Regulatory Focus Pride and COVID-19 (Analysis Code)

Emily Nakkawita, Tamar Avnet, & Namika Sagara

Contents

Setup and Data Cleaning

Load Libraries, Themes, and Functions

```
# Load libraries
library(knitr) # Required for knitting
library(readxl) # Required to load .xlsx file
library(plyr) # Required for summarySE() function
library(tidyverse) # Required for data cleaning
library(broman) # Required for myround() function that doesn't truncate digits
library(ggsignif) # Required for plots with significance stars
library(grid) # Required to print plots side by side
library(gridExtra) # Required to print plots side by side

# Load themes
mythemeweb = theme_classic(base_size = 13) +
    theme(legend.position = 'none',
          panel.grid.minor = element_blank(),
          panel.grid.major = element_blank(),
          plot.background = element_rect(fill = "transparent", color = NA),
          panel.background = element_rect(fill = "transparent", color = NA),
          strip.text.x = element_blank(),
          panel.spacing = unit(0.1, "cm"),
          panel.border = element_rect(color="black", fill=NA),
          text = element_text(family="Helvetica", size=13),
```

```
          axis.text.x = element_text(size=11, margin=unit(c(5,0,2,0), "mm")),
          axis.text.y = element_text(size=11, margin=unit(c(0,5,0,0), "mm")),
          axis.ticks = element_line(size = .5),
          axis.ticks.length=unit(-2, "mm"))
theme_set(mythemeweb)

# Define functions
summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                      conf.interval=.95, .drop=TRUE) {

    # New version of length which can handle NA's: if na.rm==T, don't count them
    length2 <- function (x, na.rm=FALSE) {
        if (na.rm) sum(!is.na(x))
        else       length(x)
    }

    # This does the summary. For each group's data frame, return a vector with
    # N, mean, and sd
    datac <- plyr::ddply(data, groupvars, .drop=.drop,
      .fun = function(xx, col) {
        c(N    = length2(xx[[col]], na.rm=na.rm),
          mean = mean   (xx[[col]], na.rm=na.rm),
          sd   = sd     (xx[[col]], na.rm=na.rm)
        )
      },
      measurevar
    )

    # Rename the "mean" column
    datac <- plyr::rename(datac, c("mean" = measurevar))

    datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean

    # Confidence interval multiplier for standard error
    # Calculate t-statistic for confidence interval:
    # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1
    ciMult <- qt(conf.interval/2 + .5, datac$N-1)
    datac$ci <- datac$se * ciMult

    return(datac)
}
```

Load and Clean Data

```
# IMPORT DATA
rf_2019 <- read.csv(file="data/RFQ_Dec2019.csv", header = TRUE)
rf_2020 <- read_excel("data/RAW DATA BeSci Analytic Engine without zipcode.xlsx", col_names = TRUE)

## PREPARE DATA FOR CLEANING AND ANALYSIS

# Select RFQ columns from 2019 data
rf_2019 <- rf_2019[,3:24]
```

```r
# Rename location column
rf_2019 <- dplyr::rename(rf_2019, state = location)

# Remove second header line in 2020 data, then rename and grab numeric scores from columns
rf_2020 <- rf_2020 %>%
  dplyr::filter(RESP_GENDER != "Gender") %>%
  dplyr::rename(gender = RESP_GENDER,
                state = SCREENER1,
                rfq_1 = Q2086001786,
                rfq_2 = Q2073796874,
                rfq_3 = Q2026604185,
                rfq_4 = Q2038294139,
                rfq_5 = Q2097149062,
                rfq_6 = Q2005076527,
                rfq_7 = Q2099986100,
                rfq_8 = Q2014137399,
                rfq_9 = Q2047821109,
                rfq_10 = Q2071138905,
                rfq_11 = Q2082031934,
                ach_pos = Q2005648795,
                prev_neg = Q2064967607,
                rf_moremot = Q2096417839,
                obtainprotect_moremot = Q2061616163,
                foodest_open = Q2078453979,
                foodest_been = Q2015373138,
                foodest_plan = Q2067606070,
                salon_open = Q2029531417,
                salon_been = Q2077930763,
                salon_plan = Q2098211869,
                vitamins = Q2004035326,
                timewithpet = Q2063144300,
                computer_me = Q2071690348__1,
                computer_kids = Q2071690348__2,
                computer_secondhand = Q2071690348__3,
                computer_refurb = Q2071690348__4,
                computer_handdown = Q2071690348__5,
                computer_specupgrade = Q2071690348__6,
                computer_none = Q2071690348__7,
                creditcard = Q2097300961,
                rf_creditcard = Q2035360696,
                lifeins = Q2026863539,
                rf_lifeins = Q2093059946,
                contactless_first = Q2002812802,
                contactless_more = Q2051778461,
                bankapp_first = Q2077157945,
                bankapp_more = Q2060012316,
                rf_shopinperson = Q2001253794,
                target = Q2001547641,
                rf_target = Q2050454100,
                frozenfood = Q2097632970,
                rf_frozenfood = Q2079137883,
                ethnicity = Q2091531022,
                education = Q2057920686,
```

```
                income = Q2014095963,
                pol_orient = Q2066820448,
                agegroup = Q2090985821)
rf_2020$rfq_1 <- str_extract(rf_2020$rfq_1, "^.{1}")
rf_2020$rfq_2 <- str_extract(rf_2020$rfq_2, "^.{1}")
rf_2020$rfq_3 <- str_extract(rf_2020$rfq_3, "^.{1}")
rf_2020$rfq_4 <- str_extract(rf_2020$rfq_4, "^.{1}")
rf_2020$rfq_5 <- str_extract(rf_2020$rfq_5, "^.{1}")
rf_2020$rfq_6 <- str_extract(rf_2020$rfq_6, "^.{1}")
rf_2020$rfq_7 <- str_extract(rf_2020$rfq_7, "^.{1}")
rf_2020$rfq_8 <- str_extract(rf_2020$rfq_8, "^.{1}")
rf_2020$rfq_9 <- str_extract(rf_2020$rfq_9, "^.{1}")
rf_2020$rfq_10 <- str_extract(rf_2020$rfq_10, "^.{1}")
rf_2020$rfq_11 <- str_extract(rf_2020$rfq_11, "^.{1}")

rf_2020$state <- str_sub(rf_2020$state, start= -2)

# Convert RFQ columns to numeric in rf_2020
num <- c("rfq_1", "rfq_2", "rfq_3", "rfq_4", "rfq_5", "rfq_6",
         "rfq_7", "rfq_8", "rfq_9", "rfq_10", "rfq_11") # Define numeric columns
rf_2020[num] <- lapply(rf_2020[num], as.character) # Convert numeric columns to character (before numer
rf_2020[num] <- lapply(rf_2020[num], as.numeric) # Convert numeric columns to numeric
rm(num) # Clear mfraw.num from workspace as it is no longer needed

# Calculate agegroup in 2019 data (consistent with 2020 age group levels)
rf_2019$agegroup <- case_when(rf_2019$age < 25 ~ "18 - 24",
                              rf_2019$age < 35 ~ "25 - 34",
                              rf_2019$age < 45 ~ "35 - 44",
                              rf_2019$age < 55 ~ "45 - 54",
                              rf_2019$age < 65 ~ "55 - 64",
                              rf_2019$age >= 65 ~ "65 or older")
rf_2019$age <- NULL

# Recode demographics in 2019 data to match 2020 coding
rf_2019$income <- case_when(rf_2019$income == "$10K-$20K" ~ "$10,000 - $20,000",
                            rf_2019$income == "$20K-$40K" ~ "$20,001 - $40,000",
                            rf_2019$income == "$40K-$70K" ~ "$40,001 - $70,000",
                            rf_2019$income == "$70K-$100K" ~ "$70,001 - $100,000",
                            rf_2019$income == "$100K-$250K" ~ "$100,001 - $250,000",
                            rf_2019$income == "$250K+" ~ "$250,001+")
rf_2019$education <- case_when(rf_2019$education == "Some High School" ~ "Other",
                            rf_2019$education == "High School" ~ "High School Diploma",
                            rf_2019$education == "Some College" ~ "High School Diploma",
                            rf_2019$education == "Associate's" ~ "Associates Degree",
                            rf_2019$education == "Bachelor's" ~ "Bachelors Degree",
                            rf_2019$education == "Master's" ~ "Graduate Degree",
                            rf_2019$education == "Doctorate" ~ "Graduate Degree",
                            rf_2019$education == "No Response" ~ "Other")
rf_2019$pol_orient <- case_when(rf_2019$pol_orient == "Very Conservative" ~ "Strongly Conservative",
                            rf_2019$pol_orient == "Very Liberal" ~ "Strongly Liberal",
                            rf_2019$pol_orient == "Conservative" ~ "Conservative",
                            rf_2019$pol_orient == "Liberal" ~ "Liberal",
                            rf_2019$pol_orient == "Slightly Conservative" ~ "Slightly Conservative",
```

```
                            rf_2019$pol_orient == "Slightly Liberal" ~ "Slightly Liberal",
                            rf_2019$pol_orient == "Moderate" ~ "Moderate")

# Add study year columns and then merge dataframes
rf_2019$year <- "2019"
rf_2020$year <- "2020"
rf <- bind_rows(rf_2019, rf_2020)
rm(rf_2019, rf_2020)

# Calculate prevention pride and promotion pride scores
rf$prev <- ((6-rf$rfq_2) + (6-rf$rfq_4) + rf$rfq_5 +
              (6-rf$rfq_6) + (6-rf$rfq_8))/5
rf$prom <- ((6-rf$rfq_1) + rf$rfq_3 + rf$rfq_7 + (6-rf$rfq_9) +
              rf$rfq_10 + (6-rf$rfq_11))/6

# Calculate promotion dominance scores
rf$promdom <- rf$prom - rf$prev

# Dummy-code year
rf$year.d <- NA
rf$year.d[rf$year == 2019] <- 0
rf$year.d[rf$year == 2020] <- 1

# Create an ordinal numeric version of the political orientation variable
rf$pol_orient.num <- recode(rf$pol_orient, `Strongly Conservative` = 1,
                            `Conservative` = 2, `Slightly Conservative` = 3,
                            `Moderate` = 4, `Slightly Liberal` = 5,
                            `Liberal` = 6, `Strongly Liberal` = 7, .default = NA_real_)

# Create an ordinal numeric version of the age group variable
rf$agegroup.num <- recode(rf$agegroup, `18 - 24` = 1, `25 - 34` = 2,
                          `35 - 44` = 3, `45 - 54` = 4, `55 - 64` = 5,
                          `65 or older` = 6, .default = NA_real_)

# Dummy-code behavioral outcome variables
rf$foodest_open_certain.d <- case_when(rf$foodest_open == "Yes" ~ 1,
                               rf$foodest_open == "Uncertain" ~ 0,
                               rf$foodest_open == "No" ~ 1)
rf$foodest_been.d <- case_when(rf$foodest_been == "Yes" ~ 1,
                               rf$foodest_been == "Does not apply" ~ 0,
                               rf$foodest_been == "No" ~ 0)
rf$foodest_plan.d <- case_when(rf$foodest_plan == "Yes" ~ 1,
                               rf$foodest_plan == "No" ~ 0)
rf$salon_open_certain.d <- case_when(rf$salon_open == "Yes" ~ 1,
                             rf$salon_open == "Uncertain" ~ 0,
                             rf$salon_open == "No" ~ 1)
rf$salon_been.d <- case_when(rf$salon_been == "Yes" ~ 1,
                             rf$salon_been == "Does not apply" ~ 0,
                             rf$salon_been == "No" ~ 0)
rf$salon_plan.d <- case_when(rf$salon_plan == "Yes" ~ 1,
                             rf$salon_plan == "No" ~ 0)
rf$vitamins_current.d <- case_when(rf$vitamins == "Currently using" ~ 1,
                               rf$vitamins == "Considering using" ~ 0,
```

```
                                rf$vitamins == "Not considering using" ~ 0)
rf$frozenfood.d <- case_when(rf$frozenfood == "Yes" ~ 1,
                             rf$frozenfood == "No" ~ 0)

# Create subsets where needed for analyses
foodest <- filter(rf, foodest_open == "Yes" | foodest_open == "Uncertain") # n = 365
salon <- filter(rf, salon_open == "Yes" | salon_open == "Uncertain") # n = 412

# Subset 2019 and 2020
rf_2019 <- dplyr::filter(rf, year == 2019)
rf_2020 <- dplyr::filter(rf, year == 2020)
```

Participants

2019 Data (Before COVID-19 Pandemic)

```
# Gender
male <- sum(rf_2019$gender == "Male", na.rm = T) # n = 143
female <- sum(rf_2019$gender == "Female", na.rm = T) # n = 135
malepct <- round(male/nrow(rf_2019)*100, digits = 0) # 51%
femalepct <- round(female/nrow(rf_2019)*100, digits = 0) # 48%

# Age Group
eighteen24 <- round(nrow(filter(rf_2019, agegroup == "18 - 24"))/
                      nrow(rf_2019)*100, digits = 0) # 8%
twentyfive34 <- round(nrow(filter(rf_2019, agegroup == "25 - 34"))/
                        nrow(rf_2019)*100, digits = 0) # 42%
thirtyfive44 <- round(nrow(filter(rf_2019, agegroup == "35 - 44"))/
                        nrow(rf_2019)*100, digits = 0) # 29%
fortyfive54 <- round(nrow(filter(rf_2019, agegroup == "45 - 54"))/
                       nrow(rf_2019)*100, digits = 0) # 12%
fiftyfive64 <- round(nrow(filter(rf_2019, agegroup == "55 - 64"))/
                       nrow(rf_2019)*100, digits = 0) # 6%
sixtyfiveplus <- round(nrow(filter(rf_2019, agegroup == "65 or older"))/
                         nrow(rf_2019)*100, digits = 0) # 3%

# Political Orientation
stronglycons <- round(nrow(filter(rf_2019, pol_orient == "Strongly Conservative"))/
                        nrow(rf_2019)*100, digits = 0) # 4%
cons <- round(nrow(filter(rf_2019, pol_orient == "Conservative"))/
                nrow(rf_2019)*100, digits = 0) # 11%
slightlycons <- round(nrow(filter(rf_2019, pol_orient == "Slightly Conservative"))/
                        nrow(rf_2019)*100, digits = 0) # 11%
moderate <- round(nrow(filter(rf_2019, pol_orient == "Moderate"))/
                    nrow(rf_2019)*100, digits = 0) # 18%
slightlylib <- round(nrow(filter(rf_2019, pol_orient == "Slightly Liberal"))/
                       nrow(rf_2019)*100, digits = 0) # 15%
lib <- round(nrow(filter(rf_2019, pol_orient == "Liberal"))/
               nrow(rf_2019)*100, digits = 0) # 22%
stronglylib <- round(nrow(filter(rf_2019, pol_orient == "Strongly Liberal"))/
                       nrow(rf_2019)*100, digits = 0) # 18%
```

2020 Data (After COVID-19 Pandemic)

```
# Gender
male <- sum(rf_2020$gender == "Male", na.rm = T) # n = 294
female <- sum(rf_2020$gender == "Female", na.rm = T) # n = 306
malepct <- round(male/nrow(rf_2020)*100, digits = 0) # 49%
femalepct <- round(female/nrow(rf_2020)*100, digits = 0) # 51%

# Age Group
eighteen24 <- round(nrow(filter(rf_2020, agegroup == "18 - 24"))/
                      nrow(rf_2020)*100, digits = 0) # 8%
twentyfive34 <- round(nrow(filter(rf_2020, agegroup == "25 - 34"))/
                        nrow(rf_2020)*100, digits = 0) # 27%
thirtyfive44 <- round(nrow(filter(rf_2020, agegroup == "35 - 44"))/
                        nrow(rf_2020)*100, digits = 0) # 8%
fortyfive54 <- round(nrow(filter(rf_2020, agegroup == "45 - 54"))/
                       nrow(rf_2020)*100, digits = 0) # 16%
fiftyfive64 <- round(nrow(filter(rf_2020, agegroup == "55 - 64"))/
                       nrow(rf_2020)*100, digits = 0) # 35%
sixtyfiveplus <- round(nrow(filter(rf_2020, agegroup == "65 or older"))/
                         nrow(rf_2020)*100, digits = 0) # 6%

# Political Orientation
stronglycons <- round(nrow(filter(rf_2020, pol_orient == "Strongly Conservative"))/
                        nrow(rf_2020)*100, digits = 0) # 11%
cons <- round(nrow(filter(rf_2020, pol_orient == "Conservative"))/
                nrow(rf_2020)*100, digits = 0) # 16%
slightlycons <- round(nrow(filter(rf_2020, pol_orient == "Slightly Conservative"))/
                        nrow(rf_2020)*100, digits = 0) # 10%
moderate <- round(nrow(filter(rf_2020, pol_orient == "Moderate"))/
                    nrow(rf_2020)*100, digits = 0) # 36%
slightlylib <- round(nrow(filter(rf_2020, pol_orient == "Slightly Liberal"))/
                       nrow(rf_2020)*100, digits = 0) # 8%
lib <- round(nrow(filter(rf_2020, pol_orient == "Liberal"))/
               nrow(rf_2020)*100, digits = 0) # 10%
stronglylib <- round(nrow(filter(rf_2020, pol_orient == "Strongly Liberal"))/
                       nrow(rf_2020)*100, digits = 0) # 9%
```

Regulatory Focus Questionnaire: Summary Statistics

2019 Data (Before COVID-19 Pandemic)

```
prommean <- myround(mean(rf_2019$prom, na.rm = T), digits = 2) # 3.59
promsd <- myround(sd(rf_2019$prom, na.rm = T), digits = 2) # 0.71
prom <- select(rf_2019, rfq_1, rfq_3, rfq_7, rfq_9, rfq_10, rfq_11)
promalpha <- myround(psych::alpha(prom, check.keys = TRUE)$total["raw_alpha"],
                     digits = 2)
# Accounts for rev-scored as expected; alpha = 0.74

prevmean <- myround(mean(rf_2019$prev, na.rm = T), digits = 2) # 3.43
prevsd <- myround(sd(rf_2019$prev, na.rm = T), digits = 2) # 0.84
prev <- select(rf_2019, rfq_2, rfq_4, rfq_5, rfq_6, rfq_8)
prevalpha <- myround(psych::alpha(prev, check.keys = TRUE)$total["raw_alpha"],
```

```
                    digits = 2)
# Accounts for rev-scored as expected; as expected; alpha = 0.84
```

2020 Data (After COVID-19 Pandemic)

```
prommean <- myround(mean(rf_2020$prom, na.rm = T), digits = 2) # 3.46
promsd <- myround(sd(rf_2020$prom, na.rm = T), digits = 2) # 0.62
prom <- select(rf_2020, rfq_1, rfq_3, rfq_7, rfq_9, rfq_10, rfq_11)
promalpha <- myround(psych::alpha(prom, check.keys = TRUE)$total["raw_alpha"],
                    digits = 2)
# Accounts for rev-scored as expected; alpha = 0.63

prevmean <- myround(mean(rf_2020$prev, na.rm = T), digits = 2) # 3.40
prevsd <- myround(sd(rf_2020$prev, na.rm = T), digits = 2) # 0.81
prev <- select(rf_2020, rfq_2, rfq_4, rfq_5, rfq_6, rfq_8)
prevalpha <- myround(psych::alpha(prev, check.keys = TRUE)$total["raw_alpha"],
                    digits = 2)
# Accounts for rev-scored as expected; alpha = 0.80
```

Results

Differences in Regulatory Focus Before Versus After Start of COVID-19 Pandemic

```
prom.year <- lm(prom ~ year.d + agegroup.num + pol_orient.num, data = rf)
summary(prom.year)
```

```
##
## Call:
## lm(formula = prom ~ year.d + agegroup.num + pol_orient.num, data = rf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.17825 -0.41374  0.00355  0.44643  1.61887
##
## Coefficients:
##                Estimate Std. Error t value             Pr(>|t|)
## (Intercept)     3.53602    0.08480  41.699 < 0.0000000000000002 ***
## year.d         -0.22496    0.04911  -4.581          0.00000530 ***
## agegroup.num    0.07459    0.01534   4.864          0.00000137 ***
## pol_orient.num -0.03261    0.01250  -2.609             0.00924 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6401 on 874 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.04561,    Adjusted R-squared:  0.04234
## F-statistic: 13.92 on 3 and 874 DF,  p-value: 0.000000007114
```

```
prom.year.data <- summary(prom.year)$coefficients

prom.year.b <- prom.year.data["year.d", "Estimate"]
```

```r
prom.year.se <- prom.year.data["year.d", "Std. Error"]
prom.year.t <- prom.year.data["year.d", "t value"]
prom.year.df <- summary(prom.year)$df[2]
prom.year.p <- prom.year.data["year.d", "Pr(>|t|)"]

confint(prom.year)
```

```
##                      2.5 %       97.5 %
## (Intercept)     3.36958777  3.702453136
## year.d         -0.32134382 -0.128580123
## agegroup.num    0.04449303  0.104693195
## pol_orient.num -0.05715125 -0.008078047
```

```r
prev.year <- lm(prev ~ year.d + agegroup.num + pol_orient.num, data = rf)
summary(prev.year)
```

```
##
## Call:
## lm(formula = prev ~ year.d + agegroup.num + pol_orient.num, data = rf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.34328 -0.52856 -0.02377  0.56414  1.81838
##
## Coefficients:
##                Estimate Std. Error t value             Pr(>|t|)
## (Intercept)     3.30559    0.10670  30.980 < 0.0000000000000002 ***
## year.d         -0.14103    0.06179  -2.282               0.0227 *
## agegroup.num    0.09819    0.01930   5.088          0.000000442 ***
## pol_orient.num -0.03174    0.01573  -2.018               0.0439 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8055 on 874 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.03611,    Adjusted R-squared:  0.03281
## F-statistic: 10.92 on 3 and 874 DF,  p-value: 0.0000004828
```

```r
prev.year.data <- summary(prev.year)$coefficients

prev.year.b <- prev.year.data["year.d", "Estimate"]
prev.year.se <- prev.year.data["year.d", "Std. Error"]
prev.year.t <- prev.year.data["year.d", "t value"]
prev.year.df <- summary(prev.year)$df[2]
prev.year.p <- prev.year.data["year.d", "Pr(>|t|)"]

confint(prev.year)
```

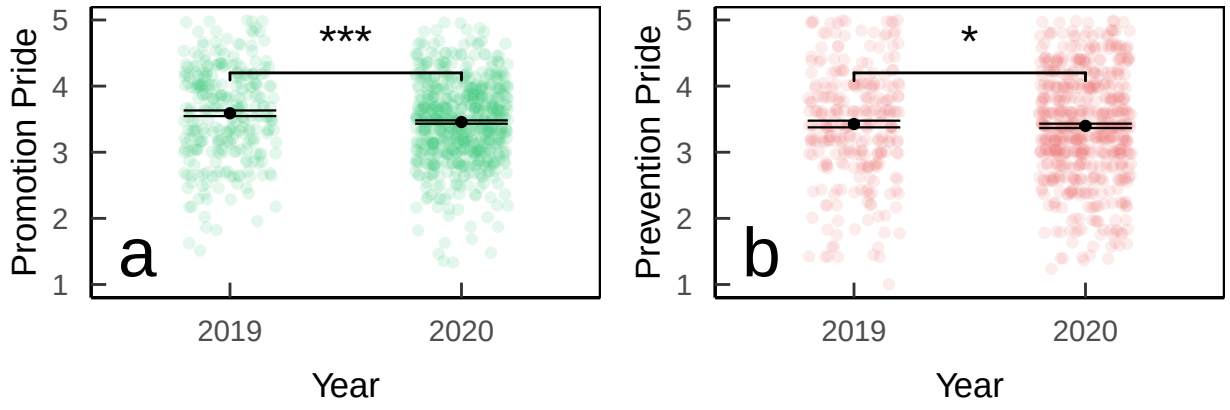
```
##                      2.5 %        97.5 %
## (Intercept)     3.09617079  3.5150076379
## year.d         -0.26230112 -0.0197509862
## agegroup.num    0.06031523  0.1360637160
## pol_orient.num -0.06261117 -0.0008634889
```

```
promsummary <- summarySE(rf, measurevar="prom", groupvars=c("year"))
promplot <- ggplot(data = rf, aes(x = year, y = prom)) +
  geom_point(alpha = .15, color = "seagreen3",
             position = position_jitter(height = .05, width = .2)) +
  geom_point(data = promsummary, color = "black", size = 1.5) +
  geom_errorbar(data = promsummary, aes(ymin = prom - se, ymax = prom + se),
                width = .4, color = "black", size = .4) +
  geom_signif(y_position = c(4.2), xmin = c(1), xmax = c(2), annotation = c("***"),
              textsize = 6, color = "black", vjust = -.4) +
  scale_y_continuous(limits = c(1, 5)) +
  annotate("text", x = .6, y = 1.5, label = "a", size = 9) +
  labs(x = "Year", y = "Promotion Pride")

prevsummary <- summarySE(rf, measurevar="prev", groupvars=c("year"))
prevplot <- ggplot(data = rf, aes(x = year, y = prev)) +
  geom_point(alpha = .15, color = "lightcoral",
             position = position_jitter(height = .05, width = .2)) +
  geom_point(data = prevsummary, color = "black", size = 1.5) +
  geom_errorbar(data = prevsummary, aes(ymin = prev - se, ymax = prev + se),
                width = .4, color = "black", size = .4) +
  geom_signif(y_position = c(4.2), xmin = c(1), xmax = c(2), annotation = c("*"),
              textsize = 6, color = "black", vjust = -.4) +
  scale_y_continuous(limits = c(1, 5)) +
  annotate("text", x = .6, y = 1.5, label = "b", size = 9) +
  labs(x = "Year", y = "Prevention Pride")
```

Figure 1: Differences in promotion pride and prevention pride before (2019) versus after (2020) the start of the COVID-19 pandemic, controlling for participants' age group and political ideology. Error bars represent standard errors of the mean.

```
grid.arrange(promplot, prevplot, ncol=2, heights = c(3, 3), widths = c(4, 4))
```



Associations Between Post-COVID-19 Regulatory Focus Pride and Behavioral Outcomes

Sensitivity to the Opening of the Market

```
foodest_open.rf <- glm(foodest_open_certain.d ~ prom + prev + agegroup.num + pol_orient.num,
                       data = rf, family = "binomial")
summary(foodest_open.rf)
```

```
##
## Call:
## glm(formula = foodest_open_certain.d ~ prom + prev + agegroup.num +
##     pol_orient.num, family = "binomial", data = rf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3202   0.4124   0.4936   0.5561   0.9011
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.8058173  0.8746300   0.921  0.35688
## prom            0.5902284  0.2013805   2.931  0.00338 **
## prev           -0.2969990  0.1599840  -1.856  0.06339 .
## agegroup.num    0.0006424  0.0844314   0.008  0.99393
## pol_orient.num  0.0318325  0.0727413   0.438  0.66167
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 463.66  on 599  degrees of freedom
## Residual deviance: 452.36  on 595  degrees of freedom
##   (279 observations deleted due to missingness)
## AIC: 462.36
##
## Number of Fisher Scoring iterations: 4
```

```
foodest_open.rf.data <- summary(foodest_open.rf)$coefficients

foodest_open.rf.prom.b <- foodest_open.rf.data["prom", "Estimate"]
foodest_open.rf.prom.se <- foodest_open.rf.data["prom", "Std. Error"]
foodest_open.rf.prom.z <- foodest_open.rf.data["prom", "z value"]
foodest_open.rf.prom.df <- summary(foodest_open.rf)$df[2]
foodest_open.rf.prom.p <- foodest_open.rf.data["prom", "Pr(>|z|)"]
foodest_open.rf.prom.or <- exp(foodest_open.rf.prom.b)

foodest_open.rf.prev.b <- foodest_open.rf.data["prev", "Estimate"]
foodest_open.rf.prev.se <- foodest_open.rf.data["prev", "Std. Error"]
foodest_open.rf.prev.z <- foodest_open.rf.data["prev", "z value"]
foodest_open.rf.prev.df <- summary(foodest_open.rf)$df[2]
foodest_open.rf.prev.p <- foodest_open.rf.data["prev", "Pr(>|z|)"]
foodest_open.rf.prev.or <- exp(foodest_open.rf.prev.b)

confint(foodest_open.rf)
```

```
##                      2.5 %     97.5 %
## (Intercept)    -0.8965881 2.53877485
## prom            0.1969949 0.98851938
## prev           -0.6149007 0.01354987
## agegroup.num   -0.1662969 0.16555736
## pol_orient.num -0.1102362 0.17555536
```

```
salon_open.rf <- glm(salon_open_certain.d ~ prom + prev + agegroup.num + pol_orient.num,
                     data = rf, family = "binomial")
```

```r
summary(salon_open.rf)
```

```
##
## Call:
## glm(formula = salon_open_certain.d ~ prom + prev + agegroup.num +
##     pol_orient.num, family = "binomial", data = rf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5015   0.4192   0.4964   0.5765   0.9639
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.242642   0.851754  -0.285  0.77574
## prom            0.531151   0.201949   2.630  0.00854 **
## prev            0.154321   0.154276   1.000  0.31717
## agegroup.num   -0.004856   0.083413  -0.058  0.95357
## pol_orient.num -0.045550   0.071391  -0.638  0.52345
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 471.21  on 599  degrees of freedom
## Residual deviance: 460.86  on 595  degrees of freedom
##   (279 observations deleted due to missingness)
## AIC: 470.86
##
## Number of Fisher Scoring iterations: 4
```

```r
salon_open.rf.data <- summary(salon_open.rf)$coefficients

salon_open.rf.prom.b <- salon_open.rf.data["prom", "Estimate"]
salon_open.rf.prom.se <- salon_open.rf.data["prom", "Std. Error"]
salon_open.rf.prom.z <- salon_open.rf.data["prom", "z value"]
salon_open.rf.prom.df <- summary(salon_open.rf)$df[2]
salon_open.rf.prom.p <- salon_open.rf.data["prom", "Pr(>|z|)"]
salon_open.rf.prom.or <- exp(salon_open.rf.prom.b)

salon_open.rf.prev.b <- salon_open.rf.data["prev", "Estimate"]
salon_open.rf.prev.se <- salon_open.rf.data["prev", "Std. Error"]
salon_open.rf.prev.z <- salon_open.rf.data["prev", "z value"]
salon_open.rf.prev.df <- summary(salon_open.rf)$df[2]
salon_open.rf.prev.p <- salon_open.rf.data["prev", "Pr(>|z|)"]
salon_open.rf.prev.or <- exp(salon_open.rf.prev.b)

confint(salon_open.rf)
```

```
##                     2.5 %     97.5 %
## (Intercept)    -1.9112513 1.43572044
## prom            0.1361121 0.92973974
## prev           -0.1485365 0.45752148
## agegroup.num   -0.1696380 0.15817236
## pol_orient.num -0.1857637 0.09469282
```

Plans to Move Forward with Marketplace Activity

```
foodest_plan.rf <- glm(foodest_plan.d ~ prom + prev + agegroup.num + pol_orient.num,
                       data = foodest, family = "binomial")
summary(foodest_plan.rf)
```

```
##
## Call:
## glm(formula = foodest_plan.d ~ prom + prev + agegroup.num + pol_orient.num,
##     family = "binomial", data = foodest)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7655  -1.0629  -0.7709   1.1896   1.7793
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.65843    0.77987   2.127  0.03346 *
## prom            0.23684    0.18054   1.312  0.18956
## prev           -0.43306    0.14724  -2.941  0.00327 **
## agegroup.num   -0.14028    0.07658  -1.832  0.06698 .
## pol_orient.num -0.20589    0.06615  -3.112  0.00186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.92  on 364  degrees of freedom
## Residual deviance: 477.94  on 360  degrees of freedom
## AIC: 487.94
##
## Number of Fisher Scoring iterations: 4
```

```
foodest_plan.rf.data <- summary(foodest_plan.rf)$coefficients

foodest_plan.rf.prom.b <- foodest_plan.rf.data["prom", "Estimate"]
foodest_plan.rf.prom.se <- foodest_plan.rf.data["prom", "Std. Error"]
foodest_plan.rf.prom.z <- foodest_plan.rf.data["prom", "z value"]
foodest_plan.rf.prom.df <- summary(foodest_plan.rf)$df[2]
foodest_plan.rf.prom.p <- foodest_plan.rf.data["prom", "Pr(>|z|)"]
foodest_plan.rf.prom.or <- exp(foodest_plan.rf.prom.b)

foodest_plan.rf.prev.b <- foodest_plan.rf.data["prev", "Estimate"]
foodest_plan.rf.prev.se <- foodest_plan.rf.data["prev", "Std. Error"]
foodest_plan.rf.prev.z <- foodest_plan.rf.data["prev", "z value"]
foodest_plan.rf.prev.df <- summary(foodest_plan.rf)$df[2]
foodest_plan.rf.prev.p <- foodest_plan.rf.data["prev", "Pr(>|z|)"]
foodest_plan.rf.prev.or <- exp(foodest_plan.rf.prev.b)

confint(foodest_plan.rf)
```

```
##                    2.5 %       97.5 %
## (Intercept)     0.1442521  3.210559396
## prom           -0.1146569  0.595105664
## prev           -0.7264123 -0.147821624
```

```
## agegroup.num   -0.2913703  0.009349337
## pol_orient.num -0.3379263 -0.078061887
```

```
salon_plan.rf <- glm(salon_plan.d ~ prom + prev + agegroup.num + pol_orient.num,
                     data = salon, family = "binomial")
summary(salon_plan.rf)
```

```
##
## Call:
## glm(formula = salon_plan.d ~ prom + prev + agegroup.num + pol_orient.num,
##     family = "binomial", data = salon)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1513  -0.7896  -0.6775  -0.4947   2.1601
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.71942    0.81289   0.885   0.3762
## prom           -0.01036    0.19196  -0.054   0.9570
## prev           -0.27554    0.15296  -1.801   0.0716 .
## agegroup.num   -0.10197    0.07878  -1.294   0.1956
## pol_orient.num -0.14881    0.06811  -2.185   0.0289 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 456.66  on 411  degrees of freedom
## Residual deviance: 446.60  on 407  degrees of freedom
## AIC: 456.6
##
## Number of Fisher Scoring iterations: 4
```

```
salon_plan.rf.data <- summary(salon_plan.rf)$coefficients

salon_plan.rf.prom.b <- salon_plan.rf.data["prom", "Estimate"]
salon_plan.rf.prom.se <- salon_plan.rf.data["prom", "Std. Error"]
salon_plan.rf.prom.z <- salon_plan.rf.data["prom", "z value"]
salon_plan.rf.prom.df <- summary(salon_plan.rf)$df[2]
salon_plan.rf.prom.p <- salon_plan.rf.data["prom", "Pr(>|z|)"]
salon_plan.rf.prom.or <- exp(salon_plan.rf.prom.b)

salon_plan.rf.prev.b <- salon_plan.rf.data["prev", "Estimate"]
salon_plan.rf.prev.se <- salon_plan.rf.data["prev", "Std. Error"]
salon_plan.rf.prev.z <- salon_plan.rf.data["prev", "z value"]
salon_plan.rf.prev.df <- summary(salon_plan.rf)$df[2]
salon_plan.rf.prev.p <- salon_plan.rf.data["prev", "Pr(>|z|)"]
salon_plan.rf.prev.or <- exp(salon_plan.rf.prev.b)

confint(salon_plan.rf)
```

```
##                    2.5 %      97.5 %
## (Intercept)    -0.8757374  2.31995811
## prom           -0.3852245  0.36926721
```

```
## prev           -0.5782633  0.02278656
## agegroup.num   -0.2568290  0.05262116
## pol_orient.num -0.2845339 -0.01694335
```

Behavior Involving Moving Forward with Marketplace Activity

```
foodest_been.rf <- glm(foodest_been.d ~ prom + prev + agegroup.num + pol_orient.num,
                       data = foodest, family = "binomial")
summary(foodest_been.rf)
```

```
##
## Call:
## glm(formula = foodest_been.d ~ prom + prev + agegroup.num + pol_orient.num,
##     family = "binomial", data = foodest)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5752  -0.8975  -0.7247   1.2664   2.0888
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.75582    0.81495   2.155  0.03120 *
## prom            0.08561    0.19117   0.448  0.65426
## prev           -0.42522    0.15459  -2.751  0.00595 **
## agegroup.num   -0.21606    0.08068  -2.678  0.00741 **
## pol_orient.num -0.16043    0.06963  -2.304  0.02123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 462.31  on 364  degrees of freedom
## Residual deviance: 439.32  on 360  degrees of freedom
## AIC: 449.32
##
## Number of Fisher Scoring iterations: 4
```

```
foodest_been.rf.data <- summary(foodest_been.rf)$coefficients

foodest_been.rf.prom.b <- foodest_been.rf.data["prom", "Estimate"]
foodest_been.rf.prom.se <- foodest_been.rf.data["prom", "Std. Error"]
foodest_been.rf.prom.z <- foodest_been.rf.data["prom", "z value"]
foodest_been.rf.prom.df <- summary(foodest_been.rf)$df[2]
foodest_been.rf.prom.p <- foodest_been.rf.data["prom", "Pr(>|z|)"]
foodest_been.rf.prom.or <- exp(foodest_been.rf.prom.b)

foodest_been.rf.prev.b <- foodest_been.rf.data["prev", "Estimate"]
foodest_been.rf.prev.se <- foodest_been.rf.data["prev", "Std. Error"]
foodest_been.rf.prev.z <- foodest_been.rf.data["prev", "z value"]
foodest_been.rf.prev.df <- summary(foodest_been.rf)$df[2]
foodest_been.rf.prev.p <- foodest_been.rf.data["prev", "Pr(>|z|)"]
foodest_been.rf.prev.or <- exp(foodest_been.rf.prev.b)

confint(foodest_been.rf)
```

```
##                     2.5 %      97.5 %
## (Intercept)     0.1726131  3.37744075
## prom           -0.2872567  0.46434290
## prev           -0.7329859 -0.12539267
## agegroup.num   -0.3755748 -0.05869583
## pol_orient.num -0.2993001 -0.02572155
```

```
salon_been.rf <- glm(salon_been.d ~ prom + prev + agegroup.num + pol_orient.num,
                     data = salon, family = "binomial")
summary(salon_been.rf)
```

```
##
## Call:
## glm(formula = salon_been.d ~ prom + prev + agegroup.num + pol_orient.num,
##     family = "binomial", data = salon)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0250  -0.7355  -0.6288  -0.5055   2.1456
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.44341    0.86782  -1.663   0.0963 .
## prom            0.30143    0.20379   1.479   0.1391
## prev           -0.15417    0.15993  -0.964   0.3350
## agegroup.num    0.05226    0.08315   0.628   0.5297
## pol_orient.num -0.16242    0.07103  -2.287   0.0222 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 429.98  on 411  degrees of freedom
## Residual deviance: 420.69  on 407  degrees of freedom
## AIC: 430.69
##
## Number of Fisher Scoring iterations: 4
```

```
salon_been.rf.data <- summary(salon_been.rf)$coefficients

salon_been.rf.prom.b <- salon_been.rf.data["prom", "Estimate"]
salon_been.rf.prom.se <- salon_been.rf.data["prom", "Std. Error"]
salon_been.rf.prom.z <- salon_been.rf.data["prom", "z value"]
salon_been.rf.prom.df <- summary(salon_been.rf)$df[2]
salon_been.rf.prom.p <- salon_been.rf.data["prom", "Pr(>|z|)"]
salon_been.rf.prom.or <- exp(salon_been.rf.prom.b)

salon_been.rf.prev.b <- salon_been.rf.data["prev", "Estimate"]
salon_been.rf.prev.se <- salon_been.rf.data["prev", "Std. Error"]
salon_been.rf.prev.z <- salon_been.rf.data["prev", "z value"]
salon_been.rf.prev.df <- summary(salon_been.rf)$df[2]
salon_been.rf.prev.p <- salon_been.rf.data["prev", "Pr(>|z|)"]
salon_been.rf.prev.or <- exp(salon_been.rf.prev.b)

confint(salon_been.rf)
```

```
##                       2.5 %      97.5 %
## (Intercept)    -3.17074968  0.23945534
## prom           -0.09320969  0.70728221
## prev           -0.46928120  0.15917714
## agegroup.num   -0.10976069  0.21699352
## pol_orient.num -0.30419364 -0.02512089
```

Usage of Vitamin Supplements for Health and Well-Being

```
vitamins_current.rf <- glm(vitamins_current.d ~ prom + prev + agegroup.num + pol_orient.num,
                           data = rf, family = "binomial")
summary(vitamins_current.rf)
```

```
##
## Call:
## glm(formula = vitamins_current.d ~ prom + prev + agegroup.num +
##     pol_orient.num, family = "binomial", data = rf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8517  -1.2139   0.7809   0.9995   1.7297
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.23431    0.63087  -3.542 0.000398 ***
## prom            0.61816    0.14813   4.173  0.00003 ***
## prev            0.05152    0.11054   0.466 0.641202
## agegroup.num    0.15001    0.05858   2.561 0.010440 *
## pol_orient.num -0.05569    0.05071  -1.098 0.272087
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 809.98  on 599  degrees of freedom
## Residual deviance: 773.48  on 595  degrees of freedom
##   (279 observations deleted due to missingness)
## AIC: 783.48
##
## Number of Fisher Scoring iterations: 4
```

```
vitamins_current.rf.data <- summary(vitamins_current.rf)$coefficients

vitamins_current.rf.prom.b <- vitamins_current.rf.data["prom", "Estimate"]
vitamins_current.rf.prom.se <- vitamins_current.rf.data["prom", "Std. Error"]
vitamins_current.rf.prom.z <- vitamins_current.rf.data["prom", "z value"]
vitamins_current.rf.prom.df <- summary(vitamins_current.rf)$df[2]
vitamins_current.rf.prom.p <- vitamins_current.rf.data["prom", "Pr(>|z|)"]
vitamins_current.rf.prom.or <- exp(vitamins_current.rf.prom.b)

vitamins_current.rf.prev.b <- vitamins_current.rf.data["prev", "Estimate"]
vitamins_current.rf.prev.se <- vitamins_current.rf.data["prev", "Std. Error"]
vitamins_current.rf.prev.z <- vitamins_current.rf.data["prev", "z value"]
vitamins_current.rf.prev.df <- summary(vitamins_current.rf)$df[2]
```

```
vitamins_current.rf.prev.p <- vitamins_current.rf.data["prev", "Pr(>|z|)"]
vitamins_current.rf.prev.or <- exp(vitamins_current.rf.prev.b)

confint(vitamins_current.rf)
```

```
##                       2.5 %      97.5 %
## (Intercept)    -3.48723298 -1.01030063
## prom            0.33171631  0.91342005
## prev           -0.16532460  0.26863835
## agegroup.num    0.03541914  0.26526634
## pol_orient.num -0.15537345  0.04367978
```

Purchases Frozen Food

```
frozenfood.rf <- glm(frozenfood.d ~ prom + prev + agegroup.num + pol_orient.num,
                     data = rf, family = "binomial")
summary(frozenfood.rf)
```

```
##
## Call:
## glm(formula = frozenfood.d ~ prom + prev + agegroup.num + pol_orient.num,
##     family = "binomial", data = rf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5300   0.3387   0.3907   0.4423   0.6335
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.80076    1.06396   1.693   0.0905 .
## prom            0.32615    0.24464   1.333   0.1825
## prev           -0.22574    0.19496  -1.158   0.2469
## agegroup.num    0.14633    0.10259   1.426   0.1538
## pol_orient.num -0.05705    0.08877  -0.643   0.5204
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 339.39  on 599  degrees of freedom
## Residual deviance: 333.73  on 595  degrees of freedom
##   (279 observations deleted due to missingness)
## AIC: 343.73
##
## Number of Fisher Scoring iterations: 5
```

```
confint(frozenfood.rf)
```

```
##                       2.5 %    97.5 %
## (Intercept)    -0.25624302 3.9209317
## prom           -0.15644198 0.8043750
## prev           -0.61314740 0.1530661
## agegroup.num   -0.05483912 0.3489626
## pol_orient.num -0.23145284 0.1175661
```