# Today

- Binomial GLM

# Binomial GLM

- Swiss breeding bird survey (www.vogelwarte.ch)
- Skilled observers, 1 km$^2$ cells

Switzerland; showing survey locations

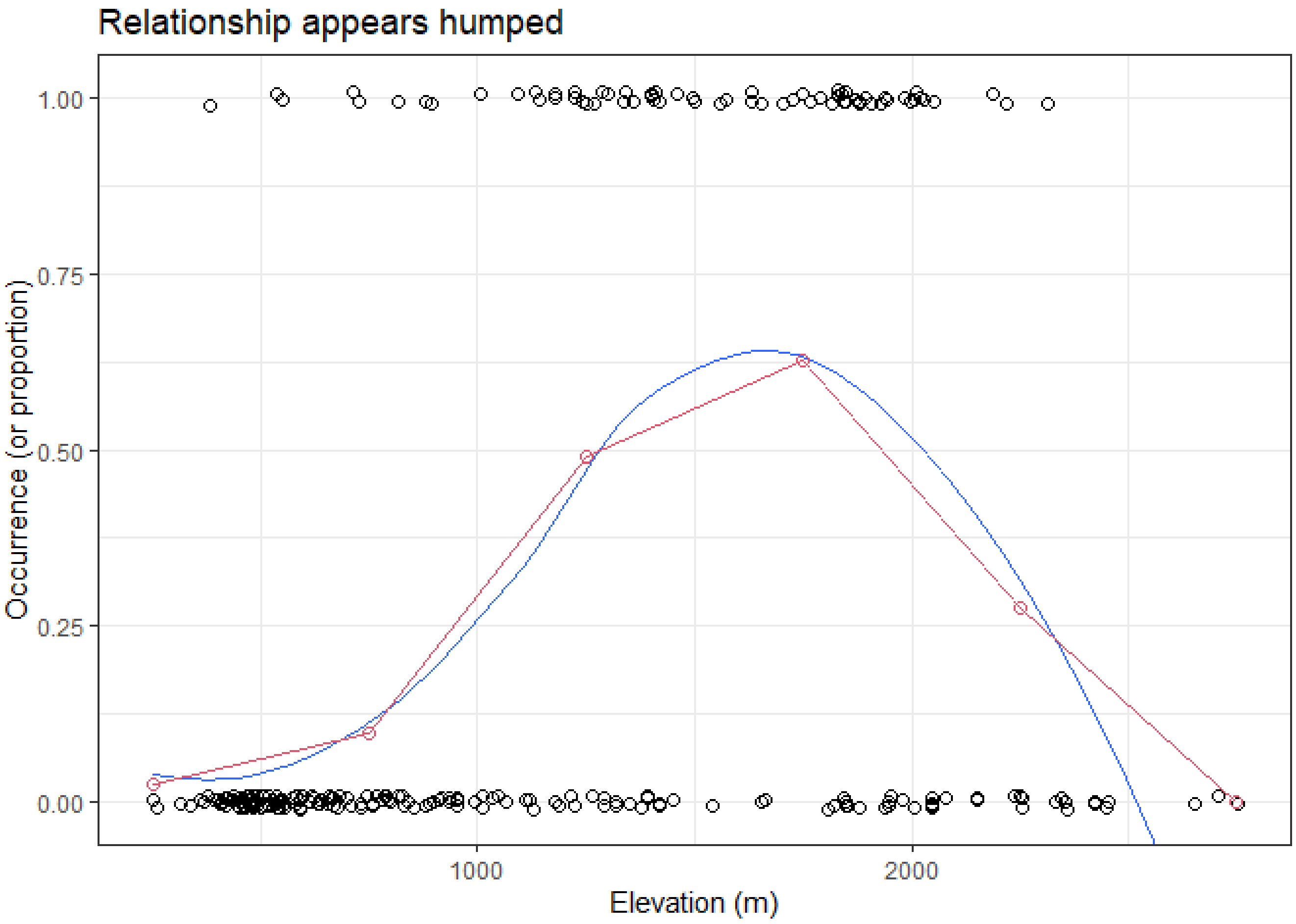
A 1 km$^2$ survey cell

- Willow tit territory presence-absence in relation to altitude

see swissbbs.R

Royle JA, Dorazio RM (2008) Hierarchical Modeling and Inference in Ecology. Academic Press, Oxford. p 87.

# EDA



Smoothing and binning are useful to visualize binary data

# Basic binomial-logit link GLM

- Linear predictor; logit link function Biology/pattern

Logit function

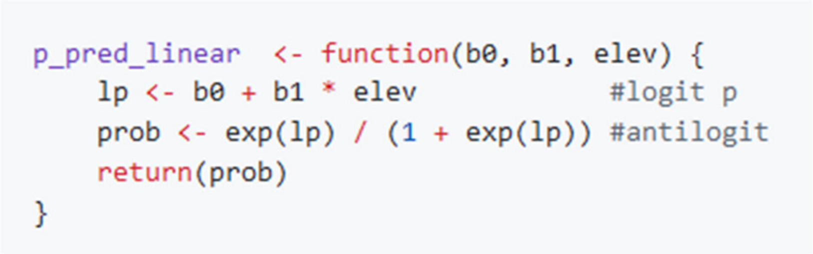
$$\mathrm{logit}(p) = \overbrace{\ln\left(\frac{p}{1-p}\right)} = \beta_0 + \beta_1 x$$

Linear function

Elevation

$$p = \frac{\exp(\mathrm{logit}(p))}{1 + \exp(\mathrm{logit}(p))}$$

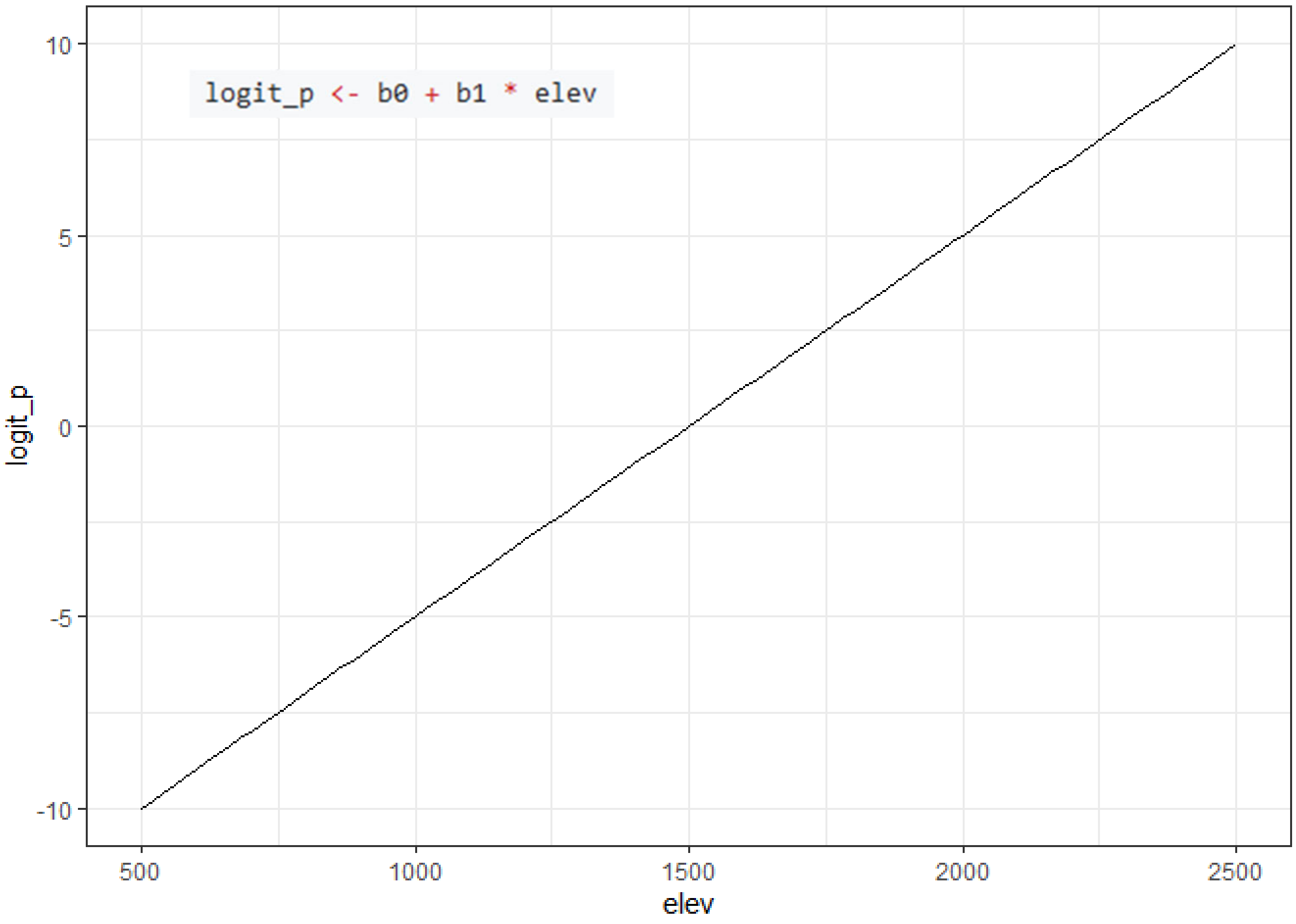
Antilogit function: backtransforms to give probability *p*



```
p_pred_linear  <- function(b0, b1, elev) {
    lp <- b0 + b1 * elev            #logit p
    prob <- exp(lp) / (1 + exp(lp)) #antilogit
    return(prob)
}
```

# Basic binomial-logit link GLM

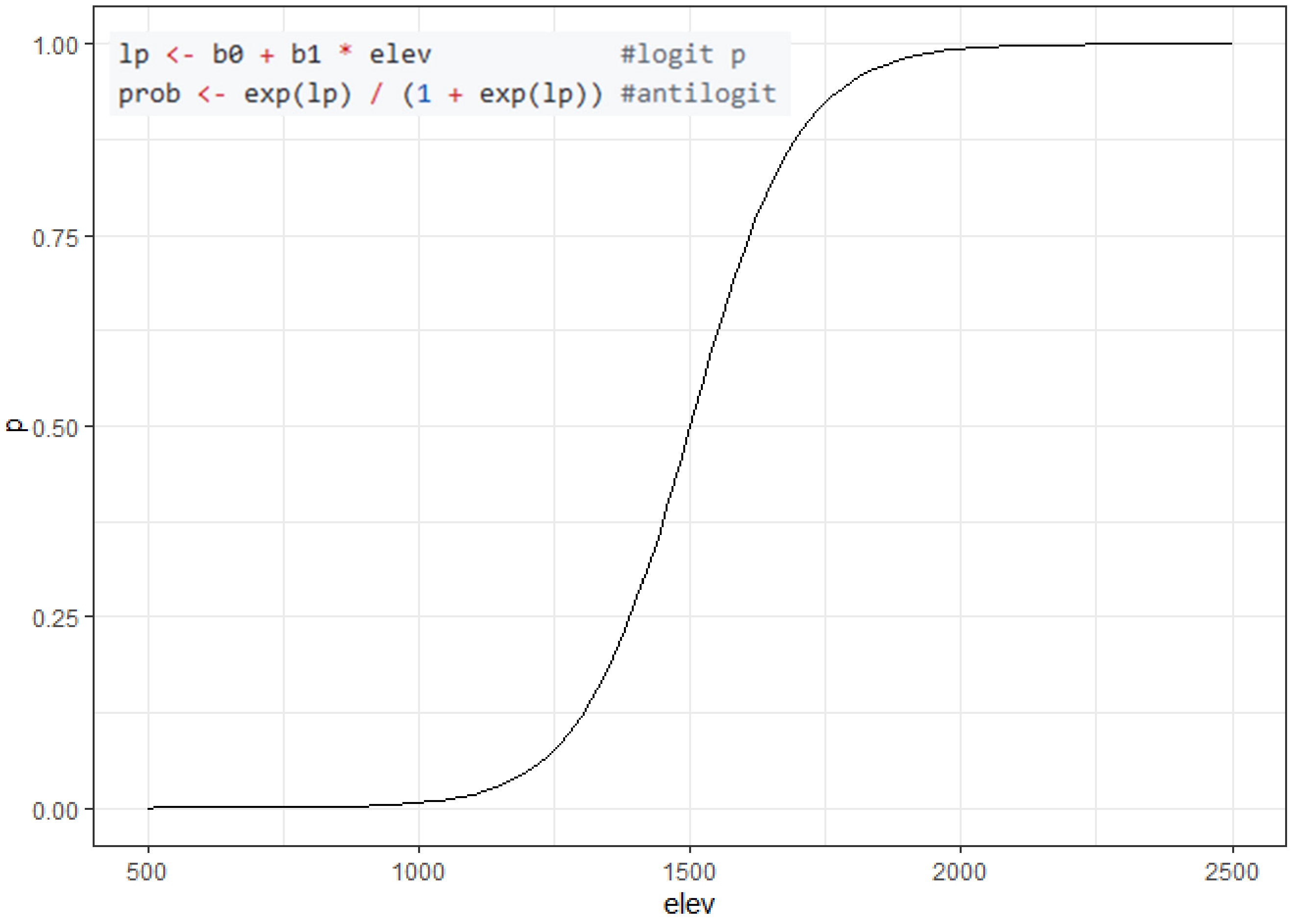
Logit scale is linear –INF to +INF



```
logit_p <- b0 + b1 * elev
```

# Basic binomial-logit link GLM

Probability (antilogit) scale is sigmoid 0 to 1, monotonic



```
lp <- b0 + b1 * elev                #logit p
prob <- exp(lp) / (1 + exp(lp)) #antilogit
```

# The humped model

- Linear predictor; logit link function

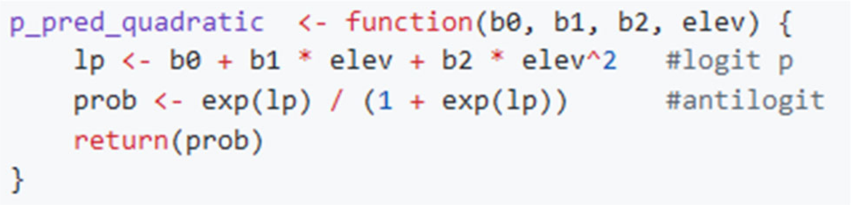
Biology/pattern

$$\text{logit}(p) = \overbrace{\ln\left(\frac{p}{1-p}\right)}^{\text{Logit function}} = \beta_0 + \beta_1 \underbrace{x}_{\text{Elevation}} + \beta_2 x^2$$

Quadratic function allows for hump

$$p = \frac{\exp(\text{logit}(p))}{1 + \exp(\text{logit}(p))}$$

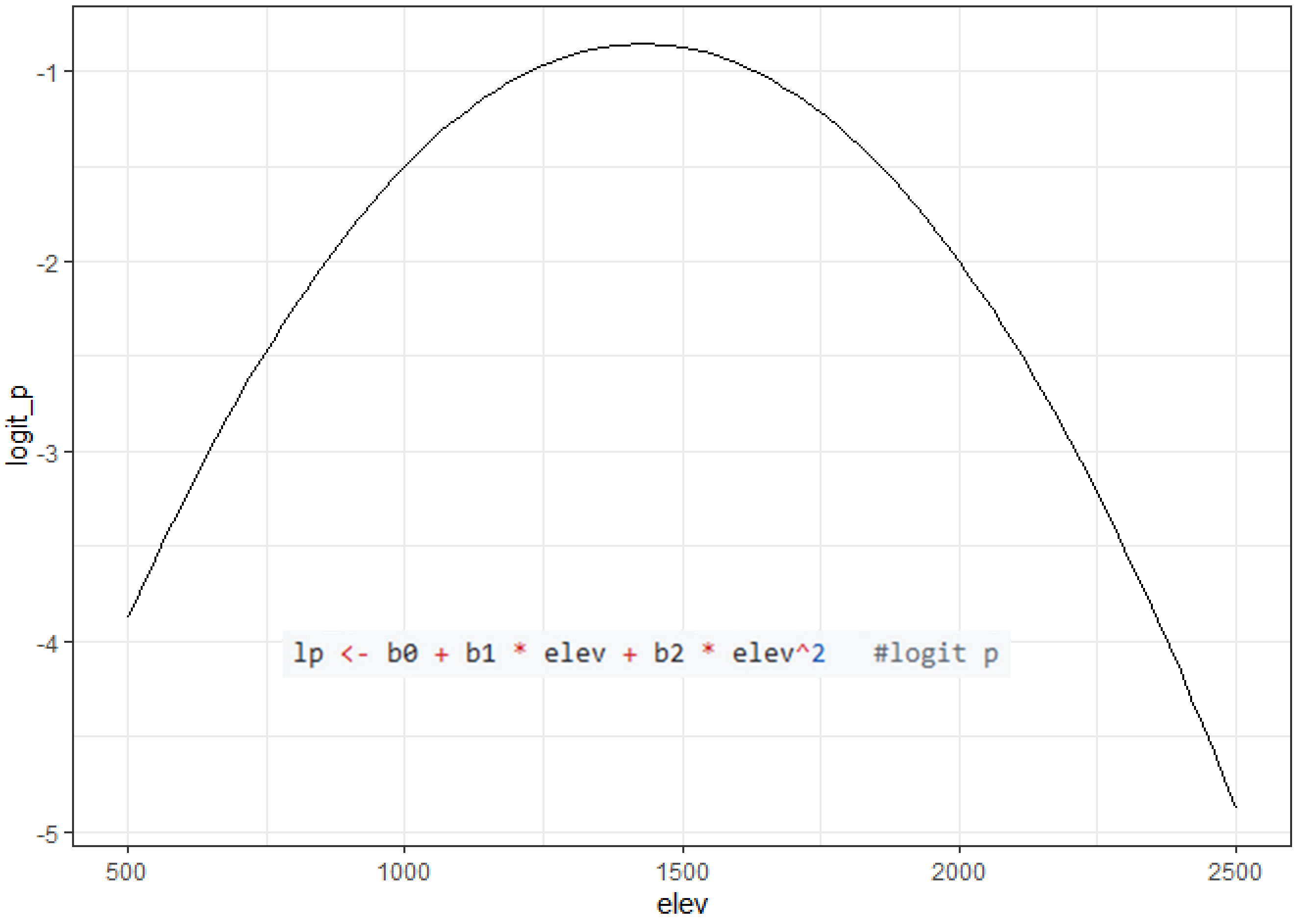
Antilogit function: backtransforms to give probability *p*



```
p_pred_quadratic  <- function(b0, b1, b2, elev) {
    lp <- b0 + b1 * elev + b2 * elev^2   #logit p
    prob <- exp(lp) / (1 + exp(lp))      #antilogit
    return(prob)
}
```

# The humped model

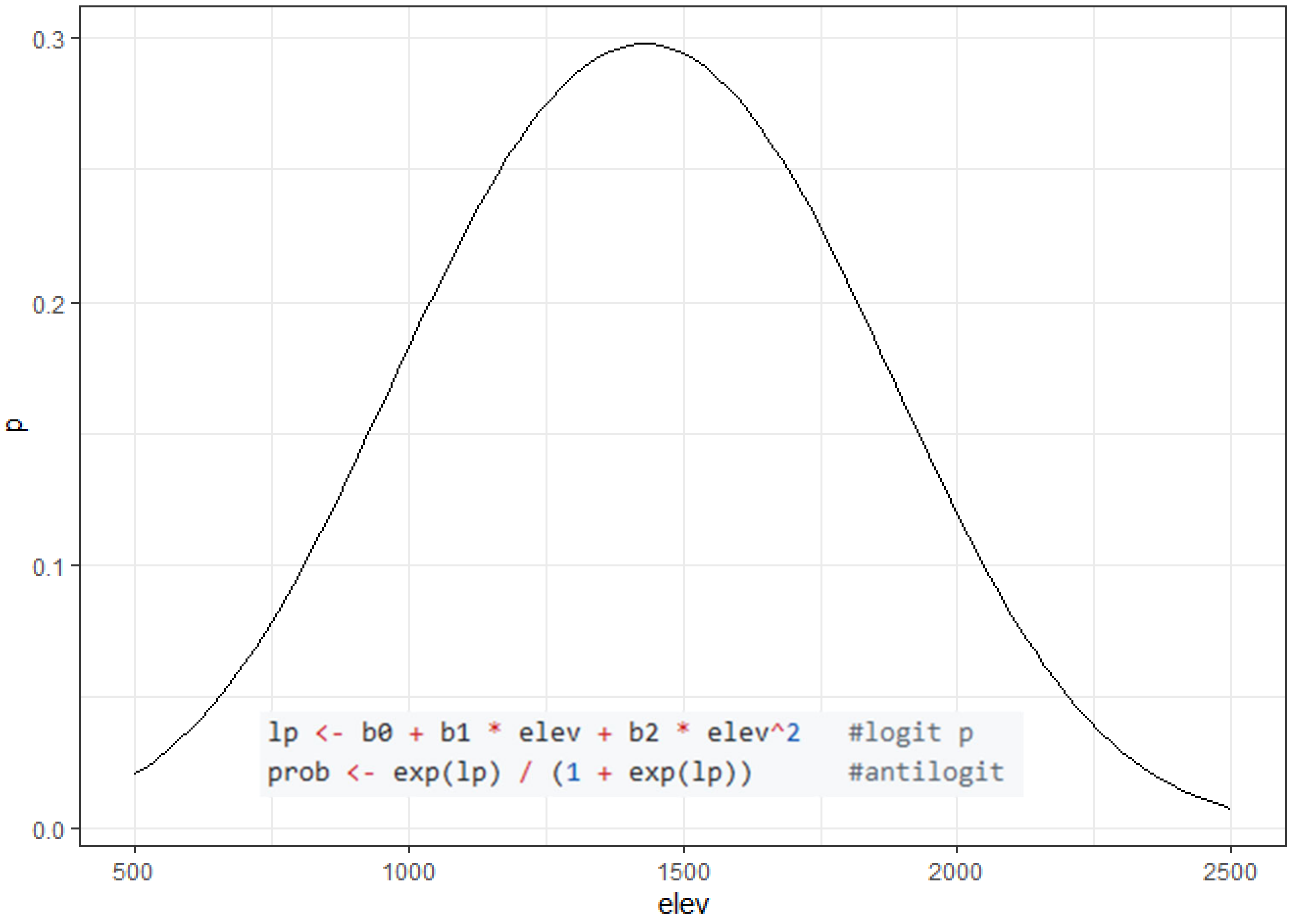
Logit scale



```
lp <- b0 + b1 * elev + b2 * elev^2   #logit p
```

# The humped model

Probability (antilogit) scale



```
lp <- b0 + b1 * elev + b2 * elev^2    #logit p
prob <- exp(lp) / (1 + exp(lp))       #antilogit
```

# The humped model

- Linear predictor; logit link function Biology/pattern

Logit function

$$\operatorname{logit}(p) = \overbrace{\ln\left(\frac{p}{1-p}\right)} = \beta_0 + \beta_1 x + \beta_2 x^2$$

Quadratic function allows for hump

Elevation

$$p = \frac{\exp(\operatorname{logit}(p))}{1 + \exp(\operatorname{logit}(p))}$$

Antilogit function: backtransforms to give probability *p*

- Data model (expected biology + stochasticity)

$$Y \sim \operatorname{Binomial}(p, N = 1)$$

Number of trials = 1

*Y* is occurrence

# Model algorithm

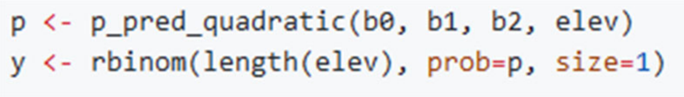
- For the probability

```
p_pred_quadratic  <- function(b0, b1, b2, elev){
       lp <- b0 + b1 * elev + b2 * elev^2   #logit p
       prob <- exp(lp) / (1 + exp(lp))      #antilogit
       return(prob)
}
```

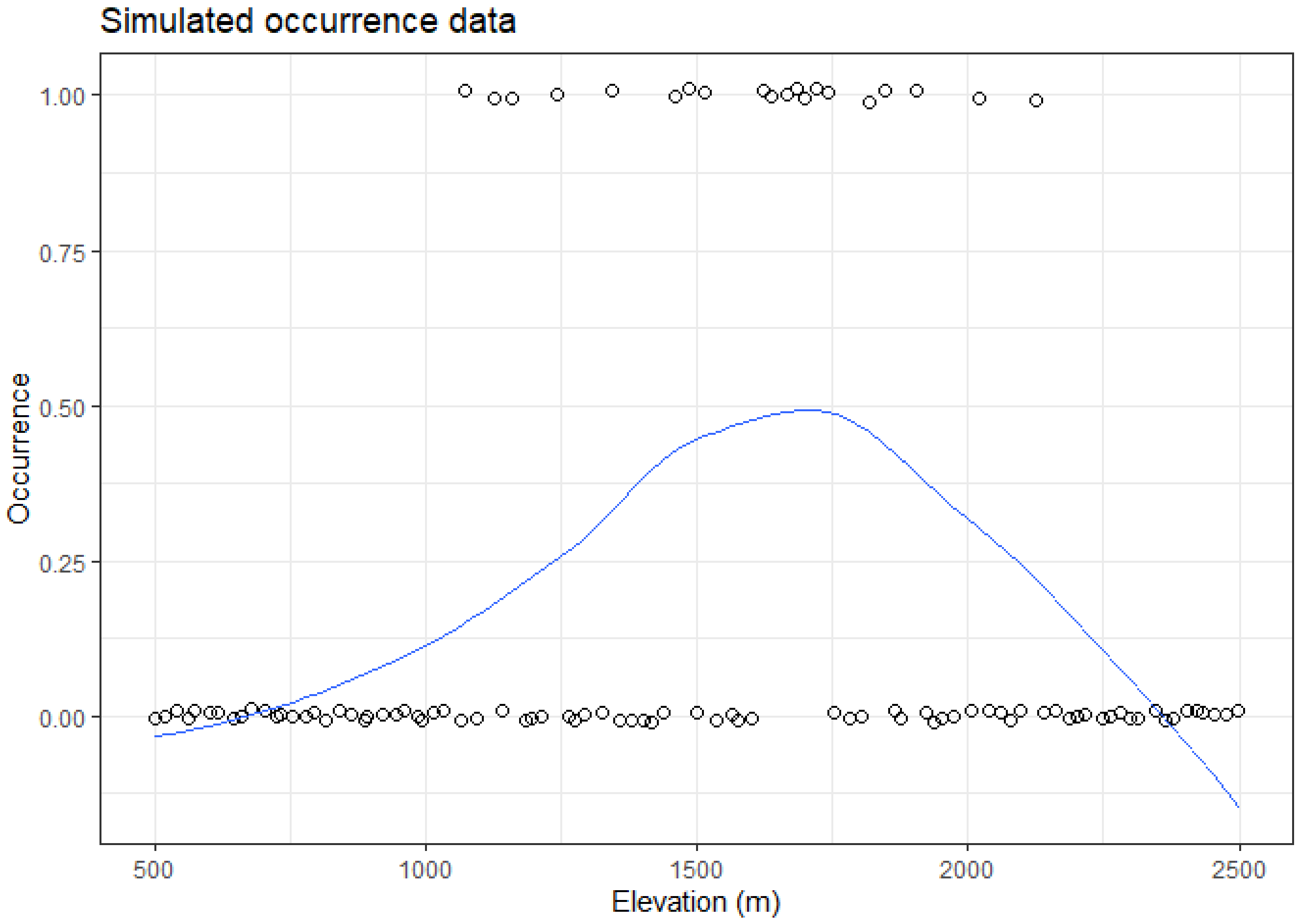
- For data generating process

```
rbinom(n, size=1, prob=p_pred_quadratic(b0,b1,b2,elev))
```

# Data generating process



```
p <- p_pred_quadratic(b0, b1, b2, elev)
y <- rbinom(length(elev), prob=p, size=1)
```



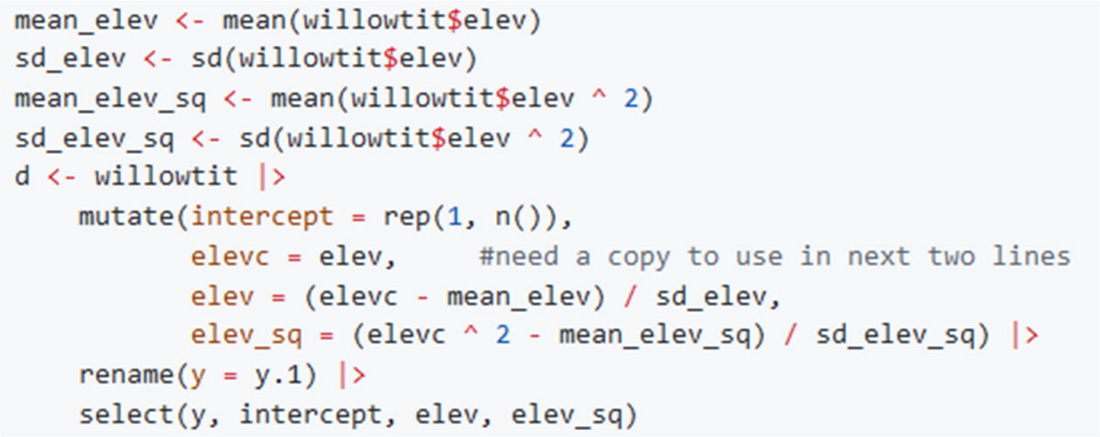
# Bayesian model

Add priors

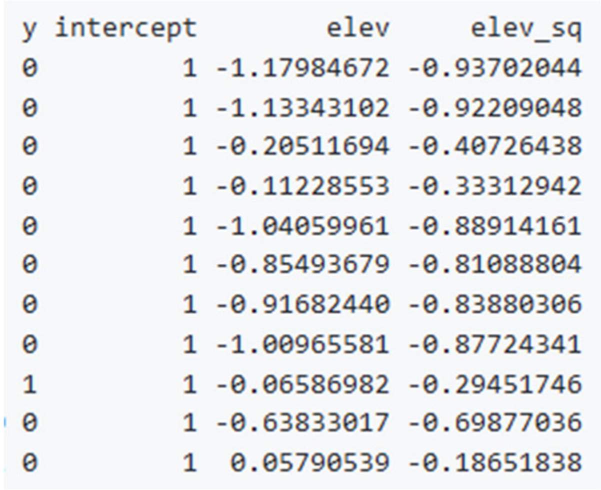
$$
\begin{aligned}
y_i &\sim \mathrm{Bernoulli}(p_i) \\
\ln\left(\frac{p}{1-p}\right) &= \beta_0 + \beta_1\,\mathrm{elev} + \beta_2\,\mathrm{elev}^2 \\
\beta_0 &\sim \mathrm{Normal}(0, 2.5) \\
\beta_1 &\sim \mathrm{Normal}(0, 2.5) \\
\beta_2 &\sim \mathrm{Normal}(0, 2.5)
\end{aligned}
$$

# Training algorithm (HMC)

## Standardized variables



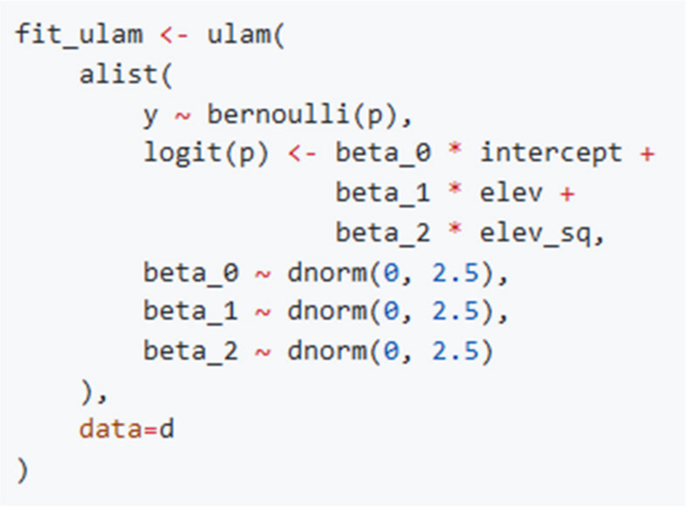
```
mean_elev <- mean(willowtit$elev)
sd_elev <- sd(willowtit$elev)
mean_elev_sq <- mean(willowtit$elev ^ 2)
sd_elev_sq <- sd(willowtit$elev ^ 2)
d <- willowtit |>
    mutate(intercept = rep(1, n()),
           elevc = elev,     #need a copy to use in next two lines
           elev = (elevc - mean_elev) / sd_elev,
           elev_sq = (elevc ^ 2 - mean_elev_sq) / sd_elev_sq) |>
    rename(y = y.1) |>
    select(y, intercept, elev, elev_sq)
```



```
y intercept        elev     elev_sq
0         1 -1.17984672 -0.93702044
0         1 -1.13343102 -0.92209048
0         1 -0.20511694 -0.40726438
0         1 -0.11228553 -0.33312942
0         1 -1.04059961 -0.88914161
0         1 -0.85493679 -0.81088804
0         1 -0.91682440 -0.83880306
0         1 -1.00965581 -0.87724341
1         1 -0.06586982 -0.29451746
0         1 -0.63833017 -0.69877036
0         1  0.05790539 -0.18651838
```

# Training algorithm (HMC)

## Ulam model specification



```
fit_ulam <- ulam(
    alist(
        y ~ bernoulli(p),
        logit(p) <- beta_0 * intercept +
                    beta_1 * elev +
                    beta_2 * elev_sq,
        beta_0 ~ dnorm(0, 2.5),
        beta_1 ~ dnorm(0, 2.5),
        beta_2 ~ dnorm(0, 2.5)
    ),
    data=d
)
```

# Training algorithm (HMC)

- Bernoulli likelihood component

Likelihood for a single data point

$$L(p) = P(Y = \underset{\substack{\uparrow \\ \text{1 for present} \\ \text{0 for absent}}}{y}|p) = \underbrace{p^y(1-p)^{1-y}}_{\text{Bernoulli distribution}}$$

y = 1:

$$L(p) = P(Y = 1|p) = p^1(1-p)^{1-1} = p$$

y = 0:

$$L(p) = P(Y = 0|p) = p^0(1-p)^{1-0} = 1-p$$

# Rescaling parameters

We fitted

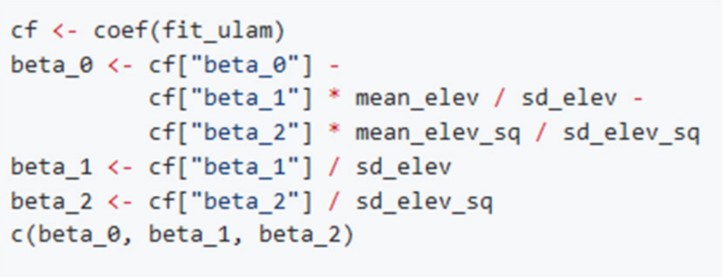
$$\eta = \beta'_0 + \beta'_1 \underset{\overset{\nearrow}{\text{elev}}}{x'_1} + \beta'_2 \underset{\overset{\nwarrow}{\text{elev^2}}}{x'_2} \qquad x' = \frac{x - \bar{x}}{\sigma_x}$$

So:

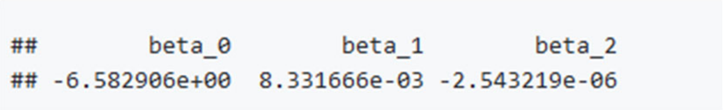
$$\begin{aligned}\eta &= \beta'_0 + \frac{\beta'_1}{\sigma_{x_1}} x_1 - \beta'_1 \frac{\bar{x}_1}{\sigma_{x_1}} + \frac{\beta'_2}{\sigma_{x_2}} x_2 - \beta'_2 \frac{\bar{x}_2}{\sigma_{x_2}} \\ &= \left(\beta'_0 - \beta'_1 \frac{\bar{x}_1}{\sigma_{x_1}} - \beta'_2 \frac{\bar{x}_2}{\sigma_{x_2}}\right) + \left(\frac{\beta'_1}{\sigma_{x_1}}\right) x_1 + \left(\frac{\beta'_2}{\sigma_{x_2}}\right) x_2 \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2\end{aligned}$$

Original scale parameters:

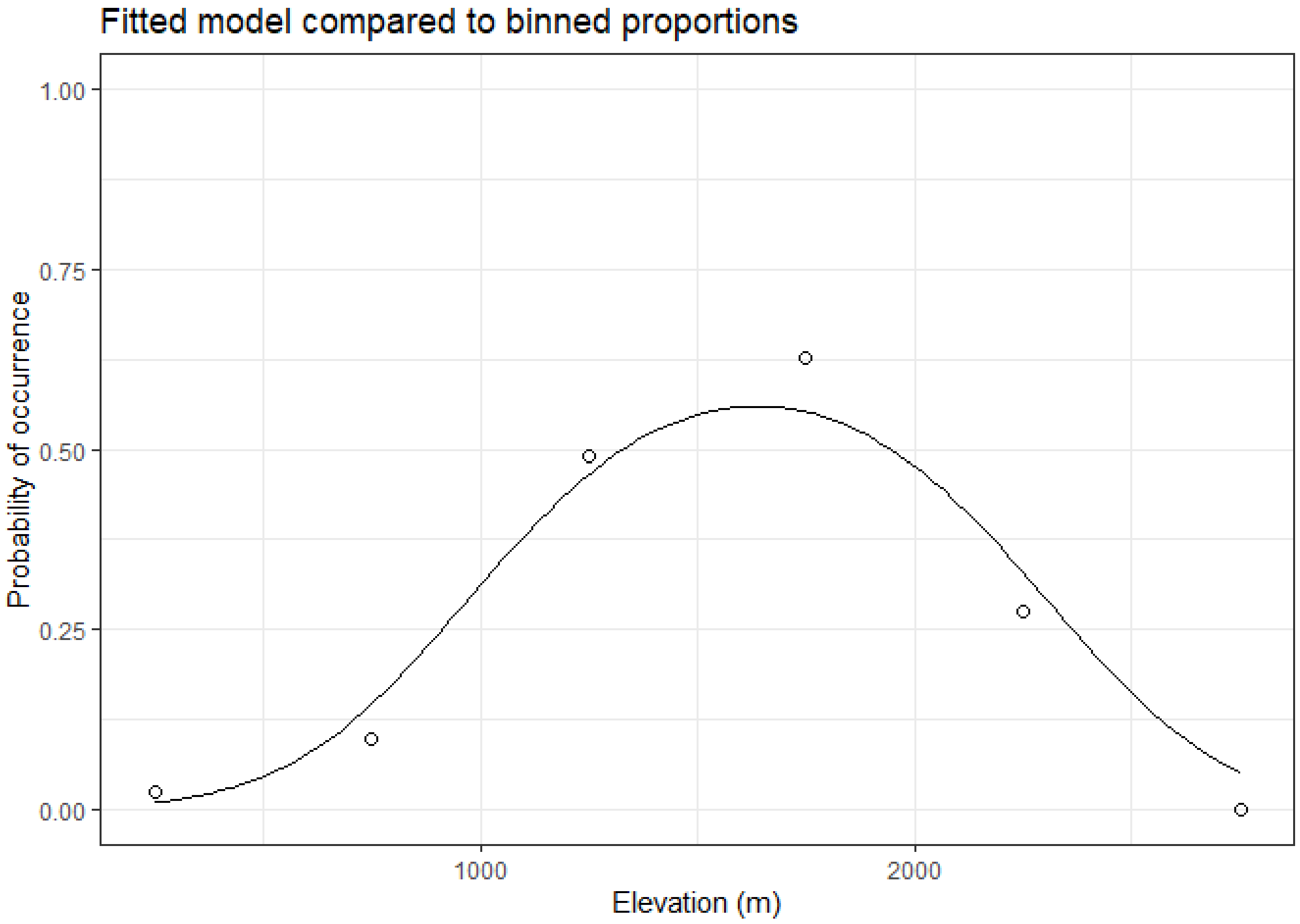
$$\beta_0 = \beta'_0 - \beta'_1 \frac{\bar{x}_1}{\sigma_{x_1}} - \beta'_2 \frac{\bar{x}_2}{\sigma_{x_2}} \qquad \beta_1 = \frac{\beta'_1}{\sigma_{x_1}} \qquad \beta_2 = \frac{\beta'_2}{\sigma_{x_2}}$$



```
cf <- coef(fit_ulam)
beta_0 <- cf["beta_0"] -
          cf["beta_1"] * mean_elev / sd_elev -
          cf["beta_2"] * mean_elev_sq / sd_elev_sq
beta_1 <- cf["beta_1"] / sd_elev
beta_2 <- cf["beta_2"] / sd_elev_sq
c(beta_0, beta_1, beta_2)
```



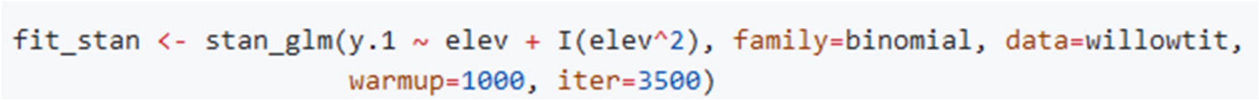
```
##        beta_0        beta_1        beta_2
## -6.582906e+00  8.331666e-03 -2.543219e-06
```



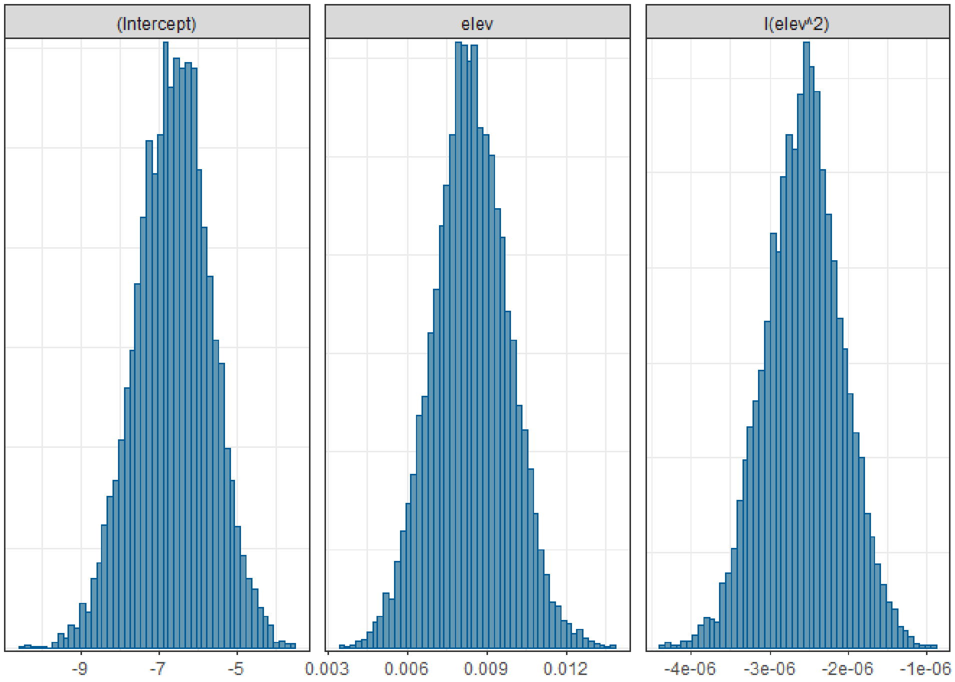
Fitted model compared to binned proportions

# Training algorithm (HMC)

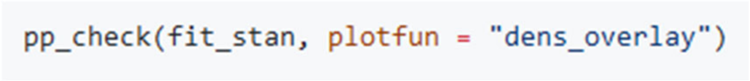
- rstanarm version



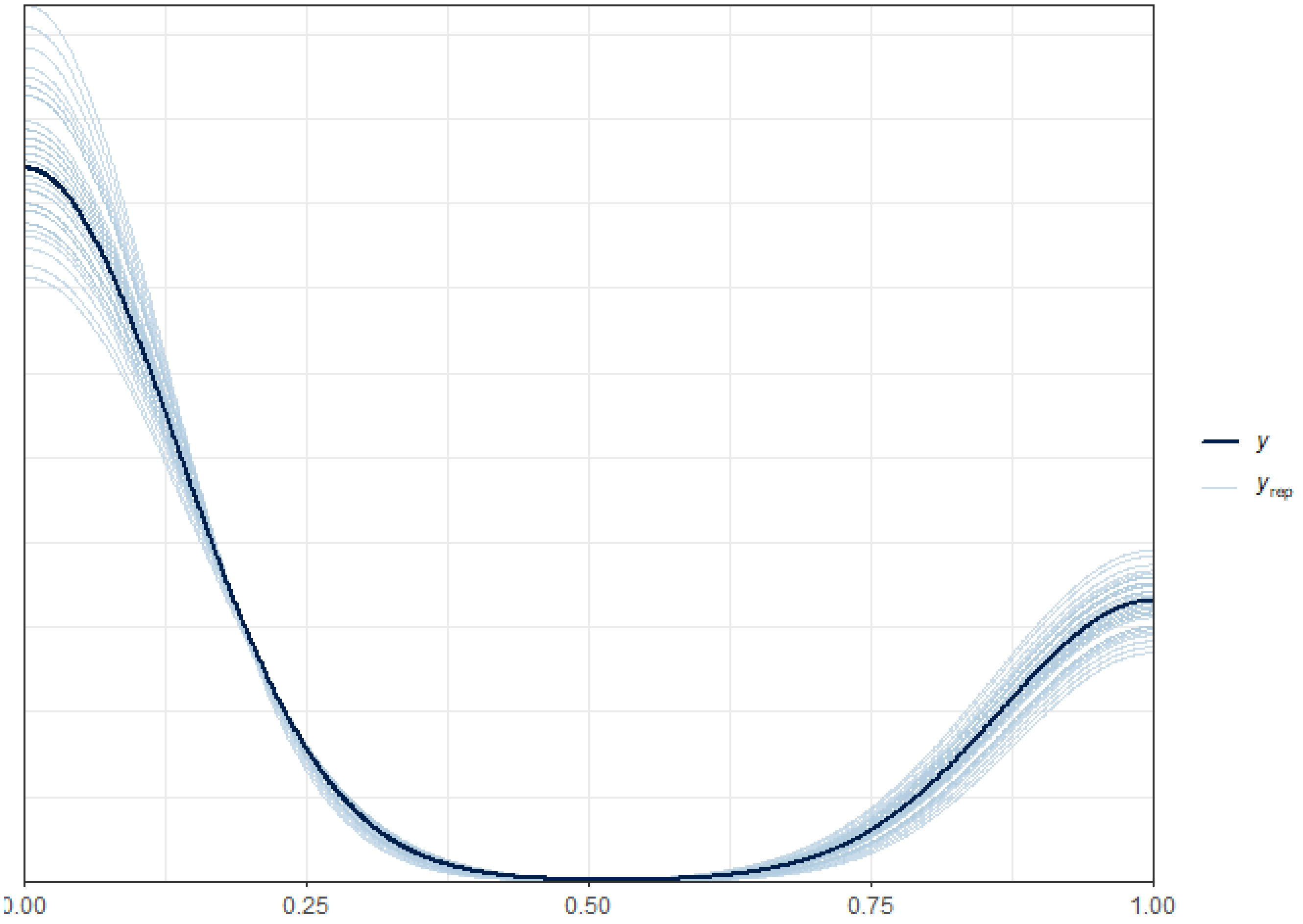
```r
fit_stan <- stan_glm(y.1 ~ elev + I(elev^2), family=binomial, data=willowtit,
                     warmup=1000, iter=3500)
```



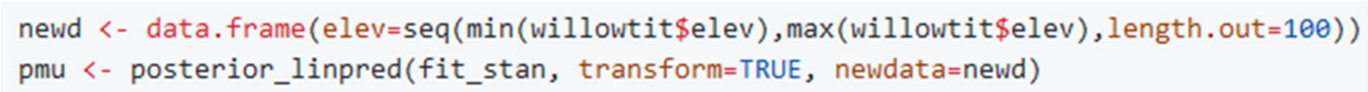
# Model check



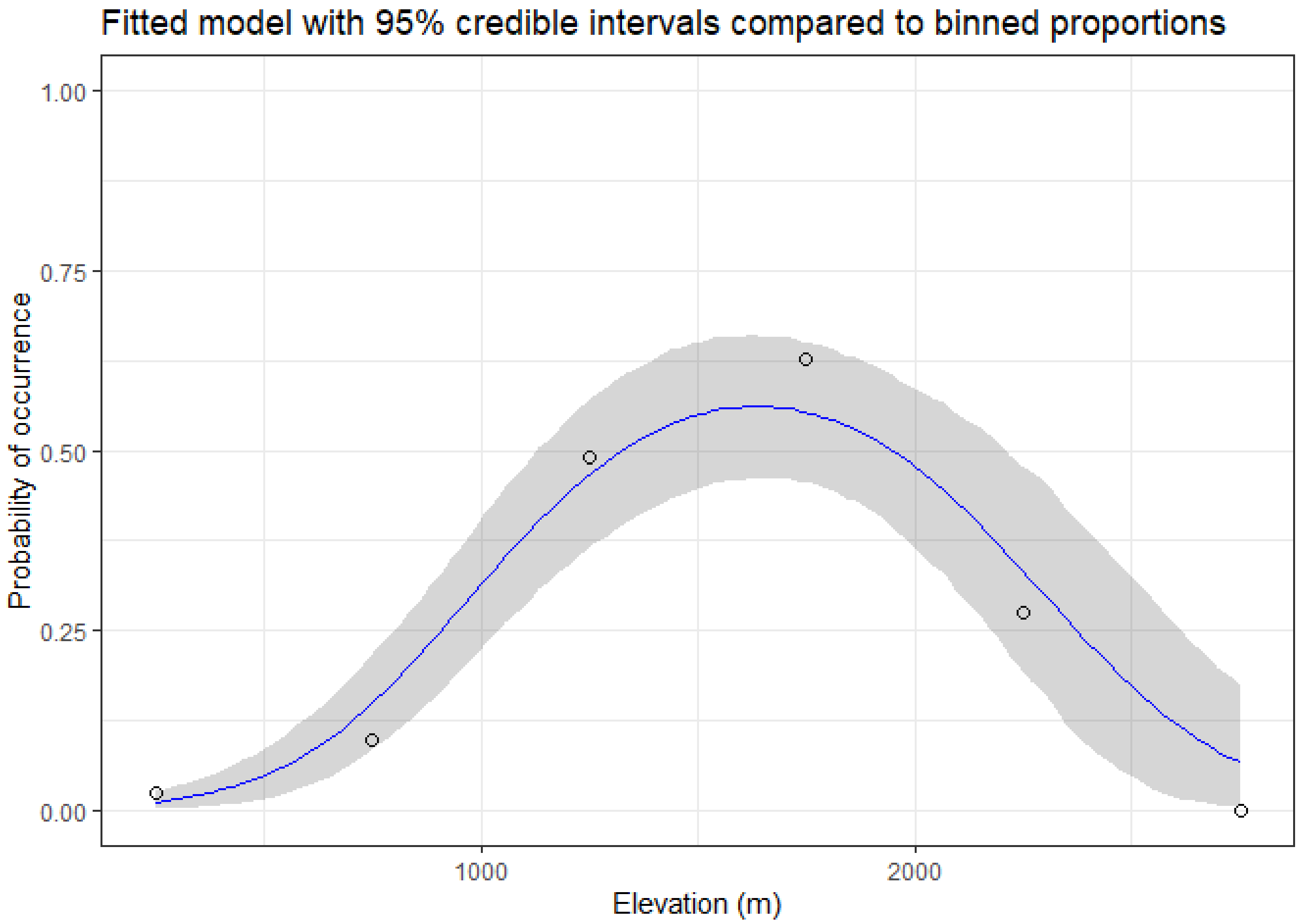
```
pp_check(fit_stan, plotfun = "dens_overlay")
```



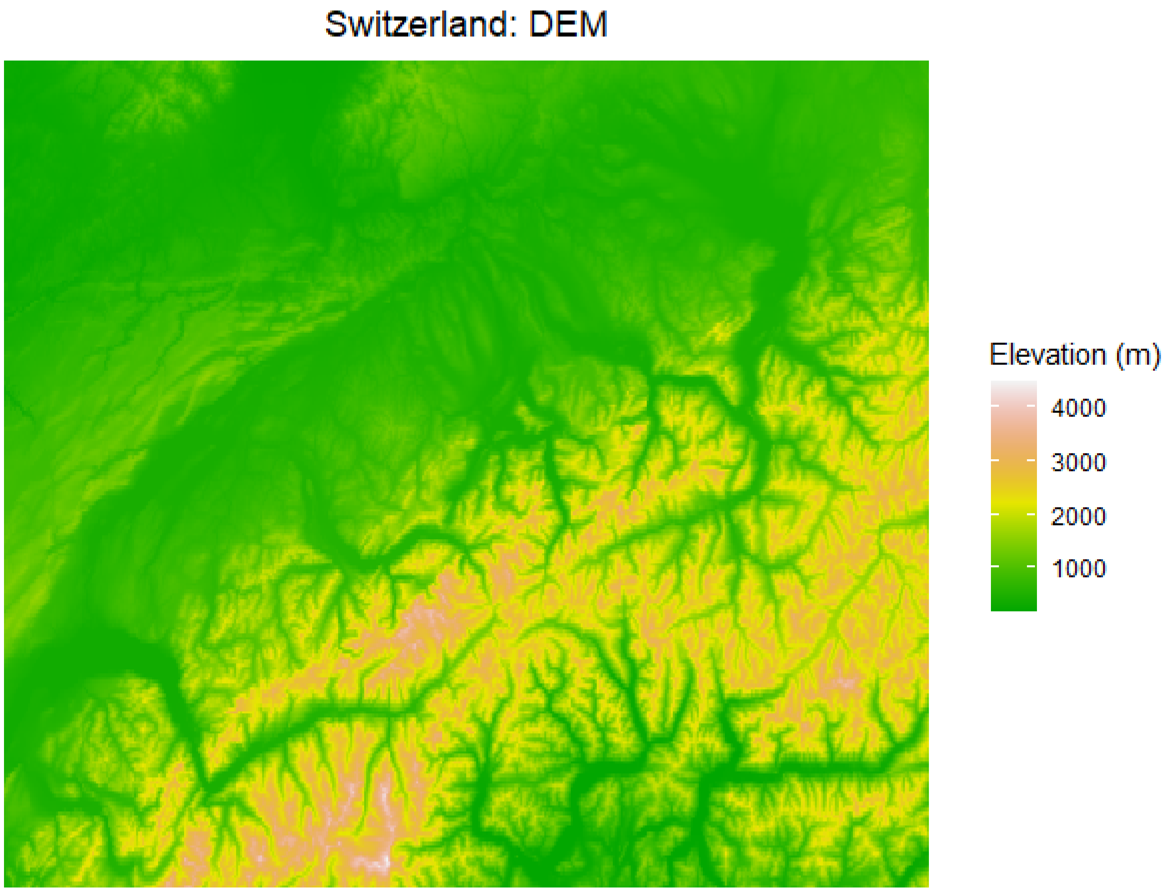
# Inference algorithm



```r
newd <- data.frame(elev=seq(min(willowtit$elev),max(willowtit$elev),length.out=100))
pmu <- posterior_linpred(fit_stan, transform=TRUE, newdata=newd)
```

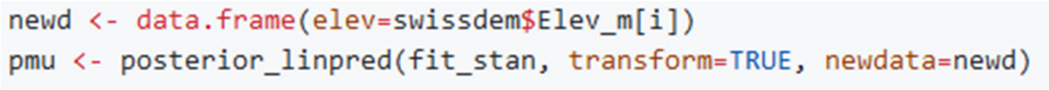


Fitted model with 95% credible intervals compared to binned proportions

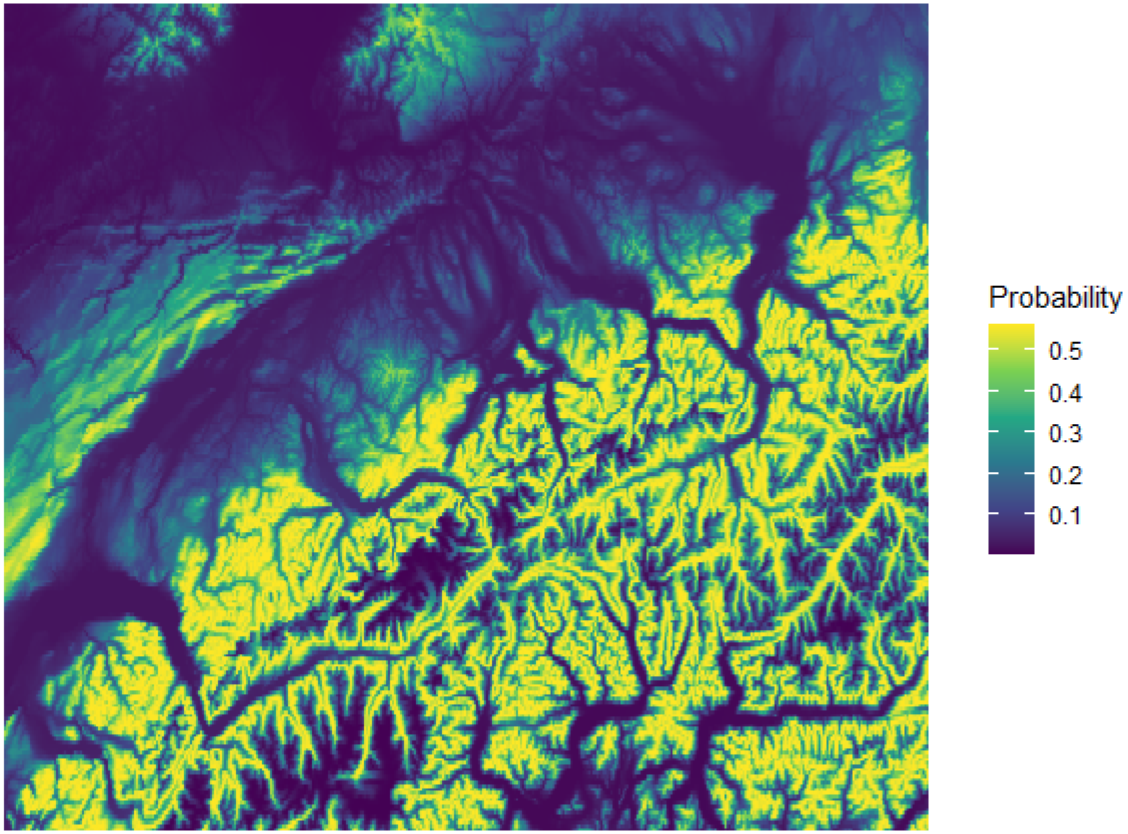


Switzerland: DEM

# Predictions as a map



```
newd <- data.frame(elev=swissdem$Elev_m[i])
pmu <- posterior_linpred(fit_stan, transform=TRUE, newdata=newd)
```



Probability of observing a willow tit in year 1

# R code

- swissbbs_binomial_GLM.Rmd
- swissbbs_binomial_GLM.md