Replication Material

Facts Shape Feelings: An Information Based Framework for Emotional Responses to Violence

Aidan Milliff

19 Feb 2020

```
if (!require('stm')) install.packages('stm'); library('stm')
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('ggthemes')) install.packages('ggthemes'); library('ggthemes')
if (!require('ggpubr')) install.packages('ggpubr'); library('ggpubr')
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
if (!require('igraph')) install.packages('igraph'); library('igraph')
if (!require('ggraph')) install.packages('ggraph'); library('ggraph')
```

```
load("stm_fit.RData")        # fit_graf
load("processed_text.RData") # paras
load("stm_covars.RData")     # covars
load("rep_data.RData")       # df
```

This file replicates the analysis and figures in "Facts Shape Feelings: An Information Based Framework for Emotional Responses to Violence."

Data and model objects from the replication material should be placed in the same directory as this file in order to compile correctly.

Topic Models

```
# EStimate new stuff
covars    <- left_join(covars, df, by = c("resp" = "ID"))
covars[is.na(covars)] <- 0

known     <- estimateEffect(1:10 ~ Perp.known, fit_graf, metadata = covars)
prosec    <- estimateEffect(1:10 ~ Prosecuted, fit_graf, metadata = covars)
blameperp <- estimateEffect(1:10 ~ Blame.perp, fit_graf, metadata = covars)
blamenone <- estimateEffect(1:10 ~ Blame.noone, fit_graf, metadata = covars)
church    <- estimateEffect(1:10 ~ churchweekly, fit_graf, metadata = covars)

anger     <- estimateEffect(1:10 ~ P1.Angry.x, fit_graf, metadata = covars)
male      <- estimateEffect(1:10 ~ male, fit_graf, metadata = covars)
```

```
# New error here
#### Topic Correlation Plot ####

cor     <- topicCorr(fit_graf, method = "huge")
```

```
Conducting the nonparanormal (npn) transformation via shrunkun ECDF....done.
Conducting Meinshausen & Buhlmann graph estimation (mb)....done
```

```
Conducting rotation information criterion (ric) selection....done
Computing the optimal graph....done
```

```
cor_mat <- cor$cor

colnames(cor_mat) <- c("1: Circumstances",
                       "2: Frust. Detectives",
                       "3: Reason/Motive",
                       "4: Finding Out",
                       "5: Anger/Blame",
                       "6: Motive confusion",
                       "7: Frust. Courts",
                       "8: What Ifs",
                       "9: Support/Community",
                       "10: Panic/Anxiety")
rownames(cor_mat) <-  c("1: Circumstances",
                        "2: Frust. Detectives",
                        "3: Reason/Motive",
                        "4: Finding Out",
                        "5: Anger/Blame",
                        "6: Motive confusion",
                        "7: Frust. Courts",
                        "8: What Ifs",
                        "9: Support/Community",
                        "10: Panic/Anxiety")

expand.grid.unique <- function(x, y, include.equals=FALSE)
{
  x <- unique(x)

  y <- unique(y)

  g <- function(i)
  {
    z <- setdiff(y, x[seq_len(i-include.equals)])

    if(length(z)) cbind(x[i], z, deparse.level=0)
  }

  do.call(rbind, lapply(seq_along(x), g))
}

vec <- as.data.frame(expand.grid.unique(colnames(cor_mat), rownames(cor_mat)))
vec$cor <- c(cor_mat[1,2:10],
             cor_mat[2,3:10],
             cor_mat[3,4:10],
             cor_mat[4,5:10],
             cor_mat[5,6:10],
             cor_mat[6,7:10],
             cor_mat[7,8:10],
             cor_mat[8,9:10],
             cor_mat[9,10])
colnames(vec) <- c("x", "y", "r")
```

```r
vec_sm <- vec %>% filter(x %in% c("3: Reason/Motive", "4: Finding Out", "5: Anger/Blame", "6: Motive co
                         y %in% c("3: Reason/Motive", "4: Finding Out", "5: Anger/Blame", "6: Motive c

graph_cors <-  vec_sm %>%
                    graph_from_data_frame(directed = F)

ggraph(graph_cors) +
  geom_edge_link(aes(color = r,
                     edge_width = abs(r),
                     edge_alpha = abs(r))) +
  geom_node_point(color = "white", size = 5) +
  geom_node_text(aes(label = name), repel = T, vjust = 1) +
  guides(edge_alpha = "none",
         edge_width = "none") +
  scale_edge_colour_gradient2(low = gray.colors(3)[3],
                              mid = gray.colors(3)[2],
                              high = gray.colors(3)[1]) +
  theme_minimal() +
  labs(title = "Topic Correlation Plot")
```

```r
#### New Effects Table ####

# use "plot" to extract predicted probabilities
# this gives expected value of a topic proportion based on a given covariate level
known_plot     <- plot(known, method = "difference", covariate = "Perp.known",
                       cov.value1 = 1, cov.value2 = 0, topics = c(2:10))
```

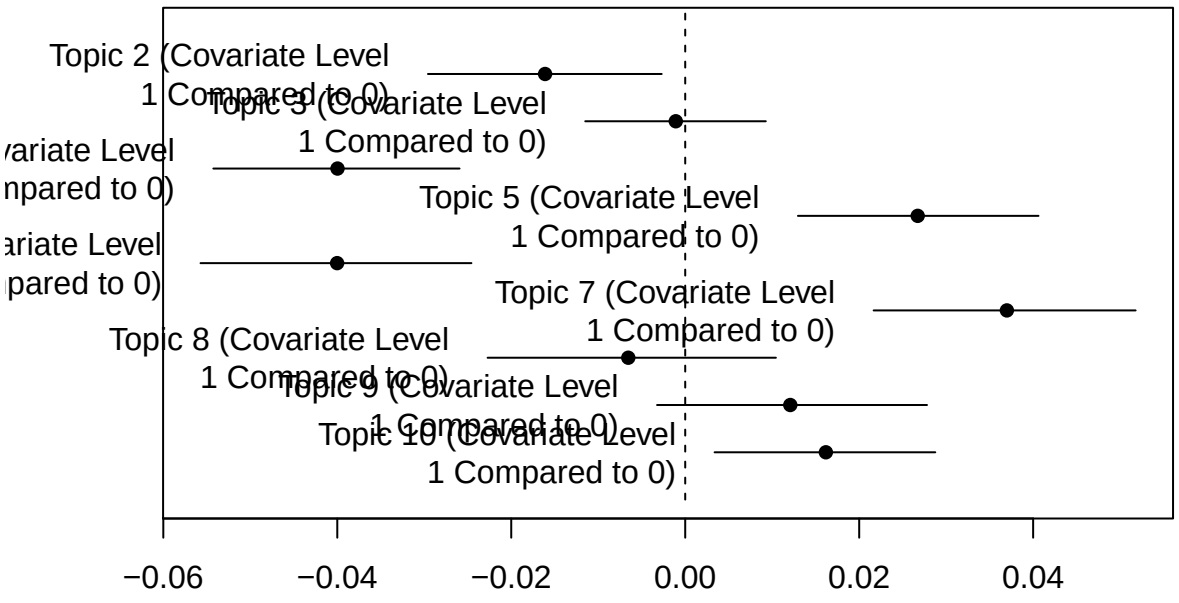

Figure 1: Appendix Figure A3

```
prosec_plot    <- plot(prosec, method = "difference", covariate = "Prosecuted",
                       cov.value1 = 1, cov.value2 = 0, topics = c(2:10))
```

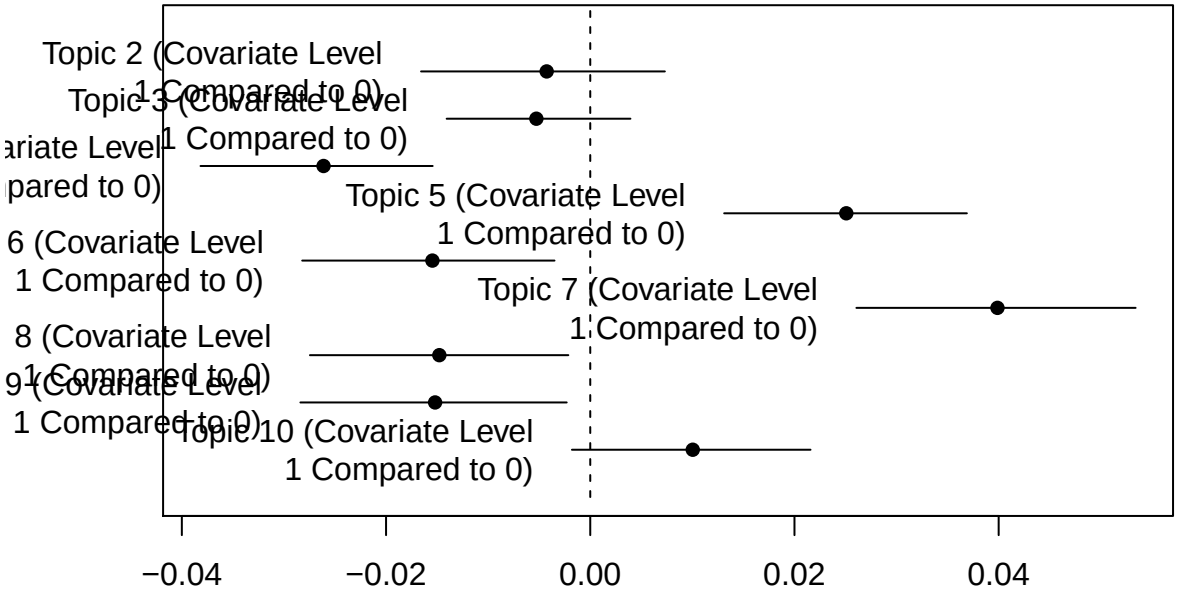

Figure 2: Appendix Figure A3

```
church_plot    <- plot(church, method = "difference", covariate = "churchweekly",
                       cov.value1 = 1, cov.value2 = 0, topics = c(2:10))
```

```
anger_plot     <- plot(anger, method = "difference", covariate = "P1.Angry.x",
                       cov.value1 = 5, cov.value2 = 2, topics = c(2:10))
```

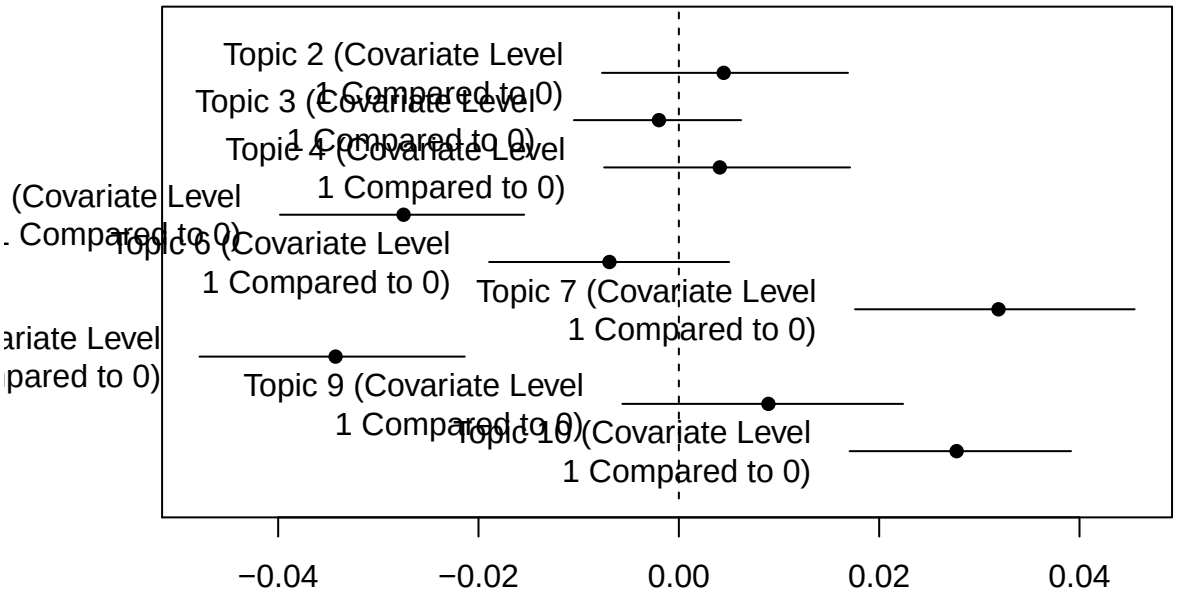

Figure 3: Appendix Figure A3

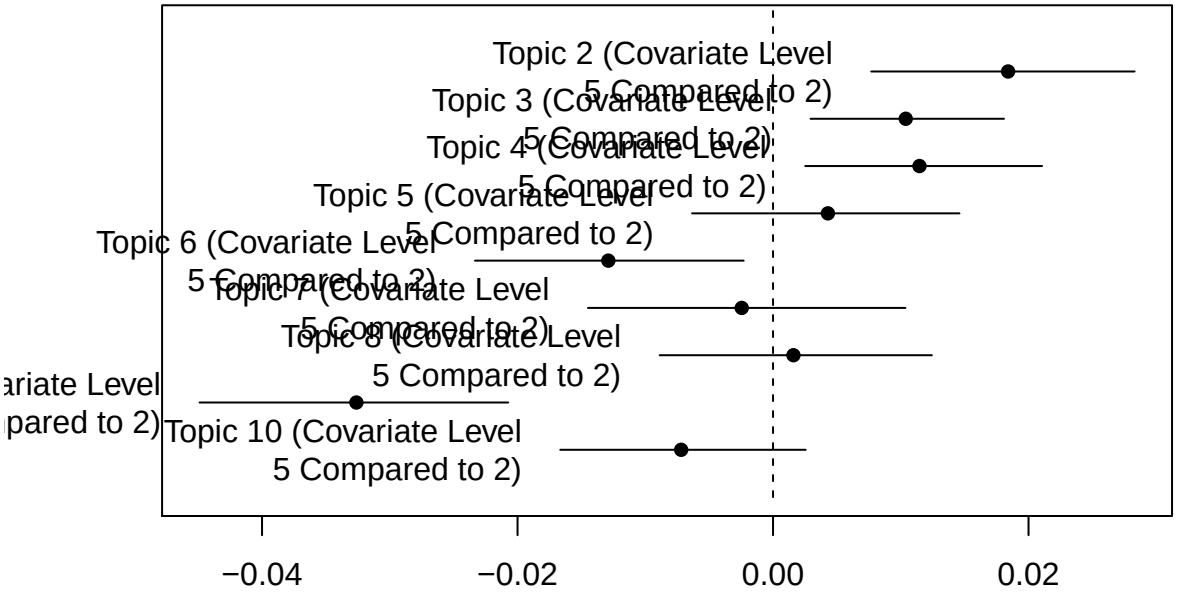

Figure 4: Appendix Figure A3

```
male_plot      <- plot(male, method = "difference",
                       covariate = "male", cov.value1 = 1, cov.value2 = 0, topics = c(2:10))
```

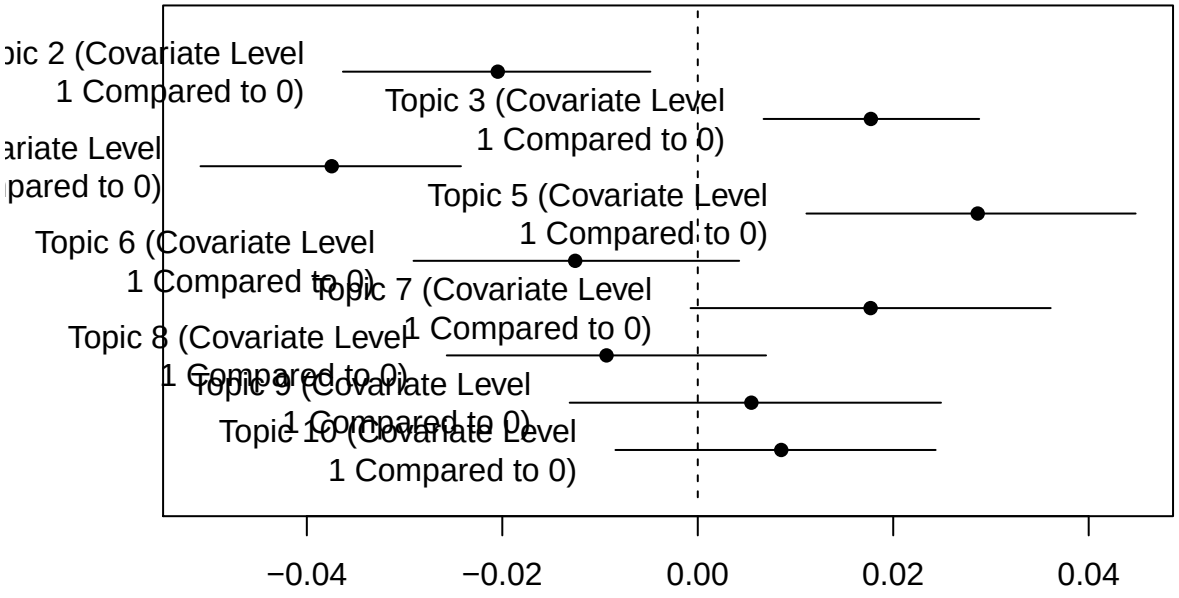

Figure 5: Appendix Figure A3

```
# next, extract predictor, topic, predicted probabilities at different values
known_mat   <- cbind(known_plot$topics, unlist(known_plot$means), unlist(known_plot$cis)[c(1,3,5,7,9,11
prosec_mat  <- cbind(prosec_plot$topics, unlist(prosec_plot$means), unlist(prosec_plot$cis)[c(1,3,5,7,9
church_mat  <- cbind(church_plot$topics, unlist(church_plot$means), unlist(church_plot$cis)[c(1,3,5,7,9
anger_mat   <- cbind(anger_plot$topics, unlist(anger_plot$means), unlist(anger_plot$cis)[c(1,3,5,7,9,11
male_mat    <- cbind(male_plot$topics, unlist(male_plot$means), unlist(male_plot$cis)[c(1,3,5,7,9,11,13
pp_table <- rbind.data.frame(known_mat, prosec_mat, church_mat, anger_mat, male_mat)
colnames(pp_table) <- c("Topic", "Mean", "ci_lo", "ci_hi")
pp_table$predictor <- c(rep("Perpetrator Known?", times = 9),
                        rep("Perpetrator Prosecuted?", times = 9),
                        rep("Weekly Church Attender?", times = 9),
                        rep("Self-Reported Anger (1-5)", times = 9),
                        rep("Male?", times = 9))
pp_table$Topic <- as.factor(pp_table$Topic)
levels(pp_table$Topic) <- c("2: Frust. Detectives",
                            "3: Reason/Motive",
                            "4: Finding Out",
                            "5: Anger/Blame",
                            "6: Motive confusion",
                            "7: Frust. Courts",
                            "8: What Ifs",
                            "9: Support/Cmty.",
                            "10: Panic/Anxiety")
pp_table$sig <- ifelse(pp_table$ci_lo < 0 & 0 < pp_table$ci_hi, "No", "Yes")
pp_table$sig <- as.factor(pp_table$sig)



out <- ggplot(pp_table) +  facet_wrap(~Topic) + geom_point(aes(y = Mean, x = predictor, color = sig, sha
```

```
                    geom_pointrange(aes(ymin = ci_lo, ymax = ci_hi, color = sig, x = predictor, y = Mean
                    theme_minimal() + scale_color_grey()  + coord_flip(expand = T) + scale_x_discrete(e
                    labs(title = "Predicted Difference in Topic Proportions",
                         subtitle = "Change in topic proportions associated with five covariates",
                         y = "Difference in Topic Proportion",
                         x = "Predictor",
                         shape  = "p < 0.05",
                         color = "p < 0.05") +
                          scale_y_continuous(breaks = c(-0.025, 0, 0.025), labels = c(-0.025, 0, 0.025))
out # Figure A3
```

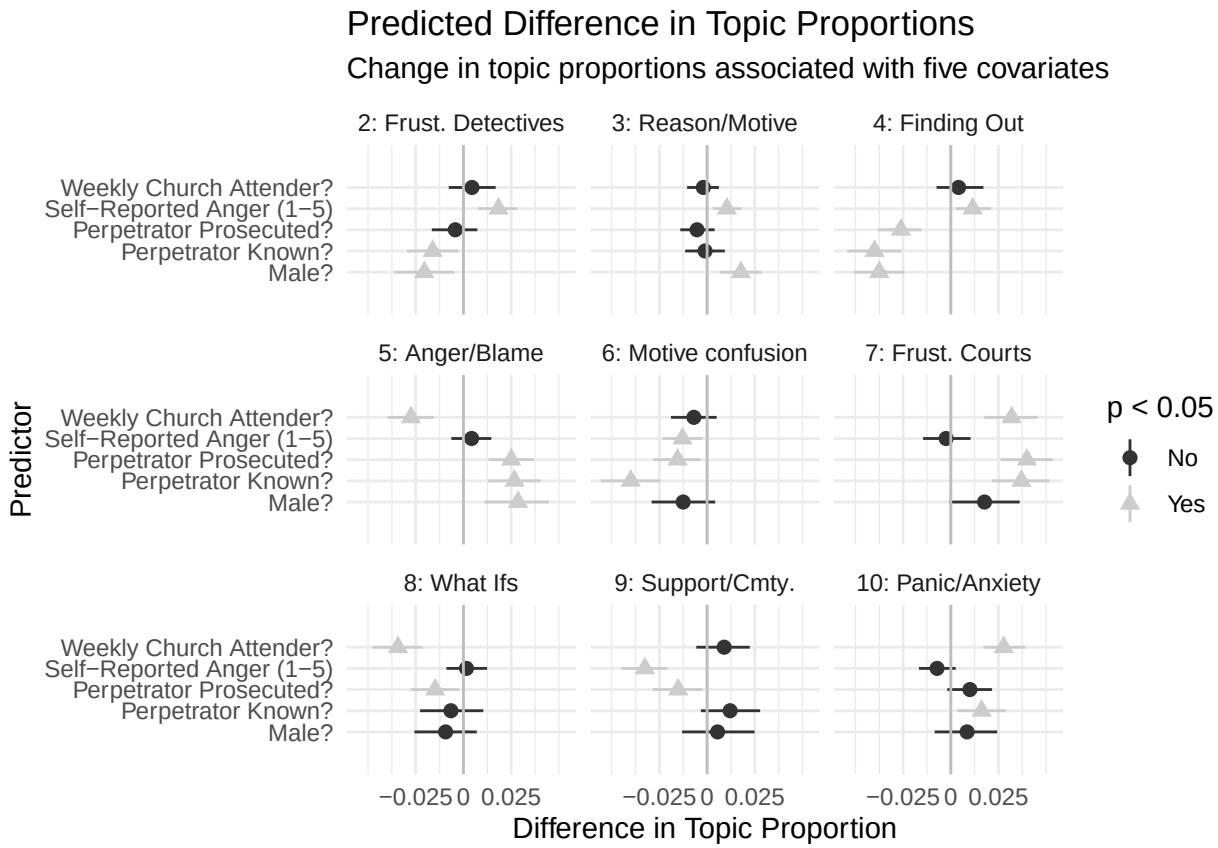

Figure 6: Appendix Figure A3

```
pp_restrict <- pp_table %>% filter(predictor %in% c("Perpetrator Known?", "Self-Reported Anger (1-5)",
                                   Topic %in% c("3: Reason/Motive", "5: Anger/Blame", "6: Motive confu
                                                "4: Finding Out", "10: Panic/Anxiety", "2: Frust. Dete

out_r <- ggplot(pp_restrict) +  facet_wrap(~Topic) + geom_point(aes(y = Mean, x = predictor, color = sig
  geom_pointrange(aes(ymin = ci_lo, ymax = ci_hi, color = sig, x = predictor, y = Mean, shape = sig)) +
  theme_minimal() + scale_color_grey()  +
  coord_flip(expand = T) +
  # scale_x_discrete(expand = c(0,4)) +
  scale_y_continuous(breaks = c(-0.025, 0, 0.025), labels = c("-0.025", "0", "0.025")) +
  theme(axis.text.x = element_text(angle = 60)) +
  labs(title = "Predicted Difference in Topic Proportions",
       subtitle = "Change in topic proportions associated with three covariates",
```

```
       y = "Difference in Topic Proportion",
       x = "Predictor",
       shape  = "p < 0.05",
       color = "p < 0.05")
out_r # Figure 4
```

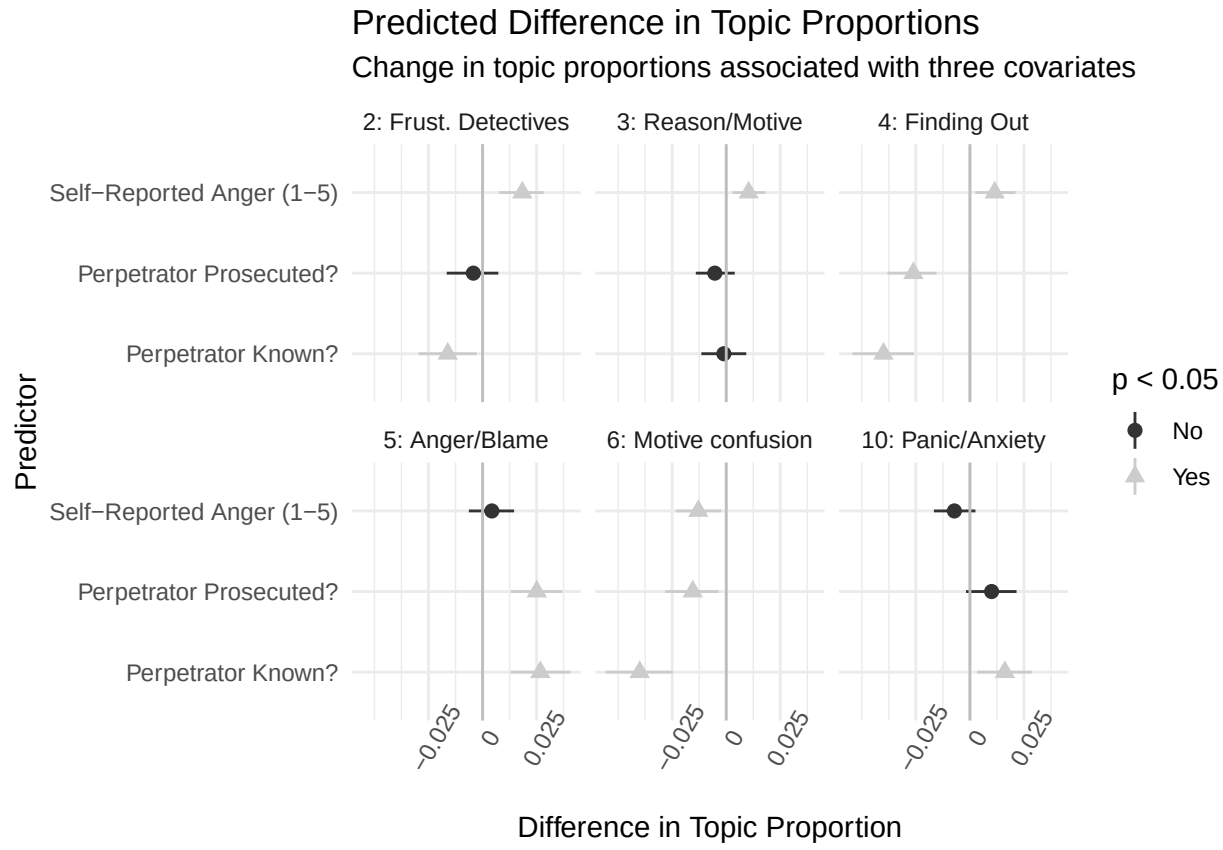

Figure 7: Figure 4

```
topicQuality(fit_graf, paras$documents)
```

```
 [1]  -87.14913  -93.86983  -86.37996 -101.12391  -99.34514 -139.11639
 [7] -118.22352  -95.56930 -113.47943  -89.85831
 [1] 9.230638 9.381033 9.674374 9.456625 9.621877 9.431092 9.123241 9.530275
 [9] 9.336564 9.526114
```

Distribution Plots

```
# Individual Plots

perpknow <- ggplot(data = df,
                   aes(x = P1.Angry,
                       group = Perp.known,
                       fill = factor(Perp.known))) +
            geom_bar(position=position_dodge(preserve = "single"),
                     width=.5) +
```

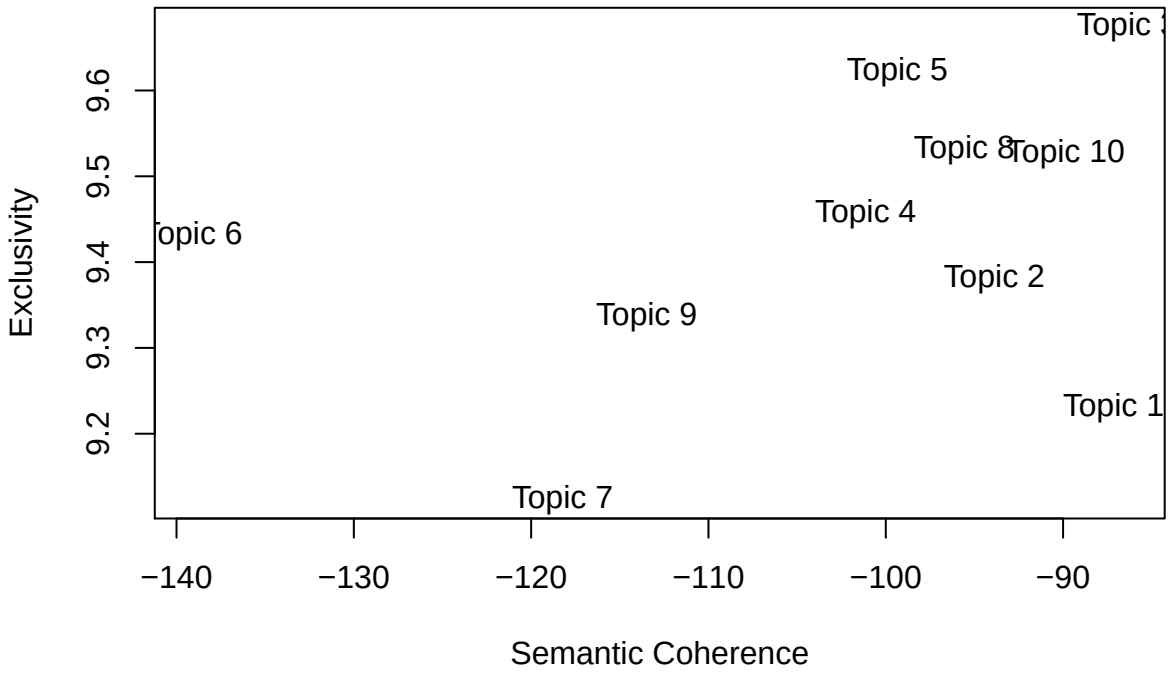

Figure 8: Appendix Figure C1

```
            labs(title = "Knowing the Identity of the Perpetrator",
                 x = "Anger Score",
                 y = "",
                 fill = "Identity\nknown") +
            scale_fill_grey(labels = c("No", "Yes")) +
            theme_minimal()

perpblame <- ggplot(data = df,
                    aes(x = P1.Angry,
                        group = Blame.perp,
                        fill = factor(Blame.perp))) +
             geom_bar(position=position_dodge(preserve = "single"),
                      width=.5) +
             labs(title = "Blaming the Perpetrator",
                  y = "",
                  x = "Anger Score",
                  fill = "Blame\nPerp.") +
             scale_fill_grey(labels = c("No", "Yes")) +
             theme_minimal()

friendblame <- ggplot(data = df,
                      aes(x = P1.Angry,
                          group = Blame.friends,
                          fill = factor(Blame.friends))) +
               geom_bar(position=position_dodge(preserve = "single"),
                        width=.5) +
               labs(title = "Blaming Friends of Victim",
                    y = "",
                    x = "Anger Score",
```

```
                    fill = "Blame\nFriends") +
               scale_fill_grey(labels = c("No", "Yes")) +
               theme_minimal()

vicblame <- ggplot(data = df,
                   aes(x = P1.Angry,
                       group = Blame.deceased,
                       fill = factor(Blame.deceased))) +
             geom_bar(position=position_dodge(preserve = "single"),
                      width=.5) +
             labs(title = "Blaming Victim",
                  x = "Anger Score",
                  y = "",
                  fill = "Blame\nVictim") +
             scale_fill_grey(labels = c("No", "Yes")) +
             theme_minimal()

noblame <- ggplot(data = df,
                  aes(x = P1.Angry,
                      group = Blame.noone,
                      fill = factor(Blame.noone))) +
           geom_bar(position=position_dodge(preserve = "single"),
                    width=.5) +
           labs(title = "Blaming Nobody",
                x = "Anger Score",
                y = "",
                fill = "Blame\nNobody") +
           scale_fill_grey(labels = c("No", "Yes")) +
           theme_minimal()

asarep <- ggplot(data = subset(df, !is.na(Asa.rep)),
                 aes(x = P1.Angry,
                     group = Asa.rep,
                     fill = factor(Asa.rep))) +
          geom_bar(position=position_dodge(preserve = "single"),
                   width=.5) +
          labs(title = "Experience of Justice System",
               x = "Anger Score",
               y = "",
               fill = "Prosec.\nGood?") +
          scale_fill_grey(labels = c("No", "Yes")) +
          theme_minimal()

detgood <- ggplot(data = df,
                  aes(x = P1.Angry,
                      group = det.good,
                      fill = factor(det.good))) +
           geom_bar(position=position_dodge(preserve = "single"),
                    width=.5) +
           labs(title = "Experience of Justice System",
                x = "Anger Score",
                y = "",
                fill = "Detec.\nGood?") +
```

```
            scale_fill_grey(labels = c("No", "Yes")) +
            theme_minimal()

punishv <- ggplot(data = df,
                  aes(x = P1.Angry,
                      group = Punish.viol,
                      fill = factor(Punish.viol))) +
            geom_bar(position=position_dodge(preserve = "single"),
                     width=.5) +
            labs(title = "Punishment Preferences",
                 x = "Anger Score",
                 y = "",
                 fill = "Prefer\nViolence") +
            scale_fill_grey(labels = c("No", "Yes")) +
            theme_minimal()

punishj <- ggplot(data = df,
                  aes(x = P1.Angry,
                      group = Punish.jail,
                      fill = factor(Punish.jail))) +
            geom_bar(position=position_dodge(preserve = "single"),
                     width=.5) +
            labs(title = "Punishment Preferences",
                 x = "Anger Score",
                 y = "",
                 fill = "Prefer\nJail") +
            scale_fill_grey(labels = c("No", "Yes")) +
            theme_minimal()

angertime <- ggplot(data = subset(df),
                    aes(x = Timesince,
                        y = P1.Angry)) +
              geom_density_2d(color = "darkgrey") +
              theme_minimal() +
              geom_point(alpha = 0.4, color = "black") +
              labs(title = "Anger and Time since Homicide",
                   x = "Time since homicide (weeks)",
                   y = "Self Reported Anger (out of five)")

# Combined Plots

allplot1 <- ggarrange(perpknow,
                      perpblame,
                      friendblame,
                      noblame,
                      common.legend = F)

allplot2 <- ggarrange(detgood,
                      punishv,
                      asarep,
                      punishj)

angerplot <- ggarrange(angertime,
```

```
                    perpknow,
                    detgood,
                    asarep,
                    common.legend = F)
```

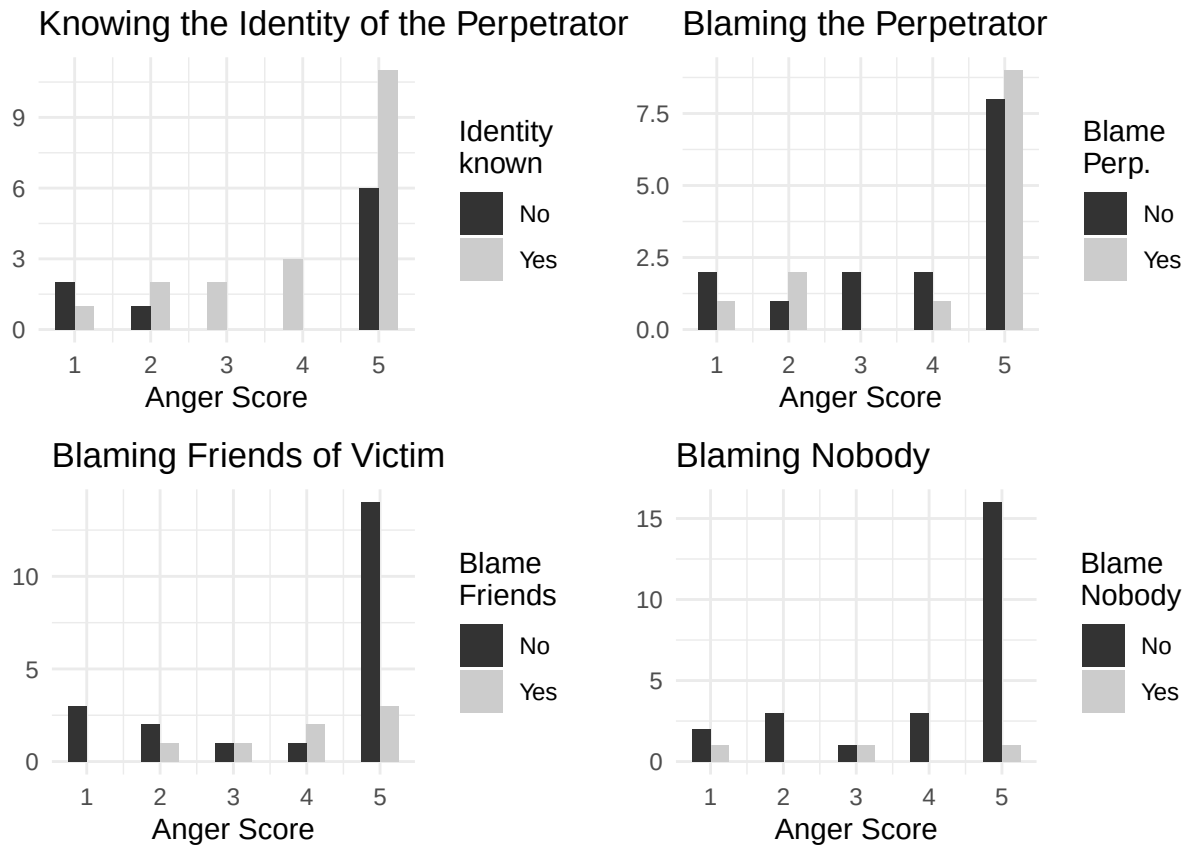

Figure 9: Allplot 1 (Figure 2 in Paper)

Online Appendix D

```
t.test2 <- function(m1,m2,s1,s2,n1,n2,m0=0,equal.variance=FALSE)
{
    if( equal.variance==FALSE )
    {
        se <- sqrt( (s1^2/n1) + (s2^2/n2) )
        # welch-satterthwaite df
        df <- ( (s1^2/n1 + s2^2/n2)^2 )/( (s1^2/n1)^2/(n1-1) + (s2^2/n2)^2/(n2-1) )
    } else
    {
        # pooled standard deviation, scaled by the sample sizes
        se <- sqrt( (1/n1 + 1/n2) * ((n1-1)*s1^2 + (n2-1)*s2^2)/(n1+n2-2) )
        df <- n1+n2-2
    }
    t <- (m1-m2-m0)/se
    dat <- c(m1-m2, se, t, 2*pt(-abs(t),df))
```

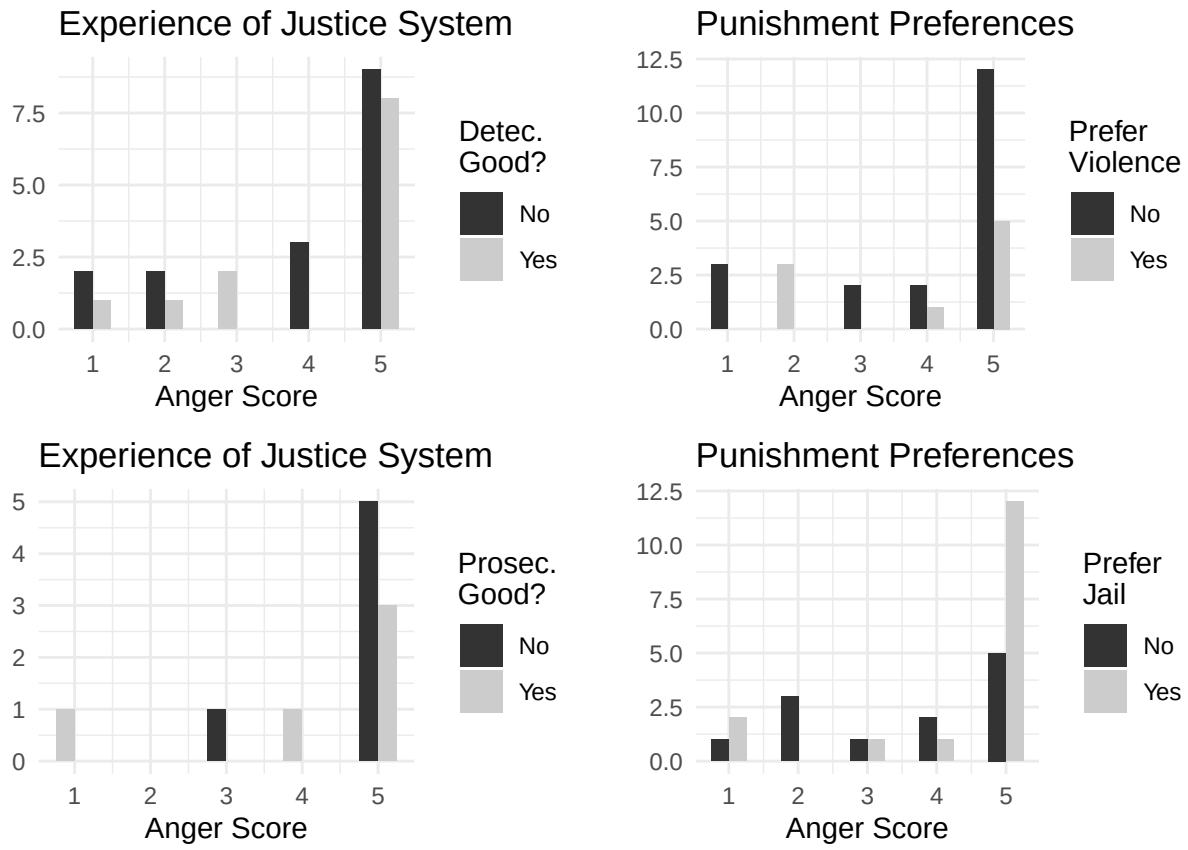

Figure 10: Allplot 2 (Figure A2 in Paper

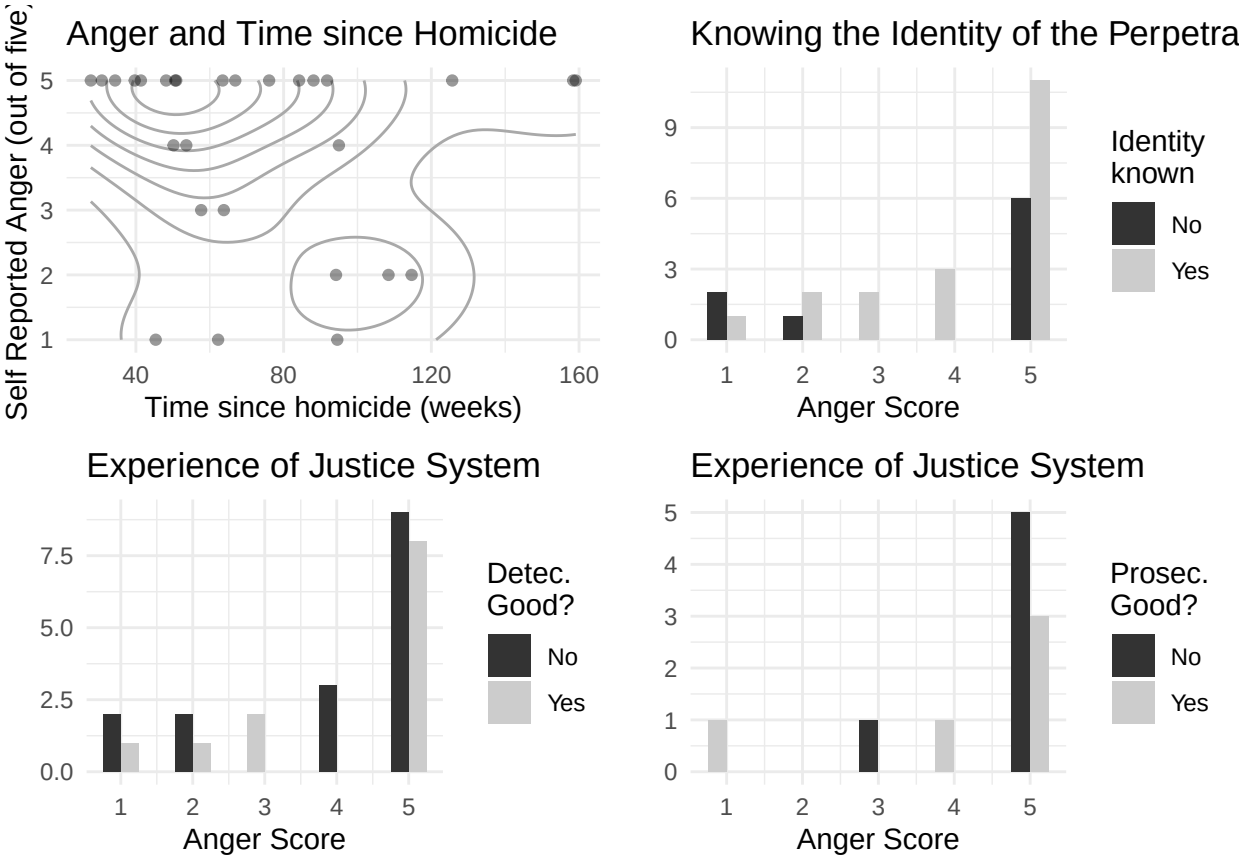

Figure 11: Anger Combined Plot (Figure A1 in Paper)

```
    names(dat) <- c("Difference of means", "Std Error", "t", "p-value")
    return(dat)
}

P1neg <- t.test2(mean(df$P1.Negative.Affect), 17.4, sd(df$P1.Negative.Affect), 6.2, 31, 100)
P1pos <- t.test2(mean(df$P1.Positive.Affect), 33.3, sd(df$P1.Positive.Affect), 7.2, 31, 100)
P2neg <- t.test2(mean(df$P2.Negative.Affect), 17.4, sd(df$P2.Negative.Affect), 6.2, 31, 100)
P2pos <- t.test2(mean(df$P2.Positive.Affect), 33.3, sd(df$P2.Positive.Affect), 7.2, 31, 100)

P12pos   <- t.test2(mean(df$P1.Positive.Affect), mean(df$P2.Positive.Affect), sd(df$P1.Positive.Affect)
P12neg   <- t.test2(mean(df$P1.Negative.Affect), mean(df$P2.Negative.Affect), sd(df$P1.Negative.Affect)

PANAS <- rbind.data.frame(P1neg, P1pos, P2neg, P2pos, P12neg, P12pos)

PANAS <- cbind(c("Present Day", "Present Day", "Post-Homicide", "Post-Homicide", "Post Homicide to Pres
                         c("Negative Affect", "Positive Affect", "Negative Affect", "Positive Affect",
                         PANAS)
colnames(PANAS) <- c("Question", "Valence", "Difference in Means", "Std. Error", "T-score", "P-value")

knitr::kable(PANAS)
```

Question	Valence	Difference in Means	Std. Error	T-score	P-value
Present Day	Negative Affect	10.5677419	2.546019	4.1506918	0.0002110
Present Day	Positive Affect	-10.2354839	2.279875	-4.4894930	0.0000678
Post-Homicide	Negative Affect	10.8580645	3.151280	3.4456042	0.0015939
Post-Homicide	Positive Affect	-18.4612903	2.069311	-8.9214663	0.0000000
Post Homicide to Present Day	Negative Affect	-0.2903226	3.955247	-0.0734019	0.9417423
Post Homicide to Present Day	Positive Affect	8.2258065	2.905698	2.8309228	0.0063278