Application to 2016 Election Data

```
## Erin's path
path_data = "~/Dropbox/Documents/2019__2020/work/kpop/2016_reweighting_example/data/"
```

```
vote_contrast <- quote((recode_vote_2016Democrat - recode_vote_2016Republican) /
                           (recode_vote_2016Democrat + recode_vote_2016Republican))

### Function for creating targets from auxiliary information and formula
create_targets <- function (target_design, target_formula) {
    target_mf <- model.frame(target_formula, model.frame(target_design))
    target_mm <- model.matrix(target_formula, target_mf)
    wts <- weights(target_design)
    colSums(target_mm * wts) / sum(wts)
}
```

Load Data

```
## SURVEY DATA (PEW)
### Load
###This needs to be loaded from dropbox/local not github bc it's too big
pew <- readRDS(paste0(path_data, "pew.rds"))

## AUXILIARY INFORMATION (CCES)
### Load
#This needs to be loaded from dropbox/local not github bc it's too big
cces <- readRDS(paste0(path_data, "cces.rds"))
### Drop invalid cases
cces <- cces %>%
    filter((CC16_401 == "I definitely voted in the General Election.") &
               !is.na(commonweight_vv_post))

## make recode_educ_white column
cces <- cces %>%
  mutate(recode_educ_white = factor(case_when(recode_race == "White" ~ as.character(recode_educ),
                              TRUE ~ "No Split"), levels = c(levels(cces$recode_educ), "No Split")))

pew <- pew %>%
  mutate(recode_educ_white = factor(case_when(recode_race == "White" ~ as.character(recode_educ),
                              TRUE ~ "No Split"), levels = c(levels(pew$recode_educ), "No Split")))

### Actual results
pres <- readRDS(paste0(path_data, "election.rds"))

natl_margin <- pres %>%
    summarise(margin = (sum(demtotal) - sum(reptotal)) /
                  (sum(demtotal) + sum(reptotal))) %>%
```

```r
    as.numeric()
natl_margin
```

```
## [1] 0.02325013
```

```r
formula_rake_demos_noeduc <- ~recode_age_bucket + recode_female + recode_race + recode_region + recode_
formula_rake_demos_weduc <- ~recode_age_bucket + recode_female + recode_race + recode_region +
    recode_educ + recode_pid_3way
formula_ps <- ~recode_age_3way + recode_female + recode_race +
    recode_region + recode_educ_3way + recode_pid_3way
#add interaction with pid
formula_retrospective <- ~recode_age_bucket:recode_pid_3way + recode_female:recode_pid_3way+
    recode_race_educ_reg:recode_pid_3way
```

```r
## Find Missing Strata
## Make "strata" variable in CCES and Pew
cces <- bind_cols(cces, cces %>%
                    unite("strata", all.vars(formula_ps), remove = FALSE) %>%
                    unite("strata_wage", c(all.vars(formula_ps), "recode_age"), remove = FALSE) %>%
                    select(strata, strata_wage))

pew <- bind_cols(pew, pew %>%
                   unite("strata", all.vars(formula_ps), remove = FALSE) %>%
                   unite("strata_wage", c(all.vars(formula_ps), "recode_age"), remove = FALSE) %>%
                   select(strata, strata_wage))


missing_strata <- unique(cces$strata)[!(unique(cces$strata) %in% unique(pew$strata))]

## recode missing age
pew$recode_age[is.na(pew$recode_age)] <- mean(pew$recode_age, na.rm = TRUE)
cces$recode_age[is.na(cces$recode_age)] <- mean(cces$recode_age, na.rm = TRUE)

### Make survey designs

## For Pew, since there are no design weights, assume SRS
pew_srs <- svydesign(ids = ~1, data = pew)
```

```
## Warning in svydesign.default(ids = ~1, data = pew): No weights or probabilities
## supplied, assuming equal probability
```

```r
cces_awt <- svydesign(ids = ~1, weights = ~commonweight_vv_post, data = cces)
```

```r
### Population targets
targets_rake_demos_noeduc <- create_targets(cces_awt, formula_rake_demos_noeduc)
targets_rake_demos_weduc <- create_targets(cces_awt, formula_rake_demos_weduc)
targets_retrospective <- create_targets(cces_awt, formula_retrospective)
```

```r
## Raking on demographics, excluding education
rake_demos_noeduc <- calibrate(design = pew_srs,
                       formula = formula_rake_demos_noeduc,
                       population = targets_rake_demos_noeduc,
                       calfun = "raking")

rake_demos_noeduc <- svydesign(~1, data = pew, weights = weights(rake_demos_noeduc))
```

```r
## Raking on demographics, including education
rake_demos_weduc <- calibrate(design = pew_srs,
                              formula = formula_rake_demos_weduc,
                              population = targets_rake_demos_weduc,
                              calfun = "raking")

rake_demos_weduc <- svydesign(~1, data = pew, weights = weights(rake_demos_weduc))

## Post-stratification
targets_ps <- svytable(formula = ~strata, design = subset(cces_awt, !(strata %in% missing_strata)))
post_stratification <- postStratify(design = pew_srs,
                         strata = ~strata,
                         population = targets_ps)

post_stratification <- svydesign(~1, data = pew, weights = weights(post_stratification))

## Retrospective weighting scheme
rake_retrospective <- calibrate(design = pew_srs,
                              formula = formula_retrospective,
                              population = targets_retrospective,
                              calfun = "raking",
                              force = TRUE)
```

```
## Warning in grake(mm, ww, calfun, bounds = bounds, population = population, :
## Failed to converge: eps=0.00688873036014645 in 51 iterations
```

```r
rake_retrospective <- svydesign(~1, data = pew, weights = weights(rake_retrospective))
```

```r
#############################################
##### PUT FINAL KPOP CODE HERE
#############################################

load("./cleaned data/Full SVD/weights_wPid_full.Rdata")

kpop <- svydesign(~1, data = pew, weights = wts_wPid[, "wtkbal_b.5x"])
kpop_mf <- svydesign(~1, data = pew, weights = wts_wPid[, "wtkbal_mf_b2x"])
```

```r
margin_summary <- round(cbind(cces = svymean(formula_rake_demos_weduc, cces_awt),
                              unweighted = svymean(formula_rake_demos_weduc, pew_srs),
            rake_demos_noeduc = svymean(formula_rake_demos_weduc, rake_demos_noeduc),
            rake_demos_weduc = svymean(formula_rake_demos_weduc, rake_demos_weduc),
            post_stratification = svymean(formula_rake_demos_weduc, post_stratification),
            rake_retrospective = svymean(formula_rake_demos_weduc, rake_retrospective),
            kpop = svymean(formula_rake_demos_weduc, kpop),
            kpop_mf = svymean(formula_rake_demos_weduc, kpop_mf)) * 100, 1) %>%
  data.frame() %>%
  rownames_to_column() %>%
  mutate(variable = case_when(str_detect(rowname, "age") ~ "4-way Age Bucket",
                             str_detect(rowname, "female") ~ "Gender",
                             str_detect(rowname, "race") ~ "Race/Ethnicity",
                             str_detect(rowname, "region") ~ "Region",
                             str_detect(rowname, "educ") ~ "Education Level",
                             str_detect(rowname, "pid") ~ "Party Identification",
                             TRUE ~ "Empty"),
         level = gsub("recode_|age_bucket|female|race|region|educ|pid_3way", "", rowname)) %>%
  select(level, everything(), -rowname, -variable)
```

```
margin_summary_educ <- round(cbind(cces = svymean(~recode_educ_white, cces_awt),
                              unweighted = svymean(~recode_educ_white, pew_srs),
            rake_demos_noeduc = svymean(~recode_educ_white, rake_demos_noeduc),
            rake_demos_weduc = svymean(~recode_educ_white, rake_demos_weduc),
            post_stratification = svymean(~recode_educ_white, post_stratification),
            rake_retrospective = svymean(~recode_educ_white, rake_retrospective),
            kpop = svymean(~recode_educ_white, kpop),
            kpop_mf = svymean(~recode_educ_white, kpop_mf)) * 100, 1) %>%
  data.frame() %>%
  rownames_to_column() %>%
  mutate(variable = case_when(str_detect(rowname, "age") ~ "4-way Age Bucket",
                             str_detect(rowname, "female") ~ "Gender",
                             str_detect(rowname, "race") ~ "Race/Ethnicity",
                             str_detect(rowname, "region") ~ "Region",
                             str_detect(rowname, "educ") ~ "Education Level",
                             str_detect(rowname, "pid") ~ "Party Identification",
                             TRUE ~ "Empty"),
         level = gsub("recode_|age_bucket|female|race|region|educ|pid_3way|educ_3way|_white", "", rownam
  select(level, everything(), -rowname, -variable)
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```
target_margin <- svycontrast(svymean(~recode_vote_2016, cces_awt, na.rm = TRUE),
                       vote_contrast)[1]

comp_df <- data.frame(
    cces = svycontrast(svymean(~recode_vote_2016, cces_awt, na.rm = TRUE),
                       vote_contrast),
    unweighted = svycontrast(svymean(~recode_vote_2016, pew_srs, na.rm = TRUE),
                        vote_contrast),
    rake_demos_noeduc = svycontrast(svymean(~recode_vote_2016, rake_demos_noeduc, na.rm = TRUE),
                        vote_contrast),
    rake_demos_weduc = svycontrast(svymean(~recode_vote_2016, rake_demos_weduc, na.rm = TRUE),
                        vote_contrast),
    post_stratification = svycontrast(svymean(~recode_vote_2016, post_stratification, na.rm = TRUE),
                        vote_contrast),
    rake_retrospective = svycontrast(svymean(~recode_vote_2016, rake_retrospective, na.rm = TRUE),
                        vote_contrast),
    kpop = svycontrast(svymean(~recode_vote_2016, kpop, na.rm = TRUE),
                        vote_contrast),
    kpop_mf = svycontrast(svymean(~recode_vote_2016, kpop_mf, na.rm = TRUE),
```

Table 1: Marginal distribution of important demographics under different weighting models.

	Target (CCES)	Unweighted Pew	Raking Demographics No Education	Raking Demographics with Education	Post-Stratification	Raking Retrospective	Kpop	Kpop Mean First
4-way Age Bucket								
18 to 35	34.8	43.7	34.8	34.8	35.4	34.8	34.4	34.8
36 to 50	22.6	32.6	22.6	22.6	26.7	22.6	23.6	22.6
51 to 64	28.9	21.0	28.9	28.9	29.7	28.9	29.5	28.9
65+	13.7	2.7	13.7	13.7	8.3	13.7	12.5	13.7
Gender								
Female	50.8	47.3	50.8	50.8	50.8	50.8	50.8	50.8
Male	49.2	52.7	49.2	49.2	49.2	49.2	49.2	49.2
Race/Ethnicity								
Black	11.8	8.9	11.8	11.8	11.5	13.4	11.7	11.8
Hispanic	6.5	7.6	6.5	6.5	6.2	6.5	5.9	6.5
Other	6.8	7.1	6.8	6.8	6.5	6.8	6.2	6.8
White	74.9	76.4	74.9	74.9	75.8	73.2	76.3	74.9
Region								
Midwest	23.4	22.3	23.4	23.4	23.2	24.7	23.6	23.4
Northeast	19.7	18.2	19.7	19.7	19.3	19.7	19.5	19.7
South	35.5	37.9	35.5	35.5	36.9	34.8	36.0	35.5
West	21.4	21.6	21.4	21.4	20.6	20.8	20.8	21.4
Education Level								
No HS	6.8	1.8	1.8	6.8	2.4	3.6	1.9	6.8
High school	30.6	19.7	21.5	30.6	28.8	29.4	31.2	30.6
Some college	23.0	16.7	17.7	23.0	24.5	23.0	24.4	23.0
2-year	10.6	11.3	10.6	10.6	9.5	11.7	10.9	10.6
4-year	18.7	28.6	26.2	18.7	22.2	20.2	20.3	18.7
Post-grad	10.4	21.9	22.3	10.4	12.6	12.2	11.3	10.4
Party Identification								
Dem	38.1	34.4	38.1	38.1	38.5	38.1	37.6	38.1
Ind	32.5	35.2	32.5	32.5	32.5	32.5	32.8	32.5
Rep	29.5	30.4	29.5	29.5	29.0	29.5	29.5	29.5

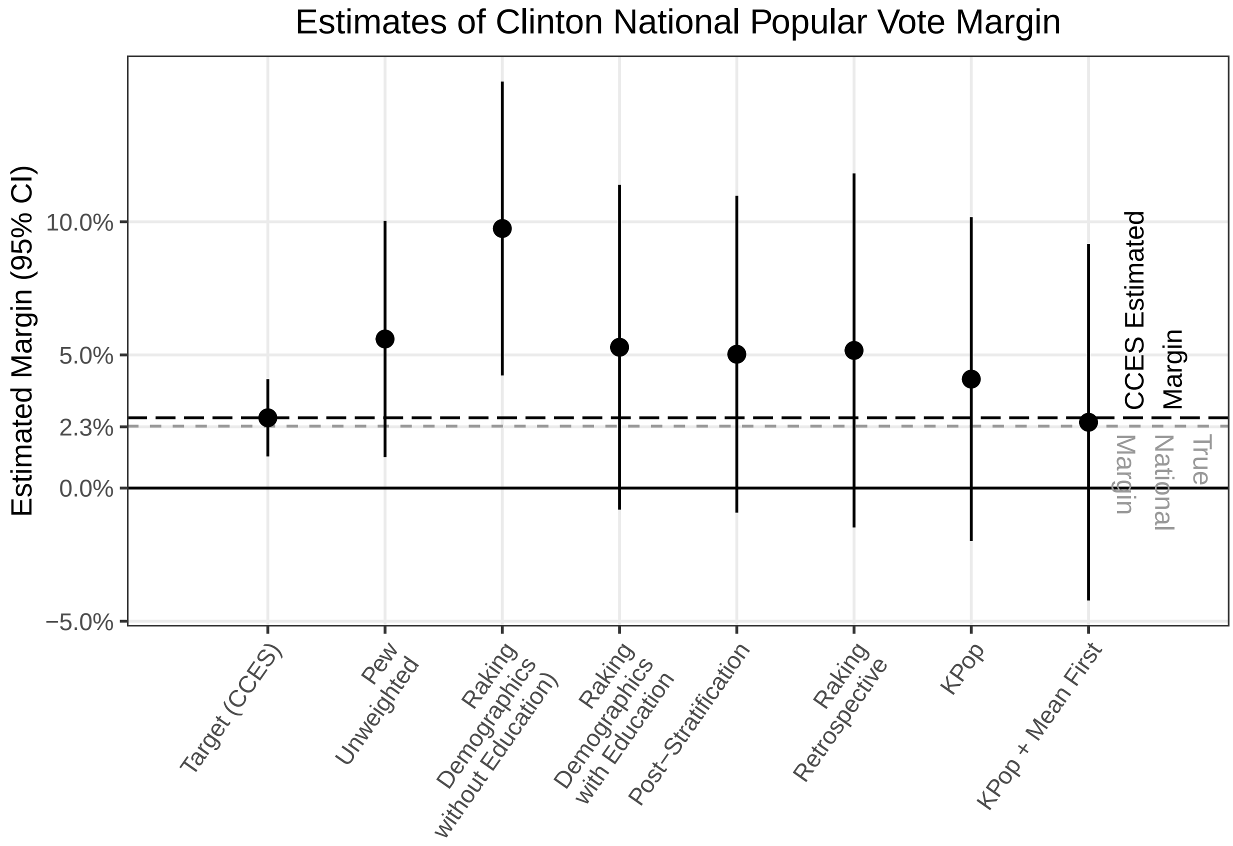
Table 2: Marginal distribution of education level for white voters under different weighting models.

	Target (CCES)	Unweighted Pew	Raking Demographics No Education	Raking Demographics with Education	Post-Stratification	Raking Retrospective	Kpop	Kpop Mean First
Education Level for White Voters								
No HS	4.6	1.1	1.1	4.5	1.8	3.0	1.2	4.8
High school	22.6	14.6	15.5	22.9	23.1	22.6	23.6	21.9
Some college	16.9	11.5	12.1	16.3	19.5	16.9	18.3	17.7
2-year	7.8	8.3	7.0	7.3	6.1	7.8	7.7	7.2
4-year	14.8	23.0	20.7	15.2	17.1	14.8	16.3	14.8
Post-grad	8.3	18.0	18.5	8.8	8.2	8.3	9.1	8.4
Error								
MAE (weighted)	0.0	6.6	5.8	0.4	1.6	0.1	1.2	0.5
MAE (unweighted)	0.0	5.9	5.4	0.4	1.7	0.3	1.4	0.4

```r
                        vote_contrast)) %>%
    pivot_longer(cols = everything(),
                 names_to = c("source", ".value"),
                 names_pattern = "(.*)\\.(.*)") %>%
    rename(est = nlcon) %>%
    mutate(est = est * 100,
           SE = SE * 100,
           err_target = est - target_margin * 100,
           source = str_replace(source, "_", " "),
           source_name = factor(source, labels = c("Target (CCES)",
                                              "KPop",
                                              "KPop + Mean First",
                                              "Post-Stratification",
                                              "Raking\nDemographics\nwithout Education)",
                                              "Raking\nDemographics\nwith Education",
                                              "Raking\nRetrospective",
                                              "Pew\nUnweighted"),
                           ordered = TRUE))

comp_df
```

```
## # A tibble: 8 x 5
##   source              est    SE err_target source_name
##   <chr>             <dbl> <dbl>      <dbl> <ord>
## 1 cces               2.64 0.740      0     "Target (CCES)"
## 2 unweighted         5.60 2.26       2.96  "Pew\nUnweighted"
## 3 rake demos_noeduc  9.75 2.81       7.11  "Raking\nDemographics\nwithout Educa~
## 4 rake demos_weduc   5.29 3.11       2.65  "Raking\nDemographics\nwith Educatio~
## 5 post stratificat~  5.03 3.04       2.39  "Post-Stratification"
## 6 rake retrospecti~  5.17 3.39       2.53  "Raking\nRetrospective"
## 7 kpop               4.09 3.10       1.45  "KPop"
## 8 kpop mf            2.47 3.41        -0.168 "KPop + Mean First"
```

```
comp_df %>%
    ggplot() +
    aes(x = source_name, y = est, ymin = est - 1.96*SE, ymax = est + 1.96*SE) +

    geom_hline(yintercept = c(0, natl_margin * 100, comp_df$est[comp_df$source == "cces"]),
               linetype = c("solid", "dashed", "longdash"),
               color = c("black", "gray60", "black")) +
    geom_pointrange() +
    scale_y_continuous(breaks = round(c(natl_margin * 100, seq(-5, 10, 5)), 1),
                       minor_breaks = NULL,
                       labels = scales::percent_format(scale = 1, accuracy = 0.1)) +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5)) +

    labs(x = NULL, y = "Estimated Margin (95% CI)") +
    ggtitle("Estimates of Clinton National Popular Vote Margin") +
    theme(axis.text.x = element_text(angle = 55, hjust = 1)) +
    annotate(geom = "text", x = 8.65, y = natl_margin * 100, label = " True\n National\n Margin", hjust
    annotate(geom = "text", x = 8.55, y = comp_df$est[comp_df$source == "cces"], label = " CCES Estimat
    theme(legend.title = element_blank()) +
  scale_x_discrete(limits=comp_df$source_name, expand = expansion(0, 1.2))
```



```
ggsave("./plots/weighted_pew_results.pdf", width = 6, height = 4)
```