Regression Models
Lecture V: Model Building: Some suggestions

DT9002: Postgraduate Certificate in Applied Statistics

Dr Joe Condon

School of Mathematical Sciences
Technological University Dublin

Model Building: Introduction

- Often in regression model we have quite a number of candidate predictors and it is not clear which subset of them should be included in the model.
- Example: For the Bread Wrapper data should the model have both temperature and polyethylene as predictors, just one of them or none?
- Example: For the Dose Response data is the quadratic the best model, or should we use the cubic or even a quartic model?
- How do we select the best model? Define the best model as:

Best Model = The minimum, plausible model which adequately describes the data.

Underfitting

Imagine the 'real' ('true') model should have 8 predictors used, but a mistake is made and only 4 predictors are used.

What are the consequences of this mistake?

- The $\beta's$ are biased - i.e. not trustworthy estimates.
- The value of s^2 will be too big - this skews all hypothesis testing, confidence interval and prediction intervals. In particular, our Type II error rate increases.

Overfitting

Imagine the 'real' ('true') model should have 4 predictors used, but a mistake is made and 8 predictors are used - 4 of which are in reality unrelated to the response.

What are the consequences of this mistake?

- The $\beta's$ for the 4 predictors are unbiased - i.e. they remain trustworthy estimates.
- The value of s^2 will remain unbiased and trustworthy as well.
- There will be an increase in the variance of the parameter estimates (the variance-covariance matrix is affected) and hence confidence interval and prediction intervals will tend to be too wide and also we again get an increase in Type II error rates.

The moral of all this is: We need to avoid underfitting and overfitting - but a small amount of overfitting is arguably less of a problem.

In particular, a modest (but not excessive) amount of overfitting will still lead to an unbiased estimate $s^2 \approx \sigma^2$ and reasonable estimates of the parameters for those predictors that should be in the model.

Model Selection for Polynomial Models

Question for the Dose-Response Data : Do we need the cubic term in the model? To answer these lets propose a few solutions.

Define the statistic R^2 as the proportion of variation in the data explained by the model. More exactly it is:

$$R^2 = \frac{SSmodel}{SStotal} = 1 - \frac{SSerror}{SStotal}$$

So the explanatory power of the model can be assessed by reference to R^2.

Proposal (1): Just look at R^2 - the model (quadratic or cubic) with the highest R^2 is explaining most variation in the data and hence is to be preferred?

Problem: As you keep adding higher order terms R^2 keeps increasing.

Degree of Polynomial	R^2	Adjusted R^2
1	0.0120	-0.0978
2	0.8428	0.8035
3	0.9044	0.8634
4	0.9656	0.9427
5	0.9743	0.9485
6	0.9759	0.9398
7	0.9907	0.9691
8	0.9908	0.9542
9	0.9998	0.9978
10	1.0000	1.0000

The adjusted R^2 above is an adjusted version with respect to the number of parameters in the model. It comes from the fact that R^2 can be written as,

$$R^2 = \frac{SSmodel}{SStotal} = 1 - \frac{SSerror}{SStotal}$$

$$Adjusted\ R^2 = 1 - \frac{SSerror \times (n-1)}{SStotal \times (n-p)}$$

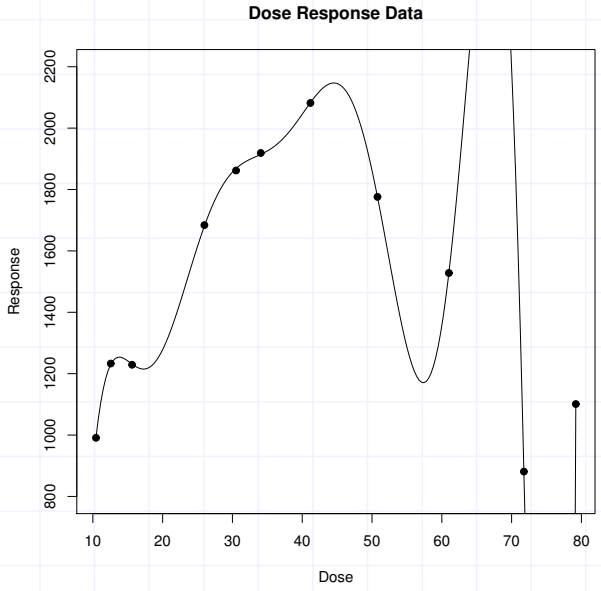
So, adjusted R^2 penalises models with more parameters.

As can be seen, using R^2 as a guide will favour a more complicated (bigger) model.

This is not good - we want to follow the law of parsimony, i.e. the simplest plausible model is best.

Adjusted R^2 is a bit better, but not the solution.

As an example, look at the polynomial of degree 10, here's a plot of the LS line it gives,



Which is an interpolating polynomial, which involves no simplification of the data.

When it comes to pure polynomial models, we can approach model fitting in the following way:

- The question is: what is the correct degree for the polynomial?
- If polynomial of degree r is fitted, then all coefficients in that polynomial should be left unconstrained - i.e. all coefficients are fitted even if some of them are statistically not significantly different from zero.
- The one exception to this rule, is the last coefficient of the polynomial, i.e. the coefficient for the x^r predictor.
- If the coefficient for x^r is not statistically significant (use a general linear hypothesis to test this) then it is removed and the simpler polynomial of degree $r-1$ is considered.
- You could proceed as follows:
  1. Start with the lowest degree polynomial that is plausible given a plot of the data.
  2. increase the degree by one, one step at a time. At each step check that that parameter for x^r is significantly different from zero - if it is, it stays in the model, if not use the polynomial with degree $r-1$.

Example: Doseresponse Data

```
> drop1(fit_dr2,test='F')
Single term deletions

Model:
activity ~ dose + I(dose^2)
          Df Sum of Sq     RSS    AIC F value    Pr(>F)
<none>                  266940 117.07
dose       1   1274269 1541210 134.35  38.189 0.0002652 ***
I(dose^2)  1   1410852 1677792 135.29  42.282 0.0001876 ***
```

```
> drop1(fit_dr3,test='F')
Single term deletions

Model:
activity ~ dose + I(dose^2) + I(dose^3)
          Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>                 162329 113.59
dose       1    423297 585626 125.71 18.2535 0.00369 **
I(dose^2)  1    225107 387436 121.16  9.7071 0.01695 *
I(dose^3)  1    104611 266940 117.07  4.5111 0.07131 .
```

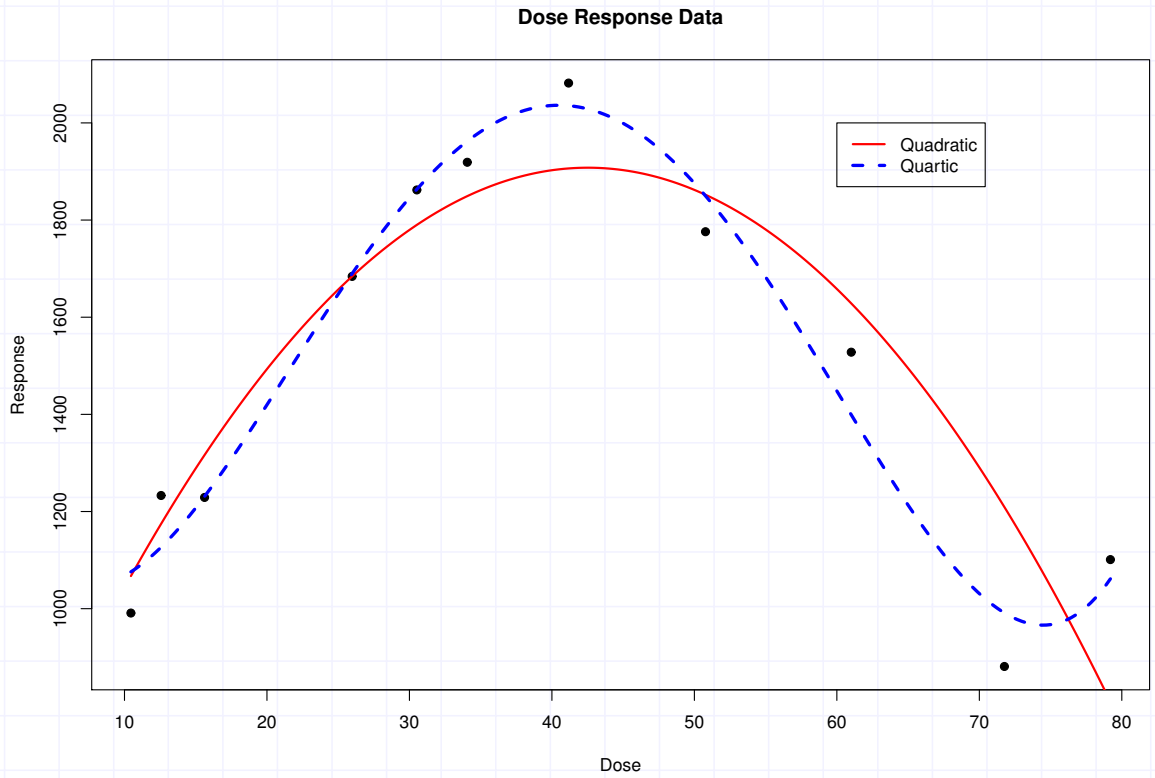

Since the degree 3 (cublic) is marginally significant, try a degree 4 (quartic).

```
> drop1(fit_dr4,test='F')
Single term deletions

Model:
activity ~ dose + I(dose^2) + I(dose^3) + I(dose^4)
          Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>                  58379 104.34
dose       1     12125  70504 104.42  1.2462 0.30698
I(dose^2)  1     51272 109651 109.28  5.2696 0.06147 .
I(dose^3)  1     85614 143993 112.28  8.7992 0.02508 *
I(dose^4)  1    103950 162329 113.59 10.6836 0.01706 *
```

Since the degree 4 is highly statistically significant, try a degree 5 (quintic)

```
> drop1(fit_dr5,test='F')
Single term deletions

Model:
activity ~ dose + I(dose^2) + I(dose^3) + I(dose^4) + I(dose
    ^5)
           Df Sum of Sq   RSS    AIC F value Pr(>F)
<none>                 43690 103.16
dose        1    4942.0 48632 102.34  0.5656 0.4859
I(dose^2)   1    2845.2 46535 101.85  0.3256 0.5929
I(dose^3)   1    4560.3 48251 102.25  0.5219 0.5024
I(dose^4)   1    8834.4 52525 103.18  1.0110 0.3608
I(dose^5)   1   14688.8 58379 104.34  1.6810 0.2514
```

OK - seems we might need a quadratic (degree 2) or a quartic (degree 4).



Some points about all this:

- At early stages we might err on the side of including higher degree terms, by accepting a higher than normal p-value for inclusion. This would somewhat offset the tendency of underfitted models to have a large Type II error rate.
- We need to refit the model at each stage - as the estimate of s^2 changes as do the parameters estimates.
- We might be a bit concerned that this method favours more complicated models - as it does not penalise models for the number of parameters included - see later for a suggested fix for this problem.

Model Selection for General Models

Example: Fitness Data;

These measurements were made on men involved in a physical fitness course. The variables are Oxygen intake rate (ml per kg body weight per minute) - the response variable, Age (years), Weight (kg), time to run 1.5 miles (minutes), heart rate while resting, heart rate while running (same time Oxygen rate measured), and maximum heart rate recorded while running.

Oxygen	Age	Weight	Time	Rest Pulse	Run Pulse	max Pulse
44.609	44	89.47	11.37	62	178	182
45.313	40	75.07	10.07	62	185	185
54.297	44	85.84	8.65	45	156	168
59.571	42	68.15	8.17	40	166	172
49.874	38	89.02	9.22	55	178	180
44.811	47	77.45	11.63	58	176	176
45.681	40	75.98	11.95	70	176	180
49.091	43	81.19	10.85	64	162	170
39.442	44	81.42	13.08	63	174	176
60.055	38	81.87	8.63	48	170	186
50.541	44	73.03	10.13	45	168	168
37.388	45	87.66	14.03	56	186	192
44.754	45	66.45	11.12	51	176	176
47.273	47	79.15	10.60	47	162	164
51.855	54	83.12	10.33	50	166	170
49.156	49	81.42	8.95	44	180	185
40.836	51	69.63	10.95	57	168	172
46.672	51	77.91	10.00	48	162	168
46.774	48	91.63	10.25	48	162	164
50.388	49	73.37	10.08	67	168	168
39.407	57	73.37	12.63	58	174	176
46.080	54	79.38	11.17	62	156	165
45.441	52	76.32	9.63	48	164	166
54.625	50	70.87	8.92	48	146	155
45.118	51	67.25	11.08	48	172	172
39.203	54	91.63	12.88	44	168	172
45.790	51	73.71	10.47	59	186	188
50.545	57	59.08	9.93	49	148	155
48.673	49	76.32	9.40	56	186	188
47.920	48	61.24	11.50	52	170	176
47.467	52	82.78	10.50	53	170	172

Method 1: All possible Regressions

In this method we fit all possible regressions, i.e.,

- The intercept only model
- Set of all possible 1 predictor models
- Set of all possible 2 predictor models

  ⋮
- Set of all possible p-1 predictor models

Then identify the best model in each parameter set and choose between those models.

The best model is the model in each set with the largest adjusted R^2. Note: if there are $p-1$ predictor variables under consideration, then there are $2^{(p-1)}$ possible regressions.

The models found with the best adjusted R^2 in each set were;

No. Predictors	Model Terms	Adjusted R^2
1	Run Time	0.7345
2	Age, Run Time	0.7474
3	Age, Run Time, Run Pulse	0.7901
4	Age, Run Time, Run Pulse, Max Pulse	0.8117
5	Age, Weight, Run Time, Run Pulse, Max Pulse	0.8176
6	Age, Weight, Run Time, Run Pulse, Rest Pulse, Max Pulse	0.8108

These are the best from the set of 63 regressions [See R printout].

So, which model do we choose?

Well look at the 6 candidate models above and choose using sequential F tests.

Or, just take the smallest model with an acceptable level of adjusted R^2. There are two problems with all this;

1. suppose you had 25 predictors to start with, then you have some 33,554,432 regressions to fit. Even with fast computing this will take some time.
2. Since you are choosing models based on selecting the best model (i.e. the model with the lowest s^2 - this is equivalent to selecting the highest adjusted R^2) from a large number of models, then the p-values you calculate are being biased (this is sometimes called interrogating the data/data dredging). Rejection of the H_0 : may be virtually guaranteed in many instances and it is not obvious how you should adjust the p-values to take account of this.

Method 2: Forward Selection

1. Consider all predictors not already included in the model one at a time for entry to the model. The single predictor selected as a candidate for entry at a stage is that predictor not already in the model that results in the lowest p-value when tested using a GLH.
2. However, for the candidate predictor to enter, it must have a p-value less than a pre-specified **entry criterion**).
3. The entry criterion is typically set higher than the standard 0.05 level. This is because of the likelihood that at the early stages of the procedure there will be underfitted models and hence p-values may be subject to upward bias.
4. Once a predictor has been included in the model it is not removed.
5. Stop when the candidate predictor at a stage fails to meet the entry criterion.

See R output for Fitness DATA.

Method 3: Backward Selection

1. Begin by calculating the p-values for a model which includes all possible predictors.
2. Predictors are considered for removal from the model one at a time.
3. The predictor chosen as a candidate for removal at any stage in that predictor with the largest p-value. However, for this predictor to be removed from the model, the p-value must be greater than a predefined significance level called the (**stay criterion**, lets say=.05).
4. Once a predictor is removed it is not considered for re-entry to the model.
5. Stop when the candidate predictor has a p-value less than the stay criterion

See R output for Fitness DATA.

Method 4: Stepwise Selection

1. The stepwise algorithm is a compromise between the forward and backward algorithms. It differs from the forward selection algorithm by allowing predictors already included in the model to be removed at later stages.
2. As in forward selection, predictors are considered for entry to the model one at a time. The candidate predictor for inclusion at any stage is that predictor not already included that has the lowest p-value.
3. For a candidate predictor to be included however, its p-value must be less than a pre-specified entry criterion.
4. At any stage where a predictor has been added to the model, the p-values are computed for all predictors in the model at that stage. A candidate predictor for removal from the model is selected as that predictor with the largest p-value.

5. Once a candidate predictor for removal is identified, it can only be removed if its p-value is greater than a pre-specified stay criterion.
6. If the candidate predictor is removed from the model, then the 'amended' model is refitted, and a check is made on those remaining predictors to see if any fail to meet the stay criterion,
7. The stepwise process ends when the candidate predictor for entry fails the entry criterion, or when the predictor to be included in the model is the one removed in the immediately preceding stage.

See R output for Fitness DATA.

Information Criteria and Model building

There are a number of Information Criteria that are routinely used in model building. The most widely used is the Akaike Information Criterion (AIC).

AIC is based on fairly complex theory from information entropy - so we will omit any deep discussion. It takes a remarkably simple form however:

AIC $= -2$ log likelihood (evaluated at $\hat{\beta}$)$+2p$

The **likelihood** for a set of data is (proportional to) the joint probability of the observed data given the model being fitted. Most of modern statistics relies on likelihood theory. The idea is to choose parameter estimates that make what was seen in the data most likely - i.e. most compatible with the evidence in the data. These are called **maximum likelihood estimates (MLE)**.

(Log) Likelihood

Likelihood:

$$L\left(\beta,\sigma^2\right) = \left(2\pi\sigma^2\right)^{\frac{-n}{2}} exp\left\{\frac{-1}{2\sigma^2}\left(Y-X\hat{\beta}\right)'\left(Y-X\hat{\beta}\right)\right\}$$

log Likelihood:

$$\ell\left(\beta,\sigma^2\right) = \frac{-n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(Y-X\hat{\beta}\right)'\left(Y-X\hat{\beta}\right)$$

NB: The least squares parameter estimates are the MLE where we assume the responses are normally distributed with equal variance.

AIC $= -2$ log likelihood (evaluated at $\hat{\beta}$)$+2p$

In this version the model with smaller AIC is preferred, therefore models with more parameters are penalised.

This is helpful, as the log likelihood is a non-decreasing function of the number of parameters (not unlike R^2).

Another famous criterion is the Bayesian Information criterion (aka Schwarz). It is very similar:

BIC $= -2$ log likelihood (evaluated at $\hat{\beta}$)$+p \log n$

The main advantage of the AIC is that is can be used to compare different models without relying on p-values - with the model with the smaller AIC preferred. This avoids some of the problems with type I error rates from multiple testing and concerns about underfitting.

Drawbacks to AIC include:

(1) we don't know its distribution
(2) it can be misleading in small samples.

In the case of drawback (2) we can use the corrected AIC or AICc:

AICc $= -2$ log likelihood (evaluated at MLE) $+2p\frac{n}{n-p-1}$

For the AIC, AICc and BIC the general idea is to control model overfitting by including a model complexity penalty in the criterion.

Method 5: Model Selection using AIC

Forward Selection: Same as before, only now the candidate for entry is that predictor which gives the greatest improvement in AIC when added to the model. Stop when no predictor gives an improvement to the AIC.

Backward Selection: Same as before, only now the candidate for removal is that predictor which gives the greatest improvement in AIC when removed from the model. Stop when no predictor improves the AIC by removal.

Stepwise Selection: Same as forward, only now at each stage check all predictors included in the model to see if removing them results in an improved AIC . Stop when neither the removal or addition of a predictor to the model results in an improvement to AIC.

See the results from R for the fitness data.

Criticisms of Stagewise algorithms

There are many of thes. Use an internet search and prepare yourself for a torrent to negative comments, including:

- There is no guarantee that they lead to the best, good or even plausible models.
- The p-values are not trustworthy as you are searching through a large number of models and cherry picking certain models which are going to have low p-values (AIC based methods could be similarly criticised). This will lead to potentially 'noise' predictors being selected and given a relevance that is false and inflated importance for other relevant predictors.
- There is no regard to scientific theory in the selection of important predictors.
- Often leads to 'overfitting' the dataset and therefore models with poor predictive power when it comes to new data.

Nevertheless, these are still widely used and do suggest some routes through a potentially vast number of models.

Method 6: Lasso Algorithm

Alternatives to stagewise selection methods include more general penalty based methods - one of these is the lasso.

The lasso (least absolute shrinkage and selection operator) can be defined as follows (NB: there are various equivalent formulations):

$$Q_{lasso} = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \ldots - \beta_{p-1} x_{ip-1})^2 + \lambda \sum_{j=1}^{p-1} |\beta_j|$$

The x's here are assumed centred and scaled so that the β_j's are all on the same scale. The estimators of the β's are those values that minimise Q_{lasso}.

The idea: The penalty says that for a β to play a big part in the model (i.e. get big) it must really work hard to earn its place by minimising the error sum of squares.

If $\lambda = 0$ then we are doing regular least squares and nothing more.

On the other hand if λ gets very big the lasso will result in no predictors being included in the model.

The lasso is sometimes recommended as it often results in some/many of the parameters shrinking to zero - i.e. that predictor is effectively removed from the model and the lasso is playing the role of predictor selection.

```
1              OLS           lasso                lasso
2  (lambda:    NA            0                    10000)
3
4  (Intercept) 102.93447948  102.95406056         47.37581
5  age          -0.22697380   -0.22705998
6  weight       -0.07417741   -0.07409882
7  runtime      -2.62865282   -2.62934835
8  restpulse    -0.02153364   -0.02159399
9  runpulse     -0.36962776   -0.36863825
10 maxpulse      0.30321713    0.30218801
```

Lasso: Estimating λ using Cross-Validation

λ is estimated by a general techniques called **cross-validation**. The idea is as follows:

- Imagine we had not one dataset but 2 datasets. Call them the 'training' dataset and 'validation' dataset.
- Fit (train) the model to the training dataset by choosing a value for λ, e.g. $\lambda = 0.1$. Calculate the $\hat{\beta}_j$'s for this value of λ.
- Assess (validate) this choice of λ by seeing how well the resulting model predicts the responses for the validation dataset. The idea of this validation (hold-out) dataset is that if you have overfitted you model by including predictors that only look significant on the training dataset but don't generalise to other datasets, then hopefully this will become apparent when the predictions for the validation data are relatively poor.

- You can measure how close the predictions are to the actual responses using the Mean Squared Error statistic:

$$MSE : \frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{y}_i(x_i)]^2$$

where:

- y_i is the recorded response for observation i on the validation dataset,
- $\hat{y}_i(x_i)$ is the predicted (fitted) value for this response using the parameters from the trained model and the predictors (x_i) for observation i:

$$\hat{y}_i(x_i) = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \ldots + \hat{\beta}_{p-1} x_{ip-1}$$

- Redo this for different value of λ and choose the λ giving the lowest MSE.

Cross-validation in Glmnet

The steps are as follows:

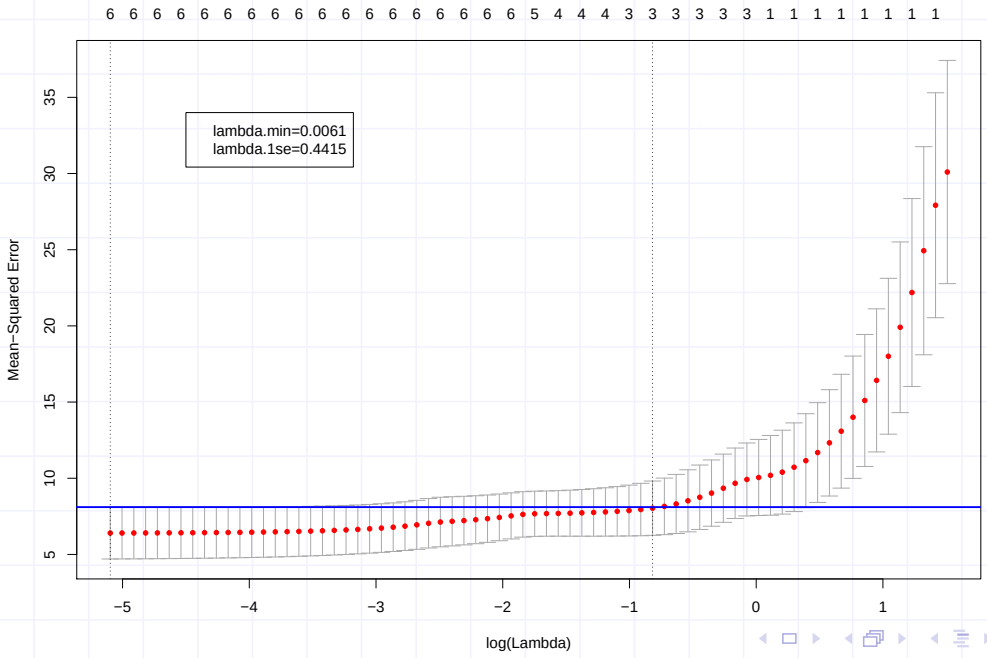
- Find the lowest value such that no predictors are included in the model - denote this λ_{max}.
- Search between λ_{max} and 0 - say use a list of 100 values equally spaced between them.
- Split the single dataset you have into 10 'folds' (sub samples) randomly.
- Fit a lasso model to 9 of the folds using a particular value for λ from the list. Use the unused fold a the validation dataset to calculate the MSE. Repeat this process for each fold - leaving a different fold out each time. This gives 10 estimates of the MSE for λ.
- Repeat for each λ in the list.

Cross-validation in Glmnet continued..

- For each λ in the list use the 10 estimates of MSE to calculate the mean MSE and the standard error of this mean (standard deviation of the 10 values divided by $\sqrt{10}$).
- The 'best' λ to use is either (a) the λ giving the lowest mean MSE, or (b) the largest λ no more that 1 standard error away from the lowest (this is called the 1.se rule).
- Using (b) to choose λ is often preferred as it results in a smaller model and therefore even less subject to overfitting.
- Once λ has been chosen, fit the model to the whole dataset using this λ.

Cross-validation Step in R

```
set.seed(2987887)
cv=cv.glmnet(preds,oxygen,family="gaussian",alpha=1)
plot(cv)
```



So, for the fitness data use $\lambda = 0.3339$.

```
> lasso_fit=glmnet(preds,oxygen,family="gaussian",alpha=1,
    lambda=cv$lambda.1se)
> coef(lasso_fit)
7 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) 94.63690740
age         -0.13498328
weight       .
runtime     -2.72818546
restpulse    .
runpulse    -0.07040887
maxpulse     .
```

NB: These are not the same as for the normal OLS fit with these predictors. Also, hypothesis testing etc. needs more work - we can no longer use our usual techniques.