stocks15_analysis.R

pw

2022-01-24

```r
# This code is an analysis of the data at 15 minute intervals.
# This code includes the following steps:
## STEP 0: LOAD LIBRARIES AND DATA
## STEP 1: PREPOCESSING
### 1.1: Difference and log-transform stock prices
### 1.2: Create morality index
### 1.3: Add lags
### 1.4: Reorder and make time series
### 1.5: Check serial dependencies; ACF & PACF
## STEP 2: DATA EXPLORATION
### 2.1: Histograms
### 2.2: Box & Violine plots
### 2.3: Scatterplots
## STEP 3: FITTING AND EVALUATING MODEL
### 3.1: Removing outliers
### 3.2: Fitting and summarizing models
### 3.3: Model comparisons
### 3.4: Standardized betas
### 3.6: Interaction plot for morality model
### 3.7: Interaction plots for foundation models

knitr::opts_chunk$set(echo = TRUE,
                      collapse = FALSE,
                      warning = FALSE,
                      tidy = TRUE)
options(width=120)


################################################################################
# STEP 0: LOAD DATA

## Read data
setwd("/home/pw/Projects/mfstocks/data/csv/60shift")
stocks15 <- read.table("data15_60shift.csv", header=TRUE, stringsAsFactors=TRUE, sep=",", na.strings="N
## Convert datetime variable to POSIX, which will allow us to create a time series later
stocks15$dt_15min <- as.POSIXct(stocks15$dt_15min, tz="EST")
## Factors are used to represent categorical data in statistical analysis. They are stored as unique in
col15 <- c("season_workday_15min", "season_month_15min",
           "tf1_15min", "tf2_15min", "tf3_15min", "tf4_15min", "tf5_15min", "tf6_15min")
stocks15[col15] <- lapply(stocks15[col15], as.factor)

################################################################################
```

```r
# STEP 1: PREPOCESSING
library(RcmdrMisc)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
## Loading required package: sandwich
```

```r
library(lubridate) # date()
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(dplyr) # %>% function
```

```
##
## Attaching package: 'dplyr'
```

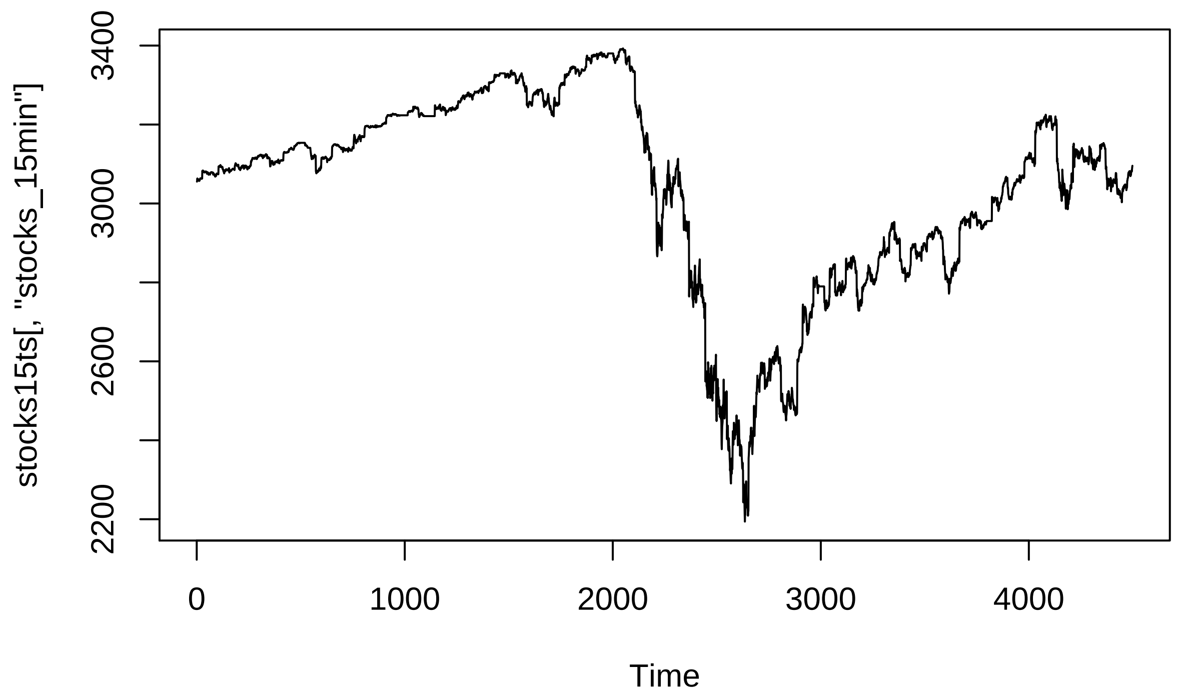
```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

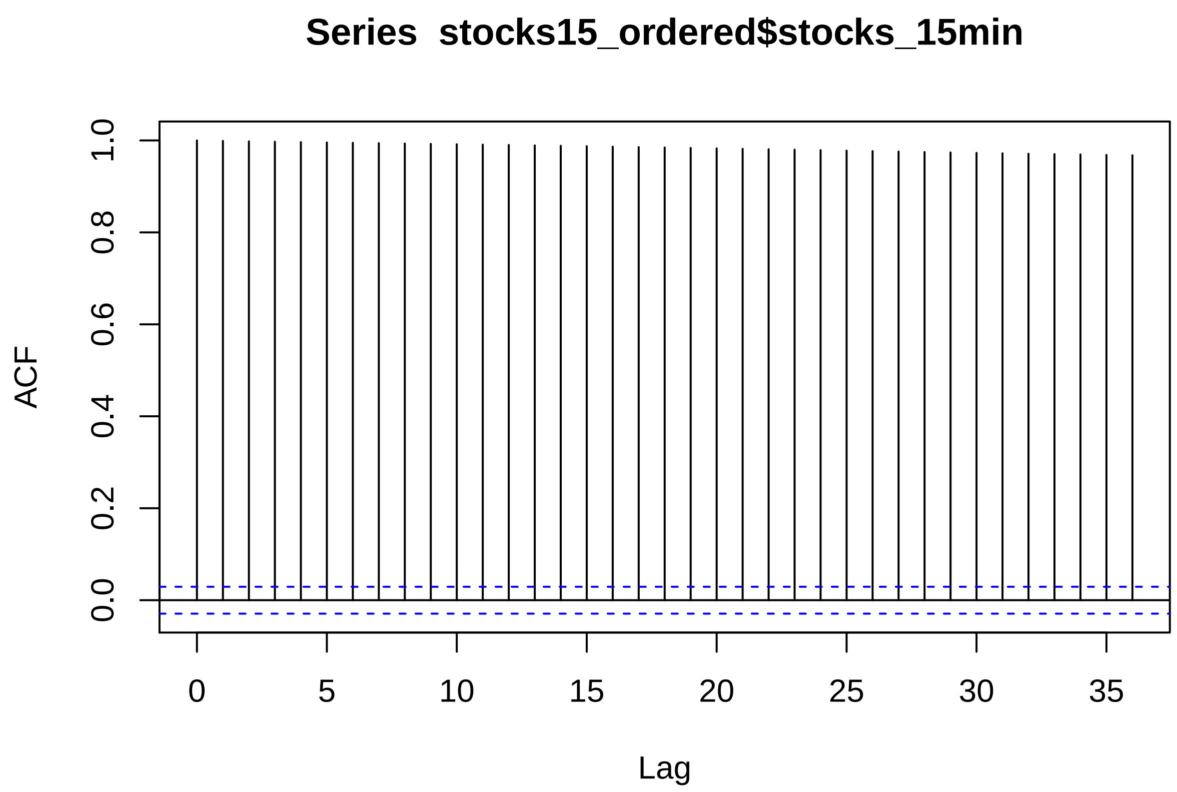
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(reshape2) # melt()

## 1.1:  Difference and log-transform stock prices

### Difference for moving averages
stocks15$stocks_15min_diff = stocks15$stocks_15min %>% diff() %>% append(NA, 0)
### Log then difference for variance stabilization
stocks15$stocks_15min_diff_ln = stocks15$stocks_15min %>% log() %>% diff() %>% append(NA, 0)
### Create variable for counting the days
stocks15$day_count =
  stocks15$dt_15min %>% date() %>% as.integer() -
  stocks15$dt_15min %>% date() %>% as.integer() %>% min() + 1
```

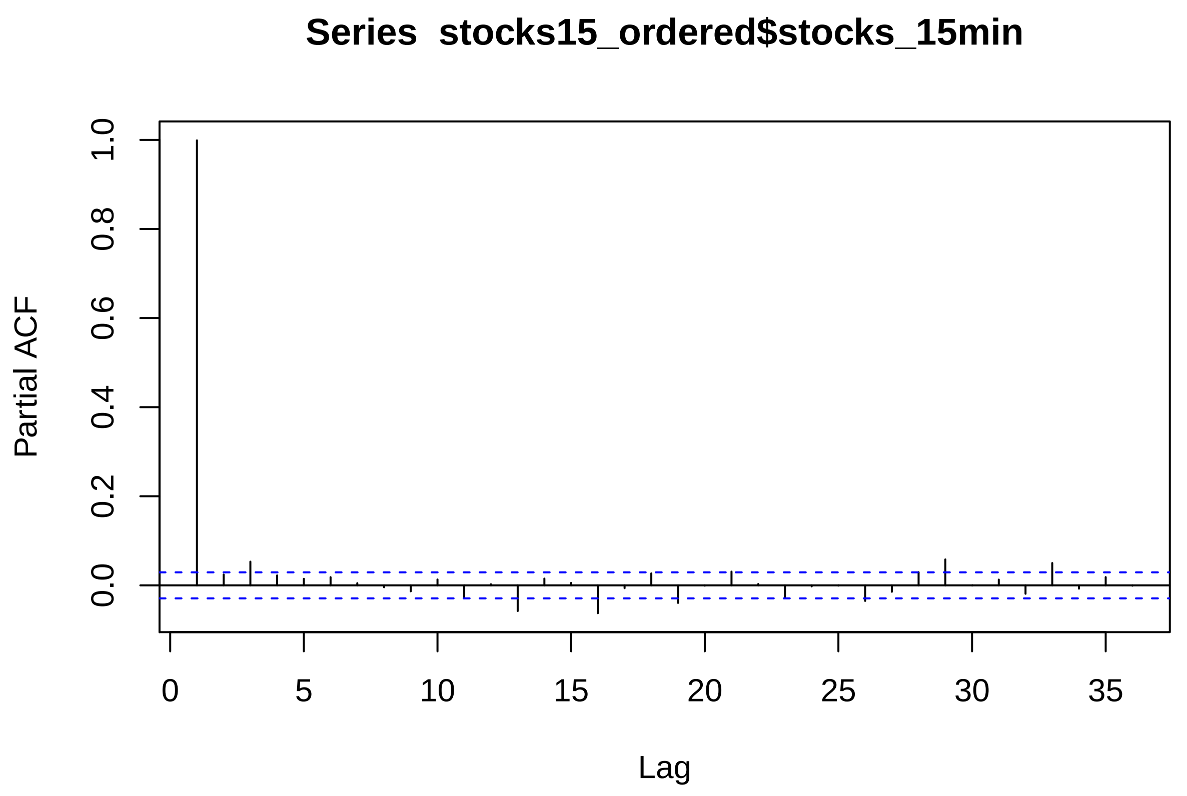
```r
## 1.2: Create morality index from the sentiment-weighted probability of each foundation

stocks15$morality =
  stocks15$care_p_15min*stocks15$care_sent_15min +
  stocks15$fairness_p_15min*stocks15$fairness_sent_15min +
  stocks15$loyalty_p_15min*stocks15$loyalty_sent_15min +
  stocks15$authority_p_15min*stocks15$authority_sent_15min +
  stocks15$sanctity_p_15min*stocks15$sanctity_sent_15min


## 1.3: Add lags

### Lag morality index
stocks15$morality_lag = lag(stocks15$morality)
### Lag foundation probabilities
stocks15$care_p_15min_lag = lag(stocks15$care_p_15min)
stocks15$fairness_p_15min_lag = (stocks15$fairness_p_15min)
stocks15$loyalty_p_15min_lag = lag(stocks15$loyalty_p_15min)
stocks15$authority_p_15min_lag = lag(stocks15$authority_p_15min)
stocks15$sanctity_p_15min_lag = lag(stocks15$sanctity_p_15min)
### Lag foundation sentiments
stocks15$care_sent_15min_lag = lag(stocks15$care_sent_15min)
stocks15$fairness_sent_15min_lag = lag(stocks15$fairness_sent_15min)
stocks15$loyalty_sent_15min_lag = lag(stocks15$loyalty_sent_15min)
stocks15$authority_sent_15min_lag = lag(stocks15$authority_sent_15min)
stocks15$sanctity_sent_15min_lag = lag(stocks15$sanctity_sent_15min)
### Lag sentiment-weighted probabilities of foundations
stocks15$care_lag = stocks15$care_p_15min_lag * stocks15$care_sent_15min_lag
stocks15$fairness_lag = stocks15$fairness_p_15min_lag * stocks15$fairness_sent_15min_lag
stocks15$loyalty_lag = stocks15$loyalty_p_15min_lag * stocks15$loyalty_sent_15min_lag
stocks15$authority_lag = stocks15$authority_p_15min_lag * stocks15$authority_sent_15min_lag
stocks15$sanctity_lag = stocks15$sanctity_p_15min_lag * stocks15$sanctity_sent_15min_lag


## 1.4: Reorder and make time series

### Reorder columns
stocks15_ordered = stocks15[, c(1, 24, 2:4, 16:21, 5, 22, 23, 25, 6:15, 27:36, 26, 37:41)]
### Make time series
stocks15ts = ts(stocks15_ordered)


## 1.5: Check serial dependencies; ACF & PACF

### Plot time series of raw stock prices
plot(stocks15ts[,"stocks_15min"])
```



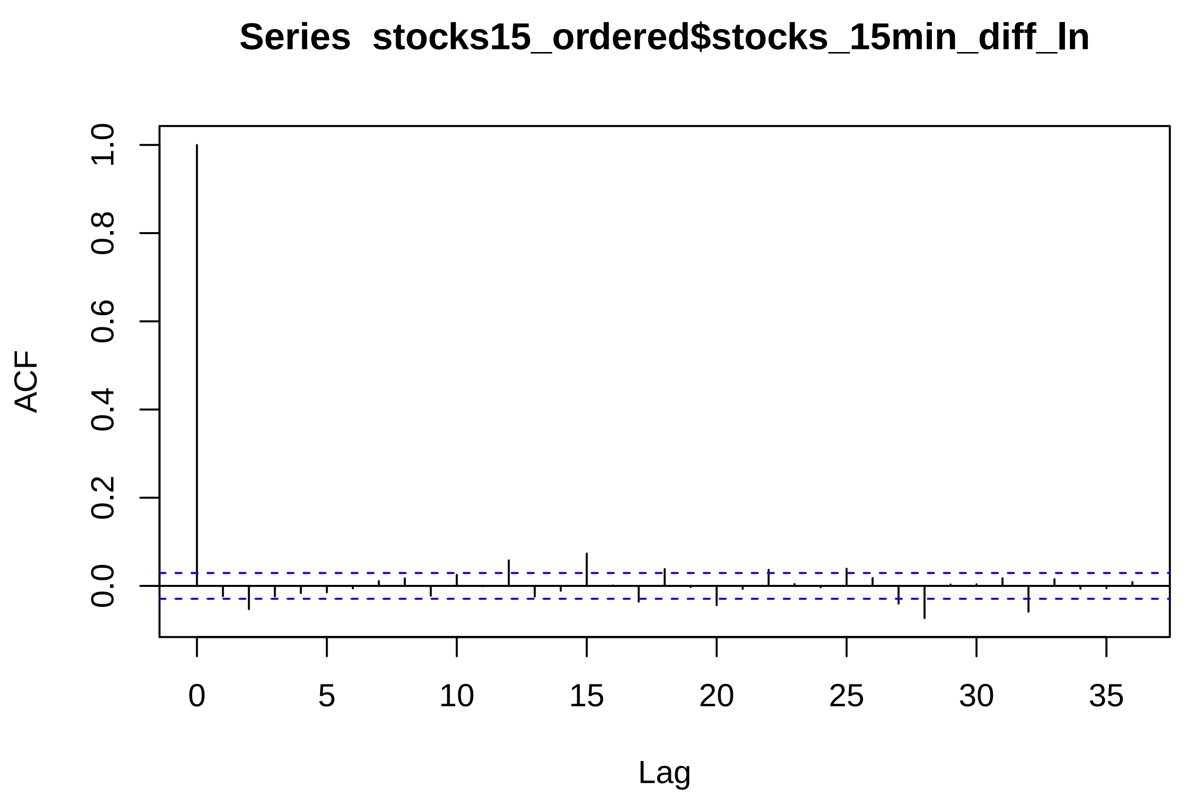
```
### Plot ACF and PACF for non-transformed data
acf(stocks15_ordered$stocks_15min, lag.max = NULL, type = c("correlation"), plot = TRUE, na.action = na
```



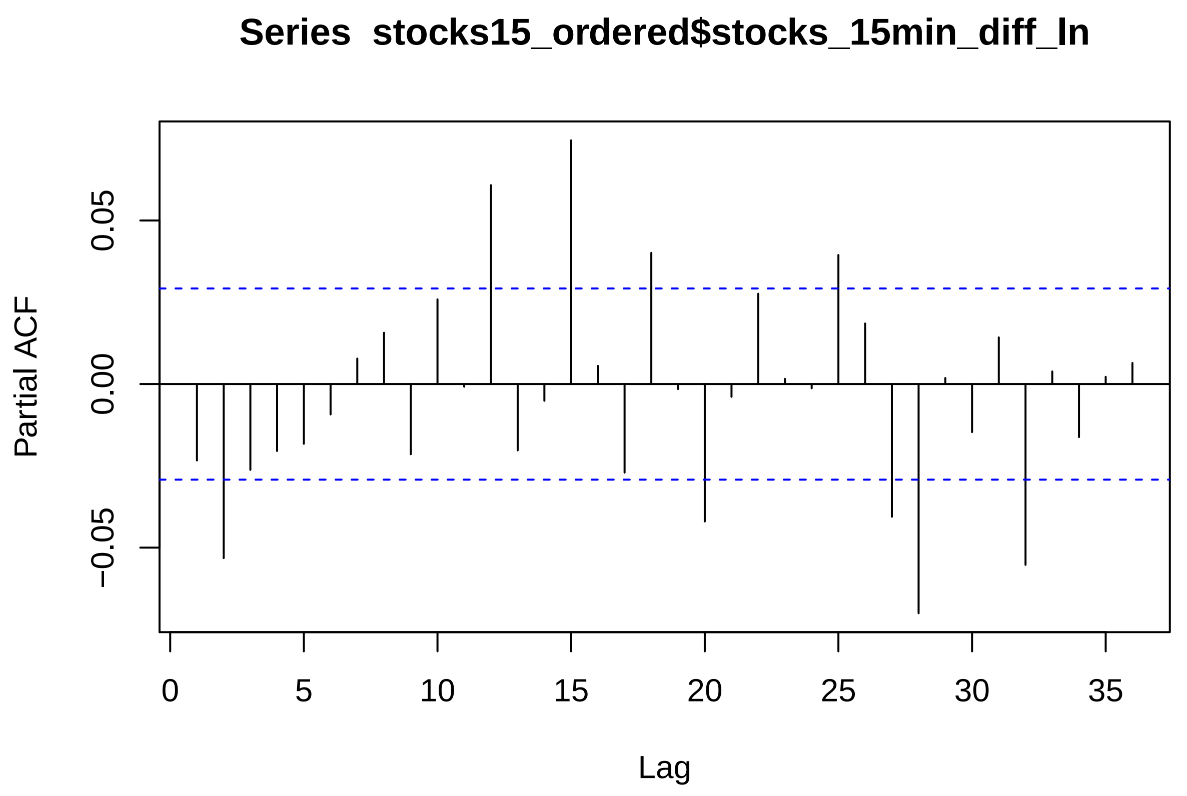
```
acf(stocks15_ordered$stocks_15min, lag.max = NULL, type = c("partial"), plot = TRUE, na.action = na.pas
```



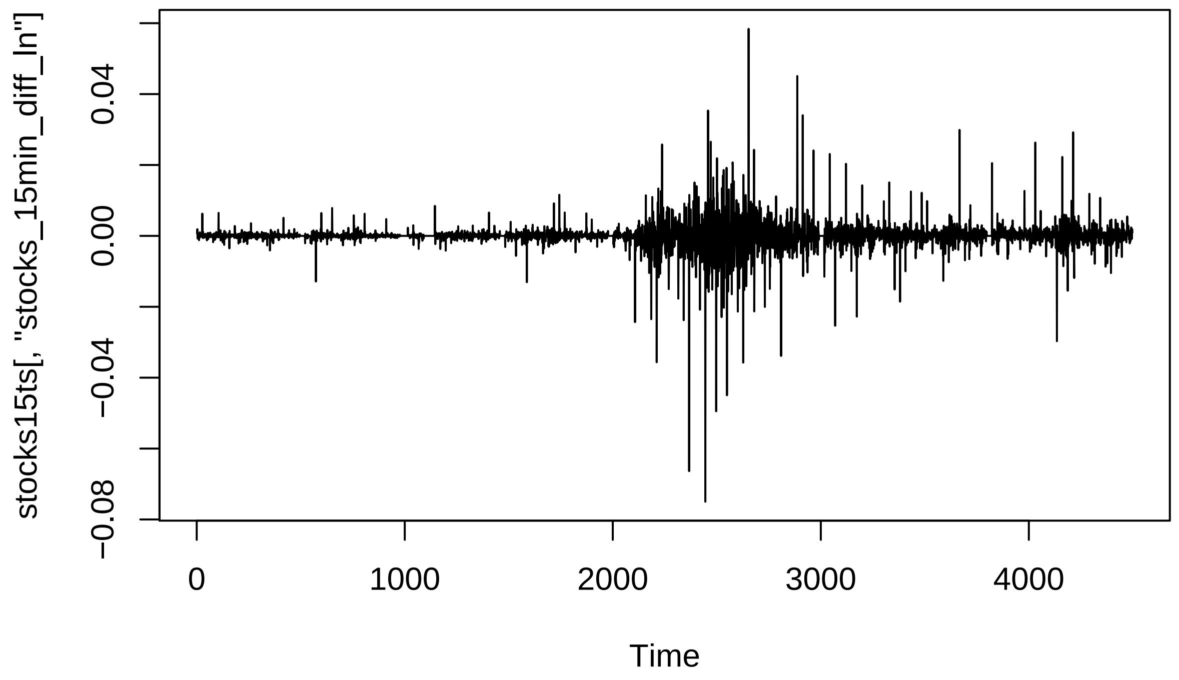
```
### Plot ACF and PACF for transformed data
acf(stocks15_ordered$stocks_15min_diff_ln, lag.max = NULL, type = c("correlation"), plot = TRUE, na.acti
```



```
acf(stocks15_ordered$stocks_15min_diff_ln, lag.max = NULL, type = c("partial"), plot = TRUE, na.action =
```



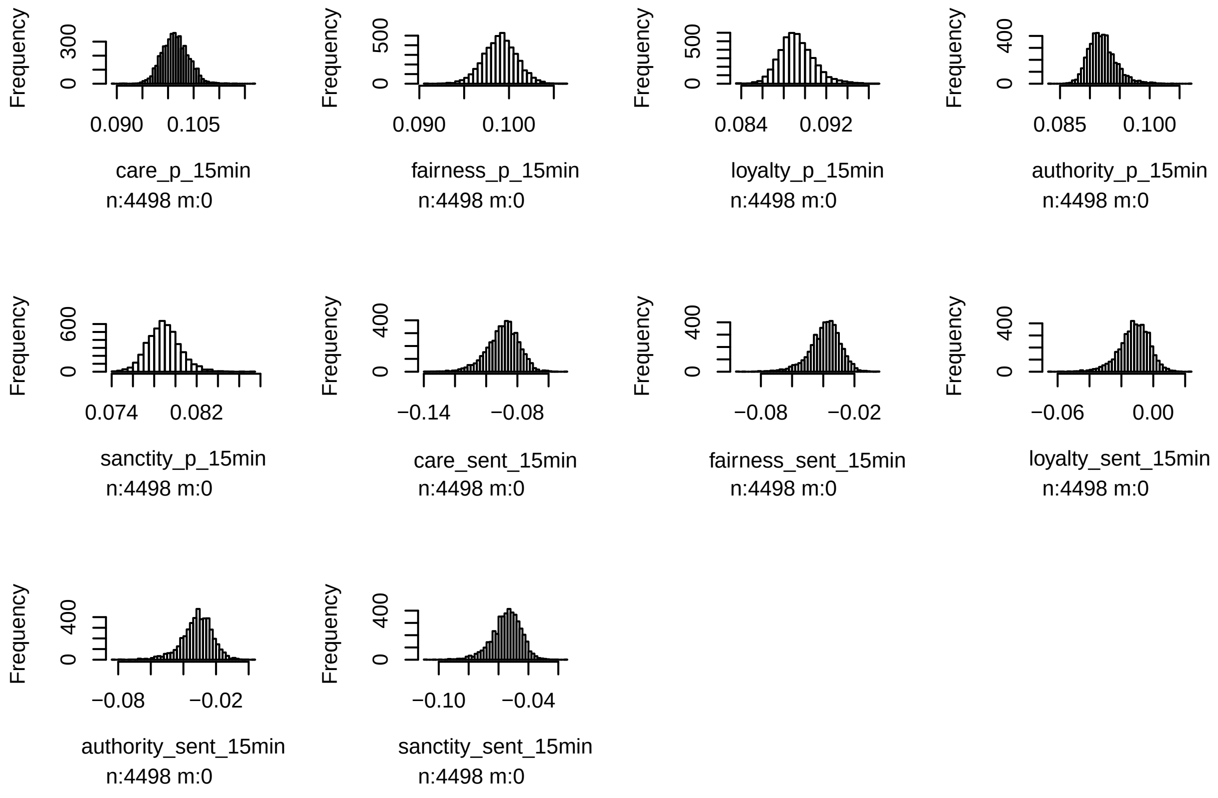
```
### Plot time series of stock prices after differencing and log-transformation
### Mildly unstable variance in contraction and recovery period
plot(stocks15ts[,"stocks_15min_diff_ln"])
```



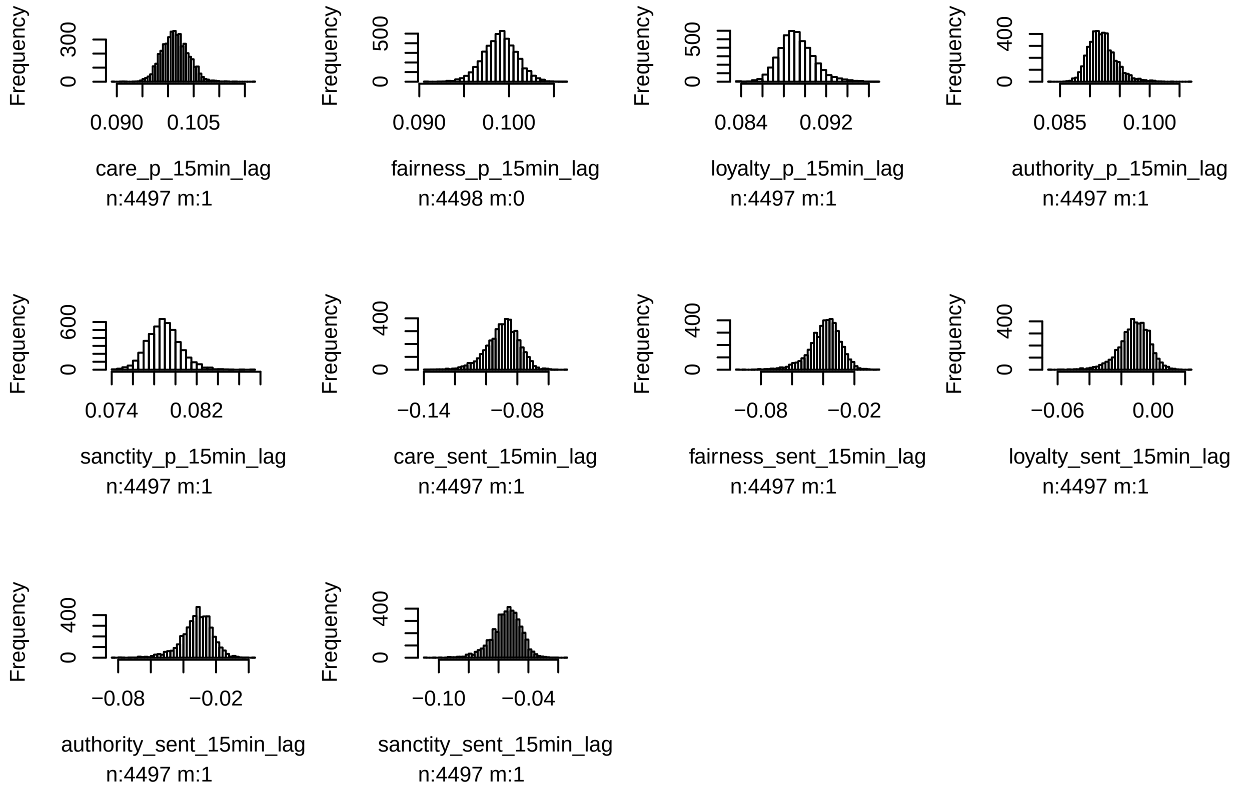
```
################################################################################
# STEP 2: DATA EXPLORATION
library(ggplot2)
library(gridExtra) # gridExtra
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

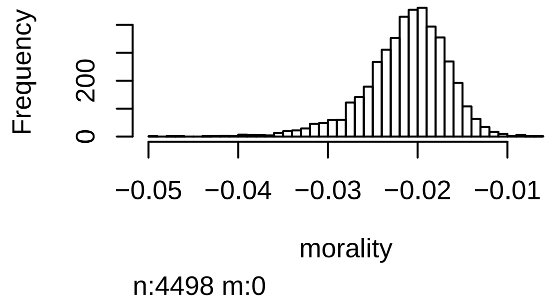
```
library(ggpubr)
theme_set(theme_pubclean())

## 2.1: Frequency histograms

### Probabilities and sentiments of foundations
hist(stocks15_ordered[16:25])
```

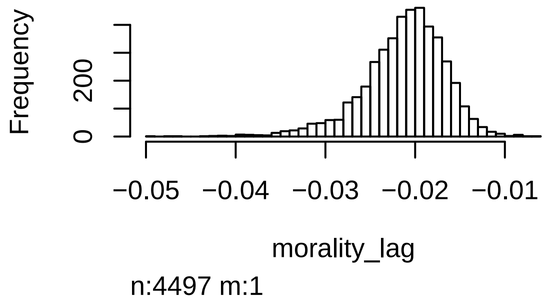


```r
### Lagged probabilities and sentiments of foundations
hist(stocks15_ordered[26:35])
```

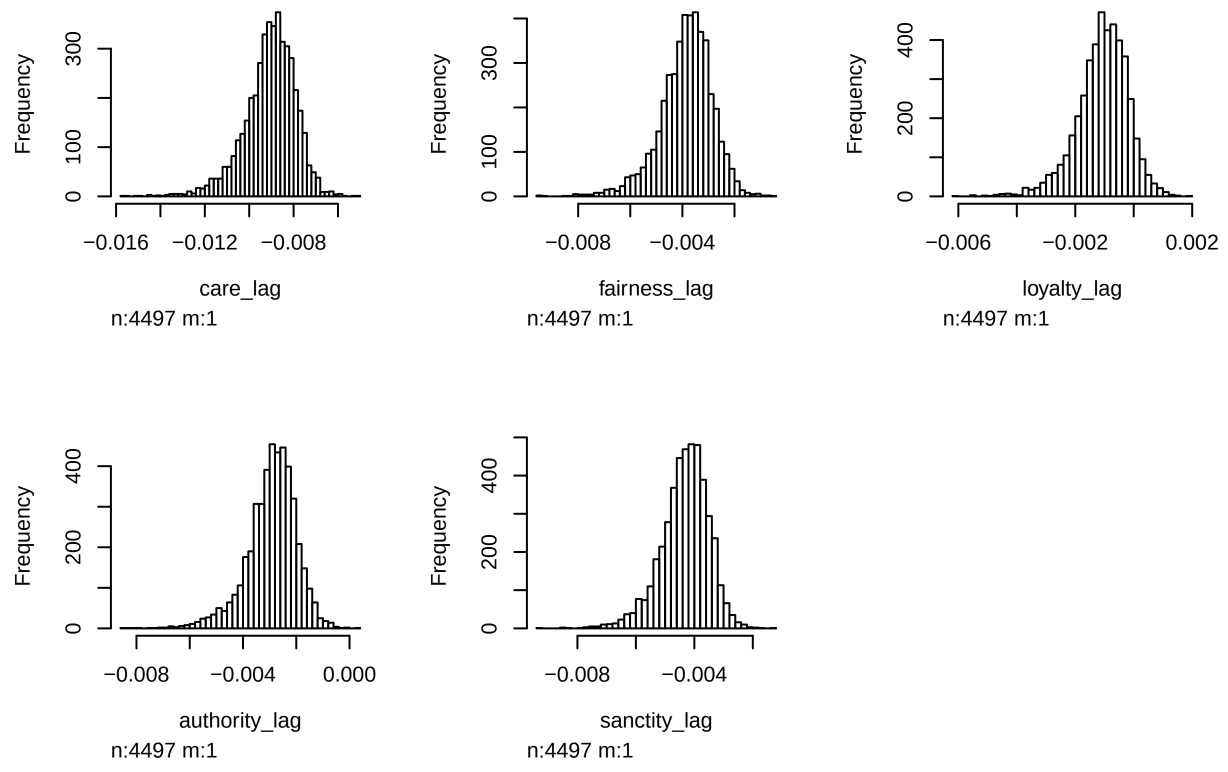


```
### Morality and lagged morality
hist(stocks15_ordered[, c(15,36)])
```

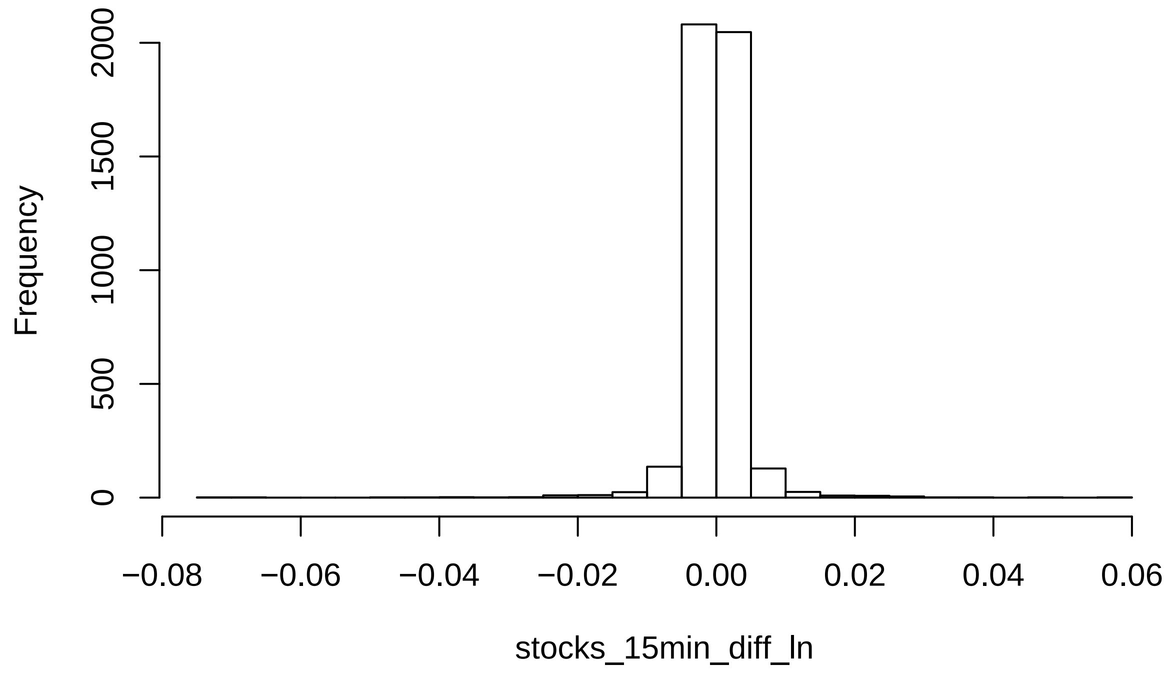




```
### Sentiment-weighted probabilities of foundations lagged
hist(stocks15_ordered[37:41])
```



```
### Transformed stock prices
hist(stocks15_ordered[14])
```

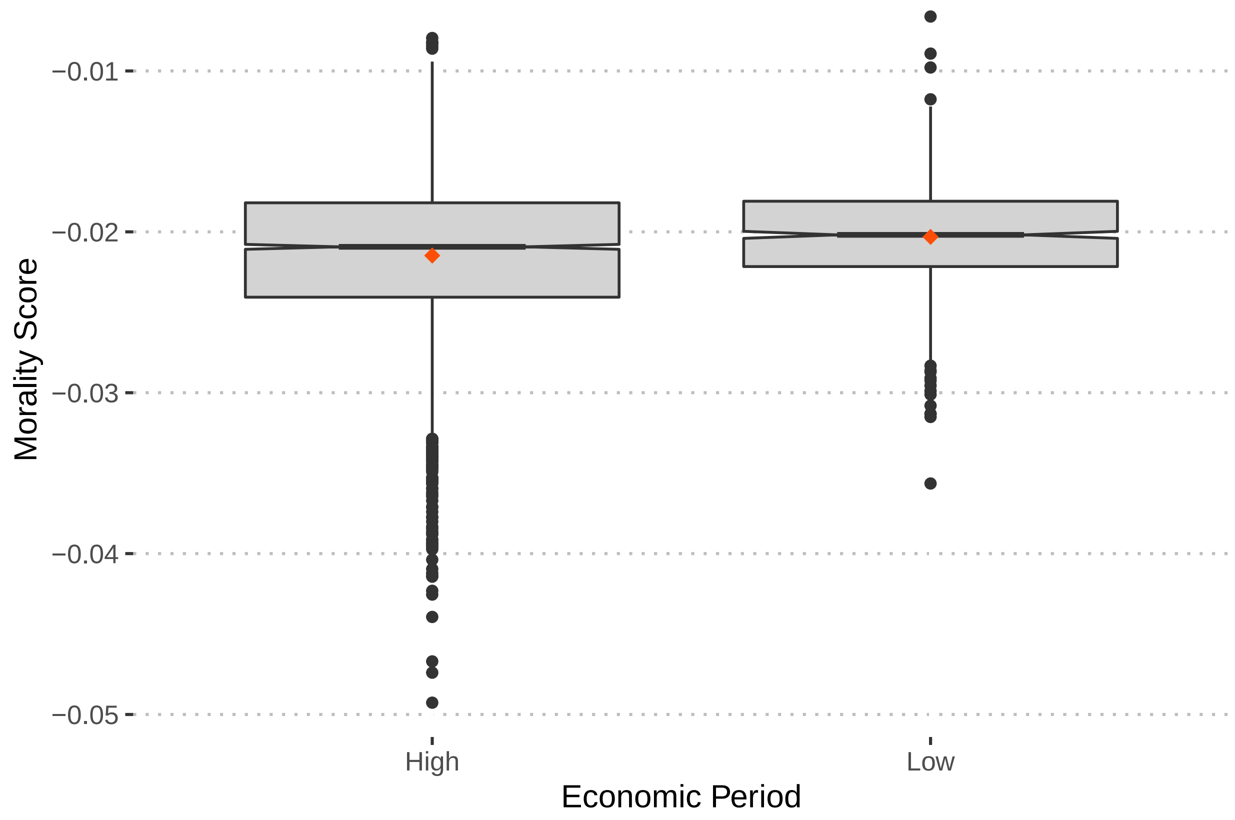


n:4497 m:1

```
## 2.2: Box & Violine plots

### Define plot functions
#### BOXPLOT
plot_boxplot <- function(input, x_labels) {
  return(input +
           geom_boxplot(notch = TRUE, fill = "lightgray") +
           stat_summary(fun.y = mean, geom = "point",shape = 18, size = 2.5, color = "#FC4E07") +
           labs(x="Economic Period", y = "Morality Score") +
           scale_x_discrete(labels=x_labels) )}
#### VIOLIN
plot_violin <- function(input, x_labels) {
  return(input +
           geom_violin(trim = FALSE) +
           stat_summary(fun.data = "mean_sdl", fun.args = list(mult = 1), geom = "pointrange", color =
           labs(x="Economic Period", y = "Morality Score") +
           scale_x_discrete(labels=x_labels) )}
#### DOUBLE VIOLIN
plot_violin2 <- function(input, x_labels) {
  return(input +
           geom_violin(aes(color = tf2_15min), trim = FALSE,position = position_dodge(0.9)) +
           geom_boxplot(aes(color = tf2_15min), width = 0.15, position = position_dodge(0.9)) +
           scale_color_manual(labels = c("High", "Low"), values = c("#00AFBB","#E7B800")) +
           scale_x_discrete(labels=x_labels) +
           theme(legend.position="right") +
           labs(x="Moral Foundations", y = "Foundation Score",colour="Economic Period") )}
```

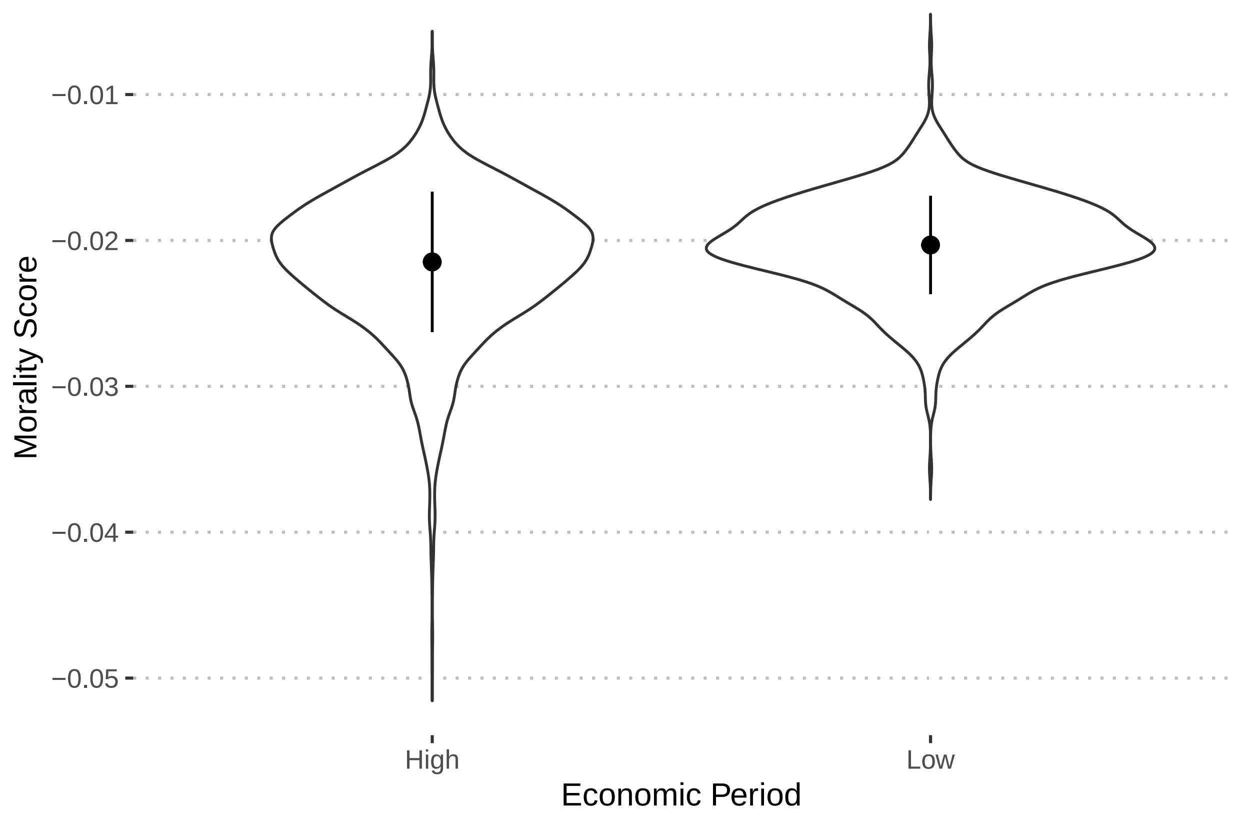
```r
### Plot functions
### MORALITY
e <- ggplot(stocks15, aes(x = tf2_15min, y = morality_lag))
plot_boxplot(e, c("High","Low"))
```

```
## Warning: 'fun.y' is deprecated. Use 'fun' instead.

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

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



```r
plot_violin(e, c("High","Low"))
```

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

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



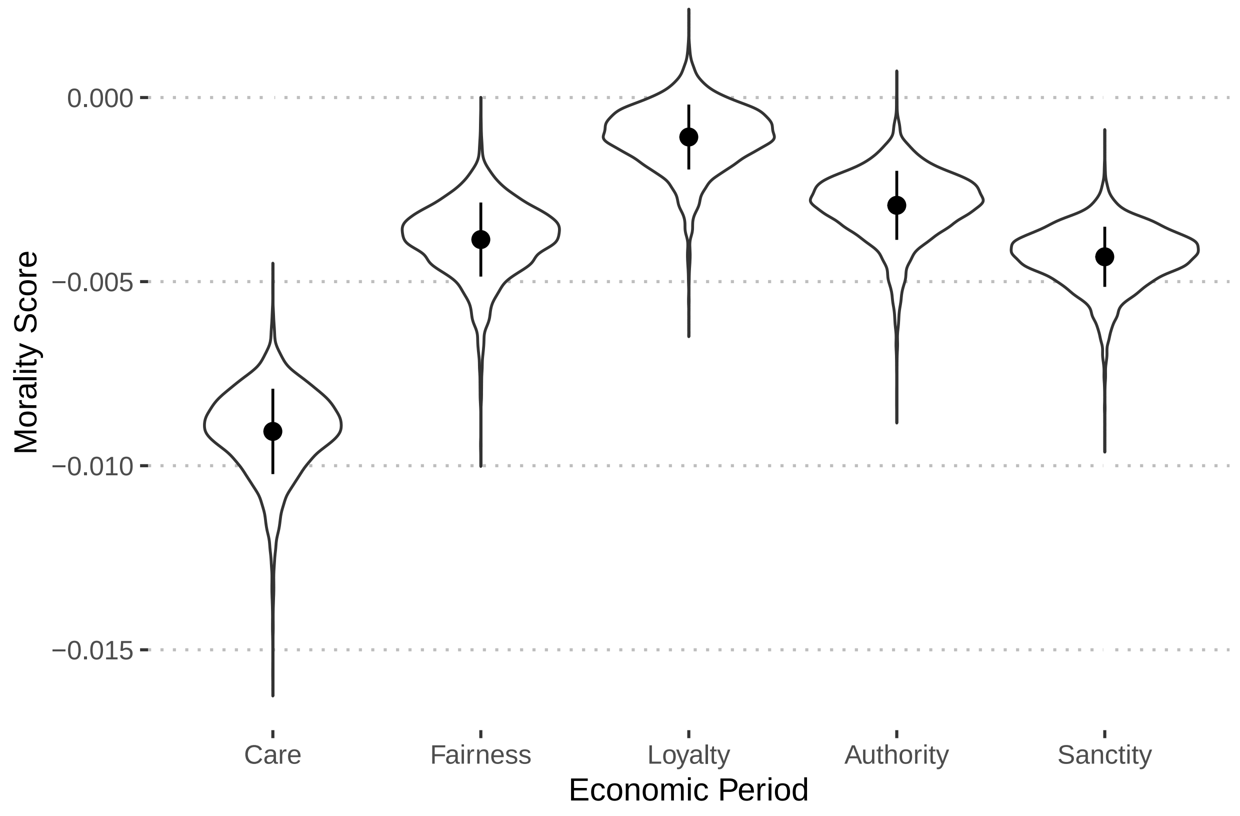
```r
### FOUNDATIONS
stocks15_foundations_long = melt(stocks15_ordered[, c(7,37:41)])
```

```
## Using tf2_15min as id variables
```

```r
e1 <-  ggplot(stocks15_foundations_long, aes(x = variable, y = value))
plot_violin(e1, c("Care","Fairness","Loyalty","Authority", "Sanctity"))
```

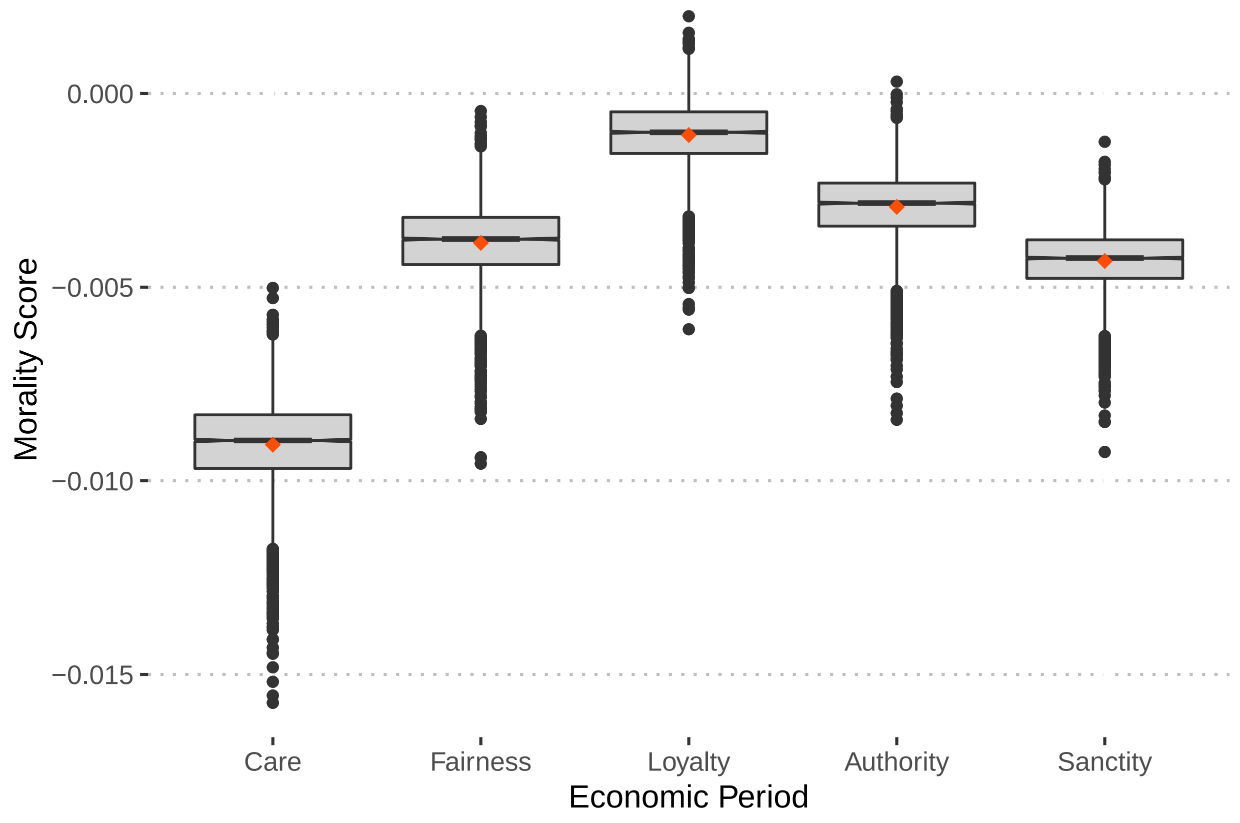
```
## Warning: Removed 5 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 5 rows containing non-finite values (stat_summary).
```



```
plot_boxplot(e1, c("Care","Fairness","Loyalty","Authority", "Sanctity"))
```

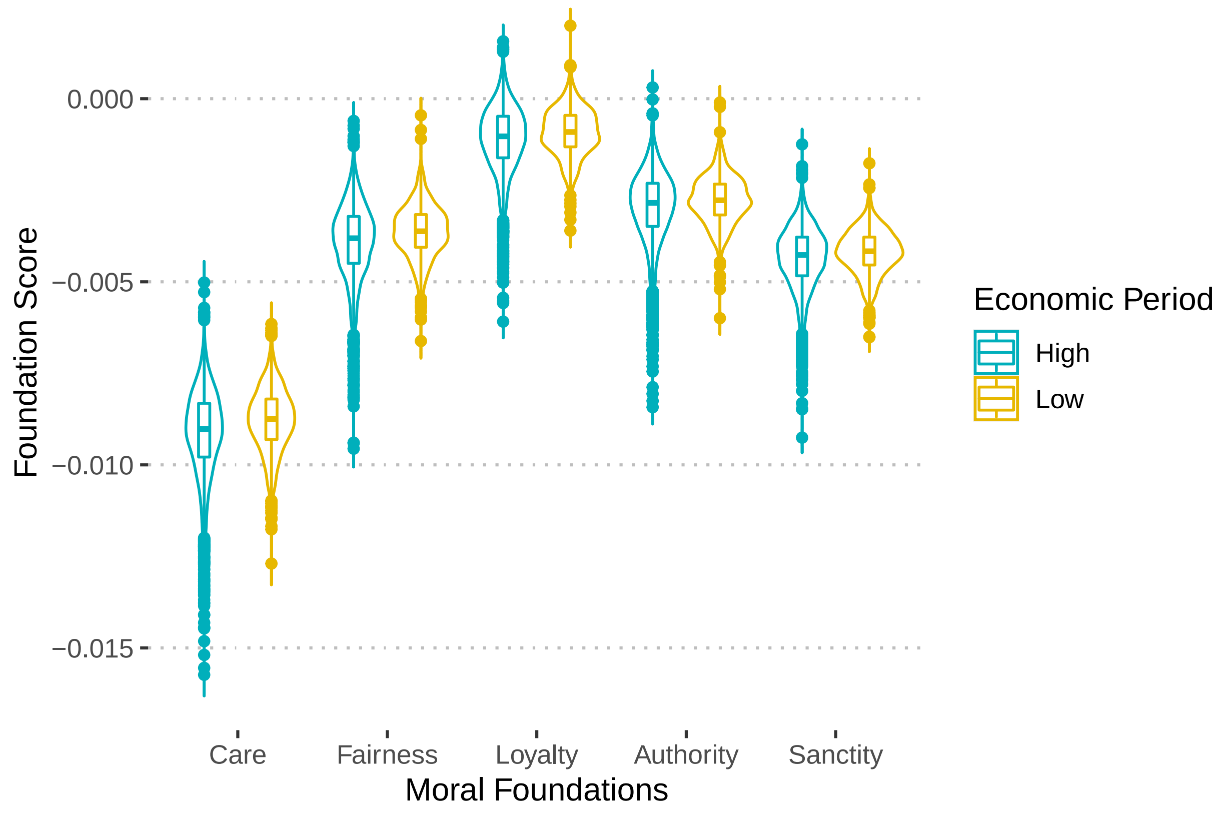
```
## Warning: ‘fun.y‘ is deprecated. Use ‘fun‘ instead.

## Warning: Removed 5 rows containing non-finite values (stat_boxplot).

## Warning: Removed 5 rows containing non-finite values (stat_summary).
```



```
plot_violin2(e1, c("Care","Fairness","Loyalty","Authority", "Sanctity"))
```

```
## Warning: Removed 5 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```



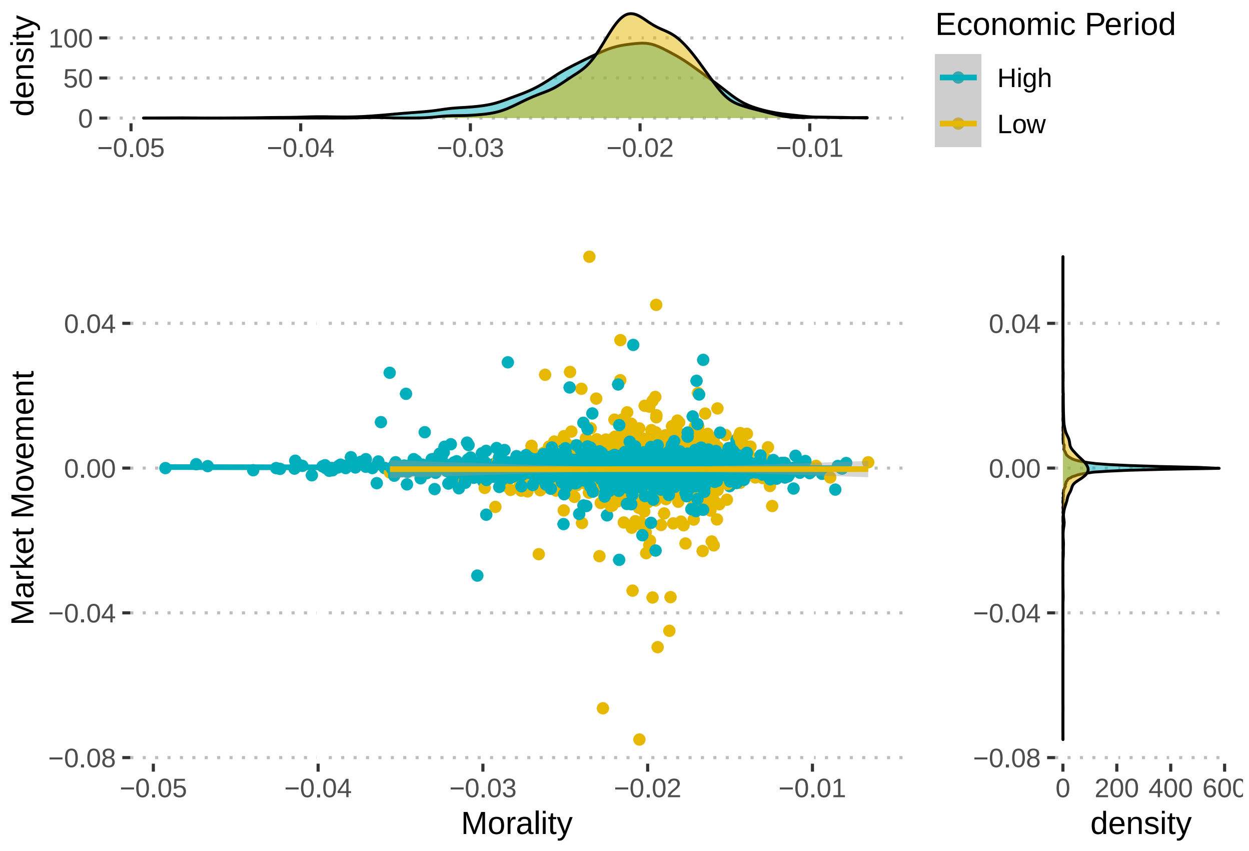
```
## 2.3: Scatterplots

### Define plot functions
#### SCATTERPLOT
plot_scatter <- function(input, x_label) {
  return(ggplot(stocks15_ordered, aes(input, stocks_15min_diff_ln, color=tf2_15min)) +
           geom_point() +
           geom_smooth(method=lm) +
           scale_color_manual(labels = c("High", "Low"), values = c('#00AFBB','#E7B800')) +
           theme(legend.position=c(0,1), legend.justification=c(0,1)) +
           labs(x=x_label, y = "Market Movement", color = "Economic Period") )}
#### X DENSITY
plot_xdensity <- function(input) {
  return(ggplot(stocks15_ordered, aes(input, fill=tf2_15min)) +
           geom_density(alpha=.5) +
           scale_fill_manual(values = c('#00AFBB','#E7B800')) +
           theme(legend.position = "none")  +
           labs(x = "") )}
#### Y DENSITY
plot_ydensity <- function() {
  return(ggplot(stocks15_ordered, aes(stocks_15min_diff_ln, fill=tf2_15min)) +
           geom_density(alpha=.5) +
           scale_fill_manual(values = c('#00AFBB','#E7B800')) +
           theme(legend.position = "none")  +
           labs(x = "") )}
### Plot functions
```

```r
### MORALITY
scatterPlot <- plot_scatter(stocks15_ordered$morality_lag, "Morality")
scatterPlot2 <- scatterPlot + theme(legend.position = "none")
xdensity <- plot_xdensity(stocks15_ordered$morality_lag)
ydensity <- plot_ydensity() + coord_flip()
legend <- get_legend(scatterPlot)
```

```
## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).
```

```r
grid.arrange(xdensity, legend, scatterPlot2, ydensity, ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4,
```

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

## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).

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

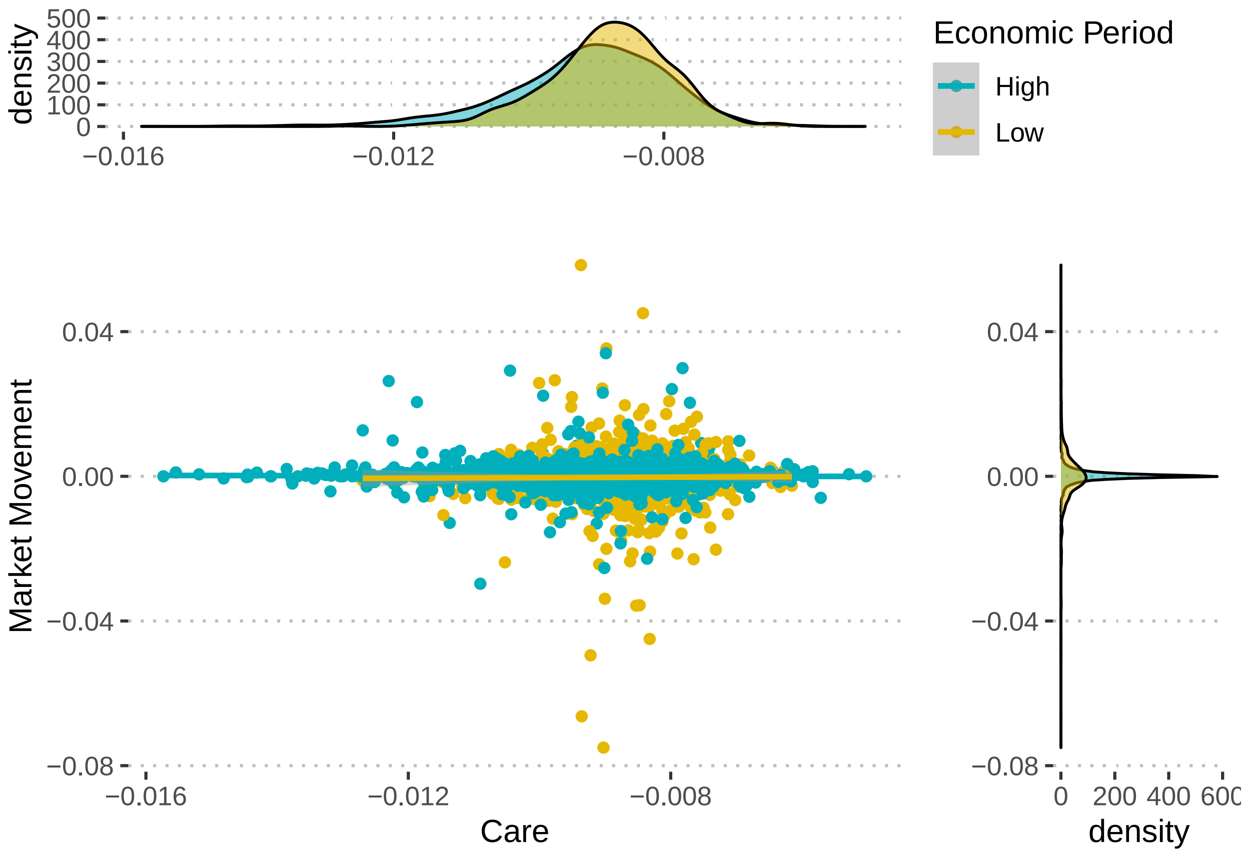


```
### FOUNDATIONS
#### Care
scatterPlot <- plot_scatter(stocks15_ordered$care_lag, "Care")
scatterPlot2 <- scatterPlot + theme(legend.position = "none")
xdensity <- plot_xdensity(stocks15_ordered$care_lag)
ydensity <- plot_ydensity() + coord_flip()
legend <- get_legend(scatterPlot)
```

```
## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).
```

```
grid.arrange(xdensity, legend, scatterPlot2, ydensity, ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4,
```

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

## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).

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

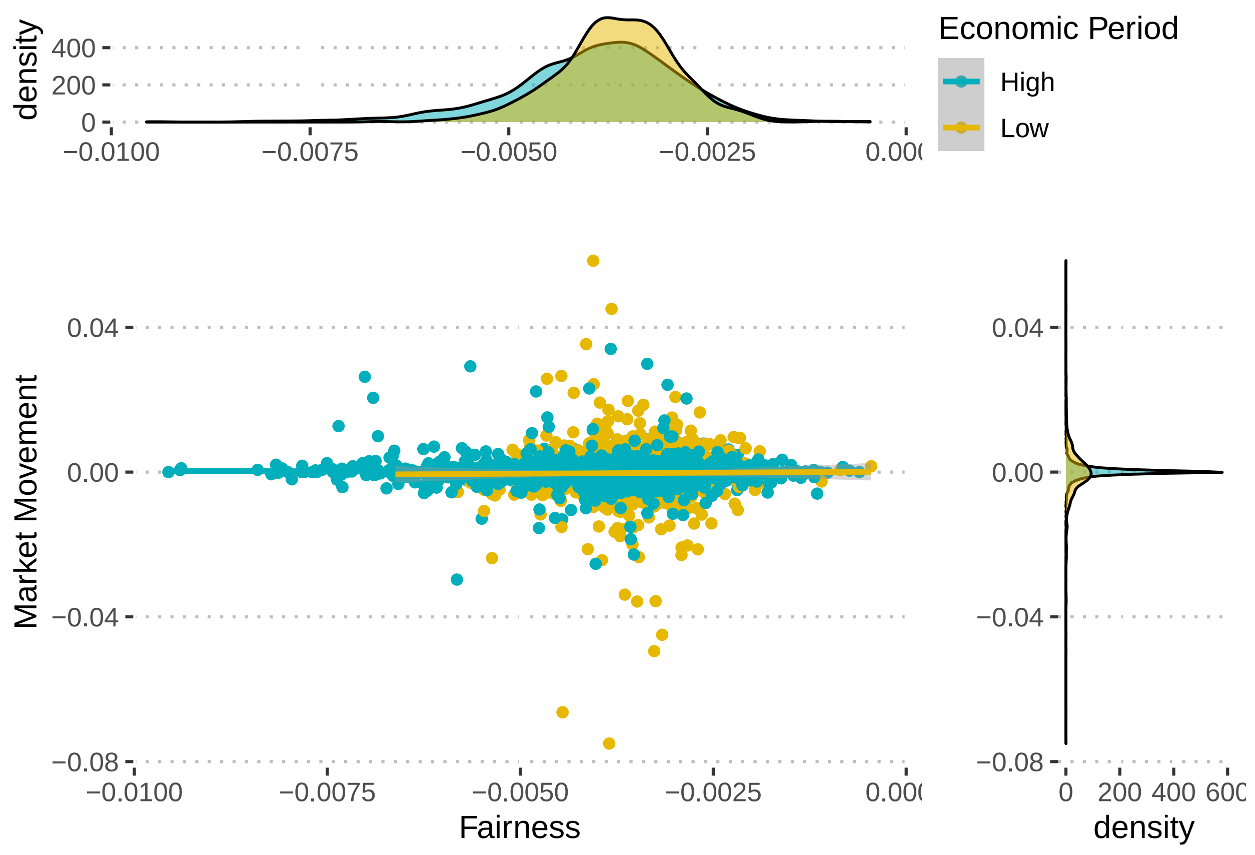


```
#### Fairness
scatterPlot <- plot_scatter(stocks15_ordered$fairness_lag, "Fairness")
scatterPlot2 <- scatterPlot + theme(legend.position = "none")
xdensity <- plot_xdensity(stocks15_ordered$fairness_lag)
ydensity <- plot_ydensity() + coord_flip()
legend <- get_legend(scatterPlot)
```

```
## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).
```

```
grid.arrange(xdensity, legend, scatterPlot2, ydensity, ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4,
```

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

## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).

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

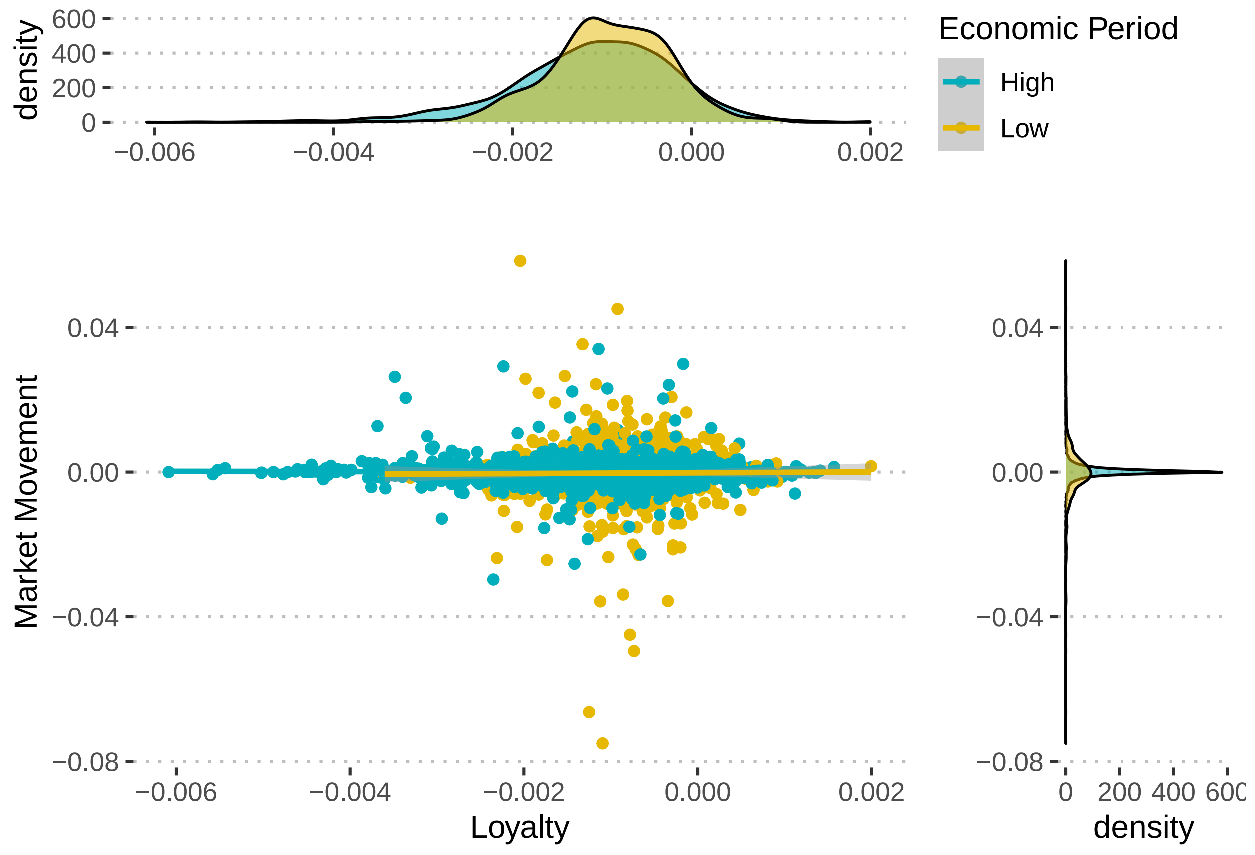


```r
#### Loyalty
scatterPlot <- plot_scatter(stocks15_ordered$loyalty_lag, "Loyalty")
scatterPlot2 <- scatterPlot + theme(legend.position = "none")
xdensity <- plot_xdensity(stocks15_ordered$loyalty_lag)
ydensity <- plot_ydensity() + coord_flip()
legend <- get_legend(scatterPlot)
```

```
## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).
```

```r
grid.arrange(xdensity, legend, scatterPlot2, ydensity, ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4,
```

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

## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).

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

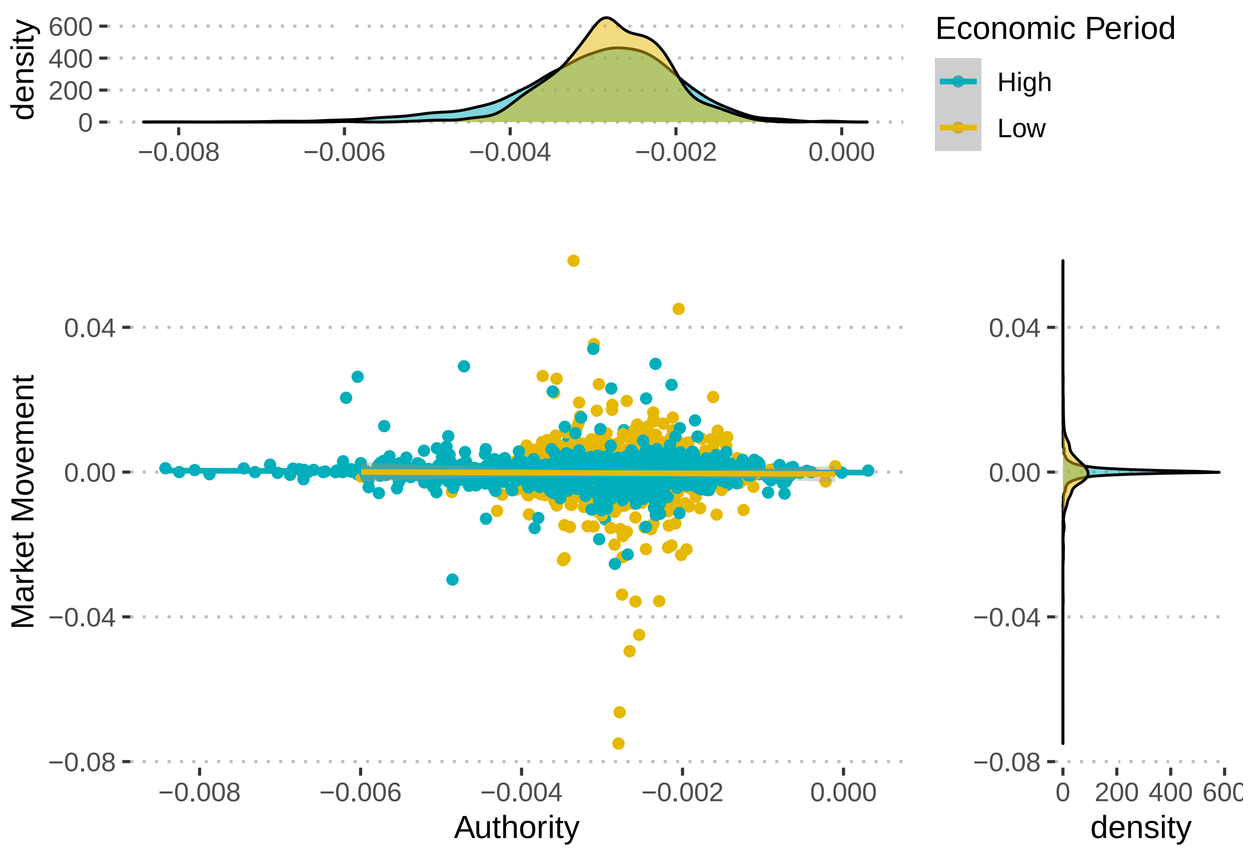


```
#### Authority
scatterPlot <- plot_scatter(stocks15_ordered$authority_lag, "Authority")
scatterPlot2 <- scatterPlot + theme(legend.position = "none")
xdensity <- plot_xdensity(stocks15_ordered$authority_lag)
ydensity <- plot_ydensity() + coord_flip()
legend <- get_legend(scatterPlot)
```

```
## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).
```

```
grid.arrange(xdensity, legend, scatterPlot2, ydensity, ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4,
```

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

## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).

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

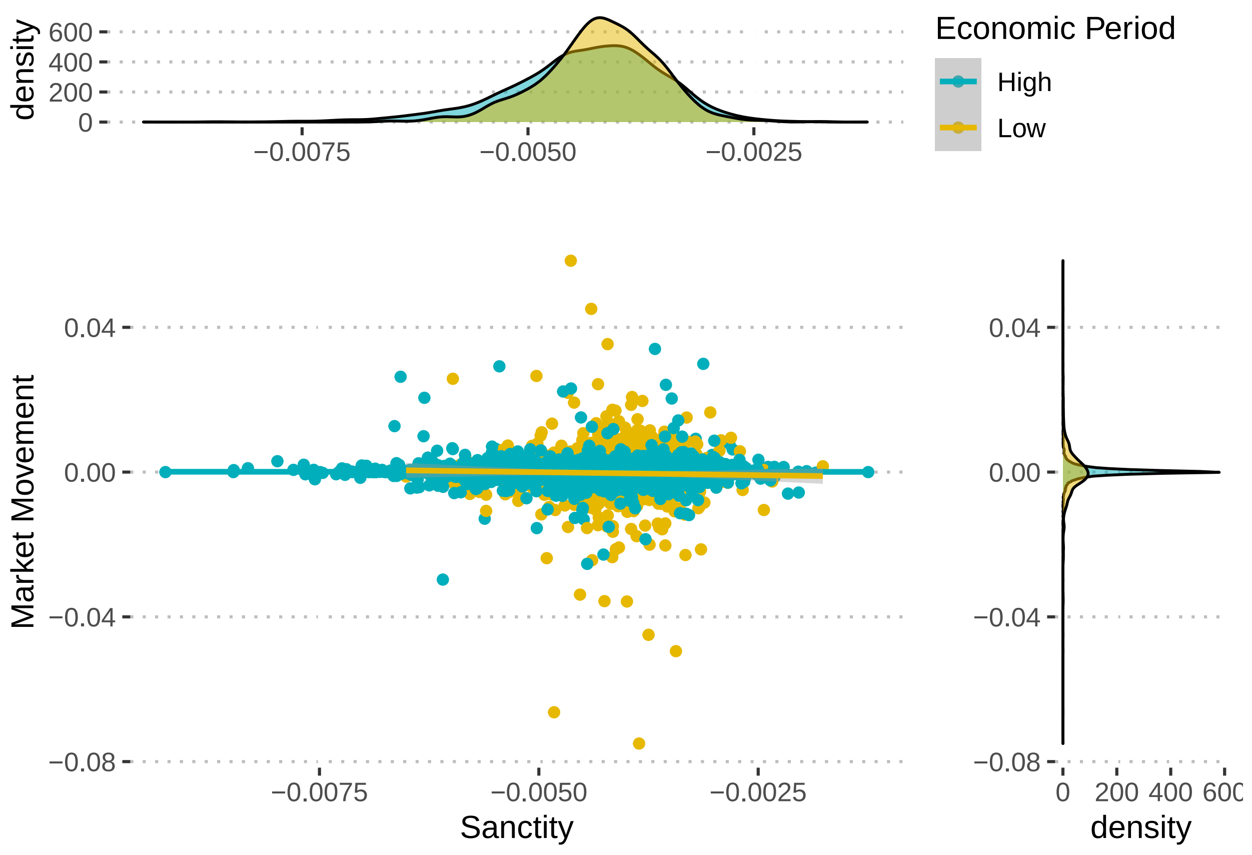


```
#### Sanctity
scatterPlot <- plot_scatter(stocks15_ordered$sanctity_lag, "Sanctity")
scatterPlot2 <- scatterPlot + theme(legend.position = "none")
xdensity <- plot_xdensity(stocks15_ordered$sanctity_lag)
ydensity <- plot_ydensity() + coord_flip()
legend <- get_legend(scatterPlot)
```

```
## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).
```

```
grid.arrange(xdensity, legend, scatterPlot2, ydensity, ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4,
```

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

## `geom_smooth()` using formula 'y ~ x'

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

## Warning: Removed 1 rows containing missing values (geom_point).

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



```
################################################################################
# STEP 3: FITTING AND EVALUATING MODEL
library(jtools) # summ()
library(lme4) # lme models
```

```
## Loading required package: Matrix
```

```
library(interactions) # interact()


## 3.1: Removing outliers
### Define function
outlier_rm_IQR <- function(data, df_str, col_str, threshold){
  data <- na.omit(data)
  Q <- quantile(data, probs=c(.25, .75), na.rm = FALSE) # 25/75 QUANTILES AFTER REMOVING ROW 1 (NA row)
  iqr <- IQR(data) # IQR AFTER REMOVING ROW 1 (NA row)
  upper <- Q[2]+threshold*iqr # Upper Range for outliers
  lower <- Q[1]-threshold*iqr # Lower Range for outliers
  df <- get(df_str)
  column <- get(df_str)[col_str]
  a <- subset.data.frame(df, column > lower)
  b <- subset.data.frame(df, column < upper)
  return(intersect(a,b))}
### Run function
stocks15_outrm_moralityonly <- outlier_rm_IQR(stocks15_ordered$morality_lag[-1], "stocks15_ordered", "mo


## 3.2: Fitting and summarizing models

### Linear model for morality
stocks15.model.lm.outrm <- lm(
  stocks_15min_diff_ln ~ tf2_15min * morality_lag,
  data = stocks15_outrm_moralityonly)
Anova(stocks15.model.lm.outrm, type="III", test="F")
```

```
## Anova Table (Type III tests)
##
## Response: stocks_15min_diff_ln
##                          Sum Sq   Df F value Pr(>F)
## (Intercept)            0.000001    1  0.0530 0.8179
## tf2_15min              0.000001    1  0.0489 0.8249
## morality_lag           0.000003    1  0.2048 0.6509
## tf2_15min:morality_lag 0.000000    1  0.0273 0.8689
## Residuals              0.075473 4467
```

```
### Linear model for foundations
stocks15.model.lm.foundations.outrm <- lm(
  stocks_15min_diff_ln ~ tf2_15min * (care_lag +fairness_lag +loyalty_lag +authority_lag +sanctity_lag)
  data = stocks15_outrm_moralityonly)
Anova(stocks15.model.lm.foundations.outrm, type="III", test="F")
```

```
## Anova Table (Type III tests)
```

```
##
## Response: stocks_15min_diff_ln
##                           Sum Sq   Df  F value    Pr(>F)
## (Intercept)             0.000000    1   0.0217   0.88284
## tf2_15min               0.000003    1   0.1583   0.69070
## care_lag                0.000001    1   0.0388   0.84382
## fairness_lag            0.000012    1   0.7186   0.39665
## loyalty_lag             0.000014    1   0.8378   0.36009
## authority_lag           0.000036    1   2.1556   0.14212
## sanctity_lag            0.000031    1   1.8454   0.17439
## tf2_15min:care_lag      0.000037    1   2.1692   0.14087
## tf2_15min:fairness_lag  0.000052    1   3.0856   0.07906 .
## tf2_15min:loyalty_lag   0.000010    1   0.5740   0.44872
## tf2_15min:authority_lag 0.000006    1   0.3494   0.55448
## tf2_15min:sanctity_lag  0.000286    1  16.9971 3.812e-05 ***
## Residuals               0.075070 4459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### Mixed-effect null model
stocks15.model.lme.null.outrm <-lmer(
  stocks_15min_diff_ln ~ 1 + (1|day_count),
  data = stocks15_outrm_moralityonly, REML=TRUE)
Anova(stocks15.model.lme.null.outrm, type="III", test="F")
```

```
## Analysis of Deviance Table (Type III Wald F tests with Kenward-Roger df)
##
## Response: stocks_15min_diff_ln
##                  F Df Df.res Pr(>F)
## (Intercept) 3e-04  1 171.84 0.9868
```

```
### Mixed-effect model for morality
stocks15.model.lme.outrm <- lmer(
  stocks_15min_diff_ln ~ (1 + season_intraday_15min + tf2_15min*morality_lag + (1 + season_intraday_15mi
  data = stocks15_outrm_moralityonly, REML = TRUE)
```

```
## Warning: Some predictor variables are on very different scales: consider rescaling

## boundary (singular) fit: see ?isSingular
```

```
Anova(stocks15.model.lme.outrm, type="III", test="F")
```

```
## Analysis of Deviance Table (Type III Wald F tests with Kenward-Roger df)
##
## Response: stocks_15min_diff_ln
##                              F Df Df.res Pr(>F)
## (Intercept)             0.2288  1 1308.1 0.6325
## season_intraday_15min   0.0006  1  174.3 0.9801
## tf2_15min               0.0843  1 3028.1 0.7715
## morality_lag            0.3969  1 2316.8 0.5288
## tf2_15min:morality_lag 0.0173  1 3387.7 0.8955
```

```
### Mixed-effect model for foundations
stocks15.model.lme.foundations.outrm <- lmer(
  stocks_15min_diff_ln ~  (1 + season_intraday_15min +
                              tf2_15min*(care_lag + fairness_lag + loyalty_lag + authority_lag + sancti
                              ( 1 + season_intraday_15min | day_count)),
  data = stocks15_outrm_moralityonly, REML = TRUE)
```

```
## Warning: Some predictor variables are on very different scales: consider rescaling

## boundary (singular) fit: see ?isSingular
```

```
Anova(stocks15.model.lme.foundations.outrm, type="III", test="F")
```

```
## Analysis of Deviance Table (Type III Wald F tests with Kenward-Roger df)
##
## Response: stocks_15min_diff_ln
##                                F Df Df.res    Pr(>F)
## (Intercept)               0.0573  1 3644.7   0.81086
## season_intraday_15min     0.0096  1  176.9   0.92195
## tf2_15min                 0.2324  1 3811.9   0.62977
## care_lag                  0.1293  1 2995.1   0.71921
## fairness_lag              1.0960  1 4170.9   0.29521
## loyalty_lag               0.6462  1 3741.5   0.42151
## authority_lag             2.2186  1 3692.7   0.13644
## sanctity_lag              2.1961  1 2162.5   0.13850
## tf2_15min:care_lag        1.7948  1 3989.0   0.18041
## tf2_15min:fairness_lag    4.5958  1 4056.8   0.03211 *
## tf2_15min:loyalty_lag     0.5404  1 3705.7   0.46230
## tf2_15min:authority_lag   0.8233  1 3698.7   0.36427
## tf2_15min:sanctity_lag   18.0085  1 3847.7 2.251e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 3.3: Model comparisons

### Comparisons between linear and mixed-effects models
##### Morality models
anova(stocks15.model.lme.outrm, stocks15.model.lm.outrm, type="Chisq")
```

```
## refitting model(s) with ML (instead of REML)

## Data: stocks15_outrm_moralityonly
## Models:
## stocks15.model.lm.outrm: stocks_15min_diff_ln ~ tf2_15min * morality_lag
## stocks15.model.lme.outrm: stocks_15min_diff_ln ~ (1 + season_intraday_15min + tf2_15min * morality_la
##                          npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## stocks15.model.lm.outrm     5 -36435 -36403  18223   -36445
## stocks15.model.lme.outrm    9 -36516 -36459  18267   -36534 89.06  4  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#### Foundations models
anova(stocks15.model.lme.foundations.outrm, stocks15.model.lm.foundations.outrm, type="Chisq")
```

```
## refitting model(s) with ML (instead of REML)

## Data: stocks15_outrm_moralityonly
## Models:
## stocks15.model.lm.foundations.outrm: stocks_15min_diff_ln ~ tf2_15min * (care_lag + fairness_lag + lo
## stocks15.model.lme.foundations.outrm: stocks_15min_diff_ln ~ (1 + season_intraday_15min + tf2_15min *
##                                       npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## stocks15.model.lm.foundations.outrm    13 -36443 -36360  18235   -36469
## stocks15.model.lme.foundations.outrm   17 -36527 -36419  18281   -36561 92.33  4  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### Comparisons between morality and foundations models
#### Linear models
anova(stocks15.model.lm.outrm, stocks15.model.lm.foundations.outrm)
```

```
## Analysis of Variance Table
##
## Model 1: stocks_15min_diff_ln ~ tf2_15min * morality_lag
## Model 2: stocks_15min_diff_ln ~ tf2_15min * (care_lag + fairness_lag +
##     loyalty_lag + authority_lag + sanctity_lag)
##   Res.Df      RSS Df  Sum of Sq      F  Pr(>F)
## 1   4467 0.075473
## 2   4459 0.075070  8 0.00040303 2.9924 0.00239 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#### Mixed-effects models
anova(stocks15.model.lme.null.outrm, stocks15.model.lme.outrm, stocks15.model.lme.foundations.outrm)
```

```
## refitting model(s) with ML (instead of REML)

## Data: stocks15_outrm_moralityonly
## Models:
## stocks15.model.lme.null.outrm: stocks_15min_diff_ln ~ 1 + (1 | day_count)
## stocks15.model.lme.outrm: stocks_15min_diff_ln ~ (1 + season_intraday_15min + tf2_15min * morality_la
## stocks15.model.lme.foundations.outrm: stocks_15min_diff_ln ~ (1 + season_intraday_15min + tf2_15min *
##                                       npar    AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## stocks15.model.lme.null.outrm           3 -36434 -36415  18220   -36440
## stocks15.model.lme.outrm                9 -36516 -36459  18267   -36534 94.102  6  < 2.2e-16 ***
## stocks15.model.lme.foundations.outrm   17 -36527 -36419  18281   -36561 27.210  8  0.0006504 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 3.4: Standardized betas
### Morality mixed-effect model
summ(stocks15.model.lme.outrm, scale=TRUE, transform.response=TRUE, confint=TRUE, digits=3)
```

```
## boundary (singular) fit: see ?isSingular

## MODEL INFO:
## Observations: 4471
## Dependent Variable: stocks_15min_diff_ln
## Type: Mixed effects linear regression
##
## MODEL FIT:
## AIC = 12638.266, BIC = 12695.914
## Pseudo-R² (fixed effects) = 0.000
## Pseudo-R² (total) = 0.054
##
## FIXED EFFECTS:
## ----------------------------------------------------------------------------------
##                                   Est.     2.5%   97.5%    t val.        d.f.       p
## ---------------------------- --------- -------- ------- --------- ---------- -------
## (Intercept)                      0.007   -0.032   0.047     0.356     276.009   0.722
## season_intraday_15min            0.001   -0.039   0.040     0.025     208.109   0.980
## tf2_15min                       -0.035   -0.120   0.049    -0.827     629.269   0.409
## morality_lag                    -0.010   -0.042   0.022    -0.632    3601.001   0.527
## tf2_15min:morality_lag          -0.006   -0.097   0.085    -0.132    4369.268   0.895
## ----------------------------------------------------------------------------------
##
## p values calculated using Satterthwaite d.f.
##
## RANDOM EFFECTS:
## -----------------------------------------------
##    Group           Parameter         Std. Dev.
## ----------- ----------------------- -----------
##  day_count        (Intercept)          0.148
##  day_count   season_intraday_15min     0.180
##  Residual                              0.973
## -----------------------------------------------
##
## Grouping variables:
## ------------------------------
##    Group     # groups    ICC
## ----------- ---------- -------
##  day_count     173      0.022
## ------------------------------
##
## Continuous variables are mean-centered and scaled by 1 s.d.
```

```
### Foundations mixed-effect model
summ(stocks15.model.lme.foundations.outrm, scale=TRUE, transform.response=TRUE, confint=TRUE, digits=3)
```

```
## boundary (singular) fit: see ?isSingular

## MODEL INFO:
## Observations: 4471
## Dependent Variable: stocks_15min_diff_ln
## Type: Mixed effects linear regression
##
```

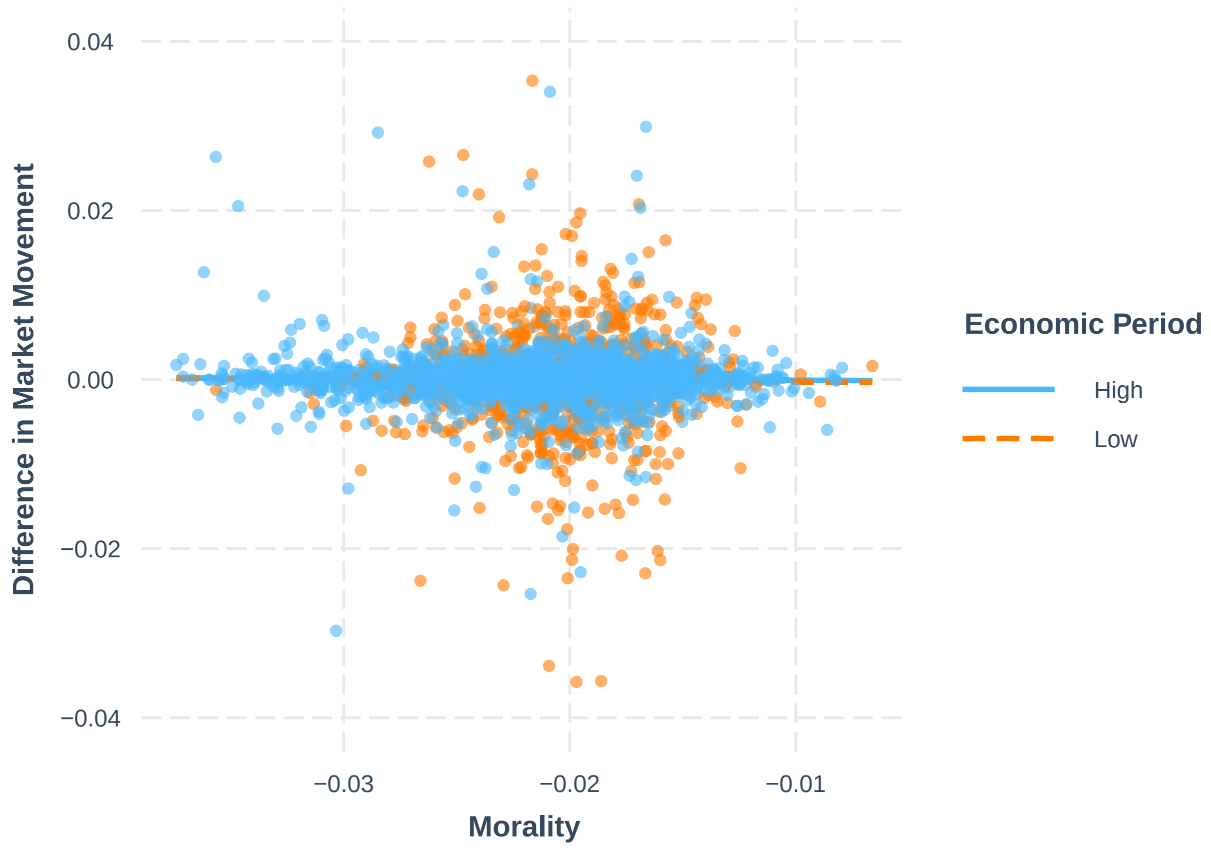
```
## MODEL FIT:
## AIC = 12657.014, BIC = 12765.905
## Pseudo-R² (fixed effects) = 0.006
## Pseudo-R² (total) = 0.061
##
## FIXED EFFECTS:
## ------------------------------------------------------------------------------------
##                                     Est.     2.5%    97.5%   t val.       d.f.       p
## ------------------------------ -------- -------- -------- -------- ---------- -------
## (Intercept)                        0.006   -0.033    0.046    0.306    271.633   0.760
## season_intraday_15min              0.002   -0.038    0.042    0.098    210.110   0.922
## tf2_15min                         -0.059   -0.145    0.026   -1.362    663.167   0.174
## care_lag                           0.015   -0.067    0.098    0.361   4046.750   0.718
## fairness_lag                      -0.050   -0.142    0.043   -1.049   4389.643   0.294
## loyalty_lag                        0.037   -0.053    0.126    0.806   4303.918   0.421
## authority_lag                     -0.065   -0.150    0.020   -1.493   4314.688   0.136
## sanctity_lag                       0.054   -0.017    0.126    1.488   3761.342   0.137
## tf2_15min:care_lag                 0.153   -0.070    0.377    1.342   4287.126   0.180
## tf2_15min:fairness_lag             0.265    0.023    0.506    2.148   4357.006   0.032
## tf2_15min:loyalty_lag              0.087   -0.144    0.318    0.737   4340.746   0.461
## tf2_15min:authority_lag           -0.102   -0.321    0.117   -0.910   4228.991   0.363
## tf2_15min:sanctity_lag            -0.425   -0.621   -0.229   -4.253   4347.225   0.000
## ------------------------------------------------------------------------------------
##
## p values calculated using Satterthwaite d.f.
##
## RANDOM EFFECTS:
## --------------------------------------------------
##    Group           Parameter         Std. Dev.
## ----------- ----------------------- -----------
##  day_count         (Intercept)          0.149
##  day_count   season_intraday_15min     0.182
##  Residual                               0.970
## --------------------------------------------------
##
## Grouping variables:
## ------------------------------
##    Group     # groups    ICC
## ----------- ---------- -------
##  day_count     173      0.023
## ------------------------------
##
## Continuous variables are mean-centered and scaled by 1 s.d.
```

```
## 3.6: Interaction plots
### Define interaction plots
#### Morality plot
plot1 <- interact_plot(stocks15.model.lme.outrm,
                       pred = morality_lag,
                       modx = tf2_15min,
                       plot.points = TRUE,
                       linearity.check = FALSE,
                       x.label = "Morality",
                       y.label = "Difference in Market Movement",
                       modx.labels = c("High","Low"),
                       legend.main = "Economic Period") +
  ylim(-0.04,0.04)
plot1a <- plot1 + theme(legend.position = "none")
#### Fairness plot
plot2 <- interact_plot(stocks15.model.lme.foundations.outrm,
                       pred = fairness_lag,
                       modx = tf2_15min,
                       plot.points = TRUE,
                       linearity.check = FALSE,
                       x.label = "Fairness",
                       y.label = "Difference in Market Movement") +
  ylim(-0.04,0.04) +
  theme(legend.position = "none")
#### Sanctity plot
plot3 <- interact_plot(stocks15.model.lme.foundations.outrm,
                       pred = sanctity_lag,
                       modx = tf2_15min,
                       plot.points = TRUE,
                       linearity.check = FALSE,
                       x.label = "Sanctity",
                       y.label = "Difference in Market Movement") +
  ylim(-0.04,0.04) +
  theme(legend.position = "none")
### Plot interactions
legend <- get_legend(plot1)
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```

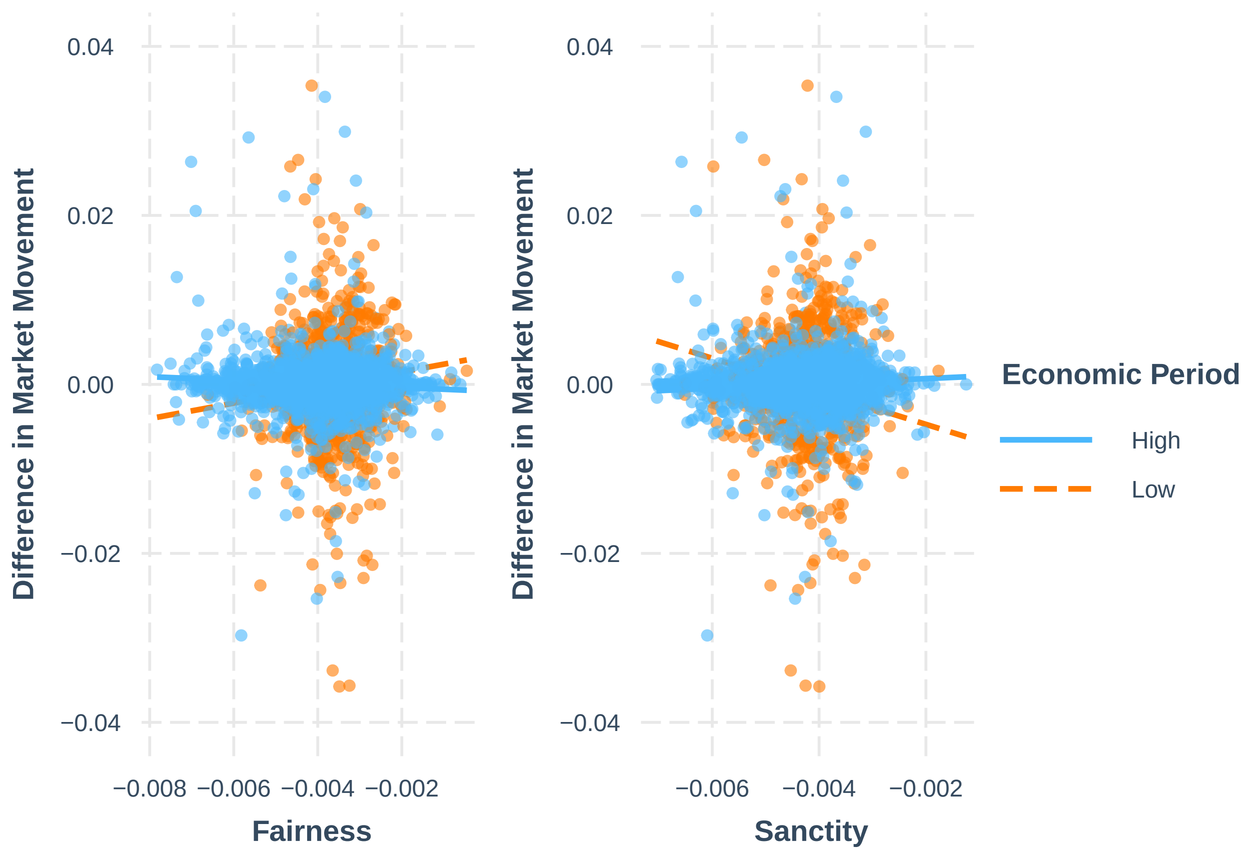
```
grid.arrange(plot1)
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



```r
grid.arrange(plot2, plot3, legend, ncol=3, widths=c(3,3,1.5))
```

```
## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 6 rows containing missing values (geom_point).
```



Difference in Market Movement
0.04
0.02
0.00
−0.02
−0.04
−0.008
−0.006
−0.004
−0.002
Fairness
Sanctity
Economic Period
High
Low