Types of model outcomes

NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R

Noam Ross

Senior Research Scientist, EcoHealth Alliance

Types of outcomes

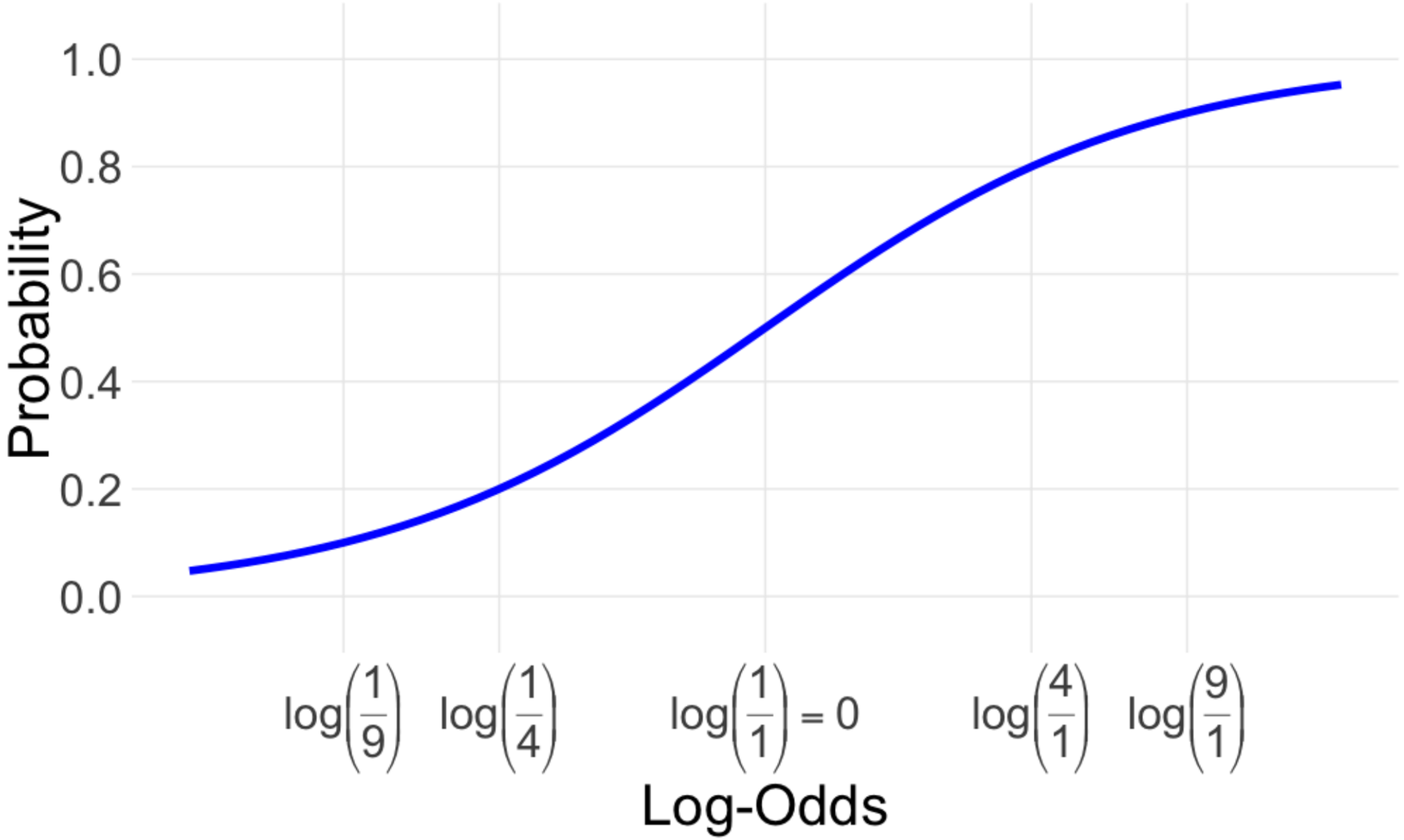
Continuous outcomes

- Speed of a motorcycle (mph)
- Fuel efficiency of a car (mpg)
- Level of pollution in soil (g/kg)

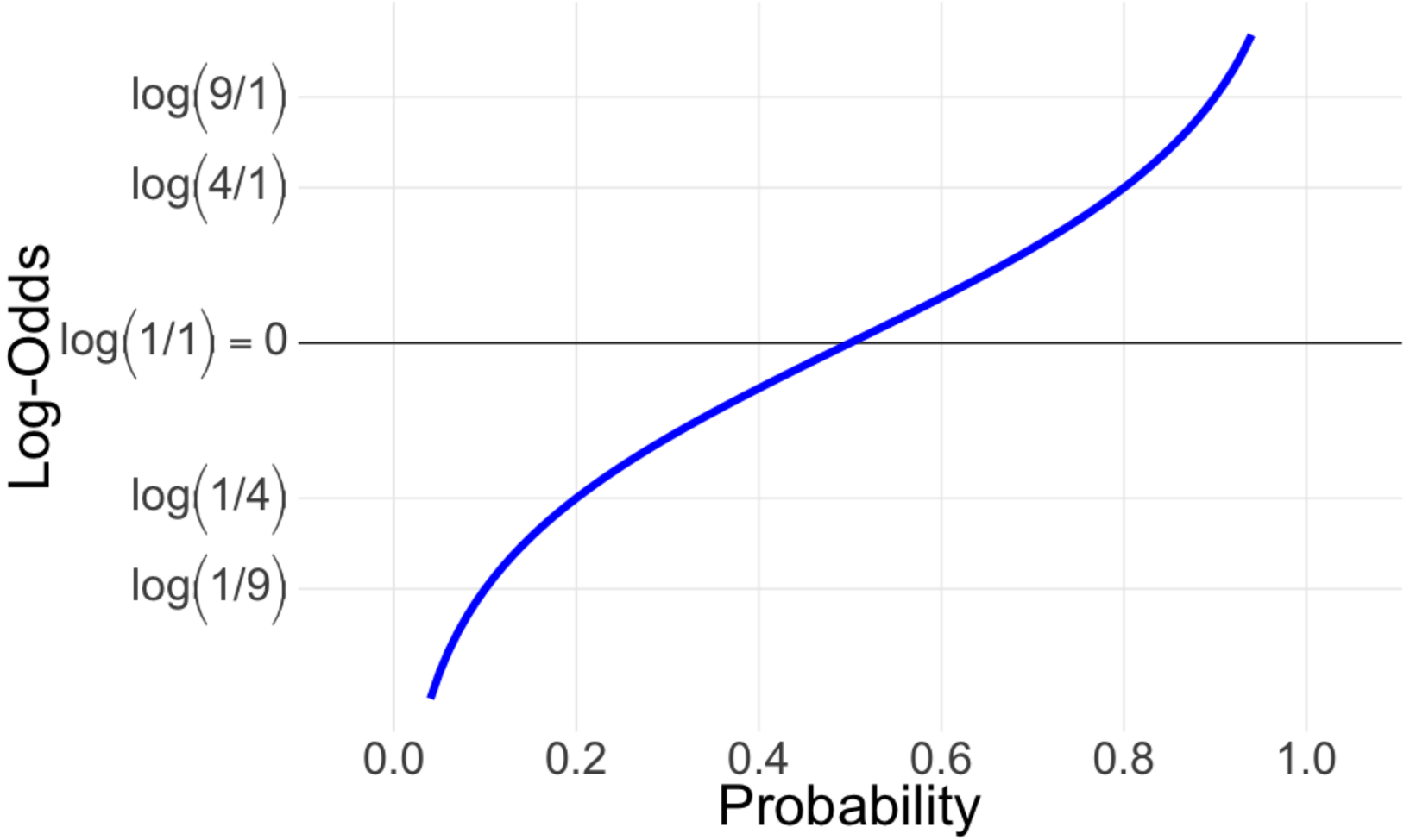
Binary outcomes

- Presence or absence of an organism in a location
- Whether a purchase was made
- Yes/No answer on a survey

Probabilities and log-odds: logistic function



Probabilities and log-odds: logit function



Logistic and logit functions in R

```
plogis() # Logistic
qlogis() # Logit


qlogis(plogis(0.5))
```

```
0.5
```

```
qlogis(0.25) == log(1/3)
```

```
TRUE
```

Logistic GAMs with mgcv

```
gam(y ~ x1 + s(x2),
    data = dat,
    family = binomial,
    method = "REML")
```

```
Family: binomial
Link function: logit
Formula:
y ~ s(x1) + s(x2)
Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.7330     0.1208    6.07 1.28e-09 ***
 ---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
        edf Ref.df Chi.sq  p-value
s(x1) 1.367  1.646  25.83 1.23e-05 ***
s(x2) 5.754  6.890  51.37 8.12e-09 ***
```

```
plogis(0.733)
```

```
0.6754633
```

```
head(csale)
```

```
  purchase n_acts bal_crdt_ratio avg_prem_balance retail_crdt_ratio
1        0     11        0.00000         2494.414           0.00000
2        0      0       36.09506         2494.414          11.49123
3        0      6       17.60000         2494.414           0.00000
4        0      8       12.50000         2494.414           0.80000
5        0      8       59.10000         2494.414          20.80000
6        0      1       90.10000         2494.414          11.49123
  avg_fin_balance mortgage_age cred_limit
1        1767.197     182.0000      12500
2        1767.197     138.9601          0
3           0.000     138.9601          0
4        1021.000     138.9601          0
5         797.000      93.0000          0
6        4953.000     138.9601          0
```

Let's practice!

NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R

datacamp

Visualizing Logistic GAMs

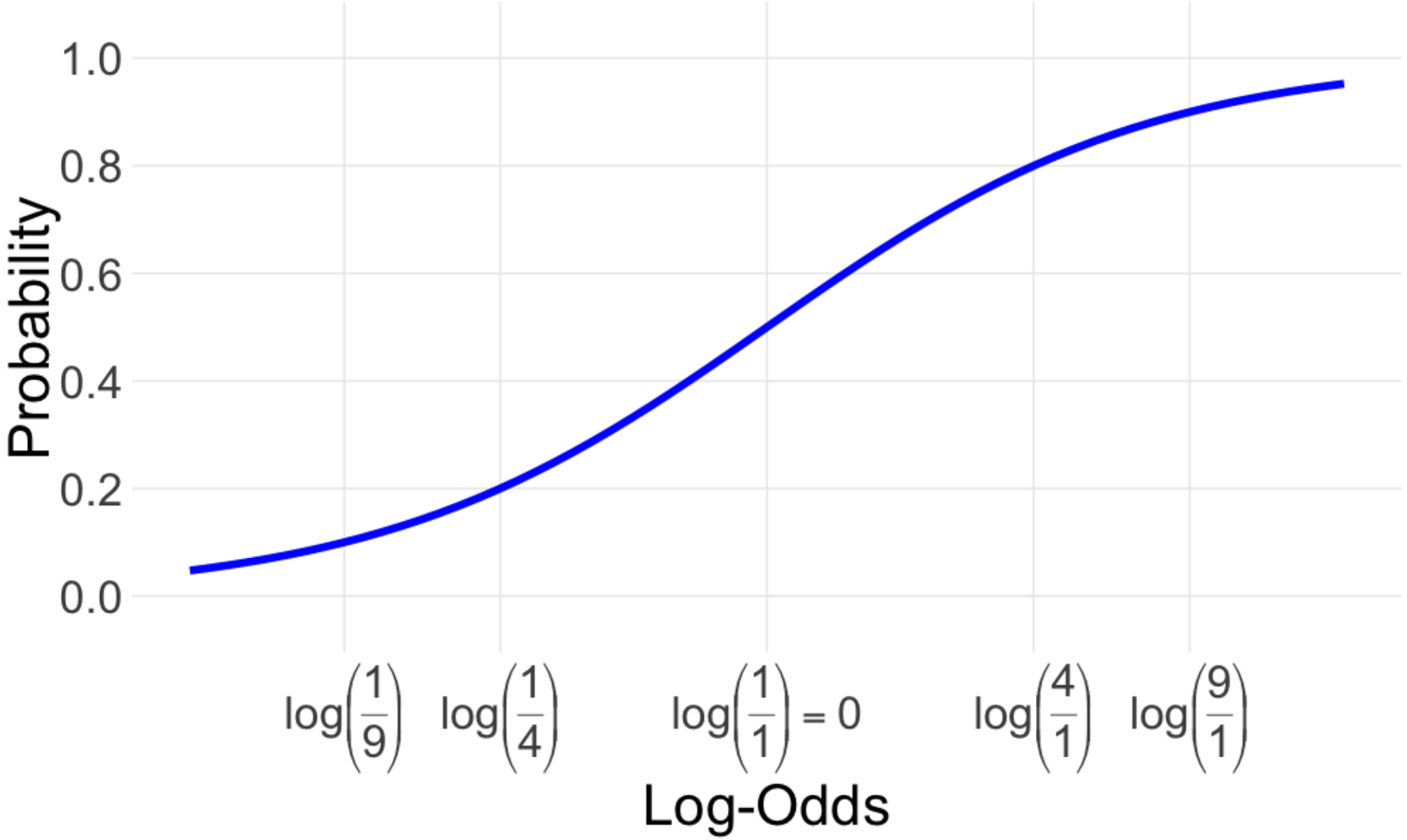
NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R

Noam Ross

Senior Research Scientist, EcoHealth Alliance

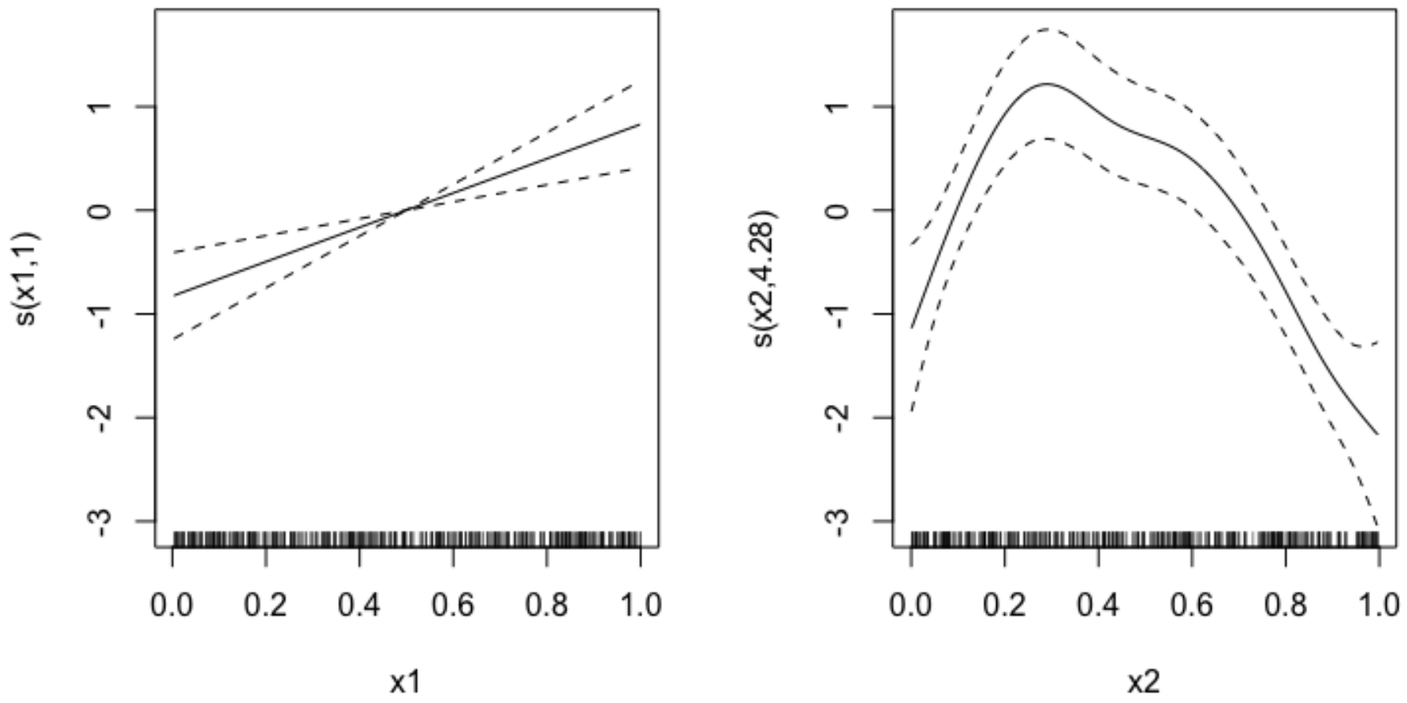
datacamp

Transforming scales



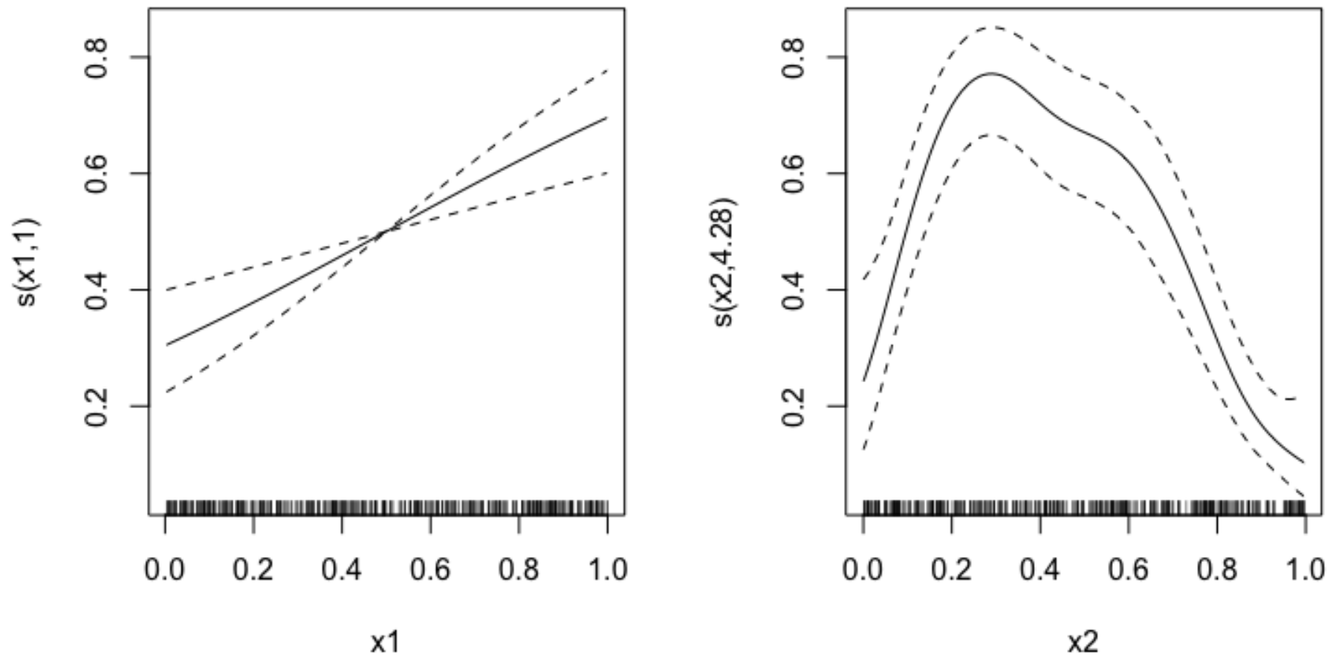
Log-odds plots

```
plot(binom_mod)
```



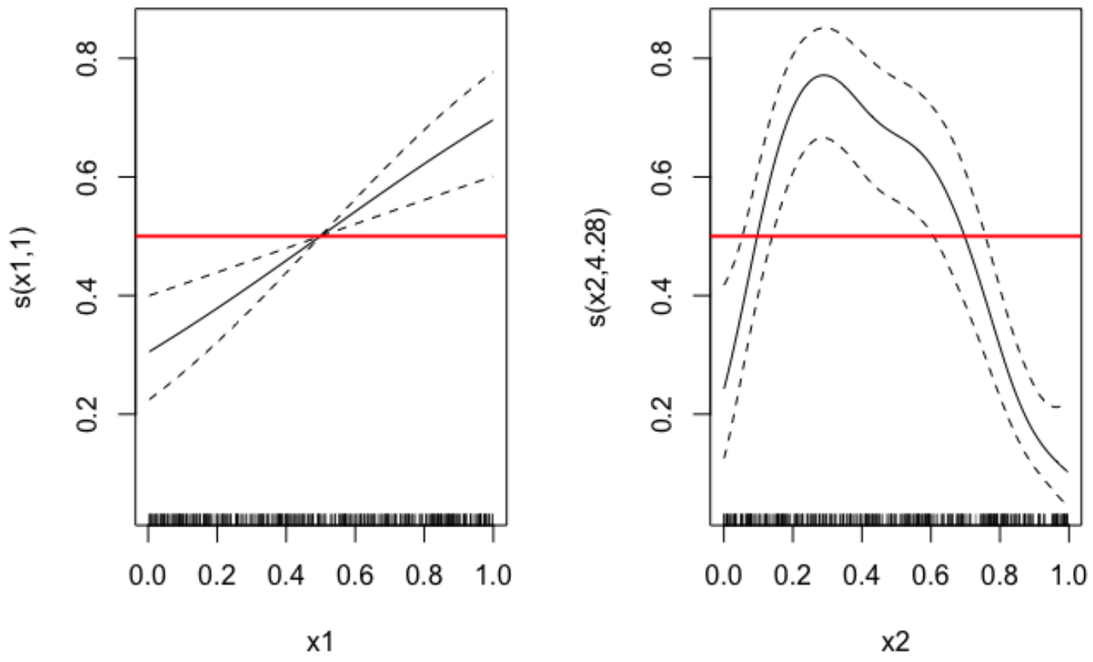
Converting partial effects

```
plot(binom_mod, pages = 1, trans = plogis)
```



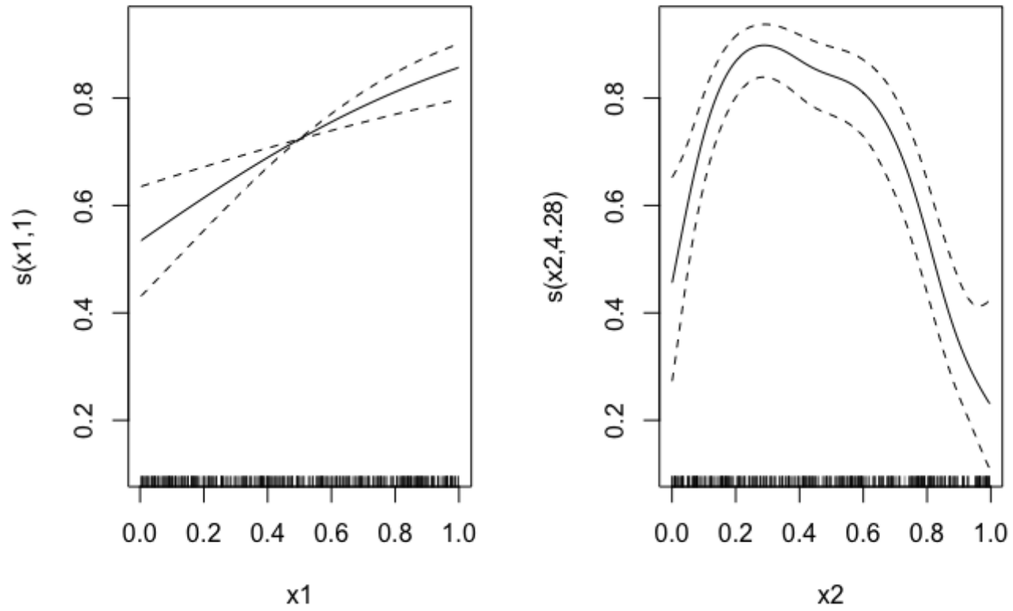
Converting partial effects (2)

```
plot(binom_mod, pages = 1, trans = plogis)
```



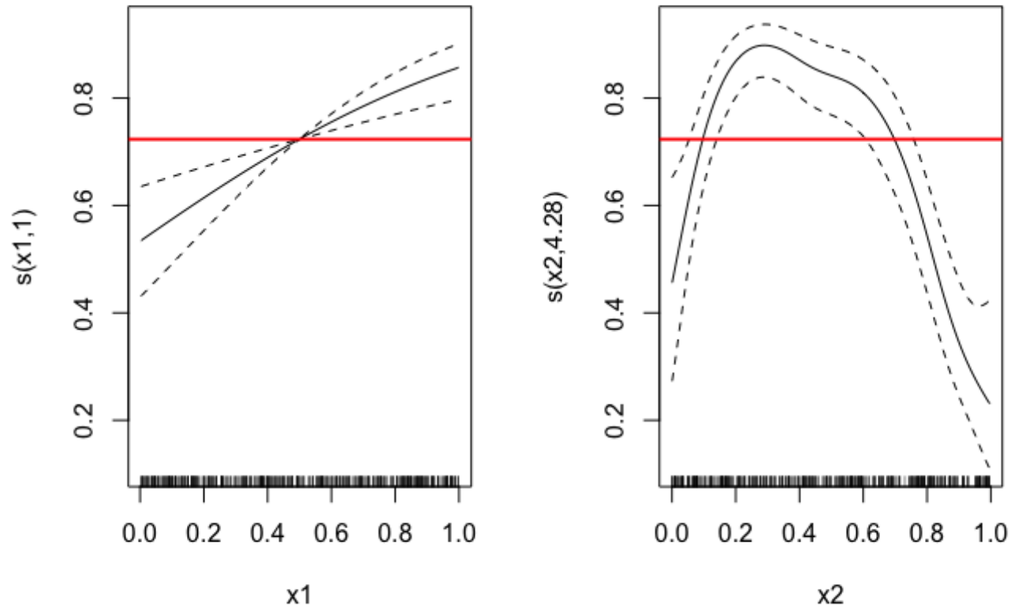
Adding an intercept

```
plot(binom_mod, pages = 1, trans = plogis,
     shift = coef(binom_mod)[1])
```



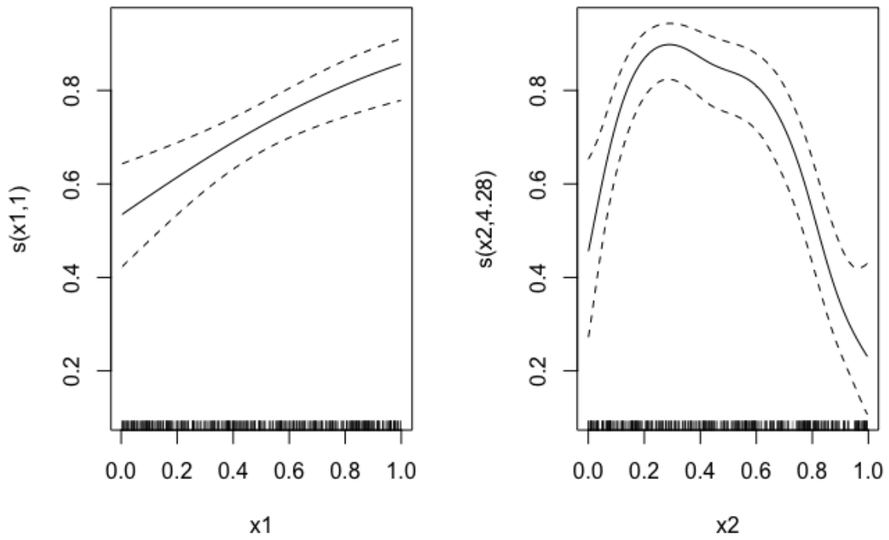
Adding an intercept (2)

```
plot(binom_mod, pages = 1, trans = plogis,
     shift = coef(binom_mod)[1])
```



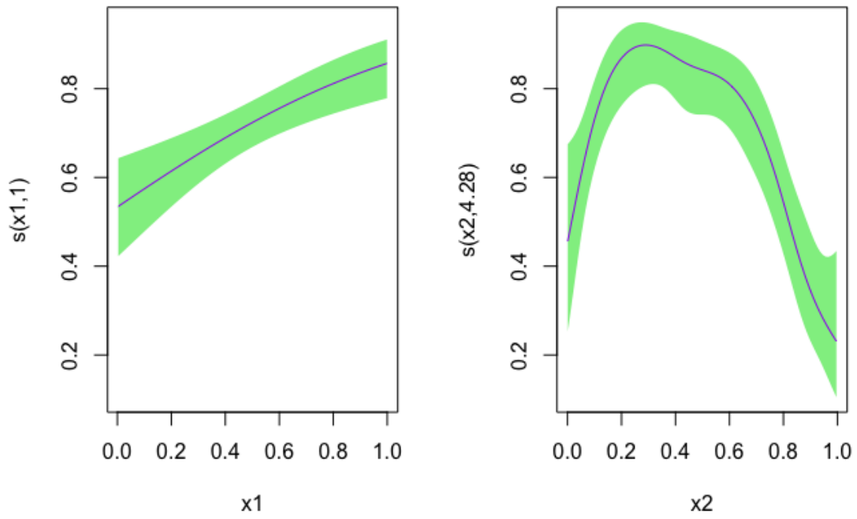
Incorporating intercept uncertainty

```
plot(binom_mod, pages = 1, trans = plogis,
     shift = coef(binom_mod)[1],
     seWithMean = TRUE)
```



Improving the plot

```
plot(binom_mod, pages = 1, trans = plogis, shift = coef(binom_mod)[1],
     seWithMean = TRUE, rug = FALSE, shade = TRUE,
     shade.col = "lightgreen", col = "purple")
```



Let's practice!

NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R

datacamp

Making predictions

NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R

Noam Ross

Senior Research Scientist, EcoHealth Alliance

mgcv's predict() function

```
predict(log_mod2)
```

```
           1            2            3            4
-0.8672827973 -2.9135420237 -0.4839780158 -0.1996086132
           5            6            7            8
-0.4416783066 -1.2351679544 -0.6148559122 -2.9135420237
...
```

Prediction types

```
predict(log_mod2, type = "link")
```

```
           1            2            3            4
-0.8672827973 -2.9135420237 -0.4839780158 -0.1996086132
           5            6            7            8
-0.4416783066 -1.2351679544 -0.6148559122 -2.9135420237
...
```

```
predict(log_mod2, type="response")
```

```
         1          2          3          4
0.29582001 0.05148818 0.38131322 0.45026288
         5          6          7          8
0.39134114 0.22527819 0.35095230 0.05148818
...
```

```
plogis(predict(log_mod2, type="link"))
```

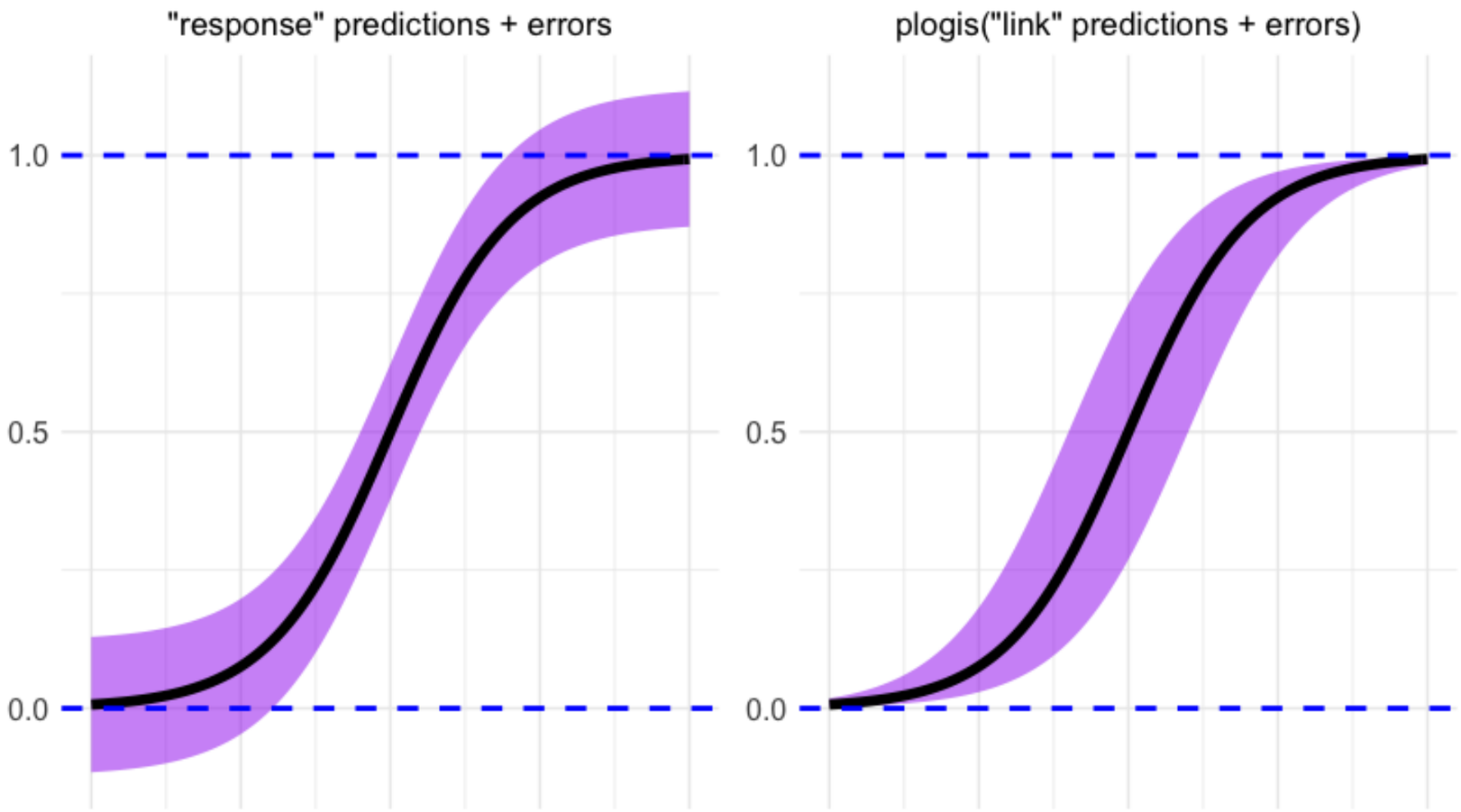
Standard errors

```
predict(log_mod2, type = "link", se.fit = TRUE)
```

```
$fit
         1          2          3          4
-0.8672828 -2.9135420 -0.4839780 -0.1996086
         5          6          7          8
-0.4416783 -1.2351680 -0.6148559 -2.9135420

$se.fit
         1          2          3          4
0.2850848 0.1646090 0.2299404 0.2159088
         5          6          7          8
0.2767443 0.7601131 0.2454877 0.1646090
```

Standard errors (2)



Predictions on new data

```
trained_model <- gam(response ~ s(predictor),
                     data = train_df,
                     family = binomial,
                     method = "REML")
```

```
# Test data
test_predictions <- predict(trained_model,
                            type = "response",
                            newdata = test_df)
```

Explaining predictions by terms

```
predict(log_mod2, type = "terms")
```

```
      s(n_acts) s(bal_crdt_ratio) s(avg_prem_balance) ...
1     1.2115213       0.3327855673         -0.135920526 ...
2    -0.8850186      -0.4058818961         -0.135920526 ...
3     0.5693622       0.2972364048         -0.135920526 ...
4     0.8974704       0.3827671103         -0.135920526 ...
5     0.8974704      -0.0727464938         -0.135920526 ...
6    -0.6228781       0.1936974771         -0.135920526 ...
7     0.3642246       0.3377181800         -0.135920526 ...
8    -0.8850186      -0.4058818961         -0.135920526 ...
9     1.0209905       0.3604064595          0.317309246 ...
10    1.7675666      -0.4533384774          0.346837355 ...
```

Explaining predictions by terms (2)

```
predict(log_mod2, type = "terms")[1, ]
```

```
          s(n_acts)    s(bal_crdt_ratio)
         1.21152126           0.33278557
 s(avg_prem_balance) s(retail_crdt_ratio)
        -0.13592053           0.06789949
  s(avg_fin_balance)      s(mortgage_age)
        -0.04057249          -0.29183903
       s(cred_limit)
        -0.37055621
```

```
plogis(sum(predict(log_mod2, type = "terms")[1, ]) + coef(log_mod2)[1])
```

```
0.29582
```

Let's practice!

NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R

datacamp

Doing more with GAMs

NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R

Noam Ross

Senior Research Scientist, EcoHealth Alliance

datacamp

Chapter 1

- GAM theory
- Fitting GAMs
- Mixing linear and nonlinear terms

Chapter 2

- Interpreting GAMs
- Visualizing GAMs
- Model-checking and concurvity

Chapter 3

- 2-D Interactions and spatial data
- Interactions with different scales
- Continuous-categorical interaction

Chapter 4

- Logistic GAMs
- Plotting logistic outputs
- Making predictions

GAMs and the Tidyverse

```
library(broom)


augment(gam_model)
tidy(gam_model)
glance(gam_model)
```

```
library(caret)


train(x, y, method = "gam", ...)
```

Other types of smooths

```
?smooth.terms
```

Other types of outcomes/distributions

```
?family.mgcv
```

See **Generalized Linear Models**

Variable selection

```
?gam.selection
```

Complex model structures

```
?gam.models
```

See **Hierarchical and Mixed Effects Models**

Thank You!

NONLINEAR MODELING WITH GENERALIZED ADDITIVE MODELS (GAMS) IN R