```
R version 4.4.0 (2024-04-24) -- "Puppy Cup"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.80 (8376) aarch64-apple-darwin20]

[History restored from /Users/js122/.Rapp.history]

> ##########################################
> ### LANE AND SCHOENHERR CITE AND SWAY? ###
> ##########################################
>
> # Purpose:
> # - Replicate Figures 2 and 3
>
> # Data Sources:
> # - Analysis2PassivePredProbs.csv
> # - Analysis2SeparatePredProbs.csv
>
> # Run on:
> # - R 4.4.0 ("Puppy Cup") on macOS Sequoia 15.1.1.
>
> # Associated codebook:
> # - Analysis2Codebook.pdf
>
> ##########################################
> ##########################################
> ##########################################
>
> library(tidyverse)
── Attaching core tidyverse packages ──────────────────────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.2     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.1
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
Warning message:
package ‘ggplot2’ was built under R version 4.4.1
> library(ggplot2)
```

```
> library(ggpubr)
>
> ######################
> ### PASSIVE PLOTS ###
> ######################
>
> # Figure 2 in the data
>
> ################################
> ### STEP 1: READ IN THE DATA ###
> ################################
>
> passiveData <- read.csv("~/Dropbox/VanityCitations/JOP\ Final/Dataverse/
Analysis2PassivePredProbs.csv")
>
> ###################################
> ### STEP 2: TURN INTO TWO SETS ###
> ###################################
>
> passivePetData <- passiveData %>% filter(pet == 1)
>
> passivePetUBCIs <- passivePetData %>% select(count, ubLow,ubMed,ubHigh)
> passivePetUBCIs <- passivePetUBCIs %>%  rename(Low = ubLow,Med = ubMed,High = ubHigh)
> passivePetUBCIs <- passivePetUBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align",
values_to = "ub",)
>
> passivePetLBCIs <- passivePetData %>% select(count, lbLow,lbMed,lbHigh)
> passivePetLBCIs <- passivePetLBCIs %>%  rename(Low = lbLow,Med = lbMed,High = lbHigh)
> passivePetLBCIs <- passivePetLBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align",
values_to = "lb",)
>
> passivePetEst <- passivePetData %>% select(count, estLow,estMed,estHigh)
> passivePetEst <- passivePetEst %>%  rename(Low = estLow,Med = estMed,High = estHigh)
> passivePetEst <- passivePetEst %>% pivot_longer(cols = c(Low,Med,High), names_to = "align",
values_to = "est",)
>
> passPetData <- left_join(passivePetEst, passivePetUBCIs, by = c("count", "align"))
> passPetData <- left_join(passPetData, passivePetLBCIs, by = c("count", "align"))
>
> passPetData$align <- factor((passPetData$align), levels=c("Low","Med","High"),
+                             labels=c("Align with Respondent", "Median Justice", "Align with
Petitioner"))
>
> passiveRespData <- passiveData %>% filter(pet == 0)
>
> passiveRespUBCIs <- passiveRespData %>% select(count, ubLow,ubMed,ubHigh)
> passiveRespUBCIs <- passiveRespUBCIs %>%  rename(Low = ubLow,Med = ubMed,High = ubHigh)
> passiveRespUBCIs <- passiveRespUBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align",
values_to = "ub",)
>
> passiveRespLBCIs <- passiveRespData %>% select(count, lbLow,lbMed,lbHigh)
> passiveRespLBCIs <- passiveRespLBCIs %>%  rename(Low = lbLow,Med = lbMed,High = lbHigh)
> passiveRespLBCIs <- passiveRespLBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align",
values_to = "lb",)
>
> passiveRespEst <- passiveRespData %>% select(count, estLow,estMed,estHigh)
> passiveRespEst <- passiveRespEst %>%  rename(Low = estLow,Med = estMed,High = estHigh)
> passiveRespEst <- passiveRespEst %>% pivot_longer(cols = c(Low,Med,High), names_to = "align",
```

```
values_to = "est",)
>
> passRespData <- left_join(passiveRespEst, passiveRespUBCIs, by = c("count", "align"))
> passRespData <- left_join(passRespData, passiveRespLBCIs, by = c("count", "align"))
>
> passRespData$align <- factor((passRespData$align), levels=c("Low","Med","High"),
+                              labels=c("Align with Respondent", "Median Justice", "Align with
Petitioner"))
>
>
> ########################
> ### STEP 3: PET PLOT ###
> ########################
>
> petPlot <- ggplot(passPetData, aes(x = count)) +
+   geom_point(aes(y = est, color = align, shape = align), size = 3) +
+   geom_line(aes(y = est, color = align), size = 1) +
+   geom_ribbon(aes(ymin = lb, ymax = ub, fill = align, color = align), alpha = 0.4, linetype = 0) +
+   theme_light() +
+   ylab("Probability Justice Votes For Petitioner\n") +
+   xlab("\nNumber of Passive Citations to \nJustice's Majority Opinions in Brief") +
+   ggtitle("Petitioner Passive Majority Citation Habits\nby Judicial Alignment with Parties") +
+   theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5)) +
+   scale_y_continuous(breaks=(seq(0.45, 0.80, 0.05)), limits = c(0.45, 0.80), expand = c(0, 0)) +
+   scale_x_continuous(breaks=(seq(0, 11, 1)), limits = c(-.12, 11.12), expand = c(0, 0)) +
+   scale_fill_manual("", values = c("gray40", "gray60", "gray80")) +
+   scale_color_manual("", values = c("gray40", "gray60", "gray80")) +
+   scale_shape_manual("",  values = c(16, 15, 17)) +
+   theme(panel.grid.minor = element_blank()) +
+   theme(legend.position = "bottom") +
+   theme(axis.text.x = element_text(size = 14, hjust = 1),
+         axis.title.x = element_text(size = 14, face = "bold")) +
+   theme(axis.text.y = element_text(size = 14),
+         axis.title.y = element_text(size = 14, face = "bold")) +
+   theme(legend.text = element_text(size = 14))
Warning message:
Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
> petPlot
>
> ########################
> ### STEP 4: RESP PLOT ###
> ########################
>
> respPlot <- ggplot(passRespData, aes(x = count)) +
+   geom_point(aes(y = est, color = align, shape = align), size = 3) +
+   geom_line(aes(y = est, color = align), size = 1) +
+   geom_ribbon(aes(ymin = lb, ymax = ub, fill = align, color = align), alpha = 0.4, linetype = 0) +
+   theme_light() +
+   ylab("Probability Justice Votes For Petitioner\n") +
+   xlab("\nNumber of Passive Citations to \nJustice's Majority Opinions in Brief") +
+   ggtitle("Respondent Passive Majority Citation Habits\nby Judicial Alignment with Parties") +
+   theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5)) +
+   scale_y_continuous(breaks=(seq(0.45, 0.80, 0.05)), limits = c(0.45, 0.80), expand = c(0, 0)) +
+   scale_x_continuous(breaks=(seq(0, 11, 1)), limits = c(-.12, 11.12), expand = c(0, 0)) +
+   scale_fill_manual("", values = c("gray40", "gray60", "gray80")) +
```

```
+   scale_color_manual("", values = c("gray40", "gray60", "gray80")) +
+   scale_shape_manual("",  values = c(16, 15, 17)) +
+   theme(panel.grid.minor = element_blank()) +
+   theme(legend.position = "bottom") +
+   theme(axis.text.x = element_text(size = 14, hjust = 1),
+         axis.title.x = element_text(size = 14, face = "bold")) +
+   theme(axis.text.y = element_text(size = 14),
+         axis.title.y = element_text(size = 14, face = "bold")) +
+   theme(legend.text = element_text(size = 14))
> respPlot
>
>
> ##################################
> ### STEP 5: SAVE PASSIVE PLOT ###
> ##################################
>
> # combine
> passivePlot <- ggarrange(petPlot, respPlot, ncol = 2, common.legend = TRUE, legend = "bottom")
> passivePlot
>
> ggsave(passivePlot, file="~/Dropbox/VanityCitations/JOP\ Final/Figures/fg2.tiff", width = 12, height
= 7)
>
> ######################
> ### SEPARATE PLOTS ###
> ######################
>
> ##################################
> ### STEP 1: READ IN THE DATA ###
> ##################################
>
> separateData <- read.csv("~/Dropbox/VanityCitations/JOP\ Final/Dataverse/
Analysis2SeparatePredProbs.csv")
>
> ##################################
> ### STEP 2: TURN INTO TWO SETS ###
> ##################################
>
> separatePetData <- separateData %>% filter(pet == 1)
>
> petUBCIs <- separatePetData %>% select(count, ubLow,ubMed,ubHigh)
> petUBCIs <- petUBCIs %>%  rename(Low = ubLow,Med = ubMed,High = ubHigh)
> petUBCIs <- petUBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align", values_to = "ub",)
>
> petLBCIs <- separatePetData %>% select(count, lbLow,lbMed,lbHigh)
> petLBCIs <- petLBCIs %>%  rename(Low = lbLow,Med = lbMed,High = lbHigh)
> petLBCIs <- petLBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align", values_to = "lb",)
>
> petEst <- separatePetData %>% select(count, estLow,estMed,estHigh)
> petEst <- petEst %>%  rename(Low = estLow,Med = estMed,High = estHigh)
> petEst <- petEst %>% pivot_longer(cols = c(Low,Med,High), names_to = "align", values_to = "est",)
>
> sepPetData <- left_join(petEst, petUBCIs, by = c("count", "align"))
> sepPetData <- left_join(sepPetData, petLBCIs, by = c("count", "align"))
>
> sepPetData$align <- factor((sepPetData$align), levels=c("Low","Med","High"),
+                            labels=c("Align with Respondent", "Median Justice", "Align with
Petitioner"))
```

```
>
>
> separateRespData <- separateData %>% filter(pet == 0)
>
> respUBCIs <- separateRespData %>% select(count, ubLow,ubMed,ubHigh)
> respUBCIs <- respUBCIs %>%  rename(Low = ubLow,Med = ubMed,High = ubHigh)
> respUBCIs <- respUBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align", values_to =
"ub",)
>
> respLBCIs <- separateRespData %>% select(count, lbLow,lbMed,lbHigh)
> respLBCIs <- respLBCIs %>%  rename(Low = lbLow,Med = lbMed,High = lbHigh)
> respLBCIs <- respLBCIs %>% pivot_longer(cols = c(Low,Med,High), names_to = "align", values_to =
"lb",)
>
> respEst <- separateRespData %>% select(count, estLow,estMed,estHigh)
> respEst <- respEst %>%  rename(Low = estLow,Med = estMed,High = estHigh)
> respEst <- respEst %>% pivot_longer(cols = c(Low,Med,High), names_to = "align", values_to = "est",)
>
> sepRespData <- left_join(respEst, respUBCIs, by = c("count", "align"))
> sepRespData <- left_join(sepRespData, respLBCIs, by = c("count", "align"))
>
> sepRespData$align <- factor((sepRespData$align), levels=c("Low","Med","High"),
+                       labels=c("Align with Respondent", "Median Justice", "Align with Petitioner"))
>
> ########################
> ### STEP 3: PET PLOT ###
> ########################
>
> petPlot2 <- ggplot(sepPetData, aes(x = count)) +
+   geom_point(aes(y = est, color = align, shape = align), size = 3) +
+   geom_line(aes(y = est, color = align), size = 1) +
+   geom_ribbon(aes(ymin = lb, ymax = ub, fill = align, color = align), alpha = 0.4, linetype = 0) +
+   theme_light() +
+   ylab("Probability Justice Votes For Petitioner\n") +
+   xlab("\nNumber of Active Citations to \nJustice's Separate Opinions in Brief") +
+   ggtitle("Petitioner Separate Active Citation Habits\nby Judicial Alignment with Parties") +
+   theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5)) +
+   scale_y_continuous(breaks=(seq(0.45, 0.80, 0.05)), limits = c(0.45, 0.80), expand = c(0, 0)) +
+   scale_x_continuous(breaks=(seq(0, 2, 1)), limits = c(-.02, 2.02), expand = c(0, 0)) +
+   scale_fill_manual("", values = c("gray40", "gray60", "gray80")) +
+   scale_color_manual("", values = c("gray40", "gray60", "gray80")) +
+   scale_shape_manual("",  values = c(16, 15, 17)) +
+   theme(panel.grid.minor = element_blank()) +
+   theme(legend.position = "bottom") +
+   theme(axis.text.x = element_text(size = 14, hjust = 1),
+         axis.title.x = element_text(size = 14, face = "bold")) +
+   theme(axis.text.y = element_text(size = 14),
+         axis.title.y = element_text(size = 14, face = "bold")) +
+   theme(legend.text = element_text(size = 14))
> petPlot2
>
> ########################
> ### STEP 4: RESP PLOT ###
> ########################
>
> respPlot2 <- ggplot(sepRespData, aes(x = count)) +
+   geom_point(aes(y = est, color = align, shape = align), size = 3) +
+   geom_line(aes(y = est, color = align), size = 1) +
```

```
+   geom_ribbon(aes(ymin = lb, ymax = ub, fill = align, color = align), alpha = 0.4, linetype = 0) +
+   theme_light() +
+   ylab("Probability Justice Votes For Petitioner\n") +
+   xlab("\nNumber of Active Citations to \nJustice's Separate Opinions in Brief") +
+   ggtitle("Respondent Separate Active Citation Habits\nby Judicial Alignment with Parties") +
+   theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5)) +
+   scale_y_continuous(breaks=(seq(0.45, 0.80, 0.05)), limits = c(0.45, 0.80), expand = c(0, 0)) +
+   scale_x_continuous(breaks=(seq(0, 2, 1)), limits = c(-.02, 2.02), expand = c(0, 0)) +
+   scale_fill_manual("", values = c("gray40", "gray60", "gray80")) +
+   scale_color_manual("", values = c("gray40", "gray60", "gray80")) +
+   scale_shape_manual("",  values = c(16, 15, 17)) +
+   theme(panel.grid.minor = element_blank()) +
+   theme(legend.position = "bottom") +
+   theme(axis.text.x = element_text(size = 14, hjust = 1),
+         axis.title.x = element_text(size = 14, face = "bold")) +
+   theme(axis.text.y = element_text(size = 14),
+         axis.title.y = element_text(size = 14, face = "bold")) +
+   theme(legend.text = element_text(size = 14))
> respPlot2
>
>
> ###################################
> ### STEP 5: SAVE SEPARATE PLOT ###
> ###################################
>
> # combine
> separatePlot <- ggarrange(petPlot2, respPlot2, ncol = 2, common.legend = TRUE, legend = "bottom")
> separatePlot
>
>
> ggsave(separatePlot, file="~/Dropbox/VanityCitations/JOP\ Final/Figures/fg3.tiff", width = 12,
height = 7)
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
>
```