Index of Lecture 3a (part)

Computing Predictions in Linear Models

We distinguish between two types of predictions (or purposes):

i) for individual observations: "real" prediction,

ii) for purposely selected combinations of predictor values: "illustrative" prediction.

All software packages for linear models offer predictions of type *i)*, directly for observed predictor patterns and for new predictor patterns:

- Stata/SAS: add extra observations to data with missing outcome,
- Minitab/R: specify new observations in separate columns/ dataset,
- Stata/SAS: special commands (`lincom`/`estimate`) give estimates for linear combinations of regression coefficients.

Fully specified predictions of type *ii)*: may be done using methods for type *i)* (perhaps tedious).

Some software offer both fully and partially specified predictions of type *ii)*:

- Stata: the `margins` command,
- Minitab/SAS: least squares means[1]; i.e., all predictors not included in prediction are set at their average value.

[1] "Least squares means" originate from experimental designed studies/data where factors are often balanced by design.

Stata: the `MARGINS` Command

- very flexible command (from version 12) with a wide range of options and setups;
  so flexible that caution is needed to not use it wrongly. . . ,
- strongly recommended to always check your predictions with simpler methods (in a few examples),
- linkage to the `marginsplot` command allows easy plotting of predicted values,
- mainly intended for "illustrative" predictions, and uses predictions from the `predict` command behind the scenes to come up with the requested predictions,
- the online help is pretty confusing ⇒ recommended to work from well-established examples, and to avoid use of numerous extra "fancy" options.

Coverage in course: worked examples (from simple to more complex) illustrating the basic features of command:[2]

- 1-predictor settings (categorical and continuous),
- 2-predictor settings, and the questions arising from omitting a predictor from a prediction,
- VER 14.12 worked example,
- plots and transformations as needed.

[2] Presentation by Sithar Dorjee in Fall 2013 offers details and references.

Simplest Examples: 1 Predictor

(1a) categorical: simple means, with model-based SE,

```
regress wpc i.herd
margins herd
lincom _cons+2.herd
```

(1b) continuous: predictions at specified set of values, with subsequent plot by `marginsplot`:

```
regress wpc milk120
margins , at( milk120=(1200(1000)5600) )
marginsplot
lincom _cons+milk120*3200
```

note: flexible format of "atspec"; e.g., 1200(1000)5600 = 1200, 2200, . . . , 5200, but list can also include statistics (e.g., mean and percentiles) and the special names "asobserved" and "asbalanced",

(1c) continuous with quadratic effect: predictions as above, but need to use factor notation,

```
regress wpc c.milk120##c.milk120
margins , at( milk120=(1200(1000)5600) )
marginsplot
```

(1d) backtransformation from transformed scale: can be specified by formula, but note CI problems,[3]

```
regress lnwpc milk120
margins , at( milk120=(1200(1000)5600))
margins , at( milk120=(1200(1000)5600)) expression(exp(predict(xb)))
marginsplot
```

[3] The correct CIs are obtained by backtransformation, but the method used by `margins` command is based on an approximate SE on original scale.

Examples with 2 Categorical Predictors

(2a) <u>additive model</u>: predictions require decision about how to weight contributions from other predictor,

- ∗ equally/balanced (standard in least squares means),
- ∗ total data weights (default choice),
- ∗ choices corresponding to specific prediction settings,

```
regress wpc i.rp i.vag_disch
margins rp vag_disch
table rp vag_disch, row col
lincom _cons+1.rp+1.vag_disch*82/1574 /* rp=1 */
margins rp vag_disch, asbalanced
lincom _cons+1.rp+1.vag_disch*0.5 /* rp=1 */
margins rp, over(vag_disch) /* same as: at(vag_disch=(0 1)) */
lincom _cons+1.rp+1.vag_disch /* rp=1, vag_disch=1 */
```

(2b) <u>model with interaction</u>: combined effect ∼ simple means, separate effects require decision about how to weight contributions from other predictor (as above for additive model),

```
regress wpc rp##vag_disch
margins rp#vag_disch
marginsplot, noci
margins rp
lincom _cons+1.rp+(1.vag_disch+1.rp#1.vag_disch)*82/1574
/* rp=1 */
margins rp, asbalanced
lincom _cons+1.rp+(1.vag_disch+1.rp#1.vag_disch)*0.5  /* rp=1 */
```

Examples with 2 Predictors (cont.)

Categorical + continuous predictor:

(2c) similar to single continuous predictor, with multiple groups (intercepts and lines),

```
regress wpc i.dyst milk120
margins dyst, at( milk120=(1200 2200 3200 4300 5500))
marginsplot, noci
lincom _cons+1.dyst+milk120*3200 /* dyst=1, milk120=3200 */
margin dyst, atmeans
lincom _cons+1.dyst+milk120*3215.096
* interaction model
regress wpc dyst##c.milk120
margins dyst, at( milk120=(1200 2200 3200 4300 5500))
marginsplot, noci
lincom _cons+1.dyst+(c.milk120+1.dyst#c.milk120)*3200
/* dyst=1, milk120=3200 */
```

Two continuous predictors:

(2d) need values (possibly lists) for both predictors ⇒ predictions usually fully specified (no averaging/weighting),

```
regress wpc parity milk120
margins , at( parity=(1(1)6) milk120=(1200 2200 3200 4300 5500))
marginsplot, noci
margins , at( milk120=(1200 2200 3200 4300 5500) parity=(1(1)6) )
marginsplot, noci /* changing roles in plot */
margins , at( milk120=(1200 2200 3200 4300 5500) (median)parity)
marginsplot
lincom _cons+milk120*1200+parity*2 /* milk120=1200, parity=2 */
margins, atmeans
lincom _cons+milk120*3215.096+parity*2.73628 /* both at means */
```

Prediction in Multivariable Models

Main challenge/thing to remember: predictions need values or weights for all predictor terms in model
$\Rightarrow$ no software can do this automatically (so that it always makes sense)!

Some issues to consider when setting up predictions:

- the purpose (e.g., "real" versus "illustrative"),
- should the prediction correspond to an average instead of a real situation? (e.g., when using weights for categorical predictors, the predictions won't correspond to real situations),[4]
- are the predictor distributions independent enough to set the values for different predictors independently?[4]
- is the predictor distribution in the observed data representative for the population or the targeted setting?[4]
- for categorical predictors, are predictions intended to facilitate pairwise comparisons in contrast to comparisons with baseline? (perhaps the main motivation of least squares means),
- if modelling is carried out on transformed scale, should any weighting take place on transformed or original scale? (as they will lead to different results).

[4] Using `margins` with its default settings implies that your answer to this question is "yes".

Predictions for VER Example 14.16

Model summary:

- outcome: `wpc`, on square-root transformed scale,
- categorical predictors: `aut_calv`, `twin`, `dyst`, `rp##vag_disch`,
- continuous predictors: `parity`, `herd_size` with quadratic term.

Some possible prediction aims:

1) illustrate combined effect of diseases (`rp`,`dyst`,`vag_disch`) on `wpc`,
2) illustrate interaction `rp#vag_disch` (effectively included under 1),
3) illustrate effect of `herd_size` on `wpc`.

1): Prediction/Estimates for combinations of disease, with backtransformed (squared) means ∼ *median* `wpc`-values:

Estimates*		$\sqrt{\texttt{wpc}}$ (mean)		`wpc` (median)	
`rp`	`vag_d`	`dyst`=0	`dyst`=1	`dyst`=0	`dyst`=1
0	0	7.517	8.059	56.50	64.95
0	1	7.503	8.046	56.30	64.73
1	0	7.906	8.448	62.51	71.38
1	1	9.384	9.926	88.06	98.53

* at: `parity`=1, `twin`=0, `herd_size`=251, `aut_calv`=0
(∼ the mean herd size, and the most frequent categories)

3): Prediction/Estimates for observed (7!) herd sizes:

Estimates*	herd sizes						
scale	125	185	201	235	263	294	333
$\sqrt{\texttt{wpc}}$	7.092	7.013	7.079	7.448	7.674	8.177	9.002
`wpc`	50.29	49.19	50.11	53.85	58.90	66.86	81.03

* at: `parity`=1, `twin`=0, `aut_calv`=0, all diseases=0

Predictions/presentation of effects on probability scale:

- easier to understand probabilities than OR's,
- not additive ⇒ more complicated (care is needed).

Illustration for Nocardia data and effect of `dcpct`:

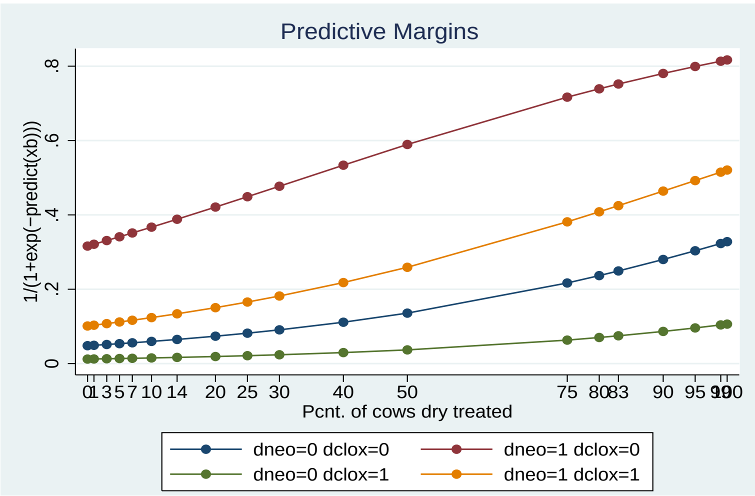
$$\text{logit}(\hat{p}) = -2.984 + 0.023\,\texttt{dcpct} + 2.212\,\texttt{dneo} - 1.412\,\texttt{dclox},$$

- OR for 1% "change" in `dcpct` $= e^{0.023} = 1.023$,
- OR for 10% "change" in `dcpct` $= e^{0.023 \cdot 10} = e^{0.23} = 1.25$.

or on probability scale[5]

$$\hat{p} = \text{logit}^{-1}(\text{logit}(\hat{p})) = 1/(1 + e^{-\text{logit}(\hat{p})})$$

can be plotted against `dcpct` values in a suitable range, for *fixed* values of all other predictors (here `dneo` and `dclox`):



note: non-linear relation!
non-parallel curves!

How to compute predictions – same options as for linear models:

- predicted values for actual/new observations,
- Stata: `margins` and `marginsplot` commands.[6]

[5] For demonstration purposes only; in a case-control design predicted probabilities *do not make sense* because the proportion of cases and controls is controlled.
[6] See next page for discussion of averaging or weighting.

Scale-dependence of Predictions with Averaging/Weighting

Main message: in models involving transformations (e.g. logistic regression), any averaging of predictions involves a choice of scale:

- different results (even after transformation to same scale),
- different interpretations.

Consider again the logistic "predictive" equation

$$\text{logit}(\hat{p}) = -2.984 + 0.023\,\texttt{dcpct} + 2.212\,\texttt{dneo} - 1.412\,\texttt{dclox},$$

— Table of predictions for $\texttt{dcpct} = 0$:

`dneo`	`dclox`	logit($\hat{p}$)	$\hat{p} = \text{logit}^{-1}(\hat{p})$
0	0	-2.984	0.0481
0	1	-4.397	0.0123
1	0	-0.772	0.316
1	1	-2.184	0.101
weighted*		-1.821	0.198
(transformed)		0.139	-1.399

* based on data counts for the 4 categories of (`dneo`,`dclox`): 22, 12, 59, 15

Interpretations:

- $\hat{p}$ (averaged): $\sim$ averaged probabilities, might correspond to a population (if data representative),
- logit($\hat{p}$) (averaged and then transformed): simpler, because backtransformation of a "natural" value.