Ravenpack news and sovereign debt trading

S.Duprey

January 4, 2016

```
### Launching all pair spread computations
### Trying to forecast the spread between ?/? bonds futures
# library("SIT")
library("RPQuantUtils")
```

```
## Loading required package: waveslim
##
## waveslim: Wavelet Method for 1/2/3D Signals (version = 1.7.5)
##
## Loading required package: sendmailR
```

```
library("RPToolsDB")
```

```
## Loading required package: RPostgreSQL
## Loading required package: DBI
## Loading required package: lubridate
##
## Attaching package: 'lubridate'
##
## The following object is masked from 'package:waveslim':
##
##     pm
```

```
library("RPBackTesting")
```

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

## Warning: package 'xts' was built under R version 3.2.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.2.3

##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Attaching package: 'PerformanceAnalytics'
##
## The following object is masked from 'package:graphics':
##
##     legend
```

```r
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```r
require(scales)
```

```
## Loading required package: scales
```

```r
require(grid)
```

```
## Loading required package: grid
```

```r
require("ppcor")
```

```
## Loading required package: ppcor
```

```r
require(graphics)
require("TTR")
```

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

```r
require(plyr)
```

```
## Loading required package: plyr
##
## Attaching package: 'plyr'
##
## The following object is masked from 'package:lubridate':
##
##     here
```

```r
# require(reshape)
require(reshape2)
```

```
## Loading required package: reshape2
```

```r
require(RColorBrewer)
```

```
## Loading required package: RColorBrewer
```

```r
require(stats)
require(Rsolnp)
```

```
## Loading required package: Rsolnp
## Loading required package: truncnorm
## Loading required package: parallel
```

```r
require(zoo)
require(xts)
require(vars)
```

```
## Loading required package: vars
## Loading required package: MASS
## Loading required package: strucchange
## Loading required package: sandwich
## Loading required package: urca
## Loading required package: lmtest
```

```r
# require(Quandl)
require(rpart)
```

```
## Loading required package: rpart
```

```r
require(randomForest)
```

```
## Loading required package: randomForest
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

```r
# require(rpart.plot)
# require(rattle)
# install.packages(pkgs = "caret", dependencies = c("Depends", "Imports"))
# require(caret)
require(xgboost)
```

```
## Loading required package: xgboost
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.2.3

## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.2.3

##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:PerformanceAnalytics':
##
##     textplot
##
## The following object is masked from 'package:stats':
##
##     lowess
```

```
# source("E:/research/Projects/sduprey/NAR-271/RCode/RP_Plotting_Utils.R")
# source("E:/research/Projects/sduprey/NAR-271/RCode/RP_Macro_Monthly_Utils.R")
# source("E:/research/Projects/sduprey/NAR-271/RCode/RP_Spread_Utils.R")

user = 'sduprey'
# JIRA Code (e.g. NAR-#)
JIRACode = 'NAR-271'
repoPath = RP_GetSharedPath(user)
# Input Data Path
inputDataPath  = paste(repoPath,'InputData/', user,'/',JIRACode,'/',sep="")
# Output Data Path
outputDataPath = paste(repoPath,'OutputData/', user,'/',JIRACode,'/',sep="")

# outputDataPathMonth <- paste(outputDataPath,"Month_2007/01_11_2015/",sep="")
# outputDataPathStrategyMonth <- paste(outputDataPath,"Month_2007/01_11_2015/",sep="")
outputDataPathMonth <- paste(outputDataPath,"Month_2007/",sep="")
outputDataPathStrategyMonth <- paste(outputDataPath,"Month_2007/",sep="")

results <- readRDS(paste(outputDataPathStrategyMonth,"all_pairs_results_month_2007.rds", sep = ""))
```

Individual pair US/Germany trading strategy

```
### Deutschland visualization
my_result_spread_name <- "US_DE"
my_pair <- c("US","DE")
my_pair[1]<-"US"
my_pair[2]<-"DE"
# Long/short position returns distribution
long_US <- results$DE_FIRST_WEIGHT>=0
long_DE <- results$DE_SECOND_WEIGHT>=0
LONG_STRATEGY_RETURN_US <- results$DE_FIRST_WEIGHT[long_US]*results$DE_FIRST_BOND_NEXT_RETURN[long_US]
LONG_STRATEGY_RETURN_DE <- results$DE_SECOND_WEIGHT[long_DE]*results$DE_SECOND_BOND_NEXT_OPEN_RETURN[lo


print("Mean return for long position both US and DE")
```

```
## [1] "Mean return for long position both US and DE"
```

```
mean((LONG_STRATEGY_RETURN_US+LONG_STRATEGY_RETURN_DE))
```

```
## Warning in LONG_STRATEGY_RETURN_US + LONG_STRATEGY_RETURN_DE: longer object
## length is not a multiple of shorter object length
```

```
## [1] 0.00683276
```

```
print("Mean return for long position for US only")
```

```
## [1] "Mean return for long position for US only"
```

```r
mean(LONG_STRATEGY_RETURN_US)
```

```
## [1] 0.0008238065
```

```r
print("Mean return for long position for DE only")
```

```
## [1] "Mean return for long position for DE only"
```

```r
mean(LONG_STRATEGY_RETURN_DE)
```

```
## [1] 0.005327057
```

```r
# Short position returns distribution
short_US <- results$DE_FIRST_WEIGHT<=0
short_DE <- results$DE_SECOND_WEIGHT<=0
SHORT_STRATEGY_RETURN_US <- results$DE_FIRST_WEIGHT[short_US]*results$DE_FIRST_BOND_NEXT_RETURN[short_US
SHORT_STRATEGY_RETURN_DE <- results$DE_SECOND_WEIGHT[short_DE]*results$DE_SECOND_BOND_NEXT_OPEN_RETURN[s
print("Mean return for short position both US and DE")
```

```
## [1] "Mean return for short position both US and DE"
```

```r
mean((SHORT_STRATEGY_RETURN_US+SHORT_STRATEGY_RETURN_DE))
```

```
## Warning in SHORT_STRATEGY_RETURN_US + SHORT_STRATEGY_RETURN_DE: longer
## object length is not a multiple of shorter object length
```

```
## [1] -0.001629671
```

```r
print("Mean return for long position for US only")
```

```
## [1] "Mean return for long position for US only"
```

```r
mean(SHORT_STRATEGY_RETURN_US)
```

```
## [1] -0.001982707
```

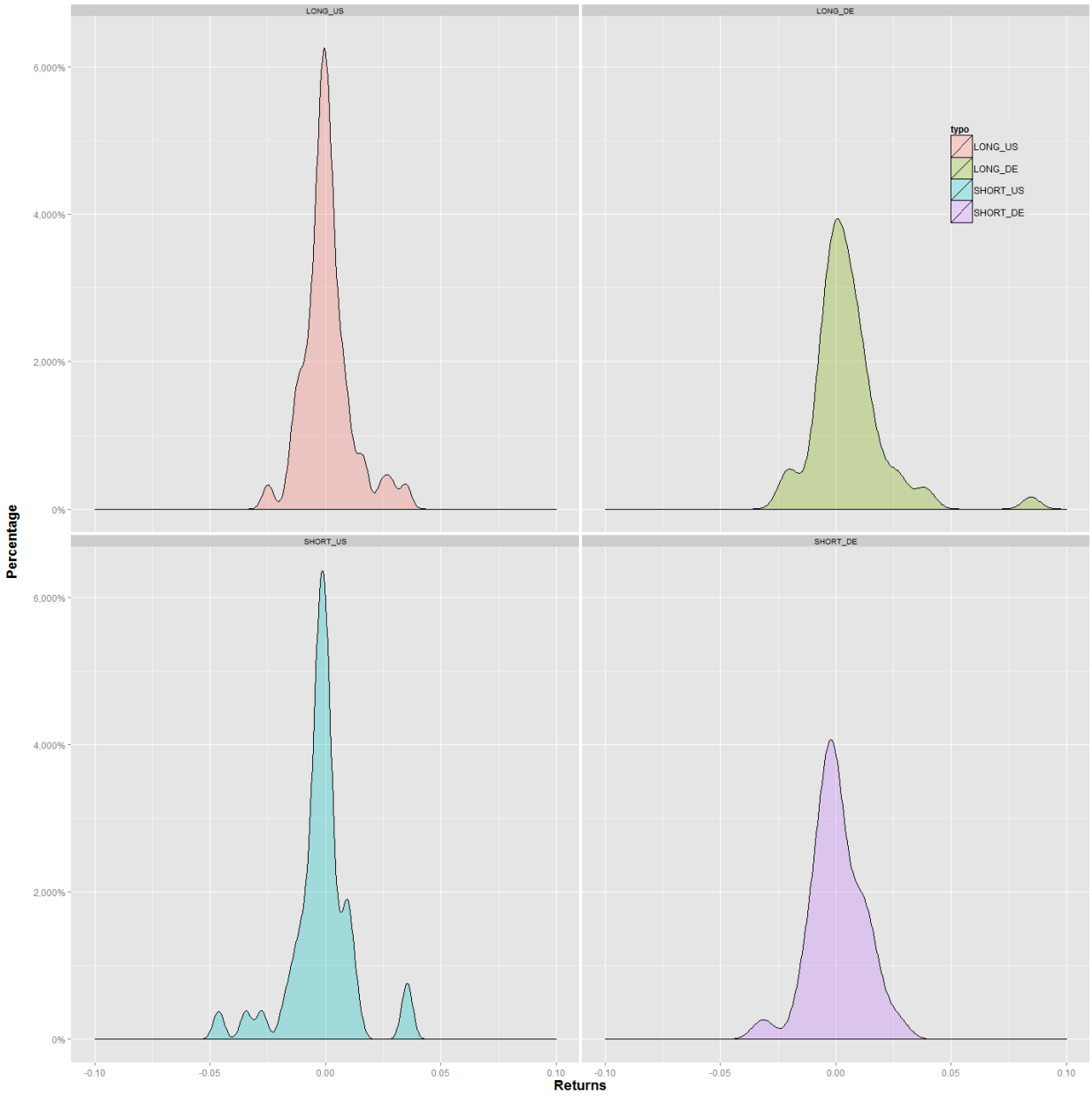
```r
print("Mean return for long position for DE only")
```

```
## [1] "Mean return for long position for DE only"
```

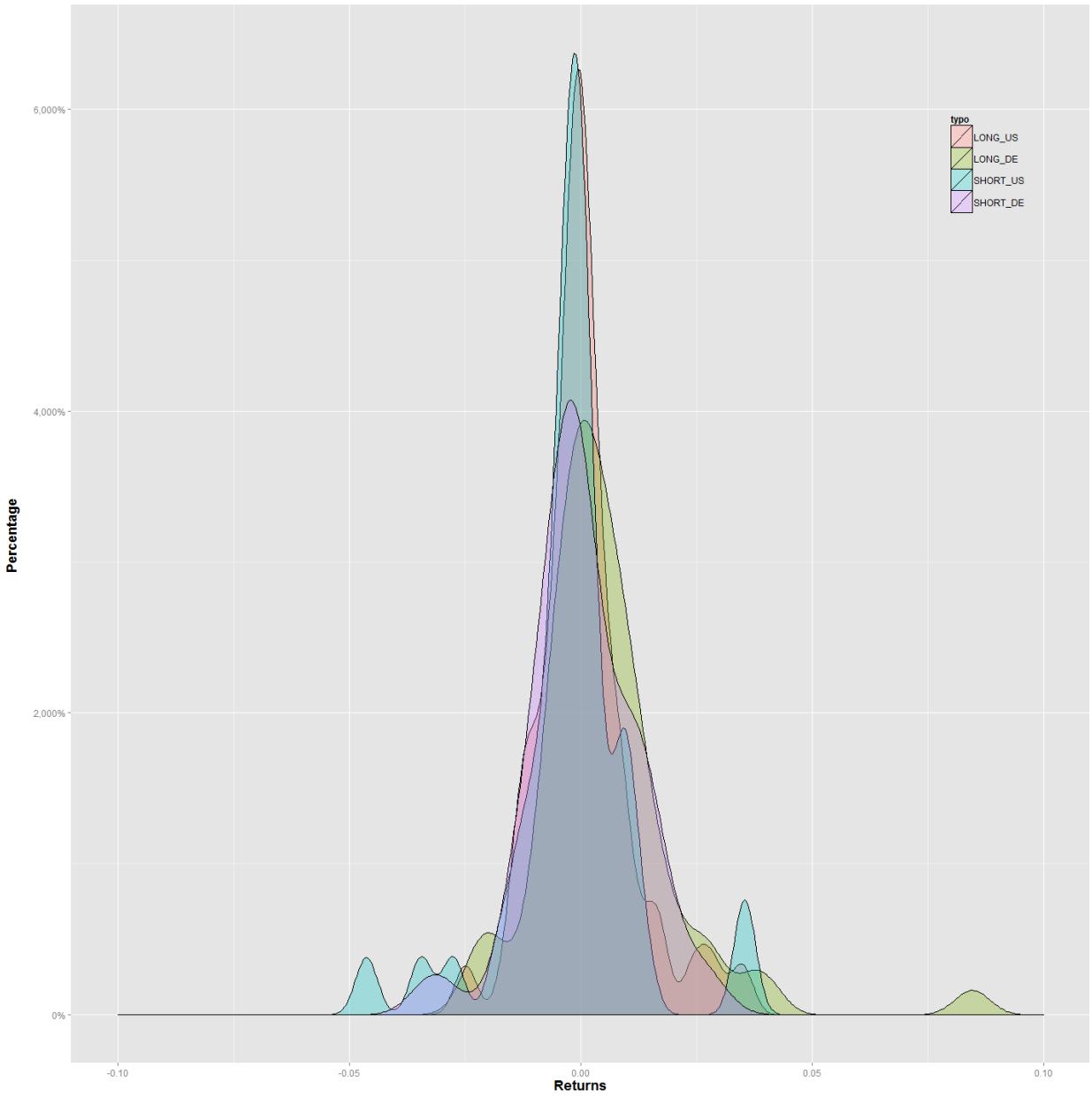
```r
mean(SHORT_STRATEGY_RETURN_DE)
```

```
## [1] 0.0005104123
```

```
### Nice visualization of superposed densities
return_decomposition_df <- data.frame(returns=LONG_STRATEGY_RETURN_US,typo="LONG_US")
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=LONG_STRATEGY_RETURN_DE,ty
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=SHORT_STRATEGY_RETURN_US,t
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=SHORT_STRATEGY_RETURN_DE,t

my_title <-"Distribution of returns per trade type"
my_xaxis_title <- paste("Returns")
my_yaxis_title <- paste("Percentage","\n")

g <- ggplot(return_decomposition_df, aes(x=returns, fill=typo)) +geom_density(alpha=.3)+
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-0.1,0.1))+facet_wrap(~typo
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```




```
g <- ggplot(return_decomposition_df, aes(x=returns, fill=typo)) +geom_density(alpha=.3)+
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-0.1,0.1))
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```




```
print("Outputing statistical results for our individual trading strategy US_DE")
```

```
## [1] "Outputing statistical results for our individual trading strategy US_DE"
```

```
WeightMatrix <- results[,c("DATES","DE_FIRST_WEIGHT","DE_SECOND_WEIGHT")]
colnames(WeightMatrix) <- c("DATE","DE_FIRST_WEIGHT","DE_SECOND_WEIGHT")
ReturnSerie <- results[,c("DE_STRATEGY_RETURN")]
turnover <- RP_GetTurnOver(WeightMatrix,-1)
RP_ReturnStats(ReturnSerie,21,TRUE,WeightMatrix,0.05,F)
```

```
## Warning in if (is.na(WeightMatrix)) {: the condition has length > 1 and
## only the first element will be used
```

```
## ************************************************************
## **********       STRATEGY STATISTICS           **********
## ************************************************************
## *****                 NO FEE                             *****
## ****                                                     ****
##   Annualized Return:      0.0376075073506816
##   Annualized Volatility:  0.0582558020743051
##   Information Ratio:     0.645558142049324
##   Hit Ratio:             0.526315789473684
##   W/L Ratio:             1.42449866331473
##   Turnover:              0.563689662853424
##   P-Value:               0.0724995334285696
##   Max Drawdown:          -0.0714420110363596
##   Drawdown Recovery:     5
##   Break-Even Fee (bps):  27.8422253966276
## ****                                                     ****
## ************************************************************
## *****                 WITH FEE                           *****
## ****                                                     ****
##   Annualized Return:      0.0308606737829419
##   Annualized Volatility:  0.0583045697815325
##   Information Ratio:     0.529301114793865
##   Hit Ratio:             0.515789473684211
##   W/L Ratio:             1.36626893887106
##   Turnover:              0.563689662853424
##   P-Value:               0.139763373803876
##   Max Drawdown:          -0.0855281748386419
##   Drawdown Recovery:     6
## ****                                                     ****
## ************************************************************

## $STATS
##   Annualized_Return Annualized_Return_Fee Annualized_Volatility
## 1        0.03760751            0.03086067             0.0582558
##   Annualized_Volatility_Fee        IR    IR_Fee  TurnOver  HitRatio
## 1                0.05830457 0.6455581 0.5293011 0.5636897 0.5263158
##   HitRatio_Fee  WLRatio WLRatio_Fee       PVal  PVal_Fee BreakEvenFee
## 1    0.5157895 1.424499    1.366269 0.07249953 0.1397634     27.84223
##   maxDrawdown drawdownRecovery maxDrawdown_Fee drawdownRecovery_Fee
## 1 -0.07144201                5     -0.08552817                    6
##
## $RETURN
##  [1] -0.0147415415  0.0098944084  0.0003817756 -0.0003242333 -0.0130223102
##  [6]  0.0262203124 -0.0030818799 -0.0084807014  0.0337137763 -0.0046859600
## [11] -0.0243604897  0.0031168435 -0.0095099736  0.0158170757  0.0222803716
## [16]  0.0163894344  0.0165189266 -0.0222926028 -0.0019988156  0.0376778689
## [21]  0.0030905772 -0.0219164646  0.0033281989 -0.0064004883 -0.0225024727
## [26]  0.0211102290  0.0088306265  0.0114855314  0.0042702137 -0.0055053949
## [31] -0.0157740481  0.0103657675 -0.0199413544 -0.0025951627  0.0354449295
## [36] -0.0122336555 -0.0270514720  0.0062234699 -0.0005128009  0.0305983735
## [41] -0.0007745535 -0.0047841905 -0.0226219139 -0.0262728080  0.0002038848
## [46] -0.0135818102  0.0181240082  0.0227395695 -0.0129369214  0.0357414085
## [51]  0.0270815742  0.0266751692 -0.0005795302  0.0079775949  0.0337399251
## [56]  0.0091660653  0.0096653244 -0.0005134620  0.0252152657  0.0173045407
```

```
## [61]  0.0186240077 -0.0040935092 -0.0033848976 -0.0031378253  0.0135181131
## [66] -0.0089855775 -0.0032922076  0.0148665435  0.0150333073 -0.0216342855
## [71]  0.0162096172  0.0207475015 -0.0157453820  0.0100720922 -0.0041580464
## [76]  0.0270882779 -0.0349400644  0.0030648041  0.0074844023 -0.0109002445
## [81]  0.0148095810  0.0125378981  0.0043995127 -0.0031560042  0.0039022537
## [86] -0.0080657650  0.0250968331  0.0169794972 -0.0199903280  0.0126879284
## [91] -0.0211508766 -0.0212604740  0.0210863362 -0.0050156459 -0.0029673031
##
## $RETURN_FEE
##  [1] -1.524910e-02  9.222475e-03 -5.556449e-04 -1.325058e-03 -1.403593e-02
##  [6]  2.548946e-02 -3.461743e-03 -9.242577e-03  3.301796e-02 -5.000813e-03
## [11] -2.512928e-02  2.410691e-03 -1.033044e-02  1.483228e-02  2.203585e-02
## [16]  1.564949e-02  1.635035e-02 -2.305980e-02 -2.444886e-03  3.695534e-02
## [21]  2.563004e-03 -2.245740e-02  2.890592e-03 -7.407416e-03 -2.352575e-02
## [26]  2.037563e-02  8.334899e-03  1.095492e-02  3.950461e-03 -6.368165e-03
## [31] -1.672804e-02  9.740477e-03 -2.085908e-02 -2.929966e-03  3.502530e-02
## [36] -1.295434e-02 -2.778292e-02  5.505403e-03 -1.033228e-03  3.009389e-02
## [41] -1.525416e-03 -5.789491e-03 -2.287535e-02 -2.679227e-02  4.279504e-06
## [46] -1.366587e-02  1.782373e-02  2.224453e-02 -1.319449e-02  3.560568e-02
## [51]  2.684028e-02  2.627008e-02 -6.014799e-04  7.075735e-03  3.277263e-02
## [56]  8.528069e-03  8.922262e-03 -7.636217e-04  2.436420e-02  1.644672e-02
## [61]  1.787779e-02 -4.634254e-03 -3.514752e-03 -3.890466e-03  1.277791e-02
## [66] -9.155670e-03 -3.695341e-03  1.449518e-02  1.429422e-02 -2.240098e-02
## [71]  1.547140e-02  2.019470e-02 -1.582714e-02  9.081623e-03 -4.205080e-03
## [76]  2.635805e-02 -3.585794e-02  2.656363e-03  7.295407e-03 -1.097940e-02
## [81]  1.422962e-02  1.204401e-02  3.646346e-03 -4.159669e-03  3.070156e-03
## [86] -8.655680e-03  2.445685e-02  1.694798e-02 -2.047484e-02  1.267423e-02
## [91] -2.155020e-02 -2.129790e-02  2.095794e-02 -5.318370e-03 -3.805535e-03
```

```
toplot_df <-  melt(results[,c("DATES", "DE_STRATEGY_TODAY","DE_FIRST_BOND","DE_SECOND_BOND")],"DATES")
my_title <- paste("Macro Sentiment ML over Ravenpack ESS metrics and taxonomy ",my_pair[1],"/",my_pair[
g<-ggplot(
  toplot_df,aes(
    x = DATES,y = value,group = variable,color = variable
  )
) +
  geom_line() +
  scale_x_date() +
  ggtitle(my_title) + xlab("Time") + ylab("Cumulated sentiment") +
  theme(title = element_text(size = 12, face = 'bold')) +
  theme(legend.position = c(0.2,0.8), legend.box = "vertical") +
  theme(legend.background = element_rect(fill = "gray90")) +
  theme(legend.key.size = unit(0.7, "cm"))
print(g)
```

Individual pair US/UK trading strategy

```
### Great Britain visualization
my_result_spread_name <- "US_GB"
my_pair[1]<-"US"
my_pair[2]<-"GB"
# Long/short position returns distribution
long_US <- results$GB_FIRST_WEIGHT>=0
long_GB <- results$GB_SECOND_WEIGHT>=0
LONG_STRATEGY_RETURN_US <- results$GB_FIRST_WEIGHT[long_US]*results$GB_FIRST_BOND_NEXT_RETURN[long_US]
LONG_STRATEGY_RETURN_GB <- results$GB_SECOND_WEIGHT[long_GB]*results$GB_SECOND_BOND_NEXT_OPEN_RETURN[lo

print("Mean return for long position both US and GB")
```

```
## [1] "Mean return for long position both US and GB"
```

```r
mean((LONG_STRATEGY_RETURN_US+LONG_STRATEGY_RETURN_GB))
```

```
## Warning in LONG_STRATEGY_RETURN_US + LONG_STRATEGY_RETURN_GB: longer object
## length is not a multiple of shorter object length
```

```
## [1] 0.01207859
```

```r
print("Mean return for long position for US only")
```

```
## [1] "Mean return for long position for US only"
```

```r
mean(LONG_STRATEGY_RETURN_US)
```

```
## [1] 0.003974042
```

```r
print("Mean return for long position for GB only")
```

```
## [1] "Mean return for long position for GB only"
```

```r
mean(LONG_STRATEGY_RETURN_GB)
```

```
## [1] 0.00557202
```

```r
# Short position returns distribution
short_US <- results$GB_FIRST_WEIGHT<=0
short_GB <- results$GB_SECOND_WEIGHT<=0
SHORT_STRATEGY_RETURN_US <- results$GB_FIRST_WEIGHT[short_US]*results$GB_FIRST_BOND_NEXT_RETURN[short_US
SHORT_STRATEGY_RETURN_GB <- results$GB_SECOND_WEIGHT[short_GB]*results$GB_SECOND_BOND_NEXT_OPEN_RETURN[s
print("Mean return for short position both US and GB")
```

```
## [1] "Mean return for short position both US and GB"
```

```r
mean((SHORT_STRATEGY_RETURN_US+SHORT_STRATEGY_RETURN_GB))
```

```
## Warning in SHORT_STRATEGY_RETURN_US + SHORT_STRATEGY_RETURN_GB: longer
## object length is not a multiple of shorter object length
```

```
## [1] -0.004035019
```

```r
print("Mean return for long position for US only")
```

```
## [1] "Mean return for long position for US only"
```

```r
mean(SHORT_STRATEGY_RETURN_US)
```

```
## [1] -0.001678582
```

```r
print("Mean return for long position for GB only")
```

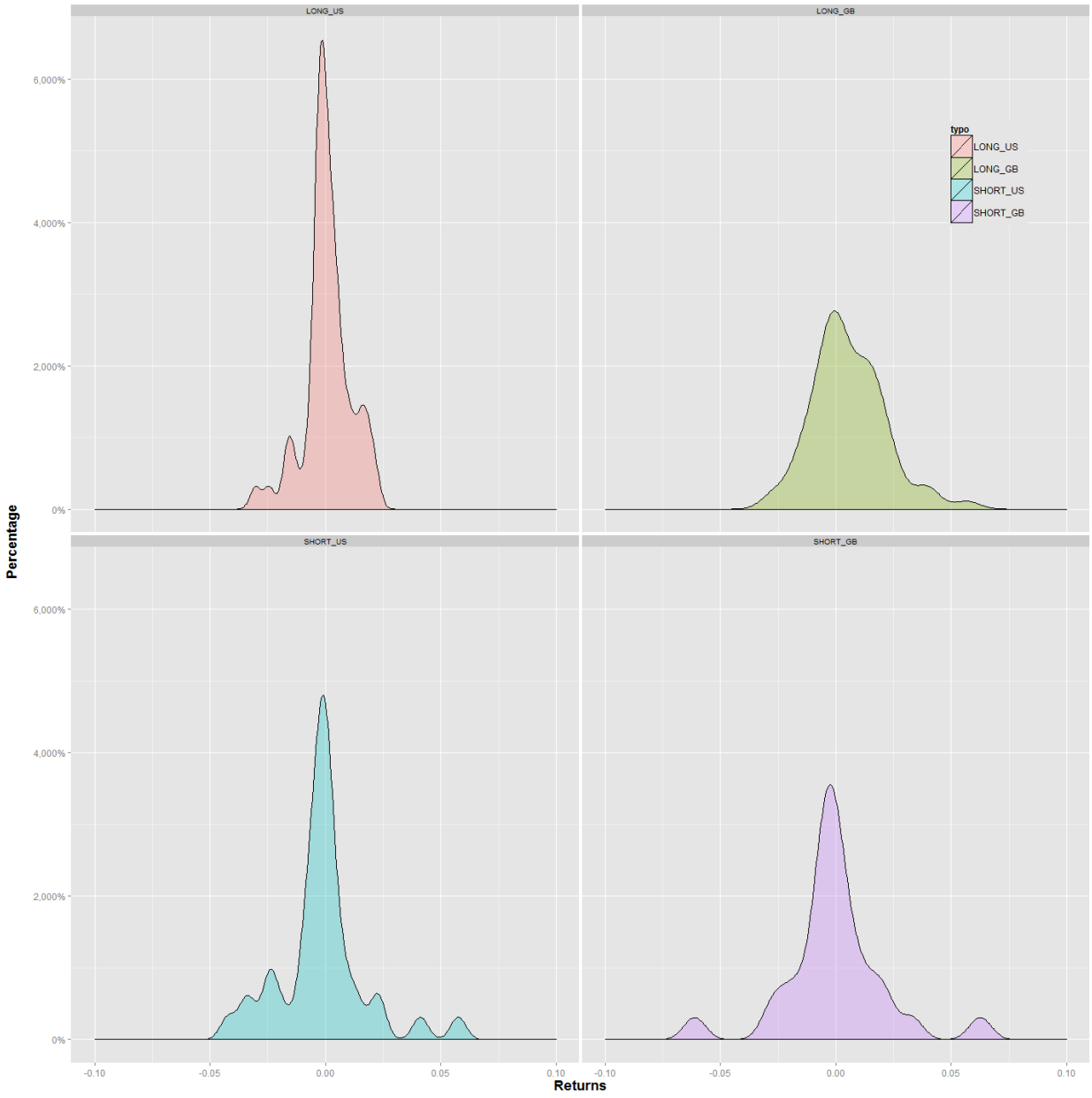
```
## [1] "Mean return for long position for GB only"
```

```r
mean(SHORT_STRATEGY_RETURN_GB)
```

```
## [1] -0.0005464763
```

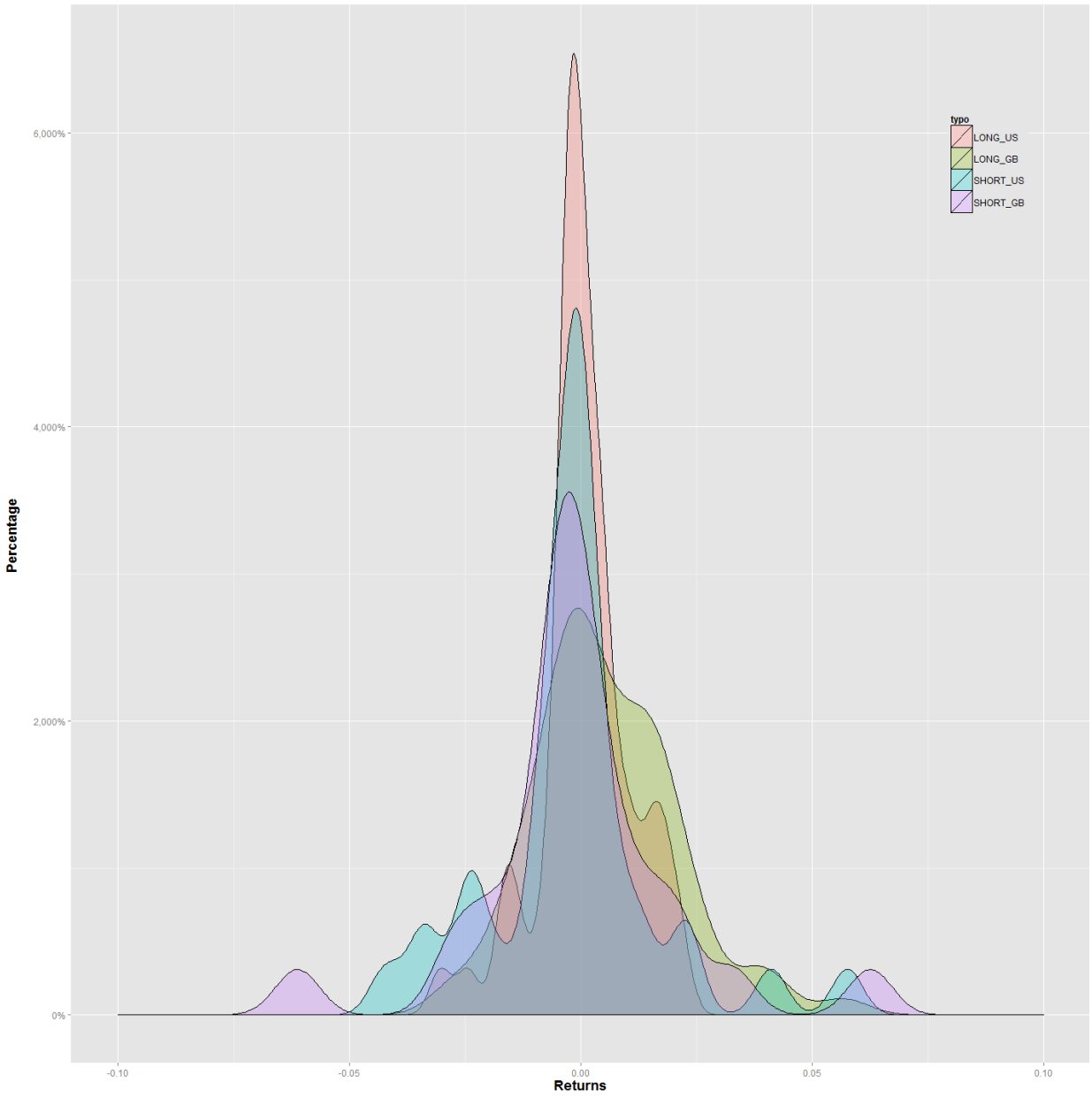
```r
### Nice visualization of superposed densities
return_decomposition_df <- data.frame(returns=LONG_STRATEGY_RETURN_US,typo="LONG_US")
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=LONG_STRATEGY_RETURN_GB,ty
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=SHORT_STRATEGY_RETURN_US,t
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=SHORT_STRATEGY_RETURN_GB,t

my_title <-"Distribution of returns per trade type"
my_xaxis_title <- paste("Returns")
my_yaxis_title <- paste("Percentage","\n")

g <- ggplot(return_decomposition_df, aes(x=returns, fill=typo)) +geom_density(alpha=.3)+
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-0.1,0.1))+facet_wrap(~typo
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```




```
g <- ggplot(return_decomposition_df, aes(x=returns, fill=typo)) +geom_density(alpha=.3)+
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-0.1,0.1))
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```




```
print("Outputing statistical results for our individual trading strategy US_GB")
```

```
## [1] "Outputing statistical results for our individual trading strategy US_GB"
```

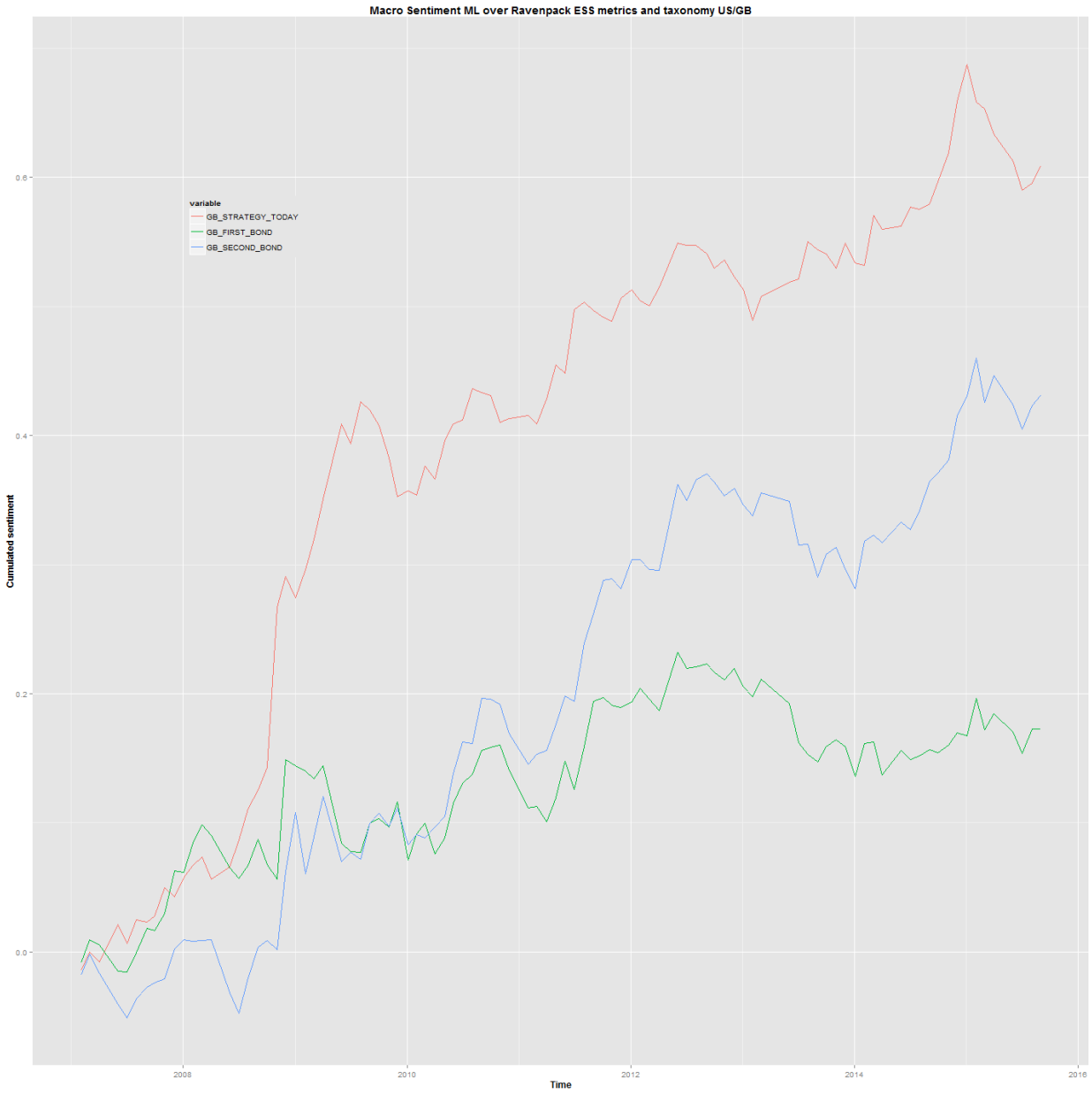
```
WeightMatrix <- results[,c("DATES","GB_FIRST_WEIGHT","GB_SECOND_WEIGHT")]
colnames(WeightMatrix) <- c("DATE","GB_FIRST_WEIGHT","GB_SECOND_WEIGHT")
ReturnSerie <- results[,c("GB_STRATEGY_RETURN")]
turnover <- RP_GetTurnOver(WeightMatrix,-1)
RP_ReturnStats(ReturnSerie,21,TRUE,WeightMatrix,0.05,F)
```

```
## Warning in if (is.na(WeightMatrix)) {: the condition has length > 1 and
## only the first element will be used
```

```
## ************************************************************
## **********       STRATEGY STATISTICS           **********
## ************************************************************
## *****                NO FEE                             *****
## ****                                                    ****
##   Annualized Return:      0.063850214335324
##   Annualized Volatility:  0.0709319349945313
##   Information Ratio:     0.900161744357415
##   Hit Ratio:             0.536842105263158
##   W/L Ratio:             1.77556898598927
##   Turnover:              0.540070451839978
##   P-Value:               0.0129739348046073
##   Max Drawdown:          -0.0943845111427219
##   Drawdown Recovery:     3
##   Break-Even Fee (bps):  49.3920010382621
## ****                                                    ****
## ************************************************************
## *****                WITH FEE                           *****
## ****                                                    ****
##   Annualized Return:      0.0573975464822265
##   Annualized Volatility:  0.0710178955504144
##   Information Ratio:     0.808212437687356
##   Hit Ratio:             0.536842105263158
##   W/L Ratio:             1.64328334599237
##   Turnover:              0.540070451839978
##   P-Value:               0.025239075826137
##   Max Drawdown:          -0.0967948811177515
##   Drawdown Recovery:     3
## ****                                                    ****
## ************************************************************

## $STATS
##   Annualized_Return Annualized_Return_Fee Annualized_Volatility
## 1        0.06385021            0.05739755            0.07093193
##   Annualized_Volatility_Fee        IR    IR_Fee  TurnOver  HitRatio
## 1                 0.0710179 0.9001617 0.8082124 0.5400705 0.5368421
##   HitRatio_Fee  WLRatio WLRatio_Fee       PVal   PVal_Fee BreakEvenFee
## 1    0.5368421 1.775569    1.643283 0.01297393 0.02523908       49.392
##   maxDrawdown drawdownRecovery maxDrawdown_Fee drawdownRecovery_Fee
## 1 -0.09438451                3     -0.09679488                    3
##
## $RETURN
##  [1] -0.0143943355  0.0140486505 -0.0078602108  0.0068235209 -0.0146493472
##  [6]  0.0181961308 -0.0020566097  0.0049652503  0.0217288301 -0.0069755063
## [11]  0.0144598046  0.0094331819  0.0063669465 -0.0172247652  0.0035804137
## [16]  0.0206474712  0.0241576848  0.0142580215  0.0175903154  0.1168042874
## [21]  0.0236454296 -0.0168139194  0.0219754262  0.0227199156  0.0312180688
## [26]  0.0175546266 -0.0155132482  0.0322613904 -0.0063635265 -0.0124418308
## [31] -0.0247680184 -0.0308359256  0.0043018405 -0.0033860024  0.0223793605
## [36] -0.0101146376  0.0291615696  0.0130047373  0.0029314152  0.0242141433
## [41] -0.0032937563 -0.0024127039 -0.0208766737  0.0032598237  0.0107792409
## [46] -0.0065352324  0.0195947225  0.0256440544 -0.0063287967  0.0479876195
## [51]  0.0058750045 -0.0068577960 -0.0048651314 -0.0030068192  0.0173202756
## [56]  0.0064418489 -0.0079062236 -0.0041992614  0.0140346453 -0.0127770707
```

```
## [61] -0.0015599247 -0.0003023358 -0.0065132701 -0.0113118386  0.0065298237
## [66] -0.0131918533 -0.0099042831 -0.0241532435  0.0179294983  0.0317916442
## [71]  0.0021374171  0.0290605906 -0.0067346787 -0.0030739111 -0.0113466916
## [76]  0.0192728992 -0.0153452134 -0.0015127132  0.0376815186 -0.0104618692
## [81]  0.0057924926  0.0145710349 -0.0018693113  0.0041737541  0.0173144127
## [86]  0.0219031513  0.0390714082  0.0281924666 -0.0291640348 -0.0055289088
## [91] -0.0195406309 -0.0210449661 -0.0227324524  0.0053592332  0.0130833290
##
## $RETURN_FEE
##  [1] -0.0149017134  0.0137885828 -0.0088039961  0.0059553190 -0.0152824698
##  [6]  0.0173194766 -0.0023715331  0.0039697078  0.0216991030 -0.0079138733
## [11]  0.0134736743  0.0087725563  0.0053727992 -0.0181649588  0.0032482166
## [16]  0.0200911558  0.0240727768  0.0139175951  0.0172287837  0.1159141321
## [21]  0.0230138712 -0.0171316770  0.0213609131  0.0223282765  0.0305716839
## [26]  0.0171731065 -0.0155689769  0.0321279365 -0.0071734772 -0.0134548633
## [31] -0.0255015100 -0.0315738831  0.0033682305 -0.0043898978  0.0215992955
## [36] -0.0104514163  0.0285138535  0.0124654867  0.0022499328  0.0239813992
## [41] -0.0039628462 -0.0030812044 -0.0211987363  0.0024758516  0.0107218695
## [46] -0.0071624561  0.0194775402  0.0254906713 -0.0068783103  0.0478381948
## [51]  0.0056994484 -0.0075292757 -0.0057273056 -0.0033790771  0.0169939010
## [56]  0.0061777255 -0.0079322883 -0.0045756339  0.0140299262 -0.0135688104
## [61] -0.0017654158 -0.0005075686 -0.0070580633 -0.0117270942  0.0062839517
## [66] -0.0132962936 -0.0100368953 -0.0249748021  0.0169467853  0.0308224663
## [71]  0.0018047401  0.0287973028 -0.0077419432 -0.0030739111 -0.0120211932
## [76]  0.0183448572 -0.0163611929 -0.0017943420  0.0374190445 -0.0107372906
## [81]  0.0049830650  0.0135977268 -0.0028496701  0.0031774230  0.0167974741
## [86]  0.0215262147  0.0387008890  0.0276568989 -0.0291886732 -0.0062163718
## [91] -0.0202377976 -0.0212300100 -0.0237559688  0.0043640831  0.0129460148
```

```
toplot_df <-  melt(results[,c("DATES", "GB_STRATEGY_TODAY","GB_FIRST_BOND","GB_SECOND_BOND")],"DATES")

my_title <- paste("Macro Sentiment ML over Ravenpack ESS metrics and taxonomy ",my_pair[1],"/",my_pair[
g<-ggplot(
  toplot_df,aes(
    x = DATES,y = value,group = variable,color = variable
  )
) +
  geom_line() +
  scale_x_date() +
  ggtitle(my_title) + xlab("Time") + ylab("Cumulated sentiment") +
  theme(title = element_text(size = 12, face = 'bold')) +
  theme(legend.position = c(0.2,0.8), legend.box = "vertical") +
  theme(legend.background = element_rect(fill = "gray90")) +
  theme(legend.key.size = unit(0.7, "cm"))
print(g)
```



Individual pair US/Japan trading strategy

```
### Japan visualization
my_result_spread_name <- "US_JP"
my_pair[1]<-"US"
my_pair[2]<-"JP"

# Long/short position returns distribution
long_US <- results$JP_FIRST_WEIGHT>=0
long_JP <- results$JP_SECOND_WEIGHT>=0
LONG_STRATEGY_RETURN_US <- results$JP_FIRST_WEIGHT[long_US]*results$JP_FIRST_BOND_NEXT_RETURN[long_US]
LONG_STRATEGY_RETURN_JP <- results$JP_SECOND_WEIGHT[long_JP]*results$JP_SECOND_BOND_NEXT_OPEN_RETURN[lo

print("Mean return for long position both US and JP")
```

```
## [1] "Mean return for long position both US and JP"
```

```
mean((LONG_STRATEGY_RETURN_US+LONG_STRATEGY_RETURN_JP))
```

```
## Warning in LONG_STRATEGY_RETURN_US + LONG_STRATEGY_RETURN_JP: longer object
## length is not a multiple of shorter object length
```

```
## [1] 0.008098824
```

```
print("Mean return for long position for US only")
```

```
## [1] "Mean return for long position for US only"
```

```
mean(LONG_STRATEGY_RETURN_US)
```

```
## [1] 0.003310682
```

```
print("Mean return for long position for JP only")
```

```
## [1] "Mean return for long position for JP only"
```

```
mean(LONG_STRATEGY_RETURN_JP)
```

```
## [1] 0.004674318
```

```
# Short position returns distribution
short_US <- results$JP_FIRST_WEIGHT<=0
short_JP <- results$JP_SECOND_WEIGHT<=0
SHORT_STRATEGY_RETURN_US <- results$JP_FIRST_WEIGHT[short_US]*results$JP_FIRST_BOND_NEXT_RETURN[short_US
SHORT_STRATEGY_RETURN_JP <- results$JP_SECOND_WEIGHT[short_JP]*results$JP_SECOND_BOND_NEXT_OPEN_RETURN[s
print("Mean return for short position both US and JP")
```

```
## [1] "Mean return for short position both US and JP"
```

```
mean((SHORT_STRATEGY_RETURN_US+SHORT_STRATEGY_RETURN_JP))
```

```
## Warning in SHORT_STRATEGY_RETURN_US + SHORT_STRATEGY_RETURN_JP: longer
## object length is not a multiple of shorter object length
```

```
## [1] 0.01194968
```

```
print("Mean return for long position for US only")
```

```
## [1] "Mean return for long position for US only"
```

```r
mean(SHORT_STRATEGY_RETURN_US)
```

```
## [1] 0.001151783
```

```r
print("Mean return for long position for JP only")
```

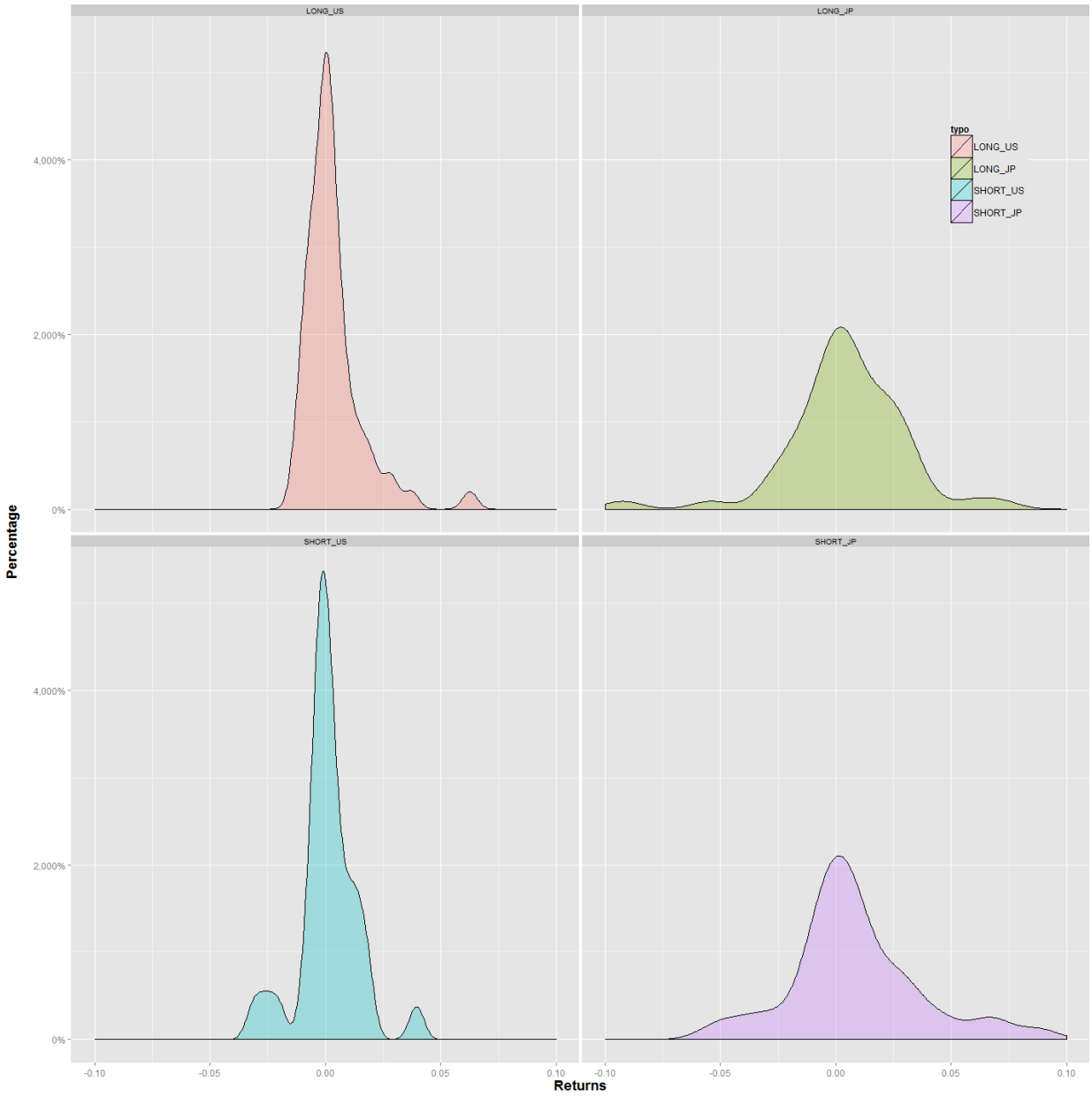
```
## [1] "Mean return for long position for JP only"
```

```r
mean(SHORT_STRATEGY_RETURN_JP)
```

```
## [1] 0.01142507
```

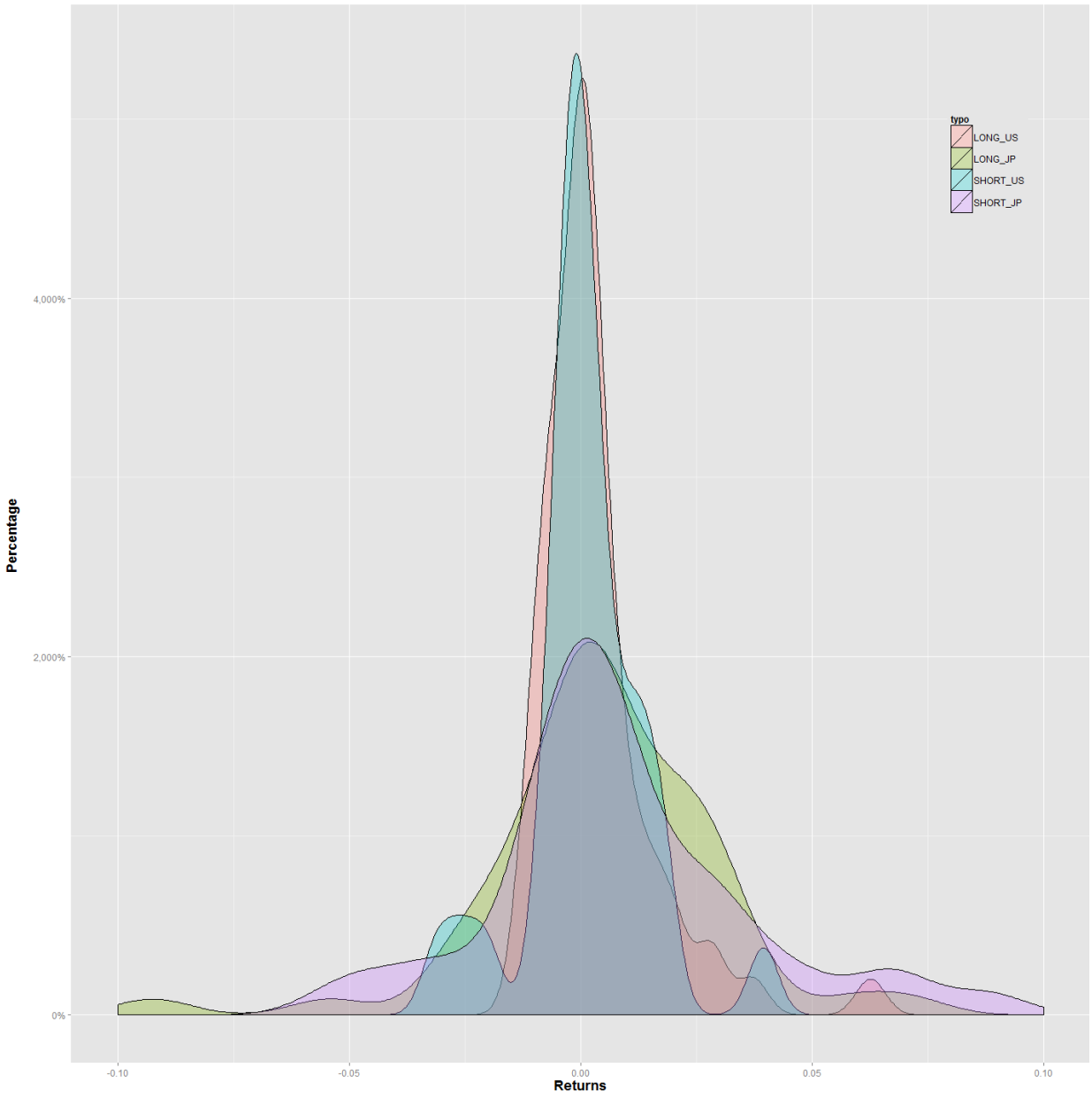
```r
### Nice visualization of superposed densities
return_decomposition_df <- data.frame(returns=LONG_STRATEGY_RETURN_US,typo="LONG_US")
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=LONG_STRATEGY_RETURN_JP,ty
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=SHORT_STRATEGY_RETURN_US,t
return_decomposition_df <- rbind(return_decomposition_df, data.frame(returns=SHORT_STRATEGY_RETURN_JP,t

my_title <-"Distribution of returns per trade type"
my_xaxis_title <- paste("Returns")
my_yaxis_title <- paste("Percentage","\n")

g <- ggplot(return_decomposition_df, aes(x=returns, fill=typo)) +geom_density(alpha=.3)+
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-0.1,0.1))+facet_wrap(~typo
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```




```
g <- ggplot(return_decomposition_df, aes(x=returns, fill=typo)) +geom_density(alpha=.3)+
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-0.1,0.1))
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```




```
print("Outputing statistical results for our individual trading strategy US_JP")
```

```
## [1] "Outputing statistical results for our individual trading strategy US_JP"
```

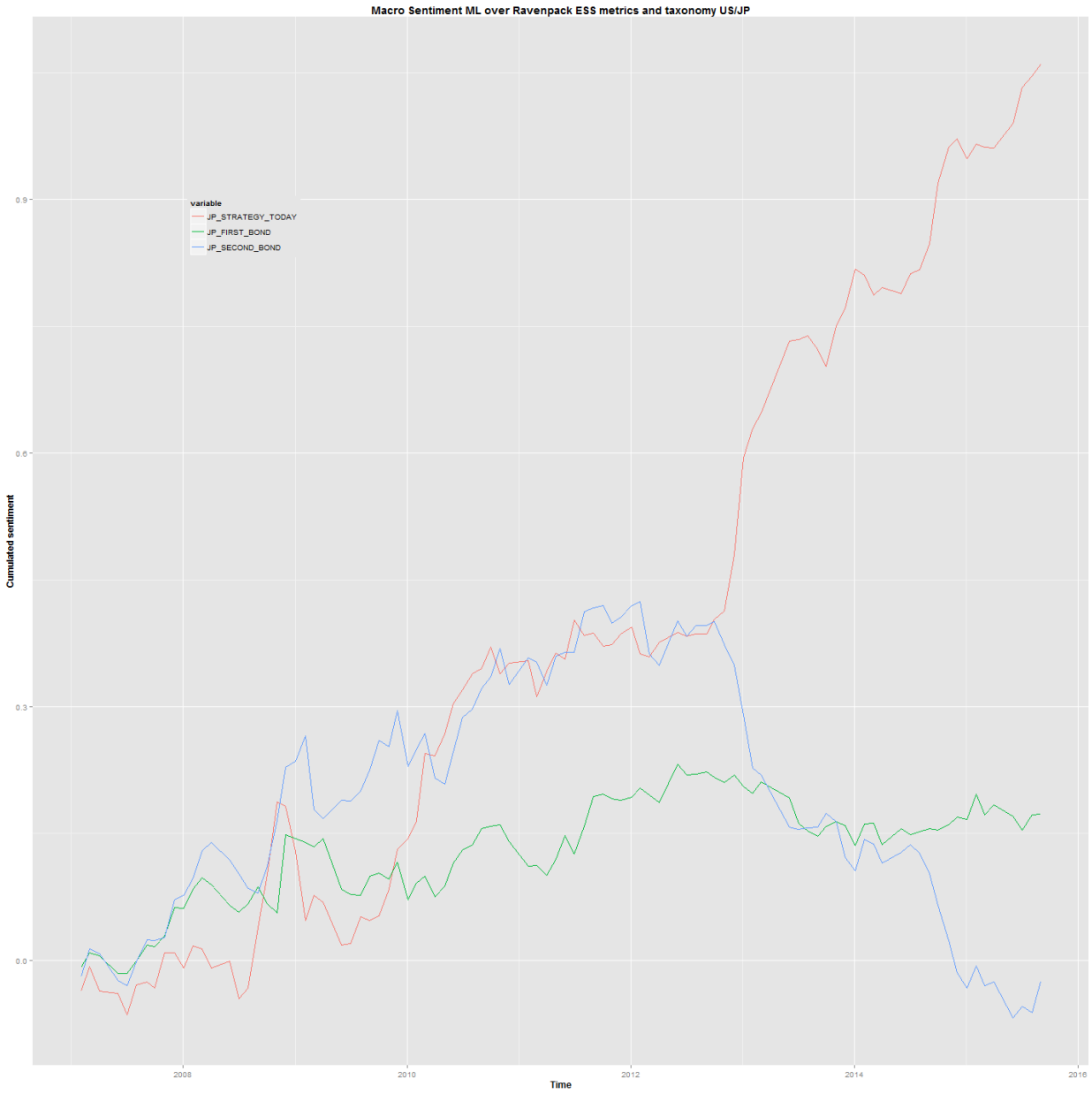
```
WeightMatrix <- results[,c("DATES","JP_FIRST_WEIGHT","JP_SECOND_WEIGHT")]
colnames(WeightMatrix) <- c("DATE","JP_FIRST_WEIGHT","JP_SECOND_WEIGHT")
ReturnSerie <- results[,c("JP_STRATEGY_RETURN")]
turnover <- RP_GetTurnOver(WeightMatrix,-1)
RP_ReturnStats(ReturnSerie,21,TRUE,WeightMatrix,0.05,F)
```

```
## Warning in if (is.na(WeightMatrix)) {: the condition has length > 1 and
## only the first element will be used
```

```
## ************************************************************
## **********       STRATEGY STATISTICS           **********
## ************************************************************
## *****                 NO FEE                             *****
## ****                                                    ****
##   Annualized Return:      0.114882853523995
##   Annualized Volatility:  0.104721540461109
##   Information Ratio:     1.09703173786543
##   Hit Ratio:             0.631578947368421
##   W/L Ratio:             1.40742956601709
##   Turnover:              0.490458149369551
##   P-Value:               0.00265980532210497
##   Max Drawdown:          -0.139656251691509
##   Drawdown Recovery:     11
##   Break-Even Fee (bps):  98.0669233819284
## ****                                                    ****
## ************************************************************
## *****                 WITH FEE                           *****
## ****                                                    ****
##   Annualized Return:      0.109053618673122
##   Annualized Volatility:  0.104735608036695
##   Information Ratio:     1.04122772300051
##   Hit Ratio:             0.631578947368421
##   W/L Ratio:             1.34587491838384
##   Turnover:              0.490458149369551
##   P-Value:               0.00425858465997352
##   Max Drawdown:          -0.142228735529995
##   Drawdown Recovery:     11
## ****                                                    ****
## ************************************************************

## $STATS
##   Annualized_Return Annualized_Return_Fee Annualized_Volatility
## 1         0.1148829             0.1090536             0.1047215
##   Annualized_Volatility_Fee       IR   IR_Fee  TurnOver  HitRatio
## 1                 0.1047356 1.097032 1.041228 0.4904581 0.6315789
##   HitRatio_Fee WLRatio WLRatio_Fee        PVal    PVal_Fee BreakEvenFee
## 1    0.6315789 1.40743    1.345875 0.002659805 0.004258585     98.06692
##   maxDrawdown drawdownRecovery maxDrawdown_Fee drawdownRecovery_Fee
## 1  -0.1396563               11      -0.1422287                   11
##
## $RETURN
##  [1] -3.686762e-02  2.888366e-02 -2.971248e-02  7.022668e-03 -2.557431e-02
##  [6]  3.488884e-02  3.674268e-03 -7.509945e-03  4.067472e-02 -7.435743e-05
## [11] -1.834110e-02  2.673699e-02 -4.208073e-03 -2.278333e-02  5.964611e-03
## [16] -4.465535e-02  1.211611e-02  6.926467e-02  6.093844e-02  8.212850e-02
## [21] -5.388129e-03 -5.468665e-02 -8.495474e-02  2.939115e-02 -8.117029e-03
## [26]  1.920339e-03  1.467937e-03  3.151046e-02 -5.363361e-03  6.224561e-03
## [31]  3.102396e-02  4.589596e-02  1.268789e-02  2.061457e-02  7.662860e-02
## [36] -2.634674e-03  2.624911e-02  3.450383e-02  1.596120e-02  1.933608e-02
## [41]  6.302224e-03  2.447606e-02 -3.159381e-02  1.208670e-02 -1.344319e-03
## [46] -4.280931e-02  2.936288e-02  2.122258e-02 -6.818524e-03  4.480623e-02
## [51] -1.828946e-02  2.681611e-03 -1.551329e-02  1.759670e-03  1.295970e-02
## [56]  7.707929e-03 -3.209309e-02 -3.770608e-03  1.743648e-02 -1.548602e-02
```

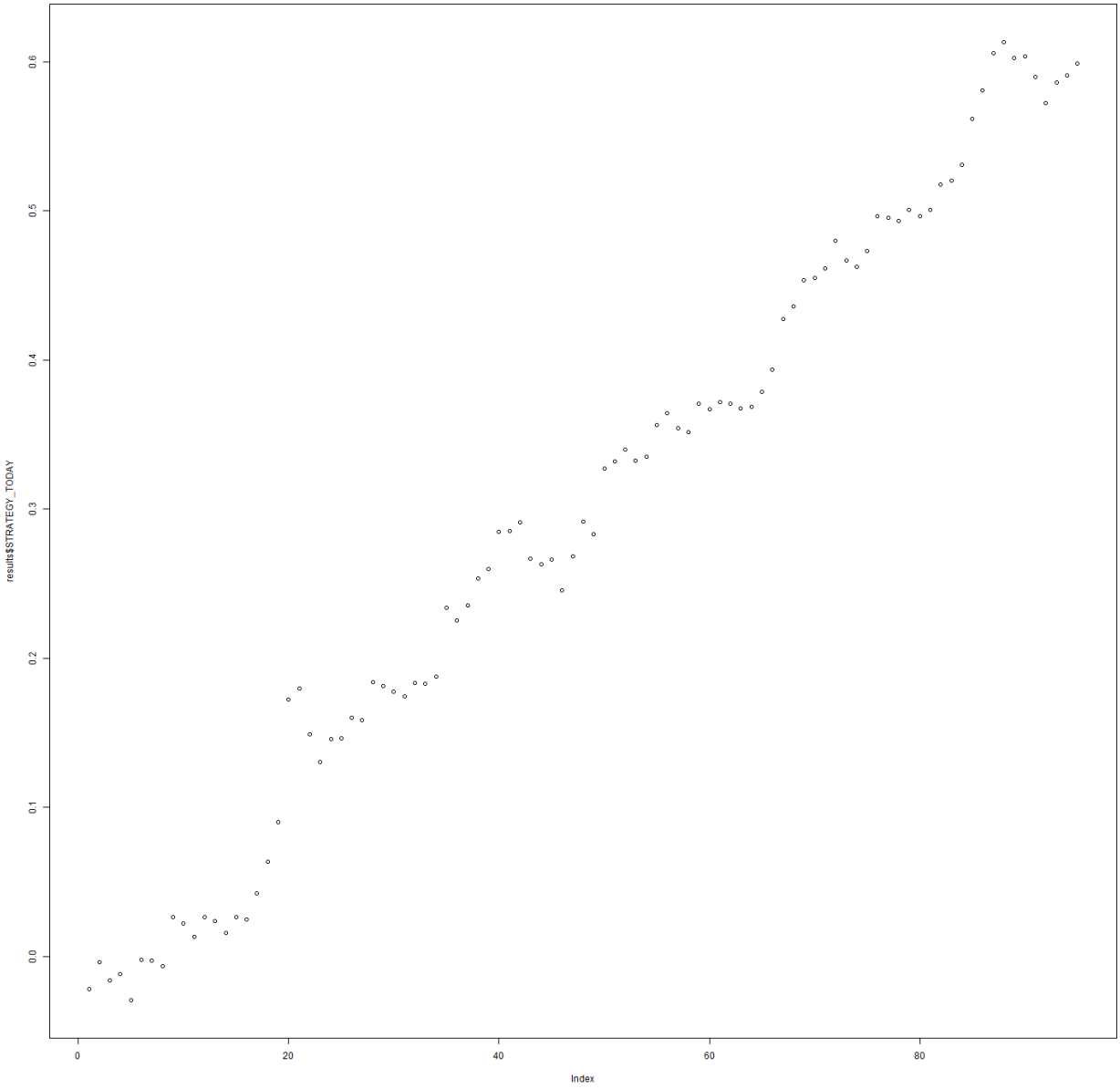
```
## [61] -3.782116e-03  1.942672e-03  1.349788e-04  1.720638e-02  9.931921e-03
## [66]  6.383025e-02  1.089873e-01  3.444364e-02  1.819703e-02 -5.769111e-03
## [71]  1.515723e-03  4.486890e-03 -1.662597e-02 -1.993546e-02  4.618041e-02
## [76]  2.222911e-02  4.453120e-02 -7.512173e-03 -2.346509e-02  9.071041e-03
## [81] -8.154362e-03  2.332726e-02  4.616968e-03  3.011477e-02  6.918661e-02
## [86]  4.211462e-02  9.179887e-03 -2.337121e-02  1.694828e-02 -3.736111e-03
## [91] -1.149941e-03 -1.168736e-02  4.221740e-02  1.334742e-02  1.301479e-02
##
## $RETURN_FEE
##  [1] -0.0373865310  0.0286267205 -0.0307431645  0.0060291731 -0.0260590465
##  [6]  0.0341104470  0.0026774387 -0.0080995121  0.0397141131 -0.0001411854
## [11] -0.0193601266  0.0257628982 -0.0045192531 -0.0229538134  0.0053016893
## [16] -0.0446553500  0.0114572516  0.0686424220  0.0605153029  0.0815045903
## [21] -0.0060691231 -0.0557434128 -0.0860440013  0.0284196465 -0.0087135466
## [26]  0.0015125586  0.0008425505  0.0310071179 -0.0063692441  0.0060805101
## [31]  0.0301946766  0.0449403601  0.0119278056  0.0204849740  0.0757019321
## [36] -0.0027814836  0.0253288338  0.0341996484  0.0158324724  0.0190845511
## [41]  0.0062641584  0.0242333620 -0.0317370386  0.0118997491 -0.0014678519
## [46] -0.0431572814  0.0289495222  0.0209113443 -0.0073489702  0.0448011859
## [51] -0.0186158463  0.0018056100 -0.0160912586  0.0017182992  0.0128554790
## [56]  0.0075266721 -0.0322952559 -0.0039818190  0.0174310303 -0.0157012761
## [61] -0.0041531689  0.0012940645  0.0001193068  0.0165830195  0.0092716253
## [66]  0.0635079121  0.1087115774  0.0341215404  0.0172145791 -0.0067754025
## [71]  0.0012210894  0.0042203313 -0.0170846247 -0.0208419849  0.0452250878
## [76]  0.0215884024  0.0438933668 -0.0075695045 -0.0237481268  0.0083412920
## [81] -0.0089709324  0.0230015604  0.0039721691  0.0297059125  0.0689367703
## [86]  0.0418817774  0.0086787773 -0.0238889055  0.0167872938 -0.0039354056
## [91] -0.0014837133 -0.0119563861  0.0416428086  0.0129505450  0.0126177892
```

```
toplot_df <-  melt(results[,c("DATES", "JP_STRATEGY_TODAY","JP_FIRST_BOND","JP_SECOND_BOND")],"DATES")

my_title <- paste("Macro Sentiment ML over Ravenpack ESS metrics and taxonomy ",my_pair[1],"/",my_pair[
g<-ggplot(
  toplot_df,aes(
    x = DATES,y = value,group = variable,color = variable
  )
) +
  geom_line() +
  scale_x_date() +
  ggtitle(my_title) + xlab("Time") + ylab("Cumulated sentiment") +
  theme(title = element_text(size = 12, face = 'bold')) +
  theme(legend.position = c(0.2,0.8), legend.box = "vertical") +
  theme(legend.background = element_rect(fill = "gray90")) +
  theme(legend.key.size = unit(0.7, "cm"))
print(g)
```



Equally weighted trading strategy plot

```
plot(results$STRATEGY_TODAY)
```



Equally weighted trading strategy metrics

```
print("Outputing statistical results for our equally weighted strategy")
```

```
## [1] "Outputing statistical results for our equally weighted strategy"
```

```
WeightMatrix <- results[,c("DATES","US_WEIGHT","DE_WEIGHT","JP_WEIGHT","GB_WEIGHT")]
colnames(WeightMatrix) <- c("DATE","US_WEIGHT","DE_WEIGHT","JP_WEIGHT","GB_WEIGHT")
ReturnSerie <- results[,c("STRATEGY_RETURN")]
turnover <- RP_GetTurnOver(WeightMatrix,-1)
RP_ReturnStats(ReturnSerie,21,TRUE,WeightMatrix,0.05,F)
```

```
## Warning in if (is.na(WeightMatrix)) {: the condition has length > 1 and
## only the first element will be used
```

```
## ************************************************************
## **********       STRATEGY STATISTICS           **********
## ************************************************************
## *****                NO FEE                            *****
## ****                                                  ****
##   Annualized Return:      0.073791949568223
##   Annualized Volatility:  0.0563971157998027
##   Information Ratio:     1.3084348112795
##   Hit Ratio:             0.621052631578947
##   W/L Ratio:             1.78201338802681
##   Turnover:              0.543292558707383
##   P-Value:               0.000386851381761672
##   Max Drawdown:          -0.0492227406580829
##   Drawdown Recovery:     5
##   Break-Even Fee (bps):  56.7675276537005
## ****                                                  ****
## ************************************************************
## *****                WITH FEE                          *****
## ****                                                  ****
##   Annualized Return:      0.0673098508982655
##   Annualized Volatility:  0.056428204113902
##   Information Ratio:     1.19284056537398
##   Hit Ratio:             0.610526315789474
##   W/L Ratio:             1.68619995447791
##   Turnover:              0.543292558707383
##   P-Value:               0.00114129791587543
##   Max Drawdown:          -0.0508020220834215
##   Drawdown Recovery:     12
## ****                                                  ****
## ************************************************************

## $STATS
##   Annualized_Return Annualized_Return_Fee Annualized_Volatility
## 1        0.07379195            0.06730985            0.05639712
##   Annualized_Volatility_Fee       IR   IR_Fee  TurnOver  HitRatio
## 1                 0.0564282 1.308435 1.192841 0.5432926 0.6210526
##   HitRatio_Fee  WLRatio WLRatio_Fee         PVal    PVal_Fee BreakEvenFee
## 1    0.6105263 1.782013      1.6862 0.0003868514 0.001141298     56.76753
##   maxDrawdown drawdownRecovery maxDrawdown_Fee drawdownRecovery_Fee
## 1 -0.04922274                5     -0.05080202                   12
##
## $RETURN
##  [1] -0.0219460399  0.0176421752 -0.0123165215  0.0045131533 -0.0177331437
##  [6]  0.0264583292 -0.0004836522 -0.0036563629  0.0320696956 -0.0039078211
## [11] -0.0092678937  0.0131456002 -0.0024285697 -0.0079179938  0.0106430607
## [16] -0.0020976632  0.0176099533  0.0211200647  0.0258570079  0.0793937057
## [21]  0.0071903931 -0.0309987406 -0.0188069931  0.0153572277  0.0004582643
## [26]  0.0135630792 -0.0016863516  0.0251319771 -0.0024740518 -0.0038778577
## [31] -0.0028720798  0.0089662849 -0.0008883985  0.0049399257  0.0450859355
## [36] -0.0083191725  0.0097848788  0.0179835029  0.0061518079  0.0247268355
## [41]  0.0007528944  0.0058480313 -0.0250197866 -0.0035079795  0.0032274647
## [46] -0.0208525253  0.0223729894  0.0232037503 -0.0086902316  0.0428585345
## [51]  0.0050607356  0.0075993884 -0.0069662954  0.0022535977  0.0213800616
## [56]  0.0077725672 -0.0099649645 -0.0028264223  0.0189064209 -0.0035420610
```

```
## [61]  0.0044782358 -0.0008146225 -0.0032507085  0.0009909278  0.0099973568
## [66]  0.0145144525  0.0334366981  0.0086813424  0.0170543039  0.0017145136
## [71]  0.0066439734  0.0181503377 -0.0130253435 -0.0042370520  0.0105545576
## [76]  0.0228686220 -0.0013432971 -0.0019773175  0.0075451382 -0.0040535643
## [81]  0.0041938212  0.0168230315  0.0023869096  0.0104796466  0.0305328345
## [86]  0.0188628025  0.0245239139  0.0075110160 -0.0105359754  0.0011746327
## [91] -0.0139059057 -0.0179876310  0.0138882557  0.0045919149  0.0077387249
##
## $RETURN_FEE
##  [1] -0.0224572648  0.0172241740 -0.0133174884  0.0035171604 -0.0185116895
##  [6]  0.0256320426 -0.0009449736 -0.0045498364  0.0314609901 -0.0043430474
## [11] -0.0101648326  0.0123472131 -0.0030323216 -0.0085596817  0.0102553319
## [16] -0.0028495207  0.0171411961  0.0205054428  0.0256324860  0.0785293063
## [21]  0.0064059071 -0.0318114732 -0.0196145155  0.0145269386 -0.0003844931
## [26]  0.0133865564 -0.0021310839  0.0247949648 -0.0033658561 -0.0044366790
## [31] -0.0033110552  0.0083025206 -0.0017842640  0.0043395153  0.0443801291
## [36] -0.0086644243  0.0089923545  0.0173303009  0.0058138278  0.0243950764
## [41]  0.0000270802  0.0051538647 -0.0251750093 -0.0039261083  0.0031473014
## [46] -0.0213304395  0.0218084933  0.0230388414 -0.0090509501  0.0427472235
## [51]  0.0047576188  0.0067887512 -0.0074667113  0.0018894960  0.0206643527
## [56]  0.0072458361 -0.0103437918 -0.0031771747  0.0185203503 -0.0042905715
## [61]  0.0040308401 -0.0011719584 -0.0036184189  0.0005611597  0.0094927374
## [66]  0.0143614388  0.0331635245  0.0083315188  0.0161362754  0.0007841981
## [71]  0.0061435349  0.0176700947 -0.0136229826 -0.0046620045  0.0098052343
## [76]  0.0220974316 -0.0019056206 -0.0021083291  0.0073055016 -0.0048755640
## [81]  0.0036038586  0.0162061036  0.0017394746  0.0096340982  0.0299583499
## [86]  0.0186215155  0.0242795548  0.0069129742 -0.0106440760  0.0006622027
## [91] -0.0144317826 -0.0182453829  0.0133864453  0.0041984423  0.0073618767
```

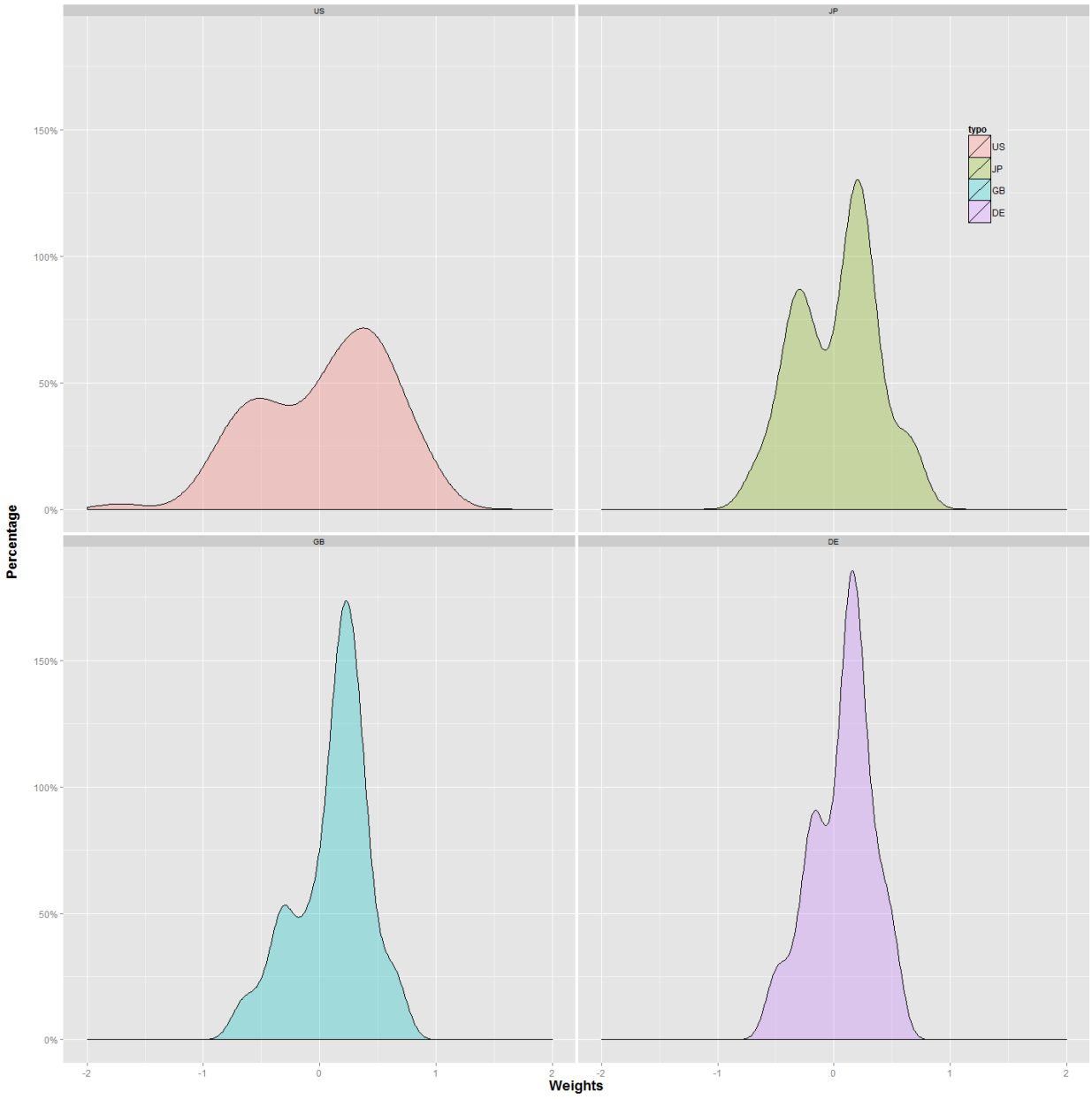
```
toplot_df <-  melt(results[,c("DATES", "STRATEGY_TODAY","EQ_WEIGHTS_BOND")],"DATES")
my_title <- paste("Macro Sentiment ML over Ravenpack ESS metrics and taxonomy ",sep="")
g<-ggplot(
  toplot_df,aes(
    x = DATES,y = value,group = variable,color = variable
  )
) +
  geom_line() +
  scale_x_date() +
  ggtitle(my_title) + xlab("Time") + ylab("Cumulated sentiment") +
  theme(title = element_text(size = 12, face = 'bold')) +
  theme(legend.position = c(0.2,0.8), legend.box = "vertical") +
  theme(legend.background = element_rect(fill = "gray90")) +
  theme(legend.key.size = unit(0.7, "cm"))
print(g)
```


```
# ExportPlot(g,outputDataPathStrategyMonth,paste(my_result_spread_name,"_bench",sep=""))

toplot_df <-  melt(results[,c("DATES", "STRATEGY_TODAY","JP_FIRST_BOND","JP_SECOND_BOND","DE_SECOND_BONI
my_title <- paste("Macro Sentiment ML over Ravenpack ESS metrics and taxonomy ",sep="")
g<-ggplot(
  toplot_df,aes(
    x = DATES,y = value,group = variable,color = variable
  )
) +
  geom_line() +
  scale_x_date() +
  ggtitle(my_title) + xlab("Time") + ylab("Cumulated sentiment") +
  theme(title = element_text(size = 12, face = 'bold')) +
  theme(legend.position = c(0.2,0.8), legend.box = "vertical") +
```

```
  theme(legend.background = element_rect(fill = "gray90")) +
  theme(legend.key.size = unit(0.7, "cm"))
print(g)
```

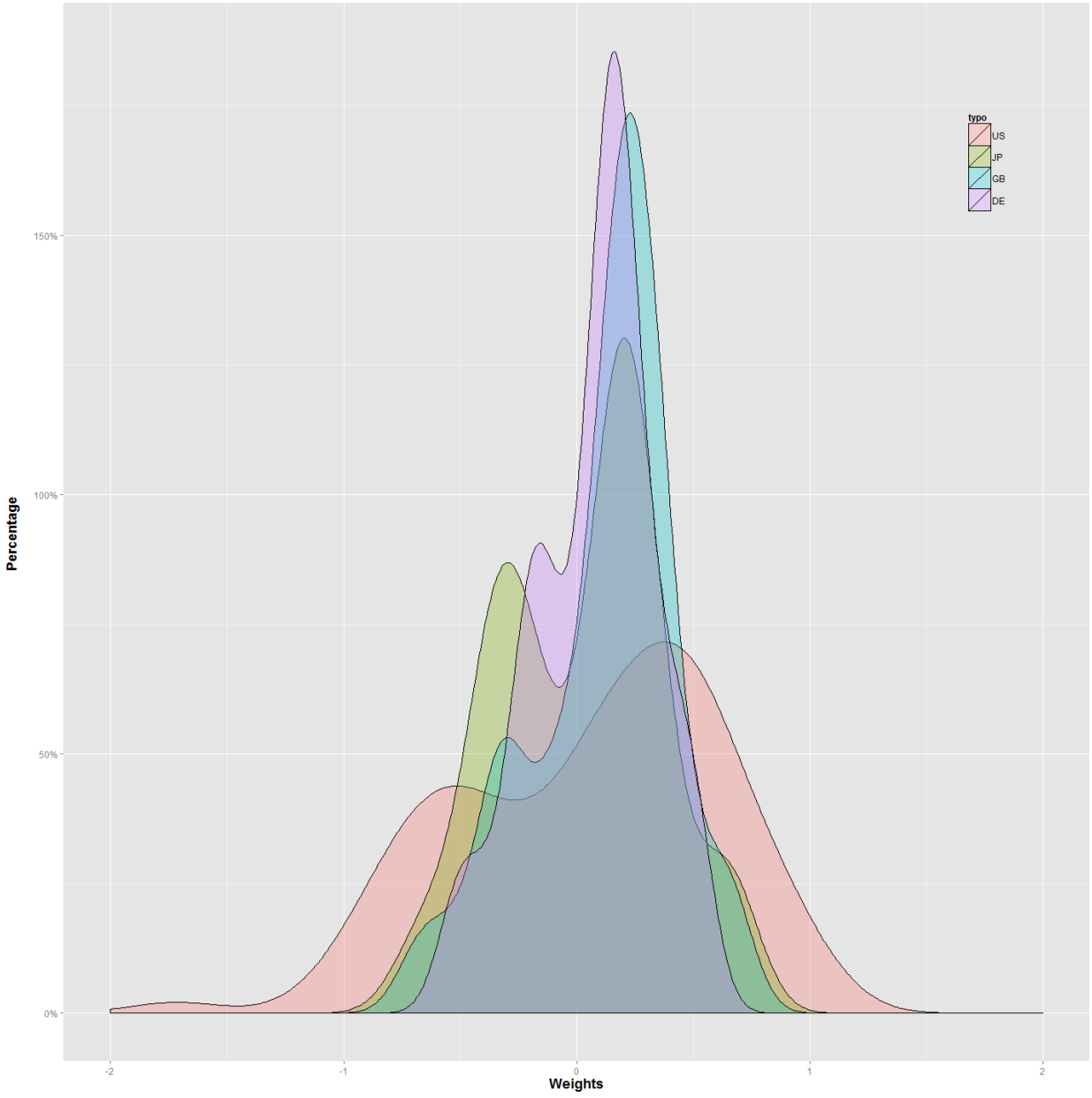

```
weights_decomposition_df <- data.frame(weights=results$US_WEIGHT,typo="US")
weights_decomposition_df <- rbind(weights_decomposition_df, data.frame(weights=results$JP_WEIGHT,typo="
weights_decomposition_df <- rbind(weights_decomposition_df, data.frame(weights=results$GB_WEIGHT,typo="
weights_decomposition_df <- rbind(weights_decomposition_df, data.frame(weights=results$DE_WEIGHT,typo="
my_title <-"Distribution of weights per country"
my_xaxis_title <- paste("Weights")
my_yaxis_title <- paste("Percentage","\n")

g <- ggplot(weights_decomposition_df, aes(x=weights, fill=typo)) +geom_density(alpha=.3)+
```

```
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-2,2))+facet_wrap(~typo, nc
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```



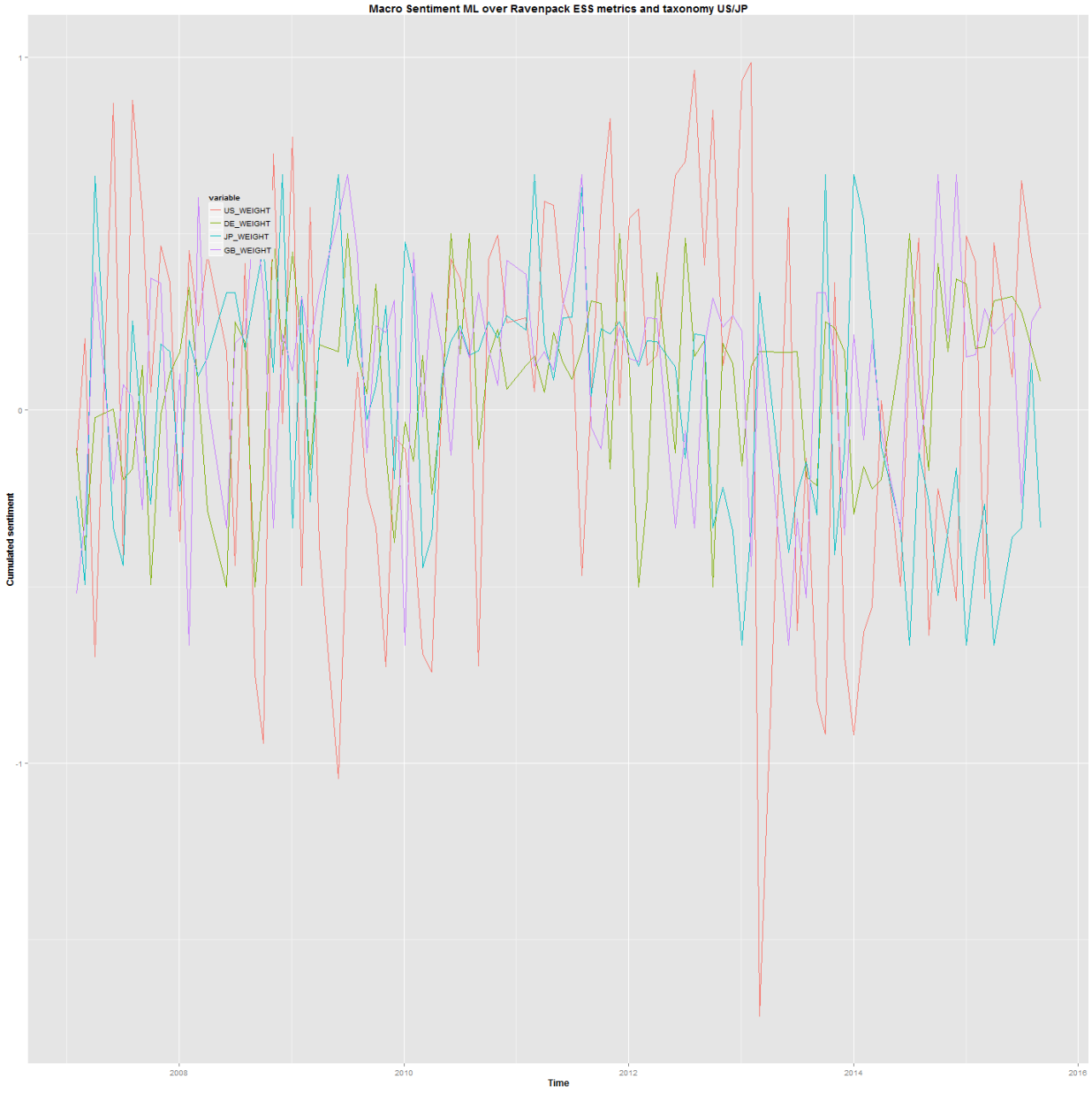

```
g <- ggplot(weights_decomposition_df, aes(x=weights, fill=typo)) +geom_density(alpha=.3)+
  scale_y_continuous(labels = percent_format())+scale_x_continuous(limits=c(-2,2))
g <- g +ylab(my_yaxis_title)+xlab(my_xaxis_title)+
```

```
  theme(axis.text.x = element_text(size=12),axis.text.y = element_text(size=12),title =element_text(siz
  theme(legend.position=c(0.9,0.85), legend.box = "vertical")+
  theme(legend.background = element_rect(fill="gray90"))+
  theme(legend.key.size = unit(1., "cm"))+
  theme(legend.text =  element_text(size=12,colour="black"))+
  theme(legend.title =  element_text(size=12,colour="black"))
print(g)
```




```
toplot_df <-  melt(results[,c("DATES","US_WEIGHT","DE_WEIGHT","JP_WEIGHT","GB_WEIGHT")],"DATES")

my_title <-paste("Macro Sentiment ML over Ravenpack ESS metrics and taxonomy ",my_pair[1],"/",my_pair[2
g<-ggplot(
  toplot_df,aes(
```

```
    x = DATES,y = value,group = variable,color = variable
  )
) +
  geom_line() +
  scale_x_date() +
  ggtitle(my_title) + xlab("Time") + ylab("Cumulated sentiment") +
  theme(title = element_text(size = 12, face = 'bold')) +
  theme(legend.position = c(0.2,0.8), legend.box = "vertical") +
  theme(legend.background = element_rect(fill = "gray90")) #+
  # theme(legend.key.size = unit(0.7, "cm"))
print(g)
```

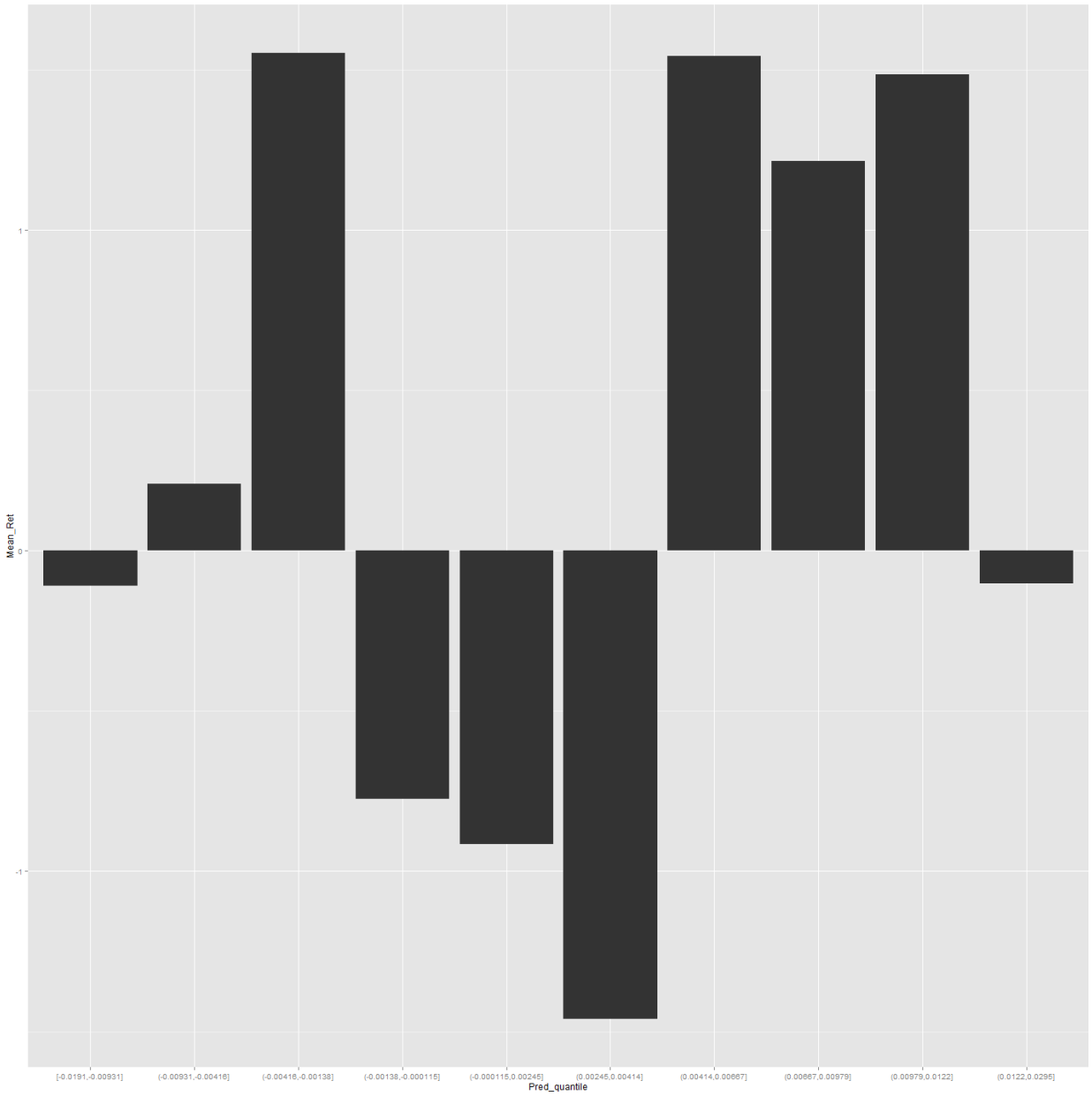


```
# ExportPlot(g,outputDataPathStrategyMonth,paste(my_result_spread_name,"_weights",sep=""))

prediction_results <- tryCatch(  readRDS(paste(outputDataPathStrategyMonth,"all_pairs_prediction_results
                                 error = function(e) {NULL})
if (!is.null(prediction_results)){

  # correlation between prediction and returns for US
  print("Correlation US unidirection prediction")
  print(cor(prediction_results$USPrediction,prediction_results$USEffective))

  for (my_quantile in c(0.1,1/5,1/3)){
    US_df <- prediction_results[,c("USPrediction","USEffective")]
    US_df$my_tiles <- with(US_df, cut(USPrediction,
                                      breaks=quantile(USPrediction, probs=seq(0,1, by=my_quantile)),
                                      include.lowest=TRUE))

    my_IR_computation <- function(x){
      IR <- mean(x,na.rm=TRUE)/sd(x,na.rm=TRUE)*sqrt(12)
      return(IR)
    }

    agg_US_df <- aggregate(US_df$USEffective, by=list(US_df$my_tiles),FUN=my_IR_computation)
    colnames(agg_US_df) <- c("Pred_quantile","Mean_Ret")
    ## Histogram plotting of the return average per prediction quantile
    g <- ggplot(agg_US_df, aes(x=Pred_quantile,y=Mean_Ret)) +
      geom_histogram(stat = "identity")
    print(g)
  }


  Predictionbullish <- prediction_results$USPrediction >= 0
  NextReturnbullish <- prediction_results$USEffective >= 0

  Predictionbearish <- prediction_results$USPrediction <= 0
  NextReturnbearish <- prediction_results$USEffective <= 0

  print("Long confusion matrix")
  print(confusionMatrix(Predictionbullish, NextReturnbullish))

  print("Short confusion matrix")
  print(confusionMatrix(Predictionbearish, NextReturnbearish))

  print("Correlation GB unidirection prediction")
  print(cor(prediction_results$GBPrediction,prediction_results$GBEffective))
  for (my_quantile in c(0.1,1/5,1/3)){
    df <- prediction_results[,c("GBPrediction","GBEffective")]
    df$my_tiles <- with(df, cut(GBPrediction,
                                breaks=quantile(GBPrediction, probs=seq(0,1, by=my_quantile)),
                                include.lowest=TRUE))
    agg_df <- aggregate(df$GBEffective, by=list(df$my_tiles),FUN=my_IR_computation)
    colnames(agg_df) <- c("Pred_quantile","Mean_Ret")
    ## Histogram plotting of the return average per prediction quantile
    g <- ggplot(agg_df, aes(x=Pred_quantile,y=Mean_Ret)) +
```

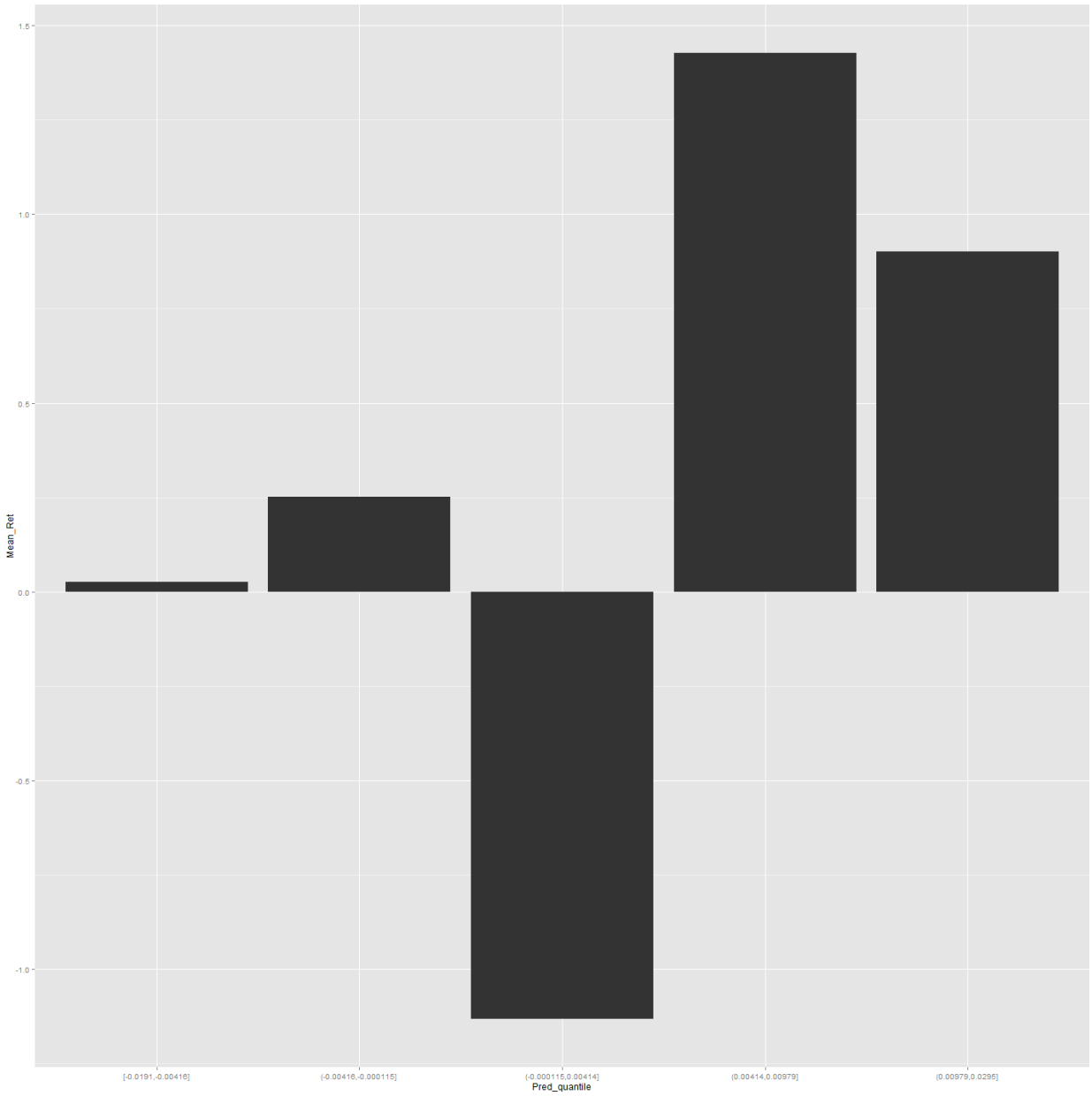
```
    geom_histogram(stat = "identity")
  print(g)
}
Predictionbullish <- prediction_results$GBPrediction >= 0
NextReturnbullish <- prediction_results$GBEffective >= 0

Predictionbearish <- prediction_results$GBPrediction <= 0
NextReturnbearish <- prediction_results$GBEffective <= 0

print("Long confusion matrix")
print(confusionMatrix(Predictionbullish, NextReturnbullish))

print("Short confusion matrix")
print(confusionMatrix(Predictionbearish, NextReturnbearish))

print("Correlation DE unidirection prediction")
print(cor(prediction_results$DEPrediction,prediction_results$DEEffective))
for (my_quantile in c(0.1,1/5,1/3)){
  df <- prediction_results[,c("DEPrediction","DEEffective")]
  df$my_tiles <- with(df, cut(DEPrediction,
                              breaks=quantile(DEPrediction, probs=seq(0,1, by=my_quantile)),
                              include.lowest=TRUE))
  agg_df <- aggregate(df$DEEffective, by=list(df$my_tiles),FUN=my_IR_computation)
  colnames(agg_df) <- c("Pred_quantile","Mean_Ret")
  ## Histogram plotting of the return average per prediction quantile
  g <- ggplot(agg_df, aes(x=Pred_quantile,y=Mean_Ret)) +
    geom_histogram(stat = "identity")
  print(g)
}
Predictionbullish <- prediction_results$DEPrediction >= 0
NextReturnbullish <- prediction_results$DEEffective >= 0

Predictionbearish <- prediction_results$DEPrediction <= 0
NextReturnbearish <- prediction_results$DEEffective <= 0

print("Long confusion matrix")
print(confusionMatrix(Predictionbullish, NextReturnbullish))

print("Short confusion matrix")
print(confusionMatrix(Predictionbearish, NextReturnbearish))

print("Correlation GB unidirection prediction")
print(cor(prediction_results$GBPrediction,prediction_results$GBEffective))

Predictionbullish <- prediction_results$GBPrediction >= 0
NextReturnbullish <- prediction_results$GBEffective >= 0

Predictionbearish <- prediction_results$GBPrediction <= 0
NextReturnbearish <- prediction_results$GBEffective <= 0

print("Long confusion matrix")
print(confusionMatrix(Predictionbullish, NextReturnbullish))
```

```
  print("Short confusion matrix")
  print(confusionMatrix(Predictionbearish, NextReturnbearish))

  print("Correlation JP unidirection prediction")
  print(cor(prediction_results$JPPrediction,prediction_results$JPEffective))
  for (my_quantile in c(0.1,1/5,1/3)){
    df <- prediction_results[,c("JPPrediction","JPEffective")]
    df$my_tiles <- with(df, cut(JPPrediction,
                                breaks=quantile(JPPrediction, probs=seq(0,1, by=my_quantile)),
                                include.lowest=TRUE))
    agg_df <- aggregate(df$JPEffective, by=list(df$my_tiles),FUN=my_IR_computation)
    colnames(agg_df) <- c("Pred_quantile","Mean_Ret")
    ## Histogram plotting of the return average per prediction quantile
    g <- ggplot(agg_df, aes(x=Pred_quantile,y=Mean_Ret)) +
      geom_histogram(stat = "identity")
    print(g)
  }

  Predictionbullish <- prediction_results$JPPrediction >= 0
  NextReturnbullish <- prediction_results$JPEffective >= 0

  Predictionbearish <- prediction_results$JPPrediction <= 0
  NextReturnbearish <- prediction_results$JPEffective <= 0

  print("Long confusion matrix")
  print(confusionMatrix(Predictionbullish, NextReturnbullish))

  print("Short confusion matrix")
  print(confusionMatrix(Predictionbearish, NextReturnbearish))

}
```

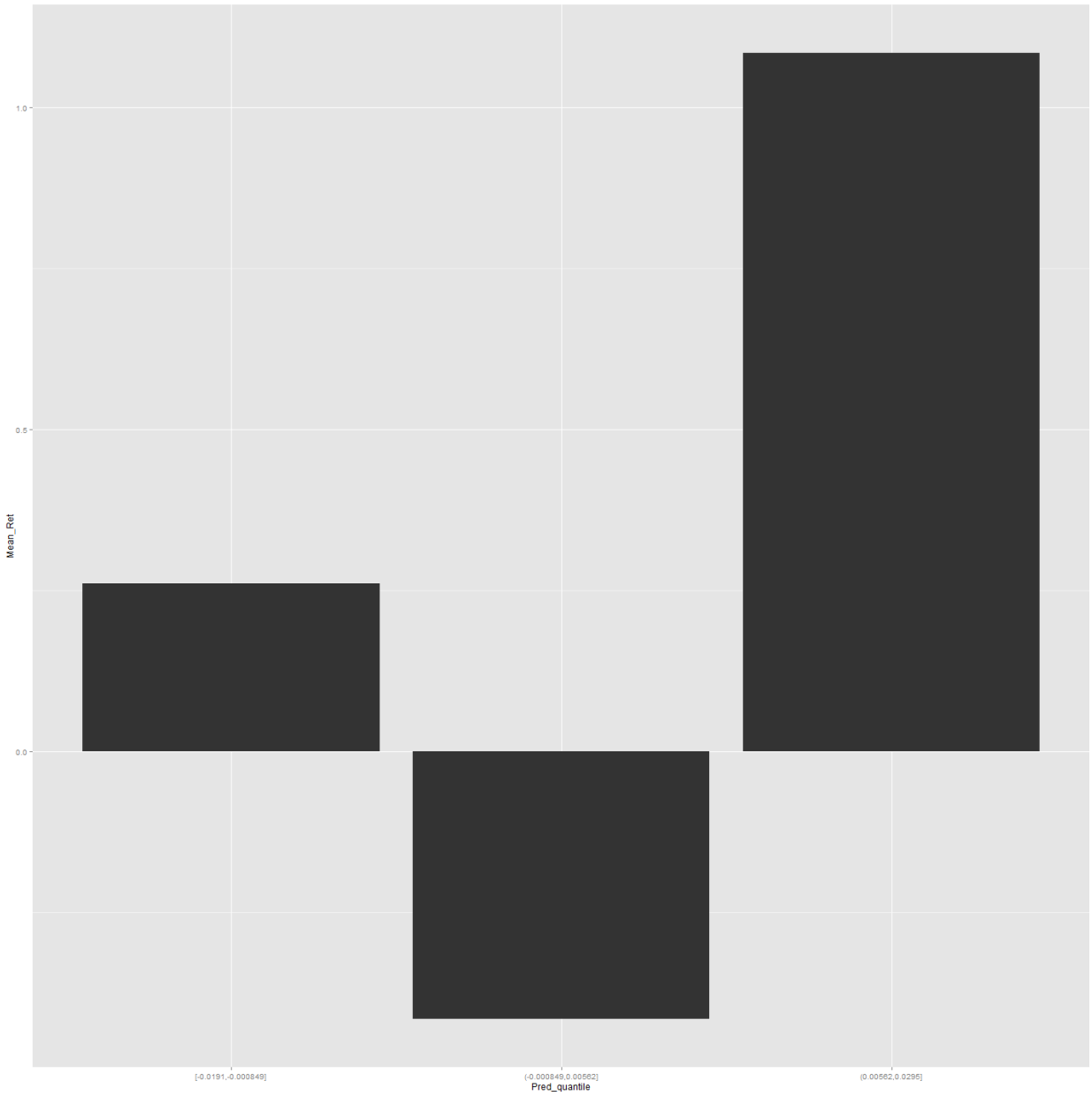
```
## [1] "Correlation US unidirection prediction"
## [1] 0.0880182

## Warning: Stacking not well defined when ymin != 0
```




```
## Warning: Stacking not well defined when ymin != 0
```

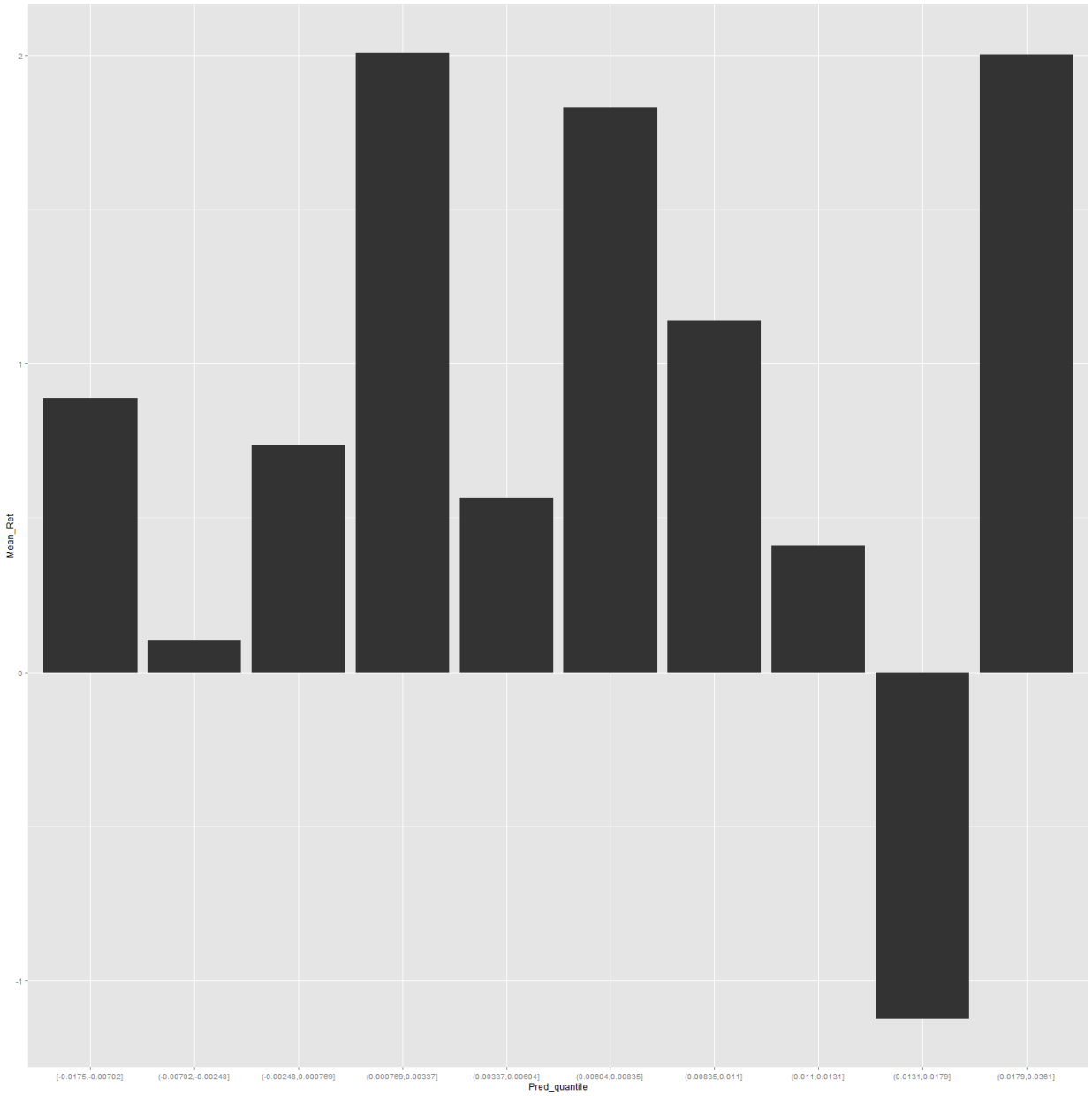


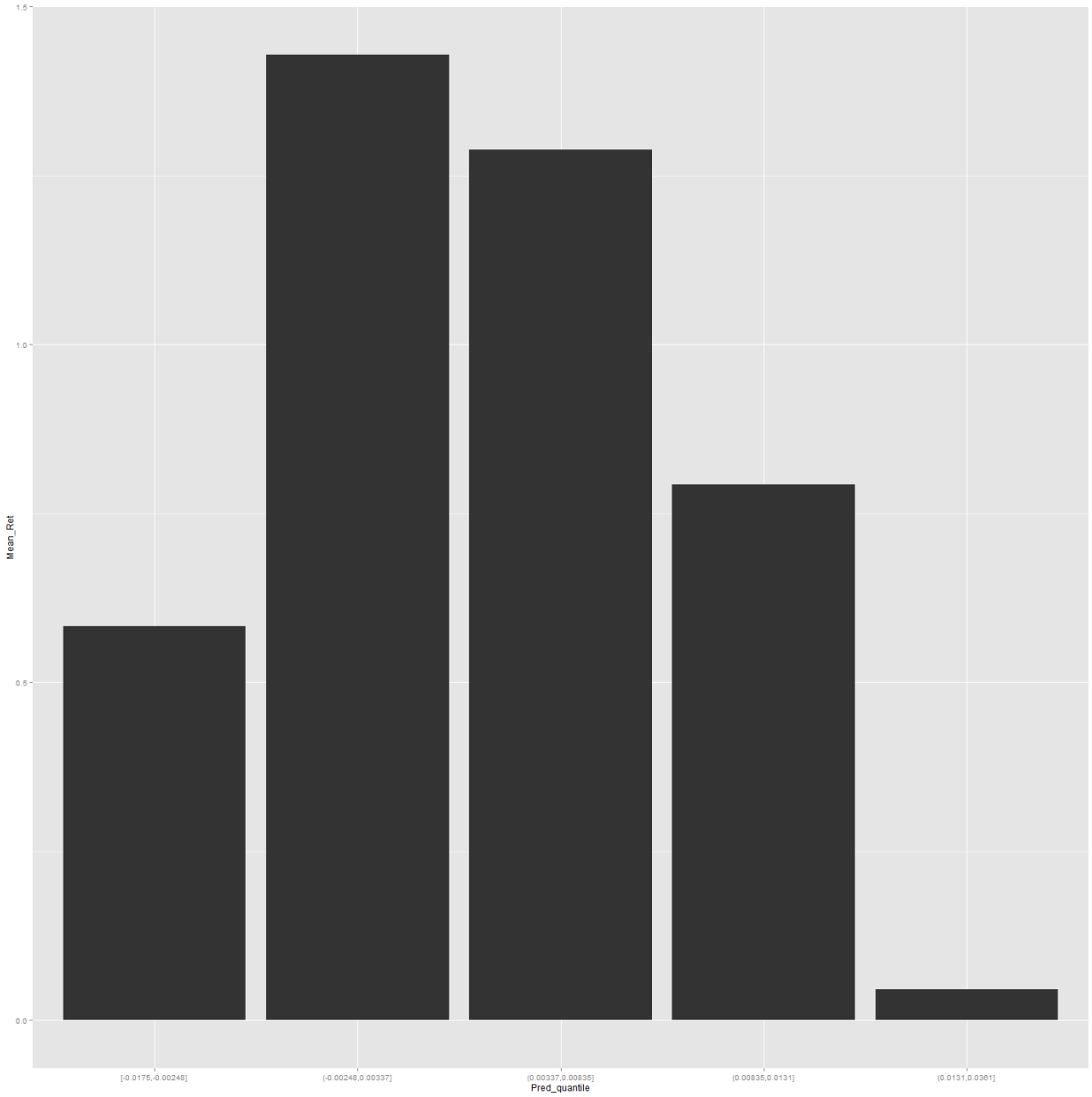

```
## Warning: Stacking not well defined when ymin != 0
```

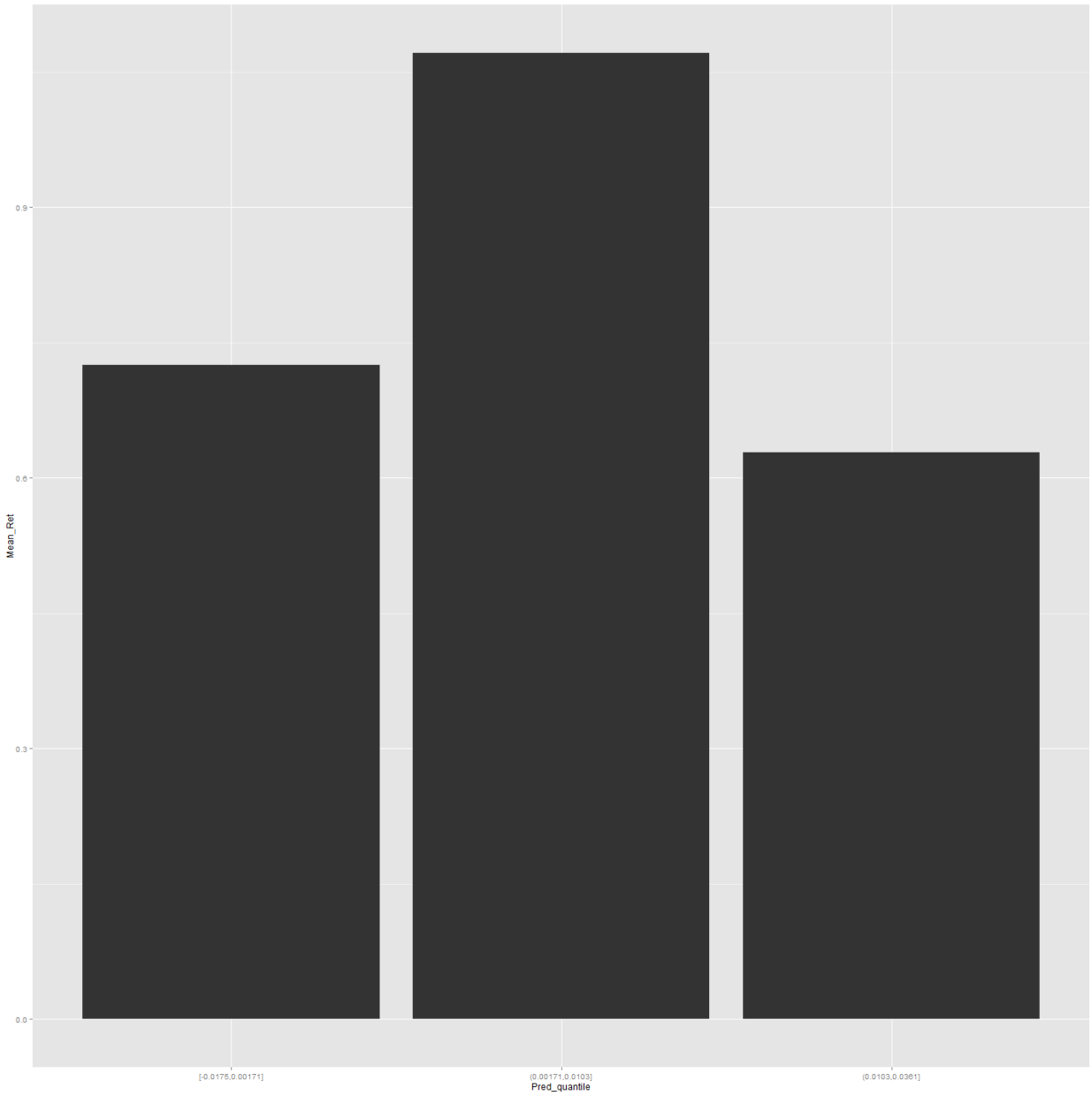



```
## [1] "Long confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    20   22
##      TRUE     30   32
##
##                Accuracy : 0.5
##                  95% CI : (0.4003, 0.5997)
##     No Information Rate : 0.5192
##     P-Value [Acc > NIR] : 0.6884
##
##                   Kappa : -0.0075
```

```
##  Mcnemar's Test P-Value : 0.3317
##
##             Sensitivity : 0.4000
##             Specificity : 0.5926
##          Pos Pred Value : 0.4762
##          Neg Pred Value : 0.5161
##              Prevalence : 0.4808
##          Detection Rate : 0.1923
##    Detection Prevalence : 0.4038
##       Balanced Accuracy : 0.4963
##
##        'Positive' Class : FALSE
##
## [1] "Short confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    32   30
##      TRUE     22   20
##
##                Accuracy : 0.5
##                  95% CI : (0.4003, 0.5997)
##     No Information Rate : 0.5192
##     P-Value [Acc > NIR] : 0.6884
##
##                   Kappa : -0.0075
##  Mcnemar's Test P-Value : 0.3317
##
##             Sensitivity : 0.5926
##             Specificity : 0.4000
##          Pos Pred Value : 0.5161
##          Neg Pred Value : 0.4762
##              Prevalence : 0.5192
##          Detection Rate : 0.3077
##    Detection Prevalence : 0.5962
##       Balanced Accuracy : 0.4963
##
##        'Positive' Class : FALSE
##
## [1] "Correlation GB unidirection prediction"
## [1] 0.006945995

## Warning: Stacking not well defined when ymin != 0
```

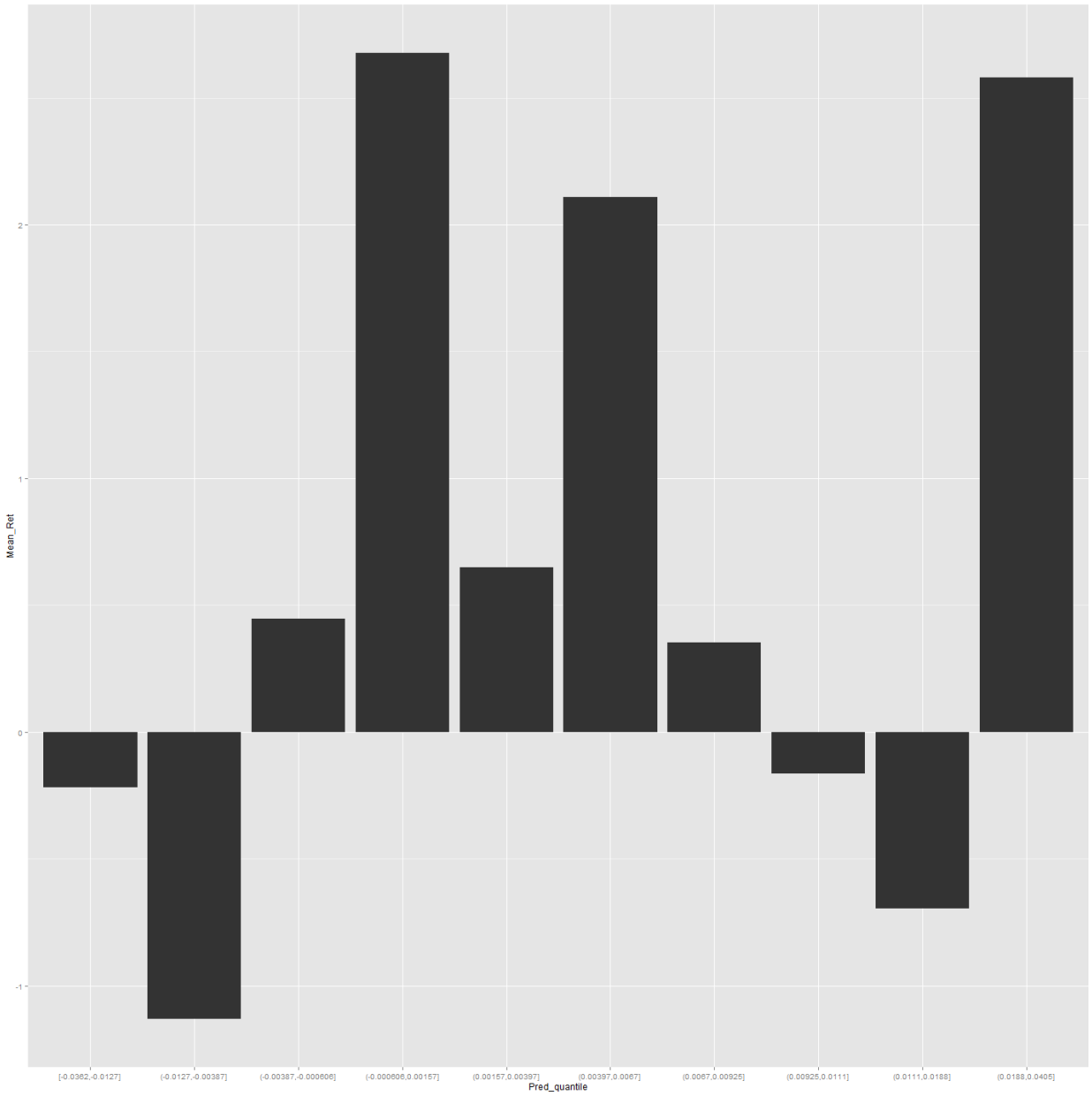

2
1
0
-1
Mean_Ret
[-0.0175,-0.00702]
(-0.00702,-0.00248]
(-0.00248,0.000769]
(0.000769,0.00337]
(0.00337,0.00604]
(0.00604,0.00835]
(0.00835,0.011]
(0.011,0.0131]
(0.0131,0.0179]
(0.0179,0.0361]
Pred_quantile


1.5
1.0
0.5
0.0
Mean_Ret
[-0.0175,-0.00248]
(-0.00248,0.00337]
(0.00337,0.00835]
(0.00835,0.0131]
(0.0131,0.0361]
Pred_quantile

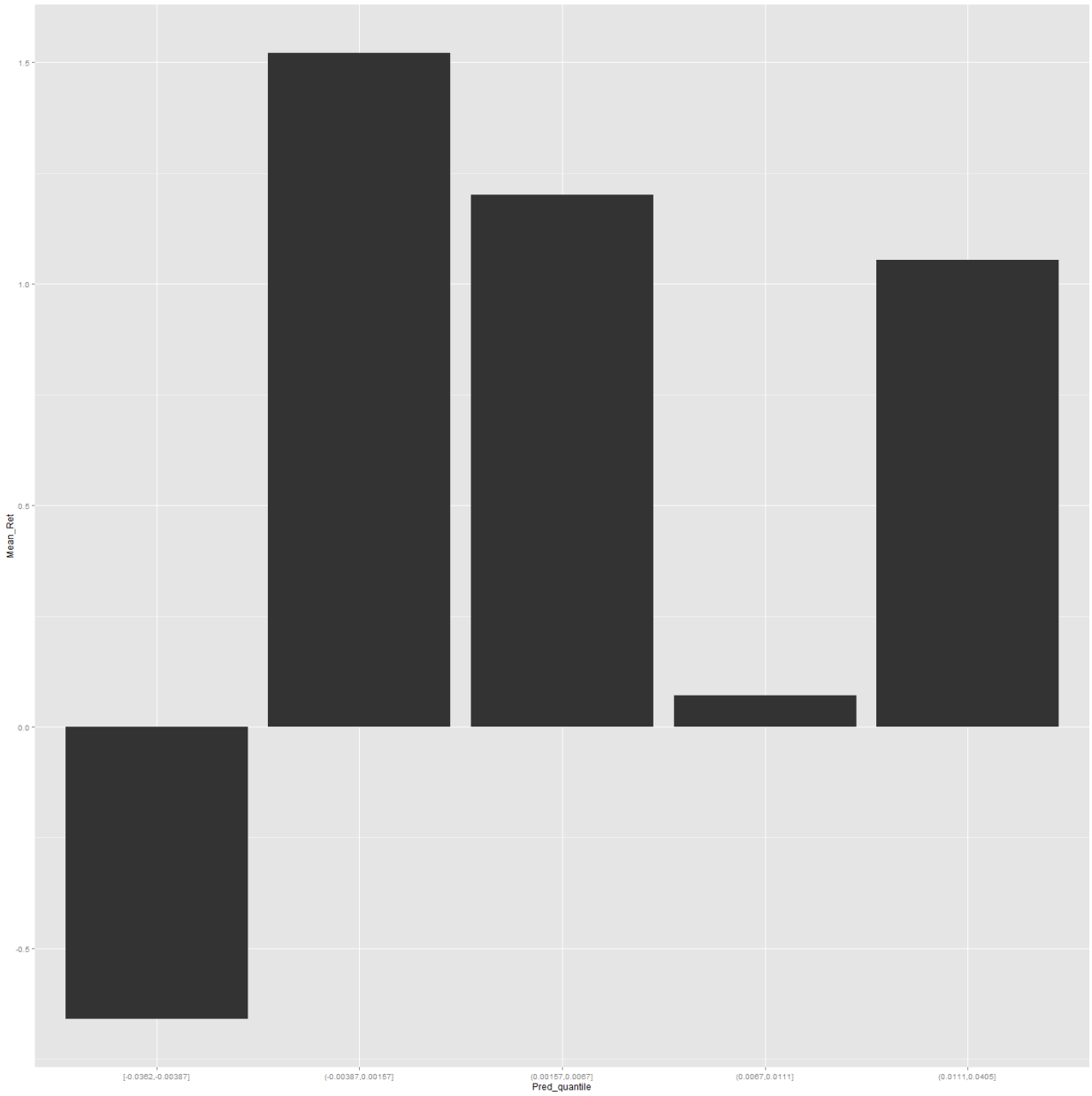



```
## [1] "Long confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    11   17
##      TRUE     34   42
##
##                Accuracy : 0.5096
##                  95% CI : (0.4097, 0.609)
##     No Information Rate : 0.5673
##     P-Value [Acc > NIR] : 0.90048
##
##                   Kappa : -0.0457
```

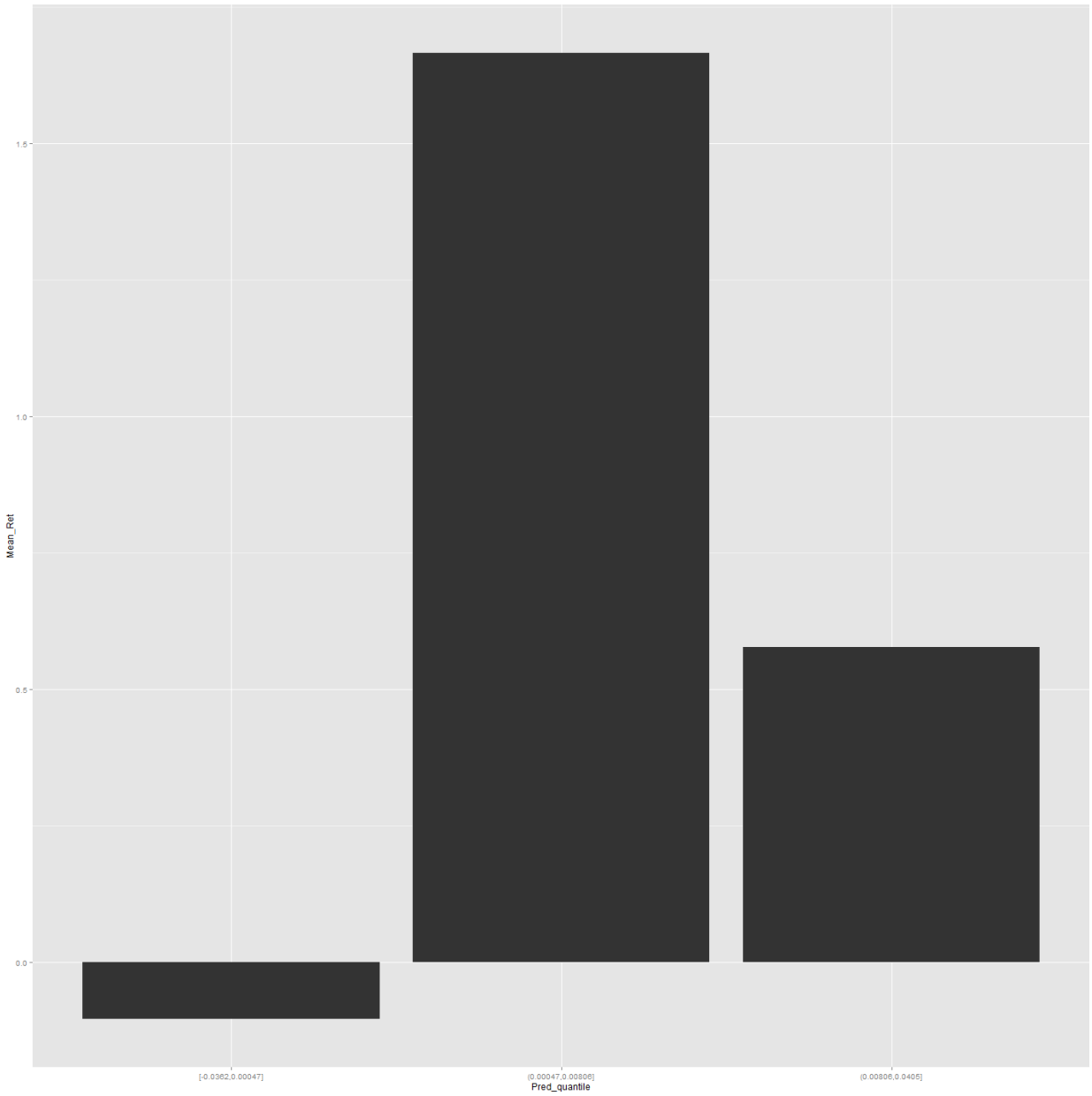
```
##  Mcnemar's Test P-Value : 0.02506
##
##             Sensitivity : 0.2444
##             Specificity : 0.7119
##          Pos Pred Value : 0.3929
##          Neg Pred Value : 0.5526
##              Prevalence : 0.4327
##          Detection Rate : 0.1058
##    Detection Prevalence : 0.2692
##       Balanced Accuracy : 0.4782
##
##        'Positive' Class : FALSE
##
## [1] "Short confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    42   34
##      TRUE     17   11
##
##                Accuracy : 0.5096
##                  95% CI : (0.4097, 0.609)
##     No Information Rate : 0.5673
##     P-Value [Acc > NIR] : 0.90048
##
##                   Kappa : -0.0457
##  Mcnemar's Test P-Value : 0.02506
##
##             Sensitivity : 0.7119
##             Specificity : 0.2444
##          Pos Pred Value : 0.5526
##          Neg Pred Value : 0.3929
##              Prevalence : 0.5673
##          Detection Rate : 0.4038
##    Detection Prevalence : 0.7308
##       Balanced Accuracy : 0.4782
##
##        'Positive' Class : FALSE
##
## [1] "Correlation DE unidirection prediction"
## [1] 0.2137453

## Warning: Stacking not well defined when ymin != 0
```




```
## Warning: Stacking not well defined when ymin != 0
```



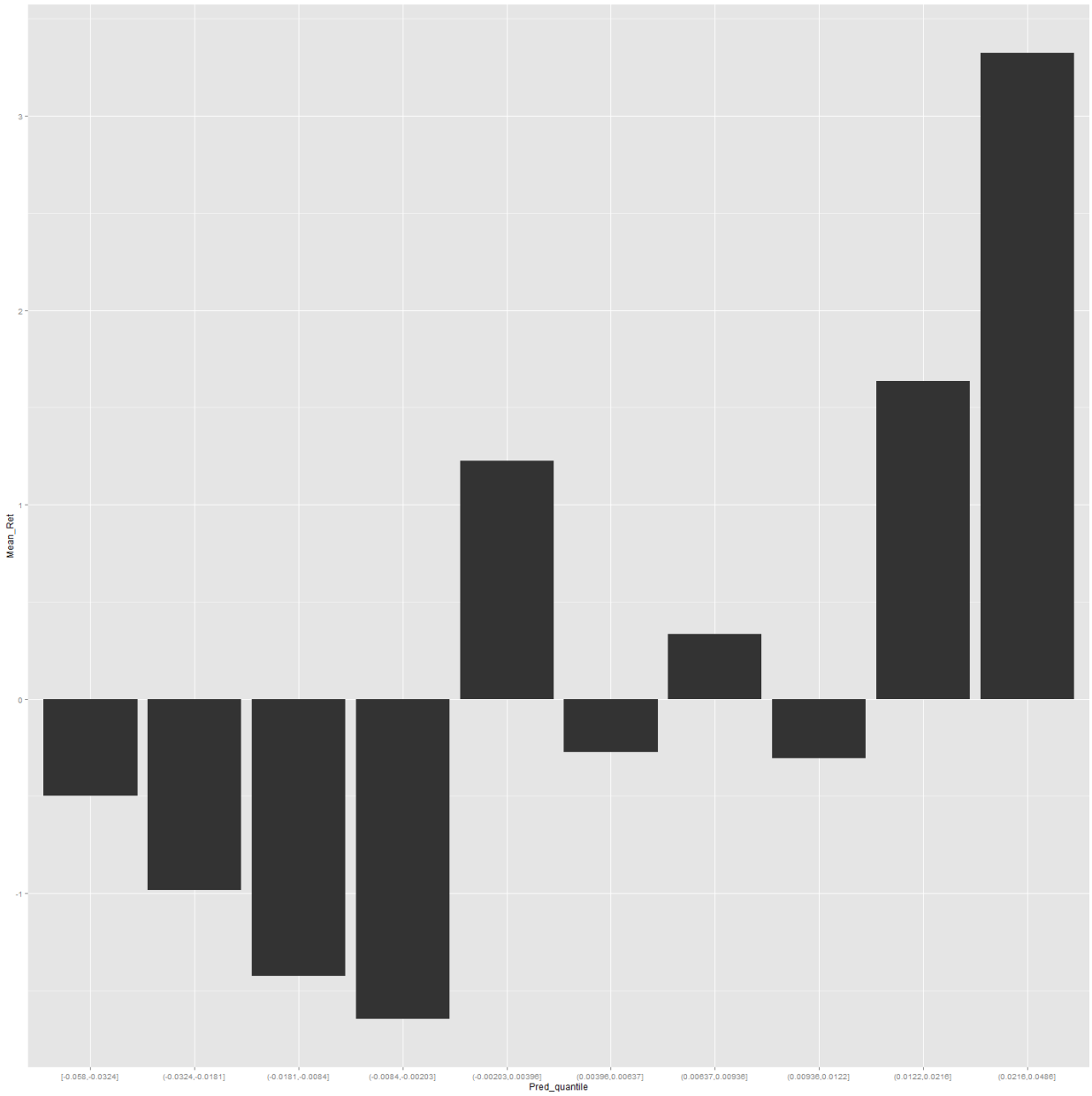

```
## Warning: Stacking not well defined when ymin != 0
```



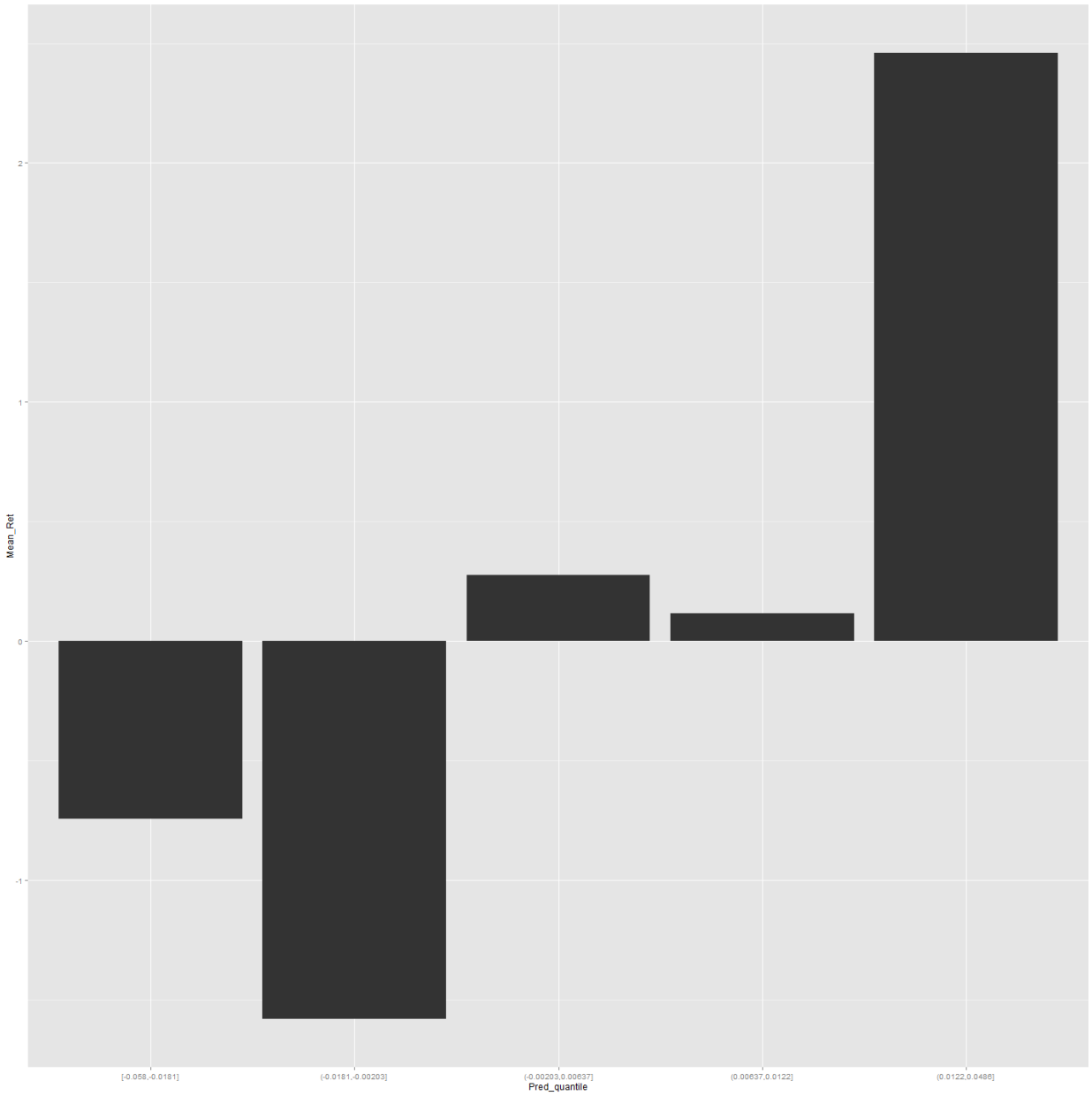

```
## [1] "Long confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    15   18
##      TRUE     29   42
##
##                Accuracy : 0.5481
##                  95% CI : (0.4474, 0.6459)
##     No Information Rate : 0.5769
##     P-Value [Acc > NIR] : 0.7571
##
##                   Kappa : 0.0423
```

```
##  Mcnemar's Test P-Value : 0.1447
##
##             Sensitivity : 0.3409
##             Specificity : 0.7000
##          Pos Pred Value : 0.4545
##          Neg Pred Value : 0.5915
##              Prevalence : 0.4231
##          Detection Rate : 0.1442
##    Detection Prevalence : 0.3173
##       Balanced Accuracy : 0.5205
##
##        'Positive' Class : FALSE
##
## [1] "Short confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    42   29
##      TRUE     18   15
##
##                Accuracy : 0.5481
##                  95% CI : (0.4474, 0.6459)
##     No Information Rate : 0.5769
##     P-Value [Acc > NIR] : 0.7571
##
##                   Kappa : 0.0423
##  Mcnemar's Test P-Value : 0.1447
##
##             Sensitivity : 0.7000
##             Specificity : 0.3409
##          Pos Pred Value : 0.5915
##          Neg Pred Value : 0.4545
##              Prevalence : 0.5769
##          Detection Rate : 0.4038
##    Detection Prevalence : 0.6827
##       Balanced Accuracy : 0.5205
##
##        'Positive' Class : FALSE
##
## [1] "Correlation GB unidirection prediction"
## [1] 0.006945995
## [1] "Long confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    11   17
##      TRUE     34   42
##
##                Accuracy : 0.5096
##                  95% CI : (0.4097, 0.609)
##     No Information Rate : 0.5673
##     P-Value [Acc > NIR] : 0.90048
```

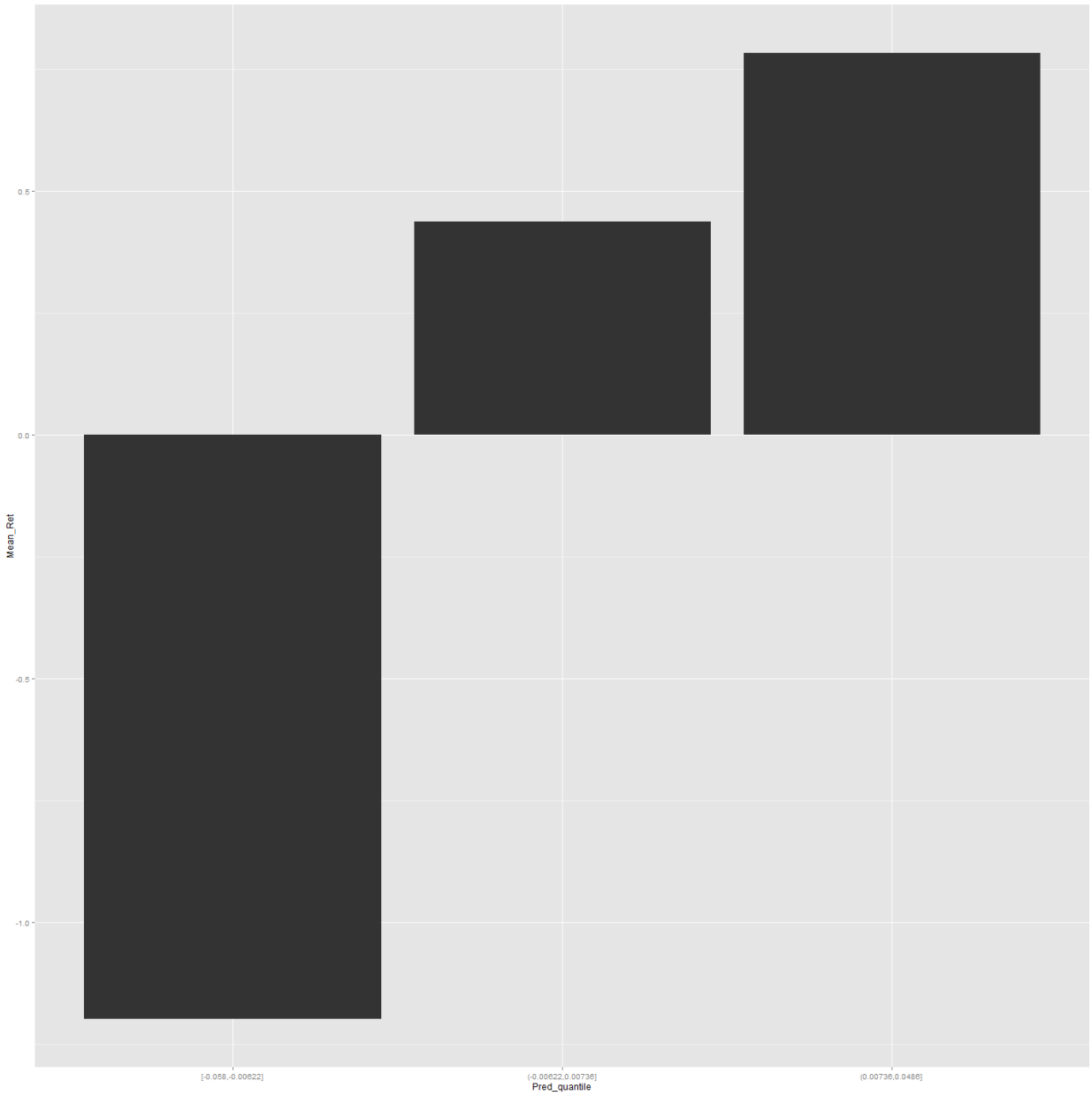
```
##
##                   Kappa : -0.0457
##  Mcnemar's Test P-Value : 0.02506
##
##             Sensitivity : 0.2444
##             Specificity : 0.7119
##          Pos Pred Value : 0.3929
##          Neg Pred Value : 0.5526
##              Prevalence : 0.4327
##          Detection Rate : 0.1058
##    Detection Prevalence : 0.2692
##       Balanced Accuracy : 0.4782
##
##        'Positive' Class : FALSE
##
## [1] "Short confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    42   34
##      TRUE     17   11
##
##                Accuracy : 0.5096
##                  95% CI : (0.4097, 0.609)
##     No Information Rate : 0.5673
##     P-Value [Acc > NIR] : 0.90048
##
##                   Kappa : -0.0457
##  Mcnemar's Test P-Value : 0.02506
##
##             Sensitivity : 0.7119
##             Specificity : 0.2444
##          Pos Pred Value : 0.5526
##          Neg Pred Value : 0.3929
##              Prevalence : 0.5673
##          Detection Rate : 0.4038
##    Detection Prevalence : 0.7308
##       Balanced Accuracy : 0.4782
##
##        'Positive' Class : FALSE
##
## [1] "Correlation JP unidirection prediction"
## [1] 0.2868498

## Warning: Stacking not well defined when ymin != 0
```




```
## Warning: Stacking not well defined when ymin != 0
```




```
## Warning: Stacking not well defined when ymin != 0
```




```
## [1] "Long confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    28   18
##      TRUE     19   39
##
##                Accuracy : 0.6442
##                  95% CI : (0.5443, 0.7357)
##     No Information Rate : 0.5481
##     P-Value [Acc > NIR] : 0.0298
##
##                   Kappa : 0.2805
```

```
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.5957
##             Specificity : 0.6842
##          Pos Pred Value : 0.6087
##          Neg Pred Value : 0.6724
##              Prevalence : 0.4519
##          Detection Rate : 0.2692
##    Detection Prevalence : 0.4423
##       Balanced Accuracy : 0.6400
##
##        'Positive' Class : FALSE
##
## [1] "Short confusion matrix"
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE    38   20
##      TRUE     18   28
##
##                Accuracy : 0.6346
##                  95% CI : (0.5345, 0.7269)
##     No Information Rate : 0.5385
##     P-Value [Acc > NIR] : 0.03013
##
##                   Kappa : 0.2627
##  Mcnemar's Test P-Value : 0.87113
##
##             Sensitivity : 0.6786
##             Specificity : 0.5833
##          Pos Pred Value : 0.6552
##          Neg Pred Value : 0.6087
##              Prevalence : 0.5385
##          Detection Rate : 0.3654
##    Detection Prevalence : 0.5577
##       Balanced Accuracy : 0.6310
##
##        'Positive' Class : FALSE
##
```

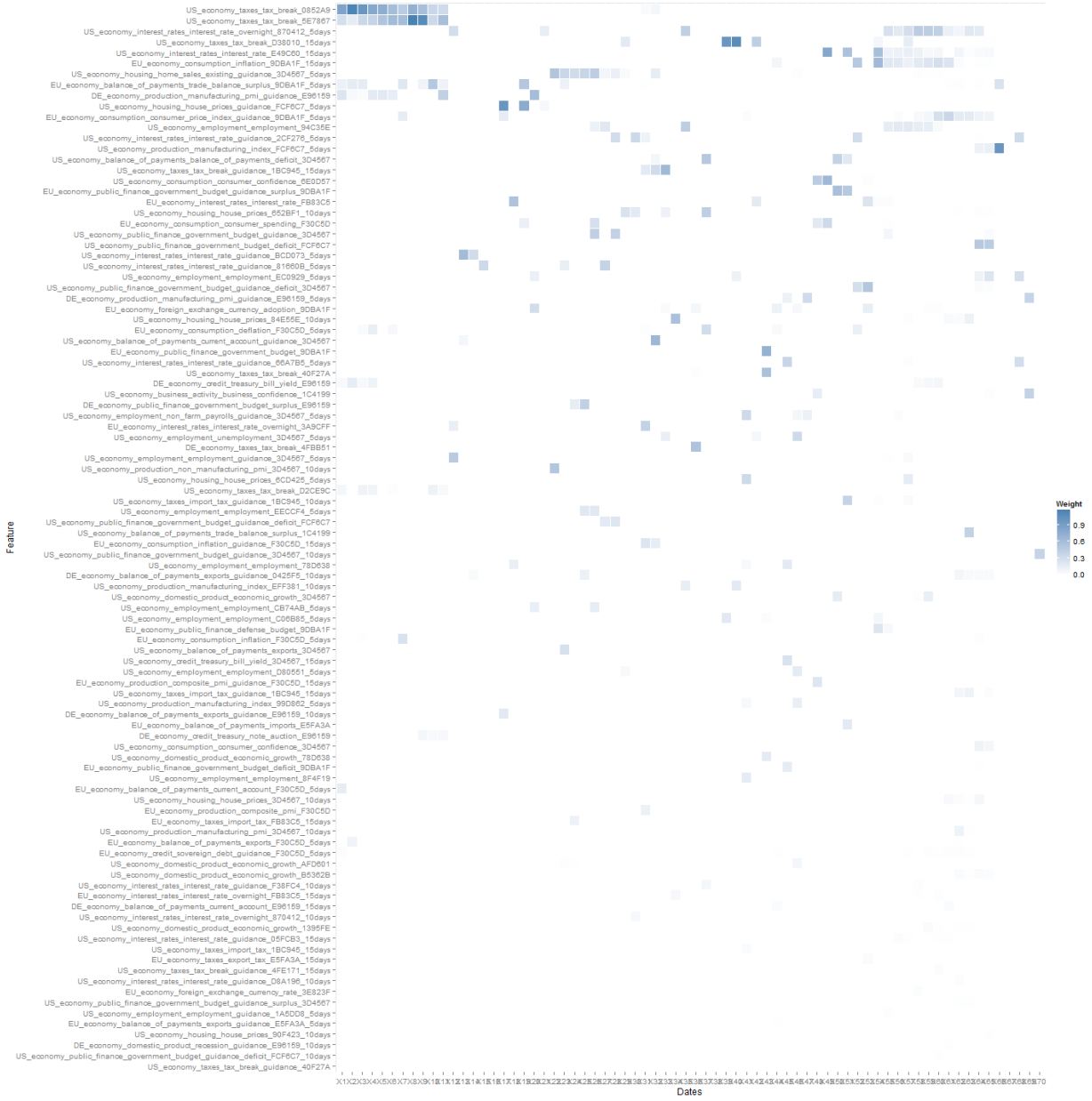
Variable importance visualization over time

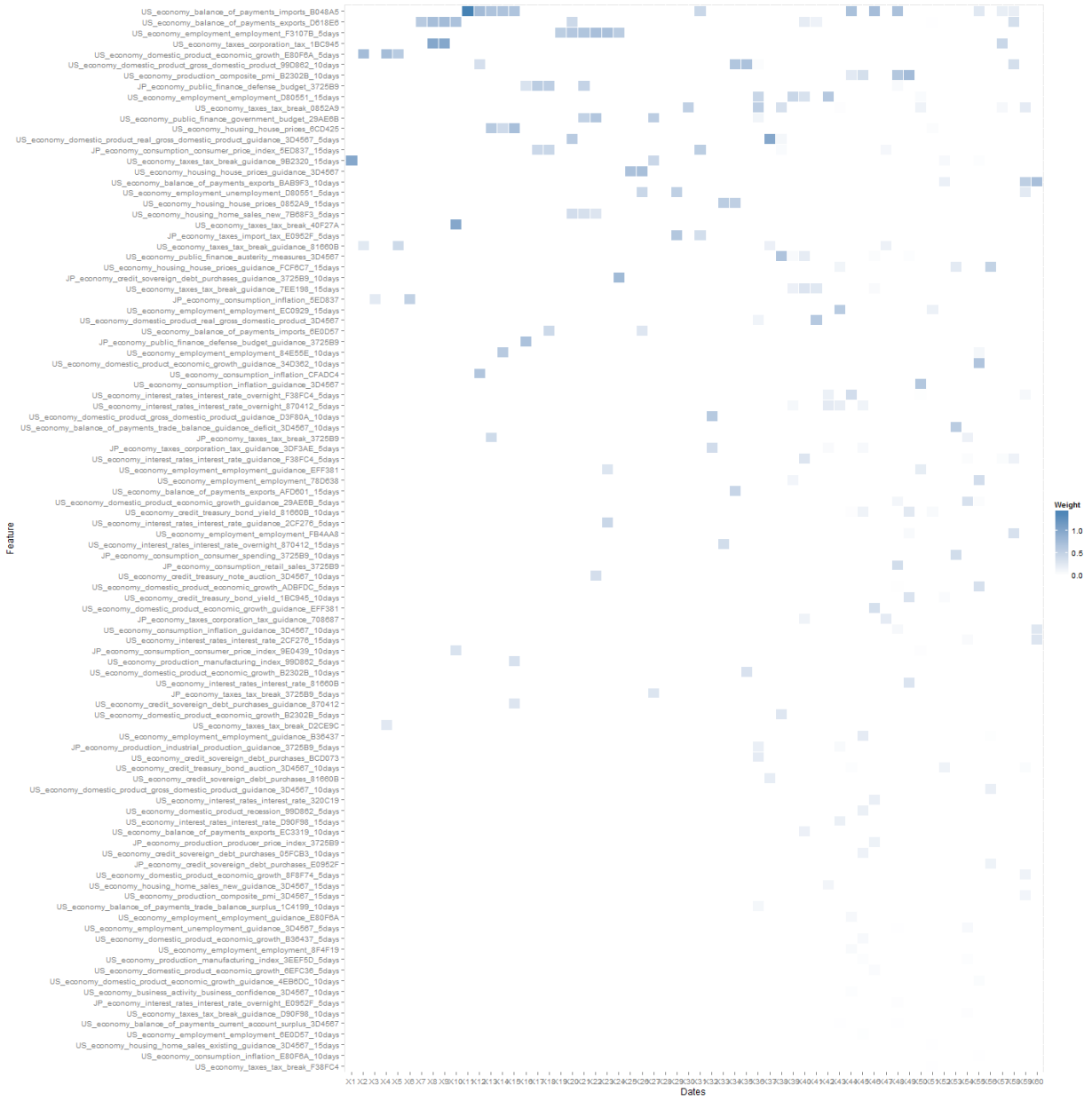
```
my_pairs = list(c("US","DE"),c("US","GB"),c("US","JP"))

my_threshold <- 100
for (my_pair in my_pairs){

  # SaveDataFrame(my_total_df,outputDataPathMonth,paste(my_pair[1], my_pair[2],"variable_importance_spr

  filename <- paste(my_pair[1], my_pair[2],"variable_importance_spread_results_month_2007.rds",sep="")
```

```
  my_total_df <- tryCatch( readRDS(paste(outputDataPathStrategyMonth,filename, sep = "")),
                           error = function(e) {NULL})
  if (!is.null(my_total_df)){


    # we only keep the first 100 most important predictors by importance count over time
    # my_filtered_df <- my_total_df[(sort(rowSums(my_total_df[,-1]), index.return=TRUE, decreasing = TR
    my_index <- rowSums(my_total_df[,-1])!=0
    my_total_df <- my_total_df[my_index,]
    my_filtered_df <- my_total_df[(sort(rowSums(my_total_df[,-1]!=0), index.return=TRUE, decreasing = TR

    my_filtered_df$Feature <- as.factor(my_filtered_df$Feature)
    my_filtered_df <- transform(my_filtered_df, Feature=reorder(Feature,(sort(rowSums(my_filtered_df[,-1
    my_filtered_df <- transform(my_filtered_df, Feature=reorder(Feature,rowSums(my_filtered_df[,-1])))


    my_filtered_df.m <- melt(my_filtered_df,id="Feature")
    colnames(my_filtered_df.m) <- c("Feature","Dates","Weight")
    # my_filtered_df.m <- ddply(my_filtered_df.m, .(variable), transform)
    # my_filtered_df.m <- ddply(my_filtered_df.m, .(variable), transform, rescale = rescale(value))
    p <- ggplot(my_filtered_df.m, aes(Dates, Feature)) + geom_tile(aes(fill = Weight),colour = "white")
      scale_fill_gradient(low = "white", high = "steelblue")
    print(p)
    # ExportPlot(p,outputDataPathStrategyMonth,paste0(my_pair[2],"heatmap"), width=10, height=15)
    print("Done")
  }
}
```

```
## [1] "Done"
```

```
## Warning in gzfile(file, "rb"): cannot open compressed
## file 'W:/OutputData/sduprey/NAR-271/Month_2007/
## USGBvariable_importance_spread_results_month_2007.rds', probable reason 'No
## such file or directory'
```


US_economy_taxes_tax_break_0852A9
US_economy_taxes_tax_break_5E7567
US_economy_interest_rates_interest_rate_overnight_870412_5days
US_economy_taxes_tax_break_D38010_15days
US_economy_interest_rates_interest_rate_E49C60_15days
EU_economy_consumption_inflation_9DBA1F_15days
US_economy_housing_home_sales_existing_guidance_3D4567_5days
EU_economy_balance_of_payments_trade_balance_surplus_9DBA1F_5days
DE_economy_production_manufacturing_pmi_guidance_E96159
US_economy_housing_house_prices_guidance_FCF6C7_5days
EU_economy_consumption_consumer_price_index_guidance_9DBA1F_5days
US_economy_employment_employment_94C35E
US_economy_interest_rates_interest_rate_guidance_2CF276_5days
US_economy_production_manufacturing_index_FCF6C7_5days
US_economy_balance_of_payments_balance_of_payments_deficit_3D4567
US_economy_taxes_tax_break_guidance_1BCB45_15days
US_economy_consumption_consumer_confidence_6E0D57
EU_economy_public_finance_government_budget_guidance_surplus_9DBA1F
EU_economy_interest_rates_interest_rate_FB83C5
US_economy_housing_house_prices_652BF1_10days
EU_economy_consumption_consumer_spending_F30C5D
US_economy_public_finance_government_budget_guidance_3D4567
US_economy_public_finance_government_budget_deficit_FCF6C7
US_economy_interest_rates_interest_rate_guidance_BCD073_5days
US_economy_interest_rates_interest_rate_guidance_81660B_5days
US_economy_employment_employment_EC0929_5days
US_economy_public_finance_government_budget_guidance_deficit_3D4567
DE_economy_production_manufacturing_pmi_guidance_E96159_5days
EU_economy_foreign_exchange_currency_adoption_9DBA1F
US_economy_housing_house_prices_84E55E_10days
EU_economy_consumption_deflation_F30C5D_5days
US_economy_balance_of_payments_current_account_guidance_3D4567
EU_economy_public_finance_government_budget_9DBA1F
US_economy_interest_rates_interest_rate_guidance_66A7B5_5days
US_economy_taxes_tax_break_40F27A
DE_economy_credit_treasury_bill_yield_E96159
US_economy_business_activity_business_confidence_1C4199
DE_economy_public_finance_government_budget_surplus_E96159
US_economy_employment_non_farm_payrolls_guidance_3D4567_5days
EU_economy_interest_rates_interest_rate_overnight_3A9CFF
US_economy_employment_unemployment_3D4567_5days
DE_economy_taxes_tax_break_4FBB51
US_economy_employment_employment_guidance_3D4567_5days
US_economy_production_non_manufacturing_pmi_3D4567_10days
US_economy_housing_house_prices_6CD425_5days
US_economy_taxes_tax_break_D2CE9C
US_economy_taxes_import_tax_guidance_1BCB45_10days
US_economy_employment_employment_EECCF4_5days
US_economy_public_finance_government_budget_guidance_deficit_FCF6C7
US_economy_balance_of_payments_trade_balance_surplus_1C4199
EU_economy_consumption_inflation_guidance_F30C5D_15days
US_economy_public_finance_government_budget_guidance_3D4567_10days
US_economy_employment_employment_78D638
DE_economy_balance_of_payments_exports_guidance_0425F5_10days
US_economy_production_manufacturing_index_EFF381_10days
US_economy_domestic_product_economic_growth_3D4567
US_economy_employment_employment_CB74AB_5days
US_economy_employment_employment_C0BB85_5days
EU_economy_public_finance_defense_budget_9DBA1F
EU_economy_consumption_inflation_F30C5D_5days
US_economy_balance_of_payments_exports_3D4567
US_economy_credit_treasury_bill_yield_3D4567_15days
US_economy_employment_employment_D80551_5days
EU_economy_production_composite_pmi_guidance_F30C5D_15days
US_economy_taxes_import_tax_guidance_1BCB45_15days
US_economy_production_manufacturing_index_99D862_5days
DE_economy_balance_of_payments_exports_guidance_E96159_10days
EU_economy_balance_of_payments_imports_E5FA3A
DE_economy_credit_treasury_note_auction_E96159
US_economy_consumption_consumer_confidence_3D4567
US_economy_domestic_product_economic_growth_78D638
EU_economy_public_finance_government_budget_deficit_9DBA1F
US_economy_employment_employment_8F4F19
EU_economy_balance_of_payments_current_account_F30C5D_5days
US_economy_housing_house_prices_3D4567_10days
EU_economy_production_composite_pmi_F30C5D
EU_economy_taxes_import_tax_FB83C5_15days
US_economy_production_manufacturing_pmi_3D4567_10days
EU_economy_balance_of_payments_exports_F30C5D_5days
EU_economy_credit_sovereign_debt_guidance_F30C5D_5days
US_economy_domestic_product_economic_growth_AFD601
US_economy_domestic_product_economic_growth_B5362B
US_economy_interest_rates_interest_rate_guidance_F38FC4_10days
EU_economy_interest_rates_interest_rate_overnight_FB83C5_15days
DE_economy_balance_of_payments_current_account_E96159_15days
US_economy_interest_rates_interest_rate_overnight_870412_10days
US_economy_domestic_product_economic_growth_1395FE
US_economy_interest_rates_interest_rate_guidance_05FCB3_15days
US_economy_taxes_import_tax_1BCB45_15days
EU_economy_taxes_export_tax_E5FA3A_15days
US_economy_taxes_tax_break_guidance_4FE171_15days
US_economy_interest_rates_interest_rate_guidance_D8A196_10days
EU_economy_foreign_exchange_currency_rate_3E823F
US_economy_public_finance_government_budget_guidance_surplus_3D4567
US_economy_employment_employment_guidance_1A5DD8_5days
EU_economy_balance_of_payments_exports_guidance_E5FA3A_5days
US_economy_housing_house_prices_90F423_10days
DE_economy_domestic_product_recession_guidance_E96159_10days
US_economy_public_finance_government_budget_guidance_deficit_FCF6C7_10days
US_economy_taxes_tax_break_guidance_40F27A
Feature
X1X2X3X4X5X6X7X8X9X10X11X12X13X14X15X16X17X18X19X20X21X22X23X24X25X26X27X28X29X30X31X32X33X34X35X36X37X38X39X40X41X42X43X44X45X46X47X48X49X50X51X52X53X54X55X56X57X58X59X60X61X62X63X64X65X66X67X68X69X70
Dates
Weight
0.9
0.6
0.3
0.0


US_economy_balance_of_payments_imports_B048A5
US_economy_balance_of_payments_exports_D618E6
US_economy_employment_employment_F3107B_5days
US_economy_taxes_corporation_tax_1BC945
US_economy_domestic_product_economic_growth_E80F6A_5days
US_economy_domestic_product_gross_domestic_product_99D862_10days
US_economy_production_composite_pmi_B2302B_10days
JP_economy_public_finance_defense_budget_3725B9
US_economy_employment_employment_D80551_15days
US_economy_taxes_tax_break_0852A9
US_economy_public_finance_government_budget_29AE6B
US_economy_housing_house_prices_6CD425
US_economy_domestic_product_real_gross_domestic_product_guidance_3D4567_5days
JP_economy_consumption_consumer_price_index_5ED837_15days
US_economy_taxes_tax_break_guidance_9B2320_15days
US_economy_housing_house_prices_guidance_3D4567
US_economy_balance_of_payments_exports_BAB9F3_10days
US_economy_employment_unemployment_D80551_5days
US_economy_housing_house_prices_0852A9_15days
US_economy_housing_home_sales_new_7B68F3_5days
US_economy_taxes_tax_break_40F27A
JP_economy_taxes_import_tax_E0952F_5days
US_economy_taxes_tax_break_guidance_81660B
US_economy_public_finance_austerity_measures_3D4567
US_economy_housing_house_prices_guidance_FCF6C7_15days
JP_economy_credit_sovereign_debt_purchases_guidance_3725B9_10days
US_economy_taxes_tax_break_guidance_7EE198_15days
JP_economy_consumption_inflation_5ED837
US_economy_employment_employment_EC0929_15days
US_economy_domestic_product_real_gross_domestic_product_3D4567
US_economy_balance_of_payments_imports_5E0D57
JP_economy_public_finance_defense_budget_guidance_3725B9
US_economy_employment_employment_84E55E_10days
US_economy_domestic_product_economic_growth_guidance_34D362_10days
US_economy_consumption_inflation_CFADC4
US_economy_consumption_inflation_guidance_3D4567
US_economy_interest_rates_interest_rate_overnight_F38FC4_5days
US_economy_interest_rates_interest_rate_overnight_870412_5days
US_economy_domestic_product_gross_domestic_product_guidance_D3F80A_10days
US_economy_balance_of_payments_trade_balance_guidance_deficit_3D4567_10days
JP_economy_taxes_tax_break_3725B9
JP_economy_taxes_corporation_tax_guidance_3DF3AE_5days
US_economy_interest_rates_interest_rate_guidance_F38FC4_5days
US_economy_employment_employment_guidance_EFF381
US_economy_employment_employment_78D638
US_economy_balance_of_payments_exports_AFD601_15days
US_economy_domestic_product_economic_growth_guidance_29AE6B_5days
US_economy_credit_treasury_bond_yield_81660B_10days
US_economy_interest_rates_interest_rate_guidance_2CF276_5days
US_economy_employment_employment_FB4AA8
US_economy_interest_rates_interest_rate_overnight_870412_15days
JP_economy_consumption_consumer_spending_3725B9_10days
JP_economy_consumption_retail_sales_3725B9
US_economy_credit_treasury_note_auction_3D4567_10days
US_economy_domestic_product_economic_growth_ADBFDC_5days
US_economy_credit_treasury_bond_yield_1BC945_10days
US_economy_domestic_product_economic_growth_guidance_EFF381
JP_economy_taxes_corporation_tax_guidance_708687
US_economy_consumption_inflation_guidance_3D4567_10days
US_economy_interest_rates_interest_rate_2CF276_15days
JP_economy_consumption_consumer_price_index_9E0439_10days
US_economy_production_manufacturing_index_99D862_5days
US_economy_domestic_product_economic_growth_B2302B_10days
US_economy_interest_rates_interest_rate_81660B
JP_economy_taxes_tax_break_3725B9_5days
US_economy_credit_sovereign_debt_purchases_guidance_870412
US_economy_domestic_product_economic_growth_B2302B_5days
US_economy_taxes_tax_break_D2CE9C
US_economy_employment_employment_guidance_B36437
JP_economy_production_industrial_production_guidance_3725B9_5days
US_economy_credit_sovereign_debt_purchases_BCD073
US_economy_credit_treasury_bond_auction_3D4567_10days
US_economy_credit_sovereign_debt_purchases_81660B
US_economy_domestic_product_gross_domestic_product_guidance_3D4567_10days
US_economy_interest_rates_interest_rate_320C19
US_economy_domestic_product_recession_99D862_5days
US_economy_interest_rates_interest_rate_D90F98_15days
US_economy_balance_of_payments_exports_EC3319_10days
JP_economy_production_producer_price_index_3725B9
US_economy_credit_sovereign_debt_purchases_05FCB3_10days
JP_economy_credit_sovereign_debt_purchases_E0952F
US_economy_domestic_product_economic_growth_8F8F74_5days
US_economy_housing_home_sales_new_guidance_3D4567_15days
US_economy_production_composite_pmi_3D4567_15days
US_economy_balance_of_payments_trade_balance_surplus_1C4199_10days
US_economy_employment_employment_guidance_E80F8A
US_economy_employment_unemployment_guidance_3D4567_5days
US_economy_domestic_product_economic_growth_B36437_5days
US_economy_employment_employment_8F4F19
US_economy_production_manufacturing_index_3EEF5D_5days
US_economy_domestic_product_economic_growth_6EFC36_5days
US_economy_domestic_product_economic_growth_guidance_4EB6DC_10days
US_economy_business_activity_business_confidence_3D4567_10days
JP_economy_interest_rates_interest_rate_overnight_E0952F_5days
US_economy_taxes_tax_break_guidance_D90F98_10days
US_economy_balance_of_payments_current_account_surplus_3D4567
US_economy_employment_employment_6E0D57_10days
US_economy_housing_home_sales_existing_guidance_3D4567_15days
US_economy_consumption_inflation_E80F6A_10days
US_economy_taxes_tax_break_F38FC4
Feature
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 X21 X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35 X36 X37 X38 X39 X40 X41 X42 X43 X44 X45 X46 X47 X48 X49 X50 X51 X52 X53 X54 X55 X56 X57 X58 X59 X60
Dates
Weight
1.0
0.5
0.0

```
## [1] "Done"
```