

Cross Validated is a question and answer site for people interested in statistics, machine learning, data analysis, data mining, and data visualization. It only takes a minute to sign up.

Anybody can ask a question



Anybody can answer

Sign up to join this community

The best answers are voted up and rise to the top



use of weights in svyglm vs glm

Asked 10 years, 1 month ago Modified 4 years, 3 months ago Viewed 12k times

- ▲ I would like to know how the treatment of weights differs between `svyglm` and `glm`
- 25 I am using the `twang` package in R to create propensity scores which are then used as weights, as follows (this code comes from the `twang` documentation):

library(twang)
library(survey)
set.seed(1)

data(lalonde)

ps.lalonde <- ps(treat ~ age + educ + black + hispan + nodegree + married + re74 +
re75,
data = lalonde)

lalonde\$w <- get.weights(ps.lalonde, stop.method="es.mean")
design.ps <- svydesign(ids=~1, weights=~w, data=lalonde)

glm1 <- svyglm(re78 ~ treat, design=design.ps)

summary(glm1)

...
Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 6685.2 374.4 17.853 <2e-16 ***
treat -432.4 753.0 -0.574 0.566

Compare this to:

```
glm11 <- glm(re78 ~ treat, weights=w , data=lalonde)
summary(glm11)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept) 6685.2       362.5 18.441 <2e-16 ***
treat       -432.4       586.1 -0.738 0.461
```

So the parameter estimates are the same but the standard errors for the treatment are quite different.

How does the treatment of weights differ between `svyglm` and `glm` ?

r survey

Share Cite Improve this question Follow

edited Apr 25, 2013 at 16:37

asked Apr 24, 2013 at 17:35



Joe King

3,384

8

37

60

2 Answers

Sorted by: Highest score (default)





14



There are lots of different sorts of weights and they get kind of confusing. You have to be pretty careful when you're using different functions or software that you're using the kind of weights you think you're using.

The `svyglm` function uses survey weights - these weight the importance of each case to make them representative (to each other, after `twang`). I'm not sure what weight does in `glm()` - I think they represent the accuracy of the measures. (If you're using the binomial family, they have different meaning).

The survey weights (in `surveyglm`) are the weights that you want, to give you the correct standard errors.

(There are also frequency weights, analytic weights, and importance weights).

Share Cite Improve this answer Follow

answered Apr 24, 2013 at 21:05



Jeremy Miles

15.6k 6 35 69

(+1) thank you. do you know an accessible reference for the survey weights, other than the docs for `surveyglm`) ? – **Joe King** Apr 24, 2013 at 21:52

2 I like Lumley's book: [amazon.com/Complex-Surveys-Analysis-Series-Methodology/dp/...](https://www.amazon.com/Complex-Surveys-Analysis-Series-Methodology/dp/...) – **Jeremy Miles** Apr 25, 2013 at 18:21

1 Thanks for the reference., By accessible I meant something available online, sorry. I don't have easy access to good libraries.... – **Joe King** Apr 28, 2013 at 18:23

Hmmm... I don't recall coming across anything, but I'll see what I can find. – **Jeremy Miles** Apr 29, 2013 at 2:27

1 This post by Thomas Lumley is exactly the thing: notstatschat.rbind.io/2020/08/04/weights-in-statistics – **bschneider** Jan 8 at 23:04



14



`survey` computes the standard errors with consideration of the loss of precision introduced by sampling weights. Weights in `glm` simply adjust the weight given to the errors in the least squares estimation, so the standard errors aren't correct. Here's a selection from Lumley (2010):

In a model-based analysis it would be necessary to specify the random part of the model correctly to get correct standard errors, but all our standard error estimates are design-based and so are valid regardless of the model. It is worth noting that the "sandwich", or "model-robust", or "heteroskedasticity-consistent" standard errors sometimes used in model-based regression analysis are almost identical to the design-based standard errors we will use; the main difference being in the handling of stratification.

So without strata in your design, you will likely find that using `sandwich` will get you identical or near-identical SE estimates.

```
library(sandwich)
coefs <- vcovHC(glm11, type="HC0")
lmtest::coefTest(glm11, coefs)
```

In my test, they didn't compute out exactly when using "HC0" or "HC1", but were very close. `svyglm` is now reporting a z-value instead of t-value as well.

Share Cite Improve this answer Follow

edited Feb 8, 2019 at 1:47



Mark White

9,734 4 29 73

answered Jul 27, 2016 at 17:01



commscho

535 4 10