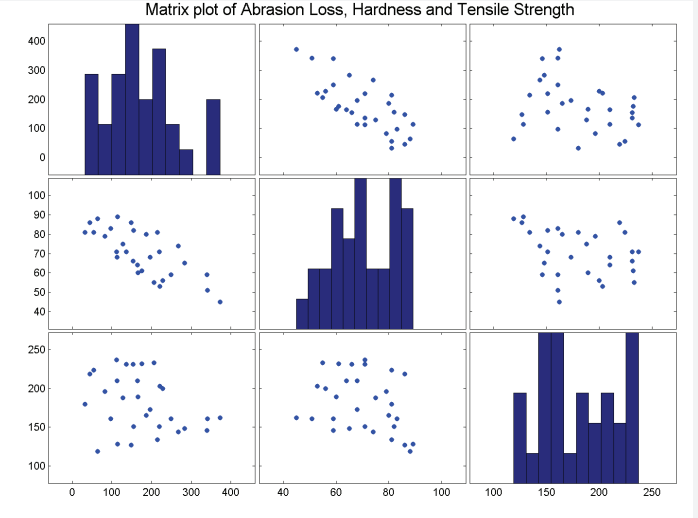
## Example: Abrasion Resistance Matrix Plot Code

- Code assumes data read in one matrix called `data`
- One column per variable (response first)
- Matrix is simple to produce in MATLAB:

```
>> % Assumes all data is read in as a matrix
>> plotmatrix(data)
>> title('Matrix plot of Abrasion Loss, Hardness and
   Tensile Strength')
```

- Only issue: Painful to put axis labels on the plot
- Workaround: put them in title (in right order)

## Example: Abrasion Resistance Matrix Plot




## Example: Abrasion Resistance Matrix Plot Features

- Upper and lower triangles are mirror image of each other
- If response is first, then first row gives first indication of association between response and explanatory variables
- Other plots in upper triangle summarise associations between the explanatory variables
- Histogram on diagonal useful to understand spread of data
- Impression for this dataset:
  - moderate-strong negative linear association between abrasion losses and hardness
  - weak negative association between abrasion losses and tensile strength
  - weak negative association between hardness and tensile strength

## Matrix Plot General Comments

- Matrix plots are excellent for:
  - examining relationships between pairs of variables;
  - detecting obvious outliers in one or two variables; and
  - displaying a large number of variables.
- However, it can be difficult to extract higher order relationships (interactions) between several variables and outlier in more than two variables.
- Other possibilities: `glyphplot`, `andrewsplot` or `parallelcoords`

## Example: Abrasion Resistance Correlation and Regression

- Often worthwhile getting correlation matrix:

```
>> corr(data)
```

- which gives:

```
ans =

    1.0000   -0.7377   -0.2984
   -0.7377    1.0000   -0.2992
   -0.2984   -0.2992    1.0000
```

- Fitting multiple regression model :

```
>> % extract response vector
>> y=data(:,1);
>> % create design matrix
>> X=[ones(size(data,1),1) data(:,2:end)];
>> % fit regression model
>> [B,BINT,R,RINT,STATS] = regress(y,X);
```

## Example: Abrasion Resistance Regression Results

- **Multiple coefficient of determination** $R^2$ (proportion of variation in abrasion loss explained by model) obtained from:

```
>> STATS(1)
```

- which gives $R^2 = 84\%$
- Regression coefficient vector (`B` vector) is:

```
B =

  885.1611
   -6.5708
   -1.3743
```

## Example: Abrasion Resistance Regression Coefficients

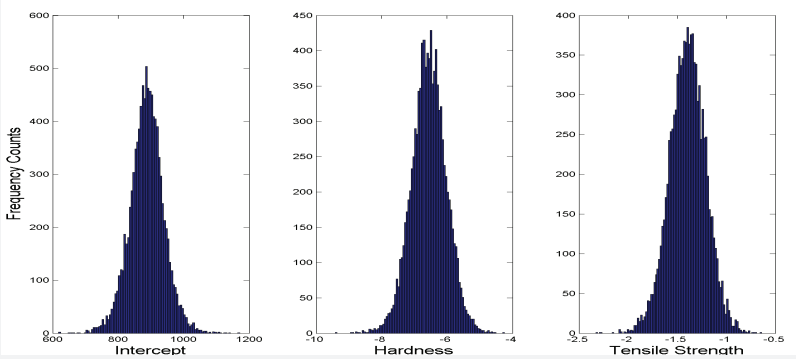
- 95% confidence interval for regression coefficients assuming normally distributed errors given by `BINT` matrix:

```
BINT =

  1.0e+003 *

    0.7585    1.0119
   -0.0078   -0.0054
   -0.0018   -0.0010
```

- Better to use bootstrapping (observation resampling appropriate here):

```
>> % observation resampling (random X)
>> bootbetas = bootstrp(nsim,@(y,x) regress(y,x),y,X);
>> prctile(bootbetas,[2.5 97.5],1)
>> figure;
>> subplot(1,3,1);hist(bootbetas(:,1),100);xlabel('Intercept')
>> subplot(1,3,2);hist(bootbetas(:,2),100);xlabel('Hardness')
>> subplot(1,3,3);hist(bootbetas(:,3),100);xlabel('Tensile Strength')
```

## Example: Abrasion Resistance Bootstrap Coefficients




- 95% confidence intervals for regression coefficients using bootstrapping (does not require normal errors assumptions)
- Given by each column of `prctile(bootbetas,[2.5 97.5],1)`:

```
ans =

  780.9881   -7.6261   -1.7635
  992.9718   -5.4824   -1.0359
```

## Hypothesis Testing of Regression Coefficients 1

| "state of nature" | Don't Reject $H_0$ | Reject $H_0$ |
|---|---|---|
| $H_0 : \beta_i = 0$ is "true" | OK | Type I error |
| $H_1 : \beta_i \neq 0$ is "true" | Type II error | OK |

- We want to reject $H_0$ when it is true with a small probability
- That is, we want to keep the probability of Type I error $\leq \alpha = 0.05$, say
- Similarly, we want to minimize type II error as well
- There are two ways to do a hypothesis test
- 1. Reject $H_0$ if 0 is not inside the $(1 - \alpha)$ confidence interval for $\beta_i$
- 2. Compute $p$-value – a measure of evidence against $H_0$

| "$p$-value" range | evidence |
|---|---|
| $< 0.01$ | very strong evidence against $H_0$ |
| $[0.01, 0.05]$ | strong evidence against $H_0$ |
| $[0.05, 0.10]$ | weak evidence against $H_0$ |
| $> 0.1$ | little or no evidence against $H_0$ |

## Hypothesis Testing of Regression Coefficients 2

- Suppose we want to carry out hypothesis test of each coefficient $\beta_i$ being zero:
  - Null hypothesis $H_0 : \beta_i = 0$
  - Alternative hypothesis $H_A : \beta_i \neq 0$
- Could use above confidence intervals which has double sided 95% confidence:
  - if zero is included within the interval, then "**there is insufficient evidence at the 95% level to reject the null hypothesis**"
  - if zero is excluded within the interval, then "**there is sufficient evidence at the 95% level to reject the null hypothesis**"
- Or you can calculate $p$-values for single sided tests (depending on which direction is most relevant):
  - $H_0 : \beta_i \leq 0$ and $H_A : \beta_i > 0$ (use for positive $\hat{\beta}_i$)
  - $H_0 : \beta_i \geq 0$ and $H_A : \beta_i < 0$ (use for negative $\hat{\beta}_i$)
- using boostrap simulations of coefficient (see next slide)

## Example: Abrasion Resistance Testing Coefficients

- MATLAB code for calculating each type of $p$-value:

```
>> pnegative=sum(bootbetas>0)/nsim
>> ppositive=sum(bootbetas<0)/nsim
```

- which for this example gives:

```
>> pnegative=sum(bootbetas>0)/nsim

pnegative =

     1     0     0

>> ppositive=sum(bootbetas<0)/nsim

ppositive =

     0     1     1
```

- i.e. none of the bootstrap simulations were of opposite sign to OLS estimate in $\hat{\boldsymbol{\beta}}$
- In practice this means the $p$-value is less than the reciprocal of the number of bootstrap simulations
- Should be reported as $p < 1e - 5$ for `nsim=100000`

## Diagnostic Plots for Multiple Regression

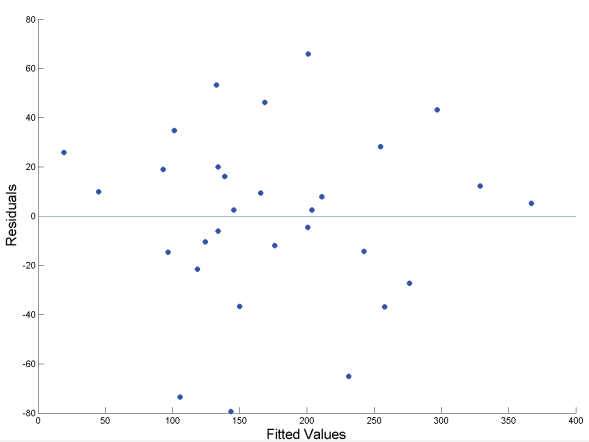
Common diagnostics for multiple regression models:

1. residuals against each explanatory variable;
2. residuals against predictions;
3. leave-one-out change in predictions (or coefficients) against leverages;
4. histogram of residuals;
5. residuals against auxiliary variables (e.g. variables left out of model, or time); and

- Interpretation of (1)-(3) same as for simple linear regression
- (4) should look close to normal if you want to use confidence intervals (or hypothesis testing approaches) based on normal error distribution assumption
- Remember: normality assumption avoided using bootstrap
- (5) highlights other predictors, or correlation with errors over time (thus breaking uncorrelated errors assumption)

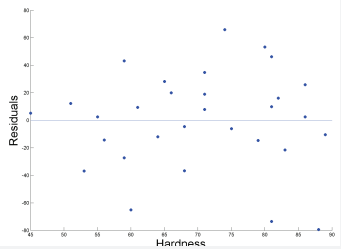
## Example: Abrasion Resistance Diagnostic Plots

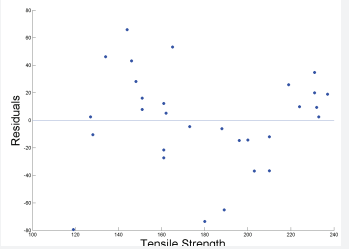
1. Residuals against predictions




## Example: Abrasion Resistance Diagnostic Plots

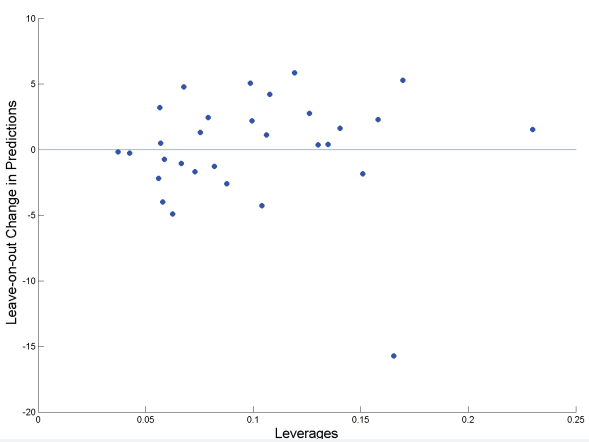
2 Residuals against both explanatory variables



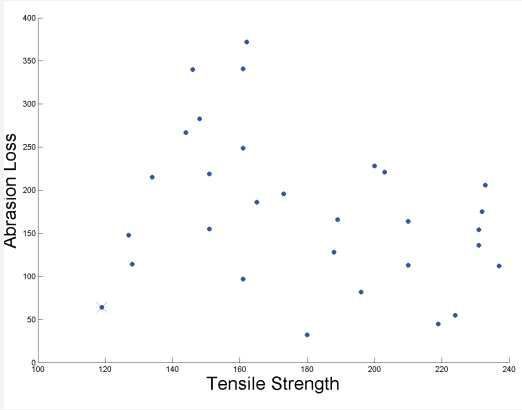

## Example: Abrasion Resistance Diagnostic Plots

3 Leave-one-out change in predictions against leverages



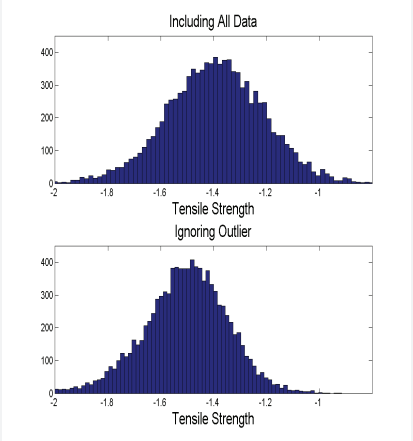

## Example: Abrasion Resistance Diagnostic Plots

- Previous plot suggests an outlier (did you spot it?)
- They are often not easy to spot in large multiple regression problems (shows power of diagnostic plot)
- Looking again at the original abrasion loss and tensile strength data suggest a possible outlier:



## Example: Abrasion Resistance Diagnostic Plots

- Bootstrapped "`Tensile Strength`" coefficients show effect of outlier
- Notice wider spread and possible second mode (around -1.25) when using all data
- Much less spread (uncertainty) in coefficient when outlier is ignored
- **Bootstrap simulations provide another useful diagnostic for influential outliers**




## Summary of Diagnostic Plots

Summary diagnostic plots to evaluate OLS assumptions:

| Assumption | Diagnostic Plots | Check For |
|---|---|---|
| Random and representative of population. (e.g. no outliers) | Scatterplot (Matrix plot)<br><br>Histogram of residuals<br>Leverages against leave-one-out statistics<br>Boostrap correlations/coefficients | No outlying responses<br>Random scatter<br>No outlying residuals<br>No outliers with high leverage<br><br>Multimodel behaviour or large change if outlier ignored |
| Linearity | Residual against explanatory variables | No remaining pattern<br>Constant mean of zero |
| Constant Variance | Residual against explanatory variables<br>Residuals against predicted values | Constant spread<br>Constant spread |
| Normality | Histogram of residuals | Normal shape (if assumed) |
| Uncorrelated | Residuals against explanatory variables or time (if relevant) | No clustering of residuals |

## Model Choice Statistics

- Abrasion model has **small number of explanatory variables** and it is clear that the models provide an **adequate fit**
- If there are a large number of explanatory variables an **objective procedure** is needed to decide how many explanatory variables (and which ones) need to be included in the model to provide an "adequate fit".
- Need to outline **model choice statistics** to assess model adequacy
- If there are a **moderate number of potential explanatory variables**: compare the performance of all $2^p$ models (called the **all possible regressions selection procedure**)
- For large models: **methodical algorithms** commonly used to efficiently search for terms to be included in the model

## Overfitting

- Key principle: model should not **overfit** the data
- An overfitted model has more terms in the model than is required to provide an "adequate fit" to the data
- Consequence of overfitting:
  - model will **perform well on the observed sample** of data used to fit the model
  - but will **perform poorly for predictions on future data**
- Extrapolation of an overfitted model will also tend provide very poor predictions
- An overfitted model explains the random errors in the sample data rather than capture the true underlying relationship
- Thus future predictions will be poor as the overfitted model tries to predict the random noise, which will be different in future observations as it is random!

## Parsimony

- It is desirable to find the simplest model required for the application, which is captured by the concept of **parsimony**:
  - *If two competing models have statistically the same predictive ability then the* **parsimonious model** *is the one with the smaller number of parameters*

## All Possible Regressions

- Considers models containing all possible combinations of the explanatory variables
- For example, if there are 3 explanatory variables $X_1, X_2$ and $X_3$
- There are $2^3 = 8$ subsets of the explanatory variables:
  1. intercept only
  2. $X_1$
  3. $X_2$
  4. $X_3$
  5. $X_1$ and $X_2$
  6. $X_1$ and $X_3$
  7. $X_2$ and $X_3$
  8. all three variables
- Lots of summary statistics to evaluate performance
- In practice, the choice of the criteria is left upto the modeler
- But good practice to make judgement based on many statistics

## Adjusted $R^2$

- (Multiple) coefficient of determination $R^2$ always increases (or at least stays the same) if more explanatory variables are included in the model.
- So useful criteria for model choice
- Adjusted $R^2$ statistics overcomes this issue:

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-(p+1)} \tag{21}$$

- Penalizes models with large numbers of parameters $p$
- Many, many! other statistics
- A **very general approach** is to use cross-validation

## Leave-one-out is Cross-validation

- We have already met the idea of leave-one-out predictions (or coefficients) to investigate outliers
- The leave-one-out idea can also be used to provide a measure of the overall model fit, but which penalizes against overfitting
- Leave-one-out is just a special case of more general **cross-validation** and approach
- So far, the sample of data has been used both to:
  1. estimate the model, and then
  2. assess the model performance.
- In some sense, the information contained in the data is being used twice, which can lead to **overoptimistic estimates of the model performance**.
- A model that performs well on the original sample may perform poorly on future predictions

## Holdout Method

- Simple way to ameliorate this problem, is to randomly split the entire sample into two non-overlapping subsets:
  - **training set** used to fit model
  - **test set** used to evaluate performance
- Known as the **holdout method**
- Typical ratio is 2:1 in favour of training set
- Advantage of this method is that it is less prone to overestimating the model performance due to overfitting and does not require any extra computations
- However, the regression estimates and performance can have a high variance, due to the reduced sample size in each subset.
- The performance evaluation also depends heavily on split into training and test sets, particularly for small datasets
- Different "runs" will likely give different results!

## Cross-Validation

- **K-fold cross validation** improves on holdout method
- Dataset is randomly divided into $K$ subsets and the holdout method is repeated in $K$ trials
- In each trial, one of the $K$ subsets is used as the test set and the other $K-1$ subsets are used as the training set
- Mean sum of square (MSE) of the errors (or similar performance measure) across all $K$ test set trials is computed
- Every observation is in a test set exactly once, and gets to be in a training set $K-1$ times
- Principal advantages are (a) variance of the regression estimates is reduced as the training sample is larger than in holdout method and (b) result is less dependent on initial random allocation to $K$ sets
- Principal disadvantages are (a) the extra computations involved and (b) results can change on each run
- Typically 10-fold cross-validation is the default

## Leave-one-out Cross-Validation and PRESS

- Special case of $K$-fold cross validation is **leave-one-out cross-validation**
- i.e. where one observation is left out at a time or $n$-fold
- Same idea is same as leave-one-out concept we met before
- Mean sum of square of the errors (MSE) for the leave one predictions is then calculated
- Some computer packages give the total sum of square of errors instead, which is called the Prediction REsidual Sum of Squares (PRESS) statistic:

$$\text{PRESS} = \sum_{i=1}^{n} (y_i - \hat{y}_{-i})^2, \tag{22}$$

  where $y_{-i}$ is the prediction for observation $i$ obtained when the regression model is estimated with observation $i$ left out
- Main advantage is that the results will be same on every run, as there is no random allocation

## Cross-Validation and PRESS Comments

- No matter which approach you take, you select the model with the smallest prediction error (total or mean) sum of squares
- PRESS sounds computationally expensive, as you need to fit the regression model $n$ times to each training set
- However, PRESS can be calculated directly from the regression model estimated using the complete dataset
- By adjusting for the influence of each observation (i.e. using the leverage $h_{ii}$):

$$\text{PRESS} = \sum_{i=1}^{n} \left( \frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2 \tag{23}$$

- Hence, the regression model need only be estimated once using the complete dataset, and PRESS is simply the sum of square of the residuals corrected for the leverage

## MATLAB Implementation of Cross-validation

- Thankfully MATLAB implements the tedious cross-validation process using `crossval` function
- First need to create function handle to fit model to training data and predict on test data:

```
>> regf = @(Xtrain, ytrain, Xtest)(Xtest * regress(ytrain,Xtrain))
```

- Then apply cross-validation to dataset:
  `>> cvMSE = crossval('mse',X,y,'predfun',regf)`
- MATLAB defaults to 10-fold cross-validation
- First input `'mse'` tells `crossval` function to calculate MSE
- Following `X` and `y` inputs are design matrix and response vector, to which cross-validation is to be applied
- Last input `'predfun'` sets prediction function handle to `regf`

## Some Explanation of MATLAB Cross-validation Options

- If you specify one of `'Kfold'`, `'Holdout'` or `'Leaveout'` options to `crossval` function then default behaviour is overridden:

```
>> cvMSE = crossval('mse',X,y,'predfun',regf,'Kfold',5) % 5-fold
>> cvMSE = crossval('mse',X,y,'predfun',regf,'Holdout',1/3) % holdout 2:1
>> cvMSE = crossval('mse',X,y,'predfun',regf,'Leaveout',1) % Leave-one-out
```

- Note: only specify one of these options!
- Value of `'Holdout'` as $1/3$, specifies the proportion of data in test set (total number in test set can be specified instead)
- Value of `'Leaveout'` can only be 1
- `crossval` has many options, see: `help crossval` or `doc crossval`

## Example: Abrasion Resistance Cross-validation

- Repeat cross-validation for all model subsets:

```
>> regf = @(Xtrain, ytrain, Xtest)(Xtest * regress(ytrain,Xtrain))
>> cvMSE = crossval('mse',X,y,'predfun',regf)
>> cvMseNoStrength = crossval('mse',X(:,1:2),y,'predfun',regf)
>> cvMSENoHardness = crossval('mse',X(:,[1,3]),y,'predfun',regf)
```

- which gives:

```
>> [cvMse cvMseNoStrength cvMseHardness]

ans =

  1.0e+003 *

    1.5259    3.8825    7.5040
```

- Adjusted $R^2$ available from `regstats` function:
- Conclusion: all variables needed in model as 10-fold cross-validation MSE is smallest for this model

## Example: Abrasion Resistance Adjusted-$R^2$

- Adjusted $R^2$ available from `regstats` function:
  `>> regstats(y,data(:,2:end),'linear',{'adjrsquare'})`
- Which gives $R^2_{adj} = 83\%$ (and 53% for Hardness only model and 6% for Strength only model)
- Overall conclusion: all variables needed in model according to both 10-fold cross-validation and adjusted-$R^2$ statistics

- NOTE: **no assumptions about the distribution of the errors have been required**

## Comparsion of Nested Models

- Two models are nested if the set of variables included in the simpler model (**restricted model**) are a subset of those used in more complex model (**unrestricted model**)
- An ANalysis Of the VAriance (ANOVA) explained by adding the extra terms into the model enables a statistical test as to whether they significantly improve the model fit
- Consider a restricted model ($\beta_{k+1} = \beta_{k+2} = \ldots = \beta_p = 0$):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon. \quad (24)$$

- with $(k+1)$ terms, and it's unrestricted form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \beta_{k+1} x_{k+1} + \ldots + \beta_p x_p + \epsilon \quad (25)$$

- with $(p+1)$ terms, where $p > k$

## Hypothesis test for Nested Models

- Wish to test the null hypothesis:

$$H_0 : \beta_{k+1} = \beta_{k+2} = \ldots = \beta_p = 0,$$

- against the alternative hypothesis:

  $H_1$ : At least one of the coefficients $\beta_{k+1}, \beta_{k+2}, \ldots, \beta_p$ is nonzero.

- It is possible to apply bootstrapping (i.e. using confidence intervals or $p$-values) to evaluate the evidence to reject the null hypothesis based on an appropriate statistic
- **Unfortunately, applying this in a stepwise algorithm for model selection is challenging to implement in MATLAB**
- So we will revert to traditional testing ideas under normality assumption for errors

## $F$-test Statistic for Nested Models

- Denote the Residual Sum of Squares as $RSS = \sum (y_i - \hat{y}_i)^2$
- The test statistic is given by:

$$F = \frac{\text{Explained Variance}}{\text{Unexplained Variance}} = \frac{(RSS_{k+1} - RSS_{p+1})/(p-k)}{RSS_{p+1}/[n-(p+1)]}$$

where:

$$\begin{aligned} RSS_{p+1} &= \text{RSS of unrestricted model} \\ RSS_{k+1} &= \text{RSS of the restricted model} \\ n &= \text{sample size} \\ p+1 &= \#\text{ of terms in unrestricted model} \\ p-k &= \#\text{ of terms constrained to zero in restricted model.} \end{aligned}$$

## $F$-test Statistic Properties

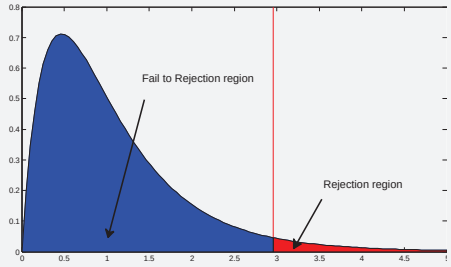
- Essentially, the $F$-test is ratio of the additional variance explained by the extra terms in the more complex model (accounting for the degrees of freedom used up when including them) to that left over in the residuals:
  - If the extra variables are useful for prediction then they will "**explain more variability on average**" than in the random errors (i.e. $F$ statistic will be large)
  - If the extra variables are not useful for prediction the they will "**explain little or the same variability on average**" than in random errors (i.e. $F$ statistic will be small)
- Notice that the denominator in the $F$-test statistic is just the estimated error variance for unrestricted model:

$$\text{Unexplained Variance} = \hat{\sigma}^2_{p+1} = \frac{RSS_{p+1}}{n-(p+1)}$$

## $F$-test Statistic Under Normal Assumption

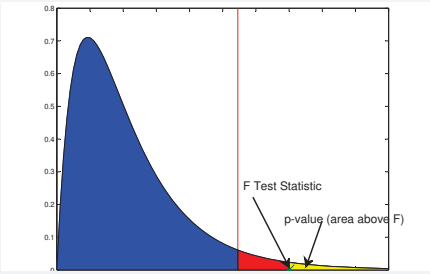
- If we assume the errors in the regression model are normally distributed, in addition to satisfying the usual OLS error assumptions
- Then the $F$-test statistic is known to follow a particular distribution **under the null hypothesis**:
  - $F$ distribution on $v_1 = p - k$ numerator degrees of freedom and $v_2 = n - (p + 1)$ denominator degrees of freedom
  - The null hypothesis is rejected if the $F$ statistic is within the rejection region $F > F_{v_1,v_2}(\alpha)$ for a $100(1-\alpha)\%$ significance test
  - Or equivalently, we reject the null hypothesis if the $p$-value for the $F$ statistics is smaller than the significance level
- See wikipedia for formal definition of $F$-distribution - not expected for this course

## $F$-test Distribution and Rejection Region




- If $F$ statistic calculated for two models is in red region where $F > F_{v_1,v_2}(\alpha)$ then "there is sufficient evidence at the $\alpha\%$ significance level to reject the null hypothesis of zero coefficients (i.e. extra terms should be included as they significantly improve model fit)
- If $F$ statistic is in blue region where $F < F_{v_1,v_2}(\alpha)$ then "there is insufficient evidence at the $\alpha\%$ significance level to reject the null hypothesis of zero coefficients (i.e. extra terms should be left out as they do not significantly improve model fit)

## $F$-test Distribution and $p$-value




- $p$-value is probability of getting the particular $F$ test statistic or something more unusual under the null hypothesis (yellow region)
- Reject null hypothesis if $p$-value for $F$-test statistic is smaller than $\alpha$ and fail to reject otherwise
- Using $p$-value or rejection region ideas are completely equivalent

## General Model Descriptors

- Before discussing variable selection algorithms it is useful to know a few descriptors used for very large models:

| | |
|---|---|
| Maximal Model | Contains all explanatory variables that may be of interest<br>Most complicated model to consider<br>Many terms likely to be insignificant |
| Minimal Adequate Model | Likely a simplified model with less terms than the maximal<br>All terms significantly improve the model fit |
| Null Model | A single parameter, i.e. intercept only<br>Equivalent to having using mean $y = \bar{y}$<br>Usually, not a good fit and no explanatory power |

- Maximal and null models are useful benchmarks with which to judge the performance of others
- Note: in general there isn't a unique minimal adequate model

## Variable Selection Algorithms

- **Aim of model variable selection** is to find the **minimal adequate model** in an efficient manner
- Essentially, these algorithms try to avoid fitting all $2^p$ possible models, by building models (or deconstructing them!) in a hierarchical fashion (one term at a time)
- There are many statistical procedures to accomplish this, but here we will consider the three most commonly used:
- Namely: **forward, backward and stepwise selection**
- We will also discuss some of the key pitfalls/dangers with using these algorithms (they are very controversial)[3]
- **Main piece of advice: NEVER solely rely on these algorithms and always validate the chosen model with subject matter expertise**

[3] See Wikipedia on "stepwise regression"

## Forward Selection

- Forward selection algorithm:
  1. Start with the **null model** (first restricted model)
  2. Create list of all **new potential explanatory variables**
  3. For each of these variables in turn, add only this variable to give a **new unrestricted model**
  4. Evaluate the **performance gain** of each new unrestricted model separately using the above $F$-test statistic (or similar performance statistic)
     - 5a If all the $F$-tests are insignificant (e.g. $p > 0.05$) then retain the restricted model and STOP, this gives the **minimal adequate model**
     - 5b Otherwise, add the single variable that has the largest $F$-test statistic and goto step 6
  6. Start the process again at step 2, with the significant variable from step 5 added to the restricted model
- No terms are dropped from the model during forward model selection

## Backward Selection

- Backward selection algorithm:
  1. Start with the **maximal model** (first unrestricted model)
  2. Create list of all **potentially removable explanatory variables**
  3. For each of these variables in turn, remove only this variable to give a **new restricted model**
  4. Evaluate the **performance drop** of each new restricted model separately using the above $F$-test statistic (or similar performance statistic)
     - 5a If all the $F$-tests are significant (e.g. $p < 0.05$) then retain the unrestricted model and STOP, this gives the **minimal adequate model**
     - 5b Otherwise, add the single variable that has the smallest $F$-test statistic and goto step 6
  6. Start the process again at step 2, with the least useful variable from step 5 removed from the unrestricted model
- No terms are added to the model during backward model selection

## Stepwise Selection

- Stepwise selection is essentially a combination of the forward and backward procedures.
- Stepwise is essentially forward selection, but each "**entry step**" is immediately followed by a "**deletion step**"
- Deletion step re-evaluates variables entered at previous steps
- If all the variables in the current model significantly improve the fit, then none are deleted
- The procedure is stopped when no new variables can significantly improve the model fit
- The reason for considering a **Deletion Step** is that a variable that may have been useful at an early stage in the model build may be superfluous at a later stage after further variables have entered (may be duplication of explanatory information)
- Generally, the significance levels for the entry and deletion steps are the same
- Condition to prevent infinite loops are needed

## General Comments

- Don't use selection algorithms unless you have to!
- Stepwise selection is generally preferred
- Worthwhile trying all approaches you have available
- If considering a polynomial model of order $p$, then usually best to retain the terms of lower order
- Further, if the investigator knows that a certain variable is physically important for prediction then generally this variable should be included in the model (even if the $F$-test indicates it is not important)
- Physical relevance generally overrides statistical significance
- Clearly, a lot of hypothesis tests are being carried out in variable selection algorithms (called "**multiple testing**" problem)
- Therefore, there is a very high probability of making at least one Type I error (including some irrelevant variables) or Type II error (not including important variables)

## Further General Comments

Some of the problems attributed with selection algorithms:

- Generally model selection is ignored, so multiple testing effects and degrees of freedom used are ignored;
- Essentially there will be more uncertainty in our model, than suggested if we ignore model selection stage
- Performance measures like $R^2$ values will be biased to be high;
- Confidence intervals for coefficients and predictions are falsely narrow;
- They have severe problems in the presence of multicollinearity (we'll come to that next)

**General advice: always validate the physical validity of the chosen model using subject matter expertise**

## Multicollinearity in Regression

- In the ideal world all explanatory variables would be uncorrelated with each other (each one then spans an orthogonal dimension in the explanatory vector space)
- Then it is possible to interpret each coefficient on their own, whilst keeping all the other variables fixed
- In real world applications this is rarely the case and we have some level of "**multicollinearity**" between the explanatory variables (see correlation between Hardness and Tensile Strength in Abrasion example)
- Essentially there is an overlap in the linear information content of two or more variables (i.e. two or more explanatory vectors span similar dimensions)
- The presence of multi-collinearity complicates interpretation, analysis and fitting of models

## Multicollinearity in Regression

- Firstly, it is impossible to disentangle the relative contributions of effects of collinear variables on the response variable
- Therefore, the coefficients for collinear terms will be related and cannot be interpreted on their own
- Common to see inappropriate signs of coefficients (i.e. suggesting effect of explanatory variable is opposite to reality)
- Uncertainty estimates (e.g. confidence intervals) are much higher for collinear variables, as effects cannot be disentangled they are very uncertain
- Dependence causes problems for model selection algorithms
- Extremely highly collinear variables can make design matrix ill-conditioned (due to linear dependence in columns) making evaluation of matrix inverse $(\mathbf{X}'\mathbf{X})^{-1}$ numerically unstable

## Multicollinearity - The Good News

- Despite all these problems there is some good news...
- Provided you are only interested in predictions, and not trying to interpret individual coefficients in model, then you don't have to do anything (as long as matrix inverse $(\mathbf{X}'\mathbf{X})^{-1}$ exists!) as the predictions are unaffected by multicollinearity

## Multicollinearity - How to Avoid or Ameliorate?

- Multicollinearity is sometimes avoided by screening variables before modelling commences using subject matter expertise
- Generally, drop the least important collinear variable
- Mean correction of the explanatory variables (particular power terms, $x, x^2, x^3, \ldots$) can substantially aid in the numerical conditioning
- There are various approaches to avoid or ameliorate the effects of multicollinearity, e.g. ridge regression or principal component analysis, but these are beyond the scope of this course
- If the collinearity is between just two variables, then this can be highlighted using the correlation matrix between all the explanatory variables
- But if the collinearity is between three or more variables then alternative statistics, e.g. variation inflation factors (VIF), can be used but these are beyond the scope of this course[4]

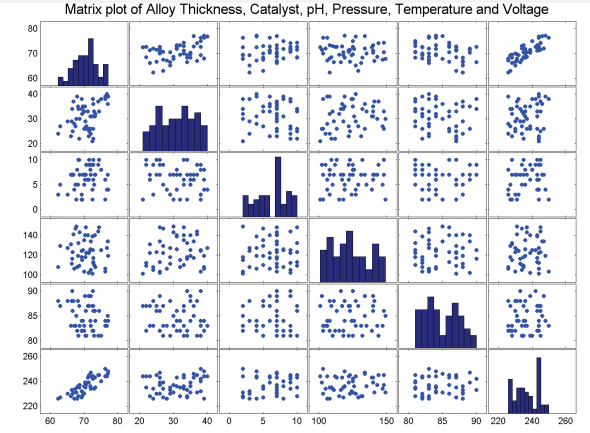
[4] See Wikipedia on "Multicollinearity"

## Example: Alloy Thickness Data

- Article investigating metal deposition using electroplating: Conklin (June, 2009) *3.4 per million: It's a Marathon, Not a Sprint*, Quality Process Journal
- An alloy is layered onto a metal substrate in an acid bath
- For simplicity, assume a single layer is applied
- The key output (response variable) is thickness of the deposit. A minimum thickness is required to ensure the performance.
- A team of engineers and technicians is studying the process, with the goals of reducing variation and fine tuning the key input levels for best effect.
- The key inputs (explanatory variables) are:
  - Catalyst - acid bath catalyst concentration;
  - pH - acid bath pH level;
  - Pressure - pressure in the acid bath tank;
  - Temperature - temperature in the acid bath; and
  - Voltage - voltage applied.

## Example: Alloy Thickness Matrix Plot

```
>> plotmatrix(data)
>> title('Matrix plot of Alloy Thickness, Catalyst, pH,
   Pressure, Temperature and Voltage')
```



## Example: Alloy Thickness Correlation Matrix

```
>> corr(data)
```

| | | | | | |
|---|---|---|---|---|---|
| 1.0000 | 0.3935 | 0.1468 | -0.0407 | -0.2522 | 0.8201 |
| 0.3935 | 1.0000 | -0.1594 | 0.1671 | -0.1545 | 0.1567 |
| 0.1468 | -0.1594 | 1.0000 | 0.0639 | 0.0486 | 0.1853 |
| -0.0407 | 0.1671 | 0.0639 | 1.0000 | 0.0412 | 0.1203 |
| -0.2522 | -0.1545 | 0.0486 | 0.0412 | 1.0000 | 0.1276 |
| 0.8201 | 0.1567 | 0.1853 | 0.1203 | 0.1276 | 1.0000 |

## Example: Alloy Thickness Subjective Impression

- There is a strong positive linear association between `Voltage` and `Thickness`
- There is a moderate positive linear association between `Catalyst` concentration and `Thickness`
- There is a weak (positive/negative) linear association between `pH/Temperature` and `Thickness`
- There is essentially no association between `Pressure` and `Thickness`
  - There is very little association between the explanatory variables
  - So we expect no issues associated with multicollinearity
  - Sample data provide good coverage across range of sensible values
  - Observation study - data collected whilst in operation, so explanatory variable values are not fixed
  - Hence observation resampling relevant bootstrap approach here

## Example: Alloy Thickness Stepwise Selection

- Easy to do stepwise selection in `MATLAB`

```
>> [B,SE,PVAL,INMODEL,STATS,NEXTSTEP,HISTORY]=stepwisefit(data(:,2:end),data(:,1))
```

- Note: `data(:,2:end)` matrix has only explanatory variables, not columns of ones for intercept
- `data(:,1)` is response data vector
- There are lots of options for outputs and inputs, see `doc stepwisefit`
- Commonly used input options are:
  - `penter` - significance level for entry step
  - `premove` - significance level for deletion step
- which are set to 5% below:

```
>> [B,SE,PVAL,INMODEL,STATS,NEXTSTEP,HISTORY]=stepwisefit(data(:,2:end),data(:,1),
                                                  'penter',0.05,'premove',0.05);
```

## Example: Alloy Thickness Stepwise `MATLAB` Output 1

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.1548]    [  0.0356]    'In'        [7.7744e-005]
    [ 0.0864]    [  0.0800]    'Out'       [      0.2862]
    [-0.0420]    [  0.0135]    'In'        [      0.0032]
    [-0.4036]    [  0.0698]    'In'        [6.6247e-007]
    [ 0.4288]    [  0.0278]    'In'        [1.4798e-019]
```

- Step 0: Start with null model (intercept/mean only)
- Step 1 Entry: Column 5 (`Voltage`) entered with $p = 3.2e-13 < 5\%$
- Step 2 Entry: Column 4 (`Temperature`) entered with $p = 1.3e-6 < 5\%$
- Step 2 Deletion: None have $p > 5\%$
- Step 3 Entry: Column 1 (`Catalyst`) entered with $p = 0.00083 < 5\%$
- Step 3 Deletion: None have $p > 5\%$

## Example: Alloy Thickness Stepwise MATLAB Output 2

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.1548]    [  0.0356]    'In'        [7.7744e-005]
    [ 0.0864]    [  0.0800]    'Out'       [     0.2862]
    [-0.0420]    [  0.0135]    'In'        [     0.0032]
    [-0.4036]    [  0.0698]    'In'        [6.6247e-007]
    [ 0.4288]    [  0.0278]    'In'        [1.4798e-019]
```

- Step 4 Entry: Column 3 (`Pressure`) entered with $p = 0.0032 < 5\%$
- Step 4 Deletion: None have $p > 5\%$
- Step 5 Entry: Remaining term has $p = 0.286 > 5\%$ so "**minimal adequate model**" found at Step 4:

$$y = \beta_0 + \beta_1 Voltage + \beta_2 Temperature + \beta_3 Catalyst + \beta_4 Pressure + \epsilon$$

## Example: Alloy Thickness Stepwise MATLAB Output 3

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.1548]    [  0.0356]    'In'        [7.7744e-005]
    [ 0.0864]    [  0.0800]    'Out'       [     0.2862]
    [-0.0420]    [  0.0135]    'In'        [     0.0032]
    [-0.4036]    [  0.0698]    'In'        [6.6247e-007]
    [ 0.4288]    [  0.0278]    'In'        [1.4798e-019]
```

- Lower table gives summary of minimal adequate model
- First column gives coefficients:

$$y = \beta_0 + 0.4288\, Voltage - 0.4036\, Temperature + 0.1548\, Catalyst - 0.0420\, Pressure + \epsilon$$

- Also gives coefficients if non-included term(s) were entered individually:

$$y = \beta_0 + \beta_1 Voltage + \beta_2 Temperature + \beta_3 Catalyst + \beta_4 Pressure + 0.0864\, pH + \epsilon$$

- Notice: other coefficients will generally change due to dependence (correlation) between them:

$$\begin{aligned} y \quad = \quad & 4.4543 + 0.4224\, Voltage - 0.4026\, Temperature + 0.1627\, Catalyst \\ & - 0.0431\, Pressure + 0.0864\, pH + \epsilon \end{aligned}$$

## Example: Alloy Thickness Stepwise MATLAB Output 4

- Default display output is then:

```
Initial columns included:  none
Step 1, added column 5, p=3.19733e-013
Step 2, added column 4, p=1.3158e-006
Step 3, added column 1, p=0.000830253
Step 4, added column 3, p=0.00315094
Final columns included:  1 3 4 5
    'Coeff'      'Std.Err.'    'Status'    'P'
    [ 0.1548]    [  0.0356]    'In'        [7.7744e-005]
    [ 0.0864]    [  0.0800]    'Out'       [     0.2862]
    [-0.0420]    [  0.0135]    'In'        [     0.0032]
    [-0.4036]    [  0.0698]    'In'        [6.6247e-007]
    [ 0.4288]    [  0.0278]    'In'        [1.4798e-019]
```

- Standard error (square root of variance) given in second column
- Third column gives `In/Out` status
- Last column is $p$-value for testing null hypothesis $H_0 : \beta_i \leq 0$ (or $\beta_i \geq 0$ whichever is relevant) under final model
- Notice: $p$-value different to entry/deletion $p$-value
- Output $p$-value assumes normal errors, you will calculate this again using bootstrap without normal assumption
- Other outputs (`STATS`, `NEXTSTEP` and `HISTORY`) not relevant in this course

## Example: Alloy Thickness Minimal Adequate Model

- Now go back to using `regress` function to fit final model:

```
>> y=data(:,1);
>> X=[ones(size(data,1),1) data(:,[6 5 2 4])];
>> [B,BINT,R,RINT,STATS] = regress(y,X);
```

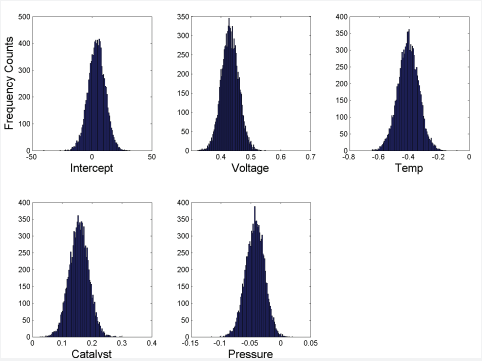
- which gives:

$$y = 3.6833 + 0.4288\, Voltage - 0.4036\, Temperature + 0.1548\, Catalyst - 0.0420\, Pressure + \epsilon$$

- Final $R^2 = 87.3\%$ (compared to $R^2 = 87.6\%$ if pH also included)
- So little drop in performance from ignoring pH

## Example: Alloy Thickness Bootstrapped Coefficients

- Now go back to using `regress` function to fit final model:




- It is clear that bootstrap coefficients are well away from zero, so expect $p$-values (under null hypothesis) to be close to zero

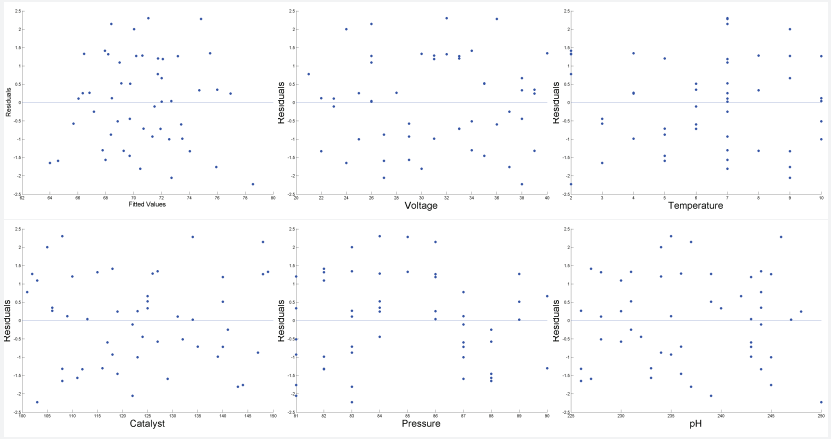
## Example: Alloy Thickness Bootstrap $p$-values

- Calculating the $p$-values for each of the 5 coefficients:

```
>> ppositive=sum(bootbetas<0)/nsim

ppositive =

    0.2921         0    1.0000         0    0.9977

>> pnegative=sum(bootbetas>0)/nsim

pnegative =

    0.7079    1.0000         0    1.0000    0.0023
```

- So $p$-values are $p_{intercept} = 0.29$, $p_{voltage} < 1e-5$, $p_{temperature} < 1e-5$, $p_{catalyst} < 1e-5$ and $p_{pressure} = 0.0023$
- As expected the $p$-values are all less than 5% significance level (else they would have been dropped during Step 4 Deletion)
- Notice these are very similar to last column of `stepwisefit` results, which assume normally distributed errors

## Example: Alloy Thickness Regression Diagnostics




- OLS assumptions look fine, except possibly quadratic with `Pressure` may be needed

## Example: Alloy Thickness Model Performance

- Adjusted-$R^2 = 86.1\%$ (compared to adjusted-$R^2 = 86.2\%$ if `pH` also included)
- So adjusted-$R^2$ suggest `pH` could be included, but only very slight improvement
- Leave-one-out cross-validation MSE for final model is 1.89 (compared to 1.88 if `pH` also included)
- What does this mean?
- The choice of $F$-test statistic for deciding which variables to include in model influences which variables are chosen
- Different model choice statistics can lead to different results
- So try out all available statistics in package (`MATLAB` is limited as it only allows $F$-test) you are using for model fitting.

## Further General Comments

- These model selection algorithms are not failsafe
- There is no guarantee they give "best model" overall
- Always try range of the model selection procedures, and look for consistency in the results and where the differences lie
- **Only use these algorithms when you have to**, i.e. when you have a huge number of explanatory variables
- There are a lot of subjective choices (i.e. test statistic, significance level, algorithm) required in model building, all of which can influence the final results; so be careful!
- Always have in mind the physical sensibility of the model and it's proposed application
- Remember: there is a big difference between physical significance and statistical significance