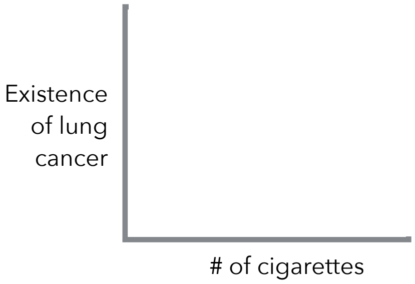
Lesson #13: Generalized Linear Models: Logistic, Poisson Regression

So far, we've fit linear models to predict continuous dependent variables. In this lesson, we'll learn how to use the **Generalized Linear Model** to predict outcome variables that are categorical (binary) or that represent counts.

1. Suppose we want to predict whether an individual will develop lung cancer. We'll model lung cancer as a function of the number of cigarettes smoked. Sketch a scatterplot we might expect from a large number of individuals.

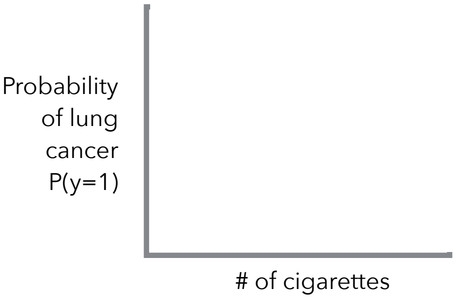


2. Suppose we fit a regression line to that scatterplot. Identify some problems we'd have with our linear model. Which assumptions (or conditions) of linear regression are violated?

$$\text{Linear model: } \text{cancer} = b_0 + b_1(\text{cigarettes})$$

Problems:

3. It looks like our linear regression techniques won't work for binary dependent variables. As an alternative, let's try to model the **probability of developing lung cancer** as a function of the number of cigarettes smoked. Using your intuition (and what you know about probabilities), fit a curve that you think might describe this relationship.



We know probabilities must always be positive and must range from 0-1.

A linear model $P(y=1) = b_0 + b_1 x$ is unbounded.

To ensure we get a positive result, you might think to try:

absolute values: $P(y=1) = |b_0 + b_1 x|$

squares: $P(y=1) = (b_0 + b_1 x)^2$

As we'll see, we're actually going to use an exponential function to ensure we get positive values: $P(y=1) = e^{b_0 + b_1 x}$.

To make sure we can only get values between 0 and 1, we can divide this function by something slightly larger than itself. For reasons which we'll learn, we'll divide it by itself plus one. This gives us the:

$$\text{logistic function: } P(y=1) = \frac{e^{b_0 + b_1 x}}{1 + e^{b_0 + b_1 x}} = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

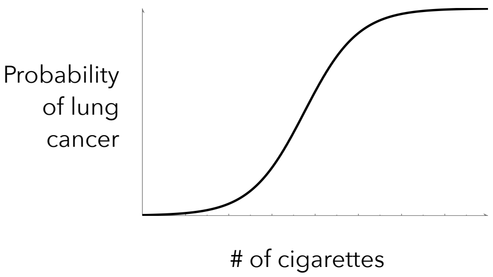
4. If we fit this logistic model to our (fictitious) lung cancer example, we might find the function displayed to the right.

   Notice the logistic function:

   a) Is asymptotic with respect to 0 and 1.

   b) Is monotonically increasing as x increases.

   c) Is continuous



$$P(y)=\frac{1}{1+e^{-(b_0+b_1x_1)}}$$

I still haven't explained <u>why</u> we use the exponential function and <u>why</u> we have the "+1" in the denominator. To do so, let's try to derive things in the opposite direction.

Suppose we like linear regression, so we <u>really want to use a linear function</u> in this lung cancer example. We know linear regression should only be used to model continuous, unbounded outcome variables, but we want to use it to model a probability (or proportion) that ranges from 0 to 1.

Let's see if we can convert a probability into a continuous, unbounded variable.

- Our probability ranges from 0 to 1.
- We could convert the probability into odds: $\text{odds}=\frac{P(y)}{1-P(y)}$. Odds approach 0 to infinity.
- If we take the log of the odds, we have something that has an unbounded range: $\ln(\text{odds})=\ln\frac{P(y)}{1-P(y)}$

This final function, the **log-odds** or **logit**, can approach both negative and positive infinity, so it can be modeled by a linear function: $\ln(\text{odds})=\ln\frac{P(y)}{1-P(y)}=b_0+b_1x$

Let's take that **logit** and solve for p:

Given $\ln\frac{p(y)}{1-p(y)}=b_0+b_1x_1$. Solve for p:

$$\frac{p(y)}{1-p(y)}=e^{b_0+b_1x_1}$$

$$p(y)=\left[1-p(y)\right]e^{b_0+b_1x_1}$$

$$p(y)=e^{b_0+b_1x_1}-p(y)e^{b_0+b_1x_1}$$

$$p(y)+p(y)e^{b_0+y_1b_1}=e^{b_0+y_1b_1}$$

$$p(y)\left(1+e^{b_0+y_1b_1}\right)=e^{b_0+y_1b_1}$$

$$p(y)=\frac{e^{b_0+b_1x_1}}{1+e^{b_0+b_1x_1}}=\frac{1}{1+e^{-(b_0+b_1x_1)}}$$

Look at that! We ended up with the logistic function from the bottom of the previous page. The logistic and logit functions are inverses.

A 1994 study of 17,096 college students found that 3,314 (19.385%) identified themselves as *frequent binge drinkers*.

Source: http://www.ndsn.org/jan95/college.html

Henry Wechsler, Ph.D., et.al., "Health and Behavioral Consequences of Binge Drinking in College," Journal of the American Medical Association, Dec. 7, 1994, p. 1672-1677; "The Disruptions of Campus Drunkenness," US News & World Report, Dec. 19, 1994, p. 12

5. Suppose the college students in this study are a random sample of all college students. What's our best guess for (estimate of) the proportion of all college students who identify themselves as binge drinkers?

The best estimate is the one that **maximizes the likelihood** of observing 3,314 binge drinkers from 17,096 students.

If we knew the population proportion, we could calculate the probability of observing 3,314 binge drinkers using the binomial distribution: $P(Y=3314)=\binom{17096}{3314}p^{3314}(1-p)^{17096-3314}$

Since we don't know the population proportion (p), we'll assume it's the value that maximizes that probability.

$$\text{Best estimate of p} = \max\left\{L(Y=3314)\right\} = \max\left\{\binom{17096}{3314}p^{3314}(1-p)^{17096-3314}\right\}$$

We can simplify things a bit before we find the value of p that maximizes that likelihood.
First, let's eliminate the combination, since it's constant across all values of p.

$$\text{Best estimate of p} = \max\left\{p^{3314}(1-p)^{17096-3314}\right\} \qquad = \max\left\{\prod_{i=1}^{N}P(x_i)^{y_i}\left(1-P(x_i)\right)^{1-y_i}\right\}$$

Then, instead of multiplying, let's take the natural log of this likelihood function:

$$\text{Best estimate of p} = \max\left\{\ln\left(p^{3314}(1-p)^{17096-3314}\right)\right\} \qquad = \max\left\{\ln\left(\prod_{i=1}^{N}P(x_i)^{y_i}\left(1-P(x_i)\right)^{1-y_i}\right)\right\}$$

We can use properties of logarithms to simplify:

$$\text{Best estimate of p} = \max\left\{3314\ln(p)+(17096-3314)\ln(1-p)\right\} \quad = \max\left\{\sum_{i=1}^{N}y_i\ln P(x_i)+(1-y_i)\ln\left(1-P(x_i)\right)\right\}$$

We can now maximize this by setting the first derivative equal to zero and solving for p:

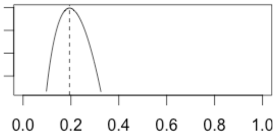
$$\frac{dL}{dp}\left\{3314\ln(p)+(17096-3314)\ln(1-p)\right\}=0$$

$$\frac{3314}{p}-\frac{17096-3314}{1-p}=0$$

$$\frac{3314(1-p)-(17096-3314)p}{p(1-p)}=0$$

$$3314-17096p=0$$

$$p=\frac{3314}{17096}\approx .19385$$



$$\frac{dL}{dp}\left\{\sum y_i\ln(p)+\left(n-\sum y_i\right)\ln(1-p)\right\}=0$$

$$\frac{\sum y_i}{p}-\frac{n-\sum y_i}{1-p}=0$$

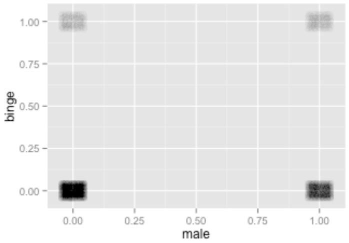
$$\frac{\sum y_i(1-p)-\left(n-\sum y_i\right)p}{p(1-p)}=0$$

$$\sum y_i-np=0$$

$$p=\frac{\sum y_i}{n}$$

6. The following table and plot display the results of that binge drinking survey by gender. Fill-in the blanks below.

	Female	Male	Total
Not binge drinker	8254	5528	13,782
Binge drinker	1690	1624	3,314
Total	9944	7152	17,096



	Female	Male
P(binge drinker) =		$\frac{1624}{7152} = 0.227$
Odds(binge drinker) =		$\frac{0.227}{1-0.227} = 0.294$
ln(odds) =		$\ln\frac{0.227}{1-0.227} = \text{-}1.23$
Odds ratio =		

7. Suppose we code our gender variable as **X = 0 (for females)** and **X = 1 (for males)**. Then, we could find:

	Female (X = 0)	Male (X = 1)
ln(odds) =	$\ln(0.205) = -1.59$	$\ln(0.294) = -1.23$

If we model these log-odds as a linear function: $\ln(\text{odds}) = b_0 + b_1(x)$

	$-1.59 = b_0 + b_1(0)$	$-1.23 = -1.59 + b_1(1)$
	Therefore, b$_0$ = _______________	Therefore, b$_1$ = $-1.23 + 1.59 = 0.36$

Our linear model is: $\ln(\text{odds})$ = ____

Odds(binge drinker) = ___________________________ ___________________________

Odds ratio = ____

P(binge drinker) =

Relative probability = ___________________________________

8. Believe it or not, we've just gone through our first example of **logistic regression**. We use logistic regression to model (the log-odds of) a binary dependent variable.

   Let's replicate this example in R:

```
# Input data from this binge drinking study and store it in a data.frame called “drink”
male <- c(rep(1,7152), rep(0,9944))
binge <- c(rep(1,1624), rep(0,5528), rep(1,1690), rep(0,8254))
drink <- data.frame(gender = factor(male, labels=c("female", "male")),
                    drink = factor(binge, labels=c("Not binge", "binge")))

drink.logmodel <- glm(drink ~ gender, data=drink, family=binomial(link=“logit")) # Fit logistic model
summary(drink.logmodel)       # Summarize model
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.58597    0.02670 -59.400   <2e-16 ***
gendermale   0.36104    0.03885   9.292   <2e-16 ***
---
    Null deviance: 16814  on 17095  degrees of freedom
Residual deviance: 16728  on 17094  degrees of freedom
AIC: 16732
```

   Those match our results

   Those coefficient were estimated from **maximum likelihood estimation** (like we used earlier).
   They represent the coefficients that maximize the likelihood of observing the data we actually observed.

   We'll figure out the rest of that output later. For now, let's replicate our odds ratio by using the exponential function:

```
exp(coef(drink.logmodel))     # Exponential function of our model coefficients
```

```
(Intercept)  gendermale
  0.2047492   1.4348145
```

   Matches our results

   We can calculate the predicted probabilities of binge drinking for each group:

```
# Create new data.frame with 1 male and 1 female
new <- gender=c(0,1)
new <- data.frame(gender = factor(gender, labels=c("female", “male")))

# Calculate predictions for our new data.frame (“response” gives us predicted probabilities)
predict(drink.logmodel, newdata=new, type=“response")
```

```
   female      male
0.1699517 0.2270694
```

   These match our results

   Before we see any more examples, let's take a look at the command used to fit our logistic regression model:

```
drink.logmodel <- glm(drink ~ gender, data=drink, family=binomial(link=“logit"))
```

   The new parts of this command are **glm** and **family=binomial(link=“logit”)**.

   What do they represent?

9. GLM stands for the *generalized linear model*.

In linear regression, we predict the expected value of an outcome variable as a linear combination of predictors. With this model, a constant change in a predictor is associated with a constant change in the outcome variable. This works when our dependent variable can, essentially, vary indefinitely in either direction.

Linear models are not appropriate for other types of dependent variables, such as:

- Number of students attending St. Ambrose as a function of tuition. The number of students is a count that must be positive, so we shouldn't use a linear function that can predict negative numbers of students.

- Whether a student returns to St. Ambrose or drops out as a function of the student's first semester GPA. Since the outcome can only be 0 (drop out) or 1 (return), a linear function isn't appropriate.

Generalized linear models handle these situations by:

- allowing dependent variables to have arbitrary distributions (other than normal distributions)
- allowing an arbitrary (link) function of the dependent variable to vary linearly with the predicted values.

To model enrollment as a function of tuition, we might choose a Poisson model (to model counts) and a log link.

To model drop-outs, we might chose a binomial model (for probabilities) and a logit (log-odds) link function.

Generalized linear models consist of three components:

1) **Random component** specifying the distribution of y given values of the predictor variables
   If y is a continuous variable, its probability distribution might be normal;
   if y is binary, the distribution might be binomial;
   if y represents counts, the distribution might be Poisson

2) **Systematic (linear) component** representing a linear combination of the predictors $\eta_i = b_0 + b_1x_1 + \ldots + b_kx_k$
   Predictors may be continuous, categorical, polynomial terms, interactions, transformed variables, etc.

3) **Link function** linking the random and systematic components (the expected value of y to the predictors)
   $$g\left(E[y]\right) = \eta_i = b_0 + b_1x_1 + \ldots + b_kx_k$$
   Some common link functions include: .

   - Identity link -- $g\left(E[y]\right) = \mu_y$ -- which is used in standard linear models.
   - Log link -- $g\left(E[y]\right) = \ln\left(\mu_y\right)$ -- which is used for count data in log-linear models.
   - Logit link -- $g\left(E[y]\right) = \ln\left(\mu_y / \left(1 - \mu_y\right)\right)$ -- which is used for binary dependent variables.

Because the link function is invertible, we can write $E[y] = g^{-1}\left(\eta_i\right) = g^{-1}\left(b_0 + b_1x_1 + \ldots + b_kx_k\right)$

With this, the generalized linear model can be thought of as

- a linear model for a transformation of the expected value of the dependent variable, or
- a nonlinear regression model for the dependent variable

Let's take a more in-depth look at some other logistic regression examples:

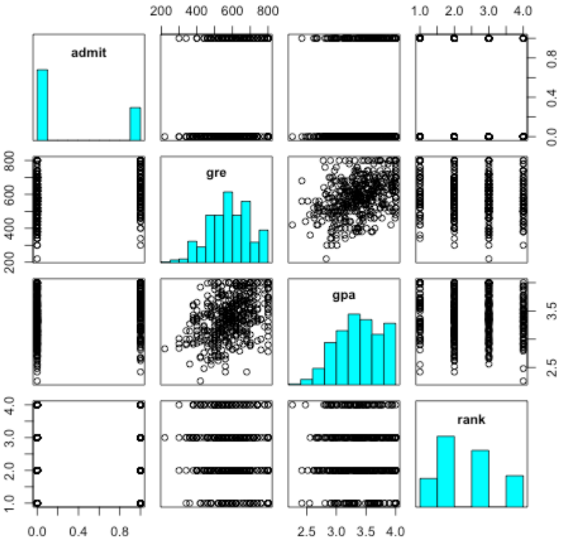
Scenario: A researcher is interested in predicting admission to graduate school based on GRE scores, undergraduate GPAs, and the reputation of the undergraduate school attended.

Our dataset consists of 400 observations.
The dependent variable, **admit**, is binary:

admit = 1 = the student was admitted
admit = 0 = the student was not admitted

31.75% of the students in the data were admitted.

The rank variable represents the prestige of the undergraduate institution on a scale from 1-4.



10. Let's begin by modeling admission as a function of GRE scores. We'll fit a logistic model:

```
gre.model <- glm(admit ~ gre, family=binomial, data=admit)
summary(gre.model)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.901344   0.606038  -4.787 1.69e-06 ***
gre          0.003582   0.000986   3.633  0.00028 ***
---
    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 486.06  on 398  degrees of freedom
AIC: 490.06
```

Before we look at these maximum likelihood coefficients, let's evaluate the fit of our model.
You're already somewhat familiar with AIC, but what do the null and residual deviance values represent?

We know we can use our model to predict $P(x_i)$, the probability that each student is admitted to graduate school.

To calculate the likelihood of our model, we multiply $\prod_{i=1}^{N} P(x_i)^{y_i}\left(1-P(x_i)\right)^{1-y_i}$ across all observations in our data.

Each probability will be between 0-1, so the likelihood of the model will be between 0-1. Models with larger likelihoods fit better than models with smaller likelihoods.

The natural logarithm of the likelihood of our model gives us **LL = log-likelihood**. The natural log of any number between 0-1 will be negative, so LL will always be negative (with numbers closer to zero representing better fit).

If we calculate $\text{Deviance} = -2LL$, we have a measure that can be thought of as similar to SSE. Larger values of deviance mean our model does not fit as well. The reason why we calculate deviance is because it follows an approximate chi-square distribution (and, therefore, we can use it to test models and estimate p-values).

So, in the output displayed above: null deviance = deviance from a model with no predictors.
residual deviance = deviance from a model with GRE scores as a predictor

To compare our model with the null model, we simply take: $\text{Deviance}_{\text{fitted}} - \text{Deviance}_{\text{null}} \sim \chi^2_{df=\text{\# of predictors in fitted model}}$

In this case, we have: $499.98 - 486.06 = 13.92 \sim \chi^2_{df=1}$, which is associated with a p-value of p = 0.00019.

11. Our model, which we just found is better than a null model, is: $\ln(\text{odds of admission}) = -2.9 + 0.0036(GRE)$

Suppose a student has a (below average) GRE score of 400. What are this student's odds of admission?

odds of admission for GRE of 400 = ______________________________

Convert those odds to a probability of admission:

probability of admission for GRE of 400 = ______________________________

12. This time, calculate the odds of admission for a student with a GRE score of 401:

odds of admission for GRE of 401 = ______________________________

13. Calculate the odds ratio for admission for a student with a GRE of 401 versus a student with a GRE of 400:

odds ratio = ______________________________

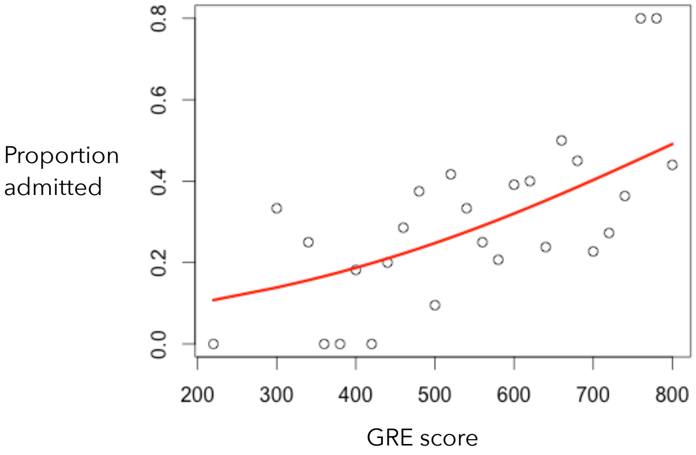
We can calculate this odds ratio (along with a confidence interval) in R with:

```
exp(cbind(OR = coef(gre.model), confint(gre.model)))  # Exp = exponential function; cbind = put together
```

```
                    OR      2.5 %    97.5 %
(Intercept) 0.0549493 0.01624471 0.1755632
gre         1.0035886 1.00168137 1.0055682
```

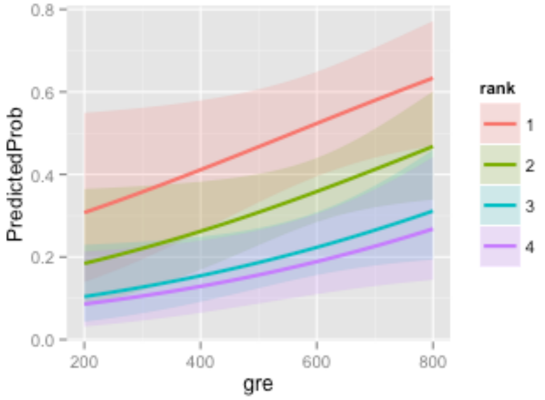
14. We could calculate these odds and probabilities across all possible GRE scores. We could also evaluate our model fit by examining the predicted probabilities against the proportion of students admitted across all GRE scores:

GRE	Odds	P(admit)
200	0.112	0.101
300	0.161	0.139
400	0.23	0.187
500	0.329	0.248
600	0.471	0.32
700	0.674	0.403
800	0.965	0.491



15. Let’s see what improvement in fit we get if we use all 4 predictor variables:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
rank2       -0.675443   0.316490  -2.134 0.032829 *
rank3       -1.340204   0.345306  -3.881 0.000104 ***
rank4       -1.551464   0.417832  -3.713 0.000205 ***
---
    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 470.52
```



We could compare this model’s deviance (458.52) to the deviance from our model with a single predictor (486.06). What distribution would we compare it to? How many degrees of freedom would we have? Would we conclude this full model fits better than the model with a single predictor?

16. Here are the odds ratios, with confidence intervals. Interpret:

```
                   OR       2.5 %    97.5 %
(Intercept) 0.0185001 0.001889165 0.1665354
gre         1.0022670 1.000137602 1.0044457
gpa         2.2345448 1.173858216 4.3238349
rank2       0.5089310 0.272289674 0.9448343
rank3       0.2617923 0.131641717 0.5115181
rank4       0.2119375 0.090715546 0.4706961
```

17. To see if the rank of a student’s undergraduate school improves our prediction, I conducted a Wald test on terms 4-6 in our model. Interpret:

```
wald.test(b = coef(admitlogit), Sigma = vcov(admitlogit), Terms = 4:6)
Wald test: Chi-squared test:
X2 = 20.9, df = 3, P(> X2) = 0.00011
```

I also compared rank 2 institutions to rank 3. Interpret:

```
wald.test(b = coef(admitlogit), Sigma = vcov(admitlogit), L = l)
Wald test: Chi-squared test:
X2 = 5.5, df = 1, P(> X2) = 0.019
```

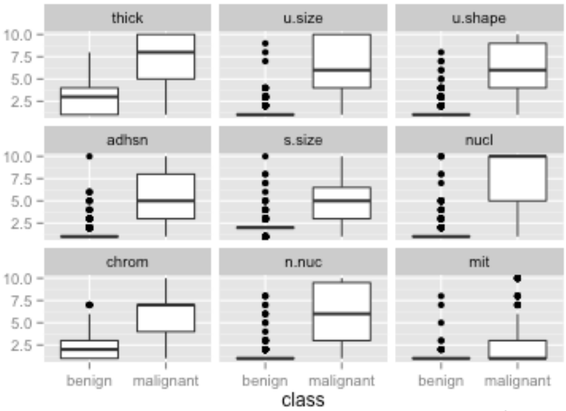
Scenario: Dr. William H. Wolberg from the University of Wisconsin Hospitals provided data from biopsy assessments of breast tumors for 683 patients (ending on July 15th, 1992). Each of 9 attributes of the tumors has been scored on a scale of 1-10 and the tumors have also been classified as either benign or malignant.

```
thick = clump thickness              adhsn = marginal adhesion       chrom = bland chromatin
u.size = uniformity of cell size     s.size = epithelial cell size   n.nuc = normal nucleoli
u.shape = uniformity of cell shape   nucl = bare nuclei              mit = mitoses
                                                                     class = benign or malignant
```

Our goal will be to use the 9 attributes to predict whether a tumor is benign or malignant.

18. I'll admit that I don't know what most of these variables represent, so let's inspect the distribution of each variable for both benign and malignant tumors.

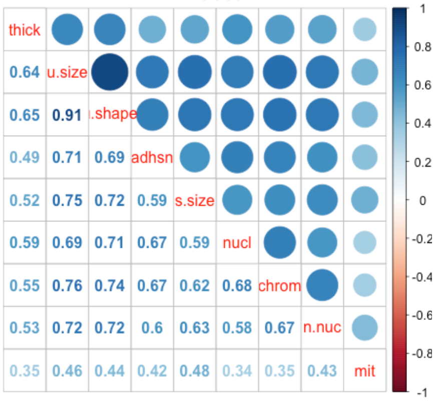
From these boxplots, which variable(s) look like they might help us predict whether tumors are benign or malignant?



Helpful variable(s) = ______________________________________

Unhelpful variable(s) = ____________________________________

Let's also take a look at the correlations between the predictors:



From the correlelogram on the right, it appears as though we might have a multicollinearity problem. Let's calculate the VIF:

```
thick  u.size u.shape   adhsn  s.size    nucl   chrom   n.nuc     mit
1.188   2.819   2.763   1.186   1.347   1.142   1.215   1.220   1.042
```

Do these VIF values indicate we have a multicollinearity problem?

19. Before we fit any models, let's randomly split our data into training and test datasets:

Training dataset: 474 observations (70% of the data); 34.2% of tumors are malignant. We'll fit our models to this data.
Test dataset: 209 observations (30% of the data); 36.8% of the tumors are malignant. We'll test our models on this data.

Let's fit a logistic model with all 9 predictors. How could we determine if this model fits better than a null model?

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -12.07441    1.97761  -6.106 1.02e-09 ***
thick         0.75495    0.22554   3.347 0.000816 ***
u.size       -0.26684    0.28954  -0.922 0.356730
u.shape       0.22282    0.28379   0.785 0.432360
adhsn         0.58666    0.20702   2.834 0.004600 **
s.size       -0.06207    0.23246  -0.267 0.789455
nucl          0.52526    0.14448   3.635 0.000277 ***
chrom         0.71157    0.23284   3.056 0.002243 **
n.nuc         0.44724    0.17153   2.607 0.009125 **
mit           0.44050    0.41143   1.071 0.284318
---
    Null deviance: 608.809  on 473  degrees of freedom
Residual deviance:  55.022  on 464  degrees of freedom
AIC: 75.022
```

20. Since we used the training dataset to construct this model, it probably fits that data pretty well. Let's see how well it fits:
    a) Use our full model to predict probabilities for each observation in the training dataset
    b) If a predicted probability is greater than 0.50, let's predict the tumor is malignant.
    c) Create a table showing how well the predicted classifications (benign or malignant) agree with the actual data

```
          benign malignant
benign       306         4
malignant      6       158
```

The model only misclassified 10 of the 474 (2.1%) observations.
Let's see how it fits the new data in the test dataset

```
          benign malignant
benign       127         4
malignant      5        73
```

The model only misclassified 9 of the 209 (4.3%) observations.
That's not too bad, but maybe we can do better.

21. In an effort to improve our model, let's use k-folds cross-validation. Below, I've pasted output from the model with the smallest average mean square error from our k-folds cross-validation. Explain how this process works.

```
Best Model:
              Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -11.5782399  1.7150431 -6.750991 1.468389e-11
thick         0.8101726  0.1836012  4.412675 1.021013e-05
adhsn         0.5290541  0.1932962  2.737012 6.200000e-03
nucl          0.5007588  0.1269485  3.944582 7.993917e-05
chrom         0.6695838  0.2124473  3.151764 1.622873e-03
n.nuc         0.4021677  0.1288032  3.122341 1.794187e-03
```

Let's test this model on our test dataset.

```
          benign malignant
benign       127         4
malignant      5        73
```

This model did not improve our fit.

At this point, if we wanted to improve our prediction, we'd probably turn to another technique (such as discriminant analysis or some machine learning algorithm).

Scenario: Approximately 2-3% of Americans have *dermatophyte onychomycosis* (a toenail infection). The infection is caused by a fungus and does not only disfigure the nails but can also cause physical pain and impair the ability to work.

Researchers conducted a randomized, double-blind clinical trial of treatments for this toenail infection. 378 patients were randomly allocated into two oral antifungal treatments:

- 250 mg/day terbinafine, or
- 200 mg/day itraconazole.

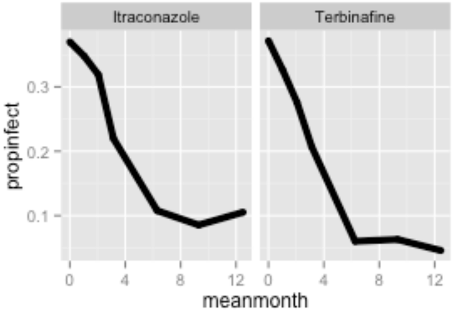
The patients were them monitored over time. The data generated from this study include:

y = onycholysis (0 = none or mild; 1 = moderate or severe)
x_1 = treatment (0 = itraconazole; 1 = terbinafine)
x_2 = month = number of months since first treatment

Do patients receiving one treatment experience a greater decrease in their probability of having onycholysis than those receiving the other treatment?

Source: De Backer, M. et al. (1998). Twelve weeks of continuous oral therapy for toenail onychomycosis caused by dermatophytes: a double-blind comparative trial of terbinafine 250 mg/day versus itraconazole 200 mg/day. Journal of the American Academy of Dermatology. 38: 57-63.

22. Before we begin, let's take a look at the proportion of patients in each treatment with toenail infections over time.



Based on this plot, which treatment appears to be more effective?

From this plot, how can we tell what proportion of the 378 patients even had the toenail infection?

23. Here are the estimated odds-ratios from: $\ln(\text{odds}) = b_0 + b_1(\text{Terbinafine}) + b_2(\text{month}) + b_3(\text{Terbinafine})(\text{month})$

```
Coefficients:
                             Estimate Std. Error z value Pr(>|z|)       95% CI
(Intercept)                -0.5566273  0.1089628  -5.108 3.25e-07       (-0.770, -0.343)
treatmentTerbinafine       -0.0005817  0.1561463  -0.004   0.9970       (-0.307,  0.305)
month                      -0.1703078  0.0236199  -7.210 5.58e-13       (-0.217, -0.124)
treatmentTerbinafine:month -0.0672216  0.0375235  -1.791   0.0732       (-0.141,  0.006)
```

Let's rearrange some of the terms in this model to see if we can gain a better understanding.

$$\text{Original model: } \ln(\text{odds}) = b_0 + b_1(\text{Terbinafine}) + b_2(\text{month}) + b_3(\text{Terbinafine x month})$$

Model for itraconazole (terbinafine = 0)

$$\ln(\text{odds}) = b_0 + b_1(0) + b_2(\text{month}) + b_3(0 \text{ x month})$$
$$\ln(\text{odds}) = b_0 + b_2(\text{month})$$

Model for terbinafine (terbinafine = 1)

$$\ln(\text{odds}) = b_0 + b_1(1) + b_2(\text{month}) + b_3(1 \text{ x month})$$
$$\ln(\text{odds}) = b_0 + b_1 + b_2(\text{month}) + b_3(\text{month})$$
$$\ln(\text{odds}) = (b_0 + b_1) + (b_2 + b_3)(\text{month})$$

Based on these rewritten models, which model parameters represent the increased effectiveness of terbinafine?

24. Calculate the following:

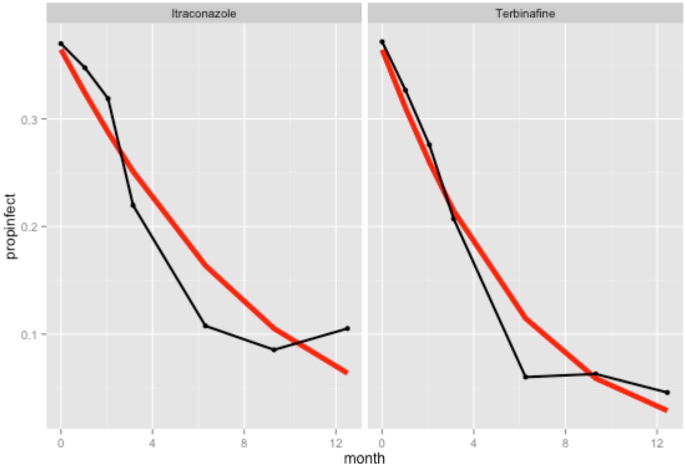
<u>Odds itraconazole (terbinafine = 0)</u>

<u>Odds for terbinafine (terbinafine = 1)</u>

<u>Odds ratio</u>

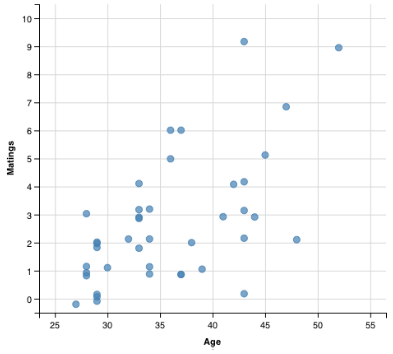
25. Sketch a graph of the odds ratio as a function of time. Try the reciprocal so you're comparing the odds of infection for terbinafine compare to the odds of infection for itraconazole. How can we interpret this graph?

26. To the right, I plotted our logistic model on top of the proportion of patients with infections at each month. Evaluate the fit of the model.



Scenario: When elephants reach maturity (around the age of 14), they have to compete with all adult males for mating opportunities. Elephants get bigger as they get older -- and females are generally more receptive to larger males -- so the number of matings should increase with age.

We're going to model the number of matings as a function of the age of the males.



Elephant #	Age	# of Matings
1	27	0
2	28	1
…	…	…
41	52	9
Mean =	35.85	2.682927
Variance =	43.28	5.071951

Source: gpk package: http://cran.r-project.org/web/packages/gpk/gpk.pdf

27. Our dependent variable represents a count (number of matings). Counts tend to be: (1) discrete, (2) positive, (3) positively skewed with a high proportion of zeros. If we try to fit an ordinary least squares regression line, we'll run into problems:
    (1) the relationship between X and Y is nonlinear
    (2) counts tend to be heteroskedastic
    (3) our line will predict negative values (which cannot exist).

We can use the generalized linear model to predict counts with the natural log link function: $g\left(E\left[y\right]\right)=\ln\left(\mu_y\right)=b_0+b_1x_1$

This link function ensures our predicted values for y are positive and positively skewed. To see this more clearly, we can use the exponential function: $e^{\ln(\mu_y)}=e^{b_0+b_1x_1}$

$$\mu_y=e^{b_0+b_1x_1}$$

When we use this log link function, we're conducting a *Poisson regression*. If you took MATH 300, you might remember the Poisson distribution. We used it to calculate the probability of eating at least 5 bug parts when we eat peanut butter. We learned the Poisson distribution can be thought of as a binomial distribution with an infinite number of trials. One thing to note about the Poisson distribution is that its mean is equal to its variance.

Based on the scatterplot for our elephant data, it looks as though when age increases:
   a. the (mean) number of matings increases
   b. the variability (dispersion) in number of matings increases

Since the dispersion increases with the mean for our count dependent variable, Poisson regression might be a good choice. We can then model: $\ln\left(\text{matings}\right)=b_0+b_1\left(\text{age}\right)$

Before I do that, let's fit a null model: $\ln\left(\text{matings}\right)=b_0$

I used R to estimate the coefficients of this null model:

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.98691    0.09535   10.35   <2e-16 ***
———
    Null deviance: 75.372  on 40  degrees of freedom
AIC: 178.82
```

So our null model is: $\ln\left(\text{matings}\right)=0.98691$. Convert this to predict the number of matings under this null model. What does this number represent?

28. Using R, I estimated the coefficients of a model including the age predictor:

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.58201    0.54462  -2.905  0.00368
Age          0.06869    0.01375   4.997 5.81e-07
---
    Null deviance: 75.372  on 40  degrees of freedom
Residual deviance: 51.012  on 39  degrees of freedom
AIC: 156.46
```

I then used the model to predict the average number of matings at ages 30, 31, and 45.
By what amount are the matings increasing each year? What are the predicted variances in matings at each age?

$$\text{matings at 30 years} = e^{-1.58201+0.06869(30)} = 1.614098$$

$$\text{matings at 31 years} = e^{-1.58201+0.06869(31)} = 1.728872$$

$$\text{matings at 45 years} = e^{-1.58201+0.06869(45)} = 4.522968$$

29. In general, we can interpret the parameters of a Poisson regression like this:

b_0: the mean of the Poisson distribution when our predictor equals zero
b_1: a 1-unit increase in X is associated with an e^{b1} increase in the expected count of Y

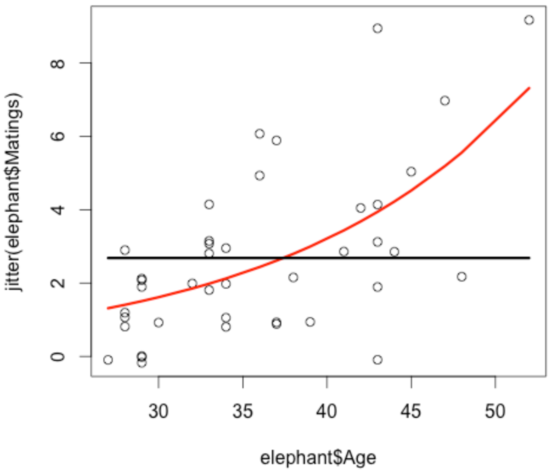
In our example, the b_0 parameter doesn't make sense, since age=0 is not meaningful.

What does the b_1 parameter represent in our scenario? Interpret that coefficient of 0.06869.

30. We can plot our predictions (from both the null and age-based models) on top of our data. Obviously, the model with age as a predictor appears to better fit the data.

We can also compare models by comparing the decrease in deviance values from our null to full models.

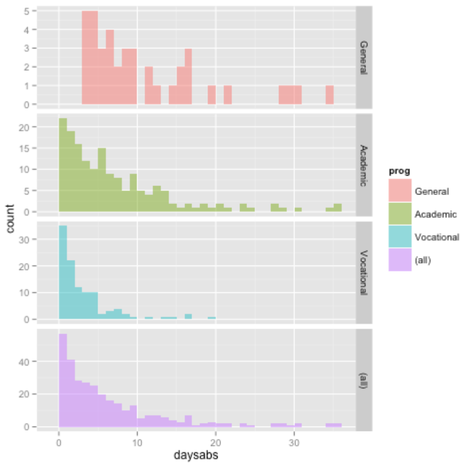
Using R, I conducted a chi-squared test that the difference in deviance values between our two models is zero. R reported a p-value of 0.000000799. Interpret this value.

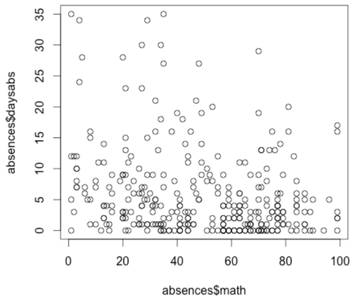


Scenario: Suppose we want to predict the number absences for students at a high school. Here's our data:

- math = math test score
- daysabs = days absent
- prog = type of program

Student	gender	math	daysabs	prog
1	male	63	4	academic
2	female	20	2	general
…		…		
314	female	77	2	vocational
Mean =			5.955	
Variance =			49.519	





Source: UCLA TISE

31. Even though the variance is much higher than the mean (even within each program type), let's try to fit a Poisson regression model with all our predictors.

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     2.7594786  0.0637731  43.270  < 2e-16 ***
gendermale     -0.2424762  0.0467765  -5.184 2.18e-07 ***
math           -0.0069561  0.0009354  -7.437 1.03e-13 ***
progAcademic   -0.4260327  0.0567308  -7.510 5.92e-14 ***
progVocational -1.2707199  0.0779143 -16.309  < 2e-16 ***
---

    Null deviance: 2217.7  on 313  degrees of freedom
Residual deviance: 1746.8  on 309  degrees of freedom
AIC: 2640.2

Exponentiated coefficients:
   (Intercept)     gendermale           math   progAcademic progVocational
    15.7916072      0.7846824      0.9930680      0.6530950      0.2806295
```

Based the deviance, the fit of this model is terrible. That's probably because of that over-dispersion.

What could we do about this? One thing we could do is fit a negative binomial regression model.
The negative binomial model is just like the Poisson model except it includes an extra parameter to model dispersion.

I fit this model using R and found:

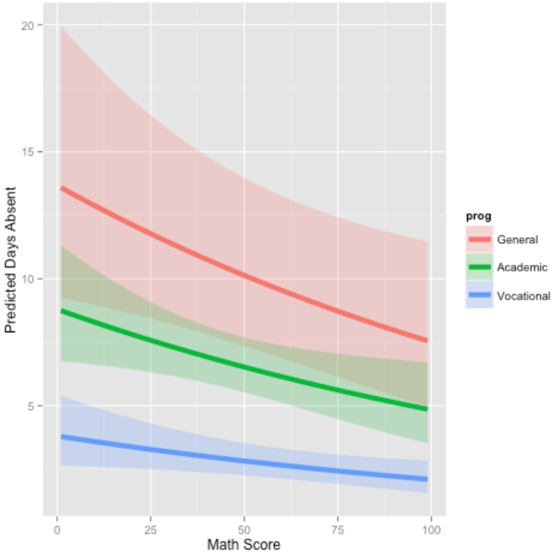
```
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     2.707484   0.204275  13.254  < 2e-16 ***
gendermale     -0.211086   0.121989  -1.730   0.0836 .
math           -0.006236   0.002492  -2.502   0.0124 *
progAcademic   -0.424540   0.181725  -2.336   0.0195 *
progVocational -1.252615   0.199699  -6.273 3.55e-10 ***
---
(Dispersion parameter for Negative Binomial(1.0473) family taken to be 1)

    Null deviance: 431.67  on 313  degrees of freedom
Residual deviance: 358.87  on 309  degrees of freedom
AIC: 1740.3

Exponentiated coefficients:
   (Intercept)     gendermale           math   progAcademic progVocational
    14.9915148      0.8097046      0.9937839      0.6540708      0.2857565
```

The deviance decreased by a large amount, so this model does fit our data better.

Suppose we're interested in seeing the the type of program (academic, general, vocational) predicts absences. To do so, we can fit a reduced model that does not include the program predictor. We can then test the difference in deviance between the models. I did this in R and found a p-value of 0.000000000313. Interpret.


Predicted Days Absent
Math Score
prog
General
Academic
Vocational