Collective Emotions and Social Resilience in the Digital Traces After a Terrorist Attack

David García and Bernard Rimé

August 31st, 2018 (refactored on October 10th, 2022)

1. Data description

```
library(sfsmisc)
library(ggplot2)
library(zoo)
library(dplyr)
library(magrittr)
library(arm)
library(texreg)
Sys.setlocale("LC_ALL", 'en_US.UTF-8')
source("Scripts/AuxFunctions.R")
```

```
load("Data/Tweets.RData")
Tweets %>% mutate(day=as.Date(date)) -> Tweets
Tweets$user <- Tweets$userid
Tweets %>% group_by(date) -> Tweets
print(paste("N tweets:",nrow(Tweets))) # 7666170
```

```
## [1] "N tweets: 17899591"
```

```
Tweets %>% filter(date>="2015-04-01" & date < "2015-10-01")-> TweetsBL
TweetsBL %>% mutate(w=weekdays(as.Date(date))) -> TweetsBL
print(paste("N tweets in baseline:",nrow(TweetsBL))) # 7666170
```

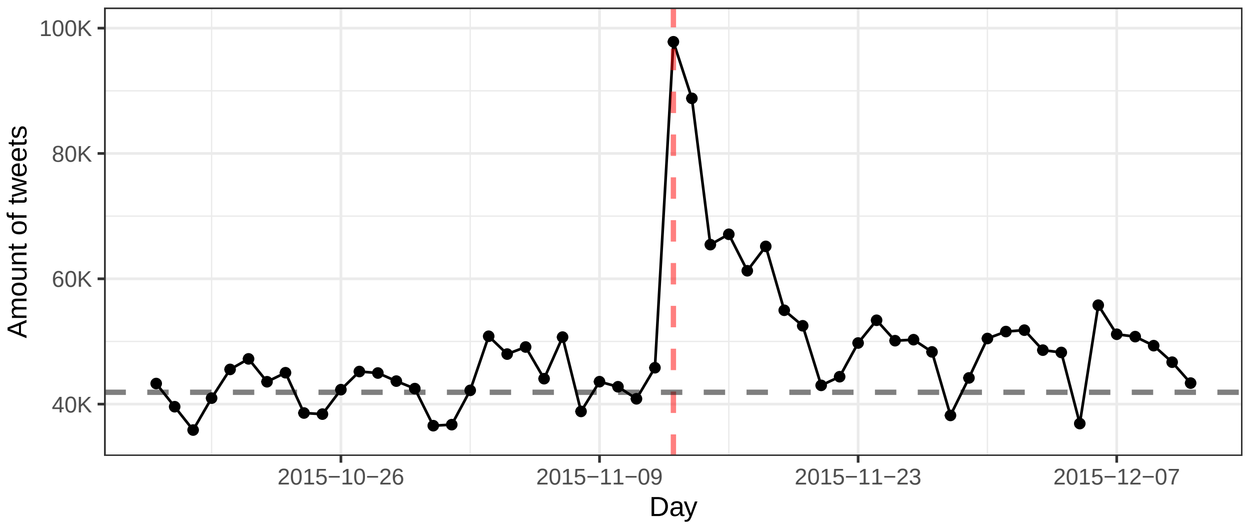
```
## [1] "N tweets in baseline: 7666170"
```

```
TweetsBL %>% group_by(date) %>% summarise(nt=length(n)) %>% summarise(mntw=mean(nt)) -> ntweetsBL
Tweets %>% summarise(ntweets=length(n)) -> dts
dts$bl <- rep(ntweetsBL$mntw, length(dts$ntweets))
```

```
dts %>% filter(date >= "2015-10-16"  & date < "2015-12-12") -> dts2

print(paste("N tweets in window:",sum(dts2$ntweets))) # 2766054
```

```
## [1] "N tweets in window: 2766054"
```

```r
plt <- ggplot()
plt <- plt + geom_line(data = dts2, mapping=aes(x=as.Date(date), y=ntweets))
plt <- plt + geom_point(data = dts2, mapping=aes(x=as.Date(date), y=ntweets))
plt <- plt + scale_x_date("Day", date_breaks = "2 weeks") +
      scale_y_continuous(name="Amount of tweets", breaks=seq(20000,100000,20000),
                         labels=c("20K","40K","60K","80K","100K")) +
      expand_limits(y=c(35000,100000))
plt <- plt + theme_bw() + geom_vline(xintercept=as.numeric(as.Date("2015-11-13")),
                                    col=rgb(1,0,0,0.5), lwd=1, lty=2)
plt <- plt + geom_hline(yintercept = dts$bl[1], col=rgb(0,0,0,0.5), lwd=1,lty=2)
plt
```



2. Affective Reactions

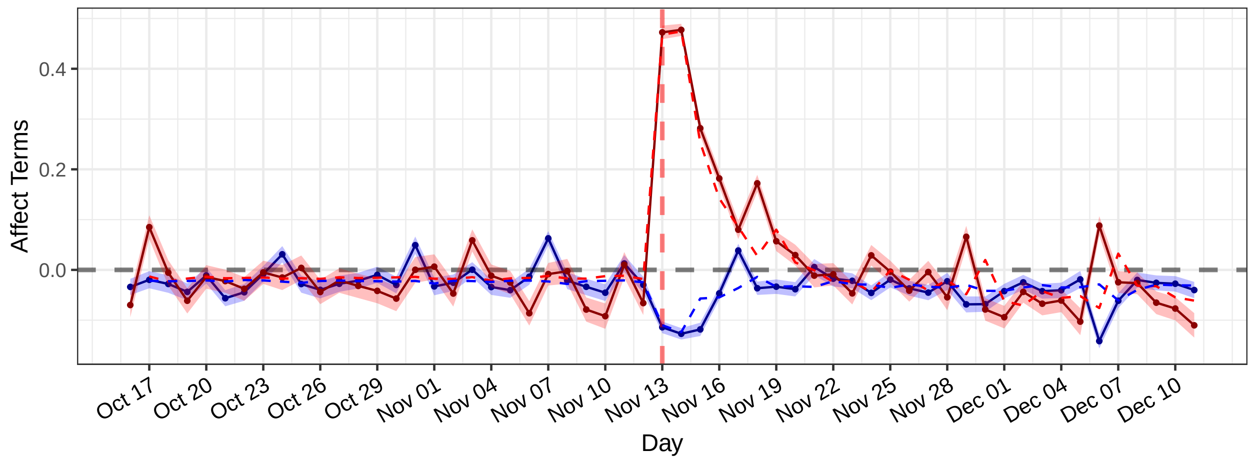
```r
TweetsBL %>% group_by(w) %>%summarise(bl=mean(posemo/n)) -> posBL
TweetsBL %>% group_by(w) %>%summarise(bl=mean(negemo/n)) -> negBL

Tweets %>% filter(as.Date(date) >= "2015-10-16"  & as.Date(date) < "2015-12-12") -> Tsel
posts <- ciTS(Tsel, "posemo", R=10000)
posts$w <- weekdays(as.Date(posts$date))
posts <- inner_join(posts, posBL)
save(posts, file="temp/posemoCITS.RData")

negts <- ciTS(Tsel, "negemo", R=10000)
negts$w <- weekdays(as.Date(negts$date))
negts <- inner_join(negts, negBL)
save(negts, file="temp/negemoCITS.RData")
```

```r
load("temp/posemoCITS.RData")
load("temp/negemoCITS.RData")

d1 <- as.Date("2015-11-13")
dtbreaks <- c(d1 + seq(-27,0,by=3), d1 + seq(3,27,by=3))
```

```
plt <- plotts(ts=posts, ylab="Affect Terms", col="darkblue", bgcolor = rgb(0,0,1,0.25))
plt <- plotts(ts=negts, col="darkred", bgcolor = rgb(1,0,0,0.25), plt=plt)
plt <- plt + scale_x_date("Day", breaks = dtbreaks, date_labels = "%b %d")
plt <- plt + theme(axis.text.x = element_text(angle = 30, hjust = 1, colour = "black", size = 10))

Zday <- which(posts$date == "2015-11-13")[1]
negFit <- TSmodel1(log(negts$mid/negts$bl), Zday)
negDF <- data.frame(x=negts$date[2:length(negts$date)],y=negFit$fitted.values)
plt <- plt + geom_line(data=negDF, aes(x=as.Date(x),y=y), col= rgb(1,0,0), lty=2)

posFit <- TSmodel1(log(posts$mid/posts$bl), Zday)
posDF <- data.frame(x=posts$date[2:length(posts$date)],y=posFit$fitted.values)
plt <- plt + geom_line(data=posDF, aes(x=as.Date(x),y=y), col= rgb(0,0,1), lty=2)

plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.
```

```
## Warning: Use of `ts$bl` is discouraged. Use `bl` instead.
```




```
texreg(list(posFit, negFit), custom.model.names = c("PA","NA"),
       digits=4, bold=0.05)
```

```
simulates <- coef(sim(posFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.03375430 -0.02389288 -0.01391375
```

	PA	NA
(Intercept)	$\mathbf{-0.0239}^{***}$	$\mathbf{-0.0175}^{*}$
	(0.0051)	(0.0075)
ypre:postFALSE	−0.0652	−0.0644
	(0.1276)	(0.2073)
ypre:postTRUE	$\mathbf{0.2604}^{*}$	$\mathbf{0.5669}^{***}$
	(0.1124)	(0.0772)
zTRUE	$\mathbf{-0.0882}^{**}$	$\mathbf{0.4812}^{***}$
	(0.0310)	(0.0532)
z2TRUE	$\mathbf{-0.0701}^{*}$	$\mathbf{0.2240}^{***}$
	(0.0323)	(0.0626)
AIC	-222.5376	-165.9640
BIC	-210.3855	-153.8119
Log Likelihood	117.2688	88.9820
Deviance	0.0498	0.1366
Num. obs.	56	56

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 1: Statistical models

```
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

```
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.31689173 -0.06297917  0.18725203
```

```
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##       2.5%        50%      97.5%
## 0.04004091 0.26017888 0.47673619
```

```
print("zTRUE")
```

```
## [1] "zTRUE"
```

```
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.14858019 -0.08822092 -0.02866958
```

```r
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##         2.5%          50%        97.5%
## -0.133078072 -0.070313969 -0.007230268
```

```r
library(tseries)
confint.default(posFit)
```

```
##                       2.5 %       97.5 %
## (Intercept)    -0.03385149 -0.013877238
## ypre:postFALSE -0.31531176  0.184887999
## ypre:postTRUE   0.04013304  0.480760085
## zTRUE          -0.14885563 -0.027450230
## z2TRUE         -0.13336449 -0.006749594
```

```r
summ <- summary(posFit)
print(paste("Positive terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "Positive terms model R2: 0.320636290180013"
```

```r
shapiro.test(posFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  posFit$residuals
## W = 0.88288, p-value = 5.772e-05
```

```r
kpss.test(posFit$residuals)
```

```
## Warning in kpss.test(posFit$residuals): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  posFit$residuals
## KPSS Level = 0.24758, Truncation lag parameter = 3, p-value = 0.1
```

```r
cor.test(sqrt(abs(posFit$residuals)), posFit$fitted.values)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(posFit$residuals)) and posFit$fitted.values
```

```
## t = 1.1988, df = 54, p-value = 0.2359
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1064014  0.4066892
## sample estimates:
##       cor
## 0.1610029
```

```
simulates <- coef(sim(negFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##         2.5%          50%        97.5%
## -0.032234067 -0.017585585 -0.002637136
```

```
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

```
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.47047061 -0.06626153  0.34466279
```

```
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.4160162 0.5666934 0.7171887
```

```
print("zTRUE")
```

```
## [1] "zTRUE"
```

```
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.3783450 0.4808526 0.5829322
```

```r
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.1028266 0.2251994 0.3458624
```

```r
confint.default(negFit)
```

```
##                      2.5 %       97.5 %
## (Intercept)    -0.03223176 -0.002839171
## ypre:postFALSE -0.47070331  0.341947757
## ypre:postTRUE   0.41563237  0.718138851
## zTRUE           0.37693212  0.585414794
## z2TRUE          0.10136199  0.346683142
```

```r
summ <- summary(negFit)
print(paste("Negative terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "Negative terms model R2: 0.814094076168924"
```

```r
shapiro.test(negFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  negFit$residuals
## W = 0.91346, p-value = 0.0006731
```

```r
kpss.test(negFit$residuals)
```

```
## Warning in kpss.test(negFit$residuals): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  negFit$residuals
## KPSS Level = 0.070959, Truncation lag parameter = 3, p-value = 0.1
```

```r
cor.test(sqrt(abs(negFit$residuals)), negFit$fitted.values)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(negFit$residuals)) and negFit$fitted.values
## t = -1.7877, df = 54, p-value = 0.07944
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.47006823  0.02827807
## sample estimates:
##        cor
## -0.2363798
```

```
TweetsBL %>% group_by(w)  %>% summarise(bl=mean(sad/n)) -> sadBL
TweetsBL %>% group_by(w) %>% summarise(bl=mean(ang/n)) -> angBL
TweetsBL %>% group_by(w) %>% summarise(bl=mean(anx/n)) -> anxBL

angts <- ciTS(Tsel, "ang", R=10000)
angts$w <- weekdays(as.Date(angts$date))
angts <- inner_join(angts, angBL)
save(angts, file="temp/angCITS.RData")

anxts <- ciTS(Tsel, "anx", R=10000)
anxts$w <- weekdays(as.Date(anxts$date))
anxts <- inner_join(anxts, anxBL)
save(anxts, file="temp/anxCITS.RData")

sadts <- ciTS(Tsel, "sad", R=10000)
sadts$w <- weekdays(as.Date(sadts$date))
sadts <- inner_join(sadts, sadBL)
save(sadts, file="temp/sadCITS.RData")
```

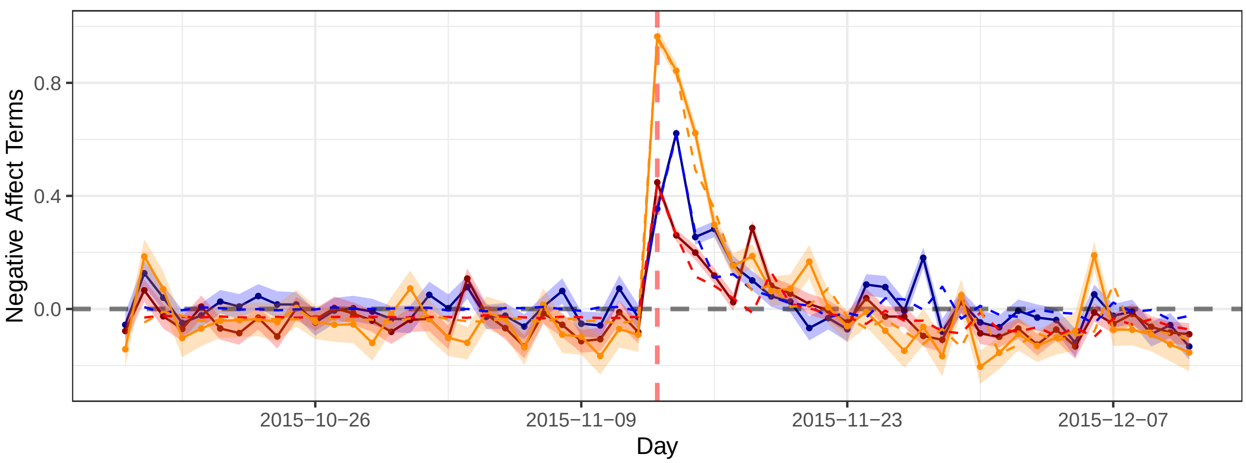
```
load("temp/anxCITS.RData")
load("temp/angCITS.RData")
load("temp/sadCITS.RData")

plt <- plotts(ts=sadts, ylab="Negative Affect Terms",
              col="darkblue", bgcolor = rgb(0,0,1,0.25))
plt <- plotts(ts=angts, col="darkred", bgcolor = rgb(1,0,0,0.25), plt=plt)
plt <- plotts(ts=anxts, col="darkorange", bgcolor = rgb(1,140/255,0,0.25), plt=plt)

Zday <- which(posts$date == "2015-11-13")[1]
sadFit <- TSmodel1(log(sadts$mid/sadts$bl), Zday)
DF <- data.frame(x=sadts$date[2:length(sadts$date)],y=sadFit$fitted.values)
plt <- plt + geom_line(data=DF, aes(x=as.Date(x),y=y), col= rgb(0,0,1), lty=2)

angFit <- TSmodel1(log(angts$mid/angts$bl), Zday)
DF <- data.frame(x=angts$date[2:length(angts$date)],y=angFit$fitted.values)
plt <- plt + geom_line(data=DF, aes(x=as.Date(x),y=y), col= rgb(1,0,0), lty=2)

anxFit <- TSmodel1(log(anxts$mid/anxts$bl), Zday)
DF <- data.frame(x=anxts$date[2:length(anxts$date)],y=anxFit$fitted.values)
plt <- plt + geom_line(data=DF, aes(x=as.Date(x),y=y), col= rgb(1,140/255,0), lty=2)

plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.

## Warning: Use of `ts$bl` is discouraged. Use `bl` instead.
```

	Sadness	Anger	Anxiety
(Intercept)	0.0003	$\mathbf{-0.0268}^{*}$	$\mathbf{-0.0314}^{*}$
	(0.0093)	(0.0101)	(0.0133)
ypre:postFALSE	−0.1116	0.0452	0.1161
	(0.2353)	(0.1903)	(0.2072)
ypre:postTRUE	$\mathbf{0.4337}^{***}$	$\mathbf{0.5444}^{***}$	$\mathbf{0.6220}^{***}$
	(0.0824)	(0.1170)	(0.0728)
zTRUE	$\mathbf{0.3446}^{***}$	$\mathbf{0.4700}^{***}$	$\mathbf{0.9988}^{***}$
	(0.0674)	(0.0689)	(0.0914)
z2TRUE	$\mathbf{0.4604}^{***}$	0.0422	$\mathbf{0.2713}^{*}$
	(0.0723)	(0.0847)	(0.1128)
AIC	-137.0705	-135.4260	-104.2293
BIC	-124.9184	-123.2739	-92.0772
Log Likelihood	74.5352	73.7130	58.1146
Deviance	0.2289	0.2357	0.4115
Num. obs.	56	56	56

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 2: Statistical models




```
texreg(list(sadFit, angFit, anxFit),
       custom.model.names = c("Sadness","Anger","Anxiety"),
       digits=4, bold=0.05)
```

```
simulates <- coef(sim(sadFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##          2.5%           50%         97.5%
## -0.0181113075  0.0003868619  0.0186559027
```

```r
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

```r
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##       2.5%        50%      97.5% 
## -0.5608988 -0.1106419  0.3447773
```

```r
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```r
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5% 
## 0.2728708 0.4328643 0.5955657
```

```r
print("zTRUE")
```

```
## [1] "zTRUE"
```

```r
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5% 
## 0.2117403 0.3439702 0.4772113
```

```r
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5% 
## 0.3200145 0.4605028 0.6014994
```

```r
summ <- summary(sadFit)
confint.default(sadFit)
```

```
##                      2.5 %     97.5 %
## (Intercept)    -0.01789236 0.01852496
## ypre:postFALSE -0.57273097 0.34961751
## ypre:postTRUE   0.27225981 0.59508548
## zTRUE           0.21248197 0.47668488
## z2TRUE          0.31867101 0.60218966
```

```r
print(paste("Sadness terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "Sadness terms model R2: 0.72509524065696"
```

```r
shapiro.test(sadFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sadFit$residuals
## W = 0.9678, p-value = 0.1393
```

```r
kpss.test(sadFit$residuals)
```

```
## Warning in kpss.test(sadFit$residuals): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  sadFit$residuals
## KPSS Level = 0.27604, Truncation lag parameter = 3, p-value = 0.1
```

```r
cor.test(sqrt(abs(sadFit$residuals)), sadFit$fitted.values)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(sadFit$residuals)) and sadFit$fitted.values
## t = -1.4816, df = 54, p-value = 0.1443
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.43779302  0.06883722
## sample estimates:
##        cor
## -0.1976401
```

```r
simulates <- coef(sim(angFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```r
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##         2.5%          50%        97.5%
## -0.046598739 -0.026912208 -0.006506174
```

```r
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

```r
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.32532254  0.04559417  0.41346108
```

```r
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```r
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.3179959 0.5436455 0.7704012
```

```r
print("zTRUE")
```

```
## [1] "zTRUE"
```

```r
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.3348275 0.4694797 0.6036234
```

```r
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.12541854  0.04178015  0.20787647
```

```r
summ <- summary(angFit)
confint.default(angFit)
```

```
##                      2.5 %       97.5 %
## (Intercept)    -0.04655919 -0.007003452
## ypre:postFALSE -0.32780784  0.418184553
## ypre:postTRUE   0.31499807  0.773720618
## zTRUE           0.33490204  0.605124221
## z2TRUE         -0.12376716  0.208086024
```

```r
print(paste("Anger terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "Anger terms model R2: 0.630283227850662"
```

```r
shapiro.test(angFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  angFit$residuals
## W = 0.85791, p-value = 9.754e-06
```

```r
kpss.test(angFit$residuals)
```

```
## Warning in kpss.test(angFit$residuals): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  angFit$residuals
## KPSS Level = 0.1018, Truncation lag parameter = 3, p-value = 0.1
```

```r
cor.test(sqrt(abs(angFit$residuals)), angFit$fitted.values)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(angFit$residuals)) and angFit$fitted.values
## t = -1.4684, df = 54, p-value = 0.1478
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.43637155  0.07058583
## sample estimates:
##       cor
## -0.195951
```

```r
simulates <- coef(sim(anxFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```r
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##         2.5%          50%        97.5%
## -0.058193910 -0.031327981 -0.005550118
```

```r
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

```r
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##       2.5%        50%      97.5% 
## -0.2990154  0.1148511  0.5175668
```

```r
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```r
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5% 
## 0.4787574 0.6220100 0.7623145
```

```r
print("zTRUE")
```

```
## [1] "zTRUE"
```

```r
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5% 
## 0.8234776 0.9990888 1.1773444
```

```r
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##       2.5%        50%      97.5% 
## 0.05024091 0.27241023 0.49378781
```

```r
summ <- summary(anxFit)
confint.default(anxFit)
```

```
##                       2.5 %      97.5 %
## (Intercept)    -0.05746755 -0.00540306
## ypre:postFALSE -0.29007127  0.52218201
## ypre:postTRUE   0.47935823  0.76457837
## zTRUE           0.81974455  1.17793407
## z2TRUE          0.05025417  0.49241980
```

```r
print(paste("Anxiety terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "Anxiety terms model R2: 0.84568694624521"
```

```r
shapiro.test(anxFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  anxFit$residuals
## W = 0.94105, p-value = 0.008617
```

```r
kpss.test(anxFit$residuals)
```

```
## Warning in kpss.test(anxFit$residuals): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  anxFit$residuals
## KPSS Level = 0.082316, Truncation lag parameter = 3, p-value = 0.1
```

```r
cor.test(sqrt(abs(anxFit$residuals)), anxFit$fitted.values)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(anxFit$residuals)) and anxFit$fitted.values
## t = -0.93662, df = 54, p-value = 0.3531
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3768102  0.1411579
## sample estimates:
##        cor
## -0.1264347
```

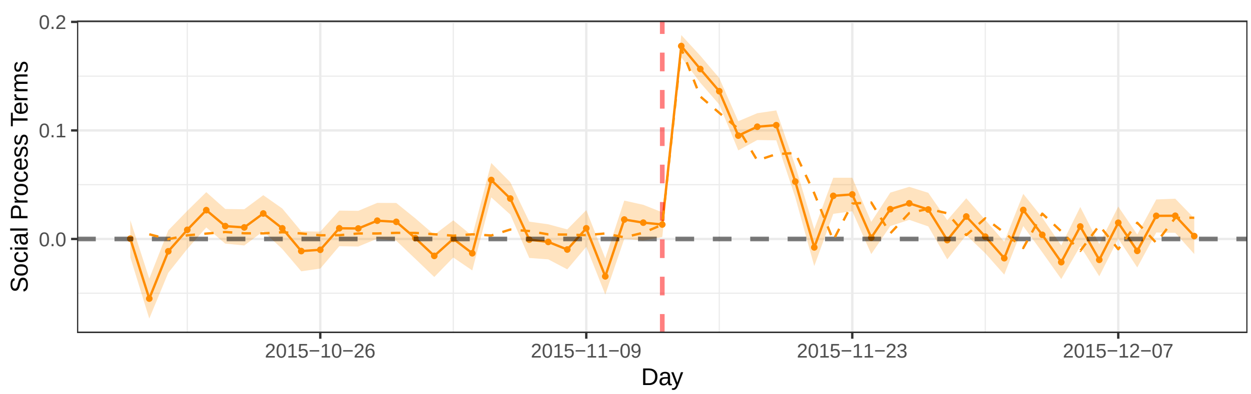
3. Social processes

```r
TweetsBL %>% group_by(w) %>%summarise(bl=mean(soc/n)) -> SocBL
TweetsBL %>% group_by(w) %>%summarise(bl=mean(prosoc/n)) -> ProSocBL
TweetsBL %>% group_by(w) %>%summarise(bl=mean(frenchValues/n)) -> FVBL

Socts <- ciTS(Tsel, "soc", R=10000)
Socts$w <- weekdays(as.Date(Socts$date))
Socts <- inner_join(Socts, SocBL)
save(Socts, file="temp/SocCITS.RData")
```

```r
ProSocts <- ciTS(Tsel, "prosoc", R=10000)
ProSocts$w <- weekdays(as.Date(ProSocts$date))
ProSocts <- inner_join(ProSocts, ProSocBL)
save(ProSocts, file="temp/ProSocCITS.RData")

FVts <- ciTS(Tsel, "frenchValues", R=10000)
FVts$w <- weekdays(as.Date(FVts$date))
FVts <- inner_join(FVts, FVBL)
save(FVts, file="temp/frenchValuesCITS.RData")
```

```r
load("temp/SocCITS.RData")

plt <- plotts(ts=Socts, ylab="Social Process Terms",
              col="darkorange", bgcolor = rgb(1,144/255,0,0.25))

Zday <- which(Socts$date == "2015-11-13")[1]
SocFit <- TSmodel1(log(Socts$mid/Socts$bl), Zday)
DF <- data.frame(x=Socts$date[2:length(Socts$date)],y=SocFit$fitted.values)
plt <- plt + geom_line(data=DF, aes(x=as.Date(x),y=y), col= rgb(1,144/255,0), lty=2)

plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.

## Warning: Use of `ts$bl` is discouraged. Use `bl` instead.
```




```r
texreg(SocFit, custom.model.names = c("Social"), digits=4, bold=0.05)
```

```r
simulates <- coef(sim(SocFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```r
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##         2.5%          50%        97.5%
## -0.002653046  0.004291746  0.011149793
```

	Social
(Intercept)	0.0043
	(0.0036)
ypre:postFALSE	0.0810
	(0.1453)
ypre:postTRUE	**0.7146*****
	(0.0754)
zTRUE	0.0076
	(0.0238)
z2TRUE	**0.1616*****
	(0.0238)
AIC	-253.0178
BIC	-240.8657
Log Likelihood	132.5089
Deviance	0.0289
Num. obs.	56

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 3: Statistical models

```
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

```
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.20638603  0.08009316  0.36951916
```

```
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.5677525 0.7154598 0.8628589
```

```
print("zTRUE")
```

```
## [1] "zTRUE"
```

```
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##         2.5%          50%        97.5%
## -0.038899070  0.007727576  0.053358673
```

```r
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.1148412 0.1613072 0.2082989
```

```r
summ <- summary(SocFit)
confint.default(SocFit)
```

```
##                        2.5 %     97.5 %
## (Intercept)    -0.002708557 0.01125288
## ypre:postFALSE -0.203714229 0.36576467
## ypre:postTRUE   0.566851413 0.86239467
## zTRUE          -0.039102850 0.05437763
## z2TRUE          0.114842490 0.20827864
```

```r
print(paste("Social process terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "Social process terms model R2: 0.727505894368885"
```

```r
shapiro.test(SocFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  SocFit$residuals
## W = 0.99025, p-value = 0.9307
```

```r
kpss.test(SocFit$residuals)
```

```
## Warning in kpss.test(SocFit$residuals): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  SocFit$residuals
## KPSS Level = 0.079223, Truncation lag parameter = 3, p-value = 0.1
```

```r
cor.test(sqrt(abs(SocFit$residuals)), SocFit$fitted.values)
```

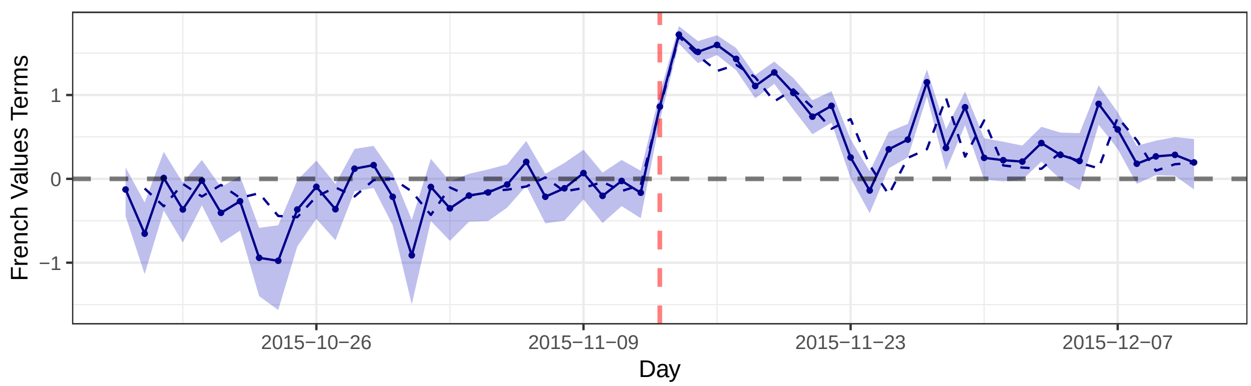
```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(SocFit$residuals)) and SocFit$fitted.values
## t = -0.28917, df = 54, p-value = 0.7736
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2991287  0.2259155
## sample estimates:
##         cor
## -0.03932035
```

```
load("temp/frenchValuesCITS.RData")
plt <- plotts(ts=FVts, ylab="French Values Terms",
              col="darkblue", bgcolor = rgb(0,0,0.75,0.25))

Zday <- which(FVts$date == "2015-11-13")[1]
FVFit <- TSmodel1(log(FVts$mid/FVts$bl), Zday)
DF <- data.frame(x=FVts$date[2:length(FVts$date)],y=FVFit$fitted.values)
plt <- plt + geom_line(data=DF, aes(x=as.Date(x),y=y), col= rgb(0,0,144/255), lty=2)
plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.
```

```
## Warning: Use of `ts$bl` is discouraged. Use `bl` instead.
```




```
texreg(FVFit, custom.model.names = c("French Values Terms"), digits=4, bold=0.05)
```

```
simulates <- coef(sim(FVFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.19662335 -0.06307364  0.06732120
```

```
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

	French Values Terms
(Intercept)	-0.0639
	(0.0674)
ypre:postFALSE	$\mathbf{0.4023}^{*}$
	(0.1987)
ypre:postTRUE	$\mathbf{0.8916}^{***}$
	(0.1011)
zTRUE	$\mathbf{0.9704}^{**}$
	(0.3472)
z2TRUE	$\mathbf{0.9944}^{**}$
	(0.3492)
AIC	46.9645
BIC	59.1166
Log Likelihood	-17.4822
Deviance	6.1217
Num. obs.	56

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 4: Statistical models

```
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## 0.004657652 0.401118883 0.786836661
```

```
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.6944170 0.8913761 1.0840821
```

```
print("zTRUE")
```

```
## [1] "zTRUE"
```

```
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.2994093 0.9689755 1.6584137
```

```
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.2985236 0.9857517 1.6811810
```

```r
summ <- summary(FVFit)
confint.default(FVFit)
```

```
##                       2.5 %     97.5 %
## (Intercept)    -0.19614136 0.06825486
## ypre:postFALSE  0.01295194 0.79171428
## ypre:postTRUE   0.69345962 1.08968676
## zTRUE           0.28983941 1.65101132
## z2TRUE          0.30993792 1.67893619
```

```r
print(paste("French Values terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "French Values terms model R2: 0.715291698184095"
```

```r
shapiro.test(FVFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  FVFit$residuals
## W = 0.98072, p-value = 0.5071
```

```r
kpss.test(FVFit$residuals)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  FVFit$residuals
## KPSS Level = 0.5964, Truncation lag parameter = 3, p-value = 0.02296
```

```r
cor.test(sqrt(abs(FVFit$residuals)), FVFit$fitted.values)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(FVFit$residuals)) and FVFit$fitted.values
## t = -1.2658, df = 54, p-value = 0.211
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.41417327  0.09749468
## sample estimates:
##       cor
## -0.169757
```

	Prosocial Terms
(Intercept)	**0.0408*****
	(0.0081)
ypre:postFALSE	0.2317
	(0.1509)
ypre:postTRUE	**0.6544*****
	(0.0449)
zTRUE	**0.1405*****
	(0.0326)
z2TRUE	**0.4072*****
	(0.0328)
AIC	-218.8325
BIC	-206.6804
Log Likelihood	115.4162
Deviance	0.0532
Num. obs.	56

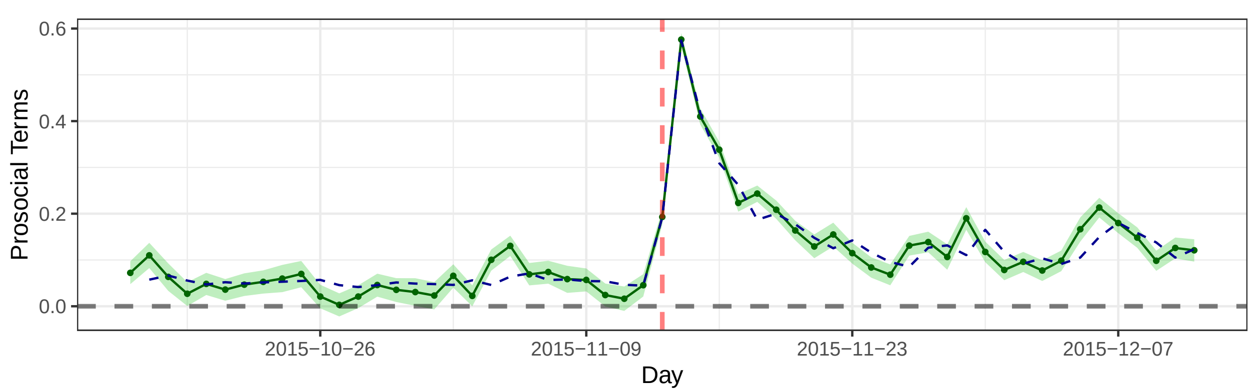
$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 5: Statistical models

```
load("temp/ProSocCITS.RData")
plt <- plotts(ts=ProSocts, ylab="Prosocial Terms",
              col="darkgreen", bgcolor = rgb(0,0.75,0,0.25))

Zday <- which(ProSocts$date == "2015-11-13")[1]
ProSocFit <- TSmodel1(log(ProSocts$mid/ProSocts$bl), Zday)
DF <- data.frame(x=ProSocts$date[2:length(ProSocts$date)],y=ProSocFit$fitted.values)
plt <- plt + geom_line(data=DF, aes(x=as.Date(x),y=y), col= rgb(0,0,144/255), lty=2)
plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.
```

```
## Warning: Use of `ts$bl` is discouraged. Use `bl` instead.
```




```
texreg(ProSocFit, custom.model.names = c("Prosocial Terms"), digits=4, bold=0.05)
```

```r
simulates <- coef(sim(ProSocFit, n.sims=20000))
print("Intercept:")
```

```
## [1] "Intercept:"
```

```r
quantile(simulates[10001:20000,1], probs = c(0.025, 0.5, 0.975))
```

```
##       2.5%        50%      97.5%
## 0.02473692 0.04066771 0.05708683
```

```r
print("ypre:postFALSE")
```

```
## [1] "ypre:postFALSE"
```

```r
quantile(simulates[10001:20000,2], probs = c(0.025, 0.5, 0.975))
```

```
##        2.5%         50%       97.5%
## -0.06694451  0.23512715  0.52642854
```

```r
print("ypre:postTRUE")
```

```
## [1] "ypre:postTRUE"
```

```r
quantile(simulates[10001:20000,3], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.5627855 0.6547865 0.7407239
```

```r
print("zTRUE")
```

```
## [1] "zTRUE"
```

```r
quantile(simulates[10001:20000,4], probs = c(0.025, 0.5, 0.975))
```

```
##       2.5%        50%      97.5%
## 0.07643358 0.13996283 0.20531268
```

```r
print("z2TRUE")
```

```
## [1] "z2TRUE"
```

```r
quantile(simulates[10001:20000,5], probs = c(0.025, 0.5, 0.975))
```

```
##      2.5%       50%     97.5%
## 0.3434912 0.4079185 0.4708813
```

```r
summ <- summary(ProSocFit)
confint.default(ProSocFit)
```

```
##                       2.5 %     97.5 %
## (Intercept)      0.02487821 0.05667417
## ypre:postFALSE -0.06403942 0.52742572
## ypre:postTRUE   0.56647622 0.74229704
## zTRUE           0.07656915 0.20447626
## z2TRUE          0.34297309 0.47141470
```

```r
print(paste("prosocial terms model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "prosocial terms model R2: 0.905744466411463"
```

```r
shapiro.test(ProSocFit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ProSocFit$residuals
## W = 0.93085, p-value = 0.003221
```

```r
kpss.test(ProSocFit$residuals)
```

```
## Warning in kpss.test(ProSocFit$residuals): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ProSocFit$residuals
## KPSS Level = 0.072436, Truncation lag parameter = 3, p-value = 0.1
```

```r
cor.test(sqrt(abs(ProSocFit$residuals)), ProSocFit$fitted.values)
```

```
##
##  Pearson's product-moment correlation
##
## data:  sqrt(abs(ProSocFit$residuals)) and ProSocFit$fitted.values
## t = -0.48618, df = 54, p-value = 0.6288
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3233061  0.2003612
## sample estimates:
##         cor
## -0.06601682
```

4. Emotional Synchronization Effect

```
Tweets$emo <- Tweets$posemo + Tweets$negemo

Tweets %>% filter(date>="2015-11-13" & date <= "2015-11-27")-> TweetsDFAttack
TweetsDFAttack %>% group_by(userid) -> TweetsDFAttack
TweetsDFAttack %>% summarise(avgEmo=mean((posemo+negemo)/n), nT=n()/15, # 15 days after the attacks
                             avgSoc=mean(soc/n), avgProsoc=mean(prosoc/n),
                             avgFV=mean(frenchValues/n), avgPA=mean(posemo/n), avgNA=mean(negemo/n),
                             avgAng=mean(ang/n), avgAnx = mean(anx/n), avgSad=mean(sad/n)) -> userDF

Tweets %>% filter(date>="2015-08-13" & date <= "2015-11-12")-> TweetsDFBL
TweetsDFBL %>% group_by(userid) -> TweetsDFBL
TweetsDFBL %>% summarise(nTBL = length(n)/92,  # number of days in baseline period
                         avgEmoBL=mean((posemo+negemo)/n), socBL=mean(soc/n),  prosocBL=mean(prosoc/n),
                         FVBL=mean(frenchValues/n),
                         PABL=mean(posemo/n), NABL=mean(negemo/n),
                         anxBL=mean(anx/n), sadBL=mean(sad/n), angerBL=mean(ang/n), iBL=mean(i/n)) -> us

userDF <- full_join(userDF, userDFBL, by="userid")

Tweets %>% filter(date>="2015-11-28" & date <= "2016-02-27")-> TweetsDFPost
TweetsDFPost %>% group_by(userid) -> TweetsDFPost
TweetsDFPost %>% summarise( nTpost=n(),
  PAPost = mean(posemo/n),  NAPost = mean(negemo/n),
  anxPost = mean(anx/n),  sadPost = mean(sad/n), angPost = mean(ang/n),
  EmoPost = mean((posemo+negemo)/n),
  socPost=mean(soc/n), prosocPost =mean(prosoc/n), FVPost = mean(frenchValues/n)) -> userDFPost
userDF <- full_join(userDF, userDFPost, by="userid")

save(userDF, file="temp/userDF.RData")

sel <- data.frame(userid=userDF$userid, sel=userDF$avgEmo>userDF$avgEmoBL)
sel$user <- sel$userid

pTS <- PairTSW(Tweets, sel, "soc", R=1000,w=30)
save(pTS, file="temp/SynchSoc.RData")

pTS <- PairTSDifW(Tweets, sel, "soc", R=1000,w=30)
save(pTS, file="temp/SynchDifSoc.RData")

pTS <- PairTSW(Tweets, sel, "prosoc", R=1000,w=30)
save(pTS, file="temp/SynchProSoc.RData")

pTS <- PairTSDifW(Tweets, sel, "prosoc", R=1000,w=30)
save(pTS, file="temp/SynchDifProSoc.RData")

pTS <- PairTSW(Tweets, sel, "frenchValues", R=1000,w=30)
save(pTS, file="temp/SynchFV.RData")

pTS <- PairTSDifW(Tweets, sel, "frenchValues", R=1000,w=30)
save(pTS, file="temp/SynchDifFV.RData")
```

	Model 1
(Intercept)	**0.0365*****
	(0.0005)
PABL	**0.3093*****
	(0.0078)
NABL	**0.3345*****
	(0.0127)
socBL	-0.0096
	(0.0104)
iBL	**0.2411*****
	(0.0076)
log(nTBL)	$-$**0.0015*****
	(0.0001)
AIC	-170257.2162
BIC	-170195.6190
Log Likelihood	85135.6081
Deviance	88.8427
Num. obs.	49001

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 6: Self-selection model based on personality correlates

```
load("temp/userDF.RData")
attach(userDF)

sum(!is.na(nT) & nT>0)
sum(!is.na(nTBL) & nTBL>0)
sum(!is.na(nTpost) & nTpost>0)
sum((!is.na(nT) & nT>0) & (!is.na(nTBL) & nTBL>0) & (!is.na(nTpost) & nTpost>0))

periodUserDF <- subset(userDF, (!is.na(nT) & nT>0) & (!is.na(nTBL) & nTBL>0) & (!is.na(nTpost) & nTpost
save(periodUserDF, file="temp/periodUserDF.RData")
```

4.0.1 Self-selection

```
load("temp/periodUserDF.RData")
periodUserDF %>% dplyr::select(avgEmo, PABL, NABL, socBL, iBL, nTBL) -> cDF

lin <- bayesglm(avgEmo ~ PABL+NABL+socBL+iBL+log(nTBL), data=cDF)

texreg(lin, digits=4, bold=0.05, caption="Self-selection model based on personality correlates")
```

```
confint(lin)
```

```
##                    2.5 %      97.5 %
## (Intercept)  0.035532334  0.037525639
## PABL         0.294195301  0.324772563
## NABL         0.310028013  0.360011595
## socBL       -0.030275168  0.010757290
```

```
## iBL          0.226220989  0.256190957
## log(nTBL)   -0.001774636 -0.001225168
```

```
summ <- summary(lin)
print(paste("Self-selection model R2:", (1-summ$deviance/summ$null.deviance)))
```

```
## [1] "Self-selection model R2: 0.0928057185364379"
```

4.1 Difference time series visualizations

```
load("temp/SynchSoc.RData")

d1 <- as.Date("2015-11-13")
dtbreaks <- c(d1 + seq(-4*7*8,0,by=7*4), d1 + seq(7*8,4*7*8,by=7*4))

Tts <- data.frame(date=pTS$date, mid=pTS$Tmid*100, low=pTS$Tlow*100, hi=pTS$Thi*100)
Fts <- data.frame(date=pTS$date, mid=pTS$Fmid*100, low=pTS$Flow*100, hi=pTS$Fhi*100)
plt <- plotts(Tts, nolog=TRUE,
              zero=FALSE, dtbreaks= "4 weeks", col="red", bgcolor = rgb(1,0,0,0.25))
plt <- plotts(Fts, col="blue", nolog=TRUE, add=TRUE, plt=plt,
             bgcolor= rgb(0,0,1,0.25), zero=FALSE)
plt <- plt + geom_vline(xintercept=as.numeric(as.Date("2015-11-27")))
plt <- plt + scale_x_date("Day", breaks = dtbreaks, date_labels = "%b %d", limits=c(as.Date("2015-06-30
plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.

## Warning: Removed 61 row(s) containing missing values (geom_path).

## Warning: Removed 61 rows containing missing values (geom_point).

## Warning: Removed 61 row(s) containing missing values (geom_path).

## Warning: Removed 61 rows containing missing values (geom_point).
```

```
load("temp/SynchSoc.RData")
refTS <- pTS
load("temp/SynchDifSoc.RData")
pTS$low <- pTS$low/refTS$Flow*100
pTS$mid <- pTS$mid/refTS$Fmid*100
pTS$hi <- pTS$hi/refTS$Fhi*100

plt <- plotts(pTS, nolog=TRUE,  bgcolor= rgb(0.95,0.75,0,0.25), col="darkorange",
              zero=TRUE)
plt <- plt + geom_vline(xintercept=as.numeric(as.Date("2015-11-27")))
plt <- plt + scale_x_date("Day", breaks = dtbreaks, date_labels = "%b %d", limits=c(as.Date("2015-06-30
plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.
```

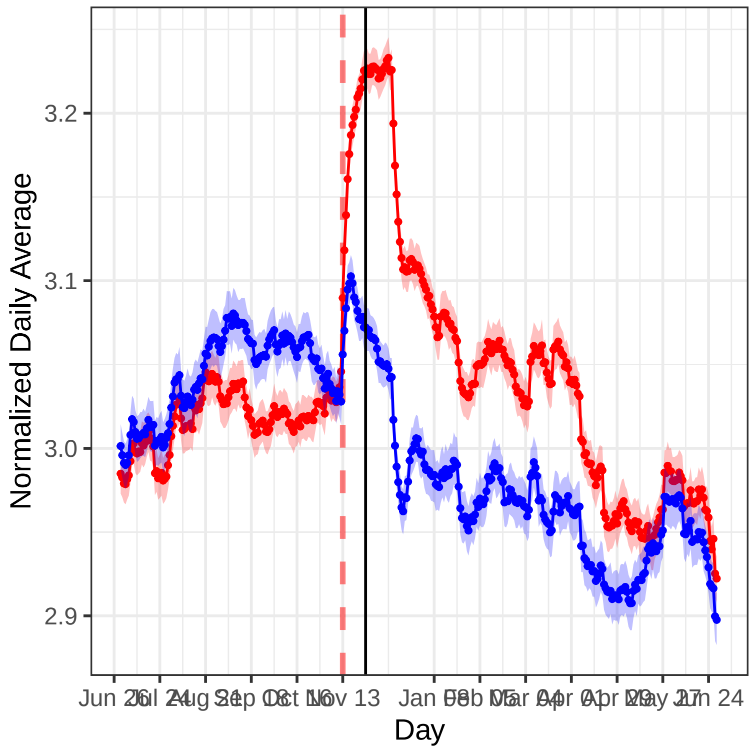


Figure 1: Social Process Terms Synchronization

```
## Warning: Removed 61 row(s) containing missing values (geom_path).

## Warning: Removed 61 rows containing missing values (geom_point).
```

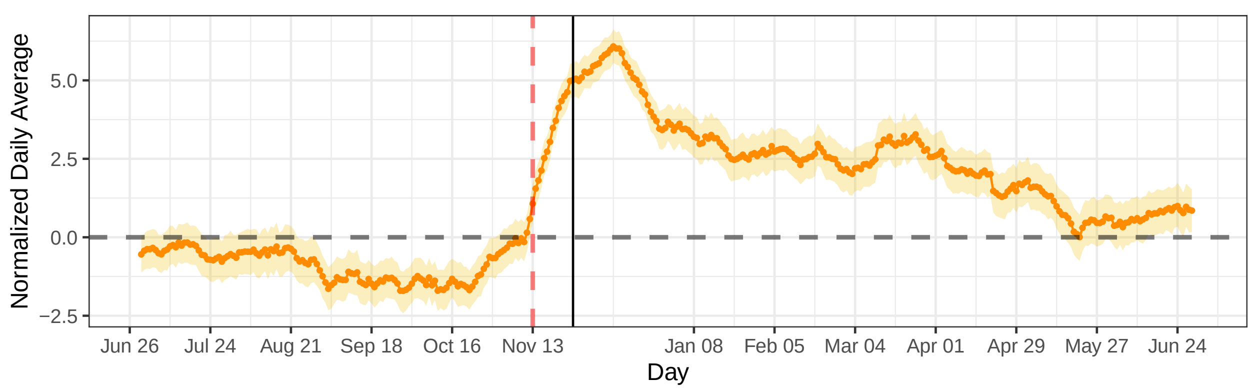


Figure 2: Social Process Terms Synchronization

```
load("temp/SynchProSoc.RData")
d1 <- as.Date("2015-11-13")
dtbreaks <- c(d1 + seq(-4*7*8,0,by=7*8), d1 + seq(7*8,4*7*8,by=7*8))

Tts <- data.frame(date=pTS$date, mid=pTS$Tmid, low=pTS$Tlow, hi=pTS$Thi)
Fts <- data.frame(date=pTS$date, mid=pTS$Fmid, low=pTS$Flow, hi=pTS$Fhi)
plt <- plotts(Tts, nolog=TRUE,
```

```
              zero=FALSE, dtbreaks= "8 weeks", col="red", bgcolor = rgb(1,0,0,0.25))
plt <- plotts(Fts, col="blue", nolog=TRUE, add=TRUE, plt=plt,
             bgcolor= rgb(0,0,1,0.25), zero=FALSE)
plt <- plt + geom_vline(xintercept=as.numeric(as.Date("2015-11-27")))
plt <- plt + scale_x_date("Day", breaks = dtbreaks, date_labels = "%b %d")
plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.
```

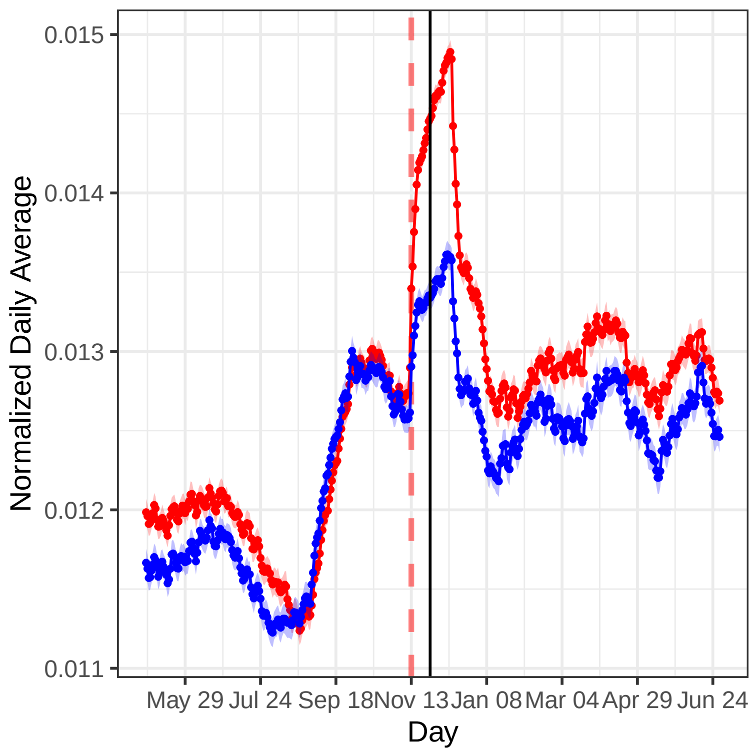


Figure 3: Prosocial Terms Synchronization

```
load("temp/SynchProSoc.RData")
refTS <- pTS
load("temp/SynchDifProSoc.RData")
pTS$low <- pTS$low/refTS$Flow*100
pTS$mid <- pTS$mid/refTS$Fmid*100
pTS$hi <- pTS$hi/refTS$Fhi*100

plt <- plotts(pTS, nolog=TRUE,  bgcolor= rgb(0,0.5,0,0.25), col="darkgreen",
              zero=TRUE)
plt <- plt + geom_vline(xintercept=as.numeric(as.Date("2015-11-27")))
plt <- plt + scale_x_date("Day", breaks = dtbreaks, date_labels = "%b %d", limits=c(as.Date("2015-06-30
plt
```

```
## Warning: Use of `ts$mid` is discouraged. Use `mid` instead.
```

```
## Warning: Removed 61 row(s) containing missing values (geom_path).
```

Warning: Removed 61 rows containing missing values (geom_point).

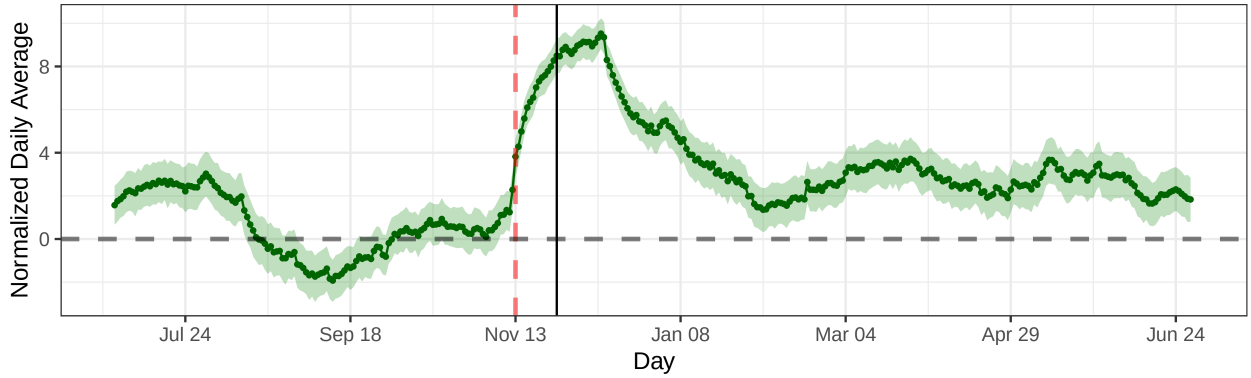


Figure 4: Prosocial Terms Synchronization

```
load("temp/SynchFV.RData")
d1 <- as.Date("2015-11-13")
dtbreaks <- c(d1 + seq(-4*7*8,0,by=7*8), d1 + seq(7*8,4*7*8,by=7*8))

Tts <- data.frame(date=pTS$date, mid=pTS$Tmid, low=pTS$Tlow, hi=pTS$Thi)
Fts <- data.frame(date=pTS$date, mid=pTS$Fmid, low=pTS$Flow, hi=pTS$Fhi)
plt <- plotts(Tts, nolog=TRUE,
              zero=FALSE, dtbreaks= "8 weeks", col="red", bgcolor = rgb(1,0,0,0.25))
plt <- plotts(Fts, col="blue", nolog=TRUE, add=TRUE, plt=plt,
             bgcolor= rgb(0,0,1,0.25), zero=FALSE)

plt <- plt + geom_vline(xintercept=as.numeric(as.Date("2015-11-27")))
plt <- plt + scale_x_date("Day", breaks = dtbreaks, date_labels = "%b %d")
plt
```

Warning: Use of ‘ts$mid‘ is discouraged. Use ‘mid‘ instead.

```
load("temp/SynchFV.RData")
refTS <- pTS
load("temp/SynchDifFV.RData")
pTS$low <- pTS$low/refTS$Flow*100
pTS$mid <- pTS$mid/refTS$Fmid*100
pTS$hi <- pTS$hi/refTS$Fhi*100


plt <- plotts(pTS, nolog=TRUE,  bgcolor= rgb(0,0,0.5,0.25), col="darkblue",
              zero=TRUE)
plt <- plt + geom_vline(xintercept=as.numeric(as.Date("2015-11-27")))
plt <- plt + scale_x_date("Day", breaks = dtbreaks, date_labels = "%b %d", limits=c(as.Date("2015-06-30
plt
```

Warning: Use of ‘ts$mid‘ is discouraged. Use ‘mid‘ instead.

Warning: Removed 61 row(s) containing missing values (geom_path).

Warning: Removed 61 rows containing missing values (geom_point).

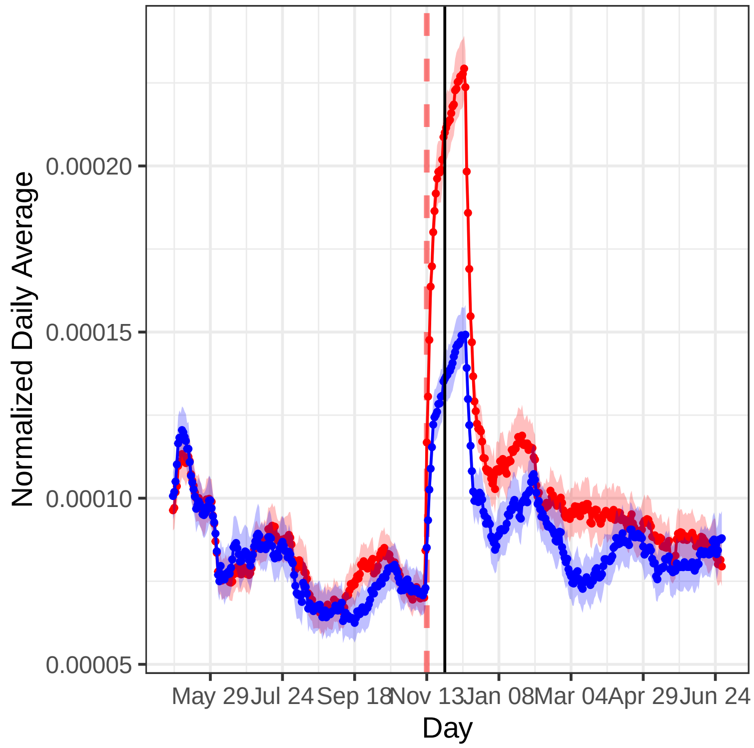


Figure 5: French Value Terms Synchronization

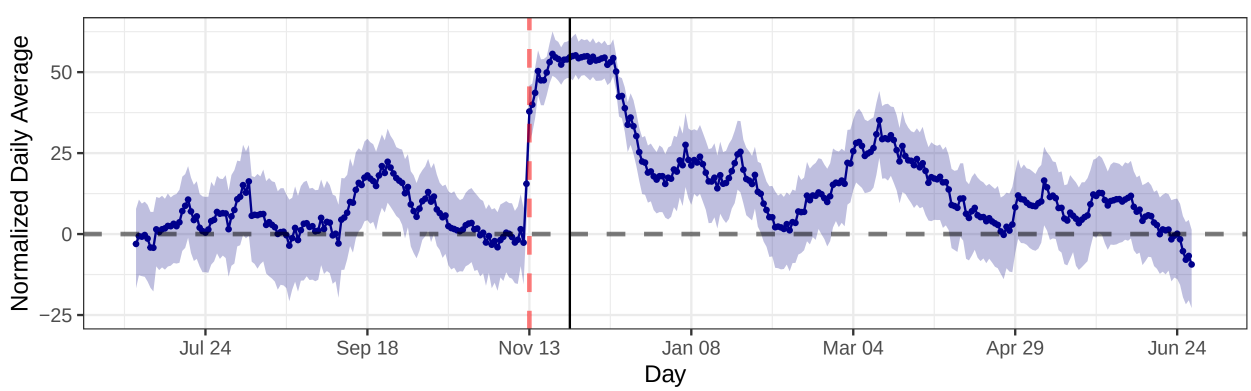


Figure 6: French Value Synchronization

4.2 Mediation analysis

```
library(mediation)
Nrep <- 10000

load("periodUserDF.RData")
periodUserDF %>% dplyr::select(avgEmo, PABL, NABL, socBL, iBL, nTBL) -> cDF
prelin <- glm(avgEmo ~ PABL+NABL+socBL+iBL+log(nTBL), data=cDF)

userDF <- periodUserDF
userDF$emo <- as.numeric(scale(userDF$avgEmo - predict(prelin, newdata=userDF)))
userDF$socBL <- as.numeric(scale(userDF$socBL))
userDF$prosocBL <- as.numeric(scale(userDF$prosocBL))
userDF$FVBL <- as.numeric(scale(userDF$FVBL))
userDF$PABL <- as.numeric(scale(userDF$PABL))
userDF$NABL <- as.numeric(scale(userDF$NABL))
userDF$iBL <- as.numeric(scale(userDF$iBL))
userDF$lnTBL <- as.numeric(scale(log(userDF$nTBL)))

userDF$socPost <- as.numeric(scale(userDF$socPost))
userDF$avgSoc <- as.numeric(scale(userDF$avgSoc))
med.fit <- glm(avgSoc ~ emo + socBL + PABL + NABL + iBL + lnTBL, data = userDF)
out.fit <- glm(socPost ~ avgSoc + emo + socBL + PABL + NABL + iBL + lnTBL, data = userDF)

med.soc <- mediate(med.fit, out.fit, treat = "emo", mediator = "avgSoc", control.value=quantile(userDF$e
save(med.soc, file="temp/med.soc.RData")

userDF$prosocPost <- as.numeric(scale(userDF$prosocPost))
userDF$avgProsoc <- as.numeric(scale(userDF$avgProsoc))
med.fit <- glm(avgProsoc ~ emo + prosocBL + PABL + NABL + iBL + lnTBL, data = userDF)
out.fit <- glm(prosocPost ~ avgProsoc + emo + prosocBL + PABL + NABL + iBL + lnTBL, data = userDF)

med.prosoc <- mediate(med.fit, out.fit, treat = "emo", mediator = "avgProsoc",  control.value=quantile(u
save(med.prosoc, file="temp/med.prosoc.RData")

userDF$FVPost <- as.numeric(scale(userDF$FVPost))
userDF$avgFV <- as.numeric(scale(userDF$avgFV))
med.fit <- glm(avgFV ~ emo + FVBL + PABL + NABL + iBL + lnTBL, data = userDF)
out.fit <- glm(FVPost ~ avgFV + emo + FVBL + PABL + NABL + iBL + lnTBL, data = userDF)

med.FV <- mediate(med.fit, out.fit, treat = "emo", mediator = "avgFV",  control.value=quantile(userDF$em
save(med.FV, file="temp/med.FV.RData")

userDF$PAPost <- as.numeric(scale(userDF$PAPost))
userDF$avgPA <- as.numeric(scale(userDF$avgPA))
med.fit <- glm(avgPA ~ emo + socBL + PABL + NABL + iBL + lnTBL, data = userDF)
out.fit <- glm(PAPost ~ avgPA + emo + socBL + PABL + NABL + iBL + lnTBL, data = userDF)

med.PA <- mediate(med.fit, out.fit, treat = "emo", mediator = "avgPA", control.value=quantile(userDF$emo
save(med.PA, file="temp/med.PA.RData")

userDF$NAPost <- as.numeric(scale(userDF$NAPost))
userDF$avgNA <-as.numeric(scale(userDF$avgNA))
```

```r
med.fit <- glm(avgNA~ emo + socBL + PABL + NABL + iBL + lnTBL, data = userDF)
out.fit <- glm(NAPost ~ avgNA + emo + socBL + PABL + NABL + iBL + lnTBL, data = userDF)

med.NA <- mediate(med.fit, out.fit, treat = "emo", mediator = "avgNA", control.value=quantile(userDF$emo
save(med.NA, file="temp/med.NA.RData")
```

```r
library(mediation)
load("temp/med.soc.RData")
summary(med.soc)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
## (Inference Conditional on the Covariate Values Specified in 'covariates')
##
##                 Estimate 95% CI Lower 95% CI Upper p-value
## ACME              0.0160       0.0131         0.02  <2e-16 ***
## ADE               0.0224       0.0105         0.03   2e-04 ***
## Total Effect      0.0384       0.0267         0.05  <2e-16 ***
## Prop. Mediated    0.4172       0.2960         0.62  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 49001
##
##
## Simulations: 10000
```

```r
load("temp/med.prosoc.RData")
summary(med.prosoc)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
## (Inference Conditional on the Covariate Values Specified in 'covariates')
##
##                 Estimate 95% CI Lower 95% CI Upper p-value
## ACME             0.00922      0.00722         0.01  <2e-16 ***
## ADE              0.01571      0.00442         0.03  0.0076 **
## Total Effect     0.02492      0.01375         0.04  <2e-16 ***
## Prop. Mediated  0.36975      0.23530         0.69  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 49001
##
##
## Simulations: 10000
```

```r
load("temp/med.FV.RData")
summary(med.FV)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
## (Inference Conditional on the Covariate Values Specified in `covariates')
##
##                 Estimate 95% CI Lower 95% CI Upper p-value
## ACME            3.44e-03     4.45e-04         0.01  <2e-16 ***
## ADE            -5.29e-03    -1.56e-02         0.00    0.27
## Total Effect   -1.85e-03    -1.07e-02         0.01    0.67
## Prop. Mediated -1.85e+00    -1.20e+01        11.50    0.67
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 49001
##
##
## Simulations: 10000
```

```r
load("temp/med.PA.RData")
summary(med.PA)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
## (Inference Conditional on the Covariate Values Specified in `covariates')
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME             0.0567       0.0422         0.07  <2e-16 ***
## ADE              0.0678       0.0498         0.09  <2e-16 ***
## Total Effect     0.1245       0.1119         0.14  <2e-16 ***
## Prop. Mediated   0.4556       0.3353         0.58  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 49001
##
##
## Simulations: 10000
```

```r
load("temp/med.NA.RData")
summary(med.NA)
```

```
##
## Causal Mediation Analysis
```

```
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
## (Inference Conditional on the Covariate Values Specified in `covariates')
##
##                Estimate 95% CI Lower 95% CI Upper p-value
## ACME            0.05632      0.04770         0.07  <2e-16 ***
## ADE             0.00522     -0.00718         0.02     0.4
## Total Effect    0.06154      0.05065         0.07  <2e-16 ***
## Prop. Mediated  0.91516      0.74132         1.13  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 49001
##
##
## Simulations: 10000
```

5. Tweet-level analysis

```
library(lmtest)
library(lme4)
```

```
load("Data/TweetPairsDF.RData")

TweetPairsDF$presocB <- TweetPairsDF$presoc>0
TweetPairsDF$preprosocB <- TweetPairsDF$preprosoc>0
TweetPairsDF$prefrenchvaluesB <- TweetPairsDF$prefrenchValues>0

TweetPairsDF$preposemoB <- TweetPairsDF$preposemo>0
TweetPairsDF$prenegemoB <- TweetPairsDF$prenegemo>0
```

```
linSocPos <- glmer((soc>0) ~ presocB * preposemoB + (1|userid), data=TweetPairsDF, family=binomial)
save(linSocPos, file="temp/LinSocPos.RData")

linPosSoc <- glmer((posemo>0) ~ presocB * preposemoB + (1|userid), data=TweetPairsDF, family=binomial)
save(linPosSoc, file="temp/LinPosSoc.RData")

linSocNeg <- glmer((soc>0) ~ prenegemoB * presocB + (1|userid), data=TweetPairsDF, family=binomial)
save(linSocNeg, file="temp/LinSocNeg.RData")

linNegSoc <- glmer((negemo>0) ~ presocB * prenegemoB + (1|userid), data=TweetPairsDF, family=binomial)
save(linNegSoc, file="temp/LinNegSoc.RData")
```

```
load("temp/LinSocPos.RData")
load("temp/LinPosSoc.RData")
load("temp/LinSocNeg.RData")
load("temp/LinNegSoc.RData")
texreg(list(linSocPos, linPosSoc, linSocNeg, linNegSoc), custom.model.names = c("SocP", "PosS", "SocN",
```

	SocP	PosS	SocN	NegS
(Intercept)	$\mathbf{-0.76}^{***}$	$\mathbf{-0.63}^{***}$	$\mathbf{-0.76}^{***}$	$\mathbf{-1.21}^{***}$
	(0.00)	(0.00)	(0.00)	(0.00)
presocBTRUE	$\mathbf{0.20}^{***}$	$\mathbf{0.03}^{***}$	$\mathbf{0.22}^{***}$	$\mathbf{0.03}^{***}$
	(0.01)	(0.01)	(0.01)	(0.01)
preposemoBTRUE	$\mathbf{0.06}^{***}$	$\mathbf{0.15}^{***}$		
	(0.01)	(0.01)		
presocBTRUE:preposemoBTRUE	-0.01	$\mathbf{0.04}^{***}$		
	(0.01)	(0.01)		
prenegemoBTRUE			$\mathbf{0.08}^{***}$	$\mathbf{0.21}^{***}$
			(0.01)	(0.01)
prenegemoBTRUE:presocBTRUE			$\mathbf{-0.06}^{***}$	
			(0.01)	
presocBTRUE:prenegemoBTRUE				0.01
				(0.01)
AIC	1130448.71	1150383.36	1130418.37	996823.83
BIC	1130507.21	1150441.86	1130476.88	996882.33
Log Likelihood	-565219.35	-575186.68	-565204.19	-498406.91
Num. obs.	890994	890994	890994	890994
Num. groups: userid	53004	53004	53004	53004
Var: userid (Intercept)	0.23	0.23	0.23	0.24

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 7: Statistical models

```
confint(linSocPos, method="Wald")
```

```
##                                 2.5 %       97.5 %
## .sig01                             NA           NA
## (Intercept)                -0.76781824 -0.750354550
## presocBTRUE                 0.18336069  0.209017602
## preposemoBTRUE              0.04494223  0.069814515
## presocBTRUE:preposemoBTRUE -0.03076320  0.007950357
```

```
confint(linPosSoc, method="Wald")
```

```
##                                 2.5 %      97.5 %
## .sig01                             NA          NA
## (Intercept)                -0.63606301 -0.61871714
## presocBTRUE                 0.01619627  0.04176321
## preposemoBTRUE              0.13296422  0.15749915
## presocBTRUE:preposemoBTRUE  0.01941860  0.05790618
```

```
confint(linSocNeg, method="Wald")
```

```
##                                 2.5 %      97.5 %
## .sig01                             NA          NA
## (Intercept)                -0.76920436 -0.75226502
## prenegemoBTRUE              0.06852298  0.09528390
## presocBTRUE                 0.20414970  0.22698117
## prenegemoBTRUE:presocBTRUE -0.08524804 -0.04324233
```

```r
confint(linNegSoc, method="Wald")
```

```
##                                  2.5 %      97.5 %
## .sig01                              NA          NA
## (Intercept)                -1.215023173 -1.19673694
## presocBTRUE                 0.018945383  0.04385579
## prenegemoBTRUE              0.198418836  0.22673314
## presocBTRUE:prenegemoBTRUE -0.009080913  0.03535224
```