ECON 557 – Advanced Data Analysis

Michael T. Sandfort

Department of Economics
Masters in Applied Economics Program
Georgetown University

February 10, 2023

Marginal Effects

Review: Marginal Effects in OLS

- Often, we want to know how our predictor would treat a change in the level of one of our covariates. It is common to hear this called a marginal effect (whether or not causality has been established). For example, in demand analysis we want to know the marginal effect of price (a right hand side variable) on quantity demanded (the left hand side variable).
- For minimum MSE loss with a correctly specified model, that means we are interested in
$$\frac{\partial E[y|\mathbf{x}]}{\partial x_j}$$
- In linear models where covariates enter linearly (that's two "linear"s), the marginal effect of a unit change in one particular explanatory variable on the predictor of the dependent variable is just the coefficient on that explanatory variable. That is, if the population model has
$$E[y|\mathbf{x}] = \mathbf{x}\boldsymbol{\beta} = \beta_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_k x_k$$
then the marginal effect on $E[y|\mathbf{x}]$ of a change in x_j is just β_j, or $\frac{\partial E[y|\mathbf{x}]}{\partial x_j} = \frac{\partial}{\partial x_j}\mathbf{x}\boldsymbol{\beta} = \beta_j$

Review: Marginal Effects in OLS

Set up our CPS data as usual

```
> library(haven)
> cps <- read_dta(paste0(myDataPath,"cps09mar.dta"))
> cps$wage <- cps$earnings/(cps$hours*cps$week)
> cps$lwage <- log(cps$wage)
> cps$sex <- factor(cps$female,labels=c("Male","Female"))
> cps$exper <- cps$age-cps$education-6
> cps$expersq <- (cps$exper)^2
```

Review: Marginal Effects in OLS

```
> res.lm <- lm(lwage ~ sex + exper, data=cps)
> summary(res.lm)

Call:
lm(formula = lwage ~ sex + exper, data = cps)

Residuals:
     Min       1Q   Median       3Q      Max
-11.0070  -0.3792  -0.0015   0.3912   2.6386

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.9460505  0.0068301  431.33   <2e-16 ***
sexFemale   -0.2340859  0.0059601  -39.28   <2e-16 ***
exper        0.0044936  0.0002526   17.79   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6638 on 50739 degrees of freedom
Multiple R-squared:  0.0354,Adjusted R-squared:  0.03536
F-statistic:   931 on 2 and 50739 DF,  p-value: < 2.2e-16
> library(margins)
> z <- margins(res.lm)
> z
Average marginal effects
lm(formula = lwage ~ sex + exper, data = cps)
    exper sexFemale
 0.004494   -0.2341
```

Review: Marginal Effects in OLS

```
> hist(z$dydx_exper,n=100,xlim=c(-.1,.1),border="red")
```

Review: Marginal Effects in OLS

- Nonlinear functions of explanatory variables in linear models take more work. That is, if the linear population model has

$$E[y|\mathbf{x}] = \beta_1 + \beta_2 x_2 + \beta_3 x_2^2 + \beta_4 x_4 + \cdots + \beta_k x_k$$

  then the marginal effect on $E[y|\mathbf{x}]$ of a change in x_2 is not β_2, but rather

$$\frac{\partial E[y|\mathbf{x}]}{\partial x_j} = \beta_2 + 2\beta_3 x_2.$$

- Observe that the marginal effect of x_2 depends on the value of x_2. That's different than it was in the linear model where the marginal effect was just β_2, a constant.
- If y is the wage and x_2 is experience, then the incremental effect of more experience depends on accumulated experience.
- In wage regressions, we expect $\beta_2 > 0$ and $\beta_3 < 0$, so more experience leads to higher wages, but the "wage bump" from the third year of experience is smaller than that from the second year of experience.

Review: Marginal Effects in OLS

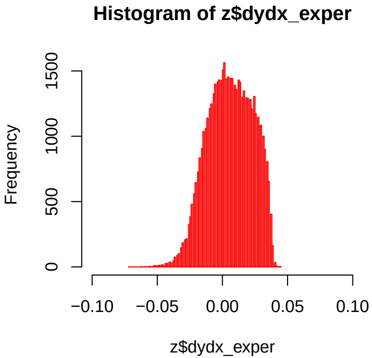
Experience enters both linearly and quadratically as a covariate.

```
> res.lm2 <- lm(lwage ~ sex + exper + I(exper^2), data=cps )
> summary(res.lm2)

Call:
lm(formula = lwage ~ sex + exper + I(exper^2), data = cps)

Residuals:
     Min       1Q   Median       3Q      Max
-10.8690  -0.3732   0.0010   0.3889   2.8747

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.646e+00  1.007e-02  262.68   <2e-16 ***
sexFemale   -2.307e-01  5.869e-03  -39.31   <2e-16 ***
exper        3.879e-02  8.934e-04   43.42   <2e-16 ***
I(exper^2)  -7.361e-04  1.842e-05  -39.97   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6536 on 50738 degrees of freedom
Multiple R-squared:  0.06484,Adjusted R-squared:  0.06478
F-statistic:  1173 on 3 and 50738 DF,  p-value: < 2.2e-16
> z <- margins(res.lm2)
> z
Average marginal effects
lm(formula = lwage ~ sex + exper + I(exper^2), data = cps)
    exper sexFemale
 0.006093   -0.2307
```

Review: Marginal Effects in OLS

`margins()` returns a list. Per-observation marginal effects are stored in `dydx_[VARNAME]`:

```
> hist(z$dydx_exper,n=100,xlim=c(-.1,.1),border="red")
```




```
> fivenum(z$dydx_exper)
[1] -0.071626905 -0.006851221  0.006398350  0.019647922  0.044674890
```

Review: Interpreting OLS Results

Why is this formulation different? $\texttt{expersq} = \texttt{exper}^2$.

```
> res.lm2a <- lm(lwage ~ sex + exper + expersq, data=cps )
> summary(res.lm2a)

Call:
lm(formula = lwage ~ sex + exper + expersq, data = cps)

Residuals:
     Min       1Q   Median       3Q      Max
-10.8690  -0.3732   0.0010   0.3889   2.8747

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.646e+00  1.007e-02  262.68   <2e-16 ***
sexFemale   -2.307e-01  5.869e-03  -39.31   <2e-16 ***
exper        3.879e-02  8.934e-04   43.42   <2e-16 ***
expersq     -7.361e-04  1.842e-05  -39.97   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6536 on 50738 degrees of freedom
Multiple R-squared:  0.06484,Adjusted R-squared:  0.06478
F-statistic:  1173 on 3 and 50738 DF,  p-value: < 2.2e-16
> margins(res.lm2a)
Average marginal effects
lm(formula = lwage ~ sex + exper + expersq, data = cps)
   exper    expersq sexFemale
 0.03879 -0.0007361   -0.2307
```

Marginal Effects in GLMs

- Computing marginal effects in GLMs is similar to computing marginal effects in OLS when covariates enter nonlinearly.
- But in GLMs, non-constant marginal effects potentially arise for two distinct reasons:
  - First (like OLS with nonlinear-in-x covariates), the linear predictor may be nonlinear in x (even though it is linear in $\boldsymbol{\beta}$):

    $$\eta = \mathbf{x}\boldsymbol{\beta} = \beta_0 + \beta_1 x + \beta_2 x^2 \implies \frac{\partial \eta}{\partial x} = \beta_1 + 2\beta_2 x.$$

  - Second (unlike OLS), the mean $\mu(\eta) = \mu(\mathbf{x}\boldsymbol{\beta})$ is derived from the linear predictor by applying the, often nonlinear, inverse link function.

Marginal Effects in GLMs

- You may already be familiar with this in the logit and probit setting, where for a continuous x_j which enters only linearly, the partial effect of x_j is

$$\frac{\partial E[y|\mathbf{x}]}{\partial x_j} = \frac{\partial Pr(y=1|\mathbf{x})}{\partial x_j} = \frac{\partial G(\mathbf{x}\boldsymbol{\beta})}{\partial x_j} = g(\mathbf{x}\boldsymbol{\beta})\beta_j$$

  where $G(\cdot)$ is the inverse link (either the logistic or normal CDF), and $g(z) = \frac{dG(z)}{dz}$ is the density of G.

- Even though x_j enters only linearly, the marginal effect of x_j in logit or probit is not constant. It depends on the levels of all other covariates $x_1, \ldots, x_k$ through the derivative of the inverse link $g(\mathbf{x}\boldsymbol{\beta})$.

- Since different sets of covariates lead to different values of the marginal effect, and perhaps we don't want to report thousands of marginal effects, we need to think about how to summarize this information.

Marginal Effects in GLMs: Two Approaches

- Average Marginal Effect (AME), or Average Partial Effect (APE)
- Partial Effect at the Average (PEA)

Average Marginal Effect (AME)

- Corresponding to the variation in $\mathbf{x}$ in our draws $\{\mathbf{x}_i\}_{i=1}^n$, we will have variation in estimated marginal effects given by

$$\frac{\partial \mu(\eta(\mathbf{x}_i \widehat{\boldsymbol{\beta}}_n))}{\partial x_j} = \frac{d\mu}{d\eta} \frac{\partial \eta(\mathbf{x}_i \widehat{\boldsymbol{\beta}}_n)}{\partial x_j}$$

  where we allow for x_j to enter possibly nonlinearly in $\eta(\cdot)$ (i.e., as experience did in our log wage regressions).

- Because η is a scalar function of $\mathbf{x}$, each of these estimated marginal effects is a scalar value. Their distribution over the data can be summarized by a histogram and/or by usual summary statistics for a univariate distribution (mean, median, mode).
- The AME is just the sample average of these effects, or

$$\frac{1}{n} \sum_{i=1}^{n} \frac{d\mu}{d\eta} \frac{\partial \eta(\mathbf{x}_i \widehat{\boldsymbol{\beta}}_n)}{\partial x_j}$$

  but other statistics of the distribution may also be interesting, depending on the application.

Partial Effect at the Average (PEA)

- Continuing the theme that the marginal effect varies with the covariates and can be computed for any given set of covariates $\mathbf{x}$, sometimes the sample averages of each x_j are computed across the $\{\mathbf{x}_i\}_{i=1}^n$, giving $\overline{\mathbf{x}}$.
- A single marginal effect can then be computed at the single "average" level of the covariates, giving

$$\frac{d\mu}{d\eta}\frac{\partial\eta(\overline{\mathbf{x}}\widehat{\boldsymbol{\beta}}_n)}{\partial x_j}$$

  which is reported as the partial effect at the average, or PEA.
- When computation was slow and expensive, this approach offered significant time savings (particularly in large samples). Currently it is used less, for reasons that will become apparent as we work through the example.

Marginal Effects in GLMs

Recalling our probit model of labor force participation:

```
> library(wooldridge)
> suppressMessages(library(tidyverse))
> data(mroz)
> res.probit <- glm(inlf ~ nwifeinc + educ + exper + I(exper^2) + age + kidslt6 + kidsge6,
+                   family=binomial(link="probit"),
+                   data=mroz)
> summary(res.probit)

Call:
glm(formula = inlf ~ nwifeinc + educ + exper + I(exper^2) + age +
    kidslt6 + kidsge6, family = binomial(link = "probit"), data = mroz)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2156  -0.9151   0.4315   0.8653   2.4553

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.2700736  0.5080782   0.532  0.59503
nwifeinc    -0.0120236  0.0049392  -2.434  0.01492 *
educ         0.1309040  0.0253987   5.154 2.55e-07 ***
exper        0.1233472  0.0187587   6.575 4.85e-11 ***
I(exper^2)  -0.0018871  0.0005999  -3.145  0.00166 **
age         -0.0528524  0.0084624  -6.246 4.22e-10 ***
kidslt6     -0.8683247  0.1183773  -7.335 2.21e-13 ***
kidsge6      0.0360056  0.0440303   0.818  0.41350
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1029.7  on 752  degrees of freedom
Residual deviance:  802.6  on 745  degrees of freedom
AIC: 818.6

Number of Fisher Scoring iterations: 4
```

Marginal Effects in GLMs

Following the calculations described above, in our probit example the marginal effect of education at $\mathbf{x}$ can be calculated as

$$\begin{aligned} \frac{d\mu}{d\eta}\frac{\partial \eta(\mathbf{x}\boldsymbol{\beta})}{\partial x_3} &= \frac{\partial}{\partial \eta}\Phi(\eta)\frac{\partial \mathbf{x}\boldsymbol{\beta}}{\partial x_3} \\ &= \frac{\partial}{\partial \eta}\Phi(\eta)\beta_3 \\ &= \phi(\mathbf{x}\boldsymbol{\beta})\beta_3 \end{aligned}$$

And at the estimated coefficients $\widehat{\boldsymbol{\beta}}$, we have for each $\mathbf{x}_i$ a marginal effect of education estimated as $\phi(\mathbf{x}_i\widehat{\boldsymbol{\beta}})\widehat{\beta}_3$ (percentage points).

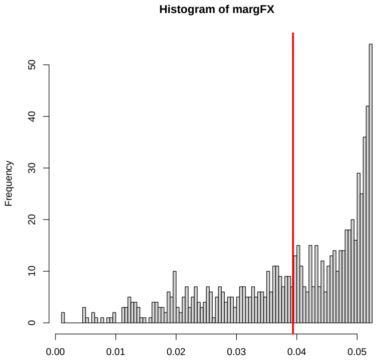
AME in the Probit Example

- Computing the marginal effects by hand is not difficult.

```
> X.data <- mroz |> mutate(iota=1) |>
+   select(iota,nwifeinc,educ,exper,expersq,age,kidslt6,kidsge6)
> etaHat <- as.matrix(X.data) %*% matrix(coef(res.probit),ncol=1)
> margFX <- dnorm(etaHat)*coef(res.probit)[3]
```

- This figure shows the distribution of the partial effect of education in our probit labor force participation model. The average partial effect is marked in red.

```
> hist(margFX,n=100,xlab=NULL); abline(v=mean(margFX),lwd=3,col="red")
```



AME in the Probit Example

- The canned version of marginal effects is

```
> margins(res.probit)
Average marginal effects
glm(formula = inlf ~ nwifeinc + educ + exper + I(exper^2) + age +     kidslt6 + kidsge6, family =
binomial(link = "probit"), data = mroz)
  nwifeinc    educ   exper     age kidslt6 kidsge6
 -0.003616 0.03937 0.02558 -0.0159 -0.2612 0.01083
```

- It matches what we got calculating by hand

```
> mean(margFX)
[1] 0.03937009
```

- Will these estimates give the correct marginal effects for experience? No.
- To fix it, we need to specify the model so that the software knows that $\texttt{expersq} \equiv \texttt{exper}^2$:

```
> model.improved = inlf~nwifeinc+educ+exper+I(exper^2)+age+kidslt6+kidsge6
> res.probit.improved <- glm(model.improved,data=mroz,family=binomial(link="probit"))
> margins(res.probit.improved)
Average marginal effects
glm(formula = model.improved, family = binomial(link = "probit"),     data = mroz)
  nwifeinc    educ   exper     age kidslt6 kidsge6
 -0.003616 0.03937 0.02558 -0.0159 -0.2612 0.01083
> res.probit <- res.probit.improved
```

PEA in the Probit Example

- We want to compute $\overline{\mathbf{x}}$ so that we can compute $\phi(\overline{\mathbf{x}}\widehat{\boldsymbol{\beta}})\widehat{\beta}_3$.
- It's easy enough to calculate the mean values for our data:

```
> X.avg <- mroz |> mutate(iota=1) |>
+   select(iota,nwifeinc,educ,exper,expersq,age,kidslt6,kidsge6) |>
+   summarize_all(mean) |>
+   as.matrix(nrow=1)
> X.avg
     iota nwifeinc     educ    exper  expersq      age   kidslt6  kidsge6
[1,]    1 20.12896 12.28685 10.63081 178.0385 42.53785 0.2377158 1.353254
```

- So we can find the PEA for education in our probit model as

```
> pea.educ = dnorm(matrix(X.avg,nrow=1) %*% matrix(coef(res.probit),ncol=1)) *
+              coef(res.probit)[3]
> pea.educ
           [,1]
[1,] 0.05112843
```

PEA in the Probit Example

- The `margins` package also allows us to compute margins at any given set of covariates using the `at=` argument.
- First, create a data frame containing the means of the covariate matrix $\mathbf{X}$:

```
> X.avg <- mroz |>
+   select(nwifeinc,educ,exper,age,kidslt6,kidsge6) |>
+   summarize_all(mean)
> X.avg
  nwifeinc     educ    exper      age   kidslt6  kidsge6
1 20.12896 12.28685 10.63081 42.53785 0.2377158 1.353254
```

- Next, compute the marginal effects at the means:

```
> margins(res.probit,at=X.avg)
Average marginal effects at specified values
glm(formula = model.improved, family = binomial(link = "probit"),     data = mroz)
 at(nwifeinc) at(educ) at(exper) at(age) at(kidslt6) at(kidsge6)  nwifeinc
        20.13    12.29     10.63   42.54      0.2377       1.353 -0.004545
    educ   exper      age kidslt6 kidsge6
 0.04948 0.03146 -0.01998 -0.3282 0.01361
```

- Which matches our earlier result exactly for education. (?!)

Marginal Effects in GLMs

- Oh...wait...it doesn't? Let's try

```
> X.avg <- mroz |> mutate(iota=1) |>
+   select(iota,nwifeinc,educ,exper,age,kidslt6,kidsge6) |>
+   summarize_all(mean) |> mutate(expersq=exper^2,.after=exper) |>
+   as.matrix(nrow=1)
> X.avg
     iota nwifeinc     educ    exper  expersq      age   kidslt6  kidsge6
[1,]    1 20.12896 12.28685 10.63081 113.0141 42.53785 0.2377158 1.353254
```

- And *now* it matches.

```
> pea.educ = dnorm(matrix(X.avg,nrow=1) %*% matrix(coef(res.probit),ncol=1)) *
+             coef(res.probit)[3]
> pea.educ
           [,1]
[1,] 0.04947932
```

- What did I change and why?

Drawbacks of the Partial Effect at the Average (PEA)

PEA is easy to compute, but it has some drawbacks:

- First, it need not represent the partial effect for any particular unit in the population. This is a common problem with the mean as a measure of central tendency – there may not actually be any element of the sample equal to, or even particularly near, the sample mean (e.g., indicator variables).

- Second, if $\mathbf{x}$ contains any nonlinear functions of covariates (e.g., experience squared) we have a bit of a problem. Since $\overline{h(x)} \neq h(\bar{x})$ for nonlinear $h()$, we have to decide which one we're going to use (and hope/check our software agrees!). We now know `margins` uses the latter of the two.

Marginal Effects in GLMs (Discrete Covariates)

- For discrete x_j (like an indicator variable), there's no such thing as an infinitesimal change in x_j. In this case, the relevant calculation is the discrete change in $G(\eta(\mathbf{x}\boldsymbol{\beta}))$ between the two distinct values of x_j.
- Supposing for the moment that x_j is a dummy variable that interacts with no other covariates, the effect of a change in x_j would be

$$G(\beta_1+\beta_2 x_2+\cdots+\beta_j\cdot 1+\cdots+\beta_k x_k)-G(\beta_1+\beta_2 x_2+\cdots+\beta_j\cdot 0+\cdots+\beta_k x_k)$$

- For most GLMs, $G()$ is nondecreasing, so this difference is positive when β_j is positive and negative when β_j is negative.
- Obviously, if x_j interacts with other covariates, the above expression gets more complicated.

Marginal Effects in GLMs (Discrete Covariates)

We don't have any indicator variables in the labor force participation regression, but this gives us an opportunity to see how the log wage regression works as a GLM.

```
> res.norm <- glm(lwage ~ sex + exper, family=gaussian(link="identity"), data=cps)
> summary(res.norm)

Call:
glm(formula = lwage ~ sex + exper, family = gaussian(link = "identity"),
    data = cps)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-11.0070   -0.3792   -0.0015    0.3912    2.6386

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.9460505  0.0068301  431.33   <2e-16 ***
sexFemale   -0.2340859  0.0059601  -39.28   <2e-16 ***
exper        0.0044936  0.0002526   17.79   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.4406729)

    Null deviance: 23180  on 50741  degrees of freedom
Residual deviance: 22359  on 50739  degrees of freedom
AIC: 102424

Number of Fisher Scoring iterations: 2
> margins(res.norm)
Average marginal effects
glm(formula = lwage ~ sex + exper, family = gaussian(link = "identity"),     data = cps)
    exper sexFemale
 0.004494   -0.2341
```

Marginal Effects in GLMs (Discrete Covariates)

- To do this by hand, we take

$$
\begin{aligned}
\Delta\mu(\mathbf{x}\boldsymbol{\beta}) &= \iota(\eta(\mathbf{x}\boldsymbol{\beta}))\big|_{\texttt{sexFemale}=1} - \iota(\eta(\mathbf{x}\boldsymbol{\beta}))\big|_{\texttt{sexFemale}=0} \\
&= \iota(\beta_0 + \beta_1 \cdot 1 + \beta_2\texttt{exper}) - \iota(\beta_0 + \beta_1 \cdot 0 + \beta_2\texttt{exper}) \\
&= \beta_0 + \beta_1 + \beta_2\texttt{exper} - \beta_0 - \beta_2\texttt{exper} \\
&= \beta_1
\end{aligned}
$$

  where $\iota()$ is the identity function.

- We would not get as much simplification if we were not in a "Gaussian response/identity link" modeling setting, but it's nice to know that when we *are* in that setting, we get the expected result.

Marginal Effects in GLMs

- A final reminder – don't expect a canned margins routine to deliver magic!
- Unless you explicitly make it aware of interactions and powers of your covariates, the computed marginal effects will not reflect those nonlinearities.
- And they will likely be incorrect.

Standard Errors for Marginal Effects

Standard Errors for Marginal Effects

- Once we have estimated the marginal effects of interest, we may also be curious how precise the estimates are.
- The marginal effects are functions of the coefficient estimates, so the precision of our estimates of the marginal effects depends directly on the precision of our coefficient estimates.
- The precision of our coefficient estimates under QML is not something we have discussed until now.

Useful Properties of GLMs from Likelihood Theory

- We start with two key results from likelihood theory.
- Recall that the log-likelihood function is given by

$$\mathcal{L}(\boldsymbol{\theta}) \equiv \ln L(\boldsymbol{\theta}) = \ln f(y|\mathbf{x}, \boldsymbol{\theta})$$

  and that we view it as a random function (y conditional on $\mathbf{x}$ is random) which we want to maximize over $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

Useful Properties of GLMs from Likelihood Theory

- Optimization problems have first order necessary and second order sufficient conditions (for interior solutions):
  - The derivative of the function must be 0 in all directions to achieve an interior optimum ("first order necessary").
  - At such a point, if the second derivative of the function is negative, the point is a local maximum ("second order sufficient").
- For the log likelihood function, the first derivative is known as the score vector $\mathbf{s}(\boldsymbol{\theta})$, where

$$\mathbf{s}(\boldsymbol{\theta}) = \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

- The second derivative is known as the Hessian matrix $\mathbf{H}(\boldsymbol{\theta})$, where

$$H_{ij} = \frac{\partial^2 \mathcal{L}}{\partial \theta_i \theta_j}$$

Useful Properties of GLMs from Likelihood Theory

- If the model is correct then, starting from the true statement that $\int f(y|x, \boldsymbol{\theta}_o) = 1$, the following two properties of all likelihoods are straightforward to show by repeated differentiation with respect to $\boldsymbol{\theta}$.
  - Zero expected score
$$\mathbb{E}\left[\mathbf{s}(\boldsymbol{\theta}_o)\right] = \mathbb{E}\left[\frac{\partial}{\partial \boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_o)\right] = \mathbf{0}_{K\times 1}$$
  - Information equality
$$\mathbb{V}\left[\mathbf{s}(\boldsymbol{\theta}_o)\right] = -\mathbb{E}\left[\mathbf{H}(\boldsymbol{\theta}_o)\right]$$

  where the expectation is taken relative to $f(y|\mathbf{x}, \boldsymbol{\theta}_o)$.

- For GLMs, these result in the statements we saw before that
$$\mathbb{E}[y] = b'(\theta) = \mu \quad \text{and} \quad \mathbb{V}[y] = \phi b''(\theta)$$

The Score Function (Example 1)

- For a univariate random variable y with exponential density $f(y|x,\theta) = f(y,\theta) = \frac{1}{\theta}e^{-\frac{y}{\theta}}$.

$$s(\theta) = \frac{\partial \mathcal{L}(\theta, y)}{\partial \theta} = -\frac{1}{\theta} + \frac{y}{\theta^2}$$

- It follows that

$$E[s(\theta)] = -\frac{1}{\theta} + \frac{E[y]}{\theta^2} = -\frac{1}{\theta} + \frac{\theta_o}{\theta^2}$$

so

$$E[s(\theta_o)] = -\frac{1}{\theta_o} + \frac{\theta_o}{\theta_o^2} = 0$$

as expected.

The Score Function (Example 2)

- For our normal linear regression, previous work shows that

$$\mathbf{s}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta, y|\mathbf{x})}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}(\theta, y|\mathbf{x})}{\partial (\sigma^2)} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{x}'(y - \mathbf{x}\boldsymbol{\beta})}{\sigma^2} \\ -\frac{1}{2\sigma^4}\left[\sigma^2 - (y - \mathbf{x}\boldsymbol{\beta})^2\right] \end{bmatrix}$$

- For $\boldsymbol{\beta}$, we have

$$E[\mathbf{s}(\boldsymbol{\theta})|\mathbf{x}][1, \ldots, k] = \frac{\mathbf{x}'(E[y] - \mathbf{x}\boldsymbol{\beta})}{\sigma^2} = \frac{\mathbf{x}'\mathbf{x}(\boldsymbol{\beta}_o - \boldsymbol{\beta})}{\sigma^2}$$

so that

$$E[\mathbf{s}(\boldsymbol{\theta})|\mathbf{x}][1, \ldots, k]|_{\boldsymbol{\beta} = \boldsymbol{\beta}_o} = \frac{\mathbf{x}'\mathbf{x}(\boldsymbol{\beta}_o - \boldsymbol{\beta}_o)}{\sigma^2} = \mathbf{0}_{k \times 1}$$

as expected.

With a True Model, Maximum Likelihood Estimates are CAN

- Showing that, with a true model, the maximum likelihood estimator is consistent and asymptotically normal is beyond the scope of this class, but we will use the following results:
  - If $f(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta})$ is the true density of $\mathbf{y}_i$ given $\mathbf{x}_i$ and $\boldsymbol{\theta}$ (specification), $\boldsymbol{\theta}_o$ uniquely maximizes the expected likelihood of $\mathbf{y}_i$ given $\mathbf{x}_i$ and $\boldsymbol{\theta}$ (identification), $\boldsymbol{\Theta} \subset \mathbb{R}^n$ is closed and bounded, and f and the log-likelihood are differentiable, then $\hat{\boldsymbol{\theta}}_{\text{ML}}$ is a consistent estimator of $\boldsymbol{\theta}_o$.
  - The ML estimate is asymptotically normal, with

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_{\text{ML}} - \boldsymbol{\theta}_o) \xrightarrow{d} N\left(\mathbf{0}, (-E[\mathbf{H}(\boldsymbol{\theta}_o)])^{-1}\right)$$

    where $\mathbf{H}$ is the Hessian of the log-likelihood.
- This is one reason we are interested in the Hessian of the likelihood function.

ML Standard Errors

- That the sampling variance should be related to the second derivative of log-likelihood is an intuitive result.
- The second derivative defines curvature of a function. "Negative Hessian" means downward curvature, consistent with a maximum.
- The more curved the log-likelihood function at the ML estimate, the more the function declines at values nearby the ML estimate.
- Differences in log-likelihood values are our basis for rejecting candidate estimates, so the more curved is $\mathcal{L}$ at our maximum likelihood estimate, the more our best estimate will distinguish itself from nearby alternatives. The curvature of the log-likelihood is sometimes called the precision.
- But note that the second derivative condition is a local condition: more curvature leads to a more prominent local maximum, but will not tell you whether your local max is a global max. Fortunately, for many models (including the gaussian, logit and probit), $\mathcal{L}$ is globally concave, so there is exactly one (unique) maximizer of $\mathcal{L}$.

The Hessian: Examples

- For a univariate random variable y with exponential density (Example 1),

$$H(\theta) = \frac{\partial^2 \mathcal{L}(\theta, y)}{\partial \theta^2} = \frac{1}{\theta^2} - \frac{2y}{\theta^3}$$

- For our normal linear regression model (Example 2),

$$\begin{aligned}
\mathbf{H}(\boldsymbol{\theta}) &= \begin{bmatrix} \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta}, y|\mathbf{x})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta}, y|\mathbf{x})}{\partial \boldsymbol{\beta} \partial (\sigma^2)} \\ \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta}, y|\mathbf{x})}{\partial \sigma^2 \partial \boldsymbol{\beta}} & \frac{\partial^2 \mathcal{L}(\boldsymbol{\theta}, y|\mathbf{x})}{\partial (\sigma^2)^2} \end{bmatrix} \\
&= \begin{bmatrix} -\frac{\mathbf{x}'\mathbf{x}}{\sigma^2} & -\frac{\mathbf{x}'(y - \mathbf{x}\boldsymbol{\beta})}{\sigma^4} \\ -\frac{\mathbf{x}'(y - \mathbf{x}\boldsymbol{\beta})}{\sigma^4} & \frac{1}{2\sigma^4} - \frac{(y - \mathbf{x}\boldsymbol{\beta})^2}{\sigma^6} \end{bmatrix}
\end{aligned}$$

The Information Matrix

- Following our earlier discussion, we define the (conditional) observed Fisher information matrix as

$$\mathcal{I}_{\mathbf{x}}(\boldsymbol{\theta}_o) \equiv -E[\mathbf{H}(\boldsymbol{\theta}_o)|\mathbf{x}]$$

- Every draw $(\mathbf{x}, y)$ generates a random contribution to the likelihood, a random contribution to the score, and a random contribution to the Hessian.

- The expected Fisher information (i.e., the expected outer product of the scores) is

$$\begin{aligned} \mathcal{I}_{\mathbf{x}}(\boldsymbol{\theta}_o) &= V[\mathbf{s}(\boldsymbol{\theta}_o)|\mathbf{x}] \\ &= E[\mathbf{s}(\boldsymbol{\theta}_o)\mathbf{s}(\boldsymbol{\theta}_o)'|\mathbf{x}] \qquad \text{(why?)} \end{aligned}$$

  where the fact that both observed and expected information are $\mathcal{I}_{\mathbf{x}}$ is just our information matrix equality result from before.

Information Matrix Examples

For Example 1, we have

$$\begin{aligned}
-E[H(\theta_o)] &= -E\left[\frac{1}{\theta_o^2} - \frac{2y}{\theta_o^3}\right] \\
&= -\frac{1}{\theta_o^2} + \frac{2E[y]}{\theta_o^3} \\
&= -\frac{1}{\theta_o^2} + \frac{2}{\theta_o^2} \\
&= \frac{1}{\theta_o^2}
\end{aligned}$$

or

$$\begin{aligned}
V[s(\theta_o)] &= V\left[\frac{-1}{\theta_o} + \frac{y}{\theta_o^2}\right] \\
&= \frac{1}{\theta_o^4} V[y] \\
&= \frac{\theta_o^2}{\theta_o^4} = \frac{1}{\theta_o^2}
\end{aligned}$$

Information Matrix Examples

For Example 2, we have

$$
\begin{aligned}
-E[H(\boldsymbol{\theta}_o)|\mathbf{x}] &= -E\left[\left[\begin{matrix} -\frac{\mathbf{x}'\mathbf{x}}{\sigma_o^2} & -\frac{\mathbf{x}'(y-\mathbf{x}\boldsymbol{\beta}_o)}{\sigma_o^4} \\ -\frac{\mathbf{x}'(y-\mathbf{x}\boldsymbol{\beta}_o)}{\sigma_o^4} & \frac{1}{2\sigma_o^4} - \frac{(y-\mathbf{x}\boldsymbol{\beta}_o)^2}{\sigma_o^6} \end{matrix}\right] \middle| \mathbf{x}\right] \\
&= \begin{bmatrix} \frac{\mathbf{x}'\mathbf{x}}{\sigma_o^2} & \frac{\mathbf{x}'E[y-\mathbf{x}\boldsymbol{\beta}_o|\mathbf{x}]}{\sigma_o^4} \\ \frac{\mathbf{x}'E[y-\mathbf{x}\boldsymbol{\beta}_o]}{\sigma_o^4} & -\frac{1}{2\sigma_o^4} + \frac{E[(y-\mathbf{x}\boldsymbol{\beta}_o)^2|\mathbf{x}]}{\sigma_o^6} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\mathbf{x}'\mathbf{x}}{\sigma_o^2} & \mathbf{0}_{k\times 1} \\ \mathbf{0}_{1\times k} & \frac{1}{2\sigma_o^4} \end{bmatrix}
\end{aligned}
$$

Estimating ML Standard Errors

Due to the information matrix equality, there are three possibilities for estimating the asymptotic VC matrix for our ML estimates:

1. The Hessian of the log-likelihood (compute the second derivatives)

$$\left[-\sum_{i=1}^{N}\mathbf{H}_i(\hat{\boldsymbol{\theta}}_{\mathsf{ML}})\right]^{-1}$$

2. The outer product of the gradient (OPG) estimator

$$\left[\sum_{i=1}^{N}\mathbf{s}_i(\hat{\boldsymbol{\theta}}_{\mathsf{ML}})\mathbf{s}_i(\hat{\boldsymbol{\theta}}_{\mathsf{ML}})'\right]^{-1}$$

3. The information matrix (compute the second derivatives and take the expectation)

$$\left[\sum_{i=1}^{N}-E[\mathbf{H}_i(\hat{\boldsymbol{\theta}}_{\mathsf{ML}})]\right]^{-1}$$

An Example (from Greene)

- Suppose we observe education (x_i) and income (y_i) for $N = 20$ individuals. Fake data for this exercise is provided in the file `example.txt`.

```
> ex.data <- read.table("example.txt",header=T)
```

- Suppose further that the DGP for this population is

$$f(y_i|x_i,\theta) = \frac{1}{\theta + x_i} e^{\frac{-y_i}{\theta + x_i}}$$

- Evidently, this is an exponential population with parameter $\theta + x_i$, so $E[y] = \theta + x$.
- The log-likelihood function for the sample is given by

$$\mathcal{L}_N(\theta) = -\sum_{i=1}^{N} \log(\theta + x_i) - \sum_{i=1}^{N} \frac{y_i}{\theta + x_i}$$

An Example (from Greene)

- Necessary first order condition for $\widehat{\theta}_{ML}$ is

$$\frac{\partial \mathcal{L}_N(\theta)}{\partial \theta} = -\sum_{i=1}^{N} \frac{1}{\widehat{\theta}_{ML} + x_i} + \sum_{i=1}^{N} \frac{y_i}{(\widehat{\theta}_{ML} + x_i)^2} = 0$$

- Messy to solve the above for $\widehat{\theta}_{ML}$, so

```
> LL.ex <- function(my.t,my.df) {
+     A <- ( my.t + my.df$x )
+     B <- ( my.df$y / A )
+     return(-sum( log(A) + B ))
+ }
> DLL.ex <- function(my.t,my.df) {
+     A <- 1/(my.t + my.df$x)
+     B <- my.df$y * A * A
+     return(sum( -A + B ))
+ }
> res <- optim(0,LL.ex,gr=DLL.ex,ex.data,method="BFGS",control=list(fnscale=-1,reltol=1e-12))
> res$par
[1] 15.60273
```

- So the optimum occurs at $\widehat{\theta}_{ML} = 15.6027271$.

An Example (from Greene)

- To compute the three estimates of the asymptotic variance of $\widehat{\theta}_{ML}$, we need the second derivative of the log-likelihood function

$$\frac{\partial^2 \mathcal{L}_N(\theta)}{\partial \theta^2} = \sum_{i=1}^{N} \frac{1}{(\theta + x_i)^2} - 2\sum_{i=1}^{N} \frac{y_i}{(\theta + x_i)^3}$$

- The first estimator of the asymptotic variance is

$$\widehat{\text{Avar}}(\widehat{\theta}_{ML}) = \left[-\sum_{i=1}^{N} \mathbf{H}_i(\hat{\boldsymbol{\theta}}_{\text{ML}}) \right]^{-1}$$

or simply

$$-\left[\sum_{i=1}^{N} \frac{1}{(\widehat{\theta}_{ML} + x_i)^2} - 2\sum_{i=1}^{N} \frac{y_i}{(\widehat{\theta}_{ML} + x_i)^3} \right]^{-1}$$

```
> A <- res$par + ex.data$x
> Avar1 <- -1/( sum(1/( A^2 )) - 2*sum(ex.data$y/( A^3 )) )
> Avar1
[1] 46.16337
```

An Example (from Greene)

- The second estimator of the asymptotic variance is the OPG estimator, or

$$\left[\sum_{i=1}^{N} \mathbf{s}_i(\hat{\boldsymbol{\theta}}_{\mathsf{ML}})\mathbf{s}_i(\hat{\boldsymbol{\theta}}_{\mathsf{ML}})'\right]^{-1}$$

which gives

$$\left[\sum_{i=1}^{N}\left(-\frac{1}{\widehat{\theta}_{ML}+x_i}+\sum_{i=1}^{N}\frac{y_i}{(\widehat{\theta}_{ML}+x_i)^2}\right)^2\right]^{-1}$$

```
> A <- res$par + ex.data$x
> Avar2 <- 1/( sum( ( -1/A + ex.data$y/(A^2) )^2 ) )
> Avar2
[1] 100.5116
```

An Example (from Greene)

- The third estimator of the asymptotic variance relies on the fact that $E[y] = \theta + x$, so the expected Hessian is

$$
\begin{aligned}
E\left[\frac{\partial^2 \mathcal{L}_N(\theta)}{\partial \theta^2} | x\right] &= E\left[\sum_{i=1}^{N} \frac{1}{(\theta + x_i)^2} - 2\sum_{i=1}^{N} \frac{y_i}{(\theta + x_i)^3} | x\right] \\
&= \sum_{i=1}^{N} \frac{1}{(\theta + x_i)^2} - 2\sum_{i=1}^{N} \frac{E[y_i | x_i]}{(\theta + x_i)^3} \\
&= \sum_{i=1}^{N} \frac{1}{(\theta + x_i)^2} - 2\sum_{i=1}^{N} \frac{\theta + x_i}{(\theta + x_i)^3} \\
&= -\sum_{i=1}^{N} \frac{1}{(\theta + x_i)^2}
\end{aligned}
$$

```
> Avar3 <- -1/( -sum(1/( A^2 )) )
> Avar3
[1] 44.2546
```

Model Misspecification

- Without a correct model, we can still get an analog to the zero expected score condition like
$$\mathbb{E}_g\left[\frac{\partial}{\partial\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^*\right] = \mathbf{0}$$
- Unfortunately, if we don't use the correct model, the information equality no longer holds, so we need to separately define the observed information
$$J(\theta) = E_g\left[\frac{\partial^2\mathcal{L}(y,\theta)}{\partial\theta\partial\theta'}\right]$$
and expected information
$$V(\theta) = E_g\left[\frac{\partial\mathcal{L}(y,\theta)}{\partial\theta}\frac{\partial\mathcal{L}(y,\theta)}{\partial\theta}\right]$$
- Even so, our QML estimator is consistent for θ^* ($\hat{\theta}_{\text{QML}} \to \theta^*$) and asymptotically normal, with
$$\sqrt{N}(\hat{\theta}_{\text{QML}} - \theta^*) \overset{A}{\sim} N(0, J^{-1}VJ^{-1})$$

"Robust" Standard Errors

The above is the basis for the Huber-Eicker-White HC ("robust") errors:

- Suppose the true DGP has

$$y_i = x_i\beta + u_i \qquad \text{with} \qquad u_i \sim N(0, \sigma_i^2)$$

  so independent, but not identically distributed errors. Let $\mathbf{y}$ be the y's and $\mathbf{X}$ hold the x's.

- Suppose the model (misspecification here) has $\sigma_i^2 = \sigma^2$.

- We still get an expected likelihood function

$$E_g[\mathcal{L}_N(\beta)] = E_g[(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)]$$

  so

$$\frac{\partial E_g[\mathcal{L}(\beta)]}{\partial \beta} = E_g[(\mathbf{y} - \mathbf{X}\beta)'\mathbf{X}]$$

- So $\beta^* = \beta$ (!) and $\hat{\beta}_{\text{QML}} \to \beta$ by the theorem.

"Robust" Standard Errors

- But to get consistent estimates of the standard errors, we need to use the QML form, where
$$J(\beta) = \frac{\partial^2 E_g[\mathcal{L}(\beta)]}{\partial \beta' \partial \beta} = \mathbf{X}^\top \mathbf{X}$$
and
$$V(\beta) = \frac{\partial E_g[\mathcal{L}(\beta)]}{\partial \beta}' \frac{\partial E_g[\mathcal{L}(\beta)]}{\partial \beta} = \mathbf{X}^\top \mathbf{u}\mathbf{u}^\top \mathbf{X}$$
- So $\sqrt{N}(\hat{\beta}_{\text{QML}} - \beta) \overset{A}{\sim} N(0, (\mathbf{X}^\top\mathbf{X})^{-1}[\mathbf{X}^\top\mathbf{u}\mathbf{u}^\top\mathbf{X}](\mathbf{X}^\top\mathbf{X})^{-1})$.
- This is exactly the Huber-Eicker-White (HC) "sandwich" covariance matrix for $\hat{\beta}$.

A Note on “Robustness”

- Note that, if the model is correctly specified, that the information matrix equality gives $J = V = \mathcal{I}$.
- Among Hal White’s many contributions was a test for model mis-specification built from this condition.
- But use of “robust” errors is now very common – many ignore the fact that the origin of these statistics was as a test for mis-specification.
- That is, if the “robust” errors are much different from the typical standard errors in your model, then you need to worry about misspecification.
- After all (Freedman, 2006), what’s the point in presenting consistent estimates for standard errors of a parameter that may be of no interest?

Standard Errors for Marginal Effects

- We have now seen how to estimate the covariance matrix for our QML estimator. Standard errors for coefficients can be taken from the diagonal of the asymptotic covariance matrix in the usual way.
- Standard errors for marginal effects take one more step due to the nonlinearity of the marginal effects – they are usually calculated by the delta method.
- The delta method is quite general – it can be used to calculate confidence intervals for any function $h(\widehat{\boldsymbol{\beta}})$ of a set of coefficient estimates where an estimate of the covariance matrix for $\widehat{\boldsymbol{\beta}}$ is available.
- To calculate confidence intervals for the marginal effects, we'll take $h()$ to be marginal effect of x_j:

$$h(\boldsymbol{\beta}) = \frac{\partial \mu(\mathbf{x}\boldsymbol{\beta})}{\partial x_j}$$

- We know the sampling variance of $\widehat{\boldsymbol{\beta}}$. We're trying to find the sampling variance of $h(\widehat{\boldsymbol{\beta}})$.

The Delta Method

- If $h()$ were linear rather than nonlinear (say $\mathbf{c}\boldsymbol{\beta}$), then our job would be easy.
- We saw before that the sampling variance of the ML estimator is the inverse of the negative expected Hessian

$$V(\widehat{\boldsymbol{\beta}}) \equiv \mathbf{A}^{-1} = E[-\mathbf{H}(\boldsymbol{\beta})]^{-1}$$

so

$$\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} N(\mathbf{0}, \mathbf{A}^{-1}).$$

- We could therefore rely on our rules for variances to say that for some vector $\mathbf{c}$, $V(\mathbf{c}\widehat{\boldsymbol{\beta}}) = \mathbf{c}\mathbf{A}^{-1}\mathbf{c}^{\top}$.

The Delta Method

- But if $h()$ is differentiable, it's almost linear in the sense that we can take a Taylor series approximation

$$h(\widehat{\boldsymbol{\beta}}) \approx h(\boldsymbol{\beta}) + (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top \frac{\partial h(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

- Rearranging terms in the above expression gives

$$\sqrt{n}[h(\widehat{\boldsymbol{\beta}}) - h(\boldsymbol{\beta})] \approx \sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top \frac{\partial h(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

so

$$h(\widehat{\boldsymbol{\beta}}) \overset{a}{\sim} N\left(h(\boldsymbol{\beta}), \frac{\partial h(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^\top} V(\boldsymbol{\beta}) \frac{\partial h(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)$$

ML Standard Errors for Marginal Effects

- Suppose $h()$ is the partial effect of x_j, entering only linearly into our probit model. Then it's easy to show that

$$\frac{\partial h(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{bmatrix} \frac{\partial h}{\partial \beta_1} & \frac{\partial h}{\partial \beta_2} & \cdots & \frac{\partial h}{\partial \beta_k} \end{bmatrix}$$

$$= \begin{bmatrix} \phi'(\mathbf{x}\boldsymbol{\beta})x_1\beta_j \\ \phi'(\mathbf{x}\boldsymbol{\beta})x_2\beta_j \\ \vdots \\ \phi'(\mathbf{x}\boldsymbol{\beta})x_j\beta_j + \phi(\mathbf{x}\boldsymbol{\beta}) \\ \vdots \\ \phi'(\mathbf{x}\boldsymbol{\beta})x_k\beta_j \end{bmatrix}'$$

- The delta method is applicable any time you want to find standard errors for a function of your coefficients, i.e., you can use it with OLS, FGLS, GMM or ML.

Example: ML SE's for Marginal Effects (APE)

```
> summary(margins(res.probit))
  factor     AME     SE       z      p   lower   upper
     age -0.0159 0.0024 -6.7392 0.0000 -0.0205 -0.0113
    educ  0.0394 0.0073  5.4186 0.0000  0.0251  0.0536
   exper  0.0256 0.0022 11.4506 0.0000  0.0212  0.0300
 kidsge6  0.0108 0.0132  0.8189 0.4129 -0.0151  0.0367
 kidslt6 -0.2612 0.0319 -8.1860 0.0000 -0.3237 -0.1986
 nwifeinc -0.0036 0.0015 -2.4604 0.0139 -0.0065 -0.0007
```

Example: ML Standard Errors for Marginal Effects

These effects are straightforward to calculate, as well:

```
> X.avg
    iota nwifeinc     educ    exper  expersq      age   kidslt6  kidsge6
[1,]    1 20.12896 12.28685 10.63081 113.0141 42.53785 0.2377158 1.353254
> k <- length(X.avg); k
[1] 8
> beta.j <-  coef(res.probit)[3]
> idx <- (matrix(X.avg,1,k) %*% matrix(coef(res.probit),k,1))[1,1]
> grad.h <- matrix(-idx*dnorm(idx)*X.avg*beta.j,nrow=1,ncol=k)
> grad.h[1,3] <- grad.h[1,3]+dnorm(idx)
> vcov.marginal <- grad.h %*% vcov(res.probit) %*% t(grad.h)
> sqrt(vcov.marginal)
            [,1]
[1,] 0.009660841
```

You can easily check that this is the same as what is returned by `margins` for the standard error on the education variable.

Example: ML SE's for Marginal Effects (PEA)

```
> options(width=150)
> summary(margins(res.probit,at=as.data.frame(X.avg)[,-1]))
   factor nwifeinc    educ   exper  expersq     age kidslt6 kidsge6     AME     SE       z      p   lower   upper
      age  20.1290 12.2869 10.6308 113.0141 42.5378  0.2377  1.3533 -0.0200 0.0032 -6.1635 0.0000 -0.0263 -0.0136
     educ  20.1290 12.2869 10.6308 113.0141 42.5378  0.2377  1.3533  0.0495 0.0097  5.1217 0.0000  0.0305  0.0684
    exper  20.1290 12.2869 10.6308 113.0141 42.5378  0.2377  1.3533  0.0315 0.0031 10.0721 0.0000  0.0253  0.0376
  kidsge6  20.1290 12.2869 10.6308 113.0141 42.5378  0.2377  1.3533  0.0136 0.0166  0.8177 0.4135 -0.0190  0.0462
  kidslt6  20.1290 12.2869 10.6308 113.0141 42.5378  0.2377  1.3533 -0.3282 0.0453 -7.2473 0.0000 -0.4170 -0.2394
 nwifeinc  20.1290 12.2869 10.6308 113.0141 42.5378  0.2377  1.3533 -0.0045 0.0019 -2.4348 0.0149 -0.0082 -0.0009
```

Bootstrapping

Cheap computing power offers an alternative to, and occasionally an improvement on, the use of asymptotic theory to approximate the sampling distributions of estimators.

We start with an estimator $T(\mathbf{W})$ and a sample $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^{N}$ from our population with distribution F. We want to know

$$P(T(\mathbf{W}) \leq t)$$

the distribution of $T(\cdot)$. As we saw with marginal effects under ML, even approximating the variance of this distribution can be an involved process. Generally, the distribution of $T(\cdot)$ is a complicated function of F, t and the sample size N.

Bootstrapping

Our guiding analogy for bootstrapping begins with the notion that our sample is itself a "population" with distribution function $\widehat{F}_N(\mathbf{w})$. Unlike the true population F, however, we can draw new samples from $\widehat{F}_N(\mathbf{w})$ at will.

In large samples, $\widehat{F}_N$ should get close to F (by the Fundamental Theorem of Statistics) so sampling from $\widehat{F}_N$ ought to be a good stand-in for sampling from F.

So if we are interested in the sampling distribution of $T(\mathbf{W})$, we might approximate it by computing its values $T(\mathbf{W}^*)$ across many draws $\mathbf{W}^* = \{\mathbf{w}_i^*\}_{i=1}^N$ from $\widehat{F}_N$. The draws $\{\mathbf{W}_b^*\}_{b=1}^B$ for some large B are called the bootstrap samples, and the process of drawing them is typically called resampling.

Bootstrapping Regression

For the regression model

$$y = m(\mathbf{x}, \boldsymbol{\beta}) + u,$$

three types of resampling (i.e., three distinct empirical distributions $\widehat{F}_N$) are commonly used in practice. Having estimated a parametric model, we can then:

- Resample from the observations $\{\mathbf{w}_i\} = \{(y_i, \mathbf{x}_i)\}$ – the nonparametric or paired bootstrap.
- Resample from the residuals $\{\widehat{u}_i\}$ – the residual bootstrap.
- Resample from the estimated model with $y_i \sim F(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ – the parametric bootstrap.

I've ordered them from most-used to least-used, which interestingly also corresponds to ordering them inversely by how much they rely on the correctness of the specified model.

Bootstrapping: Paired Bootstrap

- The paired bootstrap assumes neither that the model for the conditional mean $m()$ nor that the error specification u is correct.

- This approach simply resamples from the existing sample, so that $\mathbf{w}_i^*$ is simply a row taken at random (i.e., with probability $\frac{1}{N}$ and with replacement) from $\{(y_i, \mathbf{x}_i)\}_{i=1}^N$.

Bootstrapping: Residual Bootstrap

- The residual bootstrap assumes the model for the conditional mean in the DGP is correct, but allows that the distribution of the modeling error u may be misspecified.
- That's why this approach is often described as "intermediate" between the parametric and nonparametric bootstraps.
- Residual bootstrapping can be used without a fully specified model for u – we simply resample from the empirical distribution of the estimated residuals $\widehat{u}$ – so it works particularly well with OLS absent distributional assumptions.
- If $\{u_i^*\}$ is drawn (with replacement) from $\{\widehat{u}_i\}_{i=1}^N$ with probability $\frac{1}{N}$, then $\mathbf{w}_i^* = (m(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}) + u_i^*, \mathbf{x}_i)$.

Bootstrapping: Parametric Bootstrap

- The parametric bootstrap assumes that we have the population DGP for y correct up to parameters $\boldsymbol{\beta}$.
- Re-run the DGP for y using our parameter estimates via $F(\mathbf{x}_i, \widehat{\boldsymbol{\beta}})$, generating new y_i^*.
- In a normal regression model, we would draw $y_i^* \sim N(m(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}), \widehat{\sigma}^2)$, taking $\mathbf{w}_i^* = (y_i^*, \mathbf{x}_i)$.
- This bootstrapping scheme only works if we have a fully specified parametric model for y.
- Consistent and asymptotically normal OLS estimates may be available without a fully specified parametric model, in which case the parametric bootstrap cannot be used.

Bootstrapping

A quick example where we assume a normal regression model applies to

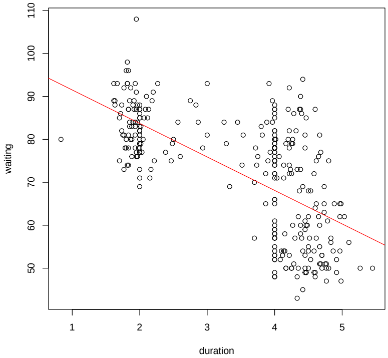
$$y = \beta_0 + \beta_1 x + u$$

where x is the duration of a geyser eruption and y is the waiting time since the last eruption.

```
> library(MASS)

Attaching package:  'MASS'
The following object is masked from 'package:dplyr':
   select
The following object is masked from 'package:wooldridge':
   cement
> data(geyser)
```

Bootstrapping

```
> plot(waiting~duration,data=geyser)
> geyser.lm <- lm(waiting~duration,data=geyser)
> abline(geyser.lm,col="red")
```

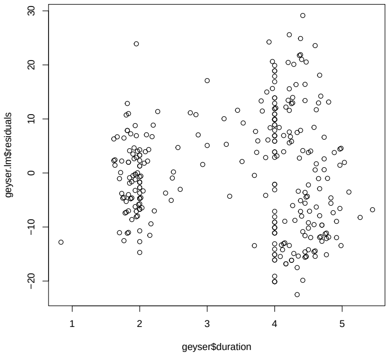


Bootstrapping

```
> summary(geyser.lm)

Call:
lm(formula = waiting ~ duration, data = geyser)

Residuals:
     Min       1Q   Median       3Q      Max
-22.5084  -8.1683  -0.4892   7.5365  29.1416

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  99.3099     1.9569   50.75   <2e-16 ***
duration     -7.8003     0.5368  -14.53   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.64 on 297 degrees of freedom
Multiple R-squared:  0.4155,Adjusted R-squared:  0.4136
F-statistic: 211.2 on 1 and 297 DF,  p-value: < 2.2e-16
```

Bootstrapping

```
> plot(geyser$duration,geyser.lm$residuals)
```



Bootstrapping

A useful function

```
> est.waiting.on.duration <- function(data) {
+     fit <- lm(waiting ~ duration, data=data)
+     return(coefficients(fit))
+ }
```

Pairs bootstrap of confidence intervals:

```
> resample.pairs <- function(this.data.frame) {
+     idx <- sample(1:nrow(this.data.frame),nrow(this.data.frame),replace=T)
+     return(this.data.frame[idx,])
+ }
> bootstrap.CIs.pairs <- function(B,alpha) {
+     beta.star <- replicate(B,est.waiting.on.duration(resample.pairs(geyser)))
+     low.quantiles <- apply(beta.star,1,quantile,probs=alpha/2)
+     high.quantiles <- apply(beta.star,1,quantile,probs=1-alpha/2)
+     C.l <- 2*coefficients(geyser.lm) - high.quantiles
+     C.u <- 2*coefficients(geyser.lm) - low.quantiles
+     CIs <- rbind(C.l,C.u)
+     return(CIs)
+ }
```

Bootstrapping

Residuals bootstrap of confidence intervals:

```
> resample.residuals <- function(lm.results) {
+     new.frame <- lm.results$model
+     new.frame[,1] <- fitted(lm.results) +
+         sample(residuals(lm.results),length(residuals(lm.results)),replace=T)
+     return(new.frame)
+ }
> bootstrap.CIs.residuals <- function(B,alpha) {
+     beta.star <- replicate(B,est.waiting.on.duration(resample.residuals(geyser.lm)))
+     low.quantiles <- apply(beta.star,1,quantile,probs=alpha/2)
+     high.quantiles <- apply(beta.star,1,quantile,probs=1-alpha/2)
+     C.l <- 2*coefficients(geyser.lm) - high.quantiles
+     C.u <- 2*coefficients(geyser.lm) - low.quantiles
+     CIs <- rbind(C.l,C.u)
+     return(CIs)
+ }
```

Bootstrapping

Parametric bootstrap of confidence intervals

```
> resample.parametric <- function(lm.results) {
+     new.frame <- lm.results$model
+     uHat <- residuals(lm.results)
+     N <- length(uHat)
+     estVar <- ( uHat^2 )/N
+     new.frame[,1] <- rnorm(N, mean=fitted(lm.results), sd=sqrt(estVar) )
+     return(new.frame)
+ }
> bootstrap.CIs.parametric <- function(B,alpha) {
+     beta.star <- replicate(B,est.waiting.on.duration(resample.parametric(geyser.lm)))
+     low.quantiles <- apply(beta.star,1,quantile,probs=alpha/2)
+     high.quantiles <- apply(beta.star,1,quantile,probs=1-alpha/2)
+     C.l <- 2*coefficients(geyser.lm) - high.quantiles
+     C.u <- 2*coefficients(geyser.lm) - low.quantiles
+     CIs <- rbind(C.l,C.u)
+     return(CIs)
+ }
```

Bootstrapping

```
> signif(bootstrap.CIs.pairs(B=1e4,alpha=0.05),3)
    (Intercept) duration
C.l        96.5    -8.70
C.u       102.0    -6.92
> signif(bootstrap.CIs.residuals(B=1e4,alpha=0.05),3)
    (Intercept) duration
C.l        95.5    -8.87
C.u       103.0    -6.74
> signif(bootstrap.CIs.parametric(B=1e4,alpha=0.05),3)
    (Intercept) duration
C.l        99.2    -7.85
C.u        99.5    -7.75
```

Are you at all suspicious of the parametric bootstrap results? Why or why not?

A Very Simple Probit Bootstrapping Routine

```
> set.seed(12345)
> B <- 1000
> Bn <- nrow(mroz)
>
> res.boot <- matrix(NA,nrow=B,ncol=length(coef(res.probit)),
+     dimnames=list(rep=seq(B), coef=names(coef(res.probit))))
>
> for (i in seq(B)) {
+   boot.data <- mroz[sample(nrow(mroz),size=Bn,replace=TRUE),]
+   res.boot[i,] <- coef(update(res.probit, data=boot.data))
+ }
```

A Very Simple Probit Bootstrapping Routine

```
> data.frame(mean_est=colMeans(res.boot),
+       t(apply(res.boot,2,quantile,c(0.025,0.975))))
                mean_est        X2.5.        X97.5.
(Intercept)  0.274858344 -0.674075893  1.3225932732
nwifeinc    -0.012269937 -0.023557746 -0.0016063638
educ         0.132703319  0.078342789  0.1887183061
exper        0.122790390  0.084160628  0.1601550366
I(exper^2)  -0.001844416 -0.002957969 -0.0005276581
age         -0.053263512 -0.072354252 -0.0365330603
kidslt6     -0.878733359 -1.107540044 -0.6539403524
kidsge6      0.034964125 -0.055994772  0.1274736517
> summary(res.probit)

Call:
glm(formula = model.improved, family = binomial(link = "probit"),
    data = mroz)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2156  -0.9151   0.4315   0.8653   2.4553

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.2700736  0.5080782   0.532  0.59503
nwifeinc    -0.0120236  0.0049392  -2.434  0.01492 *
educ         0.1309040  0.0253987   5.154 2.55e-07 ***
exper        0.1233472  0.0187587   6.575 4.85e-11 ***
I(exper^2)  -0.0018871  0.0005999  -3.145  0.00166 **
age         -0.0528524  0.0084624  -6.246 4.22e-10 ***
kidslt6     -0.8683247  0.1183773  -7.335 2.21e-13 ***
kidsge6      0.0360056  0.0440303   0.818  0.41350
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1029.7  on 752  degrees of freedom
Residual deviance:  802.6  on 745  degrees of freedom
AIC: 818.6

Number of Fisher Scoring iterations: 4
```