Goodness of Fit in Logistic Regression

As in linear regression, goodness of fit in logistic regression attempts to get at how well a model fits the data. It is usually applied after a "final model" has been selected.

As we have seen, often in selecting a model no single "final model" is selected, as a series of models are fit, each contributing towards final inferences and conclusions. In that case, one may wish to see how well more than one model fits, although it is common to just check the fit of one model. This is not necessarily bad practice, because if there are a series of "good" models being fit, often the fit from each will be similar.

Recall once again the quote from George Box:

"All Models are wrong, but some are useful."

It is not clear how to judge the fit of a model that we know is in fact wrong. Much of the goodness of fit literature is based on hypothesis testing of the following type:

$$\begin{aligned} H_0 &: \text{model is exactly correct} \\ H_A &: \text{model is not exactly correct} \end{aligned}$$

This type of testing provides no useful information. If the null hypothesis is rejected, then we have learned nothing, because we already knew that it is impossible for any model to be "exactly correct".

On the other hand, if we do not reject the model, it is almost surely because of a lack of statistical power, and as the sample size grows larger, we will eventually surely reject H_0.

These tests can be seen not only as not useful, but as harmful if non-rejection of a null hypothesis is misinterpreted as proof that the model "fits well", which is of course can be far from the truth.

If these tests are not useful (despite their popularity in some circles), what else can we do?

We can attempt to derive various descriptive measures of how well a model fits,

and then try to make a judgement concerning whether any discrepancies we see will likely affect our use of the model for its intended purpose (e.g., predictions for future subjects, or the association between any particular independent variable and the outcome).

The above is a difficult task, no perfect solutions exist, and much methodological research is still ongoing in this area.

We will look at some solutions that have been proposed, and see some examples of their use.

Goodness Of Fit Measures for Logistic Regression

The following measures of fit are available, sometimes divided into "global" and "local" measures:

- Chi-square goodness of fit tests and deviance
- Hosmer-Lemeshow tests
- Classification tables
- ROC curves
- Logistic regression R^2
- Model validation via an outside data set or by splitting a data set

For each of the above, we will define the concept, see an example, and discuss the advantages and disadvantages of each.

Chi-Square Goodness Of Fit Tests and Deviance

In linear regression, residuals can be defined as

$$y_i - \hat{y}_i$$

where y_i is the observed dependent variable for the i^{th} subject, and $\hat{y}_i$ the corresponding prediction from the model.

The same concept applies to logistic regression, where y_i is necessarily equal to either 1 or 0, and

$$\hat{y}_i = \hat{\pi}_i(x_i) = \frac{\exp(\alpha + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})}{1 + \exp(\alpha + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})}$$

Two tests can be based on these residuals:

Chi-square test: Define a standardized residual as (recall the standard deviation of the binomial distribution to be $p(1-p)$):

$$r_i = \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i(1-\hat{y}_i)}}$$

One can then form a χ^2 statistic as

$$X^2 = \sum_{i=1}^{n} r_i^2$$

The X^2 statistic follows a χ^2 distribution with $n - (p+1)$ degrees of freedom, so that p-values can be calculated (if desired, see above note about such tests).

Note how similar this is to the situation for linear regression χ^2 tests.

Technical Point: If some "covariate patterns" are repeated more than once, so that there are $J < n$ patterns, then the above test statistic changes to a sum over J rather than over n, and y_i changes to the number of successes over all individuals with that pattern. The term $\hat{y}_i$ remains the same, as it is the same across individuals with the same covariate pattern anyway.

Second Technical Point: The χ^2 distribution is not very accurate when $J \approx n$, so tests not very accurate. One way around this is to group "similar" covariate patterns together, so that $J < n$.

This is not a big concern for us, as we avoid testing anyway. However, the idea of combining similar covariate patterns is a useful one for logistic regression goodness of fit checking. In fact, we have already seen this idea used when we examined a plot of age versus CHD incidence earlier on in the course.

In general, it is likely that $J = n$ if there are one or more continuous covariates, but most often $J < n$ if all covariates are categorical.

Deviance test: A very similar test, also χ^2 distribution with $n - (p+1)$ degrees of freedom (and same technical point above) can be derived from "deviance residuals". See Hosmer and Lemeshow, page 146 for details.

Hosmer-Lemeshow methods

Continuing with the above idea re grouping, consider fitting a logistic regression model, calculating all fitted values $\hat{y}_i$, and grouping the covariate patterns according to the ordering of $\hat{y}_i$, from lowest to highest, say.

For example, if there are 100 different covariate patterns, each with fitted value $\hat{y}_j$, $j = 1, 2, \ldots, J$, create ten groupings, the first with the 10 lowest $\hat{y}_j$'s, then the next ten lowest, and so on. In this way, one can create a (2 by 10, in this case) table of observed numbers of successes in each group (or average number, if one divides by 10 in this case) versus average prediction, $\overline{\pi_k}$, $k = 1, 2, \ldots g = 10$.

One can then either plot observed versus expected (much like the age versus CHD graph we have previously seen), or create yet another χ^2 test based on the table (we will tend not to use such tests, see Hosmer and Lemeshow page 148 for specific formulae).

This is a nice idea, and such graphs can give a nice picture of overall fit across the spectrum of predicted probabilities but beware of combining categories that may in fact be quite different in observations, even if predictions are close.

Classification tables

In an idea similar to that above, one can again start by fitting a model and calculating all fitted values. Then, one can choose a cutoff value on the probability scale, say 50%, and classify all predicted values above that as predicting an event, and all below that cutoff value as not predicting the event.

Now, we construct a two-by-two table of data, since we have dichotomous observed outcomes, and have now created dichotomous "fitted values", when we used the cutoff.

Thus, we can create a table as follows:

	Observed positive	Observed negative
Predicted positive (above cutoff)	a	b
Predicted negative (below cutoff)	c	d

Of course, we hope for many counts in the a and d boxes, and few in the b and c boxes, indicating a good fit.

In an analogy with medical diagnostic testing, we can consider the following quantities:

$$\text{sensitivity} = \frac{a}{a+c}$$

and

$$\text{specificity} = \frac{d}{b+d}$$

Higher sensitivity and specificity indicate a better fit of the model.

ROC curves

Extending the above two-by-two table idea, rather than selecting a single cutoff, we can examine the full range of cutoff values from 0 to 1. For each possible cutoff value, we can form a two-by-two table.

Plotting the pairs of sensitivity and specificities (or, more often, sensitivity versus one minus specificity) on a scatter plot provides an ROC (Receiver Operating Characteristic) curve.

The area under this curve (AUC of the ROC) provides an overall measure of fit of the model.

In particular, the AUC provides the probability that a randomly selected pair of subjects, one truly positive, and one truly negative, will be correctly ordered by the test. By "correctly ordered", we mean that the positive subject will have a higher fitted value (i.e., higher predicted probability of the event) compared to the negative subject.

Logistic regression R^2

As we have seen above, having defined residuals for logistic regression, we can form the usual R^2 statistic, although it is rarely used. It is almost always rather low, since observed values need to be either 0 or 1, but predicted values are always in between these extremes. See Hosmer and Lemeshow page 164 for details (they themselves recommend not using this method).

Model validation via an outside data set or by splitting a data set

As in linear regression, one can attempt to "validate" a model built using one data set by finding a second independent data set and checking how well the second data set outcomes are predicted from the model built using the first data set.

Our comments there apply equally well to logistic regression. To summarize: Little is gained by data splitting a single data set, because by definition, the two halves must have the same model. Any lack of fit is then just by chance, and any evidence for good fit brings no new information. One is better off using all the data to build the best model possible.

Obtaining a new data set improves on the idea of splitting a single data set into two parts, because it allows for checking of the model in a different context.

If the two contexts from which the two data sets arose were different, then, at least, one can check how well the first model predicts observations from the second model. If it does fit, there is some assurance of generalisability of the first model to other contexts. If the model does not fit, however, one cannot tell if the lack of fit is owing to the different contexts of the two data sets, or true "lack of fit" of the first model.

In practice, these types of validation can proceed by deriving a model and estimating its coefficients in one data set, and then using this model to predict the Y variable from the second data set. One can then check the residuals, and so on.

Example

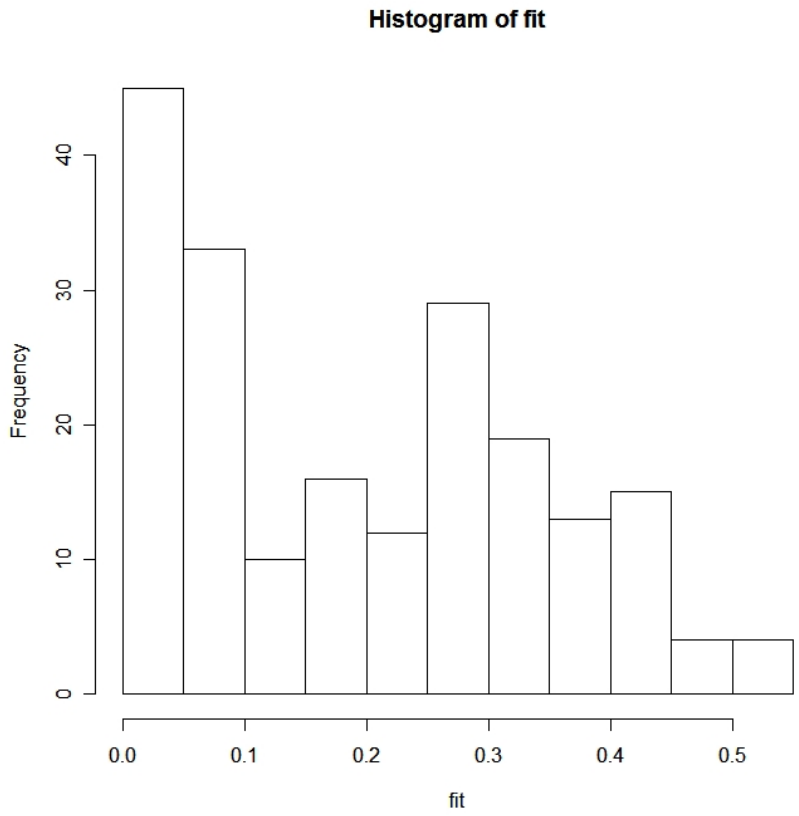
We will now apply several of the above methods in an example.

We will use the icu data previously used when we looked at multiple logistic regression.

Description	Coding	variable name
Vital Status (Main outcome)	0 = Lived 1 = Died	STA
Age	Years	AGE
Sex	0 = Male 1 = Female	SEX
Race	1 = White 2 = Black 3 = Other	RACE
Service at ICU Admission	0 = Medical 1 = Surgical	SER
Cancer Part of Present Problem	0 = No 1 = Yes	CAN
History of Chronic Renal Failure	O = No 1 = Yes	CRN
Infection Probable at ICU Admission	0 = No 1 = Yes	INF
CPR Prior to ICU Admission	0 = No 1 = Yes	CPR
Systolic Blood Pressure at ICU Admission	mm Hg	SYS
Heart Rate at ICU Admission	Beats/min	HRA
Previous Admission to an ICU within 6 Months	0 = No 1 = Yes	PRE
Type of Admission	0 = Elective 1 = Emergency	TYP
Long Bone, Multiple, Neck, Single Area, or Hip Fracture	0 = No 1 = Yes	FRA
PO2 from Initial Blood Gases	$0 > 60$ $1 \leq 60$	PO2
PH from Initial Blood Gases	$0 \geq 7.25$ $1 < 7.25$	PH
PCO2 from initial Blood Gases	$0 \leq 45$ $1 > 45$	PCO
Bicarbonate from Initial Blood Gases	$0 \geq 18$ $1 < 18$	BIC
Creatinine from Initial Blood Gases	$0 \leq 2.0$ $1 > 2.0$	CRE
Level of Consciousness at ICU Admission	O = No Coma or Stupor 1 = Deep stupor 2 = Coma	LOC

```
# Read in full data set

> icu.dat <- read.table(file="g:\\icudat.txt", header = T)
>
> summary(icu.dat)
      sta              age              sex             race            ser
 Min.   :0.0   Min.   :16.00   Min.   :0.00   Min.   :1.000   Min.   :0.000
 1st Qu.:0.0   1st Qu.:46.75   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.000
 Median :0.0   Median :63.00   Median :0.00   Median :1.000   Median :1.000
 Mean   :0.2   Mean   :57.55   Mean   :0.38   Mean   :1.175   Mean   :0.535
 3rd Qu.:0.0   3rd Qu.:72.00   3rd Qu.:1.00   3rd Qu.:1.000   3rd Qu.:1.000
 Max.   :1.0   Max.   :92.00   Max.   :1.00   Max.   :3.000   Max.   :1.000

      can              crn              inf             cpr             sys
 Min.   :0.0   Min.   :0.000   Min.   :0.00   Min.   :0.000   Min.   : 36.0
 1st Qu.:0.0   1st Qu.:0.000   1st Qu.:0.00   1st Qu.:0.000   1st Qu.:110.0
 Median :0.0   Median :0.000   Median :0.00   Median :0.000   Median :130.0
 Mean   :0.1   Mean   :0.095   Mean   :0.42   Mean   :0.065   Mean   :132.3
 3rd Qu.:0.0   3rd Qu.:0.000   3rd Qu.:1.00   3rd Qu.:0.000   3rd Qu.:150.0
 Max.   :1.0   Max.   :1.000   Max.   :1.00   Max.   :1.000   Max.   :256.0

      hra              pre              typ             fra             po2
 Min.   : 39.00   Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :0.00
 1st Qu.: 80.00   1st Qu.:0.00   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.00
 Median : 96.00   Median :0.00   Median :1.000   Median :0.000   Median :0.00
 Mean   : 98.92   Mean   :0.15   Mean   :0.735   Mean   :0.075   Mean   :0.08
 3rd Qu.:118.25   3rd Qu.:0.00   3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.:0.00
 Max.   :192.00   Max.   :1.00   Max.   :1.000   Max.   :1.000   Max.   :1.00

       ph              pco              bic             cre             loc
 Min.   :0.000   Min.   :0.0   Min.   :0.000   Min.   :0.00   Min.   :0.000
 1st Qu.:0.000   1st Qu.:0.0   1st Qu.:0.000   1st Qu.:0.00   1st Qu.:0.000
 Median :0.000   Median :0.0   Median :0.000   Median :0.00   Median :0.000
 Mean   :0.065   Mean   :0.1   Mean   :0.075   Mean   :0.05   Mean   :0.125
 3rd Qu.:0.000   3rd Qu.:0.0   3rd Qu.:0.000   3rd Qu.:0.00   3rd Qu.:0.000
 Max.   :1.000   Max.   :1.0   Max.   :1.000   Max.   :1.00   Max.   :2.000

#  We will use just three covariates in this example, age, sex, and typ.
#  Two of these are dichotomous, and one is continuous.

#  First run the logistic regression model, and get the fitted values.

> output <- glm(sta ~ age + sex + typ, family=binomial, data = icu.dat)
>
> summary(output)
```

```
Call:
glm(formula = sta ~ age + sex + typ, family = binomial, data = icu.dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2455  -0.7898  -0.4122  -0.2292   2.5102

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.50146    1.03524  -5.314 1.07e-07 ***
age          0.03489    0.01088   3.206 0.001345 **
sex         -0.22173    0.39154  -0.566 0.571185
typ          2.48540    0.75489   3.292 0.000993 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 200.16  on 199  degrees of freedom
Residual deviance: 172.75  on 196  degrees of freedom
AIC: 180.75

Number of Fisher Scoring iterations: 6

#  Get the fitted values and plot them

> fit <- output$fitted

> hist(fit)
```



Note that none are much larger than 0.5, but the event rate is only 20%.

We will first calculate the χ^2 residuals

$$r_i = \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i(1-\hat{y}_i)}}$$

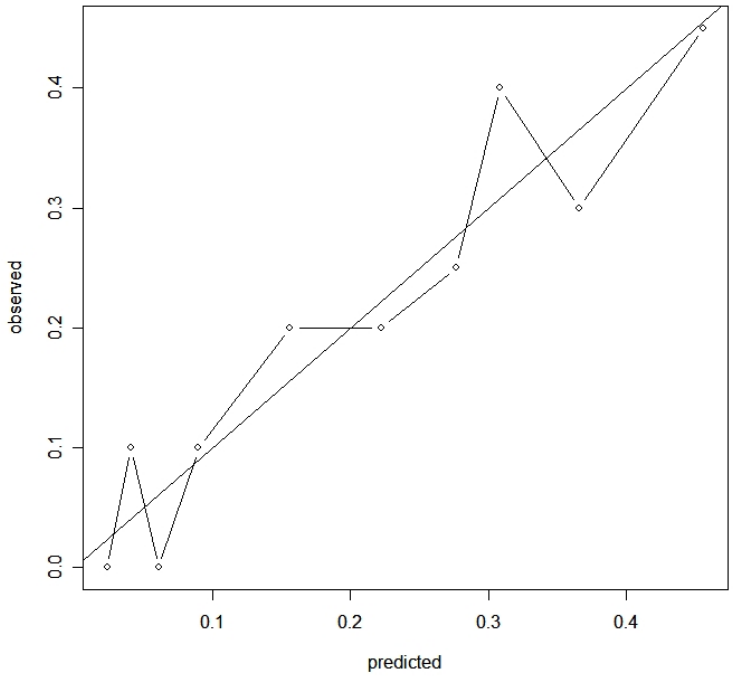
across all individuals, and then go on to other goodness of fit methods.

```
# We will now investigate each of the above methods
# for logistic regression goodness of fit.

################################################
#  Chi-square goodness of fit test
################################################

# Calculate residuals across all individuals

> r <- (icu.dat$sta - fit)/(sqrt(fit*(1-fit)))
```

```
#  Sum of squares of these residuals follows a chi-square
#  with 200 - 4 = 196 degrees of freedom

> sum(r^2)
[1] 190.7172

#  Calculate the p-value from the test

> 1- pchisq(190.7172, df=196)
[1] 0.5930901

#  So, we cannot reject the null hypothesis that this
#  model is exactly correct...but then again, we know that
#  the model is wrong!  So, not too useful a method!

#  On to something a bit more useful.

################################################
#  Hosmer-Lemeshow tests
################################################

#  Strategy to calculate the Hosmer-Lemeshow groupings:
#  Form a matrix with the outcome and fitted values,
#  and re-order according to fitted value probabilities.

#  Get indices of vector fit, from smallest to greatest

> index <- sort.list(fit)

#  Look at 10 smallest indices

> index[1:10]
 [1]  61  58 102 141 110  55  12  11  16  36

# Create a matrix of sta and fit, using this index

> hosmer <- matrix(c(icu.dat$sta[index], fit[index]), byrow=F, nrow=200)
> hosmer
      [,1]        [,2]
  [1,]    0 0.006303336
  [2,]    0 0.007588489
  [3,]    0 0.007588489
  [4,]    0 0.010722396
  [5,]    0 0.019904908
  [6,]    0 0.020597020
```

```
  [7,]    0 0.021312675
  [8,]    0 0.021786470
 ...........etc........
[100,]    1 0.183401019
[101,]    1 0.188683609
[102,]    0 0.190591487
[103,]    0 0.196031469
[104,]    1 0.199597241
[105,]    0 0.201588052
 ...........etc........
[196,]    0 0.466823972
[197,]    1 0.513483787
[198,]    0 0.539569019
[199,]    1 0.539569019
[200,]    1 0.548223198

#  Now to group into 10 groups each with 20 observations, say, and graph:

#  Create a blank vector to store results

> observed <- rep(NA, 10)

> for (i in 1:10) {observed[i] <- sum(hosmer[(20*(i-1)+1):(20*i),1])/20}

#  Look at observed rates

> observed
 [1] 0.00 0.10 0.00 0.10 0.20 0.20 0.25 0.40 0.30 0.45

#  Do same for predicted rates

#  Create a blank vector to store results

> predicted <- rep(NA, 10)

> for (i in 1:10) {predicted[i] <- sum(hosmer[(20*(i-1)+1):(20*i),2])/20}

#  Look at predicted rates

> predicted
 [1] 0.02350724 0.04103168 0.06069354 0.08979742 0.15623928 0.22267946 0.27641412
 [8] 0.30829473 0.36543618 0.45590633

#  Now plot observed versus predicted
```

```
>  plot(predicted, observed, type="b")

#  Add 45% line to plot

>  abline(a=0, b=1)
```



Note the generally reasonable fit, maybe a bit of trouble predicting some categories.

```
#####################################################
#  Classification tables
#####################################################

#  If we choose a cutoff of 50%, from the hosmer matrix,
#  we can see ...

> hosmer
       [,1]        [,2]
  [1,]    0 0.006303336
  [2,]    0 0.007588489
...........etc......
[195,]    0 0.461235253
```

```
[196,]    0 0.466823972
[197,]    1 0.513483787
[198,]    0 0.539569019
[199,]    1 0.539569019
[200,]    1 0.548223198

#  ...that only the last four are above 50%,
#  but 3/4 were in fact positive.
```

So, our two-by-two table becomes:

	Observed positive	Observed negative
Predicted positive (above cutoff)	3	1
Predicted negative (below cutoff)	37	159

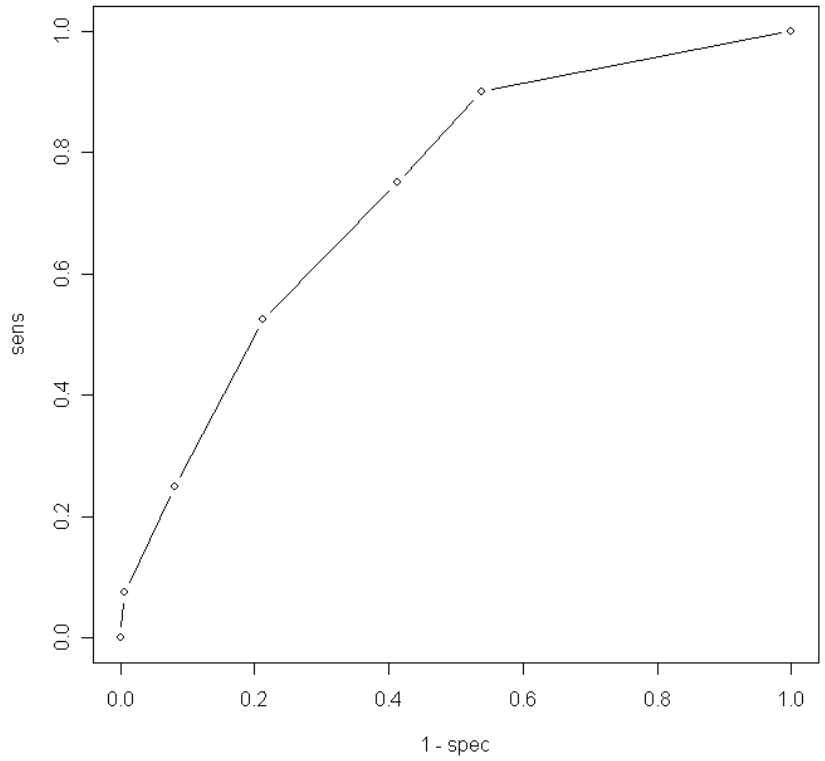
So that sensitivity = 3/40 = 7.5%, and specificity = 159/160 = 99.4%. So, it seems a cutoff of 50% will perform very badly here, missing over 90% of all true positive cases.

Let's see how some other cutoff values perform.

```
##################################################
#  ROC curves
##################################################

#  Again using the hosmer matrix, we can calculate
#  sensitivity and specificity for different
#  cutoff values, say from 0 to 1, by 10% increments.

#  We will fill in these blank vectors:

> sens <- rep(NA, 11)
> spec <- rep(NA, 11)

#  The first entries need no calculation
#  as they represent the extreme case of a 0%
#  cutoff for positivity.  So, by definition:

> sens[1] <- 1
> spec[1] <- 0
```

```
#  Cutoff of 10% for positivity (occurs at index 79)

> sens[2] = sum(hosmer[79:200,1])/40
> spec[2] = sum(1-hosmer[1:78,1])/160

#  Cutoff of 20% for positivity (occurs at index 105)

> sens[3] = sum(hosmer[105:200,1])/40
> spec[3] = sum(1-hosmer[1:104,1])/160

#  Cutoff of 30% for positivity (occurs at index 146)

> sens[4] = sum(hosmer[146:200,1])/40
> spec[4] = sum(1-hosmer[1:145,1])/160

#  Cutoff of 40% for positivity (occurs at index 178)

> sens[5] = sum(hosmer[178:200,1])/40
> spec[5] = sum(1-hosmer[1:177,1])/160

#  Cutoff of 50% for positivity (occurs at index 197)

> sens[6] = sum(hosmer[197:200,1])/40
> spec[6] = sum(1-hosmer[1:196,1])/160

#  Cutoff of 60% for positivity
> #  Since no points above this, rest all the same

> sens[7] = 0
> spec[7] = 1

> sens[8] = 0
> spec[8] = 1

> sens[9] = 0
> spec[9] = 1

> sens[10] = 0
> spec[10] = 1

> sens[11] = 0
> spec[11] = 1

> sens
 [1] 1.000 0.900 0.750 0.525 0.250 0.075
```

```
 [7]    0.000 0.000 0.000 0.000 0.000
> spec
 [1] 0.00000 0.46250 0.58750 0.78750 0.91875 0.99375 1.00000 1.00000
 [8] 1.00000 1.00000 1.00000

# Plot the graph of sens versus 1-spec to get the ROC curve

> plot(1-spec, sens, type="b")
```




```
#  We can also calculate the Area Under the ROC Curve as follows:
#  First, create separate vectors of the predicted fits
#  for positive and negative subjects

> fit.pos <- fit[icu.dat$sta==1]
> fit.neg <- fit[icu.dat$sta==0]
```

```
#  Now use nonparametric Wilcoxon test on these two samples

> wilcox.test(x=fit.pos, y=fit.neg)

        Wilcoxon rank sum test with continuity correction

data:  fit.pos and fit.neg
W = 4789.5, p-value = 1.213e-06
alternative hypothesis: true mu is not equal to 0

# Take the value of the W statistic, and divide by
# the total number of all possible pairs

> 4789.5/(160*40)
[1] 0.7483594

#  So, AUC = 74.8%.

#  Why does this work?  May have seen in 607 that Mann-Whitney
#  statistic is equivalent to probability of correctly
#  selecting the higher of a pair of numbers, which is equivalent
#  to the definition of the AUC of an ROC curve.
```

Final Note

Do not forget, amidst all these statistical ideas, that substantive knowledge and knowledge about a study design can and should play a role in thinking about a model, and how well it suits a given purpose.

If you have good *a priori* reasons to believe a variable should be in a model then simply include it, unless the evidence against it is very strong.

If the main point of a model is prediction, you might not care too much about which independent variables are included, as long as the model "fits well". But if the purpose of your model is to see which variables are important, then much attention needs to be paid to this issue.

Goodness of fit is closely related to model selection, which we will cover in the next lecture.