Propensity Scores

David M. Vock

PubH 7485/8485

Propensity Score

- Defined for a binary treatment to be $P(A = 1|X) = \pi(X)$
- Under the assumption of no unmeasured confounding and consistency, the propensity score is a balancing score
- A balancing score $b(X)$ is any function of the covariates X such that A is independent of X given $b(X)$
- This means the distribution of the covariates is the same in the treatment and control group at the same level of $b(X)$

Key Lemma

- If we assume no unmeasured confounders (i.e., A is independent of $\{Y^1, Y^0\}$ given X) then A is independent of $\{Y^1, Y^0\}$ given $\pi(X)$

Implication of this lemma

We can estimate the ATE using

1) Regression model with propensity score and treatment as covariates
2) Propensity stratification (the "nonparametric" version of 1)
3) Matching (more on this later)
4) Inverse probability (of treatment) weighting

Propensity Score Regression

- Estimate propensity score using logistic regression or random forests or some other technique.
- Include propensity score and treatment and covariates in model
- Typically include nonlinear terms for propensity score (e.g., quadratic terms, restricted cubic splines)

Propensity Model

- Positivity and near-positivity issues with ward
- For now let's leave ward out of the propensity score model (we'll come back to this later)
- Just include "main effects" for the other terms

Propensity Model

```
p1 <- glm(PHN.C1 ~ (PERSONS + VOTE96.1 + NEW + MAJORPTY + AGE)
  data = imai, family = "binomial")
round(summary(p1)$coefficients, digits = 3)
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -5.314      0.335 -15.849    0.000
## PERSONS           0.004      0.131   0.029    0.977
## VOTE96.1          0.564      0.174   3.241    0.001
## NEW               0.014      0.247   0.055    0.956
## MAJORPTY          0.141      0.163   0.865    0.387
## AGE               0.020      0.004   5.701    0.000
```

Obtain Estimated Propensity Scores and Include in Outcome Model

```
# note: rcs function is from rms library
imai$ps <- predict(p1, type = "response")
m1.ps <- glm(VOTED98 ~ PHN.C1*rcs(ps, 5), data = imai, family
round(summary(m1.ps)$coefficients, digits = 3)
```

```
##                              Estimate Std. Error z value Pr(>|z|
## (Intercept)                    -3.538      0.286 -12.383    0.00
## PHN.C1                          2.982      1.984   1.503    0.13
## rcs(ps, 5)ps                  188.724     25.515   7.397    0.00
## rcs(ps, 5)ps'                -471.988    432.163  -1.092    0.27
## rcs(ps, 5)ps''                490.781    851.988   0.576    0.56
## rcs(ps, 5)ps'''                50.678    573.418   0.088    0.93
## PHN.C1:rcs(ps, 5)ps          -187.408    185.195  -1.012    0.31
## PHN.C1:rcs(ps, 5)ps'         1601.147   3254.014   0.492    0.62
## PHN.C1:rcs(ps, 5)ps''       -2607.558   6416.154  -0.406    0.68
## PHN.C1:rcs(ps, 5)ps'''       1101.126   4283.593   0.257    0.79
```

Propensity Score Regression Adjustment

- Get predicted value for each individual in the dataset assuming that they are (a) in the treatment group and (b) in the control group
- Take the difference in the mean predicted value to get estimate of ATE

PS Regression Adjustment

```
data_trt <- data_ctr <-imai
data_trt$PHN.C1 = 1
data_ctr$PHN.C1 = 0
pred1.ps <- predict(m1.ps, newdata = data_trt, type = "respons
pred0.ps <- predict(m1.ps, newdata = data_ctr, type = "respons
ATE.ps <- mean(pred1.ps - pred0.ps)
print(ATE.ps, digits = 3)
```

```
## [1] 0.108
```

Bootstrap for PS Regression Adjustment

```
set.seed(1101985)
B <- 100
ATE.ps.boot <- NULL
n <- nrow(imai)
for(i in 1:B) {
  imai.boot <- imai[sample(1:n, n, replace = TRUE), ]
  m1.ps.boot <- glm(VOTED98 ~ PHN.C1*rcs(ps, 5), data = imai.boot,
    family = "binomial")
  data_trt.boot <- imai.boot
  data_trt.boot$PHN.C1 = 1
  data_ctr.boot <- imai.boot
  data_ctr.boot$PHN.C1 = 0
  pred1.ps.boot <- predict(m1.ps.boot, newdata = data_trt.boot,
    type = "response")
  pred0.ps.boot <- predict(m1.ps.boot, newdata = data_ctr.boot,
    type = "response")
  ATE.ps.boot <- c(ATE.boot, mean(pred1.ps.boot - pred0.ps.boot))
  }
```

Voting Example: PS Regression Adjustment Results

```
## [1] "Average Treatment Effect"

## [1] 0.108

## [1] "Bootstrap SE"

## [1] 0.0336

## [1] "Bootstrap Normal 95% CI"

## [1] 0.042 0.174
```

PS Regression Adjustment Results: Key Assumptions

Identifying

1) Consistency
2) No Unmeasured confounding

Modeling

1) Outcome model (given propensity score) correctly specified. Note this may involve extrapolation if there is not sufficient overlap in ps between treatment and control.
2) Propensity score model (given all confounders) correctly specified.

Propensity Score Stratificaition

- Estimate propensity score using logistic regression or random forests or some other technique.
- Divide the data into quintiles (or deciles) based on the estimated propensity score $\hat{\pi}(X_i)$
- Estimate the ATE using

$$\hat{\delta} = \sum_{j=1}^{5} (\overline{Y}_{1j} - \overline{Y}_{0j}) \frac{n_j}{n} \tag{1}$$

where $(\overline{Y}_{1j}, \overline{Y}_{0j})$ are the sample average response among subjects in the jth quintile receiving treatments 1 and 0, respectively, and n_j is the number of individuals in the jth quintile

Propensity Score Stratificaition

- Equivalent to ps regression approach where we categorize ps (rather than using splines or other nonlinear term)
- Bit ad hoc but seems to work reasonably well for many applications

Propensity Score Stratification - Putting it All Together

1) In practice we do not know the propensity score, estimate propensity score using logistic regression (or other flexible methods)
2) Get predicted value of $\pi(X_i)$ - i.e., $\pi(X_i; \hat{\gamma})$
3) Estimate ATE using

$$\hat{\delta} = \sum_{j=1}^{5} (\overline{Y}_{1j} - \overline{Y}_{0j}) \frac{n_j}{n} \tag{2}$$

4) Use nonparametric bootstrap to get standard error and CI

Obtain Estimated Propensity Scores and Divide Into Quintiles

```
ps <- predict(p1, type = "response")
ps_quintile <- cut(ps,
    breaks = c(0, quantile(ps, p = c(0.2, 0.4, 0.6, 0.8)), 1),
table(ps_quintile, imai$PHN.C1)
```

```
##
## ps_quintile    0    1
##           1 2140   32
##           2 2148   22
##           3 2160   36
##           4 2093   59
##           5 2041   98
```

Propensity Scores Stratification

```
n <- nrow(imai)
nj <- table(ps_quintile)
te_quintile <- tapply(imai$VOTED98[imai$PHN.C1 == 1], ps_quint
    tapply(imai$VOTED98[imai$PHN.C1 == 0], ps_quintile[imai$PH
print(round(te_quintile, 3))
```

```
##     1     2     3     4     5
## 0.162 0.217 0.073 0.013 0.123
```

```
ATE_PSS <- sum(te_quintile *nj/n)
print(round(ATE_PSS, 3))
```

```
## [1] 0.118
```

Bootstrap for PS Stratification

```
set.seed(1101985)
B <- 100
ATE_PSS.boot <- NULL
n <- nrow(imai)
for(i in 1:B) {
  imai.boot <- imai[sample(1:n, n, replace = TRUE), ]
  p1.boot <- glm(PHN.C1 ~ (PERSONS + VOTE96.1 + NEW +
      MAJORPTY + AGE),
  data = imai.boot, family = "binomial")
  ps.boot <- predict(p1.boot, type = "response")
  ps_quintile.boot <- cut(ps.boot,
        breaks = c(0, quantile(ps.boot, p = c(0.2, 0.4, 0.6, 0.8)), 1), labels = 1:5)
  nj.boot <- table(ps_quintile.boot)
    te_quintile.boot <- tapply(imai.boot$VOTED98[imai.boot$PHN.C1 == 1],
        ps_quintile.boot[imai.boot$PHN.C1 == 1], mean) -
    tapply(imai.boot$VOTED98[imai.boot$PHN.C1 == 0], ps_quintile.boot[imai.boot$PHN.C1 == 0], mean)
    ATE.boot <- sum(te_quintile.boot *nj/n)
  ATE_PSS.boot <- c(ATE_PSS.boot, ATE.boot)
  }
```

Voting Example: Propensity Score Stratification Analysis

```
## [1] "Average Treatment Effect"
## [1] 0.118
## [1] "Bootstrap SE"
## [1] 0.035
## [1] "Bootstrap Normal 95% CI"
## [1] 0.049 0.186
```

Propensity Score Stratification Results: Key Assumptions

Identifying

1) Consistency
2) No Unmeasured confounding

Modeling

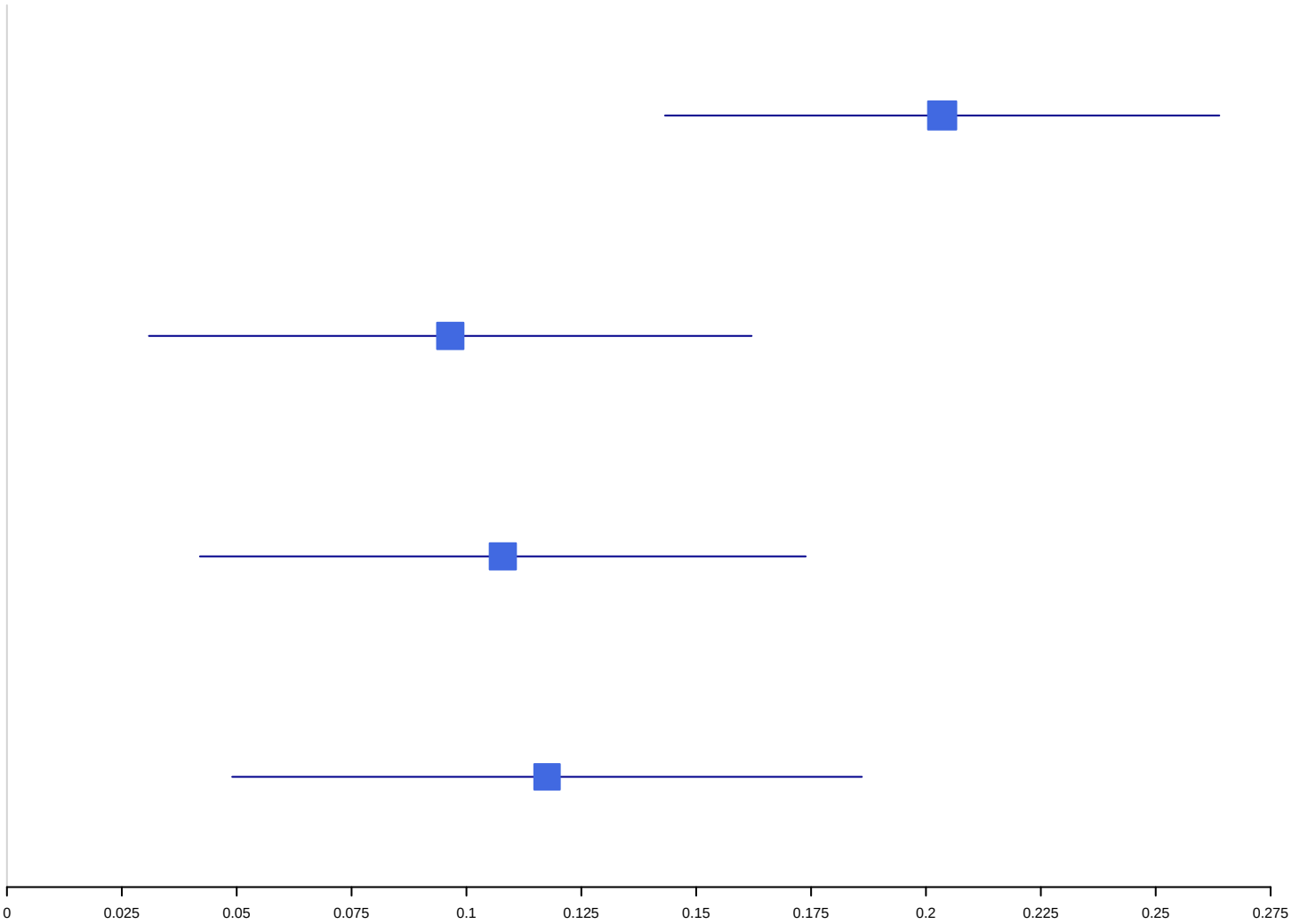
1) Outcome model (given propensity score) correctly specified. That is, the propensity score can be discretized.
2) Propensity score model (given all confounders) correctly specified.

Putting it All Together

- Compare different causal estimators with ITT
- Will add to this throughout the course

Putting it All Together

Method	ATE
Unadjusted	0.204
Regression	0.096
PS Reg	0.108
PSS	0.118



Inverse Probability Weighting: A Toy Example

- The population in the the 7 county metro area has 3.0 million and the entire state of Minnesota has a population of 5.5 million
- Suppose that I can conduct a simple random sample of 300 residents in the metro area and 700 residents "out-state"
- In our simple random sample, among those in the 7 county metro area, Gov. Walz's approval was 174/300 (58%) but in out-state it was only 336/700 (48%)
- What would your estimate of the state-wide approval be?

Inverse Probability Weighting: A Toy Example

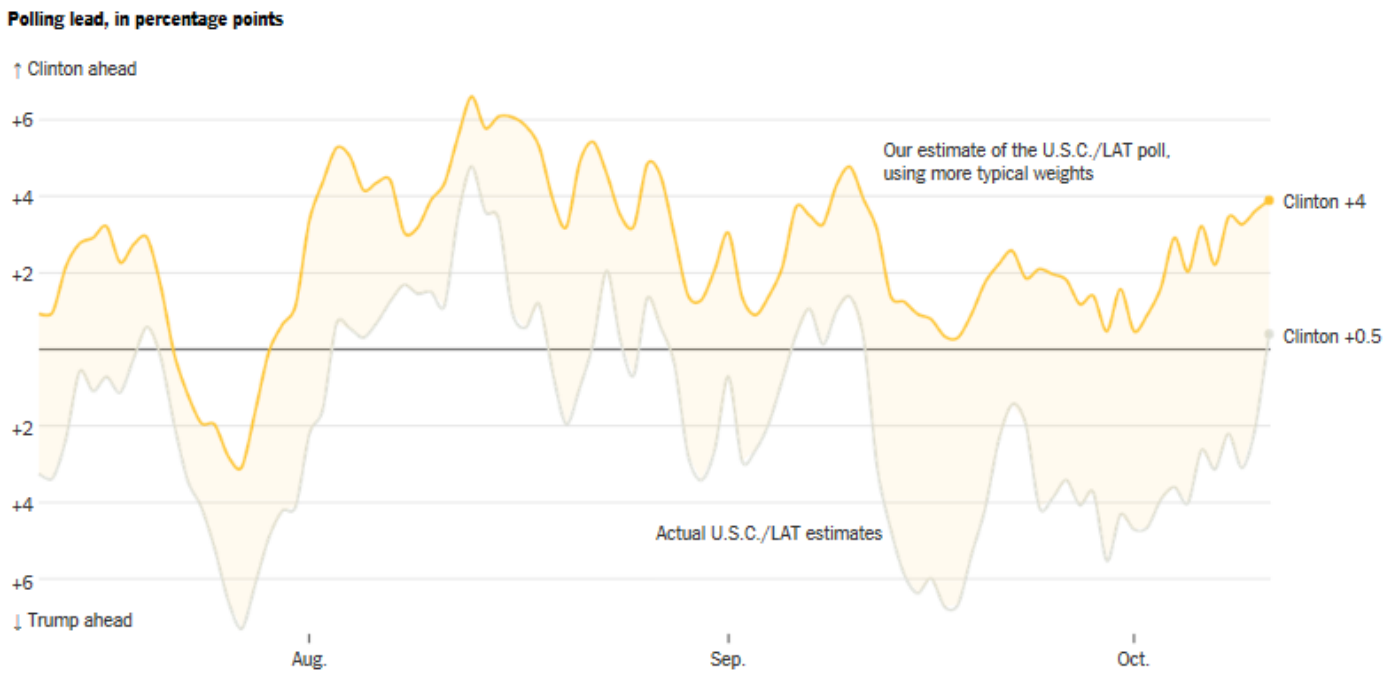
- In the example, there was an over-representation of people from "out-state" and an under-representation of those from the metro
- Those "over-represented" were "under-weighted" and those "under-represented" were "over-weighted"
- Here the imbalance by geography was by design, but even when it is not, same analysis principles apply

Survey Weights in 2016 Election Polling

TheUpshot

THE 2016 RACE

How One 19-Year-Old Illinois Man Is Distorting National Polling Averages



By Nate Cohn

Oct. 12, 2016

505

There is a 19-year-old black man in Illinois who has no idea of the role he is playing in this election.

Sign up for The Upshot Newsletter

Inverse Probability Weighting (IPW1)

- Assume that propensity model is known for now
- Remember: Assuming that we have iid data, then $\frac{1}{n}\sum_{i=1}^{n} g(Y_i)$ converges in probability to $E\{g(Y)\}$ by the weak law of large numbers
- Propose to estimate $E(Y^1)$ by $\frac{1}{n}\sum_{i=1}^{n} \frac{A_i Y_i}{\pi(X_i)}$
- Need to show that $E\left\{\frac{A_i Y_i}{\pi(X_i)}\right\} = E(Y^1)$
- Need to assume that $1 > \pi(X_i) > 0$ for all X_i (positivity assumption)
- How would I estimate $E(Y^0)$?
- What is $E\{A/\pi(X_i)\}$?

Inverse Probability Weighting (IPW2)

- Some people estimate $E(Y^1)$ by $\frac{1}{n}\sum_{i=1}^{n}\frac{A_i Y_i}{\pi(X_i;\hat{\gamma})} / \frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\pi(X_i;\hat{\gamma})}$
- Note that $\frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\pi(X_i;\hat{\gamma})} \approx 1$ so the estimators are very similar
- This is the estimator that you would get if you used the weighted.mean function in R

Inverse Probability Weighting - Putting it All Together

1) In practice we do not know the propensity score, estimate propensity score using logistic regression (or other flexible methods)
2) Get predicted value of $\pi(X_i)$ - i.e., $\pi(X_i; \hat{\gamma})$
3) Estimate $E(Y^1)$ by $\frac{1}{n}\sum_{i=1}^{n} \frac{A_i Y_i}{\pi(X_i;\hat{\gamma})}$ and $E(Y^0)$ by $\frac{1}{n}\sum_{i=1}^{n} \frac{(1-A_i) Y_i}{1-\pi(X_i;\hat{\gamma})}$
4) Take their difference to estimate δ
5) Use nonparametric bootstrap to get standard error and CI

IPW Estimates to Reduce Confounding

- Note that I could also compute the IP weighted mean of the covariates by treatment group
- If the propensity model is correctly specified, then the IP weighted mean of the covariates should be equal in the two treatment groups; i.e., the standardized mean difference should be 0
- Before we look at outcome data, calculating the weighted SMD can be used to assess how well IP weighting is doing balancing the two groups. If imbalances persist $\rightarrow$ fit new PS model

Unweighted Differences in Key Variables Between Groups

```
knitr::kable(t1)
```

	level	Not Contacted	Contacted	SMD
n		10582	247	
Voted in 1998 (%)	No	5881 (55.6)	87 (35.2)	0.418
	Yes	4701 (44.4)	160 (64.8)	
Voters in household (%)	1 Voter	5269 (49.8)	119 (48.2)	0.032
	2+ Voters	5313 (50.2)	128 (51.8)	
Age (years) (mean (SD))		49.43 (18.73)	58.31 (19.85)	0.460
Voted in 1996 (%)	No	4965 (46.9)	71 (28.7)	0.382
	Yes	5617 (53.1)	176 (71.3)	
New voter (%)	Previous Voter	8452 (79.9)	219 (88.7)	0.243
	New Voter	2130 (20.1)	28 (11.3)	
Party affilation (%)	Republican	2701 (25.5)	49 (19.8)	0.136
	Democrat	7881 (74.5)	198 (80.2)	
Ward of residence (%)	2	317 (3.0)	3 (1.2)	0.565
	3	273 (2.6)	3 (1.2)	
	4	234 (2.2)	2 (0.8)	
	5	200 (1.9)	4 (1.6)	
	6	435 (4.1)	5 (2.0)	
	7	337 (3.2)	3 (1.2)	
	8	360 (3.4)	7 (2.8)	
	9	387 (3.7)	9 (3.6)	
	10	452 (4.3)	16 (6.5)	
	11	451 (4.3)	19 (7.7)	
	12	364 (3.4)	9 (3.6)	
	13	435 (4.1)	10 (4.0)	
	14	383 (3.6)	6 (2.4)	
	15	329 (3.1)	8 (3.2)	
	16	240 (2.3)	5 (2.0)	

Weighted Differences in Key Variables Between Groups

```
ps <- predict(p1, type = "response")
imai$weight <- imai$PHN.C1/ps + (1-imai$PHN.C1)/(1-ps)
imaiSvy <- svydesign(ids = ~ 1, data = imai, weights = ~ weight)

tabWeighted <- svyCreateTableOne(vars = vars, strata = "PHN.C1F",
  data = imaiSvy, test = FALSE)
t2 <- print(tabWeighted, smd = TRUE, showAllLevels = TRUE, varLabels = TRUE)
```

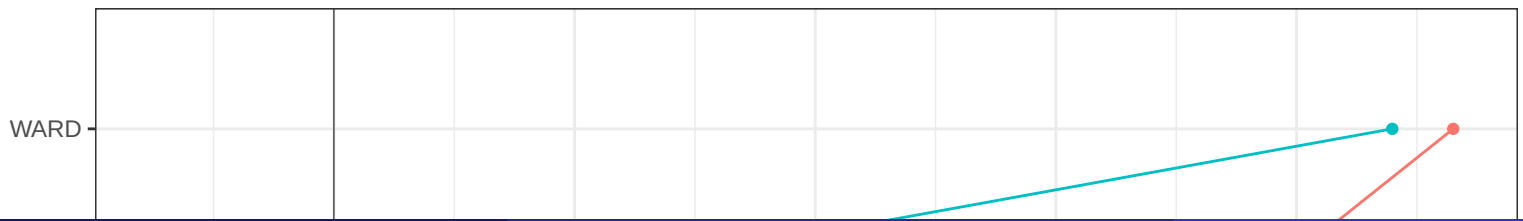
Weighted Differences in Key Variables Between Groups

```
knitr::kable(t2)
```

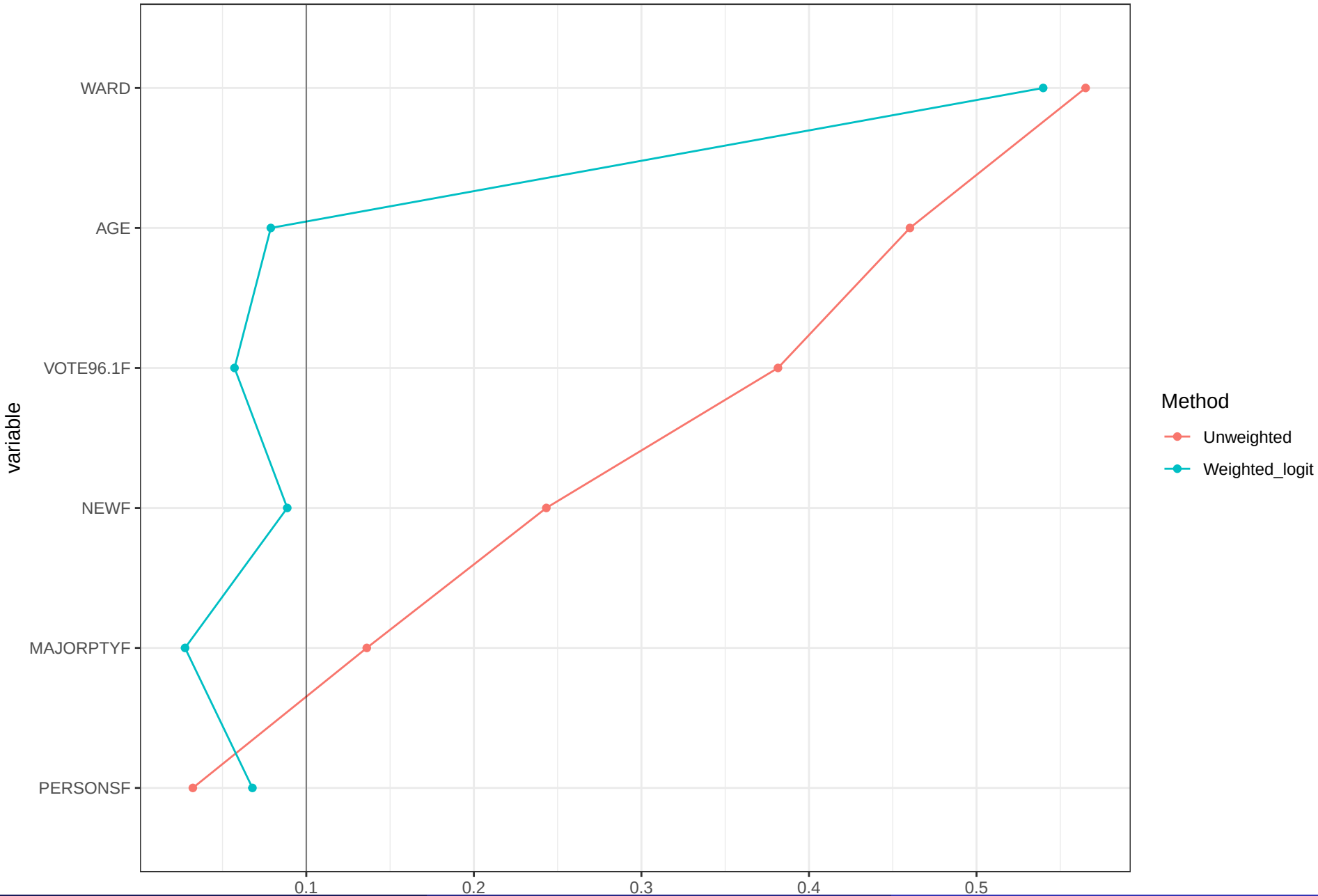
	level	Not Contacted	Contacted	SMD
n		10828.9	11083.5	
Voted in 1998 (%)	No	5992.1 (55.3)	4996.6 (45.1)	0.206
	Yes	4836.8 (44.7)	6086.9 (54.9)	
Voters in household (%)	1 Voter	5388.1 (49.8)	5890.4 (53.1)	0.068
	2+ Voters	5440.8 (50.2)	5193.0 (46.9)	
Age (years) (mean (SD))		49.63 (18.79)	48.10 (19.84)	0.079
Voted in 1996 (%)	No	5036.1 (46.5)	5470.8 (49.4)	0.057
	Yes	5792.8 (53.5)	5612.6 (50.6)	
New voter (%)	Previous Voter	8670.8 (80.1)	8469.7 (76.4)	0.089
	New Voter	2158.1 (19.9)	2613.8 (23.6)	
Party affilation (%)	Republican	2750.0 (25.4)	2949.0 (26.6)	0.028
	Democrat	8078.9 (74.6)	8134.5 (73.4)	
Ward of residence (%)	2	323.3 (3.0)	171.7 (1.5)	0.540
	3	279.2 (2.6)	84.9 (0.8)	
	4	238.2 (2.2)	71.8 (0.6)	
	5	204.3 (1.9)	143.9 (1.3)	
	6	446.8 (4.1)	145.2 (1.3)	
	7	343.7 (3.2)	262.7 (2.4)	
	8	367.9 (3.4)	436.6 (3.9)	
	9	394.5 (3.6)	452.6 (4.1)	
	10	461.2 (4.3)	751.6 (6.8)	
	11	467.4 (4.3)	625.6 (5.6)	
	12	372.0 (3.4)	416.4 (3.8)	
	13	445.4 (4.1)	319.5 (2.9)	
	14	391.9 (3.6)	177.0 (1.6)	
	15	335.6 (3.1)	451.3 (4.1)	
	16	244.5 (2.3)	304.9 (2.8)	

Plot of SMD

```
dataPlot <- data.frame(variable  = rownames(ExtractSmd(tabUnmatched)),
    Unweighted = as.numeric(ExtractSmd(tabUnmatched)),
    Weighted_logit  = as.numeric(ExtractSmd(tabWeighted)))
dataPlot <- dplyr::filter(dataPlot,
                          variable != "VOTED98F")

## Create long-format data for ggplot2
dataPlotMelt <- melt(data          = dataPlot,
    id.vars       = c("variable"),
    variable.name = "Method",
    value.name    = "SMD")

## Order variable names by magnitude of SMD
varNames <- as.character(dataPlot$variable)[order(dataPlot$Unweighted)]

## Order factor levels in the same order
dataPlotMelt$variable <- factor(dataPlotMelt$variable,
    levels = varNames)

## Plot using ggplot2
ggplot(data = dataPlotMelt,
    mapping = aes(x = variable, y = SMD, group = Method, color = Method)) +
    geom_line() +
    geom_point() +
    geom_hline(yintercept = 0.1, color = "black", size = 0.1) +
    coord_flip() +
    theme_bw() + theme(legend.key = element_blank())
```



Plot of SMD


WARD
AGE
VOTE96.1F
NEWF
MAJORPTYF
PERSONSF
variable
0.1
0.2
0.3
0.4
0.5
Method
Unweighted
Weighted_logit

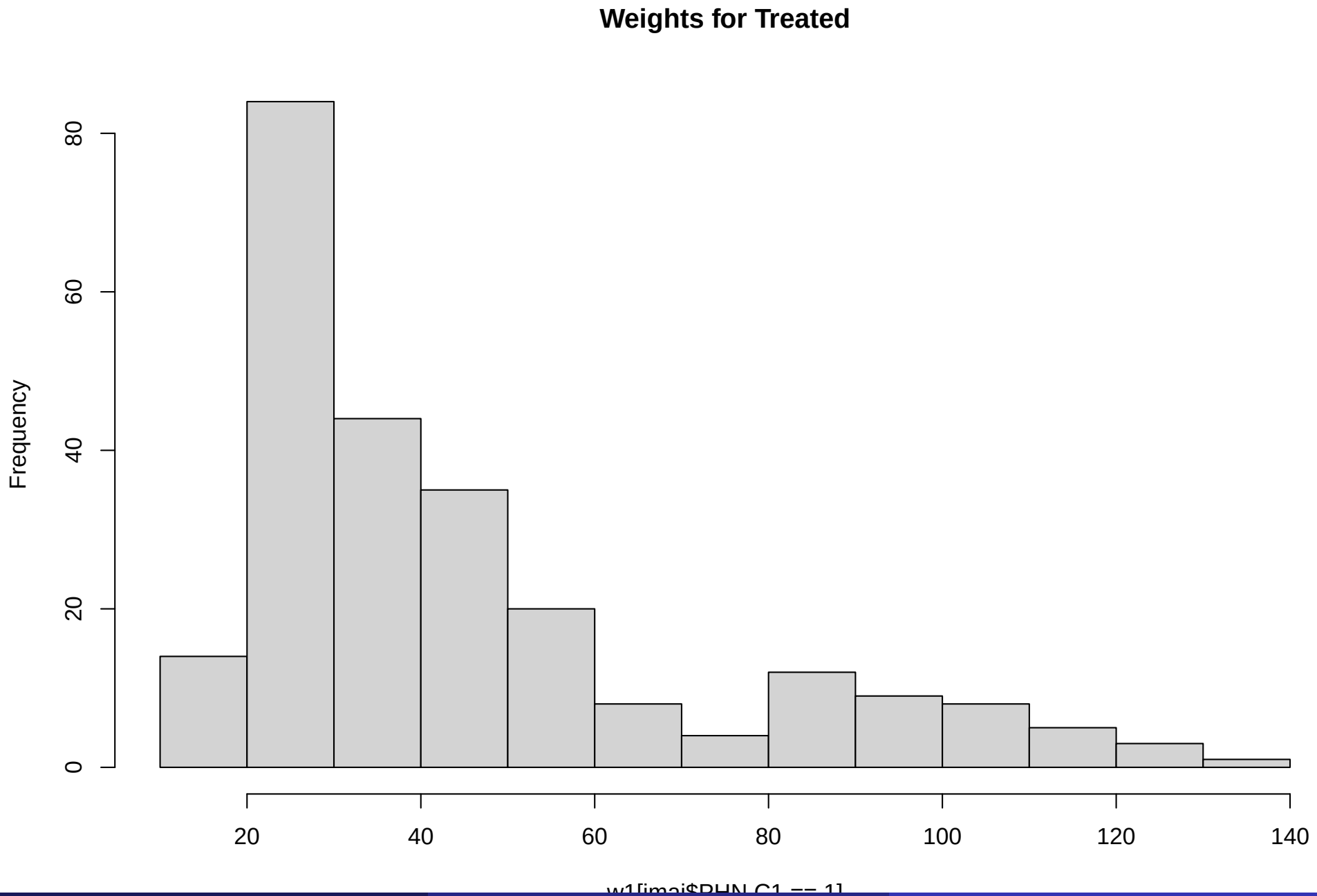
Final notes

- Some (a lot of) people do not use bootstrap to get standard errors and pretend as if γ (parameter in the propensity model) is known –> this is actually conservative. This can be implemented in survey package in R
- Might be concerned about weights that are excessively large relative to weights in the same treatment condition. Some recommend truncating weights at 99th percentile but this is very, very ad hoc.

Form Inverse Probability Weights

```
ps <- predict(p1, type = "response")
w1 <- imai$PHN.C1/ps
w0 <- (1-imai$PHN.C1)/(1-ps)
```

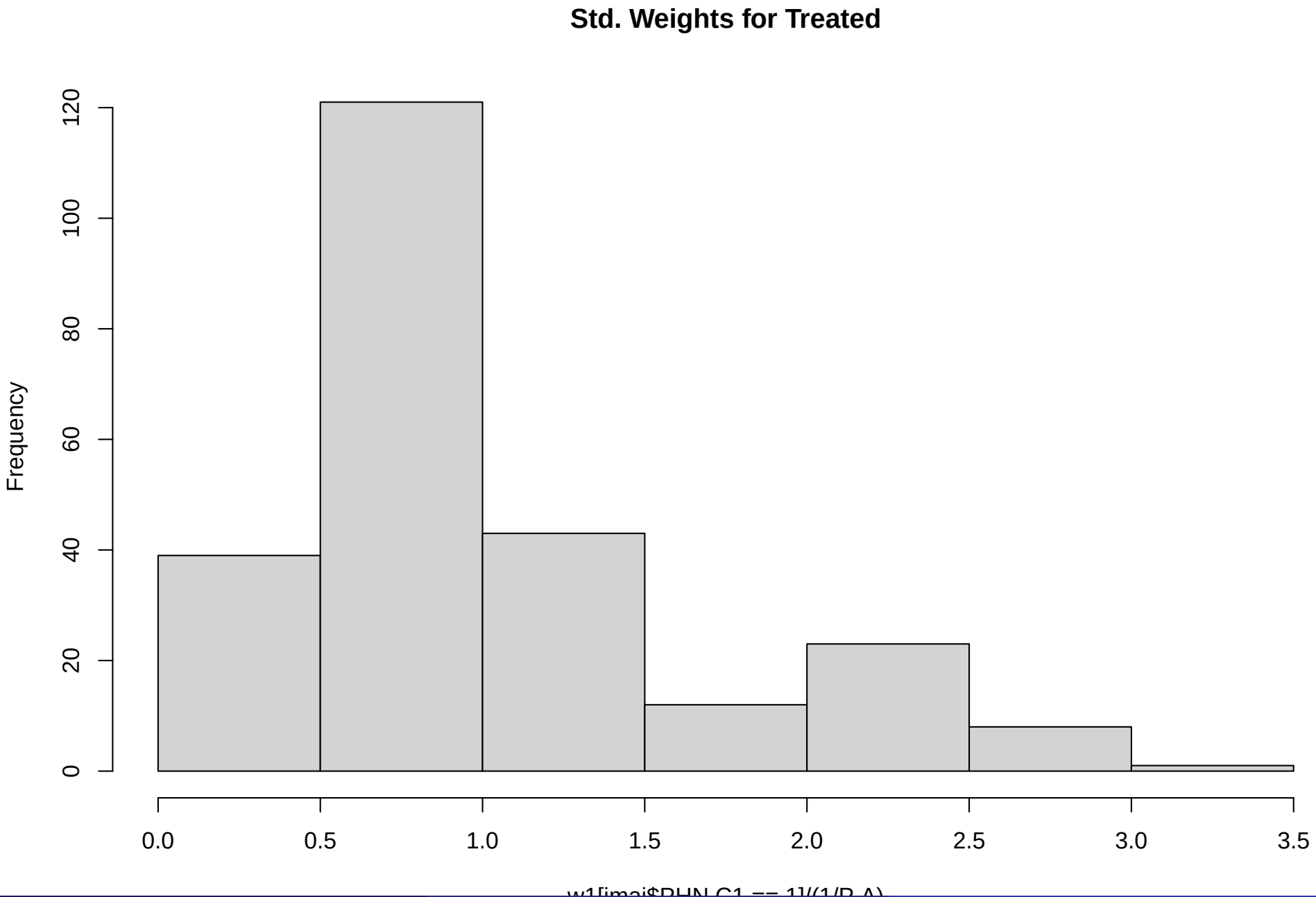
Histogram of Weights



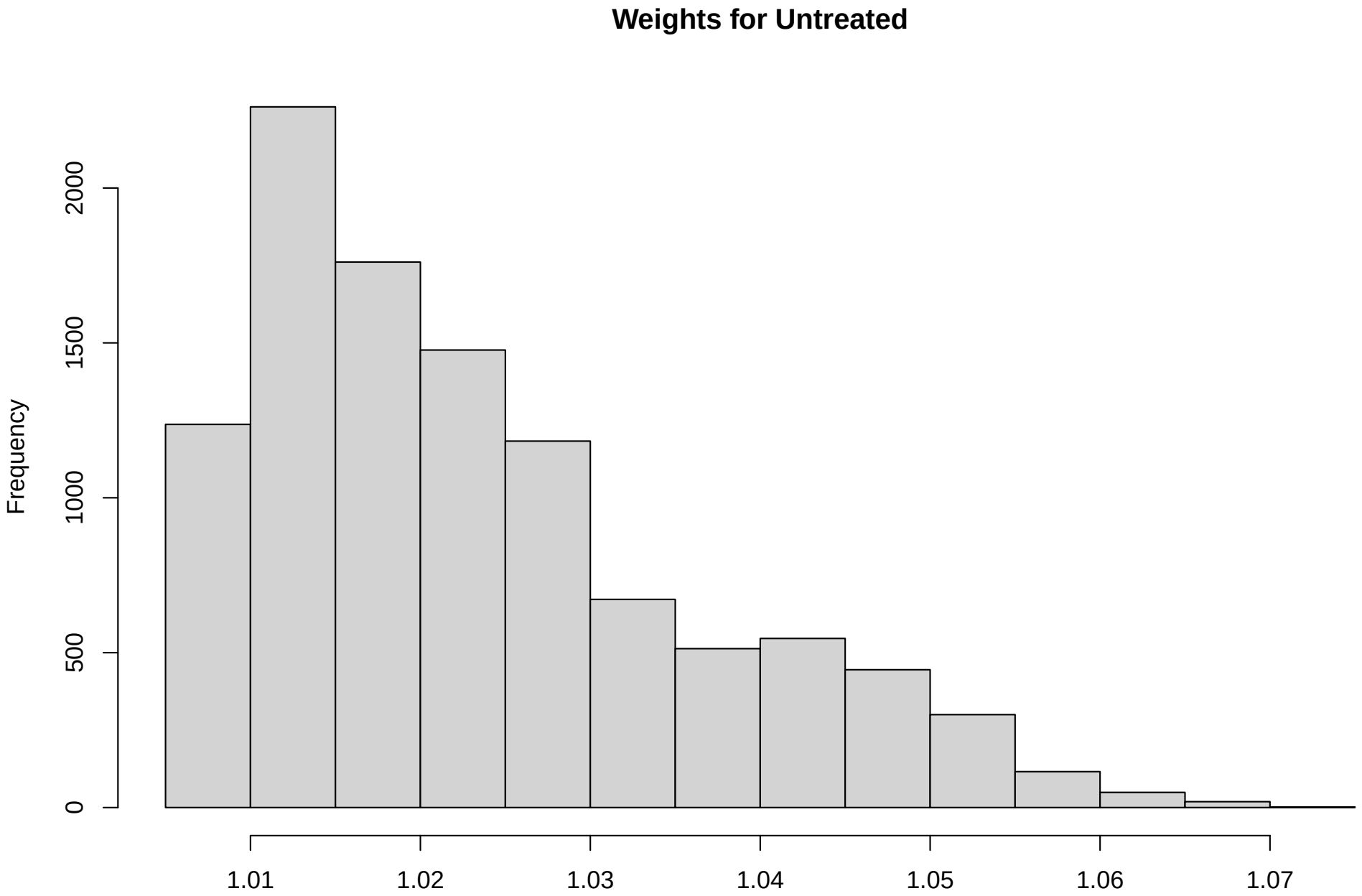
Histogram of Standardized Weights

- Even if treatment were randomized here (i.e., weights were all equal), the weights would be large due to unequal allocation ($1/P(A|X) = 1/P(A) = 42.84$)
- Could divide weights in treated by $1/P(A)$ to get idea of "excess" influence (standardized weights)

Histogram of Standardized Weights



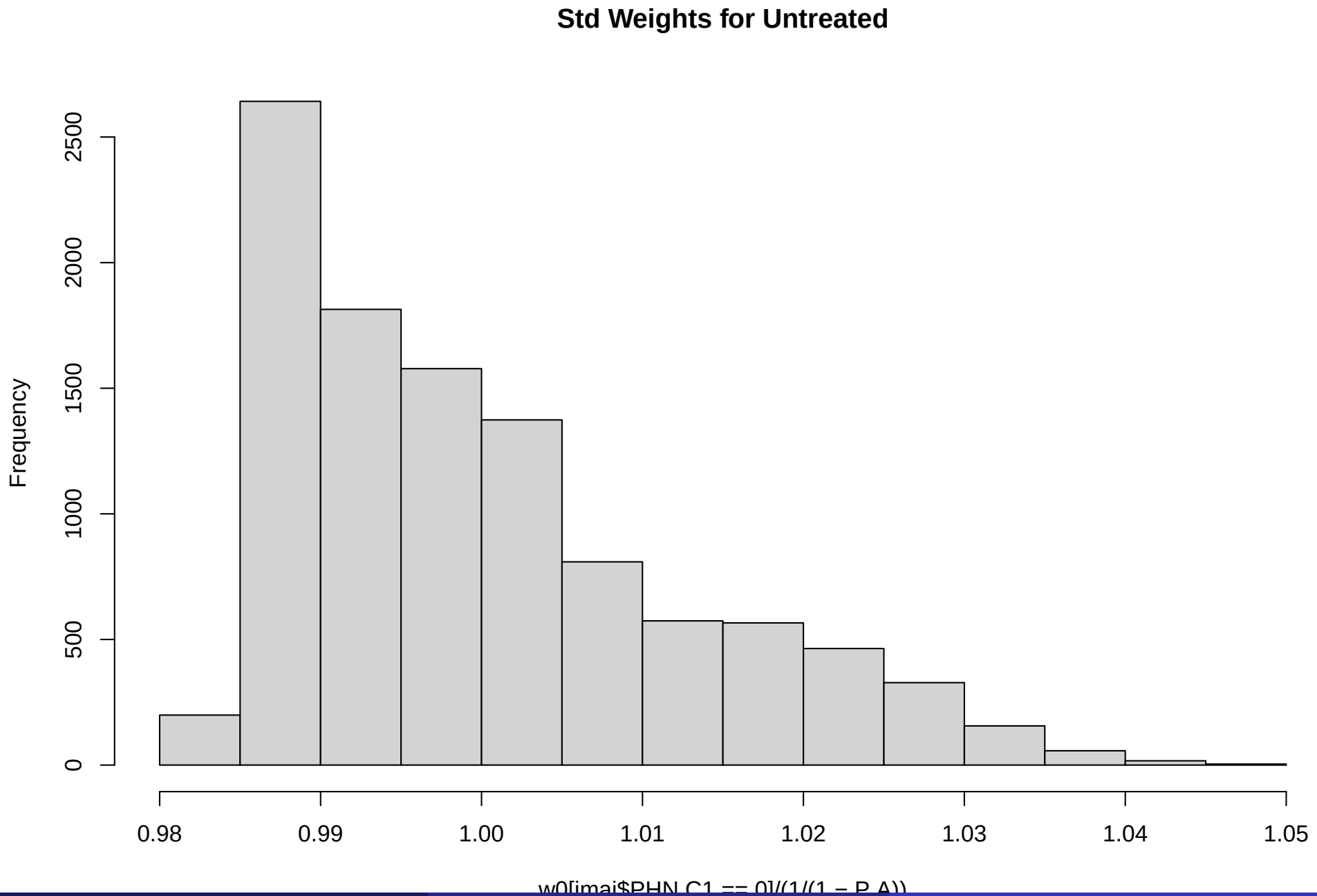
Histogram of Weights



Histogram of Standardized Weights

- Could divide weights in untreated by $1/\{1 - P(A)\}$ to get idea of "excess" influence (standardized weights)

Histogram of Standardized Weights



IPW1 Adjustment

```
ATE_IPW <- mean(imai$VOTED98*w1) - mean(imai$VOTED98*w0)
print(ATE_IPW, digits = 3)
```

```
## [1] 0.115
```

IPW2 Adjustment

```
ATE_IPW2 <- weighted.mean(imai$VOTED98, w1) -
    weighted.mean(imai$VOTED98, w0)
print(ATE_IPW2, digits = 3)
```

```
## [1] 0.103
```

Bootstrap for IPW Estimators

```
set.seed(1101985)
B <- 100
ATE_IPW.boot <- NULL
ATE_IPW2.boot <- NULL
n <- nrow(imai)
for(i in 1:B) {
  imai.boot <- imai[sample(1:n, n, replace = TRUE), ]
  p1.boot <- glm(PHN.C1 ~ (PERSONS + VOTE96.1 + NEW +
      MAJORPTY + AGE),
  data = imai.boot, family = "binomial")
  ps.boot <- predict(p1.boot, type = "response")
  w1.boot <- imai.boot$PHN.C1/ps.boot
  w0.boot <- (1-imai.boot$PHN.C1)/(1-ps.boot)
  ATE_IPW.boot <- c(ATE_IPW.boot,
    mean(imai.boot$VOTED98*w1.boot) - mean(imai.boot$VOTED98*w0.boot))
  ATE_IPW2.boot <- c(ATE_IPW2.boot,
    weighted.mean(imai.boot$VOTED98, w1.boot) - weighted.mean(imai.boot$VOTED98, w0.boot))
  }
```

Voting Example: IPW1 Analysis

```
## [1] "Average Treatment Effect"

## [1] 0.115

## [1] "Bootstrap SE"

## [1] 0.0372

## [1] "Bootstrap Normal 95% CI"

## [1] 0.0425 0.1883
```

Voting Example: IPW2 Analysis

```
## [1] "Average Treatment Effect"
## [1] 0.103
## [1] "Bootstrap SE"
## [1] 0.0362
## [1] "Bootstrap Normal 95% CI"
## [1] 0.0315 0.1736
```

Voting Example: IPW2 Analysis Using Survey Pkg

```
imaiSvy <- svydesign(ids = ~ 1, data = imai, weights = ~ weigh
fitSvymodel <- svyglm(VOTED98 ~ PHN.C1, imaiSvy, family = "bi

## Warning in eval(family$initialize): non-integer #successes
summary(fitSvymodel)

##
## Call:
## svyglm(formula = VOTED98 ~ PHN.C1, design = imaiSvy, family
##
## Survey design:
## svydesign(ids = ~1, data = imai, weights = ~weight)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.21420    0.01957 -10.947  < 2e-16 ***
## PHN.C1       0.41157    0.15369   2.678  0.00742 **
```

Voting Example: IPW2 Analysis Using Survey Pkg

- Note that this gives me a causal log odds ratio. We could transform to get difference in proportions but would need Delta theorem or bootstrap to get SE/CI

Voting Example: IPW2 Analysis Using Survey Pkg

```
## [1] "Average Treatment Effect"
```

```
## (Intercept)
##       0.103
```

IPW Results: Key Assumptions

Identifying

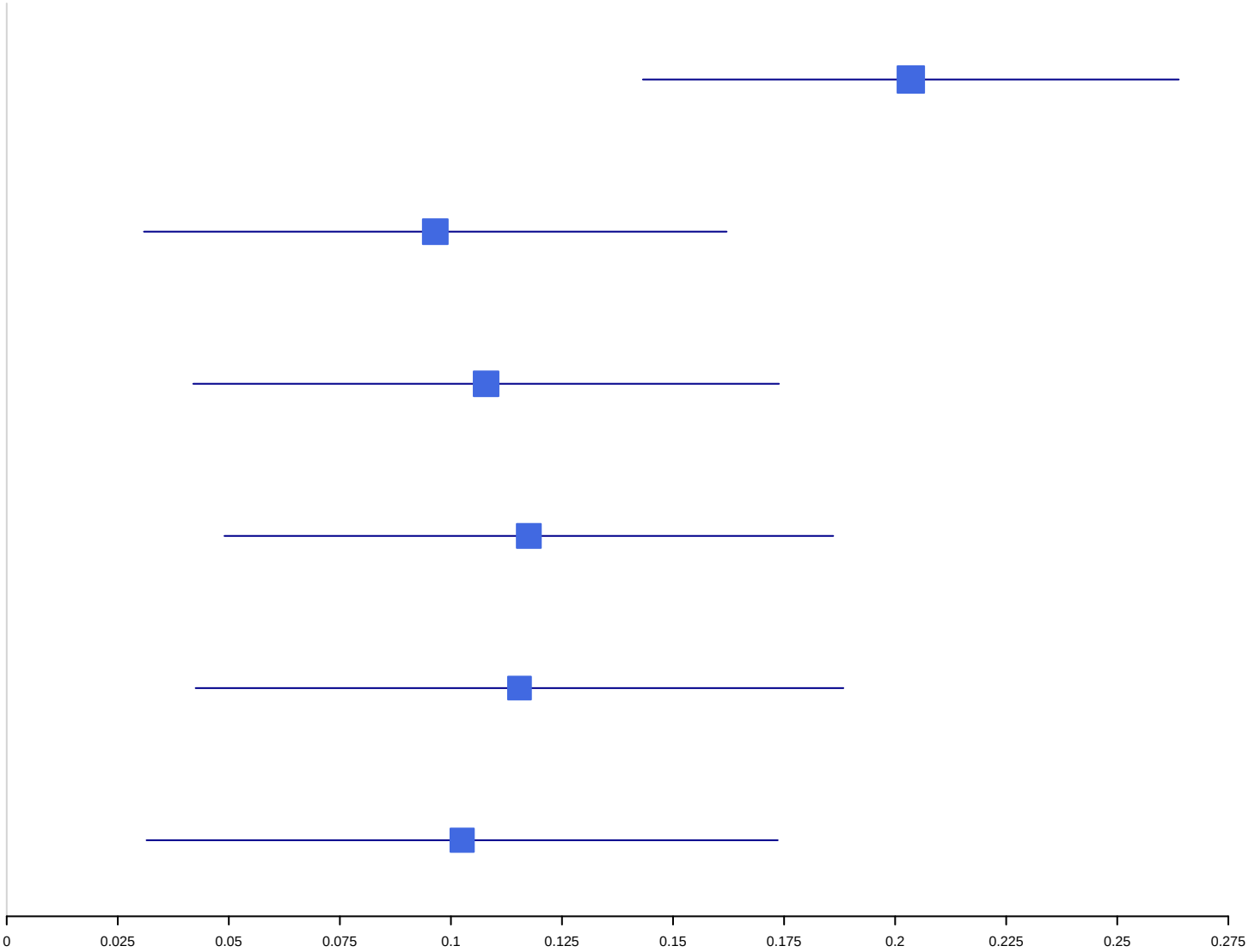
1) Consistency
2) No Unmeasured confounding
3) Positivity

Modeling

1) Propensity score model (given all confounders) correctly specified.

Putting it All Together

Method	ATE
Unadjusted	0.204
Regression	0.096
PS Reg	0.108
PSS	0.118
IPW1	0.115
IPW2	0.103



Advantages/Disadvantages of IPW versus regression model

- Both approaches require assuming consistency and no unmeasured confounding
- IPW requires assuming positivity
- IPW requires correctly specifying propensity model; regression approach requires specifying the outcome model
- IPW: can fit a single model (propensity score) and then obtain causal estimates for several outcomes
- Perception that modeling treatment allocation "easier" than outcome
- IPW tends to be more variably (even without extreme weights)

Simulation Example

- Let $X_i \sim N(0,1)$, $Y_i^1|X_i \sim N(0.5+\gamma X, 1)$ and $Y_i^0|X_i \sim N(\gamma X, 1)$. ATE $= E(Y^1) - E(Y^0) = 0.5$
- Note that in the "real world" we do not observe $\{Y_i^1, Y_i^0$ but would observe $Y_i = A_i Y_i^1 + (1-A_i)Y_i^0$. This implies that $Y_i|A_i, X_i \sim (0.5A_i + \gamma X_i, 1)$
- Let $A_i|X_i \sim$ Bernoulli(p_i) where $p_i = \exp(0+\alpha X_i)/\{1+\exp(0+\alpha X_i)\}$
- Generate a sample of size 500 consistent with this data generating mechanism with $\gamma = 1$ and $\alpha = 1$.
- With these coefficients the R^2 for regressing the outcome on X in placebo group is 0.5 and C-index for the treatment allocation ≈ 0.75
- 100 Monte Carlo datasets

Simulation Results

```
##          bias    sd
## IPW -0.00575 0.136
## REG  0.00381 0.109
```