Aalto University
School of Chemical
Technology

Workbook – Experimental design exercises in Matlab®

Mikko Mäkelä
Aalto University
School of Chemical Engineering
Department of Bioproducts and Biosystems
Espoo, Finland

Table of Contents

Foreword

This is a workbook for experimental design exercises in Matlab®. It was written as a supporting document for the courses organized at the School of Chemical Engineering at Aalto University. Performing design calculations in Matlab®, or other numerical computing software, is very useful for understanding the principles of experimental design and linear regression models. Ready-made functions and faster sequences of commands exist, but this workbook was written as didactical support. Professional design software are also available and are very valuable for the experienced user. However, the more novice user can easily end up memorizing sequences of program menus without actually understanding what is going on.

This workbook is largely based on the one originally written by Prof. Paul Geladi at the Swedish University of Agricultural Sciences in Umeå, Sweden in 2009. The calculations were performed using Matlab® (ver. 9.3, R2017b, The MathWorks, Inc.) with the Statistics and Machine Learning Toolbox (ver. 11.2) installed. However, open source alternatives do exist. Some functions might not be the same or can be missing depending on your setup. Workspace variables, commands and Matlab® outputs are given in **bold** as are vectors and matrices included in equations or in the text. If you have comments, or you find inconsistencies in this workbook, please contact me by e-mail at mikko.makela@aalto.fi.

Background literature

Box G.E.P., Draper N.R., Empirical model-building and response surfaces (1st ed.). John Wiley & Sons Inc, New York, 1987.

Box G.E.P., Hunter J.S., Hunter W.G., Statistics for experimenters (2nd ed.). John Wiley & Sons Inc, Hoboken, New Jersey, 2005.

Myers R.H., Montgomery D.C., Anderson-Cook C.M., Response surface methodology – process and product optimization using designed experiments (3rd ed.). John Wiley & Sons Inc, Hoboken, New Jersey, 2009.

Ryan T.P., Modern experimental design (1st ed.). John Wiley & Sons Inc, Hoboken, New Jersey, 2007.

1. Model building

1.1 Design matrix and coding

This example is from Leardi (2009). A chemical company was interested in the effects of three reagents A, B and C (g) on the viscosity of a polymer (10^3 mPa s), which should be higher than $46 \cdot 10^3$ mPa s. They performed a full factorial design. In general, designs can be written as matrices where the experiments are given as rows and the variables or factors as corresponding columns. The design matrix in original units, **Xori**, can be written as:

```
>> Xori=[9 3.6 9; 11 3.6 9; 9 4.4 9; 11 4.4 9; 9 3.6 11; 11 3.6 11; 9 4.4 11; 11 4.4 11]

Xori =
    9.0000    3.6000    9.0000
   11.0000    3.6000    9.0000
    9.0000    4.4000    9.0000
   11.0000    4.4000    9.0000
    9.0000    3.6000   11.0000
   11.0000    3.6000   11.0000
    9.0000    4.4000   11.0000
   11.0000    4.4000   11.0000
```

where the semicolon separates different rows. The first column of **X** now describes the values of reagent A, the second column reagent B and so on. The corresponding viscosity values were:

```
>> y=[51.8 51.6 51.0 42.4 50.2 46.6 52.0 50.0]'
y =
   51.8000
   51.6000
   51.0000
   42.4000
   50.2000
   46.6000
   52.0000
   50.0000
```

where the apostrophe transposes the row vector into a column. The design matrix is generally coded. This is done by scaling all the maximum and minimum values of a factorial design to 1 and -1, respectively. The first column of the coded design, **X**, can be obtained by:

```
(Xori(:,1)-min(Xori(:,1)))/(range(Xori(:,1))/2)-1
ans =
    -1
     1
    -1
     1
    -1
     1
```

```
   -1
    1
```

In a similar way, the entire **X** can be obtained by:

```
>> X=[(Xori(:,1)-9)/1-1 (Xori(:,2)-3.6)/0.4-1 (Xori(:,3)-9)/1-1]
X =
   -1.0000   -1.0000   -1.0000
    1.0000   -1.0000   -1.0000
   -1.0000    1.0000   -1.0000
    1.0000    1.0000   -1.0000
   -1.0000   -1.0000    1.0000
    1.0000   -1.0000    1.0000
   -1.0000    1.0000    1.0000
    1.0000    1.0000    1.0000
```

Notice the symmetry in the columns. Factorial designs are orthogonal. You can visualize this by drawing the design on a piece of paper. Or you can test it for the first two columns by:

```
>> X(:,1)'*X(:,2)
ans =
     0
```

which is the cosine angle between the first and the second column. The product of a column with itself is the square of the length of the vector, which is a constant. Thus, multiplying the transpose of **X** with **X** should provide a diagonal matrix, where all the other elements are zero:

```
>> X'*X
ans =
    8.0000         0         0
         0    8.0000         0
         0         0    8.0000
```

Questions:

- How is orthogonality explained geometrically?
- Why is orthogonality important for a design?

1.2 Estimating main effects

Now we need to decide which kind of model we want to build. The idea is to separate systematic variation in **y** from noise. If we only want to estimate the average main effects of the reagents, a linear regression equation can be written as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + e \qquad (1)$$

where β_0 denotes the average value of y in the design center, β_1, β_2 and $\beta_3$the coefficients that describe the average effects of the variables, x_1, x_2 and x_3 the coded values of the variables and e the model residual attributed to noise. In matrix notation this equation becomes:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e} \qquad (2)$$

where **y** is a vector of response values, **X** is the coded design matrix, **b** is the model vector and **e** a vector of residuals. It can be shown that a linear regression model that minimizes the sum of squared residuals can be determined by:

$$\mathbf{b} = (\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\mathbf{y} \qquad (3)$$

which is called the least-squares estimate of **b**. For our example, a column of ones first needs to be added to **X**:

```
>> X1=[ones(8,1) X]
X1 =
   1.0000   -1.0000   -1.0000   -1.0000
   1.0000    1.0000   -1.0000   -1.0000
   1.0000   -1.0000    1.0000   -1.0000
   1.0000    1.0000    1.0000   -1.0000
   1.0000   -1.0000   -1.0000    1.0000
   1.0000    1.0000   -1.0000    1.0000
   1.0000   -1.0000    1.0000    1.0000
   1.0000    1.0000    1.0000    1.0000
```

Then the model vector:

```
>> b1=inv(X1'*X1)*X1'*y
b1 =
  49.4500
  -1.8000
  -0.6000
   0.2500
```

Due to coding, the variable effects are now comparable within the design range. A bar chart illustrates the model coefficients:

```
>> bar(b1(2:4));
>> set(gca,'xticklabel',{'b1','b2','b3'})
```

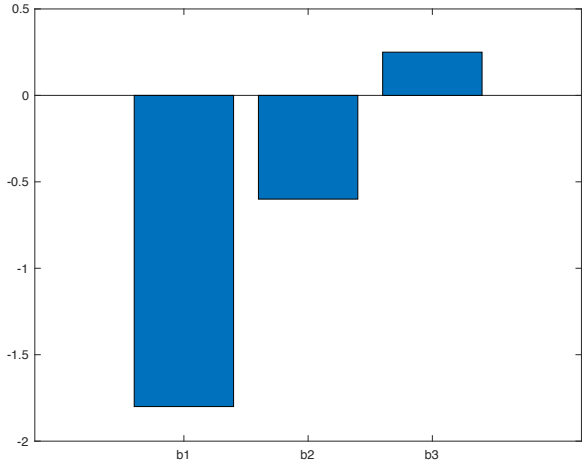

Fig. 1: The estimated main effects of the reagents on polymer viscosity.

The regression model can now be used to calculate the predicted values of **y** and the residuals **e**, see Eq. (2):

```
>> y1hat=X1*b1
y1hat =
   51.6000
   48.0000
   50.4000
   46.8000
   52.1000
   48.5000
   50.9000
   47.3000

>> e1=y-y1hat
e1 =
    0.2000
    3.6000
    0.6000
   -4.4000
   -1.9000
   -1.9000
    1.1000
    2.7000
```

Compare the observed and predicted values:

```
>> [y y1hat]
ans =
   51.8000   51.6000
   51.6000   48.0000
   51.0000   50.4000
   42.4000   46.8000
```

50.2000 52.1000
46.6000 48.5000
52.0000 50.9000
50.0000 47.3000

Questions:

- What does a positive or negative effect mean?
- What does **b(1)** describe?
- How is a coefficient value related to a change in **y** within the design range?
- What can be done with the residuals?
- Is there a suitable way to compare the observed and predicted values?

1.3 Estimating main effects and their interactions

Factorial designs enable estimating variable interactions. With two variables the respective regression equation would look like:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + e \qquad (4)$$

In our case this requires creating the corresponding columns from **X**. With three variables there are three possible two-variable interactions and an interaction for all three variables:

```
>> X2=[ones(8,1) X X(:,1).*X(:,2) X(:,1).*X(:,3) X(:,2).*X(:,3) X(:,1).*X(:,2).*X(:,3)]
X2 =
   1.0000  -1.0000  -1.0000  -1.0000   1.0000   1.0000   1.0000  -1.0000
   1.0000   1.0000  -1.0000  -1.0000  -1.0000  -1.0000   1.0000   1.0000
   1.0000  -1.0000   1.0000  -1.0000  -1.0000   1.0000  -1.0000   1.0000
   1.0000   1.0000   1.0000  -1.0000   1.0000  -1.0000  -1.0000  -1.0000
   1.0000  -1.0000  -1.0000   1.0000   1.0000  -1.0000  -1.0000   1.0000
   1.0000   1.0000  -1.0000   1.0000  -1.0000   1.0000  -1.0000  -1.0000
   1.0000  -1.0000   1.0000   1.0000  -1.0000  -1.0000   1.0000  -1.0000
   1.0000   1.0000   1.0000   1.0000   1.0000   1.0000   1.0000   1.0000
```

where the dot before the asterisk denotes direct multiplication. Now a model that includes the main effects and their interactions:

```
>> b2=inv(X2'*X2)*X2'*y
b2 =
  49.4500
  -1.8000
  -0.6000
   0.2500
  -0.8500
   0.4000
   1.9000
   1.2500
```

This model contains eight terms based on a factorial design with eight experiments. Write down the regression equation based on Eq. (4) and make the same calculations and plots as in Section 1.2. A normal probability plot can be used to illustrate the significance of the coefficients, but it should only be used with factorial designs. First sort the coefficients and build a vector of probabilities:

```
>> b2s=sort(b2(2:end))
b2s =
   -1.8000
   -0.8500
   -0.6000
    0.2500
    0.4000
    1.2500
    1.9000

>> prob=((1:7)-0.5)/7
prob =
    0.0714    0.2143    0.3571    0.5000    0.6429    0.7857    0.9286
```

The probabilities 7 to 93% fit in the range of -2 to 2 standard deviations. Plot the probabilities against the coefficients:

```
>> plot(b2s,prob,'o')
>> ylabel('Probability'); xlabel('Coefficients')
```

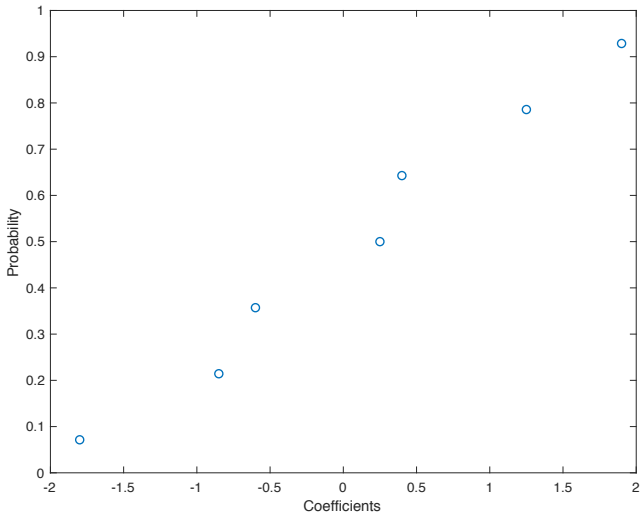

Fig. 2: A normal probability plot of the model coefficients.

Find the values of the model coefficients in Fig. 2 and make a similar plot of all of the residuals.

Questions:

- Were there large and significant interactions?
- Was this model better? Why?
- Are the main effect estimates still the same?
- How is the probability plot used with the coefficients and the residuals? What is the null hypothesis?
- Why can it only be used with coefficients from factorial designs?

1.4 Using real values

The models can also be determined based on the original real values. Recall the design matrix in original units, **Xori**:

```
>> Xori
Xori =
    9.0000    3.6000    9.0000
   11.0000    3.6000    9.0000
    9.0000    4.4000    9.0000
   11.0000    4.4000    9.0000
    9.0000    3.6000   11.0000
   11.0000    3.6000   11.0000
    9.0000    4.4000   11.0000
   11.0000    4.4000   11.0000
```

Add a column of ones and determine a main effect model:

```
>> X3=[ones(8,1) Xori]
X3 =
    1.0000    9.0000    3.6000    9.0000
    1.0000   11.0000    3.6000    9.0000
    1.0000    9.0000    4.4000    9.0000
    1.0000   11.0000    4.4000    9.0000
    1.0000    9.0000    3.6000   11.0000
    1.0000   11.0000    3.6000   11.0000
    1.0000    9.0000    4.4000   11.0000
    1.0000   11.0000    4.4000   11.0000

>> b3=inv(X3'*X3)*X3'*y
b3 =
   70.9500
   -1.8000
   -1.5000
    0.2500
```

Calculate the predicted values:

```
>> y3hat=X3*b3
y3hat =
   51.6000
   48.0000
```

50.4000
46.8000
52.1000
48.5000
50.9000
47.3000

Questions:

- Compare the predicted values with those in Section 1.2, are they the same?
- Why is the mean term **b(1)** different from that of the coded levels?
- What happens for the other regression coefficients?
- Can you think of situations where the coded or uncoded levels would be preferable?

1.5 Response surfaces

Contour plots and response surfaces are ideal for visualizing the predictions as a function of variable levels. This can be done with coded or real values. This example uses coded values and the interaction model from Section 1.3.

For plotting the response surface a grid needs to be created. Since we are using coded values, a range from -1 to 1 with a step of 0.2 can be convenient:

```
>> hor=ones(11,1)*(-1:0.2:1)
hor =
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
  -1.0000  -0.8000  -0.6000  -0.4000  -0.2000       0   0.2000   0.4000   0.6000   0.8000
1.0000
```

```
>> ver=(-1:0.2:1)'*ones(1,11)
```

ver =

-1.0000 -1.0000 -1.0000 -1.0000 -1.0000 -1.0000 -1.0000 -1.0000 -1.0000 -1.0000 -1.0000
-0.8000 -0.8000 -0.8000 -0.8000 -0.8000 -0.8000 -0.8000 -0.8000 -0.8000 -0.8000 -0.8000
-0.6000 -0.6000 -0.6000 -0.6000 -0.6000 -0.6000 -0.6000 -0.6000 -0.6000 -0.6000 -0.6000
-0.4000 -0.4000 -0.4000 -0.4000 -0.4000 -0.4000 -0.4000 -0.4000 -0.4000 -0.4000 -0.4000
-0.2000 -0.2000 -0.2000 -0.2000 -0.2000 -0.2000 -0.2000 -0.2000 -0.2000 -0.2000 -0.2000
0 0 0 0 0 0 0 0 0 0 0
0.2000 0.2000 0.2000 0.2000 0.2000 0.2000 0.2000 0.2000 0.2000 0.2000 0.2000
0.4000 0.4000 0.4000 0.4000 0.4000 0.4000 0.4000 0.4000 0.4000 0.4000 0.4000
0.6000 0.6000 0.6000 0.6000 0.6000 0.6000 0.6000 0.6000 0.6000 0.6000 0.6000
0.8000 0.8000 0.8000 0.8000 0.8000 0.8000 0.8000 0.8000 0.8000 0.8000 0.8000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000

As we have three variables one needs to be set to a constant level for creating a two-dimensional contour plot. Remember the regression equation with three variables from Section 1.3. The third variable is used at its zero level, which eliminates a lot of terms from the equation. Calculate the altitude of the contours, the multiplication between **hor** and **ver** needs to be a direct one:

>> alt=b2(1)+b2(2)*hor+b2(3)*ver+b2(5)*hor.*ver
alt =

51.0000 50.8100 50.6200 50.4300 50.2400 50.0500 49.8600 49.6700 49.4800 49.2900 49.1000
51.0500 50.8260 50.6020 50.3780 50.1540 49.9300 49.7060 49.4820 49.2580 49.0340 48.8100
51.1000 50.8420 50.5840 50.3260 50.0680 49.8100 49.5520 49.2940 49.0360 48.7780 48.5200
51.1500 50.8580 50.5660 50.2740 49.9820 49.6900 49.3980 49.1060 48.8140 48.5220 48.2300
51.2000 50.8740 50.5480 50.2220 49.8960 49.5700 49.2440 48.9180 48.5920 48.2660 47.9400
51.2500 50.8900 50.5300 50.1700 49.8100 49.4500 49.0900 48.7300 48.3700 48.0100 47.6500
51.3000 50.9060 50.5120 50.1180 49.7240 49.3300 48.9360 48.5420 48.1480 47.7540 47.3600
51.3500 50.9220 50.4940 50.0660 49.6380 49.2100 48.7820 48.3540 47.9260 47.4980 47.0700
51.4000 50.9380 50.4760 50.0140 49.5520 49.0900 48.6280 48.1660 47.7040 47.2420 46.7800

51.4500 50.9540 50.4580 49.9620 49.4660 48.9700 48.4740 47.9780 47.4820 46.9860 46.4900
51.5000 50.9700 50.4400 49.9100 49.3800 48.8500 48.3200 47.7900 47.2600 46.7300 46.2000

Now plot the contours:

```
>> C=contourf(hor,ver,alt);
>> clabel(C,'backgroundcolor','white')
>> xlabel('x1'); ylabel('x2')
```

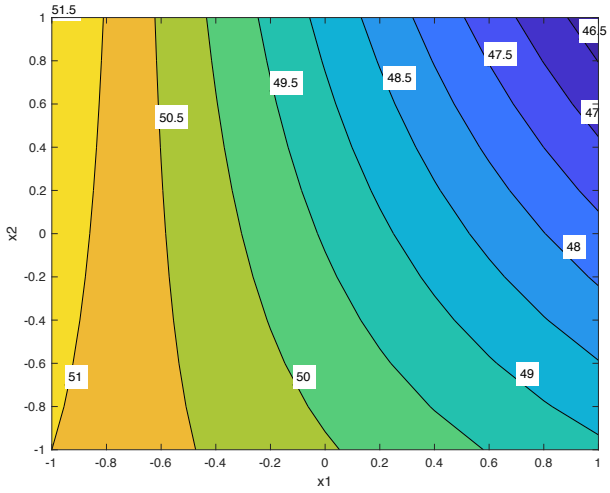

Fig. 3: A contour plot of polymer viscosity as a function of reagents A and B based on the interaction model.

As shown in Fig. 3, there seems to be no problem in reaching the target viscosity of $>46 \cdot 10^3$ mPa s with these settings if the predictions are reliable. Now make the same plot using the real values. Draw the design on a piece of paper and figure out which part of it was plotted.

Questions:

- How would you evaluate the reliability of a response contour?
- How can you calculate the exact predicted value in a specific location?

2. Diagnostics

2.1 Testing significance of coefficients

This example is from Box et al. (2005), pp. 177. The effects of temperature, concentration and catalyst type on product yield during a set of pilot experiments were determined using a full factorial design. The design using coded values:

```
>> X=[-1 -1 -1; 1 -1 -1; -1 1 -1; 1 1 -1; -1 -1 1; 1 -1 1; -1 1 1; 1 1 1];
```

where the semicolon in the end of the command hides the output. Duplicate experiments provided the following average yields:

>> y=[60 72 54 68 52 83 45 80]';

We will start with an interaction model:

>> X1=[ones(8,1) X X(:,1).*X(:,2) X(:,1).*X(:,3) X(:,2).*X(:,3) X(:,1).*X(:,2).*X(:,3)];

And the least-squares estimate of **b**:

```
>> b1=inv(X1'*X1)*X1'*y
b1 =
   64.2500
   11.5000
   -2.5000
    0.7500
    0.7500
    5.0000
         0
    0.2500
```

A bar chart illustrates the coefficients:

```
>> bar(b1(2:end))
>> set(gca,'xticklabel',{'b1','b2','b3','b12','b13','b23','b123'})
>> hold
Current plot held
```

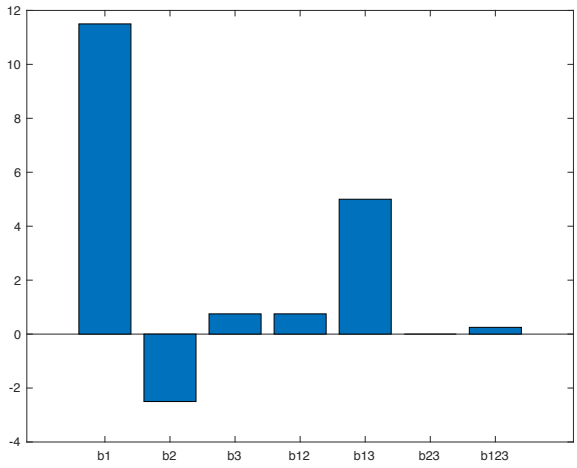

Fig. 4: A bar chart of the coefficients.

As illustrated in Fig. 4, some of the coefficients are small or zero. In order to know which ones to exclude we need to estimate their significance. For this we need an error estimate either based on model residuals or replicate experiments. In this example we are going to use the latter. The following response values were attained from the duplicates:

```
>> y1=[59 74 50 69 50 81 46 79]';
>> y2=[61 70 58 67 54 85 44 81]';
```

The values and their mean are thus:

```
>> [y1 y2 y]
ans =
   59   61   60
   74   70   72
   50   58   54
   69   67   68
   50   54   52
   81   85   83
   46   44   45
   79   81   80
```

Now we can use these values for estimating their variance. The difference of the first set of experiments from the mean is:

```
>> y1-y
ans =
   -1
    2
   -4
    1
   -2
   -2
    1
   -1
```

And the respective sum of squares:

```
>> SS1=(y1-y)'*(y1-y)
ans =
   32
```

Now the sum of squares of the second set:

```
>> SS2=(y2-y)'*(y2-y)
ans =
   32
```

The variance can be obtained as a pooled estimate corrected for the remaining degrees of freedom. We had a total of sixteen observations but calculated the means of eight duplicates, so there are eight degrees of freedom left. The variance:

```
>> s2y=(SS1+SS2)/8
s2y =
    8
```

The variance of the coefficients can be calculated based on the diagonal of the covariance matrix $\sigma^2(\mathbf{X^TX})^{-1}$ where σ^2 is the error estimate. A square root gives the standard errors:

```
>> seb1=sqrt(s2y*diag(inv(X1'*X1)))
seb1 =
    1
    1
    1
    1
    1
    1
    1
    1
```

Now we can use the Student's t test with the null hypothesis that a coefficient equals zero. The t statistic is:

```
>> z1=(b1-0)./seb1
z1 =
   64.2500
   11.5000
   -2.5000
    0.7500
    0.7500
    5.0000
         0
    0.2500
```

which should be compared with the t distribution with the corresponding degrees of freedom. Using $\alpha = 0.10$ could be appropriate here so the probability limit value for the t distribution which is double-sided is:

```
>> tinv(0.95,8)
ans =
    1.8595
```

Now which coefficients are likely to be something else than noise? It is helpful to visualize the distribution:

```
>> x=-5:0.05:5;
>> figure, plot(x,tpdf(x,8))
>> ylabel('Probability density')
```

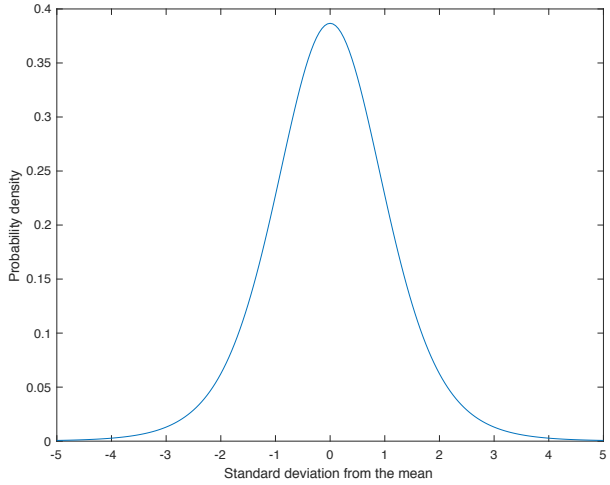

Fig. 5: The t distribution with eight degrees of freedom.

As illustrated in Fig. 5, we are more confident on accepting the alternative hypothesis that a coefficient is different from zero if the t statistic is closer to the extremes of the distribution. Confidence limits can also be calculated:

```
>> b1ci=tinv(0.95,8)*seb1
b1ci =
   1.8595
   1.8595
   1.8595
   1.8595
   1.8595
   1.8595
   1.8595
   1.8595
```

Now plot these in the earlier bar chart:

```
>> errorbar(b1(2:end),b1ci(2:end),'linestyle','none')
```

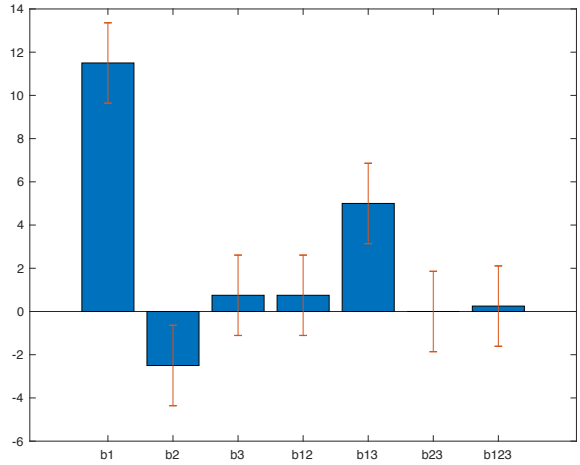

Fig. 6: A bar char of the coefficients with the respective confidence intervals ($\alpha = 0.10$).

As illustrated in Fig. 6, temperature, concentration and the interaction between temperature and catalyst type had a significant effect on product yield. We might then be confident on removing the insignificant terms from the model. However, if a variable is included in an important interaction, it cannot be removed.

Questions:

- How do the degrees of freedom and α level affect the confidence intervals?
- Is the use of $\alpha = 0.10$ appropriate here?
- The variance can also be estimated based on the model residuals. Are there problems with doing this based on saturated designs? What happens with the degrees of freedom?
- In this example the model was built based on the mean values of the duplicates. Can you think of another way of using the results? Try to repeat the calculations.

2.2 Analysis of variance (ANOVA)

In this example we will use the data from Section 2.1. However, no mean values are calculated and the performed duplicates will be treated as individual observations. Then the design also needs to include the duplicate rows:

```
>> X=[X;X]
X =
   -1   -1   -1
    1   -1   -1
   -1    1   -1
    1    1   -1
   -1   -1    1
    1   -1    1
   -1    1    1
    1    1    1
   -1   -1   -1
```

```
   1    -1    -1
  -1     1    -1
   1     1    -1
  -1    -1     1
   1    -1     1
  -1     1     1
   1     1     1
```

And the corresponding **y** values:

```
>> y=[y1;y2];
```

Use a model with the significant terms from Section 2.1. The design matrix:

```
>> X2=[ones(16,1) X X(:,1).*X(:,3)];
```

And the model vector:

```
>> b2=inv(X2'*X2)*X2'*y;
```

The principle of ANOVA with regression models is to compare the variation explained by the model against noise, i.e., the part not explained by the model. To do this, we need to know how much variation there is to begin with. Thus, the total sum of squares of **y**:

```
>> SStot=(y-mean(y))'*(y-mean(y))
SStot =
       2699
```

Now we want to distribute this to the model and the residuals. It is easy to start with the residuals. First determine the predicted values and then the residuals:

```
>> y2hat=X2*b2;
>> e2=y-y2hat
e2 =
  -0.5000
   1.5000
  -4.5000
   1.5000
  -1.0000
  -3.0000
        0
        0
   1.5000
  -2.5000
   3.5000
  -0.5000
   3.0000
   1.0000
  -2.0000
```

2.0000

Now the sum of squares of the residuals:

```
>> SSres2=e2'*e2
SSres2 =
   74
```

The sum of squares are additive. Thus, the model sum of squares is:

```
>> SSmod2=SStot-SSres2
SSmod2 =
      2625
```

ANOVA is based on the F test, which compares the mean square of the model against the mean square of the residuals based on the respective degrees of freedom. The null hypothesis is that all coefficients equal zero. The alternative hypothesis is that at least one coefficient does not equal zero and the model explains something else than noise.

The degrees of freedom are also additive. So, if there were sixteen observations in total from which we calculated the mean, there are fifteen degrees of freedom left for the model and the residuals. In addition to the mean term there are four terms in the model so the mean square:

```
>> MSmod2=SSmod2/4
MSmod2 =
  656.2500
```

This leaves eleven degrees of freedom for the residual:

```
>> MSres2=SSres2/11
MSres2 =
   6.7273
```

The mean squares are used for calculating the F ratio:

```
>> F=MSmod2/MSres2
F =
  97.5507
```

which is then compared with the F distribution based the respective degrees of freedom. A distinct distribution exists for all combinations of degrees of freedom and it is helpful to visualize the distribution:

```
>> x=0:0.01:7;
>> plot(x,fpdf(x,4,11))
>> ylabel('Probability density'); xlabel('F ratio')
```

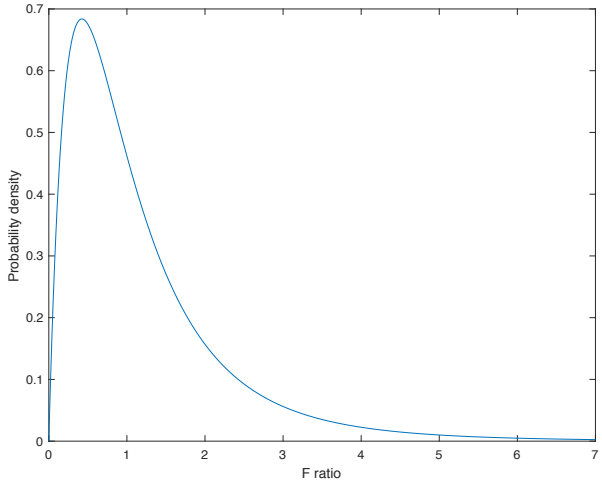

Fig. 7: The F distribution with four and eleven degrees of freedom.

As illustrated in Fig. 7, we are more comfortable in accepting the alternative hypothesis if the F ratio is large. In our case there seems to be no problem, the F distribution is one-sided and the probability limit value for $\alpha = 0.05$ is:

>> finv(0.95,4,11)
ans =
3.3567

The ANOVA is generally summarized in a table such as Table 1 below. See which parameters are additive and which are not. The mean squares of the individual model terms can also be determined and p values can be used.

Table 1: An ANOVA table based on the previous example.

Source	Degrees of freedom	Sum of squares	Mean Square	F ratio
Total corrected	15	2700		
Model	4	2630	656	97.6
Residual	11	74.0	6.73	

Questions:

- What was the null hypothesis? Is this a useful hypothesis?
- The mean of the response values was not included in the ANOVA. Are there situations where this should also be tested?
- Is the significance of a model enough, or should we also look at something else?

- What is an ideal number of degrees of freedom for the residual?

2.3 Coefficient of determination

It is useful of quantify how much of the original variation the model actually explains. This can be done through the coefficient of determination, the R^2 value. The calculation is based on the sum of squares. With the data from Section 2.2, the R^2 value is simply:

```
>> R2=SSmod2/SStot
R2 =
   0.9726
```

Or:

```
>> R2=1-SSres2/SStot
R2 =
   0.9726
```

The R^2 is always in the range 0-1. The value determined above indicates that the model explains 97% of the variation in **y** around its mean. It can also be useful to plot the predicted values against the observed ones:

```
>> plot(y,y2hat,'o')
>> axis('image'); refline(1,0)
>> ylabel('Predicted'); xlabel('Observed')
```

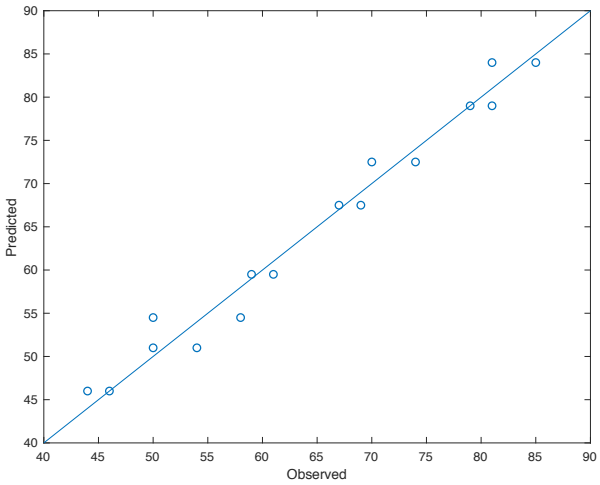

Fig. 8: Predicted vs. observed based on the model from Section 2.2.

If the R^2 value would equal 1, all the points would lie on the straight line in Fig. 8.

Questions:

- Are R^2 values of 0 or 1 realistic in practical situations?
- Can you think of any weaknesses in using the R^2?

2.4 Residuals

Model residuals are useful for identifying potential outlier experiments or other anomalies. Remember that a residual describes the difference between the observed and the predicted value. Recall the residuals from Section 2.2:

```
>> e2
e2 =
  -0.5000
   1.5000
  -4.5000
   1.5000
  -1.0000
  -3.0000
        0
        0
   1.5000
  -2.5000
   3.5000
  -0.5000
   3.0000
   1.0000
  -2.0000
   2.0000
```

Different plots can be used. As an example, the residuals can be plotted against experiment or row number in the design:

```
>> exp=(1:1:16)';
>> subplot(2,2,1), plot(exp,e2,'o')
>> title('(a)'); ylabel('Residual'); xlabel('Experiment'); refline(0,0)
```

Or a hypothetical run order:

```
>> runOrder=randperm(length(exp))';
>> [runOrder_sorted,index]=sort(runOrder);
>> e2_sorted=e2(index,:);
>> subplot(2,2,2), plot(runOrder,e2_sorted,'o')
>> title('(b)'); ylabel('Residual'); xlabel('Run order'); refline(0,0)
```

Raw residuals are however not the most useful. A normal probability shows normally distributed residuals on a line:

```
>> e2_sorted2=sort(e2)
>> prob=((1:length(e2_sorted2))-0.5)/length(e2_sorted2)
>> subplot(2,2,3), plot(e2_sorted2,prob,'o')
>> title('(c)'), ylabel('Probability'), xlabel('Residual')
```

Standardized residuals can also be used. The mean square of the residuals provides an estimate of model error and its square root is convenient for standardization:

```
>> e2s=e2./sqrt(MSres2)
e2s =
   -0.1928
    0.5783
   -1.7350
    0.5783
   -0.3855
   -1.1566
         0
         0
    0.5783
   -0.9639
    1.3494
   -0.1928
    1.1566
    0.3855
   -0.7711
    0.7711
```

```
>> subplot(2,2,4), plot(exp,e2s,'o')
>> title('(d)'); ylabel('Standardized residual'); xlabel('Experiment'); refline(0,0)
```

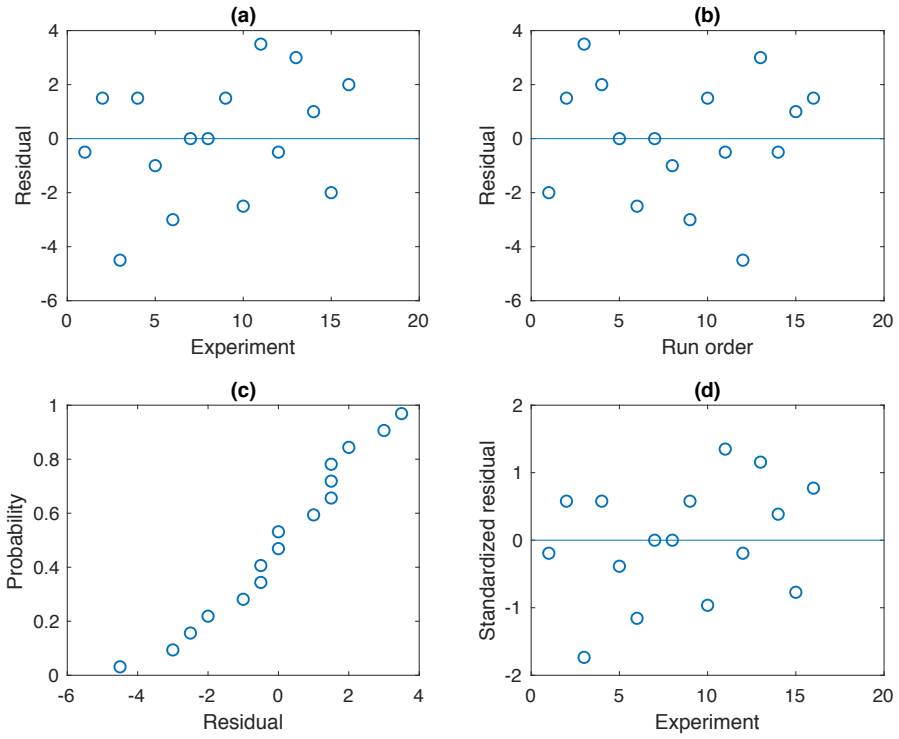

Fig. 9: (a) Model residuals based on experiment number, (b) residuals based on run order, (c) a normal probability plot of the residuals and (d) standardized residuals based on experiment number.

Questions:

- What kind of properties are expected of the residuals?
- How can the normal probability plot or standardized residuals be used to identify potential outliers?
- Can signs of non-linearity be detected?

References

Box, G.E.P., Hunter, J.S. and Hunter, W.G. (2005) Statistics for Experimenters, John Wiley & Sons, Inc., Hoboken, New Jersey.

Leardi, R. (2009) Experimental design in chemistry: a tutorial. Analytica Chimica Acta 652, 161-172. doi: 10.1016/j.aca.2009.06.015.