

Generalized Linear Models

Objectives

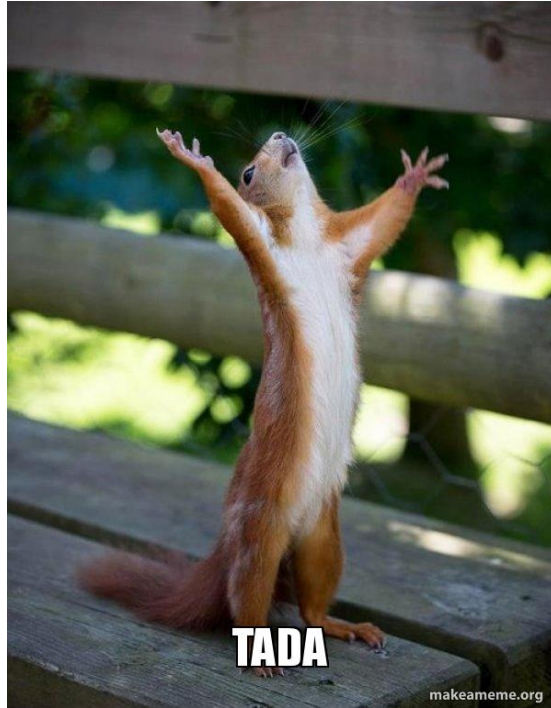
- GLM framework
- matrix notation
- linear algebra
- design matrix / categorical variable
- glm function
- link functions
- logistic regression

GLM

Generalized linear model framework using matrix notation

...

$$\mathbf{y} \sim [\mathbf{y} | \mu, \sigma]$$
$$g(\mu) = \mathbf{X}\beta$$



Motivation

GLMs:

- t-test
- ANOVA/ANCOVA
- linear regression
- logistic / probit regression
- Poisson regression
- log-linear regression
- survival analysis
- AND MORE!

GLM

Generalized linear model framework

$$\mathbf{y} \sim [\mathbf{y} | \mu, \sigma]$$
$$g(\mu) = \mathbf{X}\beta$$

Elements

- prob. function to define the RV ($\mathbf{y}$)
- parameters or moments of the prob. function (μ, σ)
- link function ($g(\mu)$); deterministic transformation of parameters to new scale

GLM

Generalized linear model framework

$$\mathbf{y} \sim [\mathbf{y} | \mu, \sigma]$$
$$g(\mu) = \mathbf{X}\beta$$

Elements

- inverse-link function ($g^{-1}(\mathbf{X}\beta)$); deterministic transformation of linear combination back to parameter scale
- design matrix of the known explanatory variables ($\mathbf{X}$)
- coefficient parameters (β); needs estimating

Linear Regression

index notation

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}$$

matrix notation

$$\mathbf{y} \sim \text{Normal}(\mu, \sigma)$$
$$\mu = g^{-1}(\mathbf{X}\beta)$$
$$g(\mu) = \mathbf{X}\beta$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{1,2} & x_{1,3} \\ 1 & x_{2,2} & x_{2,3} \\ 1 & x_{3,2} & x_{3,3} \\ \vdots & \vdots & \vdots \\ n & x_{n,2} & x_{n,3} \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

n = sample size x_2 & x_3 are independent variables

Linear Algebra

$$g(\mu) = \mathbf{X}\beta$$

$\mathbf{X}$ is called the Design Matrix.

β is a vector of coefficients.

...

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,2} & x_{1,3} \\ 1 & x_{2,2} & x_{2,3} \\ 1 & x_{3,2} & x_{3,3} \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

...

$$\mathbf{X}\beta = \begin{bmatrix} \beta_0 \times 1 + \beta_1 \times x_{1,2} + \beta_2 \times x_{1,3} \\ \beta_0 \times 1 + \beta_1 \times x_{2,2} + \beta_2 \times x_{2,3} \\ \beta_0 \times 1 + \beta_1 \times x_{3,2} + \beta_2 \times x_{3,3} \end{bmatrix}$$

...

$$\mathbf{X}\beta = \begin{bmatrix} \beta_0 + \beta_1 x_{1,2} + \beta_2 x_{1,3} \\ \beta_0 + \beta_1 x_{2,2} + \beta_2 x_{2,3} \\ \beta_0 + \beta_1 x_{3,2} + \beta_2 x_{3,3} \end{bmatrix}$$

$$g(\mu) = \mathbf{X}\beta = \begin{bmatrix} \text{lt}_1 \\ \text{lt}_2 \\ \text{lt}_3 \end{bmatrix}$$

lt = linear terms

...

$$\mu = g^{-1}(\mathbf{X}\beta) = \mathbf{X}\beta/1 = \begin{bmatrix} \text{lt}_1/1 \\ \text{lt}_2/1 \\ \text{lt}_3/1 \end{bmatrix}$$

...

$$\mu = \mathbf{X}\beta = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}$$

Linear Algebra

- $\mathbf{y}'\mathbf{y}$
- $\mathbf{y}' \cdot \mathbf{y}$

...

$$= \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

...

$$= [(1 \times 1) + (2 \times 2) + (3 \times 3)] = [14]$$

...

```
y <- matrix(c(1,2,3),nrow=3,ncol=1)
t(y)%*%y
```

```
      [,1]
[1,]    14
```

Linear Algebra

When can we do matrix multiplication?

...

```
first = t(y)
dim(first)
```

```
[1] 1 3
```

```
...
```

```
second = y
dim(second)
```

```
[1] 3 1
```

```
...
```

When this is true...

```
ncol(first)==nrow(second)
```

```
[1] TRUE
```

Linear Algebra

Note that

$$\mathbf{y}'\mathbf{y} \neq \mathbf{y}\mathbf{y}'$$

```
...
```

```
t(y)%*%y
```

```
      [,1]
[1,]    14
```

```
y%*%t(y)
```

```
      [,1] [,2] [,3]
[1,]     1     2     3
[2,]     2     4     6
[3,]     3     6     9
```

Elephant Linear Regression Example



Categorical Variable

	weight	sex	age.years
1	11488.991	Male	10
2	4105.442	Female	12
3	4299.308	Female	15

Elephant Linear Regression Example

Categorical Variable

```
model = glm(weight ~ sex + age.years,  
            data = dat,  
            family = gaussian(link = identity)  
            )  
summary(model)
```

```
Call:
glm(formula = weight ~ sex + age.years, family = gaussian(link = identity),
     data = dat)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2552.820	112.577	22.68	<2e-16 ***
sexMale	6828.965	87.891	77.70	<2e-16 ***
age.years	145.748	5.781	25.21	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 382352.1)

Null deviance: 2639772182 on 199 degrees of freedom
 Residual deviance: 75323364 on 197 degrees of freedom
 AIC: 3143.4

Number of Fisher Scoring iterations: 2

Elephant Linear Regression Example

Categorical Variable

x_2 as an indicator of sex, female (0) or male (1)

x_3 elephant age

$$\text{weight} \sim \text{Normal}(\mu, \sigma) \mu = \mathbf{X}\beta = \begin{bmatrix} \beta_0 + (\beta_1 \times 1) + (\beta_2 \times 10) \\ \beta_0 + (\beta_1 \times 0) + (\beta_2 \times 12) \\ \beta_0 + (\beta_1 \times 0) + (\beta_2 \times 15) \end{bmatrix}$$

$$\hat{\mu} = \mathbf{X}\hat{\beta} = \begin{bmatrix} 2552.82 + (6828.96 \times 1) + (145.74 \times 10) \\ 2552.82 + (6828.96 \times 0) + (145.74 \times 12) \\ 2552.82 + (6828.96 \times 0) + (145.74 \times 15) \end{bmatrix}$$

$$\hat{\mu} = \mathbf{X}\hat{\beta} = \begin{bmatrix} 2552.82 + 6828.96 + 1457.4 \\ 2552.82 + 0 + 1748.88 \\ 2552.82 + 0 + 2186.1 \end{bmatrix}$$

$$\hat{\mu} = \mathbf{X}\hat{\beta} = \begin{bmatrix} 10401.98 \\ 6779.52 \\ 5322.02 \end{bmatrix}$$

So, what does β_1 mean?

glm and design matrix

```
X = model.matrix(~ sex + age.years,
                 data = dat
                 )
head(X)
```

	(Intercept)	sexMale	age.years
1	1	1	10
2	1	0	12
3	1	0	15
4	1	0	12
5	1	1	20
6	1	1	4

```
glm(weight ~ 0 + X,
     data = dat
     )
```

Call: glm(formula = weight ~ 0 + X, data = dat)

Coefficients:

X(Intercept)	XsexMale	Xage.years
2552.8	6829.0	145.7

Degrees of Freedom: 200 Total (i.e. Null); 197 Residual

Null Deviance: 1.561e+10

Residual Deviance: 75320000 AIC: 3143

Sex variable is arranged by ‘Dummy Coding’

Closed Form Estimator

Ordinary Least Squares Estimator

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

X(Intercept)	XsexMale	Xage.years
2552.8203	6828.9645	145.7483

```
y = dat$weight  
c(solve(t(X)%*%X)%*%t(X)%*%y)
```

```
[1] 2552.8203 6828.9645 145.7483
```

Link functions

- $g(\mu) = \mathbf{X}\beta$
- $\mu = g^{-1}(\mathbf{X}\beta)$

...

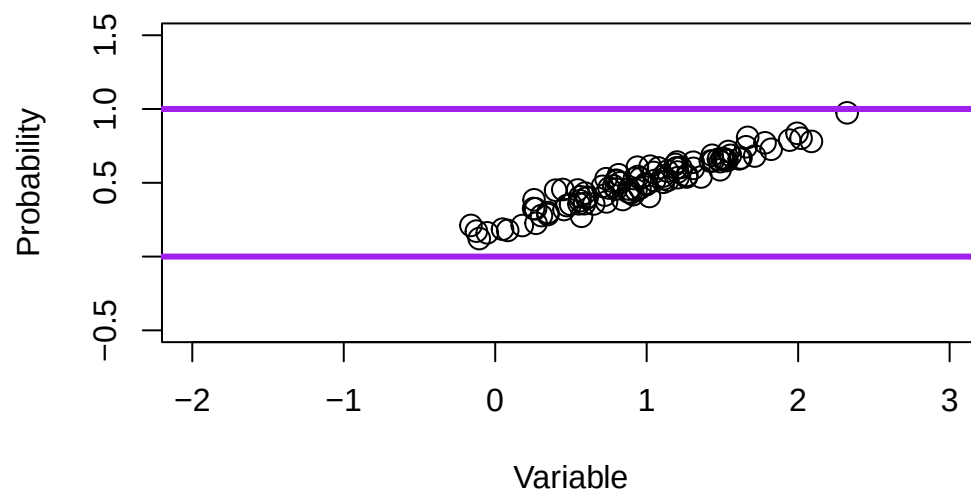
Link functions map parameters from one support to another.

...

Why is that important for us?

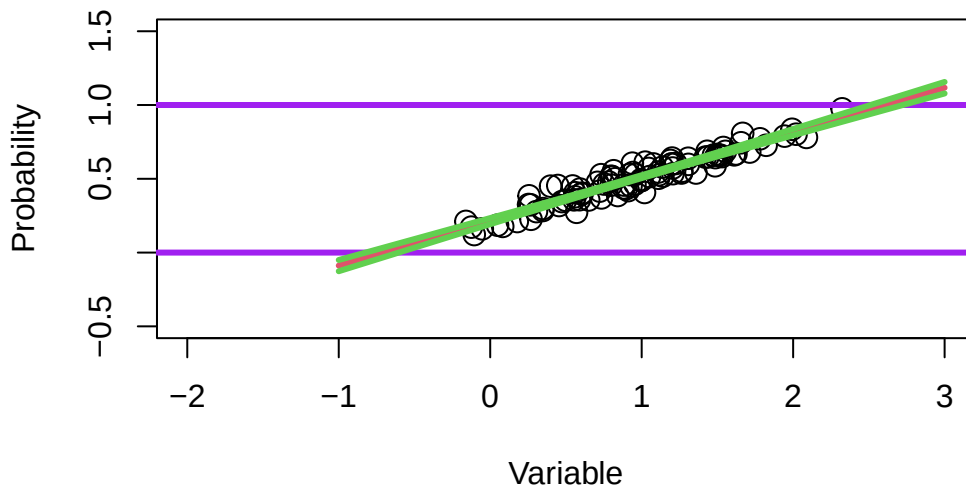
Link functions

To put a linear model on parameters of interest and ensure the parameter support is maintained.



Link functions

To put a linear model on parameters of interest and ensure the parameter support is maintained.



Link functions

Optimization algorithms do not like boundaries (e.g., 0 or 1). It's easier to **guess** values to evaluate in a maximum likelihood optimization when there are no bounds. $(-\infty, \infty)$

Link functions

Identity

No transformation is needed because the parameter support is maintained.

$$\mu \in (-\infty, \infty)$$

$$\mathbf{X}\beta \in (-\infty, \infty)$$

A Model by another name

Model Name	$[y \theta]$	Link
ANOVA (x_1 is categorical/multiple levels)	Normal	identity
ANCOVA (x_1 is categorical, x_2 is continuous))	Normal	identity
t-test (x_1 is categorical with 2 levels)	Normal	identity

Model Name	$[y \theta]$	Link
Linear Regression x_1 is continuous	Normal	identity
Multiple Linear Regression x_p is continuous	Normal	identity
Logistic Regression	Binomial	logit
Probit Regression	Binomial	probit
Log-linear Regression	Poisson	log
Poisson Regression	Poisson	log
Survival Analysis	Exponential	log
Inverse Polynomial	Gamma	Reciprocal

[Nelder and Wedderburn \(1972\)](#)

Logistic Regression (logit link)

$$\mathbf{y} \sim \text{Binomial}(N, p)$$

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

inverse-logit (expit)

$$p = g^{-1}(\mathbf{X}\beta) = \text{logit}^{-1}(\mathbf{X}\beta) = \frac{e^{\mathbf{X}\beta}}{e^{\mathbf{X}\beta} + 1}$$

Logistic Regression

Full model

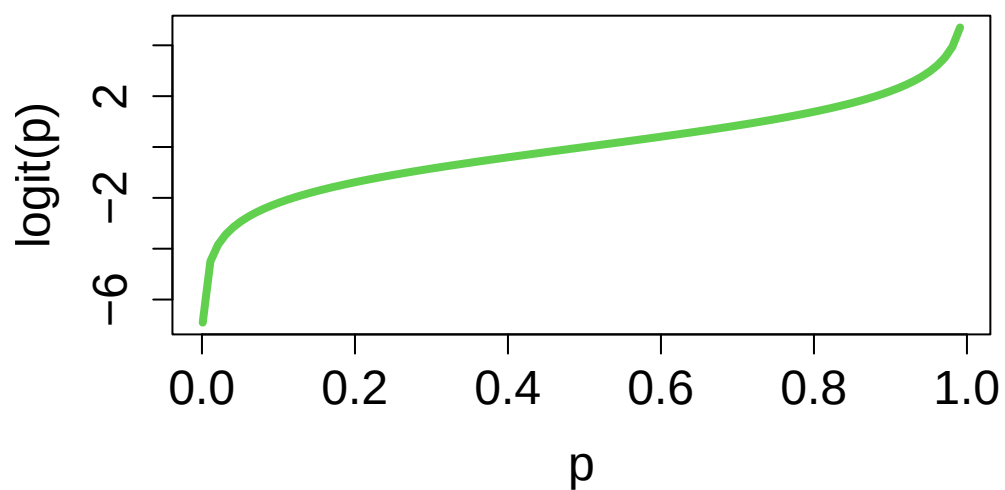
$$\begin{aligned}\mathbf{y} &\sim \text{Binomial}(N, p) \\ \text{logit}(p) &= \mathbf{X}\beta\end{aligned}$$

Remember, $p \in [0, 1]$

$$\mathbf{X}\beta \in (-\infty, \infty)$$

logit/p mapping

```
p = seq(0.001,0.999,  
        by = 0.01  
      )  
logit.p = qlogis(p)  
par(cex.lab=1.5,cex.axis=1.5)  
plot(p,logit.p,type="l",lwd=4,col=3,xlab='p',ylab="logit(p)")
```



Logistic Regression Simulation

...

```
# Design matrix  
set.seed(43534)  
Var1 = rnorm(100)  
X = model.matrix(~Var1)  
head(X)
```

(Intercept) Var1

```

1      1  1.4371396
2      1 -0.4835090
3      1  1.0370452
4      1 -0.5894099
5      1  1.2633564
6      1 -1.9008071

```

Logistic Regression Simulation

```

# Marginal coefficients (on logit-scale)
beta = c(-2,4)

# Linear terms
lt = X%*%beta

```

...

```

# Transformation via link function to probability scale
p = plogis(lt)
head(round(p,digits=2))

```

```

[,1]
1 0.98
2 0.02
3 0.90
4 0.01
5 0.95
6 0.00

```

...

```

# Generate a sample dataset
set.seed(14353)
y = rbinom(n=length(p),size=1,p)
y

```

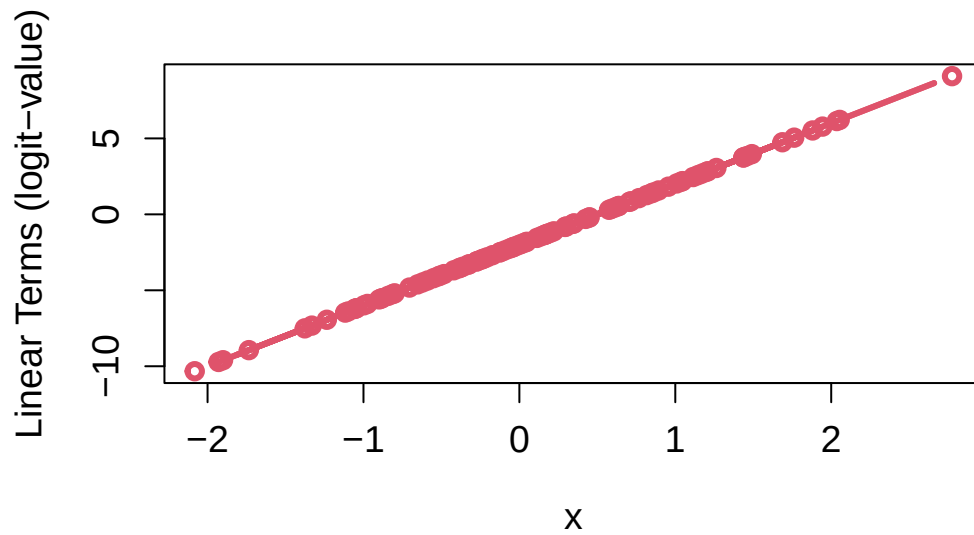
```

[1] 1 0 1 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
[38] 0 1 1 0 1 0 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
[75] 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0

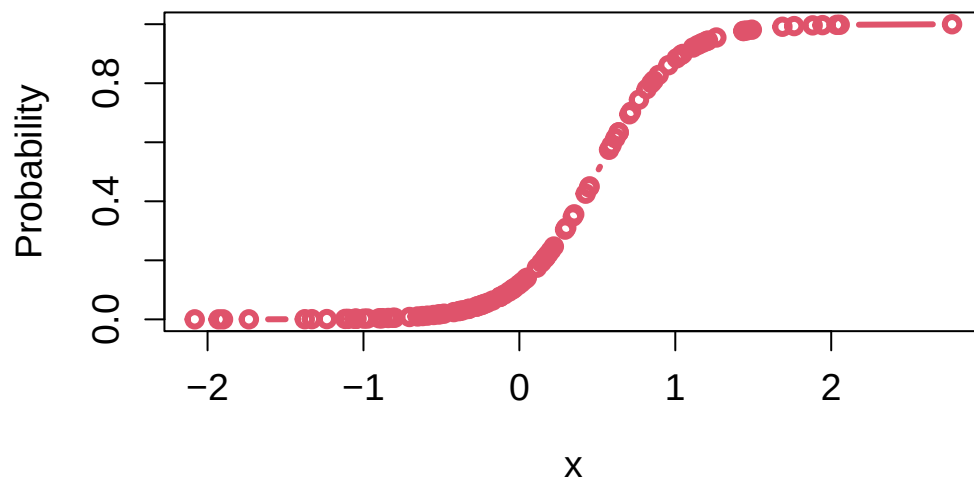
```

Logistic Regression Simulation

```
# Plot the linear 'terms' and explanatory variable  
par(cex.lab=1.2,cex.axis=1.2)  
plot(Var1,lt,type="b",lwd=3,col=2,  
      xlab="x",ylab="Linear Terms (logit-value)")
```



```
index=order(Var1)  
plot(Var1[index],p[index],type="b",lwd=3,col=2,xlab="x",ylab="Probability")
```



Logistic Regression Estimation

```
# Fit model to sample
model1 = glm(y ~ 0 + X,
             family = binomial(link = "logit")
             )
summary(model1)
```

Call:

```
glm(formula = y ~ 0 + X, family = binomial(link = "logit"))
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
X(Intercept)	-2.5304	0.6473	-3.909	9.25e-05	***
XVar1	5.4198	1.2232	4.431	9.38e-06	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 138.63 on 100 degrees of freedom
Residual deviance: 39.14 on 98 degrees of freedom
AIC: 43.14

Number of Fisher Scoring iterations: 7

Logistic Regression Evaluation

```
# Sample many times
n.sim = 1000
y.many = replicate(n.sim,
                    rbinom(n = length(p),
                           size = 1,
                           p)
                    )
dim(y.many)
```

```
[1] 100 1000
```

```
...
```

```
# Estimate coefs for all 100 samples
coef.est = apply(y.many, 2, FUN = function(y){
  model1 = glm(y ~ 0 + X,
               family = binomial(link = "logit")
               )

  model1$coefficients
})
```

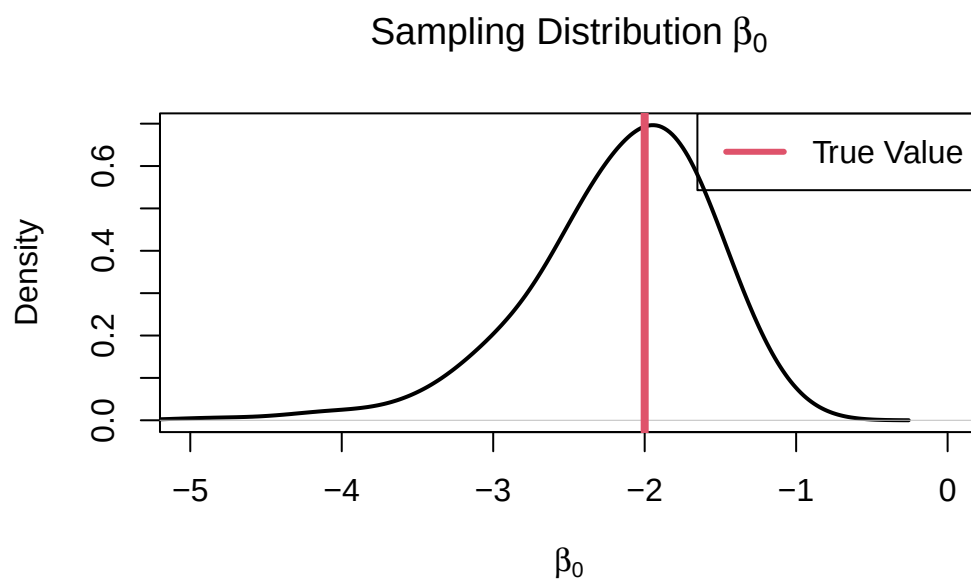
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

```
dim(coef.est)
```

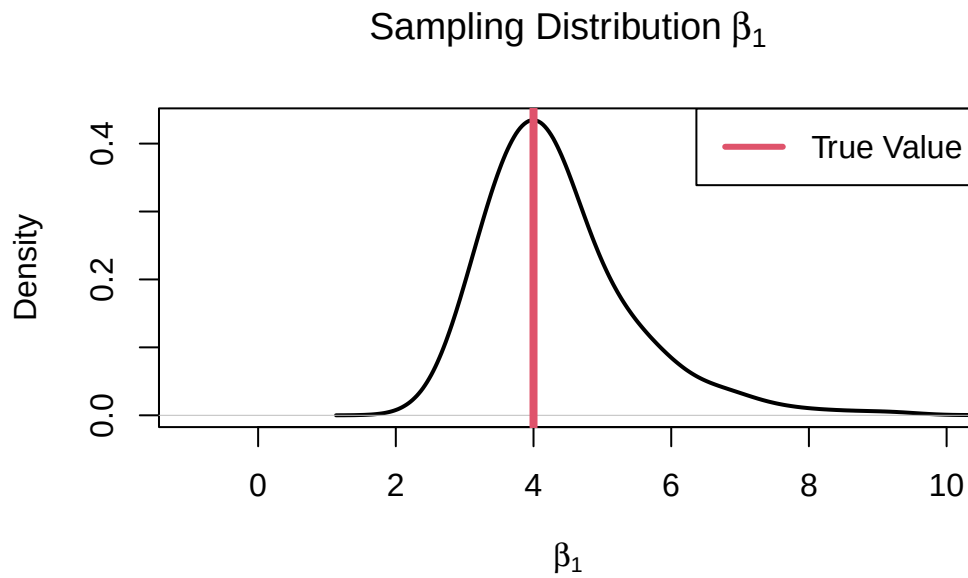
```
[1] 2 1000
```

```
...
```

Plot Intercept



Plot Slope



Evaluate Sampling Distributions

Is our model biased?

```
# Relative Bias  
(mean(coef.est[1,]) - beta[1]) / beta[1]
```

```
[1] 0.0879452
```

```
(mean(coef.est[2,]) - beta[2]) / beta[2]
```

```
[1] 0.09751027
```

Evaluate Sampling Distributions

Are we going to estimate the sign of the coef correctly?


```
# Probability of estimating a coef sign correctly
length(which(coef.est[1,]<0))/n.sim
```

```
[1] 1
```

```
length(which(coef.est[2,]>0))/n.sim
```

```
[1] 1
```

Logistic Regression Biased Correction

brglm: Bias Reduction in Binomial-Response Generalized Linear Models

```
brglm::brglm(y ~ 0 + X,
             family = binomial(link = "logit")
             )
```

```
# Relative Bias
(mean(coef.est[1,])-beta[1])/beta[1]
```

```
[1] 0.007785692
```

```
(mean(coef.est[2,])-beta[2])/beta[2]
```

```
[1] 0.008799186
```

Recap

- GLM framework: What are key components?
- matrix notation - why learn this?
- link functions - what are they?