Exposure-response modeling for binary data

Outline

- Encountering binary data
- Groundwork
  - Notational conventions
  - Modeling from a probabilistic viewpoint
- Bernoulli distribution
  - Defining models
  - Maximum likelihood
- Visualizing relationships
- Models, model checking, and measures of effect

Key learning objectives

At the end of today's session, I hope you will know:

- How to make effective exploratory analysis plots for binary data
- The basic concepts of maximum likelihood estimation for binary data models
- How to fit and evaluate binary data models
- How to interpret model terms

What do we mean by binary data?

- Outcomes that have two possible values
- Can be categorical by nature or created by discretizing a categorical or continuous variable
- Examples
  - Objective response per RECIST (response / non-response)
  - Incident adverse event (yes / no)
  - Alive without disease progression vs. progressive disease or died
  - Coin flip (heads / tails)

What makes binary data interesting?

- If you are coming from a Pop PK background, moving away from models with Normal (or log-Normal) residuals may force you to think at a new level of abstraction.
- The Bernoulli distribution for binary data is (in most respects) as simple a statistical distribution as there is. Good place to start for many basic statistical concepts.
- There are fewer choices to make and fewer assumptions to check when dealing with binary data (e.g. as compared to time-to-event data).

Basic Notation

For now, we will use this high-level notation:

- Greek letters are model parameters
  - μ model parameter ("intercept")
  - β model parameter (coefficient for effect of exposure or covariate)
- Upper case Roman letters are random variables
  - C, D, T : exposure (think of a steady-state exposure metric for now, e.g. $\mathrm{CAVG}_{\mathrm{ss}}$), or dose, or just treatment indicator.
  - X : covariates
  - Y : As-yet-unrealized / unobserved response ("DV")
- Lower case Roman letters are observed values
  - y : Observed value for Y

Probability versus statistics

In some cases (e.g., when setting up a model) we will be thinking in the data generating / probability direction:

$$\mu,\ \beta,\ C,\ X \quad \overset{\text{Probability}}{\longrightarrow} \quad Y$$

In other cases we will be thinking in the model estimating / statistics direction:

$$\mu,\ \beta \quad \overset{\text{Statistics}}{\longleftarrow} \quad C,\ X,\ y$$

Probability models: continuous variable

Throughout the course we will conceptualize models from a probabilistic or `data generating` viewpoint.

We'll use expressions like:

$$\text{Height} \sim \text{Normal}(\mu, \sigma)$$

Read as:

Height follows a normal distribution with mean μ and standard deviation σ

If we know μ and σ we can

- Make probabilistic statements about height in the population
- Simulate heights

Probability models: binary variable

Suppose we define a binary variable as

$$Tall = \begin{cases} 1 & Height > 200cm \\ 0 & Height \leq 200cm \end{cases}$$

Then our model might be

$$\text{Tall} \sim \text{Bernoulli}(\pi)$$

Read as:

Being tall follows a Bernoulli distribution with the probability of being tall equal to π

If we know π, we can

- Make probabilistic statements about the number of tall people in a random sample from the population
- Simulate data

Probability density (mass) function

Both of these link to specific probability density or probability mass functions:

The *normal* (or *Gaussian*) pdf for height

$$p(\mathit{height} \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(\mathit{height} - \mu)^2\right\}$$

and the *Bernoulli* pmf for Tall

$$p(\mathit{tall} \mid \pi) = \pi^{\text{tall}}(1-\pi)^{1-\text{tall}}$$

The likelihood function

- Suppose you observed a measured value $Y = y_{obs}$.
- If we view the pdf or pmf as a function of the parameters, conditional on some observed data y_{obs}, we refer to the function as a likelihood function
- It is the ***same expression*** as the PDF or pmf, but we now view it as a function of the parameters given the data instead of as a function of the data given the parameters.

$$L(\theta \mid y_{obs}, x) = p(y_{obs} \mid \theta, x)$$

Modeling from a probabilistic point of view: The likelihood function

$$L(\theta \mid y_{obs}, x) = p(y_{obs} \mid \theta, x)$$

- During model development we generally do not know the values of the parameters θ and use the observed data to estimate those parameters.
- The likelihood function contains information about what those parameter values might be.
- We will talk about two different approaches that exploit the likelihood function to estimate θ:
  - Maximum likelihood estimation
  - Bayesian statistical analysis

The Bernoulli likelihood for one observation

Likelihood for a single Bernoulli observation

$$l(\pi \mid Y_i = y_i) = P(Y_i = y_i) = \begin{cases} \pi & y_i = 1 \\ 1 - \pi & y_i = 0 \end{cases}$$

We often see this written more compactly as

$$l(\pi \mid Y_i = y_i) = \pi^{y_i}(1 - \pi)^{(1-y_i)}$$

The Bernoulli joint likelihood function

Joint likelihood for a sample of independent Bernoulli observations

$$
\begin{aligned}
l(\pi|\mathbf{Y}=\mathbf{y}) &= \prod_{i=1}^{n} P(Y_i = y_i) \\
&= \prod_{i=1}^{n} \pi^{y_i}(1-\pi)^{(1-y_i)} \\
&= \pi^{(\#\text{ of "ones"})}(1-\pi)^{n-(\#\text{ of "ones"})}
\end{aligned}
$$

Joint log likelihood:

$$
\begin{aligned}
L(\pi \mid \mathbf{Y}=\mathbf{y}) &= \log l(\pi|\mathbf{Y}=\mathbf{y}) \\
&= (\#\text{ of "ones"})\log(\pi) + (n - \#\text{ of "ones"})\log(1-\pi)
\end{aligned}
$$

Maximum likelihood estimation for a simple model

Maximum likelihood estimates are the values of the parameters (p) which maximize the likelihood.

Derivative of joint log likelihood:

$$\frac{\mathrm{d}L}{\mathrm{d}\pi} = \frac{(\#\,\text{of successes})}{\pi} - \frac{(n - \#\,\text{of successes})}{1-\pi}$$

$$\frac{\mathrm{d}L}{\mathrm{d}\pi} = 0 \iff \pi = \frac{(\#\,\text{of successes})}{n}$$

- Estimator for $\hat{\pi}$ is as expected from probability perspective
- Note: common value of π for all subjects

Maximum likelihood for regression models

- When we have a full logistic regression model with covariates, there is no analytical solution to the likelihood equations.
- However there is a numerical root finder that is specially tailored to the structure of logistic (and all GLM) models: the `Newton-Raphson` method ([see @2795]).
- Historically, that was one reason for using GLMs.
- That reason is often not compelling any more, but there are still advantages in interpretation, e.g. versus non-monotonic models.

Workbook 01: Exploring the binomial density

- Binomial density
- Likelihood function

Binary Data Example

- This data set is comprised of a two-week study (protocol A) and a six-week study (protocol B)
- The data set includes patients of type `PT2`, a patient type that (let's say) has not been studied at higher dose levels.
- Suppose further that a dose level under consideration for a phase 2 study in `PT2` would have typical value exposures near 2.5 ug/mL.
- The objective of *this* analysis will be to determine whether the AE rate will be "sufficiently low" in `PT2` at that exposure.

Data snapshot

```
. # A tibble: 5 x 6
.   STUDYID USUBJID PBO   CAVGSS AE01  AETOXGR
.   <fct>   <fct>   <chr> <chr>  <chr> <fct>
. 1 PROTA   UID-001 PBO   0.00   0     Mild
. 2 PROTA   UID-002 PBO   0.00   0     Mild
. 3 PROTA   UID-003 PBO   0.00   0     Mild
. 4 PROTA   UID-004 TRT   1.98   0     Mild
. 5 PROTA   UID-005 TRT   1.37   0     Mild
```

- `STUDYID`: protocol
- `USUBJID`: unique subject id
- `AE01`: adverse event of grade 3 or higher

Visualizing relationships

Objective:

Plot the probability of a grade 3 or higher AE vs predictor

Types of plots will depend on the predictor variable:

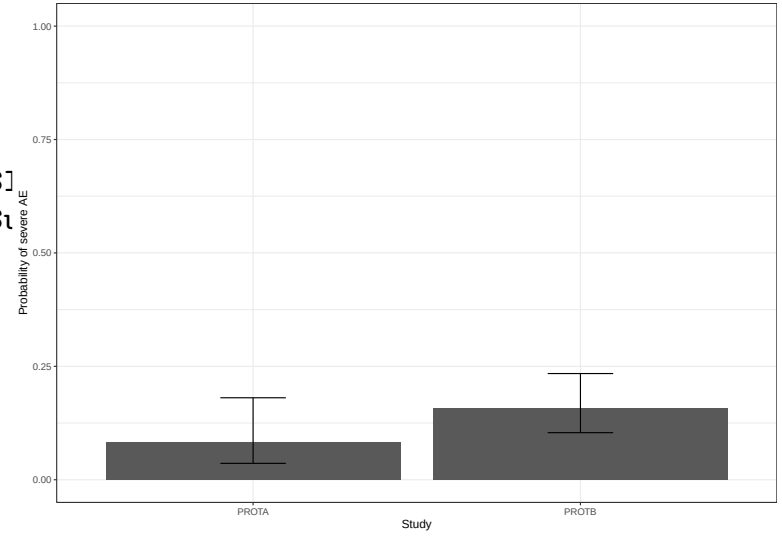
- Categorical variable
- Continuous variable

Relationship with categorical variable

- x-axis: Categorical variable (`STUDYID`)
- y-axis: Probability of Grade 3+ AE (`AE01`)
- geom: `bar`, `point`

Probability vs categorical variable

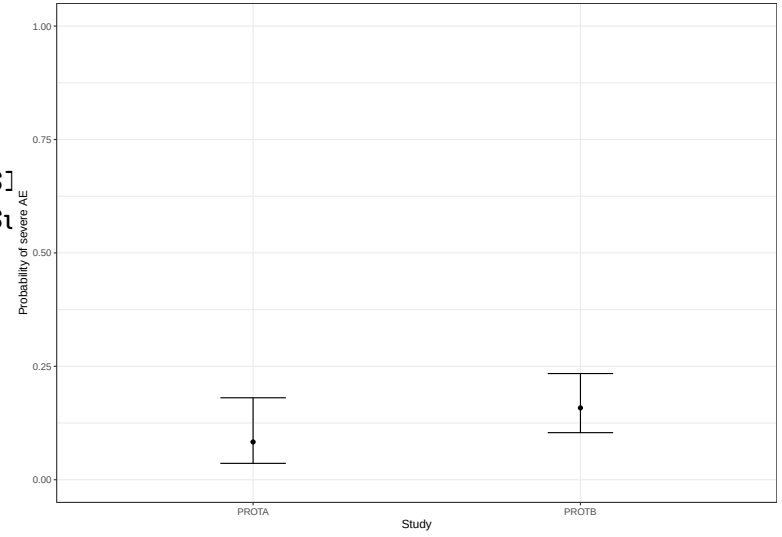
```
aedat %>%
  group_by(STUDYID) %>%
  summarise(x = sum(AE01), n=n()) %>%
  mutate(phat = x / n,
         lcl = binom.confint(x,n,methods = "wilson")$l
         ucl = binom.confint(x,n,methods = "wilson")$u
  ggplot(aes(x=STUDYID, y=phat)) +
  geom_col() +
  geom_errorbar(aes(ymin=lcl, ymax=ucl),
                width = 0.2) +
  ylim(0,1) +
  labs(x='Study', y='Probability of severe AE')
```



Note: `binom.confint` is in the `binom` package

Alternative to bar plot

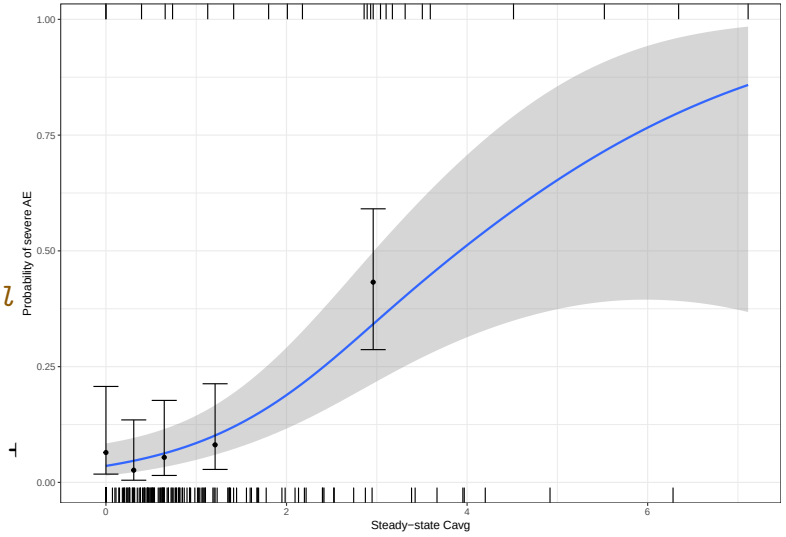
```
aedat %>%
  group_by(STUDYID) %>%
  summarise(x = sum(AE01), n=n()) %>%
  mutate(phat = x / n,
         lcl = binom.confint(x,n,methods = "wilson")$l
         ucl = binom.confint(x,n,methods = "wilson")$u
  ggplot(aes(x=STUDYID, y=phat)) +
  geom_point() +
  geom_errorbar(aes(ymin=lcl, ymax=ucl),
                width = 0.2) +
  ylim(0,1) +
  labs(x='Study', y='Probability of severe AE')
```



Note: `binom.confint` is in the `binom` package

Probability vs a continuous variable

```
ggplot() +
  # Add tick marks at top and bottom
  geom_rug(data = filter(dat_eda, AE01 == 0),
           aes(x = CAVGSS), sides = "b") +
  geom_rug(data = filter(dat_eda, AE01 == 1),
           aes(x = CAVGSS), sides = "t") +
  # Add smooth mean function.   Note: Use of gam with
  #  binomial family to keep predictions on (0,1) scal
  geom_smooth(data = dat_eda,
              aes(x = CAVGSS, y = AE01),
              method='gam', formula=y~s(x),
              method.args = list(family='binomial')) +
  # Add points and CIs
  stat_summary(
    data = dat_eda,
    aes(x = MedConc, y = AE01, group = MedConc),
    fun = function(y) sum(y) / length(y),
    geom = "point"
  ) +
  stat_summary(
    data = dat_eda,
    aes(x = MedConc, y = AE01),
    # Wilson CI recommended by Agresti and Coull (2000) review paper
    fun.min = function(y) {
      binom.confint(sum(y),length(y),
                    methods = "wilson")$lower
      },
    fun.max = function(y) {
      binom.confint(sum(y), length(y),
                    methods = "wilson")$upper
      },
    geom = "errorbar"
  ) +
  labs(x='Steady-state Cavg', y='Probability of severe AE')
```



Plotting tips

- Break exposure into quartiles, compare incidence rate across quartiles
  - Textbook variance formula $\frac{p(1-p)}{n}$ isn't the best choice with small sample sizes and probabilities near zero or one
  - Recommendation: Use Wilson interval (implemented in `binom.confint`)
- Rug plots (`geom_rug`)
- Stratify by other covariates of interest

Workbook 02: Visualizing binary data

Measures of effect: odds ratio

- `Conditional probability` of an event, conditional on treatment "1".

$$P(Y = 1 \mid T = 1) \in (0, 1)$$

- `Odds` of an event, conditional on treatment "1":

$$\frac{P(Y = 1 \mid T = 1)}{P(Y = 0 \mid T = 1)} = \frac{P(Y = 1 \mid T = 1)}{1 - P(Y = 1 \mid T = 1)} \in (0, \infty)$$

- `Odds ratio` of an event, for treatment "1" versus treatment "0":

$$\frac{P(Y = 1 \mid T = 1) \,/\, P(Y = 0 \mid T = 1)}{P(Y = 1 \mid T = 0) \,/\, P(Y = 0 \mid T = 0)} \in (0, \infty)$$

Measures of effect: relative risk

Relative risk of an event, for treatment "1" versus treatment "0":

$$\frac{P(Y = 1 \mid T = 1)}{P(Y = 1 \mid T = 0)} \in (0, \infty)$$

Anecdotally, this is often the preferred / most interpretable way to quantify efficacy.

NB: Odds ratio and relative risk are sometimes confused with each other. Note the difference.

Other measures of effect for binary data

Difference in probability of events, for treatment "1" versus treatment "0":

$$P(Y = 1 \mid T = 1) - P(Y = 1 \mid T = 0) \in (-1, 1)$$

Often undesirable: do you want to treat the difference between 3% and 5% the same way that you treat the difference between 23% and 25% ?

The logit transform

- The logit, or "log odds" function, `qlogis()` in R

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) \in (-\infty, \infty)$$

- The standard logistic function (also called the "expit") is the inverse of the logit, `plogis()` in R:

$$p = \text{expit}(x)$$

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

$$= \frac{\exp(x)}{1+\exp(x)}$$

Other "link functions"

- The `logit` function takes us from the unit interval to the full Real line:

$$(0,1) \xrightarrow{\text{logit}} \mathbb{R}$$

- An alternative "link" function is the `probit` :

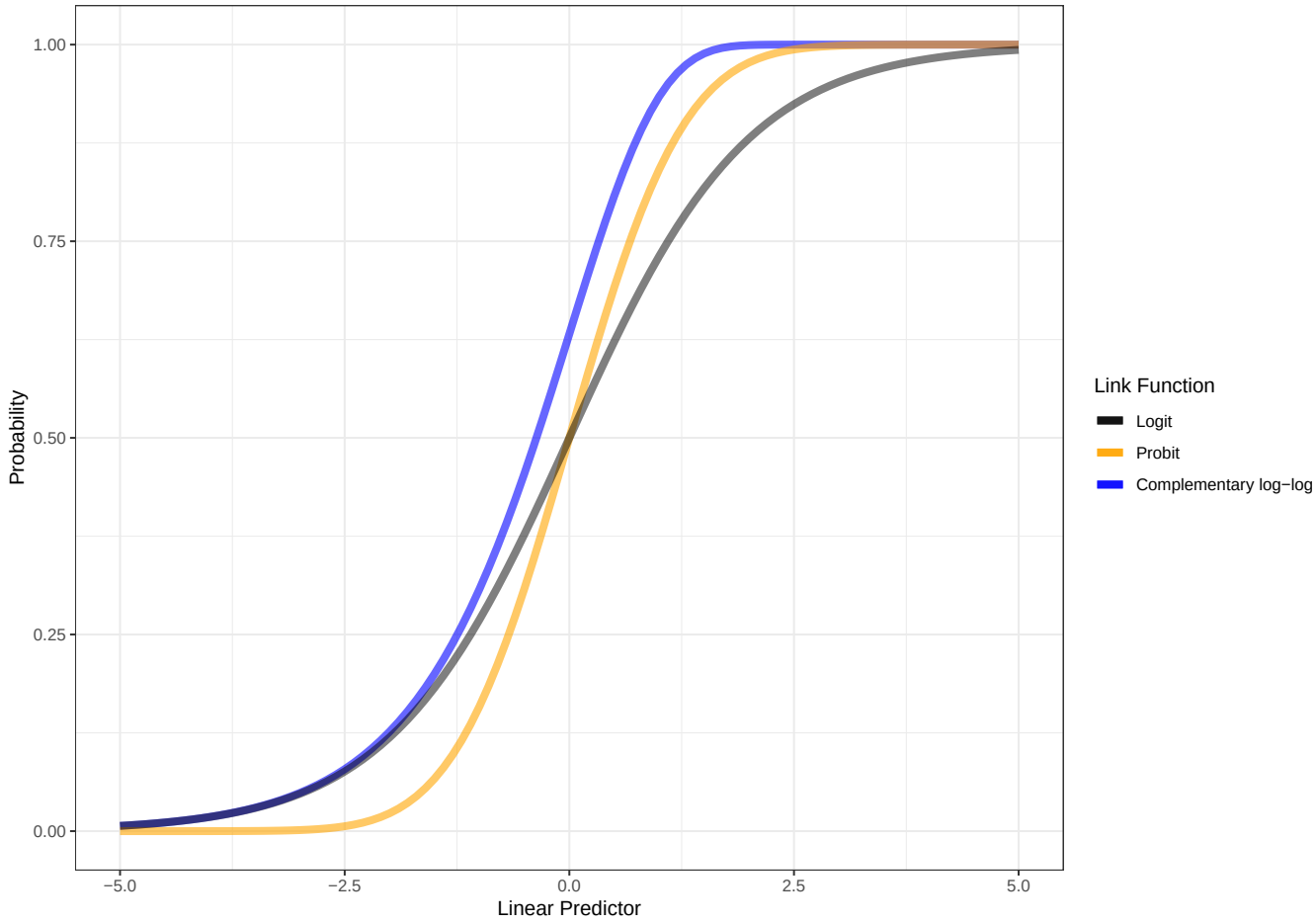
$$\text{probit}(p) = \Phi^{-1}(p)$$

where Φ is the Normal Cumulative Density Function (CDF):

$$\Phi(x) = P(\text{Std. Normal Variate} < x)$$

- Alternative "link" function: `complementary log-log`
  - $\text{cloglog}(p) = \log(-\log(1-p)$
  - It's an asymmetrical link function
  - Shows up when transforming other models to binary outcomes, e.g. Poisson model to binary model

More About Link Functions



Workbook 03: Exploring odds ratios, relative risks, and the logit transformation

A logit-link GLM

A GLM with a logit link and Bernoulli (or more generally, Binomial) distribution is referred to as a logistic regression.

A logistic regression with exposure (C_i) as the sole predictor would be expressed as:

$$Y_i \sim \text{Ber}(\pi_i) \text{ where } \pi_i = \text{expit}(\mu + \beta C_i) \; ; \; i = 1, \ldots, n$$

Or equivalently:

$$Y_i \sim \text{Ber}(\pi_i) \text{ where } \text{logit}(\pi_i) = \mu + \beta C_i \; ; \; i = 1, \ldots, n$$

Logistic regression is a type of GLM

- The `generalized` in ***generalized linear model*** refers to the non-Normal residuals (in this case, Bernoulli residuals).
- The `linear` in ***generalized linear model*** refers to the fact that the right hand side of, e.g.

$$\text{logit}(\pi_i) = \mu + \beta C_i$$

is linear in the parameters (i.e. it is a linear function of μ and β).

Quiz

Which (if any) of the following is linear in the parameters?

$$\mu + \beta \log(C) \qquad \mu + \frac{\beta}{C} \qquad \mu + \frac{\beta_1 C}{(\beta_2 + C)}$$

Anatomy of a GLM

Taking the following model as an example:

$$Y_i \sim \mathrm{Ber}(\pi_i) \text{ where } \mathrm{logit}(\pi_i) = \mu + \beta C_i^* \ ; \ i = 1, \ldots, n$$

Standard terminology to refer to the model components is:

- $Y_i \sim \mathrm{Ber}(\pi_i)$ is the distribution component of the model. (Sometimes also called the `random` component of the model, but we avoid that terminology is it becomes ambiguous in a `GLMM` context that includes random effects).
- The logit transformation is the link function.
- $\mu + \beta C_i^*$ is the linear predictor.

Key Assumptions for Logistic Regression

- Residual distribution assumptions:
  - Observations are independent (conditional on covariates and exposure).
  - Observations associated with identical covariate values are identically distributed.
  - For Bernoulli residuals, the only remaining distributional "assumption" is that the data are binary (pretty easy to check!).
- Linear predictor:
  - All relevant predictors are in the model and suitably transformed.
  - Interaction terms included where necessary.
  - "Plays nicely" with link function.
- Missing data assumptions.

Fitting a logistic regression in R

```
mod01_glm <- glm(AE01 ~ CAVGSS + BWT + PTTYPE + SEXTXT,
                 data = aedat,
                 family = binomial(link='logit'))
```

- `AE01`: binary outcome variable
- `CAVGSS + BWT + PTTYPE + SEXTXT`: linear predictor
- `family = binomial(link='logit')`: distribution and link function.
  - `logit` is the default link
  - other link functions include `probit` and `cloglog`

General R model syntax

The right-hand side of the formula syntax:

```
~ CAVGSS + BWT + PTTYPE + SEXTXT
```

specifies that the linear predictor is

$$\begin{aligned} \beta_0 \quad + \quad & \beta_{\mathrm{C}}\mathrm{C} + \beta_{\mathrm{W}}\mathrm{W} + \\ & \beta_{\mathrm{PT1}}\mathrm{I}_{\mathrm{PT=PT1}} + \beta_{\mathrm{PT2}}\mathrm{I}_{\mathrm{PT=PT2}} + \beta_{\mathrm{M}}\mathrm{I}_{\mathrm{SEX=M}} \end{aligned}$$

Factor variables in R models

- Most modeling and plotting functions in R treat factor variables differently from numerical variables.
- For modeling functions, the first factor level is treated as the "reference level".
- Choice of reference level determines the interpretation of the model intercept.
- As our field uses *another tool* that doesn't allow character variables, you will probably end up with some categorical variables that have numerical values. . .
  - Remember to "factorize" your categorical variables!
  - `as.factor(SEX)` can even be used directly in model formula!

Output of fitted model

```
summary(mod01_glm)
```

```
.
. Call:
. glm(formula = AE01 ~ CAVGSS + BWT + PTTYPE + SEXTXT, family = binomial(link = "logit"),
.     data = aedat)
.
. Deviance Residuals:
.     Min       1Q   Median       3Q      Max
. -1.8703  -0.4502  -0.3383  -0.2204   2.8435
.
. Coefficients:
.             Estimate Std. Error z value Pr(>|z|)
. (Intercept) -7.00249    4.27313  -1.639   0.1013
. CAVGSS       0.81127    0.17369   4.671    3e-06 ***
. BWT          0.03671    0.05932   0.619   0.5360
. PTTYPEPT1    1.46361    0.69982   2.091   0.0365 *
. PTTYPEPT2    0.92954    0.90557   1.026   0.3047
. SEXTXTMALE   0.06964    0.85167   0.082   0.9348
. ---
. Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
.
. (Dispersion parameter for binomial family taken to be 1)
.
.     Null deviance: 141.36  on 179  degrees of freedom
. Residual deviance: 106.29  on 174  degrees of freedom
. AIC: 118.29
.
. Number of Fisher Scoring iterations: 5
```

Interpreting coefficients in logit-link models: Intercept

- The logistic or expit function can be applied to the intercept or linear predictor to transform it to a probability.
  - $expit(\beta_0) = \frac{1}{1+exp(-\beta_0)}$
  - The probability of an AE when all predictors are 0

Interpretation of effects for categorical covariates

- Let p_1 refer to probability of AE for a `PT1` patient and
- Let p_0 refer to probability of AE for `HV` with exactly the same exposure and covariate values.
- Then:

$$\text{logit}(p_1) - \text{logit}(p_0) = \beta_{\text{PT1}}$$

And since the logit function is the log-odds function, that implies:

$$\frac{p_1/(1-p_1)}{p_0/(1-p_0)} = \exp(\beta_{\text{PT1}})$$

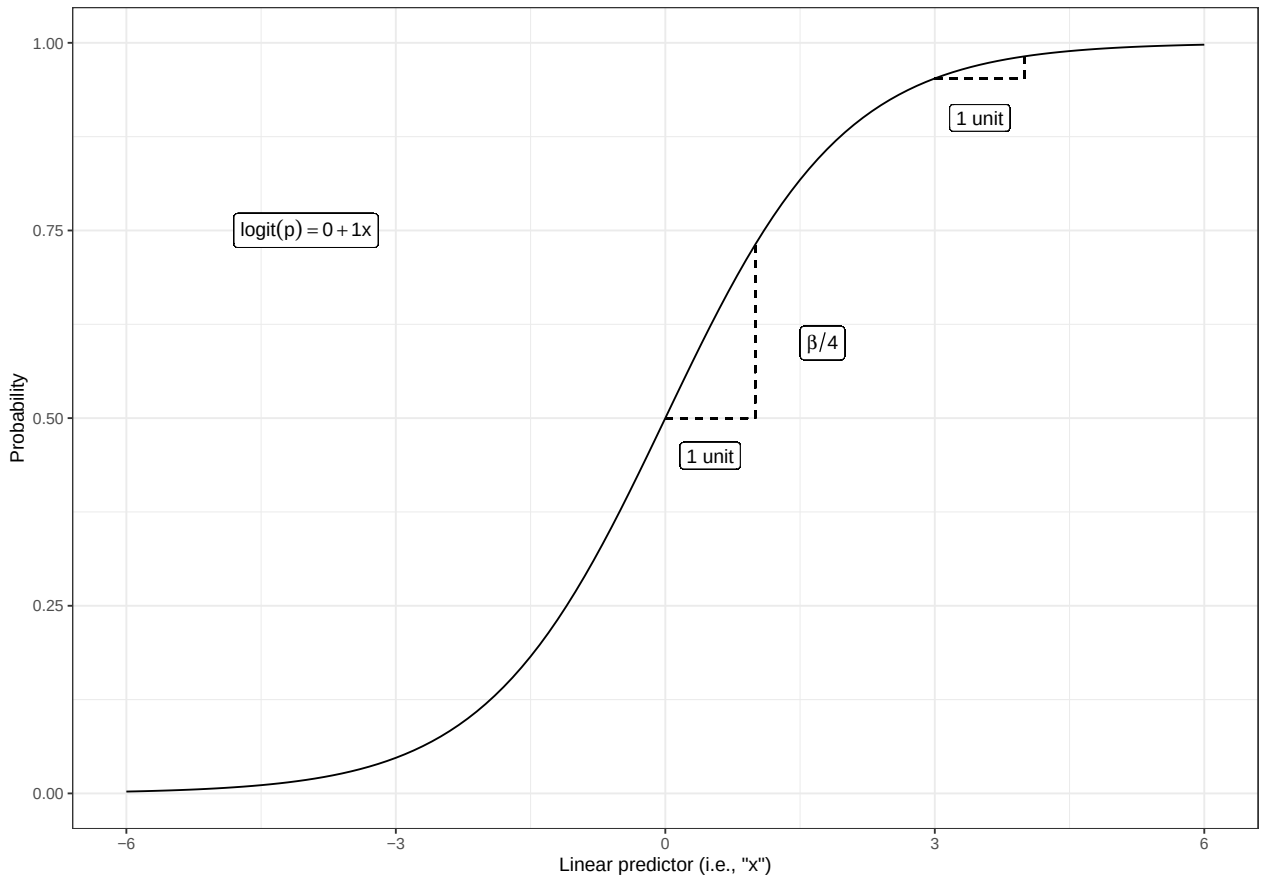
In other words, $\exp(\beta_{\text{PT1}})$ is the odds ratio for the effect of being type `PT1` versus `HV`.

Interpretation of effects for continuous covariates

- For continuous covariates, the odds ratio adjustments are exponential "per unit"
  - For example, the coefficient for body weight is $\beta_{WT} = 0.0367$
  - So, a 10 kg difference in weight corresponds to an odds increase of $\exp(10 \times 0.0367) = 1.44$
- May be advantageous to scale exposure (e.g. divide by 1000) in order to avoid exponentiated coefficients like 1.000123, which might indicate a consequential effect despite being very close to 1
  - try 1.000123^{1000}.

Interpreting covariate effects on the probability scale

- A unit of improvement in x means different things depending upon the reference x
- Largest change always occurs at the inflection point
  - The derivative of the logistic function is maximized at $p = 0.5$ and is maximized by $\beta/4$



Interpretation of covariate effects for probit-link models

- You really can't.
- That's part of the reason for the popularity of logit-link models.
- How much does that really matter?
  - It's certainly nice to have directly interpretable coefficients.
  - But in many cases, the predictive inferences matter more than the direct inferences on parameters.
  - Recommendation: choose link function based on what fits the data better, not based on mathematical convenience. Probit link handles models where tails are more "certain"
- Probit models also arise from a different modeling framework: a latent variable determined by covariates with random Gaussian noise with a threshold for a "true" outcome

Aside: Choice of exposure metric

- Observed vs **model-predicted**
- Which summary measure?
  - Cmin, Cmax, Cavg, AUC, ...
  - Depends on the endpoint and substantive knowledge
- What time horizon? Depends on the context ...
  - When dose 'holidays' or reductions are rare
    - "Early" (e.g., cycle 1)
    - Steady-state
  - When dose holidays or reductions are common, there are a number of common practices:
    - Average to the end of the study and average up to the event (or end of study). PROBLEMATIC!
    - Best to develop a TTE dynamic model with time-varying exposure
    - Decision about initial dose or entire regimen?

Workbook 04: Fitting a logistic regression model

Model diagnostics and comparison

- We'll look at two main types of diagnostics for assessing model fit
  - Residual-based
  - Simulation-based
- Our main method for comparing models will be the quality of out-of-sample predictions

Residual Diagnostics

- Common types of residuals:
  - "Response" residuals (the usual `DV-PRED`).
  - Deviance residuals.
  - Pearson residuals.
- For **all** of these, lower your expectations:
  - Plots usually look "chunky".
  - Natural consequence of binary data.
  - Some sort of smoother needed to aid the eye.

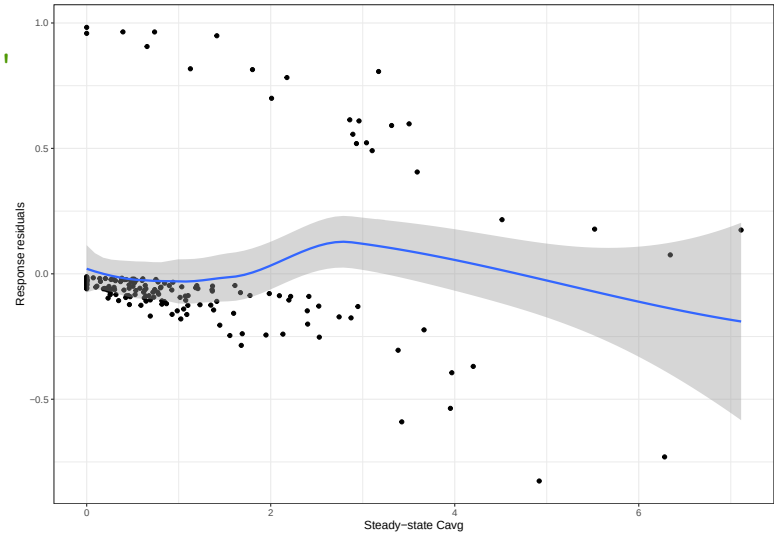
Response residuals

- Easiest residuals to conceptualize.
- Not expected to be "homoscedastic" or even symmetric around the $y = 0$ line.
- Still expected to be at the $y = 0$ line "on average".
- Advantage: departures from $y = 0$ are on the probability scale.
- Don't be fooled:
  - Consequential departures from the $y = 0$ line can be obscured by the plotting scale.
  - Response residuals range from -1 to 1.
  - Plot appearance can depend on distribution of covariates apart from the model

Example of response residuals

```
dat_plus <- dat_mod
dat_plus$res <- residuals(mod01_glm, type = "response'
dat_plus$pred <- fitted(mod01_glm)

ggplot(dat_plus, aes(x = CAVGSS, y = res)) +
  geom_point() +
  geom_smooth() +
  labs(x='Steady-state Cavg', y='Response residuals')
```



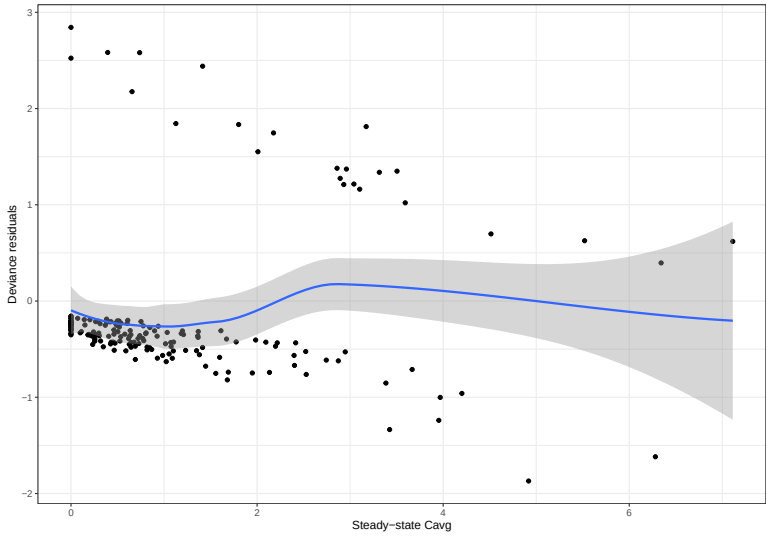
Other residuals (Pearson, Deviance)

- Pearson residuals are like standardizing residuals in linear models
- Deviance residuals are the contributions to the log-likelihood of each data point
- Benefits:
  - More nearly symmetric and homoscedastic (if model is correct).
  - Deviance residuals should be asymptotically normally distributed
- Suggestions:
  - Start with response residuals. You know what these mean.
  - Compare with deviance residuals. If the deviance residual plots look better, take comfort.
  - Don't spend too much time trying to get any residual plots to look "good". They won't. Use them to suggest model refinements and then move on.
- The total residual deviance should be "close" to the residual degrees of freedom

Example of deviance residuals

```
dat_plus <- dat_mod
dat_plus$res <- residuals(mod01_glm, type = "deviance'
dat_plus$pred <- fitted(mod01_glm)

ggplot(dat_plus, aes(x = CAVGSS, y = res)) +
  geom_point() +
  geom_smooth() +
  labs(x='Steady-state Cavg', y='Deviance residuals')
```



Simulation-based Diagnostics

- Randomized quantile residuals
  - Use model to simulate data and calculate quantile of observed values against simulated data
  - Similar to normalized prediction distribution errors (npdes) in pharmacometrics
  - Smooths out discrete residual values
  - DHARMa Package in R makes this easy (`simulateResiduals`)
- Visual Predictive Checks (VPCs)
  - The choice of summary statistics is problem dependent
  - Will see more examples in the next hands-on portion

Quantile residuals for continuous data

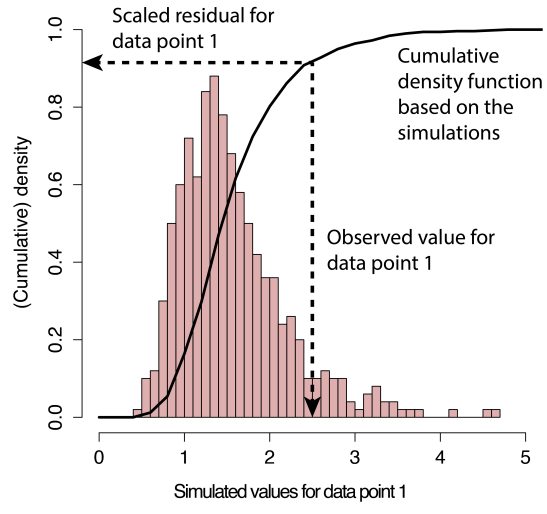
Suppose we have a continuous random variable: $Y \sim f(\theta)$. Then,

$$F(y_i) \sim \text{Uniform}(0, 1)$$

where $F(x) = \int_{-\infty}^{x} f(y)\, dy$ is the cumulative distribution function for Y.

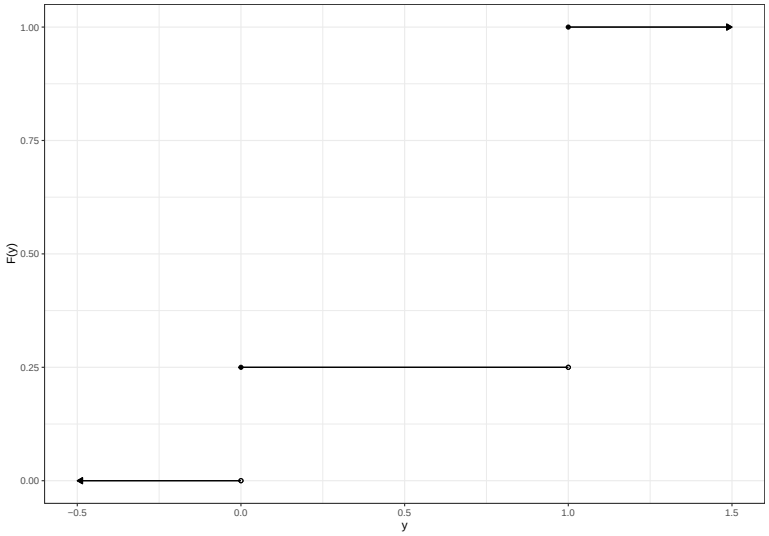
So, for continuous data, we could

- Simulate many replicates from our fitted model to approximate $F(Y)$
- Evaluate the approximate $F(Y)$ at y_i



Randomized quantile residuals for binary data

For binary data, $F(y)$ is a step function, with steps at 0 and 1.



Dunn and Smyth (1997 JCGS) defined the ***randomized quantile residual***.
For binary data, this is:

$$r_{q,i} = \begin{cases} u \sim U(0, \hat{\pi}_i) & \text{if } y_i = 0 \\ u \sim U(\hat{\pi}_i, 1) & \text{if } y_i = 1 \end{cases}$$

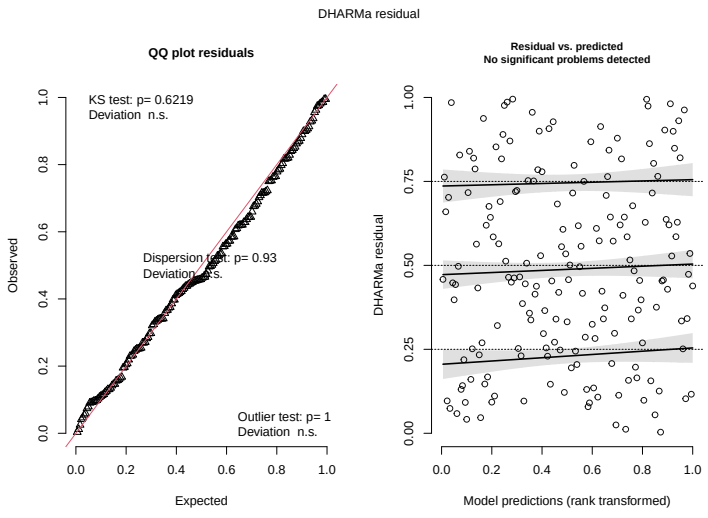
These residuals have nice properties (see, e.g., the DHARMa vignette)

Importantly, departures from a uniform distribution indicate lack of fit.

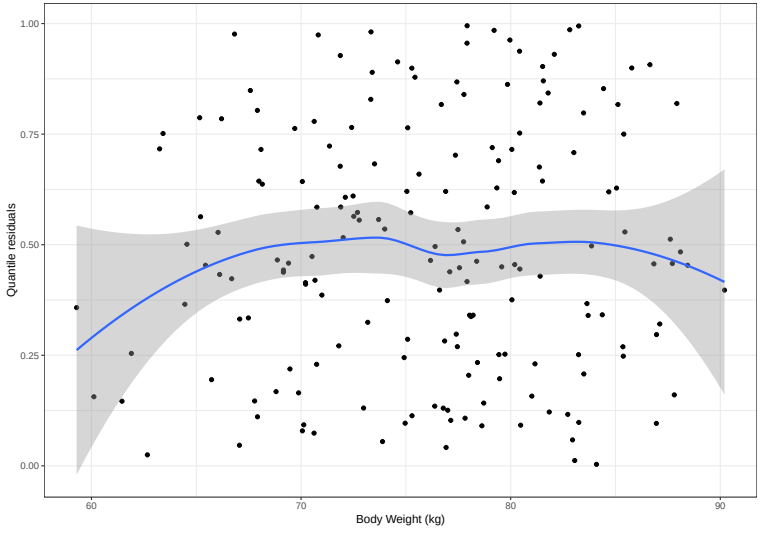
Randomized quantile residuals example

```
dharma_resids = simulateResiduals(mod01_glm,
                                  n = 1000,
                                  integerResponse = TRUE)

plot(dharma_resids)
```




```
aedat %>% ungroup() %>%
  mutate(quantile_residual = dharma_resids$scaledR
  ggplot(aes(x=BWT, y=quantile_residual)) +
  geom_point() +
  geom_smooth() +
  labs(y='Quantile residuals', x='Body Weight (kg)
```



Visual Predictive Checks

A general VPC "recipe":

- Simulate many replicates of the DV using the estimated model and observed predictors
- Determine summary statistic(s) of interest
- Calculate summary statistic for observed data
- Calculate summary statistic for each simulated replicate
- Plot distribution(s) of simulated summary statistics
- Overlay observed value

Simulate data for a VPC

```
# Simulate using the simulate function in stats
aedat_pp = bind_cols(aedat,
                     stats::simulate(mod01_glm, nsim=500)) %>%
  pivot_longer(cols=sim_1:sim_500)

aedat_pp %>% ungroup() %>%
  select(USUBJID, PTTYPE, AE01, Quartile, name, value) %>%
  slice_tail(n=4)
```

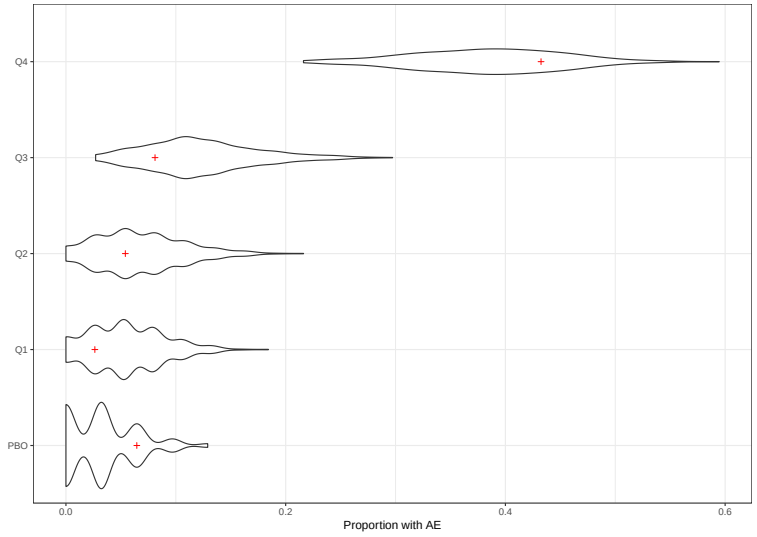
```
. # A tibble: 4 x 6
.   USUBJID PTTYPE  AE01 Quartile name    value
.   <fct>   <fct>  <int> <chr>    <chr>   <dbl>
. 1 UID-180 PT1        0 Q2       sim_497     0
. 2 UID-180 PT1        0 Q2       sim_498     0
. 3 UID-180 PT1        0 Q2       sim_499     0
. 4 UID-180 PT1        0 Q2       sim_500     0
```

Generate VPC for categorical predictor

```
# Observed data summary
obs_summary <- aedat %>%
  group_by(Quartile) %>%
  summarise(phat_obs = mean(AE01))

#Simulated data summary
sim_summary <- aedat_pp %>%
  group_by(name,Quartile) %>%
  summarise(phat_sim = mean(value))

# VPC
sim_summary %>%
  ggplot(aes(x=Quartile, y=phat_sim)) +
  geom_violin() +
  geom_point(data=obs_summary, aes(y=phat_obs), col='red',shape=3) +
  labs(x='', y='Proportion with AE') +
  coord_flip()
```



VPC for continuous variable: define summary statistic

- Summary statistic = non-parametric estimate of exposure-response relationship, evaluated at a fixed grid of values
  - Fit generalized additive model (smoother) to each simulated dataset
  - Predict at a fixed grid of values (5^{th} to 95^{th} percentile)

```
summary_function <- function(.data, .x_name, .y_name='value') {
  .data <- .data %>% ungroup() %>%
    rename('xvar' = all_of(.x_name),
           'yvar'= all_of(.y_name))
  x_grid <- with(.data, seq(from = quantile(xvar, probs=0.05),
                            to = quantile(xvar,probs = 0.95),
                            length = 100))
  fit <- gam(yvar ~ s(xvar), family=binomial(link='logit'), data=.data)
  predictions <- predict(fit, newdata = data.frame(xvar=x_grid), type='response')
  return( data.frame(xvar=x_grid, prediction = predictions))
}
```

Compute summary statistics for observed data

```
obs_summary <- summary_function(aedat, .x_name = 'CAVGSS', .y_name='AE01') %>%
  mutate(type='Observed')

head(obs_summary)
```

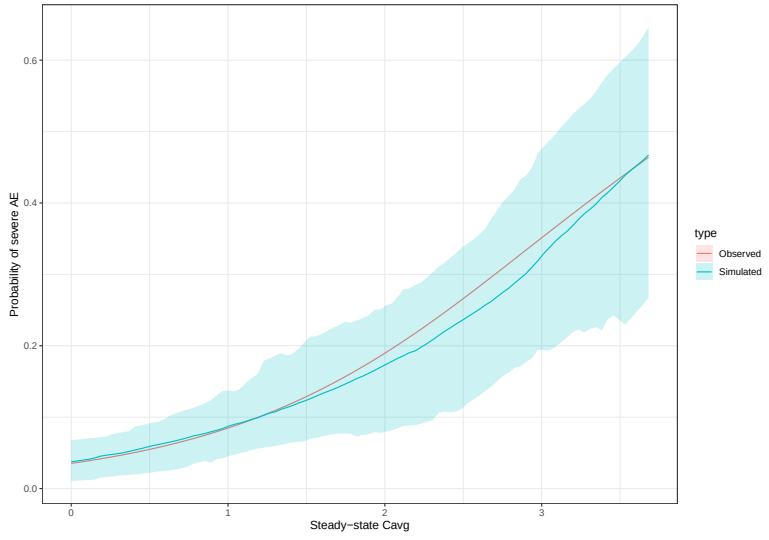
```
.         xvar prediction     type
. 1 0.00000000 0.03526032 Observed
. 2 0.03718346 0.03644987 Observed
. 3 0.07436693 0.03767799 Observed
. 4 0.11155039 0.03894582 Observed
. 5 0.14873385 0.04025454 Observed
. 6 0.18591731 0.04160537 Observed
```

Compute summary on each simulated study

```
sim_summary <- aedat_pp %>%
  # Nest everying except the simulation name
  nest(cols=-name) %>%
  # Use 200 sims for demonstration
  slice(1:200) %>%
  # Compute summary stats for each simulated dataset
  mutate(predictions = map(cols, ~summary_function(.x, .x_name='CAVGSS'))) %>%
  select(name, predictions) %>%
  unnest(cols=predictions) %>%
  # Summarise across simulated data sets
  group_by(xvar) %>%
  summarise(qlo = quantile(prediction, probs = 0.05),
            qhi = quantile(prediction, probs = 0.95),
            prediction=median(prediction)
            ) %>%
  mutate(type = 'Simulated')
```

Plot VPC

```
sim_summary %>% bind_rows(obs_summary) %>%
  ggplot(aes(x=xvar, y=prediction)) +
  geom_line(aes(col=type, group=type)) +
  geom_ribbon(aes(ymin=qlo, ymax=qhi, fill=type),
              alpha=0.2) +
  labs(x='Steady-state Cavg',
       y='Probability of severe AE')
```



Model Comparison

- Likelihood and information criteria
  - Likelihood (or -2 log-likelihood) measures in-sample model fit
  - Cross-validation to approximate out-of-sample deviance
  - IC approximate out-of-sample deviance
    - AIC $= -2 \times$ Log-likelihood $+ 2 \times k$
    - BIC $= -2 \times$ Log-likelihood $+ \log(N) \times k$
    - Lower is better
- Classification accuracy
  - Classification accuracy scores (e.g. sensitivity, specificity, Kappa)
  - Receiver operating characteristic curve (ROC) and its AUC

Workbook 05: Model evaluation and comparison

What we haven't covered

- Interactions
  - Does the exposure-response relationship depend on another covariate?
  - Aka, effect modification
- Non-linear effects
  - Parametric (e.g., Emax) models for binary data not well served by R
  - Semi-parametric models (e.g., generalized additive models) are using the `mgcv` package
- Forest plots for covariate effects
- Clinical trial simulation

Key learning objectives

- How to make effective exploratory analysis plots for binary data
  - We've seen plots for categorical and continuous predictors
- The basic concepts of maximum likelihood estimation for binary data models
- How to fit and evaluate binary data models
  - Using `glm` to fit models
  - Using residuals and VPCs to evaluate models
- How to interpret model terms
  - Parameters in logistic regression models inform us about odds ratios

Break

Next up: Bayesian analysis of binary data

References