Outcome Analysis of the Sold-Vs-Not Design

Jake Bowers

December 08, 2021

```
load(here::here("Data", "finaldat.rda"), verbose = TRUE)
```

```
Loading objects:
  finaldat
```

```
load(here::here("Data", "wrkdat.rda"), verbose = TRUE)
```

```
Loading objects:
  wrkdat
  allvars2
  allvars
  allcovs
  designvars
  outcomes
  covs
  covsCensus
```

```
load(here::here("Analysis", "match_data_prep.rda"), verbose = TRUE)
```

```
Loading objects:
  wdat17p
  wdat17i
  wdat18p
  wdat18i
  covs3
  allcovs
  outcomes
  designvars
  covsCensus
  covs
```

```
load(here::here("Analysis", "design_soldvsnot.rda"), verbose = TRUE)
```

```
Loading objects:
  parms1_res
  parms1
  dat17i
  dat17p
```

Analyses Registered

We registered the following analyses and research designs in https://osf.io/sxe8q and will submit differences from the planned analyses there, too.

Data Setup for the Analyses

Not all outcome variables are available in endline survey. See the `pap_soldvsnot.Rmd` for more details.

Hypothesis	Outcome	Variable(s)
H1	Neighbors' crime victimization in the last 12 months	vic12
H2	Neighbors' insecurity perceptions. Country and neighborhood	c_sec_i and n_sec_i
H3	Existence of "bocas" in the neighborhood	boca1_i
H4	Social Disorder Index	social_dis
H5	Neighbors' insertion in neighborhood	activities_index
H6	Law's perceived of impact on public security	ps_impact_i
H7	Law's perceived of impact on drug trafficking	dt_impact_i

```
regoutcomes <- c("n_sec_i", "c_sec_i", "vic12_i", "dt_impact_i", "ps_impact_i", "boca1_i", "social_dis", "activities_index")
regcovs <- c("ideol_si_i", "educ_i", "sex_i", "age_i", "robb_2016", "vrobb_2016") ## ,"n_sec_i" this will be used later  at phar
regdesignvars <- c("id", "Q56", "treat", "ronda", "ph_type")
```

We want the versions of the variable without imputation for missing values

```
## Checking the relationship between ph_type and treat
with(dat17p, table(soldvsnot17, ph_type, exclude = c()))
```

```
           ph_type
soldvsnot17  1  2  3
          0 42  0  0
          1  0 10  6
```

```
## finaldat includes both 2017 and 2018 and is at the individual level
with(finaldat, table(treat, ph_type, exclude = c()))
```

```
     ph_type
treat   1   2   3   4   5
    0 875  10   0   0  40
    1  10 197 126  40   0
```

```
## This is the outcome data
fdat18i <- finaldat %>%
  dplyr::select(one_of(c(regdesignvars, regoutcomes, regcovs))) %>%
  filter(ronda == 2018) %>%
  mutate_if(is.character, as.numeric)

## Since we do not have the same people in 2017 and 2018, we do covariance
## adjustment either by: ## (1) the rebar method (following Sales and Hansen)
## and/or lin approach using indivi level data and (2) more simply just using the
## phramacy level data. The idea is that a difference score might have more
## statistical power, but we don't observe the same people twice.

replace_NA_0 <- function(x) {
  ifelse(x %in% c(88, 99), NA, x)
}
## Make all 88 and 99 responses into NA
outdat3 <- fdat18i %>%
  dplyr::select(-c("id", "age_i")) %>%
  mutate_all(replace_NA_0)
```

```
stopifnot(sum(is.na(outdat3$boca1_i)) == 288) ## make sure to preserve missings
outdat3$id <- fdat18i$id
outdat3$age_i <- fdat18i$age_i
## How much missing data is there?
outdat3 %>% summarise_all(~ sum(is.na(.)))
```

```
  Q56 treat ronda ph_type n_sec_i c_sec_i vic12_i dt_impact_i ps_impact_i boca1_i social_dis activities_index ideol_si_i educ_i
1   0     0     0       0       1       2       0          44          42     288          0                4         41      1
  sex_i robb_2016 vrobb_2016 id age_i
1     0       698        698  0     0
```

```
##  vrobb_2016 and robb_2016 are all missing because they are not recorded for the  2018 subjects. They are pharmacy  level.
##  Leaving them here as placeholders. Otherwise very little missing data.
```

Now merge the pharmacy level design info onto the individual level data

```
stopifnot(all.equal(names(parms1_res$fm_p), row.names(dat17p)))
dat17p$parms1_fm_p <- parms1_res$fm_p

designdat_p <- dat17p %>% dplyr::select(c(
  "Q56", "parms1_fm_p", "soldvsnot17", "ph_type",
  "n_sec_i_mean", "robb_2016_mean", "vrobb_2016_mean"
))

## Two pharmacies (the placebos) were dropped
stopifnot(length(unique(designdat_p$Q56)) == 58)
stopifnot(unique(designdat_p$ph_type) != 5)

outdat4 <- inner_join(outdat3, designdat_p)
stopifnot(isTRUE(all.equal(sort(unique(designdat_p$Q56)), sort(unique(outdat4$Q56)))))
outdat4 %>% summarise_all(~ sum(is.na(.)))
```

```
  Q56 treat ronda ph_type n_sec_i c_sec_i vic12_i dt_impact_i ps_impact_i boca1_i social_dis activities_index ideol_si_i educ_i
1   0     0     0       0       0       1       0          42          37     256          0                3         38      1
  sex_i robb_2016 vrobb_2016 id age_i parms1_fm_p soldvsnot17 n_sec_i_mean robb_2016_mean vrobb_2016_mean
1     0       638        638  0     0         269           0            0              0               0
```

```
dim(outdat4)
```

```
[1] 638  24
```

```
## Dropped 6 non-selling pharmacies. (60 indivs)
dim(outdat3)
```

```
[1] 698  19
```

```
## Remove non-selling pharmacies dropped during the design-search process
##  Remove variables that are all  missing
outdat5 <- outdat4 %>%
  filter(!is.na(parms1_fm_p)) %>%
  dplyr::select(-c("robb_2016", "vrobb_2016")) %>%
  ungroup()
dim(outdat5)
```

```
[1] 369  22
```

A quick check to ensure that we don't have too much missing data.

```
## Do we need to worry about means (maybe)
regcovs1 <- c(grep("robb", regcovs, value = TRUE, invert = TRUE), "robb_2016_mean", "vrobb_2016_mean", "n_sec_i_mean")
outdat5 %>%
  dplyr::select(all_of(regoutcomes)) %>%
  summarize_all(~ length(unique(.)))
```

```
  n_sec_i c_sec_i vic12_i dt_impact_i ps_impact_i boca1_i social_dis activities_index
1       4       4       2           4           4       3          7               19
```

```
outdat5 %>%
  dplyr::select(all_of(regcovs1)) %>%
  summarize_all(~ length(unique(.)))
```

```
  ideol_si_i educ_i sex_i age_i robb_2016_mean vrobb_2016_mean n_sec_i_mean
1         11      9     2    70             28               7           14
```

Checking heterogeneity within set across individuals

No strong relationships between the types of people living in a place and whether that place had a pharmacy selling marijuana conditional on set. Here, checking this relationship allowing for non-linear relationships between education and age and treatment status.

```r
library(splines)
xb_i <- balanceTest(soldvsnot17 ~ ns(educ_i, 3) + sex_i + ns(age_i, 3) + robb_2016_mean + vrobb_2016_mean + ideol_si_i + strata(
xb_i$results[, , "parms1_fm_p"]
```

```
                 stat
vars               Control Treatment std.diff   adj.diff pooled.sd        z p
  ns(educ_i, 3)1   0.17101   0.15418 -0.08249 -0.016829   0.20401  1.41824 1
  ns(educ_i, 3)2   0.40382   0.40982  0.07756  0.006000   0.07736  0.38816 1
  ns(educ_i, 3)3  -0.03608  -0.04074 -0.01196 -0.004659   0.38971 -1.15533 1
  sex_i            1.61605   1.57803 -0.07747 -0.038011   0.49068  0.05256 1
  ns(age_i, 3)1    0.21786   0.19043 -0.10946 -0.027428   0.25057 -0.72504 1
  ns(age_i, 3)2    0.39216   0.37395 -0.14126 -0.018213   0.12894  0.05060 1
  ns(age_i, 3)3   -0.10863  -0.09738  0.04512  0.011249   0.24931  1.32492 1
  robb_2016_mean  39.79943  39.31214 -0.01697 -0.487288  28.71685 -0.85224 1
  vrobb_2016_mean  1.59312   1.53179 -0.02347 -0.061331   2.61349 -1.13072 1
  ideol_si_i       5.46809   5.25949 -0.07870 -0.208591   2.65040 -0.28228 1
  (ideol_si_i)     0.94269   0.91329 -0.11607 -0.029399   0.25329  0.98983 1
```

```r
xb_i$overall["parms1_fm_p", ]
```

```
chisquare        df   p.value
  10.0000   10.0000    0.4405
```

```r
lm_i <- lm_robust(soldvsnot17 ~ ns(educ_i, 3) + sex_i + ns(age_i, 3) + robb_2016_mean + vrobb_2016_mean + ideol_si_i, fixed_effe
thef <- lm_i$proj_fstatistic
p_thef <- pf(thef["value"], df1 = thef["numdf"], df2 = thef["dendf"], lower.tail = FALSE)
p_thef
```

```
 value
0.8384
```

Does treatment assignment relate to missingness?

No evidence against the idea that missingness on outcomes is random within set.

```r
outdat5 %>%
  dplyr::select(all_of(regoutcomes)) %>%
  summarize_all(~ sum(is.na(.)))
```

```
  n_sec_i c_sec_i vic12_i dt_impact_i ps_impact_i boca1_i social_dis activities_index
1       0       0       0          23          22     147          0                0
```

```r
missing_outcome_vars0 <- sapply(outdat5[, regoutcomes], function(x) {
  any(is.na(x))
})
missing_outcome_vars <- names(missing_outcome_vars0[missing_outcome_vars0])

outdat5 <- outdat5 %>%
  mutate(across(one_of(missing_outcome_vars), ~ as.numeric(is.na(.)), .names = "missing_{col}"))

missing_test_i <- balanceTest(soldvsnot17 ~ missing_dt_impact_i + missing_ps_impact_i + missing_boca1_i + strata(parms1_fm_p) +
missing_test_i$results[, , "parms1_fm_p"]
```

```
                     stat
vars                  Control Treatment std.diff adj.diff pooled.sd      z      p
  missing_dt_impact_i 0.05158   0.06358  0.04951  0.01201    0.2425 0.3381 0.7353
  missing_ps_impact_i 0.05444   0.06936  0.06262  0.01492    0.2383 0.4041 0.6862
  missing_boca1_i     0.37249   0.41618  0.08902  0.04369    0.4908 0.1606 0.8724
```

```r
missing_test_i$overall["parms1_fm_p", ]
```

```
chisquare        df   p.value 
   1.8472    4.0000    0.7638 
```

Estimating average effects and testing the weak null of no effects

Tests of of the weak null combined with estimates of the ATE

First, we do the simple thing — estimate the ATE and test the weak null of no effects under asymptotic assumptions.

```r
## Get cluster size and condition on it just to avoid bias as described by
## Aronow and Middleton using the general approach from Lin --- mean centering
## the covariate within block.
outdat5 <- outdat5 %>%
  group_by(Q56) %>%
  mutate(n_clus = n()) %>%
  ungroup()

summary(outdat5[, regoutcomes])
```

```
    n_sec_i         c_sec_i        vic12_i       dt_impact_i     ps_impact_i       boca1_i        social_dis    activities_index
 Min.   :1.00   Min.   :1.0   Min.   :0.000   Min.   :1.00   Min.   :1.00   Min.   :1.00   Min.   :1.00   Min.   :0.000   
 1st Qu.:2.00   1st Qu.:2.0   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:1.00   1st Qu.:1.00   1st Qu.:4.00   1st Qu.:0.500   
 Median :3.00   Median :2.0   Median :1.000   Median :2.00   Median :2.00   Median :1.00   Median :4.00   Median :0.665   
 Mean   :2.95   Mean   :2.3   Mean   :0.843   Mean   :1.99   Mean   :1.73   Mean   :1.45   Mean   :3.88   Mean   :0.653   
 3rd Qu.:4.00   3rd Qu.:3.0   3rd Qu.:1.000   3rd Qu.:3.00   3rd Qu.:2.00   3rd Qu.:2.00   3rd Qu.:4.00   3rd Qu.:0.750   
 Max.   :4.00   Max.   :4.0   Max.   :1.000   Max.   :3.00   Max.   :3.00   Max.   :2.00   Max.   :4.00   Max.   :1.000   
                                              NA's   :23    NA's   :22    NA's   :147                                  
```

```r
unadj_fn <- function(ynm) {
  thedat <- outdat5 %>%
    filter(!is.na(!!rlang::sym(ynm))) %>%
    group_by(Q56) %>%
    mutate(n_clus = n()) %>%
    ungroup()
  thedat <- thedat %>%
    group_by(parms1_fm_p) %>%
    mutate(all_same_n = length(unique(n_clus)), n_clus_c = n_clus - mean(n_clus), valid_block = length(unique(soldvsnot17)) > 1)
    filter(valid_block) %>%
    droplevels() %>%
    ungroup()
  if (all(thedat$all_same_n != 1)) {
    fmla <- paste(ynm, "~soldvsnot17*n_clus_c", sep = "")
  } else {
    fmla <- paste(ynm, "~soldvsnot17", sep = "")
  }
  mod <- lm_robust(as.formula(fmla), data = thedat, fixed_effects = ~parms1_fm_p, clusters = Q56)
  res <- tidy(mod) %>% filter(term == "soldvsnot17")
  res$n_clus_used <- length(unique(thedat$n_clus))
  return(res)
}

## unadj_fn("boca1_i")
## unadj_fn(regoutcomes[1])

unadj_coefs_lst <- lapply(regoutcomes, unadj_fn)
unadj_coefs <- bind_rows(unadj_coefs_lst)
unadj_coefs
```

```
         term estimate std.error statistic p.value conf.low conf.high     df         outcome n_clus_used
1 soldvsnot17 0.178500   0.12979   1.37529  0.1880 -0.09670   0.45370 15.958         n_sec_i           4
2 soldvsnot17 0.018410   0.08351   0.22045  0.8283 -0.15866   0.19548 15.958         c_sec_i           4
3 soldvsnot17 0.050921   0.03529   1.44302  0.1684 -0.02390   0.12574 15.958         vic12_i           4
4 soldvsnot17 0.001829   0.07031   0.02602  0.9796 -0.14730   0.15096 15.901     dt_impact_i           6
5 soldvsnot17 0.036142   0.08895   0.40633  0.6899 -0.15255   0.22484 15.859     ps_impact_i           6
```

```
6 soldvsnot17 0.033097   0.11226   0.29482  0.7742 -0.21720   0.28339  9.952          boca1_i          11
7 soldvsnot17 0.064138   0.06080   1.05496  0.3072 -0.06477   0.19305 15.958       social_dis           4
8 soldvsnot17 0.024971   0.03017   0.82779  0.4200 -0.03899   0.08893 15.958 activities_index           4
```

```
kableExtra::kable(unadj_coefs)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome	n_clus_used
soldvsnot17	0.1785	0.1298	1.3753	0.1880	-0.0967	0.4537	15.958	n_sec_i	4
soldvsnot17	0.0184	0.0835	0.2204	0.8283	-0.1587	0.1955	15.958	c_sec_i	4
soldvsnot17	0.0509	0.0353	1.4430	0.1684	-0.0239	0.1257	15.958	vic12_i	4
soldvsnot17	0.0018	0.0703	0.0260	0.9796	-0.1473	0.1510	15.901	dt_impact_i	6
soldvsnot17	0.0361	0.0889	0.4063	0.6899	-0.1526	0.2248	15.859	ps_impact_i	6
soldvsnot17	0.0331	0.1123	0.2948	0.7742	-0.2172	0.2834	9.952	boca1_i	11
soldvsnot17	0.0641	0.0608	1.0550	0.3072	-0.0648	0.1930	15.958	social_dis	4
soldvsnot17	0.0250	0.0302	0.8278	0.4200	-0.0390	0.0889	15.958	activities_index	4

```
## The simple version with the full sample and no statistical adjustment: Mostly
## to learn about power loss and bias reduction from our matched design:

## A function that estimates the effects without the matched design.
simp_fn <- function(ynm) {
  thedat <- outdat4 %>%
    filter(!is.na(!!rlang::sym(ynm))) %>%
    group_by(Q56) %>%
    mutate(n_clus = n()) %>%
    ungroup()
  thedat <- thedat %>%
    mutate(n_clus_c = n_clus - mean(n_clus)) %>%
    droplevels() %>%
    ungroup()
  fmla <- paste(ynm, "~soldvsnot17*n_clus_c", sep = "")
  mod <- lm_robust(as.formula(fmla), data = thedat, clusters = Q56)
  res <- tidy(mod) %>% filter(term == "soldvsnot17")
  res$n_clus_used <- length(unique(thedat$n_clus))
  return(res)
}

simp_coefs_lst <- lapply(regoutcomes, simp_fn)
simp_coefs <- bind_rows(simp_coefs_lst)
simp_coefs
```

```
         term estimate std.error statistic  p.value  conf.low conf.high    df          outcome n_clus_used
1 soldvsnot17  0.29240   0.10846    2.6959 0.013706  0.066521    0.5183 20.51          n_sec_i           4
2 soldvsnot17  0.12053   0.08910    1.3527 0.190535 -0.064768    0.3058 21.00          c_sec_i           4
3 soldvsnot17  0.05983   0.02637    2.2693 0.034172  0.004922    0.1147 20.51          vic12_i           4
4 soldvsnot17  0.03891   0.07514    0.5178 0.610023 -0.117428    0.1952 20.83      dt_impact_i           6
5 soldvsnot17  0.01176   0.06250    0.1881 0.852801 -0.119063    0.1426 19.01      ps_impact_i           6
6 soldvsnot17  0.06619   0.09033    0.7327 0.472400 -0.122494    0.2549 19.57          boca1_i          13
7 soldvsnot17  0.14479   0.04963    2.9174 0.008367  0.041430    0.2482 20.51       social_dis           4
8 soldvsnot17  0.01609   0.02730    0.5893 0.561645 -0.040519    0.0727 22.12 activities_index           4
```

Looks like we had more bias that mattered for some outcomes more so that others: social_dis, and vic12_i stand out as showin much larger post-adjustment differences.

```
kableExtra::kable(simp_coefs)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome	n_clus_used
soldvsnot17	0.2924	0.1085	2.6959	0.0137	0.0665	0.5183	20.51	n_sec_i	4
soldvsnot17	0.1205	0.0891	1.3527	0.1905	-0.0648	0.3058	21.00	c_sec_i	4
soldvsnot17	0.0598	0.0264	2.2693	0.0342	0.0049	0.1147	20.51	vic12_i	4
soldvsnot17	0.0389	0.0751	0.5178	0.6100	-0.1174	0.1952	20.83	dt_impact_i	6
soldvsnot17	0.0118	0.0625	0.1881	0.8528	-0.1191	0.1426	19.01	ps_impact_i	6
soldvsnot17	0.0662	0.0903	0.7327	0.4724	-0.1225	0.2549	19.57	boca1_i	13
soldvsnot17	0.1448	0.0496	2.9174	0.0084	0.0414	0.2482	20.51	social_dis	4
soldvsnot17	0.0161	0.0273	0.5893	0.5616	-0.0405	0.0727	22.12	activities_index	4

```
left_join(select(unadj_coefs, estimate, std.error, outcome, p.value), select(simp_coefs, estimate, std.error, outcome, p.value),
```

```
  estimate.x std.error.x          outcome p.value.x estimate.y std.error.y p.value.y
```

```
1   0.178500     0.12979          n_sec_i    0.1880    0.29240     0.10846  0.013706
2   0.018410     0.08351          c_sec_i    0.8283    0.12053     0.08910  0.190535
3   0.050921     0.03529          vic12_i    0.1684    0.05983     0.02637  0.034172
4   0.001829     0.07031      dt_impact_i    0.9796    0.03891     0.07514  0.610023
5   0.036142     0.08895      ps_impact_i    0.6899    0.01176     0.06250  0.852801
6   0.033097     0.11226          boca1_i    0.7742    0.06619     0.09033  0.472400
7   0.064138     0.06080       social_dis    0.3072    0.14479     0.04963  0.008367
8   0.024971     0.03017 activities_index    0.4200    0.01609     0.02730  0.561645
```

```
## Also try a multilevel model for those more familiar with that approach.
library(lme4)
library(lmerTest)

## A function that estimates the effects without the matched design.
lmer_fn <- function(ynm) {
  thedat <- outdat5 %>% filter(!is.na(!!rlang::sym(ynm)))
  fmla <- paste(ynm, "~(1|parms1_fm_p:Q56) + (1|Q56)+soldvsnot17", sep = "")
  mod <-
    lmer(as.formula(fmla), data = thedat) # ,control=lmerControl(optimizer="bobyqa"))
  res0 <- summary(mod)$coefficients
  res0_ci <- confint(mod)
  res1 <- data.frame(
    term = row.names(res0), estimate = res0[, "Estimate"], std.error = res0[, "Std. Error"],
    "statistic" = res0[, "t value"], p.value = res0[, 5], conf.low = res0_ci["soldvsnot17", 1], conf.high = res0_ci["soldvsnot17
  )
  res <- res1 %>% filter(term == "soldvsnot17")
  return(res)
}

lmer_coefs_lst <- lapply(regoutcomes, lmer_fn)
lmer_coefs <- bind_rows(lmer_coefs_lst)
lmer_coefs
```

```
                       term estimate std.error statistic p.value conf.low conf.high          outcome
soldvsnot17...1 soldvsnot17 0.172927   0.14922    1.1589  0.2553 -0.11923   0.46461          n_sec_i
soldvsnot17...2 soldvsnot17 0.037574   0.09175    0.4095  0.6848 -0.14178   0.21725          c_sec_i
soldvsnot17...3 soldvsnot17 0.056506   0.03796    1.4886  0.1374 -0.01788   0.13089          vic12_i
soldvsnot17...4 soldvsnot17 0.026025   0.08393    0.3101  0.7584 -0.13878   0.18982      dt_impact_i
soldvsnot17...5 soldvsnot17 0.050335   0.08177    0.6156  0.5425 -0.10927   0.21075      ps_impact_i
soldvsnot17...6 soldvsnot17 0.008979   0.09512    0.0944  0.9254 -0.17706   0.19514          boca1_i
soldvsnot17...7 soldvsnot17 0.053975   0.06262    0.8619  0.3953 -0.06856   0.17645       social_dis
soldvsnot17...8 soldvsnot17 0.025928   0.02832    0.9155  0.3669 -0.02952   0.08129 activities_index
```

Compare the two different approaches

```
## No real substantive differences
unadj_mods <- left_join(select(unadj_coefs, estimate, outcome, p.value, conf.low, conf.high),
  select(lmer_coefs, estimate, outcome, p.value, conf.low, conf.high),
  by = "outcome", suffix = c(".lm", ".lmer")
)

kableExtra::kable(unadj_mods %>% select(
  outcome, estimate.lm, estimate.lmer, p.value.lm, p.value.lmer, conf.low.lm, conf.high.lm,
  conf.low.lmer, conf.high.lmer
))
```

outcome	estimate.lm	estimate.lmer	p.value.lm	p.value.lmer	conf.low.lm	conf.high.lm	conf.low.lmer	conf.high.lmer
n_sec_i	0.1785	0.1729	0.1880	0.2553	-0.0967	0.4537	-0.1192	0.4646
c_sec_i	0.0184	0.0376	0.8283	0.6848	-0.1587	0.1955	-0.1418	0.2172
vic12_i	0.0509	0.0565	0.1684	0.1374	-0.0239	0.1257	-0.0179	0.1309
dt_impact_i	0.0018	0.0260	0.9796	0.7584	-0.1473	0.1510	-0.1388	0.1898
ps_impact_i	0.0361	0.0503	0.6899	0.5425	-0.1526	0.2248	-0.1093	0.2108
boca1_i	0.0331	0.0090	0.7742	0.9254	-0.2172	0.2834	-0.1771	0.1951
social_dis	0.0641	0.0540	0.3072	0.3953	-0.0648	0.1930	-0.0686	0.1764
activities_index	0.0250	0.0259	0.4200	0.3669	-0.0390	0.0889	-0.0295	0.0813

As a check on the preceding, do this at the level of the pharmacy:

```
outdat5p <- outdat5 %>%
  group_by(Q56) %>%
  summarize(across(one_of(c("soldvsnot17", regoutcomes)), mean, na.rm = TRUE), n_clus = n())
outdat6p <- left_join(outdat5p, dat17p[, c("Q56", "parms1_fm_p")], by = "Q56")
stopifnot(nrow(outdat5p) == nrow(outdat6p))

unadj_fn_p <- function(ynm) {
  thedat <- outdat6p %>% filter(!is.na(!!rlang::sym(ynm)))
  thedat <- thedat %>%
    group_by(parms1_fm_p) %>%
    mutate(
      n_uniq_sets =
        n(), n_clus_c = n_clus - mean(n_clus)
    ) %>%
    filter(n_uniq_sets > 1) %>%
    droplevels()
  fmla <- paste(ynm, "~soldvsnot17*n_clus_c", sep = "")
  mod <- lm_robust(as.formula(fmla), data = thedat, fixed_effects = ~parms1_fm_p)
  res <- tidy(mod) %>% filter(term == "soldvsnot17")
  res$n_clus_used <- length(unique(thedat$n_clus))
  return(res)
}

unadj_fn_p("boca1_i")
```

```
         term estimate std.error statistic p.value conf.low conf.high df outcome n_clus_used
1 soldvsnot17 -0.02413    0.1092    -0.221  0.8283  -0.2583    0.2101 14 boca1_i           4
```

```
unadj_fn_p(regoutcomes[1])
```

```
         term estimate std.error statistic p.value conf.low conf.high df outcome n_clus_used
1 soldvsnot17   0.1674    0.1281     1.307  0.2109  -0.1056    0.4404 15 n_sec_i           4
```

```
unadj_coefs_p_lst <- lapply(regoutcomes, unadj_fn_p)
unadj_coefs_p <- bind_rows(unadj_coefs_p_lst)
unadj_coefs_p
```

```
         term  estimate std.error statistic p.value conf.low conf.high df          outcome n_clus_used
1 soldvsnot17  0.167415   0.12809   1.30704  0.2109 -0.10560   0.44043 15          n_sec_i           4
2 soldvsnot17  0.024950   0.08611   0.28974  0.7760 -0.15860   0.20850 15          c_sec_i           4
3 soldvsnot17  0.045830   0.03658   1.25283  0.2294 -0.03214   0.12380 15          vic12_i           4
4 soldvsnot17  0.004138   0.07320   0.05653  0.9557 -0.15188   0.16015 15      dt_impact_i           4
5 soldvsnot17  0.040372   0.08782   0.45971  0.6523 -0.14681   0.22756 15      ps_impact_i           4
6 soldvsnot17 -0.024131   0.10919  -0.22100  0.8283 -0.25832   0.21006 14          boca1_i           4
7 soldvsnot17  0.062552   0.06314   0.99068  0.3376 -0.07203   0.19713 15       social_dis           4
8 soldvsnot17  0.021798   0.03093   0.70471  0.4918 -0.04413   0.08773 15 activities_index           4
```

```
kableExtra::kable(unadj_coefs_p)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome	n_clus_used
soldvsnot17	0.1674	0.1281	1.3070	0.2109	-0.1056	0.4404	15	n_sec_i	4
soldvsnot17	0.0250	0.0861	0.2897	0.7760	-0.1586	0.2085	15	c_sec_i	4
soldvsnot17	0.0458	0.0366	1.2528	0.2294	-0.0321	0.1238	15	vic12_i	4
soldvsnot17	0.0041	0.0732	0.0565	0.9557	-0.1519	0.1602	15	dt_impact_i	4
soldvsnot17	0.0404	0.0878	0.4597	0.6523	-0.1468	0.2276	15	ps_impact_i	4
soldvsnot17	-0.0241	0.1092	-0.2210	0.8283	-0.2583	0.2101	14	boca1_i	4
soldvsnot17	0.0626	0.0631	0.9907	0.3376	-0.0720	0.1971	15	social_dis	4
soldvsnot17	0.0218	0.0309	0.7047	0.4918	-0.0441	0.0877	15	activities_index	4

Now checking with covariance adjustment:

```
adj_fn <- function(ynm) {
  thedat <- outdat5 %>%
    filter(!is.na(!!rlang::sym(ynm))) %>%
    group_by(Q56) %>%
    mutate(n_clus = n()) %>%
    ungroup()
  thedat <- thedat %>%
    group_by(parms1_fm_p) %>%
```

```r
    mutate(
      all_same_n = length(unique(n_clus)), n_clus_c = n_clus - mean(n_clus),
      ideol_c = ideol_si_i - mean(ideol_si_i, na.rm = TRUE),
      educ_c = educ_i - mean(educ_i, na.rm = TRUE),
      sex_c = sex_i - mean(sex_i, na.rm = TRUE),
      age_c = age_i - mean(age_i, na.rm = TRUE),
      valid_block = length(unique(soldvsnot17)) > 1
    ) %>%
    filter(valid_block) %>%
    droplevels() %>%
    ungroup()
  thedat$ideol_c_NA <- as.numeric(is.na(thedat$ideol_c))
  thedat$ideol_c_imp <- ifelse(is.na(thedat$ideol_c), 0, thedat$ideol_c)
  fmla <- paste(ynm, "~soldvsnot17*(n_clus_c+ideol_c_imp+ideol_c_NA+educ_c+sex_c+age_c)", sep = "")
  mod <- lm_robust(as.formula(fmla), data = thedat, fixed_effects = ~parms1_fm_p, clusters = Q56)
  res <- tidy(mod) %>% filter(term == "soldvsnot17")
  res$n_clus_used <- length(unique(thedat$n_clus))
  return(res)
}

## adj_fn("boca1_i")
## adj_fn(regoutcomes[1])

adj_coefs_lst <- lapply(regoutcomes, adj_fn)
adj_coefs <- bind_rows(adj_coefs_lst)
adj_coefs
```

```
         term  estimate std.error statistic p.value conf.low conf.high    df          outcome n_clus_used
1 soldvsnot17  0.188886   0.13642    1.3846  0.1871 -0.10279   0.48056 14.49          n_sec_i           4
2 soldvsnot17 -0.023667   0.09385   -0.2522  0.8044 -0.22432   0.17698 14.49          c_sec_i           4
3 soldvsnot17  0.031966   0.04144    0.7714  0.4529 -0.05664   0.12057 14.49          vic12_i           4
4 soldvsnot17 -0.009579   0.07456   -0.1285  0.8997 -0.17069   0.15153 12.98      dt_impact_i           6
5 soldvsnot17 -0.020463   0.08946   -0.2287  0.8226 -0.21354   0.17261 13.13      ps_impact_i           6
6 soldvsnot17  0.045757   0.11564    0.3957  0.7002 -0.21016   0.30168 10.53          boca1_i          11
7 soldvsnot17  0.052430   0.06309    0.8311  0.4194 -0.08245   0.18731 14.49       social_dis           4
8 soldvsnot17  0.027230   0.03288    0.8283  0.4209 -0.04306   0.09752 14.49 activities_index           4
```

```r
kableExtra::kable(adj_coefs)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high	df	outcome	n_clus_used
soldvsnot17	0.1889	0.1364	1.3846	0.1871	-0.1028	0.4806	14.49	n_sec_i	4
soldvsnot17	-0.0237	0.0938	-0.2522	0.8044	-0.2243	0.1770	14.49	c_sec_i	4
soldvsnot17	0.0320	0.0414	0.7714	0.4529	-0.0566	0.1206	14.49	vic12_i	4
soldvsnot17	-0.0096	0.0746	-0.1285	0.8997	-0.1707	0.1515	12.98	dt_impact_i	6
soldvsnot17	-0.0205	0.0895	-0.2287	0.8226	-0.2135	0.1726	13.13	ps_impact_i	6
soldvsnot17	0.0458	0.1156	0.3957	0.7002	-0.2102	0.3017	10.53	boca1_i	11
soldvsnot17	0.0524	0.0631	0.8311	0.4194	-0.0825	0.1873	14.49	social_dis	4
soldvsnot17	0.0272	0.0329	0.8283	0.4209	-0.0431	0.0975	14.49	activities_index	4

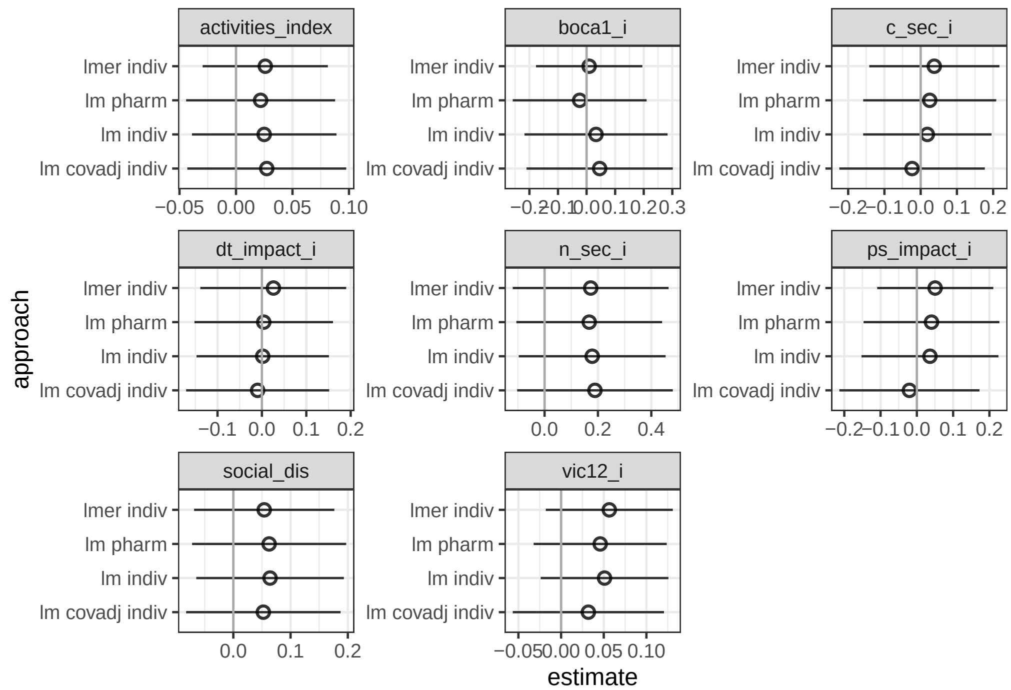
```r
unadj_coefs$approach <- "lm indiv"
lmer_coefs$approach <- "lmer indiv"
unadj_coefs_p$approach <- "lm pharm"
adj_coefs$approach <- "lm covadj indiv"

all_unadj <- bind_rows(unadj_coefs, lmer_coefs, unadj_coefs_p, adj_coefs)
all_unadj$outcome_n <- as.numeric(as.factor(all_unadj$outcome))
all_unadj$approach_n <- as.numeric(as.factor(all_unadj$approach)) - 1
all_unadj$outcome_n_yvals <- all_unadj$outcome_n + all_unadj$approach_n / 5

g <- ggplot(all_unadj, aes(x = estimate, y = approach)) +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high), shape = 21, alpha = .8) +
  geom_vline(xintercept = 0, color = "dark gray") +
  facet_wrap(~outcome, scales = "free") +
  theme_bw()
```

All approaches agree.

```
print(g)
```



Test for all outcomes

Checking to see if a more powerful test would reveal any effects. Simplify the hypothesis to a hypothesis of no effects for *any* outcome (other than `boca1_i`).

```
outdat6p$soldvsnot17F <- factor(outdat6p$soldvsnot17)

with(outdat6p %>% filter(!is.na(boca1_i)), table(soldvsnot17, parms1_fm_p, exclude = c()))
```

```
           parms1_fm_p
soldvsnot17 1.1 1.10 1.11 1.12 1.13 1.14 1.15 1.16 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
          0   1    1    1    1    1    1    1    1   1   1   1   1   2   1   1   2
          1   1    1    1    1    1    1    1    1   1   0   1   1   1   1   1   1
```

```
multi_outcome_fmla <- as.formula(paste(paste(regoutcomes[!(regoutcomes %in% c("boca1_i"))], collapse = "+"), "~soldvsnot17F|parm

multi_outcome_test_p <- independence_test(multi_outcome_fmla, data = outdat6p)
multi_outcome_test_perm_p <- independence_test(multi_outcome_fmla, data = outdat6p, distribution = approximate())

multi_outcome_test_p
```

```
    Asymptotic General Independence Test

data:  n_sec_i, c_sec_i, vic12_i, dt_impact_i, ps_impact_i, social_dis, activities_index by soldvsnot17F (0, 1)
     stratified by parms1_fm_p
maxT = 1.4, p-value = 0.7
alternative hypothesis: two.sided
```

```
multi_outcome_test_perm_p
```

```
    Approximative General Independence Test

data:  n_sec_i, c_sec_i, vic12_i, dt_impact_i, ps_impact_i, social_dis, activities_index by soldvsnot17F (0, 1)
     stratified by parms1_fm_p
maxT = 1.4, p-value = 0.7
alternative hypothesis: two.sided
```

```r
## Checking for whether we lose power because of strange distributions. Using a
## rank-based test
outdat6p <- outdat6p %>% mutate(across(one_of(regoutcomes), rank, .names = "rank_{col}"))

multi_outcome_test_rank_p <- independence_test(multi_outcome_fmla,
  data = outdat6p,
  ytrafo = function(data) {
    trafo(data, numeric_trafo = rank_trafo)
  }
)
multi_outcome_test_rank_perm_p <- independence_test(multi_outcome_fmla,
  data = outdat6p,
  ytrafo = function(data) {
    trafo(data, numeric_trafo = rank_trafo)
  },
  distribution = approximate()
)

multi_outcome_test_rank_p
```

```
    Asymptotic General Independence Test

data:  n_sec_i, c_sec_i, vic12_i, dt_impact_i, ps_impact_i, social_dis, activities_index by soldvsnot17F (0, 1)
     stratified by parms1_fm_p
maxT = 1.5, p-value = 0.6
alternative hypothesis: two.sided
```

```r
multi_outcome_test_rank_perm_p
```

```
    Approximative General Independence Test

data:  n_sec_i, c_sec_i, vic12_i, dt_impact_i, ps_impact_i, social_dis, activities_index by soldvsnot17F (0, 1)
     stratified by parms1_fm_p
maxT = 1.5, p-value = 0.6
alternative hypothesis: two.sided
```

Sensitivity Analysis

Since we have no strong effects, relationships where a test of the null hypothesis of no effects passes the $p < .05$ threshold, we do not do any sensitivity analysis.

References