Exercise 10 – ANOVA (cont'd)

Repeat the one-way ANOVA of the dataset from Exercise 8. An alloy is being prepared with three different catalysts: A, B, and C. The concentrations of a major element are determined for each catalyst in several samples:

```
A=transpose([42.7 45.6 43.1 41.6 NaN])
B=transpose([44.9 48.3 46.2 NaN NaN])
C=transpose([41.9 44.2 40.5 43.7 41.0])
```

To construct the matrix, you first need to make sure that your data columns are of equal

```
X=[A B C]
[p table stats]=anova1(X)
```

Inspect the table and the figure that are produced. Note that, compared to the previous exercise, the anova1 command now also generates additional statistics, whish are stored in the structure 'stats'.

1. This variable 'stats' can be now used for making pair wise comparisons between the means of the three treatments:

```
c=multcompare(stats)
```

This command generates a matrix 'c'. Each row of this matrix has five columns.
Columns 1 and 2: the treatments that are being compared (e.g. treatments 1 and 3)
Column 4: the difference between the corresponding means (e.g. μ_1-μ_3)
Columns 3 and 5: the minimum and maximum values that could be expected for the difference in means (column 4), given the variances in the data and a 95% confidence interval. (Remember *t*-test?) If the two means (e.g. μ_1 and μ_3) are statistically the same, then this interval (between values in columns 3 and 5) should include zero! If it does not, then the two means are statistically different with 95% confidence.

The command also generates an interactive graph. Explore this graph by clicking on its data points.

2. Now, we'd like to make a transition from 1-factor analysis to many-factor analysis. The latter is done in Matlab via the 'anovan' command ('n' stands for N-factor). But first, let's try using 'anovan' for our 1-factor problem above. The format of the anovan command is slightly different, so the variables need to be reshaped:

```
X=X(:)
factor=repmat(1:3,5,1)
factor=factor(:)
```

This writes the observations in a single column X and creates the matrix 'factor', which specifies the levels of the experimental factors for each row in X.

Now you can run the analysis:

```
p=anovan(X,factor)
```

Verify that the result is the same as before.

3. Now, let's look at the effects of two factors. Load the data from the file 'normtemp.mat', which is by now familiar to you and contains measurements of normal body temperature, patient gender, and heart rate. Let's define these into separate variables:

```
T=normtemp(:,1)
gen=normtemp(:,2)
hr=normtemp(:,3)
```

We would like to know if the heart rate is affected by either the temperature or gender of the patient. First, plot the data to inspect it visually:

```
plot(T,hr,'o')
plot(hr,gen,'o')
```

Hard to tell, isn't it? Now let's perform ANOVA:

```
p=anovan(hr,[T gen])
```

The outputs are the standard ANOVA table and the probabilities that the effects of each factor are null. What can you conclude? The command also allows seeing if the effects of one factor (body temperature) depend on the levels of another factor (gender)

```
p=anovan(hr,[T gen], 'model','interaction')
```

Let's visualize our results. The first 65 rows in our data are for men, the last 65 are for women. The distribution of data can be visualized with a box plot:

```
boxplot(hr,gen)
```

This produces a box plot of heart rate, according to gender. The red line is the median, the "whiskers" are the maximum and minimum values, and the box extends from the lower quartile to the upper quartile in the data. Do the same for temperature:

```
boxplot(hr,T)
```

Does this graph support the conclusion of ANOVA?
Compare with the results of a linear fit and correlation:

```
corrcoef(hr,T)
fit(hr,T,'poly1')
```

Can you use ANOVA to verify if the normal body temperature depends on gender?