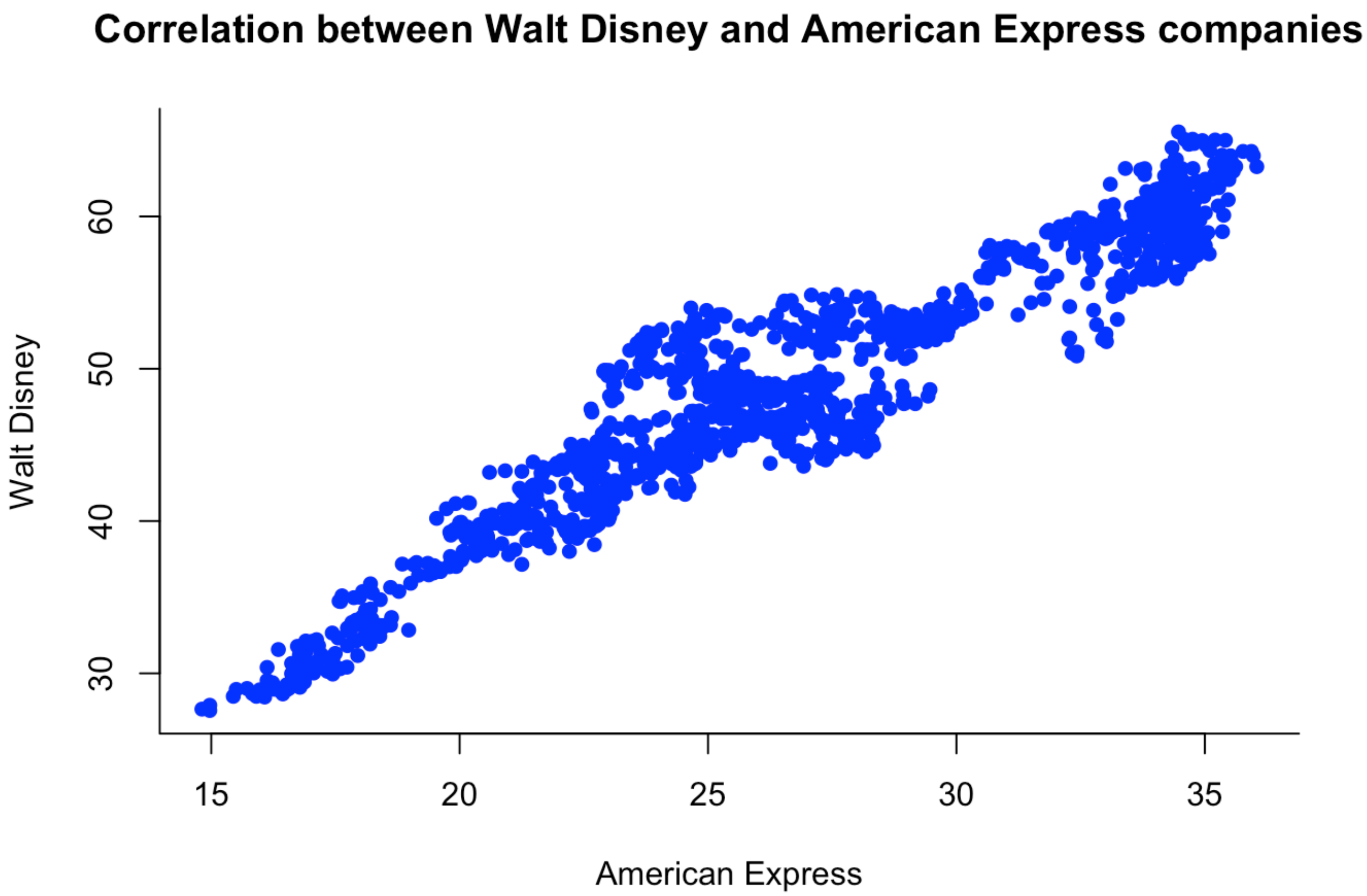
```
par(mfrow=c(3,1))
plot.ts(Walt_Disney$Close,main='Walt Disney market closure',ylab='',col='yellow3',
lwd=1.5)
plot.ts(American_exp$Close,main='American Express market closure',ylab='',col='red
',lwd=1.5)
plot.ts(Goldman$Close,main='Goldman market closure',ylab='',col='blue',lwd=1.5)
```



So, it could be reasonable that we have a linkage between them.

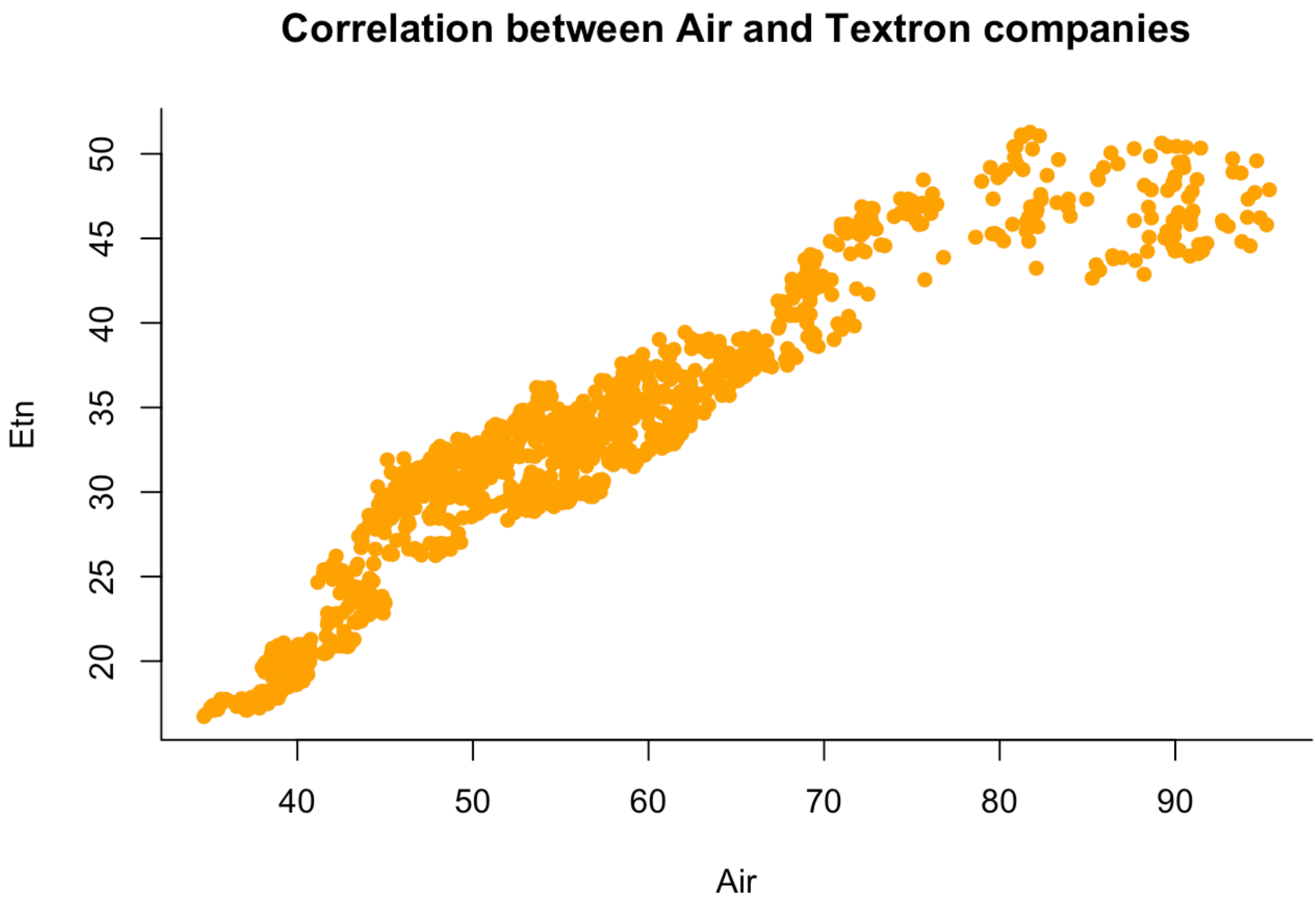
Furthermore, as shown from the plot below, the correlation from these two companies turns out to be really strong.

```
plot(Walt_Disney$Close, American_exp$Close, main = "Correlation between Walt Disne
y and American Express companies", pch = 20,xlab = "American Express", ylab = "Wal
t Disney",col = "blue", lwd = 2.5, bty = "l")
```



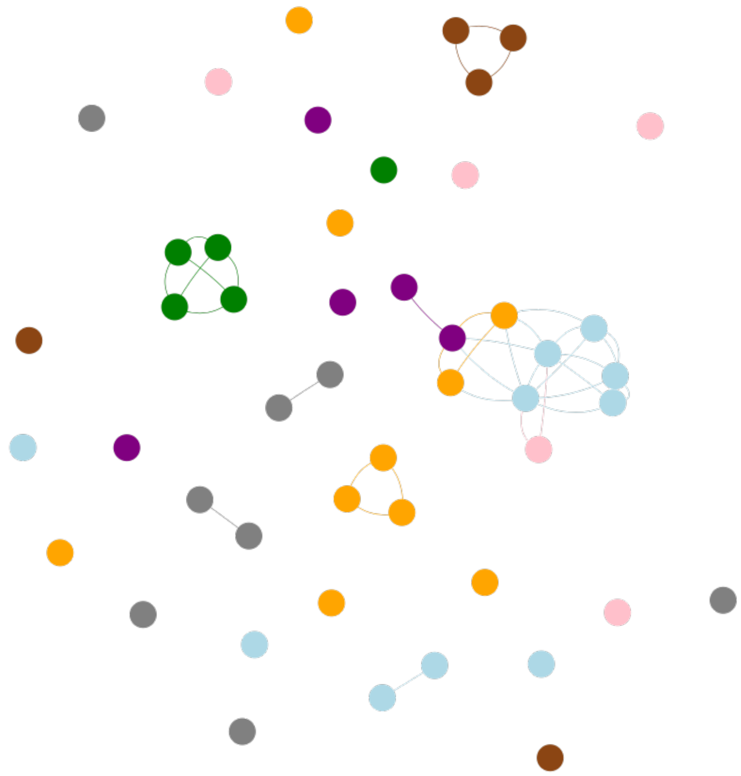
The same happens for **Air** and **Etn** companies that are linked togheter. In the same way we'll show the correlation with the plot below.

```
plot(Air$Close, Etn$Close, main = "Correlation between Air and Textron companies",
pch = 20,xlab = "Air", ylab = "Etn",col = "orange", lwd = 2.5, bty = "l")
```



Interactive Network

Select by id



Kendall Correlation Method

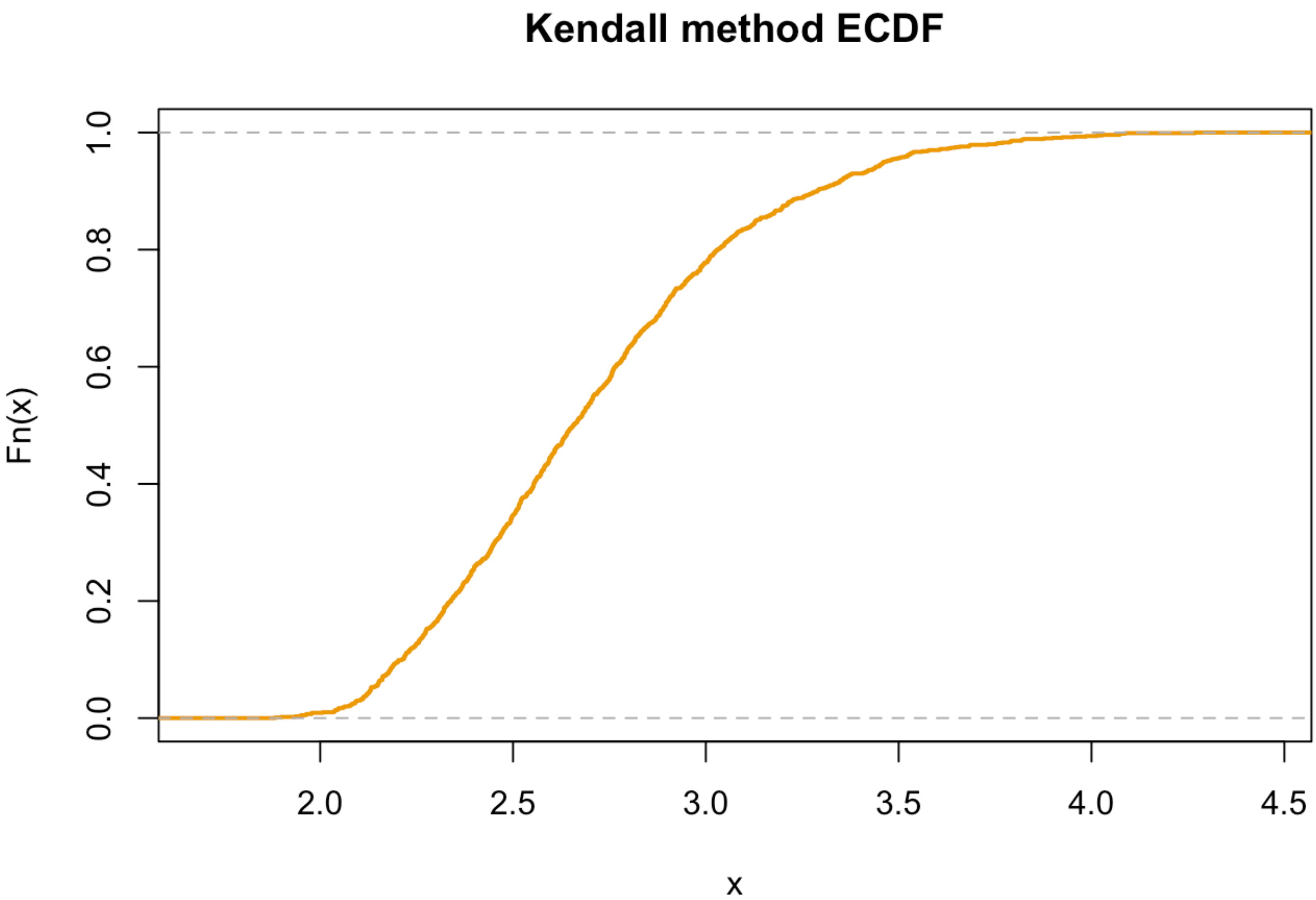
Here we’ve computed the same steps done in the previous chunks but using *Kendall* method.

In the test session, we discovered that the Kendall method takes too much time to perform the calculation, so we **parallelized** the code which use of multiple compute resources to solve a computation problem.

The code we used is the same for Pearson so we didn’t comment the code.

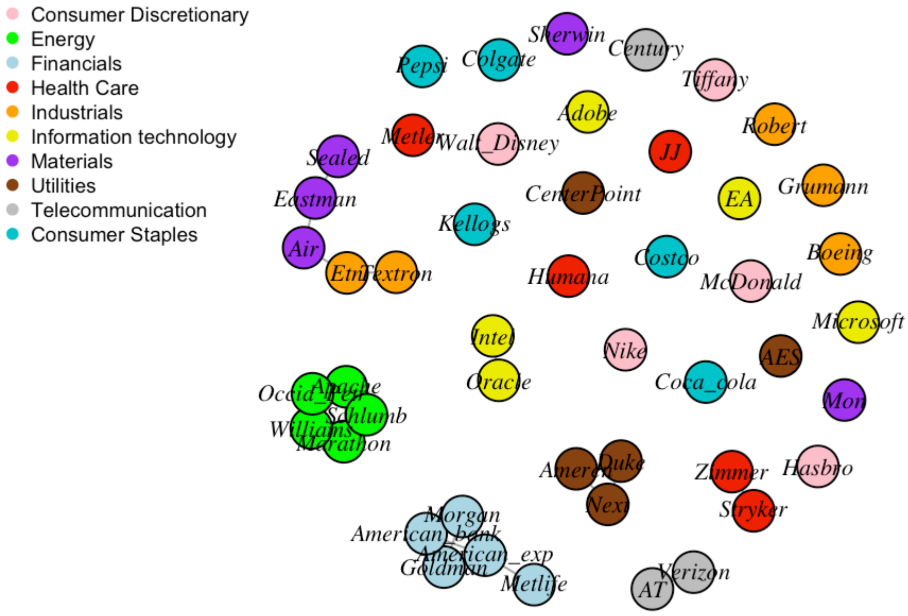
```
cl <- makeCluster(c("localhost","localhost"), type = "SOCK")

registerDoSNOW(cl = cl)


#setup parallel backend to use many processors
cores=detectCores()
cl <- makeCluster(cores[1]-1) # We use 3 cores
registerDoParallel(cl)

#kendall parallelization

R_hat <- cor(X[,2:49], method = "kendall")

n <- nrow(X)
B = 1000
R_star_kendall <- c()
X1 <- X[,2:49]

system.time(R_star_kendall<- foreach( b=1:B ,.combine=c)%dopar%{

  idx <- sample(1:n,replace = T)
  bsamp <- X1[idx,]
  list(cor(bsamp, method = "kendall"))

})

stopCluster(cl) #stop working with 3 cores

#save(R_star_kendall,file='/Users/Dario/Desktop/Brutti/HW3/R_star_kendall.RData')
```

```
load('/Users/Dario/Desktop/Brutti/HW3/R_star_kendall.RData')
R_hat <- cor(X[,2:49], method = "kendall")
matrice_max <- matrix(NA,nrow = 48,ncol = 48)

vettore_kendall <- c()
for(k in 1:length(R_star)){
  matrice_max <- matrix(NA,nrow = 48,ncol = 48)
  for(i in 1:nrow(R_hat)){
    for(j in 1:ncol(R_hat)){
      matrice_max[i,j] <- abs(R_star_kendall[[k]][i,j] - R_hat[i,j])
    }

  }
  vettore_kendall <- c(vettore_kendall,max(matrice_max))
}


vettore_kendall_radici <- c()
for (i in vettore_kendall){
  vettore_kendall_radici <- c(vettore_kendall_radici,sqrt(n)*i)
}




ecdf_kendall <- ecdf(vettore_kendall_radici)
plot(ecdf_kendall,main='Kendall method ECDF',lwd=2,col="orange2" )
```




```
quant_kendall <- quantile(vettore_kendall,probs = 0.95)


adj_kendall <- matrix(NA,nrow(R_hat),ncol(R_hat))
epsilon <- 0.35
for ( j in 1:nrow(R_hat)){
  for ( k in 1:ncol(R_hat)){

    I.C_kendall <- round(c(R_hat[j,k] - quant_kendall/sqrt(n),R_hat[j,k] + quant_k
endall/sqrt(n)),2)

    if ( I.C_kendall < -epsilon || I.C_kendall > epsilon && j!=k ){
      adj_kendall[j,k] <- 1

    }else{
      adj_kendall[j,k] <- 0
    }

  }
}
```

```
#name of companies in matrix
vect1 <- c()
for(i in colnames(X[,2:49])){
  v1=strsplit(i,"log_")
  for(j in v1[[1]]){
    if(j!=""){
      vect1 <- c(vect1,j)
    }
  }
}


rownames(adj_kendall) <- vect1
colnames(adj_kendall) <- vect1


# Let's create the graph again
G_kendall <- graph.adjacency(adj_kendall,mode = "undirected")



#colour nodes
V(G_kendall)[1:5]$color='pink'
V(G_kendall)[6:10]$color='green'
V(G_kendall)[11:15]$color='lightblue'
V(G_kendall)[16:20]$color='red2'
V(G_kendall)[21:25]$color='orange'
V(G_kendall)[26:30]$color='yellow2'
V(G_kendall)[31:35]$color='purple'
V(G_kendall)[36:40]$color='saddlebrown'
V(G_kendall)[41:43]$color='grey'
V(G_kendall)[44:48]$color='turquoise3'


V(G_kendall)$label.cex=.7
V(G_kendall)$label.font=3
V(G_kendall)$label.color='black'
plot(G_kendall,vertex.size=15)
legend("topleft",
  legend = c("Consumer Discretionary","Energy","Financials","Health Care","Industr
ials","Information technology","Materials","Utilities","Telecommunication","Consum
er Staples"),
  col = c('pink','green','lightblue','red2','orange','yellow2','purple','saddlebro
wn','grey','turquoise3'),
  pch =20,
  bty = "n",
  pt.cex = 1,
  cex = 0.6,
  text.col = "black",
  horiz = F )
```



How we can see from the graph above, the results we obtained are a little bit different respect to Pearson application. One reason it could be that Kendall is a more robust measure because it's less sensitive to outliers.

After many trials we observed that, to get an acceptable graph view, in according to have a good correlation quality between the companies, and at the same time for having a well-grouping which respect the sector of belonging, the good α and ϵ values look different respect Pearson.

We know that we have a relation between α and ϵ. Let's go to talk a little bit about these two values. We used α to create our Confidence Intervals which we used then to verify if there was an intersection with [-ϵ, ϵ].

Based on this, we created the adj matrix which described the connections between the companies.

So, it's easy to notice the logical link these two values have.

For the same value of ϵ, we have more connections between then nodes when α increases.

On the other hand, when ϵ assumes a large value, all the nodes will be no connected.

Coming back our "Kendall graph", we have that most of the companies are connected as the same as in the Pearson method.

Changes, like the IT companies (Microsoft, Oracle and so on) which are not connected anymore by each other, may have been caused by the different threshold we used.

Select by id