

Arizona State University

From the Selected Works of Joseph M Hilbe

November 2, 2016

Calculating Odds Ratios from Probabilities

Joseph M Hilbe



Available at: https://works.bepress.com/joseph_hilbe/76/

Calculating Odds Ratios from Probabilities

Joseph M. Hilbe

Arizona State University; Jet Propulsion Laboratory

hilbe@asu.edu : © 2014 J Hilbe 30Jun, 2014

1 Defining Terms

Odds ratios can at times be confusing. This appears to particularly be the case when attempting to derive the odds ratios of model explanatory predictors from the model case-wise predicted probabilities. The task is to use observation probabilities to determine parameter estimates.

Typically, most analysts consider that logistic regression can be used to either,

- a) *determine the predicted probability that the binary response term (0,1) is equal to 1.*
- or to
- b) *obtain the odds ratios, or exponentiated parameter estimates (coefficients), of the various explanatory predictors in the model.*

The predicted or fitted values of a logistic regression are many times used for the purpose of classification. But predicted probabilities are likely used more to help the analyst, or others who are relying on the model results, to make decisions based on the model. Suppose we read an article detailing study results concerning a logistic model with *died of lung cancer or emphysema* as the response variable and *smoked most of the patient's adult life, current age or age at death, gender, fitness level for age, and obesity index value* as explanatory predictors. If we learn that unfit obese smoking female patients in the study have a probability of death by means of lung cancer or emphysema of 94%, and fit non-obese female non-smoking patients have a similar probability of death of 17%, we may well --- given that we are female --- stop smoking, hit the gym, and lose weight. I simply made-up this study and its values, but the point remains that the predicted probabilities calculated from a well-fitted logistic model are many times used to make decisions, or to understand some issue in a better manner.

The fitted value of a logistic model is a predicted probability that the response (dependent variable, or y) has the value of 1. This is generally considered to be the level (0, 1) of the response which is of interest to us. Separate probability values are derived for each pattern of covariates in the model. Consider the following data, which we structure as a model.

y	X1	X2	X3
1	1	1	5
1	0	2	2
0	0	2	2
1	0	1	8
0	1	3	4
1	1	3	4

- y is a binary response with values of 0,1.
- X1 is a binary predictor with values of 0,1.
- X2 is a categorical predictor, with values of 1,2, and 3.
- X3 is a continuous predictor, with values ranging from 2-8.

There are six observations in the model, but only four covariate patterns. A covariate pattern is distinct set of predictor values, the response term not being involved. The four patterns are

COVARIATE PATTERNS	NUMBER OF OBSER. FOR EACH PATTERN	NUMBER OF Y=1's EACH PATTERN
1-1-5	1	1
0-2-2	2	1
0-1-8	1	1
1-3-4	2	1

The notion of covariate patterns and how many times $y=1$ is the case for each pattern is the basis of establishing a grouped logistic model, which I do not address in this brief monograph. The logic of what is discussed here though is directly applicable for grouped as well as for observation-based models.

Traditionally, probabilities are symbolized as $\hat{\mu}$, $\hat{\pi}$, \hat{p} , or \hat{y} . The *hat* over each letter indicates that the value is estimated. When the terms are understood to be estimated values, they are at times displayed without the hat. \hat{y} is rarely used to represent the predicted fit of a logistic model, which is defined as the probability that y , the response, is equal to 1.

For this monograph, the response term of a logistic regression is a binary variable, having values of 0 and 1. I shall generically refer to it as y . The predicted probability that $y=1$ will be symbolized as μ , or μ . The hat over the term is to be understood. The linear predictor is referred to as either xb or the Greek letter η (eta). It is defined as the sum of the product of regression coefficients and their respective predictor values, including the intercept.

$$xb_i = \sum_{i=1}^n \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_j X_{ij}$$

where β_0 is the regression intercept and β_1 through β_j are the coefficients of the various model predictors, X_1 through X_j . The subscript i is the observation number of the regression terms. The number of observations in the model is symbolized as n . For example, suppose that the coefficients and predictor values of an observation in a logistic regression appears as follows

$$xb = 1.5 + 0.5 * 1 - 0.25 * 6 + 2 * 0$$

With the intercept having a value of 1.5, the three coefficients are 0.5, -0.25, and 2, and associated predictor values are 1, 6, and 0. The linear predictor for that observation is

$$1.5 + 0.5*1 - 0.25*6 + 2*0 = 0.5$$

As explained in more detail later in this monograph, the fitted or predicted probability that the response term is equal to 1 given the linear predictor may be calculated using the *inverse logistic link function*

$$\frac{\exp(xb)}{1 + \exp(xb)} = \frac{1}{1 + \exp(-xb)}$$

The predicted probability of the linear predictor value, 0.5, is 0.62246. This predicted probability is also referred to as μ , or *mu*.

2 Model with a single Binary predictor --- 9 observations

We can imagine that we create a logistic regression from some simple data. We then calculate the predicted fit, or the probability that the response (y) is one. I then convert the fitted probabilities to observation odds ratios and show how to determine the model odds ratios and intercept from them. I describe this relationship for a binary predictor, more than one predictor, for a categorical response, for a categorical response and associated binary predictor. Finally I show how to determine odds ratio of a continuous predictor based on the individual fitted values.

Our initial data set consists of a response term, y, and one binary predictor, x. R and Stata are used to display examples. R code is displayed in separate tables. I advise typing the code, or pasting it, into the respective software editor:

R: click on *File > New Script*

Stata: type *doedit* on the command line, or select it from the menu.

R : mod1 (data)

```
=====
y <- c(1,1,0,0,1,0,0,1,1)
x <- c(0,1,1,1,0,0,1,0,1)
table(y,x)
=====
```

STATA

```
. l y x

+-----+
| y   x |
+-----+
1. | 1   0 |
2. | 1   1 |
3. | 0   1 |
4. | 0   1 |
5. | 1   0 |
+-----+
6. | 0   0 |
7. | 0   1 |
8. | 1   0 |
9. | 1   1 |
+-----+
```

```
. tab y x
```

y	x		Total
	0	1	
0	1	3	4
1	3	2	5
Total	4	5	9

The odds ratio of X may easily be calculated as follows:

```
ODDS X==1
. di 2/3
.66666667
```

```
ODDS X==0
. di 3/1
3
```

```
ODDS RATIO X==1 to X==0
. di (2/3) / (3/1)
.22222222
```

That is...

=====

To obtain the odds of x==1: for x==1, take the ratio of y==1 to y==0, or 2/3 = 0.666667.
 To obtain the odds of x==0: for x==0, take the ratio of y==1 to y==0, or 3/1 = 3.

To obtain the odds ratio of x==1 to x==0, divide. Therefore, 0.666667/3 = 0.222222

=====

A logistic regression model may be presented displaying a table of coefficients and associated standard errors, Z values, *p*-values and confidence intervals. Confidence intervals are calculated separately when using R. Coefficients are also referred to as slopes and parameter estimates. They are parameter estimates since, as shall be discovered, we may determine the odds ratios of each predictor given the predicted probabilities of the model. The natural log of a logistic regression odds ratio is a predictor coefficient or slope.

$$\log(\text{Odds Ratio}) = \text{coefficient}$$

$$\exp(\text{coefficient}) = \text{odds ratio}$$

Related model statistics can be determined as:

```
COEFFICIENT OF X
. di log(.22222222)
-1.5040774
```

```
ODDS RATIO X==0 to X==1
. di 1/.22222222
4.5
```

```
INTERCEPT {when model parameterized as odds ratio(s)}
. di 3/1 /* ODDS X==0 */
3
```

```
INTERCEPT {when model parameterized as coefficient(s)}
. di log(3) /* COEFFICIENT intercept*/
1.0986123
```

Now we determine how well our hand calculations compare to actually modeling the data using a logistic model.

R : mod1

```
=====
summary( mod1 <- glm( y ~ x, family=binomial))
exp(coef(mod1))
xb <- mod1$linear.predictors
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data1 <-data.frame(y,x,mu,xb,o)
data1
=====
```

```
. glm y x, fam(bin) nolog nohead
```

		OIM				
	y	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
x		-1.504077	1.47196	-1.02	0.307	-4.389066 1.380911
_cons		1.098612	1.154701	0.95	0.341	-1.164559 3.361784

Recall that parameter odds ratios are obtained by exponentiating the coefficients.

```
. glm y x, fam(bin) nolog nohead eform
```

		OIM				
	y	Odds Ratio	Std. Err.	z	P> z	[95% Conf. Interval]
x		.2222222	.3271023	-1.02	0.307	.0124123 3.978526
_cons		3	3.464102	0.95	0.341	.3120602 28.84059

```
. predict mu
. predict xb, xb
. l
```

	y	x	mu	xb
1.	1	0	.75	1.098612
2.	1	1	.4	-.4054651
3.	0	1	.4	-.4054651
4.	0	1	.4	-.4054651
5.	1	0	.75	1.098612
6.	0	0	.75	1.098612
7.	0	1	.4	-.4054651
8.	1	0	.75	1.098612
9.	1	1	.4	-.4054651

The odds is calculated as $\mu/(1-\mu)$. Note that the log of the odds is the logit link function, and the basis of the name, logistic regression. In this manner the predicted probability for each observation may be converted to an odds.

```
. gen o=mu/(1-mu)
. l
```

	y	x	mu	xb	o
1.	1	0	.75	1.098612	3
2.	1	1	.4	-.4054651	.6666667
3.	0	1	.4	-.4054651	.6666667
4.	0	1	.4	-.4054651	.6666667
5.	1	0	.75	1.098612	3
6.	0	0	.75	1.098612	3
7.	0	1	.4	-.4054651	.6666667
8.	1	0	.75	1.098612	3
9.	1	1	.4	-.4054651	.6666667

To calculate the model odds ratio of X, divide odds of $x==1$ by odds of $x==0$

```
. di .6666667/3
.22222223
```

This value corresponds to what is displayed in the regression output. The intercept is simply the odds of $x==0$, or 3.

I shall add a few more observations to determine if this same relationship between the predicted probabilities and odds ratio of a parameter estimate.

3 Add 5 more observations

R : mod2 (data)

```
=====
y <- c(1,1,0,0,1,0,0,1,1,1,1,0,1,0)
x <- c(0,1,1,1,0,0,1,0,1,0,1,1,1,1)
table(y,x)
=====
```

```
. tab y x
```

	x		
y	0	1	Total
0	1	5	6
1	4	4	8
Total	5	9	14

```
. 1
```

```

+-----+
| y   x |
+-----+
1. | 1   0 |
2. | 1   1 |
3. | 0   1 |
4. | 0   1 |
5. | 1   0 |
+-----+
6. | 0   0 |
7. | 0   1 |
8. | 1   0 |
9. | 1   1 |
10. | 1   0 |
+-----+
11. | 1   1 |
12. | 0   1 |
13. | 1   1 |
14. | 0   1 |
+-----+

```

R : mod2

```

=====
summary( mod2 <- glm( y ~ x, family=binomial))
exp(coef(mod2))
xb <- mod2$linear.predictors
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data2 <-data.frame(y,x,mu,xb,o)
data2
=====

```

```
. drop mu xb o
. glm y x, nolog nohead fam(bin)
```

```

-----
          |
          y |      Coef.      OIM      Std. Err.      z      P>|z|      [95% Conf. Interval]
-----+-----
          x | -1.609438      1.30384      -1.23      0.217      -4.164918      .9460425
      _cons |  1.386294      1.118034      1.24      0.215      -.805012      3.577601
-----

```

```
. glm y x, nolog nohead fam(bin) eform
```

```

-----
          |
          y | Odds Ratio      OIM      Std. Err.      z      P>|z|      [95% Conf. Interval]
-----+-----
          x |      .2      .2607681      -1.23      0.217      .015531      2.575497
      _cons |      4      4.472136      1.24      0.215      .4470826      35.78757
-----

```

```
. predict mu
. predict xb, xb
. gen o=mu/(1-mu)
```



```
. 1
```

	y	x	mu	xb	o
1.	1	0	.8	1.386294	4
2.	1	1	.4444444	-.2231435	.8
3.	0	1	.4444444	-.2231435	.8
4.	0	1	.4444444	-.2231435	.8
5.	1	0	.8	1.386294	4
6.	0	0	.8	1.386294	4
7.	0	1	.4444444	-.2231435	.8
8.	1	0	.8	1.386294	4
9.	1	1	.4444444	-.2231435	.8
10.	1	0	.8	1.386294	4
11.	1	1	.4444444	-.2231435	.8
12.	0	1	.4444444	-.2231435	.8
13.	1	1	.4444444	-.2231435	.8
14.	0	1	.4444444	-.2231435	.8

The model odds ratio for X is ratio of odds for $x=1/x=0$

```
. di .8/4
.2
```

The exponentiated Intercept $x=0 = 4$. The true model intercept is $\log(4)$, or 1.386294. This value is the same as displayed in the above regression output.

When we add another predictor to the model, the relationship between the probabilities and associated odds is more complex.

4 Add extra covariate

```
. glm y x1 x2, fam(bin) nolog nohead
```

		Coef.	OIM Std. Err.	z	P> z	[95% Conf. Interval]	
y							
x1		-1.817754	1.587367	-1.15	0.252	-4.928935	1.293427
x2		2.836778	1.492793	1.90	0.057	-.0890423	5.762599
_cons		.2607284	1.309508	0.20	0.842	-2.305859	2.827316

```
. glm y x1 x2, fam(bin) nolog nohead eform
```

		Odds Ratio	OIM Std. Err.	z	P> z	[95% Conf. Interval]	
y							
x1		.1623901	.2577726	-1.15	0.252	.0072342	3.645258
x2		17.06071	25.46811	1.90	0.057	.9148069	318.1741
_cons		1.297875	1.699577	0.20	0.842	.0996731	16.90004

R : mod3

```
=====
y <- c(1,0,1,1,0,0,0,1,0,1,1,1,1,0)
x1 <- c(1,1,1,1,1,1,1,1,1,0,0,0,0,0)
x2 <- c(1,1,1,1,0,0,0,0,0,1,1,1,0,0)
summary( mod3 <- glm( y ~ x1 + x2, family=binomial))
exp(coef(mod3))
xb <- mod3$linear.predictors
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data3 <-data.frame(y,x1,x2,mu,xb,o)
data3
=====
```

```
. drop mu xb o
. predict xb, xb
. predict mu
. gen o=mu/(1-mu)
. gsort -x1 -x2
```

```
. 1
+-----+
| y   x1  x2      mu      xb      o |
+-----+
1. | 1   1   1   .7824076   1.279753   3.59575 |
2. | 0   1   1   .7824076   1.279753   3.59575 |
3. | 1   1   1   .7824076   1.279753   3.59575 |
4. | 1   1   1   .7824076   1.279753   3.59575 |
5. | 0   1   0   .1740739  -1.557026   .210762 |
+-----+
6. | 0   1   0   .1740739  -1.557026   .210762 |
7. | 0   1   0   .1740739  -1.557026   .210762 |
8. | 1   1   0   .1740739  -1.557026   .210762 |
9. | 0   1   0   .1740739  -1.557026   .210762 |
10. | 1   0   1   .9567898   3.097507  22.14268 |
+-----+
11. | 1   0   1   .9567898   3.097507  22.14268 |
12. | 1   0   1   .9567898   3.097507  22.14268 |
13. | 1   0   0   .5648153   .2607284  1.297875 |
14. | 0   0   0   .5648153   .2607284  1.297875 |
+-----+
```

```
3.59575   x1=1; x2=1
22.14268  x1=0; x2=1
.210762   x1=1; x2=0
1.297875  x1=0; x2=0
```

Odds Ratio: X1 (x1=1; x2=1) / (x1=0; x2=1)

```
. di 3.59575/22.14268
.16239001
```

Odds Ratio: X2 (x1=1; x2=1) / (x1=1; x2=0)

```
. di 3.59575/.210762
17.060713
```

Odds: Intercept (x1=0; x2=0)

```
1.297875
```

The values correspond to what we observe in the regression output.

5 Categorical Predictor

I next determine if the same relationship exists for categorical predictors. I create a data set with *died* as the response and *typ* as a three level categorical predictor. The data displayed in the listings below are to be typed into the data editor, where it is then ready to be evaluated.

```
. clear          /* type values into data editor */
. tab died typ
```

	1=Died;	typ			
	0=Alive	1	2	3	Total
	0	6	1	4	11
	1	2	1	1	4
	Total	8	2	5	15

```
. gsort -died -typ
. l died typ t1 t2 t3
```

	died	typ	t1	t2	t3
1.	1	3	0	0	1
2.	1	2	0	1	0
3.	1	1	1	0	0
4.	1	1	1	0	0
5.	0	3	0	0	1
6.	0	3	0	0	1
7.	0	3	0	0	1
8.	0	3	0	0	1
9.	0	2	0	1	0
10.	0	1	1	0	0
11.	0	1	1	0	0
12.	0	1	1	0	0
13.	0	1	1	0	0
14.	0	1	1	0	0
15.	0	1	1	0	0

R : mod4

```
=====
died <- c(1,1,1,1,0,0,0,0,0,0,0,0,0,0,0)
typ  <- c(3,2,1,1,3,3,3,3,2,1,1,1,1,1,1)
t1   <- c(0,0,1,1,0,0,0,0,0,1,1,1,1,1,1)
t2   <- c(0,1,0,0,0,0,0,0,0,1,0,0,0,0,0)
t3   <- c(1,0,0,0,0,1,1,1,1,0,0,0,0,0,0)
summary(mod4 <- glm(died ~ factor(typ), family=binomial))
exp(coef(mod4))
xb <- mod4$linear.predictors
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data4 <- data.frame(died,typ,t1,t2,t3,mu,xb,o)
data4
=====
```

```
. glm died t2 t3, fam(bin) nolog nohead eform
```

	died	Odds Ratio	OIM Std. Err.	z	P> z	[95% Conf. Interval]	
t2	3	4.898979	0.67	0.501	.1222123	73.64234	
t3	.75	1.038328	-0.21	0.835	.0497309	11.31088	
_cons	.3333333	.2721655	-1.35	0.178	.0672784	1.651513	

```
. predict xb, xb
. predict mu
. gen o=mu/(1-mu)
. l died typ t1 t2 t3 xb mu o
```

	died	typ	t1	t2	t3	xb	mu	o
1.	1	3	0	0	1	-1.386294	.2	.25
2.	1	2	0	1	0	-2.22e-16	.5	1
3.	1	1	1	0	0	-1.098612	.25	.3333333
4.	1	1	1	0	0	-1.098612	.25	.3333333
5.	0	3	0	0	1	-1.386294	.2	.25
6.	0	3	0	0	1	-1.386294	.2	.25
7.	0	3	0	0	1	-1.386294	.2	.25
8.	0	3	0	0	1	-1.386294	.2	.25
9.	0	2	0	1	0	-2.22e-16	.5	1
10.	0	1	1	0	0	-1.098612	.25	.3333333
11.	0	1	1	0	0	-1.098612	.25	.3333333
12.	0	1	1	0	0	-1.098612	.25	.3333333
13.	0	1	1	0	0	-1.098612	.25	.3333333
14.	0	1	1	0	0	-1.098612	.25	.3333333
15.	0	1	1	0	0	-1.098612	.25	.3333333

Odds Ratio: t2

```
(t2==1; t3==0) / (t2==0; t3==0)
. di 1/.3333333
3.0000003
```

Odds Ratio: t3

```
(t2=0; t3=1) / (t2==0; t3==0)
. di .25/.3333333
.75000008
```

Odds: Intercept

```
(t2==0; t3==0)
.3333333
```

The values displayed above correspond to what is displayed in the regression output.

The above table can still be used if the reference level is changed. I shall change the reference level to *t3*, which is the default reference level for SAS.

6 Reference changed to level 3 (t3)

```
. glm died t1 t2 , fam(bin) nolog nohead eform
```

	died	Odds Ratio	OIM Std. Err.	z	P> z	[95% Conf. Interval]	
t1		1.333333	1.845916	0.21	0.835	.0884104	20.10823
t2		4	7.211103	0.77	0.442	.1168247	136.9574
_cons		.25	.2795085	-1.24	0.215	.0279427	2.236723

Using the above table of observation odds values, the model odds ratios may be calculated as:

Odds Ratio: t1

```
(t1==1; t2==0) / (t1==0; t2==0)
. di .3333333/.25
1.3333332
```

Odds Ratio: t2

```
(t1==0; t2==1) / (t1==0; t2==0)
. di 1/.25
4
```

Odds: Intercept

```
(t1==0; t3==0)
.25
```

R : mod5

```
=====
# data from mod4
summary(mod5 <- glm(died ~ t1 + t2, family=binomial))
exp(coef(mod5))
xb <- mod5$linear.predictors
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data5 <- data.frame(died, typ, t1, t2, t3, mu, xb, o)
data5
=====
```

Now we add more complexity to the model by adding another binary predictor.

7 Add binary predictor

```
. glm died x t2 t3, fam(bin) nolog nohead eform
```

	died	Odds Ratio	OIM Std. Err.	z	P> z	[95% Conf. Interval]	
x		.5152636	.726038	-0.47	0.638	.0325562	8.155013
t2		4.660429	8.918793	0.80	0.421	.109511	198.3326
t3		.8657819	1.235649	-0.10	0.920	.0527921	14.19869
_cons		.4164325	.3844513	-0.95	0.343	.0681889	2.543171

R : mod6

```
=====
died <- c(1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0)
typ  <- c(3,2,1,1,3,3,3,3,2,1,1,1,1,1,1,1)
t1   <- c(0,0,1,1,0,0,0,0,0,0,1,1,1,1,1,1)
t2   <- c(0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0)
t3   <- c(1,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0)
x     <- c(1,1,0,0,0,1,0,1,1,0,1,1,1,0,0,0)
summary(mod6 <- glm(died ~ x + t2 + t3, family=binomial))
exp(coef(mod6))
xb <- mod6$linear.predictors
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data6 <- data.frame(died, typ, t1, t2, t3, mu, xb, o)
data6
=====
```

```
. drop xb mu o
. predict xb, xb
. predict mu
. gen o=mu/(1-mu)
. l
```

	died	t1	t2	t3	typ	x	xb	mu	o
1.	1	0	0	1	3	1	-1.68323	.1566683	.185773
2.	1	0	1	0	2	1	0	.5	1
3.	1	1	0	0	1	0	-.8760308	.294001	.4164325
4.	1	1	0	0	1	0	-.8760308	.294001	.4164325
5.	0	0	0	1	3	0	-1.020153	.2649976	.3605398
6.	0	0	0	1	3	1	-1.68323	.1566683	.185773
7.	0	0	0	1	3	0	-1.020153	.2649976	.3605398
8.	0	0	0	1	3	1	-1.68323	.1566683	.185773
9.	0	0	1	0	2	1	0	.5	1
10.	0	1	0	0	1	0	-.8760308	.294001	.4164325
11.	0	1	0	0	1	1	-1.539107	.1766651	.2145725
12.	0	1	0	0	1	1	-1.539107	.1766651	.2145725
13.	0	1	0	0	1	1	-1.539107	.1766651	.2145725
14.	0	1	0	0	1	0	-.8760308	.294001	.4164325
15.	0	1	0	0	1	0	-.8760308	.294001	.4164325

Odds Ratio: X /* hold typ constant for both; then ratio of x=1/x=0 */

(x==1; typ==1) / (x==0; typ==1)

```
. di .2145725/.4164325
.51526358
```

(x==1; typ==3) / (x==0; typ==3)

```
.185773 /.3605398
.5152635
```

```
Odds Ratio: t2      /* hold x=1 constant; then ratio of x=2/x=1 */
(x==1; typ==2)/(x==1; typ==1)
. di 1/.2145725
4.6604295
```

```
Odds Ratio: t3      /* hold x constant; then ratio of x=3/x=1 */
(x==1; typ==3)/(x==1; typ==1)
. di .185773/.2145725
.86578196
```

```
Or
(x==0; typ==3) / (x==0; typ==1) /* hold x constant; then ratio of x=3/x=1 */
. di .3605398/ .4164325
.86578209
```

```
Odds: Intercept
(x==0; typ==1)
.4164325
```

8 Continuous predictor

Finally, I demonstrate how to calculate the model odds ratios from the probabilities of a continuous predictor.

```
. clear          /* then type new values into the data editor */
. glm y x, fam(bin) nolog nohead eform
```

```
-----
              |              OIM
              y | Odds Ratio   Std. Err.      z    P>|z|    [95% Conf. Interval]
-----+-----
              x |   .9648635   .1086147   -0.32   0.751    .7738309    1.203056
      _cons |   2.008914   5.349508    0.26   0.793    .010872    371.2049
-----
```

R : mod7

```
=====
y <- c(1,1,1,1,1,1,1,0,0,0,0,0,0,0)
x <- c(34,25,24,23,20,18,16,29,27,27,25,22,21,21,17)
summary( mod7 <- glm( y ~ x, family=binomial))
exp(coef(mod7))
xb <- mod7$linear.predictors
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data7 <-data.frame(y,x,mu,o)
data7

=====

. predict mu
. gen o=mu/(1-mu)
. gsort -y -x      /* sort descending order, both y and x */
. l
```

	y	x	mu	o
1.	1	34	.3731935	.5953888
2.	1	25	.4510021	.8215005
3.	1	24	.459873	.8514164
4.	1	23	.4687694	.8824216
5.	1	20	.4955553	.9823778
6.	1	18	.5134363	1.055229
7.	1	16	.5312829	1.133483
8.	0	29	.415883	.7119858
9.	0	27	.4333588	.7647853
10.	0	27	.4333588	.7647853
11.	0	25	.4510021	.8215005
12.	0	22	.4776857	.9145559
13.	0	21	.4866162	.9478604
14.	0	21	.4866162	.9478604
15.	0	17	.5223668	1.093657

Odds ratios are obtained by taking the ratio of two contiguous values. Note that the three examples displayed below give the same odds ratio value: for 18-17, 22-21, and 25-24 years old. Notice that the value of y makes no difference when calculating the odds ratio values.

Odds Ratio: X

$x=18/x=17$

```
. di 1.055229 /1.093657
.96486284
```

$x=22/x=21$

```
. di .9145559 /.9478604
.9648635
```

$x=25/x=24$

```
. di .8215005 /.8514164
.96486337
```

Intercept = take the exponentiation of the regression intercept.

9 Continuous with binary predictor

What happens when we add a binary predictor, which is simply a two-level categorical predictor? I let y and x in the previous section remain the same, but rename x to xc. I'll then add a xd, for dichotomous (xb is has already been assigned for the linear predictor). R output is provided in this section.

R : mod8

```
=====
y <- c(1,1,1,1,1,1,1,0,0,0,0,0,0,0)
xc <- c(34,25,24,23,20,18,16,29,27,27,25,22,21,21,17)
xd <- c(1,0,0,0,1,1,0,1,1,0,0,0,0,1,1)
summary( mod8 <- glm( y ~ xc + xd, family=binomial))
exp(coef(mod8))
xb <- mod8$linear.predictors
```



```
mu <- 1/(1+exp(-xb))
o <- mu/(1-mu)
data8 <- data.frame(y, xc, xd, mu, o)
data8
```

Partial output of relevant R results are displayed as:

```
> summary( mod8 <- glm( y ~ xc + xd, family=binomial))
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.77020	2.69784	0.285	0.775
xc	-0.03367	0.11376	-0.296	0.767
xd	-0.26263	1.04667	-0.251	0.802

```
> exp(coef(mod8))
```

	xc	xd
(Intercept)	2.1601985	0.7690241

```
> data8
```

	y	xc	xd	mu	o
1	1	34	1	0.3459094	0.5288402
2	1	25	0	0.4821465	0.9310482
3	1	24	0	0.4905565	0.9629263
4	1	23	0	0.4989718	0.9958958
5	1	20	1	0.4586575	0.8472595
6	1	18	1	0.4754157	0.9062712
7	1	16	0	0.5576297	1.2605495
8	0	29	1	0.3849145	0.6257902
9	0	27	1	0.4009739	0.6693765
10	0	27	0	0.4653616	0.8704232
11	0	25	0	0.4821465	0.9310482
12	0	22	0	0.5073877	1.0299942
13	0	21	0	0.5157995	1.0652600
14	0	21	1	0.4503111	0.8192107
15	0	17	1	0.4838179	0.9373009

Odds Ratio: xc (continuous predictor)

(xd==1;xc==18) / (xd==1; xc==17)

```
> 0.9062712/ 0.9373009
```

```
[1] 0.9668946
```

(xd==0;xc==22) / (xd==0; xc==21)

```
> 1.0299942/1.0652600
```

```
[1] 0.9668947
```

Tactic: contiguous predictors for xc, as with xd the same for both levels.

Odds Ratio: xd (binary predictor)

(xd==1;xc==27) / (xd==0;xc==27)

```
> 0.6693765/0.8704232
```

```
[1] 0.7690242
```

Tactic: with the same value for the continuous predictor, compare the odds of xd=1/xd=0

Again, the intercept must be taken from the output since it is the value of the linear predictor with values of 0 for all predictors.

10 Conclusion

We have covered a number of different types of models with differing structures, incorporating combinations of continuous, categorical and binary predictors. It is obvious that the case-level odds, which are derived directly from the predicted probabilities, relate in specific ways to determine the odds ratios for the model predictors. At initial look it appears that there is no one-to-one relationship between case level probabilities and predictor odds ratios, but there is.

For additional discussion on this subject, refer to Hilbe (2009).

Note: An amended version of this monograph will be included in the second edition of *Logistic Regression Models*, due to be published in 2018. A brief version of this monograph will be included in Hilbe (2016), which is due to be completed in October, 2014 (published 9 July, 2015 – Chapman & Hall/CRC gives next year copyright to all books published on 1 July or later in the year). Please cite this monograph if used in any manner in a publication.

Reference

Hilbe, J.M (2009), *Logistic Regression Models*, Boca Raton, FL: Chapman & Hall/CRC.

Hilbe, J.M (2016), *Practical Guide to Logistic Regression Modeling*, Boca Raton, FL: Chapman & Hall/CRC.