Maximum Likelihood Methods
for the Social Sciences
POLS 510 · CSSS 510

# Models of Binary Data


**Christopher Adolph**

**Political Science *and* CSSS**
**University of Washington, Seattle**


Brent Wadden *Black Bank* and *3 Grey Swoops* 2015

# Plan for today

Today

*How to estimate a regression model with a binary outcome variable*

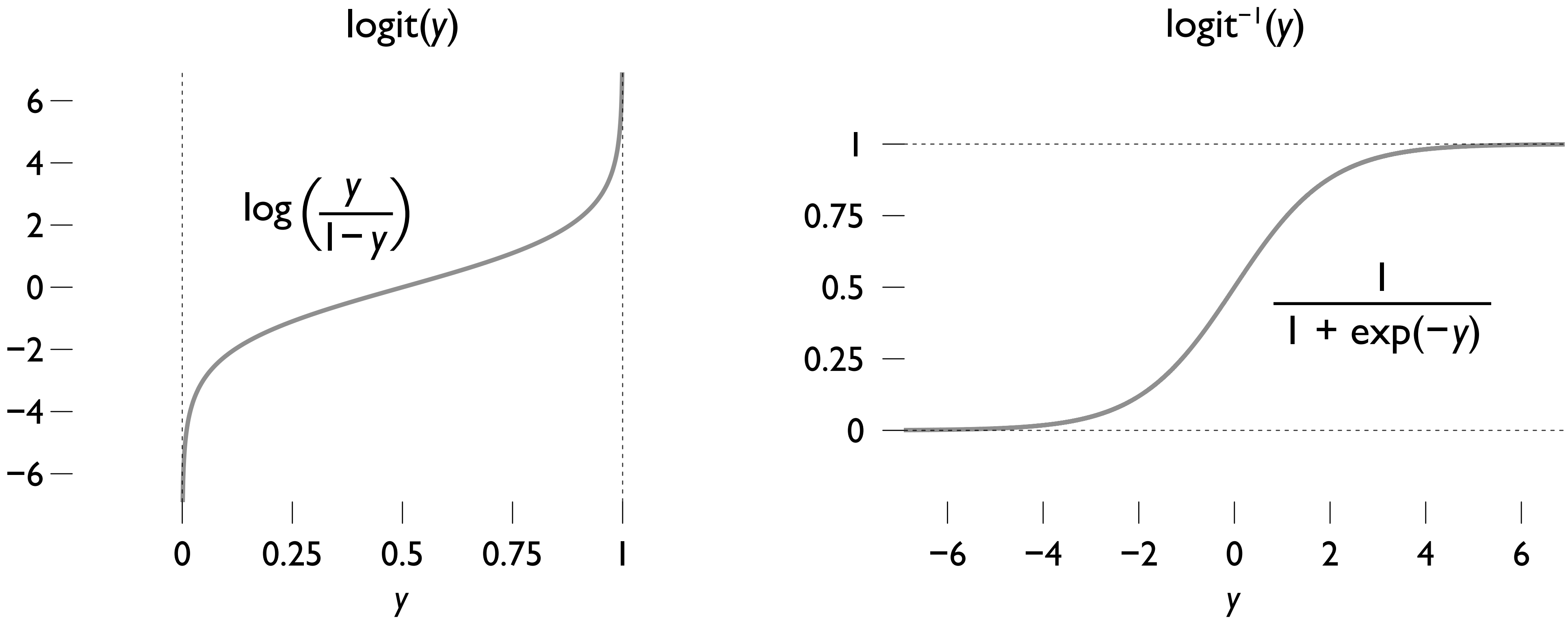
Next time

*How to interpret the results*

Next week

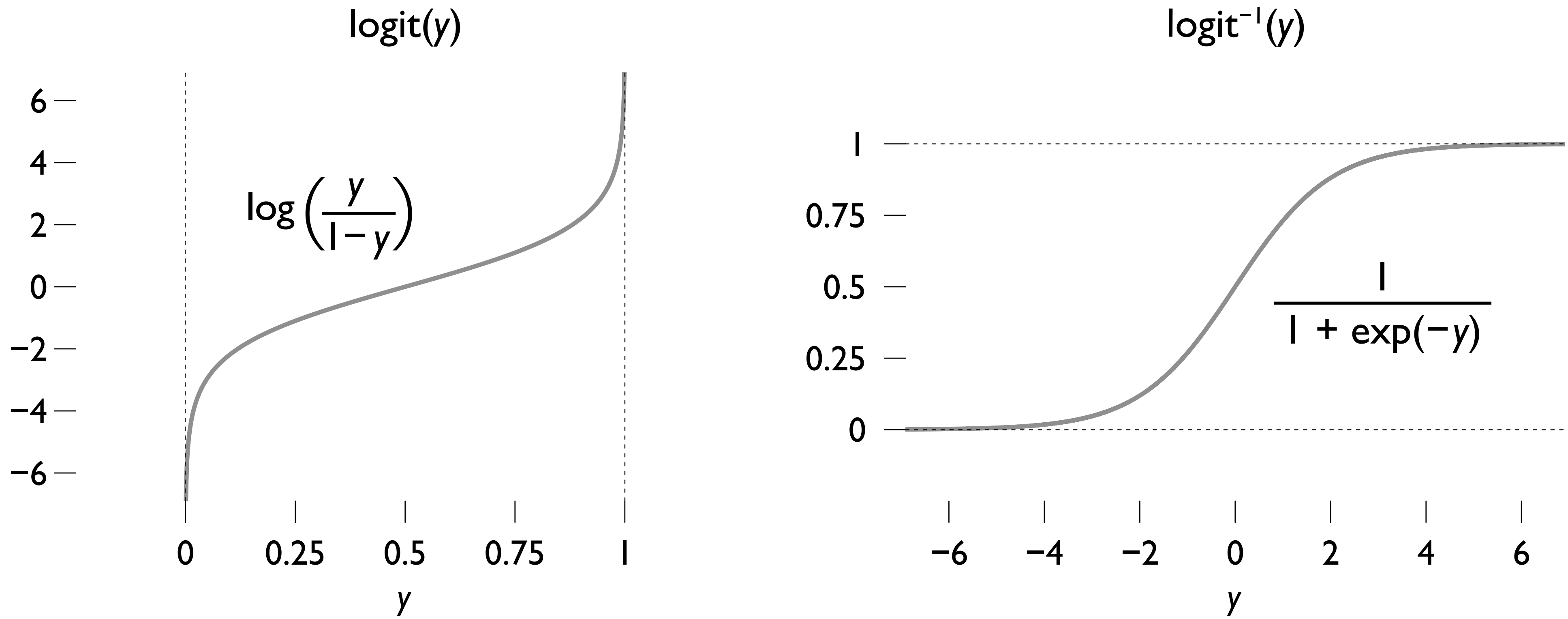
*How to judge whether the model fits the data well*

But first: *exactly what is "logit"?*




Several *distinct* meanings of "logit," including the logit *function* & the logit *model*

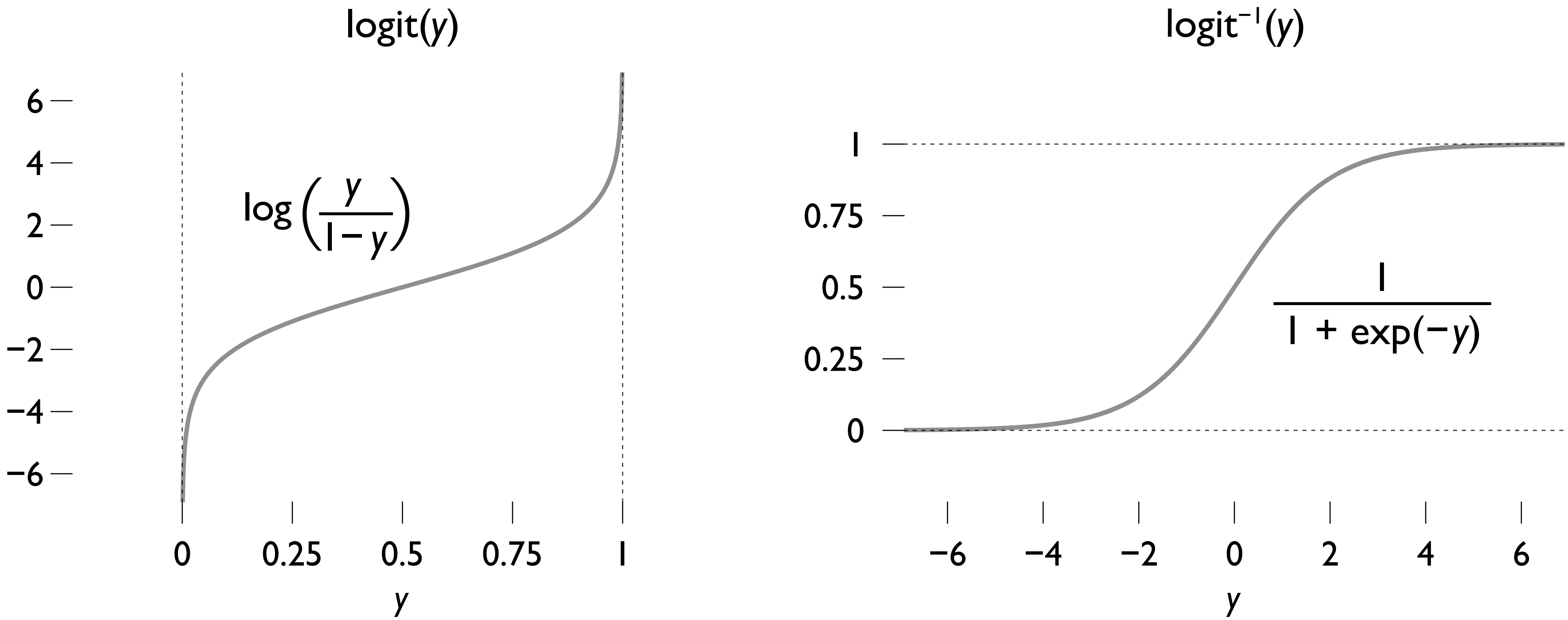
Let's start with the logit function,
which is a tool for transforming continuous data




The logit function is a simple transformation mapping values in (0,1) to $(-\infty, \infty)$:

$$\text{logit}(y) = \log\left(\frac{y}{1-y}\right)$$

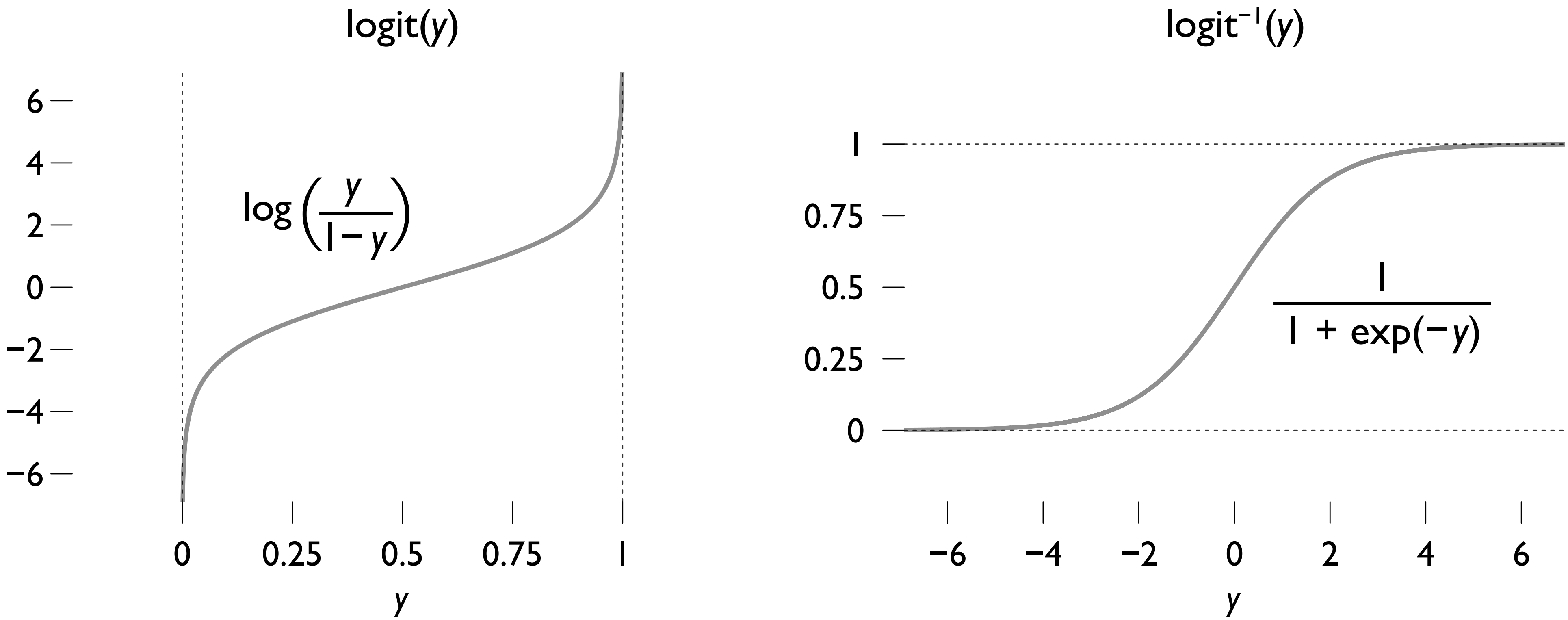
The logit transformation compresses differences near 0.5
and exaggerates differences in the neighborhood of 0 or 1




Logit transforms are often applied to proportion data to make them unbounded, e.g., for use as the outcome in a linear regression

Note that you cannot compute logit(0) or logit(1)

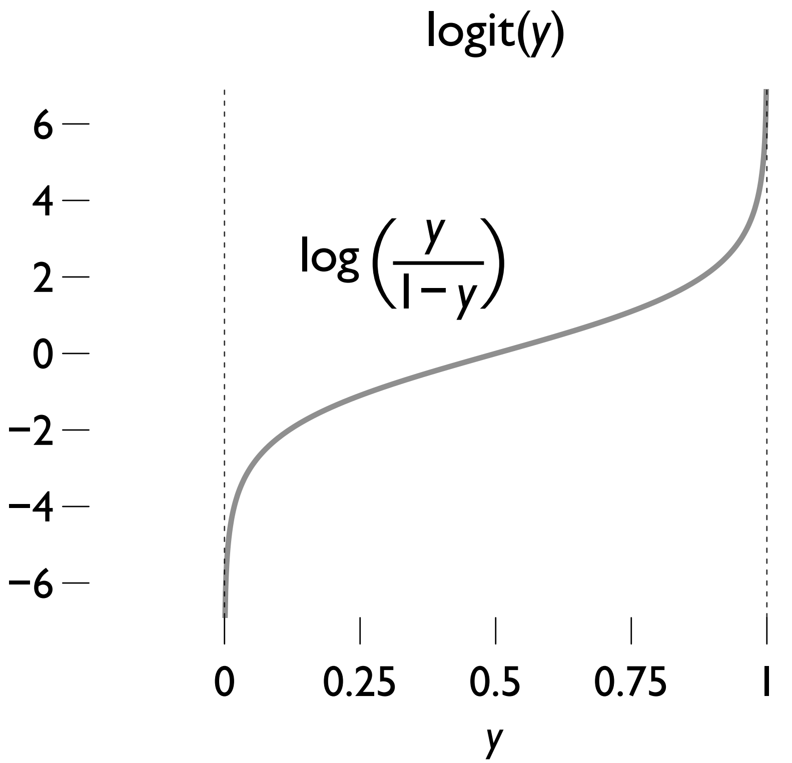
Be careful if your data contain exact 0's or 1's:
you can't just "round them off" to 0.0001 or 0.9999,
as these values become extreme outliers on the logit scale!

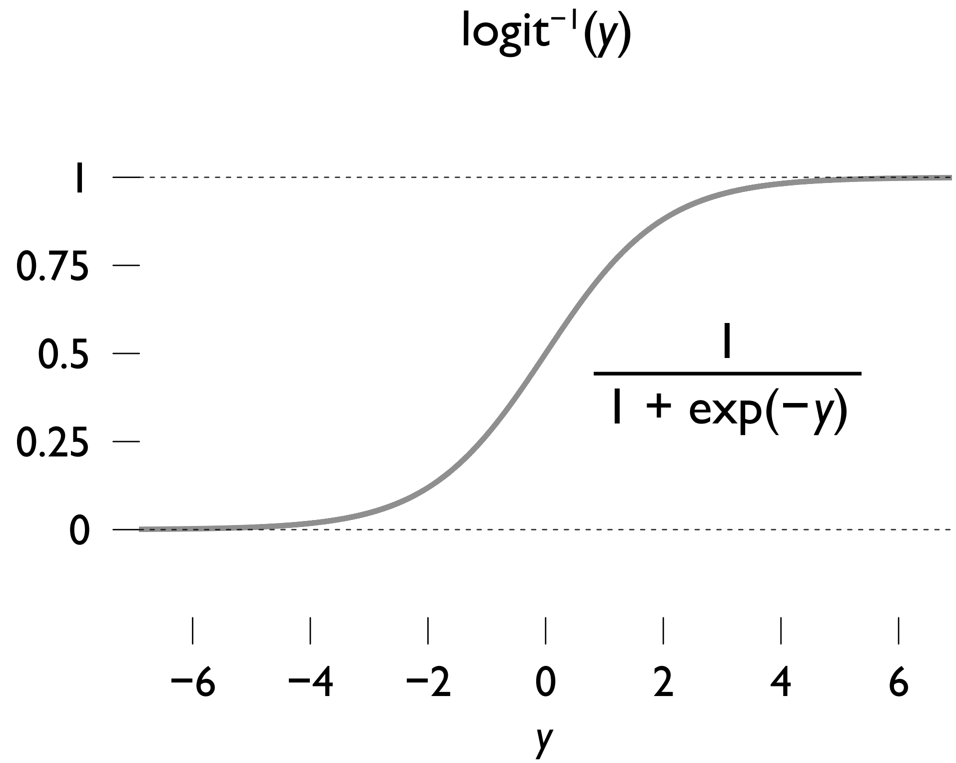



The inverse of this function (the inverse-logit) maps from $(-\infty, \infty)$ to (0,1):

$$\text{logit}^{-1}(y) = \frac{\exp(y)}{1 + \exp(y)} = \frac{1}{1 + \exp(-y)}$$

The inverse-logit transform exaggerates differences in the midrange of $y$ and compresses differences in the neighborhood of relatively small or large $y$'s






These are useful transformations for *continuous data* (akin to the log transformation)

You could use logit transformed variables in a linear regression, for example

*How would you interpret the results on the original scale of $y$?*
Compute $\mathrm{logit}^{-1}(\hat{y}|\mathbf{x}_{\mathrm{hyp}})$ for $\mathbf{x}_{\mathrm{xhyp}}$ of substantive interest

The above would still be linear regression on a transformed, continuous outcome
– not the logit *model*, which is a nonlinear model of a binary outcome

# Binary data & social science

Binary data are perhaps the most common social science measurement:

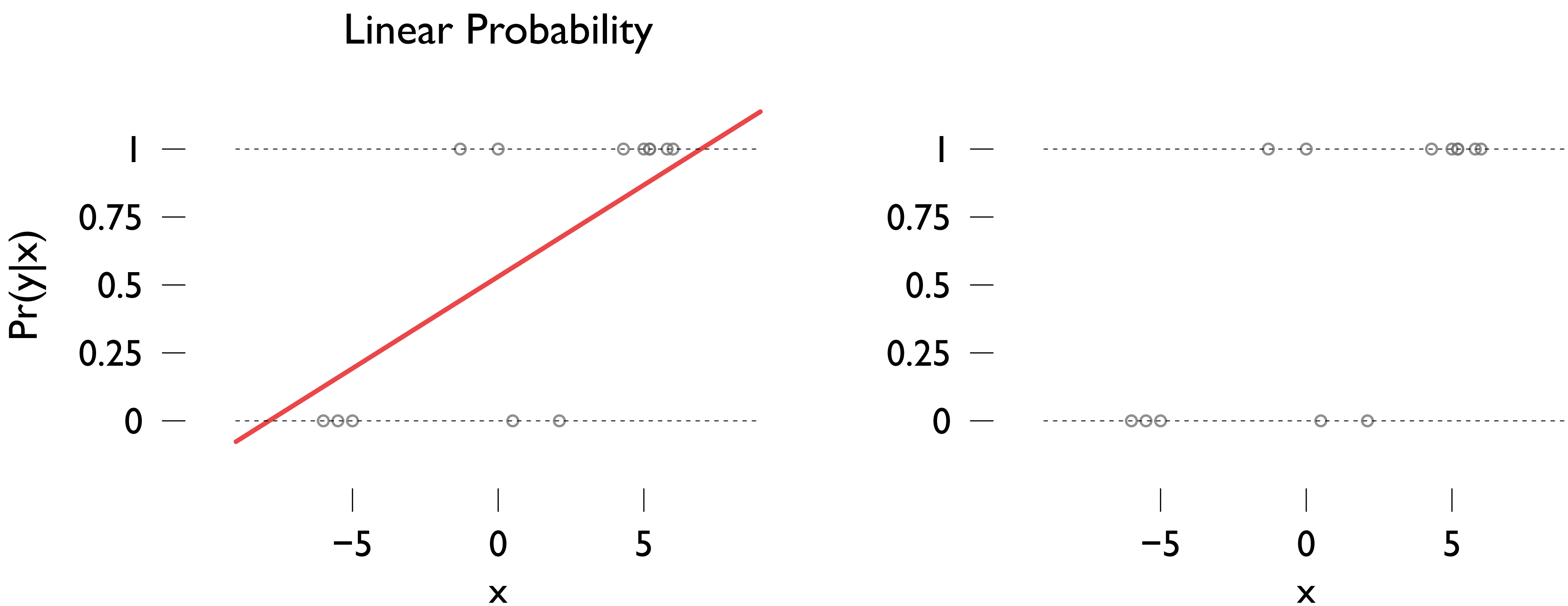
- Did you vote?
- Did an event occur (conflict, democratic transition, strike)?
- Is an institution/custom present in a culture?
- Did an individual commit a crime?
- Did a student complete a degree program?

# Binary data & the MO of MLE

Moreover, binary data analysis is fundamental to many advanced topics:

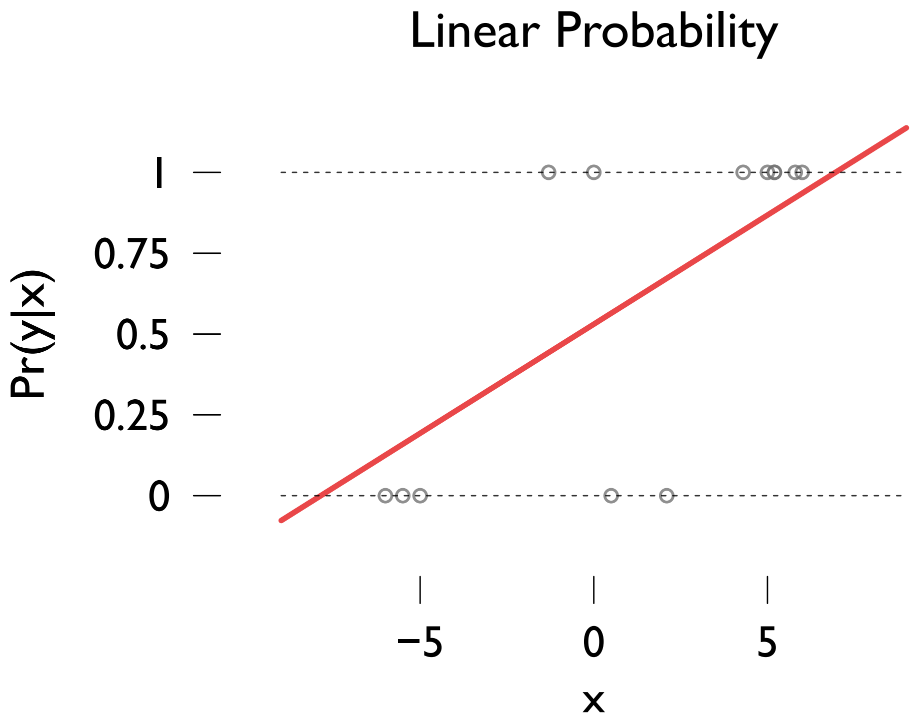
- Event history models
- Network models
- Models of sample selection
- Models of censoring
- Causal models

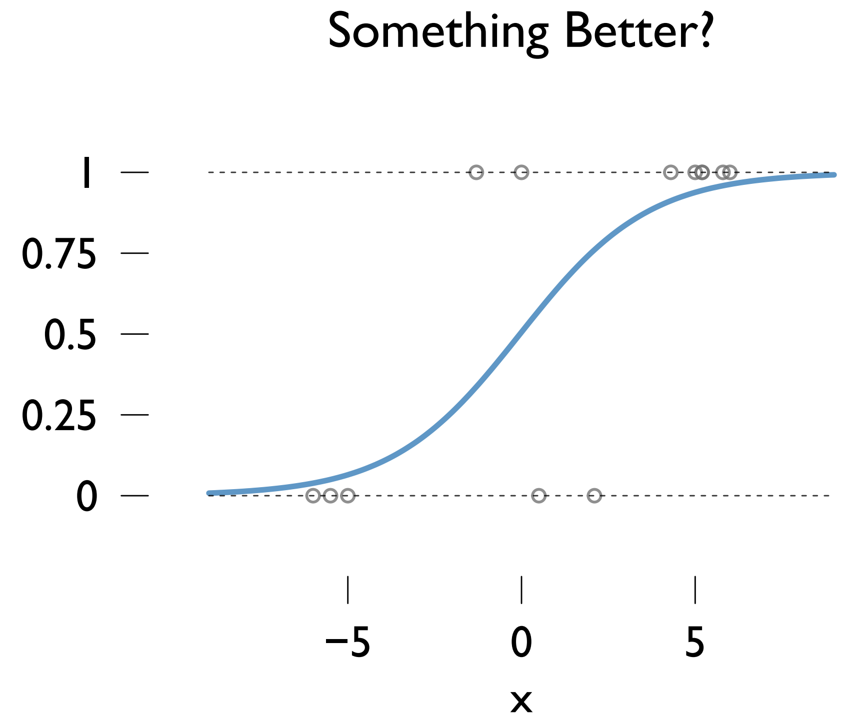
Finally, models of binary data are the easiest way
to understand the *modus operandi* of maximum likelihood




Once upon a time, most people used linear regression to model 1s & 0s

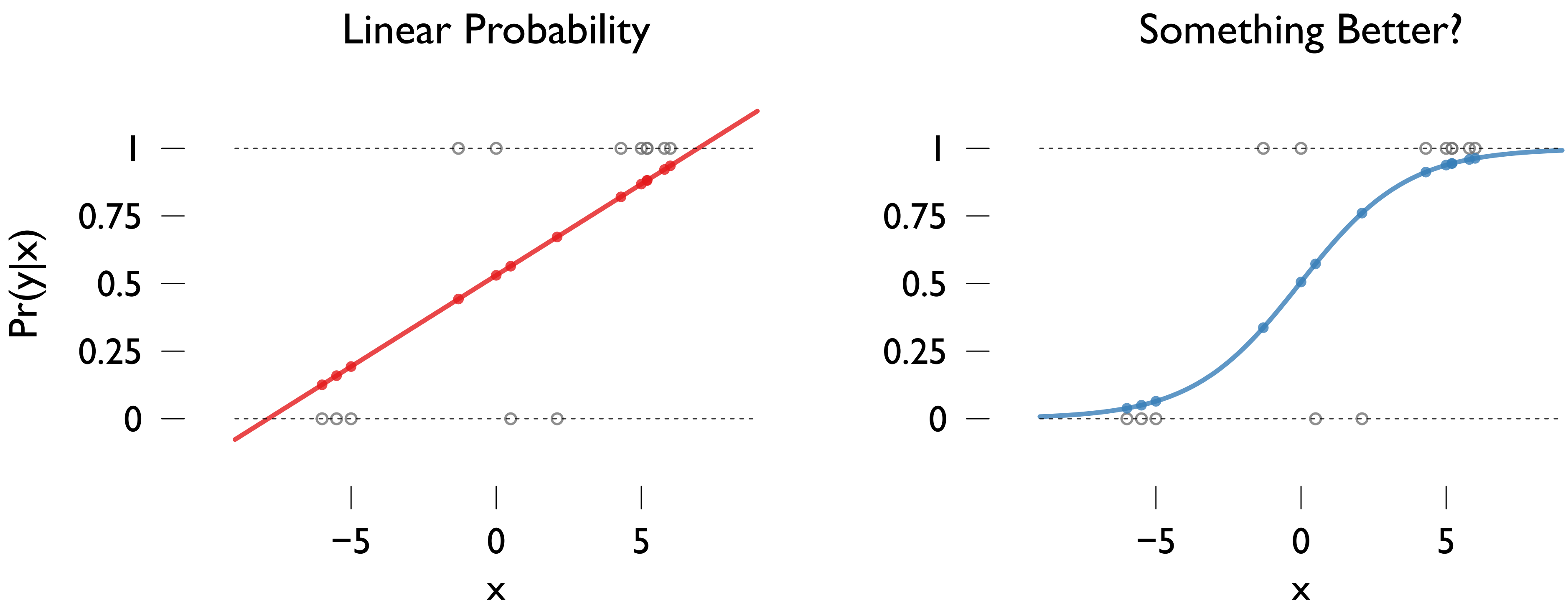
Justification: LS is unbiased, as shown by Gauss-Markov theorem




But three big problems:

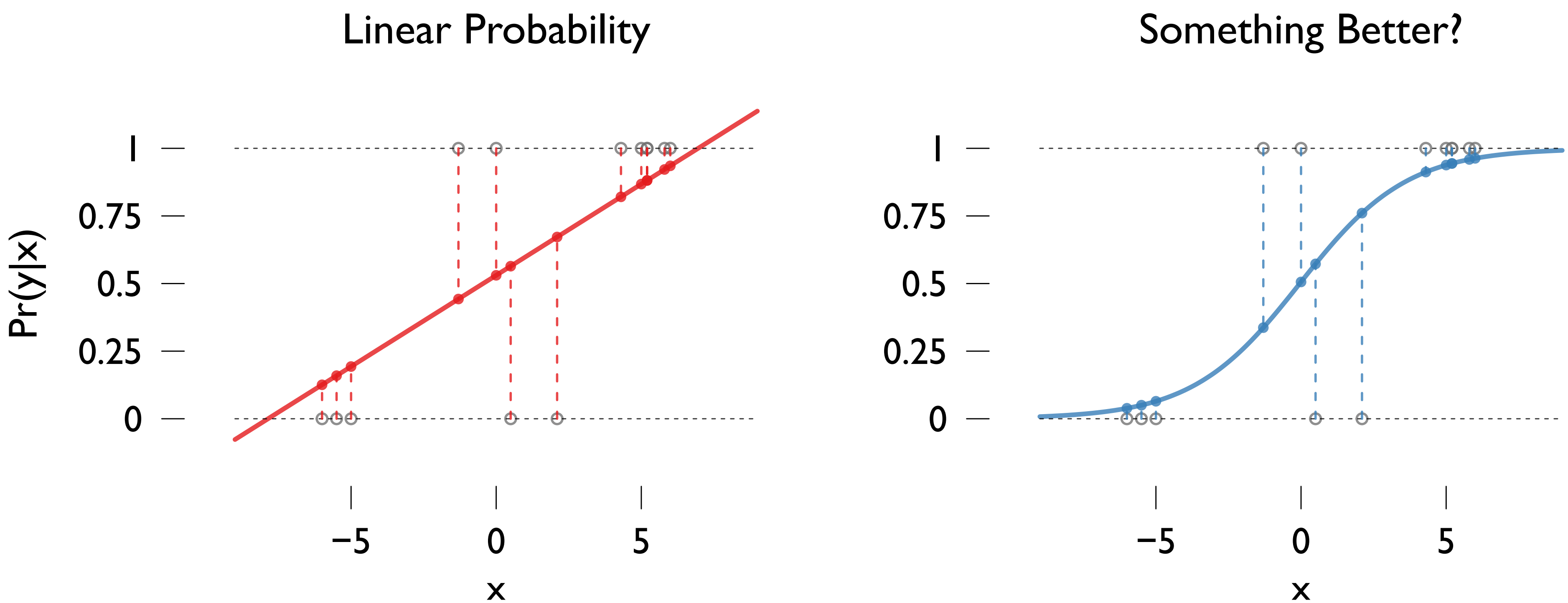
1. Functional form (poor fit)




But three big problems:

1. Functional form (poor fit)

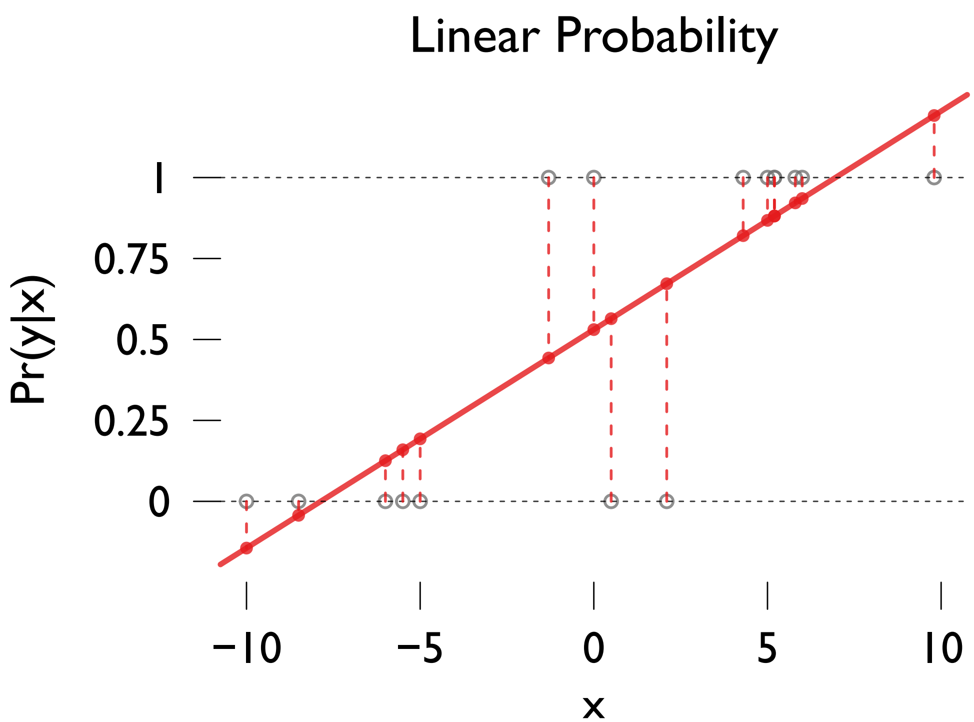
2. Massive heteroskedasticity (inefficiency)

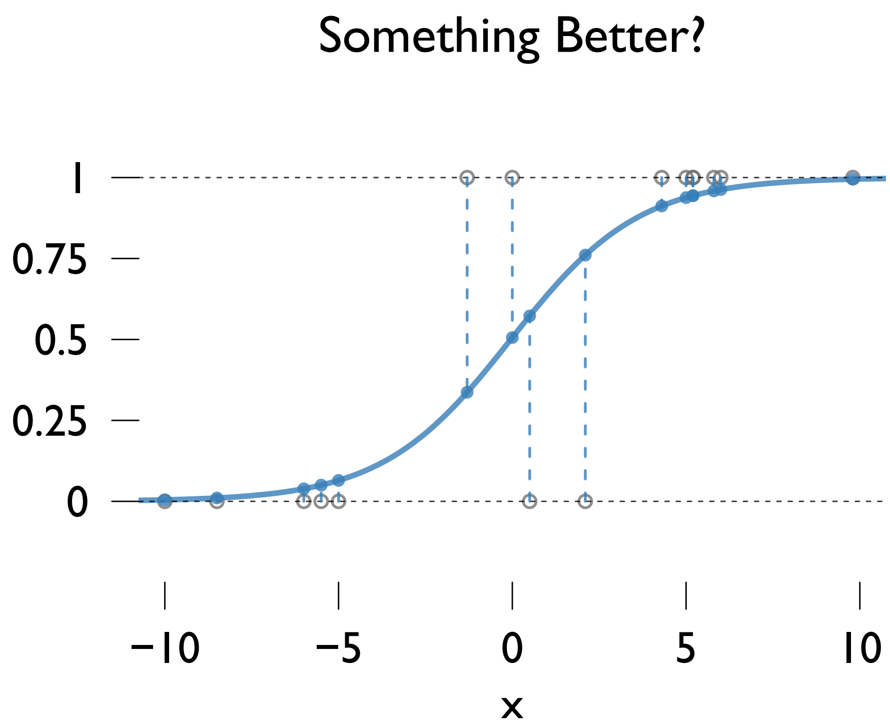



But three big problems:

1. Functional form (poor fit)

2. Massive heteroskedasticity (inefficiency)




But three big problems:

1. Functional form (poor fit)

2. Massive heteroskedasticity (inefficiency)

3. Impossible predictions (*prima facie* inappropriate)

Unless your *only* concern is unbiasedness,
the "linear probability model" is inappropriate for binary data

# A sensible model of binary data

iid binary data precisely fit the assumptions of the Bernoulli distribution

Recall the Bernoulli had the following pdf

$$\Pr(y_i = 1|\pi_i) = f_{\text{Bernoulli}}(y_i|\pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

where

$$\mathbb{E}(y_i) = \pi_i$$

We can easily form the likelihood from the joint probability:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\pi}|\mathbf{y}) &\propto \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} \\ \log \mathcal{L}(\boldsymbol{\pi}|\mathbf{y}) &\propto \sum_{i=1}^{n} y_i \log \pi_i + (1-y_i)\log(1-\pi_i) \end{aligned}$$

A simple likelihood, easy to maximize numerically – approximately quadratic

# Systematic component of the binary choice model

$$\begin{aligned} y_i &\sim \text{Bernoulli}(y_i|\pi_i) \\ \pi_i &= g(\mathbf{x}_i\boldsymbol{\beta}) \end{aligned}$$

What is the functional form of $\pi$? That is, what should we choose for $g(\cdot)$?

- $\pi$ is bounded $[0, 1]$
- An S-curve is an attractive option
- Any $\mathbf{x}_i\boldsymbol{\beta}$ in $[-\infty, \infty]$ then leads to a $\pi$ in $[0, 1]$
- Exact choice of a function is arbitrary
- cdfs are often S-curves, so let's try some of them

# Systematic component: logit

A popular choice: the cdf of the standard logistic distribution

$$\pi_i = \text{logit}^{-1}(\mathbf{x}_i\boldsymbol{\beta}) = \frac{\exp(\mathbf{x}_i\boldsymbol{\beta})}{1+\exp(\mathbf{x}_i\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{x}_i\boldsymbol{\beta})}$$

This leads to the *logit model*, which is very mathematically tractable

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\pi}|\mathbf{y}) &\propto \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} \\
\mathcal{L}(\boldsymbol{\beta}|\mathbf{y}) &\propto \prod_{i=1}^{n} \left(\frac{1}{1+\exp(-\mathbf{x}_i\boldsymbol{\beta})}\right)^{y_i} \left(1-\frac{1}{1+\exp(-\mathbf{x}_i\boldsymbol{\beta})}\right)^{1-y_i} \\
\mathcal{L}(\boldsymbol{\beta}|\mathbf{y}) &\propto \prod_{i=1}^{n} (1+\exp(-\mathbf{x}_i\boldsymbol{\beta}))^{-y_i} (1+\exp(\mathbf{x}_i\boldsymbol{\beta}))^{-(1-y_i)} \\
\log \mathcal{L}(\boldsymbol{\beta}|\mathbf{y}) &\propto \sum_{i=1}^{n} -y_i \log(1+\exp(-\mathbf{x}_i\boldsymbol{\beta})) - (1-y_i)\log(1+\exp(\mathbf{x}_i\boldsymbol{\beta}))
\end{aligned}$$

# Systematic component: probit

Back to models of binary outcomes. . .

Another popular choice: the cdf of the standard Normal distribution

$$\begin{aligned}\pi_i &= \int_{-\infty}^{0} (2\pi)^{-1/2} \exp\left(-\frac{1}{2}(y_i^* - \mathbf{x}_i\boldsymbol{\beta})^2\right) \mathrm{d}y_i^* \\ &= \Phi(\mathbf{x}_i\boldsymbol{\beta}) = \mathrm{probit}^{-1}(\mathbf{x}_i\boldsymbol{\beta})\end{aligned}$$

The probit is the inverse of the Normal cdf, and *vice versa*

As with logit,
"probit" can refer to a function or a model based on that function

The Normal cdf is not analytically defined but can be approximated quickly

# Systematic component: probit

Another popular choice: the cdf of the standard Normal distribution

$$\pi_i = \Phi(\mathbf{x}_i\boldsymbol{\beta}) = \text{probit}^{-1}(\mathbf{x}_i\boldsymbol{\beta})$$

How do we derive an MLE for probit?

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\pi}|\mathbf{y}) &\propto \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} \\
\mathcal{L}(\boldsymbol{\beta}|\mathbf{y}) &\propto \prod_{i=1}^{n} \Phi(\mathbf{x}_i\boldsymbol{\beta})^{y_i}\left(1-\Phi(\mathbf{x}_i\boldsymbol{\beta})\right)^{1-y_i} \\
\log\mathcal{L}(\boldsymbol{\beta}|\mathbf{y}) &\propto \sum_{i=1}^{n} y_i \log\Phi(\mathbf{x}_i\boldsymbol{\beta}) + (1-y_i)\log\left(1-\Phi(\mathbf{x}_i\boldsymbol{\beta})\right)
\end{aligned}$$

Use `pnorm()` in `R` to calculate $\Phi(\cdot)$

# Systematic component: choices. . .

We could continue using still more mathematical function for S-curves

Before we do, let's think about how we choose a specific S-curve for our data

Two questions:

1. Can we justify the choice of a particular systematic component?

2. Does it make a difference which systematic component we choose?

# Latent variables justification

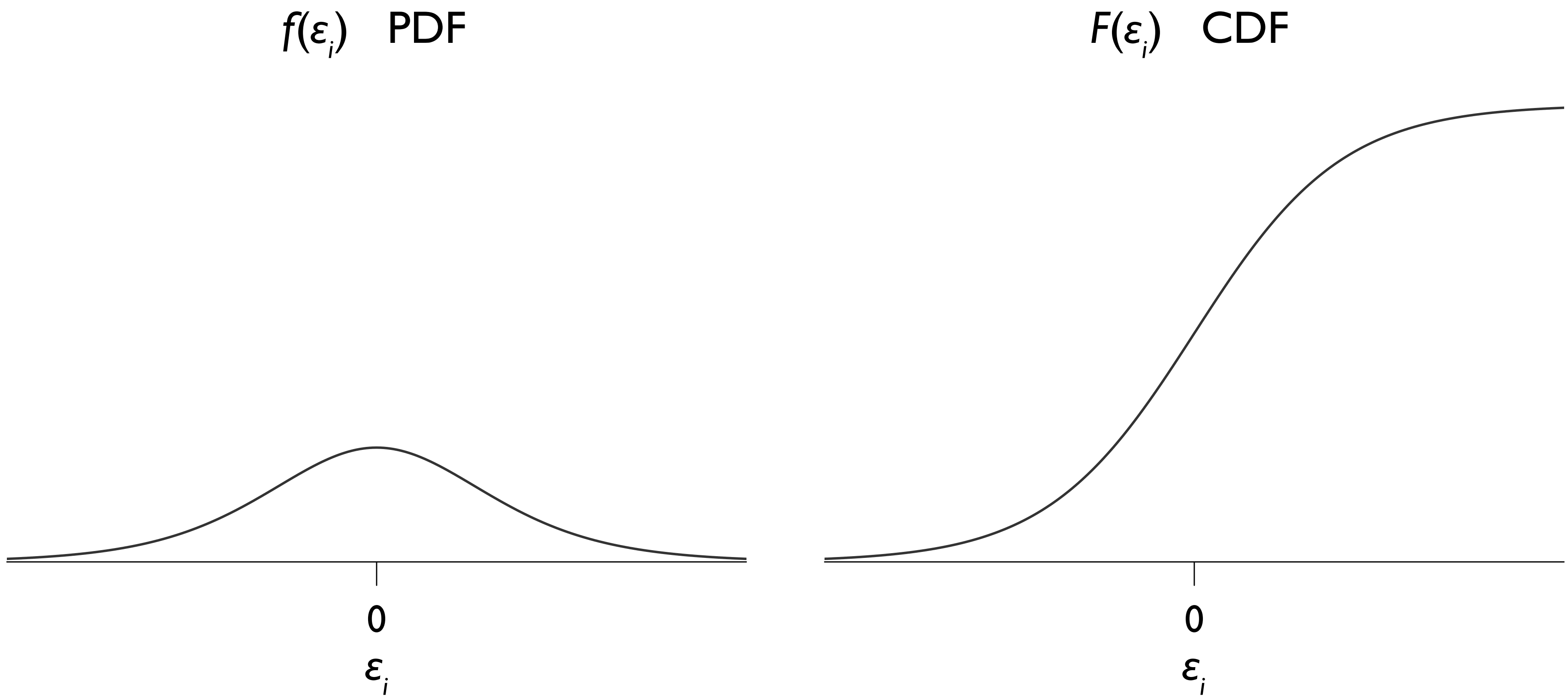
One justification for choosing a particular function form:

Concepts are seldom really binary "underneath"

Often study a continuous concept with imperfect binary measurement

| Underlying concept | Observable outcomes |
|---|---|
| Insect health | alive or dead |
| International tension | war or peace |
| Conservatism | *yea* or *nay* on roll call |
| Aptitude | Correct or incorrect exam answer |

In each case, we observe binary realizations $y_i$ of an underlying (possibly unmeasurable) continuous variable $y_i^*$




If we observed $y_i^*$ we could use a linear model

$$y_i^* = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i$$

where $\varepsilon_i$ is a random variable following any *symmetrical* distribution (possibly but not necessarily the Normal distribution)

The mean of $\varepsilon_i$ is zero, and the variance is fixed

# Latent variables justification

If we observed $y_i^*$ we could use a linear model

$$y_i^* = \mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i$$

where $\varepsilon_i$ is a random variable following any *symmetrical* distribution (possibly but not necessarily the Normal distribution)

The mean of $\varepsilon_i$ is zero, and the variance is fixed

We don't observe $y_i^*$, but we assume it created $y_i$ as follows:

$$y_i = \begin{cases} 1 & \text{if } y_i^* > \tau \\ 0 & \text{if } y_i^* \leq \tau \end{cases}$$

In this setup, $\tau$ is a *cutpoint* along the scale of $y_i^*$

We usually don't know where $\tau$ is

# Latent variables justification

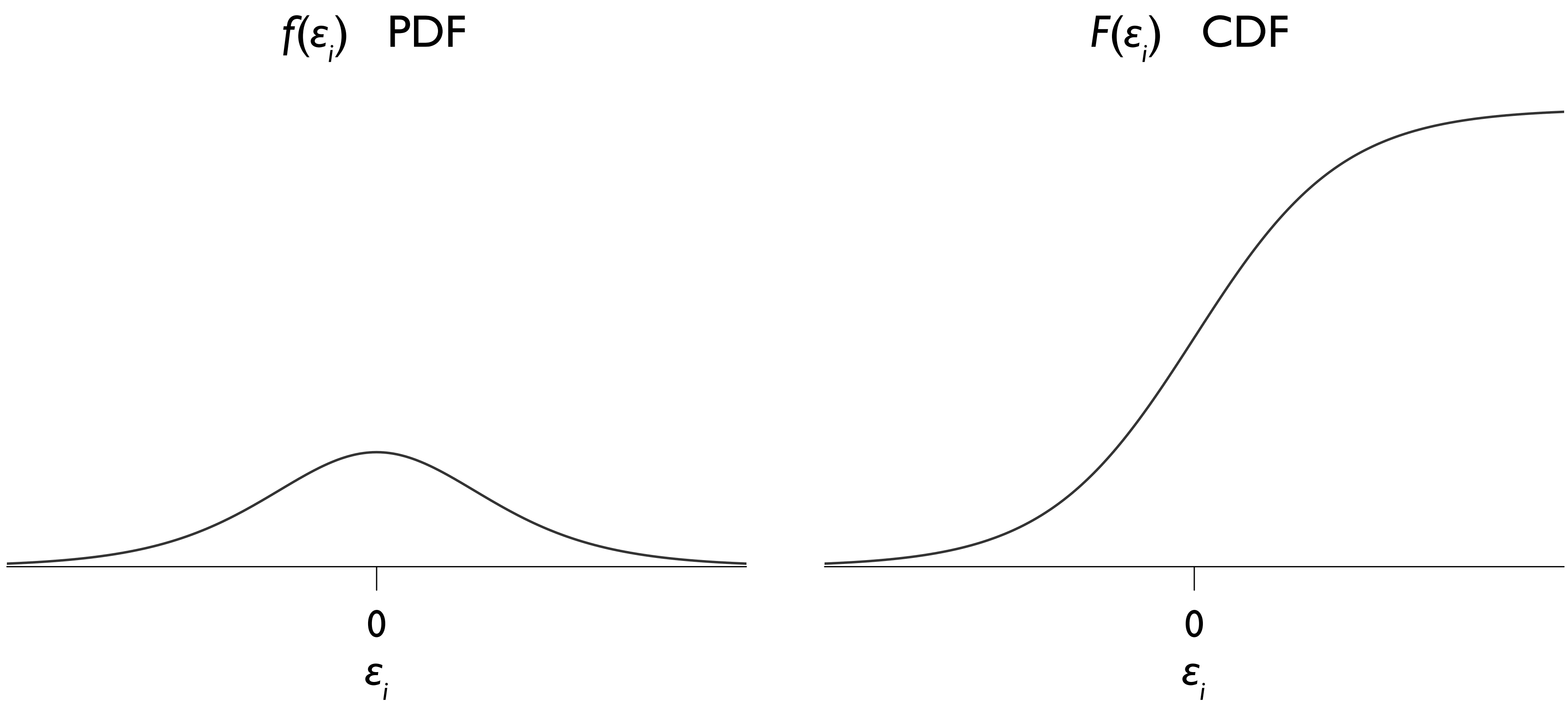
For identification and without loss of generality, assume $\tau = 0$, so that

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \leq 0 \end{cases}$$

In a sense, regression with $y_i$ is a "degenerate" form of linear regression where we observe only the sign of the original dependent variable $y_i^*$

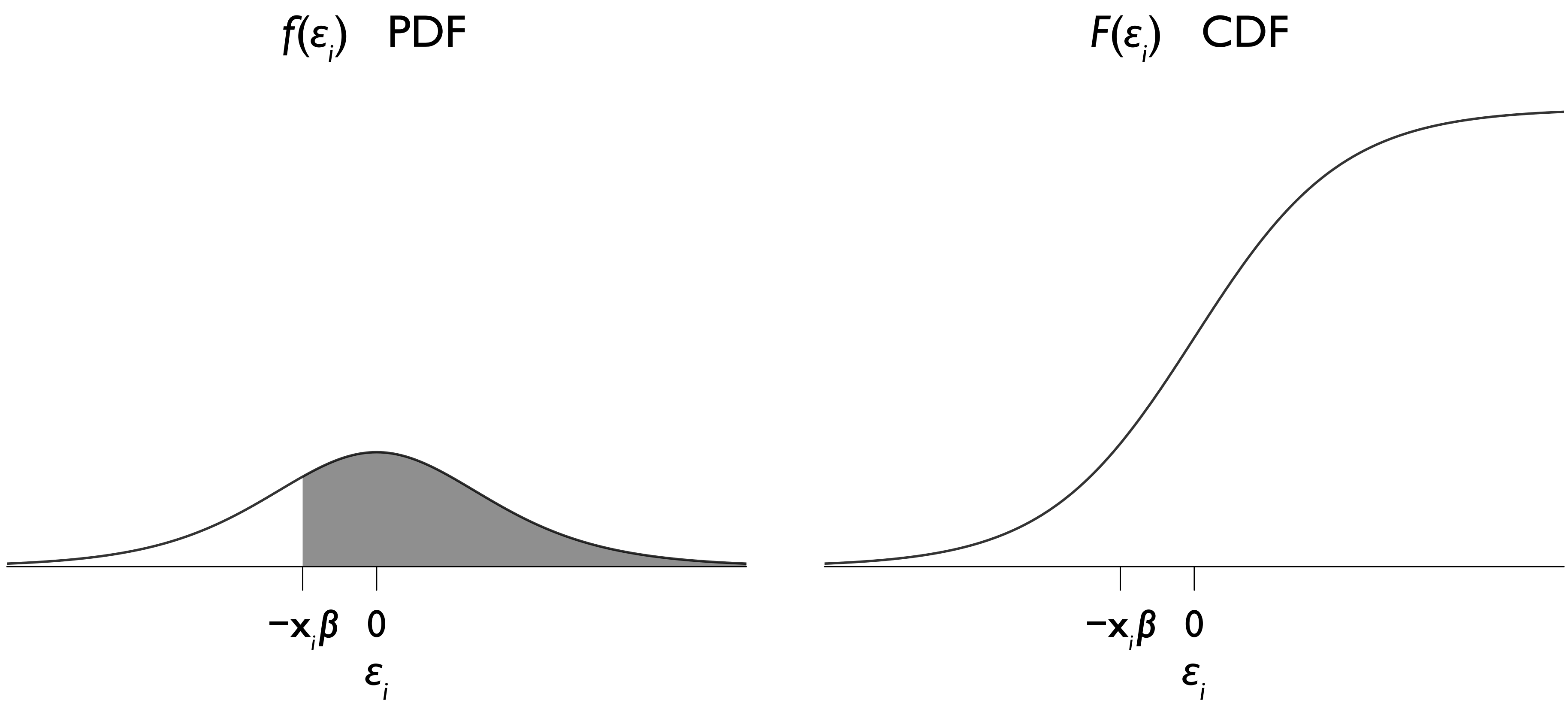
Aside: this implies binary outcome variables contain less *information* per observation than any other kind of outcome variable

With a binary outcome we need much larger $N$ to precisely estimate parameters, compared to linear regression



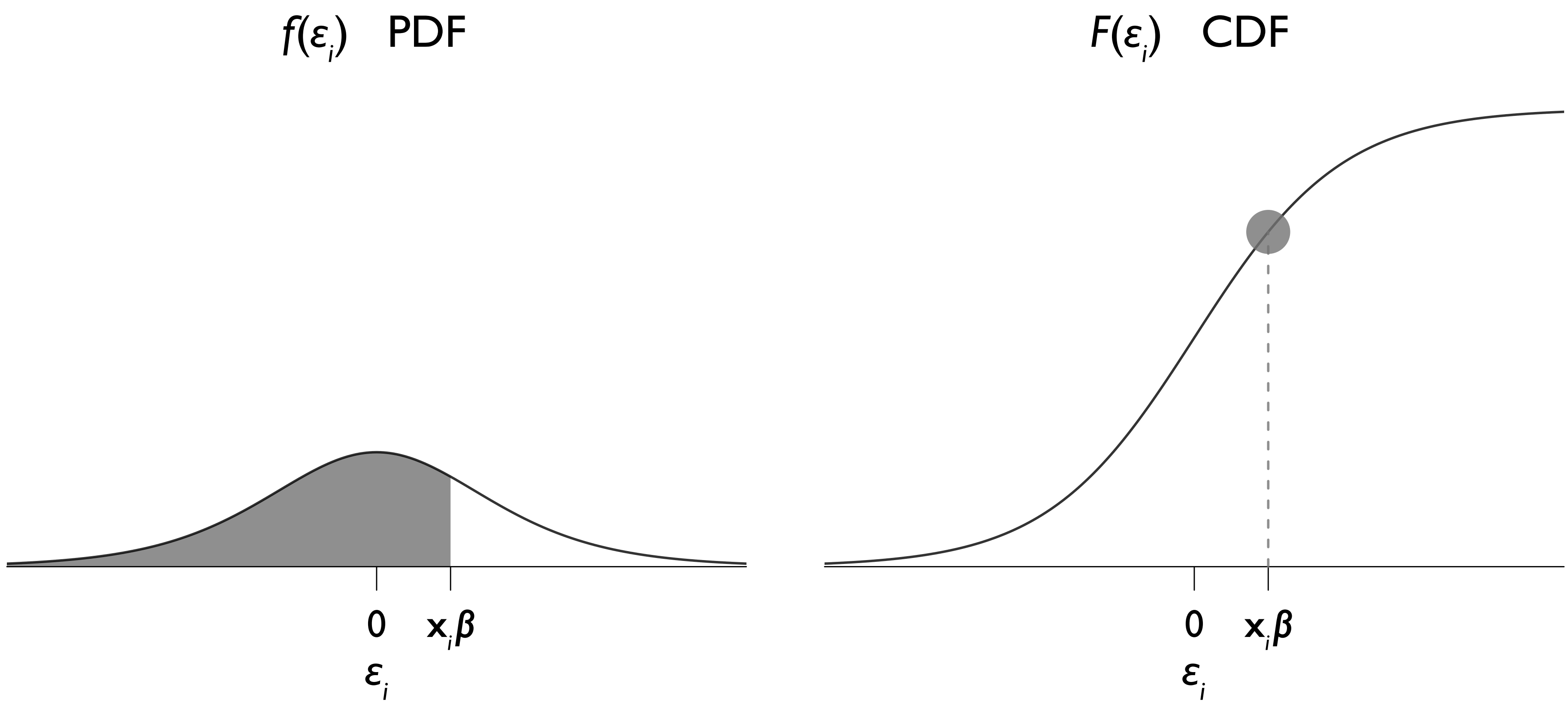

Can we derive an estimator based on the latent variable setup?

$$\Pr(y_i^* > 0 | \boldsymbol{\beta}, \mathbf{x}_i) \quad = \quad \Pr(\mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i > 0)$$




Can we derive an estimator based on the latent variable setup?

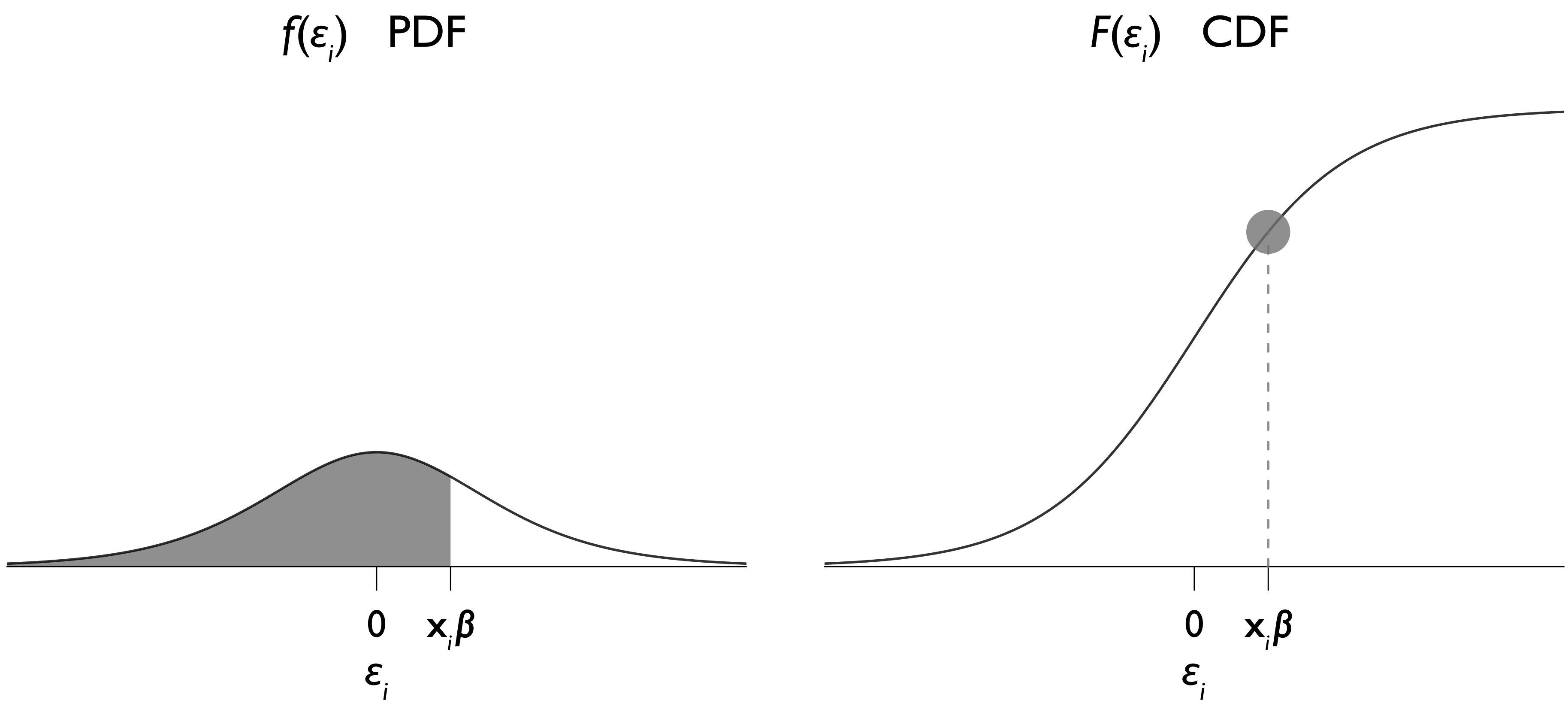
$$
\begin{aligned}
\Pr(y_i^* > 0 | \boldsymbol{\beta}, \mathbf{x}_i) &= \Pr(\mathbf{x}_i \boldsymbol{\beta} + \varepsilon_i > 0) \\
&= \Pr(\varepsilon_i > -\mathbf{x}_i \boldsymbol{\beta})
\end{aligned}
$$




Can we derive an estimator based on the latent variable setup?

$$
\begin{aligned}
\Pr(y_i^* > 0|\boldsymbol{\beta}, \mathbf{x}_i) &= \Pr(\mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i > 0) \\
&= \Pr(\varepsilon_i > -\mathbf{x}_i\boldsymbol{\beta}) \\
&= \Pr(\varepsilon_i < \mathbf{x}_i\boldsymbol{\beta})
\end{aligned}
$$

where the last step follows from the symmetry of $f(\cdot)$




Note that $\Pr(\varepsilon_i < \mathbf{x}_i\boldsymbol{\beta})$ is the definition of a cdf:

$$\begin{aligned}\Pr(y_i^* > 0|\boldsymbol{\beta}, \mathbf{x}_i) &= F(\mathbf{x}_i\boldsymbol{\beta}) \\ \Pr(y_i = 1|\boldsymbol{\beta}, \mathbf{x}_i) &= F(\mathbf{x}_i\boldsymbol{\beta})\end{aligned}$$

So to justify choice of a specific cdf as the systematic component, we could argue for a specific pdf of the latent variable

# Latent variables justification

Suppose we thought the latent variable was Normal

Since we can't observe it, we can assume it is $f_{\mathcal{N}}(0,1)$ w/o loss of generality

This implies the binary variable is related to $\mathbf{x}_i\boldsymbol{\beta}$ through the Normal cdf

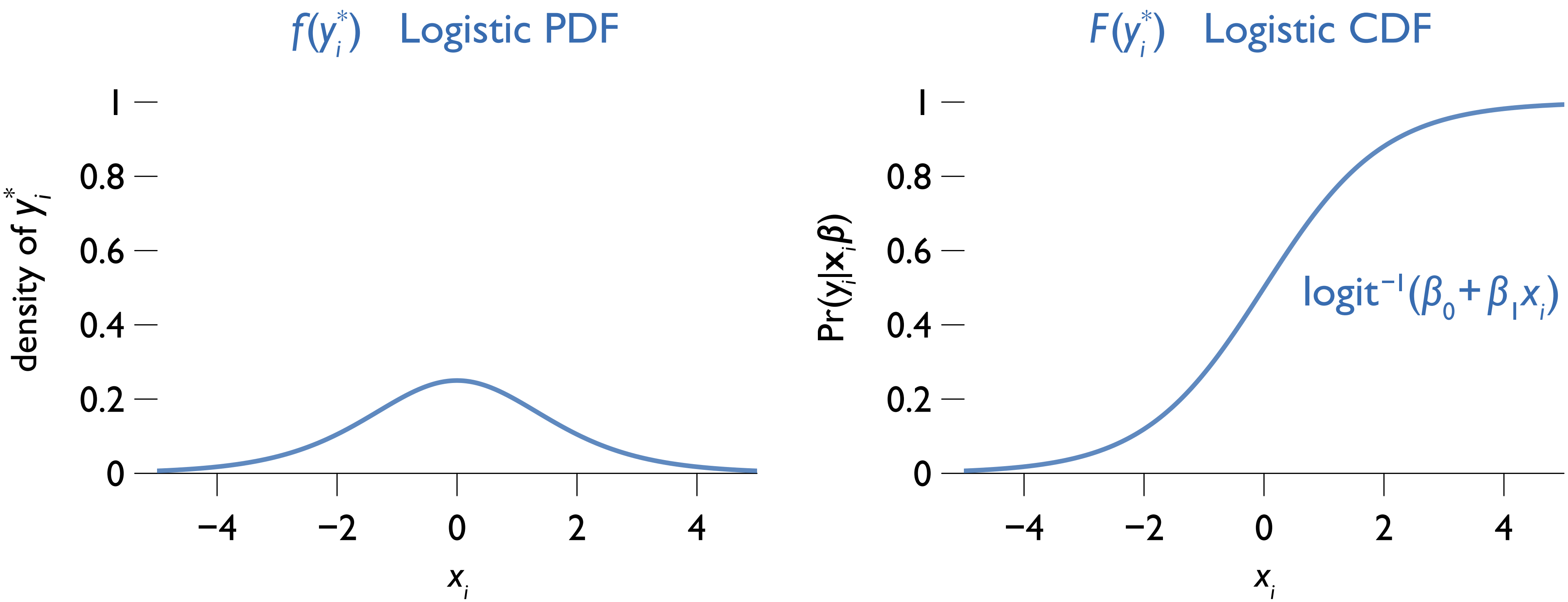
$\Rightarrow$ the Probit model

Or, suppose we assume the latent variable was standard logistic distributed:

$$\begin{aligned} y_i^* &= f_{\text{Standard Logistic}}(\mathbf{x}_i\boldsymbol{\beta}) = \frac{\exp(-\mathbf{x}_i\boldsymbol{\beta})}{[1+\exp(\mathbf{x}_i\boldsymbol{\beta})]^2} \\ \Pr(y_i = 1|\mathbf{x}_i, \boldsymbol{\beta}) &= F_{\text{Standard Logistic}}(\mathbf{x}_i\boldsymbol{\beta}) = \frac{1}{1+\exp(-\mathbf{x}_i\boldsymbol{\beta})} \end{aligned}$$

$\Rightarrow$ The logit model

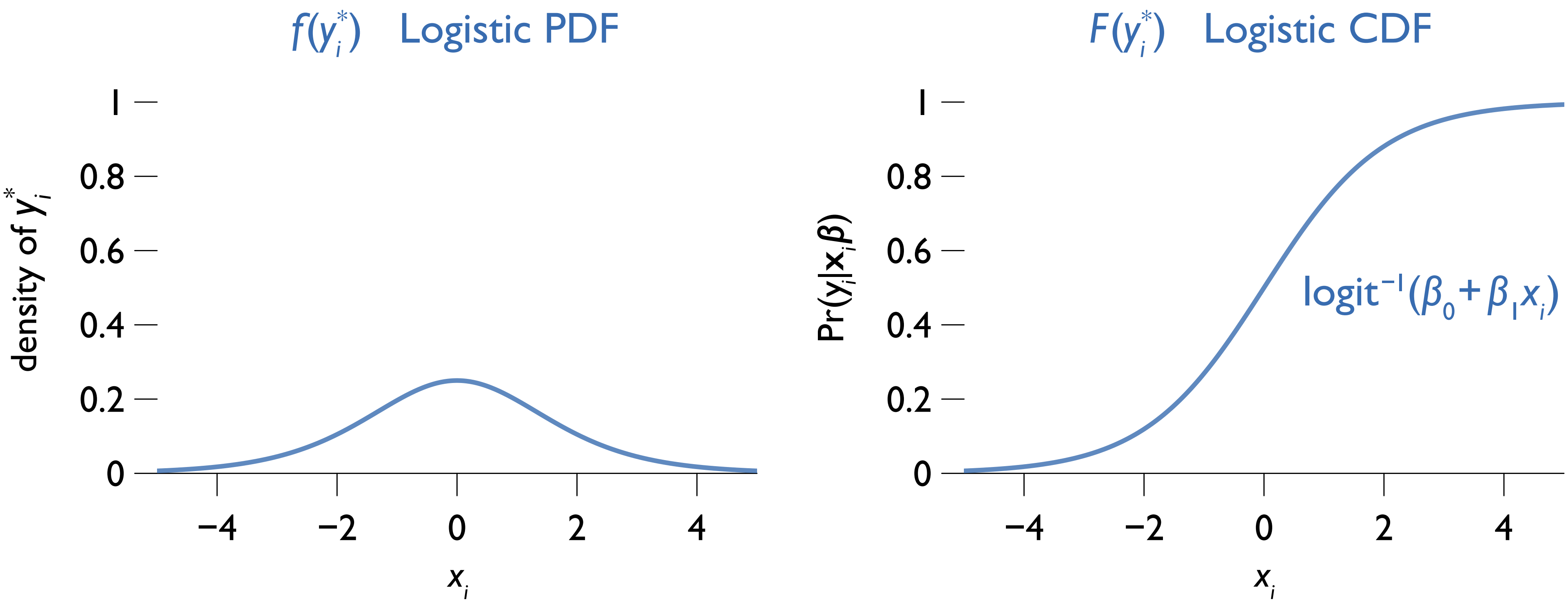
*Does it matter which we choose? Let's look at some S-curves to find out*




The logistic CDF – which happens to be the inverse-logit function – is the most popular systematic component for models of binary data

Above right is a plot of this model: $\text{logit}^{-1}(\mathbf{x}_i\boldsymbol{\beta})$

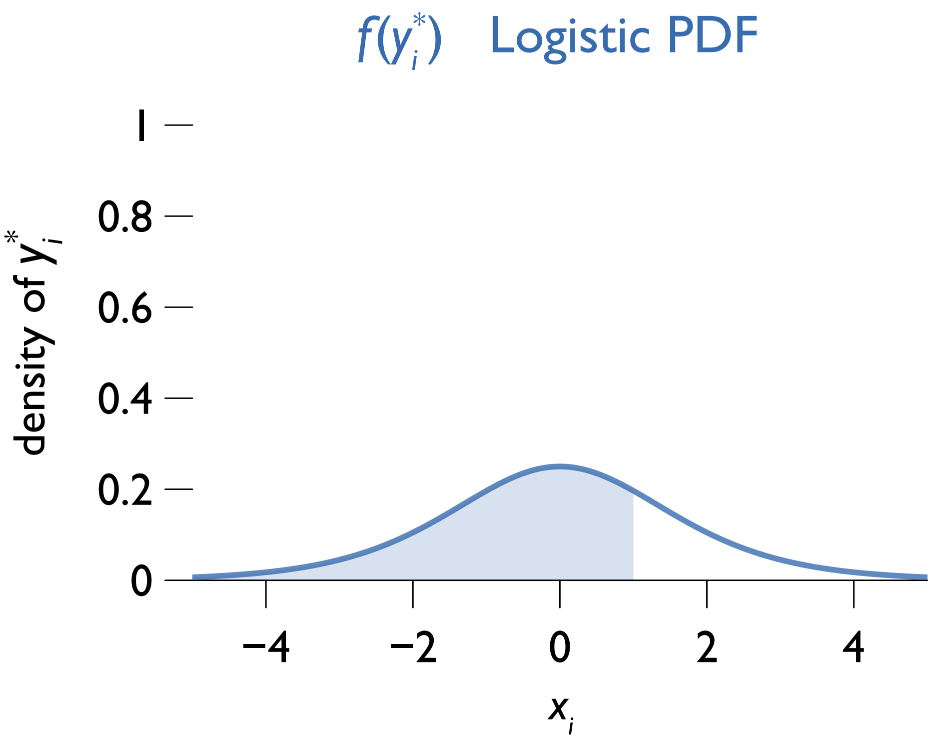
This would be an appropriate model for $\Pr(y_i|x_i\beta)$ if we believed the latent variable $y_i^* = \text{x}_i\boldsymbol{\beta}$ followed a Standard Logistic distribution (left panel)

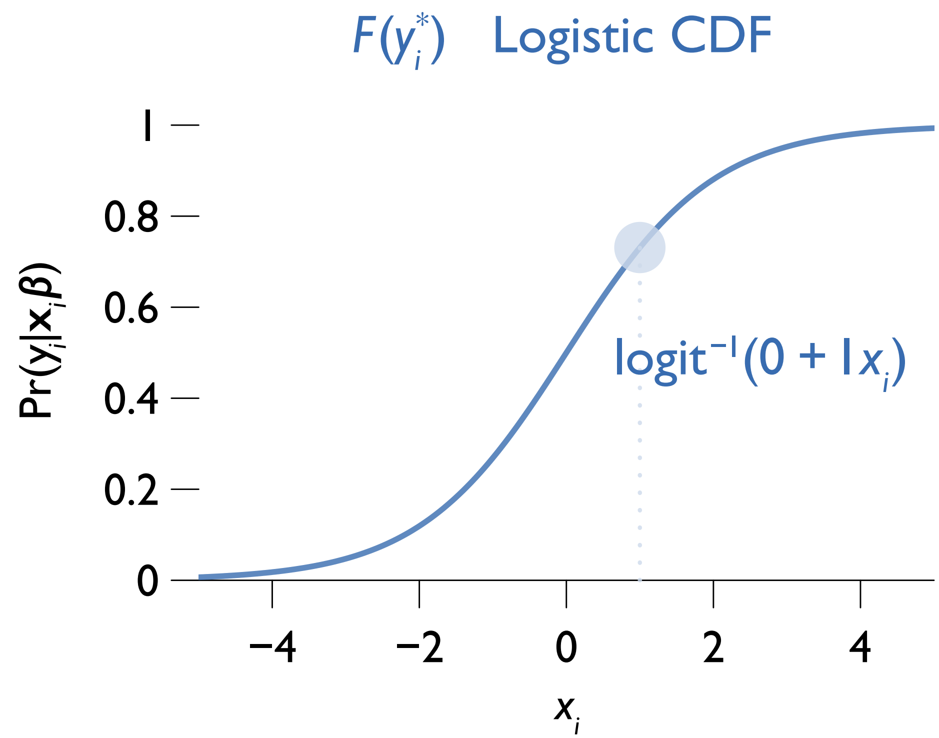



Let's assume we have a single covariate $x_i$ with coefficient $\beta_1$ and a constant $\beta_0$

Thus our systematic component is $\text{logit}^{-1}(\beta_0 + \beta_1 x_i)$

The horizontal axes above show the value of $x_i$
used to compute the latent variable $y_i^* = \beta_0 + \beta_1 x_i$

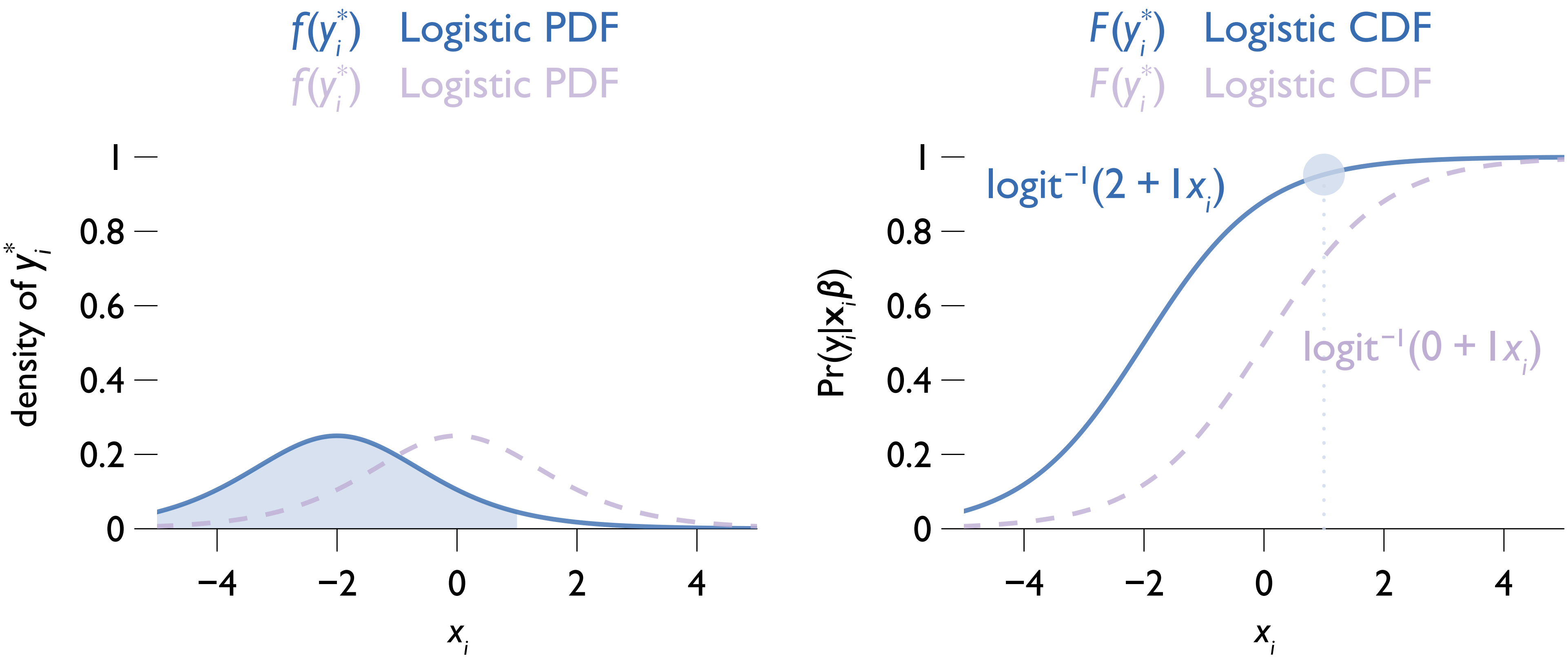



Assume that $\beta_0 = 0$ and $\beta_1 = 1$

Suppose that $x_i = 1$ Now the shaded circle marks the probability that $y_i = 1$

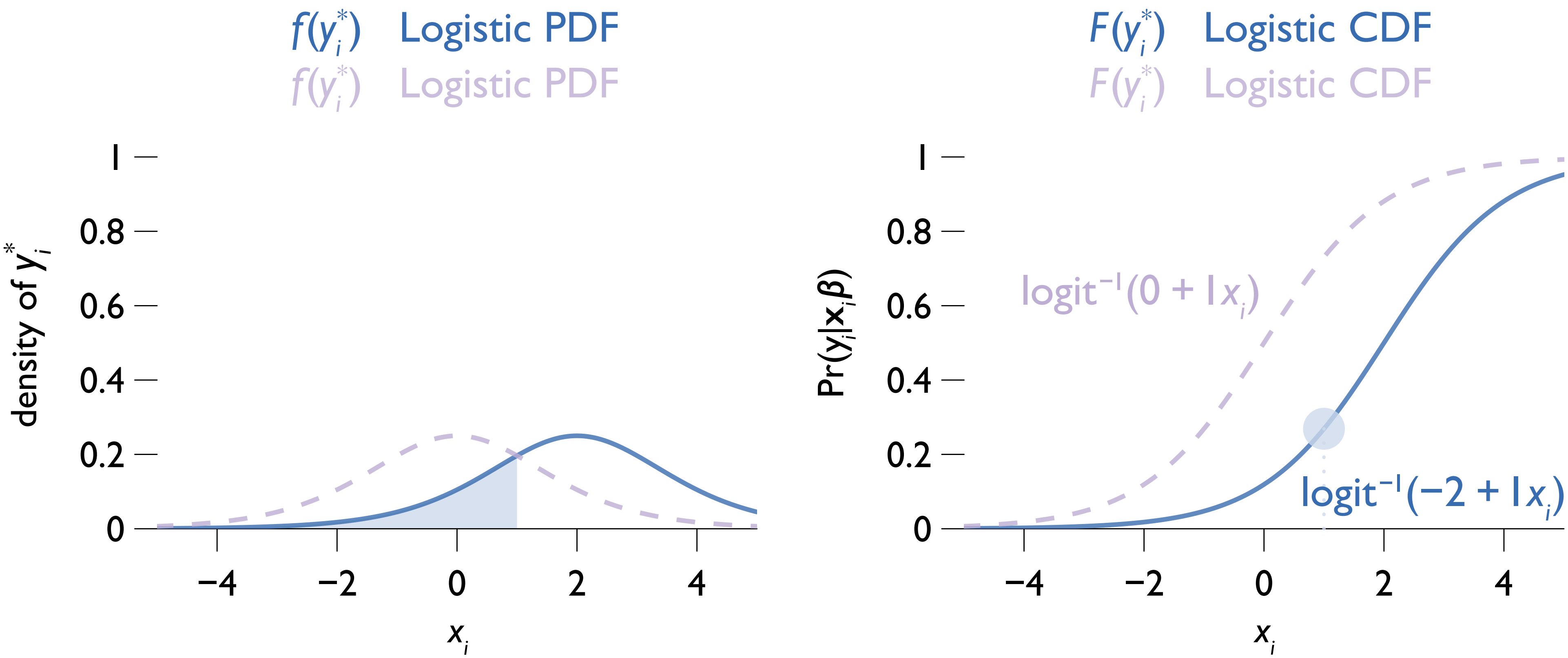
What happens to the logit curve – and this probability – as we vary $\beta_0$ or $\beta_1$?




If we increase $\beta_0$ to 2, then the same value of $x_i$ will lead to a higher value of $y_i^*$

This shifts the logit curve to the left,
analogous to increasing the constant in linear regression

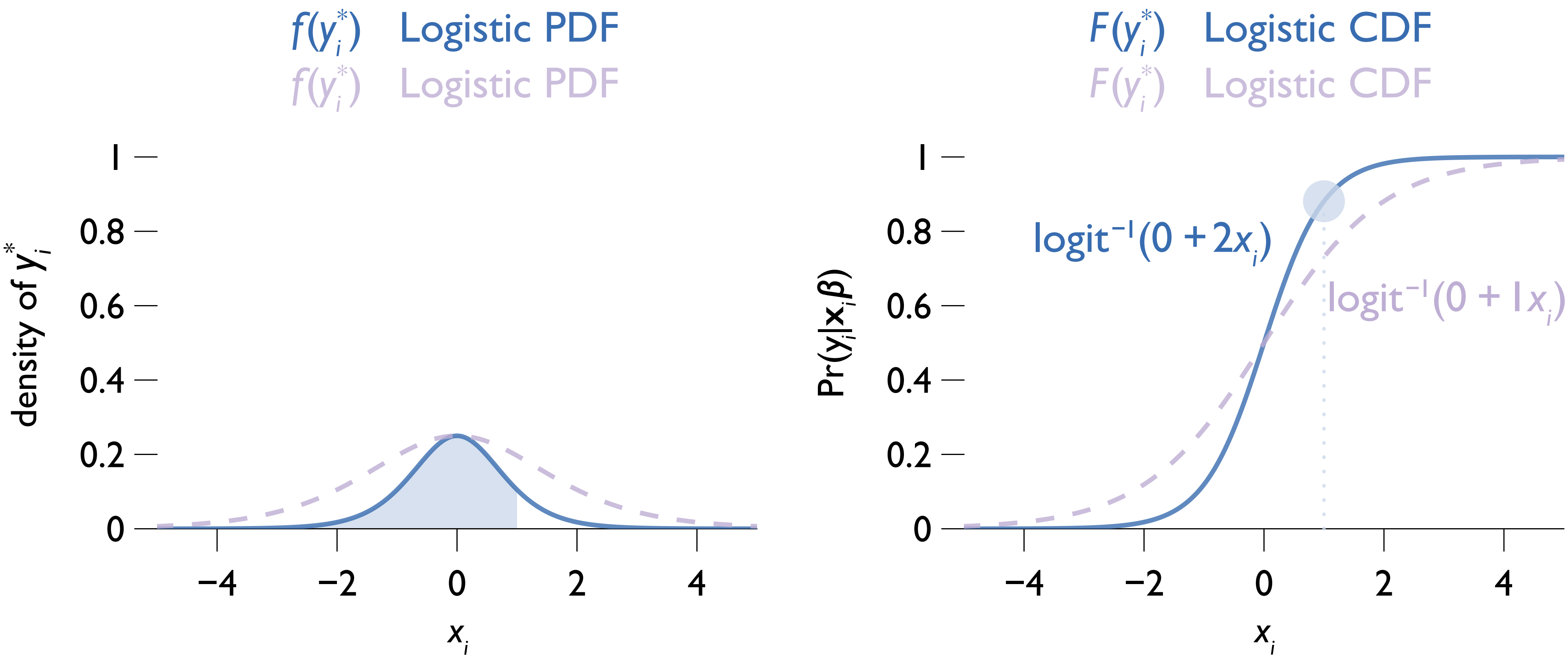
Although the curvature of the logit curve is unchanged,
the change in probability at any given value of $x_i$ is nonlinear




Likewise, if we lower $\beta_0$ to 2,
then the same value of $x_i$ will lead to a lower value of $y_i^*$

This shifts the logit curve to the right,
analogous to decreasing the constant in linear regression

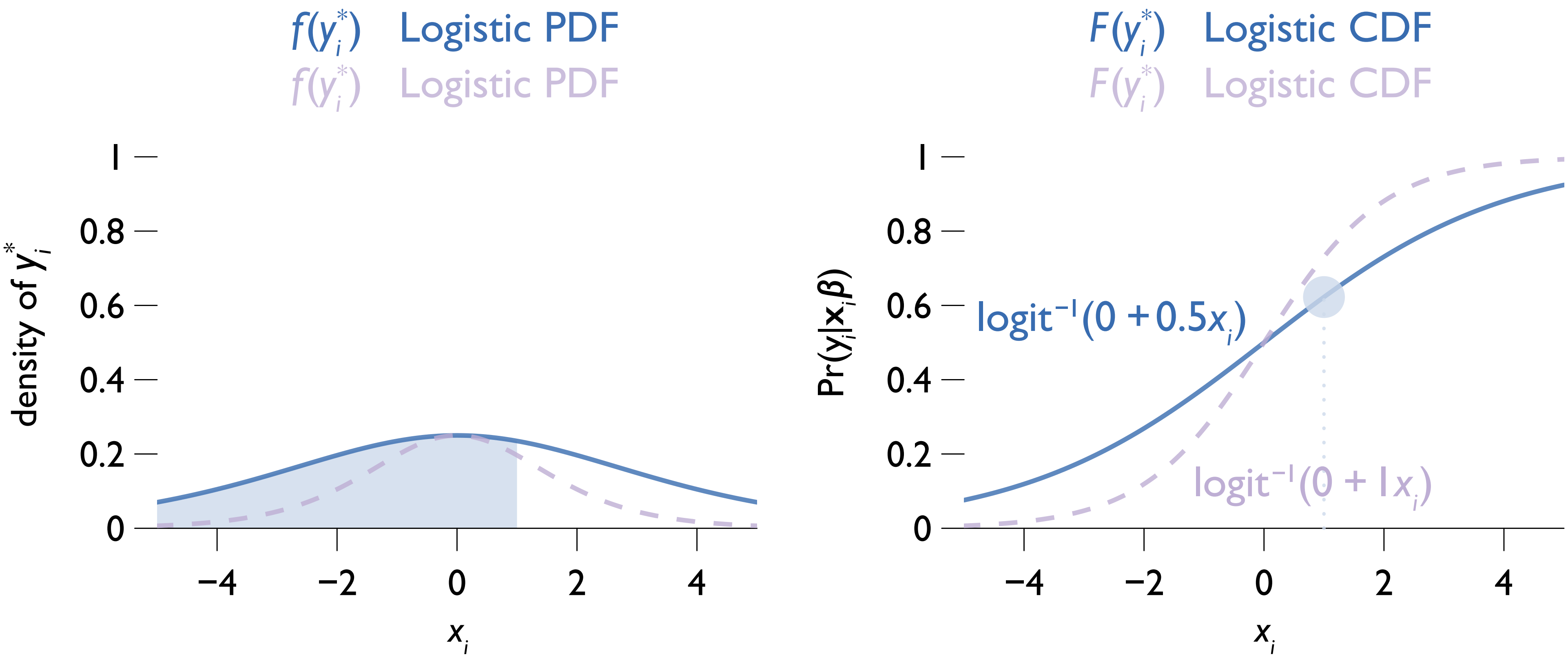
Again, there is no change in curvature




Raising $\beta_1$ from 1 to 2 does change the shape of the curve

It is now steeper, analogous to shifting the slope of a regression line

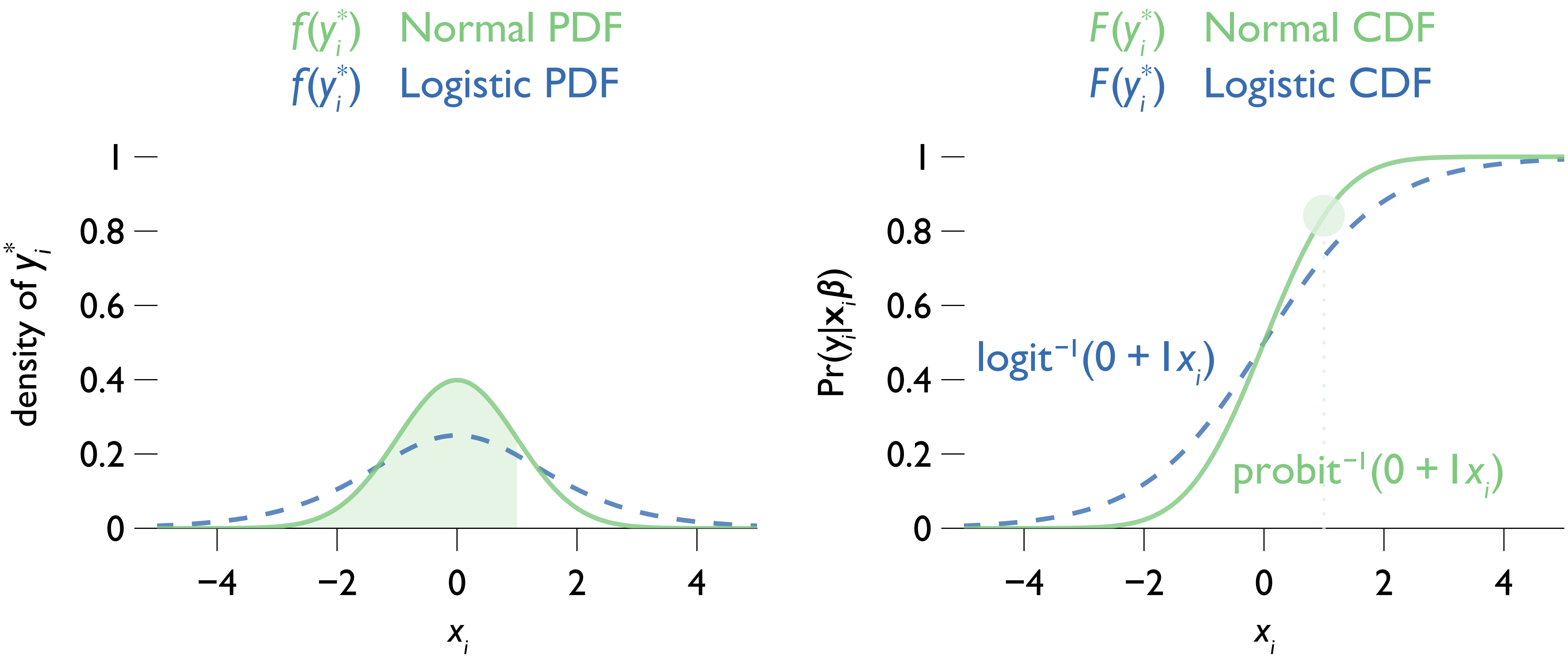
Note the change is nonlinear,
with strongest effects near a probability of 0.5




Lowering $\beta_1$ from 1 to 0.5 flattens the curve

The logit is now a bit more linear

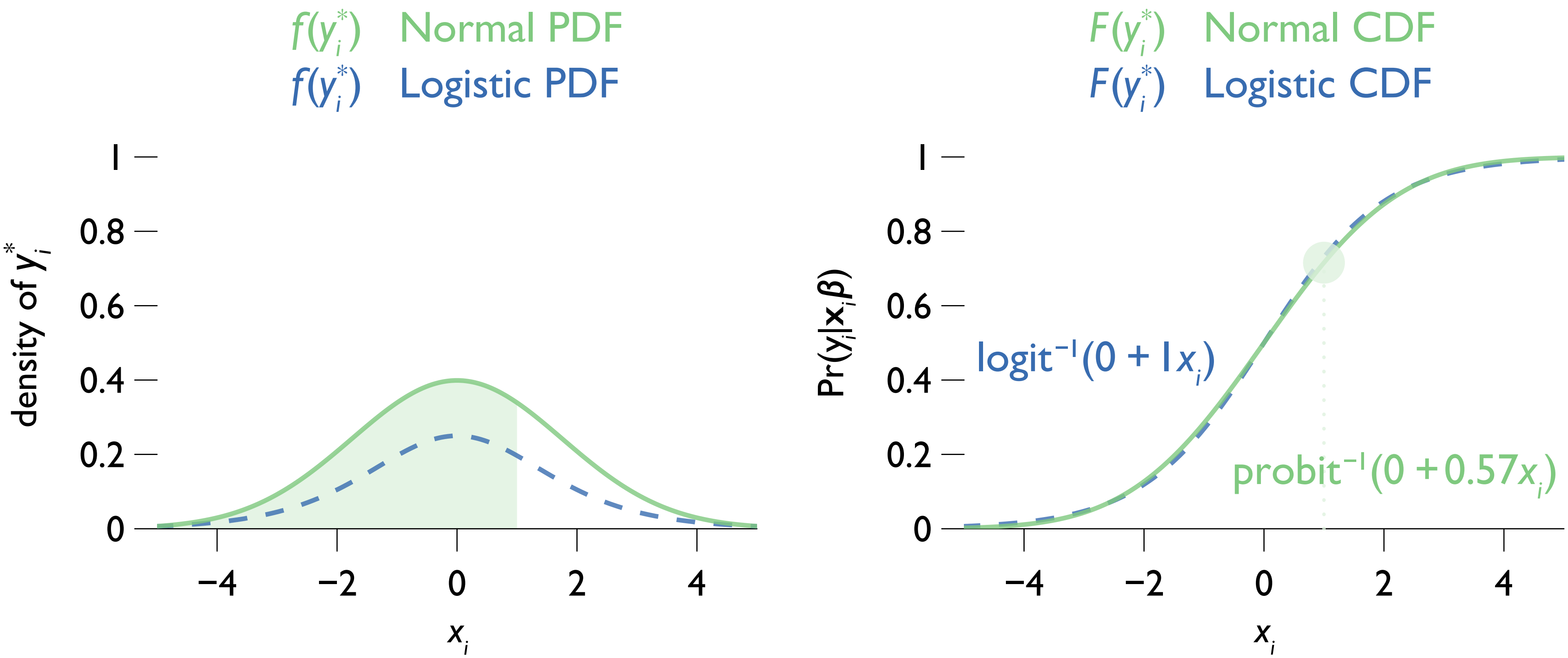
What would it take to make the logit completely linear?




Let's compare logit and probit

Notice the probit has thinner tails than the logit

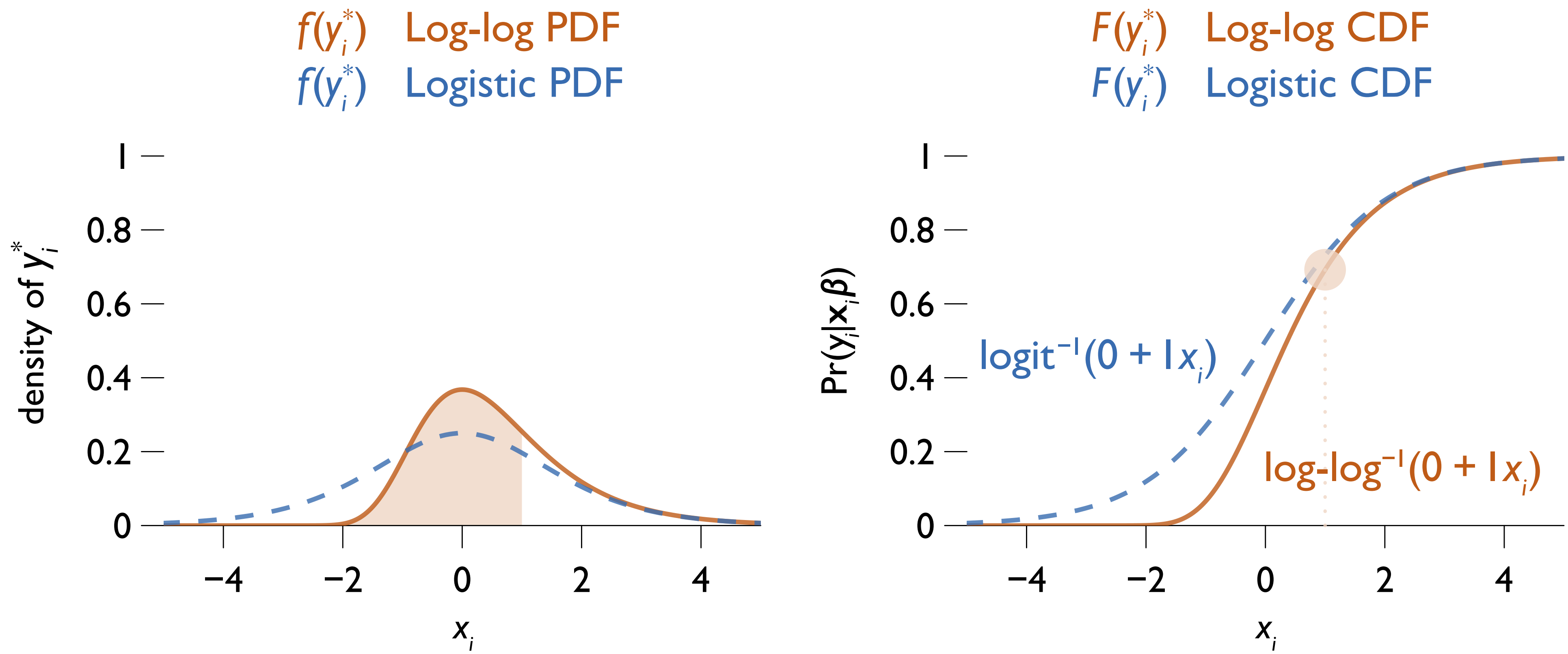
Does this mean there is a big difference between logit and probit models?




*No. There exists a probit curve to closely match any given logit curve*

The MLE will find this probit curve,
so there's only a trivial difference between logit and probit for binary outcomes

If there's no real difference between logit and probit,
does it matter which S-curve we use?

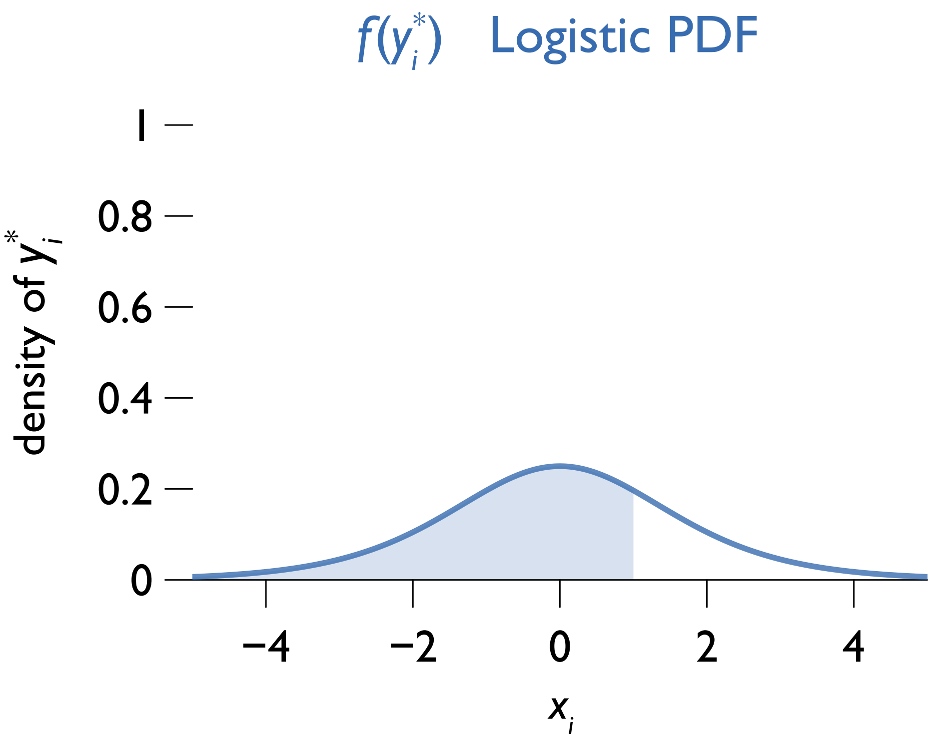


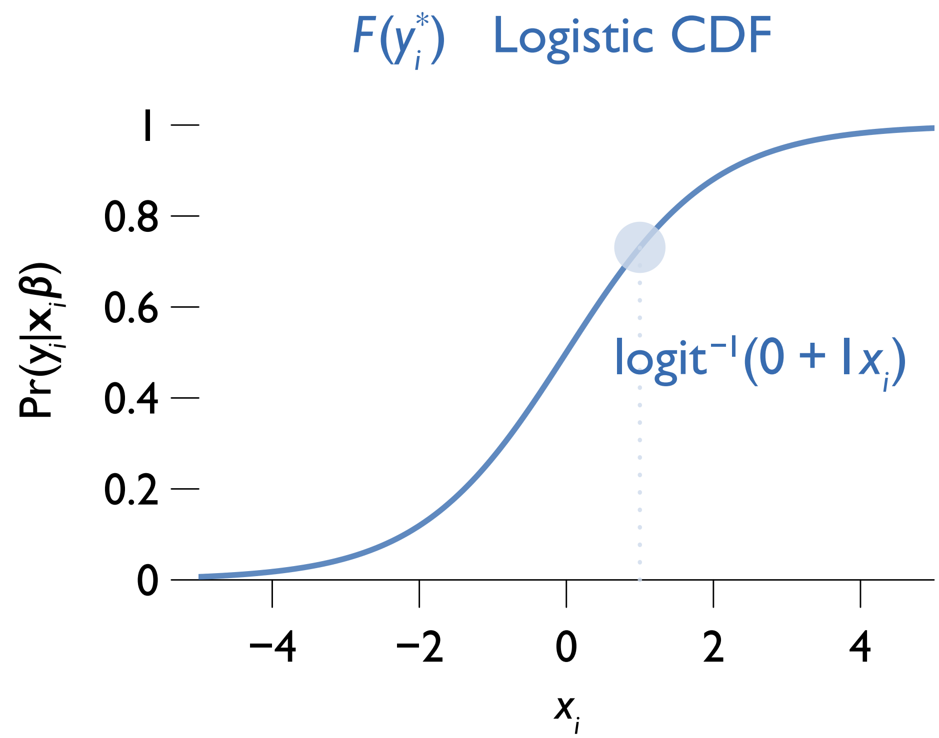

What if we abandon symmetry? (What might this mean substantively?)

If we assume a particular asymmetry in the latent variable, such as shown above, we can obtain an appropriate S-curve

This example is the log-log model $\pi = \exp[-\exp(-\mu)]$

How might we flip this curve?
Use the complementary log-log, $\pi = 1 - \exp[-\exp(\mu)]$ or reverse-code $y_i$

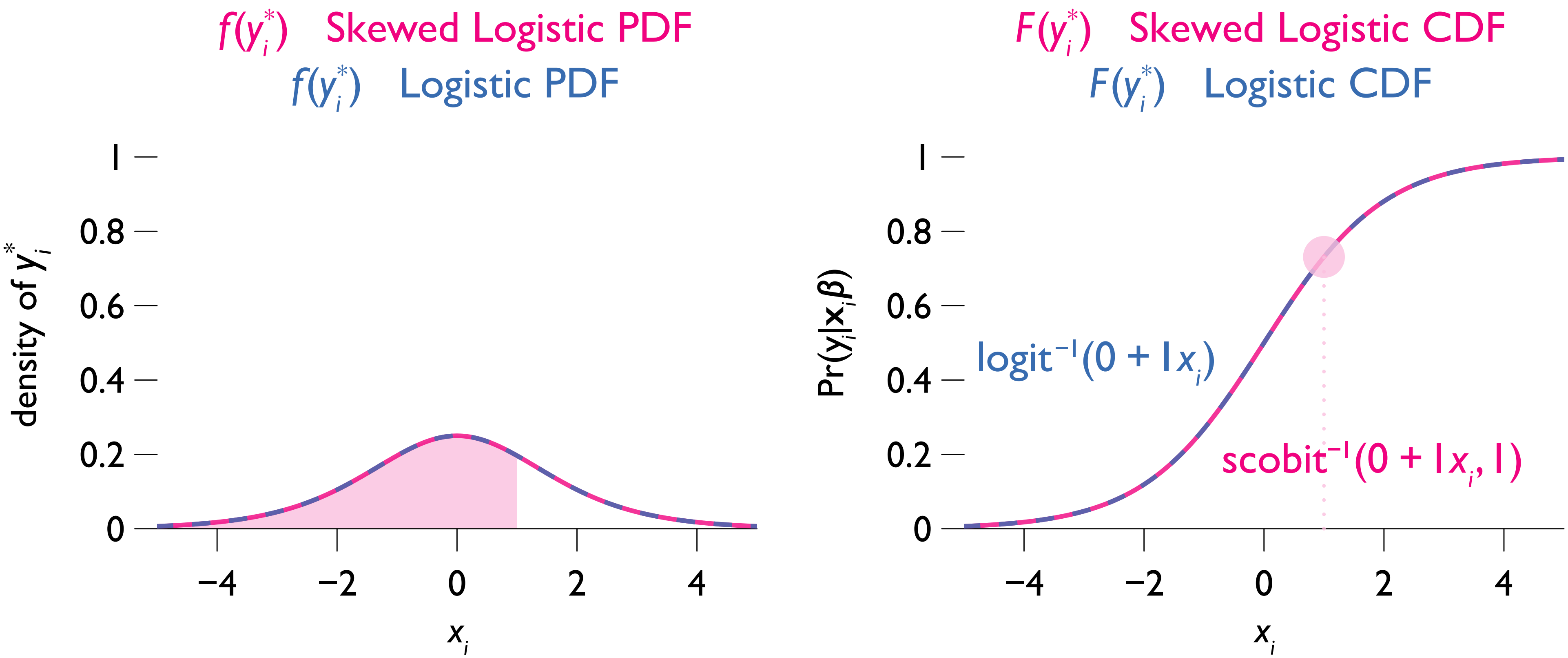



But what if we want to *test* asymmetry, rather than assume it?

Jonathan Nagler proposed "scobit,"
a generalization of logit with a skew parameter $\alpha$:

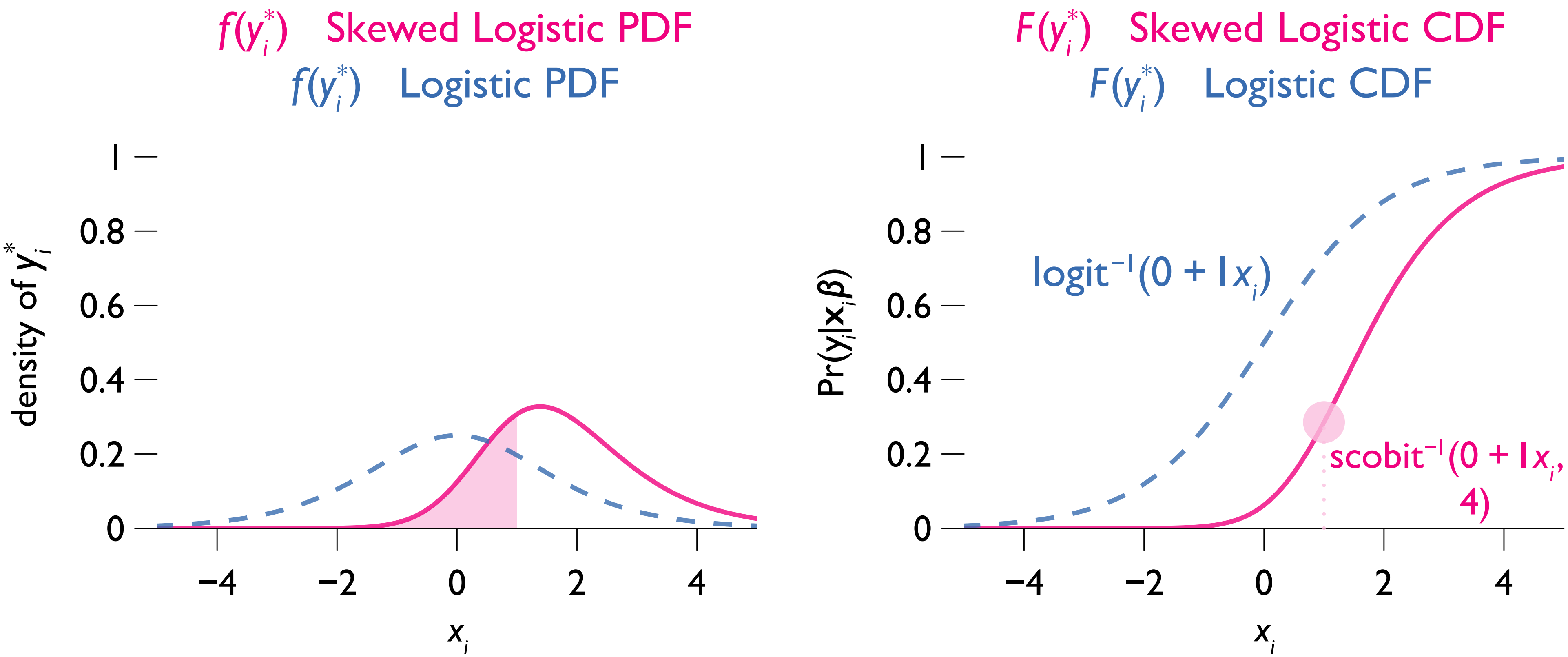
$$1 - \frac{1}{(1 + \exp(\mathbf{x}_i \boldsymbol{\beta}))^{\alpha}}$$




A scobit with $\alpha = 1$ produces logit as a special case

Other $\alpha$'s produce different skews

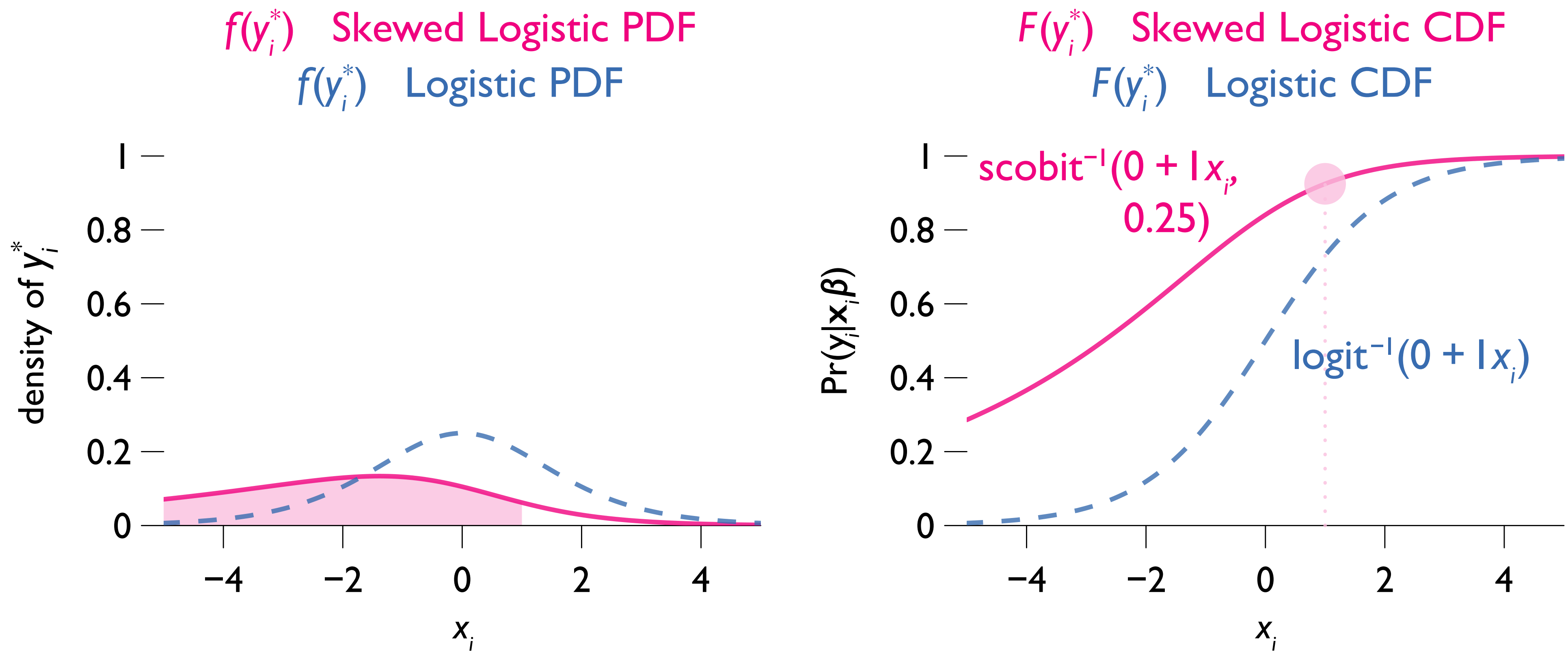
Scobit allows the model to find any asymmetry that may be present




*Here we see $\alpha > 1$ produces a log-log style curve*

While it's often a good idea to let the data reveal the structure of the model, scobit may be the exception

Takes an enormous amount of data, because curves are similar and binary outcomes contain little information




*Here we see* $\alpha < 1$ *produces a clog-log style curve*

Virtually impossible to estimate unless skew is extreme or data massive?

In practice,
use logit or probit unless you have a strong reason to suspect a specific asymmetry

In which case, use log-log or clog-log as appropriate

# Interpreting logit models

So far, we've learned how to estimate binary choice models

Now, we'll learn how to interpret these estimates

Later, we'll learn to see if our estimates are any good

Everything today will be in terms of logit, but most applies, *mutatis mutandis*, to probit, log-log, clog-log, scobit, etc.

*Running example: Why do people vote?*

We will explore a simple dataset using a naive model of voting to illustrate interpretation of logistic regression parameters

Caveat: the goal is *not* to deeply understand voting;
the model is incomplete

# Voting data: NES 2000

Turnout is binary: people either vote ($\text{Vote}_i = 1$) or they don't ($\text{Vote}_i = 0$)

Many factors could influence turn-out; we focus on age and education

American National Election Survey (ANES) in 2000: "Did you vote in 2000 election?"

```
      vote00 age hsdeg coldeg
 [1,]      1  49     1      0
 [2,]      0  35     1      0
 [3,]      1  57     1      0
 [4,]      1  63     1      0
 [5,]      1  40     1      0
 [6,]      1  77     0      0
 [7,]      0  43     1      0
 [8,]      1  47     1      1
 [9,]      1  26     1      1
[10,]      1  48     1      0
 ...
```

(Notice the structure of `hsdeg` and `coldeg`)

# A logit model

Let's model $\mathrm{Vote_i}$ as a function of $\mathrm{Age}$, $\mathrm{HSDeg}$, and $\mathrm{ColDeg}$.

Formally,

$$\Pr(\mathrm{Vote}_i|\pi_i) = f_{\mathrm{Bern}}(\pi_i)$$

$$\pi_i = \frac{1}{1 + \exp(-\mu_i)}$$

$$\mu_i = \beta_0 + \beta_{\mathrm{Age}}\mathrm{Age}_i + \beta_{\mathrm{HSDeg}}\mathrm{HSDeg}_i + \beta_{\mathrm{ColDeg}}\mathrm{ColDeg}_i$$

That's a mouthful! A more concise version?

# A logit model

Let's model $\mathrm{Vote_i}$ as a function of $\mathrm{Age}$, $\mathrm{HSDeg}$, and $\mathrm{ColDeg}$.

Slightly more compact notation:

$$\begin{aligned}\mathrm{Vote}_i &\sim \mathrm{Bernoulli}(\pi_i) \\ \pi_i &= \mathrm{logit}^{-1}(\beta_0 + \beta_{\mathrm{Age}}\mathrm{Age}_i + \beta_{\mathrm{HSDeg}}\mathrm{HSDeg}_i + \beta_{\mathrm{ColDeg}}\mathrm{ColDeg}_i)\end{aligned}$$

*Note:* You can't use the "$+\ \varepsilon_i$" notation with logit or any other nonlinear model!

# A logit model: how I write models in a paper

*Still more compact notation that might appear in a paper:*

I model respondent $i$'s decision to vote in the 2000 U.S. presidential election $(\text{Vote}_i = 1)$ using logistic regression:

$$\text{Vote}_i \sim \text{Bernoulli}(\pi_i) \quad (1)$$

$$\pi_i = \text{logit}^{-1}(\mathbf{x}_i \boldsymbol{\beta}) \quad (2)$$

where $\mathbf{x}_i$ is a vector of covariates including the respondent's *Age* in years, whether the respondent has (at least) a high school degree (*HS Degree*), and whether the respondent has a college degree (*College Degree*). All variables are taken from the ANES (American National Election Study, 2000).

*This is very helpful if we add additional stochastic layers or hierarchical structure to the model, but overkill here*

# A logit model: how I write models in a paper

*For a simple logit model, we can be even more concise. . .*

I model respondent $i$'s decision to vote in the 2000 U.S. presidential election $(\text{Vote}_i = 1)$ using logistic regression:

$$\text{Vote}_i \sim \text{Bernoulli}(\text{logit}^{-1}(\mathbf{x}_i \boldsymbol{\beta})) \tag{3}$$

where $\mathbf{x}_i$ is a vector of covariates including the respondent's *Age* in years, whether the respondent has (at least) a high school degree (*HS Degree*), and whether the respondent has a college degree (*College Degree*). All variables are taken from the ANES (American National Election Study, 2000).

*This is how I would write up this model in a paper*
*if I felt the need to write out the model formally*

# A logit model

Let's model $\mathrm{Vote_i}$ as a function of $\mathrm{Age}$, $\mathrm{HSDeg}$, and $\mathrm{ColDeg}$

Back to this notation for this lecture:

$$\begin{aligned} \mathrm{Vote}_i &\sim \mathrm{Bernoulli}(\pi_i) \\ \pi_i &= \mathrm{logit}^{-1}(\beta_0 + \beta_{\mathrm{Age}}\mathrm{Age}_i + \beta_{\mathrm{HSDeg}}\mathrm{HSDeg}_i + \beta_{\mathrm{ColDeg}}\mathrm{ColDeg}_i) \end{aligned}$$

*Note:* I use this odd notation here to make it easier to talk about specific coefficients across models. Don't imitate it in papers or homeworks!

We expect to Age and Education to increase Pr(Vote).

Thus we expect $\beta_{\mathrm{Age}} > 0$, $\beta_{\mathrm{HSDeg}} > 0$, $\beta_{\mathrm{ColDeg}} > 0$.

This is the (now familiar) logit model

We estimate it using ML, such as through `optim` in `R`

# Exciting R output

```
$par
(Intercept)        xage      xhsdeg     xcoldeg
-2.23246823  0.03169878  1.29593503  1.18035804

$value
[1] 1027.954

$counts
function gradient
      49        8

$convergence
[1] 0

$hessian
            (Intercept)       xage      xhsdeg    xcoldeg
(Intercept)   349.69863  15813.476   308.88407   73.52431
xage        15813.47596 810140.667 13386.99697 3074.98330
xhsdeg        308.88407  13386.997   308.88407   73.52431
xcoldeg        73.52431   3074.983    73.52431   73.52431
```

Table 1: *Determinants of voting in the 2000 presidential election.* Logit coefficients estimated using data from the 2000 American National Election Survey.

| | est. | s.e. | $p$-value |
|---|---|---|---|
| Age | $0.032$ | $0.003$ | <0.001 |
| High School Grad | $1.296$ | $0.178$ | <0.001 |
| College Grad | $1.180$ | $0.134$ | <0.001 |
| Constant | $-2.232$ | $0.257$ | <0.001 |
| log likelihood | $-1027.95$ | | |
| $N$ | $1798$ | | |

We can make the `R` output prettier

We'll also calculate $p$-values, using $p = 1 - F_t\left(\hat{\beta}/\sqrt{\mathrm{var}(\hat{\beta})}, n-k\right)$

(Aside: where do we find these quantities?)

All parameters have expected signs and are very significant

Declare victory and publish?

Table 2: *Determinants of voting in the 2000 presidential election.* Logit coefficients estimated using data from the 2000 American National Election Survey.

| | est. | s.e. | $p$-value |
|---|---|---|---|
| Age | $0.032$ | $0.003$ | <0.001 |
| High School Grad | $1.296$ | $0.178$ | <0.001 |
| College Grad | $1.180$ | $0.134$ | <0.001 |
| Constant | $-2.232$ | $0.257$ | <0.001 |
| log likelihood | $-1027.95$ | | |
| $N$ | $1798$ | | |

What do these estimates *mean*? Are the effects small or large? Which is "biggest"?

Table 2: *Determinants of voting in the 2000 presidential election.* Logit coefficients estimated using data from the 2000 American National Election Survey.

| | est. | s.e. | $p$-value |
|---|---|---|---|
| Age | $0.032$ | $0.003$ | <0.001 |
| High School Grad | $1.296$ | $0.178$ | <0.001 |
| College Grad | $1.180$ | $0.134$ | <0.001 |
| Constant | $-2.232$ | $0.257$ | <0.001 |
| log likelihood | $-1027.95$ | | |
| $N$ | $1798$ | | |

What do these estimates *mean*? Are the effects small or large? Which is "biggest"?

What is $\Pr(\text{Vote}|\text{Age} = a, \text{HSDeg} = h, \text{ColDeg} = c)$, for interesting $a$, $h$, $c$?

What is the $\Delta\Pr(\text{Vote})$ if we take a person on average age and give them a college degree? or increase their age by ten years?

What are the confidence intervals for any of these quantities?

. . . Maybe we're not finished with the analysis

# 4 ways to interpret logit results

1. Graphical display of response surface

2. Derivatives calculated at points on the response surface

3. Directly intepret coefficients on the scale of estimation (the logit)

4. Calculate interesting expected values & first differences

All four are trivially expressed by the coefficients in linear regression, but not in logit or other non-linear models

# Plotting the response surface

The response surface is a complete summary of $\mathbb{E}(\pi|\mathbf{x} = \mathbf{x}_c)$
for any specific possible values of $\mathbf{x}$ across $k$ covariates

This is a manifold (bumpy blanket) in $k + 1$ dimensions

(Approximates a linear regression (flat sheet) only to extent that

$$\text{For all } \mathbf{x}_c, \Pr(y|\mathbf{x} = \mathbf{x}_c) \approx \bar{y}$$

That is, only for very poorly fitting/uninformative models!)

Unwieldy, especially if $> 2$ predictor variables

Even then, not everyone can read 3D plots well

Confidence intervals can be tricky to display – but see Sun and Genton (2011)
"Functional boxplots" *J. of Computational & Graphical Stat.*

# Derivative methods

For linear regression:

$$\frac{\partial \hat{y}}{\partial x_j} = \hat{\beta}_j$$

Derivative is a constant – very convenient summary

For logit:

$$\frac{\partial \hat{\pi}}{\partial x_j} = \hat{\beta}_j \hat{\pi}(1 - \hat{\pi})$$

$$\frac{\partial \hat{\pi}}{\partial x_j} = \hat{\beta}_j \times \frac{1}{1 + \exp(-\hat{\beta}_0 - \sum_{k=1}^{K} \hat{\beta}_k x_k)} \times \left(1 - \frac{1}{1 + \exp(-\hat{\beta}_0 - \sum_{k=1}^{K} \hat{\beta}_k x_k)}\right)$$

Derivative depends on $\hat{\pi}$, which depends on all the $x$'s and $\beta$'s:
not so convenient

Derivative is at maximum when $\hat{\pi} = 0.5$; instantaneous effect at this point is $\beta_j/4$

Derivative methods are useful if reader has a calculator,
descriptive statistics for all $x$'s, and a lot of patience . . .

# Direct interpretation: log odds

Logit coefficients are obscure: understanding them requires some background

Define the *odds* of $y$ given $\mathbf{x}$ as

$$O_{y|\mathbf{x}} = \frac{\Pr(y|\mathbf{x})}{1 - \Pr(y|\mathbf{x})}$$

For example, if betting markets have the odds of Clinton beating Trump at $11/2$, then

$$O_{\text{Clinton}} = \frac{\Pr(\text{Clinton})}{\Pr(\text{Trump})} = \frac{11}{2} = \frac{.846}{.154}$$

$$\Pr(\text{Clinton}) = \frac{11}{11+2} = .846$$

$$\Pr(\text{Trump}) = \frac{2}{11+2} = .154$$

# Direct interpretation: log odds

Notice that

$$\boldsymbol{\pi} = \text{logit}^{-1}(\mathbf{x}\boldsymbol{\beta}) \qquad \Rightarrow \qquad \text{logit}(\boldsymbol{\pi}) = \mathbf{x}\boldsymbol{\beta}$$

$$\boldsymbol{\pi} = \frac{1}{1 + \exp(-\mathbf{x}\boldsymbol{\beta})} \qquad \Rightarrow \qquad \log\left(\frac{\boldsymbol{\pi}}{1 - \boldsymbol{\pi}}\right) = \mathbf{x}\boldsymbol{\beta}$$

$\log\left(\frac{\boldsymbol{\pi}}{1 - \boldsymbol{\pi}}\right)$ is the log odds – hence "logit"

A $1$ unit increase in $x_k$ increases the log odds by $\beta_k$

. . . Okay, but how do we use this?

# Direct interpretation: log odds

Next define the *relative risk*, a useful summary of the effect of $\mathbf{x}$ on $y$:

$$\text{RR} = \frac{\Pr(y|\mathbf{x})}{\Pr(y|\mathbf{x}')}$$

*At the average age,* the RR of voting given a college degree versus high school is

$$\text{RR}_{\text{coldeg,hsdeg}} = \frac{\Pr(\text{Vote}|\text{coldeg})}{\Pr(\text{Vote}|\text{HSdeg})} = \frac{.8501}{.6363} = 1.3360$$

A College grad is 33.6% more likely to vote than a HS grad, at age 47

The value of the *relative risk* depends on other covariates

Here, the relative risk with respect to education depends on the level of age

# Direct interpretation: log odds

Finally, define the odds ratio as:

$$\mathrm{OR}_{\mathbf{x},\mathbf{x}'} = \frac{O_{y|\mathbf{x}}}{O_{y|\mathbf{x}'}} = \frac{\Pr(y|\mathbf{x})}{1-\Pr(y|\mathbf{x})} \Big/ \frac{\Pr(y|\mathbf{x}')}{1-\Pr(y|\mathbf{x}')}$$

Note this is *not* the same as the relative probability (or relative "risk"):

$$\mathrm{OR}_{\mathrm{ColDeg},\mathrm{HSDeg}} = \frac{O_{\mathrm{Vote}|\mathrm{ColDeg}}}{O_{\mathrm{Vote}|\mathrm{HSDeg}}} = \frac{\Pr(\mathrm{Vote}|\mathrm{ColDeg})}{1-\Pr(\mathrm{Vote}|\mathrm{ColDeg})} \Big/ \frac{\Pr(\mathrm{Vote}|\mathrm{HSDeg})}{1-\Pr(\mathrm{Vote}|\mathrm{HSDeg})}$$

$$\mathrm{OR}_{\mathrm{ColDeg},\mathrm{HSDeg}} = 3.2415$$

The OR is fixed regardless of age, but its meaning is unclear

The odds ratio is *not* the relative risk and *not* directly interesting

Exception for rare events:
the odds ratio approaches the relative risk as $\Pr(y) \to 0$

# Direct interpretation: log odds

Odds ratios are usually uninterpretable and often misleading,
at least if we treat them as relative risks

But odds ratios are what $\boldsymbol{\beta}$ shifts:

$$\log O_{y|\mathbf{x}}/O_{y|\mathbf{x}'} = \log O_{y|\mathbf{x}} - \log O_{y|\mathbf{x}'}$$

$$\log O_{y|\mathbf{x}}/O_{y|\mathbf{x}'} = \mathbf{x}\boldsymbol{\beta} - \mathbf{x}'\boldsymbol{\beta}$$

$$\log O_{y|\mathbf{x}}/O_{y|\mathbf{x}'} = (\mathbf{x} - \mathbf{x}')\boldsymbol{\beta}$$

And so for $\Delta x_k = 1$,

$$O_{y|x_k}/O_{y|x'_k} = \exp(\beta_k)$$

A one unit increase in $x_k$ increases the odds ratio by $\exp(\beta_k)$

# Direct interpretation: log odds

If you have rare events – that is, $\Pr(y \approx 0)$ for the whole dataset – it may be useful to note that $\exp(\hat{\beta})$ is the change in the odds ratio

Otherwise, odds ratios are not a good way to express results

Requires the reader have a calculator, knowledge of odds ratios, and a fair bit of imagination

Why restrict your audience and risk misunderstanding?

So on to a better way . . .

# Computing quantities of interest

Given some counterfactual $\mathbf{x}_c$'s, our goal to obtain "quantities of interest" like

- Expected Values: $\mathbb{E}\left(y|\mathbf{x}_c\right)$
- First Differences: $\mathbb{E}\left(y|\mathbf{x}_{c2}\right) - \mathbb{E}\left(y|\mathbf{x}_{c1}\right)$
- Relative Risks: $\dfrac{\mathbb{E}\left(y|\mathbf{x}_{c2}\right)}{\mathbb{E}\left(y|\mathbf{x}_{c1}\right)}$
- or any function of the above

Getting point estimates is easy: e.g., for EVs, just plug $\mathbf{x}_c$ into $\dfrac{1}{1+\exp(-\mathbf{x}_c\boldsymbol{\beta})}$

# Computing quantities of interest using point estimates

*What is the probability a high school graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.15 + 0.0309 \times 42.7 + 1.21 \times 1 + 1.10 \times 0) = 62.8\%$$

*What is the probability a college graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.15 + 0.0309 \times 42.7 + 1.21 \times 1 + 1.10 \times 1) = 83.5\%$$

*How much does a college degree raise the chance of voting for a high school grad?*

$$83.5\% - 62.8\% = 20.8\%$$

*Is this statistically significant? What's the 95% CI?*

Confidence intervals of these quantities are tiresome to calculate analytically, so let's try simulation

Let's start with what we know about the MLEs:

| | $\hat{\beta}_0$ | $\hat{\beta}_{\text{Age}}$ | $\hat{\beta}_{\text{HSdeg}}$ | $\hat{\beta}_{\text{Coldeg}}$ |
|---|---|---|---|---|
| *estimates* | $-2.15$ | $0.0309$ | $1.21$ | $1.10$ |
| *var-cov* | | | | |
| $\hat{\beta}_0$ | $0.0658$ | $-0.000682$ | $-0.0358$ | $-0.00135$ |
| $\hat{\beta}_{\text{Age}}$ | $-0.000682$ | $0.0000115$ | $0.000178$ | $0.0000227$ |
| $\hat{\beta}_{\text{HSdeg}}$ | $-0.0358$ | $0.000178$ | $0.0322$ | $-0.00388$ |
| $\hat{\beta}_{\text{Coldeg}}$ | $-0.00135$ | $0.0000227$ | $-0.00388$ | $0.0170$ |
| *correlations* | | | | |
| $\hat{\beta}_0$ | $1.000$ | $-0.785$ | $-0.777$ | $-0.040$ |
| $\hat{\beta}_{\text{Age}}$ | $-0.785$ | $1.000$ | $0.292$ | $0.051$ |
| $\hat{\beta}_{\text{HSdeg}}$ | $-0.777$ | $0.292$ | $1.000$ | $-0.166$ |
| $\hat{\beta}_{\text{Coldeg}}$ | $-0.040$ | $0.051$ | $-0.166$ | $1.000$ |

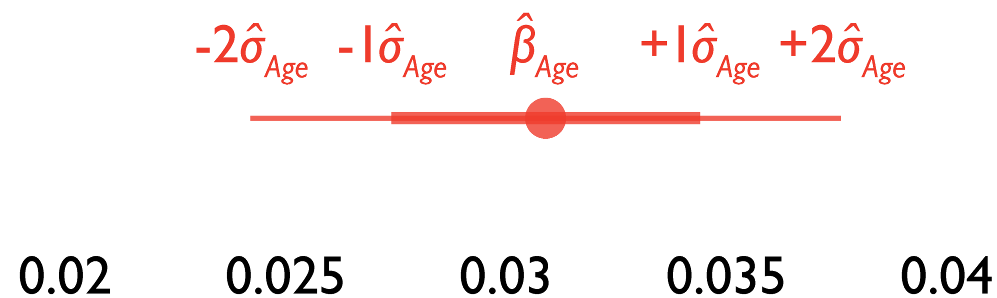
Consider the model estimates of $\beta_{\text{Age}}$ and $\beta_{\text{HSdeg}}$ (se's in parentheses):

$\hat{\beta}_{\text{Age}} = 0.0309\ (0.00339)$ $\quad$ $\hat{\beta}_{\text{HSdeg}} = 1.21\ (0.179)$ $\quad$ $r_{\hat{\beta}_{\text{Age}},\hat{\beta}_{\text{HSdeg}}} = 0.292$

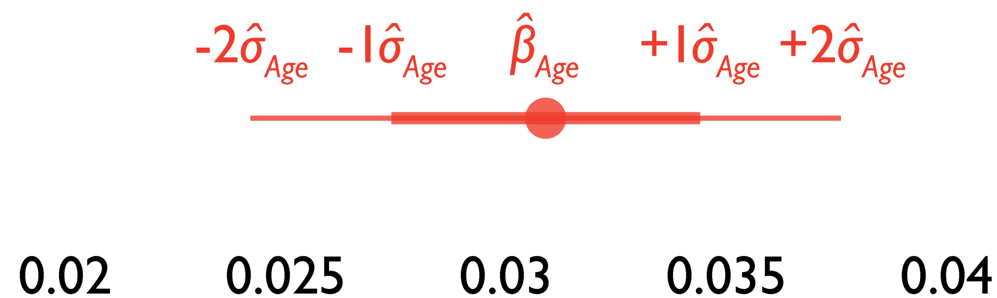
These describe the 1st and 2nd moments of an asymptotically Normal distribution

In computing functions of $\beta$, how do we account for this uncertainty?

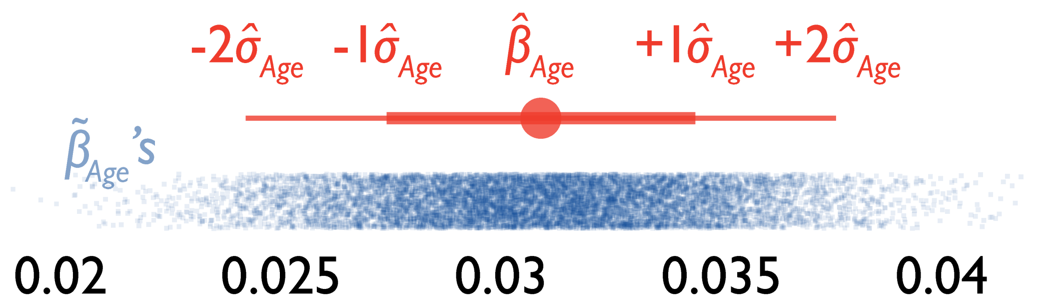
Perhaps we can capture and pass on this variability using *simulation*



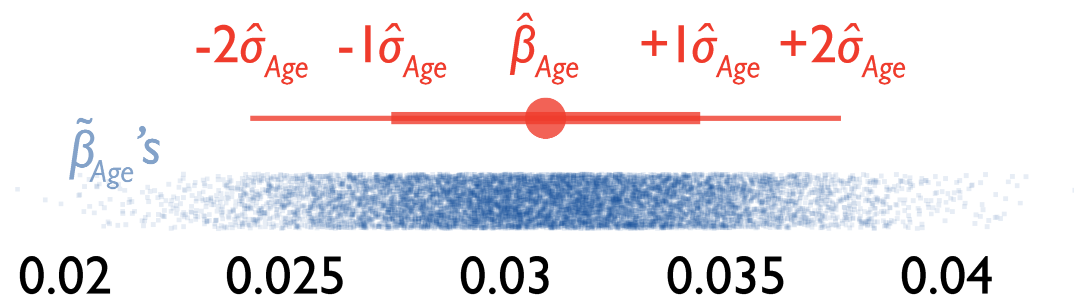

We often summarize uncertainty with confidence intervals around parameters



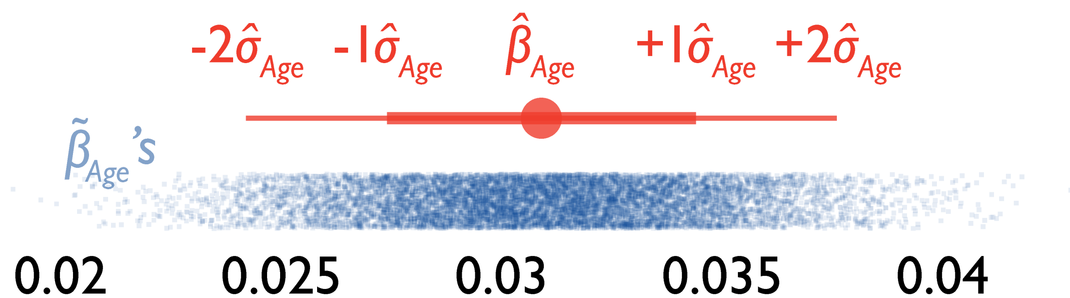

These intervals represent $\hat{\beta}_{\text{Age}} = 0.0309\ (0.00339)$,
assuming the uncertainty in the parameter estimate is Normally distributed



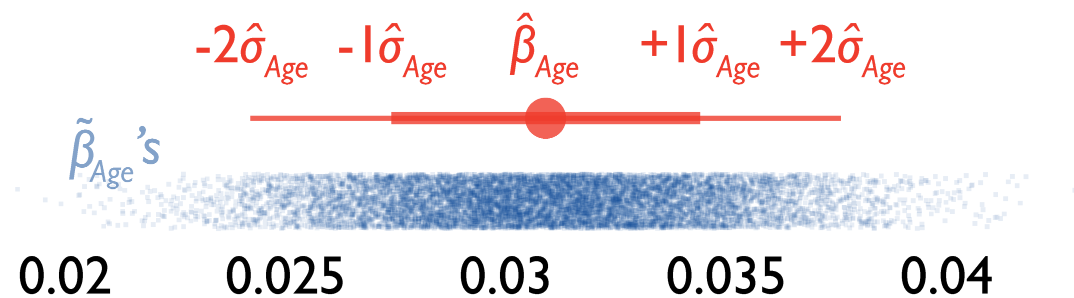

But we could use random *draws* (here, 10,000 draws)
from the distribution of the parameter to capture the same uncertainty



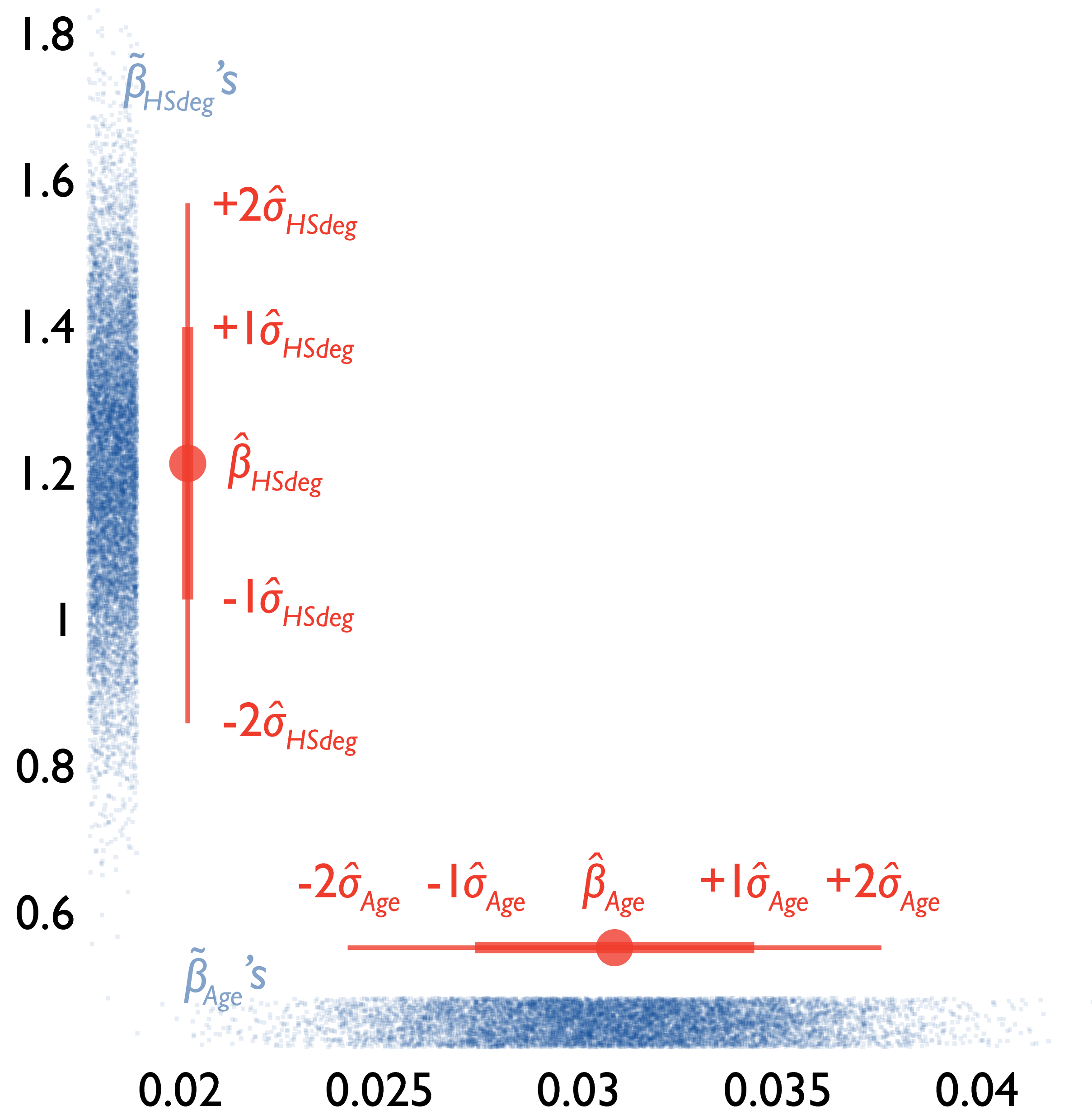

68% [95%] of the draws fall in the $\pm 1$ [$\pm 2$] $\hat{\sigma}$ confidence interval



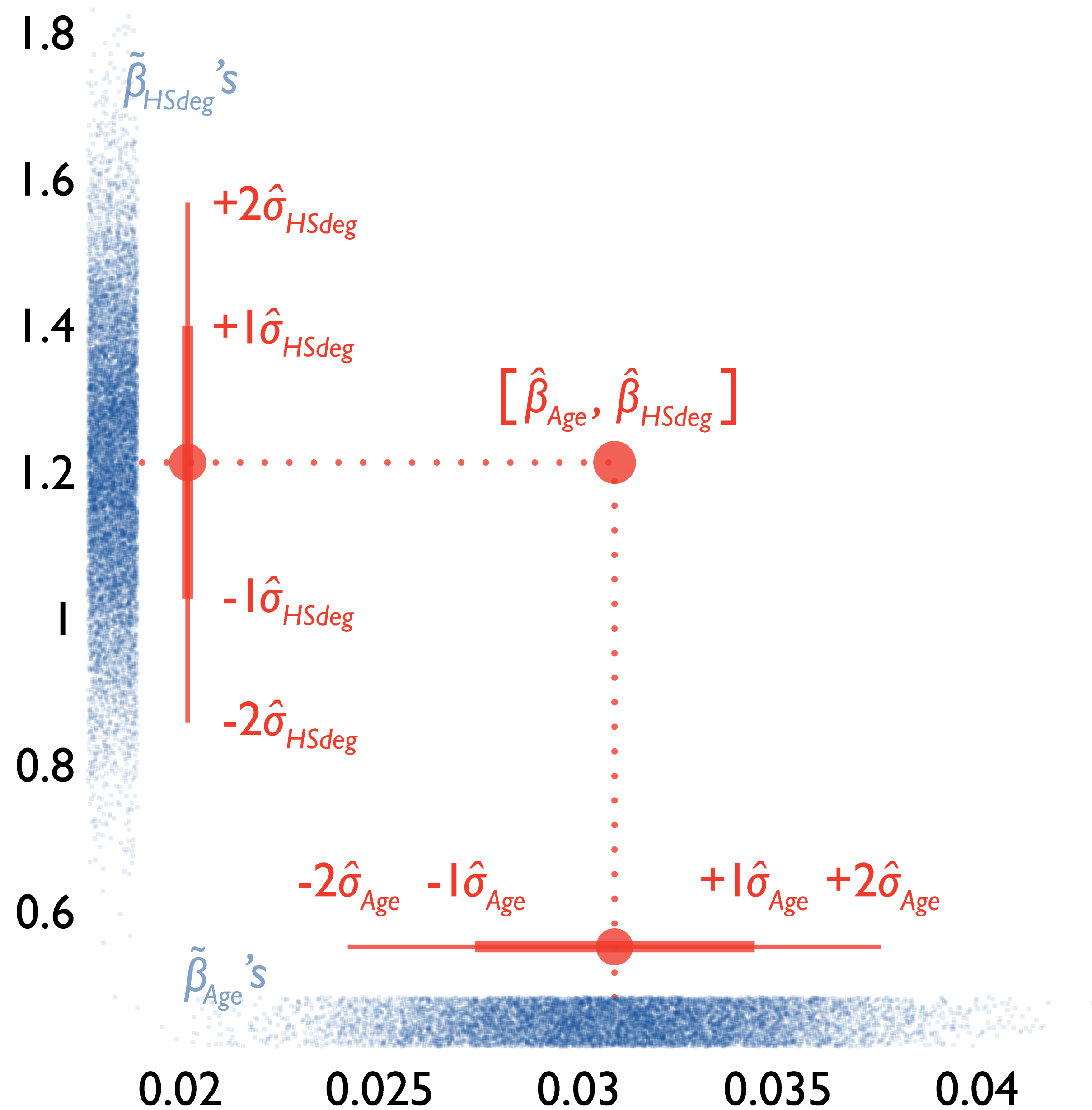

Whatever other quantities we calculate with these 10,000 draws will, across their 10,000 versions, contain all the uncertainty we have about $\hat{\beta}_{\text{Age}}$



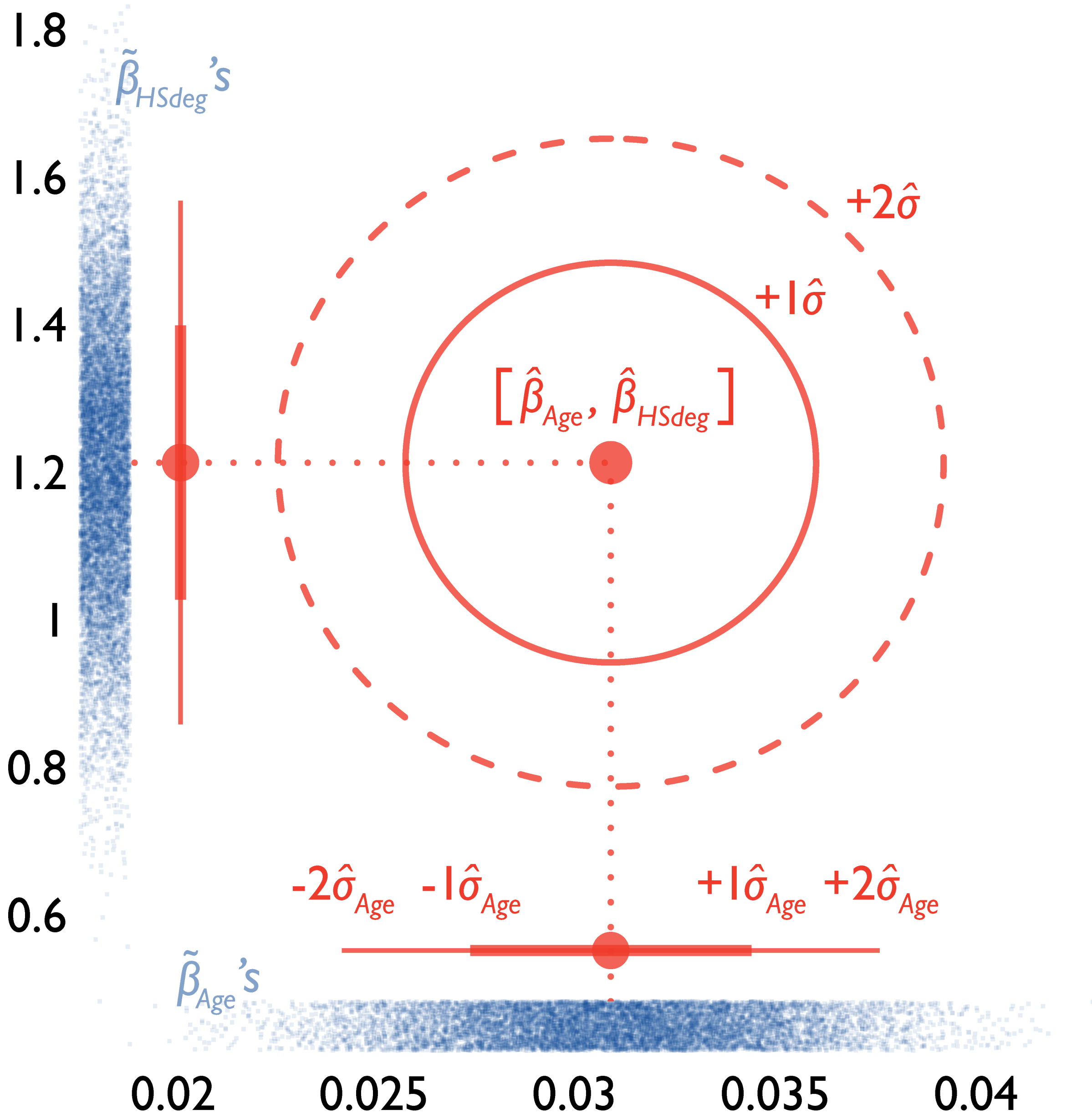

We can capture uncertainty in any estimated parameter using draws from its predictive distribution



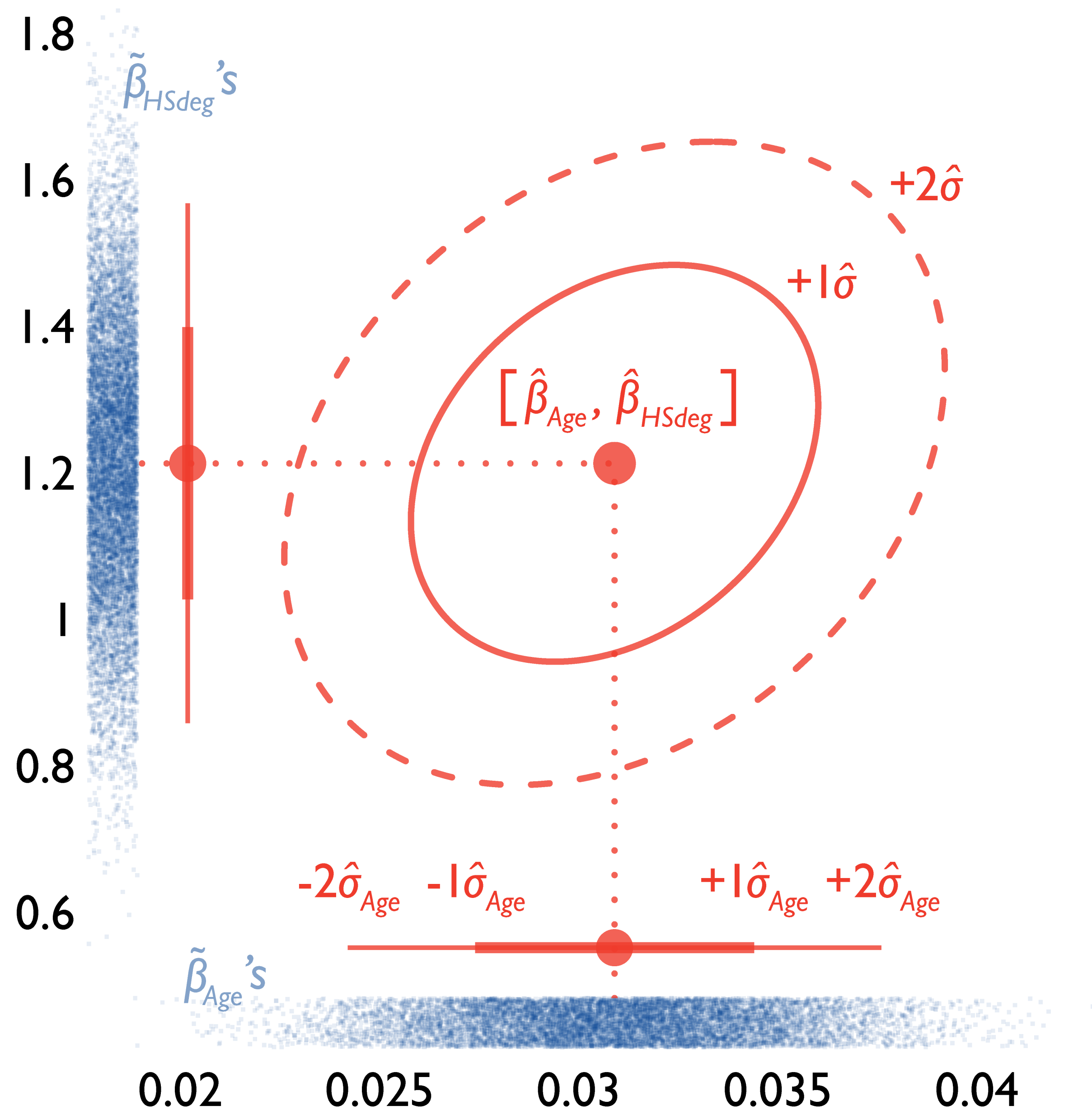

For example, here is the estimated distribution of $\hat{\beta}_{\text{HSdeg}}$, which we again assume is Normally distributed



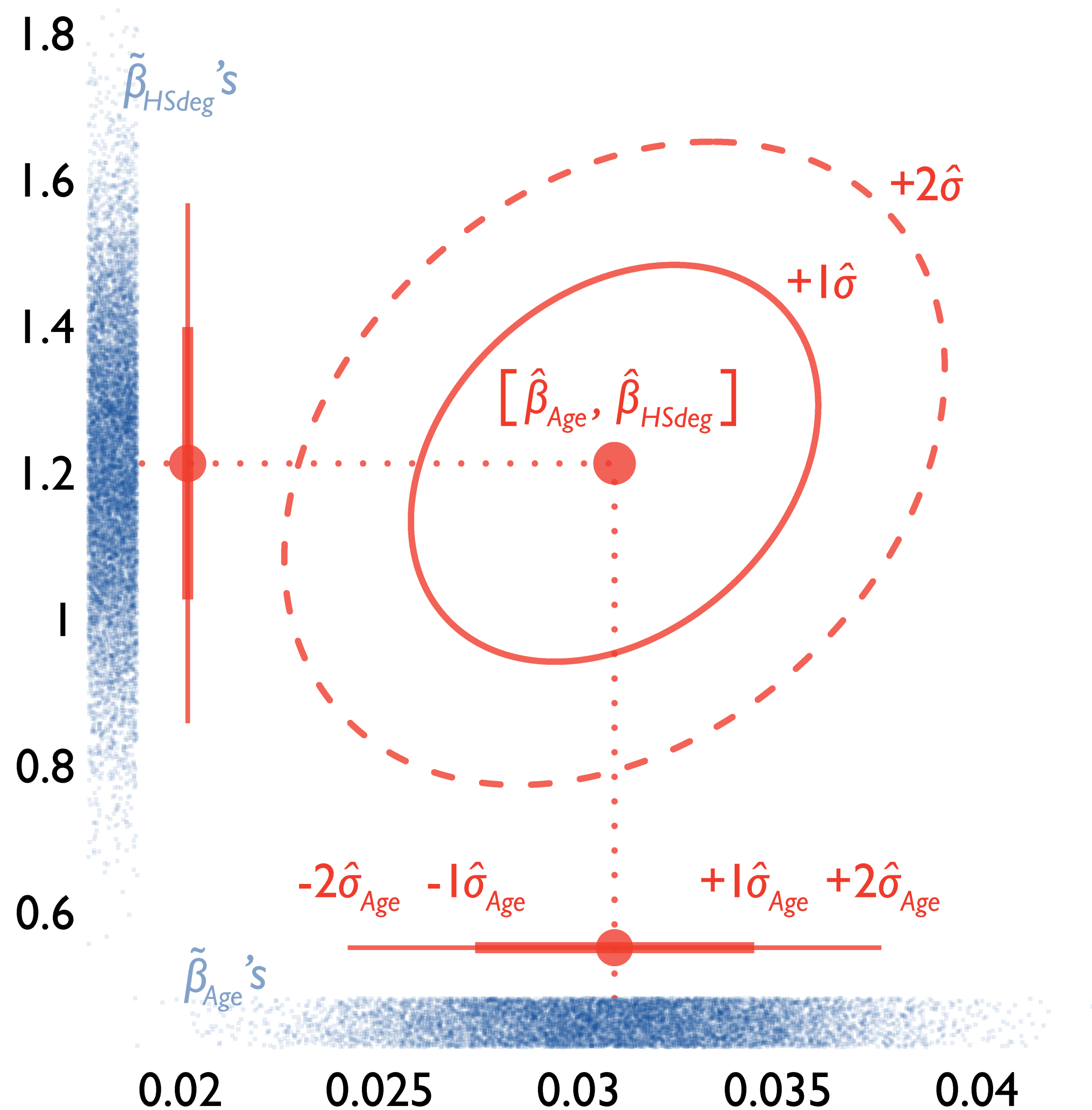

But we want the *joint* distribution of $\hat{\beta}_{\text{Age}}$ and $\hat{\beta}_{\text{HSdeg}}$ –
can we draw $\hat{\beta}_{\text{Age}}$ and $\hat{\beta}_{\text{HSdeg}}$ separately from Normal distributions?



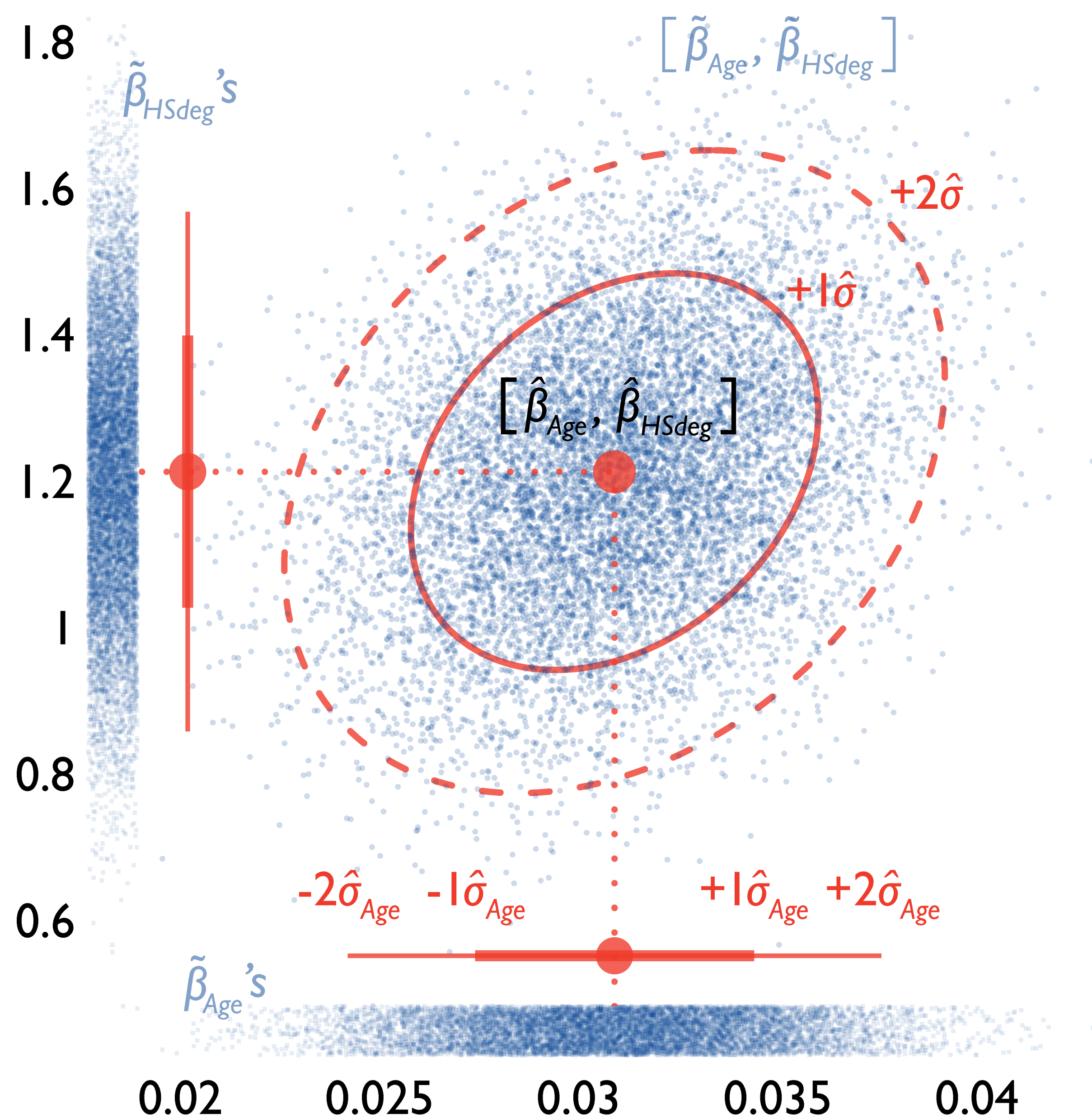

Not unless we're willing to assume they're *uncorrelated*. . .



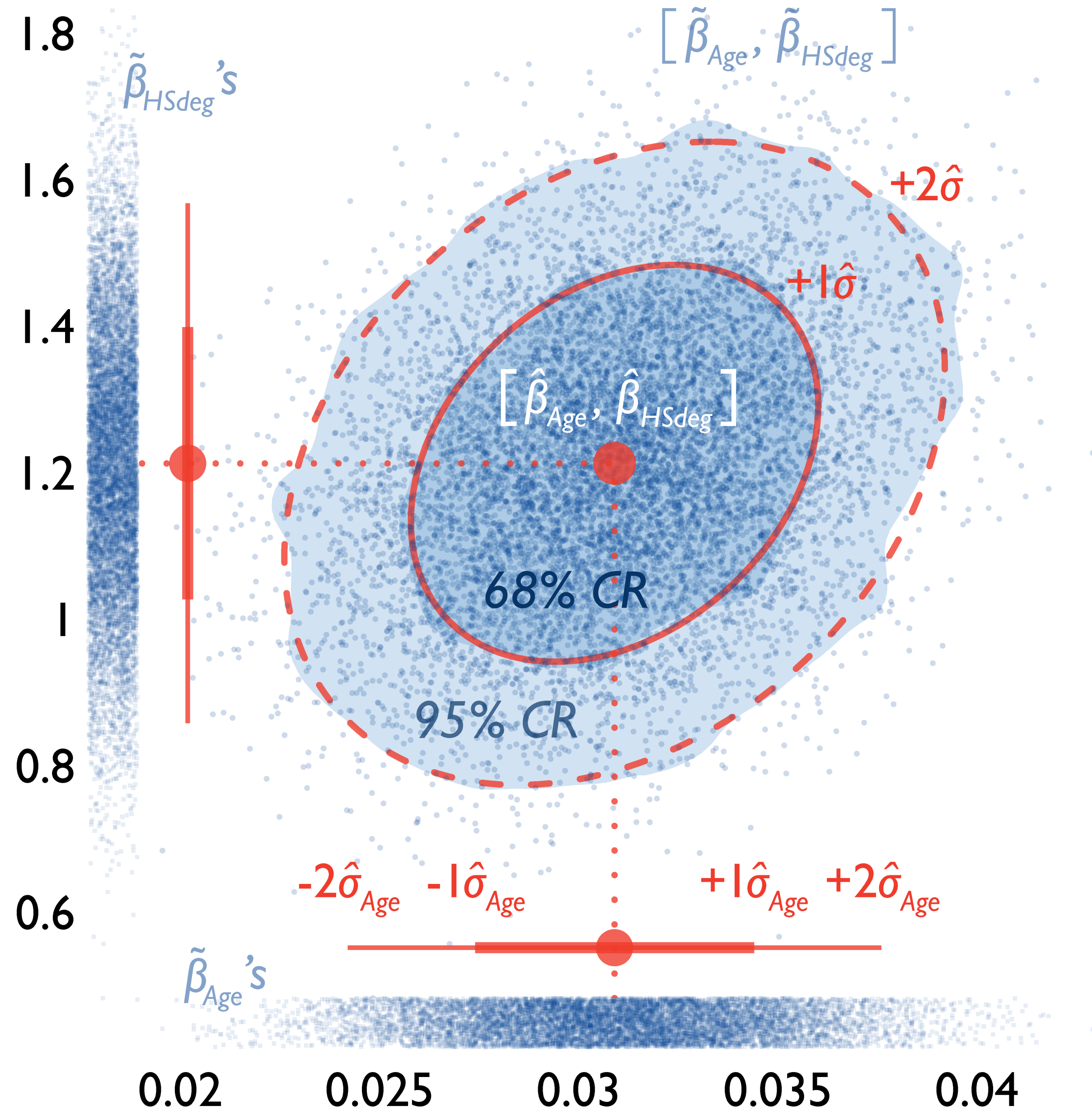

and we *know* these parameters are correlated at $r = 0.292$



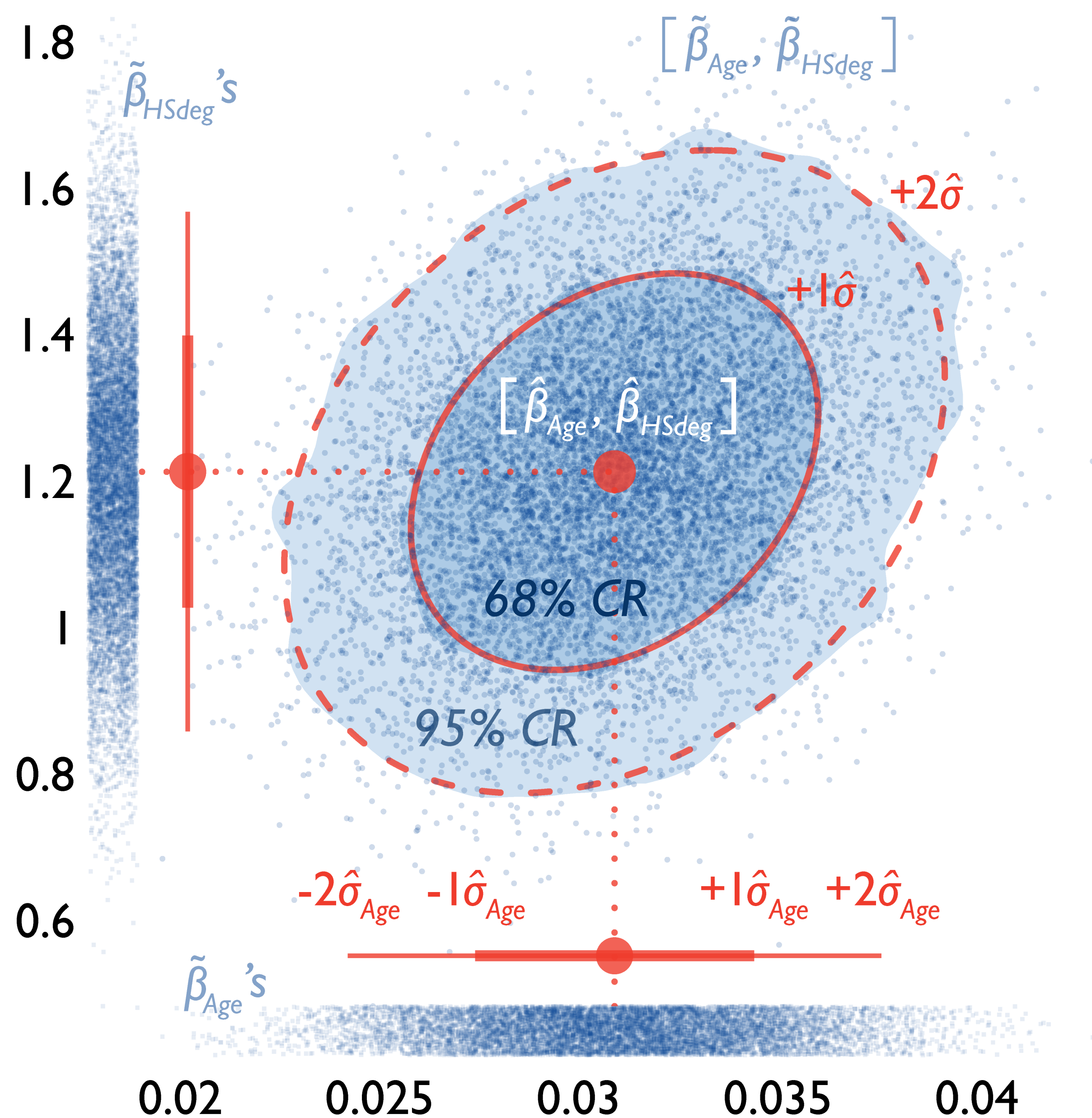

Let's try to simulate these two parameters *jointly*



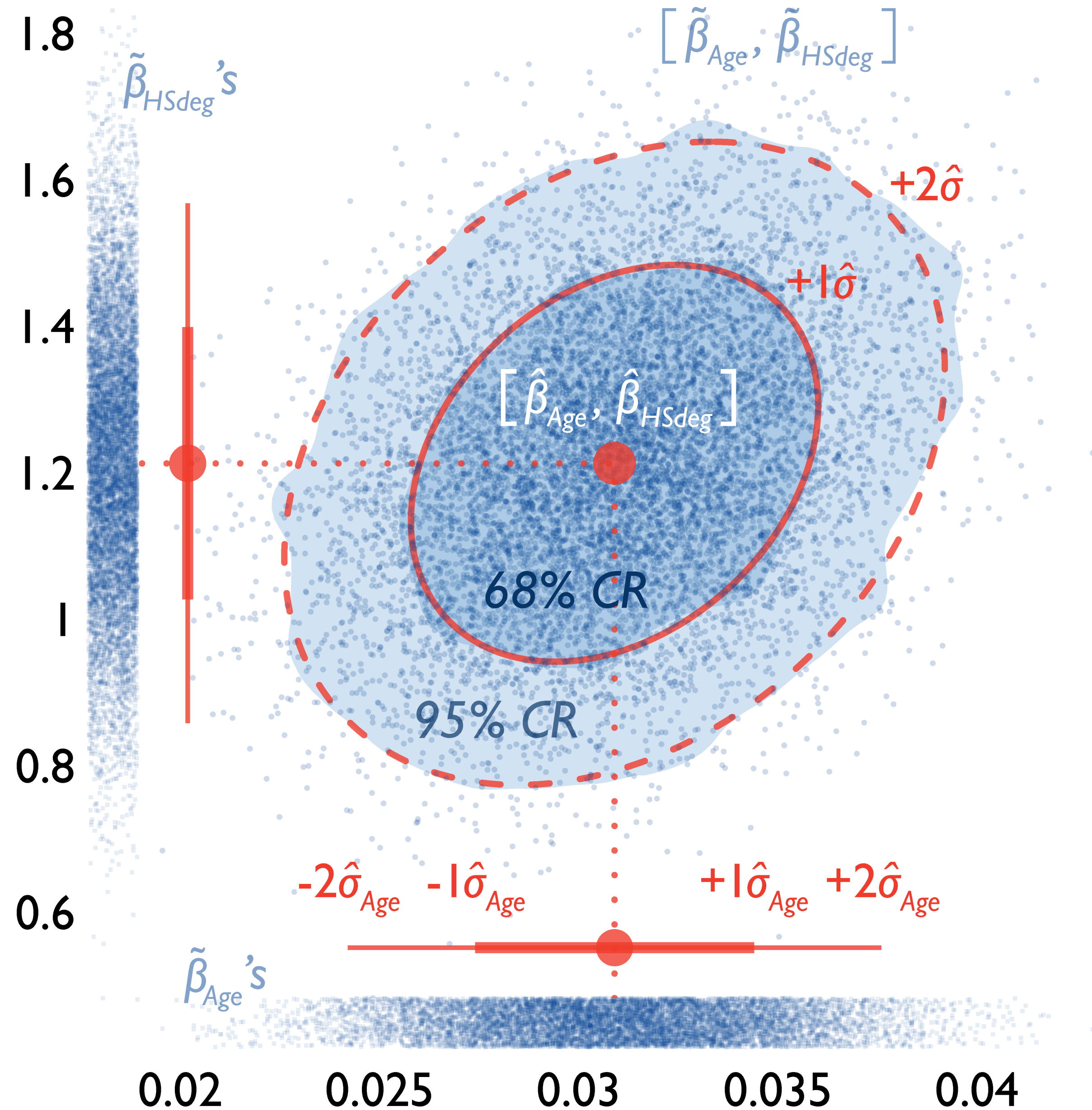

Draw 10,000 simulations $\left[\tilde{\boldsymbol{\beta}}_{\text{Age}}, \tilde{\boldsymbol{\beta}}_{\text{HSdeg}}\right]$ from $\text{MVN}\left(\begin{bmatrix}0.0309\\1.21\end{bmatrix}, \begin{bmatrix}0.0000115 & 0.000178\\0.000178 & 0.0322\end{bmatrix}\right)$



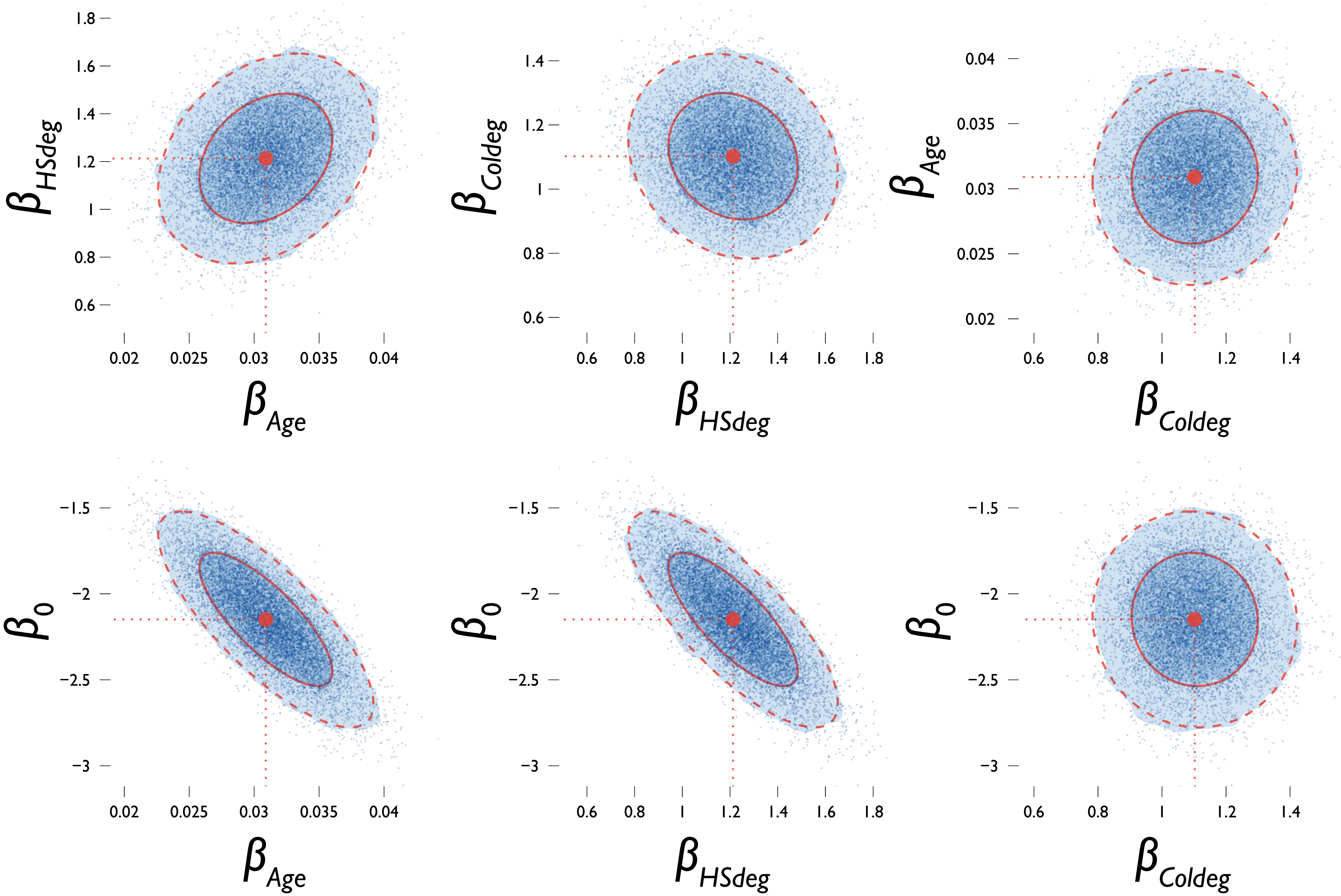

The central 68% [95%] of all draws match up closely with
the theoretical 68% [95%] confidence regions of the parameters



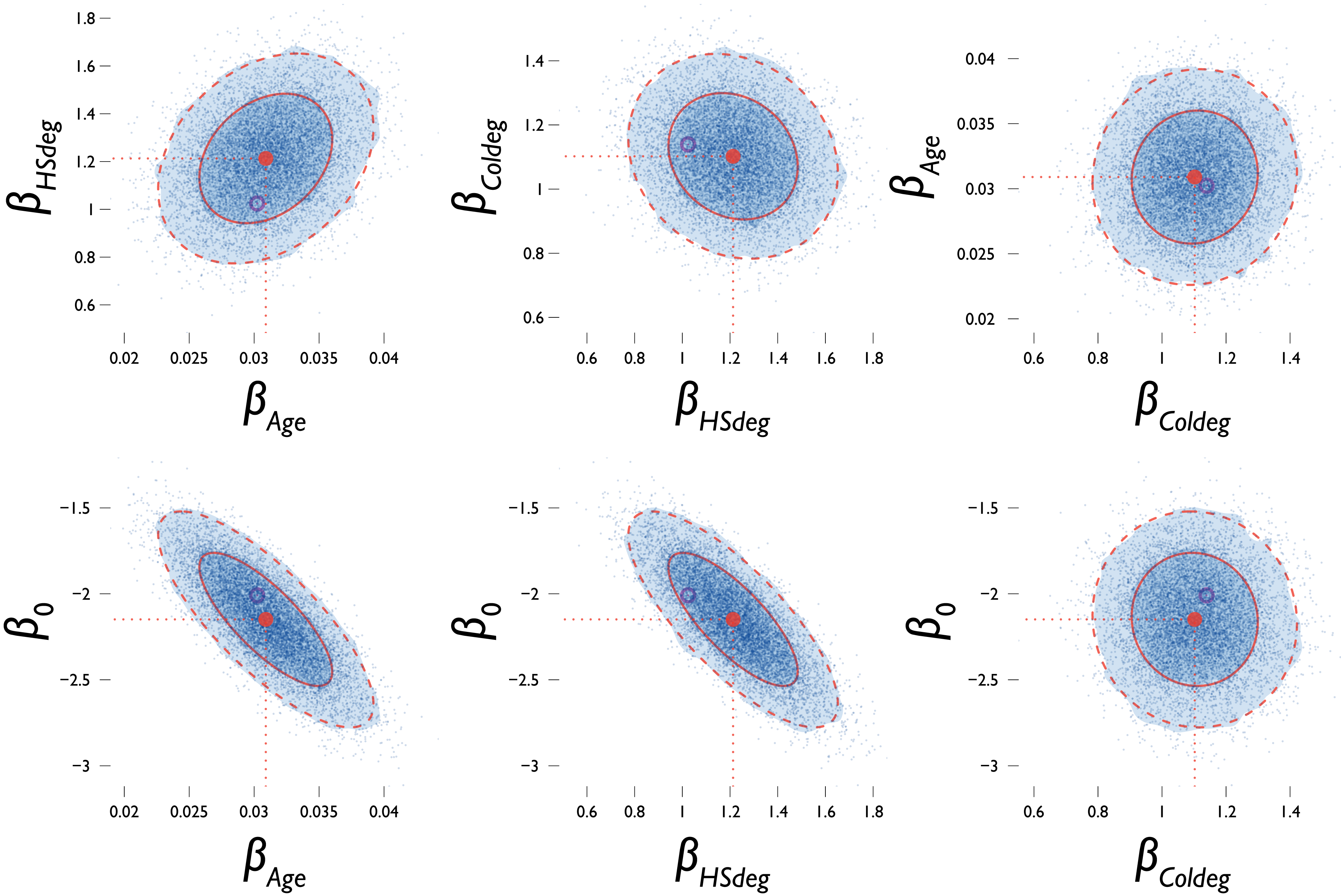

Why might these parameter estimates be positively correlated?
What implications does this have for computing quantities of interest?




Of course, we have not two but *four* estimated parameters,
so we should draw all four at once to account for all estimated covariances




Collectively, the 10,000 simulations of $[\tilde{\beta}_0, \tilde{\beta}_{\text{Age}}, \tilde{\beta}_{\text{HSdeg}}, \tilde{\beta}_{\text{Coldeg}}]$ capture everything the model "knows" about them




Let's look at a specific draw of simulated $\tilde{\beta}$'s
and compute our quantities of interest

# Computing quantities of interest using point estimates

*What is the probability a high school graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.15 + 0.0309 \times 42.7 + 1.21 \times 1 + 1.10 \times 0) = 62.8\%$$

*What is the probability a college graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.15 + 0.0309 \times 42.7 + 1.21 \times 1 + 1.10 \times 1) = 83.5\%$$

*How much does a college degree raise the chance of voting for a high school grad?*

$$83.5\% - 62.8\% = 20.8\%$$

Recall that the above were the *point estimates*
of our quantities of interest

# Simulating quantities of interest: Draw 8 of 10,000

*What is the probability a high school graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.01 + 0.0302 \times 42.7 + 1.02 \times 1 + 1.14 \times 0) = 60.9\%$$

*What is the probability a college graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.01 + 0.0302 \times 42.7 + 1.02 \times 1 + 1.14 \times 1) = 82.9\%$$

*How much does a college degree raise the chance of voting for a high school grad?*

$$82.9\% - 60.9\% = 22.1\%$$

Each simulation of the QoI will differ slightly from the mean;
collectively, these differences capture our uncertainty

# Simulating quantities of interest: Draw 9 of 10,000

*What is the probability a high school graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.08 + 0.0330 \times 42.7 + 0.99 \times 1 + 1.00 \times 0) = 61.4\%$$

*What is the probability a college graduate of average age (42.7 years) votes?*

$$\Pr(\text{Vote}|\text{Age} = 42.7, \text{HSdeg} = 1, \text{Coldeg} = 0)$$

$$= \text{logit}^{-1}(-2.08 + 0.0330 \times 42.7 + 0.99 \times 1 + 1.14 \times 1) = 81.3\%$$

*How much does a college degree raise the chance of voting for a high school grad?*

$$81.3\% - 61.4\% = 19.8\%$$

Each simulation of the QoI will differ slightly from the mean;
collectively, these differences capture our uncertainty

# Simulating quantities of interest: Summary

Just as the middle 95% of simulated $\beta$'s matched the 95% CI,
the middle 95% of simulated $\mathrm{Pr}(\mathrm{Vote}|\mathrm{Age}, \mathrm{HSdeg}, \mathrm{Coldeg})$'s provide a 95% CI

*What is the probability a high school graduate of average age (42.7 years) votes?*

The mean across 10,000 simulations was 62.8%, with a 95% CI of [59.8%, 65.8%]

*What is the probability a college graduate of average age (42.7 years) votes?*

The mean across 10,000 simulations was 83.5%, with a 95% CI of [80.2%, 86.4%]

*How much does a college degree raise the chance of voting for a high school grad?*

The average difference across 10,000 simulations
was 20.7%, with a 95% CI of [16.5%, 24.9%]

These closely match the point estimates computed above,
but now have measures of uncertainty as well

# Simulating quantities of interest: formalization

Let's formalize the simulation procedure we've used to interpret our logit result

We consider a generic simulation method (King, Tomz, Wittenberg 2000)
for models of the form:

$$y_i \sim f(\mu_i, \boldsymbol{\alpha}) \qquad \mu_i = g(\boldsymbol{\beta}, \mathbf{x}_i)$$

(for convenience, $\boldsymbol{\theta} = \mathrm{vec}(\boldsymbol{\beta}, \boldsymbol{\alpha})$)

This formulation includes all the models in POLS/CSSS 510

# Algorithm for simulating EVs and their confidence intervals

1. Choose a counterfactual $\mathbf{x}_c$
2. Estimate the model and obtain the parameter vector, $\hat{\boldsymbol{\theta}}$, and its variance covariance matrix, $\hat{\mathrm{V}}(\hat{\boldsymbol{\theta}})$
3. Draw $\tilde{\boldsymbol{\theta}}$ from the multivariate normal $f_{\mathcal{MVN}}\left(\hat{\boldsymbol{\theta}}, \hat{\mathrm{V}}(\hat{\boldsymbol{\theta}})\right)$
4. Calculate $\tilde{\mu}_c = g(\tilde{\boldsymbol{\beta}}, \mathbf{x}_c)$
5. Draw $\tilde{y}_c \sim f(\tilde{\mu}_c, \tilde{\alpha})$, from the *model's* distribution.
   This is a *predicted value*. [Aside]
6. Repeat steps 3 to 5 many times, averaging the results to get one *expected value*
7. Repeat step 6 many times to obtain a vector of expected values
8. Summarize this vector using means, sd's, or confidence intervals

# The algorithm applied to logit

1. Choose a counterfactual $\mathrm{x}_c$
2. Estimate the model and obtain the parameter vector, $\hat{\boldsymbol{\beta}}$, and its variance covariance matrix, $\hat{\mathrm{V}}(\hat{\boldsymbol{\beta}})$
3. Draw $\tilde{\boldsymbol{\beta}}$ from the multivariate normal $f_{\mathcal{MVN}}\left(\hat{\boldsymbol{\beta}}, \hat{\mathrm{V}}(\hat{\boldsymbol{\beta}})\right)$
4. Calculate $\tilde{\pi}_c = 1/(1 + \exp(-\mathbf{x}_c\tilde{\boldsymbol{\beta}})$
5. Draw $\tilde{y}_c \sim \mathrm{Bernoulli}(\tilde{\pi}_c)$: this is a *predicted value* (a 1 or a 0)
6. Repeat steps 3 to 5 many times, averaging the results to get one *expected value* (i.e., a $\hat{\pi}_c$)
7. Repeat step 6 many times to obtain a vector of expected values
8. Summarize this vector using means, sd's, or confidence intervals

# A shortcut

A good idea to simulate at least 10,000 PVs to get a single EV

And at least 10,000 EVs to get its distribution

Which means doing 100 million sims for each $\mathrm{x}_c$!

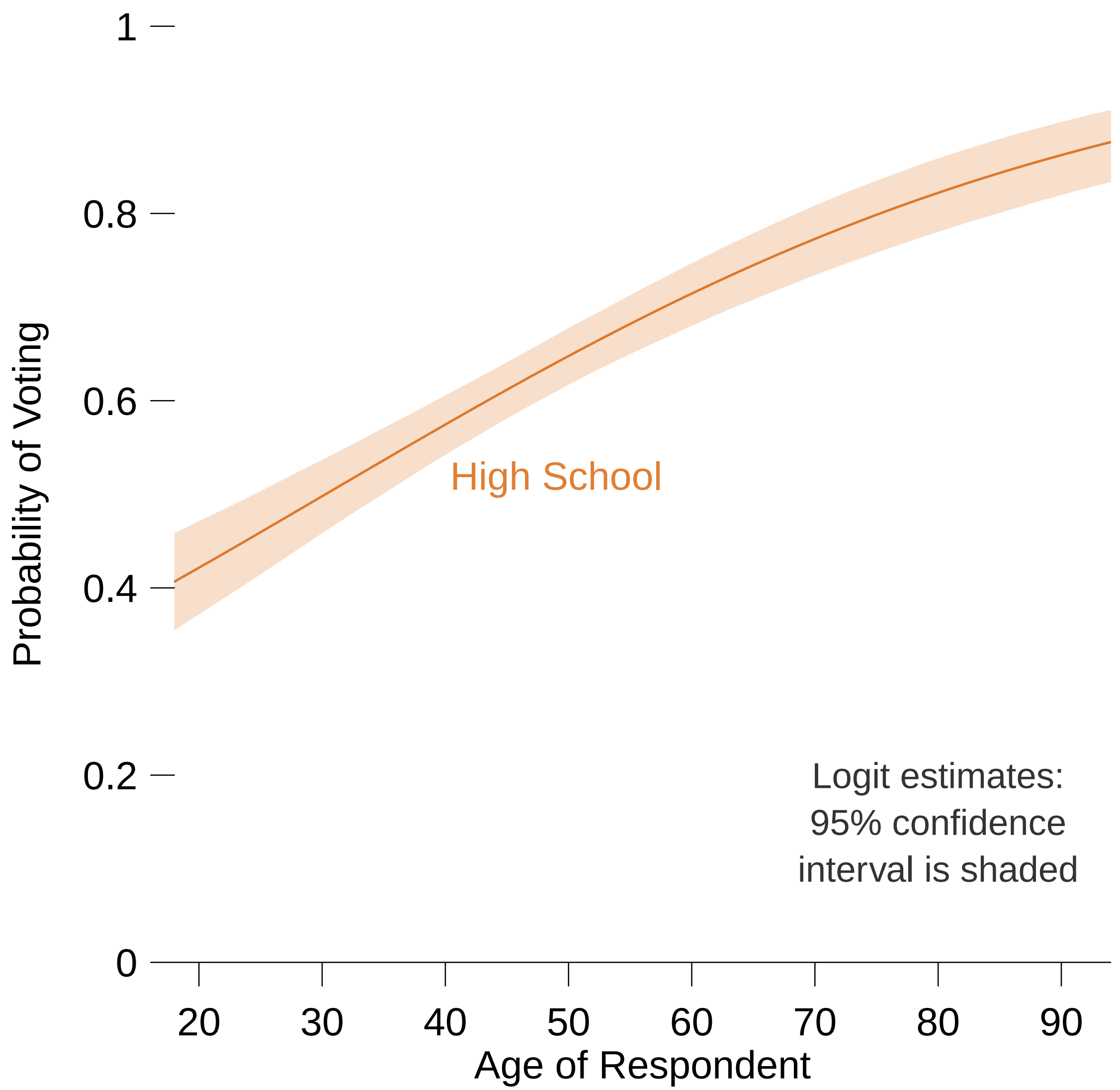
Important shortcut!

If $\mathbb{E}(y|\mu_c) = \hat{\mu}_c$ (the case for linear regression, logit, probit, Poisson, etc.),
you can skip steps 5 and 6: just iterate the fourth step 10,000 times to get your EVs

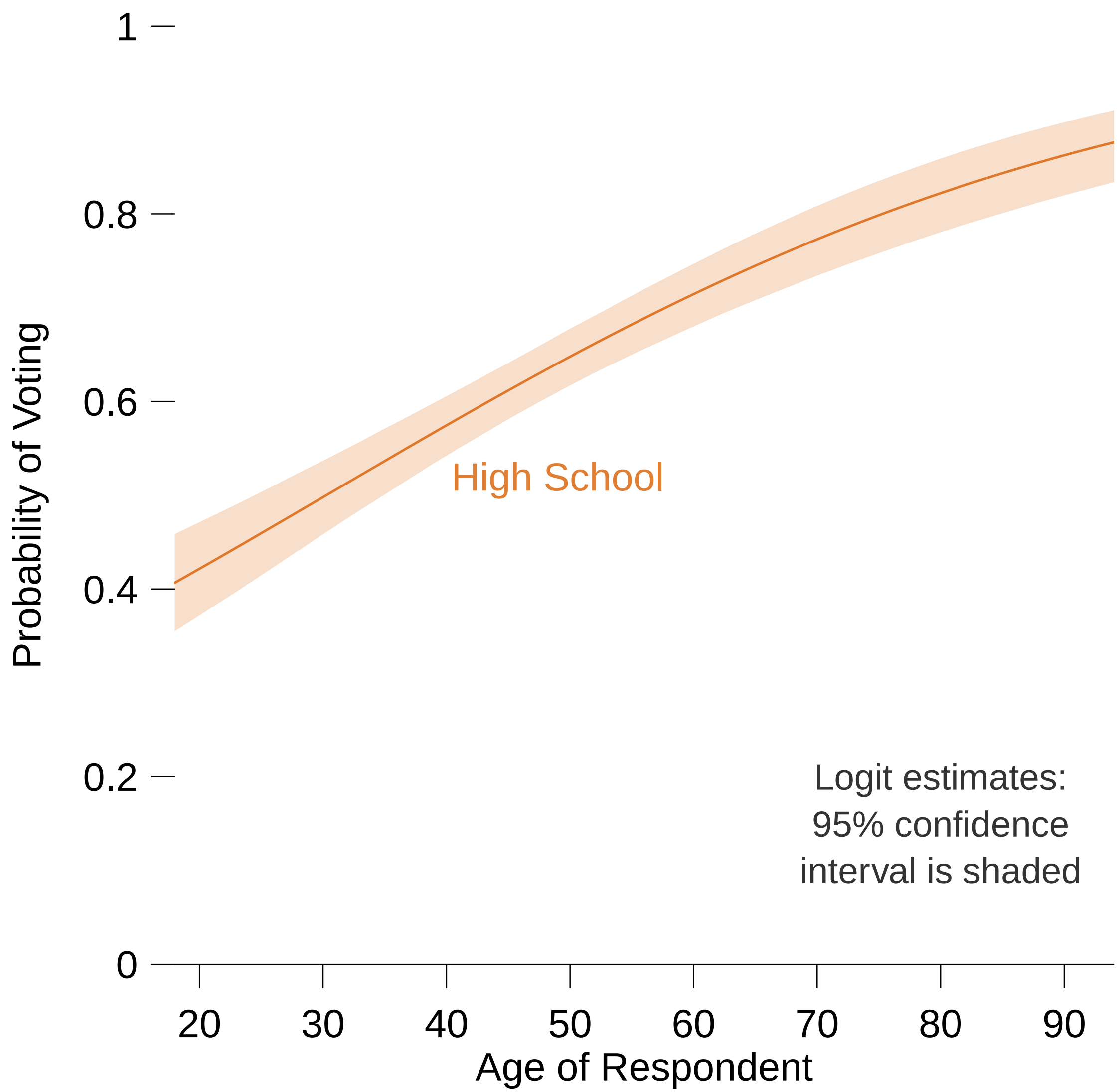
Want predicted values? Need to draw from the model distribution,
to include "fundamental uncertainty"

EVs include only "estimation uncertainty;" PVs include both

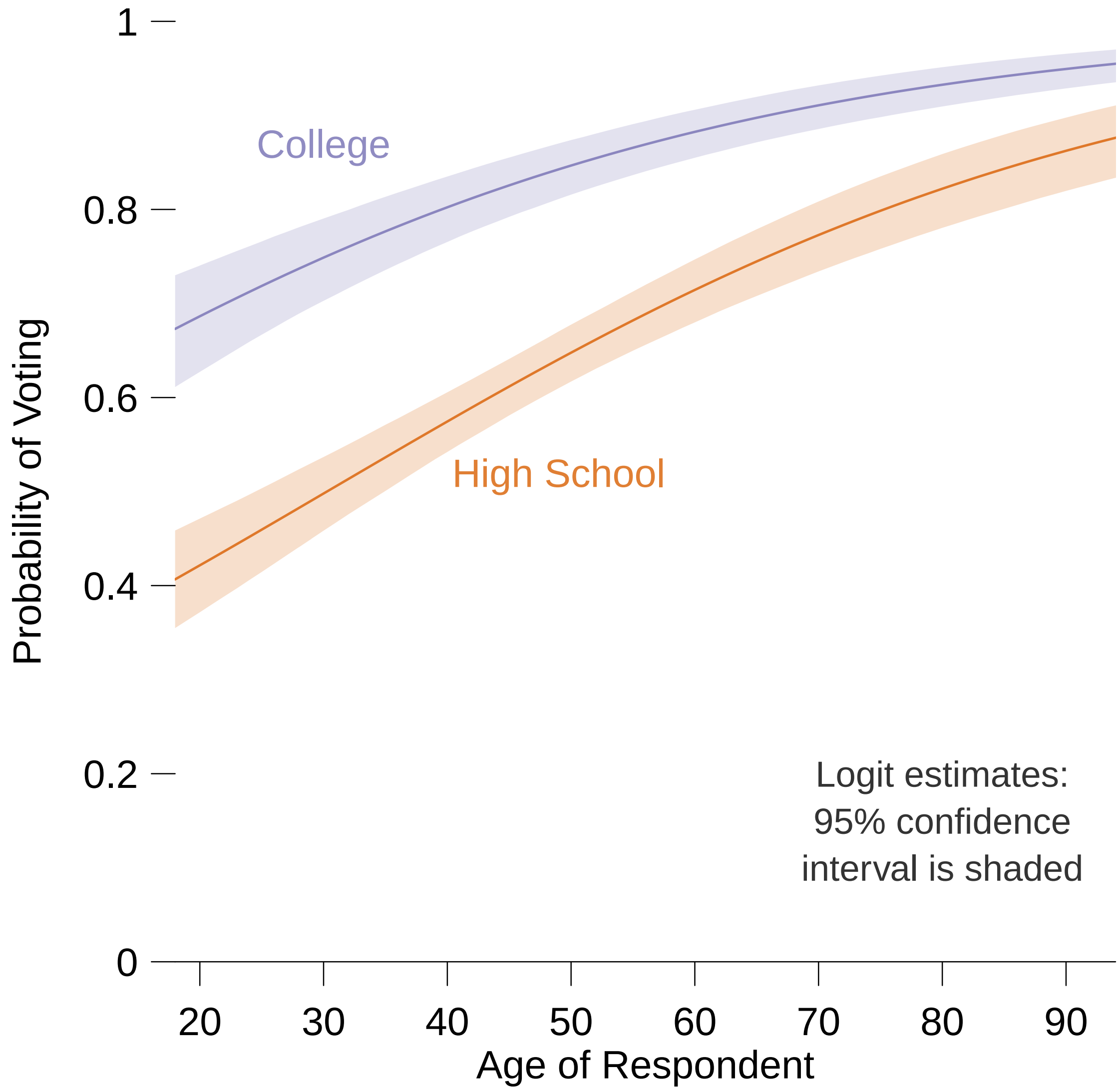
Now let's simulate EVs for the Votes data,
starting with high school grads at each age




Plotting each of our EV's and CI's over the range of ages gives us a slice of the response surface, with uncertainty (10,000 sims each)

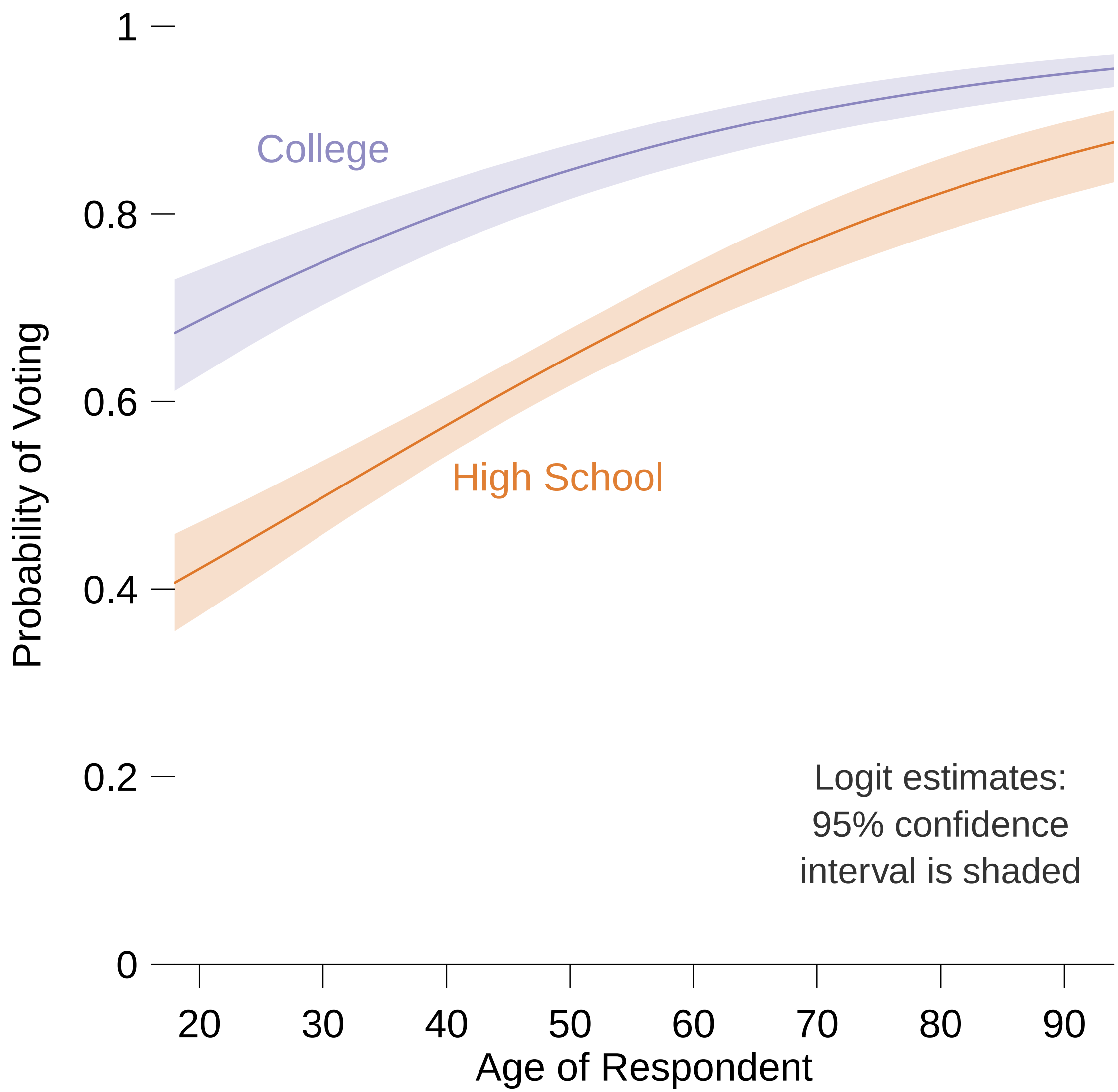



No knowledge of logit or odds ratios required – any curious person can read it




Why stop with one level of education?

Repeating for college grads gives full response surface in a single 2D plot




Claim: this plot is a better summary of the regression than Table 1

If you can publish only the table or the plot, choose the plot

# Save this sort of table for the web appendix

Table 3: *Determinants of voting in the 2000 presidential election.* Logit coefficients estimated using data from the 2000 American National Election Survey.

| | est. | s.e. | $p$-value |
|---|---|---|---|
| Age | $0.075$ | $0.017$ | $< 0.001$ |
| Age$^2$ | $-0.000443$ | $0.000166$ | $0.007$ |
| High School Grad | $1.124$ | $0.180$ | $< 0.001$ |
| College Grad | $1.080$ | $0.131$ | $< 0.001$ |
| Constant | $-3.019$ | $0.418$ | $< 0.001$ |
| log likelihood | $-1101.37$ | | |
| $N$ | $1798$ | | |

Above is a new specification we'll use in subsequent plots. . .

What's changed? How would you interpret this using odds ratios?

# When simulation is the only reasonable option

This lecture builds up a plot from an article posted on the course site (King, Tomz, and Wittenberg, *AJPS* 2000).

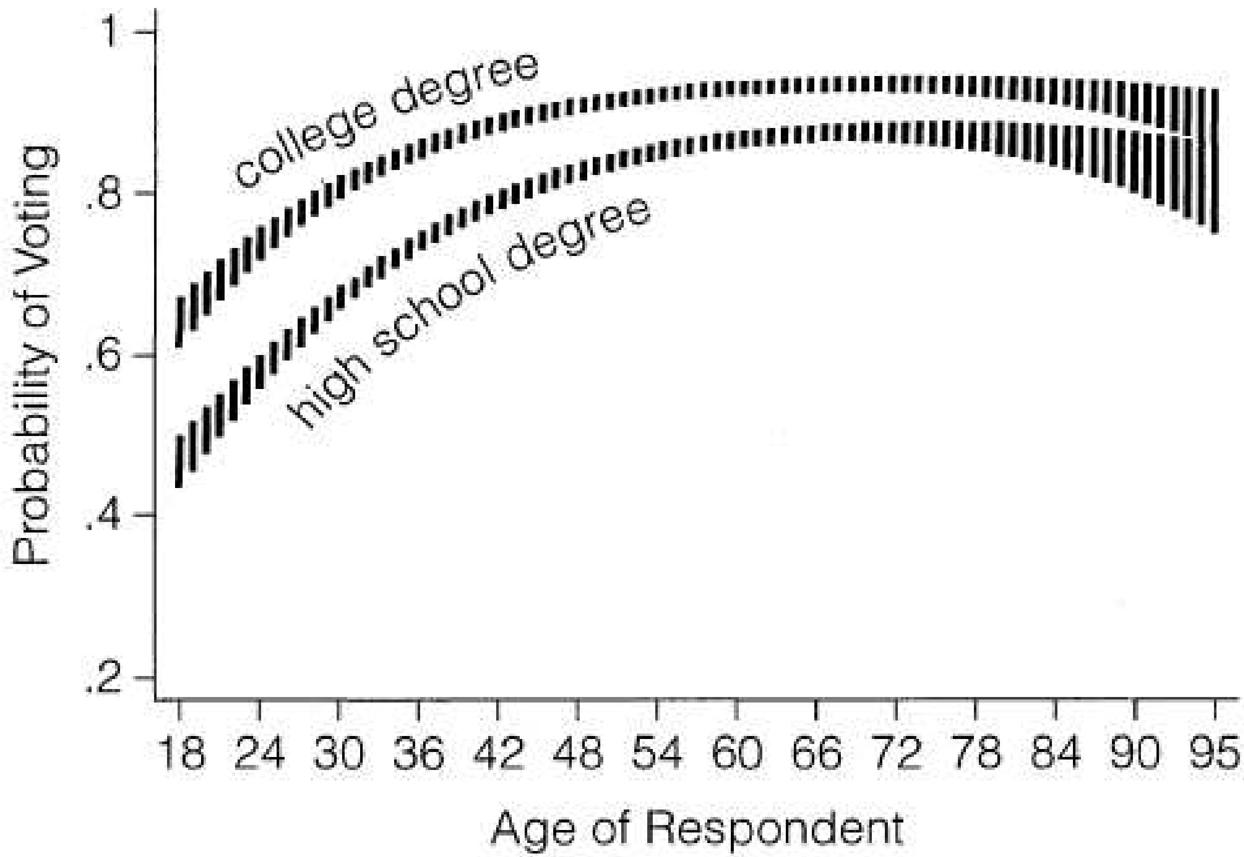
KTW add an extra term, $\mathrm{Age}^2$, which has a negative effect
– intuitively, past some age, voting may be difficult

This would add a great deal of complexity to interpretation under most alternatives:

*If you like odds ratios, you'll love combining several odds ratio interaction terms!*

How hard would it be to add a quadratic term to our simulation method?

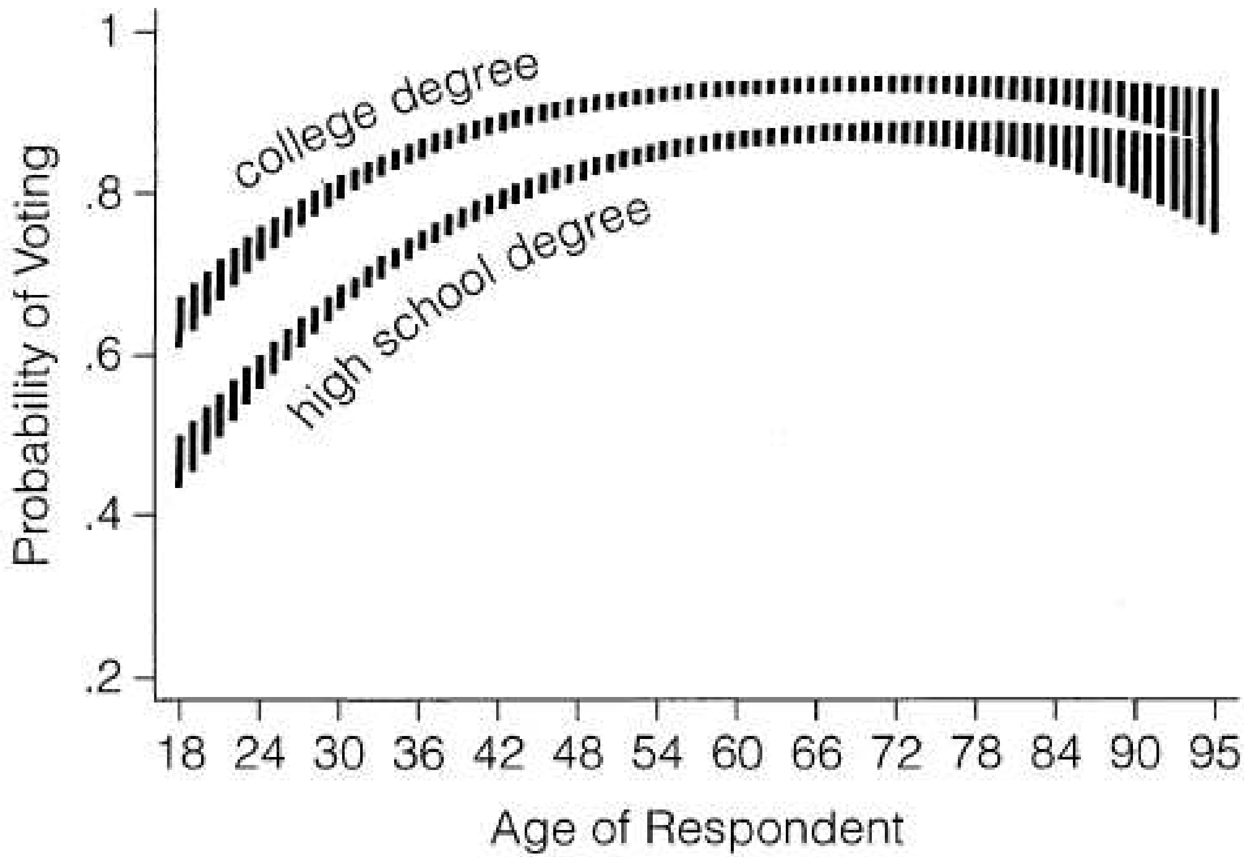
**FIGURE 1** Probability of Voting by Age




Vertical bars indicate 99-percent confidence intervals

$\text{Age}^2$ adds *zero* complexity to the expected values plot

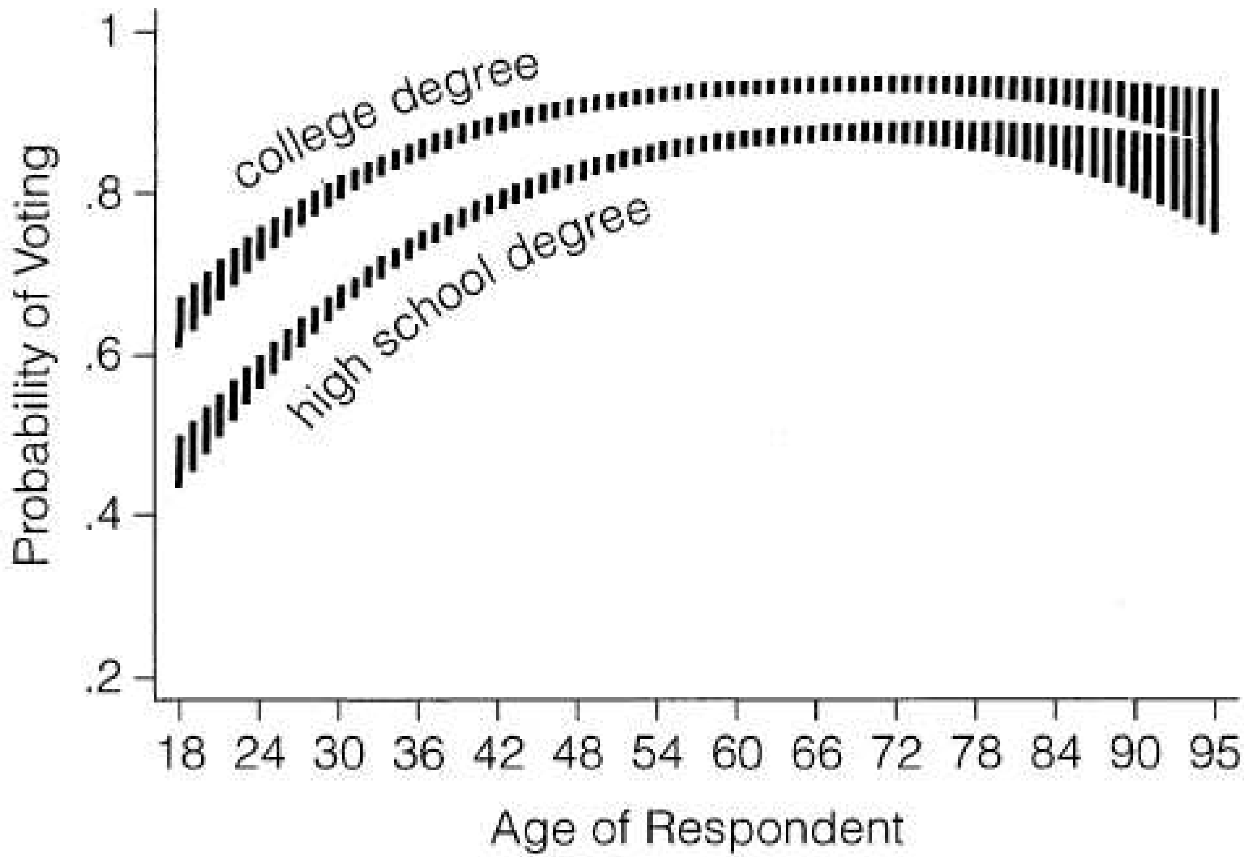
**FIGURE 1** **Probability of Voting by Age**




Vertical bars indicate 99-percent confidence intervals

There's a hidden flaw in this plot: can you spot it?

**FIGURE 1** Probability of Voting by Age




Vertical bars indicate 99-percent confidence intervals

*18 year-old college graduates?* Take care in selecting counterfactuals!

At a minimum, stay inside the sampled space ("convex hull")

# Graphing simulations from models

Challenges for presentation:

1. Our QoIs and their CIs trace out non-linear functions

2. Contain lots of information: easy to produce hundreds of data points

3. Often, counterfactuals producing QoIs vary on multiple dimensions

Once we simulate our quantities of interest and confidence intervals,
we need visual methods for presenting them

I offer two `R` packages on my website to make this easier:

**simcf** – generic primitive functions for simulating QoIs

**tile** – a graphics package for plotting QoIs with CIs

Learn more about tile in CSSS 569; only a taste here

# Graphing simulations from models

Two proposed general graphical methods using the **tile** package:

- **lineplot**: For models with at least one continuously varying counterfactual variable, use a line varying over that counterfactual, and as many lines as you have discrete counterfactuals

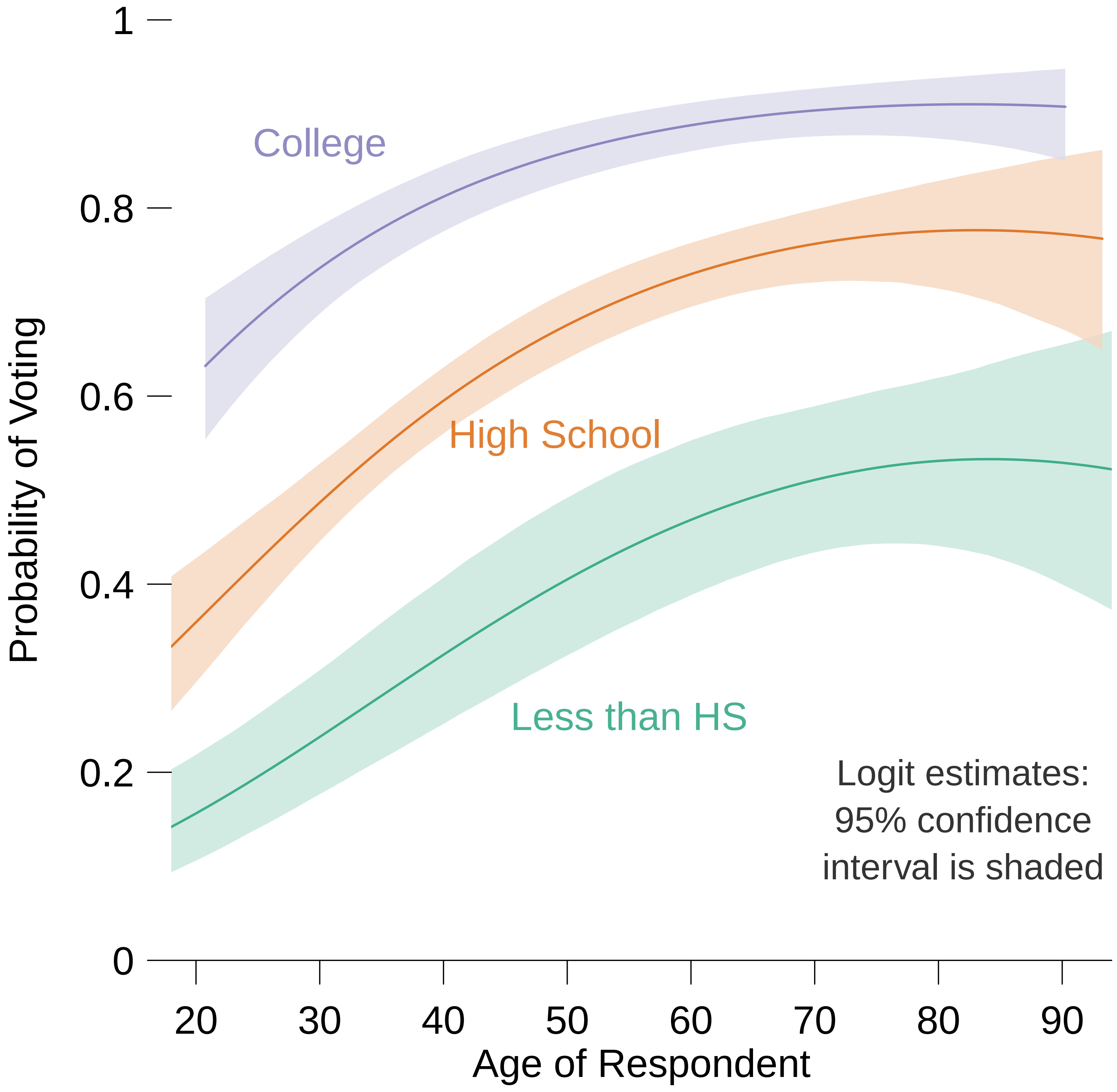
  Note this will focus attention on your continuous counterfactuals, which may not be your intention

- **ropeladder**: For *any* model, including those with only discrete counterfactual variables, or a combination of discrete and continuous

  ropeladders put equal attention on covariates that vary continuously and those that vary discretely.

  Can accommodate large numbers of different counterfactual scenarios

Let's use lineplots for our voting example, since we have one continuous and one categorical covariate


1
0.8
0.6
0.4
0.2
0
Probability of Voting
College
High School
Less than HS
Logit estimates:
95% confidence
interval is shaded
20
30
40
50
60
70
80
90
Age of Respondent

# Logit interpretation example

```
# Clear memory
rm(list=ls())

# Load libraries
library(MASS)
library(simcf)
library(tile)
library(RColorBrewer)
```

Notes:

I usually start programs with a command to clear memory

Then I load all the libraries I need

The `tile` and `simcf` libraries are available at `chrisadolph.com` under Software

# Logit interpretation example

```
# Load data
file <- "nes00a.csv"
data <- read.csv(file, header=TRUE)

# Estimate logit model using optim()
# Construct variables and model objects
y <- data$vote00
x <- cbind(data$age,data$age^2,data$hsdeg,data$coldeg)
```

Notes:

To use optim, we need the data in a vector of y's and a matrix of X's

If we had missing data, we would also need to listwise delete.

Note that you need to `cbind()` the x and y data together before listwise deleting with `na.omit()`, then pull the x and y apart again.

We'll look at an easier way in a moment.

# Logit interpretation example

```
# Likelihood function for logit
llk.logit <- function(param,y,x) {
  os <- rep(1,length(x[,1]))
  x <- cbind(os,x)
  b <- param[ 1 : ncol(x) ]
  xb <- x%*%b
  sum( y*log(1+exp(-xb)) + (1-y)*log(1+exp(xb)));
               # optim is a minimizer, so min -ln L(param|y)
}
```

Notes:

We need to program in the likelihood as a function that can take in the parameters, data, and covariates, and pass back the likelihood at those parameters

# Logit interpretation example

```
# Fit logit model using optim
ls.result <- lm(y~x)  # use ls estimates as starting values
stval <- ls.result$coefficients  # initial guesses
logit.result.opt <- optim(stval,llk.logit,method="BFGS",
                          hessian=T,y=y,x=x)
                   # call minimizer procedure
pe.opt <- logit.result.opt$par   # point estimates
vc.opt <- solve(logit.result.opt$hessian)  # var-cov matrix
se.opt <- sqrt(diag(vc.opt))    # standard errors
ll.opt <- -logit.result.opt$value  # likelihood at maximum
```

Notes:

After we load the data and program the likelihood, we send them to optim

Optim is a minimizer, which sends back the parameters that make $-L$ as small as possible

Then we extract the point estimates, variance-covariance, standard errors, and likelihod at the max

## Logit interpretation example

```
# Estimate logit model using glm()
# Set up model formula and model specific data frame
model <- vote00 ~ age + I(age^2) + hsdeg + coldeg
mdata <- extractdata(model, data, na.rm=TRUE)
```

Notes:

Alternatively, we could use the command `glm()` to do logit and several other similar models.

`glm()` needs a model formula, and a dataframe with listwise deleted data

`extractdata()` from `simcf` simplifies listwise deletion

We keep all data in a dataframe while deleting only based on model variables

# Logit interpretation example

```
# Run logit & extract results
logit.result <- glm(model, family=binomial, data=mdata)
pe.glm <- logit.result$coefficients  # point estimates
vc.glm <- vcov(logit.result)         # var-cov matrix
```

Notes:

Running the model using `glm()` uses less code, but ends with the same estimates

`glm()` won't work for ever MLE in this course, though;
just those that belong to the Exponential family

# Logit interpretation example

```
## Simulate quantities of interest using simcf
##
## We could do this from the optim or glm results;
## here, we do from glm

# Simulate parameter distributions
sims <- 10000
simbetas <- mvrnorm(sims, pe.glm, vc.glm)
```

Notes:

Regardless of whether we use optim or glm,
we use the same simulation method to understand the results

We summarize the model using draws from the predictive distributions of the estimated parameters

This is approximately MVN for all MLEs,
regardless of the distribution of the response variable

# Logit interpretation example

```
# Set up counterfactuals:  all ages, each of three educations
xhyp <- seq(18,97,1)
nscen <- length(xhyp)
nohsScen <- hsScen <- collScen <- cfMake(model, mdata, nscen)
```

Notes:

We set up counterfactual levels of the continuous covariate of interest, age

This determines the number of scenarios for our lineplots

We then make counterfactual objects using the `simcf` package (on course site)

Each of these objects has a many scenarios as there are counterfactuals of age

We need three sets of these counterfactuals – one for each level of education

**This step (and following) varies greatly depending on which counterfactuals & graphics you want to make.**

# Logit interpretation example

```
for (i in 1:nscen) {
  # No High school scenarios (loop over each age)
  nohsScen <- cfChange(nohsScen, "age", x = xhyp[i], scen = i)
  nohsScen <- cfChange(nohsScen, "hsdeg", x = 0, scen = i)
  nohsScen <- cfChange(nohsScen, "coldeg", x = 0, scen = i)
}
```

Notes:

We loop over each level of age

At each iteration, we need to set the appropriate values of the covariates in our counterfactuals

`cfChange()` takes a counterfactual scenario under construction, the name of a covariate, a new level for the variable, and the number of the scenario to be changed to that level

We apply `cfChange()` repeatedly to setup all our scenarios

We could set in addition `xpre` to do first differences between `x` and `xpre`

# Logit interpretation example

```
for (i in 1:nscen) {
  # No High school scenarios (loop over each age)
  nohsScen <- cfChange(nohsScen, "age", x = xhyp[i], scen = i)
  nohsScen <- cfChange(nohsScen, "hsdeg", x = 0, scen = i)
  nohsScen <- cfChange(nohsScen, "coldeg", x = 0, scen = i)

  # HS grad scenarios (loop over each age)
  hsScen <- cfChange(hsScen, "age", x = xhyp[i], scen = i)
  hsScen <- cfChange(hsScen, "hsdeg", x = 1, scen = i)
  hsScen <- cfChange(hsScen, "coldeg", x = 0, scen = i)

  # College grad scenarios (loop over each age)
  collScen <- cfChange(collScen, "age", x = xhyp[i], scen = i)
  collScen <- cfChange(collScen, "hsdeg", x = 1, scen = i)
  collScen <- cfChange(collScen, "coldeg", x = 1, scen = i)
}
```

# Logit interpretation example

```
# Simulate expected probabilities for all scenarios
nohsSims <- logitsimev(nohsScen, simbetas, ci=0.95)
hsSims <- logitsimev(hsScen, simbetas, ci=0.95)
collSims <- logitsimev(collScen, simbetas, ci=0.95)
```

Notes:

We run each set of counterfactuals through an appropriate simulator from `simcf`

Since we are using a logit model, and want expected probabilities, we use `logitsimev`, which stands for Logit Simulate Expected Values

The output is a list with point estimates `pe`, lower bounds `lower`, and upper bounds `upper`

# Logit interpretation example

```
# Get 3 nice colors for traces
col <- brewer.pal(3,"Dark2")

# Set up lineplot traces of expected probabilities
nohsTrace <- lineplot(x=xhyp,
                      y=nohsSims$pe,
                      lower=nohsSims$lower,
                      upper=nohsSims$upper,
                      col=col[1],
                      extrapolate=list(data=mdata,
                                       cfact=nohsScen$x,
                                       omit.extrapolated=TRUE),
                      plot=1)
```

Notes:

We put the simulated responses, CIs, and corresponding hypothetical ages into a trace

We need one trace for each education level

# Logit interpretation example

```
# Get 3 nice colors for traces
col <- brewer.pal(3,"Dark2")

# Set up lineplot traces of expected probabilities
nohsTrace <- lineplot(x=xhyp,
                      y=nohsSims$pe,
                      lower=nohsSims$lower,
                      upper=nohsSims$upper,
                      col=col[1],
                      extrapolate=list(data=mdata,
                                       cfact=nohsScen$x,
                                       omit.extrapolated=TRUE),
                      plot=1)
```

Notes:

Optionally, we include instructions to omit extrapolations outside the convex hull

Inputs to `extrapolate` are very picky: we need the model data, and the hypothetical data in exactly the same order of columns

# Logit interpretation example

```
# Get 3 nice colors for traces
col <- brewer.pal(3,"Dark2")

# Set up lineplot traces of expected probabilities
nohsTrace <- lineplot(x=xhyp,
                      y=nohsSims$pe,
                      lower=nohsSims$lower,
                      upper=nohsSims$upper,
                      col=col[1],
                      extrapolate=list(data=mdata,
                                       cfact=nohsScen$x,
                                       omit.extrapolated=TRUE),
                      plot=1)
```

Notes:

Leave the `extrapolate=` lines out if they are giving you trouble

*Start small with `tile` code, then add features!*

# Logit interpretation example

```
hsTrace <- lineplot(x=xhyp,
                    y=hsSims$pe,
                    lower=hsSims$lower,
                    upper=hsSims$upper,
                    col=col[2],
                    extrapolate=list(data=mdata,
                                     cfact=hsScen$x,
                                     omit.extrapolated=TRUE),
                    plot=1)

collTrace <- lineplot(x=xhyp,
                      y=collSims$pe,
                      lower=collSims$lower,
                      upper=collSims$upper,
                      col=col[3],
                      extrapolate=list(data=mdata,
                                       cfact=collScen$x,
                                       omit.extrapolated=TRUE),
                      plot=1)
```

# Logit interpretation example

```
# Set up traces with labels and legend
labelTrace <- textTile(labels=c("Less than HS", "High School",
                                "College"),
                       x=c( 55,    49,     30),
                       y=c( 0.26,  0.56,   0.87),
                       col=col,
                       plot=1)

legendTrace <- textTile(labels=c("Logit estimates:",
                                 "95% confidence",
                                 "interval is shaded"),
                        x=c(82, 82, 82),
                        y=c(0.2, 0.15, 0.10),
                        cex=0.9,
                        plot=1)
```

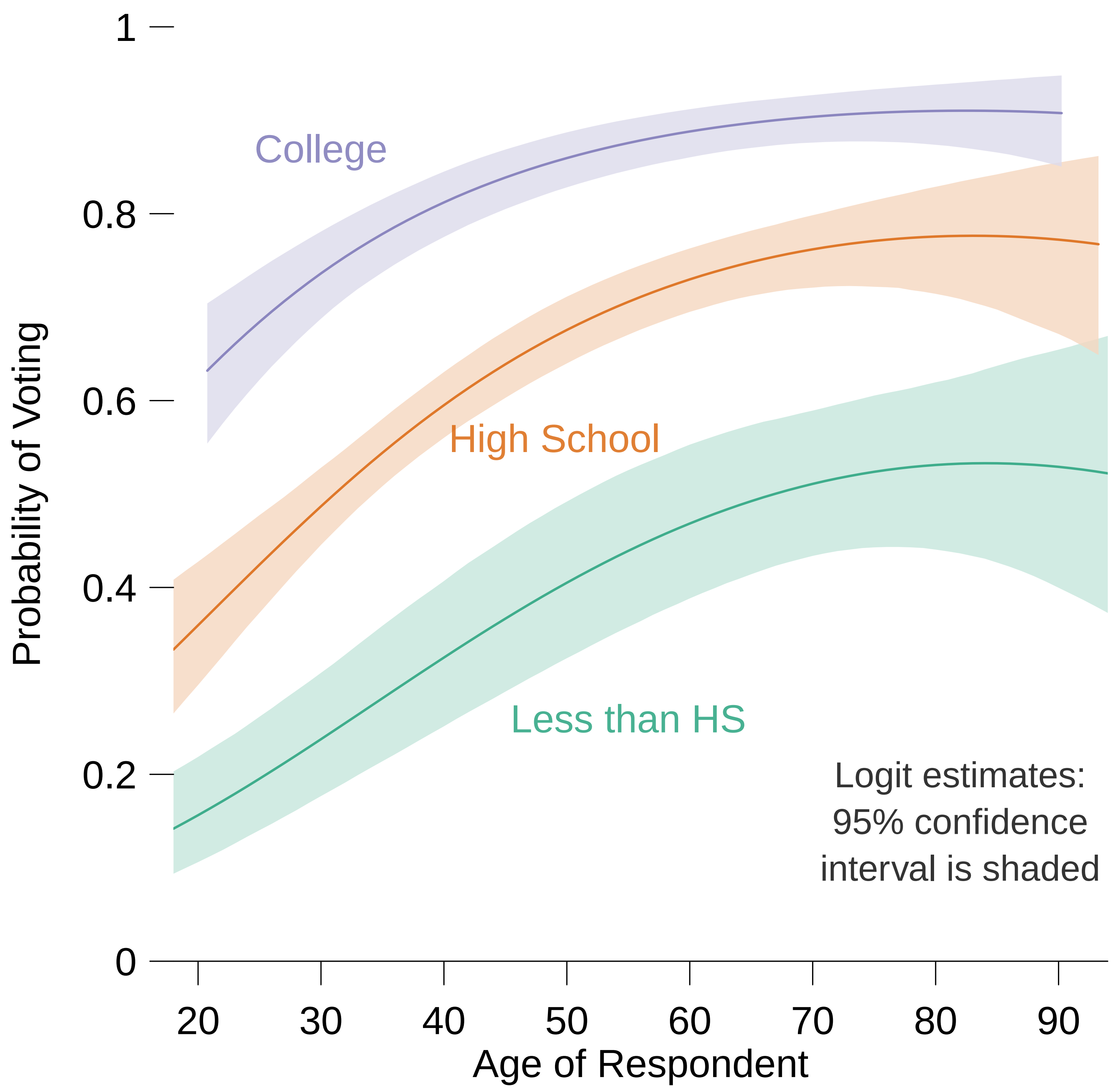
# Logit interpretation example

```
# Plot traces using tile
tile(nohsTrace,
     hsTrace,
     collTrace,
     labelTrace,
     legendTrace,
     limits=c(18,94,0,1),
     xaxis=list(at=c(20,30,40,50,60,70,80,90)),
     yaxis=list(label.loc=-0.5, major=FALSE),
     xaxistitle=list(labels="Age of Respondent"),
     yaxistitle=list(labels="Probability of Voting"),
     width=list(null=5,yaxistitle=4,yaxis.labelspace=-0.5),
     output=list(file="educationEV",width=5.5)
     )
```

This makes a pdf called `educationEV.pdf`;
leave out the `output=` line to have this plot to the screen

The `width` argument lets of tweak the layout.

Setting `null=5` stretches the size of each plot in the graphic to 5 inches from a default of 2



1
0.8
0.6
0.4
0.2
0
Probability of Voting
College
High School
Less than HS
Logit estimates:
95% confidence
interval is shaded
20
30
40
50
60
70
80
90
Age of Respondent

# Summing up . . .

Now you know how to

- Pick a binary choice model
- Obtain ML estimates of it
- Interpret those estimates
- Present your results to a broad audience

But are your results are any good?

Essential step: testing goodness of fit.

# Evaluating the fit of logit models

So far, we've learned how to estimate binary data models

*and* how to interpret these estimates

But are the estimates *any good*?

As before, everything will be in terms of logit, but most applies, *mutatis mutandis*, to probit, log-log, clog-log, scobit, etc.

Most of it even applies to all other MLEs – I'll note where it doesn't

# Recall the Votes data

Response is individual turn-out, a binary variable

People either vote ($\text{Vote}_i = 1$), or they don't ($\text{Vote}_i = 0$)

Many factors could influence turn-out; we focused on age and education

American National Election Survey: "Did you vote in 2000 election?"

```
      vote00 age hsdeg coldeg married  vote96
 [1,]      1  49     1      0       1       1
 [2,]      0  35     1      0       1       0
 [3,]      1  57     1      0       1       1
 [4,]      1  63     1      0       0       1
 [5,]      1  40     1      0       1       1
 [6,]      1  77     0      0       1       1
 [7,]      0  43     1      0       1       0
 [8,]      1  47     1      1       0       1
 [9,]      1  26     1      1       0       1
[10,]      1  48     1      0       0       1
 ...
```

Note two new variables

# Comparing two models

We modeled $\mathrm{Vote_i}$ as a function of $\mathrm{Age_i}$, $\mathrm{HSDeg_i}$, and $\mathrm{ColDeg_i}$

Let's call this Model 1 (M1)

Now add the variable $\mathrm{Married}$ to make Model 2 (M2)

Formally,

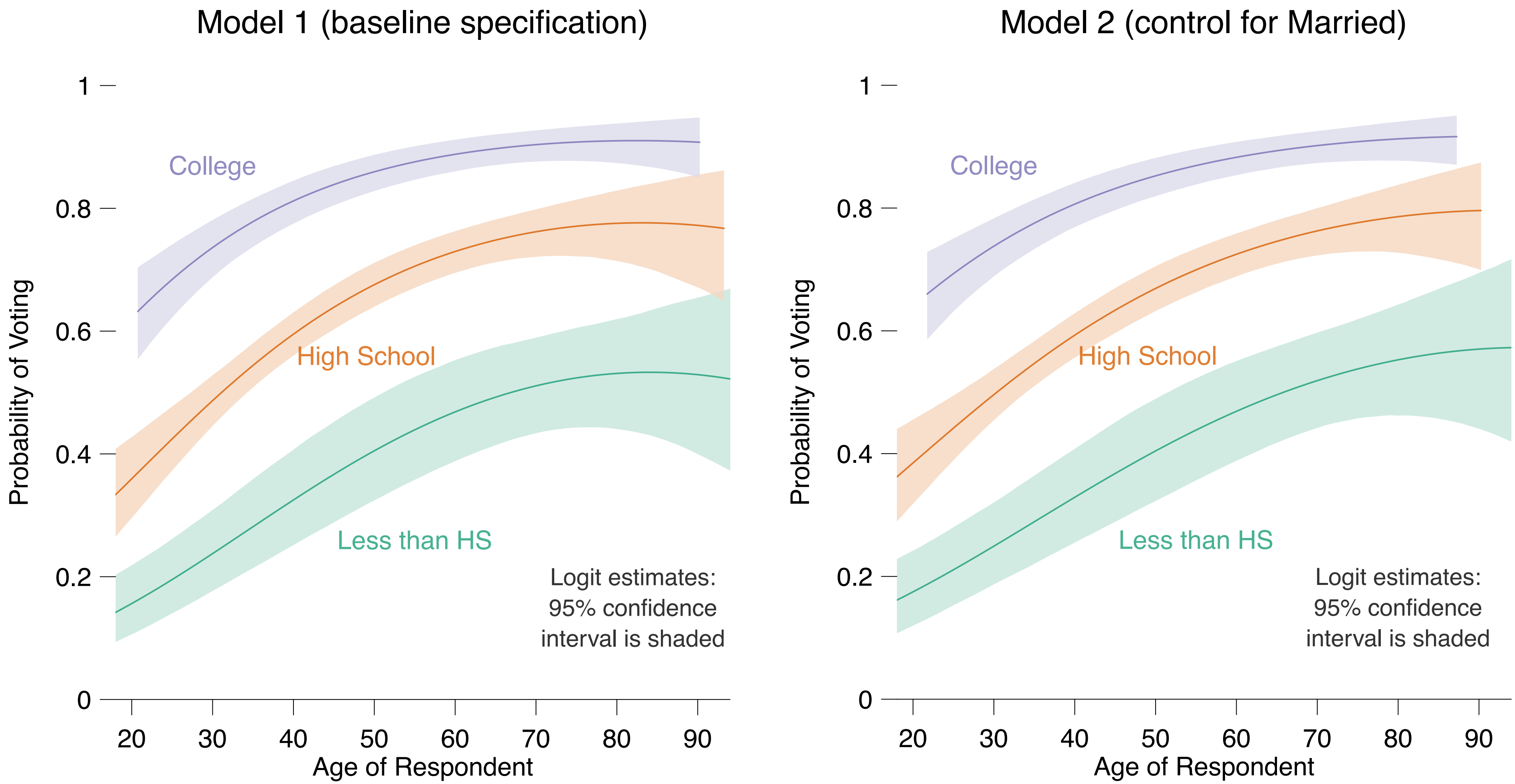
$$\begin{aligned}\mathrm{Vote}_i &\sim \mathrm{Bernoulli}(\pi_i) \\ \pi_i &= \mathrm{logit}^{-1}(\beta_0 + \beta_1 \mathrm{Age}_i + \beta_2 \mathrm{Age}_i^2 + \beta_3 \mathrm{HSDeg}_i + \beta_4 \mathrm{ColDeg}_i + \beta_5 \mathrm{Married}_i)\end{aligned}$$

Like age and education, we expect being married to increase Pr(Vote)

Table 4: *Determinants of voting in the 2000 presidential election.* Entries are logit coefficients, with standard errors in parentheses. Data taken from the 2000 American National Election Survey.

| | M1 | M2 |
|---|---|---|
| Age | $0.075$ | $0.061$ |
| | $(0.017)$ | $(0.017)$ |
| Age$^2$ | $-0.000443$ | $-0.000318$ |
| | $(0.000166)$ | $(0.000170)$ |
| High School Grad | $1.124$ | $1.099$ |
| | $(0.180)$ | $(0.181)$ |
| College Grad | $1.080$ | $1.053$ |
| | $(0.131)$ | $(0.132)$ |
| Married | | $0.373$ |
| | | $(0.110)$ |
| Constant | $-3.019$ | $-2.866$ |
| | $(0.418)$ | $(0.421)$ |
| log likelihood | $-1101.370$ | $-1099.283$ |
| $N$ | $1783$ | $1783$ |

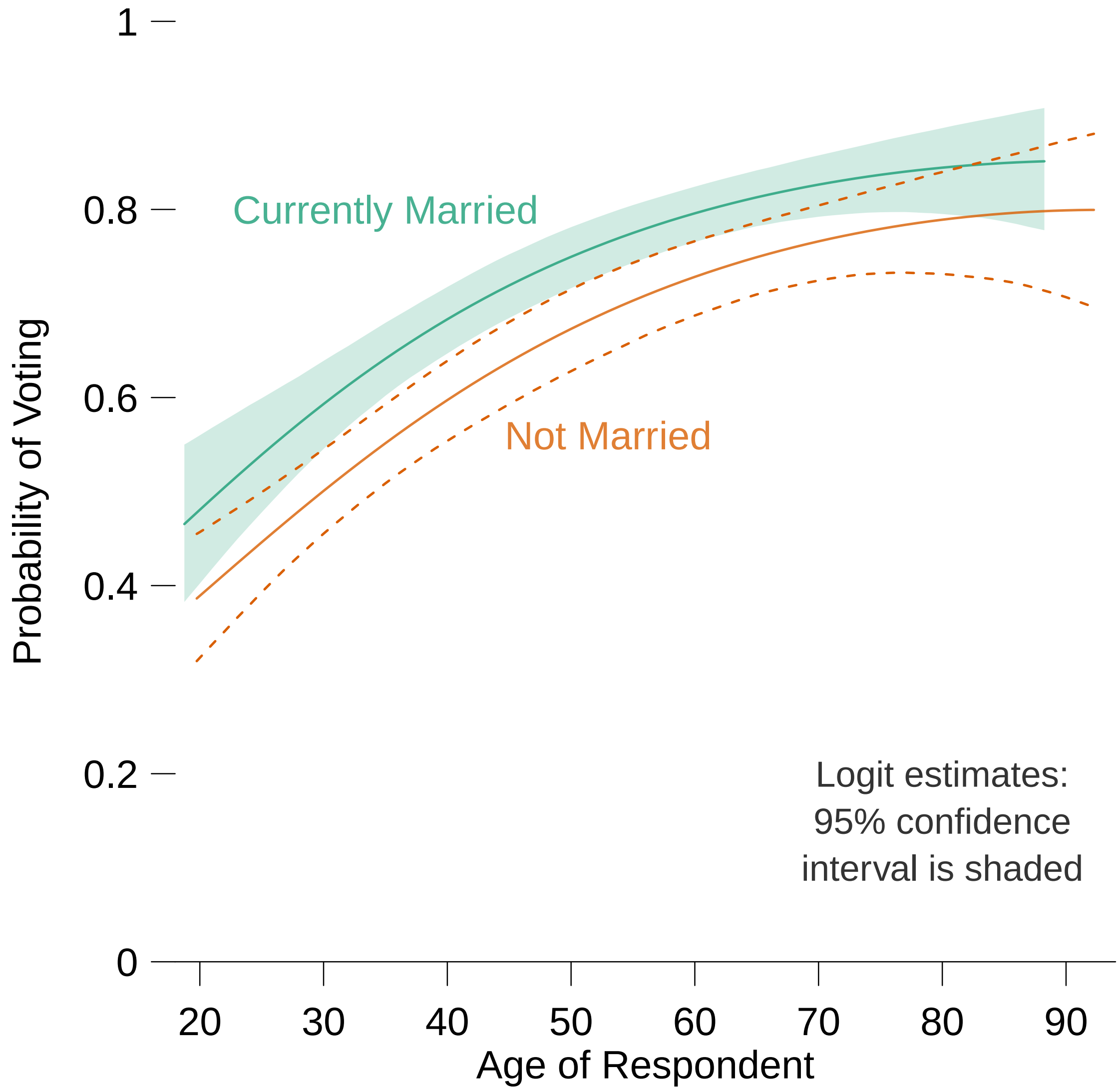
A conventional way to compare two logit models    *Substantive difference?*




An intelligible way to compare two logit models

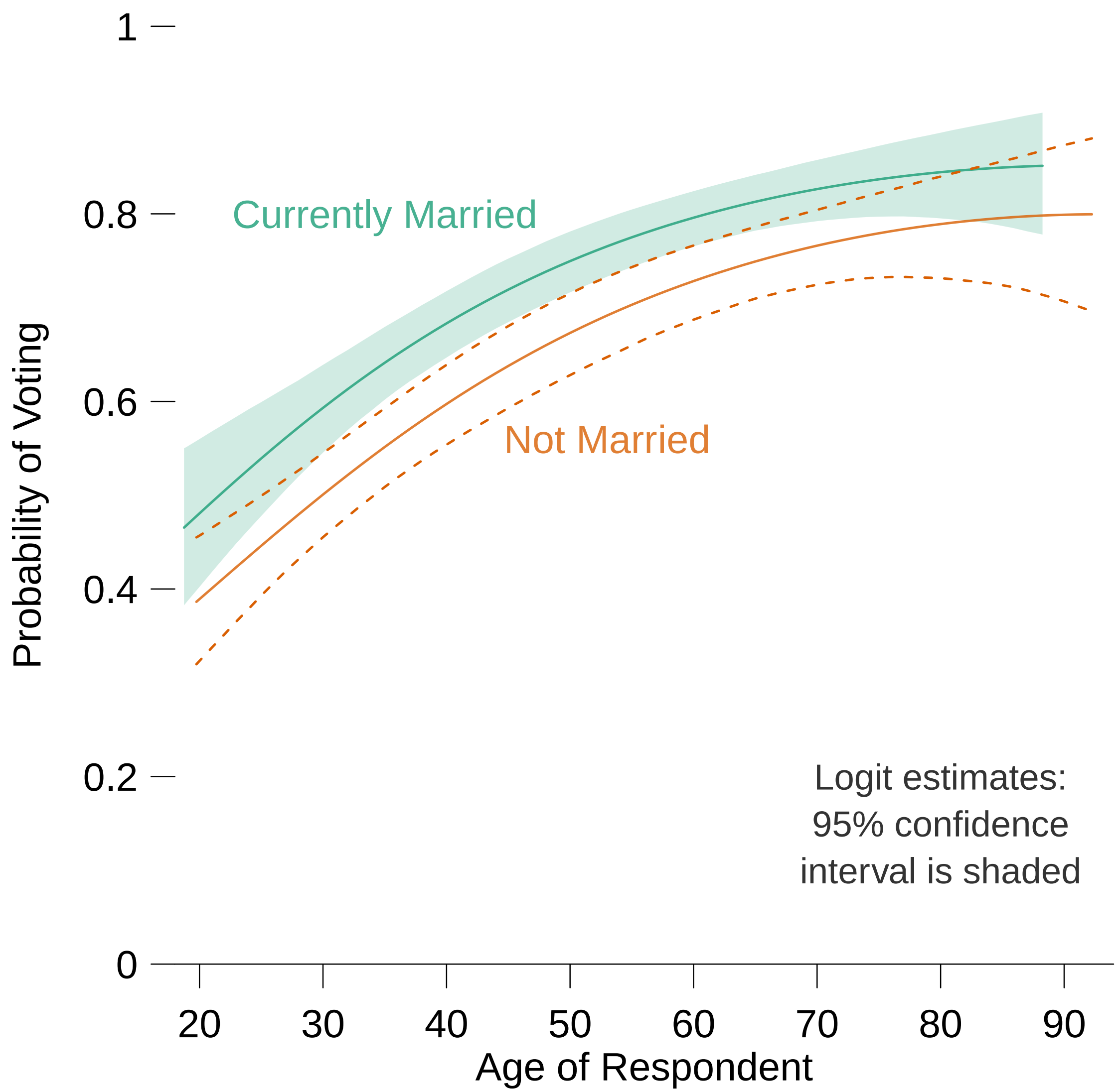
Holding marriage constant at its mean,
the age and education effects haven't discernably changed

*Same words to describe either plot ≡ "substantively the same relationship"*



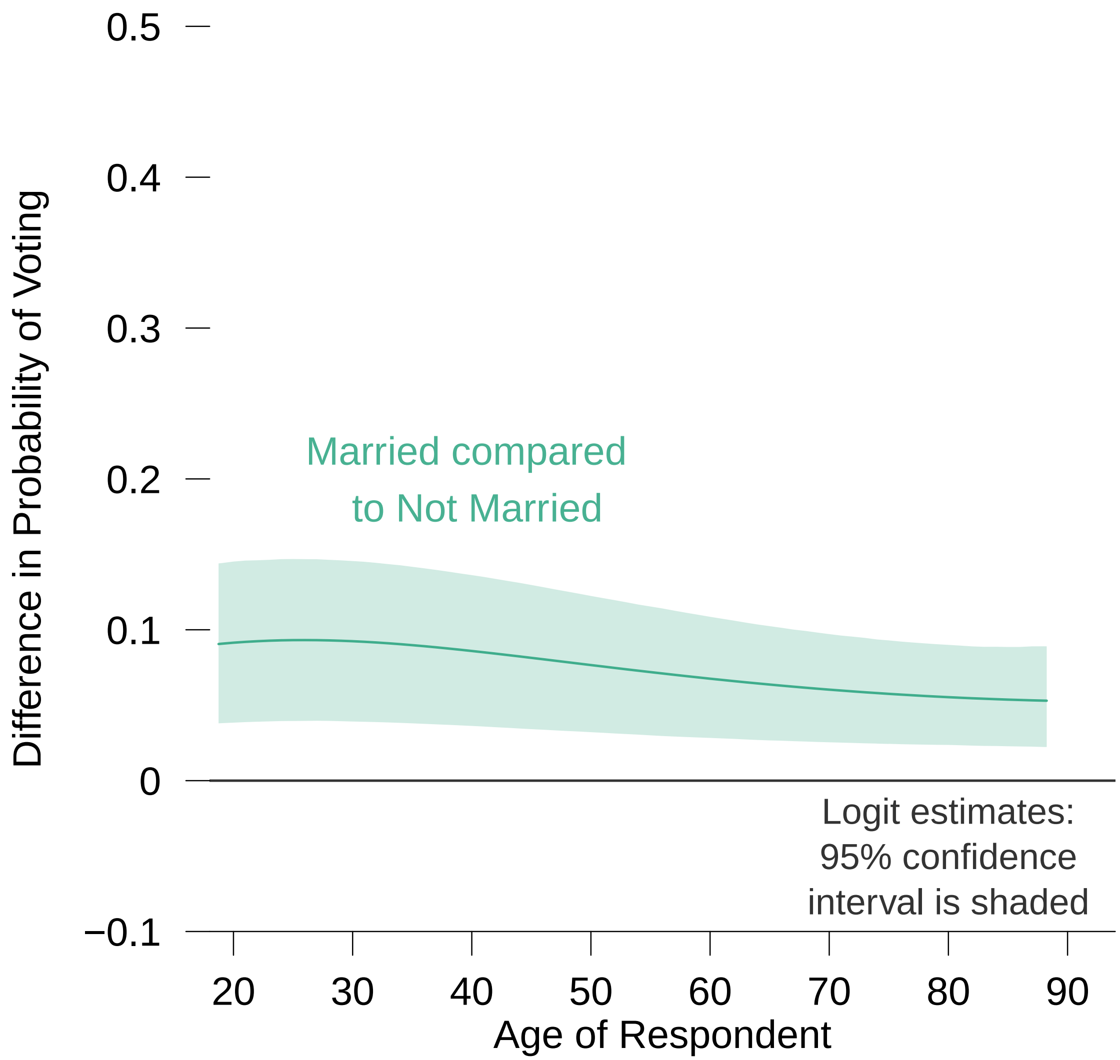

Holding age and education constant at their means, marriage has a moderate effect

Is this effect statistically significant?



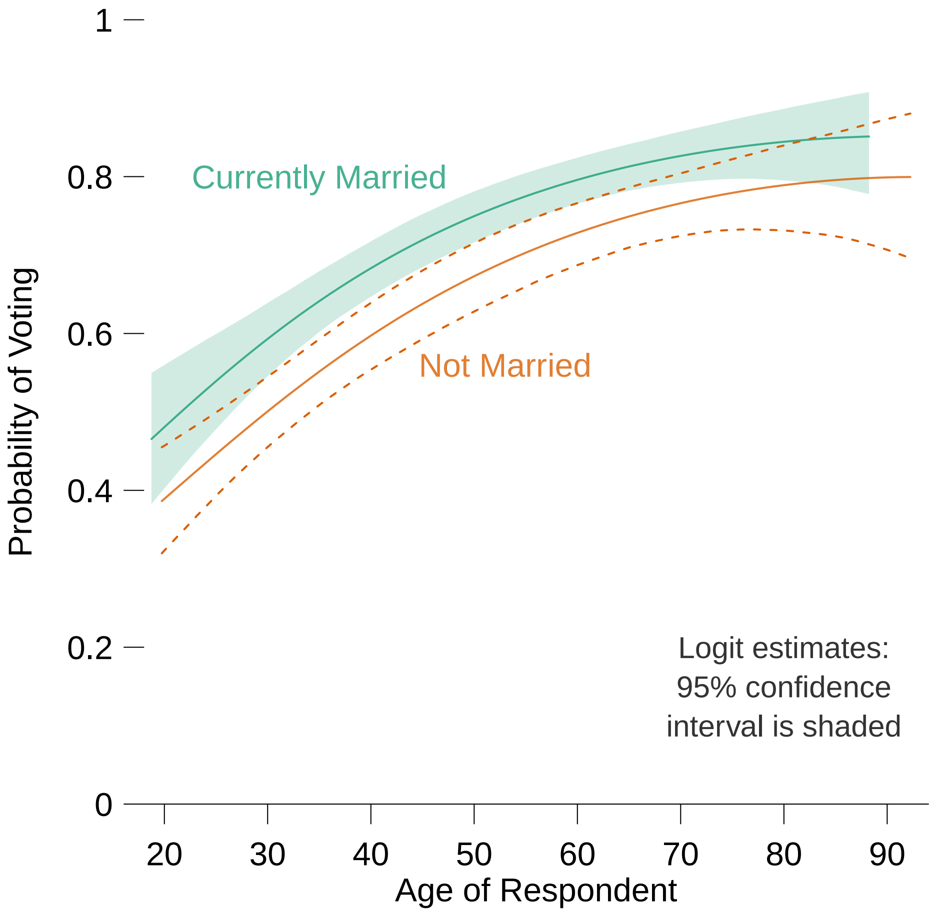

A common misconception about confidence intervals:
*expected values* with overlapping CIs don't always imply an insignificant *difference*

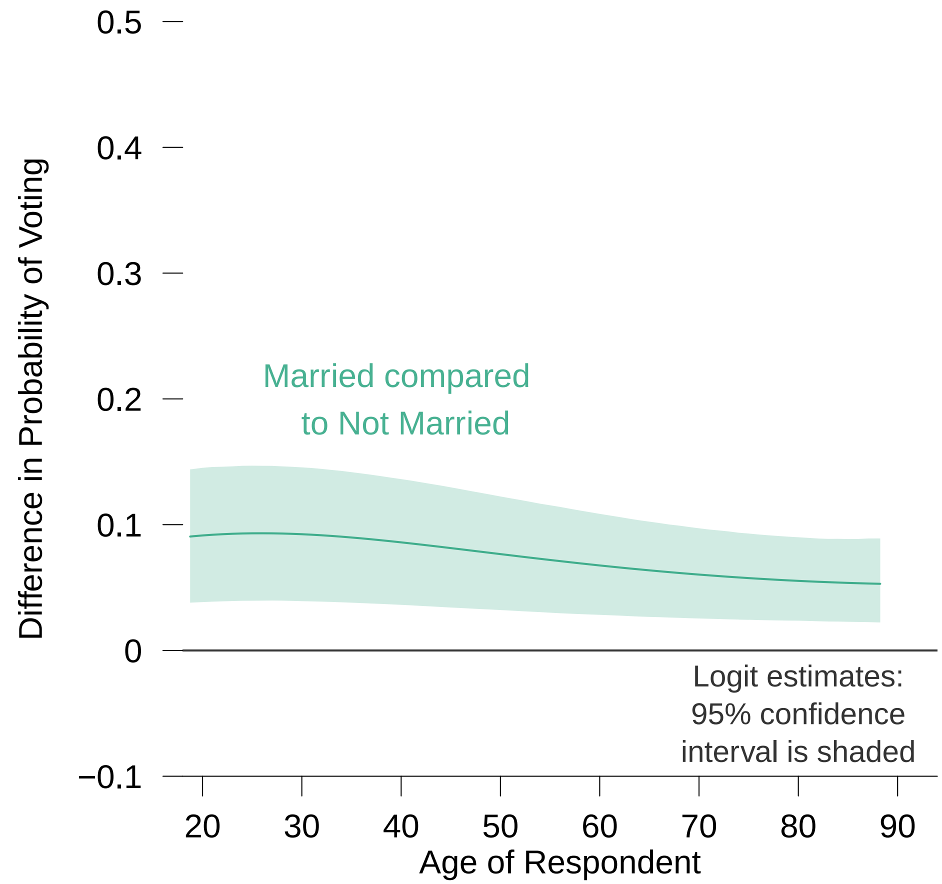
Not necessarily the case in "close calls"




Right way to assess statistical significance: simulate the CI of the first difference

Here the first difference is always bounded away from zero, hence always significant



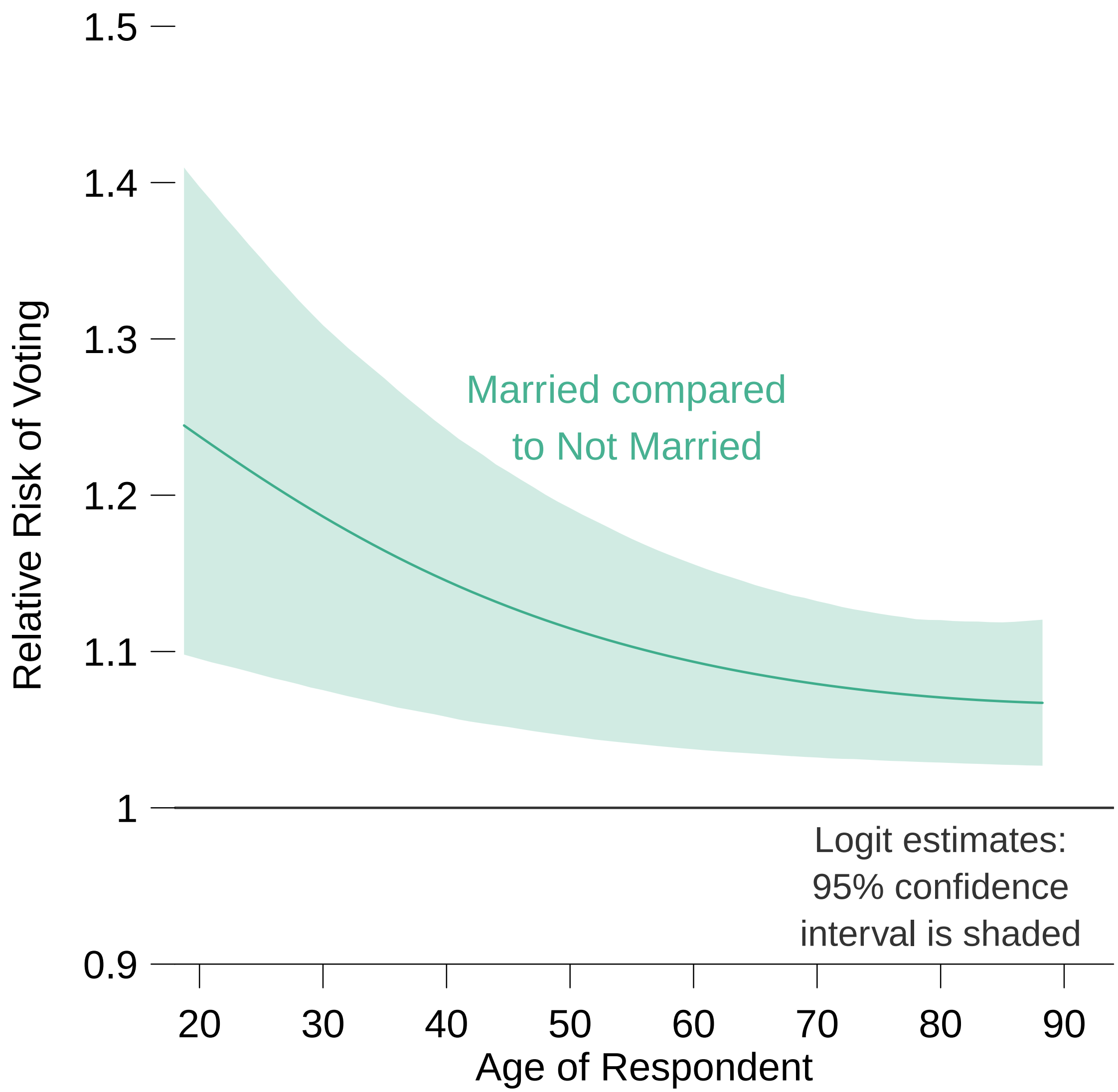




Why the difference?

EV estimates difference & *location*; estimating more things increases uncertainty

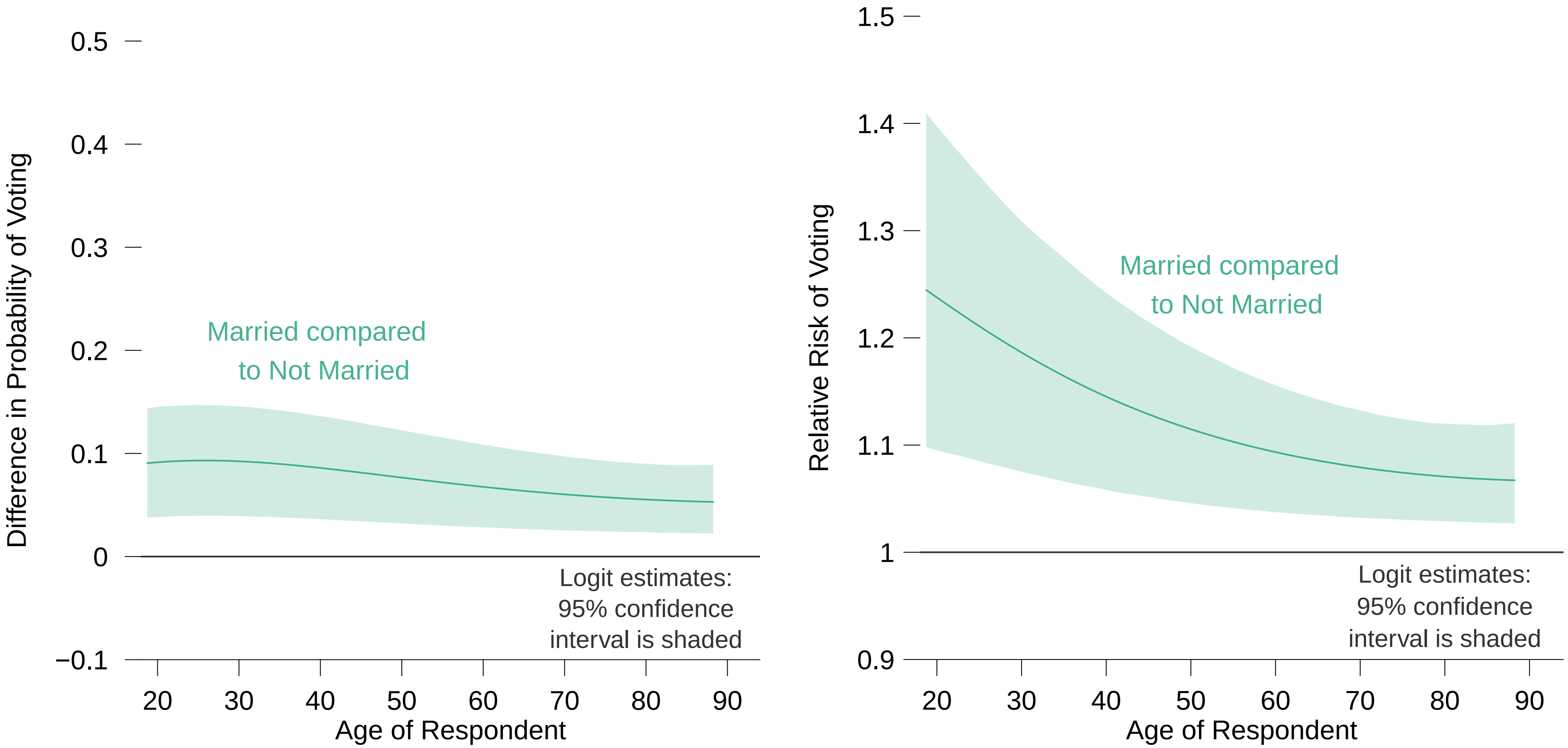
FD estimates the difference only, so it has a tighter CI

Avoid mistakenly rejecting significant first differences!




Consider showing relative risks instead of (or in support of) first diffs

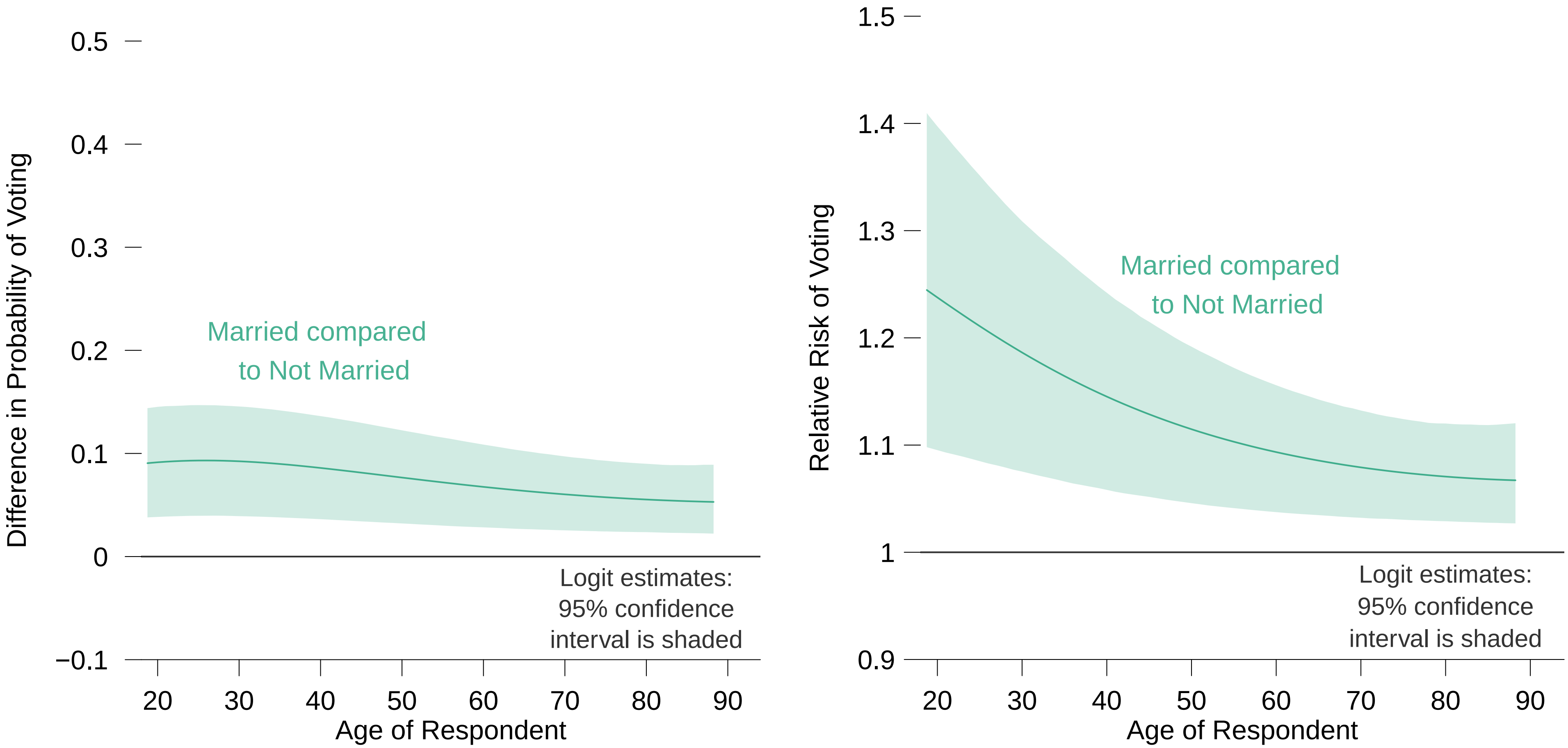
Shows "how many times more likely": often the best way to show magnitude




Setting up counterfactuals for FDs or RRs is tricky (see sample code)

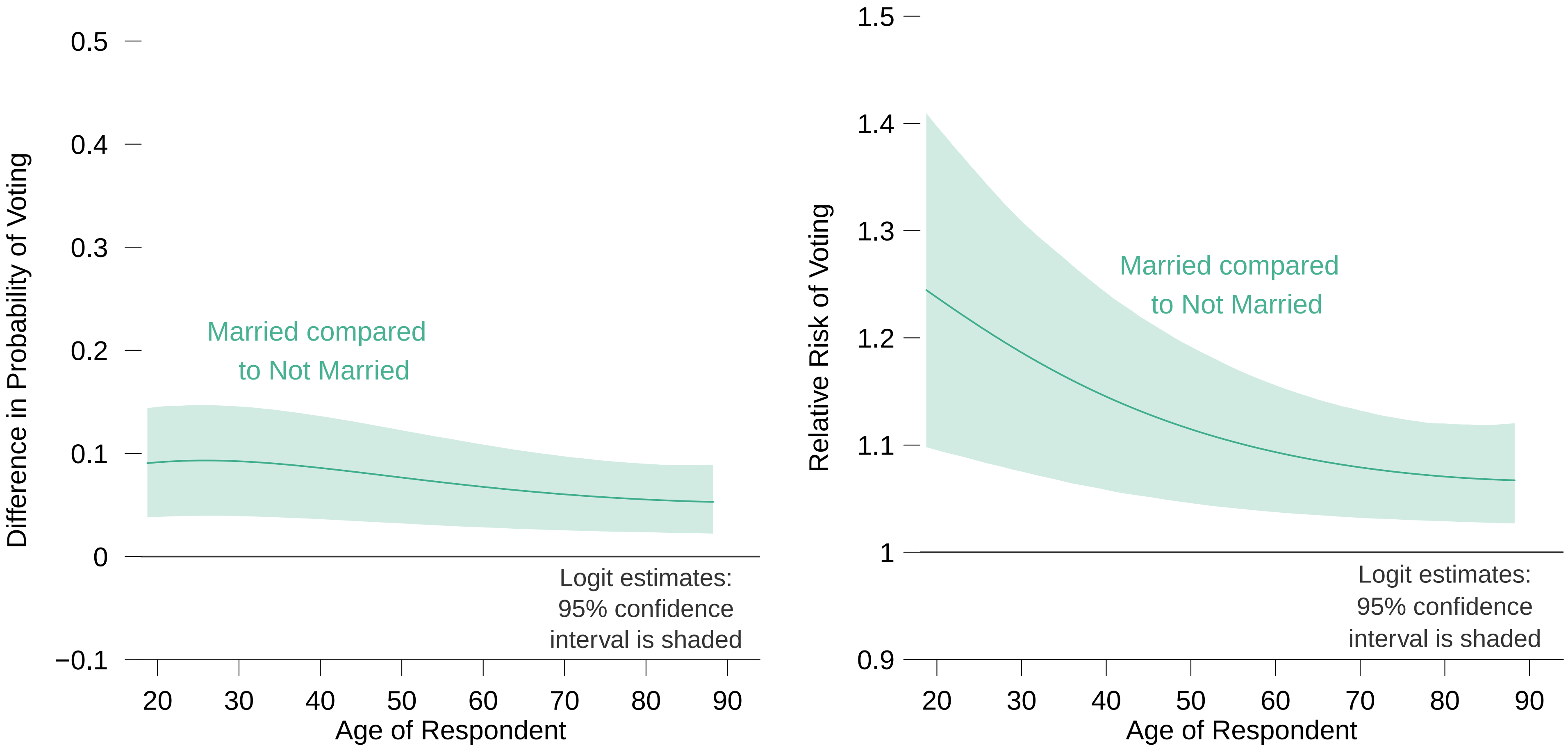
Be very careful to completely lay out before and after scenarios

Here I set before and after *age* to the same value (which varied across the plot)
but I set Married to different values (0 before, 1 after)




Take care in selecting the before and after values of *all* covariates

This is the most common place students make errors,
usually with *huge* substantive consequences




Now we know the *substantive* effect of including Marriage as a covariate

But should we use Model 1 or Model 2? Do we keep Marriage or not?

To decide, we should consider model goodness of fit

# 3 paradigms for goodness of fit

## In-sample fit

*Test the model's performance on the same data used to fit the model*

Easy to do – you already have the data

The default approach unless otherwise stated

Tends towards overconfidence and overfitting:
treats models that predict quirks of the sample as good models of the population

## Out-of-sample fit

*Test the model's performance on a separate sample from the target population*

Expensive, as you need to reserve/find data not used in the main analysis

Requires the analyst to construct a special test; not a default output

Avoids overconfidence: shows how the model *generally* fits the population

## Cross-validation

*Can you have the best of both worlds?*

# Within each paradigm, the same tests can be applied

*Five (mostly) generic ways to check MLE goodness of fit*

1. $t$-test / Wald statistic
2. Likelihood Ratio test
3. Bayesian Information Criterion (BIC)
4. Akaike Information Criterion (AIC)
5. Inspection of "residuals" (GLM only)

*Six additional ways to check GOF for binary outcomes*

6. Percent correctly predicted
7. Separation plots
8. Sensitivity and specificity / ROC plots
9. Concordance index / "Area under the curve" (AUC)
10. Actual *versus* predicted plots
11. Error *versus* predicted plots

# $t$-tests

We start with general test of goodness of fit applied in-sample

A simple test of whether a parameter is significantly different from zero is:

$$p = 1 - F_t\left(\hat{\beta}/\sqrt{\mathrm{Var}(\hat{\beta})}, n-k\right)$$

We calculate the $t$-statistic for Married, and find:

$$\frac{\hat{\beta}}{\sqrt{\mathrm{Var}(\hat{\beta})}} = \frac{0.373}{0.110} = 3.393$$

which has a $p$-value of $0.0007 = \frac{7}{10{,}000}$

*what does this mean?*

A more general form (Wald test) can be applied to multiple parameters at once

# Likelihood ratios

Likelihood ratio test of nested models:

$$\mathrm{LR} = -2 \log \frac{\mathcal{L}(\mathcal{M}_1)}{\mathcal{L}(\mathcal{M}_2)}$$
$$\mathrm{LR} = 2 \left( \log \mathcal{L}(\mathcal{M}_2) - \log \mathcal{L}(\mathcal{M}_1) \right) \sim f_{\chi^2}(m)$$

where $m$ is the number of restrictions placed on model 2 by model 1 (e.g., parameters held constant in model 1)

In our example,
$$\mathrm{LR} = 2 \times (-1099.283 + 1101.370) = 4.174$$

with a $p$-value from the $\chi^2$ distribution of 0.041

LR tests require same data, and nested models [aside: missing data]

# Information Criteria

Penalized LR tests:

Bayesian Information Criterion

$$\begin{aligned}\mathrm{BIC}_{\mathrm{M2}} &= \log n \times p_2 - 2 \log \mathcal{L}(\mathcal{M}_2) \\ \mathrm{BIC}_{\mathrm{M1}} &= \log n \times p_1 - 2 \log \mathcal{L}(\mathcal{M}_1)\end{aligned}$$

where $p$ are the number of parameters and $n$ the number of observations

Lower values of BIC are better – penalty for extra obs & params

In our example, $\mathrm{BIC}_{\mathrm{M2}} - \mathrm{BIC}_{\mathrm{M1}}$ is $3.311664$,
which by Raftery's guidelines indicates "weak" evidence for *M1* over M2

An alternative, Akaike's Information Criterion, agrees:

$$\begin{aligned}\mathrm{AIC}_{\mathrm{M2}} &= 2p_2 - 2 \log \mathcal{L}(\mathcal{M}_2) \\ \mathrm{AIC}_{\mathrm{M1}} &= 2p_1 - 2 \log \mathcal{L}(\mathcal{M}_1)\end{aligned}$$

In our example, $\mathrm{AIC}_{\mathrm{M2}} - \mathrm{AIC}_{\mathrm{M1}} = -2.174388$ ("weak" evidence for *M2*)

# Residual inspection

Residual inspection a mainstay of linear regression analysis

- Identification of outliers, especially those with high leverage
- Robust regression (downweights outliers) / resistant regression (ignores them)

More difficult for binary models – the distribution of residuals is unclear

Bayesian alternatives for outlier detection (optional reading: Albert and Chib)

GLM alternatives available in example code

Less useful for binary data – *what's an outlier when only two outcomes are possible?*

We'll revisit residuals with count data

Now some methods tailored for binary outcomes. . .

# Percent Correctly Predicted

*Percent Correctly Predicted:* The percentage of observations, y, such that

$$\hat{y} \geq .5 \text{ and } y = 1 \qquad \text{or} \qquad \hat{y} < .5 \text{ and } y = 0$$

How do we formalize this?

$$\text{PCP} = \frac{1}{n}\left[\sum_{\forall y_i = 1} \mathbb{I}(\hat{y}_i \geq 0.5) + \sum_{\forall y_i = 0} \mathbb{I}(\hat{y}_i < 0.5)\right]$$

# Percent Correctly Predicted

*Percent Correctly Predicted:* The percentage of observations, y, such that

$$\hat{y} \geq .5 \text{ and } y = 1 \qquad \text{or} \qquad \hat{y} < .5 \text{ and } y = 0$$

An alternative way to write this may be easier to turn into code:

$$\text{PCP} = \frac{1}{n}\left[\sum_{\forall i} y_i \times \mathbb{I}(\hat{y}_i > 0.5) + (1 - y_i) \times \mathbb{I}(\hat{y}_i < 0.5)\right]$$

For the model with "married", $\text{PCP}_{\text{m2}} = 69.770\%$

For the model without "married", $\text{PCP}_{\text{m1}} = 69.994\%$

Note a PCP less than 50% would be horrid (why?)

$\bar{y} = 0.65675$, so these PCPs are less impressive than you might think (Why?)

If you report PCP, be sure to also say what $\bar{y}$ is:
$\text{PCP} - \bar{y}$ is how much improvement in prediction comes from your covariates

# Percent Correctly Predicted

PCP is easy to calculate but flawed

PCP treats these cases *identically*:

$\hat{y} > .51$ and $y = 1$ (Model is hedging its bets)

$\hat{y} > .99$ and $y = 1$ (Model is going "all in")

. . . but treats these cases *differently*:

$\hat{y} > .51$ and $y = 1$ (Model is hedging its bets)

$\hat{y} > .49$ and $y = 1$ (Model is hedging its bets)

PCP's sharp emphasis on probability=0.5 is a severe limitation

Can we avoid this misplaced emphasis?

Let's try some graphical methods

# Separation Plots

Greenhill, Ward, and Sacks suggest a simple graphical approach to binary GOF

Sort the observations by the predicted probability of the outcome;
plot 1s as red lines, and 0s as tan lines

A model that correctly predicts observations should separate red and tan lines

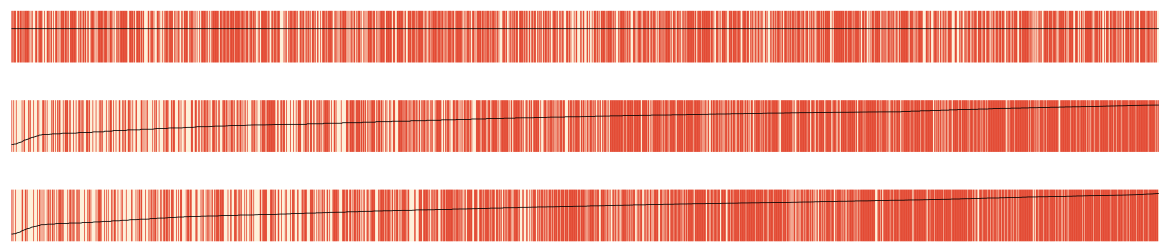
Red lines lying to the left and tan lines lying to the right are mispredictions

A horizontal black line traces out the predicted probability for each case

Null Model (No covariates)

Model 1 (Age, Edu)

Model 2 (Age, Edu, Married)



Helps when comparing lots of models (compact, stackable plots)

Could even use the R `locator()` function to identify the most surprising cases
and the models that get them right

# Perfect separation

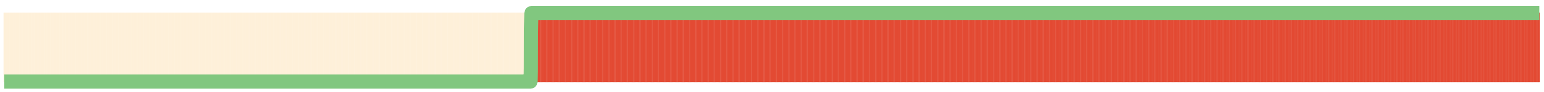
Suppose the separation plot looked like this:

Things to think about:

1. What kind of model would produce perfect separation?

# Perfect separation

Suppose the separation plot looked like this:



Things to think about:

1. What kind of model would produce perfect separation?

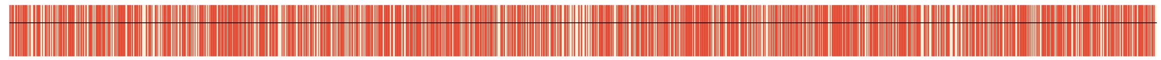
2. I've added a line tracing the probability of each case.
What is the slope of this line near the separation between tan and red?

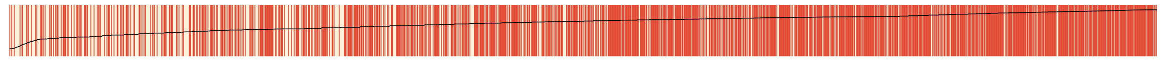
3. If there is a binary covariate $z$ that is 0 for every $y = 0$ and 1 for every $y = 1$,
what is the logit coefficient for $z$?

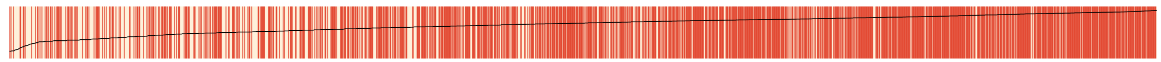
4. Could a logit model fit "too well"?

# Separation Plots

Let's return to the data & models for turnout:

Null Model (No covariates) 

Model 1 (Age, Edu) 

Model 2 (Age, Edu, Married) 

We can visualize percent correctly predicted using these plots:

- Find the spot where the probability line crosses 50%
- Red observations to the *left* of this point are *incorrectly* classified
- Red observations to the *right* of this point are *correctly* classified

We noted $\pi^* = 50\%$ was an arbitrary threshold

Can we conceptualizehow different thresholds $\pi^*$ would classify the data?

# Sensitivity and Specificity

Let $\pi^*$ be a threshold at some level in [0,1], above which we suppose $\hat{y} = 1$

Given a threshold $\pi^*$, we can compute the model's sensitivity and specificity:

**Sensitivity** is the fraction of true positives the model tends to identify, or

$$\text{True Positive Rate} \qquad \Pr(\hat{y} \geq \pi^* | y = 1)$$

**Specificity** is the fraction of true negatives the model tends to identify, or

$$\text{True Negative Rate} \qquad \Pr(\hat{y} < \pi^* | y = 0)$$

While we'd like high sensitivity *and* specificity, they trade off as we shift $\pi^*$ (Why?)

A sensitive model is better when you especially fear false negatives
(detecting a virulent infectious disease, like the Ebola virus)

A specific model is better when false positives are too costly to wade through
(if treating a disease has big side-effects, like many cancers)

# Sensitivity and Specificity

We can compute in-sample specificity & sensitivity for any $\pi^*$ & model

| $\pi^* = 0.50$ | Sensitivity (TPR) | Specificity (TNR) |
|---|---|---|
| Model 0: Null | 100.0% | 0.0% |
| Model 1: Age + Edu | 89.1% | 33.5% |
| Model 2: Age + Edu + Married | 88.2% | 34.3% |

*What do these mean? And what do they suggest about model selection?*

$\pi^*$ is arbitrary, so why restrict our attention to one threshold?

In our case, better performance occurs at a higher threshold like $\pi^* = 0.65$

| $\pi^* = 0.65$ | Sensitivity (TPR) | Specificity (TNR) |
|---|---|---|
| Model 0: Null | 100.0% | 0.0% |
| Model 1: Age + Edu | 68.2% | 64.2% |
| Model 2: Age + Edu + Married | 68.2% | 64.2% |

*What's changed and why? What else would change if we set $\pi^* = 0.66$?*

# Receiver Operating Characteristic Plots

Unless we have a $\pi^*$ of particular interest,
we're more interested in the range of sensitivity and specificity

We'd also like to understand the tradeoff better,
and see the threshold that maximizes their sum

Receiver operator characteristic plots developed in WWII to evaluate radar operators

ROC plots show sensitivity & specificity under all thresholds from 0 to 1

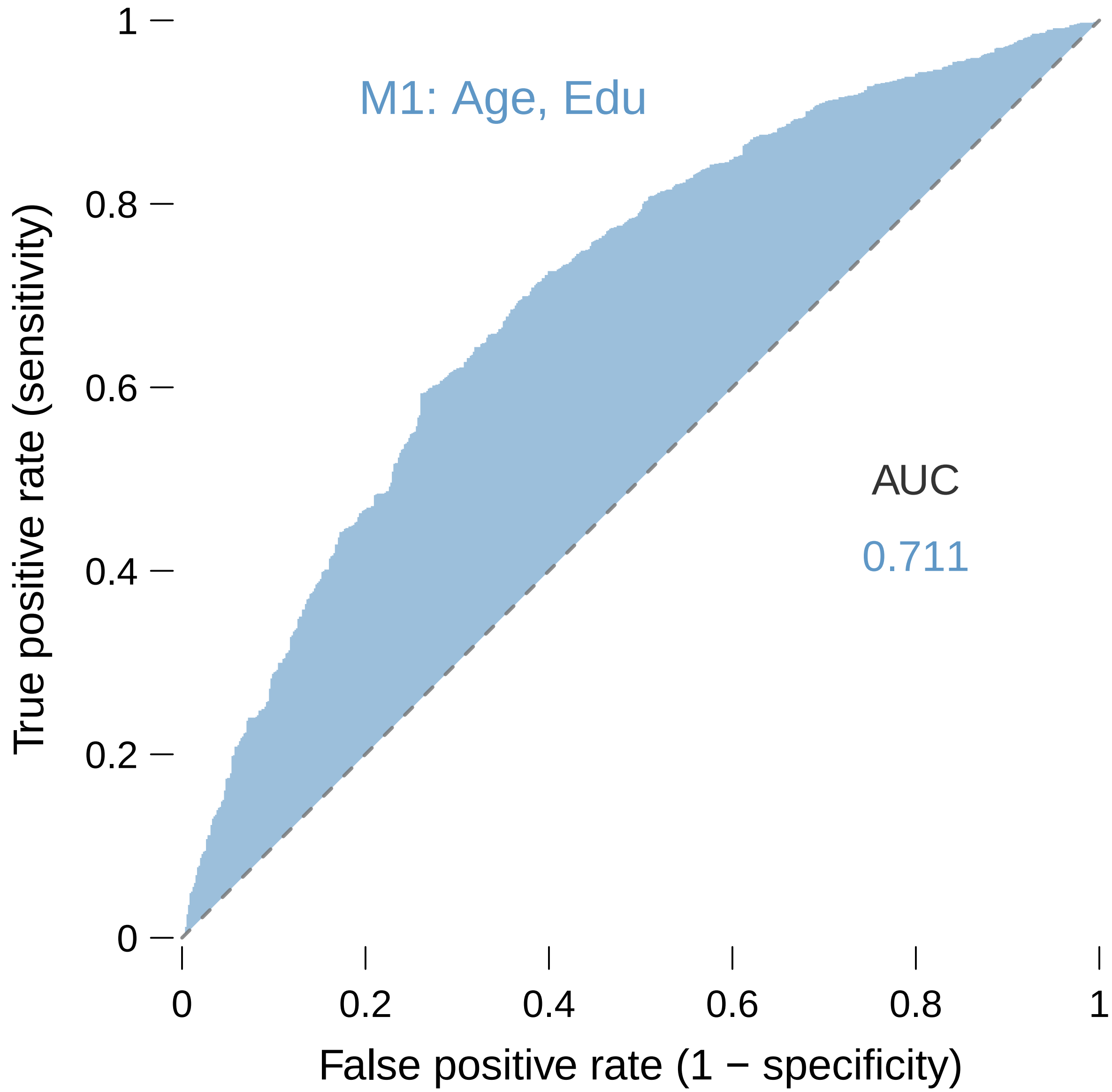
To produce a ROC curve:

1. sort cases from greatest probability of a 1 to least probability

2. start a line at the origin of the graph and work over the cases:

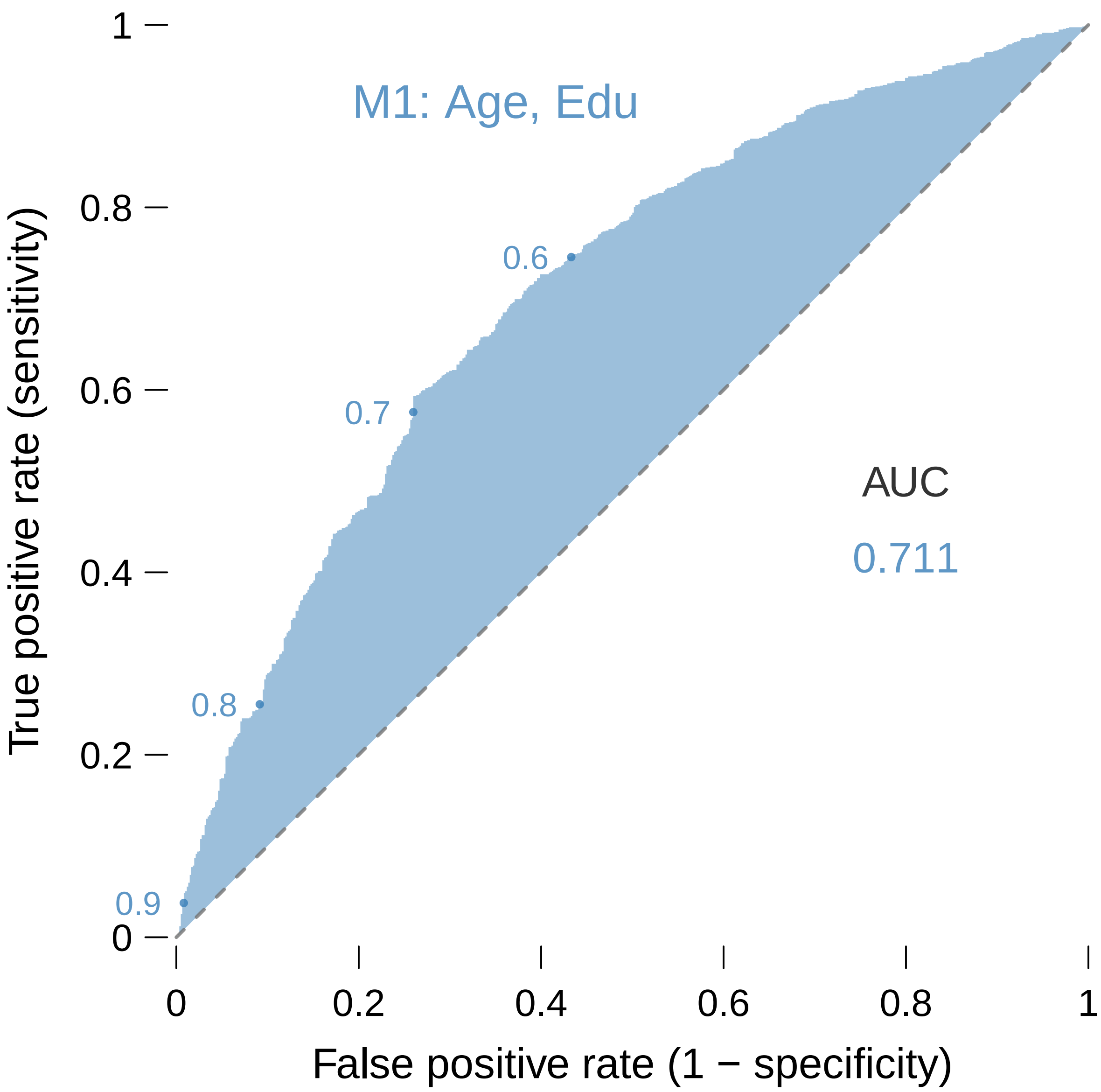
    - move up $\dfrac{1}{\text{total positives}}$ for each positive case

    - move right $\dfrac{1}{\text{total negatives}}$ for each negative case



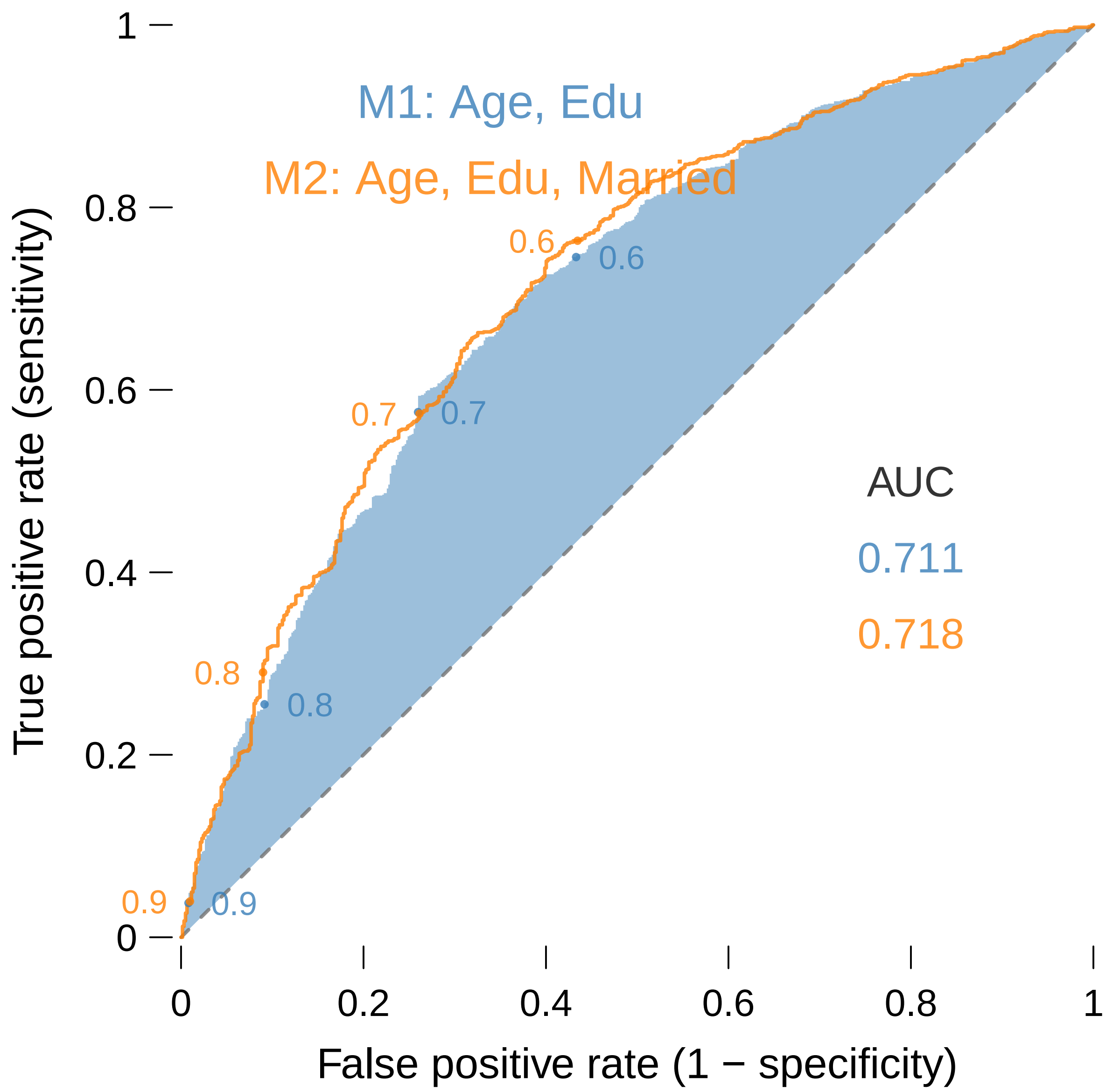

Above is the ROC plot for Model 1

Note we can read the horizontal axis two ways




ROC plots are easier to interpret if we plot a few example thresholds

E.g., if we require $\pi^* = 0.8$ to predict a 1, sens$\approx 26\%$ & spec$\approx 91\%$




Overlapping ROC plots helps us see which models are better at sensing 1s and which are better at sensing 0s (and at what cost in missed 0s and 1s)

# AUC / Concordance Index

The area under the ROC is one measure of GOF (concordance index or AUC)

Models that increase the concordance index are better –
even if the model doesn't change PCP (PCP is hard to move for rare events)

Concordance index is my favorite "one number summary" for logit & probit

(Aside for economists: AUC of ROC is equivalent to Gini coefficient)

Criticisms of AUC: still arbitrary, with equal weight on each threshold and different penalties for different models (see Hand 2009)

Even with AUC, the shape of the ROC is still useful to study:
impossible to fully summarize performance in a single number

# Actual versus Predicted Plots

Plotting $y$ versus $\hat{y}$ is very informative in linear regression

Ideally, all points should be close to the $45^\circ$ line

Some points will be poorly predicted because of noise ($\sigma^2$),
but clusters of deviations from the line suggest potential omitted variables

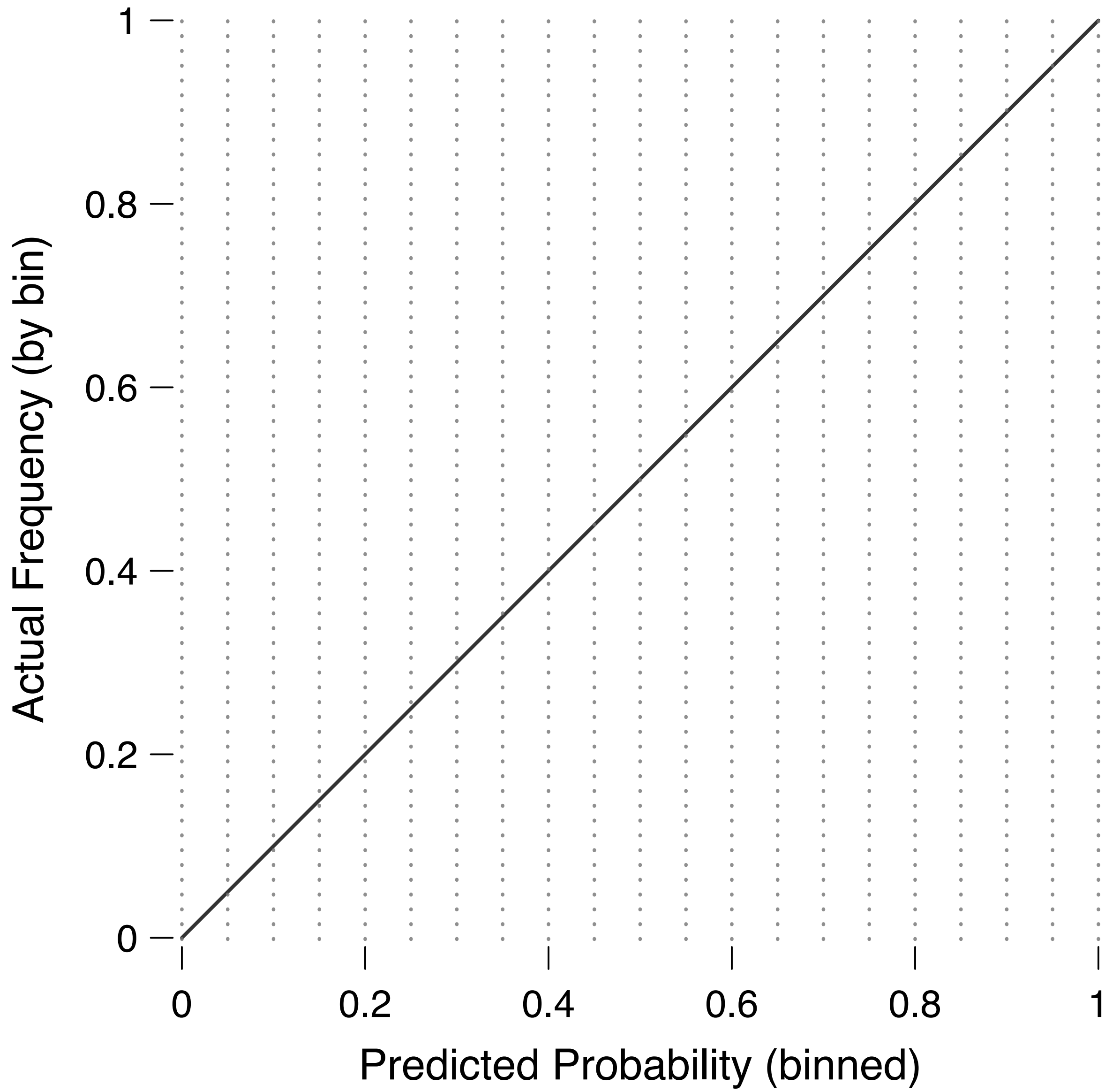
Systematic nonlinearity suggests incorrect functional form

$y$ versus $\hat{y}$ plot for binary data would be as useful

But binary data don't plot well

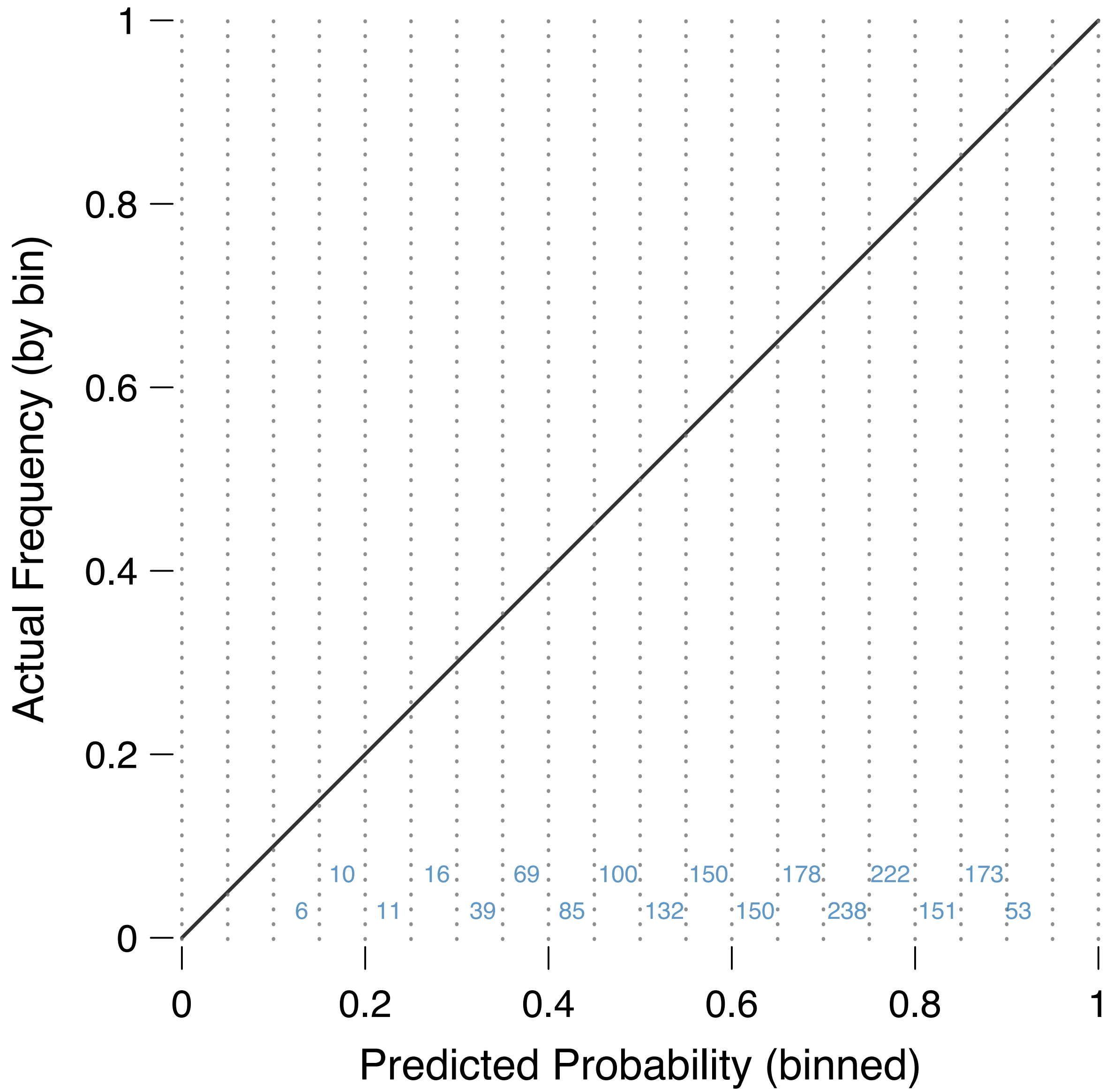
Sorting the data by predicted probability has helped so far

*Solution: bin* the sorted data into ranges of predicted probability



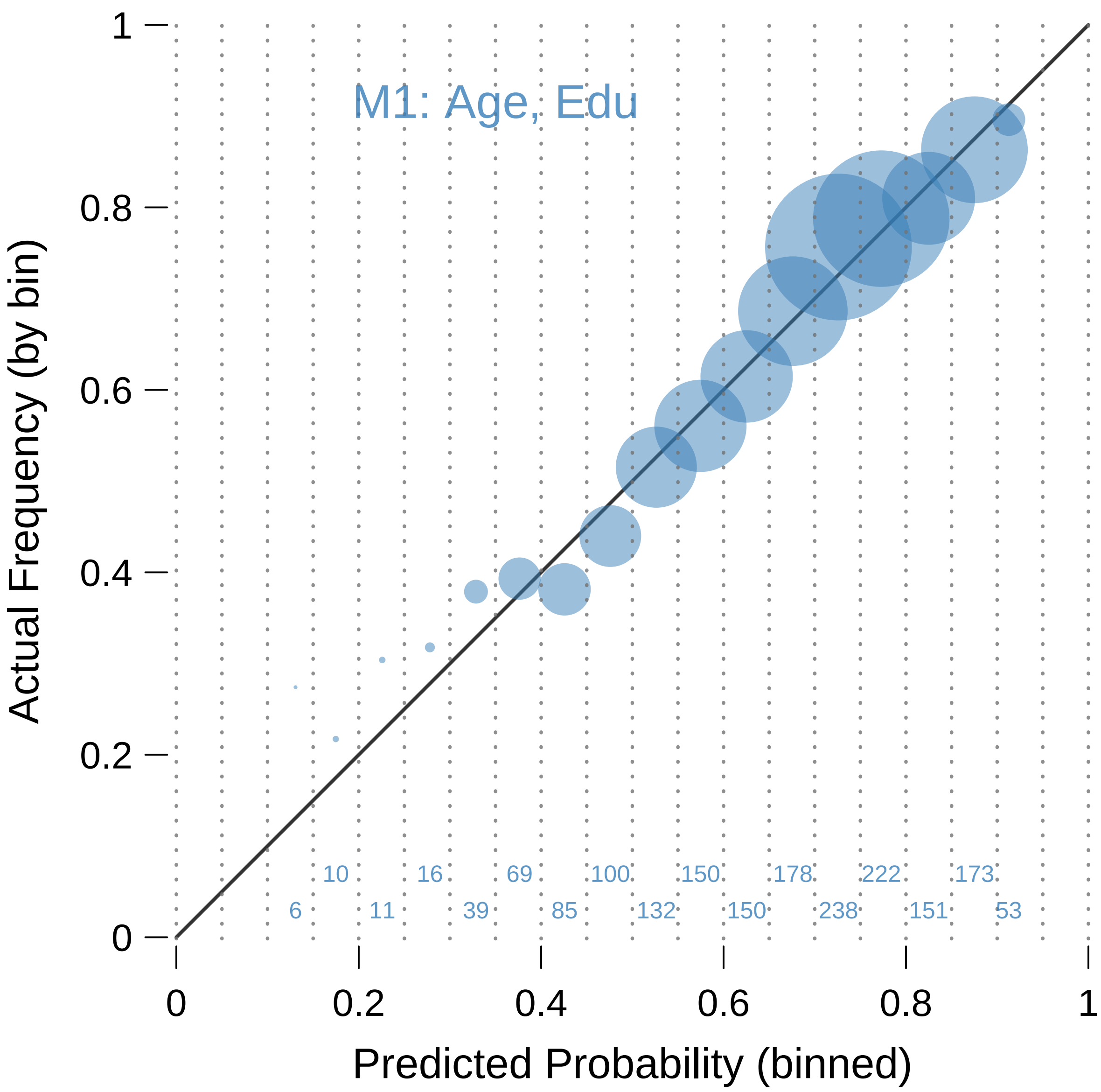

Consider 20 bins, each containing cases with predicted probs in a common interval

1st bin has cases with $\Pr(\text{Vote}) \in [0\%, 5\%)$; the 2nd has $\Pr(\text{Vote}) \in [5\%, 10\%)$; etc.



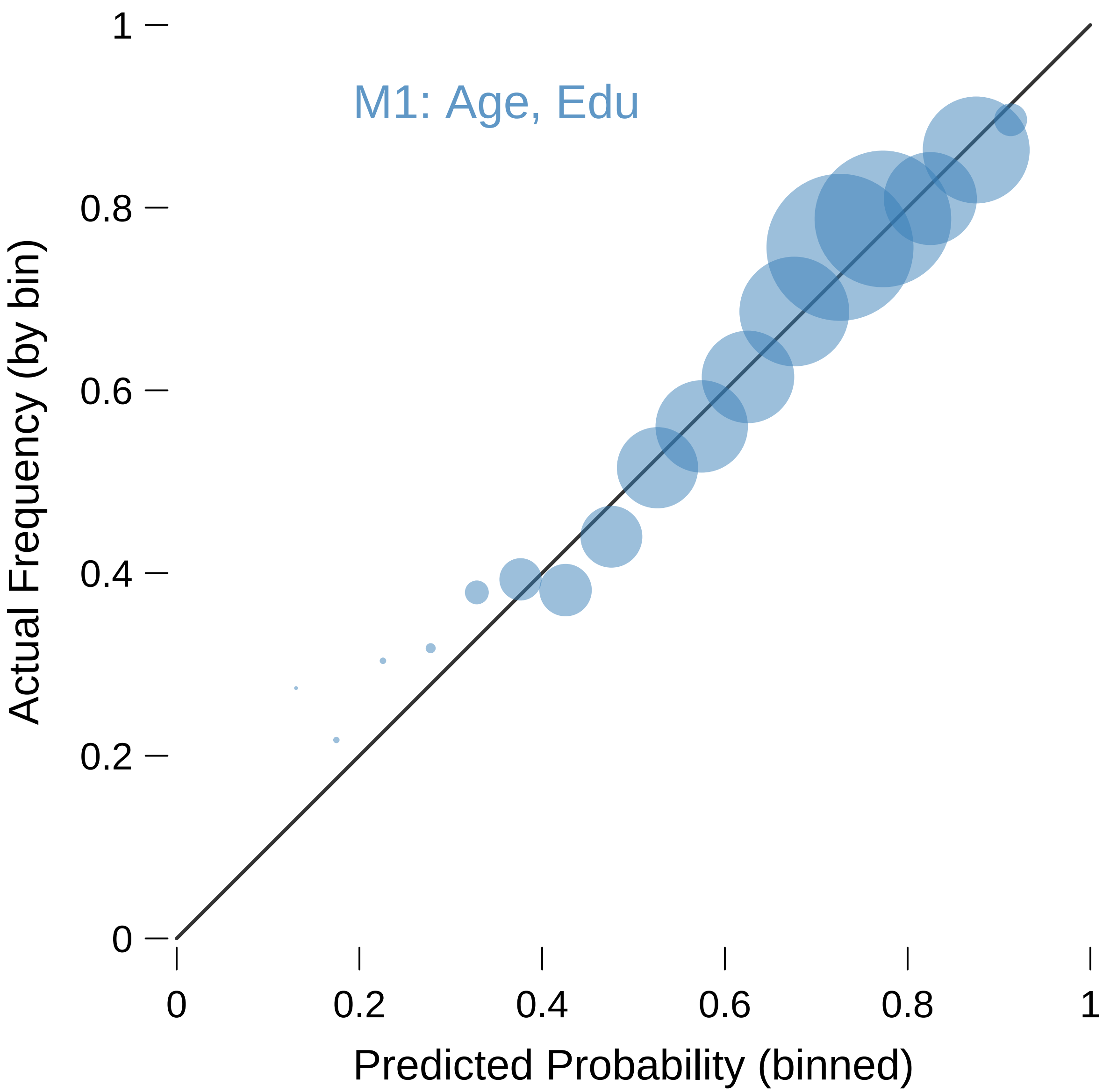

Some bins have more cases than others

Let's ignore bins with fewer than 5 cases (why?)



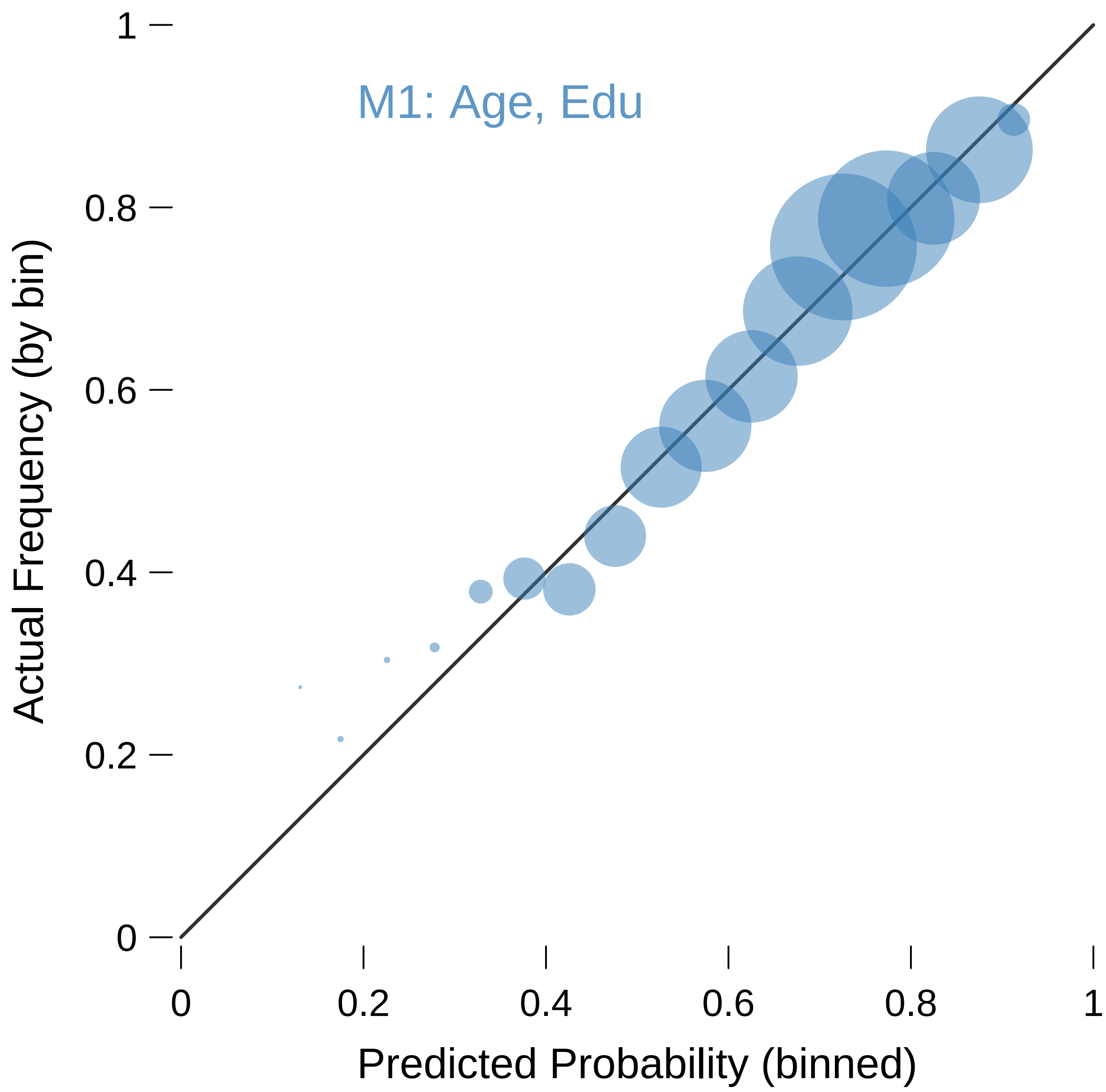

For each bin, compute avg predicted probability and avg empirical frequency

Scale *area* of circles to number of cases to highlight reliable results



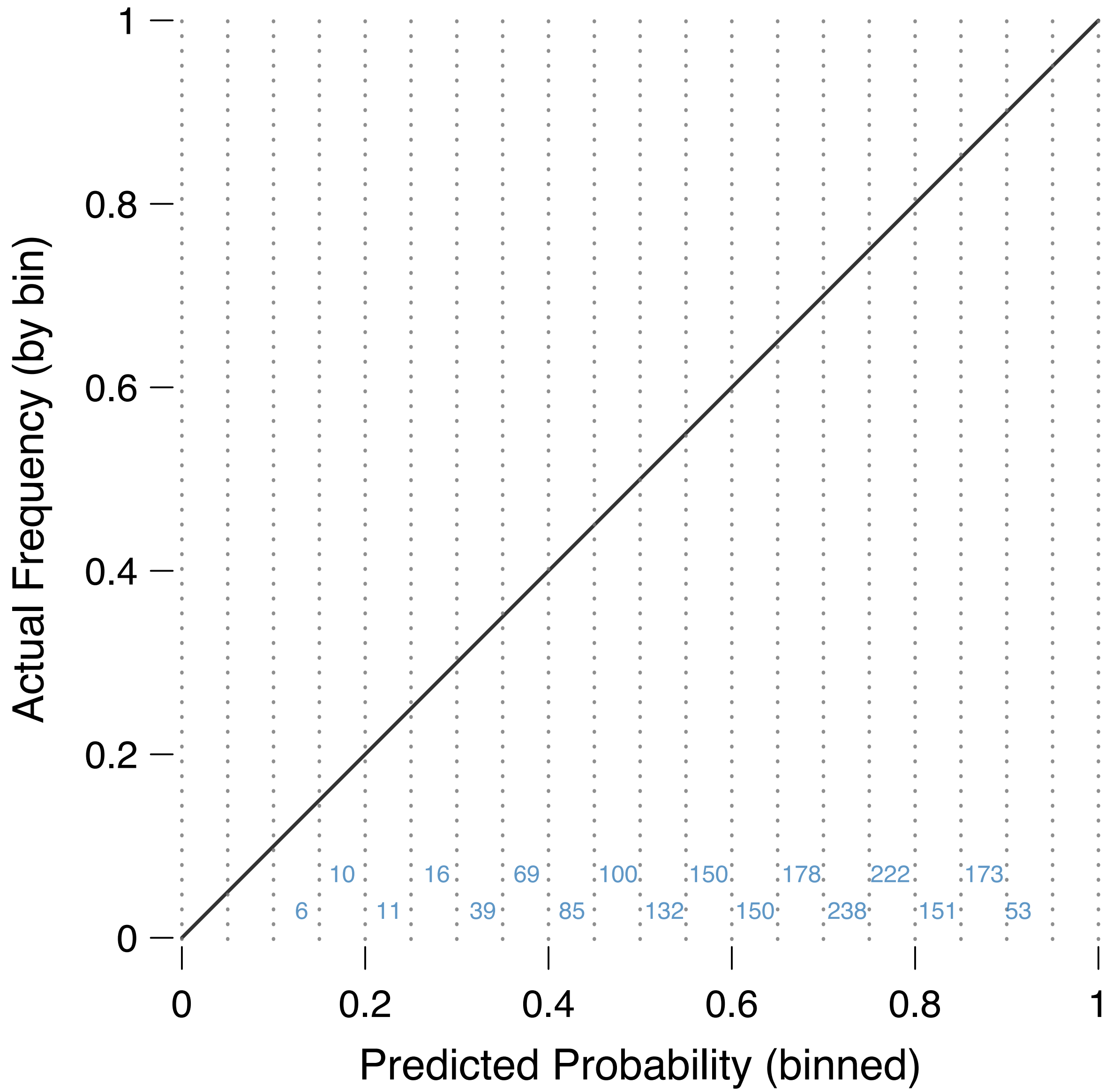

Not bad fit – close to 45° line

A better model would also have most/largest circles near [0,0] or [1,1] (why?)

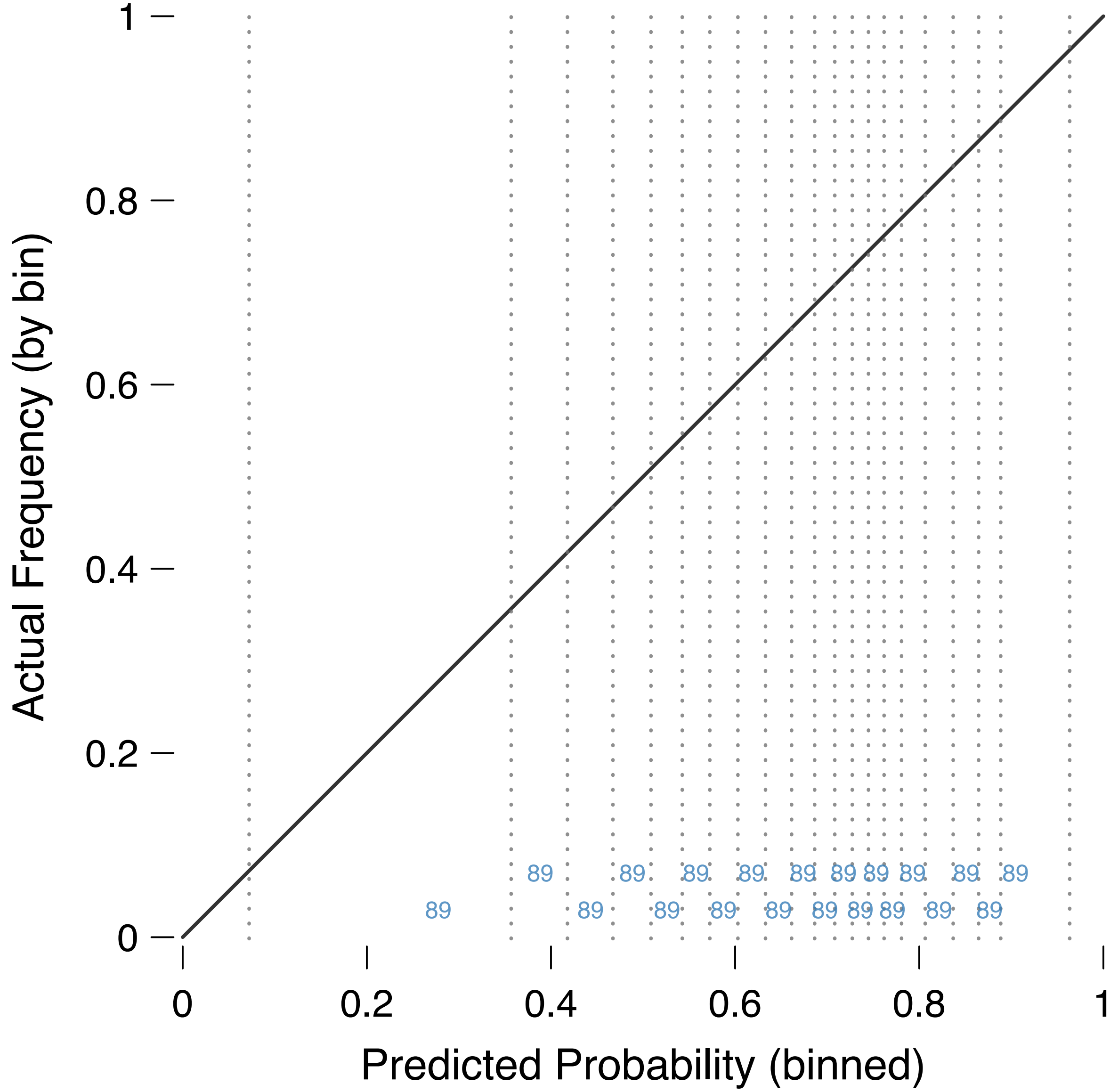



Worse at predicting low probability cases?

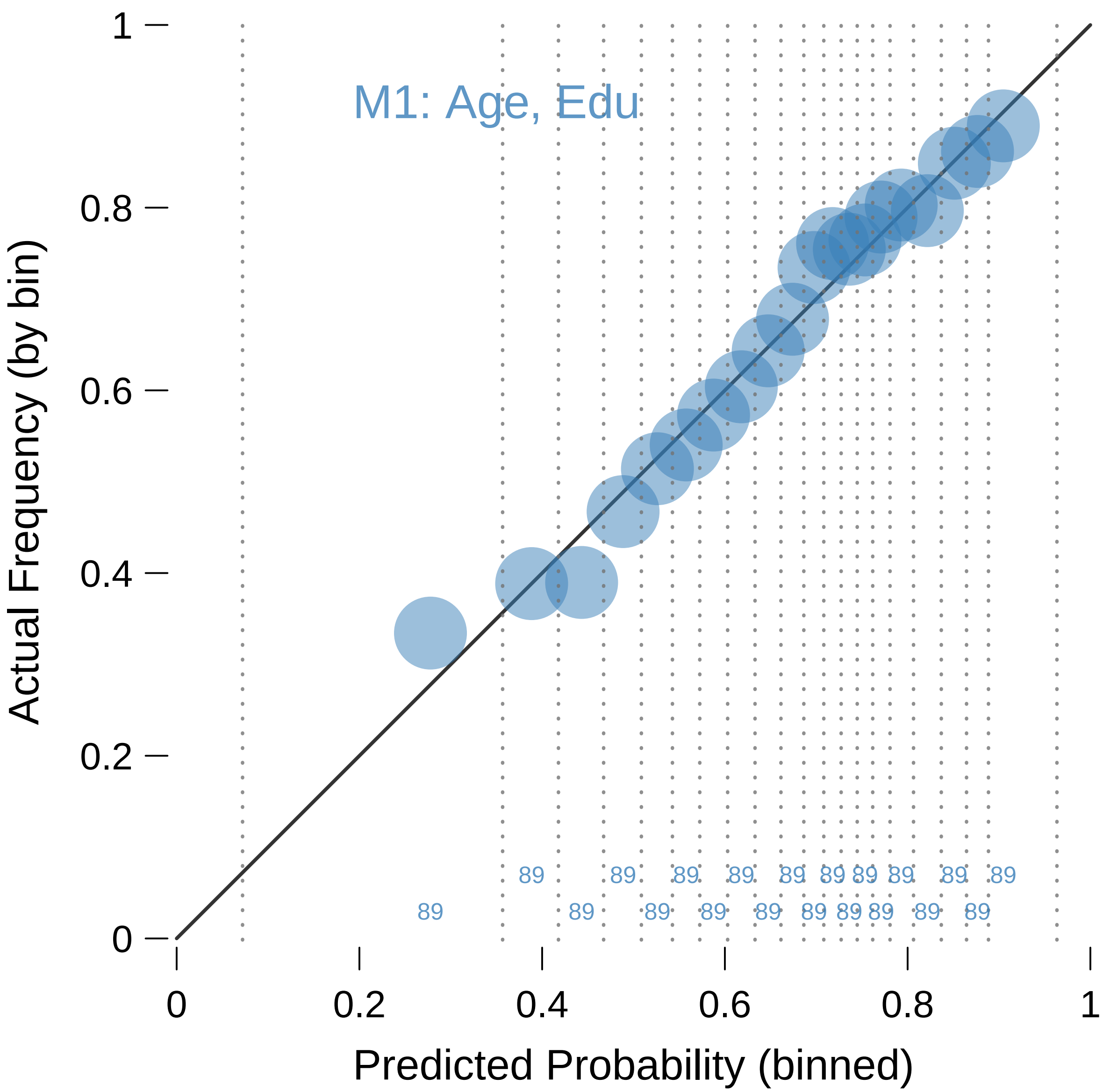
Maybe, but it's hard to tell with few cases per bin in those ranges



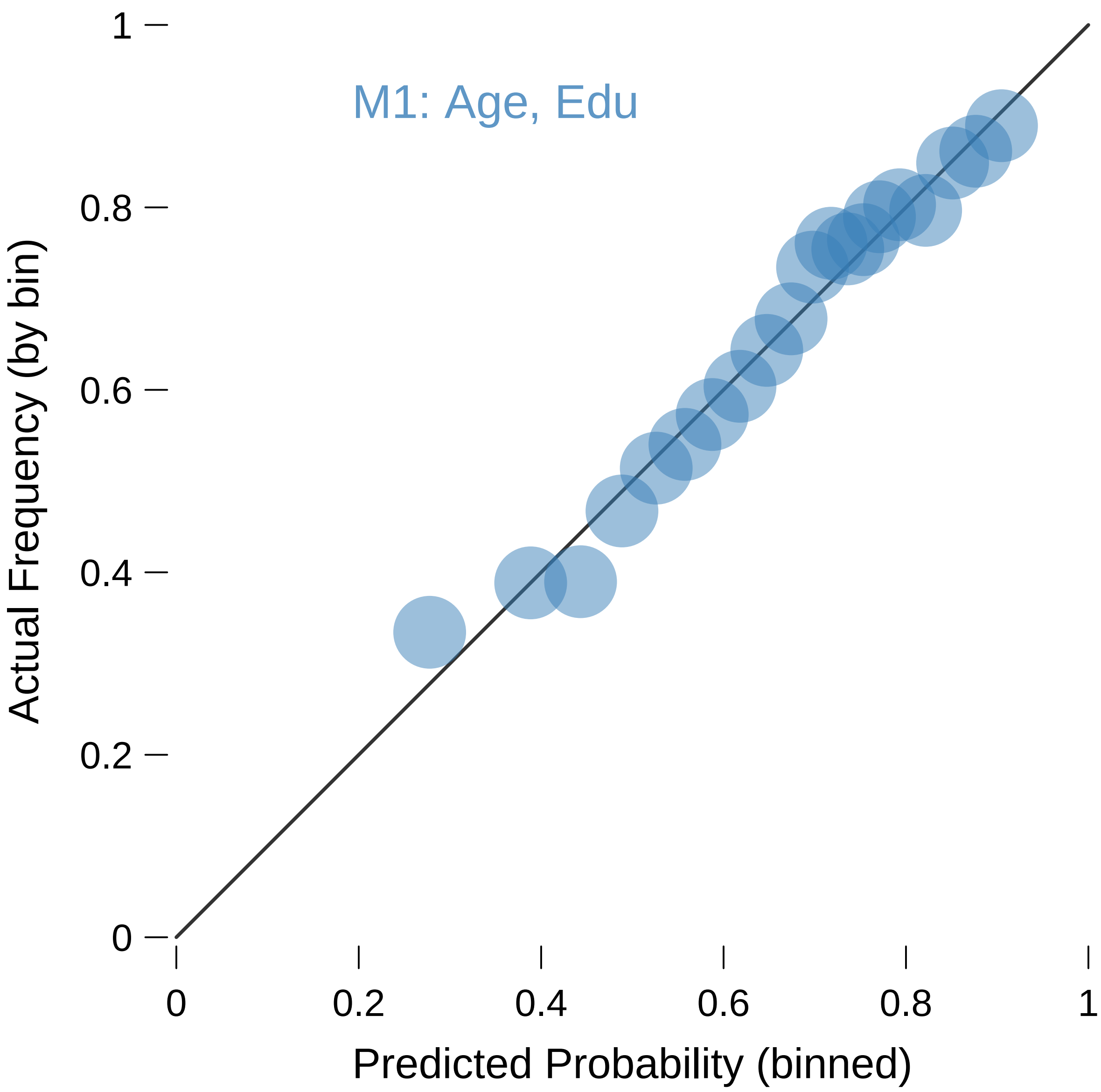

Unless you have tons of data, equally spaced bins are suboptimal




Better to have variable-width bins with a reasonable number of cases

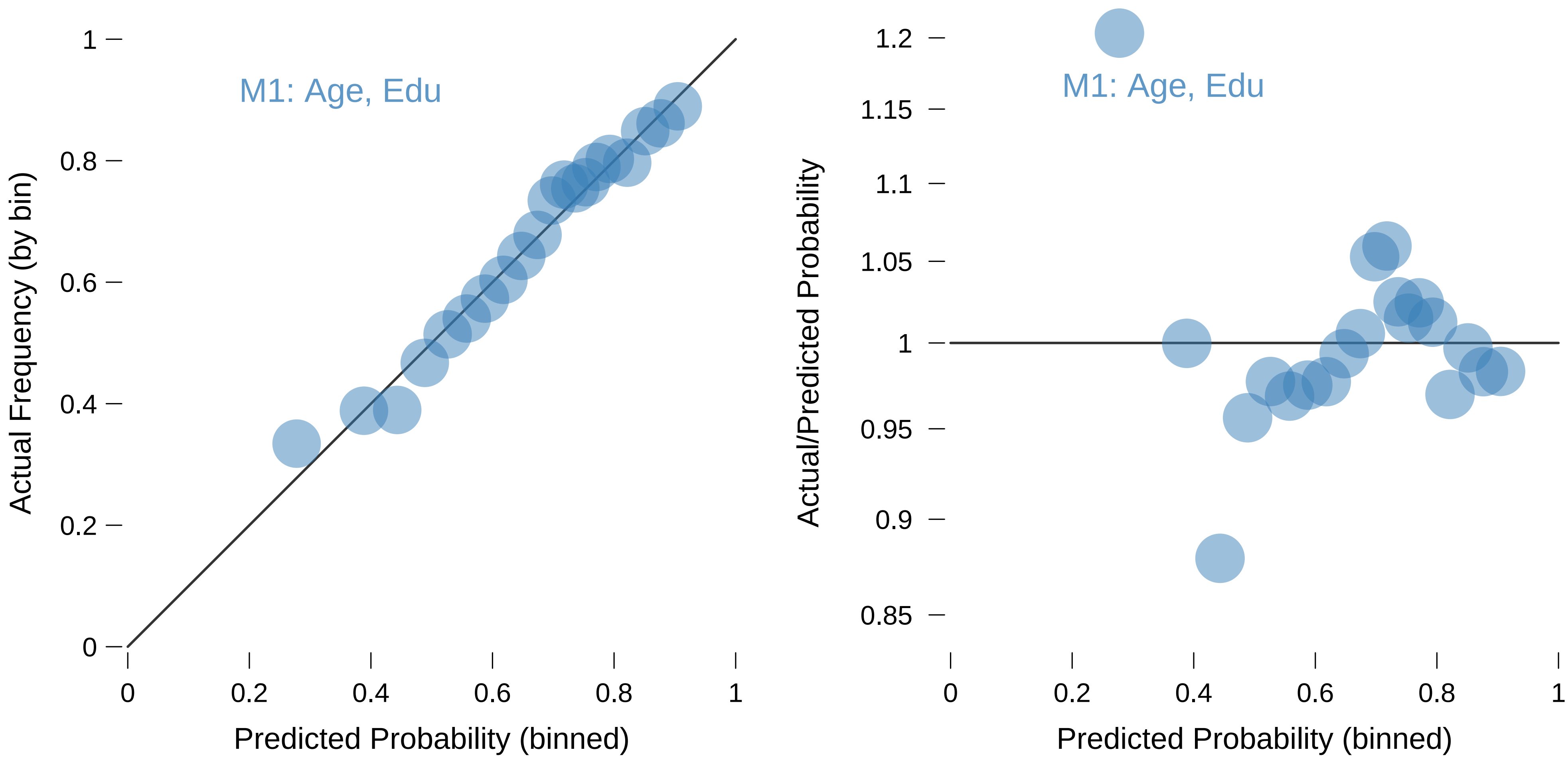



It does look like the model is a bit worse at predicting the behavior of unlikely voters




One problem with actual vs predicted plots: they show absolute error

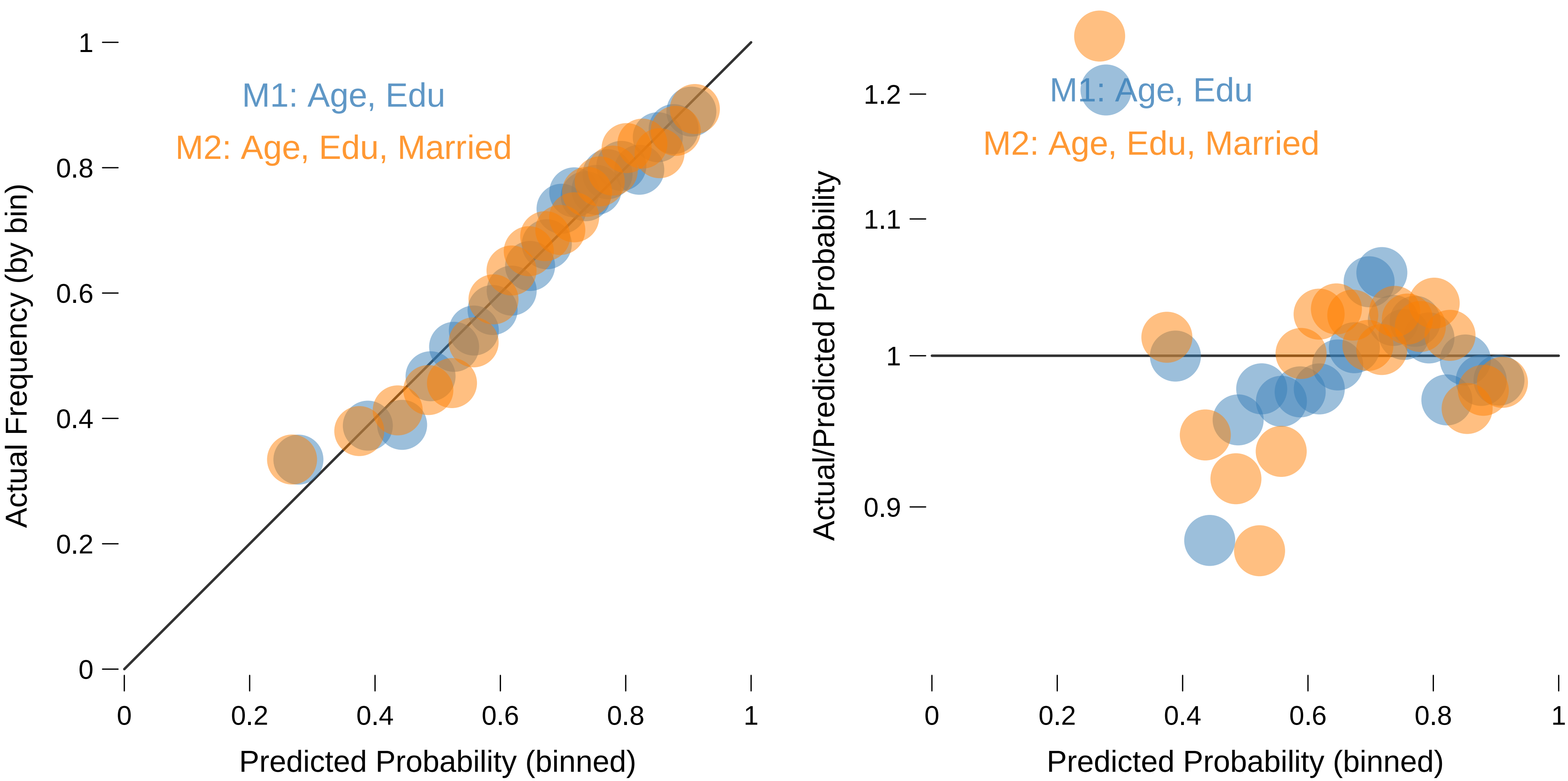
But we're often more interested in relative risk (e.g. for rare events)




The plot at right shows the ratio of actual to predicted probability

"Scales up" errors for low prob events, just as relative risks do compared to first diffs

Relatively speaking, the errors in the lowest bin are larger: we miss by 20%




Model comparison is easy: just overlap AvP and EvP plots

These models are similar, with generally good performance

In other cases, careful study of deviations can suggest model improvements

# Out-of-sample tests

Sometimes we're only (or mostly) interested in inference within a sample

*Sample is a census; e.g., how legislators voted on a specific bill*

If so, in-sample fit is directly relevant – but still risks overfitting!

Usually, we're interested in the sample as a means to infer population parameters

*Presidential Turnout: just one sample of voters from one election*

If so, in-sample fit is both overconfident and indirect:
how would we fit other samples (or the average sample) from the population?

Solution: use the in-sample parameter estimates to predict a new test sample

Expensive: need to gather out-of-sample (OOS) data ($)
or hold out part of the original sample (so se's will be larger)

In 2000 ANES, voters told us whether they voted in 1996 – use as an OOS test?

| | 1996 Data | 2000 Data |
|---|---|---|
| Age | 0.156 | 0.061 |
| | (0.018) | (0.017) |
| Age$^2$ | $-0.00106$ | $-0.000318$ |
| | (0.000192) | (0.000170) |
| High School Grad | 1.415 | 1.099 |
| | (0.199) | (0.181) |
| College Grad | 1.199 | 1.053 |
| | (0.150) | (0.132) |
| Married | 0.375 | 0.373 |
| | (0.122) | (0.110) |
| Constant | $-5.184$ | $-2.866$ |
| | (0.422) | (0.422) |
| log likelihood | $-863.1$ | $-1028.7$ |
| AIC | 1738.2 | 2069.5 |
| BIC | 1771.1 | 2102.4 |
| $N$ | 1783 | 1783 |

Above are the estimates for each year run *separately*

Good robustness check: are parameters (or better still, the QoIs) similar?

| | 1996 Data | 2000 Data |
|---|---|---|
| Age | 0.156 | 0.061 |
| | (0.018) | (0.017) |
| Age$^2$ | $-0.00106$ | $-0.000318$ |
| | (0.000192) | (0.000170) |
| High School Grad | 1.415 | 1.099 |
| | (0.199) | (0.181) |
| College Grad | 1.199 | 1.053 |
| | (0.150) | (0.132) |
| Married | 0.375 | 0.373 |
| | (0.122) | (0.110) |
| Constant | $-5.184$ | $-2.866$ |
| | (0.422) | (0.422) |
| log likelihood | $-863.1$ | $-1028.7$ |
| AIC | 1738.2 | 2069.5 |
| BIC | 1771.1 | 2102.4 |
| $N$ | 1783 | 1783 |

Perhaps a more monotononic relationship between turnout & age in 1996 (why?)

Next step is an OOS test: use the 2000 parameters to predict the 1996 outcome

# Out-of-sample tests

We can use any goodness of fit measure we like in our OOS test

Predict 1996 data with 2000 parameters, compare to in-sample 1996 fit using:

*separation plots* *PCP* *ROC* *AUC* *AvP plots*

If the outcome were more continuous, you could also compare:

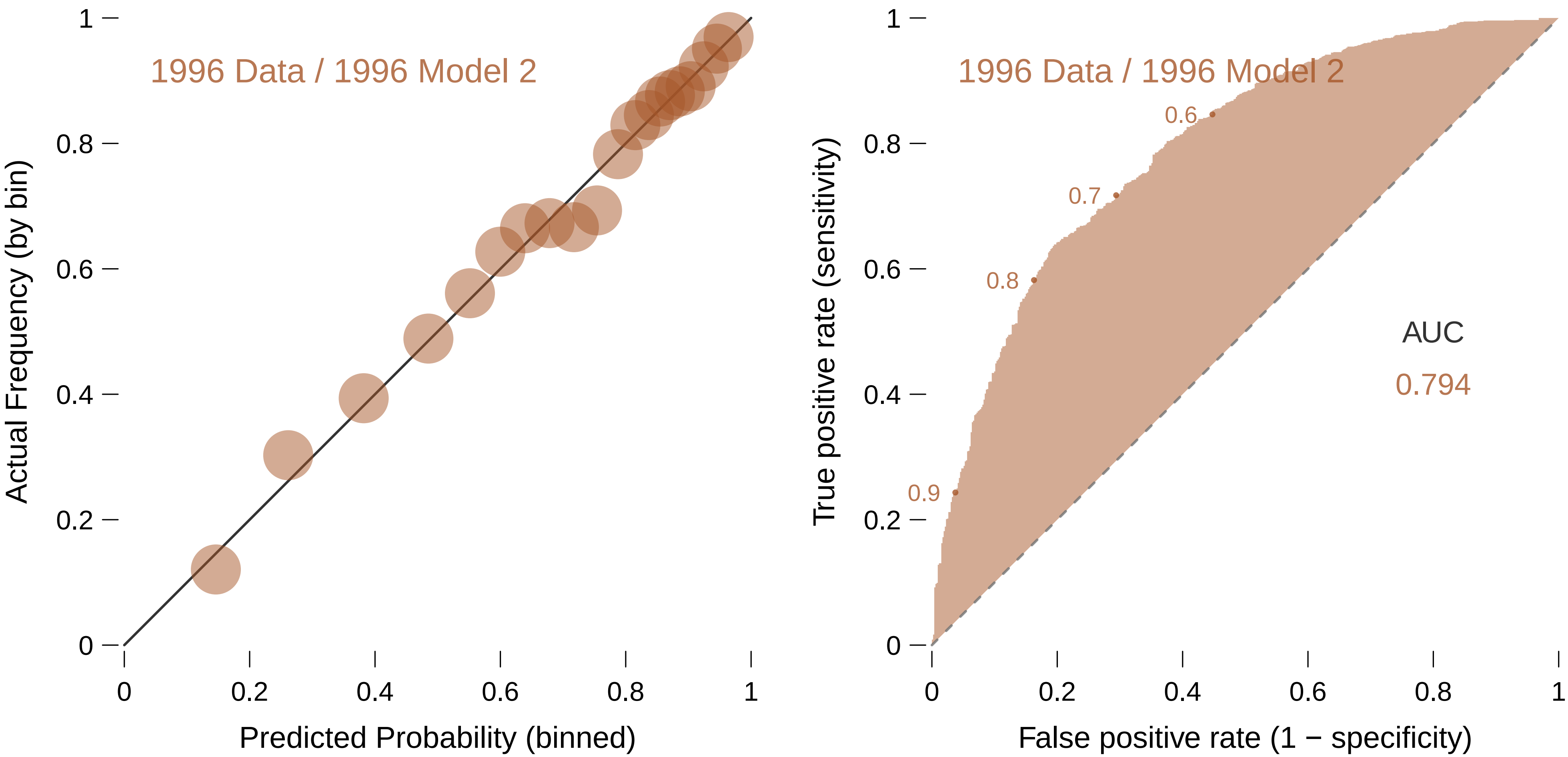
*mean absolute error (MAE)* *RMSE* *coverage of prediction intervals*

"Does the model fit out-of-sample nearly as well as in-sample?"

*If not, maybe there is overfitting; consider a simpler model*

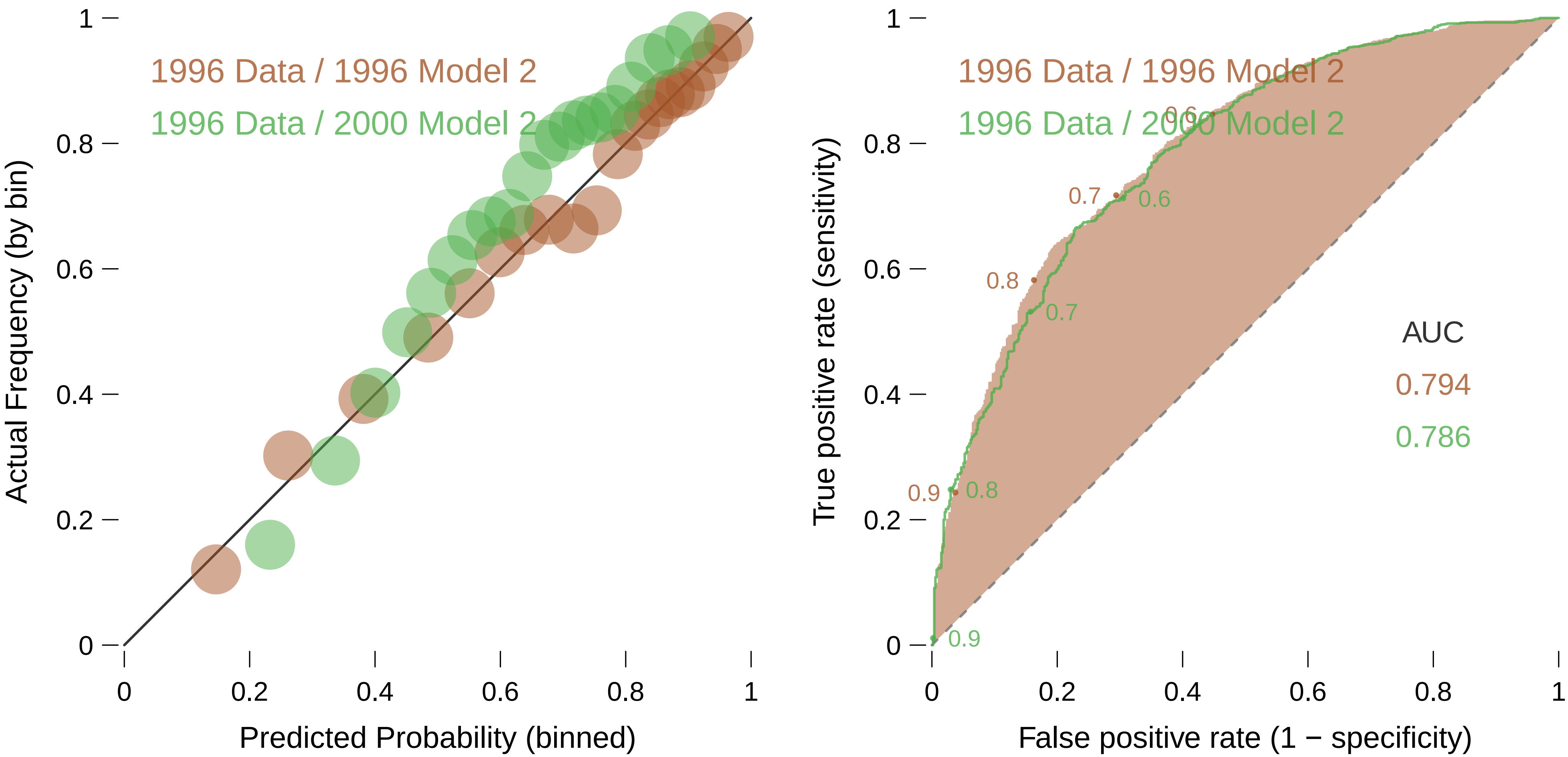
"Is the out-of-sample fit adequate?"

*This is a strong test of goodness of fit, unlike in-sample tests*



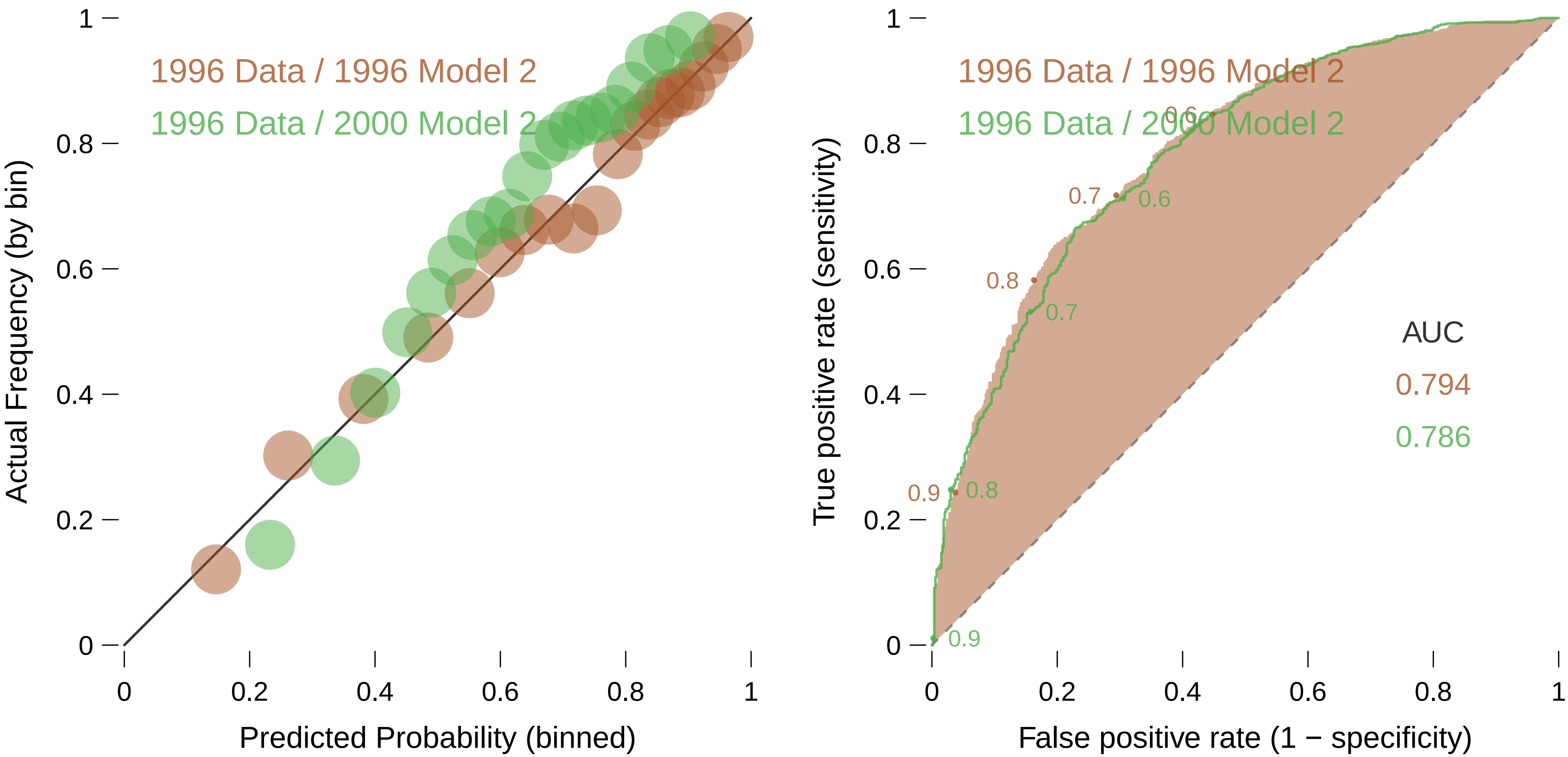

Model 2's in-sample fit to the 1996 turnout data according to AvP, ROC & AUC

How closely can we approximate this using the 2000 model?



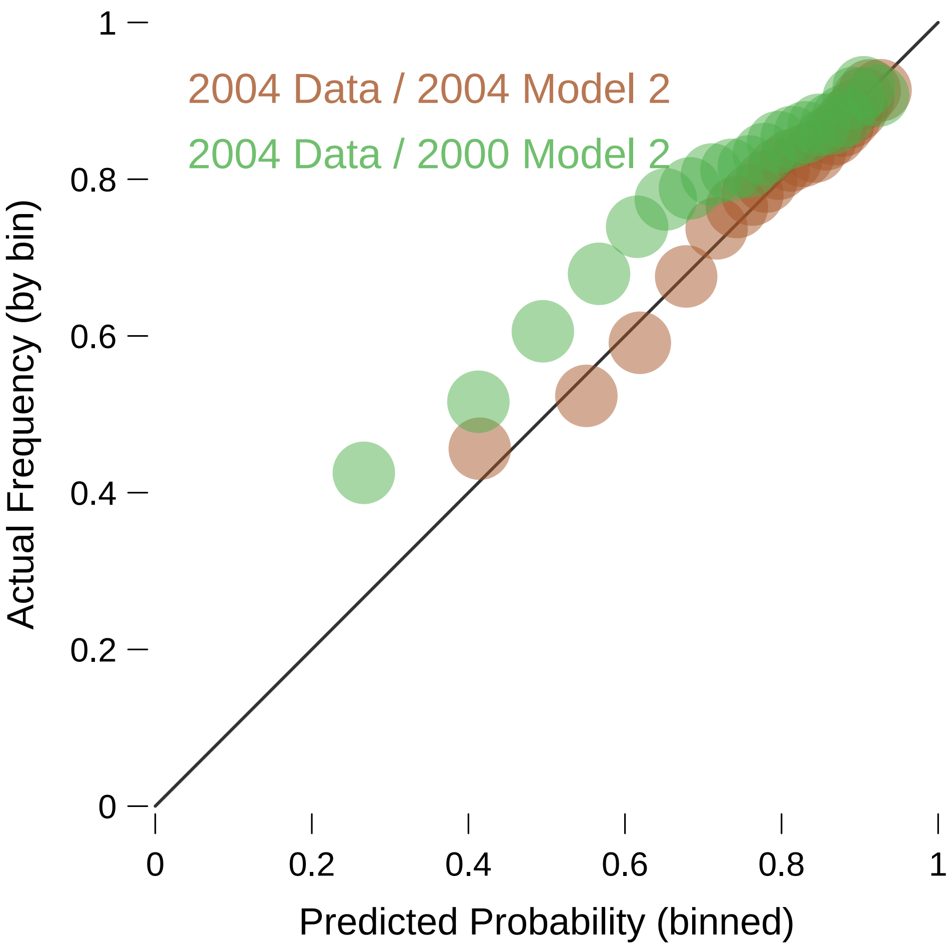

The out-of-sample test of the 2000 model to the 1996 data

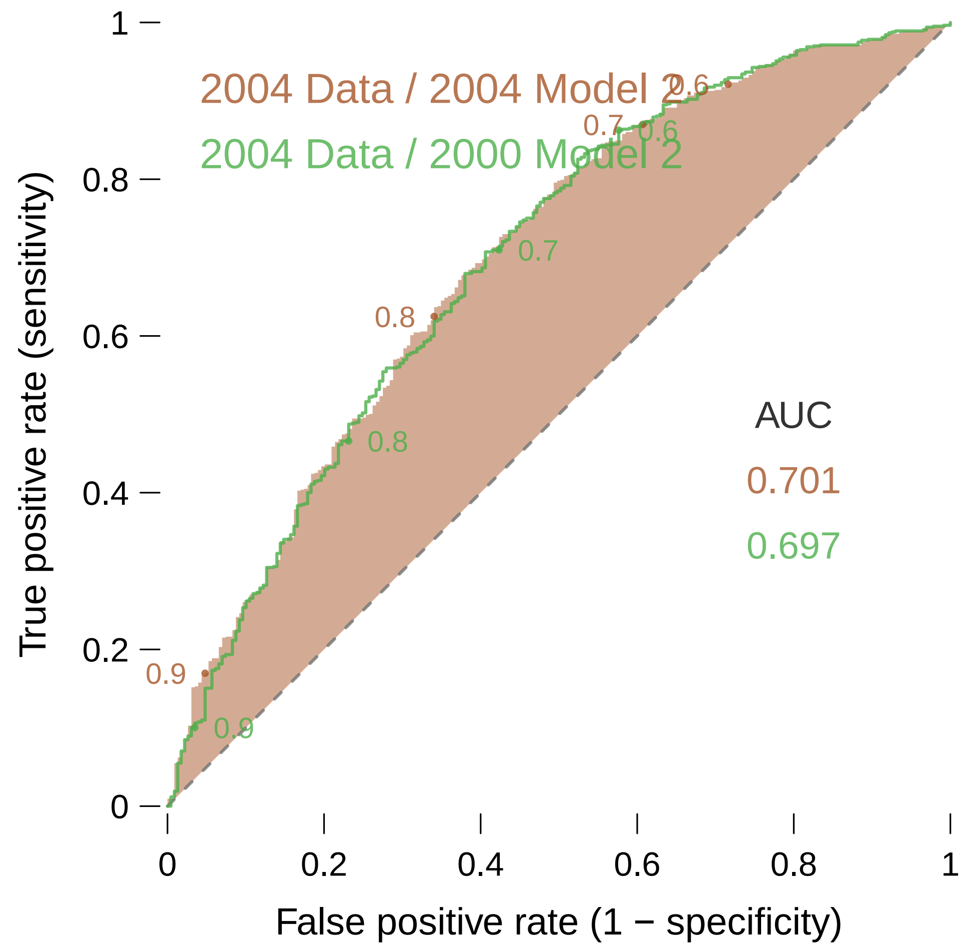
Predicts fairly well, but hints of nonlinearity – why?




2000 model expects participation of the elderly to tail off more than it does

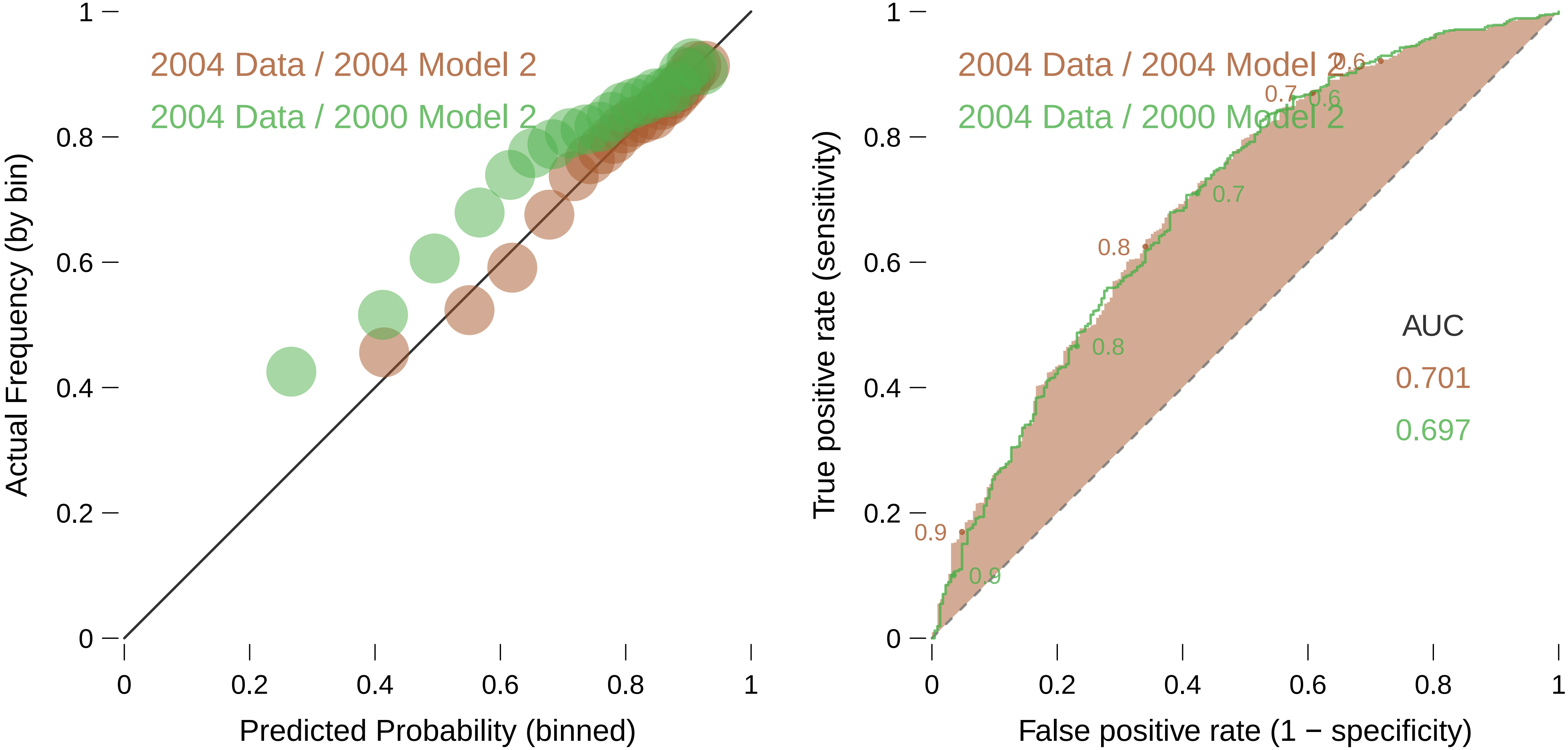
But could that be sample selection bias? Only people who lived to 2000 are sampled!




A truly separate OOS test: predict fresh 2004 ANES data using the 2000 model

But 2004 ANES has massive social desirability bias (78% report voting!)
Pattern of misfit tracks Deufel & Kedar's (2010) theory of who lies about voting




A good out of sample test is expensive and hard to find

Can we find another way to improve on in-sample testing?

# Another way?

Out of sample tests are more persuasive than in-sample:
it's easy to fit the data to which a model has been fitted

But we often lack an out of sample dataset
(or if we have one, we often want to pool it into our analysis)

Can we generate an OOS test using the original data?

# Cross-validation

Out of sample tests are more persuasive than in-sample:
it's easy to fit the data to which a model has been fitted

But we often lack an out of sample dataset
(or if we have one, we often want to pool it into our analysis)

Can we generate an OOS test using the original data?

Amazingly, the answer is yes, by exploiting an analogy:

Subsample : Sample :: Sample : Population

If this analogy applies, we should be able to split the data,
and accurately predict one half the data using a model fit on the other half

Could repeat for many different splits (slow & biased but more robust)

Or just leave out one case at a time (fast & unbiased by less robust)

This is **cross-validation**: Using CV to calculate any GOF technique yields more reliable results than in-sample versions of the test

# Cross-validation

To cross-validate, we need an error metric

For linear models, this is usually the absolute prediction error
or the root mean squared error

For binary, we'll start with percent correctly predicted

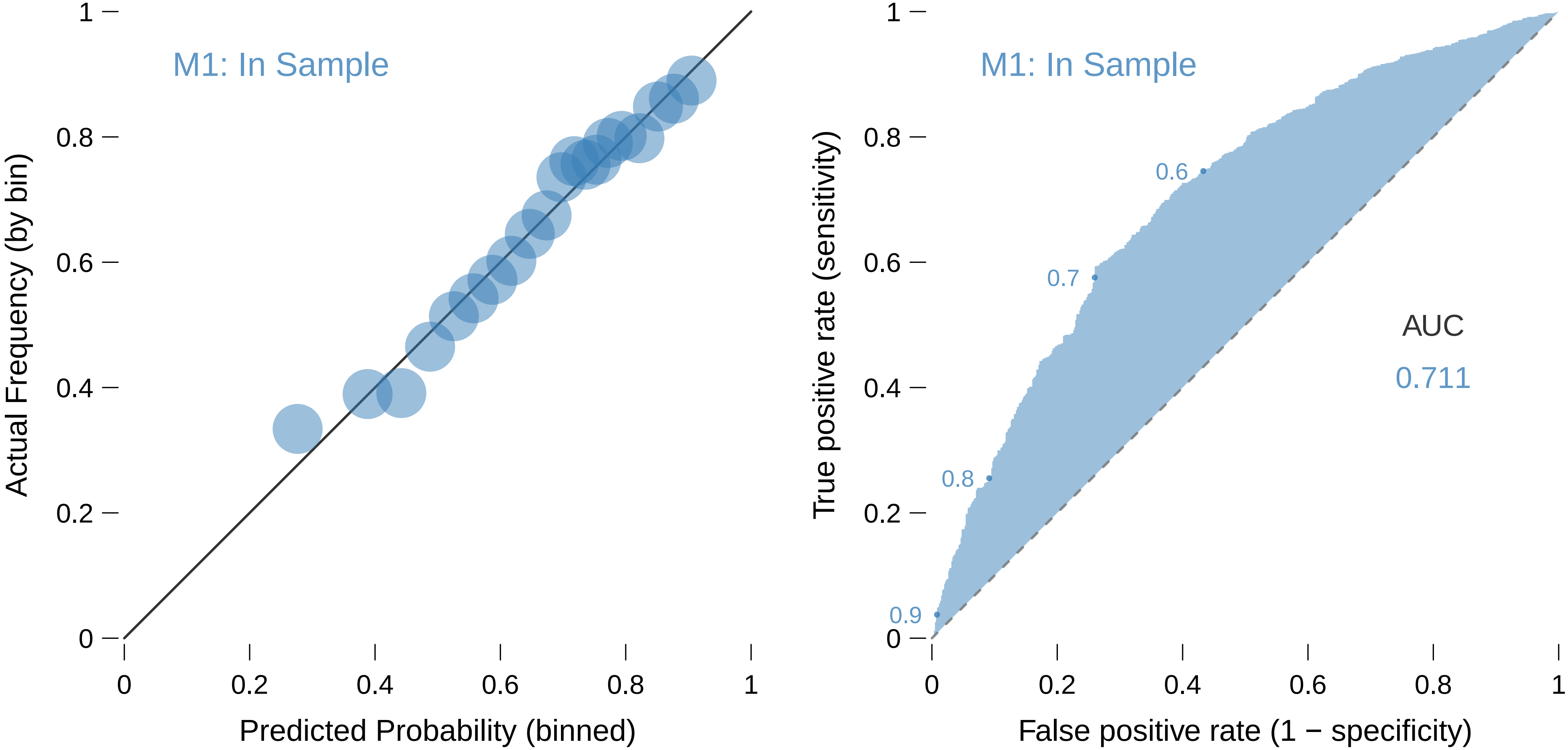
| | in-sample PCP | cv-PCP |
|---|---|---|
| Model 1 | $69.99\%$ | $69.94\%$ |
| Model 2 | $69.77\%$ | $69.56$ |

Here, CV error is almost equal to in-sample error

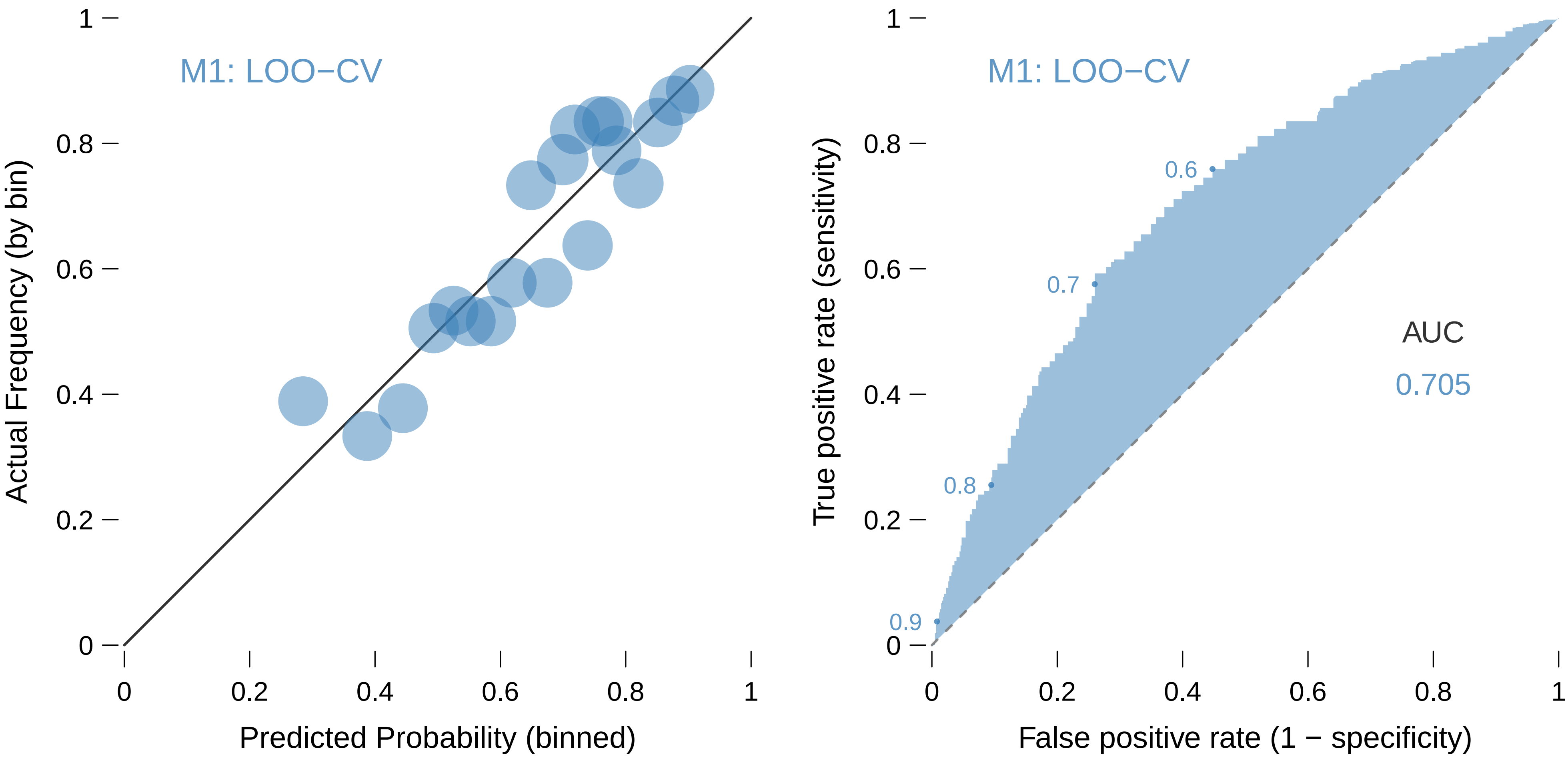
Typically it's higher; sometimes much higher

Persuasive evidence that we have a good fit, not a spurious one

But PCP is a limited GoF test; we can do better

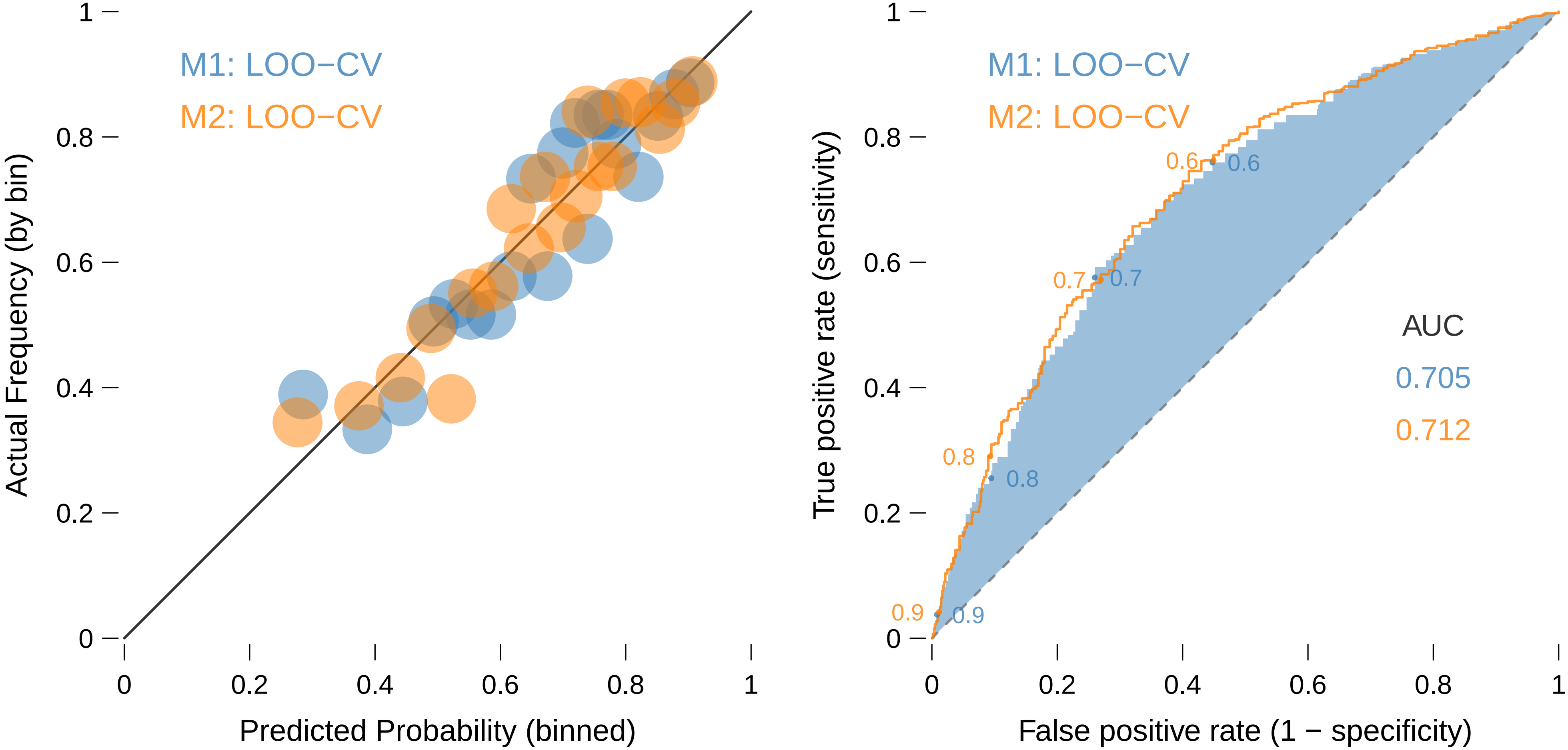



Recall our in-sample fits for Model 1



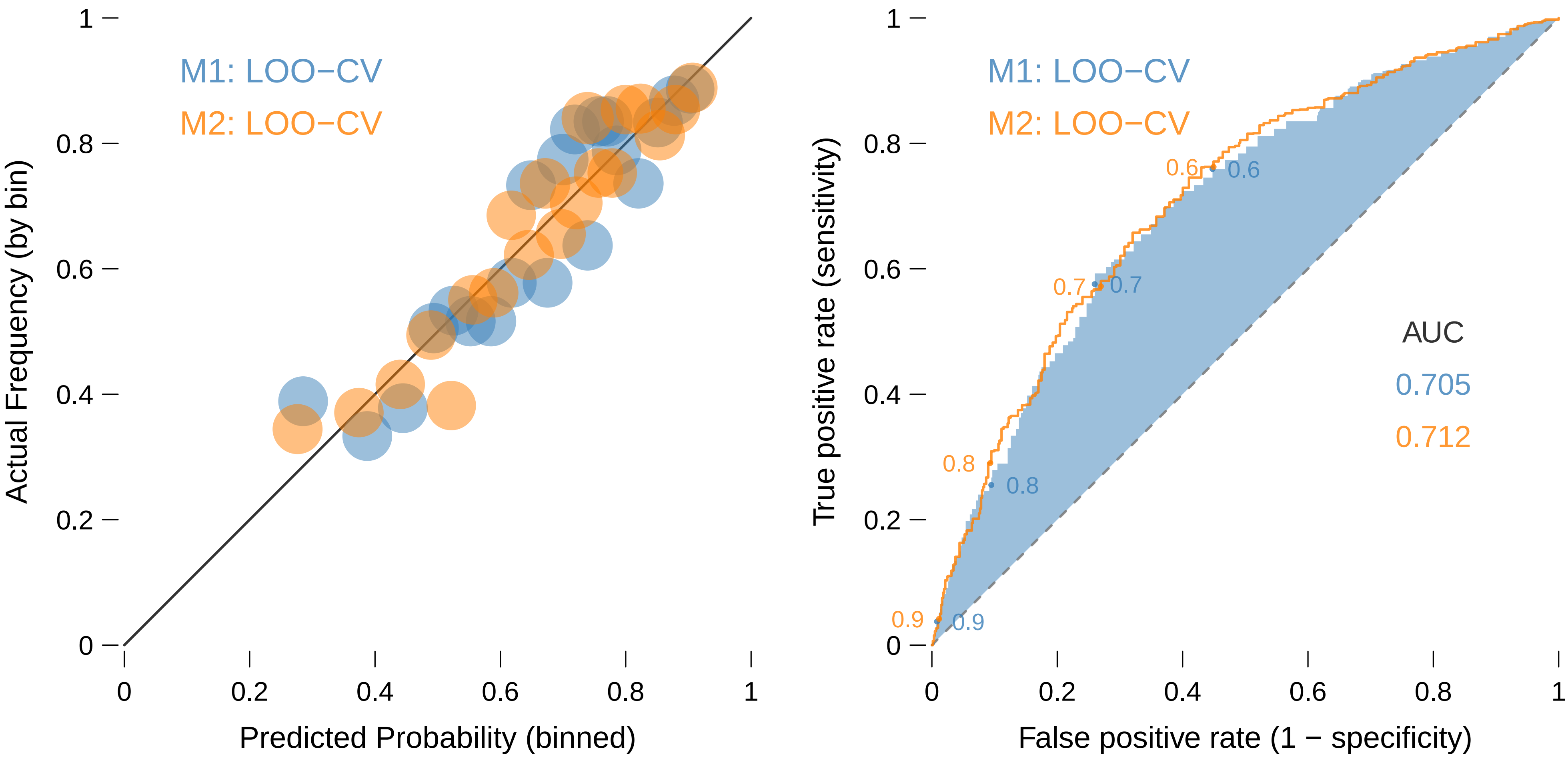

Now compare to our cross-validated fit – slightly weaker, but still pretty good

*Intuition check: How does leave-one-out cross validation work for these tests?*



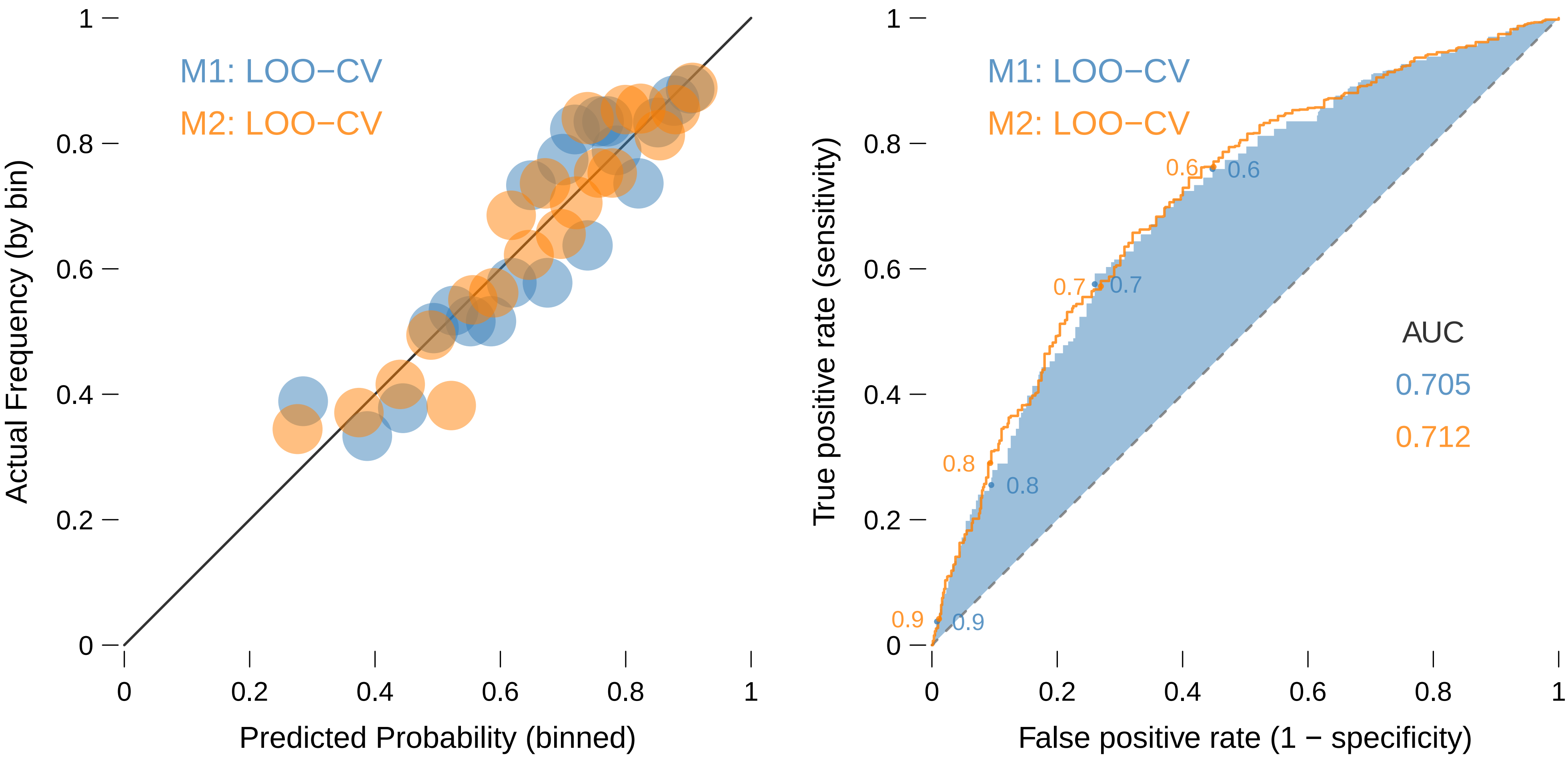

And now we have more faith than ever that
Model 2 is a genuine albeit small improvement on Model 1

*When should you use cross-validation? When you care if you're right*



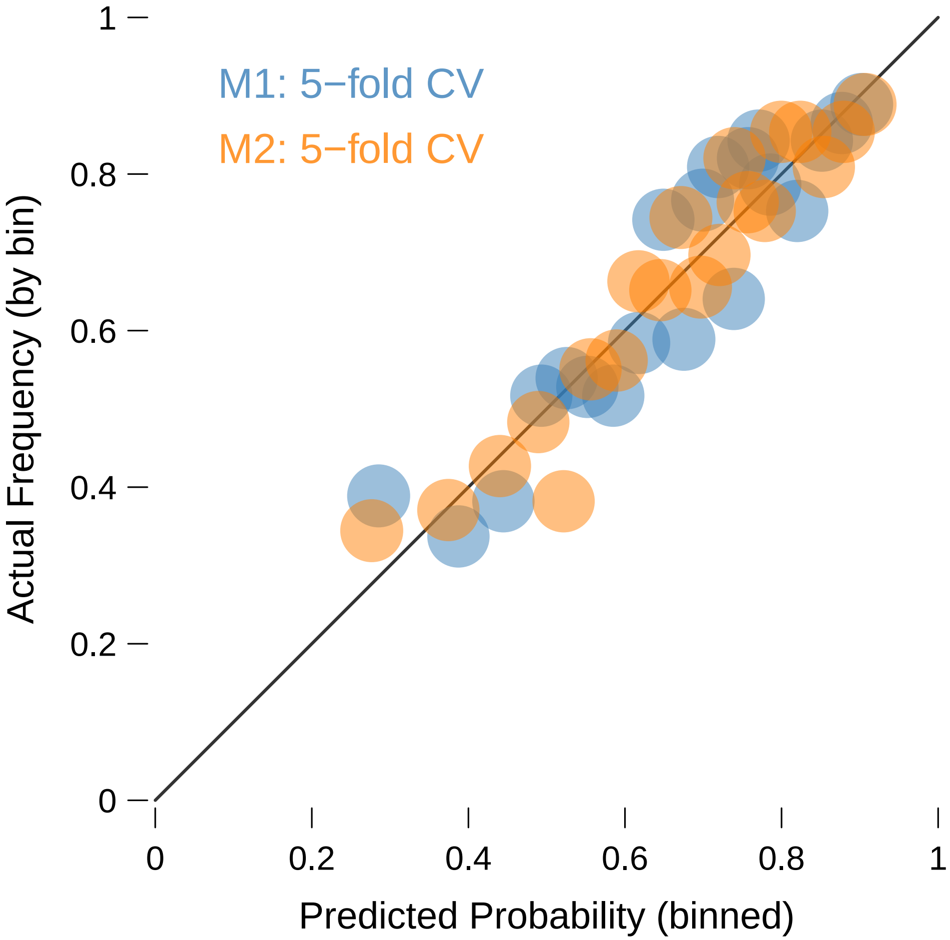

*Did it matter that we did "leave-one-out" instead of $k$-fold CV?*

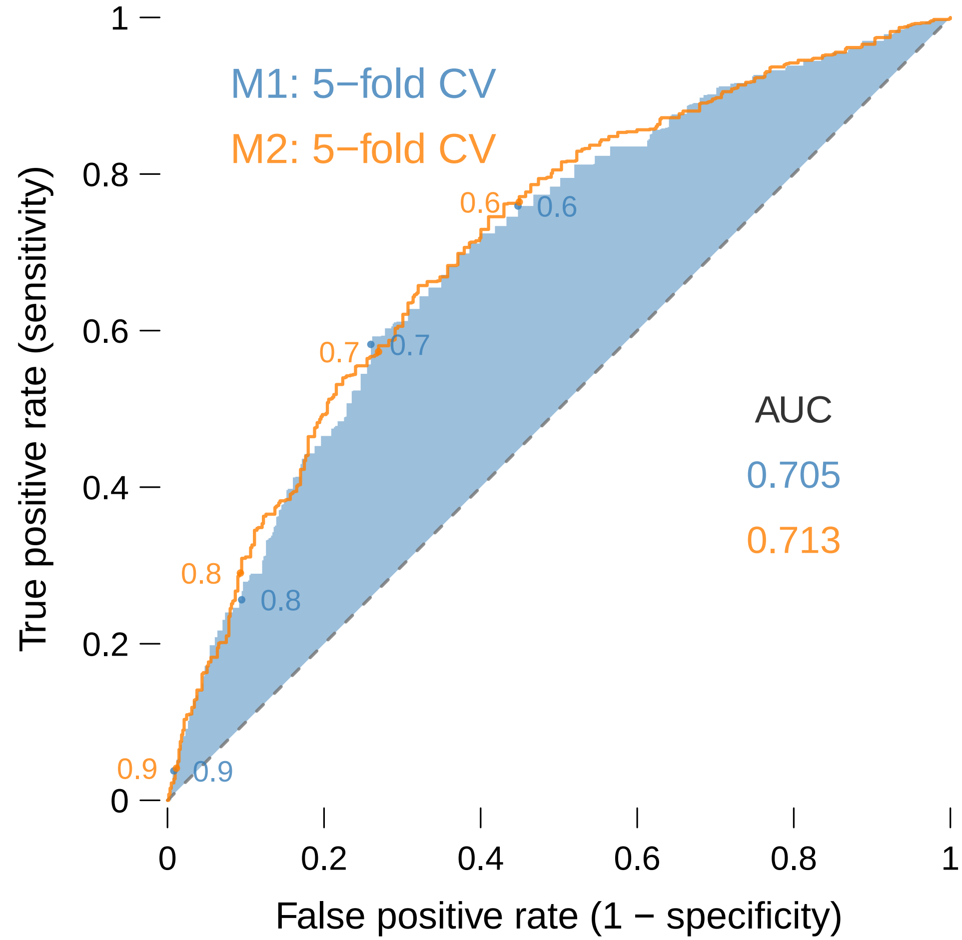
Perhaps repeatedly splitting the data into, say,
$k = 5$ equal samples would be more robust?




*Did it matter that we did "leave-one-out" instead of $k$-fold CV?*

Intuition: randomly select 80% of cases to "train" the model,
then predict remaining 20% "test" cases; repeat & average prediction error




*Did it matter that we did "leave-one-out" instead of $k$-fold CV?*

LOOCV vs $k$-fold CV makes no difference here – but sometimes it might

# GOF: statistics versus graphics

Single number GOF tests are like grades:
one number to tell if you're "good" or "bad"
Always graded on a curve – but not always certain what the curve is

Graphical GOF tests are like evaluations:
Form the basis for targeted improvement
Tells us where the "bad grade" came from

- Did your model fail one test but perform brilliantly otherwise?
- Is your model mediocre across the board?
- Is there a pattern to your model's failures?

Best GOF tests make you think about the model

Remember! You can use a diagnostic graph to make your paper better even if you don't have the space to publish the graph in the paper

# Another model: lagged vote

Back to 2000 data

There's one more variable to try: the lagged behavior of the voter

Past behavior looks like a great predictor:

```
      vote00 age hsdeg coldeg married  vote96
 [1,]      1  49     1      0        1      1
 [2,]      0  35     1      0        1      0
 [3,]      1  57     1      0        1      1
 [4,]      1  63     1      0        0      1
 [5,]      1  40     1      0        1      1
 [6,]      1  77     0      0        1      1
 [7,]      0  43     1      0        1      0
 [8,]      1  47     1      1        0      1
 [9,]      1  26     1      1        0      1
[10,]      1  48     1      0        0      1
 ...
```

# Wow! What an improvement in fit

We re-use our explanatory variables (Age, HSdeg, Coldeg)

Then add a new control, the lagged dependent variable (LDV)

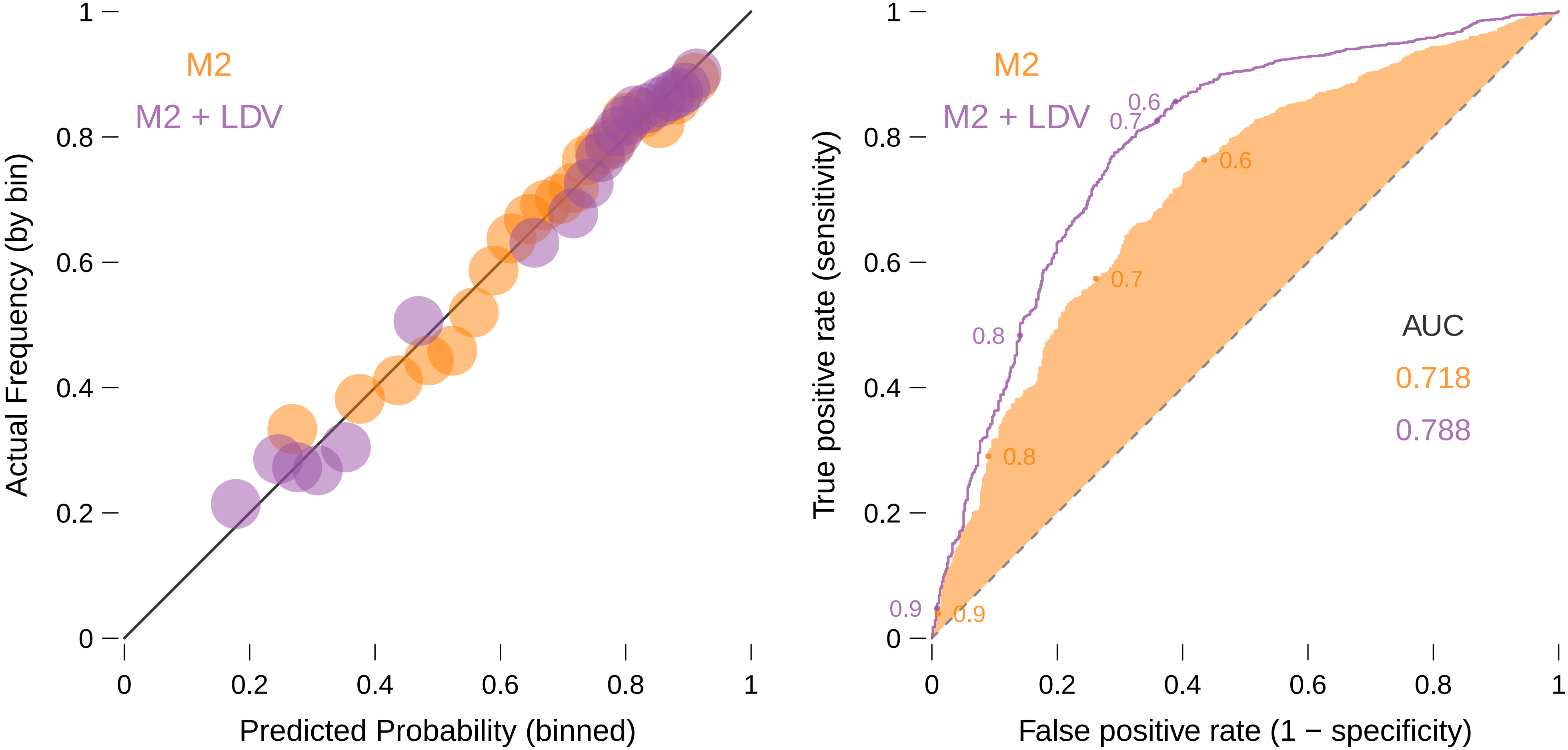
$$\Pr(\text{Vote}_{i,2000}) = \text{logit}^{-1}(\mathbf{x}_{i,2000}\boldsymbol{\beta} + \gamma\text{Vote}_{i,1996})$$

$t$-stat for $\hat{\gamma} = 14.8$ ($p$-value $= 0.0000000\ldots$)

likelihood ratio $= 235.7$ on 1 df ($p$-value $= 0.000000\ldots$)

reduction in BIC $= 228.2$ (*huge*)

PCP $= 77.29\%$ (vs. 69.77% without LDV or 65.7% for the null model)




Actual vs. Predicted is much improved: circles have moved towards [0,0] and [1,1]

ROC show huge gains in sensitivity and specificity at most thresholds

| | M2 | M3 |
|---|---|---|
| Age | 0.061 | −0.004 |
| | (0.017) | (0.019) |
| $\text{Age}^2$ | −0.000318 | 0.000139 |
| | (0.000170) | (0.000188) |
| High School Grad | 1.099 | 0.712 |
| | (0.181) | (0.199) |
| College Grad | 1.053 | 0.756 |
| | (0.132) | (0.141) |
| Married | 0.373 | 0.283 |
| | (0.110) | (0.119) |
| Vote96 | | 1.949 |
| | | (0.132) |
| Constant | −2.866 | −1.758 |
| | (0.422) | (0.462) |
| log likelihood | −1028.7 | −910.9 |
| AIC | 2069.5 | 1835.8 |
| BIC | 2102.4 | 1874.2 |
| $N$ | 1783 | 1783 |

But all our substantive effects have shrunk! Omitted variable bias in M2?

| | M2 | M3 |
|---|---|---|
| Age | $0.061$ | $-0.004$ |
| | $(0.017)$ | $(0.019)$ |
| Age$^2$ | $-0.000318$ | $0.000139$ |
| | $(0.000170)$ | $(0.000188)$ |
| High School Grad | $1.099$ | $0.712$ |
| | $(0.181)$ | $(0.199)$ |
| College Grad | $1.053$ | $0.756$ |
| | $(0.132)$ | $(0.141)$ |
| Married | $0.373$ | $0.283$ |
| | $(0.110)$ | $(0.119)$ |
| Vote96 | | $1.949$ |
| | | $(0.132)$ |
| Constant | $-2.866$ | $-1.758$ |
| | $(0.422)$ | $(0.462)$ |
| log likelihood | $-1028.7$ | $-910.9$ |
| AIC | $2069.5$ | $1835.8$ |
| BIC | $2102.4$ | $1874.2$ |
| $N$ | $1783$ | $1783$ |

Maybe not: we know $\text{Vote96} = f(\text{Age}, \text{Educ})$, so LDV is *absorbing* these variables

# Prediction versus Explanation

Prediction and explanation are different goals

**Explanation:** What is the relationship between some particular $x$ and $y$?

E.g., "Does age affect the probability of voting, and if so, how much?"

**Prediction:** What $y$ do we expect to see under given circumstances?

E.g., "What percentage of the voters will turn out in some district?"

When we are *predicting*, we can include or exclude any $x$ to aid fit.

When we are *explaining*, we are usually interested in getting a precise estimate of the effect of a particular $x$, as suggested by an underlying causal model

The best *predictive* model is the one with the best fit, with caveats

1. What kind of fit? Out-of-sample or CV

2. Curve-fitting hinders prediction,
   plus we don't care what the parameters are *per se*

3. Often relies on a boring mediating or spurious variable:
   Best predictor of turnout is whether you plan to vote

The best *explanatory* model depends on which relationship we want to explore

1. May depend on which $x$ we are interested in (i.e., which QoI we want to estimate)

2. We often exclude a better fitting model
   because it reduces precision of parameters too much

3. This should be done in a principled way –
   think about mediation, try to sketch the causal model

# Conclusions

This week, we've seen three crucial steps in maximum likelihood modeling:

1. Model specification and estimation

2. Model interpretation

3. Model re-specification and selection

Though our discussion was in terms of binary choice,
much carries over to other models

Even your code will be a good template

Next up: ordered and multinomial choice