

Confounding and Collinearity in Multivariate Logistic Regression

We have already seen confounding and collinearity in the context of linear regression, and all definitions and issues remain essentially unchanged in logistic regression.

Recall the definition of confounding:

Confounding: A third variable (not the independent or dependent variable of interest) that distorts the observed relationship between the exposure and outcome. Confounding complicates analyses owing to the presence of a third factor that is associated with both the putative risk factor and the outcome.

Criteria for a confounding factor:

1. A confounder must be a risk factor (or protective factor) for the outcome of interest.
2. A confounder must be associated with the main independent variable of interest.
3. A confounder must not be an intermediate step in the causal pathway between the exposure and outcome.

All of the above remains true when investigating confounding in logistic regression models.

In linear regression, one way we identified confounders was to compare results from two regression models, with and without a certain suspected confounder, and see how much the coefficient from the main variable of interest changes.

The same principle can be used to identify confounders in logistic regression. An exception possibly occurs when the range of probabilities is very wide (implying an s-shaped curve rather than a close to linear portion), in which case more care can be required (beyond scope of this course).

As in linear regression, collinearity is an extreme form of confounding, where variables become “non-identifiable”.

Let's look at some examples.

Simple example of collinearity in logistic regression

Suppose we are looking at a dichotomous outcome, say cured = 1 or not cured = 0, from a certain clinical trial of Drug A versus Drug B. Suppose by extreme bad

luck, all subjects randomized to Drug A were female, and all subjects randomized to drug B were male. Suppose further that both drugs are equally effective in males and females, and that Drug A has a cure rate of 30%, while Drug B has a cure rate of 50%.

We can simulate a data set that follows this scenario in R as follows:

```
# Suppose sample size of trial is 600, with 300 on each medication
> drug <- as.factor(c(rep("A", 300), rep("B", 300)))

# Ensure that we have collinearity of sex and the medication
> sex <- as.factor(c(rep("F", 300), rep("M", 300)))

# Generate cure rates of 30% and then 50%
> cure <- c(rbinom(300, 1, 0.3), rbinom(300, 1, 0.5))

# Place variables into a data frame, check descriptive statistics
> cure.dat <- data.frame(cure=cure, sex=sex, drug=drug)

> summary(cure.dat)
      cure      sex      drug
Min.   :0.00  F:300   A:300
1st Qu.:0.00  M:300   B:300
Median :0.00
Mean   :0.42
3rd Qu.:1.00
Max.   :1.00

# Run a logistic regression model for cure with both variables in the model
> output <- glm(cure ~ drug + sex, family = binomial)

# Use usual summary when there is collinearity
> summary(output)

Call:
glm(formula = cure ~ drug + sex, family = binomial)

Deviance Residuals:
      Min       1Q   Median       3Q      Max
```

```
-1.2637 -0.8276 -0.8276 1.0935 1.5735
```

```
Coefficients: (1 not defined because of singularities)
```

```
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.8954      0.1272  -7.037 1.96e-12 ***
drugB        1.0961      0.1722   6.365 1.96e-10 ***
sexM         NA         NA      NA      NA
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)
```

```
Null deviance: 816.35 on 599 degrees of freedom
Residual deviance: 774.17 on 598 degrees of freedom
AIC: 778.17
```

```
Number of Fisher Scoring iterations: 4
```

Notice that R has automatically eliminated the sex variable, and we see that the OR for drug B compared to drug A is $\exp(1.0961) = 2.99$, which is close to correct, because $OR = (.5/(1-.5))/(.3/(1-.3)) = 2.33$, and the CI is $(\exp(1.0961 - 1.96*0.1722), \exp(1.0961 + 1.96*0.1722)) = (2.13, 4.19)$.

In fact, this exactly matches the observed OR, from the table of data we simulated:

```
> table(cure.dat$cure, cure.dat$drug)
```

```
      A  B
0 213 135
1  87 165
> 213*165/(87*135)
[1] 2.992337
```

```
# Why was sex eliminated, rather than drug?
# Depends on order entered into the glm statement
```

```
# Check the other order:
```

```
> output <- glm(cure ~ sex + drug, family = binomial)
> summary(output)
```

```
Coefficients: (1 not defined because of singularities)
```

```

                Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.8954      0.1272  -7.037 1.96e-12 ***
sexM         1.0961      0.1722   6.365 1.96e-10 ***
drugB                NA          NA     NA     NA
---

```

```
# Exactly the same numerical result, but for sex rather than drug.
```

Second example of collinearity in logistic regression

A more subtle example can occur when two variables act to be collinear with a third variable.

Collinearity can also occur in continuous variables, so let's see an example there:

```

# Create any first independent variable (round to one decimal place)
> x1 <- round(rnorm(400, mean=0, sd=1), 1)

# Create any second independent variable (round to one decimal place)
> x2 <- round(rnorm(400, mean = 4, sd=2), 1)

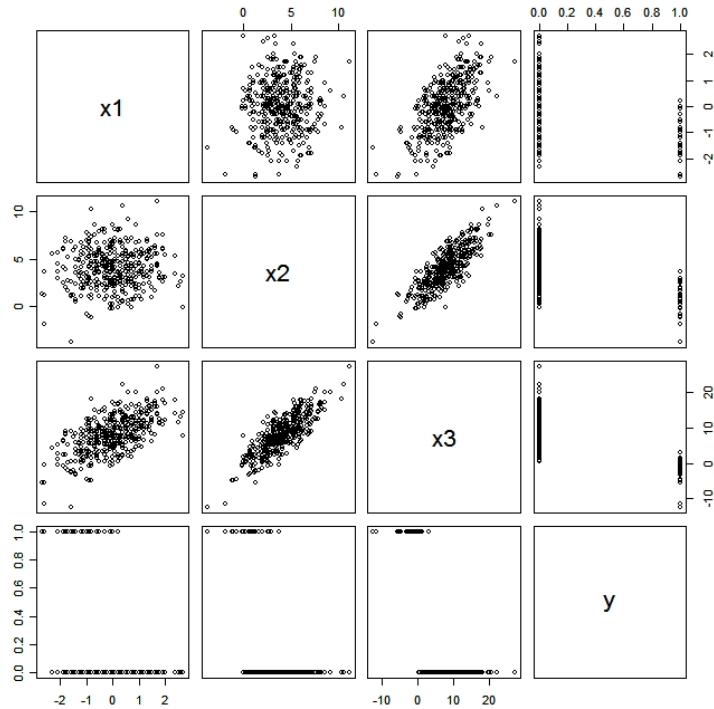
# Now create a third independent variable that is a direct function
# of the first two variables
> x3 <- 3*x1 + 2 *x2

# Create a binary outcome variable that depends on all three variables
# Note that the probability of the binomial is an inv.logit function
> y <- rbinom(400, 1, exp(x1 + 2*x2 -3 * x3)/(1+ exp(x1 + 2*x2 -3 * x3)))

# Put all variables into a data frame
> collinear.dat <- data.frame(x1=x1, x2=x2, x3=x3, y=y)

# If looked at pairwise, the perfect collinearity is not obvious
> pairs(collinear.dat)

```



One can see high correlations, but cannot tell that there is perfect collinearity. But let's see what happens if we run an analysis:

```
> output <- glm(y ~ x1 + x2 + x3, data = collinear.dat, family = binomial)
Warning message:
fitted probabilities numerically 0 or 1 occurred in:
glm.fit(x = X, y = Y, weights = weights, start = start, etastart = etastart,

# Note the warning message below...R has detected collinearity

> summary(output)
```

Call:

```
glm(formula = y ~ x1 + x2 + x3, family = binomial, data = collinear.dat)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.811e+00	-3.858e-03	-7.593e-05	-2.107e-08	3.108e+00

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.7036	0.9001	1.893	0.0584 .

```

x1          -7.8162      1.8864  -4.144  3.42e-05 ***
x2          -4.6763      1.1203  -4.174  2.99e-05 ***
x3              NA          NA      NA      NA

```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)
```

```

Null deviance: 242.030 on 399 degrees of freedom
Residual deviance: 30.292 on 397 degrees of freedom
AIC: 36.292

```

```
Number of Fisher Scoring iterations: 11
```

```
# x3 has been eliminated, other variables reasonably estimated.
```

```

# If we want to get the CIs automatically, rerun model without x3
# and use logistic.regression.or.ci function, or more simply,
# just use the built-in confint function in R

```

```
> output <- glm(y ~ x1 + x2, data = collinear.dat, family = binomial)
```

```
Warning message:
```

```
fitted probabilities numerically 0 or 1 occurred in:
```

```
glm.fit(x = X, y = Y, weights = weights, start = start, etastart = etastart,
```

```
# Above warning message refers to strong results, not collinearity
```

```
> logistic.regression.or.ci(output)
```

```
$regression.table
```

```
Call:
```

```
glm(formula = y ~ x1 + x2, family = binomial, data = collinear.dat)
```

```
Deviance Residuals:
```

```

      Min       1Q   Median       3Q      Max
-1.811e+00 -3.858e-03 -7.593e-05 -2.107e-08  3.108e+00

```

```
Coefficients:
```

```

      Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.7036     0.9001  1.893  0.0584 .
x1          -7.8162     1.8864 -4.144  3.42e-05 ***
x2          -4.6763     1.1203 -4.174  2.99e-05 ***

```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 242.030 on 399 degrees of freedom
 Residual deviance: 30.292 on 397 degrees of freedom
 AIC: 36.292

Number of Fisher Scoring iterations: 11

\$intercept.ci

[1] -0.06050707 3.46779874

\$slopes.ci

	[,1]	[,2]
[1,]	-11.513417	-4.119032
[2,]	-6.872168	-2.480528

\$OR

	x1	x2
	0.0004031407	0.0093129598

\$OR.ci

	[,1]	[,2]
[1,]	9.995083e-06	0.01626024
[2,]	1.036228e-03	0.08369899

Two very strong effects, not surprising given data set
 # construction

Check a few fitted values

> output\$fitted[1:6]

	1	2	3	4	5	6
	6.932227e-01	4.244699e-03	1.219659e-07	2.220446e-16	1.269672e-10	2.220446e-16

Note how close some values are to zero, others much higher.

In real practice, most collinearity problems happen when several categorical variables line up to “perfectly predict” another variable.

Example of confounding in logistic regression

Let's consider a similar example again, but with x_3 not quite perfectly derived from the first two variables:

```
# Create any first independent variable (round to one decimal place)

x1 <- round(rnorm(400, mean=0, sd=1), 1)

# Create any second independent variable (round to one decimal place)

x2 <- round(rnorm(400, mean = 4, sd=2), 1)

# Now create a third independent variable that is a
# related by not a direct function of the first two variables
# because of the error term added

x3 <- round(3*x1 + 2 *x2 + rnorm(400, mean = 0, sd=5), 1)

# Create a binary outcome variable that depends on all three variables
# Note that the probability of the binomial is an inv.logit function

# We will use smaller effects this time as well, more realistic.
# Note that a coefficient of 0.2 has an OR of
# exp(0.2) = 1.22 / one unit change

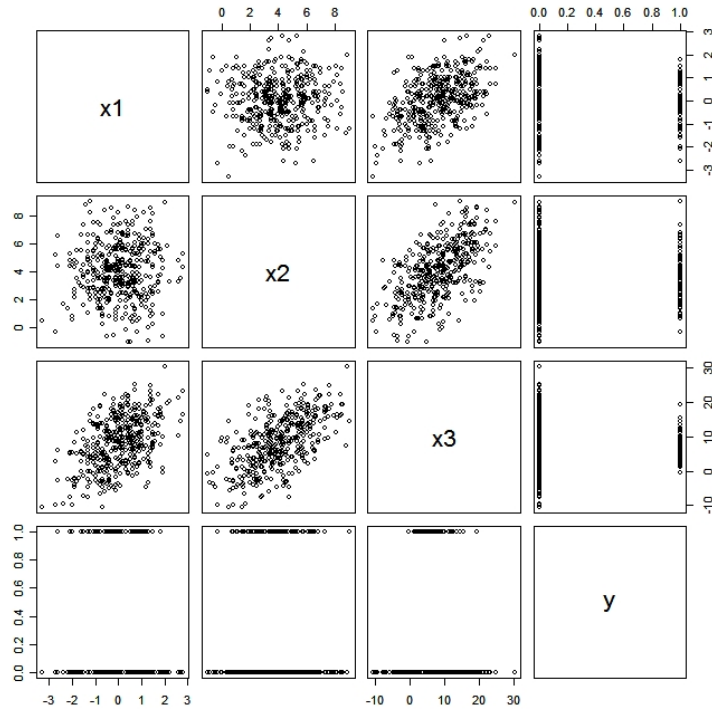
y <- rbinom(400, 1, exp(.2*x1 + .3*x2 -.3 * x3)/(1+ exp(.2*x1 + 2*x2 -3 * x3)))

# Put all variables into a data frame

confounding.dat <- data.frame(x1=x1, x2=x2, x3=x3, y=y)

# If looked at pairwise, the very strong confounding is not obvious
# because it arises from three variables working together

pairs(confounding.dat)
```

Note the smaller effects as shown in the graphics.

Now to analyze the data, comparing univariate to multivariate model outputs.

```
# First univariate logistic regressions for each of the three variables
```

```
> output <- glm(y ~ x1, data = confounding.dat, family = binomial)
> logistic.regression.or.ci(output)
$regression.table
```

Call:

```
glm(formula = y ~ x1, family = binomial, data = confounding.dat)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.3020	0.1260	-10.337	< 2e-16	***
x1	-0.3484	0.1203	-2.897	0.00377	**

```
$OR
```

```
      x1
0.7058417
```

```
$OR.ci
```

```
[1] 0.5576294 0.8934473
```

```
> output <- glm(y ~ x2, data = confounding.dat, family = binomial)
> logistic.regression.or.ci(output)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.97316	0.26430	-3.682	0.000231	***
x2	-0.07689	0.06109	-1.259	0.208142	

\$OR

```
      x2
0.9259914
```

\$OR.ci

```
[1] 0.8215029 1.0437700
```

```
> output <- glm(y ~ x3, data = confounding.dat, family = binomial)
> logistic.regression.or.ci(output)
```

\$regression.table

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.85231	0.16717	-5.098	3.43e-07	***
x3	-0.05655	0.01683	-3.359	0.000781	***

\$OR

```
      x3
0.9450173
```

\$OR.ci

```
[1] 0.9143465 0.9767169
```

Now let's run a logistic regression with all three variables included:

```
> output <- glm(y ~ x1 + x2 + x3, data = confounding.dat, family = binomial)
> logistic.regression.or.ci(output)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.06718	0.27464	-3.886	0.000102	***
x1	-0.16353	0.14926	-1.096	0.273257	
x2	0.04321	0.08537	0.506	0.612738	

```
x3          -0.05209    0.02629  -1.981  0.047561 *
---
```

```
$OR
      x1          x2          x3
0.8491411 1.0441583 0.9492457
```

```
$OR.ci
      [,1]      [,2]
[1,] 0.6337649 1.137710
[2,] 0.8832837 1.234333
[3,] 0.9015722 0.999440
```

To investigate the above results for confounding, let's form a comparative table:

Variable	Multivariate		Univariate	
	OR	CI	OR	CI
x1	0.85	(0.63, 1.11)	0.71	(0.56, 0.89)
x2	1.04	(0.88, 1.23)	0.93	(0.82, 1.04)
x3	0.95	(0.90, 1.00)	0.95	(0.91, 0.98)

Note how drastically different the results are, especially for x1. All CIs cross 1 in the multivariate model, but only x2 crosses 1 in the univariate models, the CI widths are smaller in the univariate models. OR's also change by large amounts.

As x2 may not be contributing much, we can also run a model with just x1 and x3.

```
> output <- glm(y ~ x1 + x3, data = confounding.dat, family = binomial)
> logistic.regression.or.ci(output)
```

Coefficients:

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.96730    0.18904  -5.117 3.11e-07 ***
x1           -0.19265    0.13802  -1.396  0.1628
x3           -0.04310    0.01933  -2.230  0.0258 *
```

```
---
$OR
      x1          x3
0.8247687 0.9578197
```

```
$OR.ci
      [,1]      [,2]
[1,] 0.6292859 1.0809767
[2,] 0.9222138 0.9948003
```

Not much change from the model with all three variables.

We will soon see how we can run all interesting models with a single command using the `bic.glm` model selection function. This will allow us to investigate confounding and model selection at the same time.

Real example of confounding in logistic regression

Low birth weight is of concern, because infant mortality rates and birth defect rates are very high for low birth weight babies. A woman's behavior during pregnancy (including diet, smoking habits, and receiving prenatal care) can greatly alter the chances of carrying the baby to term and, consequently, of delivering a baby of normal birth weight.

The following data are collected:

Variable	Coding
Low Birth Weight (0 = Birth Weight \geq 2500g, 1 = Birth Weight $<$ 2500g)	low
Age of the Mother in Years	age
Weight in Pounds at the Last Menstrual Period	lwt
Race (1 = White, 2 = Black, 3 = Other)	race
Smoking Status During Pregnancy (1 = Yes, 0 = No)	smoke
History of Premature Labor (0 = None 1 = One, etc.)	ptl
History of Hypertension (1 = Yes, 0 = No)	ht
Presence of Uterine Irritability (1 = Yes, 0 = No)	ui
Number of Physician Visits During the First Trimester (0 = None, 1 = One, 2 = Two, etc.)	ftv
Birth Weight in Grams	bwt

We might suspect some confounding. For example, smoking may be related to weight and hypertension, and so on.

We will follow all of our usual steps in analyzing these data. Recall that the steps are:

1. Look at various descriptive statistics to get a feel for the data. For logistic regression, this usually includes looking at descriptive statistics within “outcome = yes = 1” versus “outcome = no = 0” groups.
2. The above “by outcome group” descriptive statistics are often sufficient for discrete covariates, but you may want to prepare some graphics for continuous variables.
3. For all continuous variables being considered, calculate a correlation matrix of each variable against each other variable. This allows one to begin to investigate possible confounding and collinearity.
4. Similarly, for each categorical/continuous independent variable pair, look at the values for the continuous variable in each category of the other variable.
5. Finally, create tables for all categorical/categorical independent variable pairs.

6. Perform a simple logistic regression for each independent variable. This begins to investigate confounding (we will see in more detail next class), as well as providing an initial “unadjusted” view of the importance of each variable, by itself.
7. Think about any “interaction terms” that you may want to try in the model.
8. Perform some sort of model selection technique, or, often much better, think about avoiding any strict model selection by finding a set of models that seem to have something to contribute to overall conclusions.
9. Based on all work done, draw some inferences and conclusions. Carefully interpret each estimated parameter, perform “model criticism”, possibly repeating some of the above steps (for example, run further models), as needed.
10. Other inferences, such as predictions for future observations, and so on.

```
# Read in the data set, save as a data frame

> lbw.dat <- read.table(file="g:\\lbw.txt", header=T)

# Convert factor variables

> lbw.dat$smoke <- as.factor(lbw.dat$smoke)
> lbw.dat$race <- as.factor(lbw.dat$race)
> lbw.dat$ptl <- as.factor(lbw.dat$ptl)
> lbw.dat$ht <- as.factor(lbw.dat$ht)
> lbw.dat$ui <- as.factor(lbw.dat$ui)

# Summarize

> summary(lbw.dat)
```

low	age	lwt	race	smoke	ptl
Min. :0.0000	Min. :14.00	Min. : 80.0	1:96	0:115	0:159
1st Qu.:0.0000	1st Qu.:19.00	1st Qu.:110.0	2:26	1: 74	1: 24
Median :0.0000	Median :23.00	Median :121.0	3:67		2: 5
Mean :0.3122	Mean :23.24	Mean :129.8			3: 1
3rd Qu.:1.0000	3rd Qu.:26.00	3rd Qu.:140.0			
Max. :1.0000	Max. :45.00	Max. :250.0			

ht	ui	ftv	bwt
0:177	0:161	Min. :0.0000	Min. : 709
1: 12	1: 28	1st Qu.:0.0000	1st Qu.:2414

```
Median :0.0000   Median :2977
Mean   :0.7937   Mean    :2945
3rd Qu.:1.0000   3rd Qu.:3475
Max.   :6.0000   Max.    :4990
```

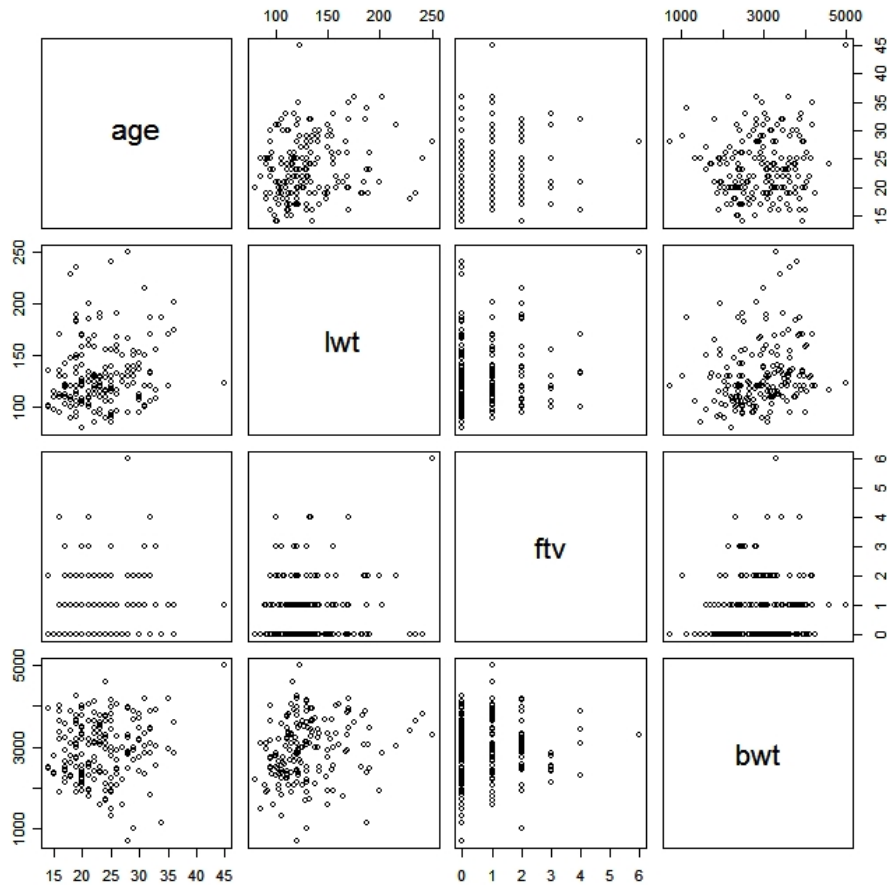
```
# Examining the categorical variables, ptl has few cases at 2 or 3, so
# combine these categories with category 1.
```

```
> for (i in 1:length(lbw.dat$ptl)) { if (lbw.dat$ptl[i] == 2 | lbw.dat$ptl[i] == 3)
    lbw.dat$ptl[i] <- 1}
```

```
> summary(lbw.dat$ptl)
 0  1  2  3
159 30  0  0
```

```
# Look at some correlations for continuous variables
```

```
> pairs(list(age=lbw.dat$age, lwt=lbw.dat$lwt, ftv=lbw.dat$ftv, bwt=lbw.dat$bwt))
```



Some correlations to keep in mind, e.g., age and lwt, although nothing too extreme.

Although included up to this point, the outcome variable low is in fact just a dichotomized version of the bwt variable, so the latter is omitted for the rest of these analyses.

Should also check some tables and values of continuous variables against categorical variables, I leave this as an exercise. [And, since we will soon see another way to check for confounding, this is not always needed.]

```
# Run univariate regressions
```

```
> output <- glm(low ~ age, data = lbw.dat, family=binomial)
> logistic.regression.or.ci(output)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.38458	0.73212	0.525	0.599


```
age          -0.05115    0.03151   -1.623    0.105
```

```
$OR
```

```
  age
0.9501333
```

```
$OR.ci
```

```
[1] 0.8932232 1.0106694
```

```
> output <- glm(low ~ lwt, data = lbw.dat, family=binomial)
> logistic.regression.or.ci(output)
```

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.99831	0.78529	1.271	0.2036
lwt	-0.01406	0.00617	-2.279	0.0227 *

```
$OR
```

```
  lwt
0.98604
```

```
$OR.ci
```

```
[1] 0.9741885 0.9980358
```

```
> output <- glm(low ~ race, data = lbw.dat, family=binomial)
> logistic.regression.or.ci(output)
```

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.1550	0.2391	-4.830	1.36e-06 ***
race2	0.8448	0.4634	1.823	0.0683 .
race3	0.6362	0.3478	1.829	0.0674 .

```
$OR
```

```
  race2  race3
2.327536 1.889234
```

```
$OR.ci
```

```
  [,1]  [,2]
[1,] 0.9385074 5.772384
[2,] 0.9554579 3.735596
```

```
> output <- glm(low ~ smoke, data = lbw.dat, family=binomial)
> logistic.regression.or.ci(output)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.0871	0.2147	-5.062	4.14e-07	***
smoke1	0.7041	0.3196	2.203	0.0276	*

\$OR

smoke1
2.021944

\$OR.ci

[1] 1.080660 3.783111

```
> output <- glm(low ~ pt1, data = lbw.dat, family=binomial)
> logistic.regression.or.ci(output)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.0571	0.1813	-5.831	5.5e-09	***
pt11	1.4626	0.4144	3.529	0.000417	***

\$OR

pt11
4.317073

\$OR.ci

[1] 1.916128 9.726449

```
> output <- glm(low ~ ht, data = lbw.dat, family=binomial)
> logistic.regression.or.ci(output)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.8771	0.1650	-5.315	1.07e-07	***
ht1	1.2135	0.6083	1.995	0.0461	*

\$OR

ht1
3.365385

\$OR.ci

[1] 1.021427 11.088221

```
> output <- glm(low ~ ui, data = lbw.dat, family=binomial)
```

```
> logistic.regression.or.ci(output)
```

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.9469	0.1756	-5.392	6.97e-08	***
ui1	0.9469	0.4168	2.272	0.0231	*

```
$OR
```

```
  ui1
2.577778
```

```
$OR.ci
```

```
[1] 1.138905 5.834499
```

```
> output <- glm(low ~ ftv, data = lbw.dat, family=binomial)
```

```
> logistic.regression.or.ci(output)
```

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.6868	0.1948	-3.525	0.000423	***
ftv	-0.1351	0.1567	-0.862	0.388527	

```
$OR
```

```
  ftv
0.8736112
```

```
$OR.ci
```

```
[1] 0.6425933 1.1876819
```

```
# Run a multivariate model
```

```
> output <- glm(low ~ age + lwt + race + smoke + ptl + ht + ui + ftv,
  data = lbw.dat, family=binomial)
```

```
> logistic.regression.or.ci(output)
```

```
$regression.table
```

```
Call:
```

```
glm(formula = low ~ age + lwt + race + smoke + ptl + ht + ui +
  ftv, family = binomial, data = lbw.dat)
```

```
Deviance Residuals:
```

Min	1Q	Median	3Q	Max
-1.6305	-0.7894	-0.5094	0.9119	2.2257

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.644476	1.223889	0.527	0.59849
age	-0.039548	0.038305	-1.032	0.30186
lwt	-0.015078	0.007034	-2.143	0.03207 *
race2	1.218791	0.533168	2.286	0.02226 *
race3	0.819439	0.450466	1.819	0.06890 .
smoke1	0.859459	0.409836	2.097	0.03599 *
pt11	1.218512	0.463015	2.632	0.00850 **
ht1	1.860429	0.708161	2.627	0.00861 **
ui1	0.719299	0.463419	1.552	0.12062
ftv	0.050900	0.175456	0.290	0.77174

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 234.67 on 188 degrees of freedom
 Residual deviance: 196.75 on 179 degrees of freedom
 AIC: 216.75

Number of Fisher Scoring iterations: 4

\$intercept.ci

[1] -1.754303 3.043254

\$slopes.ci

	[,1]	[,2]
[1,]	-0.11462392	0.035527619
[2,]	-0.02886411	-0.001290897
[3,]	0.17380164	2.263780165
[4,]	-0.06345768	1.702336654
[5,]	0.05619565	1.662721742
[6,]	0.31101897	2.126005387
[7,]	0.47245923	3.248398230
[8,]	-0.18898481	1.627582912
[9,]	-0.29298669	0.394786830

\$OR

age	lwt	race2	race3	smoke1	pt11
0.9612237	0.9850356	3.3830948	2.2692275	2.3618819	3.3821520

ht1	ui1	ftv
6.4264914	2.0529937	1.0522177

```

$OR.ci
      [,1]      [,2]
[1,] 0.8917014 1.036166
[2,] 0.9715485 0.998710
[3,] 1.1898195 9.619383
[4,] 0.9385138 5.486753
[5,] 1.0578046 5.273645
[6,] 1.3648151 8.381320
[7,] 1.6039338 25.749063
[8,] 0.8277991 5.091553
[9,] 0.7460321 1.484068

```

Compare univariate to multivariate results:

Variable	Multivariate		Univariate	
	OR	CI	OR	CI
age	0.96	(0.89, 1.04)	0.95	(0.89, 1.01)
lwt	0.99	(0.97, 1.00)	0.99	(0.97, 1.00)
race2	3.39	(1.19, 9.62)	2.02	(0.93, 5.77)
race3	2.27	(0.94, 5.49)	1.89	(0.96, 3.74)
smoke1	2.36	(1.06, 5.27)	2.02	(1.08, 3.78)
ptl1	3.38	(1.36, 8.38)	4.32	(1.92, 9.73)
ht1	6.42	(1.60, 25.75)	3.37	(1.02, 11.09)
ui1	2.05	(0.83, 5.09)	2.58	(1.14, 5.83)
ftv	1.05	(0.75, 1.48)	0.87	(0.64, 1.19)

Maybe some confounding with race, ht1, ftv, etc. We could investigate this further here, but will rather revisit this example after covering model selection and the `bic.glm` program, which makes such investigations much easier.