Module 7: Generalized linear models

Jake Ferguson (jakeferg@hawaii.edu)

Module goals

1. Generalize regression to count and categorical data
2. Identify proper distribution to use for particular datasets
3. Evaluate model quality

Linear regression

The lm framework:

$$\mu = \beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_p x_p$$
$$Y \sim N(\mu, \sigma^2)$$

We assume that the residuals are normally distributed around the mean, μ.

Limitations of `lm`

Although linear models are a very useful framework, there are some situations where they are not appropriate

- the range of Y is restricted (e.g. binary, count)
- the variance of Y depends on the mean

Generalized linear models (glms) extend the linear model framework to address both of these issues

Generalized linear regression

In a glm there is some transformation of the mean, $g(\mu)$, called the **link** function, that results in a linear model.

$$
\begin{aligned}
g(\mu) &= \beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_p x_p \\
\mu &= g^{-1}\left(\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_p x_p\right) \\
Y &\sim f(\mu, \theta)
\end{aligned}
$$

We assume that the residuals are distributed around the mean, μ, following some distribution $f(\cdot)$ (e.g., Binomial, Poisson, Negative Binomial). Where θ is any relevant additional parameters needed to model the variance.

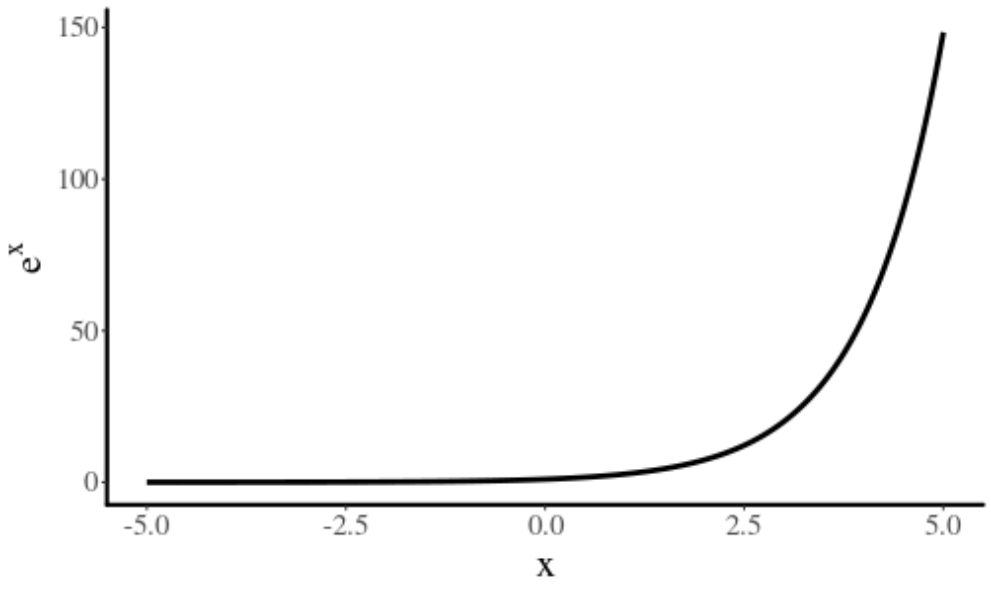
By choosing an appropriate **link** function, $g(\cdot)$, we will ensure that the mean only takes on values that are supported by the distribution (for example only positive values for the Poisson or between 0 and 1 for the Binomial).

Modeling counts: Poisson regression

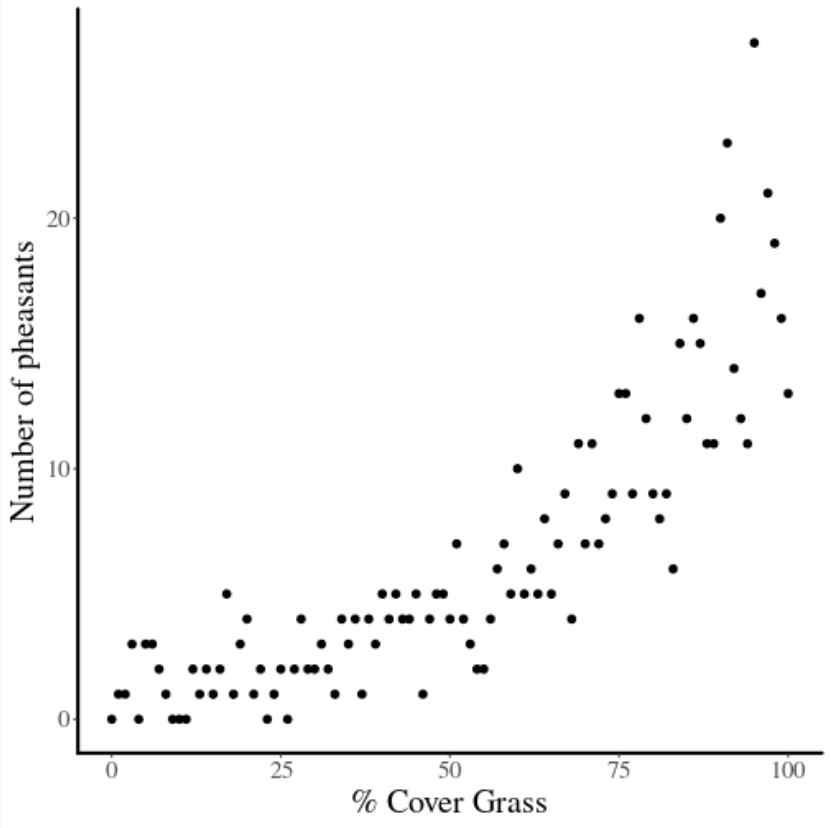
Use the log-link function: $g(\mu) = ln(\mu)$. Then the the inverse links is $e^{\cdot}$.

$$\mu = e^{\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_p x_p}$$
$$Y \sim \text{Poisson}(\mu)$$

$$E[Y] = \mu, \quad Var[Y] = \mu$$



Continuous predictor variables



$$\mu = e^{0.01+0.03 \cdot X}$$
$$Y \sim \text{Poisson}(\mu)$$

Using `glm`

The syntax for `glm` follows `lm` closely:

```
glm(RESPONSE ~ X1 + X2, data=data.csv, family=poisson(link="log"))
```

We've added the `family` argument to specify which distribution to use. We've also added `(link="log")` to specify the link function we use.

This website has a nice table of families and their link functions in `glm`:
https://data.princeton.edu/R/GLMs

glm model output

```
summary(glm(Pheasants ~ grassCover, data=pheas.dat, family=poisson))
```

```
##
## Call:
## glm(formula = Pheasants ~ grassCover, family = poisson, data = pheas.dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1059  -0.7261  -0.1055   0.5581   2.2426
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.029698   0.125016   0.238    0.812
## grassCover  0.029486   0.001648  17.896   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 484.156  on 100  degrees of freedom
## Residual deviance:  94.911  on  99  degrees of freedom
## AIC: 424.02
##
```

How to interpret β_0 and β_1?

```
coef(glm(Pheasants ~ grassCover, data=pheas.dat, family=poisson))
```

```
## (Intercept)  grassCover
##  0.02969773  0.02948565
```

On the **link** scale parameters mean the same as in `lm`.

$$\ln(\mu) = \beta_0 + \beta_1 X$$

β_0: intercept of the log-mean

β_1: slope of the log-mean

On the **natural** scale they may mean something different:

$$\mu = e^{\beta_0 + \beta_1 X}$$
$$\mu = e^{\beta_0} \cdot e^{\beta_1 X}$$

β_0: mean log-fecundity $at X = 0$

β_1: rate of change in fecundity with X

Deviance

```
Null deviance: 484.156 on 100 degrees of freedom
```

```
Residual deviance: 94.911 on 99 degrees of freedom
```

Deviance is a measure of model fit.

Null deviance is the difference in fit between the saturated model and the intercept model

- the *saturated* model has a parameter at each observation
- the intercept only has a single parameter

Residual deviance is the difference in fit between the saturated model and the fitted model

Why Deviance?

There is no R^2 for glm's typically so we can instead use the proportion deviance explained (called the pseudo R^2).

$$R^2_{\text{pseudo}} = 1 - \frac{\text{Residual deviance}}{\text{Null deviance}}$$

- **Null deviance** $\approx$ equivalent of total sum of squares.
- **Residual deviance** $\approx$quivalent of residual sum of squares.

For the pheasant model, $R^2_{pseudo} = 0.8$.

Exercise 7A

Fit a count model

Offsets in Poisson regression

Count data (Y) are often collected:

- over varying lengths of time
- in sample units that have different areas

So we are often interested in modeling rates:

$$Y/\text{time}$$

or densities:

$$Y/\text{Area}$$

Modeling beaver densities in MN

Each route is a different length.

Beaver density `glm`

Using route length as an **offset** controls for differences in survey effort.

```
##
## Call:
## glm(formula = num.col ~ rte.name + offset(log(rte.km)), family = poisson,
##     data = beav.dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.5792  -1.3366  -0.1332   1.3064   7.2427
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -0.07145    0.02529  -2.826  0.00472 **
## rte.nameC_st_louis      -0.62041    0.03196 -19.411  < 2e-16 ***
## rte.nameCass            -0.64000    0.03268 -19.584  < 2e-16 ***
## rte.nameCass_crow       -0.71424    0.03038 -23.507  < 2e-16 ***
## rte.nameEly_finger      -0.22439    0.03158  -7.106 1.19e-12 ***
## rte.nameHay_kelliher    -0.84487    0.03514 -24.045  < 2e-16 ***
## rte.nameItasca          -0.83408    0.03774 -22.101  < 2e-16 ***
## rte.nameKabetogama       0.54214    0.02975  18.221  < 2e-16 ***
## rte.nameKanabec         -1.21305    0.04222 -28.734  < 2e-16 ***
## rte.nameKawishiwi       -0.68557    0.04620 -14.840  < 2e-16 ***
## rte.nameKooch_north     -0.18667    0.03528  -5.291 1.22e-07 ***
```

Model selection

Can use same tools as linear regression.

- t-tests on parameter estimates
- Confidence intervals (`confint`)
- ANOVA (f-tests)
- AIC

Is the Poisson suitable?

An important assumption in the Poisson distribution is that the mean and variance are equal: $\mathbf{E}[Y] = \mathbf{Var}[Y]$.

How can we test this assumption?

- Examine residual plots
- Formal goodness of fit tests exist (Pearson's χ^2,)
- Fit model with overdispersion (next section) and compare via AIC

Overdispersion

Reasons data may be overdispersed

- Omitted variables
- Measurement error
- Wrong distribution

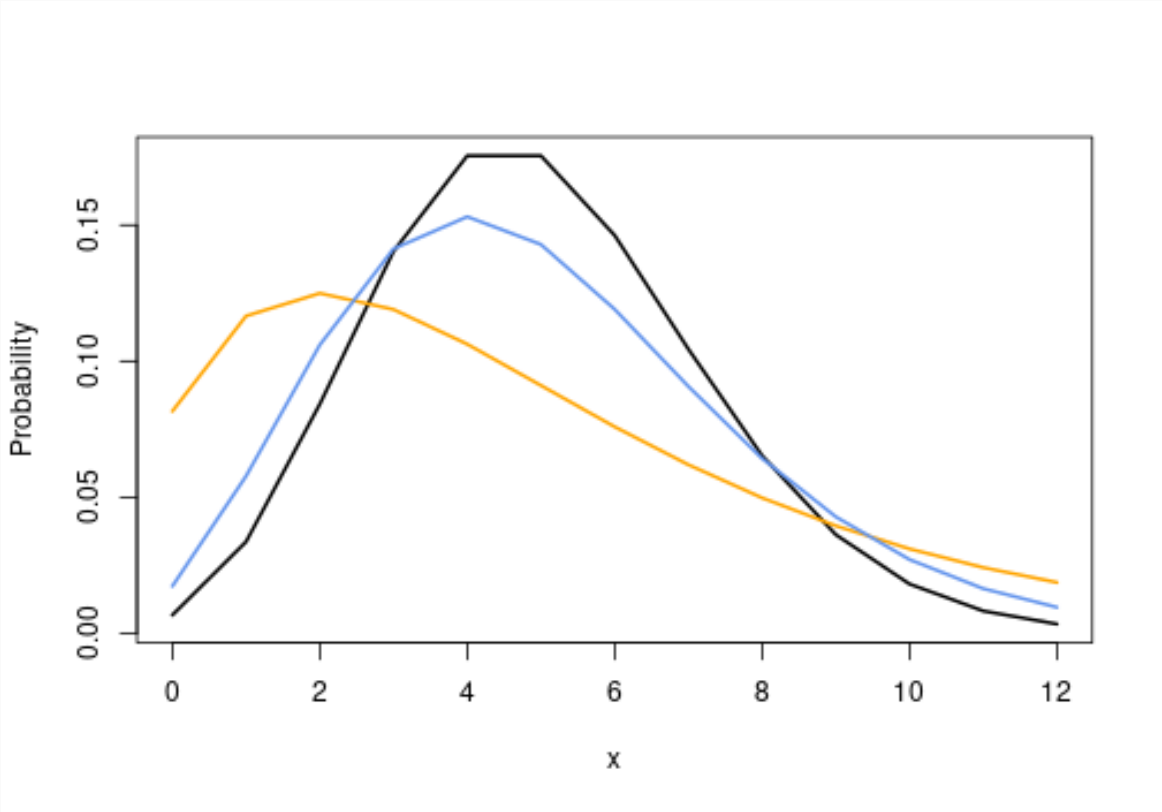
Consequences of overdispersion

- Standard errors may underestimated
- More complex models than necessary may be selected

Negative Binomial

The true variance is often higher (**overdispersion**) than the Poisson distribution. The Negative Binomial distribution allows for this overdispersion.

$$\mathrm{E}[y] = \mu$$
$$\mathrm{Var}[y] = \mu + \mu^2/\theta$$



Negative binomial regression

```
library(MASS)
library(MuMIn)
beaver.pois ← glm(num.col ~ rte.name + offset(log(rte.km)), data=beav.dat, family=po
beaver.nb ← glm.nb(num.col ~ rte.name + offset(log(rte.km)), data=beav.dat)

model.sel(list(beaver.pois, beaver.nb))
```

```
## Model selection table
##      (Int) rte.nam off(log(rte.km))        family  class init.theta link df
## 2 -0.07145       +                + NB(24.9368,l) negbin       24.9  log 17
## 1 -0.07145       +                +          p(l)    glm                16
##      logLik   AICc  delta weight
## 2 -1428.402 2892.8   0.00      1
## 1 -1772.298 3578.4 685.56      0
## Abbreviations:
## family: NB(24.9368,l) = 'Negative Binomial(24.9368,log)',
##         p(l) = 'poisson(log)'
## Models ranked by AICc(x)
```

Based on this output is **overdispersion** present?

`glm.nb` output

```
beaver.nb
```

```
##
## Call:  glm.nb(formula = num.col ~ rte.name + offset(log(rte.km)), data = beav.dat,
##     init.theta = 24.93683249, link = log)
##
## Coefficients:
##             (Intercept)       rte.nameC_st_louis             rte.nameCass
##                 -0.07145                -0.62041                -0.64000
##       rte.nameCass_crow       rte.nameEly_finger     rte.nameHay_kelliher
##                 -0.71424                -0.22439                -0.84487
##          rte.nameItasca       rte.nameKabetogama          rte.nameKanabec
##                 -0.83408                 0.54214                -1.21305
##       rte.nameKawishiwi      rte.nameKooch_north         rte.nameNorthome
##                 -0.68557                -0.18667                -0.46886
##        rte.nameRed_lake       rte.nameS_st_louis    rte.nameSouthern_pine
##                 -1.19178                -0.77329                -0.99222
## rte.nameWest_vermillion
##                 -0.48115
##
## Degrees of Freedom: 320 Total (i.e. Null);  305 Residual
##   (40 observations deleted due to missingness)
## Null Deviance:        1781
```

Exercise 7B

Negative binomial regression

Modeling categorgies: Binomial (logistic)

We've covered:

1. Continuous response, continuous predictor (regression)
2. Continuous response, discrete predictor (t-test, ANOVA)

Binomial regression has a **discrete response** with continuous or discrete predictors.

Examples: survival, annual recruitment, presence/absence, disease infection/recovery

Logistic regression

$$\operatorname{logit}(p) \equiv \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_p x_p$$

$$p = \frac{e^{\beta_0+\beta_1 \cdot x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 \cdot x_1+\cdots+\beta_p x_p}}$$

$$Y \sim \operatorname{binomial}(N,\, p)$$

The binomial distribution properties:

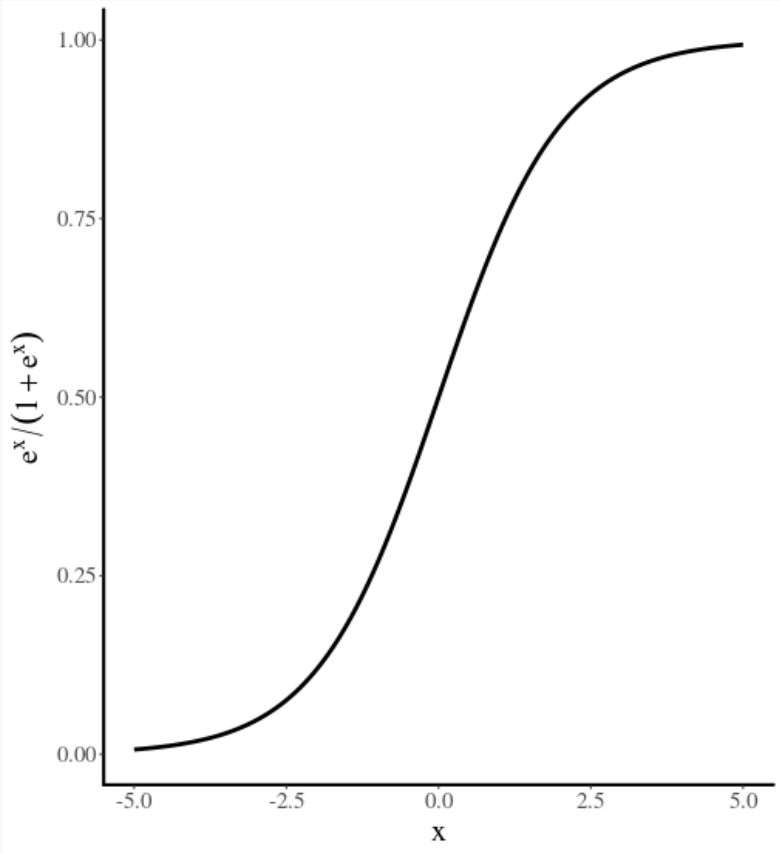
$$E[Y] = Np$$

$$Var[Y] = Np(1-p)$$

Odds

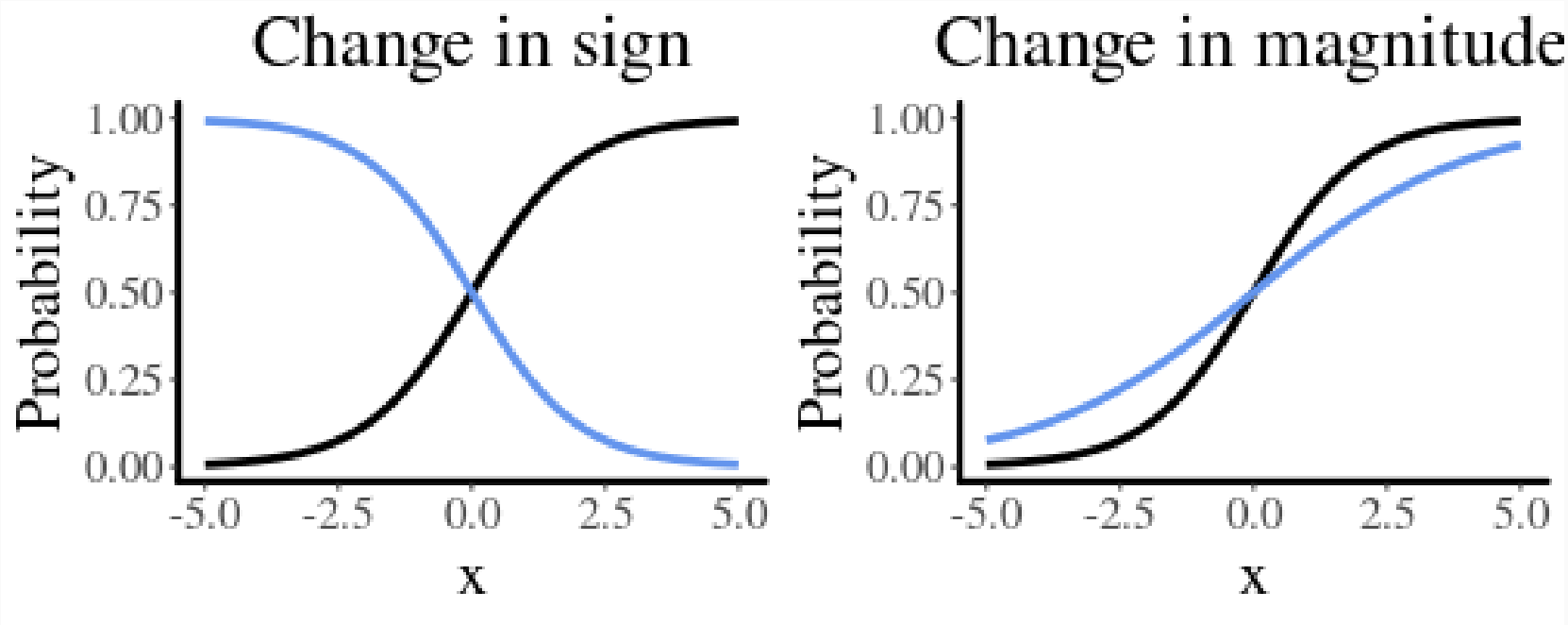
$\frac{p}{1-p}$ is called the **odds** and gives the relative probability of success. It is often used in betting.

p	odds	logit.log.odds.
0.10	0.11	-2.2
0.25	0.33	-1.1
0.50	1.00	0.0
0.75	3.00	1.1
0.90	9.00	2.2



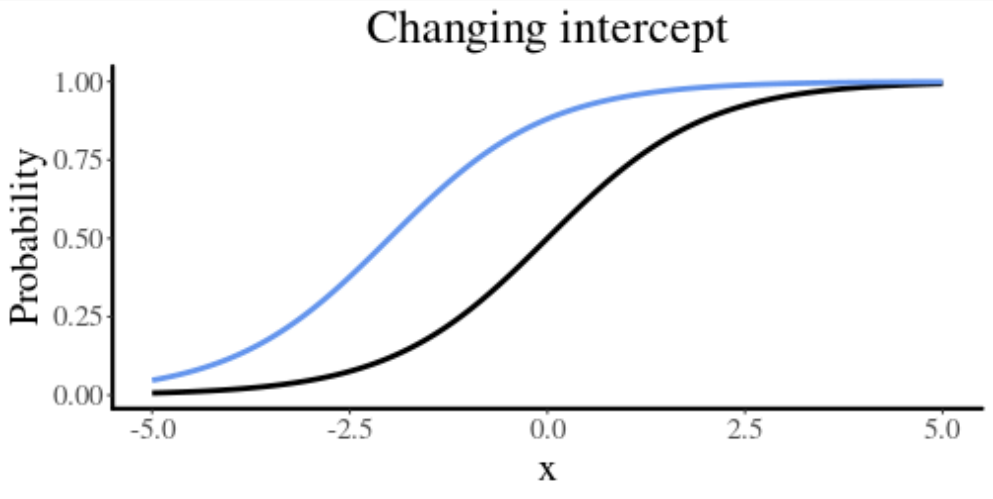
Interpreting parameters in logistic

For a continuous predictor variable, x_1 , the regression coefficient, β_1 , represents the change in log-odds per unit change in x_1 holding other predictors constant.



Interpreting parameters in logistic

The intercept determines the probability at $x = 0$: $p = \frac{e^{\beta_0}}{1+e^{\beta_0}} = ?$

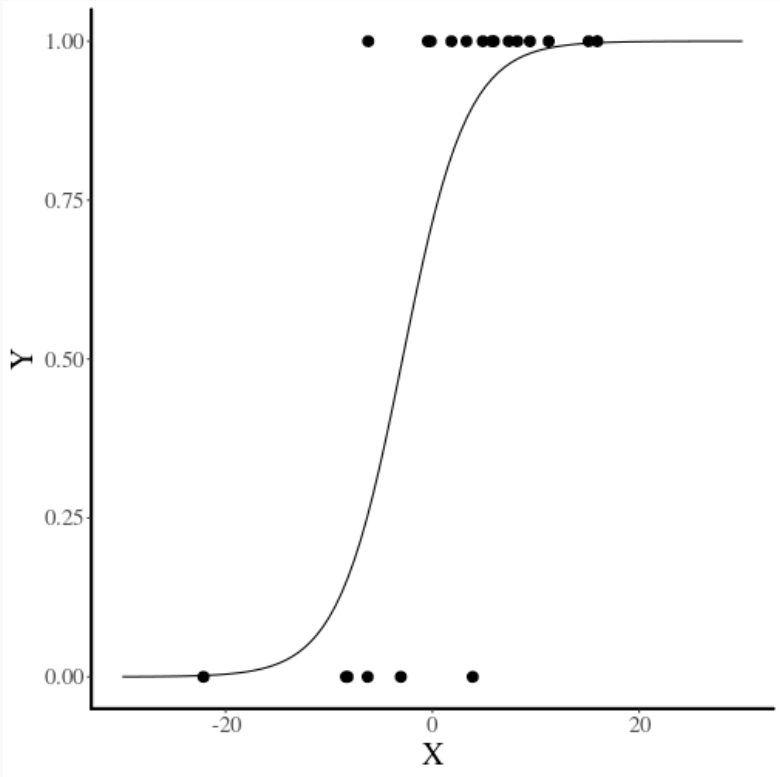


Response data in logistic regression

We can have response data of 0 or 1...

```
glm.fit  ← glm(y ~ x, data=sim.dat, family=binomial)
```

x	y
-6.264538	0
1.836433	1
-8.356286	0
15.952808	1
3.295078	1
-8.204684	0
4.874290	1
7.383247	1
5.757814	1

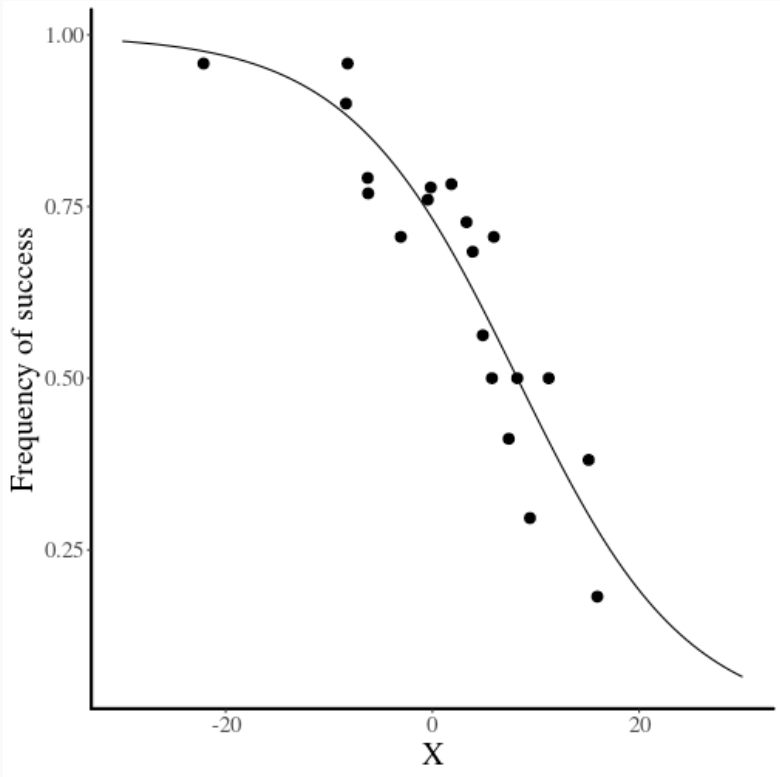


Response data in logistic regression II

... or the response data can be a vector of successes and failures

```
glm(cbind(successes, fails) ~ x, data=sim.dat, family=binomial)
```

successes	fails	x
19	5	-6.264538
18	5	1.836433
18	2	-8.356286
2	9	15.952808
16	6	3.295078
23	1	-8.204684
9	7	4.874290
7	10	7.383247
9	9	5.757814

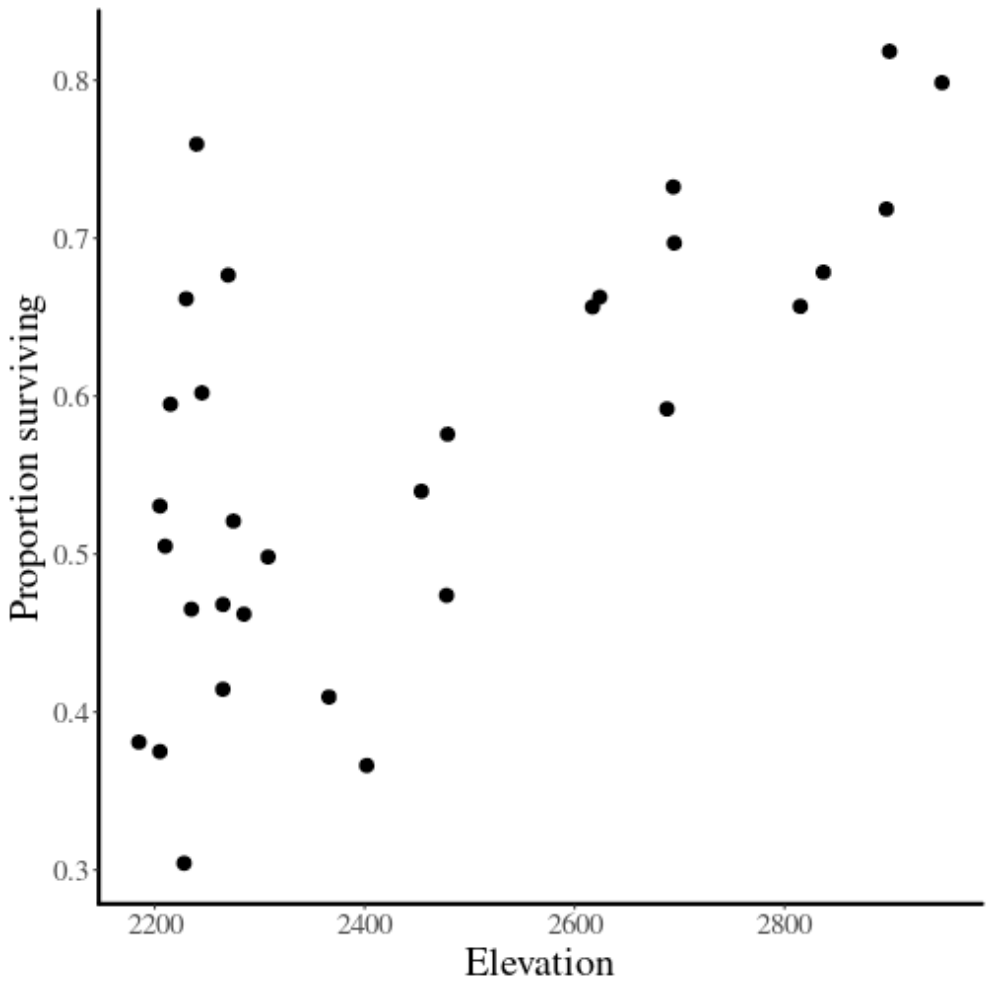


Example: Haleakalā silverswords

Krushelnycky, P.D., et al, 2013. Climate-associated population declines reverse recovery and threaten future of an iconic high-elevation plant. Global Change Biology, 19(3)

Image: wikimedia

Survival

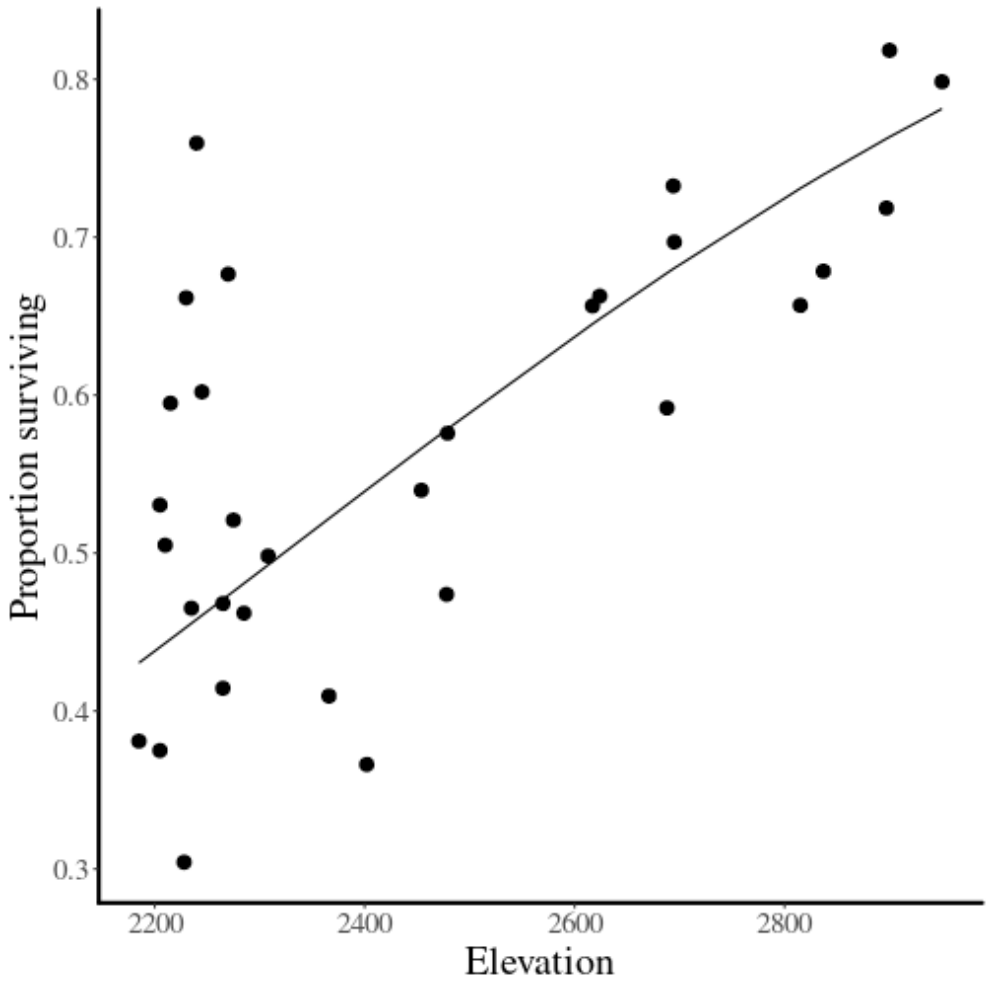

Proportion surviving
0.8
0.7
0.6
0.5
0.4
0.3
2200
2400
2600
2800
Elevation

Modeling survival with `glm`

```
##
## Call:
## glm(formula = cbind(Survive, Die) ~ Elev, family = binomial,
##     data = silver.dat)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10.9712   -2.6427    0.1903    3.5485   12.7223
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.717e+00  1.410e-01  -33.45   <2e-16 ***
## Elev         2.031e-03  5.949e-05   34.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2074.36  on 30  degrees of freedom
## Residual deviance:  794.65  on 29  degrees of freedom
## AIC: 1005.2
##
## Number of Fisher Scoring iterations: 3
```

How did we do?

$R^2_{pseudo} = 0.62$



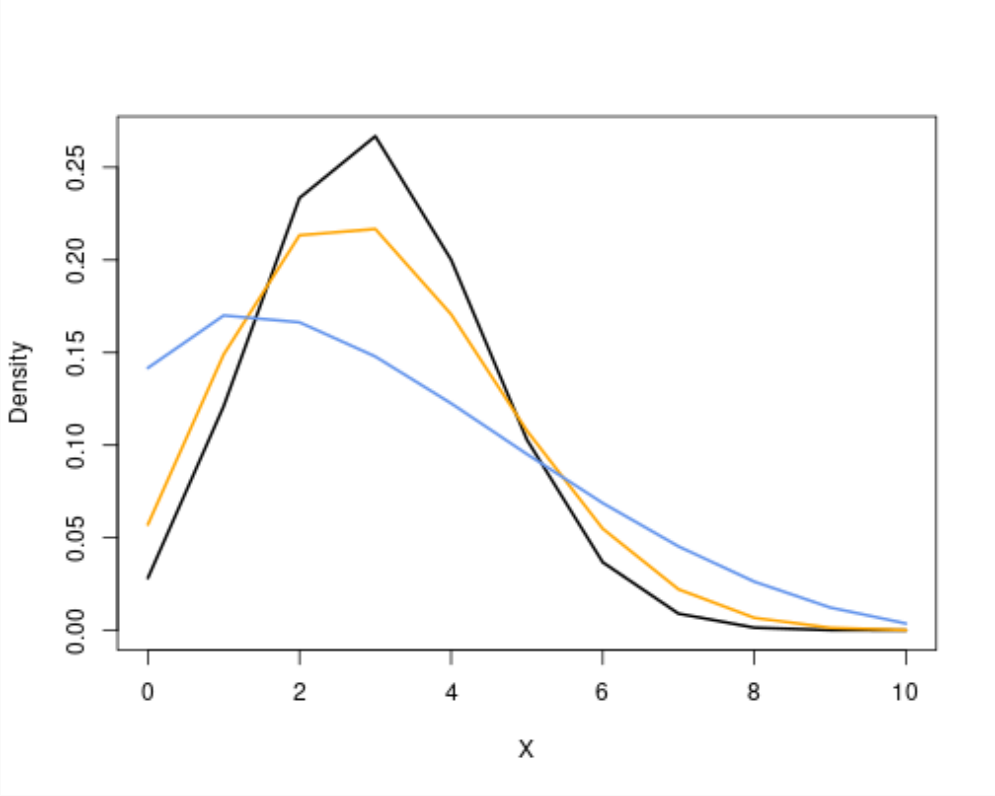
Exercise 7C

Modeling categorical data

What about overdispersion?

The beta-binomial distribution allows for overdispersion

$$E[Y] = N\mu,$$
$$Var[Y] = N\mu\,(1-\mu) + \rho N(N-1)\mu\,(1-\mu)$$



Beta-binomial regression

```
library(VGAM) #contains beta-binomial regression
model.bin ← glm(cbind(Survive, Die) ~ scale(Elev), data=silver.dat, family=binomial)
model.bb ← vglm(cbind(Survive, Die) ~ scale(Elev), data=silver.dat, family=betabinom

summary(model.bb)
```

```
##
## Call:
## vglm(formula = cbind(Survive, Die) ~ scale(Elev), family = betabinomial,
##     data = silver.dat)
##
## Pearson residuals:
##                      Min      1Q  Median     3Q   Max
## logitlink(mu)  -1.8795 -0.5691 -0.1327 0.5302 2.663
## logitlink(rho) -0.7086 -0.6495 -0.4718 0.1017 4.234
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  0.27837    0.07451   3.736 0.000187 ***
## (Intercept):2 -3.21794    0.26498 -12.144  < 2e-16 ***
## scale(Elev)    0.37971    0.07861   4.830 1.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing parameter estimates

Binomial model

```
coef(model.bin)
```

```
## (Intercept) scale(Elev)
##   0.2459422   0.5098857
```

```
confint(model.bin, method="profile")
```

```
##                  2.5 %    97.5 %
## (Intercept) 0.2203443 0.2716179
## scale(Elev) 0.4807028 0.5392651
```

Beta-binomial model

```
coef(model.bb)
```

```
## (Intercept):1 (Intercept):2   scale(Elev)
##     0.2783720    -3.2179363     0.3797051
```

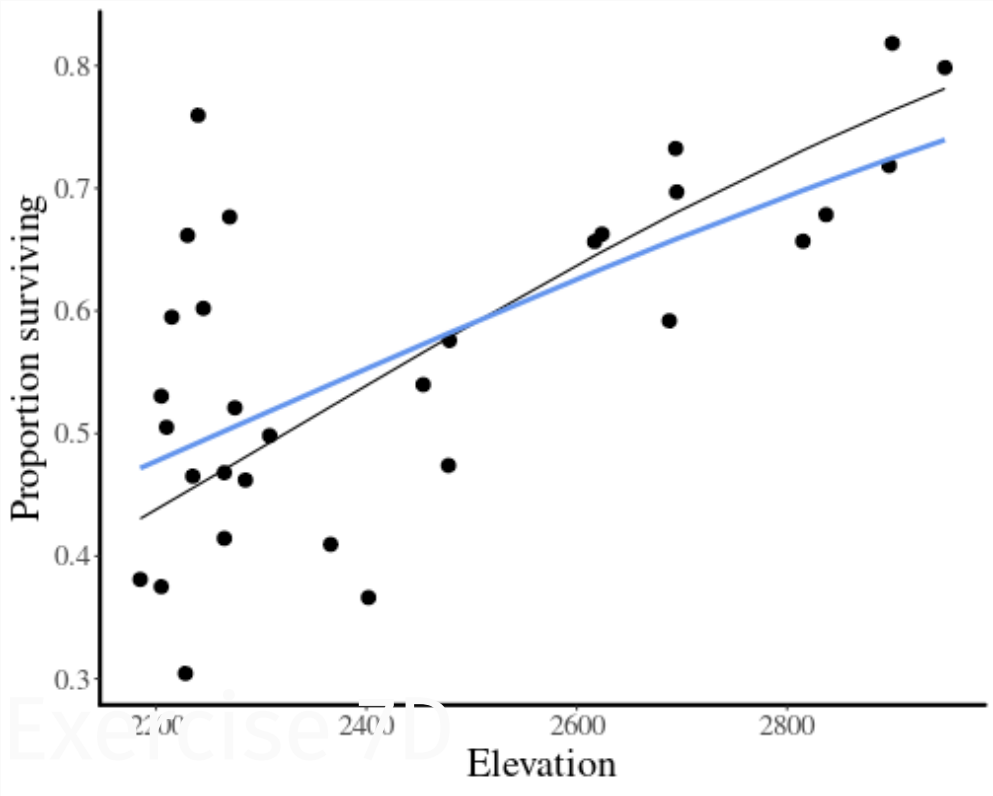
```
confintvglm(model.bb, method="profile")
```

```
##                      2.5 %     97.5 %
## (Intercept):1  0.1278870  0.4282708
## (Intercept):2 -3.7124930 -2.6589521
## scale(Elev)    0.2237321  0.5373084
```

Comparing model fits

```
print(c(AIC(model.bin), AIC(model.bb)))
```

```
## [1] 1005.151  341.455
```



Summary

- GLM's extend the application of LM's beyond the normal distribution.
    - Often useful with biological data
- Standard distributions in `glm` cannot handle overdisperison.
- Overdispersion can have strong influences on SE's and p-values
    - Not accounting for overdispersion can lead to overconfidence in estimates and model.