Logistic Regression

- Used when the response variable is binary

- We model the log of the odds for $Y_i = 1$

$$
\ln \left( \frac{P(Y_i = 1)}{1 - P(Y_i = 1)} \right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik}
$$

- Which can be converted to

$$
P(Y_i = 1) = \frac{\exp(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik})}{1 + \exp(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_k X_{ik})}
$$

- Unlike OLS regression, logistic regression does not assume...

  - linearity between the independent variables and the dependent
  - normally distributed errors
  - homoscedasticity

- It does assume...

  - we have independent observations
  - that the independent variables be linearly related to the logit of the dependent variable (somewhat difficult to check)

- Maximum likelihood estimation (MLE) is used to calculate the regression coefficient estimates

  - Ordinary Least Squares (OLS) minimizes the sum of the squared residuals
  - MLE finds the parameter estimates that maximize the log-likelihood function

* The Global Null model (simplest possible model) has only an intercept and is the model:

$$
\text{logit}(\pi_i) = c \quad \text{for all } i
$$

and some constant $c$

[i.e. covariates don’t affect $\pi_i = P(Y_i = 1)$]

* Full model vs. Global Null model has the flavor of an overall F-test from multiple regression (i.e. are any of the variables in the model useful).
**Example:** Incidence of bird on islands

Dichotomous response called *incidence*.

\[
\text{incidence} = \begin{cases} 
1 & \text{if island occupied by bird} \\
0 & \text{if bird did not breed there}
\end{cases}
\]

Two continuous predictor variables:
- *area* - area of island in \(km^2\)
- *isolation* - distance from mainland in km

\[
\begin{array}{ccc}
\text{incidence} & \text{area} & \text{isolation} \\
1 & 7.928 & 3.317 \\
2 & 1.925 & 7.554 \\
3 & 2.045 & 5.883 \\
4 & 4.781 & 5.932 \\
5 & 1.536 & 5.308 \\
6 & 7.369 & 4.934 \\
\end{array}
\]

We expect *incidence* to be lower for high *isolation*, and incidence to be higher for high *area*.

Fit the full additive model (2 covariates):

\[
\text{area} + \text{isolation}
\]

\[
\begin{array}{c}
\text{null deviance: 68.029 on 49 degrees of freedom} \\
\text{residual deviance: 28.402 on 47 degrees of freedom} \\
\text{AIC: 34.402}
\end{array}
\]

Deviance (or residual deviance)
- This is used to assess the model fit.
- In logistic regression, the *deviance* has the flavor of RSS in ordinary regression.
- The smaller the deviance, the better the fit.

Null deviance - like RSS in ordinary regression when only an overall mean is fit (see R output: n=50, and df are 49).

Residual deviance - like RSS from the full model fit (see R output: n=50, and df are 47).
A comparison of the **Null deviance** and **Residual deviance** is used to test the global null hypothesis. In this case...

\[ H_0 : \beta_1 = \beta_2 = 0 \]
\[ H_A : \text{At least one } \beta_j \text{ not equal to 0} \]

A likelihood ratio test is used for this nested test which follows a central \( \chi^2 \) distribution under \( H_0 \) being true.

\[ \chi^2_q = -2[\log \text{likelihood}_{red} - \log \text{likelihood}_{full}] \]
\[ = (-2LL_{red}) - (-2LL_{full}) \]
\[ = (2LL_{saturated} - 2LL_{red}) - (2LL_{saturated} - 2LL_{full}) \]
\[ = \text{Reduced model deviance} - \text{Full model deviance} \]
\[ = \text{Null deviance} - \text{Residual deviance} \]

The **deviance** is saved in the model fit output, and it can be requested...

\[ \chi^2_q = \text{Reduced model deviance} - \text{Full model deviance} \]

> chi.sq=glm.out.null$deviance-glm.out.full$deviance
> pchisq(chi.sq,2,lower.tail=FALSE)

[1] 2.483693e-09

Same \( p \)-value as in previous page output.

- Global null hypothesis test for *incidence* data:

From the *summary* output...

Null deviance: 68.029 on 49 degrees of freedom
Residual deviance: 28.402 on 47 degrees of freedom

\( \chi^2_2 \) test:

> chi.sq=68.029 - 28.402
> pchisq(chi.sq,2,lower.tail=FALSE)

[1] 2.483741e-09

This can also be done using the full vs. reduced likelihood ratio test (use test = "Chisq"):  

> glm.out.null=glm(incidence ~ 1,family=binomial(logit))
> anova(glm.out.null,glm.out.full,test="Chisq")

Analysis of Deviance Table

Model 1: incidence ~ 1
Model 2: incidence ~ area + isolation

Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1 49 68.029
2 47 28.402 2 39.627 2.484e-09

⇒ Reject \( H_0 \).

- Individual tests for coefficients

After rejecting the **Global Null hypothesis**, we can consider individual Z-tests for the predictors.

> summary(glm.out.full)

```
Coefficients: Estimate Std. Error z value Pr(>|z|)
(Intercept) 6.6417 2.9218 2.273 0.02302 *
area 0.5807 0.2478 2.344 0.01909 *
isolation -1.3719 0.4769 -2.877 0.00401 **
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

*isolation* is a significant predictor, given we’ve already accounted for *area*.

*area* is a significant predictor, given we’ve already accounted for *isolation*.  

---

11
Another way to view the data:

```r
plot(area, isolation, pch = (incidence + 1), col = (incidence + 1))
title("Each point represents an island")
legend(6, 9, c("incidence 0", "incidence 1"),
col = c(1, 2), pch = c(1, 2))
```

![Graph showing the relationship between area and isolation with incidence level]

- **Interpretation of the parameters:**
  
  \[ \hat{\beta}_{\text{area}} = 0.5807 \]
  
  or \[ e^{\hat{\beta}_{\text{area}}} = 1.7873 \]

  Holding \textit{isolation} constant...

  A 1 km\(^2\) increase in \textit{area} is associated with an increase in the odds of seeing a bird by a factor of 1.7873.

  \[ e^{\hat{\beta}_{\text{area}}} \] represents the multiplicative effect (applied to the odds) of a 1-unit change in \textit{area}.

  Increasing the area of an island by 1 km\(^2\), increases the odds of seeing a bird by a multiplicative factor of 1.7873.

  It increases the odds by 78.73%.

Another way to express it,

\[
\text{Odds}_{[x_1 + 1 \text{ km}^2]} = 1.7872 \times \text{Odds}_{[x_1 \text{ km}^2]}
\]

\[
\hat{\beta}_{\text{isolation}} = -1.3719
\]

or \[ e^{\hat{\beta}_{\text{isolation}}} = 0.2536 \]

Holding \textit{area} constant...

A 1 km increase in \textit{isolation} is associated with a decrease in the odds of seeing a bird by a factor of 0.2536.

\[
\text{Odds}_{[x_2 + 1 \text{ km}]} = 0.2536 \times \text{Odds}_{[x_2 \text{ km}]}
\]

- **Diagnostics:** Outliers, Influential data

The \textit{car} library diagnostics can also be used on generalized linear models...

\textit{rstudent, hatvalues, cookd, vif, outlier.test, and av.plots.}

```r
> plot(cooks.distance(glm.out.full), pch = 16)
> identify(1:n, cooks.distance(glm.out.full))
```

![Graph showing diagnostic plots]
• **Diagnostics**: Goodness of fit

Since the responses are all 0’s and 1’s, it is more difficult to check how well the model fits our data (compared to ordinary regression).

If you have data points fairly evenly spread across your x-values, you could try to check the fit using the Hosmer-Lemeshow Goodness of Fit Test.

We will return to our original example data on lead levels in children’s blood relative to soil lead levels to show how this test works.

The fitted value is a probability (or \( \hat{p} \)).

The logistic regression provides a \( \hat{p} \) for every x-value.

To check the fit, we will partition the observations into 10 groups based on the x-values.

```r
> break.points = quantile(soil, seq(0, 1, 0.1))
> group.soil = cut(soil, breaks = break.points)
> table(group.soil)

<table>
<thead>
<tr>
<th>group.soil</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(40,89.4]</td>
<td>(89.4,151]</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>(151,239]</td>
<td>(239,361]</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>(361,527]</td>
<td>(527,750]</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>(750,891]</td>
<td>(891,1330]</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>(1330,1780]</td>
<td>(1780,5260]</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>
```
For each group \( g \), we will estimate a \( \hat{p}_g \).

This \( \hat{p}_g \) does not consider the other fitted \( \hat{p} \) values (unlike the logistic regression fitted values which all fall along a smooth curve).

```r
> group.est.p=tapply(highbld,group.soil,mean)
> group.est.p=as.vector(group.est.p)
> round(group.est.p,4)
[1] 0.2727 0.1429 0.2143 0.0714 0.5714
0.7692 0.6429 0.7143 0.9286 1.0000
```

Now we will compare the ‘freely fit’ estimated probabilities, with the logistic regression (restricted) fitted probabilities.

The vertical lines represent the grouping structure of the observations.

If the dots fall close to the fitted logistic curve, it’s a reasonably good fit.

The short red lines represent +/- 2 standard errors of each \( \hat{p} \).

The **Hosmer-Lemeshow Test** takes these values and tests the goodness of fit using a \( \chi^2 \) test statistic.

\[ H_0 : \text{Fit is sufficient} \]
\[ H_A : \text{Fit is not sufficient (the curve doesn’t explain the data very well)} \]

```r
> hosmerlem(highbld, glm.out$fitted, g = 10)
X^2 Df P(>Chi)
12.2422056 8 0.1407200
```

The \( p \)-value=0.1407, so we do not reject null.

The logistic regression model describes the data reasonably well.

The **Hosmer-Lemeshow Test** function isn’t in an \textbf{R} library, but it is shown below. To use it, just copy and paste it into the \textbf{R} interpreter window.

```r
hosmerlem=function (y, yhat, g = 10) {
## y is the original response.
## yhat are the fitted values from the model.
## g is the number of groups requested.
## cuts = cut(yhat,
## breaks = quantile(yhat, probs = seq(0,1,1/g)),
## include.lowest = T)
obs = xtabs(cbind(1 - y, y) ~ cutyhat)
expect = xtabs(cbind(1 - yhat, yhat) ~ cutyhat)
chisq = sum((obs - expect)^2/expect)
P = 1 - pchisq(chisq, g - 2)
c("X^2" = chisq, Df = g - 2, "P(>Chi)" = P)
}
## A function to do the Hosmer-Lemeshow test in R.
## R Function is due to Peter D. M. Macdonald,
## McMaster University.

One criticism of this test is that it depends on the chosen breakpoints for the group. See...

• Issues to consider
  – Influential observations
  – multicollinearity
    * When you reject $H_0$, but none of the individual regression coefficients are significant
    * high correlation among predictors
    * same solution as for linear regression
  – nonlinearity in the logit
    * awkward to check
    * no consensus on how to deal with
  – poor model fit
    * Can consider a goodness-of-fit test (Hosmer-Lemeshow $\chi^2$ test)
  – High separation or discrimination...

• What is high separation?

If there’s no overlap of the x’s that lead to a 0 and the x’s that lead to a 1.

Depending on the degree of separation, this can cause problems with fitting the model.

This inflates SE of regression coefficient.

If there’s perfect separation, coefficients can not be estimated.

## Code for graphic of Hosmer-Lemeshow Test:

```r
## Break the x-axis into 10 groups:
break.points=quantile(soil,seq(0,1,0.1))
group.soil=cut(soil,breaks=break.points)
table(group.soil)
## For each group, get the estimated p-hat:
group.est.p=tapply(highbld,group.soil,mean)
group.est.p=as.vector(group.est.p)
se=as.vector(sqrt(group.est.p*(1-group.est.p)/table(group.soil)))
## Get mean x-values for each group for plotting purposes:
group.x.means=tapply(soil,group.soil,mean)
group.x.means=as.vector(group.x.means)
## Overlay plots:
xvalues=seq(0,5300)
yvalues=predict(glm.out,list(soil=xvalues),type="response")
plot(highbld~soil,type="n")
rug(jitter(soil[highbld==0]))
rug(jitter(soil[highbld==1]),side=3)
lines(xvalues,yvalues,col="blue",lwd=2)
points(group.x.means,group.est.p,cex=2,pch=16)
abline(v=vlines)
legend(2700,.6,c("grouped p-hat","logistic reg. p-hat","95% CI for grouped p-hat"),
      col=c("black","blue","red"),lty=c(-1,1,1),
pch=c(16,-1,-1),lwd=2)
up=group.est.p+2*se
down=group.est.p-2*se
for (i in 1:10){
  lines(c(group.x.means[i],group.x.means[i]),
       c(up[i],down[i]),col="red",lwd=2)
}
points(group.x.means,group.est.p,ces=2,pch=16)

hosmerlem=function (y, yhat, g = 10) {
  cutyhat = cut(yhat, breaks =
                quantile(yhat, probs = seq(0,1, 1/g)),
                include.lowest = T)
  obs = xtabs(cbind(1 - y, y) ~ cutyhat)
  expect = xtabs(cbind(1 - yhat, yhat) ~ cutyhat)
  chisq = sum((obs - expect)^2/expect)
  P = 1 - pchisq(chisq, g - 2)
  c("X^2" = chisq, Df = g - 2, "P(>Chi)" = P)
}

hosmerlem(highbld, glm.out$fitted, g = 10)
```

```r
hosmerlem=function (y, yhat, g = 10) {
  cutyhat = cut(yhat, breaks =
                quantile(yhat, probs = seq(0,1, 1/g)),
                include.lowest = T)
  obs = xtabs(cbind(1 - y, y) ~ cutyhat)
  expect = xtabs(cbind(1 - yhat, yhat) ~ cutyhat)
  chisq = sum((obs - expect)^2/expect)
  P = 1 - pchisq(chisq, g - 2)
  c("X^2" = chisq, Df = g - 2, "P(>Chi)" = P)
}

hosmerlem(highbld, glm.out$fitted, g = 10)
```