```r
options(stringsAsFactors = FALSE)

# Initial settings
rm(list = ls())
library(cjoint)
library(dplyr)

data1 = read.csv("input_data_all_crisis.csv", header = TRUE, check.names=FALSE)

col_names <- c("Sex", "Crisis", "Interest")

data1[, col_names] <- lapply(data1[, col_names], factor)

# Make an attribute list
attribute_list <- list()
attribute_list[["Sex"]] <- c("Male", "Female")
attribute_list[["Crisis"]] <- c("economic", "social", "environmental")
attribute_list[["Interest"]] <- c("voters", "public", "experts", "party")

# Specify baselines
baselines <- list()
baselines[["Sex"]] <- c("Male")
baselines[["Crisis"]] <- c("economic")
baselines[["Interest"]] <- c("voters")

# Specify Conjoint Design
conjoint_design <- makeDesign(
    type="constraints",
    attribute.levels=attribute_list
)

# Save Names of Attributes and Levels
names.only <- do.call("rbind", lapply(attribute_list, as.data.frame))
names(names.only)[1] <- "Level"
names.only$Attribute <- rownames(names.only)
names.only$Attribute <- sub(".\\d", "", names.only$Attribute)
names.only$Level <- as.character(names.only$Level)

attributes.only <- names.only %>%  select(Attribute) %>%  distinct()

# Estimate and save AMCEs
getamce <- function(arg1){

    data.name <- deparse(substitute(arg1))

    results <- amce(
        Choice ~ Sex + Crisis + Interest,
        data = arg1,
        cluster=TRUE,
        respondent.id = "R_Id",
        design = conjoint_design,
        baselines = baselines,
        weights = "Weight"
    )

    results.summary <- summary(results)

    out <- results.summary$amce %>%
        select(Attribute, Level, Estimate, `Std. Err`) %>%
        full_join(
            names.only,
            by = c("Attribute" = "Attribute", "Level" = "Level")
        ) %>%
        bind_rows(attributes.only) %>%
        arrange(Attribute, Level)  %>%
        mutate(
            p_raw = 2*pnorm(-abs(`Estimate`/`Std. Err`)),
            sig   = (p_raw < 0.05),
            data = data.name,
            respondents = results.summary$respondents,
            order = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
        ) %>%
        arrange(order)

    out$var.names <- paste0(out$Attribute,":")
    out$var.names <- ifelse(
        !is.na(out$Level) & is.na(out$Estimate),
        paste0("(Baseline = ", out$Level, ")"),
        out$var.names
    )
    out$var.names <- ifelse(
        !is.na(out$Estimate),
        paste0("   ", out$Level),
        out$var.names
    )

    out$var.order <- paste(out$Attribute, "1")
    out$var.order <- ifelse(
        !is.na(out$Level) & is.na(out$Estimate),
        paste(out$Attribute, "2"),
        out$var.order
    )
    out$var.order <- ifelse(
        !is.na(out$Estimate),
        paste(out$Attribute, "3"),
        out$var.order
    )

    out <- out[order(out$var.order),]
    out$order <- 1:nrow(out)
    out <- out[order(-out$order),]
    out$order <- 1:nrow(out)

    write.csv(x = out, file = paste0("amce_output_all_crisis.csv"))
}

getamce(data1)
```

```r
# Initial settings
rm(list = ls())
library(cjoint)
library(dplyr)
library(ggtext)
options(stringsAsFactors = FALSE)

# Code to clean data
clean_data <- function(arg1){

  data.name <- deparse(substitute(arg1))

  df <- read.csv(paste0("amce_output_all_crisis.csv")) %>%
    rename(
        est = `Estimate`,
        se = `Std..Err`,
        n = `respondents`,
        order = `order`
    )

    df$var.names <- ifelse(df$var.names == "Sex:", "Sex:", df$var.names)
    df$var.names <- ifelse(df$var.names == "Interest:", "Interest:", df$var.names)
    df$var.names <- ifelse(df$var.names == "Crisis:", "Crisis:", df$var.names)

    return(df)
}

data_clean <- clean_data(data1)

# Save Colour
g <- ggplot(data_clean, aes(y = est, x = order, colour = Attribute))

c <- ggplot_build(g)$data[[1]]["colour"] %>% distinct(colour)

default_colour_palett  <- c$colour
default_colour_palett2 <- c(default_colour_palett, "lightgrey", "black")

# Function for customized theme
mytheme <- function(base_size = 13, base_family = "") {
    theme_grey(
        base_size = base_size,
        base_family = base_family
    ) %+replace%
    theme(
        axis.text.x = element_text(
            size = base_size,
            colour = "black",
            hjust = .5 ,
            vjust = 1
        ),
        axis.text.y = element_text(
            size = base_size,
            colour = "black",
            hjust = 0,
            vjust = 0.5
        ),
        axis.ticks = element_line(colour = "grey50"),
        axis.title.y = element_text(
            size = base_size,
            angle = 90,
            vjust = .01,
            hjust = .1
        ),
        legend.position = "none"
    )
}

# Make a chart for the main results using all respondents
p <- ggplot(data_clean, aes(y = est, x = order, colour = Attribute)) +
    coord_flip(ylim = c(-.7, .2), xlim = c(1, 12)) +
    mytheme(8) +
    geom_hline(
        yintercept = 0,
        linewidth = .5,
        colour = "darkgrey",
        linetype = "solid"
    ) +
    geom_pointrange(aes(ymin = est - 1.96 * se, ymax = est + 1.96 * se)) +
    scale_y_continuous(
        name = "Change in Pr(Preferred Candidate)",
        breaks = round(seq(-.7, .2, .1), 2),
        labels = labels = c(
            "\u20130.7",
            "\u20130.6",
            "\u20130.5",
            "\u20130.4",
            "\u20130.3",
            "\u20130.2",
            "\u20130.1",
            "0",
            "0.1",
            "0.2"
        )
    ) +
    scale_x_continuous(
        name="",
        breaks = data_clean$order,
        labels = data_clean$var.names
    ) +
    scale_colour_manual(values = default_colour_palett) +
    ggtitle(paste0("All Respondents (*N* = ", data_clean[1, "n"], ")")) +
    theme(plot.title = element_markdown())

ggsave("amce_plot_all_crisis.png", width = 6.5, height = 4.3)
```