Example ANOVA in Scholl and Molo (2024)

Fabio Molo

We simulate data to run an ANOVA example and generate Figure 1 in Scholl and Molo (2024).

Setup

```
library(ggplot2)
library(ggrepel)
```

Simulate data and run ANOVA

```
n <- 14 # n observations
p <- 8 # p potential confounders
te <- 0 # treatment effect beta_t
coef_size <- 1 # beta_p (taken as constant for all confounders)
distr <- "rnorm" # for standard-normally distributed confounders
confounder_mean <- 1 # mean of potential confounding variables
confounder_sd <- 0.5 # standard deviation of potential confounding variables
```

Function to generate confounders:

```
# function to simulate n observations of p confounding variables
# from a given distribution function
generate_confounders <- function(n, p, distr = "rnorm", ...) {
    res <- sapply(1:p, FUN = function(x) {
        do.call(distr, args = list(
          n = n,
          mean = confounder_mean,
          sd = confounder_sd,
          ...)
          )
        })
}
```

Function to generate individual anovas:

```r
generate_anovas <- function(n, p, te, coef_size, distr) {

    # record the initial random seed that will be used in this function
    seed <- .Random.seed

    # generate potential confounders
    X <- generate_confounders(n = n, p = p, distr = distr)

    # set effect of counfounders
    beta <- rep(coef_size, p)

    # randomize treatment
    treat <- sample(rep(0:1, n / 2))

    # compute the "true" outcome y
    y <- te * treat + X %*% beta

    # fit linear regression model for treatment
    fit_lm <- lm(y ~ treat)

    # conduct analysis of variance
    fit <- anova(fit_lm)

    # get differences in means between groups
    # (may be used for non-binary confounders)
    mean_diff <- apply(
        X, 2, function(x) {
            mean(x[which(treat == 1)]) - mean(x[which(treat == 0)])
            }
        )

    res <- list(
        y = y,
        treat = treat,
        X = X,
        mean_diff = mean_diff,
        var_within = fit["Residuals", "Mean Sq"],
        var_between = fit["treat", "Mean Sq"],
        f = fit["treat", "F value"],
        pval = fit["treat", "Pr(>F)"],
        ci = confint(fit_lm)
        )

    # return random seed as an attribute (analogous to stats:::simulate.lm)
    attr(res, "seed") <- seed
```

```
    return(res)
}
```

We want to show an example with some imbalance of confounders, but not an extreme case. We consider the sum of mean differences in all confounders between treatment and control group as a measure of overall balance. We run 101 simulations. Then we choose the simulation at the 75% quantile (i.e. the '75th-worst case') in terms of overall balance.

```
n_simul <-101
sum_mean_diff <- numeric(n_simul)
seeds <- vector(mode = "list", length = n_simul)

set.seed(1)
for (i in 1:n_simul) {
    anova_sim <- generate_anovas(
        n = n, p = p, te = te, coef_size = coef_size, distr = distr
        )
    sum_mean_diff[i] <- sum(anova_sim$mean_diff)
    seeds[[i]] <- attr(anova_sim, "seed")
}

# retrieve the 75%-quantile case be choosing the seed that
# corresponds to the median sum of mean differences
seed_index <- which(sum_mean_diff == quantile(sum_mean_diff, 0.75))
.Random.seed <- seeds[[seed_index]]
example_anova <- generate_anovas(
  n = n,
  p = p,
  te = te,
  coef_size = coef_size,
  distr = distr
  )

# store data for plotting in a data frame
plot_df <- data.frame(
    treat = ifelse(
        example_anova$treat == 1, "Treatment group", "Control group"
        ),
    y = example_anova$y
    )

group_means <- aggregate(y ~ treat, data = plot_df, FUN = mean)
```

The raw (simulated) experimental data is as follows, with one row corresponding to one observation:

```
print(plot_df[order(plot_df$treat), c("treat", "y")], row.names = FALSE)
```

```
         treat         y
   Control group  7.707023
   Control group  9.150743
   Control group  8.336769
   Control group  8.952040
   Control group 11.080828
   Control group  8.622989
   Control group  6.790455
 Treatment group  8.950577
 Treatment group  9.859927
 Treatment group  6.968693
 Treatment group 10.162250
 Treatment group  8.560251
 Treatment group  8.771956
 Treatment group 10.658256
```

The group means are 8.663 and 9.133 for the control and treatment group, respectively. The ANOVA F-statistic is 0.469 with a p-value of 0.5064. The variance within (i.e. the Residual Mean squares) is 1.649 and the variance between groups is 0.774.

Plot Figure 1

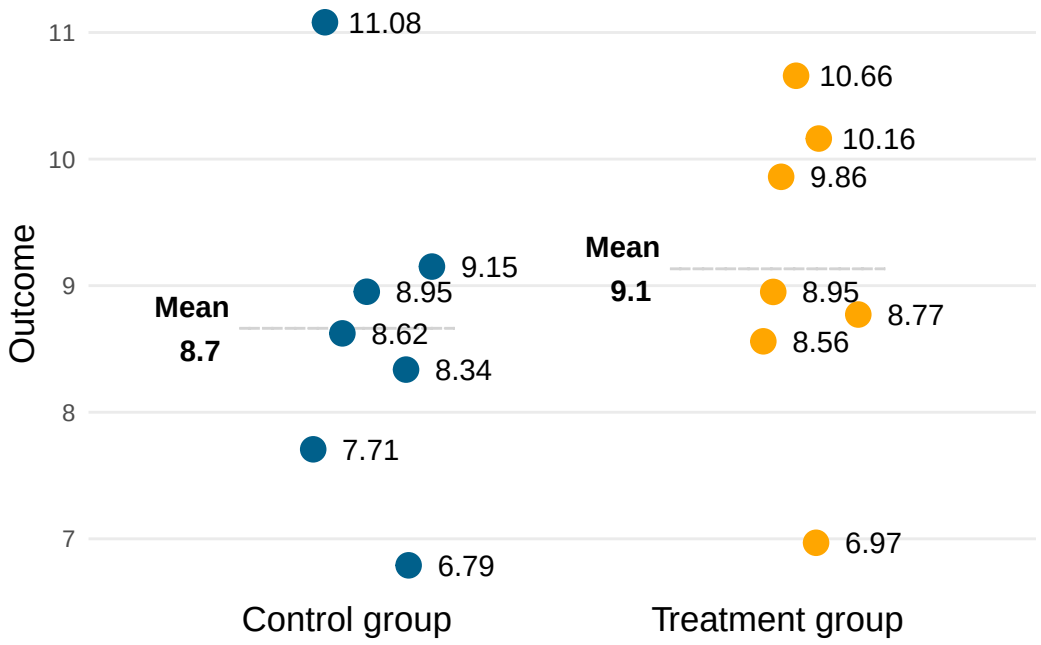
We can now plot the simulated data to obtain Figure 1 of Scholl and Molo (2024).

```
jittersettings <- position_jitter(height = 0, width = 0.2, seed = 22)

fig1 <- ggplot(data = plot_df, aes(x = treat, y = y)) +
    geom_crossbar(
        data = group_means,
        aes(y = y, ymin = y, ymax = y),
        colour = "lightgrey",
        linetype = "dashed",
        alpha = 0.5,
        fatten = 0.5,
        width = 0.5
        ) +
    geom_text(
        data = group_means,
        aes(label = paste("Mean \n", round(y, 1))),
        size = 4,
        nudge_x = -0.35,
        family = "Helvetica",
        fontface = "bold"
        ) +
    geom_point(
```

```
        aes(color = treat),
        size = 4,
        position= jittersettings
        ) +
    scale_color_manual(values = c(
      "Control group" = "#00608b", # blue
      "Treatment group" = "#ffa600" # yellow-orange
      #"Control group"= "#58508D", # purple-blue
      #"Treatment group" = "#FF6361" # red
      )) +
    geom_text_repel(
        aes(label = format(round(y, 2), nsmall = 2)),
        #nudge_x = 0.2,         # Nudge labels to the right of the points
        direction = "x",       # Only nudge horizontally
        hjust = -0.35,             # Align text to the left
        segment.size = 0,    # Line connecting point and text
        #segment.alpha = 0,
        segment.color = NA,
        point.padding = 0,  # Padding between point and text
        box.padding = 0,     # Padding around the text box
        position = jittersettings,
        size = 4,
        family = "Helvetica"
        ) +
    theme_minimal() +
    ylab("Outcome") +
    # reverse order on x-axis
    scale_x_discrete(limits = rev(levels(plot_df$treat))) +
    theme(
        legend.position = "none",
        text = element_text(family = "Helvetica"),
        axis.title.x = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.y = element_blank(),
        axis.title.y = element_text(family="Helvetica", size = 13),
        axis.text.x = element_text(family="Helvetica", size = 13, color = "black")
        )
    #ggtitle("A hypothetical randomized trial")

fig1
```




```
ggsave("figures/figure1.pdf", fig1, height = 4, width = 7)
```