LOGISTIC REG

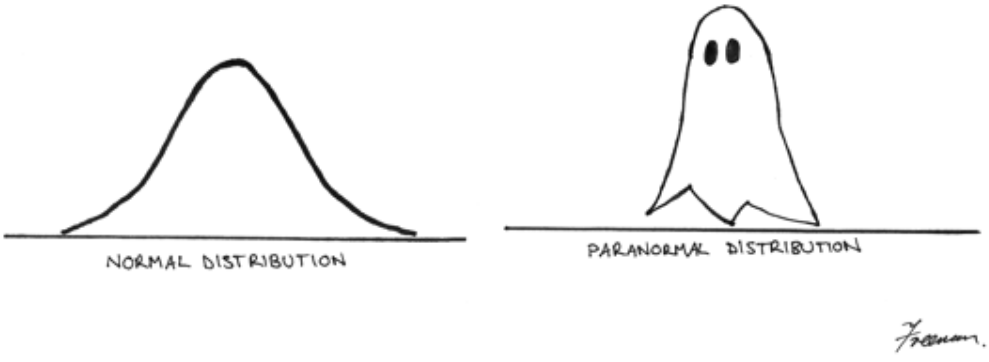
Research Methods in Psychology I & II ▪ Department of Psychology ▪ Colorado State University

BY THE END OF THIS UNIT YOU WILL:

1. Understand the idea behind modeling a binary outcome using a maximum likelihood estimator with a logit link (i.e., a logistic regression).
2. Know how to estimate a binary logistic regression model and interpret the parameters estimates in terms of logits and Odds Ratios (OR).
3. Know how to plot the results of a binary logistic regression.
4. Learn how to calculate predicted probabilities of outcomes across values of the predictors.
5. Gain introductory knowledge of marginal effects on the probability scale.
6. Gain introductory knowledge of how interactions are interpreted in binary logistic regression.
7. Know the assumptions of a binary logistic regression model.
8. Be prepared to use the tools from Unit 9 to assess assumptions.

What is binary logistic regression?

So far in the 652/653 sequence, we have worked with the general linear model to fit ordinary least squares (OLS) regression models (i.e., linear regression) for continuous outcomes. In this unit, we will begin to work more generally with the generalized linear model (GLM). GLMs are a broad class of models that include general linear models, but also includes other types of outcomes, including categorical and count variables. In this unit, we will explore a very commonly used categorical outcome — a binary outcome. To fit this type of generalized linear model, we will use a binomial distribution (rather than the normal distribution that we've been using so far) and a logit link to link the binary outcome with the predictors. To fit these models we will use a maximum likelihood estimator rather than the ordinary least squares approach that we've utilized up until this point in the course sequence.



Example Dataset Description — predictors of drinking on 21st birthday

A research team sought to examine factors associated with 21st birthday drinking among female students at a large University. Female students who were nearing age 21 and self-classified as regular drinkers were eligible for the study. In total, 200 students were recruited and agreed to take part in the study. Students were instructed to report to the lab two weeks prior to their 21st birthday. During this lab session, students completed a brief survey that measured alcohol use during the past month (using the Timeline Follow Back Method) and their weight was recorded. One week prior to their 21st birthday, participants were sent a link for an online survey to measure positive alcohol expectancies for drinking on their 21st birthday. Within three days prior to their 21st birthday, students reported to the lab and were given a diary-based data collection form to record several items on their 21st birthday. Students were instructed to record the food that they consumed during the day, the degree to which they were in a partying mood just prior to the celebration, and the quantity and type of drinks that they consumed during the first two hours of the celebration. The students were also given a small breathalyzer machine to measure BAC 2 hours after consumption of their first drink.

The dataset called bac_obs.csv contains the data:

weight: weight in kilograms

alcexp: positive alcohol expectancy for drinking on the impending 21st birthday, a multi-item scale that ranges from 1-7, where a higher score indicates more positive expectations about the role alcohol will play

typ_drks: the number of standard alcohol drinks consumed in the past 30 days

pmood: a rating on a scale from 1-9 on the respondent's mood to party on the 21st birthday, where 1 means never been less in the mood to party, and 9 means never been more in the mood to party

absorb: a score calculated from the food diaries to determine how full the participant was when they began drinking, the score ranges from 1 to 8, where 1 means a completely full stomach, and 8 means a completely empty stomach

alc_gm: a score calculated from the drinking diary to estimate the grams of alcohol consumed on the 21st birthday

bac: the participant's blood alcohol content, measured as grams of alcohol per deciliter of blood on the 21st birthday

Prepare Data and Get Descriptive Statistics

Load libraries

```
library(modelr)
library(tidyverse)
library(descriptr)
library(margins)
```

Import data

```
obs <- read_csv("bac_obs.csv")
```

Dichotomize two of the variables for demonstration purposes

```
# Create dichotomized versions of bac (>.08 vs. <= .08) and
# typ_drks (> average of 2 per day vs. <= average of 2 per day).

obs <- mutate(obs,
        bac_over = ifelse(bac > .08, 1, 0),
        typ_hvy = ifelse(typ_drks > 60, 1, 0))
```

For purposes of demonstration, we will work with dichotomized versions of BAC and typical drinking.

For BAC, we'll compare people who have a BAC over the legal limit (.08), to those with a BAC at or under .08.

For typical drinking, we'll compare people who report consuming more than 2 drinks per day on average (we'll call this group the heavy drinkers), to those who consume two or fewer drinks per day (we'll call this group light drinkers).

Let's Start Simple — A Two by Two Crosstab

Create cross table of typical heavy drinking & BAC > .08

```
ds_cross_table(obs$typ_hvy, obs$bac_over)
```

```
   Cell Contents
|---------------|
|     Frequency |
|       Percent |
|       Row Pct |
|       Col Pct |
|---------------|

 Total Observations:  200

-------------------------------------------------------------
|              |                 bac_over                   |
-------------------------------------------------------------
|      typ_hvy |            0 |            1 |    Row Total |
-------------------------------------------------------------
|            0 |           90 |           71 |          161 |
|              |        0.45 |        0.355 |              |
|              |        0.56 |        0.44 |          0.8 |
|              |        0.97 |        0.66 |              |
-------------------------------------------------------------
|            1 |            3 |           36 |           39 |
|              |        0.015 |        0.18 |              |
|              |        0.08 |        0.92 |          0.2 |
|              |        0.03 |        0.34 |              |
-------------------------------------------------------------
| Column Total |           93 |          107 |          200 |
|              |        0.465 |        0.535 |              |
-------------------------------------------------------------
```

90/161 = .56, 56% of light drinkers had a BAC <= .08.

3/39 = .08, 8% of heavy drinkers had a BAC <= .08.

71/161 = .44, 44% of light drinkers had a BAC over .08.

36/39 = .92, 92% of heavy drinkers had a BAC over .08.

Calculate proportions for BAC categories by typical drinking categories

```
# calculate proportion of light drinkers and heavy drinkers who had a BAC at or under .08
prop_light_under <-  90/161
prop_heavy_under <- 3/39

# calculate proportion of light drinkers and heavy drinkers who had a BAC over .08
prop_light_over <-  71/161
prop_heavy_over <- 36/39
```

These calculations provide us with the **proportion of people in the sample** with a BAC of a certain level. A proportion is a known quantity calculated from a sample. If the sample is a random sample drawn from the population, then the proportion is an estimate of the probability of the event (e.g., a BAC over .08) occurring for a random person sampled from the population. The probability that an event will occur is also commonly referred to as the risk of the event. For example, the probability (i.e., risk) that a heavy drinker will have a BAC > .08 on her 21st birthday is .92.

$$\hat{p} = \frac{(chances\ for)}{(total\ chances)} = \frac{36}{39} = .92$$

Probability of having a BAC > .08 for heavy drinkers

Risk Ratio (aka Relative Risk)

We might then be interested in comparing the probability or risk of having a BAC > .08 across our two groups (light and heavy drinkers). To accomplish this we simply create a ratio, specifically a risk ratio. A risk ratio compares the risk of an outcome (e.g., BAC > .08) among one group (e.g., heavy drinkers) with the risk among another group (light drinkers). It does so by dividing the risk in group 1 by the risk in group 2.

$$RR = \frac{\hat{P}_{group\ 1}}{\hat{P}_{group\ 2}} = \frac{.92}{.44} = 2.09$$

Calculate risk ratio of BAC > .08 by heavy drinking status

```
RR = prop_heavy_over/prop_light_over
RR
```

```
[1] 2.093174
```

A RR of 1.0 is indicative of identical risk across the two groups. A RR >1.0 indicates an increased risk for the group in the numerator, while a RR < 1.0 indicates a decreased risk for the group in the numerator. Here, we see that the risk of having a BAC > .08 is about 2.1 times higher for heavy drinkers as compared to light drinkers.

Probability to Odds

Odds are another way to describe the likelihood that an event will occur. However, while the probability or risk is defined by the chances that an event will occur divided by the total chances, the odds is defined by the chances the event will occur divided by **the chances the event will not occur.** Consider the odds that a heavy drinker will have a BAC > .08.

$$\hat{odds} = \frac{(chances\ for)}{(chances\ against)} = \frac{36}{3} = 12$$

Calculate the odds of BAC > .08 for heavy and light drinkers

```
# calculate the odds of BAC being over .08 for heavy drinkers
odds_heavy <- prop_heavy_over/prop_heavy_under
odds_heavy

# calculate the odds of BAC being over .08 for light drinkers
odds_light <- prop_light_over/prop_light_under
odds_light
```

```
[1] 12
```

```
[1] 0.7888889
```

The odds that a heavy drinker will have a BAC > .08 are 12 (i.e., 36/3), while the odds that a light drinker will have a BAC > .08 are .79 (i.e., 71/90).

Note that if the probability that an event will occur is .5, then the odds that an event will occur is 1. This denotes even odds — the event is equally likely to occur or not-occur. Also note that taking the inverse of the odds that an event will occur (that is, 1/odds of event), gives the odds that the event will not occur. For example 1/12 = .083. The odds that a heavy drinker will have a BAC <= .08 are .083.

Transformation of probability to odds

```
trans_probodds <- tibble(prob = seq(0, 1, by = .05)) %>%
  mutate(odds = prob/(1-prob))
```

prob	odds
0.00	0.00000000
0.05	0.05263158
0.10	0.11111111
0.15	0.17647059
0.20	0.25000000
0.25	0.33333333
0.30	0.42857143
0.35	0.53846154
0.40	0.66666667
0.45	0.81818182
0.50	1.00000000
0.55	1.22222222
0.60	1.50000000
0.65	1.85714286
0.70	2.33333333
0.75	3.00000000
0.80	4.00000000
0.85	5.66666667
0.90	9.00000000
0.95	19.00000000
1.00	Inf

You can also compute the odds based on the probability using the equation below. The transformation from probability to odds is a monotonic transformation, meaning the odds increase as the probability increases or vice versa. Probability ranges from 0 to 1. Odds range from 0 to positive infinity.

$$\hat{odds} = \frac{\hat{p}}{1-\hat{p}}$$

Odds Ratio

In the probability scale, we compared the probability of an event in one group to the probability of the event in another group, and we called this a risk ratio. Similarly, we can compare the odds of an event in one group to the odds of an event in another group. This is called an odds ratio (OR).

$$OR = \frac{\hat{odds}\ in\ group\ 1}{\hat{odds}\ in\ group\ 2} = \frac{12}{.79} = 15.21$$

Calculate the odds ratio to compare the odds across the two groups

```
# calculate the odds ratio
OR <- odds_heavy/odds_light
OR

# reverse the numerator and denominator
OR_rev <- odds_light/odds_heavy
OR_rev
```

```
[1] 15.21127
[1] 0.06574074
```

The odds that a heavy drinker will have a BAC > .08 are about 15 times higher than the odds that a light drinker will have a BAC > .08. What if we had put the non-heavy drinkers in the numerator? In this case, the OR is .07. The odds that a light drinker will have a BAC > .08 are about .07 times the odds that a heavy drinker will have a BAC > .08. In other words, the odds are less than 1/10 that of a heavy drinker, or about 93% lower ((1-.07) · 100).

You can also take the inverse of the OR to reverse the results, so 1/15.211 = .066, and 1/.066 = 15.211.

The Relationship Between a Continuous Predictor and a Binary Outcome

We can extend this simple two by two table to something slightly more complex. Let's consider the relationship between grams of alcohol consumed (a continuous variable) and having a BAC over .08. To visualize the relationship, let's begin with a plot.

Calculate the proportion of people at each level of alc_gm who had a BAC over .08

```
avg_over_bygrams <- obs %>%
  group_by(alc_gm) %>%
  summarize(proportion_over = mean(bac_over)) %>%
  ungroup()
```

Plot mean of BAC across levels of alc_gm

```
ggplot(avg_over_bygrams, aes(y = proportion_over, x = alc_gm)) +
  geom_point() +
  labs(title = "Proportion of women with BAC over .08 by alcohol consumed",
       x = "Grams of alcohol consumed", y = "Proportion of women with BAC over.08")
```

Given that we want to model the relationship between grams of alcohol consumed and the likelihood of having a BAC over .08, we might be inclined to fit a regression model, regressing bac_over on alc_gm. One option would be an OLS regression model. However, we can see from the plot to the left that a straight line won't fit our data well because the probability can't be greater than 1 or less than 0.

Further, from this plot, we can see that it is highly unlikely that someone who consumed fewer than 25 grams of alcohol would have a BAC over .08, and it is highly likely that someone who consumed greater than 35 grams of alcohol would have a BAC over .08. In the middle of these two end points, we see a rather linear relationship, where each extra gram consumed increases the likelihood of having a BAC over .08. Therefore, similar to the non-linear OLS models that we studied in Unit 9, there is not a constant effect of the continuous predictor on the probability of the outcome, but rather the effect changes as we move across the values of the predictor.

Because we can see from the plot that the relationship is not linear, but rather S-shaped, we can use a transformation of the probabilities. Specifically we will apply a logit transformation. Much in the same way that we used a log transformation in Unit 9 to be able to fit a non-linear relationship at the level of the raw variables using a linear model of the transformed scores, we can use a logit transformation to fit a S-shaped curve at the level of the raw variables using a linear model of the transformed probabilities (logits).

The Logistic Regression Model

When we fit a logistic regression model, we don't model the probability, or the odds. Instead, we model the log of the odds, also called the logit. That is the natural log (ln) of the odds of the outcome given the predictors. We do this because it can be difficult to model a variable that has restricted range, such as the probability which can only range from 0 to 1, or the odds which can't be less than 0. The transformation from probability to log odds alleviates the restricted range problem. The log odds transformation takes a probability ranging between 0 and 1 and turns it into a log odds ranging from negative infinity to positive infinity.

In a binary logistic regression the outcome variable is dichotomous, that is, it can take on only two categories— for example, alive or dead, win or lose. In OLS regression, the outcome is continuous and we assume that the probability distribution is normal (i.e., recall the empirical rule). In binary logistic regression, the outcome is dichotomous and we assume a binomial distribution. A binomial random variable is the number of successes (e.g., alive vs. dead; win vs. lose) across cases (e.g., people) in a dataset. The probability distribution of a binomial random variable is called a binomial distribution.

A logistic regression model allows us to model the relationship between a binary outcome (e.g., bac_over) and one or more predictors. A logistic regression models the logit as a linear relationship with the predictor variables using a maximum likelihood (ML) estimator. Logistic regression can't be solved by a series of closed form equations like a OLS regression model can. That is why we use ML to estimate the model parameters.

Briefly, ML begins with a mathematical expression known as the Likelihood Function of the sample data. Think of the likelihood of a set of data as the probability of obtaining that particular set of data, given the chosen probability distribution model (e.g., normal, binomial for categorical outcomes, poisson for count outcomes). The starting mathematical expression contains the unknown model parameters (e.g., the intercept and the slopes that relate the predictor to the outcome). The values of these parameters that maximize the sample likelihood are known as the maximum likelihood estimates, and we will obtain estimates of the model parameters in this way (e.g., the intercept, slopes).

A logistic regression equation for a single predictor is written as (where p is the probability):

$$\hat{logit} = log(\hat{odds}) = \beta_0 + \beta_1 x_i$$

To transform the logit y-hats back to probabilities (p), we use:

$$\hat{p} = \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}}$$

Logistic Regression for a Continuous Predictor

Fit a logistic regression model for alc_gm and bac_over

```
# first center alc_gm at the mean
obs <- mutate(obs, alc_gm_m = alc_gm - mean(alc_gm))

# fit the logistic regression model
logreg0 <- glm(bac_over ~ alc_gm_m, data = obs, family=binomial("logit"))
summary(logreg0)
```

glm is the function for a generalized linear model. By specifying family = "binomial" we indicate that the outcome is categorical. The ols_regress command that we used with lm models isn't appropriate for glm, so we must use the base R summary function to look at the results.

```
Call:
glm(formula = bac_over ~ alc_gm_m, family = "binomial", data = obs)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.72104  -0.46772   0.05849   0.43081   2.49055

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.44099    0.23031   1.915   0.0555 .
alc_gm_m     0.44883    0.06356   7.062 1.64e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 276.28  on 199  degrees of freedom
Residual deviance: 130.32  on 198  degrees of freedom
AIC: 134.32

Number of Fisher Scoring iterations: 6
```

The intercept (.44099) is the predicted log odds of bac_over when alc_gm_m = 0 (i.e., the mean grams of alcohol consumed since we centered it). The estimate divided by the standard error (.23031) give the z value (z*) — you can think of this like the t* in a OLS model. The z distribution is the normal distribution, so there are no df like with the t-distribution. For alpha of .05 (2-sided test), a |z*| that exceeds |1.96| is statistically significant. The p-value is the probability of obtaining a z* of this magnitude or larger if the null hypothesis were true (i.e., null hypotheses work the same as in OLS regression, so typically the null hypothesis is that the estimate is 0).

The slope (.44883) is the predicted change in the log odds of bac_over for a 1 unit increase in alc_gm_m. Again, the estimate divided by the standard error (.06356) gives the z value (z*), and the p-value is the probability of obtaining a z* of this magnitude or larger if the null hypothesis were true (i.e., that the slope relating x to y is 0). Here we see the effect is positive (higher grams of alcohol consumed is associated with a larger log odds of having a BAC over .08) and statistically significant.

Notice that the summary command provides symbols to denote significance (labeled Signif. codes) that correspond to the p-values for an alpha of .05 (two-sided). Three stars (***) means $p<.001$, two stars (**) means $p<.01$, and one star (*) means $p<.05$. A period (.) denotes $p<.10$.

Significance Testing for Logistic Regression Slopes

When we divide the estimate of the intercept and slope(s) by the standard error in a logistic regression, this is referred to as a Wald Z Test. This test assumes that the distribution of the maximum likelihood estimators is normal. Simulation studies have shown that if the sample size or the number of events is small, the normality assumption may not be met. For that reason, other methods for testing the significance of the intercept and each slope have been proposed. Venzon and Moolgavkar (1988) proposed a likelihood-based confidence interval estimator called the profile log-likelihood, and this method is implemented in the confint function for glm models in R. A value of the profile log-likelihood is computed by specifying a value for the coefficient of interest (e.g., the slope for alc_gm_m) and then finding the value of the coefficient that maximizes the log-likelihood. This process is repeated over a grid of values of the specified coefficient until the best solution is found. We can use the confint function in R to obtain profile log-likelihood confidence intervals for our intercept and slope(s).

Venzon, D.J., & Moolgavkar, S.H. (1988). A method for computing profile-likelihood-based confidence intervals. *Journal of the Royal Statistical Society, 37(1),* 87-94.

Obtain profile log likelihood 95% confidence intervals for log odds estimates

```
confint(logreg0, level = .95)
```

```
Waiting for profiling to be done...
                  2.5 %     97.5 %
(Intercept) 0.0008690559 0.9101733
alc_gm_m    0.3364614210 0.5873176
```

In the usual way, if the 95% CI doesn't include 0 for the log odds estimate, then the estimate is significantly different than 0 (for alpha = .05). In this case, both the intercept and slope are significantly different from zero.

Transforming Log Odds Coefficients back to Odds

Recall from Unit 8 that after fitting a model in which one or more of the variables had been log transformed, we back transformed the coefficients so that we could discuss them in terms of the original variables (i.e., percent change). We can do something similar in logistic regression so that we can discuss the effects in terms of odds rather than log odds. In this case, we take the anti-log (inverse log) of the coefficients. We will exponentiate the intercept and slope (i.e., use the exp function) to accomplish this task. Once exponentiated, the intercept will represent the odds of the outcome when all predictors are 0, and each slope coefficient will represent the odds ratio (i.e., the expected change in the odds of the outcome for a 1 unit increase in the predictor of interest).

Obtain the Odds Ratio (OR)

```
exp(coef(logreg0))
```

```
(Intercept)    alc_gm_m
   1.554245    1.566476
```

The odds of having a BAC > .08 among people who consume an average amount of alcohol is 1.55.

The odds having a BAC > .08 are 1.57 times higher for each additional gram of alcohol consumed.

```
Coefficients:
            Estimate
(Intercept)  0.44099
alc_gm_m     0.44883
---
```

Transform log odds coef to percent change

```
# function to interpret regression slope for ln transformed y, original x
ln.y <- function(slope, x_chg) {
  new_slope <- 100 * (exp(slope * x_chg)-1)
  return(new_slope)
}

ln.y(slope = .44883, x_chg = 1)
```

```
[1] 56.64783
```

The odds of having a BAC > .08 are 57% higher for each one unit increase in alc_gm.

95% Confidence Intervals for Exponentiated Coefficients

We can get profile log likelihood 95% confidence intervals for these exponentiated coefficients as follows.

Get confidence intervals for exponentiated fitted estimates

```
exp(cbind(OR = coef(logreg0), confint(logreg0)))
```

```
Waiting for profiling to be done...
                   OR    2.5 %   97.5 %
(Intercept) 1.554245 1.000869 2.484753
alc_gm_m    1.566476 1.399985 1.799156
```

In the metric of odds, even odds = 1, so if the 95% CI includes 1, then the effect is not statistically significant (for alpha = .05). So, unlike the 95% CIs for the log odds where we look to determine if the interval contains 0 (meaning no effect), here we look to determine if the interval contains 1 (meaning no effect). These will match up with the 95% CIs for the log odds estimates in terms of rejecting the null hypotheses—they're just expressed in a different metric.

Obtain Predicted Values Based on Model Estimates

Using the same techniques as we did for our linear models, we can obtain predicted scores of the outcome for any prototypical values of interest. We can obtain those in terms of log odds, odds, and probability. For example, we can determine the log odds, odds and probability of having a BAC over .08 among people who consumed the average grams of alcohol in the sample.

Predicted value in log odds $\hat{logit} = log(\hat{odds}) = \beta_0 + \beta_1 x_i$

Predicted value in odds $\hat{odds} = e^{\beta_0 + \beta_1 x_i}$

Predicted value in probability $\hat{p} = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$

Get predicted values (manually)

```
# predicted value when alc_gm is at the mean (alc_gm_m = 0)
logodds <- 0.44099 + (0.44883*0)
odds <- exp(logodds)
prob <- odds/(1+odds)

logodds
odds
prob
```

```
[1] 0.44099
[1] 1.554245
[1] 0.6084949
```

Get predicted values using data_grid and add_predictions

```
pred_grid0 <- tibble(alc_gm_m = 0) %>%
  add_predictions(logreg0) %>%
  mutate(odds = exp(pred),
         predprob = (odds/(1+odds)))
pred_grid0
```

For the add_predictions function performed on a logistic regression, the predicted value (pred) is in the metric of log odds .

pred <dbl>	odds <dbl>	predprob <dbl>
0.4409899	1.554245	0.6084949

Among people who consume an average amount of alcohol (32.8 grams), the log odds of having a BAC over .08 is .44, the odds are 1.55, and the probability is .61.

Plot of the Fitted Model

First, let's get the predicted values across a range of alc_gm

```
pred_grid0 <- data_grid(obs, alc_gm_m = seq_range(alc_gm_m, 10)) %>%
  add_predictions(logreg0) %>%
  mutate(odds = exp(pred),
           predprob = (odds/(1+odds)),
           alc_gm = alc_gm_m + mean(obs$alc_gm))
```

Plot in metric of log odds

```
ggplot(pred_grid0, aes(x = alc_gm, y = pred)) +
  geom_line(size = 1) +
  labs(title = "Model fitted log odds of BAC > .08 as function of grams of alcohol consumed",
      x = "Grams of Alcohol Consumed", y = "Log odds of BAC > .08")
```

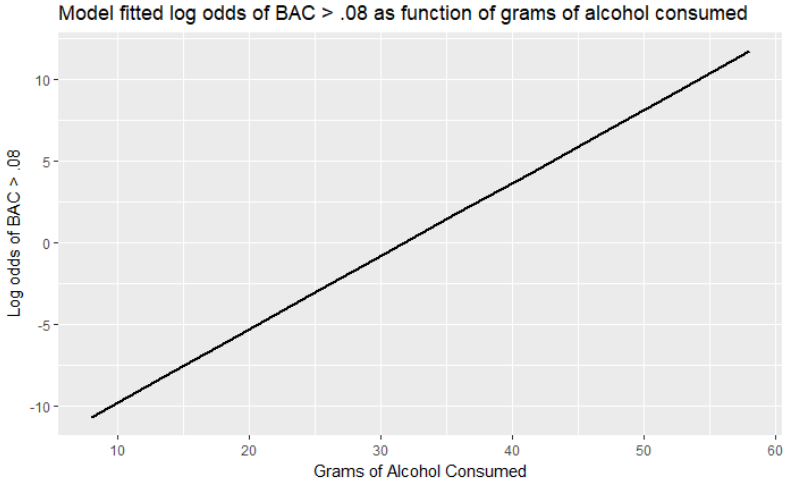
Plot in metric of odds

```
ggplot(pred_grid0, aes(x = alc_gm, y = odds)) +
  geom_line(size = 1) +
  labs(title = "Model fitted odds of BAC > .08 as function of grams of alcohol consumed",
      x = "Grams of Alcohol Consumed", y = "Odds of BAC > .08")
```

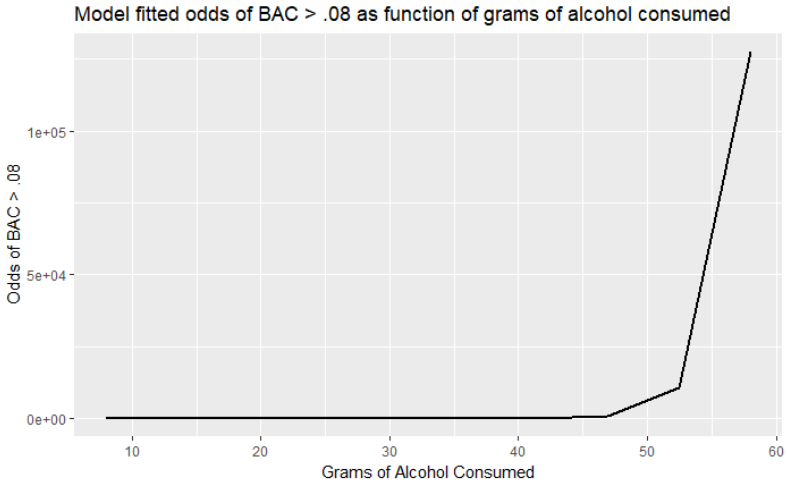
Plot in metric of probability

```
ggplot(pred_grid0, aes(x = alc_gm, y = predprob)) +
  geom_line(size = 1) +
  labs(title = "Model fitted probabilities of BAC > .08 as function of grams of alcohol consumed",
      x = "Grams of Alcohol Consumed", y = "Probability of BAC > .08")
```

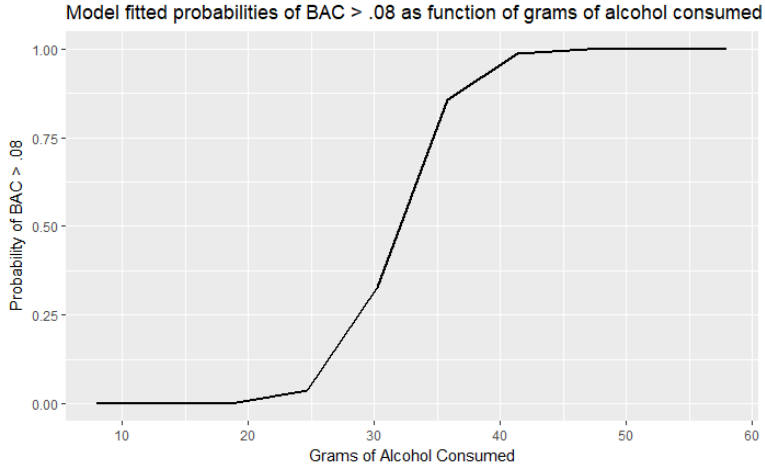
Plots



The logistic regression model specifies a linear relationship between the predictor and the log odds of the outcome.



The logistic regression model specifies a non-linear relationship between the predictor and the odds of the outcome.



The logistic regression model specifies a S-shaped relationship between the predictor and the probability of the outcome.

Let's Return to Our Simple Two by Two Example

Let's fit a logistic regression model to determine if the odds of having a BAC > .08 differ as a function of typical drinking status.

Fit a logistic regression model to our two by two example

```
logreg1 <- glm(bac_over ~ typ_hvy, data = obs, family=binomial("logit") )
summary(logreg1)

exp(cbind(OR = coef(logreg1), confint(logreg1)))
```

```
Call:
glm(formula = bac_over ~ typ_hvy, family = "binomial", data = obs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2649  -1.0785   0.4001   1.2796   1.2796

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.2371     0.1587  -1.494    0.135
typ_hvy       2.7220     0.6215   4.380 1.19e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 276.28  on 199  degrees of freedom
Residual deviance: 242.10  on 198  degrees of freedom
AIC: 246.1

Number of Fisher Scoring iterations: 5

Waiting for profiling to be done...
                    OR     2.5 %    97.5 %
(Intercept)  0.7888889 0.5764987  1.075408
typ_hvy     15.2112676 5.2134256 64.872148
```

The intercept is the log odds of having a BAC over .08 among people who are light drinkers. When we exponentiate this value (e.g., exp(-.2371) we get .7889 which is the odds that people in this group will have a BAC over .08. If we take the odds divided by 1 + the odds (e.g., .7889/(1+.7889) we get the predicted probability (.4410) that people in this group will have a BAC over .08. Likewise, we can use the equation to obtain the log odds of a BAC over .08 for heavy drinkers: log odds = -.2371 + 2.7220*1 = 2.4849. We can then use the same technique to obtain the odds and predicted probability of having a BAC > .08 for people in this group. Notice the correspondence with the values we obtained at the beginning of the unit.

Notice also that the calculated odds ratio based on this model is 15.21, indicating that the odds of having a BAC over .08 are more than 15 times higher among people who are heavy drinkers as compared to people who are light drinkers. The 95% CI for the OR does not include 1, indicating that the odds of having a BAC over .08 are significantly larger for heavy drinkers compared to light drinkers.

Obtain log odds, odds, and probability

```
pred_grid1 <- data_grid(obs, typ_hvy) %>%
  add_predictions(logreg1) %>%
  mutate(odds = exp(pred),
         predprob = (odds/(1+odds)))
```

typ_hvy <dbl>	pred <dbl>	odds <dbl>	predprob <dbl>
0	-0.2371298	0.7888889	0.4409938
1	2.4849066	12.0000000	0.9230769

Add an Additional Predictor to our Two by Two Example

Let's complete this example by adding an additional predictor to the model — alcexp. In terms of interpreting the log odds coefficients and ORs, multiple predictors work the same in this setting as they did for the linear model. The slope for each predictor will represent the unique effect after adjusting for all other predictors in the model.

Regress bac_over on typ_hvy and alcexp

```
# center alcexp at the mean
obs <- mutate(obs, alcexp_m = alcexp - mean(alcexp))

#fit the regression model
logreg2 <- glm(bac_over ~ typ_hvy + alcexp_m, data = obs, family=binomial("logit"))
summary(logreg2)

exp(cbind(OR = coef(logreg2), confint(logreg2)))
```

```
Call:
glm(formula = bac_over ~ typ_hvy + alcexp_m, family = "binomial"
    data = obs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7035  -0.9017   0.2473   0.9363   2.3172

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1129     0.1729  -0.653  0.51372
typ_hvy       2.1011     0.6454   3.256  0.00113 **
alcexp_m      1.2473     0.2661   4.688 2.76e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 276.28  on 199  degrees of freedom
Residual deviance: 214.34  on 197  degrees of freedom
AIC: 220.34

Number of Fisher Scoring iterations: 5

Waiting for profiling to be done...
                   OR     2.5 %    97.5 %
(Intercept) 0.8932098 0.6358694  1.255022
typ_hvy     8.1749305 2.6341400 36.013458
alcexp_m    3.4810138 2.1198312  6.040294
```

The intercept is the log odds of having a BAC over .08 among people who are light drinkers and have an average expectancy for alcohol use.

The slope for typ_hvy (2.10), is the predicted difference in the log odds of having a BAC over .08 for people who are heavy drinkers compared to those who are light drinkers (i.e., a 1 unit increase in typ_hvy), holding constant alcohol expectancies. By exponentiating this value we get the OR (8.17) and the 95% CI for the OR does not contain 1, indicating that the effect is statistically significant. Comparing people with the same level of alcohol expectancies, the odds that heavy drinkers will have a BAC over .08 are 8.17 times higher than light drinkers.

The slope for alcexp_m (1.25) is the predicted change in the log odds of having a BAC over .08 for a one unit increase in alcohol expectancies, holding constant typ_hvy. By exponentiating this value we get the OR (3.48) and the 95% CI for the OR does not contain 1, indicating that the effect is statistically significant. Comparing people with the same score for typ_hvy, the odds of having a BAC over .08 are 3.48 times higher for each one unit increase in alcohol expectancies.

Predicted Scores and Plots

Get a prediction matrix

```
pred_grid2 <- data_grid(obs, typ_hvy, alcexp_m = seq_range(alcexp_m, 10)) %>%
  add_predictions(logreg2) %>%
  mutate(odds = exp(pred),
         predprob = (odds/(1+odds)),
         typ_hvy = factor(typ_hvy, levels = c(0,1), labels = c("light drinker", "heavy drinker")),
         alcexp = alcexp_m + mean(obs$alcexp))
```

typ_hvy	alcexp_m	pred	odds	predprob	alcexp
light drinker	-2.0752	-2.70137966	0.06711286	0.06289199	2.01
light drinker	-1.6252	-2.14008405	0.11764495	0.10526147	2.46
light drinker	-1.1752	-1.57878844	0.20622480	0.17096714	2.91
light drinker	-0.7252	-1.01749284	0.36150014	0.26551605	3.36
light drinker	-0.2752	-0.45619723	0.63368884	0.38788833	3.81
light drinker	0.1748	0.10509838	1.11081988	0.52625044	4.26
light drinker	0.6248	0.66639398	1.94720300	0.66069524	4.71
light drinker	1.0748	1.22768959	3.41333421	0.77341394	5.16
light drinker	1.5248	1.78898519	5.98337742	0.85680281	5.61
light drinker	1.9748	2.35028080	10.48851450	0.91295654	6.06
heavy drinker	-2.0752	-0.60030744	0.54864294	0.35427336	2.01
heavy drinker	-1.6252	-0.03901183	0.96173933	0.49024828	2.46
heavy drinker	-1.1752	0.52228377	1.68587341	0.62768163	2.91
heavy drinker	-0.7252	1.08357938	2.95523857	0.74717075	3.36
heavy drinker	-0.2752	1.64487499	5.18036225	0.83819719	3.81
heavy drinker	0.1748	2.20617059	9.08087536	0.90080227	4.26
heavy drinker	0.6248	2.76746620	15.91824920	0.94089223	4.71
heavy drinker	1.0748	3.32876181	27.90377001	0.96540244	5.16
heavy drinker	1.5248	3.89005741	48.91369467	0.97996542	5.61
heavy drinker	1.9748	4.45135302	85.74287721	0.98847168	6.06

$$\hat{logit} = -.1129 + 2.1011 \cdot 0 + 1.2473 \cdot -2.0752 = -2.7014$$

$$\hat{odds} = e^{-2.7014} = .0671$$

$$\hat{p} = \frac{.0671}{1 + .0671} = .0629$$

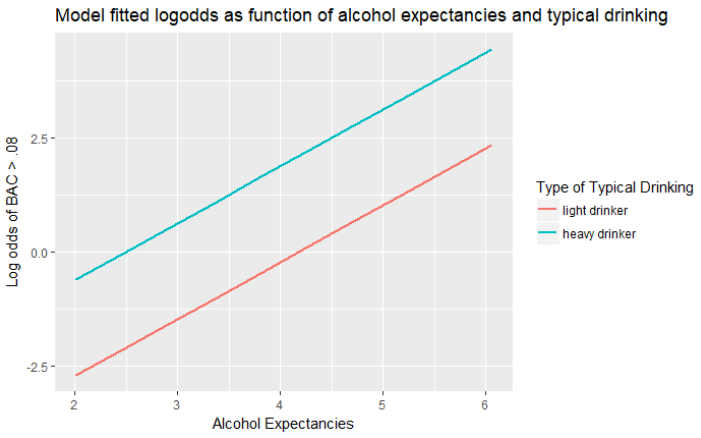
Plot the results

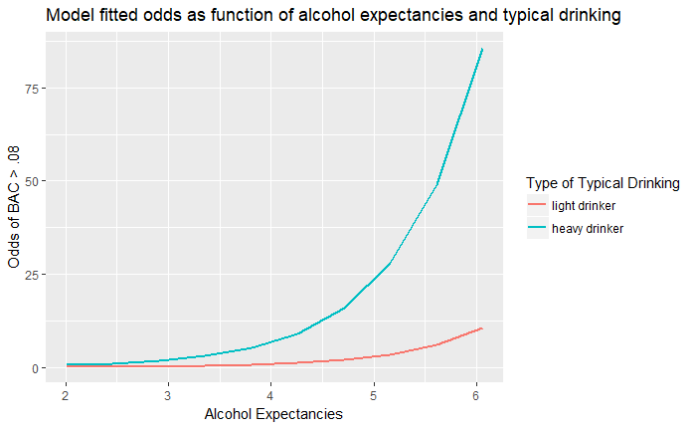
Make a plot of the model results - log odds

```
ggplot(pred_grid2, aes(x = alcexp, y = pred, group = factor(typ_hvy), colour = typ_hvy)) +
  geom_line(size = 1) +
  guides(color=guide_legend("Type of Typical Drinking")) +
  labs(title = "Model fitted log odds as function of alcohol expectancies and typical drinking",
       x = "Alcohol Expectancies", y = "Log odds of BAC > .08")
```

Make a plot of the model results - odds

```
ggplot(pred_grid2, aes(x = alcexp, y = odds, group = factor(typ_hvy), colour = typ_hvy)) +
  geom_line(size = 1) +
  guides(color=guide_legend("Type of Typical Drinking")) +
  labs(title = "Model fitted odds as function of alcohol expectancies and typical drinking",
       x = "Alcohol Expectancies", y = "Odds of BAC > .08")
```





On the log odds scale, the effect of alcohol expectancies on the outcome is constant. On the odds scale, it looks like the effect of alcohol expectancies is not constant, but it IS constant when you compute the odds ratio at different levels.

These are the predicted values obtained in the same way as before, but instead of a range of alc_exp ranging from lowest to highest, I asked for specific values: alcexp_m = c (-2,-1,0,1,2).

typ_hvy	alcexp_m	pred	odds	predprob
light drinker	-2	-2.6075809	0.07371264	0.06865212
light drinker	-1	-1.3602574	0.25659473	0.20419848
light drinker	0	-0.1129338	0.89320980	0.47179652
light drinker	1	1.1343898	3.10927564	0.75664811
light drinker	2	2.3817134	10.82343137	0.91542218
heavy drinker	-2	-0.5065087	0.60259575	0.37601232
heavy drinker	-1	0.7408149	2.09764411	0.67717402
heavy drinker	0	1.9881384	7.30192808	0.87954605
heavy drinker	1	3.2354620	25.41811230	0.96214718
heavy drinker	2	4.4827856	88.48079935	0.98882442

The change in the odds of BAC > .08 for a light drinker compared to a heavy drinker is the same at all levels of alcexp_m:

For alcexp_m = -1:
2.09764411/.25659473 = 8.17

For alcexp_m = 1:
25.41811230/3.10927564 = 8.17

The change in the odds of BAC > .08 is the same for any chosen 1 unit increase of alcexp_m, for both light and heavy drinkers:

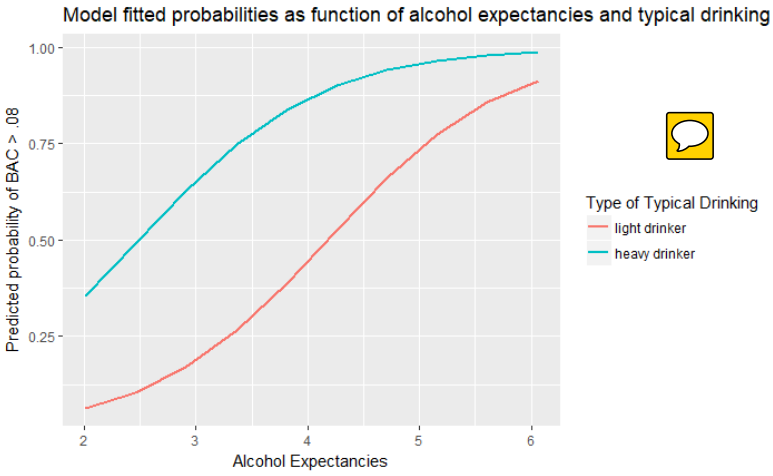
3.10927564/.89320980 = 3.48
10.82343137/3.10927564 = 3.48

2.09764411/.60259575 = 3.48
7.30192808/2.09764411 = 3.48

Plot the Results in the Probability Metric

Make a plot of the model results - probability

```
ggplot(pred_grid2, aes(x = alcexp, y = predprob, group = factor(typ_hvy), colour = typ_hvy)) +
  geom_line(size = 1) +
  guides(color=guide_legend("Type of Typical Drinking")) +
  labs(title = "Model fitted probabilities as function of alcohol expectancies and typical drinking",
       x = "Alcohol Expectancies", y = "Predicted probability of BAC > .08")
```



Unlike the situation for the interpretation of coefficients in the metric of log odds or odds, on the probability scale, the slope for alcohol expectancy changes, and the difference between the two lines at any given value of alcohol expectancies (the effect of heavy drinking status) also changes as we move across the x-axis. When we transform the results of a logistic regression model to consider the effects in the probability metric, the regression coefficients depend on the values of ALL of the predictors in the model. This is very different from linear regression, or a logistic regression when only the log odds or odds ratios are considered. In these latter cases, a partial regression coefficient doesn't depend on any other variables unless it is explicitly involved in an interaction. Therefore, if we want to discuss the effects in terms of probability, we need special techniques and tools to appropriately describe and evaluate the model.

The margins Package for the Probability Metric

The margins package can help us understand the effect of a covariate on the probability of y at different levels of the covariates. For a continuous covariate, margins computes the slope drawn tangent to the curvilinear line (i.e., see figure on previous page). For a categorical covariate, margins computes the effect of a change from one level to another (e.g., men compared to women). Because of this difference, we need to specify categorical predictors as factors before fitting the model. If you have multicategory variables, then just set the categorical variable as a factor rather than making dummy codes.

Obtain marginal effect, choose values of x

```
# make typ_hvy a factor
obs <- mutate(obs, typ_hvy.f = factor(typ_hvy, levels = c(0,1), labels = c("light", "heavy")))

# fit model to feed to margins (don't center any variables)
demo_margins <- glm(bac_over ~ typ_hvy.f + alcexp, data=obs, family=binomial("logit"))

# marginal effects at chosen levels of the covariates
pick <- margins(demo_margins, at = list(typ_hvy.f = c("light", "heavy"), alcexp = c(2:6)))
summary(pick)
```

Slope to interpret

Levels of the covariates

These are the slopes that represent the effect on the probability of the outcome given levels of the covariates

factor	typ_hvy	alcexp	AME	SE	z	p	lower	upper
alcexp	light	2	0.0727	0.0218	3.3388	0.0008	0.0300	0.1154
alcexp	light	3	0.1900	0.0220	8.6365	0.0000	0.1469	0.2331
alcexp	light	4	0.3081	0.0657	4.6864	0.0000	0.1793	0.4370
alcexp	light	5	0.2420	0.0295	8.2005	0.0000	0.1842	0.2999
alcexp	light	6	0.1054	0.0270	3.9095	0.0001	0.0526	0.1582
alcexp	heavy	2	0.2843	0.0530	5.3678	0.0000	0.1805	0.3881
alcexp	heavy	3	0.2824	0.1049	2.6929	0.0071	0.0769	0.4880
alcexp	heavy	4	0.1431	0.0757	1.8892	0.0589	-0.0054	0.2915
alcexp	heavy	5	0.0501	0.0284	1.7620	0.0781	-0.0056	0.1058
alcexp	heavy	6	0.0153	0.0099	1.5519	0.1207	-0.0040	0.0346
typ_hvyheavy	light	2	0.2893	0.1777	1.6283	0.1035	-0.0589	0.6375
typ_hvyheavy	light	3	0.4660	0.1469	3.1717	0.0015	0.1781	0.7540
typ_hvyheavy	light	4	0.4224	0.0830	5.0907	0.0000	0.2598	0.5850
typ_hvyheavy	light	5	0.2215	0.0597	3.7123	0.0002	0.1046	0.3385
typ_hvyheavy	light	6	0.0808	0.0421	1.9164	0.0553	-0.0018	0.1634
typ_hvyheavy	heavy	2	0.2893	0.1777	1.6283	0.1035	-0.0589	0.6375
typ_hvyheavy	heavy	3	0.4660	0.1469	3.1717	0.0015	0.1781	0.7540
typ_hvyheavy	heavy	4	0.4224	0.0830	5.0907	0.0000	0.2598	0.5850
typ_hvyheavy	heavy	5	0.2215	0.0597	3.7123	0.0002	0.1046	0.3385
typ_hvyheavy	heavy	6	0.0808	0.0421	1.9164	0.0553	-0.0018	0.1634

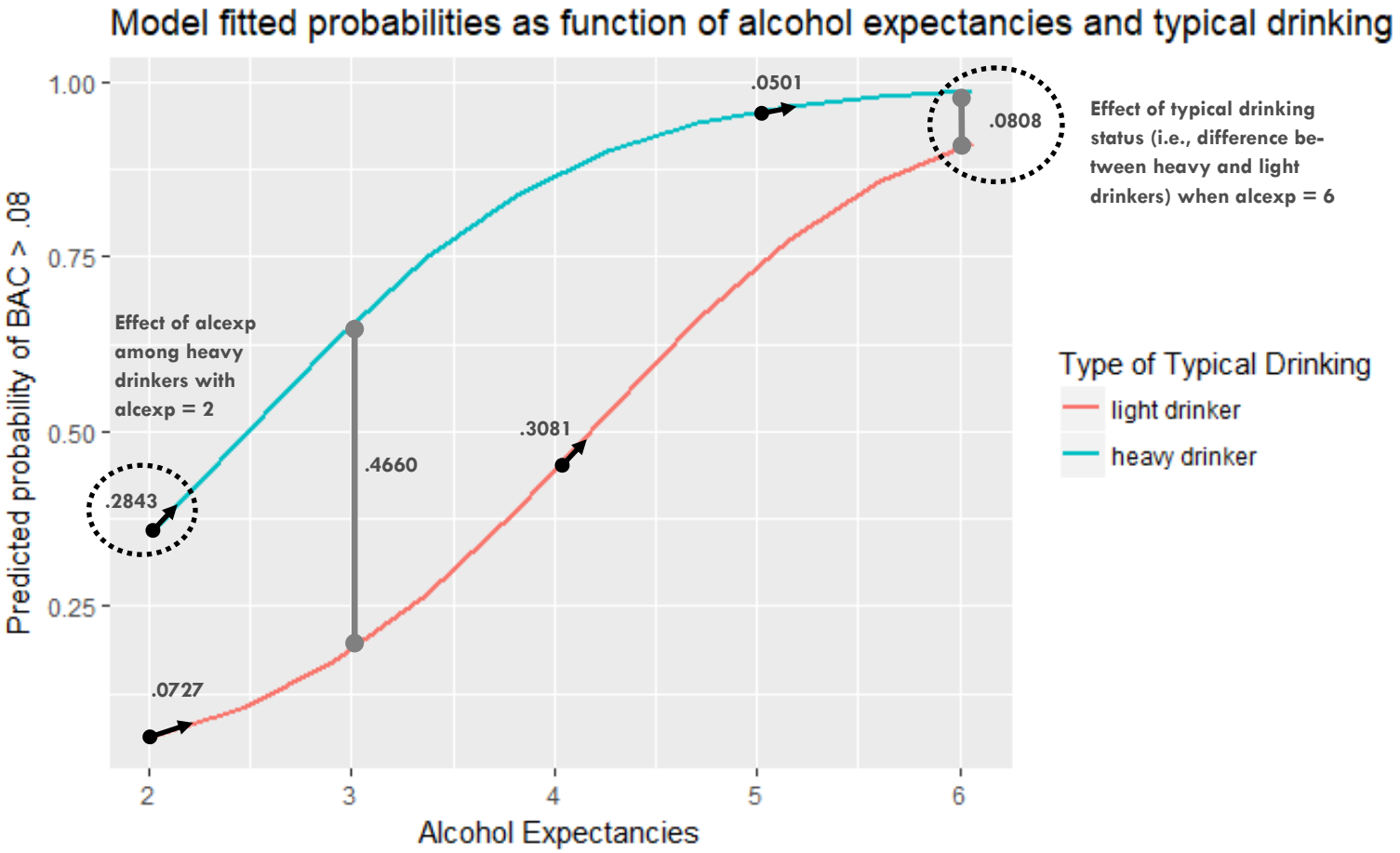
Here, we get the effect of the covariate (variable listed under factor) on the PROBABILITY of having a BAC over .08, at each combination and level of our covariates. The slope is labeled AME, for average marginal effect. We also get a standard error (SE), z*, p-value, and the 95% CI for each effect.

For the effect of alcohol expectancies (see rows where factor = alcexp), we can see that the effect is always significant for light drinkers, but for heavy drinkers, the effect of alcohol expectancies is only significant when alcohol expectancies are low. This IS NOT indicative of an interaction — it's just the nature of the effect on the probability scale.

For the effect of heavy drinking status (all rows where factor = typ_hvy), the difference is significant at the middle levels of alcohol expectancies, but not at the highest and lowest levels. Note that we get the same effects twice here because the effects are printed for each level of the covariate (you can ignore the lower set — i.e., the last five lines of the output).

Map the Results onto the Graph

factor	typ_hvy	alcexp	AME
alcexp	light	2	0.0727
alcexp	light	3	0.1900
alcexp	light	4	0.3081
alcexp	light	5	0.2420
alcexp	light	6	0.1054
alcexp	heavy	2	0.2843 ←
alcexp	heavy	3	0.2824
alcexp	heavy	4	0.1431
alcexp	heavy	5	0.0501
alcexp	heavy	6	0.0153
typ_hvyheavy	light	2	0.2893
typ_hvyheavy	light	3	0.4660
typ_hvyheavy	light	4	0.4224
typ_hvyheavy	light	5	0.2215
typ_hvyheavy	light	6	0.0808 ←
typ_hvyheavy	heavy	2	0.2893
typ_hvyheavy	heavy	3	0.4660
typ_hvyheavy	heavy	4	0.4224
typ_hvyheavy	heavy	5	0.2215
typ_hvyheavy	heavy	6	0.0808



Average Marginal Effects

The margins package will also give you the average of all of the marginal effects across all combinations.

Average marginal effects

```
ame <- margins(demo_margins)
summary(ame)
```

```
          factor    AME     SE      z      p  lower  upper
          alcexp 0.2266 0.0372 6.0999 0.0000 0.1538 0.2994
typ_hvy.fheavy 0.3651 0.0835 4.3712 0.0000 0.2014 0.5287
```

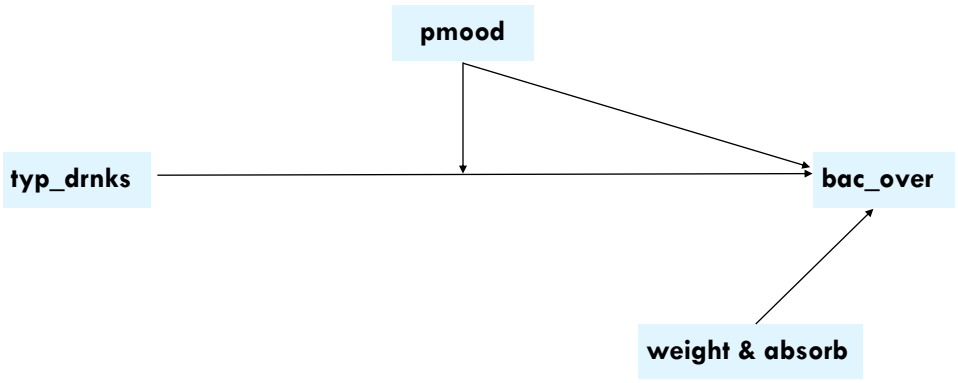
Rather than presenting the effects at each specified level of the covariates, these estimates average the effects across all levels. For alcexp, the .2266 represents the average effect (in the probability scale) across all levels of alcexp and heavy drinking status. For heavy drinking status, the .3651 represents the average difference in the predicted probability between heavy drinkers and light drinkers across all values of alcexp. Both are significantly different than zero.

Interactions in Logistic Regression

Examination of the effects of interactions in the metric of log odds works the same as it does for linear regression, and for Odds Ratios it is mostly the same. However, if your desire is to understand the interaction effects for a binary outcome in terms of probability, then the process becomes more complex. We'll start with the simpler case of understanding the interaction in terms of log odds and Odds Ratios, and then dig into the case of interactions in terms of probabilities.

Example 1

Continuing with the BAC study, let's imagine that a researcher wants to determine if the effect of typical drinking (the original continuous measure) on the odds of having a BAC over .08 differs by the degree to which the student is in a partying mood on the night of her 21st birthday. We will also control for the participants weight and the amount of food she had eaten during the day.



Fit logistic regression model with interaction

```
# center the predictors
obs <- mutate(obs,
            typ_drks_m = typ_drks - mean(typ_drks),
            pmood_m = pmood - mean(pmood),
            weight_m = weight - mean(weight),
            absorb_m = absorb - mean(absorb))

# fit the model
logint <- glm(bac_over ~ typ_drks_m + pmood_m + typ_drks_m*pmood_m + weight_m + absorb_m, data = obs,
  family = binomial("logit"))
summary(logint)

# get OR and CIs
exp(cbind(OR = coef(logint), confint(logint)))
```

Interpretation of Model with Interactions, Log Odds

```
Call:
glm(formula = bac_over ~ typ_drks_m + pmood_m + typ_drks_m *
    pmood_m + weight_m + absorb_m, family = binomial("logit"),
    data = obs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6388  -0.6589   0.1037   0.8213   1.9974

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.27270    0.19502   1.398  0.16202
typ_drks_m         0.14267    0.02291   6.226 4.77e-10 ***
pmood_m            0.46918    0.14413   3.255  0.00113 **
weight_m          -0.09015    0.02309  -3.904 9.47e-05 ***
absorb_m           0.22572    0.22467   1.005  0.31505
typ_drks_m:pmood_m 0.04797    0.01622   2.957  0.00311 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 276.28  on 199  degrees of freedom
Residual deviance: 172.40  on 194  degrees of freedom
AIC: 184.4
```

Interpretation of parameter estimates:

Intercept: The predicted log odds of having a BAC over .08 (bac_over) when all predictors are 0 (i.e., at the mean since we centered all of them).

typ_drks_m: The predicted change in the log odds of bac_over for a one unit increase in typical drinks among people who reported an average partying mood, holding constant weight and absorb. This effect is significant, indicating that among people with an average partying mood, more frequent typical drinking is associated with a higher log odds of having a BAC over .08. We can calculate the effect of typical drinks at any other value of partying mood in the usual way by multiplying the interaction slope by the value of pmood_m of interest and then adding this value to the slope for typ_drks_m. For example, the slope for typ_drks_m when pmood_m = 1 (one unit above the mean) is .14 + (.05 ▪ 1) = .19.

pmood_m: The predicted change in the log odds of bac_over for a one unit increase in partying mood among people who reported an average score for typical drinking, holding constant weight and absorb. This effect is significant, indicating that among people with an average level of typical drinking, being more in the mood to party on one's 21st birthday is associated with a higher log odds of having a BAC over .08.

weight_m: The predicted change in the log odds of bac_over for a one unit increase in weight, holding constant all other predictors. This is a significant difference, indicating that holding all other variables constant, each one unit increase in weight is associated with a lower log odds of having a BAC over .08.

absorb: The predicted change in the log odds of BAC over .08 for a one unit increase in absorb, holding constant all other predictors. This is not a significant effect.

typ_drks_m:pmood_m: This is the interaction term between typical drinks and mood. It is the predicted **difference** in the effect of typical drinks on log odds of bac_over for each one unit increase in pmood_m, holding constant all other predictors. Each one unit increase in typical drinks has a larger effect on the log odds of having a BAC over .08 as partying mood increases. It is statistically significant.

Following the estimation of a model with an interaction, when describing the model in terms of log odds, you can apply all of the same techniques that we used for linear modeling to calculate simple slopes, regions of significance, etc.

Interpretation of Model with Interaction, Odds Ratios

```
Waiting for profiling to be done...
                            OR     2.5 %    97.5 %
(Intercept)          1.3135008 0.9004153 1.9420633
typ_drks_m           1.1533436 1.1073422 1.2122545
pmood_m              1.5986811 1.2168780 2.1481586
weight_m             0.9137934 0.8707156 0.9536942
absorb_m             1.2532308 0.8112372 1.9680531
typ_drks_m:pmood_m   1.0491443 1.0174705 1.0848043
```

First, let's define what each value under OR generally represents. The value listed under OR for the intercept is the odds of having a BAC over .08 when all predictors are 0. The value listed under OR for typ_drks, pmood_m, weight_m, and absorb_m are all Odds Ratios. The value listed under OR for the interaction terms is a ratio of odds ratios.

OR for typ_drks_m is 1.15, and is interpreted as the expected change in the odds of having a BAC over .08 for a one unit increase in typical drinks among people with an average mood. We can use the coefficient for the interaction term to calculate the OR for the effect of typ_drks_m at different levels of mood. For example, let's calculate the OR when pmood_m is 1 (i.e., one unit above the mean). This is accomplished by **MULTIPLYING** the OR for typ_drks_m by 1 times the OR for the interaction. Note that this is different than in the log odds case in which we add the interaction term to the slope of the predictor. In the Odds Ratio metric, the interaction is multiplicative rather than additive.

OR when pmood_m = 1: 1.15 ▪ (1.05 ▪ 1) = 1.21

Plot the Model Results

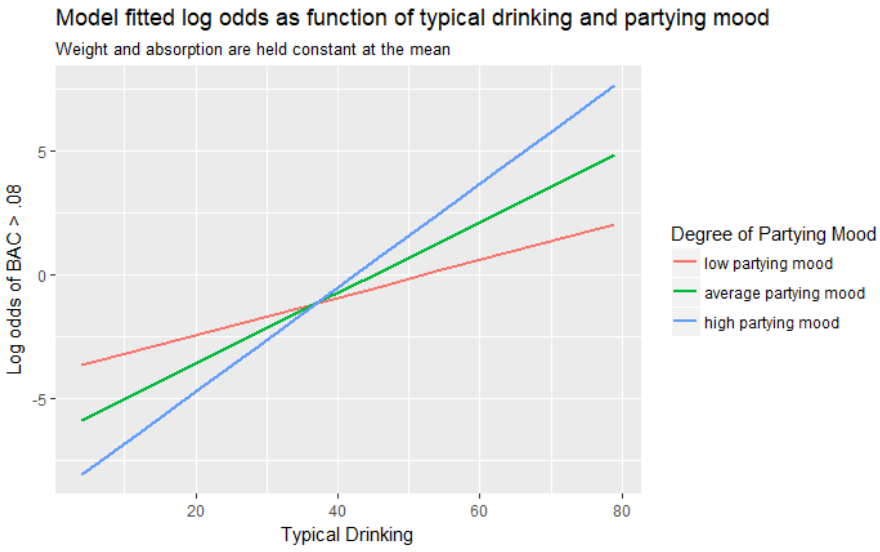
Prepare data for plotting

```
sd(obs$pmood)

pred_grid_int <- data_grid(obs,
                      typ_drks_m = seq_range(typ_drks_m, 10),
                      pmood_m = c(-1.396973, 0, 1.396973),
                      weight_m = 0,
                      absorb_m = 0) %>%
 add_predictions(logint) %>%
 mutate(odds = exp(pred),
          predprob = (odds/(1+odds)),
          typ_drks = typ_drks_m + mean(obs$typ_drks),
          pmood.f = factor(pmood_m, levels = c(-1.396973, 0, 1.396973),
               labels = c("low partying mood", "average partying mood", "high partying mood")))
```

Plot the model in terms of log odds

```
ggplot(pred_grid_int, aes(x = typ_drks, y = pred, group = pmood.f, colour = pmood.f)) +
 geom_line(size = 1) +
 guides(color=guide_legend("Degree of Partying Mood")) +
 labs(title = "Model fitted log odds as function of typical drinking and partying mood",
     subtitle = "Weight and absorption are held constant at the mean",
     x = "Typical Drinking", y = "Log odds of BAC > .08")
```



In terms of log odds you can do all of the same sorts of things that we did with linear models to depict the interaction effects. For example, plotting the simple slopes or creating a Johnson-Neyman graph, and the same techniques are used. You can also exponentiate the coefficients to get the effects in the metric of the OR. When you interpret the effects of the interactions in logistic regression in terms of odds, you are referring to the multiplicative interaction between the variables.

A New Example, and an Introduction to Additive Interactions in Logistic Regression

In logistic regression, we can consider two types of interaction: multiplicative interaction (the kind we just considered) and additive interaction. Multiplicative interaction happens on the odds scale, while additive interaction happens on the probability scale. It is easiest to understand the difference between multiplicative and additive interactions when considering two binary variables that each elicit some risk. To demonstrate this, we will build on the practice activity that you completed earlier in this unit. Hilt and colleagues (1986) were interested in the effect of asbestos exposure on lung cancer, and whether the ill effect of exposure on lung cancer differed by smoking status. Data from their paper is located in the datafile called asbestos.csv. Collapsing across smoking status to consider those who never smoked to those who were current or previous smokers and crossing this collapsed smoking status variable with asbestos exposure yields the table at the top of the next page.

Load libraries

```
library(tidyverse)
library(car)
library(margins)
library(descriptr)
```

Import data

```
asbestos <- read_csv("asbestos.csv")
```

Format variables

```
asbestos <- mutate(asbestos,
          smoker.f = factor(smoking_status, levels = c(1,2,3), labels = c("current", "former","never")),
          ev_smoker = ifelse(smoking_status == 3, 0, 1),
          ev_smoker.f = factor(ev_smoker, levels = c(0,1), labels = c("no", "yes")),
          asb_exp.f = factor(asbestos, levels = c(0,1), labels = c("no", "yes")),
          cancer.f = factor(lung_cancer, levels = c(0,1), labels = c("no", "yes")),
          category = ifelse(ev_smoker == 0 & asbestos == 0, 0,
                            ifelse(ev_smoker == 0 & asbestos == 1, 1,
                                   ifelse(ev_smoker == 1 & asbestos == 0, 2, 3))),
          category.f = factor(category, levels = c(0,1,2,3),
             labels = c("nonsmoker/no asbestos", "nonsmoker/asbestos", "smoker/no asbestos", "smoker/asbestos")))
```

Generate a table of descriptive statistics

```
asb_summary <- asbestos %>%
 group_by(ev_smoker.f, asb_exp.f, category.f) %>%
 summarize(count = n(),
         num_cancer = sum(lung_cancer),
         prop_cancer = num_cancer/count,
         odds_cancer = prop_cancer/(1-prop_cancer)) %>%
 ungroup()

asb_summary
```

Explore the Hilt Data — Consider the Probability of Cancer

ev_smoker.f <fctr>	asb_exp.f <fctr>	category.f <fctr>	count <int>	num_cancer <int>	prop_cancer <dbl>	odds_cancer <dbl>
no	no	nonsmoker/no asbestos	5057	6	0.001186474	0.001187884
no	yes	nonsmoker/asbestos	749	5	0.006675567	0.006720430
yes	no	smoker/no asbestos	12383	118	0.009529193	0.009620872
yes	yes	smoker/asbestos	3130	141	0.045047923	0.047172968

Let's start by looking at how the proportion of people with cancer varies across the four groups (column labeled prop_cancer above). About 4.5% of the people who smoked and were exposed to asbestos have cancer, that is a substantially higher percentage than for any of the other groups. It seems like the risk of getting cancer is much higher if both risk factors are present, in other words, it seems like smoking and asbestos exposure interact to predict cancer. The effect of the two risk factors together exceeds the effect of each risk factor considered individually. Epidemiologists call this comparison of the probabilities of the outcome across two binary risk exposures the interaction contrast (IC):

$$IC = \hat{p}[Y^{1,1} = 1] - \hat{p}[Y^{0,1} = 1] - \hat{p}[Y^{1,0} = 1] + \hat{p}[Y^{0,0} = 1]$$

probability of cancer if both a smoker and exposed to asbestos	probability of cancer if a nonsmoker, but exposed to asbestos	probability of cancer if a smoker, but not exposed to asbestos	probability of cancer if neither a smoker or exposed to asbestos	
.0450	.0067	.0096	.0012	= .03

The IC answers the following question: Compared to people with neither risk factor, does the presence of both risk factors have an effect on the probability of the outcome that is different from the sum of the effect of each risk factor separately?

If the IC is 0, the answer to this question is no, and there is no additive interaction between the two risk factors.

If the IC is positive, the interaction of the two risk factors is considered to be super-additive. Having both risk factors synergistically increases the probability of the outcome.

If the IC is negative the interaction of the two risk factors is considered to be sub-additive. Having both risk factors is associated with a lower probability of the outcome than would be expected based on the individual effect of each risk factor.

Consider the Odds Ratio of Asbestos Exposure Across Smoker Categories

ev_smoker.f <fctr>	asb_exp.f <fctr>	category.f <fctr>	count <int>	num_cancer <int>	prop_cancer <dbl>	odds_cancer <dbl>
no	no	nonsmoker/no asbestos	5057	6	0.001186474	0.001187884
no	yes	nonsmoker/asbestos	749	5	0.006675567	0.006720430
yes	no	smoker/no asbestos	12383	118	0.009529193	0.009620872
yes	yes	smoker/asbestos	3130	141	0.045047923	0.047172968

Among smokers, does asbestos exposure increase the odds of lung cancer?

```
print(OR_asbestos.exposure_smokers <- 0.047172968/0.009620872)
```

```
4.90319
```

Among smokers, the odds of cancer are about 4.9 times higher if exposed to asbestos.

Among nonsmokers, does asbestos exposure increase the odds of lung cancer?

```
print(OR_asbestos.exposure_nonsmokers <- 0.006720430/0.001187884)
```

```
5.65748
```

Among nonsmokers, the odds of cancer are about 5.7 times higher if exposed to asbestos.

Contrast these two odds ratios

```
mult_int <- OR_asbestos.exposure_smokers/OR_asbestos.exposure_nonsmokers
mult_int
```

```
0.8666739
```

This is a ratio of the odds ratios, and because it is smaller than one, the OR for smokers is smaller than the OR for nonsmokers (which of course we see above).

Fit Logistic Regression Model

Fit a logistic regression model to consider the interaction of asbestos exposure and smoking status

```
add_int <- glm(lung_cancer ~ asbestos + ev_smoker + asbestos*ev_smoker, data=asbestos, family = binomial("logit"))
summary(add_int)

exp(cbind(OR = coef(add_int), confint(add_int)))
```

```
Call:
glm(formula = lung_cancer ~ asbestos + ev_smoker + asbestos *
    ev_smoker, family = binomial("logit"), data = asbestos)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.3036  -0.1384  -0.1384  -0.1157   3.6706

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         -6.7356     0.4085 -16.489  < 2e-16 ***
asbestos             1.7330     0.6068   2.856  0.00429 **
ev_smoker            2.0918     0.4188   4.994  5.9e-07 ***
asbestos:ev_smoker  -0.1431     0.6198  -0.231  0.81742
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2895.8  on 21318  degrees of freedom
Residual deviance: 2635.7  on 21315  degrees of freedom
AIC: 2643.7

Number of Fisher Scoring iterations: 9

Waiting for profiling to be done...
                           OR        2.5 %       97.5 %
(Intercept)        0.001187884 0.0004719303  0.002408341
asbestos           5.657482070 1.6268547250 18.829435373
ev_smoker          8.099171070 3.8920599893 20.699273637
asbestos:ev_smoker 0.866673578 0.2541420326  3.083345820
```

By fitting a logistic regression model, we can determine if the ratio of ORs is statistically significant. The OR for the interaction term in the logistic regression model contrasts the OR for the effect of asbestos exposure for smokers and non-smokers. The OR for the effect of asbestos exposure among non-smokers is greater than that for smokers because the base rate of lung cancer is smaller for non-smokers not exposed to asbestos (the reference category for the nonsmokers' model) as compared to smokers not exposed to asbestos (the reference category for the smokers' model). Thus the ratio of the odds ratios across these two groups (i.e., the interaction term) is less than one (although not significantly different than 1), capturing the fact that the OR for nonsmokers is larger than that for smokers.

The interaction on the odds ratio scale is considered multiplicative. This is demonstrated by the manner in which we calculate the effect of asbestos exposure on lung cancer for smokers using the odds ratios: $5.66 \cdot (1 \cdot .87) = 4.90$.

Additive vs. Multiplicative Interaction for Risk Exposures

In sum, on the additive scale, where we are concerned with probabilities, we see that there is a super-additive effect of having both risk factors. On the multiplicative scale, where we are concerned with odds, we see a sub-multiplicative effect. Which one is correct? Well, they both are, they are just answering different questions.

From a public health perspective, the additive interaction results may be more important. Consider the prop_cancer column in our descriptives table again. The effect of asbestos exposure on the probability scale among nonsmokers is .006675567 - .001186474 = .005 (so fewer than 1 person per 100 people), while the effect of asbestos exposure on the probability scale among smokers is .045047923 - .009529193 = .036 (so nearly 4 people per 100 people). If we worked to prevent the exposure of asbestos among smokers, we would expect to be able to prevent more cases of lung cancer than if we worked to prevent the exposure of asbestos among nonsmokers.

ev_smoker.f <fctr>	asb_exp.f <fctr>	category.f <fctr>	count <int>	num_cancer <int>	prop_cancer <dbl>	odds_cancer <dbl>
no	no	nonsmoker/no asbestos	5057	6	0.001186474	0.001187884
no	yes	nonsmoker/asbestos	749	5	0.006675567	0.006720430
yes	no	smoker/no asbestos	12383	118	0.009529193	0.009620872
yes	yes	smoker/asbestos	3130	141	0.045047923	0.047172968

Additional Methods for Presenting the Interaction Effects

When presenting the interactive effects of two binary exposures, it's best to present the results on both the additive and multiplicative scale. Let's consider a couple of additional techniques that you can consider to more thoroughly present your results.

First, presenting the results of a logistic regression model in which the same reference group is used for all comparisons is helpful. Here, for our four groups, we could select the nonsmoking participants who weren't exposed to asbestos as the reference group for comparing to all three other groups.

Compare all groups to the lowest risk group

```
print(OR_nonsmoker.asbestos <- 0.006720430/0.001187884)
print(OR_smoker.no_asbestos <- 0.009620872/0.001187884)
print(OR_smoker.asbestos <- 0.047172968/0.001187884)


new_int <- glm(lung_cancer ~ category.f, data = asbestos, family = binomial("logit"))
summary(new_int)

exp(cbind(OR = coef(new_int), confint(new_int)))
```

```
[1] 5.65748
[1] 8.099168
[1] 39.71176

Call:
glm(formula = lung_cancer ~ category.f, family = binomial("logit"),
    data = asbestos)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.3036  -0.1384  -0.1384  -0.1157   3.6706

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -6.7356     0.4085 -16.489  < 2e-16 ***
category.fnonsmoker/asbestos   1.7330     0.6068   2.856  0.00429 **
category.fsmoker/no asbestos   2.0918     0.4188   4.994  5.9e-07 ***
category.fsmoker/asbestos      3.6816     0.4175   8.819  < 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: 2895.8  on 21318  degrees of freedom
Residual deviance: 2635.7  on 21315  degrees of freedom
AIC: 2643.7

Number of Fisher Scoring iterations: 9

Waiting for profiling to be done...
                                      OR        2.5 %       97.5 %
(Intercept)                 0.001187884 4.719303e-04 2.408341e-03
category.fnonsmoker/asbestos  5.657482070 1.626855e+00 1.882944e+01
category.fsmoker/no asbestos  8.099171070 3.892060e+00 2.069927e+01
category.fsmoker/asbestos    39.711776451 1.914924e+01 1.012926e+02
```

Here we see that the odds of cancer are substantially higher for all groups compared to the lowest risk group (i.e., reference group — non-smokers not exposed to asbestos).

Additive Interactions: Relative Excess Risk Due to Interaction

From the calculated odds ratios with a common reference group, we can calculate two additional statistics to present the effects: the Relative Excess Risk due to Interaction (RERI) and the Associated Proportion (AP).

```
print(OR_nonsmoker.asbestos <- 0.006720430/0.001187884)
print(OR_smoker.no_asbestos <- 0.009620872/0.001187884)
print(OR_smoker.asbestos <- 0.047172968/0.001187884)
```

```
[1] 5.65748
[1] 8.099168
[1] 39.71176
```

***RERI* = OR_{11} - OR_{10} - OR_{01} + *1* = 39.71176 - 8.099168 - 5.65748 + 1 = 26.955**

***AP* = *RERI* / OR_{11} = 26.95512 / 39.71176 = .679**

OR_{11} = Exposed to both risk factors

OR_{10} = Exposed to first risk factor only

OR_{01} = Exposed to second risk factor only

Calculate the RERI

```
print(RERI <- OR_smoker.asbestos - OR_smoker.no_asbestos - OR_nonsmoker.asbestos + 1)
```

```
26.95512
```

Calculate the AP

```
print(AP <- RERI/OR_smoker.asbestos)
```

```
0.678769
```

The RERI represents the additional risk (in this case expressed in terms of the OR) associated with exposure to both risk factors. The AP represents the proportion of the outcome (e.g., lung cancer) among those with both exposures that is attributable to their interaction. Mathur and Vanderweele (2018) provide a R function to determine if the RERI is statistically significant. To use it, put the R program called "Additive Interactions Function.R" in the same project folder as the Notebook that you are running. Then execute the code below. You call in the logistic regression model in which the outcome is regressed on the two risk factors and the interaction between the two (we called this model "add_int"). This function requires installing a package called msm, so please install it first: install.packages("msm").

A significance test for RERI

```
source("Additive interactions function.R")

additive_interactions(add_int)
```

Stat <fctr>	Est <dbl>	CI.lo <dbl>	CI.hi <dbl>	p.val.0 <dbl>	p.val.epi <dbl>	p.val.suff.cause <dbl>
RERI	26.9551233	4.51178044	49.3984662	9.286927e-03	0.01465409	0.01170594
AP	0.6787690	0.53588626	0.8216518	0.000000e+00	*NA*	*NA*
asbestos	0.1203118	-0.01116065	0.2517842	3.643985e-02	*NA*	*NA*
ev_smoker	0.1833853	0.13110168	0.2356689	3.108402e-12	*NA*	*NA*
asbestos:ev_smoker	0.6963029	0.54812102	0.8444848	0.000000e+00	*NA*	*NA*

The p-value for the RERI is small (see p-value under p.val.0 for RERI), indicating that the additive interaction is statistically significant. The Mathur and Vanderweele paper (in the Unit 11 folder) describes the other values in the table.

A Few Last Items for Logistic Regression: Assessment of the Overall Model

Recall that with linear models, we also obtained information about the overall model fit, including the R^2 and a F-test. We can assess something similar for logistic regression to examine overall model fit. At the bottom of the output you see the null and residual deviance and the AIC. To determine if the specified model fits better than a null model (i.e., a model with no predictors), we need to compute the difference between the residual deviance for the model with predictors and the null model. This difference is chi-squared distributed with degrees of freedom equal to the differences in degrees of freedom between the current and the null model (i.e., the number of predictor variables in the model). This is called a likelihood ratio test.

Likelihood ratio test for the null and full model

```
# difference in log-likelihoods
with(logreg2, null.deviance - deviance)

# difference in degrees of freedom
with(logreg, df.null - df.residual)

# p-value for difference in log-likelihoods
with(logreg2, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
```

```
[1] 61.93736
[1] 2
[1] 3.552006e-14
```

```
Call:
glm(formula = bac_over ~ typ_hvy + alcexp_m, family = "binomial"
    data = obs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7035  -0.9017   0.2473   0.9363   2.3172

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1129     0.1729  -0.653  0.51372
typ_hvy       2.1011     0.6454   3.256  0.00113 **
alcexp_m      1.2473     0.2661   4.688 2.76e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 276.28  on 199  degrees of freedom
Residual deviance: 214.34  on 197  degrees of freedom
AIC: 220.34

Number of Fisher Scoring iterations: 5

Waiting for profiling to be done...
                   OR     2.5 %    97.5 %
(Intercept) 0.8932098 0.6358694  1.255022
typ_hvy     8.1749305 2.6341400 36.013458
alcexp_m    3.4810138 2.1198312  6.040294
```

The difference in the deviance between the null and full model is 61.937 (i.e., 276.28 - 214.34). There is a difference of 2 df between these two models. The p-value to assess whether the full model is significantly better than the reduced model is very small $p < .001$ (listed in the output in scientific notation as 3.552006e-14). This tells us that these two predictors together significantly predict our outcome (i.e., our model fits significantly better than a model with no predictors).

You can also compare two different nested models in the same way we learned about for linear models. Once each model is estimated, you would execute the following code:

```
anova(M0, M1, test = "Chisq"))
```

Where M0 and M1 are your two nested models.

In addition you can compare two models (whether nested or non-nested) using the AIC (Akaike Information Criterion). A better model is a model with a lower AIC. There is no formal significance test for comparing two models using the AIC.

Evaluate the Accuracy of Our Predictions

If the intent of our model is to most accurately predict the outcome, we can evaluate the accuracy of our predictions. Let's return to the multiple logistic regression with the BAC data (logreg2). To do this, we first use the predicted probabilities of our model for each case to classify the case as either a 1 (predicted to have a BAC over .08) or a 0 (predicted to have a BAC less than or equal to .08). We will use a threshold of .5. If the predicted probability of the outcome is above .5 we will classify the prediction as a 1, otherwise if the predicted probability is less than or equal to .5, we will classify the prediction as a 0.

	id	predprob	classify
1	1	0.13549766	0
2	2	0.33055810	0
3	3	0.92452846	1
4	4	0.09841834	0
5	5	0.43006844	0
6	6	0.96413060	1
7	7	0.17274232	0
8	8	0.90517228	1
9	9	0.73896403	1
10	10	0.21342925	0

Obtain a confusion matrix to evaluate the accuracy of the predictions

```
obs_pred <- obs %>%
  add_predictions(logreg2) %>%
  mutate(predprob = exp(pred) / (1 + exp(pred)),
         classify = ifelse(predprob > .5, 1, 0))

conf_matrix <- ds_cross_table(obs_pred$classify, obs_pred$bac_over)
conf_matrix
```

Confusion Matrix

```
Total Observations:  200

-------------------------------------------------------------
|             |                bac_over                     |
-------------------------------------------------------------
|    classify |            0 |            1 |    Row Total |
-------------------------------------------------------------
|           0 |           68 |           31 |           99 |
|             |   TN   0.34  |   FN  0.155  |              |
|             |         0.69 |         0.31 |          0.5 |
|             |         0.73 |         0.29 |              |
-------------------------------------------------------------
|           1 |           25 |           76 |          101 |
|             |   FP  0.125  |   TP   0.38  |              |
|             |         0.25 |         0.75 |          0.5 |
|             |         0.27 |         0.71 |              |
-------------------------------------------------------------
| Column Total |           93 |          107 |          200 |
|             |        0.465 |        0.535 |              |
-------------------------------------------------------------
```

A confusion matrix is a table that describes the performance of a logistic regression model. The matrix compares the predicted outcomes (labeled classify in the table) to the observed outcomes (labeled bac_over in the table). Four cells emerge:

1. True Positives (TP): We predict a score of 1 (BAC > .08) and they do have a score of 1.
2. True Negatives (TN): We predict a score of 0 (BAC <= .08), and they have a score of 0.
3. False Positives (FP): We predict a score of 1, but they actually have a score of 0.
4. False Negatives (FN): We predict a score of 0, but they actually have a score of 1.

Using the Confusion Matrix, we can calculate two statistics that further describe the accuracy of our predictions: Sensitivity and Specificity. Sensitivity (also referred to as the True Positive Rate) is the percentage of 1's actually observed that were correctly predicted by our model. Specificity (also referred to as the True Negative Rate) is the percentage of 0's actually observed that were correctly predicted by our model.

Sensitivity = TP/(TP + FN) = 76/107 = .71

Specificity = TN/(TN + FP) = 68/93 = .73

So, when people actually do have a BAC over .08, our model accurately predicts this 71% of the time. And, when people actually do have a BAC at .08 or under, our model accurately predicts this 73% of the time.

Assumptions of a Logistic Regression Model

1. There is a linear relationship between the continuous predictors in your model and the logit transformation of your outcome.
2. The observations are independent. If they are not, you must adjust for the nesting or estimate a multilevel model.
3. Note that there is NO assumption of normally distributed errors/residuals or homogeneity of variance of the errors/ residuals like in OLS regression.

There are two papers by Zhongheng Zhang in the Unit 11 Dropox folder that show examples of model checking and diagnostics for logistic regression in R. He uses the car package, the same that you studied and utilized in Unit 9, so most everything will be familiar to you.