Using Predictive Margins to Make Clearer Explanations

Bill Rising

StataCorp LP

Indian Stata Users Group Meeting
1 August 2013

Goals

- This will be an interactive demonstration
- Looking at estimation in particular
- Looking at nice ways to make good images of models
- It would be nice to do this as a narrative of data analysis

Available Datasets

- Stata has many datasets available to play with
- They are here: **File > Example Datasets...**
  - `Example datasets installed with Stata` are installed
  - `Stata 13 manual datasets` come from Stata's webserver
- The dataset we would like is the low birthweight dataset from Hosmer and Lemeshow's book on logistic regression
  - Click the `[R]` manual, then search for `logisitic`
  - Click the link `lbw.dta`

```
. webuse lbw
```

A Quick Peek at the Dataset

- We can look at the data as a table

  `. edit`

- We can take look at the contents of the dataset

  `. codebook`

- We can get summary statistics of the data

  `. summarize`

Summaries for Categories

- We can make a oneway table of mean birthweights for each smoking status
  tabulate smoke, sum(bwt)
  - It appears that smoking status
- If we would like many tables at once, we can use the `tab1` command

```
. tab1 race smoke ui, sum(bwt)
```

- Now, to look for possible interactions, we can make twoway tables

```
. tab2 race smoke ui, sum(bwt)
```

Looking at Correlations

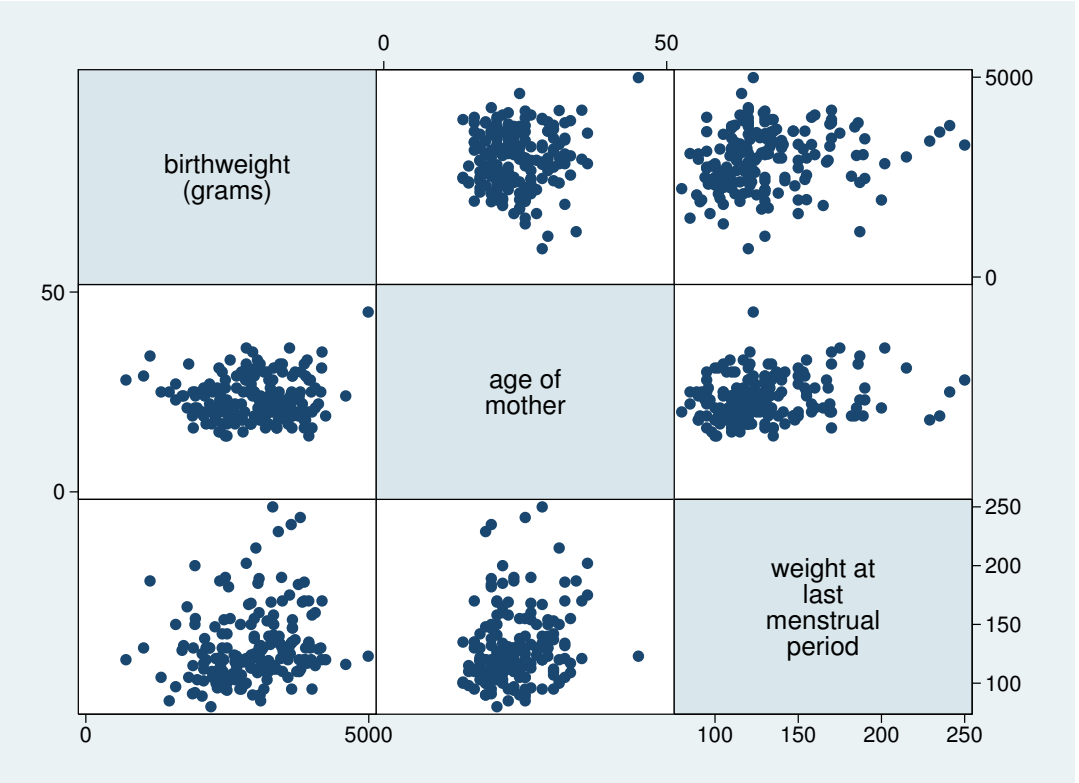
- If we would like to look at correlations, we can

```
. pwcorr bwt age lwt, sig
```

Graphing Correlations

- It is nicer to draw a matrix of scatterplots, though

```
. graph matrix bwt age lwt
```



Fitting Something Simple

- Here is a simple model (simpler than it should be)

```
. regress bwt lwt ui smoke
```

- The 0-1 variables `ui` and `smoke` have been included just like the continuous variable `lwt`
  - This is OK for the coeffiecients, but has some drawbacks for more complex models, as we will see

Working with Categorical Variables

- We would now like to include `race` in the model
  - It cannot simply be added to the list of covariates, because it has 3 categories
- To include a categorical variable, put an `i.` in front of its name—this declares the variable to be a categorical variable, or in Stataese, a *factor variable*
- Example:

```
. regress bwt lwt i.ui i.smoke i.race
```

- If we wanted "black" as the reference class, we could do that, too:

```
. regress bwt lwt i.ui i.smoke b2.race
```

Aside: Including Reference Classes

- By default, Stata does not show the base reference class in the regression table
- To change this behavior, either
  - Add the `base` option to each estimation command
  - Type `set showbaselevels on` to show the base levels for the rest of the current session
  - Type `set showbaselevels on, perm` to show the base levels for the rest of the current session
- Let's turn the base levels on forever

```
. set showbaselevels on, perm
```

Adding Interactions and Quadratics

- We can build the model up by including the interaction of `race` and `smoke`, which looked important in the tables

```
. regress bwt lwt i.ui i.smoke##i.race
```

  - The `##` notation is for an interaction, including both main and interaction effects—it replaces the `*` notation in textbooks
  - The `i.` was not needed for this interaction, because Stata assumes interactions are between categorical variables, by default

- Let's now add a quadratic in age to the model
  - This seems odd, but is surprising in this dataset
- To add the quadratic, we can interact age with itself

```
. regress bwt c.age##c.age lwt i.ui i.smoke##i.race
```

What Have We Here?

- The end result is complex
- We can easily interpret the coefficient for `lwt`
  - All other things being equal, mothers who weigh 1 pound more have babies which are about 3.0 grams heavier, on average
- We cannot easily interpret the coefficients for age, smoking status, and race, though
- Hence, the coefficient table is not particularly useful by itself
  - What is the shape of the parabola for age?
  - What is the effect of smoking within each race category?
- We could do arithmetic to answer these questions, but it would be nicer to see the answers directly
- This will be done using *predictive margins*

Asking a Different Question

- Suppose, instead of being asked to interpret the coefficients for age, we were asked to see the role of age in the model
- We could phrase this as

  > "What would we guess the mean weights of the babies would be as the age of the mothers range from 15 to 45?"

- As it stands, this question is somewhat hard to answer

Simplifying the Question

- Why not start by asking about one age:

  "What is a good guess at the mean weight of babies whose mothers are 25?"

- There are two paths we can take here:
  - We could plug in 25 for the age for all the women in our sample, leave all other covariates the same, predict the birthweights, and average the results
    - Order: predict with partial info, then average
  - We could average all the other covariates, set age to 25, and predict
    - Order: average, then predict with partial info
- The first path is called 'predictive margins' or 'average predictive margins'—it is the one we will take

A Predictive Margin

- Stata implements predictive margins using the `margins` command
- Here is what we could use as our best guess of the mean weight of babies of women aged 25:

```
. margins, at(age=25)
```

- Computationally, the point estimate could be computed by
  - Changing age to 25 everywhere
  - Using `predict` to get predicted values
  - Finding the mean of those values
- `margins` does more—it computes standard errors of the means
  - These are not standard errors of prediction for an individual

Margins Across Multiple Values

- Now we can look at the role age plays in our model by looking at a range of values
- We just need to give a *numlist* to specify the ages

```
. margins, at(age=(15(5)45))
```

- Good news: we see the weights drop and then rise
- Bad news: the notation is factor-variable like, so we need to look up the levels to get the values for age

Picturing Predictive Margins

- After running the margins command, we can draw a picture of its results using the `marginsplot` command
- Here is a very simple example

```
. marginsplot
```

  - This gives a good view of the parabolic shape
  - The wide confidence intervals for the older ages show some uncertainty

What About Comparing Groups?

- Suppose we would like to compare weights across the three race categories
- This is done by including `race` in the `varlist` for the `margins` command:

```
. margins race
```

  - We can do this because we specified `i.race` in the model
  - Aside: this is equivalent to the following

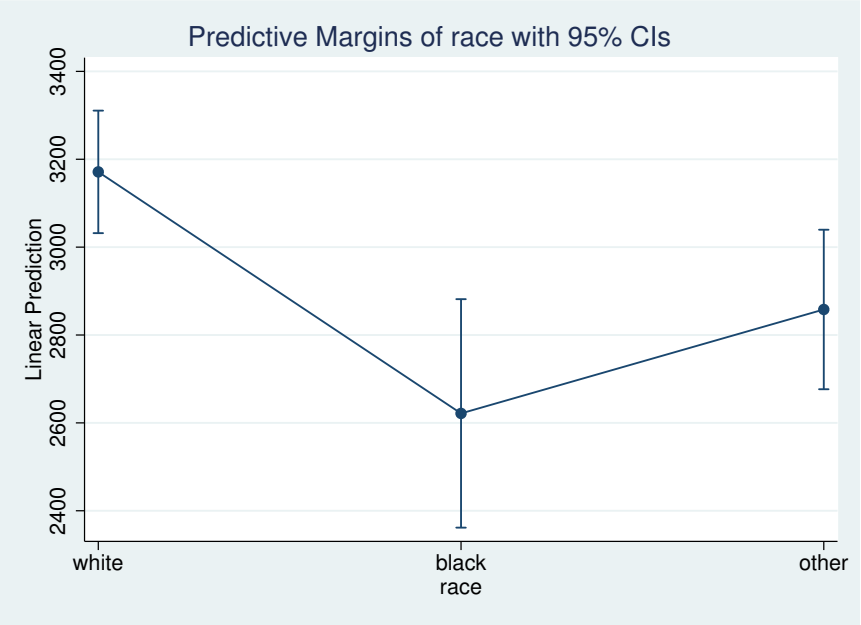
```
. margins, at(race=(1/3))
```

- These are simpler to look at as a table...

Graphing the Group Means

- ...but we can graph still graph them these using `marginsplot`

```
. marginsplot
```



- The graph is a little odd to see

More Complex Margins

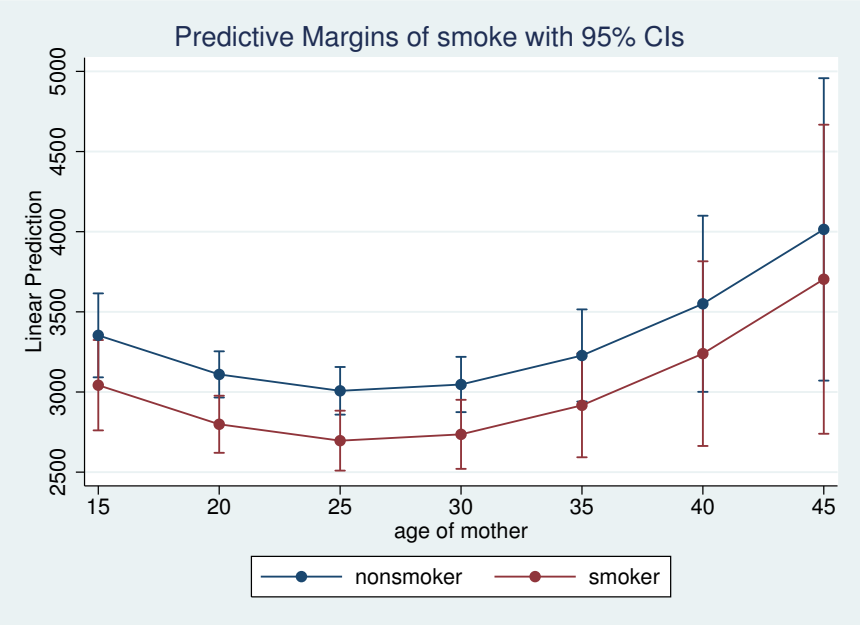
- There is no reason for us to limit our predictive margins to be computed over just one variable
- We could just as well look to see how age and smoking status work together
- Here is the the margins command

```
. margins smoke, at(age==(15(5)45))
```

Still a Simple Graph

- The `marginsplot` still makes a simple graph

```
. marginsplot
```



- The confidence intervals overlap a bit

Driving the Point Home

- If we wanted to specify smoking and age, we could use the interaction notation in the `margins` command
  - This is true even though there were no interactions—all that is done is that all possible combinations of smoking status and race are included
- The command is not bad...

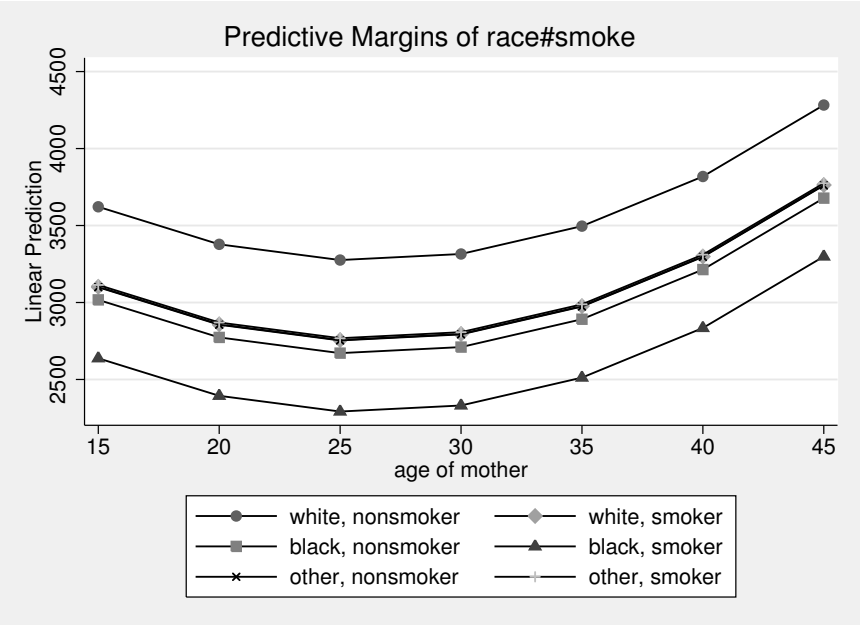
```
. margins race#smoke, at(age==(15(5)45))
```

- ... but the output is
- So...from now on, the output from the `margins` commands will not be shown in the handouts

We Can Still Visualize This

- We can still make a picture

```
. marginsplot, noci scheme(s2mono)
```

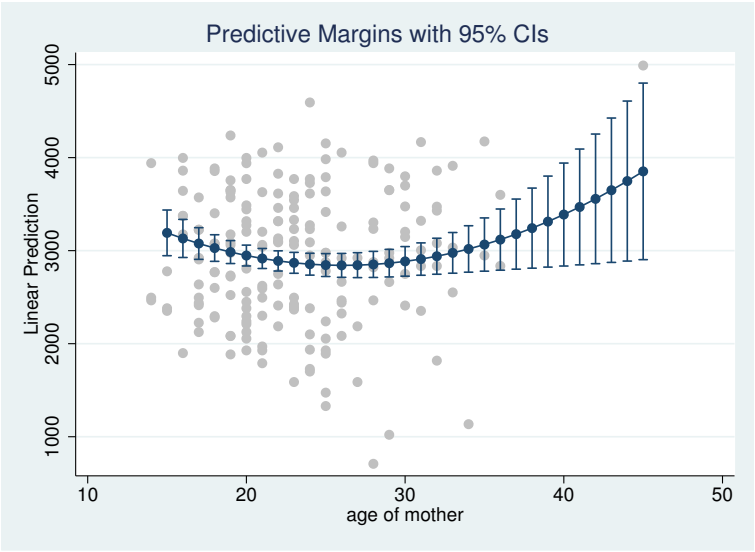


- The `s2mono` makes the overlapping points easier to see
- The `noci` option squelches the confidence intervals to make a better graph

A Fancy Overlay

- In this particular dataset, there is an outlier: a combination of a woman who is much older with a baby which is much heavier
- We can make a graph which shows the effect of age together with a scatterplot

```
. quietly margins, at(age==(15(1)45))
. marginsplot, legend(off) ///
      addplot(scatter bwt age, mcolor(gs12) below)
```



A Richer Dataset

- Now will switch over to the `nhanes2` dataset

  ```
  . webuse nhanes2
  ```

- This is a richer dataset from the National Health and Nutrition !! survey
- These data come from a data with a complex sampling design
- We can see that the sampling design has been saved with the dataset

  ```
  . svyset
  ```

- This will not make things much more complex: we will just need to put the `svy:` prefix in front of our estimation commands

Looking at Diabetes

- Let's look at the chances of having diabetes
- Here is a simple model

```
. svy: logistic diabetes age i.sex i.race bmi
```

- We can see that age and bmi both increase the odds of diabetes by about 6% for each unit increase
- Of course, this says nothing about how the probabilities of having diabetes change

A Better Explanation

- If we look at average predictive margins, we can see the roles of age and race more clearly
- Here is our `margins` command

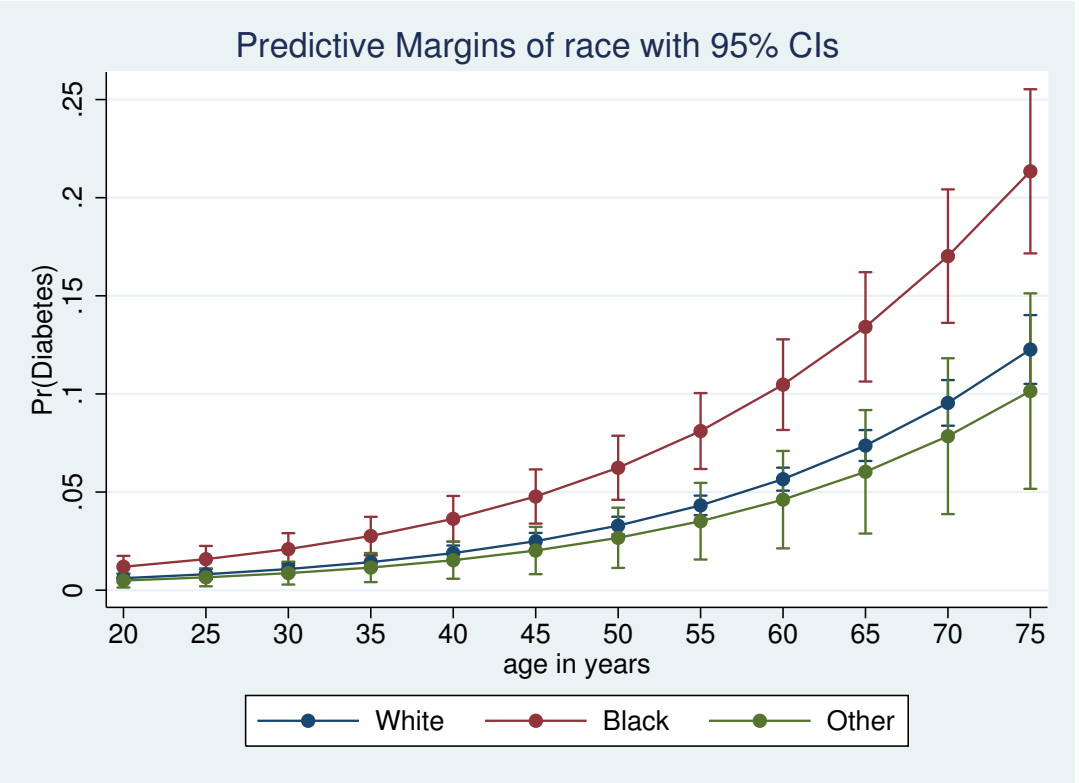
```
. margins race, at(age==(20(5)75)) vce(uncond)
```

  - The `vce(uncond)` option should be used to get the proper standard errors when using survey data

Here Is the Picture

- We can get a nice picture

```
. marginsplot, legend(rows(1))
```



This is Better than Odds Ratios

- This type of graph is something that makes explaining a logistic model much easier than via odds ratios
- It is as applicable to the general population as much as your belief that your sample is representative of the general population
  - Which is important for the odds ratios also
- Here, a picture is worth a thousand hard words

For Probit Fans

- If you prefer probit models, we can use the same type of logic

```
. svy: probit diabetes age i.sex i.race bmi
```

  - Now the coefficients are at all interpretable
- We can still get margins

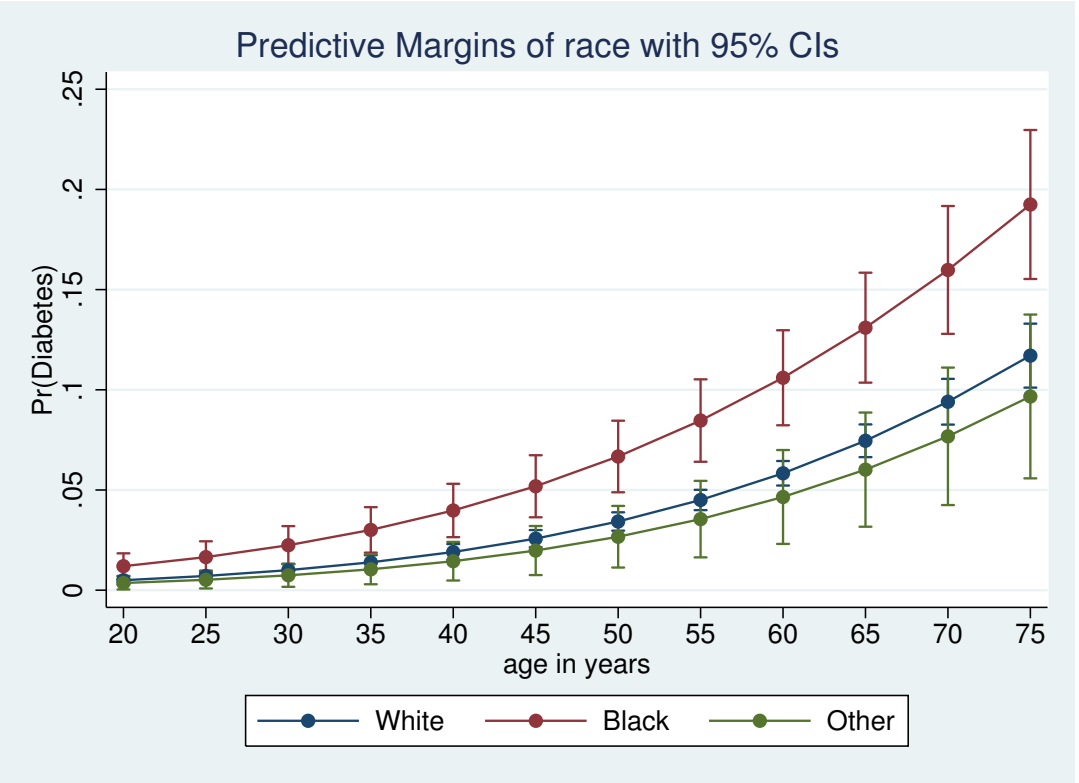
```
. margins race, at(age==(20(5)75)) vce(uncond)
```

- Creating the predictive margins still works the same

Picturing a Probit

- We can still get a very similar nice picture

```
. marginsplot, legend(rows(1))
```



Interactions

- Here is a model with interactions

```
. svy: logit diabetes c.age##c.age##race bmi i.sex
```

- If we look at the output, the higher-level interactions are needed in the model
- They are nearly impossible to picture or to talk about, however

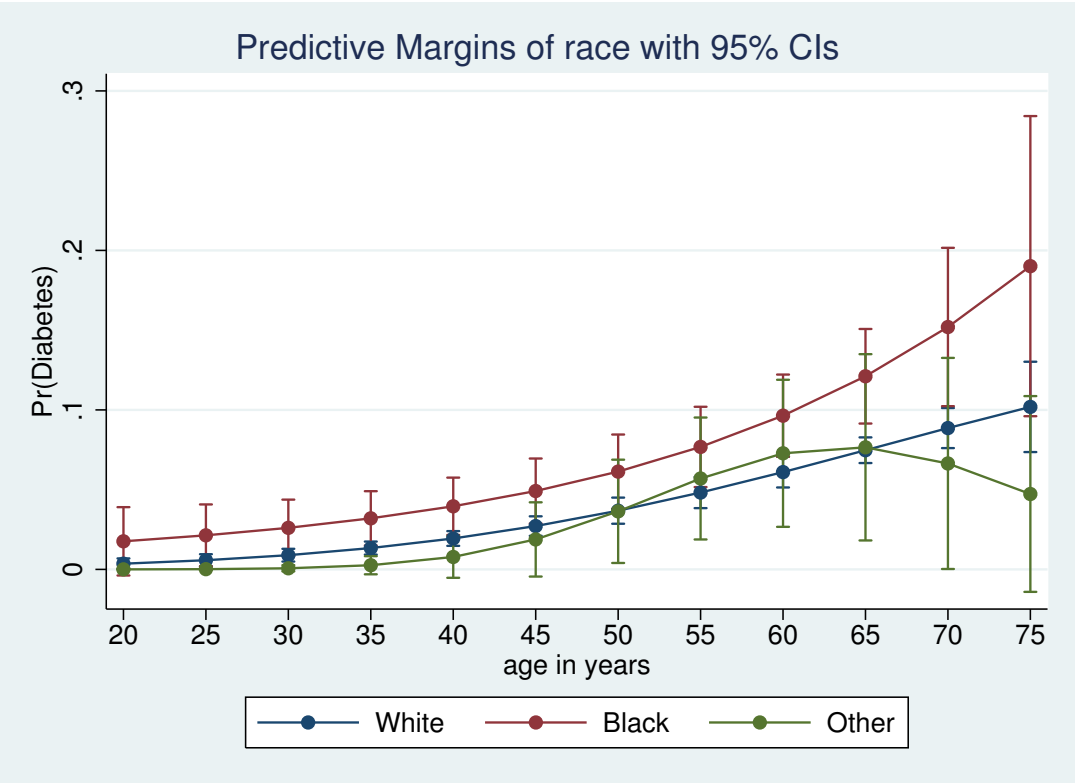
Visualizing Interactions with `marginsplot`

- Here are the margins for this complex model

```
. margins race, at(age==(20(5)75)) vce(uncond)
```

- And a nice, informative picture

```
. marginsplot, legend(rows(1))
```



Bothered by Counterfactuals

- You might have been bothered by the idea in the preceding examples that we used

  ```
  . margins race, ...
  ```

- This sets every observtion to each race category while computing the predictive margins
- If you would rather compute the predictive margins within each race, use the `over` option

  ```
  . margins, at(age==(20(5)75)) vce(uncond) over(race)
  ```

- The differences from before are small, and the picture is similar (not shown in handouts)

  ```
  . marginsplot, legend(rows(1))
  ```

Conclusion

- Predictive margins are wonderful for being able to explain models
- This is a help whenever the natural metric is different than the model metric
- This is even better for models with interaction terms

A Fun Plot

- Just For Fun
  - Believe it or not, it is possible to make a contour plot of predictive margins

    ```
    . do margcon
    ```
  - Here is the picture