

Untitled

Manvydas Sokolovas ir Paulius Kostickis

10/10/2016

Reikiamos bibliotekos:

```
library("quantmod")
library("forecast")
library("xts")
library("dplyr")
library("tidyverse")
```

Nuskaityta lentelė:

```
data=read.csv("rawdata.csv")

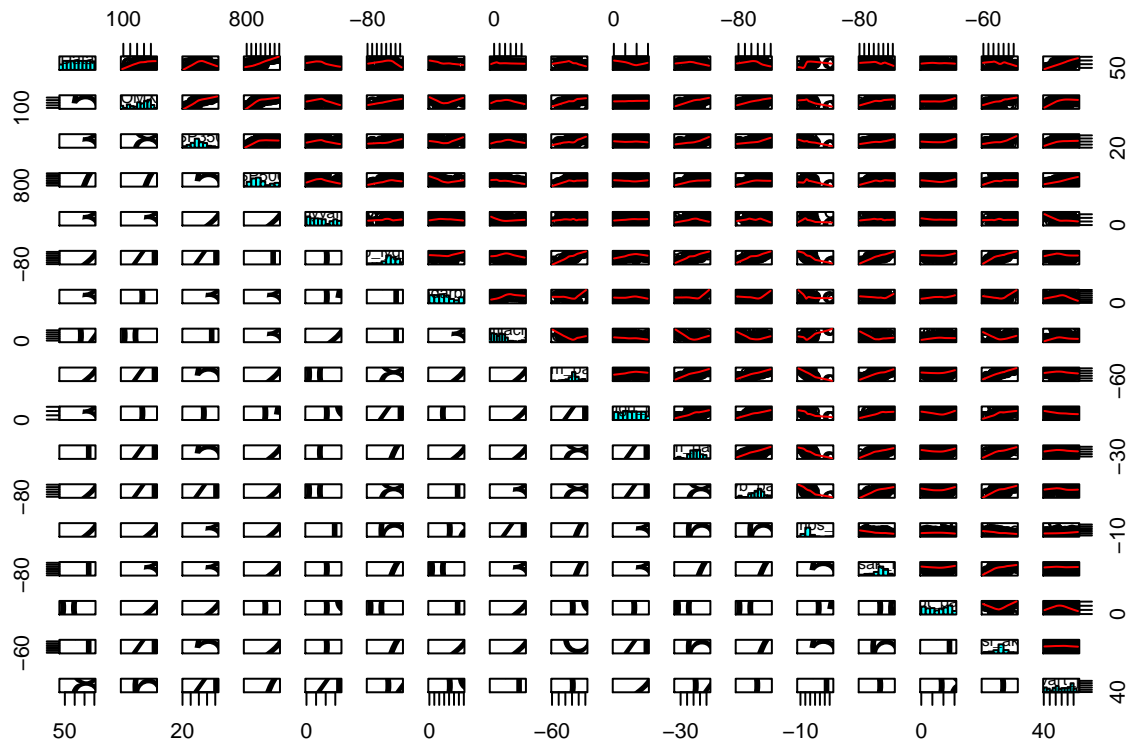
data[ data == ":" ] = NA
data=data[complete.cases(data),]
rownames(data)<-NULL
```

r ir grafikai:

```
panel.hist <- function(x, ...) #ši funkcija reikalinga grafikų lentelei išbrėžti (histogramos pateik
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor,...) #ši funkcija reikalinga grafikų le
#išbrėžti (koreliacijos koeficiento radimui)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = 3)
}

pairs(data,upper.panel=panel.smooth,diag.panel=panel.hist, lower.panel=panel.cor)
```



r ir p:

```
panel.hist <- function(x, ...) #ši funkcija reikalinga grafikų lentelei išbrėžti (histogramos pateiktos)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}
```

```
panel.cor2 <- function(x, y, digits=2, cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  test <- cor.test(x,y)
  Signif <- ifelse(round(test$p.value,3)<0.001,"p<0.001",paste("p=",round(test$p.value,3)))
  text(0.5, 0.25, paste("r=",txt))
  text(.5, .75, Signif)
}
```

```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor,...) #ši funkcija reikalinga grafikų lentelei išbrėžti (koreliacijos koeficiento radimui)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
```

```

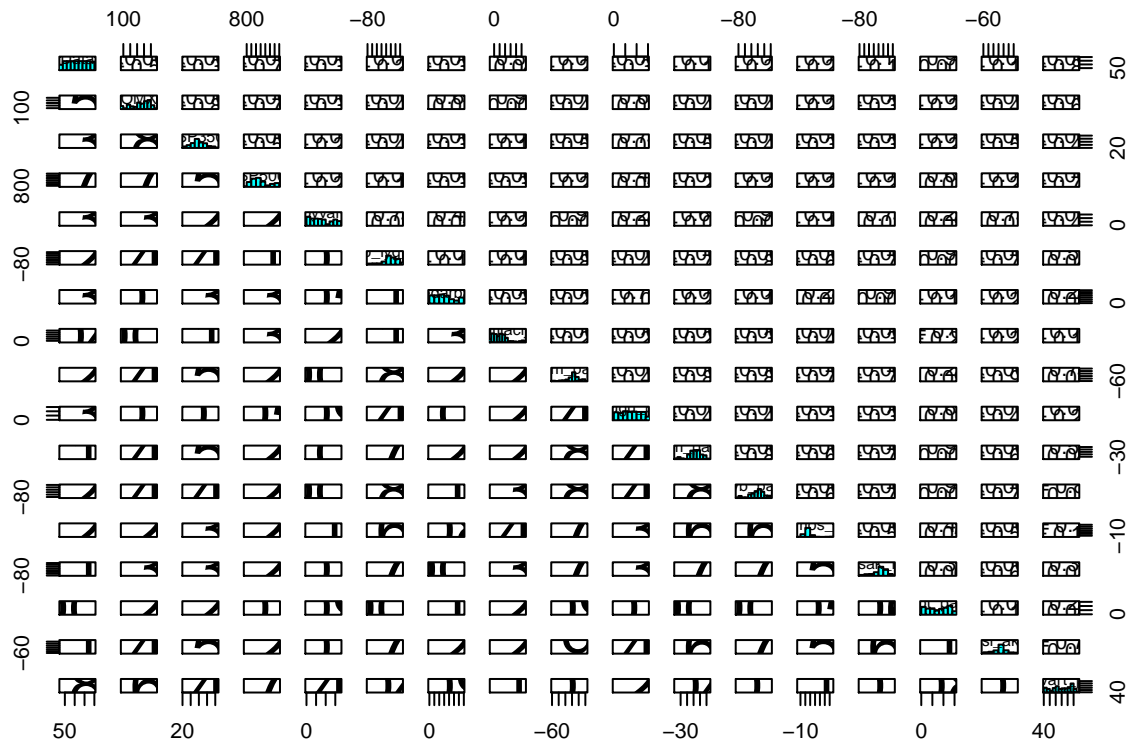
r <- abs(cor(x, y))
txt <- format(c(r, 0.123456789), digits = digits)[1]
txt <- paste0(prefix, txt)
if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
text(0.5, 0.5, txt, cex = 3)
}

```

```

pairs(data, upper.panel=panel.cor2, diag.panel=panel.hist, lower.panel=panel.cor)

```



r, p ir grafikai:

```

panel.hist <- function(x, ...) #ši funkcija reikalinga grafikų lentelei išbrėžti (histogramos pateik
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

```

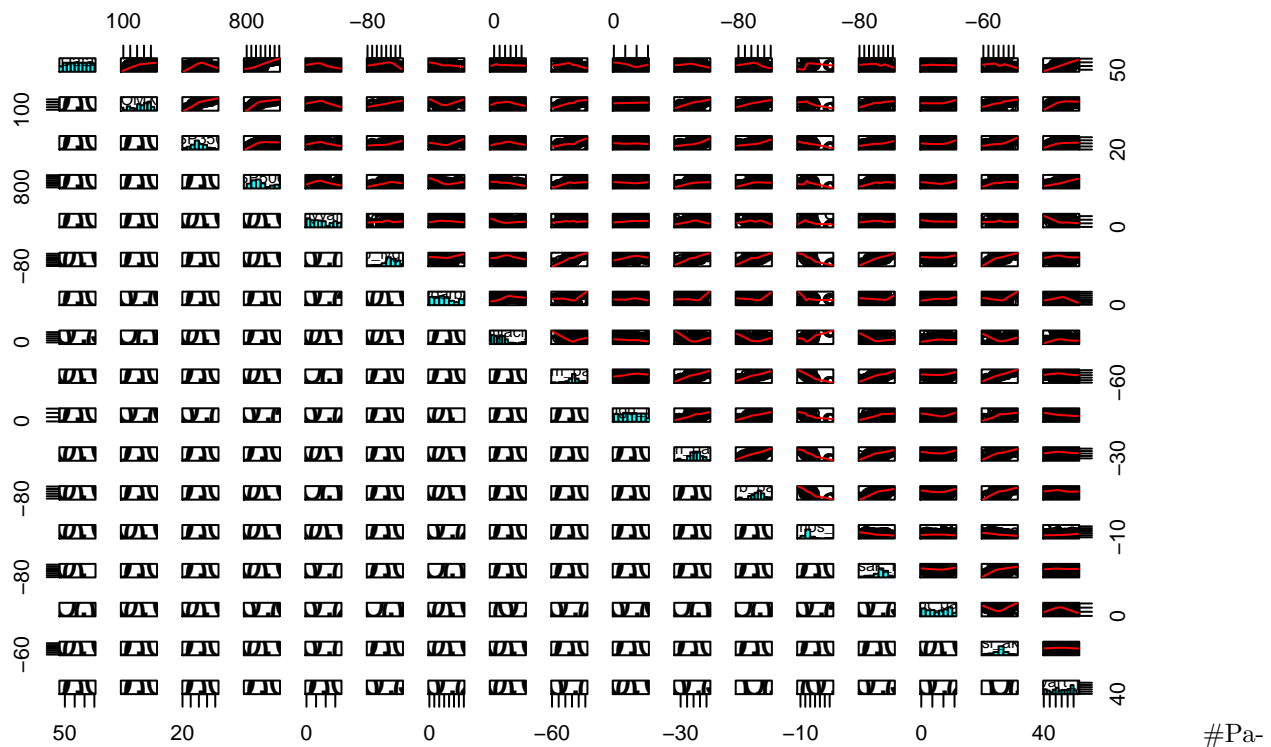
```

panel.cor2 <- function(x, y, digits=2, cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  test <- cor.test(x,y)
  Signif <- ifelse(round(test$p.value,3)<0.001,"p<0.001",paste("p=",round(test$p.value,3)))
  text(0.5, 0.25, cex=2, paste("r=",txt))
}

```

```
text(.5, .75, cex=2, Signif)
}
```

```
pairs(data, upper.panel=panel.smooth, diag.panel=panel.hist, lower.panel=panel.cor2)
```



bandyt taikyt Fisherio raw testa. Ar koreliacija reikšminga pagal kintamųjų skaičių

Duomenų klasės pakeitimas į numeric:

```
data1 <- data
eil_nr <- 1:17
klase_pries <- cbind(data.frame(sapply(data, class)), eil_nr); colnames(klase_pries) <- c("Klase pries", "eil_nr")
stulpeliai = c(5, 7, 10, 15);

data1[,stulpeliai] <- as.numeric( as.character( unlist( data1[, stulpeliai])) )

data1[,stulpeliai] <- as.numeric(as.matrix(data1[,stulpeliai]))
klase_po <- cbind(data.frame(sapply(data1, class)), klase_pries) ; colnames(klase_po) <- c("Klase po", "eil_nr")
```

Standartizavimas:

```
norm.duom <- scale(data1[,2:17])

# patikriname ar tikrai gavome vidurkį 0 ir variaciją 1:
colMeans(norm.duom)
```

##	OMX	SP350	SP500	Apyvarta
##	1.626582e-16	-4.001632e-16	-2.995492e-16	2.533812e-16
##	darb_lyg_luk	nedarbas	Infliacija	mazm_pasitik

```
##      4.236377e-17      -6.133414e-17      -6.736002e-18      4.027399e-18
## paslaugu_pasitik      pram_pasitik      statyb_pasitik      turimos_akc
##      -1.365461e-17      1.211263e-17      1.617046e-17      -2.194780e-17
##      uzsak_luk      vartot_pasitik      versl_aktyv      ilg_vart_prek
##      -1.694145e-18      1.583569e-17      -1.752984e-17      -9.539965e-17
```

```
apply(norm.duom, 2, sd)
```

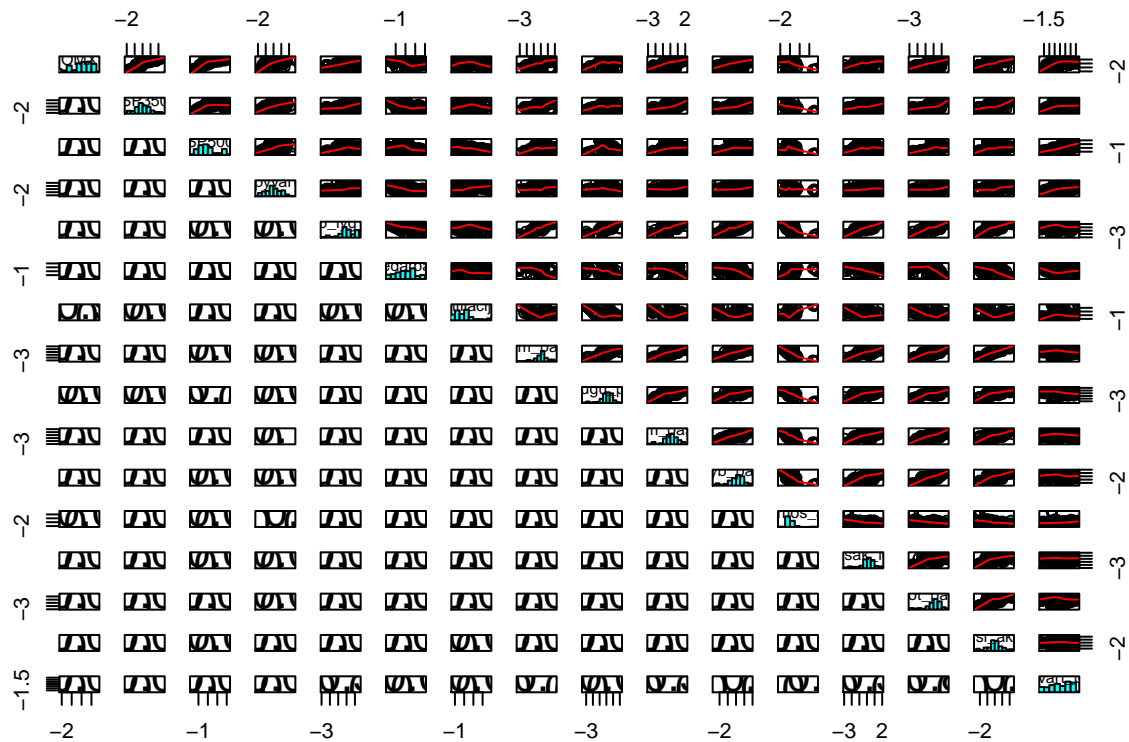
```
##      OMX      SP350      SP500      Apyvarta
##      1      1      1      1
## darb_lyg_luk      nedarbas      Infliacija      mazm_pasitik
##      1      1      1      1
## paslaugu_pasitik      pram_pasitik      statyb_pasitik      turimos_akc
##      1      1      1      1
##      uzsak_luk      vartot_pasitik      versl_aktyv      ilg_vart_prek
##      1      1      1      1
```

r, p ir grafikai:

```
panel.hist <- function(x, ...) #ši funkcija reikalinga grafikų lentelei išbrėžti (histogramos pateik
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

panel.cor2 <- function(x, y, digits=2, cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  test <- cor.test(x,y)
  Signif <- ifelse(round(test$p.value,3)<0.001,"p<0.001",paste("p=",round(test$p.value,3)))
  text(0.5, 0.25, cex=2, paste("r=",txt))
  text(.5, .75, cex=2, Signif)
}

pairs(norm.duom,upper.panel=panel.smooth,diag.panel=panel.hist, lower.panel=panel.cor2)
```



Atvaizdavimas:

```
ggplot(data=norm.duom) +
  geom_point(mapping = aes(x = OMX, y = SP350))
```

```
## Error: ggplot2 doesn't know how to deal with data of class matrix
```

```
#####
```