R-Code Visualizing of IR

Stefan Glogger

2 August 2017

Overview

Here we do the visualization. The mathematical formulas and setting of the parameters are done in separate files. Use the cache option in Markdown to safe computation time.

1. Calculate Sentiment
   1.1. 1 month Sentiment (survey regarding expectations for one month)
   1.2. 6 month Sentimen (survey regarding expectations for six months)
2. Import Data
   2.1. Sentix

Open Questions

-> QUEST

Functions and Parameters separate

```
source("parameters.R")
source("functions.R")
```

```
## Loading required package: cccp
## Loading required package: Rglpk
## Loading required package: slam
## Using the GLPK callable library version 4.47
## Loading required package: timeSeries
## Loading required package: timeDate
## Financial Risk Modelling and Portfolio Optimisation with R (version 0.4-1)
```

Data Import

Sentix

```
load(file.path(folderData, "Sentix", "SentixCalculated"))
```

There might be a problem with duplicated dates!

```
dates <- as.Date(sentix[[1]][,1], format = "%d.%m.%Y")
sum(duplicated(dates))
```

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

```
sum(dates==as.Date("2013-04-05"))
```

```
## [1] 2
```

```
dates <- unique(dates)
```

dispersion

```
sentixP1disp <- data.frame(DAX = unique(sentix[["DAX"]])$P_disp,
                           row.names = as.Date(unique(sentix[["DAX"]])[,1], format = "%d.%m.%Y"))
sentixP1disp$TEC = unique(sentix[["TEC"]])$P_disp[unique(sentix[["TEC"]])$Datum %in% dates]
sentixP1disp$ESX50 = unique(sentix[["ESX50"]])$P_disp[unique(sentix[["ESX50"]])$Datum %in% dates]
sentixP1disp$SP5 = unique(sentix[["SP5"]])$P_disp[unique(sentix[["SP5"]])$Datum %in% dates]
sentixP1disp$NASDAQ = unique(sentix[["NASDAQ"]])$P_disp[unique(sentix[["NASDAQ"]])$Datum %in% dates]
sentixP1disp$NIKKEI = unique(sentix[["NIKKEI"]])$P_disp[unique(sentix[["NIKKEI"]])$Datum %in% dates]
sentixP1disp$BUND = unique(sentix[["BUND"]])$P_disp[unique(sentix[["BUND"]])$Datum %in% dates]
sentixP1disp$TBOND = unique(sentix[["TBOND"]])$P_disp[unique(sentix[["TBOND"]])$Datum %in% dates]


sentixI1disp <- data.frame(DAX = unique(sentix[["DAX"]])$I_disp,
                           row.names = as.Date(unique(sentix[["DAX"]])[,1], format = "%d.%m.%Y"))
sentixI1disp$TEC = unique(sentix[["TEC"]])$I_disp[unique(sentix[["TEC"]])$Datum %in% dates]
sentixI1disp$ESX50 = unique(sentix[["ESX50"]])$I_disp[unique(sentix[["ESX50"]])$Datum %in% dates]
sentixI1disp$SP5 = unique(sentix[["SP5"]])$I_disp[unique(sentix[["SP5"]])$Datum %in% dates]
sentixI1disp$NASDAQ = unique(sentix[["NASDAQ"]])$I_disp[unique(sentix[["NASDAQ"]])$Datum %in% dates]
sentixI1disp$NIKKEI = unique(sentix[["NIKKEI"]])$I_disp[unique(sentix[["NIKKEI"]])$Datum %in% dates]
sentixI1disp$BUND = unique(sentix[["BUND"]])$I_disp[unique(sentix[["BUND"]])$Datum %in% dates]
sentixI1disp$TBOND = unique(sentix[["TBOND"]])$I_disp[unique(sentix[["TBOND"]])$Datum %in% dates]


sentixG1disp <- data.frame(DAX = unique(sentix[["DAX"]])$G_disp,
                           row.names = as.Date(unique(sentix[["DAX"]])[,1], format = "%d.%m.%Y"))
sentixG1disp$TEC = unique(sentix[["TEC"]])$G_disp[unique(sentix[["TEC"]])$Datum %in% dates]
sentixG1disp$ESX50 = unique(sentix[["ESX50"]])$G_disp[unique(sentix[["ESX50"]])$Datum %in% dates]
sentixG1disp$SP5 = unique(sentix[["SP5"]])$G_disp[unique(sentix[["SP5"]])$Datum %in% dates]
sentixG1disp$NASDAQ = unique(sentix[["NASDAQ"]])$G_disp[unique(sentix[["NASDAQ"]])$Datum %in% dates]
sentixG1disp$NIKKEI = unique(sentix[["NIKKEI"]])$G_disp[unique(sentix[["NIKKEI"]])$Datum %in% dates]
sentixG1disp$BUND = unique(sentix[["BUND"]])$G_disp[unique(sentix[["BUND"]])$Datum %in% dates]
sentixG1disp$TBOND = unique(sentix[["TBOND"]])$G_disp[unique(sentix[["TBOND"]])$Datum %in% dates]
```

```
sentixP6disp <- data.frame(DAX = unique(sentix[["DAXm"]])$P_disp,
                           row.names = as.Date(unique(sentix[["DAXm"]])[,1], format = "%d.%m.%Y"))
sentixP6disp$TEC = unique(sentix[["TECm"]])$P_disp[unique(sentix[["TECm"]])$Datum %in% dates]
sentixP6disp$ESX50 = unique(sentix[["ESX50m"]])$P_disp[unique(sentix[["ESX50m"]])$Datum %in% dates]
sentixP6disp$SP5 = unique(sentix[["SP5m"]])$P_disp[unique(sentix[["SP5m"]])$Datum %in% dates]
sentixP6disp$NASDAQ = unique(sentix[["NASDAQm"]])$P_disp[unique(sentix[["NASDAQm"]])$Datum %in% dates]
sentixP6disp$NIKKEI = unique(sentix[["NIKKEIm"]])$P_disp[unique(sentix[["NIKKEIm"]])$Datum %in% dates]
sentixP6disp$BUND = unique(sentix[["BUNDm"]])$P_disp[unique(sentix[["BUNDm"]])$Datum %in% dates]
sentixP6disp$TBOND = unique(sentix[["TBONDm"]])$P_disp[unique(sentix[["TBONDm"]])$Datum %in% dates]


sentixI6disp <- data.frame(DAX = unique(sentix[["DAXm"]])$I_disp,
                           row.names = as.Date(unique(sentix[["DAXm"]])[,1], format = "%d.%m.%Y"))
sentixI6disp$TEC = unique(sentix[["TECm"]])$I_disp[unique(sentix[["TECm"]])$Datum %in% dates]
sentixI6disp$ESX50 = unique(sentix[["ESX50m"]])$I_disp[unique(sentix[["ESX50m"]])$Datum %in% dates]
sentixI6disp$SP5 = unique(sentix[["SP5m"]])$I_disp[unique(sentix[["SP5m"]])$Datum %in% dates]
sentixI6disp$NASDAQ = unique(sentix[["NASDAQm"]])$I_disp[unique(sentix[["NASDAQm"]])$Datum %in% dates]
sentixI6disp$NIKKEI = unique(sentix[["NIKKEIm"]])$I_disp[unique(sentix[["NIKKEIm"]])$Datum %in% dates]
sentixI6disp$BUND = unique(sentix[["BUNDm"]])$I_disp[unique(sentix[["BUNDm"]])$Datum %in% dates]
sentixI6disp$TBOND = unique(sentix[["TBONDm"]])$I_disp[unique(sentix[["TBONDm"]])$Datum %in% dates]


sentixG6disp <- data.frame(DAX = unique(sentix[["DAXm"]])$G_disp,
                           row.names = as.Date(unique(sentix[["DAXm"]])[,1], format = "%d.%m.%Y"))
sentixG6disp$TEC = unique(sentix[["TECm"]])$G_disp[unique(sentix[["TECm"]])$Datum %in% dates]
sentixG6disp$ESX50 = unique(sentix[["ESX50m"]])$G_disp[unique(sentix[["ESX50m"]])$Datum %in% dates]
sentixG6disp$SP5 = unique(sentix[["SP5m"]])$G_disp[unique(sentix[["SP5m"]])$Datum %in% dates]
sentixG6disp$NASDAQ = unique(sentix[["NASDAQm"]])$G_disp[unique(sentix[["NASDAQm"]])$Datum %in% dates]
sentixG6disp$NIKKEI = unique(sentix[["NIKKEIm"]])$G_disp[unique(sentix[["NIKKEIm"]])$Datum %in% dates]
sentixG6disp$BUND = unique(sentix[["BUNDm"]])$G_disp[unique(sentix[["BUNDm"]])$Datum %in% dates]
sentixG6disp$TBOND = unique(sentix[["TBONDm"]])$G_disp[unique(sentix[["TBONDm"]])$Datum %in% dates]
```

herfindah

```
sentixP1herf <- data.frame(DAX = unique(sentix[["DAX"]])$P_herf,
                           row.names = as.Date(unique(sentix[["DAX"]])[,1], format = "%d.%m.%Y"))
sentixP1herf$TEC = unique(sentix[["TEC"]])$P_herf[unique(sentix[["TEC"]])$Datum %in% dates]
sentixP1herf$ESX50 = unique(sentix[["ESX50"]])$P_herf[unique(sentix[["ESX50"]])$Datum %in% dates]
sentixP1herf$SP5 = unique(sentix[["SP5"]])$P_herf[unique(sentix[["SP5"]])$Datum %in% dates]
sentixP1herf$NASDAQ = unique(sentix[["NASDAQ"]])$P_herf[unique(sentix[["NASDAQ"]])$Datum %in% dates]
sentixP1herf$NIKKEI = unique(sentix[["NIKKEI"]])$P_herf[unique(sentix[["NIKKEI"]])$Datum %in% dates]
sentixP1herf$BUND = unique(sentix[["BUND"]])$P_herf[unique(sentix[["BUND"]])$Datum %in% dates]
sentixP1herf$TBOND = unique(sentix[["TBOND"]])$P_herf[unique(sentix[["TBOND"]])$Datum %in% dates]


sentixI1herf <- data.frame(DAX = unique(sentix[["DAX"]])$I_herf,
                           row.names = as.Date(unique(sentix[["DAX"]])[,1], format = "%d.%m.%Y"))
sentixI1herf$TEC = unique(sentix[["TEC"]])$I_herf[unique(sentix[["TEC"]])$Datum %in% dates]
sentixI1herf$ESX50 = unique(sentix[["ESX50"]])$I_herf[unique(sentix[["ESX50"]])$Datum %in% dates]
sentixI1herf$SP5 = unique(sentix[["SP5"]])$I_herf[unique(sentix[["SP5"]])$Datum %in% dates]
sentixI1herf$NASDAQ = unique(sentix[["NASDAQ"]])$I_herf[unique(sentix[["NASDAQ"]])$Datum %in% dates]
```

```
sentixI1herf$NIKKEI = unique(sentix[["NIKKEI"]])$I_herf[unique(sentix[["NIKKEI"]])$Datum %in% dates]
sentixI1herf$BUND = unique(sentix[["BUND"]])$I_herf[unique(sentix[["BUND"]])$Datum %in% dates]
sentixI1herf$TBOND = unique(sentix[["TBOND"]])$I_herf[unique(sentix[["TBOND"]])$Datum %in% dates]


sentixG1herf <- data.frame(DAX = unique(sentix[["DAX"]])$G_herf,
                           row.names = as.Date(unique(sentix[["DAX"]])[,1], format = "%d.%m.%Y"))
sentixG1herf$TEC = unique(sentix[["TEC"]])$G_herf[unique(sentix[["TEC"]])$Datum %in% dates]
sentixG1herf$ESX50 = unique(sentix[["ESX50"]])$G_herf[unique(sentix[["ESX50"]])$Datum %in% dates]
sentixG1herf$SP5 = unique(sentix[["SP5"]])$G_herf[unique(sentix[["SP5"]])$Datum %in% dates]
sentixG1herf$NASDAQ = unique(sentix[["NASDAQ"]])$G_herf[unique(sentix[["NASDAQ"]])$Datum %in% dates]
sentixG1herf$NIKKEI = unique(sentix[["NIKKEI"]])$G_herf[unique(sentix[["NIKKEI"]])$Datum %in% dates]
sentixG1herf$BUND = unique(sentix[["BUND"]])$G_herf[unique(sentix[["BUND"]])$Datum %in% dates]
sentixG1herf$TBOND = unique(sentix[["TBOND"]])$G_herf[unique(sentix[["TBOND"]])$Datum %in% dates]



sentixP6herf <- data.frame(DAX = unique(sentix[["DAXm"]])$P_herf,
                           row.names = as.Date(unique(sentix[["DAXm"]])[,1], format = "%d.%m.%Y"))
sentixP6herf$TEC = unique(sentix[["TECm"]])$P_herf[unique(sentix[["TECm"]])$Datum %in% dates]
sentixP6herf$ESX50 = unique(sentix[["ESX50m"]])$P_herf[unique(sentix[["ESX50m"]])$Datum %in% dates]
sentixP6herf$SP5 = unique(sentix[["SP5m"]])$P_herf[unique(sentix[["SP5m"]])$Datum %in% dates]
sentixP6herf$NASDAQ = unique(sentix[["NASDAQm"]])$P_herf[unique(sentix[["NASDAQm"]])$Datum %in% dates]
sentixP6herf$NIKKEI = unique(sentix[["NIKKEIm"]])$P_herf[unique(sentix[["NIKKEIm"]])$Datum %in% dates]
sentixP6herf$BUND = unique(sentix[["BUNDm"]])$P_herf[unique(sentix[["BUNDm"]])$Datum %in% dates]
sentixP6herf$TBOND = unique(sentix[["TBONDm"]])$P_herf[unique(sentix[["TBONDm"]])$Datum %in% dates]


sentixI6herf <- data.frame(DAX = unique(sentix[["DAXm"]])$I_herf,
                           row.names = as.Date(unique(sentix[["DAXm"]])[,1], format = "%d.%m.%Y"))
sentixI6herf$TEC = unique(sentix[["TECm"]])$I_herf[unique(sentix[["TECm"]])$Datum %in% dates]
sentixI6herf$ESX50 = unique(sentix[["ESX50m"]])$I_herf[unique(sentix[["ESX50m"]])$Datum %in% dates]
sentixI6herf$SP5 = unique(sentix[["SP5m"]])$I_herf[unique(sentix[["SP5m"]])$Datum %in% dates]
sentixI6herf$NASDAQ = unique(sentix[["NASDAQm"]])$I_herf[unique(sentix[["NASDAQm"]])$Datum %in% dates]
sentixI6herf$NIKKEI = unique(sentix[["NIKKEIm"]])$I_herf[unique(sentix[["NIKKEIm"]])$Datum %in% dates]
sentixI6herf$BUND = unique(sentix[["BUNDm"]])$I_herf[unique(sentix[["BUNDm"]])$Datum %in% dates]
sentixI6herf$TBOND = unique(sentix[["TBONDm"]])$I_herf[unique(sentix[["TBONDm"]])$Datum %in% dates]


sentixG6herf <- data.frame(DAX = unique(sentix[["DAXm"]])$G_herf,
                           row.names = as.Date(unique(sentix[["DAXm"]])[,1], format = "%d.%m.%Y"))
sentixG6herf$TEC = unique(sentix[["TECm"]])$G_herf[unique(sentix[["TECm"]])$Datum %in% dates]
sentixG6herf$ESX50 = unique(sentix[["ESX50m"]])$G_herf[unique(sentix[["ESX50m"]])$Datum %in% dates]
sentixG6herf$SP5 = unique(sentix[["SP5m"]])$G_herf[unique(sentix[["SP5m"]])$Datum %in% dates]
sentixG6herf$NASDAQ = unique(sentix[["NASDAQm"]])$G_herf[unique(sentix[["NASDAQm"]])$Datum %in% dates]
sentixG6herf$NIKKEI = unique(sentix[["NIKKEIm"]])$G_herf[unique(sentix[["NIKKEIm"]])$Datum %in% dates]
sentixG6herf$BUND = unique(sentix[["BUNDm"]])$G_herf[unique(sentix[["BUNDm"]])$Datum %in% dates]
sentixG6herf$TBOND = unique(sentix[["TBONDm"]])$G_herf[unique(sentix[["TBONDm"]])$Datum %in% dates]
```

Stocks

QUEST: take data of Yahoo Finance

Take data from Yahoo Finance. Take closing course from *dateMin* to *dateMax* for several indexes.

Take the following as sources of the data:

- DAX *^GDAXI*
- TEC *^TECDAX*
- ESX50 *^STOXX50E*
- SP500 *^GSPC*
- NASDAQ *^NDX*
- NIKKEI *^N225*
- BUND not from yahoo, manually from bundesbank *BBK01.WT0557*
- TBOND workaround with ETF *TLH*

```
# install.packages("quantmod")
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following object is masked from 'package:timeSeries':
##
##     time<-
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
# ?getSymbols
```

```
stocks <- data.frame(Datum = dates)

# DAX
dax <- new.env()
getSymbols("^GDAXI", env = dax, src = "yahoo", from = dateMin, to = dateMax)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

```
## Warning: ^GDAXI contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
```

```
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "GDAXI"
```

```
DAX <- data.frame(dax$GDAXI[dates,"GDAXI.Close"])
colnames(DAX) <- "Close" # somehow the column name cannot be given directly
DAX$Datum <- as.Date(row.names(DAX))

stocks$DAX <- merge(stocks, DAX, by = "Datum", all.x = T)$Close


# TEC
tec <- new.env()
getSymbols("^TECDAX", env = tec, src = "yahoo", from = dateMin, to = dateMax)
```

```
## Warning: ^TECDAX contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "TECDAX"
```

```
TEC <- data.frame(tec$TECDAX[dates, "TECDAX.Close"])
colnames(TEC) <- "Close"
TEC$Datum <- as.Date(row.names(TEC))

stocks$TEC <- merge(stocks, TEC, by = "Datum", all.x = T)$Close


# ESX50
esx50 <- new.env()
getSymbols("^STOXX50E", env = esx50, src = "yahoo", from = dateMin, to = dateMax)
```

```
## Warning: ^STOXX50E contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "STOXX50E"
```

```
ESX50 <- data.frame(esx50$STOXX50E[dates,"STOXX50E.Close"])
colnames(ESX50) <- "Close"
ESX50$Datum <- as.Date(row.names(ESX50))

stocks$ESX50 <- merge(stocks, ESX50, by = "Datum", all.x = T)$Close


# SP500
sp500 <- new.env()
getSymbols("^GSPC", env = sp500, src = "yahoo", from = dateMin, to = dateMax)
```

```
## [1] "GSPC"
```

```
SP500 <- data.frame(sp500$GSPC[dates,"GSPC.Close"])
colnames(SP500) <- "Close"
SP500$Datum <- as.Date(row.names(SP500))
# sum(is.na(SP500$Close))

stocks$SP5 <- merge(stocks, SP500, by = "Datum", all.x = T)$Close
```

```
# NASDAQ
nasdaq <- new.env()
getSymbols("^NDX", env = nasdaq, src = "yahoo", from = dateMin, to = dateMax)
```

```
## [1] "NDX"
```

```
NASDAQ <- data.frame(nasdaq$NDX[dates,"NDX.Close"])
# sum(is.na(NASDAQ[,"NDX.Close"]))
colnames(NASDAQ) <- "Close"
NASDAQ$Datum <- as.Date(row.names(NASDAQ))

stocks$NASDAQ <- merge(stocks, NASDAQ, by = "Datum", all.x = T)$Close


# NIKKEI
nikkei <- new.env()
getSymbols("^N225", env = nikkei, src = "yahoo", from = dateMin, to = dateMax)
```

```
## Warning: ^N225 contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "N225"
```

```
NIKKEI <- data.frame(nikkei$N225[dates,"N225.Close"])
colnames(NIKKEI) <- "Close"
NIKKEI$Datum <- as.Date(row.names(NIKKEI))

stocks$NIKKEI <- merge(stocks, NIKKEI, by = "Datum", all.x = T)$Close
```

Bundesanleihe not to get from yahoo

```
env <- new.env()
getSymbols("FGBLU7.EX", env = bund, src = "yahoo", from = dateMin, to = dateMax)
getSymbols("FGBLH8.EX" env = bund, src = "yahoo", from = dateMin, to = dateMax)
```

new: Den Bund-Future habe ich bei onvista in 5-Jahresstücken geladen und zusammengebaut. Dezimaltrennzeichen umgestellt im .csv

```
BUND <- read.csv(file.path(folderData, "Bundfuture", "Bundfuture2001-2017.csv"), sep = ";")
BUND[,1] <- as.Date(BUND[,1], format = "%d.%m.%Y")
BUND <- BUND[BUND[,1] %in% dates,]
BUND <- as.data.frame(BUND)

stocks$BUND <- merge(stocks, BUND, by = "Datum", all.x = T)$Schluss
```

old: Bundesanleihen von https://www.bundesbank.de/Navigation/DE/Statistiken/Zeitreihen_Datenbanken/Makrooekonomische_Zeitreihen/its_details_value_node.html?tsId=BBK01.WT0557 Zeitreihe BBK01.WT0557: Ungewogene Umlaufsrendite der an der EUREX jeweils lieferbaren Bundeswertpapiere / Mittlere RLZ von 9 bis einschl. 10 Jahre / Tageswerte

```
# BUND <- read.csv(file.path(folderData, "Indexdaten", "BBK01.WT0557.csv"), sep = "\t")
# colnames(BUND) <- c("Datum", "Kurs")
# BUND[,1] <- as.Date(BUND[,1], format = "%d.%m.%Y")
# BUND <- BUND[BUND[,1] %in% dates,]
# BUND <- as.data.frame(BUND)
#
```

```
# stocks$BUND <- merge(stocks, BUND, by = "Datum", all.x = T)$Kurs
```

Treasury bond

new: Beim T-Bond ist es die 10 Year Treasury Note, auf welche das TBOND Sentiment abzielt. Diese habe ich bei FRED geladen: https://fred.stlouisfed.org/series/DGS10

```
TBOND <- read.csv(file.path(folderData, "10 year T-Notes", "DGS10.csv"), sep = ",")
TBOND[,1] <- as.Date(TBOND[,1], format = "%Y-%m-%d")
TBOND[,2] <- as.numeric(as.character(TBOND[,2])) # was a factor first and factors are stored via index
```

```
## Warning: NAs durch Umwandlung erzeugt
```

```
colnames(TBOND) <- c("Datum", "DGS10")
TBOND <- TBOND[TBOND[,1] %in% dates,]
TBOND <- as.data.frame(TBOND)

stocks$TBOND <- merge(stocks, TBOND, by = "Datum", all.x = T)$DGS10
```

old: from *Link Yahoo* iShares 10-20 Year Treasury Bond ETF (TLH)

```
# tbond <- new.env()
#
# getSymbols("TLH", env = tbond, src = "yahoo", from = dateMin, to = dateMax)
# TBOND <- data.frame(tbond$TLH[dates,"TLH.Close"])
# colnames(TBOND) <- "Close"
# TBOND$Datum <- as.Date(row.names(TBOND))
#
# stocks$TBOND <- merge(stocks, TBOND, by = "Datum", all.x = T)$Close
```

Data Preparation

na's

There might be dates missing.

```
colSums(is.na.data.frame(stocks))
```

```
##  Datum    DAX    TEC  ESX50    SP5 NASDAQ NIKKEI   BUND  TBOND
##      0     25     22     41     26     26     32     56     22
```

We delete dates with missing values.

Work with expressions to keep code nice: https://stackoverflow.com/questions/1743698/evaluate-expression-given-as-a-string

```
stocks <- stocks[complete.cases(stocks),]
dates <- stocks[,1]

updateDates <- function(d){
    return(d[as.Date(rownames(d)) %in% dates, ])
}


i = sentixDataNames[1]
parse(text = paste0(i, " <- ", "updateDates(", i, ")"))
```

```
## expression(sentixI1disp <- updateDates(sentixI1disp))
```

```
for (i in sentixDataNames){
    eval(parse(text = paste0(i, " <- ", "updateDates(", i, ")")))
}
### not needed any more (done in three lines above) :) :) :)
# sentixI1disp <- updateDates(sentixI1disp)
# sentixP1disp <- updateDates(sentixP1disp)
# sentixG1disp <- updateDates(sentixG1disp)
# sentixI1herf <- updateDates(sentixI1herf)
# sentixG1herf <- updateDates(sentixP1herf)
# sentixP1herf <- updateDates(sentixG1herf)
#
# sentixI6disp <- updateDates(sentixI6disp)
# sentixP6disp <- updateDates(sentixP6disp)
# sentixG6disp <- updateDates(sentixG6disp)
# sentixI6herf <- updateDates(sentixI6herf)
# sentixG6herf <- updateDates(sentixP6herf)
# sentixP6herf <- updateDates(sentixG6herf)

colSums(is.na.data.frame(stocks))
```

```
##  Datum    DAX    TEC  ESX50    SP5 NASDAQ NIKKEI   BUND  TBOND
##      0      0      0      0      0      0      0      0      0
```

```
colSums(is.na.data.frame(sentixI1disp)) + colSums(is.na.data.frame(sentixP1disp)) + colSums(is.na.data.
```

```
##    DAX    TEC  ESX50    SP5 NASDAQ NIKKEI   BUND  TBOND
##      0      0      0      0      0      0      0    693
```

```
colSums(is.na.data.frame(sentixI6disp)) + colSums(is.na.data.frame(sentixP6disp)) + colSums(is.na.data.
```

```
##    DAX    TEC  ESX50    SP5 NASDAQ NIKKEI   BUND  TBOND
##      0      0      0      0      0      0      0    693
```

remove TBOND

```
stocks <- stocks[,-which(colnames(stocks)=="TBOND")]

i <- sentixDataNames[1]
parse(text = paste0(i, " <- ", i, "[,-which(colnames(", i, ") == \"TBOND\")]"))
```

```
## expression(sentixI1disp <- sentixI1disp[, -which(colnames(sentixI1disp) ==
##     "TBOND")])
```

```
for (i in sentixDataNames){
    eval(parse(text = paste0(i, " <- ", i, "[,-which(colnames(", i, ") == \"TBOND\")]")))
}
```

regress Sentiment

```
i <- sentixDataNames[1]
parse(text = paste0(i, "Reg", " <- ", "regSent(", i, ")"))
```

```
## expression(sentixI1dispReg <- regSent(sentixI1disp))
```

```
for (i in sentixDataNames){
    eval(parse(text = paste0(i, "Reg", " <- ", "regSent(", i, ")")))
```

```
}
```

```
i <- sentixDataNames[i]
parse(text = paste0(i, "RegCov", " <- ", "cov(", i, "Reg)"))
```

```
## expression(NARegCov <- cov(NAReg))
```

```
for(i in sentixDataNames){
    eval(parse(text = paste0(i, "RegCov", " <- ", "cov(", i, "Reg)")))
}
```