Lab 9: Generalized Linear Models

ENVIRON 710: Applied Statistical Modeling*

Generalized linear models (GLM's) are an extension of regular linear models (e.g., multiple linear regression, ANOVA, ANCOVA) to situations where the probability model is not a normal distribution. As in linear regression, we are interested in the relationship between a response variable, Y, and a set of predictors, X. The response probability distribution can be any member of the exponential family of distributions, which contains the normal distribution, the binomial distribution, and the Poisson distribution, among others. A nonlinear link function relates the mean of the response linearly to a set of terms based on predictor variables. Whereas linear regression was solved by ordinary least squares, GLM's are fit by a process called *iteratively reweighted least squares*, which overcomes the problem that transforming the data to be linear also changes the variance.

In this lab, we will discuss Poisson, logistic, and binomial regression. The learning goals of the lab are to:

- understand when to use Poisson, logistic, and binomial regression;
- learn to implement GLMs in R, including special topics like offsets for modeling rates;
- conduct model selection to find the minimum adequate model for a dataset and question of interest;
- interpret GLM coefficients and make predictions from the models.

At the end of the lab, there are a few problems to answer. *Submit your answers in R Markdown to the class Sakai site under the Assignments folder*

More functions in R

- `glm()` - fits GLM's to data, with the user specifying the probability distribution, called the family (e.g. `family = poisson` or `family = binomial`).
- `logLik()` - generates the log-likelihood of fitted models.
- `AIC()` – generates the Akaike Information Criterion for one or several fitted models.
- `anova()` – computes an analysis of deviance table to evaluate fits of GLM's and carry out model comparisons. The function provides different test statistics. For models with known dispersion, the Chi-squared test or likelihood ratio test (LRT) test is most appropriate. For models where the dispersion is estimated (Gaussian, quasibinomial, quasipoisson), the F-test is most appropriate.
- `lrtest()` – a general function from package `lmtest` for carrying out likelihood ratio tests that compares nested GLM's.
- `offset()` - including an offset allows modeling of a rate (e.g. number of birds per area), rather than just a count. For example, it allows the mean Poisson variable, μ, to be divided by effort, t: e.g. $log(\mu/t) = \beta_0 + \beta_1 X$ which is equivalent to $log(\mu) = \beta_0 + \beta_1 X + log(t)$.
- `deviance()` - returns the deviance of a fitted model object.
- `inv.logit()` - given a numeric object, this function from the `boot` package returns the inverse logit of the values.

*Created by John Poulsen with edits from TAs.

Poisson Regression

In Poisson regression the response variable consists of count data. Because count data must consist of positive integers, it is often assumed to have a Poisson distribution:

$$Y_i \sim Poisson(\lambda_i).$$

The Poisson distribution variable λ_i is modeled by

$$log(\lambda_i) = X_i\beta.$$

The mean response is related to the predictor variables through a 'log link'. And thus, the logarithmic expected number of incidents is modeled by the linear function of potential predictors. So, for example, $log(y_i) = \beta_0 + \beta_1 X_i$ or $y_i = exp^{(\beta_0+\beta_1 X_i)}$. This model is used to answer questions related to responses like (a) the number of cargo ships damaged by waves, (b) daily homicide counts in California, or (c) the environmental drivers of numbers of birds along an altitudinal gradient.

Squirrel behavior

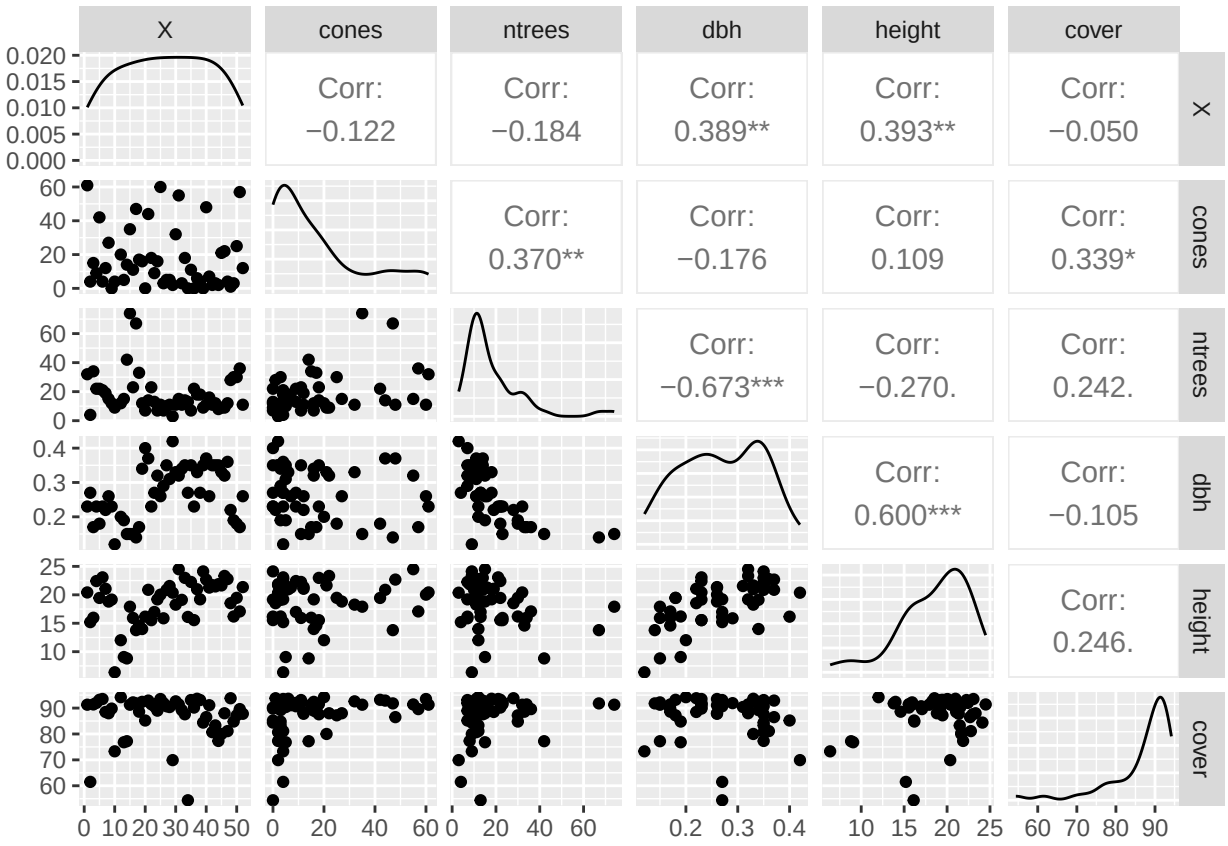
To explore Poisson regression, let's use the squirrel data set (nuts) from Zuur, Hilbe, and Ieno (2013). As originally reported by Flaherty et al (2012), researchers recorded information about squirrel behavior and forest attributes across various plots in Scotland's Abernathy Forest. In this analysis, we are going to analyze the factors that determine the number of cones (*cones*) stripped by red squirrels. The data were collected in plots with different numbers of trees (*ntrees*). Recorded variables include: mean DBH per plot (*dbh*), mean tree height per plot (*height*) and percentage canopy closure (*cover*). The stripped cone count was only taken when the mean diameter of trees was under 0.6m (dbh).

Note that the number of stripped cones are *counts*, thus it is appropriate to use GLM's with a Poisson probability distribution.

```
nuts <- read.csv("nuts.csv", header = T)
```

Take a look at the data using `ggpairs()` and `summary()`.

```
ggpairs(nuts)
```




```
summary(nuts)
```

As always, we need to inspect this plot to get a "feel" for our data. There are a couple of things to notice in this plot. First, `cones` ranges from 0 to 60, which is appropriate because the dependent variable needs to be a non-negative integer for Poisson regression. It also not normally distributed, which we don't expect it to be. Second, all the variable are continuous - no categorical independent variables. Third, there do not seem to be linear relationships between `cones` and the other variables in the raw data - that is fine too, because that is not an assumption of Poisson regression.

Let's build a full model to assess the explanatory variables that determine the number of cones stripped by squirrels. The number of cones counted in a plot likely varies with the number of trees in a plot, therefore we want to model the number of cones per tree. To do so, we need to add an offset so that the counts can be interpreted relative to some baseline or 'exposure'. In other words, we want to interpret the response variable as a rate, number of stripped cones per tree.

```
nut1 <- glm(cones ~ dbh + height + cover, offset = log(ntrees),
            family = poisson, data = nuts)
summary(nut1)
```

In other applications, we could add an offset to account for different levels of effort (e.g. hours of counting birds or plot sizes for counting a rare plant). The offset term has to be logged in the Poisson model (see the explanation for `offset` above).

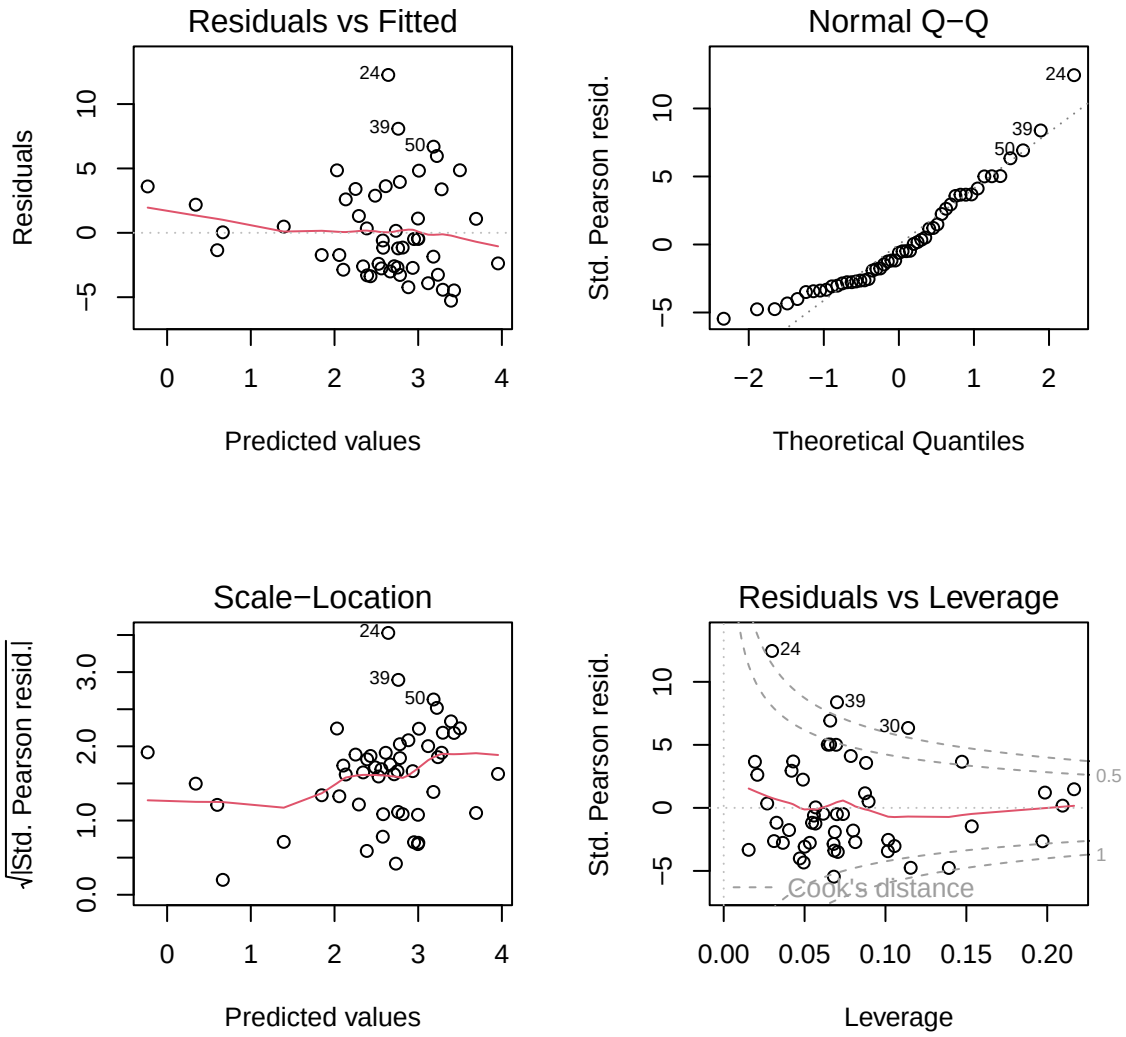
All the terms in this model are statistically significant. To understand the coefficients, we exponentiate them and interpret them as multiplicative effects.

```
(coef <- exp(coef(nut1)))
```

The intercept is the prediction if $X_1 = 0$, $X_2 = 0$, and $X_3 = 0$. After exponentiating the intercept, the mean number of stripped cones is 9.8e-04, which is the average number of cones per tree stripped by squirrels. This is a very small number. The coefficient of X_1 is the expected difference in y (on the logarithmic scale) for every additional meter of dbh. After exponentiating the coefficient, $e^{4.413} = 82.53$, the number of cones increases by a factor of 82.5 for every additional meter of dbh in a tree plot. We could interpret the effects of mean tree height, $e^{0.0373} = 1.038$, and percent cover, $e^{0.0537} = 1.055$, in the same way: for every additional meter of tree height the number of stripped cones increases by roughly 3% and for every additional percentage of tree cover the number of stripped cones increases by roughly 6%.

Under the Poisson distribution, the variance is assumed to be equal to the mean. If this is true, and the mean number of stripped cones is large (which it is at 17.9 cones) then the residuals should be independent and normally distributed. Overdispersion is the case where the variance is much greater than the mean. With overdispersion, we expect the residuals to be much larger, reflecting the extra variation beyond what is predicted under the Poisson model. One sign of overdispersion is that the residual deviance is much higher than the residual degrees of freedom (which is true of `nut1` - the residual deviance is 665.3 and the residual degrees of freedom is 48). We can use the typical residual plots to verify.

```
par(mfrow=c(2,2))
 plot(nut1)
```



As you can tell from the figures, the data are very overdispersed! A couple of problems can be detected from these figures: 1. in the scale-location plot it looks like more than 5% of the standardized residuals are greater than 2; and 2. in the leverage plot, a few observations (#11, #31) might be outliers. Let's first check the possibility of overdisperion before we mess with outliers.

For technical reasons that we won't explore in this class, overdispersion is calculated with the Pearson's χ^2 statistic:

$$\chi^2_p = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{var(y_i)}$$

and

$$\hat{\phi} = \frac{\chi^2_p}{n-p}$$

We can calculate and check for overdispersion using the `dispersiontest` function from the `AER` package.

```
dispersiontest(nut1)
```

An overdispersion ratio of 2 is considered high, so this is out of the ballpark! To handle overdispersion, we can use the quasipoisson "distribution". This is not really a distribution. Rather, all the regression standard errors are rescaled through multiplication by the square root of the overdispersion $\sqrt{261} = 16.2$. Let's run the model.

```
nut2 <- glm(cones ~ dbh + height + cover, offset = log(ntrees),
            family = quasipoisson, data = nuts)
summary(nut2)
```

Note that the quasipoisson model doesn't change the coefficient estimates, but increases the standard errors, reduces the statistics, and increases the p-values (making them more conservative). Now tree height is no longer statistically significant and canopy cover is marginally significant.

There are a couple of other ways to deal with overdispersion, including adding an observation-level random effect or employing the negative binomial distributon, which has an extra parameter for estimating dispersion.

Let's run a negative binomial GLM which has an extra parameter, *theta* to account for overdispersion. The negative binomial distribution, like the Poisson distribution, describes the probabilities of the occurrence of whole numbers greater than or equal to 0. The negative binomial is a generalization of the Poisson distribution that does not require the variance and the mean to be equivalent.

Unfortunately, the `glm.nb` function has to be specified differently than our previous models. We don't, for example, have to include the family and the offset is included as `+ offset(log(ntrees))`

```
nut3 <- glm.nb(cones ~ dbh + height + cover + offset(log(ntrees)),
               data = nuts)
 summary(nut3)
```

In this model, we retain both `dbh` and `cover` as important explanatory variables. Let's reduce the model.

```
nut4 <- update(nut3, .~.-height)
 summary(nut4)
```

Maybe we've found our final model? Notice that the residual deviance and residual degrees-of-freedom are much more similar than before. Don't try to check the overdispersion again, because by definition the negative binomial cannot be overdispersed.

Let's compare all our models with the Akaike Information Criterion.

```
AIC(nut1, nut2, nut3, nut4)
```

Notice that `nut2` does not have an AIC. The quasipoisson is not a probability distribution, doesn't have a likelihood, and therefore cannot have an AIC score. It doesn't matter, though, because we have taken care of overdispersion with the negative binomial distribution. Our final model has the lowest AIC.

Our models are all nested models, thus we can also compare them with the likelihood ratio test from the `lmtest` package. This gives us the same result.

```
lrtest(nut1, nut2, nut3, nut4)
```

The Poisson and negative binomial model are nested: Poisson is a special case with $\theta = \infty$. So a likelihood ratio test comparing the two models is testing the null hypothesis that $\theta = \infty$ against the alternative that $\theta < \infty$. Comparing `nut1` and `nut3`, the fitted log-likelihood in the negative model is much larger/better using just one additional parameter (3 regression coefficients plus θ). The value of `nut3$theta` is 1.022, clearly less than ∞.

Let's test the goodness of fit of the model, using the chi-square goodness-of-fit test. We compare the residual deviance of the reduced model to a chi-square distribution with 49 degrees-of-freedom. It is not statistically significant, so we have a good model! .

```
pchisq(nut4$deviance, nut4$df.residual, lower.tail=F)
```

We can calculate a pseudo-R^2 for GLMs to evaluate their goodness-of-fit. In linear regression, R^2 represents the proportion of variance that is explained by the predictors. In GLMs, several pseudo-R^2 have been developed, each with their limitations, but none represent the proportionate reduction in error. A pseudo-R^2 only has meaning when compared to another pseudo-R^2 of the same type, on the same data, predicting the same outcome. In this situation, the higher pseudo-R^2 indicates which model better predicts the outcome.

Veall and Zimmermann[1] concluded that from a set of six widely used measures the measure suggested by McKelvey and Zavoina had the closest correspondance to ordinary least square R^2. The Aldrich-Nelson pseudo-R^2 with the Veall-Zimmermann correction is the best approximation of the McKelvey-Zavoina pseudo-R^2. Efron, Aldrich-Nelson, McFadden and Nagelkerke approaches severely underestimate the "true" R^2.

Therefore, we can use the `DescTools` package to run `VeallZimmermann` pseudo-R^2.

```
DescTools::PseudoR2(nut3, c("VeallZimmermann"))
```

```
## VeallZimmermann
##       0.1579536
```

Logistic Regression for Binary Responses

In logistic and binomial regression, we need to distinguish between two cases of models both of which use the binomial distribution. The first is the case where the response variable is binary (presence/absence or 1/0). The second is the case where the responses are counts and each value represents the number of 'successes' observed in a specified number of trials. In this case, we model the number of successes in a series of independent Bernoulli trials, where each trial has the same probability of success. Logistic regression answers questions such as: "Which indicators of health describe whether humans are infected (infection = yes/no) with malaria?" Binomial regression answers questions like, "what predictors determine the proportion of death penalty verdicts that were overturned in each of 50 states".

Bumpus sparrows

Let's try a logistic regression model on the Bumpus sparrow data. House sparrows were found on the ground after a severe winter storm in 1898, some of which survived and others of which perished. We are going to analyze the explanatory variables that determine whether a sparrow survived or not. Although there are lots

[1] Veall, M.R., & Zimmermann, K.F. 1992. Evaluating Pseudo-R2's fpr binary probit models. Quality & Quantity, 28, pp. 151-164.

of different morphometric data, we will limit our analysis to bird weight, `WT`, and age, `AG`. Age is coded as adult (1) or young (2). If you are interested in trying the other variables, they include: total length, `TL`, alar extent, `AE`, length of beak and head, `BH`, length of humerus, `HL`, length of femur, `FL`, length of tibio-tarsus, `TT`, width of skull, `SK`, and length of keel of sternum, `KL`.

After loading the data, we have to transform the `Status` column into a numerical variable, `SV`, where 1 = survived and 0 = died. Note that you will need the `boot`, `Sleuth3`, and `ResourceSelection` packages, so install and require them.

```
sdat<- ex2016
sdat$SV <- ifelse(sdat$Status == "Survived", 1, 0)
```

Fit the full model

Plot the data with `ggpairs()`. Even though we will focus on weight and age, you will notice that many of the variables are highly correlated – which is expected with morphological measurements. Here we run three models - a full model with an interaction, a main effects model, and a model with just one main effect. Note that we set the probability distribution of the model to binomial for logistic regression (`family=binomial`).

```
lr1 <- glm(SV ~ factor(AG)*WT, family=binomial, data=sdat)
 lr2 <- glm(SV ~ factor(AG)+WT, family=binomial, data=sdat)
  lr3 <- glm(SV ~ WT, family=binomial, data=sdat)
```

Compare nested models

As with Poisson Regression, we will use the Likelihood Ratio Test to compare nested models. What is your conclusion from this test?

```
anova(lr1, lr2, lr3, test="Chisq")
```

We could also use AIC to select the best model.

```
AIC(lr1, lr2, lr3)
```

The most parsimonious model, `lr3`, is the minimum adequate model. Removing the interaction term `AG x WT` and then `AG` did not strongly affect the model. The AIC comparison also shows `lr3` to have the lowest AIC. `lr2` is not more than 2 AIC points from `lr3`, indicating that they fit the data equally well. Here we will work with the most parsimonious model.

Diagnose model fit

The χ^2 test of deviance shows evidence for lack-of-fit of the model to the data. In other words, the fitted values are significantly different from the observed values. Goodness-of-fit tests are based on the premise that the data will be divided into subsets and within each subset the predicted number of outcomes will be computed and compared to the observed number of outcomes. The H_0 is that the observed values are not different from the fitted values.

```
pchisq(lr3$deviance, lr3$df.residual, lower=F)
```

We could also calculate the Hosmer-Lemeshow goodness of fit test. The test assesses whether or not the observed rates of survival match the expected rates of survival in subgroups of the model population. The result is similar to the test of deviance.

```
hoslem.test(sdat$SV, fitted(lr3))
```

We are going to ignore the lack of fit for now, mostly because we don't have many options with a model that only has one variable. The results indicate that there is still extra variation to be explained, and so perhaps we are missing an important predictor (one of the many others in the data set).

We can calculate a pseudo-R^2 for logistic regression as we did above. It supports our conclusion that this model is not very informative.

```
DescTools::PseudoR2(lr3, c("VeallZimmermann"))
```

Interpret parameters

The coefficients of the fitted model can be interpreted in two different ways – as log-odds and odds. The coefficients provided by `glm()` are log-odds. By taking the anti-log of the coefficients, we can interpret them as odds ratios.[2]

In our example, a 1-unit difference in weight corresponds to a multiplicative change of $e^{-0.42} = 0.657$ in the odds of survival. So the coefficient of `WT` can be interpreted as: "for every gram of additional weight the odds of survival change by a factor of 0.65 (or decrease by 35%)". Another way of saying this is that for every additional gram of weight, the odds of failure increase by 1.5 times.

```
1/exp(lr3$coef[2])
```

We can take a look at the above answer by substituting in values for weight. The equation from our model is:

$$log(\frac{p}{1-p}) = logit(p) = 11.3201 - 0.4244 \cdot WT$$

The mean weight is approximately 25 grams, so let's look at the effect of a change from 25 to 26 grams on the odds of survival.

```
cf <- coefficients(lr3)
 (cf[1]+cf[2]*26)-(cf[1]+cf[2]*25)
```

For a one-unit increase in weight, the expected change in log-odds is -0.424. By taking the anti-log, we get a change in odds of 0.65, or a 35% decrease in odds of survival for each additional gram of weight.

We can then calculate the probability of survival for a specific weight or weights. To convert log-odds to probabilities, we use the inverse logit: $1/(1 + e^{-x})$. In R, the `inv.logit` function will calculate the probabilities for us. For example, the probability of survival for a bird weighing 30 grams is 19.6%. Note that to predict the response for a specific value of the covariates, the inverse logit is performed on the entire equation.

```
coef.prob1 <- c(1/(1+exp(-cf[1])), 1/(1+exp(-cf[2])))
 coef.prob1 <- inv.logit(coef(lr3))

 bird.30 <- inv.logit(cf[1] + cf[2]*30)
 bird.31 <- inv.logit(cf[1] + cf[2]*31)

 bird.31 - bird.30
```

We can evaluate how a difference in 1 gram of weight would affect the probability of survival between birds that are 30 and 31 grams. The probability of survival of a bird weighing 31 grams is 13.7%, compared to 19.6% for a 30 gram bird. This is a change of 5.8% in survival.

[2] If an outcome has a probability p, then $p/(1-p)$ is the odds of the outcome.

We could quickly find the change in probability of survival over the possible range of bird weights with the following code. What do you notice?

```
newdat <- data.frame(WT = c(23:31))
newdat$predSV <- predict(lr3, newdat, "response")
newdat$changeSV <- c(0, newdat$predSV[1:8] - newdat$predSV[2:9])
```

Finally, we can measure model accuracy by determining the proportion of observations that have been correctly classified as 'survived' from the model. Line `problr3` below extracts the probability of survival over all the observed weights in the raw data based on our simple model. If the probability is greater than 0.5, then it signifies that the bird survived. `pred.classes` codes the probabilities as 1 (survived) or 0 (perished) based on the probabilities, and then we evaluate the number of times predicted survival was equal to the observed survival. The proportion of correctly classified observations is ~59.8%, which is not great.

```
problr3 <- lr3 %>% predict(sdat, type = "response")
pred.classes <- ifelse(problr3 > 0.5, 1, 0)

mean(pred.classes == sdat$SV)
```

Just to see if a more complicated model would improve model goodness of fit, pseudo-R^2 and accuracy, let's build a more complicated model and run all of those assessments again. What do you find?

```
lr4 <- glm(SV ~ WT + TL + AE + TT + SK, family = binomial, data = sdat)
 DescTools::PseudoR2(lr4, c("VeallZimmermann"))
  hoslem.test(x = sdat$SV, y = fitted(lr4))

problr4 <- lr4 %>% predict(sdat, type = "response")
pred.classes <- ifelse(problr4 > 0.5, 1, 0)

mean(pred.classes == sdat$SV)
```

In summary:

- As with linear regression, the intercept is estimated assuming zero values for the other predictors. The intercept is not usually interpreted in a logistic regression model.
- A difference of 1 gram in weight corresponds to a negative difference of 0.424 in log-odds of bird survival. By taking the anti-log of -0.424, the change can be interpreted as a change in odds.
- We can predict the probability of survival for different scenarios (e.g. birds at different weights). This is done by fitting our logistic regression equation with different values of the predictors and then taking the inverse logit. Note that it is difficult to interpret the slope coefficients (e.g. weight) in terms of the change in probability with each one-unit change in the predictor variable because the relationship is non-linear (see below figures).
- Model fit can be assessed in several ways, including the test of deviance, pseudo-R^2, and the Hosmer-Lemeshow statistic.
- Model accuracy can be assessed by predicting the proportion of times the model currently predicts observed survival.

Plotting Results

For most people, probabilities are more intuitive than odds, so let's plot the data and the probability of survival over weight of the birds. The fitted function provides the probabilities from the model by calculating the inverse of the logit function (the logistic function). If $logit(\pi) = \eta$, then $\pi = e^{\eta}/(1 + e^{\eta})$. These are the expected probabilities ($\hat{y}$) of the observed data given the model.

```
fitted(lr3)
```

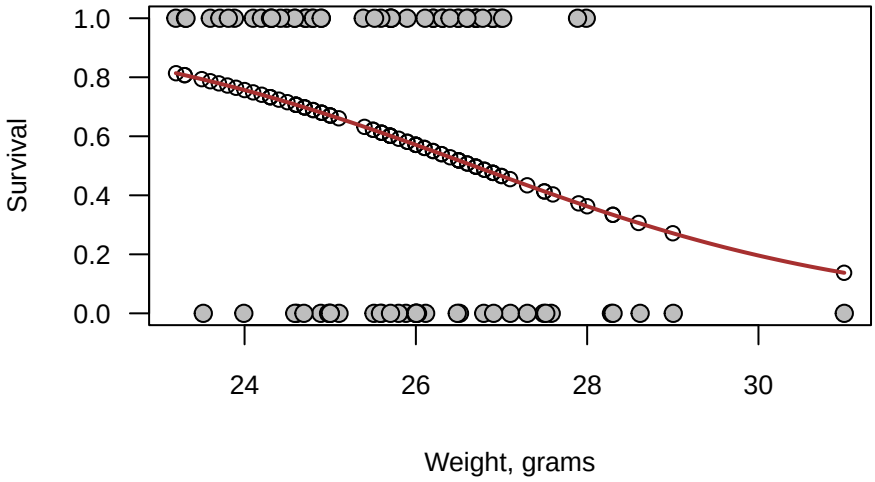
The `predict()` function returns predictions from a new set of predictor variables. If you don't specify a new

set of predictor variables, then it will use the original data giving the same results as fitted for some models.[3] `predict` returns the fitted values *before* the inverse of the link function is applied (to return the data to the same scale as the response variable), and `fitted` shows it *after* it is applied. To get results on the original (response) scale, you must use `predict(model, type = "response")`.

```
predict(lr3, type = "response")
```

First, we will plot the raw data and the fitted values from the model. To add the fitted curve to the plot we first define x as a range of values from the minimum bird weight to the maximum weight. Then we fit the curve, using `inv.logit()` to transform log-odds to proportions to fit our plot.

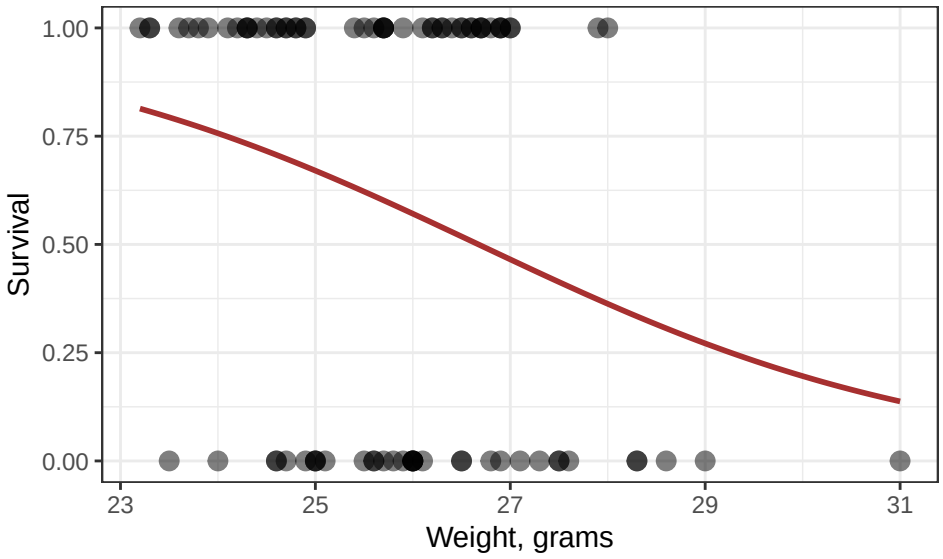
```
plot(jitter(sdat$WT), sdat$SV, las=1, pch=21, cex=1.2, bg="grey",
     xlab = "Weight, grams", ylab="Survival", cex.axis = 0.8,
               cex.lab = 0.8)

 x <- seq(min(sdat$WT), max(sdat$WT), length = 100)
  points(sdat$WT, fitted(lr3))

 curve(expr = inv.logit(lr3$coef[1] + lr3$coef[2]*x), add=T,
       lwd=2, col= jcoPalette[9])
```



Or, we could use `ggplot()`, which does all this for us.

```
ggplot(sdat, aes(x=WT, y=SV)) +
  geom_point(alpha=.5, size = 3) + labs(x = "Weight, grams", y = "Survival") +
   stat_smooth(method="glm", se=FALSE, method.args = list(family=binomial),
               col= jcoPalette[9]) +
  theme_bw()
```

[3]Note that by default predict() calculates the log-odds, $logit(\pi)$, by sticking in values of the predictor variables, e.g. X_1, into the equation: $\beta_0 + \beta_1 X_1$.



Logistic Regression for Count Data

Sometimes, we may have count data that need to be treated as proportions rather than simple counts, such as when we have a number of "successes" out of a total count (or number of trials). Examples include:

- Proportion of planted seeds that recruited into seedlings
- Proportion of death penalty verdicts overturned
- Proportion of population that survived a disease.

The model, called a logistic-binomial model, is used in settings where each data point represents the number of successes in some number of trials.

Let's look at an example where randomly chosen people were asked to respond to the following statement regarding the role of women in society: "Women should take care of running their homes and leave the running of the country up to men". Install and load the `HSAUR` package to get the data.

The observations have been grouped into counts of number of respondents who `agree` with the statement and number who `disagree` with the statement. The `sex` and years of `education` of the respondents are the predictor variables. To fit a binomial regression model to grouped data using the `glm` function, we need to specify the number of agreements and disagreements as a two-column matrix on the left hand side of the model formula with `cbind()`.

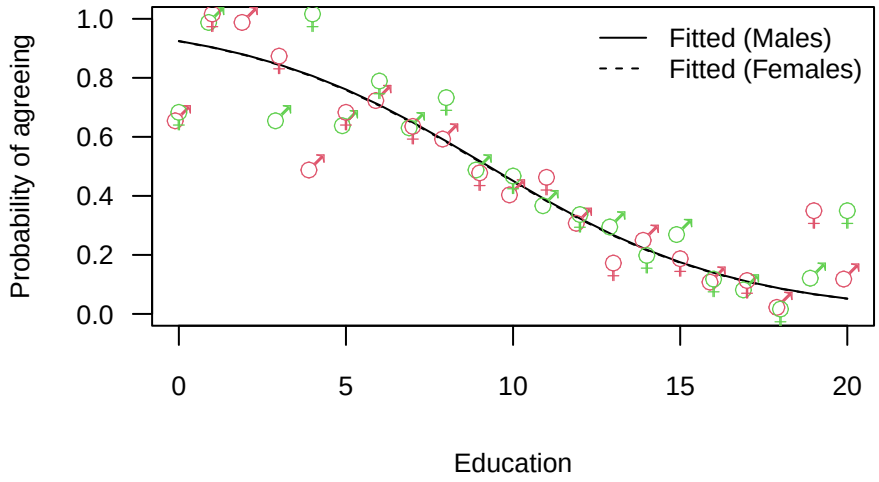
```
logreg1 <- glm(cbind(agree, disagree) ~ factor(sex) + education,
               data = womensrole, family = binomial)
```

From the summary of `logreg1`, `education` significantly predicts whether a respondent will agree with the statement, but `sex` is unimportant. As a respondent's years of education increase, his/her probability of agreeing with the statement decreases. While using the z-value of the coefficient on `sex` is enough to determine that we don't need it in the model, we could also verify by running the model with and without `sex` and looking at the change in deviance.

```
logreg2 <- glm(cbind(agree, disagree) ~ education,
               data = womensrole, family = binomial)
 anova(logreg1, logreg2, "Chisq")
```

The change in deviance from removing `sex` is very small (<1) confirming our previous conclusion. Let's look at how the probability of agreeing to the statement varies over education level for both men and women.

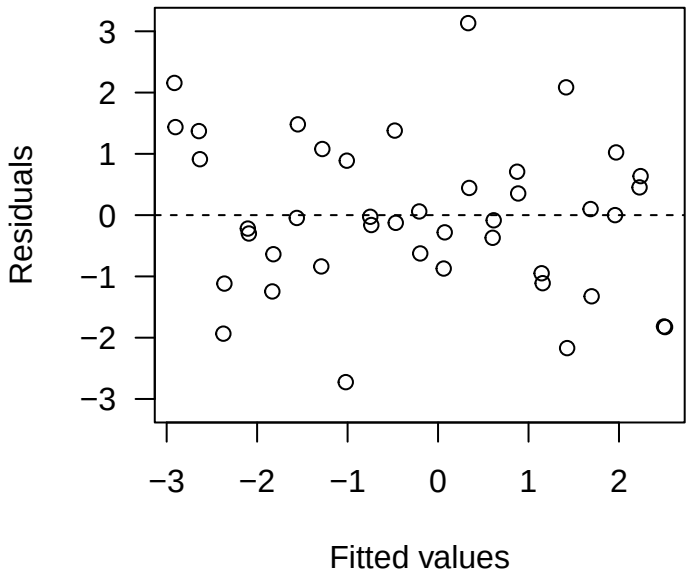
```
lr1.fitted <- predict(logreg1, type = "response")
 f <- womensrole$sex == "Female"

plot(womensrole$education, lr1.fitted, type = "n",
     ylab = "Probability of agreeing",
      xlab = "Education", ylim = c(0,1), las = 1, cex.axis = 0.8,
              cex.lab = 0.8)

lines(womensrole$education[!f], lr1.fitted[!f], lty = 1)
 lines(womensrole$education[f], lr1.fitted[f], lty = 2)
  lgtxt <- c("Fitted (Males)", "Fitted (Females)")
   legend("topright", lgtxt, lty = 1:2, bty = "n", cex = 0.8)

   y <-  womensrole$agree / (womensrole$agree +
                             womensrole$disagree)
text(womensrole$education, y, ifelse(f, "\\VE", "\\MA"),
     family = "HersheySerif", cex = 1.25, col = c(2,3))
```



We can also check out the fit of `logreg1` by using the typical diagnostics. Try `plot(logreg1)`. Note that these diagnostics are not helpful for assessing the fit of the binary logistic regression (our first example) because the data were 0's and 1's, whereas here the data are counts.

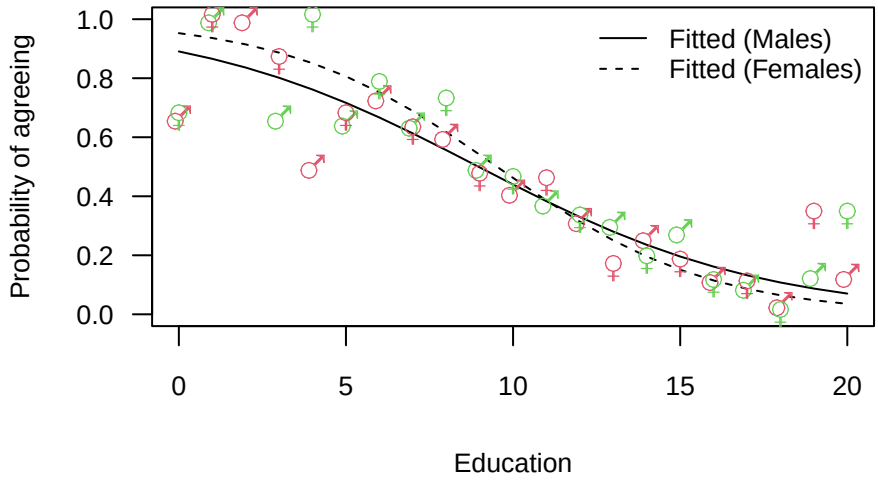
We can also create our own plot of deviance residuals plotted against fitted values using the below code. Most of the residuals fall into a horizontal band between -2 and 2 (standard errors). We expect 5% of the observations to fall outside of this band just by chance, so this pattern does not suggest a poor fit overall.

```
res <- residuals(logreg1, type = "deviance")
 plot(predict(logreg1), res, xlab="Fitted values", las = 1,
          ylab = "Residuals", ylim = max(abs(res)) * c(-1,1))
abline(h = 0, lty = 2)
```



As before with linear regression, we can also try to model the interaction between *sex* and *education*.

```
logreg2 <- glm(cbind(agree, disagree) ~ education*factor(sex),
               data = womensrole, family = binomial)
```



The interaction model leads to a deviance that is nearly 7 points lower than the main effects model. The interaction between `sex` and `education` is also statistically significant. We can plot this in the same way as above, but this time we see the effect of the interaction on the probability of agreeing to the statement.

To Do

How is this figure different from the above figure with the model without the interaction?

The interaction term can be understood in two ways. Looking from one direction, for a female, the value -0.08138 is added to the coefficient for `education` (-0.23403), and so we can see the interaction as saying that

the importance of education as a predictor increases for respondents that are female (i.e., it has an even stronger negative effect on the probability of agreeing).

Looking at the interaction from the other way, for each year of education the value -0.08138 is added to `sex` (0.90474 for a female). So we can understand the interaction as saying that the importance of `sex` as a predictor decreases as respondents gain education.

We can interpret the coefficients as log-odds or odds as before.

Note that like Poisson regression, when binomial regression is applied to count data it is possible for the data to have more variation than is explained by the model. (This was not the case for binary logistic regression, because the response variable was constrained to 0 and 1.) This overdispersion problem arises because the model does not have a variance parameter. As with the Poisson model, we can compute the estimated overdispersion and adjust our inference if the model is overdispersed. Here we implement the quasibinomial model.

```
logreg3 <- glm(cbind(agree, disagree) ~ education*factor(sex), data = womensrole,
               family = quasibinomial)
```

Then we can extract the dispersion estimate from the model and determine if the degree of overdispersion is significantly different from 1. Dispersion is 1.96, and the probability of getting that high of a dispersion is less than 0.001 given the model. Therefore we should report the results from `logreg3`, rather than `logreg2`.

```
pchisq(summary(logreg3)$dispersion * logreg3$df.residual,
       logreg3$df.residual, lower = F)
```

Problems

Your assignment is to answer the below questions. As always, turn in a R markdown document that includes your R code as an Appendix.

For each question, write a ~ 1-page description of your analysis, results, and inference. The write-up should include the following information:

- Null and alternative hypotheses of your test.
- Results of your model selection process. What is the minimum adequate model?
- Interpretation of your statistical test. Test whether your final model is a good fit to the data. Interpret the model coefficients, writing 2-3 sentences that include the appropriate reporting of the statistics.
- A description of how you checked the assumptions of your statistical test.
- A figure that demonstrates the results of your test/model. Specifically, the figure should show the (a) raw data, (b) curves that demonstrate the effect of bird-keeping over the range of one of the remaining significant predictors. Include a legend on the figure and make sure the axes are labeled appropriately.

Problem 1

Use the aircraft data, `AircraftDat.csv`, found on Sakai to evaluate the factors that led to damage of attack aircraft (bombers) during the Vietnam War. The database contains data from 30 strike missions involving two types of aircraft, the A-4 and the A-6. The regressor, x_1 is an indicator variable (A-4=0 and A-6=1), and the other regressors x_2 and x_3 are bomb load (in tons) and total months of aircrew experience. The response variable is the number of locations where damage was inflicted on the aircraft.

For this problem, develop models for every combination of variables, *excluding interactions*. In other words, run models that include all the IV's, each pair of IV's, and the individual IV's. Compare each of the models to the full model using the LRT.

Problem 2

Use the data from the package `Sleuth3`, `case2002`, to examine whether increased lung cancer is associated with bird keeping, even after accounting for factors such as age. Develop a logistic model with the log-odds of getting lung cancer explained by age `AG`, years the individual has smoked `YR`, the indicator variable for bird keeping `BK`, *and their interactions*. There are other variables in the database, which you do not need to consider. Specifically, test the hypothesis that bird keepers have higher rates of lung cancer than non-bird keepers. As requested above, plot the probability of having lung cancer for bird keepers and non-bird keepers. Finally, answer the following:

- What is the probability of having lung cancer if you are a bird keeper and have smoked for 32 years?
- What is the probability of the intercept? What does it represent exactly?

Problem 3

Mountaintop removal mining is a form of surface coal mining that involves removal of a hill or ridge to access buried coal seams. Mountaintop removal mining can have long-term environmental impacts. For example, a survey of 78 streams from valley fills found that 73 had water with sulfate concentrations greater than the threshold for toxic bioaccumulation. To test if groundwater is also contaminated, a study was conducted at 50 sites, with water samples taken from 20 wells at each site. The water samples were tested to determine whether they had higher than acceptable leves of mine-derived chemical constituents or not. The dataset can be found in *water.csv*, which provides the number of contaminated wells, Y, out of the total number of tested wells, N, at each site *site*. The study also recorded the depth of each well in feet, *well_depth*, and the distance of the well from the mine, *dist_mine*. Assess whether *well_depth*, *dist_mine*, and *site* affect the proportion of contaminated wells at a site.