

```

1  options(stringsAsFactors = FALSE)
2
3  # Initial settings
4  rm(list = ls())
5  library(cjoint)
6  library(dplyr)
7
8  # Assuming that input_data.csv is in the same folder as the script
9  data1 = read.csv("input_data_one_crisis.csv", header = TRUE, check.names=FALSE)
10
11 col_names <- c("Sex", "Interest")
12
13 data1[, col_names] <- lapply(data1[, col_names], factor)
14
15 # Make an attribute list
16 attribute_list <- list()
17 attribute_list[["Sex"]] <- c("Male", "Female")
18 attribute_list[["Interest"]] <- c("voters", "public", "experts", "party")
19
20 # Specify baselines
21 baselines <- list()
22 baselines[["Sex"]] <- c("Male")
23 baselines[["Interest"]] <- c("voters")
24
25 # Specify Conjoint Design
26 conjoint_design <- makeDesign(
27   type="constraints",
28   attribute.levels=attribute_list
29 )
30
31 # Save Names of Attributes and Levels
32 names.only <- do.call("rbind", lapply(attribute_list, as.data.frame))
33 names(names.only)[1] <- "Level"
34 names.only$Attribute <- rownames(names.only)
35 names.only$Attribute <- sub("\\d", "", names.only$Attribute)
36 names.only$Level <- as.character(names.only$Level)
37
38 attributes.only <- names.only %>% select(Attribute) %>% distinct()
39
40 # Estimate and save AMCEs
41 getamce <- function(arg1){
42
43   data.name <- deparse(substitute(arg1))
44
45   results <- amce(
46     Choice ~ Sex + Interest,
47     data = arg1,
48     cluster=TRUE,
49     respondent.id = "R_Id",
50     design = conjoint_design,
51     baselines = baselines,
52     weights = "Weight"
53   )
54
55   results.summary <- summary(results)
56
57   out <- results.summary$amce %>%
58     select(Attribute, Level, Estimate, `Std. Err`) %>%
59     full_join(
60       names.only,
61       by = c("Attribute" = "Attribute", "Level" = "Level")
62     ) %>%
63     bind_rows(attributes.only) %>%
64     arrange(Attribute, Level) %>%
65     mutate(p_raw = 2*pnorm(-abs(`Estimate`/`Std. Err`)),
66            sig_ = (p_raw < 0.05),
67            data = data.name,
68            respondents = results.summary$respondents,
69            order = c(1, 2, 3, 4, 5, 6, 7, 8))

```

```

70     ) %>%
71     arrange(order)
72
73     out$var.names <- paste0(out$Attribute, ":")
74     out$var.names <- ifelse(
75       !is.na(out$Level) & is.na(out$Estimate),
76       paste0("(Baseline = ", out$Level, ")"),
77       out$var.names
78   )
79   out$var.names <- ifelse(
80     !is.na(out$Estimate),
81     paste0(" ", out$Level),
82     out$var.names
83   )
84
85   out$var.order <- paste(out$Attribute, "1")
86   out$var.order <- ifelse(
87     !is.na(out$Level) & is.na(out$Estimate),
88     paste(out$Attribute, "2"),
89     out$var.order
90   )
91   out$var.order <- ifelse(
92     !is.na(out$Estimate),
93     paste(out$Attribute, "3"),
94     out$var.order
95   )
96
97   out <- out[order(out$var.order),]
98   out$order <- 1:nrow(out)
99   out <- out[order(-out$order),]
100  out$order <- 1:nrow(out)
101
102  write.csv(x = out, file = paste0("amce_output_one_crisis.csv"))
103 }
104
105 getamce(data1)
106

```

```

1 # Initial settings
2 rm(list = ls())
3 library(cjoint)
4 library(dplyr)
5 library(ggtext)
6 options(stringsAsFactors = FALSE)
7
8 # Clean data
9 clean_data <- function(arg1){
10
11     data.name <- deparse(substitute(arg1))
12
13     df <- read.csv(paste0("amce_output_one_crisis.csv")) %>%
14     rename(
15         est = `Estimate`,
16         se = `Std..Err`,
17         n = `respondents`,
18         order = `order`
19     )
20
21     df$var.names <- ifelse(df$var.names == "Sex:", "Sex:", df$var.names)
22     df$var.names <- ifelse(df$var.names == "Interest:", "Interest:", df$var.names)
23
24     return(df)
25 }
26
27 data_clean <- clean_data(data1)
28
29 # Save Colour
30 g <- ggplot(data_clean, aes(y = est, x = order, colour = Attribute))
31
32 c <- ggplot_build(g)$data[[1]][["colour"]] %>% distinct(colour)
33
34 default_colour_palett <- c$colour
35 default_colour_palett2 <- c(default_colour_palett, "lightgrey", "black")
36
37
38 # Function for customized theme
39 mytheme <- function(base_size = 13, base_family = "") {
40     theme_grey(
41         base_size = base_size,
42         base_family = base_family
43     ) %+replace%
44     theme(
45         axis.text.x = element_text(
46             size = base_size,
47             colour = "black",
48             hjust = .5 ,
49             vjust = 1
50         ),
51         axis.text.y = element_text(
52             size = base_size,
53             colour = "black",
54             hjust = 0,
55             vjust = 0.5
56         ),
57         axis.ticks = element_line(colour = "grey50"),
58         axis.title.y = element_text(
59             size = base_size,
60             angle = 90,
61             vjust = .01,
62             hjust = .1
63         ),
64         legend.position = "none"
65     )
66 }
67
68 # Make a chart for the main results using all respondents
69 p <- ggplot(data_clean, aes(y = est, x = order, colour = Attribute)) +

```

```

70 coord_flip(ylim = c(-.7, .2), xlim = c(1, 8)) +
71 mytheme(8) +
72 geom_hline(
73   yintercept = 0,
74   linewidth = .5,
75   colour = "darkgrey",
76   linetype = "solid"
77 ) +
78 geom_pointrange(aes(ymin = est - 1.96 * se, ymax = est + 1.96 * se)) +
79 scale_y_continuous(
80   name = "Change in Pr(Preferred Candidate)",
81   breaks = round(seq(-.7, .2, .1), 2),
82   labels = c(
83     "\u20130.7",
84     "\u20130.6",
85     "\u20130.5",
86     "\u20130.4",
87     "\u20130.3",
88     "\u20130.2",
89     "\u20130.1",
90     "0",
91     "0.1",
92     "0.2"
93   )
94 ) +
95 scale_x_continuous(
96   name="",
97   breaks = data_clean$order,
98   labels = data_clean$var.names
99 ) +
100 scale_colour_manual(values = default_colour_palett) +
101 ggtitle(paste0("All respondents (*N* = ", data_clean[1, "n"], ")")) +
102 theme(plot.title = element_markdown())
103
104 ggsave("amce_plot_one_crisis.png", width = 6.5, height = 3)
105

```