# Useful Templates

Joy Kumagai

2025-02-13

## Plotting Prior Predictive Distributions

Often during class, I've been confused on how to plot prior predictive distributions. I've made this template as a reference for myself and others if you also are lost at this step in the process.

```
# Set number of draws from the distributions you will use
n <- 50

# Set your priors for alpha, and the parameter you are most interested in seeing the effect of
alpha <- rnorm(n, 0, .2)
beta <- rnorm(n, 0, .2)
```

We want to create a plot which visualizes the effects the priors have on our response variable. Therefore, the y axis is the response variable, and the x axis is our predictor we are most interested in.

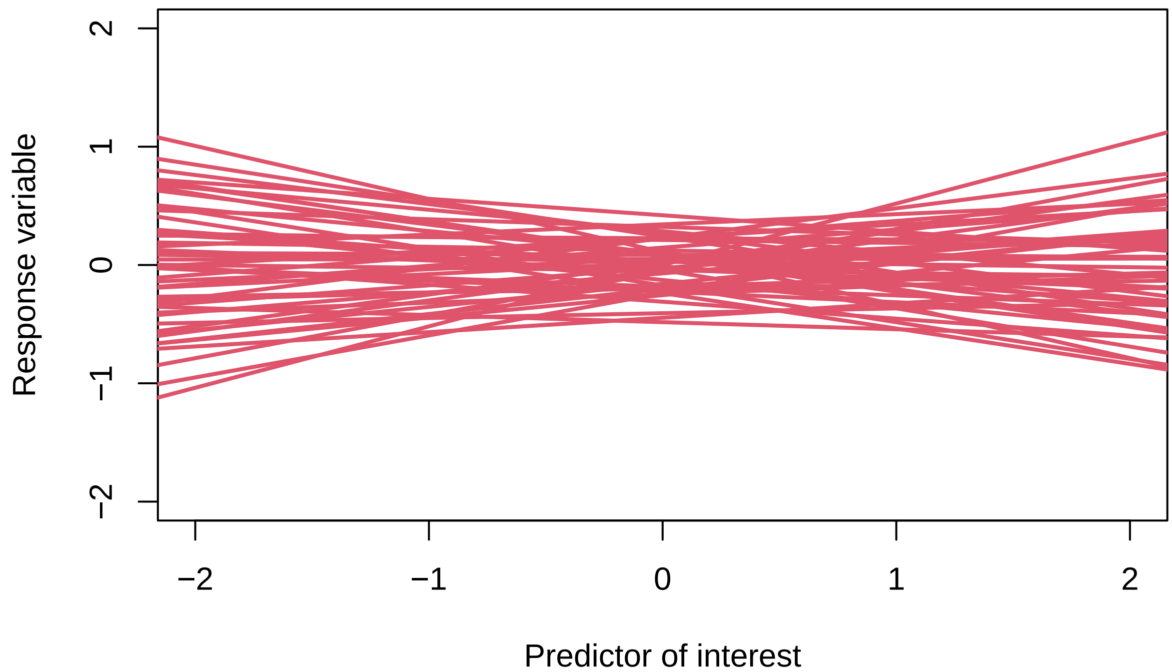
Create a plot with the x axis set by xlim and with the y axis set by ylim. Tip: If you are plotting a poisson relationship, ylim could be set to 0, as the response variable should never be negative.

The next line after setting up the for loop is the deterministic equation. For a normal distribution, this would be what mu is equal to. For example, mu = alpha + beta*X, so this line of code is equal to abline(alpha + betaX).

For a poisson distribution, the deterministic equation is lambda = exponential(alpha + beta*X). Therefore this line needs to specify a curve equal to exp(alpha + beta*X)).

Normal distribution example

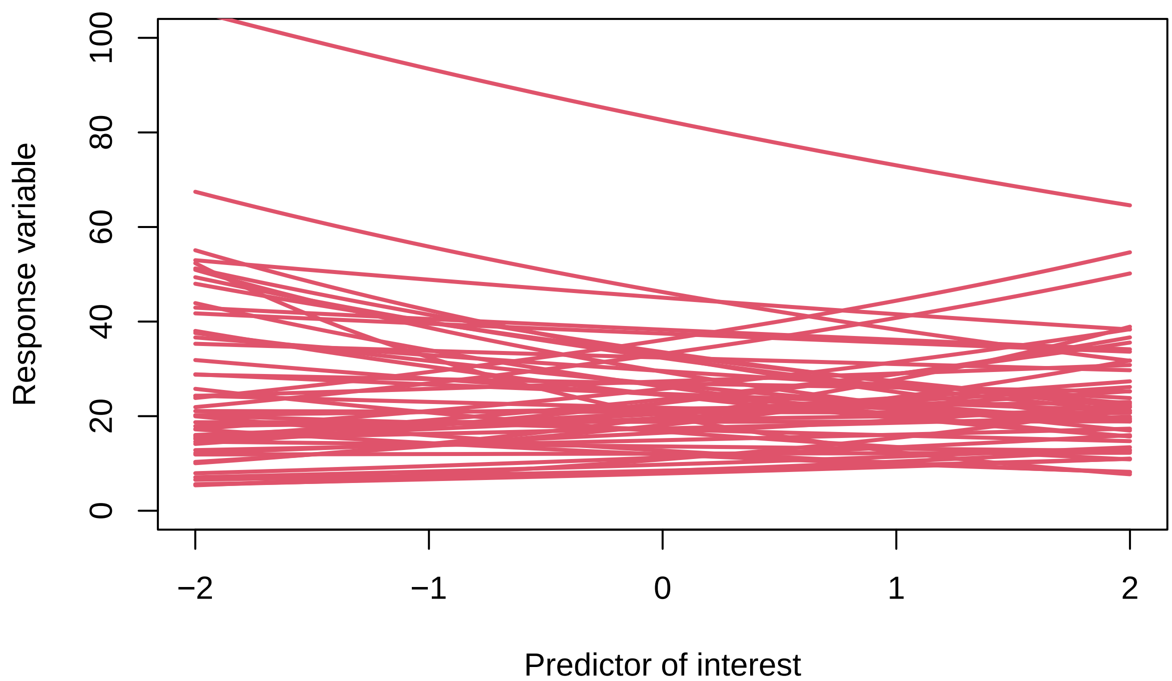
```
plot( NULL , xlim=c(-2,2) , ylim=c(-2,2) ,
      xlab="Predictor of interest" , ylab="Response variable" )
for ( j in 1:n ) {
  abline( a=alpha[j] , b=beta[j] ,                 # The deterministic equation
          lwd=2 ,                                  # width of line is 2
          col=2 )                                  # color is red
  }
```




```
# n is the number of lines that will be drawn
```

Poisson Distribution example

```
n <- 50
alpha <- rnorm(n, 3, 0.5)
beta <- rnorm(n, 0, 0.2)

# Ylim is higher here because Poission is modeling counts, not means!
plot( NULL , xlim=c(-2,2) , ylim=c(0,100) ,
      xlab="Predictor of interest" , ylab="Response variable" )
for ( j in 1:n ) {
  curve( exp(alpha[j] + beta[j]*x),
         # ^ The deterministic equation, uses x values from xlim, I think?
         add = TRUE,
         lwd=2 ,                                     # width of line is 2
          col=2 )                                     # color is red
  }
```

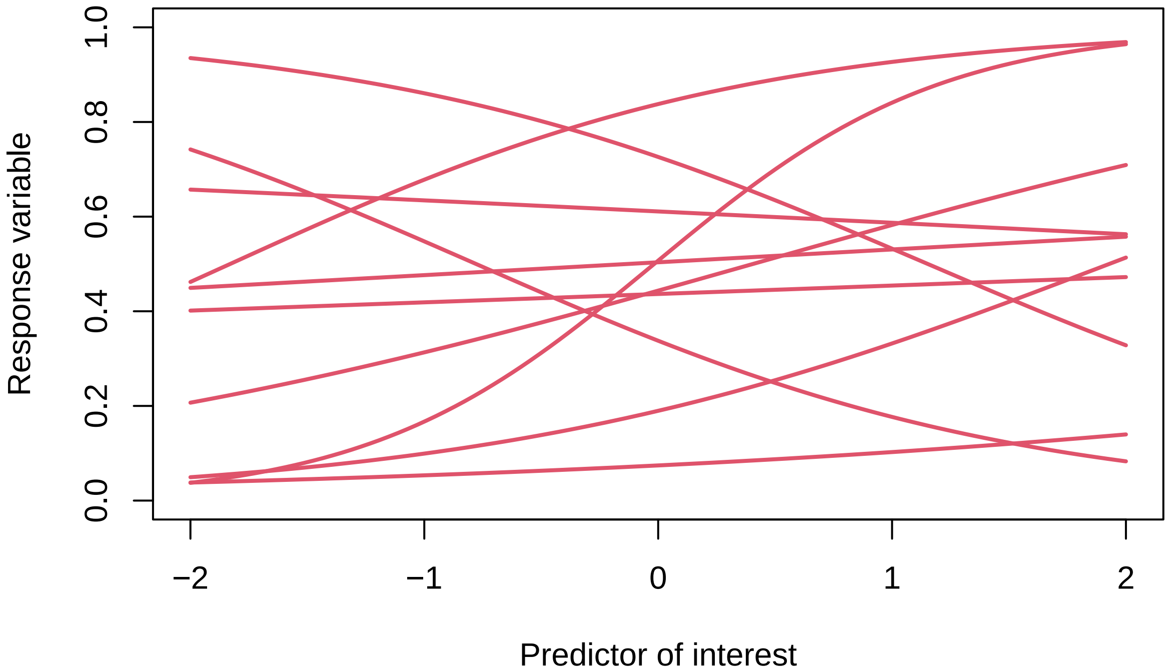



Bernoulli distribution example:

A key question: What do we expect the form of the lines on the plot to be? Straight, exponential, logistic?

```
n <- 10
alpha <- rnorm(n, 0, 1)
beta <- rnorm(n, 0, 1)

# We would expect a logistic relationship, thus we can use the lines()
# function to plot in R, which requires we specify the x and y values

# To do this, create a sequence of x values, and calculate the response
# given our relationship

xseq <- seq(-2, 2, len = 100)
p <- sapply(xseq, function(x) inv_logit(alpha + beta*x))

# Each value of our response is calculated above for a sequence of x values
# and the pairs of alpha and beta. One column corresponds to one alpha and
# beta set.

plot( NULL , xlim=c(-2,2) , ylim=c(0,1) ,
      xlab="Predictor of interest" , ylab="Response variable" )
for ( j in 1:n ) {
  lines(xseq, p[j,], # Plot the lines given the sequence of x values, and the
                     # corresponding sequence of response points.
         lwd=2 ,
```

```
            col=2 )
}
```




Tip: Abline refers to a straight line, lines draws a line that connects the points you provide and can curve.