Doing Weighted Analyses of Survey Data

Jacob Long

3/2/2018

The data

- AARP December 2016 Brain Health Survey
- Target population: Americans older than 40
- Web survey administered by GfK/Knowledge Networks
- N = 2,585
  - 1,535 general population sample (age $>$ 40)
  - 341 Hispanic oversample (conducted in Spanish as needed)
  - 399 Black oversample
  - 310 Asian oversample

Available at: http://go.osu.edu/Dec16AARP_Roper

Get the code for these slides at:
https://github.com/jacob-long/survey-analysis-demo

Key variables

- `cog_fun`: A scale self-assessment of respondent's cognitive functioning.
  - I've coded it such that each item has values of 1 (decreased a lot), 2 (decreased a little), or 3 (stayed the same)
- `engagement`: Index of social activities the respondent engages in and how often (going to church, go dancing, etc.)
- `watch_tv`, `surf_net`, and `use_facebook`: Frequency of communication activities
- `age`, `female`, `educ`, `black`, `hispanic`: Demographics

I'm sparing you the full recoding logic, etc.

Weights

3 weighting variables provided:

- `WEIGHT1` for analyzing general population sample
  - $\sum_i^{N_1} w_{1i} = 1535$
- `WEIGHT2` for analyzing race subgroups
  - $\sum_i^{N_2} w_{2i} = 1370$

Note

$\sum_i^{N_1} w_{1i} + \sum_i^{N_2} w_{2i} \neq 2585$ because the general population includes non-whites who are given weights for the sub-group analyses.

- `WEIGHT3` for analyzing all cases
  - $\sum_i^{N_3} w_{3i} = 2585$

What weights do

Weighted mean

Where x is a vector of numbers ($\{x_1, x_2, ..., x_n\}$)

Unweighted:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Weighted:

Where w is a vector of weights ($\{w_1, w_2, ..., w_n\}$)

$$\bar{x}_w = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} x_i w_i$$

Note that if, as is typical, the sum of weights is the number of cases, then $\sum_{i=1}^{n} w_i = n$.

Weighted variance

Unweighted:

$$Var(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Weighted:

$$Var_w(x) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i (x_i - \bar{x}_w)^2$$

Design effects

Normally, the standard error is:

$$SE = \frac{\sqrt{Var(x)}}{\sqrt{n}}$$

Just plugging in our weighted variance estimate for **standard errors** would be inappropriate because this is not a simple random sample.

Kish (1965) defined the *design effect* as the ratio of variance you get in a complex sample to the variance you get in a simple random sample. One way to approximate this using your weights is to calculate the "effective sample size" (also Kish, 1965).

Effective N

$$N_{eff} = \frac{(\sum_{i=1}^{n} w_i)^2}{\sum_{i=1}^{n} {w_i}^2}$$

and

$$DEff = \frac{N_{actual}}{N_{eff}}$$

When all weights $= 1$ (as with simple random samples with random nonresponse), $N_{eff} = N_{actual}$.

In our data, $N_{eff} = 1794$, so the design effect is 1.44. For simple means, you can get the proper standard errors by substituting N_{eff} in the SE calculation:

$$SE = \frac{\sqrt{Var_w(x)}}{\sqrt{N_{eff}}}$$

Software choices

Software choices

There are several commercial products designed for analysis of complex surveys in particular:

- SUDAAN
- WesVar
- Several others that are specific to narrow range of analyses and/or datasets.

General-use statistics programs:

- R
- Stata
- SAS (inconsistent support)
- SPSS (with caveats)

SPSS

Beware `WEIGHT BY` command!

- Interpreted as frequency weights
- Most survey weights we use will be inverse probability weights
- Estimated standard errors will be wrong, almost always too small

SPSS Complex Samples

- Paid add-on to SPSS, like AMOS
- OSU SoC does get this
- Can be fairly opaque in terms of correct setup

For surveys with single weight, tell SPSS it is a one-stage design sampled with replacement. SPSS will save the setup to file.

Example SPSS setup

```
CSPLAN ANALYSIS
  /PLAN FILE='/path/to/file/ex.csaplan'
  /PLANVARS ANALYSISWEIGHT=WEIGHT3
  /SRSESTIMATOR TYPE=WOR
  /PRINT PLAN
  /DESIGN
  /ESTIMATOR TYPE=WR.
```

Run this (or just follow the menus under “Analyze” -> Complex Samples" -> “Prepare for Analysis. . . ”)

Available analyses in SPSS

- Descriptives (means, frequencies, crosstabs)
- OLS (well, WLS) regression
- Logistic regression
- Ordinal regression (logit, probit, cloglog, couple other links)
- Cox regression (simple survival analysis)

SPSS code snippet

Here's a preview of the SPSS code needed to run the model we'll do later:

```
CSGLM  cog_fun WITH engagement watch_tv use_facebook
surf_net age female hispanic black educ
  /PLAN FILE='/path/to/file/ex.csaplan'
  /MODEL engagement watch_tv use_facebook surf_net age
  female hispanic  black educ
  /INTERCEPT INCLUDE=YES SHOW=YES
  /STATISTICS PARAMETER SE CINTERVAL TTEST
  /PRINT SUMMARY
  /TEST TYPE=F PADJUST=LSD
  /MISSING CLASSMISSING=EXCLUDE
  /CRITERIA CILEVEL=95.
```

R

- `survey` package, available on CRAN
- Large range of analyses, designs, utilities
  - All models supported by base `lm` and `glm`, plus ordinal regression
    - Some others implemented like negative binomial in `sjstats` package
  - SEM via `lavaan.survey`
  - Descriptives
  - Multiple imputation
  - Weighting procedures via calibration, raking, post-stratification
- A few quirks that make some procedures different than base R equivalent

R setup

Read data into R...

```
library(haven)
raw_data <- read_dta("brain_health.dta")
```

Create a `survey.design` object:

```
library(survey)
survey_data <- svydesign(data = raw_data, ids = ~1,
                         weights = ~WEIGHT3)
```

In `survey`, instead of indexing variables with `data$variable` syntax, always use formula syntax (`~variable`).

`ids = ~1` means we assume independent sampling, if dependent we provide the grouping variables to `id =`.

Stata

Many built-in abilities in its `svy` suite of utilities (akin to `xt` suite for panel data, `st` for survival data).

- Capabilities mostly overlap with R `survey` package
- After setup, pretty simple to use (just begin command with `svy:`)
- Many commands initially developed by Senior Mathematical Statistician at Bureau of Labor Statistics (he is now Assistant Director for Research and Methodology).
- One type of analysis only available in Stata: Multilevel models with regression weights.
- One limitation: Uneven support for incorporating weights into plots

Stata setup

You may use dialogs to set things up.

```
use "/path/to/file/brain_health.dta"
svyset _n [pweight=WEIGHT3], vce(linearized)
```

- `pweight` is what we use for the typical probability weights in survey datasets.
- `vce(linearized)` tells Stata we want to use the conventional method for calculating variance
  - e.g. Huber-White robust estimators for regression

Example analyses

Means in our data

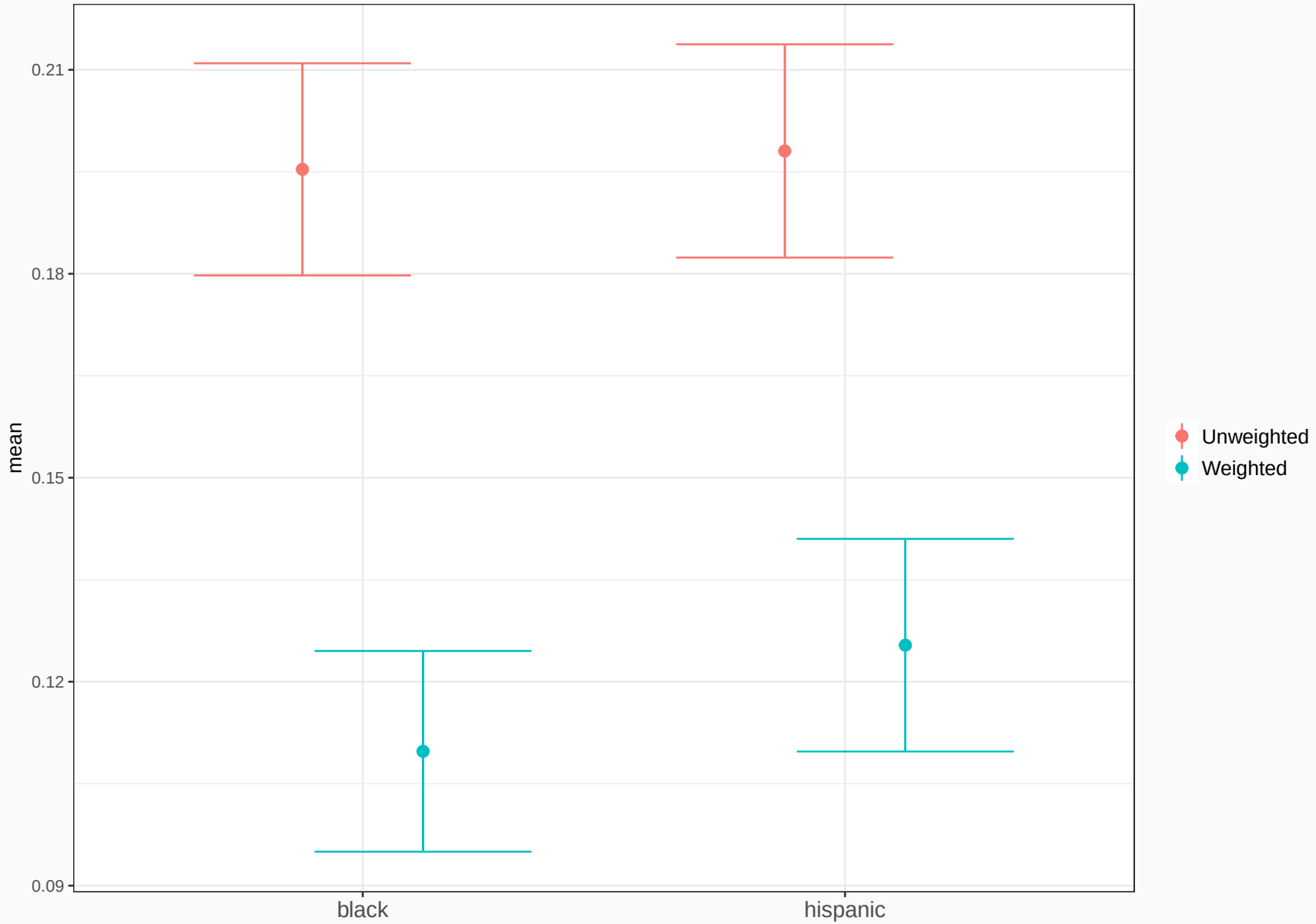
R:

```r
svymean(~cog_fun, design = survey_data, na.rm = TRUE)
```

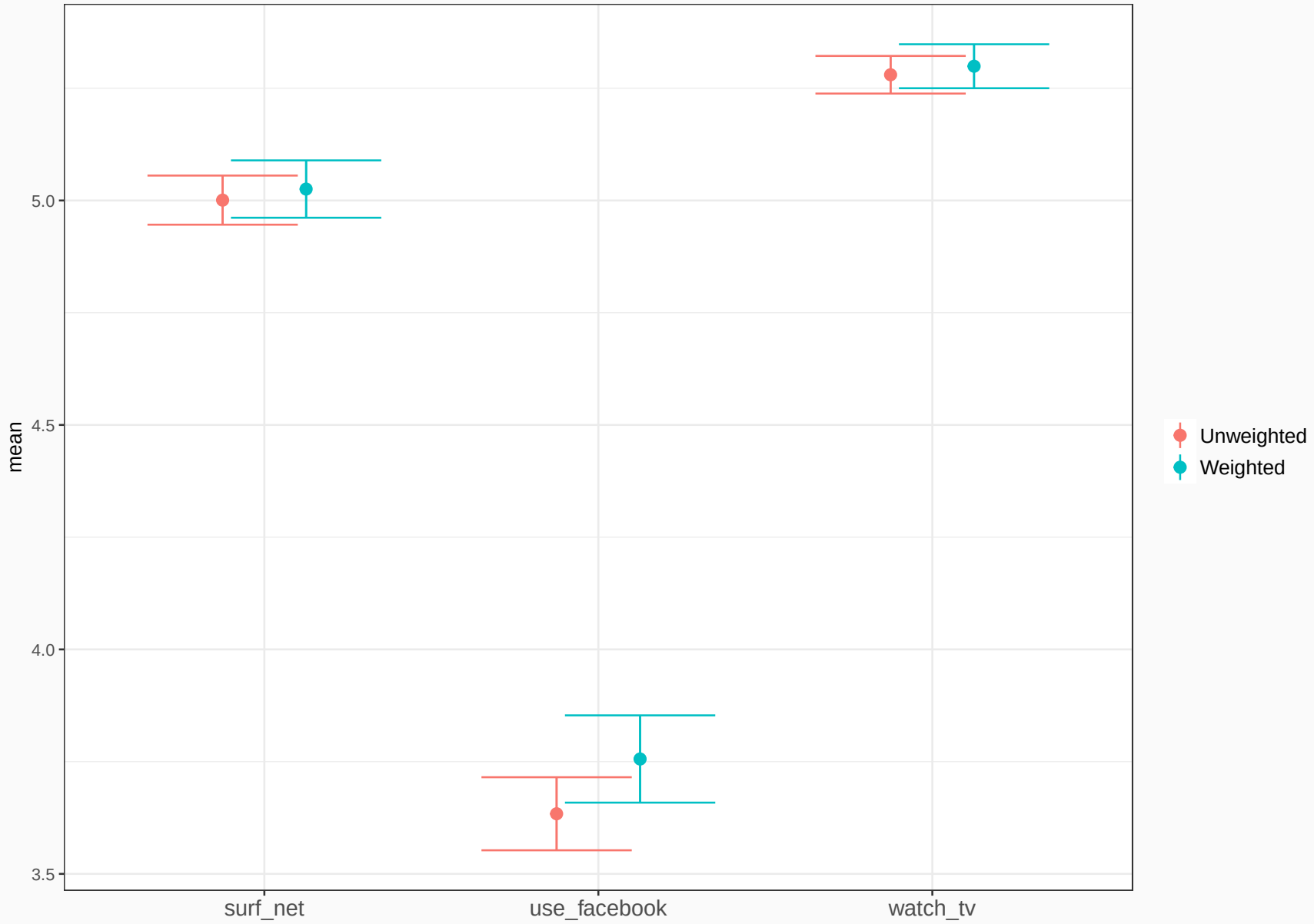
Stata:

```
svy: mean cog_fun engagement
```

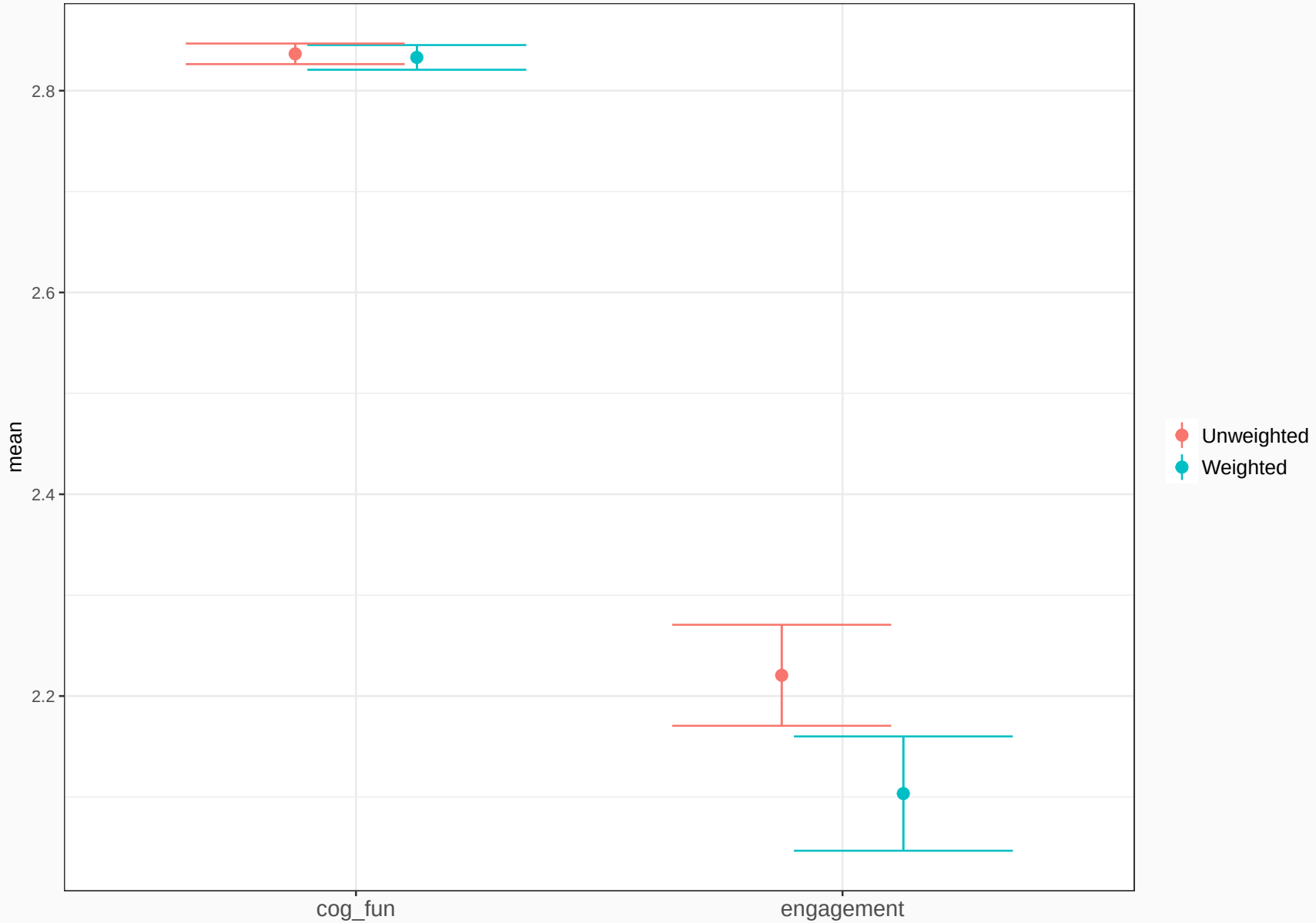
Means in our data



Means in our data



Means in our data



Regression analysis

Let's start by looking at unweighted estimates.

R:

```
lm(cog_fun ~ engagement + watch_tv + surf_net +
     use_facebook + age + female + educ + black +
     hispanic, data = raw_data)
```

Stata:

```
regress cog_fun engagement watch_tv surf_net ///
  use_facebook age female educ black hispanic
```

R Note

You will want to store the model in an object, like this:

```
model <- lm(y ~ x)
```

And use the the `summary` function or similar to inspect the results.

I'm omitting that code for simplicity in these slides.

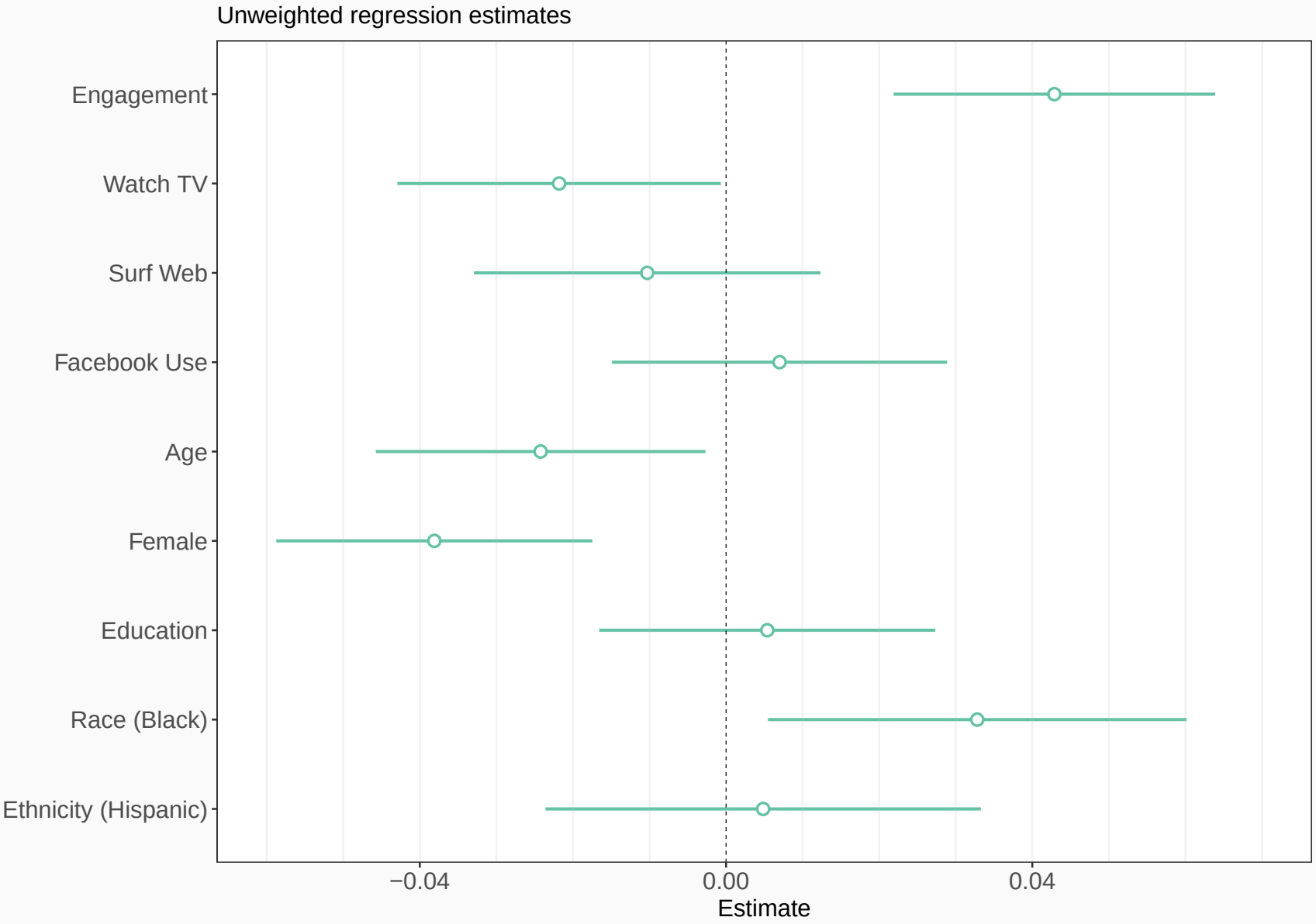
Regression analysis

	Unweighted
Engagement	0.04^{***}
	(0.01)
Watch TV	-0.02^{*}
	(0.01)
Surf Web	-0.01
	(0.01)
Facebook Use	0.01
	(0.01)
Age	-0.02^{*}
	(0.01)
R^2	0.02

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 1: Statistical models

Regression analysis



Survey-weighted regression

Now let's account for the weights and survey design:

R:

```
svyglm(cog_fun ~ engagement + watch_tv + surf_net +
         use_facebook + age + female + educ + black +
         hispanic, design = survey_data)
```

Stata:

```
svy: regress cog_fun engagement watch_tv surf_net ///
  use_facebook age female educ black hispanic
```

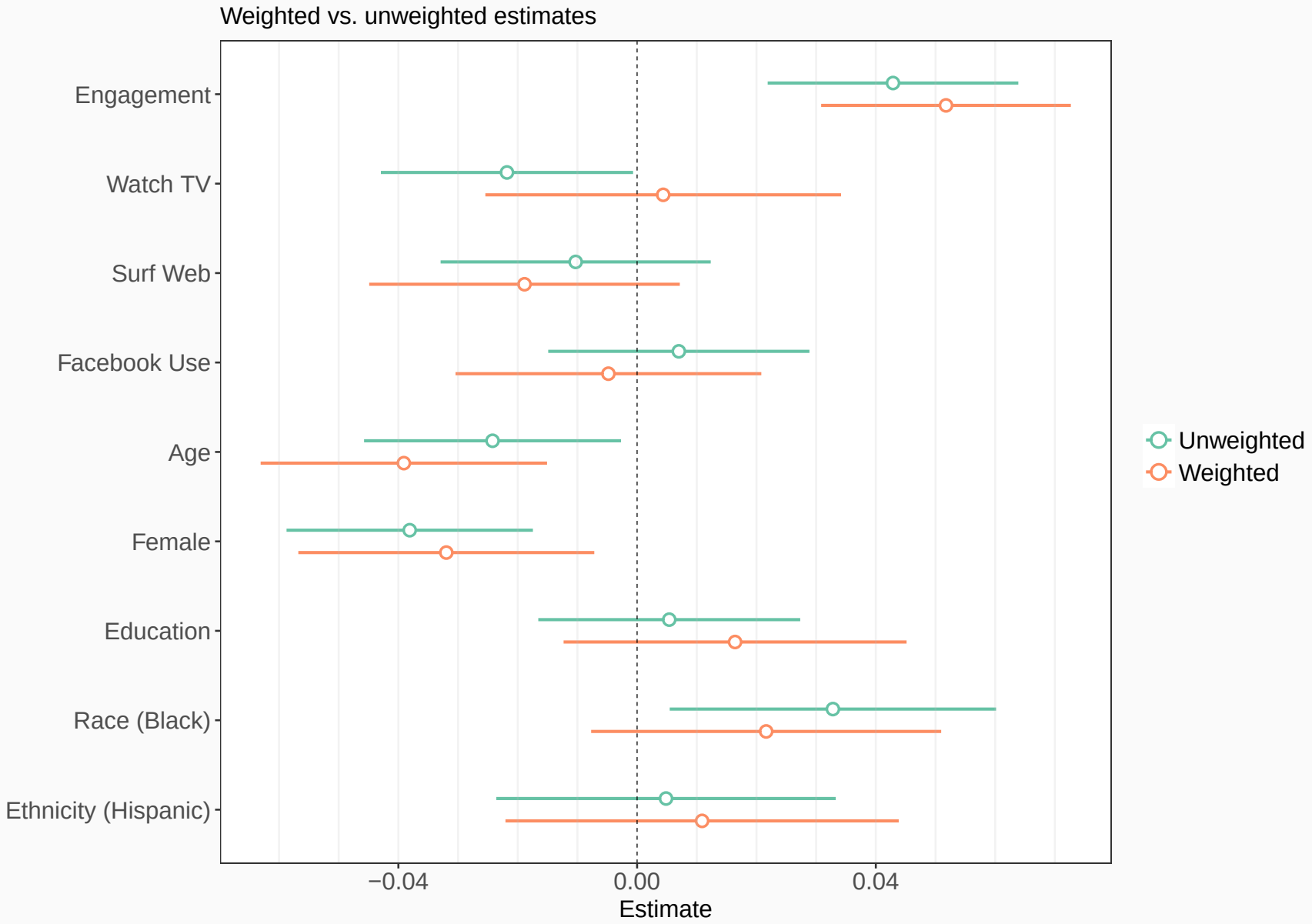
Survey-weighted regression

	Unweighted	Weighted
Engagement	0.04^{***}	0.05^{***}
	(0.01)	(0.01)
Watch TV	-0.02^{*}	0.00
	(0.01)	(0.02)
Surf Web	-0.01	-0.02
	(0.01)	(0.01)
Facebook Use	0.01	-0.00
	(0.01)	(0.01)
Age	-0.02^{*}	-0.04^{**}
	(0.01)	(0.01)
R^2	0.02	0.02

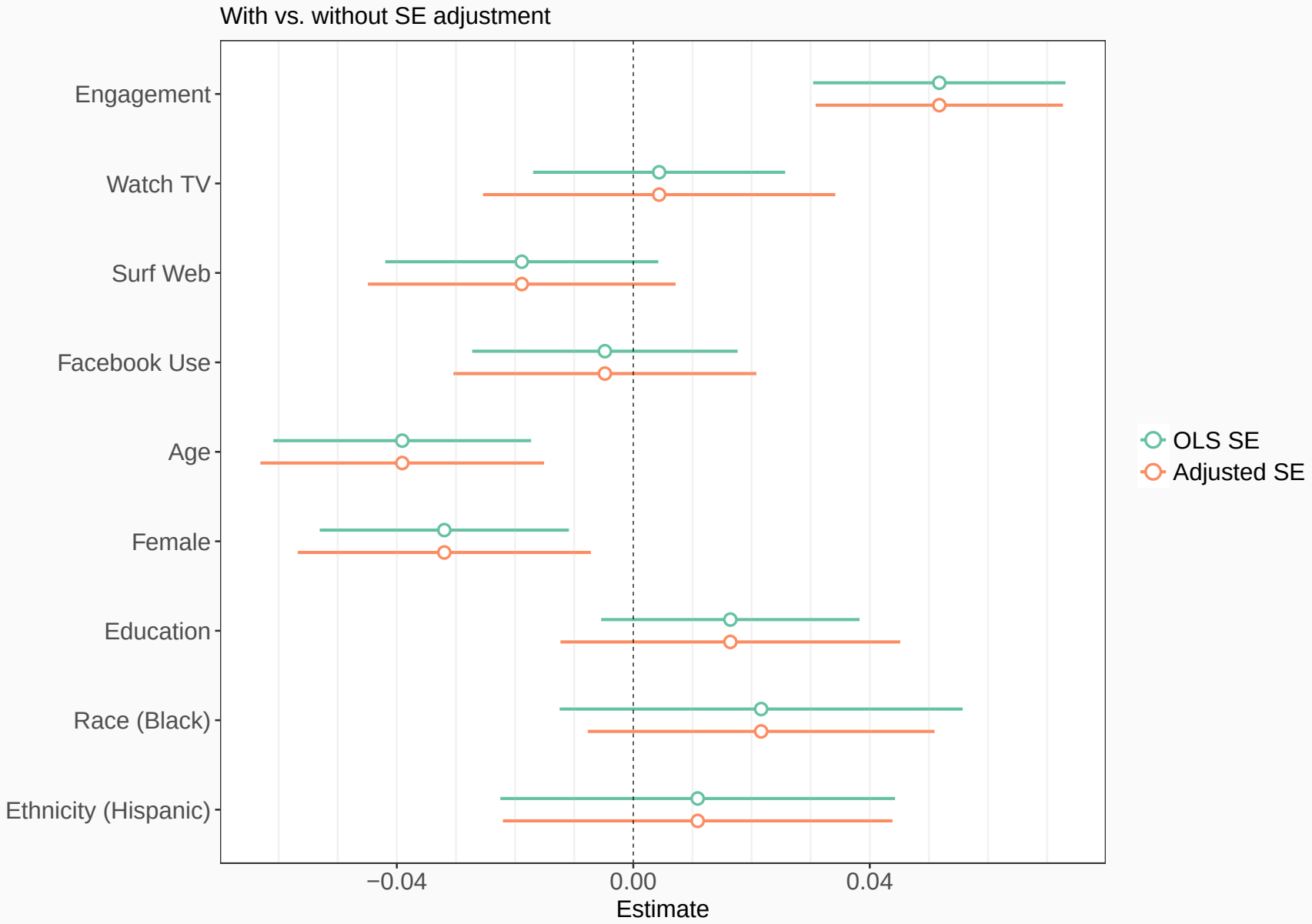
$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

Table 2: Statistical models

Survey-weighted regression



Effect of varaince adjustment



Testing ignorability of weights

Stata:

`wgttest`

- Test described in DuMouchel and Duncan (1983)
  - Also discussed in Bollen et al. (2016)
- Re-fits model with weights as predictor
  - Interaction term between weights and each predictor
- R^2-change test comparing original and re-fit models
- Looking at model coefficients can show you which predictors are affected
- Find at https://ideas.repec.org/c/boc/bocode/s444104.html

```
wgttest cog_fun engagement watch_tv surf_net ///
  use_facebook age female educ black hispanic,
  wgt(WEIGHT3)
```

Testing ignorability of weights

R:

`wgttest`

- R clone of Stata `wgttest` implemented in `jtools` package
- Find at https://cran.r-project.org/package=jtools

`pf_sv_test`

- Implements Pfefferman and Sverchkov (1993) bootstrapping procedure
- Testing correlation between model residuals and weights
  - Also squared and cubed residuals
- Also in `jtools` package

Testing ignorability of weights

```
library(jtools)
fit <- lm(cog_fun ~ engagement + watch_tv + surf_net +
            use_facebook + age + female + educ + black +
            hispanic, data = raw_data)

wgttest(fit, weights = WEIGHT3, data = raw_data)
## DuMouchel-Duncan test of model change with weights
##
## F(10,2447) = 2.206
## p = 0.015
##
## Lower p values indicate greater influence of the weights
```

Testing ignorability of weights

```
pf_sv_test(fit, weights = WEIGHT3, data = raw_data)
##
## Pfeffermann-Sverchkov test of sample weight ignorability
##
## Residual correlation = 0.01, p = 0.37
## Squared residual correlation = 0.02, p = 0.37
## Cubed residual correlation = -0.02, p = 0.38
##
## A significant correlation may indicate biased estimates
## in the unweighted model.
```