Variable Selection and Model Building

So far, we have usually worked with models in which (almost) all predictor variables were important for the response. In reality, that's often not the case. The strategy that we have largely pursued so far was as follows:

1.) Fit the full model (with all available predictors and interaction terms).

2.) Perform a thorough analysis of the fit of this model (looking at the scatterplots, R^2 and performing a residual analysis). At this stage, we're on the lookout for problems such as violations of residual assumptions or multicollinearity between the predictors.

3.) Determine transformations of the response and for of some (or all) of the predictors if necessary.

4.) Check that all terms in the model are significant (small p-values for individual t-tests for slopes). Exclude non-significant predictors as necessary and refit the model.

5.) Re-evaluate the final possibly transformed and downsized model. Repeat the residual analysis.

Theoretically, one should re-evaluate the fit after every inclusion or exclusion of a predictor variable in a regression model. For cases where we have a large pool of potential predictor variables (that may not all be important for the response) this approach is unpractical. Finding a small subset of predictors that collectively model the response well is called the VARIABLE SELECTION PROBLEM.

There are advantages and disadvantages to including more predictors in regression.

PRO: More predictors will always be able to explain the response better (smaller SS_{Res}, larger R^2).

CON: A model with fewer predictors is simpler and may be easier to explain. If more predictors are included, the variance of $\hat{y}$ for prediction becomes larger and prediction confidence intervals become less precise. Problems with multicollinearity may be increased if there are more predictors. In the "real" world, collecting data on additional predictor variables costs money.

As you have already seen, it is difficult to find "the best" regression equation for any given set of data and a set of predictors. The models may differ with respect to their explanatory power (R^2), with respect to fulfilling residual assumptions, and with respect to being easy to explain.

Thus, the discussion of variable selection is a little bit abstract. We're assuming for instance, that we already know in which form (untransformed, or transformed in a specific way) each predictor should be included in the model. In practice we would, of course, not know how to transform the predictors before including them in the model. Furthermore, transformations of one predictor may depend on the presence or absence of other predictors in the model.

In reality, variable selection methods are used only as *one tool* in the arsenal to find the optimal model. All other methods we have previously discussed such as variable transformations, identification of leverage and influential points, residual analysis etc. should still be employed simultaneously. You should not exclusively rely on the automated variable selection methods we will discuss, nor expect that they will automatically provide you with the best model for your data.

Consequences of Model Misspecification

The parameter estimates in a multiple regression model are only unbiased if the model is correctly specified. That means that all the predictors that actually have an influence on the response are accounted for in the model. That's very rarely the case in practice. What happens to the parameter estimates (and other inference procedures) if this is not the case?

In general: suppose that there are K candidate regressors that may have an influence on the response. Suppose further that $n > K + 1$ observations have been collected on all regressors and on the response. Then the full model is

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \cdots + \beta_K \mathbf{x}_K + \boldsymbol{\epsilon}$$

or

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

We assume that the list of candidate predictors $\mathbf{x}_1, \ldots, \mathbf{x}_K$ contains all predictors that actually have a true influence on the response. Now suppose, that we decide to exclude r of the predictor variables from the model. Partition the $\mathbf{X}$ matrix and the $\boldsymbol{\beta}$ vector, so that $\mathbf{X}_p$ contains the intercept column of ones and the p predictors to be retained, while $\mathbf{X}_r$ contains the r predictors to be excluded from the model.

$$\mathbf{y} = \mathbf{X}_p \boldsymbol{\beta}_p + \mathbf{X}_r \boldsymbol{\beta}_r + \boldsymbol{\epsilon}$$

For the full model (with all K predictors) we know that the parameter estimate

$$\hat{\boldsymbol{\beta}}^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

is unbiased and that the residual variance estimate is

$$\hat{\sigma}^2 = \mathrm{MS}_{Res} = \frac{\mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}^{*\prime}\mathbf{X}'\mathbf{y}}{n - K - 1} = \frac{\mathbf{y}'[\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}']\mathbf{y}}{n - K - 1}$$

What can we say about the parameter estimate of $\boldsymbol{\beta}_p$ in the reduced model, in which r predictor variables have been excluded?

$\hat{\boldsymbol{\beta}}_p$ is biased. The expected value of $\hat{\boldsymbol{\beta}}_p$ is

$$E(\hat{\boldsymbol{\beta}}_p) = \boldsymbol{\beta}_p + (\mathbf{X}_p'\mathbf{X}_p)^{-1}\mathbf{X}_p'\mathbf{X}_r\boldsymbol{\beta}_r$$

The variance of $\hat{\boldsymbol{\beta}}_p$ is $\sigma^2(\mathbf{X}_p'\mathbf{X}_p)^{-1}$. The variances of the least-squares parameter estimates in the full model are greater than or equal to the variances of the parameter estimates in the reduced model. This means that deleting variables from a model never increases the variances of the estimates of the remaining parameters.

It does not make much sense to compare the variance of the parameter estimates in the full and the reduced model, because the variance in the full model is always bigger and since $\hat{\boldsymbol{\beta}}_p$ is biased, we are not really (that much) interested in its variance. Instead, one often compares the mean squared errors of the two estimates which is a combination of the variance and the bias. Mean squared error is defined as

$$\text{MSE}(\hat{\theta}) = Var(\hat{\theta}) + [E(\hat{\theta}) - \theta]^2$$

The mean square error of the parameter estimates in the reduced model is smaller than the corresponding parameter estimates in the full model if the regression coefficients of the deleted variables are smaller than the standard errors of those estimates in the full model. That means that as long the estimated coefficients are smaller than their standard errors, we can exclude those variables from the model without raising the mean square error of the remaining variables.

The mean square error estimate of $\hat{\sigma}^2$ in the full model is an unbiased estimate of the residual variance. This is not true for the MSE of $\hat{\sigma}^2$ in reduced model, which is generally biased upward (larger than the true residual variance).

In Summary: Deleting variables from a regression model can improve the precision of the parameter estimates for the remaining variables. This can be true even if the slopes of the deleted variables are significant. In general, deleting variables will introduce bias in estimation and prediction. If the deleted variables have small effects (and introduce only a little bias), then the mean squared error of the biased estimates can be less than the variance of the unbiased estimates. In other words, the amount of bias introduced may be less than the reduction in variance.

If non-significant regressors or regressors for which the parameter estimate is less than the corresponding standard error are retained in a regression model, than the variances of the predicted estimates and of predicted response values may be unnecessarily increased.

If there is any information available in the context of the problem about the predictor variables that tells us which variables should theoretically be important for the response, then this information supersedes any "automated" variable selection criteria.

Criteria for Evaluating Subset Regression Models

When we have to decide which available predictor variables to keep in the reduced model and which ones to exclude there are often very many possible "candidate" subsets of predictors. How should the resulting models be compared? Several criteria are possible:

- R^2 or adjusted R^2
- The model MS_{Res}
- Mallows's C_p statistic (measures the MSE of fitted values)

In general, these criteria do not only depend on *which* predictors are retained but also on *how many* predictors are retained in the reduced model. This makes it a little trickier to compare them across different candidate models.

R^2 - COEFFICIENT OF MULTIPLE DETERMINATION

Recall that R^2 for multiple regression models is defined as

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_{Res}}{SS_T}$$

where SS_R and SS_{Res} denote the regression and residual sum of squares for the candidate model, respectively. The SS_R will increase with the number of parameters in the model and the SS_{Res} will therefore decrease. We can directly compare R^2 for different candidate models with the same number of predictors, but it would not be fair to use R^2 to compare models of different size.

In practice, one often proceeds as follows: Fit all possible models with a subset of k predictor variables (for $k = 1, \ldots, K$). For each value of k, find the subset of predictors that maximizes R^2. Plot these maximized R^2-values against k and try to find the "bend" in the curve. This will indicate a good number of predictors to use and the optimal subset of predictors of that size.

ADJUSTED R^2

To be able to compare models of different size, some analysts prefer to use the adjusted R^2, which takes the number of predictors in the model in account. Recall, that the adjusted R^2 for a model with $k+1$ parameters (k predictors) is defined as

$$R^2_{\text{Adj}} = 1 - \left(\frac{n-1}{n-k-1}\right)(1 - R^2)$$

Maximizing the adjusted R^2 is equivalent to minimizing the MS_{Res}.

Residual Mean Square

The residual mean square of the subset models

$$\mathrm{MS}_{Res} = \frac{SS_{Res}}{n-k-1}$$

can be used as another model evaluation criterion. As already noted, SS_{Res} decreases as the number k of predictors in the model increases. But this decrease may not be linear for all values of k. Typically, in the lower ranges of k, the SS_{Res} decreases faster than $n-k-1$, so that overall the MS_{Res} decreases. But for larger values of k, the decrease in the residual sum of square is not as large anymore, so that the mean square residual actually increases again. The subset regression model that minimizes MS_{Res} simultaneously maximizes R^2_{Adj}.

Mallows's C_p Statistic

This criterion is based on the mean square error of a fitted value:

$$E\left[(\hat{y}_i - E(y_i))^2\right] = [E(y_i) - E(\hat{y}_i)]^2 + Var(\hat{y}_i)$$

Here, $E(y_i)$ is the point on the true regression equation corresponding to the true full model and $E(\hat{y}_i)$ is the expected mean response (for infinitely large samples) for the $p = (k+1)$-term subset model. The bias terms for the n fitted values are combined into

$$SS_B(p) = \sum_{i=1}^{n} [E(y_i) - E(\hat{y}_i)]^2$$

which allows us to define the standardized total mean square error as

$$\Gamma_p = \frac{1}{\sigma^2}\left\{\sum_{i=1}^{n}[E(y_i) - E(\hat{y}_i)]^2 + \sum_{i=1}^{n} Var(\hat{y}_i)\right\} = \frac{SS_B(p)}{\sigma^2} + \frac{1}{\sigma^2}\sum_{i=1}^{n} Var(\hat{y}_i)$$

It can be shown that

$$\sum_{i=1}^{n} Var(\hat{y}_i) = p\sigma^2, \text{ and } E[SS_{Res}(p)] = SS_B(p) + (n-p)\sigma^2$$

Thus,

$$C_p = \frac{SS_{Res}(p)}{\hat{\sigma}^2} - n + 2p$$

is an estimate of the standardized total mean square error Γ_p of the p-term model. If the bias in the p-term model is negligible, then

$$E[C_p|\text{no bias}] = p$$

Thus, the C_p statistic is often plotted against p and compared to the 45° line ($y = p$). Models with small C_p values are desirable. Note, that we need a reliable estimate of σ^2 to compute the C_p statistic. Often, the MS_{Res} of the full model is used for this purpose.

Using Model Evaluation Criteria

You have now seen several potential model evaluation criteria. Which one should you use in practice? That depends a little on what the purpose of your regression model is. Regression models can be used to (1) describe data, (2) predict new observation values, (3) estimate parameters, and (4) control processes.

If the objective is to describe the data, then making the SS_{Res} small is a good goal. Here, you have to weigh removing non-significant and/or irrelevant predictor variables against small increases in SS_{Res}.

If the main goal of the model is prediction, then any statistic that minimizes the mean square error for prediction is a good choice. This could, for instance, be the PRESS statistic that describes the difference between observed values and predicted values based on leave-one-out regression.

If the purpose of the regression model is parameter estimation, maybe to understand the conceptual relationship between the response and a group of predictors, then clearly minimizing the variance of the estimated coefficients (while keeping the estimation bias reasonable) would be a good goal.

Note: The model selection criteria that we discussed above are available in R in the `leaps` package. Download and install the package using the `Package Installer` in R. There are two functions that can effectively be used to compare many models at once. They are called `leaps()` and `regsubsets()`. They differ in the amount of information they return (`regsubsets` produces more details). Given a data set with several predictor variables, both functions will compute the criteria (R^2, adjusted R^2, C_p) for you for any number of subset candidate models.

Computational Techniques for Variable Selection

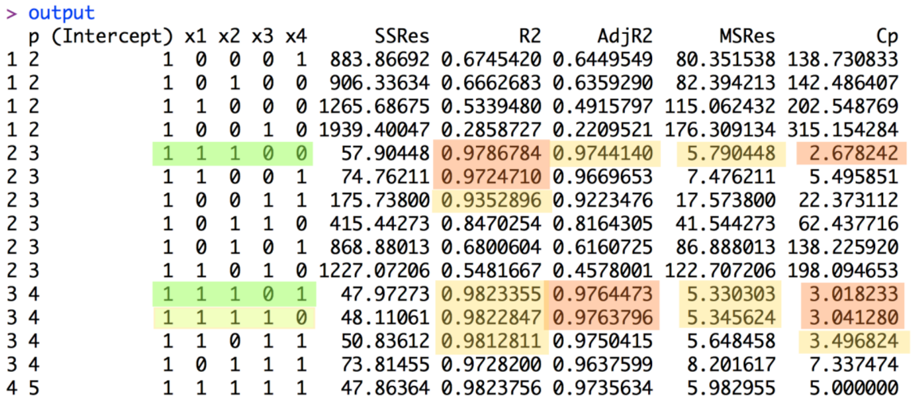
All the different criteria that you have seen so far require a good bit of work to use in an actual application. You would have to fit the full model and all possible subset candidate models, compute the criteria for each model and compare. Of course, you will also need to keep an eye on the residuals of each model at the same time. Even with the help of a function such as `leaps()`, this will still require a good bit of work and final interpretation by you (the user). That sounds almost infeasible in practice. Wouldn't it be nice, if there were a button one could push, that would take all the effort of finding the “best” subset model out of your hands and spit out an answer? There is... but don't get too excited about it.

Example: The Hald Cement Data

Cement is mixed from four ingredients: tricalcium aluminate (x_1), tricalcium silicate (x_2), tetracalcium alumino ferrite (x_3), and dicalcium silicate (x_4). To make concrete, these chemicals are mixed with aggregates (coarse gravel or sand). If the dry mix is hydrated with water, a chemical reaction takes place that makes the concrete harden. This chemical reaction produces heat. The proportion in which the above chemicals are used determines how much heat is generated. The heat (y) is measured in calories per gram of cement. The data are available in the file "Cement.txt". They contain 13 observations on different mixing proportions.

We want to investigate the effects that the four possible predictor variables $x_1, \ldots, x_4$ have on the response y. If we wanted to fit all possible candidate models, how many models would we have to fit and investigate?

As we have previously discussed, possible criteria by which to compare models are R^2, the adjusted R^2, the MS$_{Res}$, and C_p.



```
> output
  p (Intercept) x1 x2 x3 x4      SSRes        R2     AdjR2      MSRes         Cp
1 2           1  0  0  0  1  883.86692 0.6745420 0.6449549  80.351538 138.730833
1 2           1  0  1  0  0  906.33634 0.6662683 0.6359290  82.394213 142.486407
1 2           1  1  0  0  0 1265.68675 0.5339480 0.4915797 115.062432 202.548769
1 2           1  0  0  1  0 1939.40047 0.2858727 0.2209521 176.309134 315.154284
2 3           1  1  1  0  0   57.90448 0.9786784 0.9744140   5.790448   2.678242
2 3           1  1  0  0  1   74.76211 0.9724710 0.9669653   7.476211   5.495851
2 3           1  0  0  1  1  175.73800 0.9352896 0.9223476  17.573800  22.373112
2 3           1  0  1  1  0  415.44273 0.8470254 0.8164305  41.544273  62.437716
2 3           1  0  1  0  1  868.88013 0.6800604 0.6160725  86.888013 138.225920
2 3           1  1  0  1  0 1227.07206 0.5481667 0.4578001 122.707206 198.094653
3 4           1  1  1  0  1   47.97273 0.9823355 0.9764473   5.330303   3.018233
3 4           1  1  1  1  0   48.11061 0.9822847 0.9763796   5.345624   3.041280
3 4           1  1  0  1  1   50.83612 0.9812811 0.9750415   5.648458   3.496824
3 4           1  0  1  1  1   73.81455 0.9728200 0.9637599   8.201617   7.337474
4 5           1  1  1  1  1   47.86364 0.9823756 0.9735634   5.982955   5.000000
```

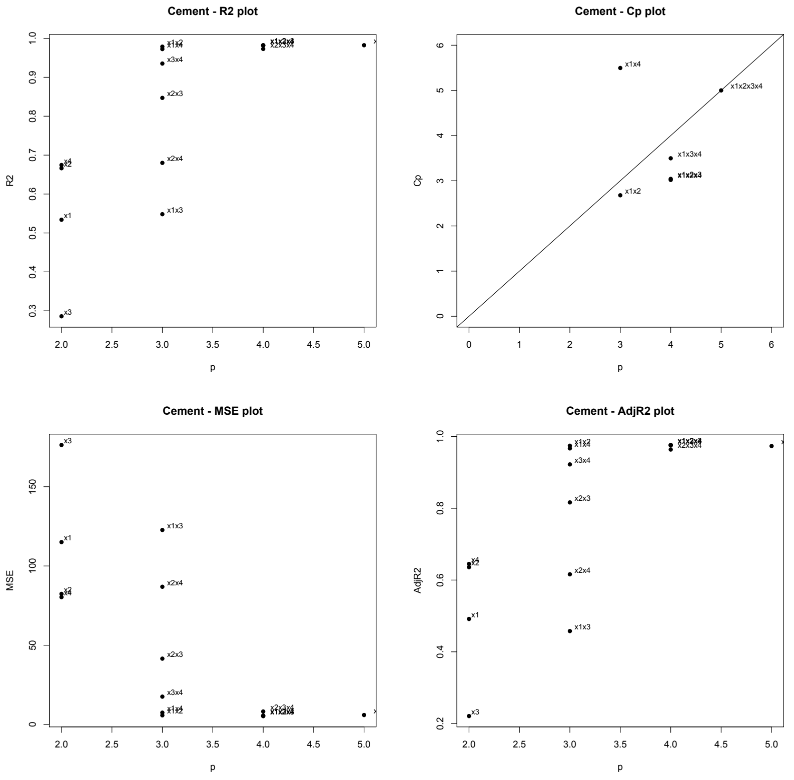
The first two columns describe how many predictors (or parameters p) are included in the model. Columns 3-7 describe *which* predictors are in the considered subset model. According to this table, which subset model is "best"? Start by considering the R^2-criterion: There is a big improvement in using two predictors over using just one. For two predictor subset models the models that use (x_1, x_2) and those that use (x_1, x_4) have very similar R^2-values (about 97% variation explained). The models with three of even four predictors do not explain much more (about 98%).

The model with the smallest MS$_{Res}$ is the model with predictors (x_1, x_2, x_4) with MS$_{Res}$ = 5.33. This same model also maximizes the adjusted R^2. The models (x_1, x_2, x_3), (x_1, x_3, x_4) and the two-regressor model (x_1, x_2) also have comparatively small MS$_{Res}$ (and comparatively large adjusted R^2) values. The MS$_{Res}$ of the (x_1, x_4) model is a little bit larger.

The C_p criterion identifies four acceptable candidate models: (x_1, x_2) has the smallest C_p value, closely followed by the (x_1, x_2, x_4), (x_1, x_2, x_3) and (x_1, x_3, x_4) models.

If we had to make a decision without any additional information about the predictor variables, the simplest model (x_1, x_2) seems like a good choice.

The R^2-values (or any of the other model evaluating criteria) can be graphed as a function of the number of parameters in the model. This allows for a fast overview of how much including additional parameters would improve the criterion.



We can also evaluate the estimated coefficients in the candidate models (here in pretty table form). The R-output from the `coef` - command is not as nicely organized.

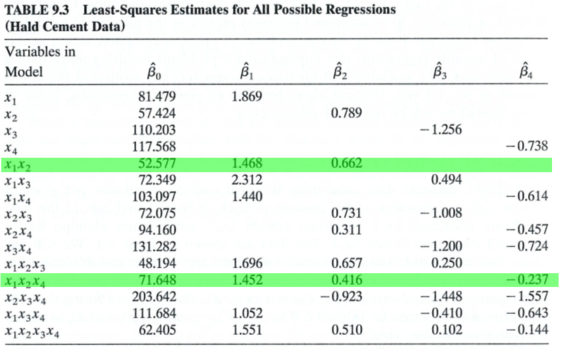
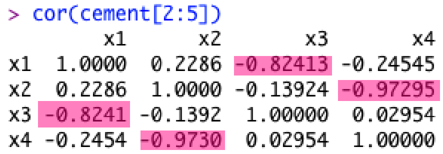


TABLE 9.3 Least-Squares Estimates for All Possible Regressions (Hald Cement Data)

Variables in Model	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$
x_1	81.479	1.869			
x_2	57.424		0.789		
x_3	110.203			−1.256	
x_4	117.568				−0.738
x_1x_2	52.577	1.468	0.662		
x_1x_3	72.349	2.312		0.494	
x_1x_4	103.097	1.440			−0.614
x_2x_3	72.075		0.731	−1.008	
x_2x_4	94.160		0.311		−0.457
x_3x_4	131.282			−1.200	−0.724
$x_1x_2x_3$	48.194	1.696	0.657	0.250	
$x_1x_2x_4$	71.648	1.452	0.416		−0.237
$x_2x_3x_4$	203.642		−0.923	−1.448	−1.557
$x_1x_3x_4$	111.684	1.052		−0.410	−0.643
$x_1x_2x_3x_4$	62.405	1.551	0.510	0.102	−0.144

How do you interpret the fact, that the signs for some of the coefficients change, depending on which model they are included in?

Note, that the different criteria can point to different subset models as the "best" choice. In practice, it might be a good idea to use the criteria to collect a small number of "finalists" which then should be subjected to a more thorough analysis to make a decision. For this particular dataset, the correlation between some pairs of predictors is rather high.



```
> cor(cement[2:5])
        x1      x2       x3       x4
x1  1.0000  0.2286 -0.82413 -0.24545
x2  0.2286  1.0000 -0.13924 -0.97295
x3 -0.8241 -0.1392  1.00000  0.02954
x4 -0.2454 -0.9730  0.02954  1.00000
```

Thus, models that simultaneously use predictors x_1 and x_3 (correlation -0.82), or models that simultaneously use predictors x_2 and x_4 (correlation -0.97) should be avoided to reduce multicollinearity between the predictors.

Stepwise Model Selection Methods

As you have just seen, an exhaustive look at all possible candidate models can be a lot of work, especially if the number of available predictor variables is large. Maybe we don't really need to consider *all* possible subset models. In our exhaustive search, we have rather quickly zeroed in on one (or sometimes two) candidate models with the same number of predictors. There are a number of different methods that add or remove predictors from a model one at a time and thus consider a much smaller number of candidate models. The methods are referred to as STEPWISE-TYPE PROCEDURES. They can be classified into three major categories: FORWARD SELECTION, BACKWARD SELECTION, and STEPWISE SELECTION which is effectively a combination of the other two methods.

FORWARD VARIABLE SELECTION: The idea here is to begin with the model that includes just the intercept and to include variables one-by-one. Variables that are already in the model will remain in the model. The predictors to be included next in the model are chosen to maximize the correlation with the response, given the predictors that already are in the model. This is the same as maximizing the partial F-statistic

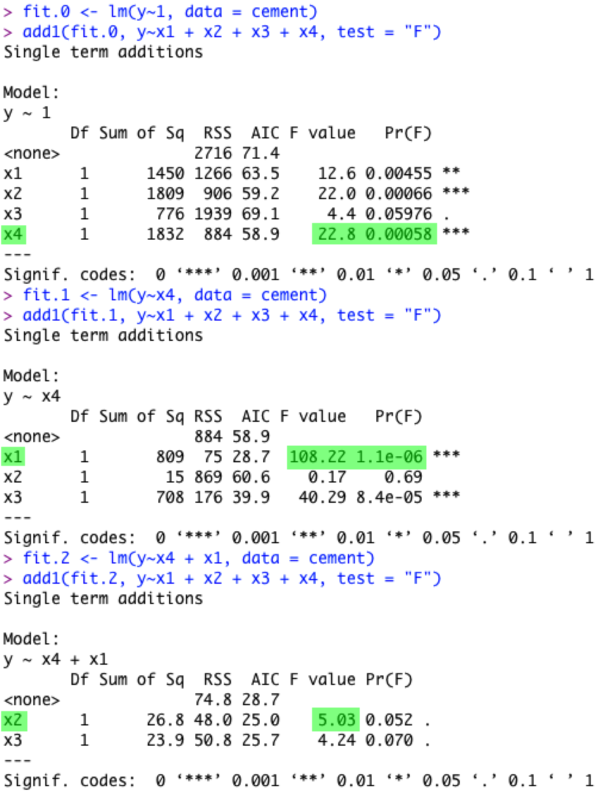
$$F = \frac{SS_R(x_{new}|x_{old})}{MS_{Res}(x_{old}, x_{new})}$$

If this partial F-statistic value exceeds a threshold value the new predictor is included (together with the old predictors already in the model). The procedure terminates, if there are no additional predictors whose inclusion in the model would raise the partial F-statistic over the threshold. Often, software produces p-values for the partial F-statistics to make threshold selection more straightforward.

Example: The Hald Cement Data

Forward variable selection can be done "by-hand" in R with the `add1()` command. This command starts with a simple model (usually $y \sim 1$, the model with just the intercept) and selects variables one by one, that will maximize the partial F-statistic. It will also report the partial F-statistic together with a p-value. The p-value tells us whether the inclusion of an additional predictor in the model is "worth" raising the number of parameters in the model by one.

Shown on the right is the R-output for three single variable additions to the model that initially contains only the intercept. What are the variables that are being added (and in which order are they added)? What is the final model chosen by this algorithm? Use cutoff $\alpha = 0.1$.



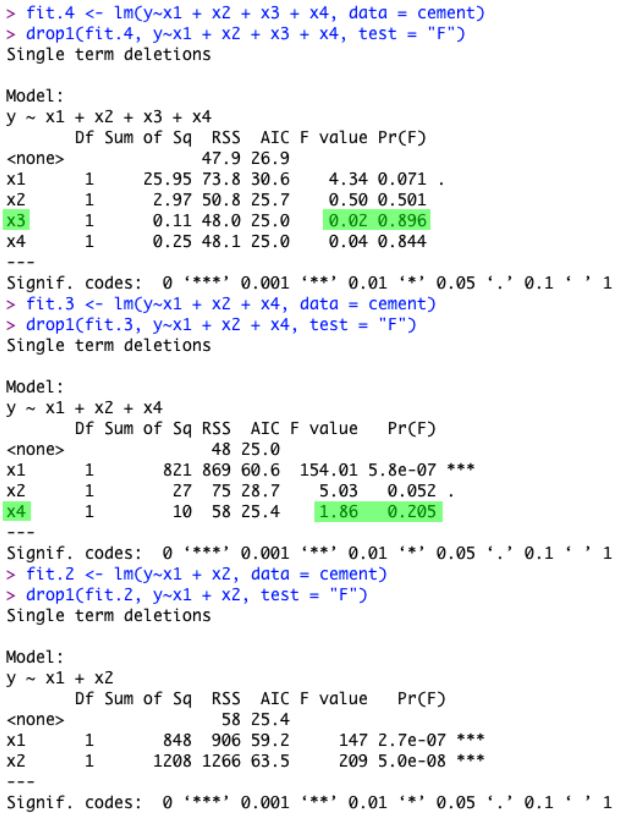
```
> fit.0 <- lm(y~1, data = cement)
> add1(fit.0, y~x1 + x2 + x3 + x4, test = "F")
Single term additions

Model:
y ~ 1
       Df Sum of Sq  RSS  AIC F value   Pr(F)
<none>               2716 71.4
x1      1      1450 1266 63.5    12.6 0.00455 **
x2      1      1809  906 59.2    22.0 0.00066 ***
x3      1       776 1939 69.1     4.4 0.05976 .
x4      1      1832  884 58.9    22.8 0.00058 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.1 <- lm(y~x4, data = cement)
> add1(fit.1, y~x1 + x2 + x3 + x4, test = "F")
Single term additions

Model:
y ~ x4
       Df Sum of Sq RSS  AIC F value   Pr(F)
<none>              884 58.9
x1      1       809  75 28.7  108.22 1.1e-06 ***
x2      1        15 869 60.6    0.17    0.69
x3      1       708 176 39.9   40.29 8.4e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.2 <- lm(y~x4 + x1, data = cement)
> add1(fit.2, y~x1 + x2 + x3 + x4, test = "F")
Single term additions

Model:
y ~ x4 + x1
       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>               74.8 28.7
x2      1      26.8 48.0 25.0    5.03 0.052 .
x3      1      23.9 50.8 25.7    4.24 0.070 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

BACKWARD VARIABLE SELECTION: Recall, that the forward algorithm starts with no predictors in the model and includes predictors one-by-one. The Backward algorithm does the exact opposite. It starts with all possible predictors in the model and excludes predictors one by one. The algorithm again uses the partial F-statistics to decide which variables to exclude next. The partial F-test statistics for each predictor are computed as if this predictor were the *last* predictor to be added to the model, given that all other variables are already in the model. That means that the variables with the *smallest* partial F-statistics and thus the largest p-values are the ones that should be excluded *first*.

Example: The Hald Cement Data

Backward variable selection can be done in R with the `drop1()` command. It works very similarly to the `add1()` command. This time, we're looking for the smallest partial F-statistic and we consider excluding variables, as long as the p-value for the partial F-statistic is larger than α (e.g., 0.1).

Shown on the right is the R-output for three single variable deletions in the model that initially contains all four predictors. What are the variables that are being deleted (and in which order are they deleted)? What is the final model chosen by this algorithm? Use cutoff $\alpha = 0.1$.



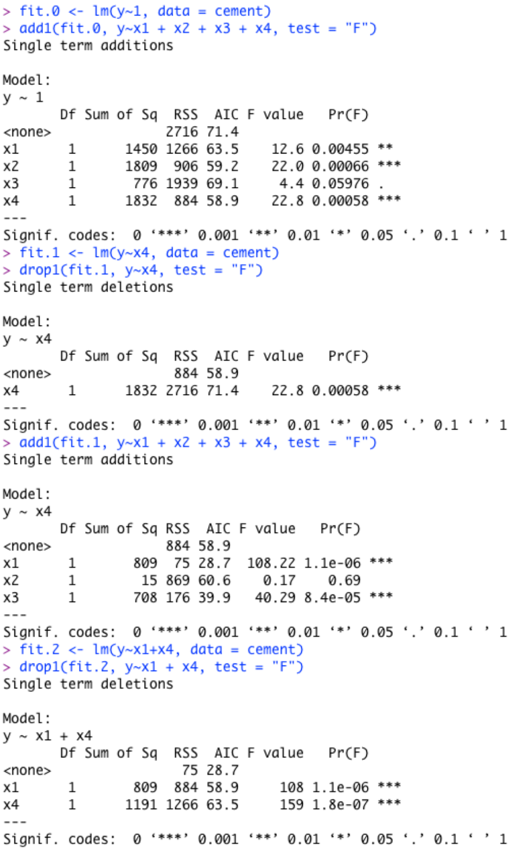
```
> fit.4 <- lm(y~x1 + x2 + x3 + x4, data = cement)
> drop1(fit.4, y~x1 + x2 + x3 + x4, test = "F")
Single term deletions

Model:
y ~ x1 + x2 + x3 + x4
       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>              47.9 26.9
x1      1     25.95 73.8 30.6    4.34 0.071 .
x2      1      2.97 50.8 25.7    0.50 0.501
x3      1      0.11 48.0 25.0    0.02 0.896
x4      1      0.25 48.1 25.0    0.04 0.844
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.3 <- lm(y~x1 + x2 + x4, data = cement)
> drop1(fit.3, y~x1 + x2 + x4, test = "F")
Single term deletions

Model:
y ~ x1 + x2 + x4
       Df Sum of Sq RSS  AIC F value   Pr(F)
<none>               48 25.0
x1      1       821 869 60.6  154.01 5.8e-07 ***
x2      1        27  75 28.7    5.03   0.052 .
x4      1        10  58 25.4    1.86   0.205
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.2 <- lm(y~x1 + x2, data = cement)
> drop1(fit.2, y~x1 + x2, test = "F")
Single term deletions

Model:
y ~ x1 + x2
       Df Sum of Sq  RSS  AIC F value   Pr(F)
<none>                58 25.4
x1      1       848  906 59.2     147 2.7e-07 ***
x2      1      1208 1266 63.5     209 5.0e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Note: The final model(s) chosen by the forward and backward variable selection algorithms do not always agree (as you've just seen in the Cement Data example). Here, the backward algorithm has chosen the model that we would have also preferred based on an exhaustive search of all possible candidate models. That is not guaranteed to always be the case, either.

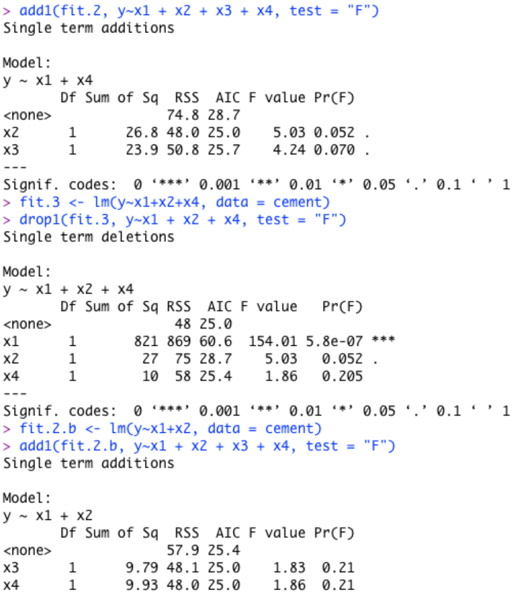
Stepwise Variable Selection: The forward and backward procedures can be combined in a number of ways to consider more candidate models. For instance, even though x_4 was chosen as the single strongest predictor for the response in the first step of the forward algorithm, after the inclusion of another variable (here x_2), x_4 could have become redundant and the stepwise algorithm would be able to exclude those variables again.

There are several different stepwise procedures. The stepwise algorithm suggested by Efroymson (1960) starts with a forward selection. After each inclusion of a new predictor, a backward (`drop1`) step is performed to allow variables to be dropped. This algorithm requires two cutoff values for the F-statistics (or their p-values). The α-cutoff for dropping a variable from the model is often chosen to be higher than the cutoff for including a variable to avoid (infinitely) long inclusion/exclusion loops.

Shown below is the R-output for a stepwise variable selection for the Cement Data example. Here both cutoffs are chosen to be 0.1. What are the variables that are added (or deleted) from the model, and in which order? What is the final model according to this method?



```
> fit.0 <- lm(y~1, data = cement)
> add1(fit.0, y~x1 + x2 + x3 + x4, test = "F")
Single term additions

Model:
y ~ 1
       Df Sum of Sq  RSS  AIC F value   Pr(F)
<none>              2716 71.4
x1      1      1450 1266 63.5    12.6 0.00455 **
x2      1      1809  906 59.2    22.0 0.00066 ***
x3      1       776 1939 69.1     4.4 0.05976 .
x4      1      1832  884 58.9    22.8 0.00058 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.1 <- lm(y~x4, data = cement)
> drop1(fit.1, y~x4, test = "F")
Single term deletions

Model:
y ~ x4
       Df Sum of Sq  RSS  AIC F value   Pr(F)
<none>               884 58.9
x4      1      1832 2716 71.4    22.8 0.00058 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> add1(fit.1, y~x1 + x2 + x3 + x4, test = "F")
Single term additions

Model:
y ~ x4
       Df Sum of Sq RSS  AIC F value   Pr(F)
<none>              884 58.9
x1      1       809  75 28.7  108.22 1.1e-06 ***
x2      1        15 869 60.6    0.17    0.69
x3      1       708 176 39.9   40.29 8.4e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.2 <- lm(y~x1+x4, data = cement)
> drop1(fit.2, y~x1 + x4, test = "F")
Single term deletions

Model:
y ~ x1 + x4
       Df Sum of Sq  RSS  AIC F value   Pr(F)
<none>                75 28.7
x1      1       809  884 58.9     108 1.1e-06 ***
x4      1      1191 1266 63.5     159 1.8e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



```
> add1(fit.2, y~x1 + x2 + x3 + x4, test = "F")
Single term additions

Model:
y ~ x1 + x4
       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>              74.8 28.7
x2      1      26.8 48.0 25.0    5.03 0.052 .
x3      1      23.9 50.8 25.7    4.24 0.070 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.3 <- lm(y~x1+x2+x4, data = cement)
> drop1(fit.3, y~x1 + x2 + x4, test = "F")
Single term deletions

Model:
y ~ x1 + x2 + x4
       Df Sum of Sq RSS  AIC F value   Pr(F)
<none>               48 25.0
x1      1       821 869 60.6  154.01 5.8e-07 ***
x2      1        27  75 28.7    5.03   0.052 .
x4      1        10  58 25.4    1.86   0.205
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> fit.2.b <- lm(y~x1+x2, data = cement)
> add1(fit.2.b, y~x1 + x2 + x3 + x4, test = "F")
Single term additions

Model:
y ~ x1 + x2
       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>              57.9 25.4
x3      1      9.79 48.1 25.0    1.83  0.21
x4      1      9.93 48.0 25.0    1.86  0.21
```

Summary

These automated variable selection processes are tempting, because they require no knowledge about the actual predictor variables. All that we have to be able to do is to compare p-values to thresholds. But the methods have some serious drawbacks. They do not always have the same answer. Thus, if you employ an automated variable selection method, you should use all three different methods. If the results do not agree, you have to investigate further (by hand...). Further, there is no guarantee, that any one of the three automated methods will identify the model that would have been identified in an exhaustive search based on the previously discussed model selection criteria.

In general, the results of the forward selection algorithm tend to agree with the results of an exhaustive search for small subset sizes and the results of the backward algorithm tend to agree with the results on an exhaustive search for large subset sizes.

Obviously, the choice of threshold for inclusion and exclusion of parameters from the model has to be made by the user and the sequence of candidate models as well as the final model can depend on this choice. The threshold α that is used for variable selection procedures should not be interpreted as a significance level (in the sense of hypothesis testing). If relatively large α-threshold values are used, then predictors that might not (at first glance) have appeared important are at least investigated in terms of their effects on the model. In some software programs, automated variable selection methods have default thresholds of $\alpha = 0.05$ for forward selection and $\alpha = 0.1$ for backward selection. Other programs use thresholds as large as 0.25.

General Strategy

In this section we have made the unrealistic assumption, that none of the predictor variables or the response need to be transformed, and that the residual assumptions are equally satisfied for all possible candidate models. In reality, that is of course regularly not the case. In which order should one then proceed?

1. Fit the largest possible model to the data (using all available untransformed predictor variables)

2. Perform a thorough (residual) analysis of this model. Identify outliers or influential observations.

3. Determine if a model transformation is necessary and transform variables as appropriate.

4. Determine if it is feasible to exhaustively consider all possible subset models

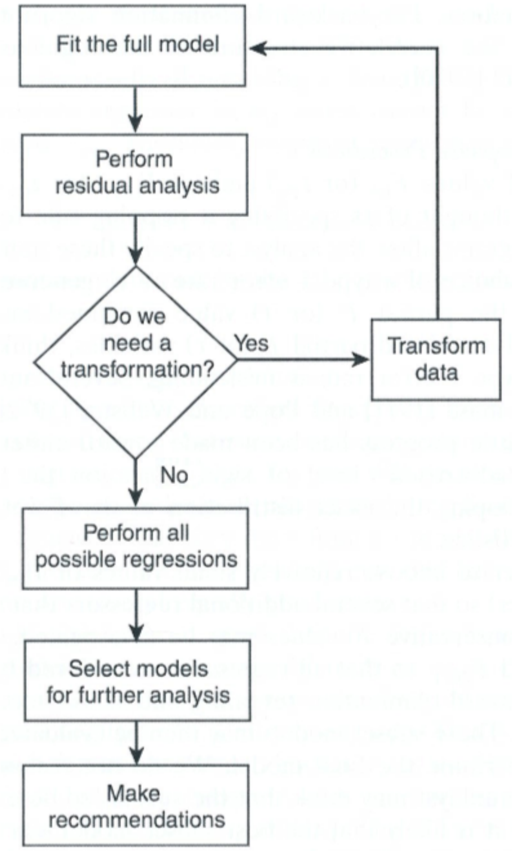
   (a) If it is feasible, fit all possible subset regression models and compare them using criteria such as R^2, adjusted R^2, the MS_{Res}, and Mallowes's C_p. Rank the best subset models.

(b) If exhaustive search is not feasible, employ one (or several) of the stepwise regression techniques to find the largest subset model for which exhaustive subset search becomes feasible. Then study the subset models under the criteria described above.

5. Compare and contrast the best models (residuals, interpretation, importance of predictors in the context etc.) suggested by each criterion.

6. Explore the need for further transformations

7. Present several of your final candidate models to the subject-matter experts (the people who asked you to analyze their data) and discuss the relative merits and weaknesses of each model with them.


Fit the full model
Perform residual analysis
Do we need a transformation?
Yes
Transform data
No
Perform all possible regressions
Select models for further analysis
Make recommendations