

# Simulation from Coppock et al 2020 Data

*Musashi Harukawa*

*9 September 2020*

## Introduction

Coppock et al. (2020, hereafter CHV20) ran a 29-week survey experiment (N=34,000) where they tested a range of hypotheses concerning the effectiveness of 49 unique political advertisements. Their findings overall show that small average effects are not due to heterogeneous treatment effects.

The setup of their experiment resembles Phase I of my proposed experiment in many ideal ways. Multiple ads are tested per objective (see Table 1 below), ad assignment is randomised, and a number of relevant covariates are collected.

In this document I use the CHV20 dataset to show that the optimal advertisement varies with individual traits, and estimate the increase in average outcome under maximal allocation.

```
library("tidyverse")
library("pander")
library("estimatr")

setwd("~/Documents/Study/DPhil/Main/Code/coppock_et_al2020")

data <- read_rds("reprod/exps.rds")
```

## Dataset

The data includes 34,000 observations of 23 variables, detailed below. Of main note are the variables:

- `pid_7_pre` a 1-7 ordinal scale of partisanship
- `ideo_5_pre` a 1-5 ordinal scale of ideology
- `female_pre` a dummy variable for gender
- `ad_*` variables indicating which ad was shown
- `favor*_rev` outcome, the favorability rating of a candidate given by initials

```
# ~~ COLUMN LEGEND ~~ #
#
# wave - NUM: Waves 2-30, 1000 each
# date - DATE: Same info as wave
# election_phase - CHAR: 22000 General, 12000 Primary
# pid_3_pre - CHAR: Party ID: Republican, Independent, Democrat
# pid_7_pre - NUM: Party ID 1-7 scale, Dem 1-3 Ind 4 Rep 5-7
# ideo5_pre - NUM: Ideology 1-5 scale, likely L-R scale 1 Left
# female_pre - NUM: Dummy Female 1
# battleground NUM: Dummy Battleground State 1
# assignment FAC: Treatment assignment, 8 treatments 1 control
# ad_id: NUM: numeric id, 51 unique,
# ad_id_fac: FAC: same as above, dtype FACTOR
# ad_title: CHAR: ad title
# ad_valence: CHAR: corresponds to assignment, save Two Ads
# ad_sponsor: CHAR: sponsor of ad, 12 unique values
# manip: NUM: Unclear, TODO: refer to paper
```

```
# favorDT_rev: NUM: Donald Trump rating (5-point)
# favorHC_rev: NUM: Hillary Clinton rating
# favorBS_rev: NUM: Bernie Sanders rating
# favorJK_rev: NUM: John Kasich rating
# favorTC_rev: NUM: Ted Cruz rating
# general_vote_HC: NUM: Dummy will vote Hillary in general election
# general_vote_DT: NUM: Dummy will vote Donald in general election
# weights: NUM: weights for robust regression
```

51 different ads are contained in the dataset. The number of individuals treated with each ad vary from 1,344 for “Role Model” (Anti Trump), to 189 for “Broken” (Pro Trump). Unlike in my design, the outcome variable is not a pre-post difference.

I summarise the number of ads per CHV20 assignment group in the table below:

```
# ~~ OUTCOME ~~
#
# 51 different ads tested, ranging from 1344 to 189 per ad
# No pre-treatment level
# We can do this one model per outcome
# Most models include Donald Trump
# Test number of ads per assignment category

data %>%
  group_by(assignment) %>%
  summarise(length(unique(ad_id))) %>%
  pander()
```

| assignment   | length(unique(ad_id)) |
|--------------|-----------------------|
| Control      | 1                     |
| Pro Clinton  | 6                     |
| Anti Clinton | 11                    |
| Pro Trump    | 3                     |
| Anti Trump   | 24                    |
| Pro Cruz     | 2                     |
| Pro Kasich   | 1                     |
| Pro Sanders  | 2                     |
| Two Ads      | 1                     |

## Predictive Model

Following CHV20, I use the `lm_robust` function from the `estimatr` package to fit my predictive model for learning the best advertisement as a function of the covariates. Similarly, I use the same pre-treatment covariates to fit the model: partisanship, ideology and gender. Unlike the functions contained in the reproduction material for CHV20, I interact the treatments on each covariate in order to account for heterogeneity in the ad effects.

```
test_treatments <- function(treatment_filter, data, dv) {
  df <- data %>% filter(assignment %in% treatment_filter) %>% drop_na(., dv)
  # Fitting the model with favorHC_rev as outcome
  formula <- formula(
    paste0(
      dv,
```

```

    "~ ad_id_fac + ad_id_fac*pid_7_pre + ad_id_fac*ideo5_pre + ad_id_fac*female_pre"
  )
)
model1 <- lm_robust(formula, weights=weights, data = df)

# Construct prediction frame `test`
covs <- c('pid_7_pre', 'ideo5_pre', 'female_pre')
n_treats <- length(unique(df$ad_id_fac))
step <- dim(df)[1]
test <- df[,covs]
for (i in 1:(n_treats-1)) {
  test <- rbind(test, df[,covs])
}
test$ad_id_fac <- sort(rep(unique(df$ad_id_fac), step))
test$yhat <- predict(model1, test)

# Reform prediction frame to view predictions side-by-side
test.predict <- data.frame(matrix(nrow=step, ncol=n_treats))
colnames(test.predict) <- sapply(
  unique(df$ad_id_fac), function(x){paste0('treatment ', x)}
)

i <- 1
for (t in unique(df$ad_id_fac)) {
  test.predict[,paste0('treatment ', t)] <- test$yhat[i:(i-1+step)]
  i <- i + step
}

# Find best ad per-observation
rank_optima <- function(x){
  ranking <- ifelse(str_extract(treatment_filter, "[A-z]+")=="Anti", min, max)
  optim_ind <- which(x==ranking(x))
  out_str <- colnames(test.predict)[optim_ind]
  return(out_str)
}

best_ad <- apply(
  X = test.predict,
  FUN = rank_optima,
  MARGIN = 1
)

# Find average outcome under random allocation
rand_out <- mean(df[[dv]])
# Find average outcome under maximal allocation
ma_df <- df[,covs]
ma_df$ad_id_fac <- sapply(best_ad, function(x){str_extract(x, "[0-9]+")})
ma_df[[dv]] <- predict(model1, ma_df)
ma_out <- mean(ma_df[[dv]])
# Calculated predicted improvement
pred_improv <- ifelse(
  str_extract(treatment_filter, "[A-z]+")=="Anti",
  (rand_out - ma_out)/rand_out,

```

```

    (ma_out - rand_out)/rand_out
  )
  return(
    list(
      Objective = treatment_filter,
      Observations = step,
      Treatments = table(best_ad),
      Improvement = pred_improv,
      "Base Outcome" = rand_out,
      "Predicted Outcome" = ma_out,
      "Simulated Data" = ma_df
    )
  )
}

```

## Outcome

I test the predicted improvement for the four subsets of the data with the largest number of observations, Pro/Anti Trump/Clinton. The tables below show a number of key statistics for each of the subsets/predictive models. The main outcome of interest is **Improvement**, which is the proportional change in predicted average outcome under maximal allocation compared to the actual average outcome under random allocation. For “Pro” campaigns, the increases in average favorability are treated as positive, whereas for “Anti” campaigns decreases are treated as positive.

The results are promising for my experiment. Each of the subsets demonstrates predicted variation in optimal ad, as demonstrated by the table under **Treatments** in each cell. Moreover, the predicted improvement ranges from 0.066% for the Pro Trump ads to 15.38% for the Anti Trump ads.

```
test_treatments("Pro Trump", data, "favorDT_rev")[1:6] %>% pander()
```

- **Objective:** Pro Trump
- **Observations:** 751
- **Treatments:**

| treatment 24 | treatment 72 | treatment 74 |
|--------------|--------------|--------------|
| 360          | 236          | 155          |

- **Improvement:** 0.006591
- **Base Outcome:** 2.549
- **Predicted Outcome:** 2.565

```
test_treatments("Anti Trump", data, "favorDT_rev")[1:6] %>% pander()
```

- **Objective:** Anti Trump
- **Observations:** 10114
- **Treatments:**

Table 3: Table continues below

| treatment 27 | treatment 38 | treatment 4 | treatment 50 | treatment 6 |
|--------------|--------------|-------------|--------------|-------------|
| 2287         | 397          | 265         | 1069         | 2090        |

  

| treatment 60 | treatment 63 | treatment 77 | treatment 8 | treatment 99 |
|--------------|--------------|--------------|-------------|--------------|
| 971          | 1027         | 719          | 844         | 445          |

- **Improvement:** *0.1538*
- **Base Outcome:** *2.422*
- **Predicted Outcome:** *2.05*

```
test_treatments("Pro Clinton", data, "favorHC_rev")[1:6] %>% pander()
```

- **Objective:** Pro Clinton
- **Observations:** *2367*
- **Treatments:**

| treatment 12 | treatment 18 | treatment 42 | treatment 9 |
|--------------|--------------|--------------|-------------|
| 1408         | 210          | 332          | 417         |

- **Improvement:** *0.1068*
- **Base Outcome:** *2.678*
- **Predicted Outcome:** *2.964*

```
test_treatments("Anti Clinton", data, "favorHC_rev")[1:6] %>% pander()
```

- **Objective:** Anti Clinton
- **Observations:** *4205*
- **Treatments:**

| treatment 45 | treatment 48 | treatment 58 | treatment 61 | treatment 64 |
|--------------|--------------|--------------|--------------|--------------|
| 345          | 1033         | 1670         | 532          | 625          |

- **Improvement:** *0.108*
- **Base Outcome:** *2.522*
- **Predicted Outcome:** *2.249*

## Next Steps

I think that this simulation could easily go into my transfer of status materials. Therefore the next steps include:

- Trying alternative predictive models to see if results are robust.
- Use data to make power estimates based on realistic effect sizes.