ACADEMIC YEAR 2017/2018
Università degli Studi di Milano
GRADUATE SCHOOL IN SOCIAL AND POLITICAL SCIENCES
APPLIED MULTIVARIATE ANALYSIS
Luigi Curini
luigi.curini@unimi.it

Do not quote without author's permission

Margins command

margins is a postestimation command, a command for use after you have fit a model using an estimation command such as **regress**, or almost **any other estimation command**.

A margin is a statistic based on a fitted model in which some of or all the covariates are fixed at some given value defined by the researcher

margins estimates and reports **margins of responses** and margins of derivatives of responses, also known as **marginal effects** (does this ring you a bell?)

1. **Obtaining margins of responses: a simple case**

Let's use our usual dataset with Lijphart:

```
regress effparty45 numiss
```

Let's predict effparty45 when `numiss`=1 using the lincom command:

```
lincom _b[_cons] +_b[numiss]
```

Now let's use *margins*:

```
margins, at(numiss=1)
marginsplot

margins, at ( numiss=(0 1))
marginsplot
```

More complex examples:

```
reg  ecogr709  const45  federal45 judrev45
```

```
lincom _b[_cons] +_b[const45]*2+_b[federal45]*1+_b[judrev45]*3
margins, at(const45=2 federal45=1 judrev45=3)
marginsplot
```

Declaring a covariate as a dummy makes things faster! Let's use our usual dataset with SWD in Europe:

```
reg  demo_satisfaction  est_europa
margins, at ( est_europa=(0 1))
```

As an alternative method: note the **i.** in front of the dummy variable: "**i**" stands for categorical variable! This allows us to estimate *margins* more easily:

```
reg  demo_satisfaction  i.est_europa
margins, at ( est_europa=(0 1))
margins est_europa
```

Now let's test the following model:

```
reg  demo_satisfaction  qualityofinstitutions
margins qualityofinstitutions
```

You are not allowed to type `margins qualityofinstitutions`; doing that will produce an error. Why? Because `qualityofinstitutions` is a continuous variable, there are an infinite number of values at which it could evaluate the margins. At what value(s) should `qualityofinstitutions` be fixed? margins requires more guidance with continuous covariates. We can provide that guidance by using the `at()` option and typing as did earlier:

```
margins, at( qualityofinstitutions=1)
margins, at( qualityofinstitutions=(1 2 3))
margins, at( qualityofinstitutions=(1 (1) 3))
```

2. Testing margins

Continuing with the previous example, it would be interesting to test whether it is significant the difference between the expected value (margins) for different combination of our IVs. To do that, you make a test of equality of *margins*.

```
regress effparty45 numiss
lincom ((_b[_cons] +_b[numiss]*3)-(_b[_cons] +_b[numiss]*1) )
margins, at(numiss=(1 3)) contrast(atcontrast(r._at) wald)
margins, at(numiss=(1 3)) contrast(atjoint)
marginsplot
```

```
lincom (_b[_cons] +_b[const45]*2+_b[federal45]*3+_b[judrev45]*3)-(_b[_cons] +_b[const45]*2+_b[federal45]*1+_b[judrev45]*3)

margins, at(const45=2 federal45=1 judrev45=3) at(const45=2 federal45=3 judrev45=3) contrast(atcontrast(r._at) wald)
```

or more simply (given that `federal45` is the only variable that changes here…):

```
margins, at(federal45=(1 3)) contrast(atcontrast(r._at) wald)
marginsplot
```

On the contrary if we have two variables that change:

```
lincom (_b[_cons] +_b[const45]*3+_b[federal45]*3+_b[judrev45]*3)-(_b[_cons] +_b[const45]*2+_b[federal45]*1+_b[judrev45]*3)

margins, at(const45=2 federal45=1) at(const45=3 federal45=3) contrast(atcontrast(r._at) wald)
marginsplot
```

3. Example with a quadratic term

Obtaining margins of derivatives of responses (a.k.a. marginal effects) are crucial for better understanding quadratic and interaction non-linear models.

Let's go back to the example on California school.

```
gen avginc2 = avginc^2
reg testscr  avginc avginc2 computer

lincom (_b[_cons]+ _b[avginc]*11 + _b[avginc2]*(11*11) + _b[computer]*300)-(_b[_cons]+ _b[avginc]*10 + _b[avginc2]*(10*10)+ _b[computer]*300)

margins, at(avginc=10 avginc2=100 computer=300) at(avginc=11 avginc2=121 computer=300) contrast(atcontrast(r._at) wald)
```

However, to fully exploit the *margins* command, we could write:

```
reg testscr  c.avginc##c.avginc
```

The **c.** operator tells Stata that `avginc` is to be treated as a continuous variable. On the other side, the ## operator is a shortcut notation for two operations. The first # tells Stata that this term is an interaction; the second # tells Stata to include the associated variable in addition to their interaction (remember: you have always to include in your equation all the constitutive terms of your interaction!). If you write:

```
reg testscr  c.avginc#c.avginc
```

It does not include the constitutive terms but just the interaction!

This is where *margins* becomes (really) useful. In three different ways.

First: now Stata knows that you are employing in the model `avginc` and its interaction (i.e., `avginc`^{2}), so there is no need anymore to specify at which value you want to fix `avginc` and `avginc2` as you were doing earlier. It is enough to fix the value of `avginc`. Stata automatically will also upload the value of `avginc`^{2}. And indeed compare:

```
reg testscr c.avginc##c.avginc
margins,    at(avginc==10) at(avginc=11) contrast(atcontrast(r._at)
wald)
```

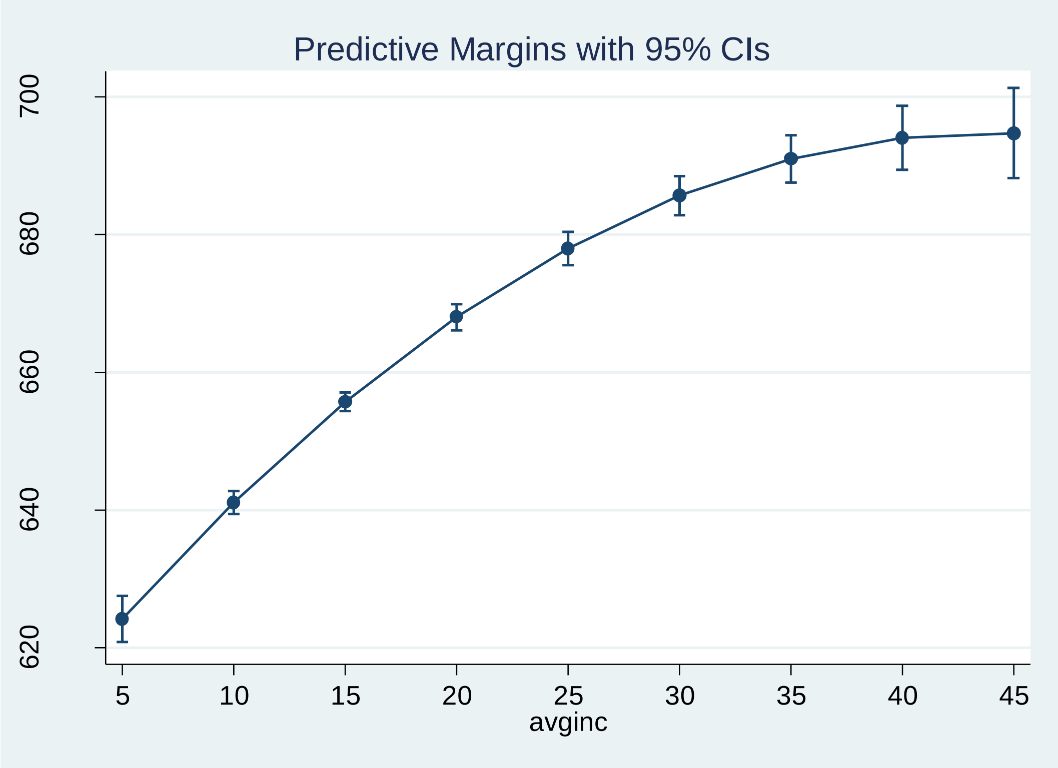
with the previous *margins* you have ran (i.e., `margins,   at(avginc=10 avginc2=100) at(avginc=11 avginc2=121) contrast(atcontrast(r._at) wald)`). You get exactly the same result (but without the extra effort to specify at which value you want to fix `avginc2`).

Second: we can use *margins* to estimate the expected value of `testscr` at various values of `avginc`. Because `avginc` ranges from (around) 5 to 45 let's predict `testscr` for each 5 points increase in this range starting from 5 up to 55. For each specified value, *margins* will call predict to generate a variable with the linear prediction and take the average of the prediction to get the predictive margin.

```
margins,   at(avginc=(5 (5) 45))
margins,   at(avginc=(5 (5) 45))vsquish
```

Using the vsquish option suppresses the extra vertical space in the legend for the `at()` option.
Now we add also the post option and visually see the non-linear relationship between DV and IV:

```
marginsplot
```



4. Marginal effects

Third: if you run the following model:

```
reg testscr c.avginc##c.avginc
```

the coefficient on `avginc` is not simple to understand: if you increase `avginc` by one unit this increases both `avginc` and `avginc` squared, and the total effect depends on what the value of `avginc` was to begin with.

With the `dydx()` option, *`margins`* calculates the derivative of the mean expected outcome with respect to the variable you specify. In this case we do not want to estimate the expected value of `testscr` at different value of `avginc`, but on the contrary the expected impact of `avginc` on `testscr` as `avginc` change by one unit!

More on details, the expected value of `testscr` is going to be equal to:

$$\widehat{\text{testscr}} = \text{a} + b_1 avging + b_2 avging^2$$

So, for example, when `avginc` is equals to 5, the expected value of `testscr` is going to be equal to: 607.3+3.85*5-0.04*5*5=625.

The marginal impact, on the contrary, is just the first derivative of the previous equation, that is:

$$\frac{\Delta \text{testscr}}{\Delta \text{avging}} = b_1 + 2 * b_2 * avging$$

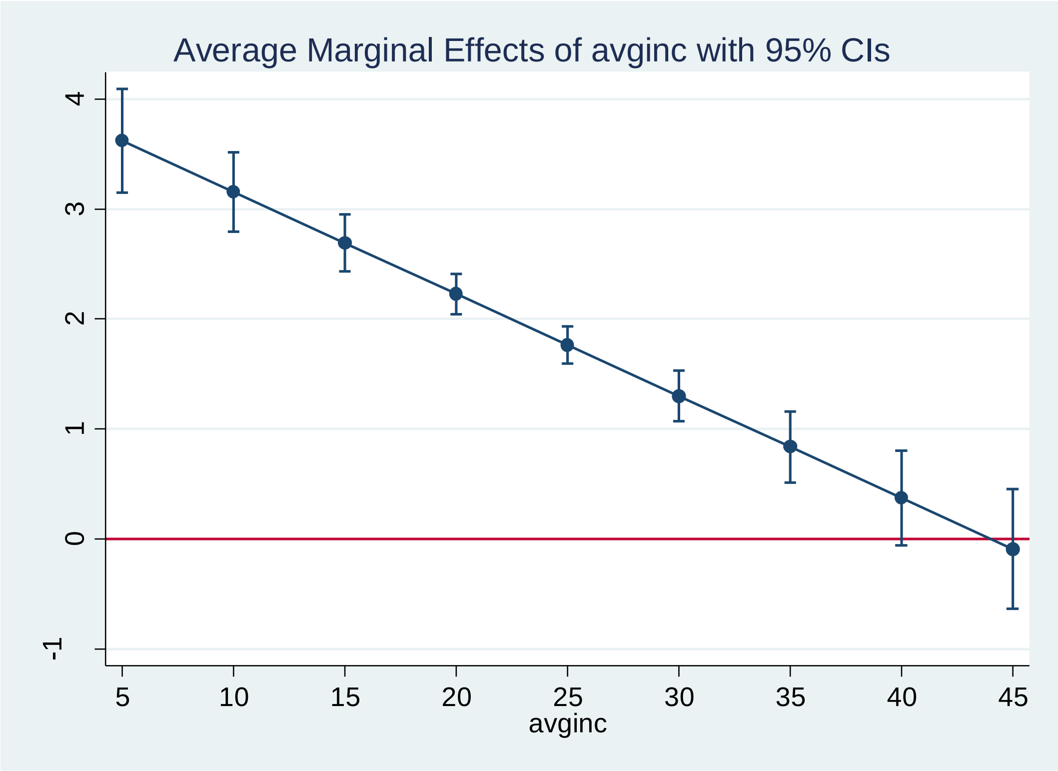
By using `margins`, it is easy to estimate the marginal impact of a unit increase of x on y:

```
margins, dydx(avginc) at(avginc=(5.3 (5) 55.3)) vsquish
```

Now we can see that a (given/one unit) change of `avginc` has a different impact on our DV according to where this unit change happens! Moreover not always the marginal impact of `avginc` is significant!

Let's graph the relationship!

```
margins, dydx(avginc) at(avginc=(5.3(5) 55.3)) vsquish post
marginsplot
marginsplot, yline(0)
```



As you can see now, the marginal impact of `avginc` is decreasing, becoming insignificant for values of `avginc` higher than 35.

Addendum:

The margins command estimates the marginal impact of one unit change in your IV. What if I want to estimate the marginal impact of .3 change in your IV? You need to use other systems (i.e., simulation). Take a look here:

https://files.nyu.edu/mrg217/public/interaction.html

If we will have time, we will discuss about that later!

Now imagine that you run this model:

```
reg testscr c.avginc##c.avginc computer
```

In this case the marginal impact of computer is going to be:

$$\frac{\Delta \text{testscr}}{\Delta \text{computer}} = b_3$$

That is, the marginal impact is fixed and equals to the coefficient of `computer`. And indeed, if you estimate the marginal impact of `computer` with margins:

```
margins, dydx(computer) at(computer=(0 (500) 3500))
```

gives you always -.0074354, i.e., the coefficient of `computer`, no matter at which value of `computer` we make it change by one unit. This is obvious, given the linear relationship between `computer` and `testscr`.

5. Interaction reprise

Back to our example on NES2004:

```
recode polknow3 (0=0 "Low pol. knowledge") (1/2=1 "Medium-high pol. knowledge"), gen (poldummy)

gen interact = progovmnt * poldummy

reg dem_therm progovmnt poldummy interact

lincom (_b[_cons]+ _b[progovmnt ]*2+_b[poldummy]*1+_b[interact]*2)
margins,  at(progovmnt==2 poldummy==1 interact==2)
```

However, to fully exploit *margins*, we should once again write:

```
reg dem_therm c.progovmnt##i.poldummy
margins, at(progovmnt==2 poldummy==1)
```

Notice that now you do not have to identify the value at which fixing the interaction term between `progovmnt` and `poldummy`, given that now Stata knows that you are running a model with the two variables interacting among themselves.

How to estimate the marginal impact of increasing `progovmnt` by 1 unit as `poldummy` changes from 0 to 1,

i.e.,

$$\frac{\Delta \text{dem\_therm}}{\Delta \text{progovmnt}} = b_1 + b_3 * poldummy$$

?

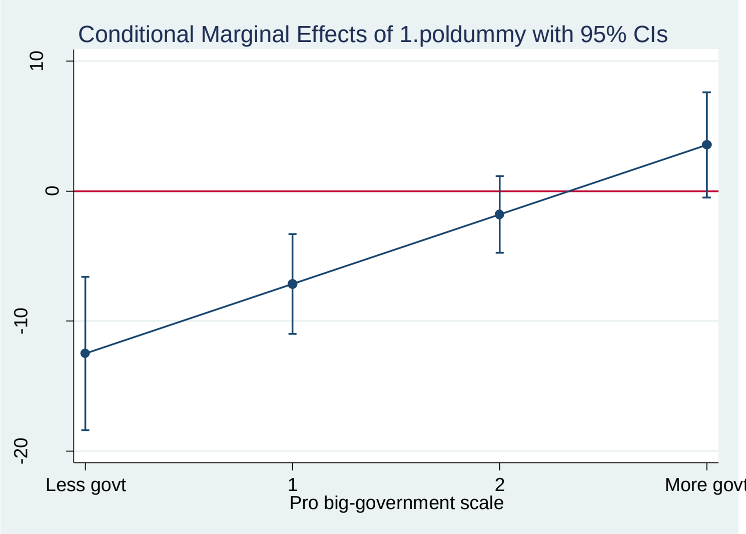
```
margins, dydx(progovmnt) at(poldummy=(0 1))
marginsplot
```

How to estimate the marginal impact of increasing `poldummy` by 1 unit as `progovmnt` changes from 0 to 3, i.e.,

$$\frac{\Delta \text{dem\_therm}}{\Delta \text{poldummy}} = b_2 + b_3 * progovmnt$$

?

```
margins, dydx(poldummy) at(progovmnt=(0 (1) 3))
marginsplot, yline(0)
```



As you can see, the marginal impact of `poldummy` is not significant for values of `progovmnt` higher than 1.