DEN 435 - Homework 3

%%%%%%%%%%%%%%%%%%%%%%%%

<u>To Do:</u>

DONE. 10/17 ---- 8.3.7 - Surface Plot

Problems:

DONE. 8.4,

DONE. 8.5,

DONE. 8.17

%%%%%%%%%%%%%%%%%%%%%%%%

Problem 8.2 - 8.3 - Tool life experiment

1. Import data

```
dFF3 = fullfact([2 2 2])
```

```
dFF3 = 8×3
     1     1     1
     2     1     1
     1     2     1
     2     2     1
     1     1     2
     2     1     2
     1     2     2
     2     2     2
```

```
treat = {'(1)';'a';'b';'ab';'c';'ac';'bc';'abc'} % Creates a cell array
```

```
treat = 8×1 cell
'(1)'
'a'
'b'
'ab'
'c'
'ac'
'bc'
'abc'
```

```
% Create a grouping variable
gtreat = [treat; treat]
```

```
gtreat = 16×1 cell
'(1)'
'a'
'b'
'ab'
'c'
'ac'
'bc'
'abc'
'(1)'
'a'
```

⋮

```
% Replace 1s and 2s with -1s and +1s, respectively, to code the design.
% Initialize array to hold coded variables
dFF3c = nan;
% Repetition structure
for i = 1:length(dFF3(:,1)) % From one to number of number of rows in column 1
    for j = 1:length(dFF3(1,:))
        if dFF3(i,j) == 1
            dFF3c(i,j) = -1;
        else
            dFF3c(i,j) = 1;
        end
    end
end

% Factors
f1 = dFF3c(:,1); % Cutting speed
f2 = dFF3c(:,2); % Metal hardness
f3 = dFF3c(:,3); % Cutting angle
% Table to hold factor levels
tbl80 = table(f1,f2,f3,'VariableNames',{'Factor1','Factor2','Factor3'})
```

tbl80 = 8×3 table

	Factor1	Factor2	Factor3
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	-1
5	-1	-1	1
6	1	-1	1
7	-1	1	1
8	1	1	1

2. Response column(s)

```
tlife1 = [221;325;354;552;440;406;605;392]
```

```
tlife1 = 8×1
   221
   325
   354
   552
   440
   406
   605
   392
```

```
tlife2 = [311;435;348;472;453;377;500;419]
```

```
tlife2 = 8×1
   311
   435
   348
   472
   453
   377
   500
   419
```

```
% Table to hold replicates
tbl831 = table(tlife1,tlife2,'VariableNames',{'Replicate1','Replicate2'})
```

tbl831 = 8×2 table

	Replicate1	Replicate2
1	221	311
2	325	435
3	354	348
4	552	472
5	440	453
6	406	377
7	605	500
8	392	419

```
% Replicates vector
rtlife = [tlife1;tlife2]
```

```
rtlife = 16×1
   221
   325
   354
   552
   440
   406
   605
   392
   311
   435
     ⋮
```

3. Examine data graphically

```
boxplot(rtlife,gtreat)
```

4. Create Fitted Model

The regression model

Stack (concatenate) the replicates into a single variable.

```
stlife = stack(tbl831,1:2) % Output is a concatenated table of the replicates.
```

stlife = 16×2 table

	Replicate1_Replicate2_Indicator	Replicate1_Replicate2
1	Replicate1	221
2	Replicate2	311
3	Replicate1	325
4	Replicate2	435
5	Replicate1	354
6	Replicate2	348
7	Replicate1	552
8	Replicate2	472
9	Replicate1	440
10	Replicate2	453
11	Replicate1	406
12	Replicate2	377
13	Replicate1	605
14	Replicate2	500
15	Replicate1	392
16	Replicate2	419

Create a table to hold the DOE test matrix and response variables.

```
tbl832 = table([f1;f1],[f2;f2],[f3;f3],[tlife1;tlife2],'VariableNames',{'CuttingSpeed','MetalHa
```

tbl832 = 16×4 table

	CuttingSpeed	MetalHardness	CuttingAngle	ToolLife
1	-1	-1	-1	221
2	1	-1	-1	325
3	-1	1	-1	354
4	1	1	-1	552
5	-1	-1	1	440
6	1	-1	1	406
7	-1	1	1	605
8	1	1	1	392
9	-1	-1	-1	311

	CuttingSpeed	MetalHardness	CuttingAngle	ToolLife
10	1	-1	-1	435
11	-1	1	-1	348
12	1	1	-1	472
13	-1	-1	1	453
14	1	-1	1	377
15	-1	1	1	500
16	1	1	1	419

Linear regression model

Recall: Regression models describe the relationship between a response (output) variable, and one or more predictor (input variables).

```
mdl = fitlm(tbl832,'ToolLife ~ CuttingSpeed + MetalHardness + CuttingAngle + CuttingSpeed*Metal
```

```
mdl = 
Linear regression model:
    ToolLife ~ 1 + CuttingSpeed*MetalHardness + CuttingSpeed*CuttingAngle + MetalHardness*CuttingAngle + CuttingSpee

Estimated Coefficients:
                                                Estimate      SE        tStat        pValue  
                                                ________    ______    ________    _________

    (Intercept)                                  413.13     12.406      33.301    7.2169e-10
    CuttingSpeed                                  9.125     12.406     0.73554       0.48302
    MetalHardness                                42.125     12.406      3.3956     0.0094221
    CuttingAngle                                 35.875     12.406      2.8918      0.020144
    CuttingSpeed:MetalHardness                   -5.625     12.406    -0.45341        0.6623
    CuttingSpeed:CuttingAngle                   -59.625     12.406     -4.8062     0.0013449
    MetalHardness:CuttingAngle                  -12.125     12.406    -0.97736       0.35702
    CuttingSpeed:MetalHardness:CuttingAngle     -17.375     12.406     -1.4005       0.19892


Number of observations: 16, Error degrees of freedom: 8
Root Mean Squared Error: 49.6
R-squared: 0.854,  Adjusted R-Squared: 0.726
F-statistic vs. constant model: 6.66, p-value = 0.0079
```

```
% Remove factors that are not statistically significant.
% mdl2 = removeTerms(mdl,'Factor1')
% mdl3 = removeTerms(mdl2,'Factor1*Factor2')
% mdl4 = removeTerms(mdl3,'Factor2*Factor3')
% mdl5 = removeTerms(mdl4,'Factor1*Factor2*Factor3')
```

To obtain a more robust model; we then iterate the regression analysis procedure on only the significant terms.

```
% Or create a new model with only statistically significant terms
mdl2 = fitlm(tbl832,'ToolLife ~ MetalHardness + CuttingAngle + CuttingSpeed:CuttingAngle')
```

```
mdl2 = 
Linear regression model:
    ToolLife ~ 1 + MetalHardness + CuttingAngle + CuttingSpeed:CuttingAngle

Estimated Coefficients:
```

```
                                  Estimate      SE       tStat       pValue  
                                  ________    ______    _______    __________

    (Intercept)                    413.13     12.231     33.778    2.8807e-13
    MetalHardness                  42.125     12.231     3.4442     0.0048565
    CuttingAngle                   35.875     12.231     2.9332      0.012529
    CuttingSpeed:CuttingAngle     -59.625     12.231    -4.8751     0.0003817


Number of observations: 16, Error degrees of freedom: 12
Root Mean Squared Error: 48.9
R-squared: 0.787,  Adjusted R-Squared: 0.733
F-statistic vs. constant model: 14.7, p-value = 0.00025
```

Summary of Fit

R-Square

- Estimates the proportion of variation in the response that can be attributed to the model rather than to random error. Calculated:
- SSmodel / SStotal

R-Square Adj

- Adjusts the R-Square statistic for the number of parameters in the model. R-Square Adj facilitates comparisons among models with different numbers of parameters. Calculated:
- 1 - MSerror / (SStotal / DFtotal)

Root Mean Square Error

- Estimates the standard deviation of the random error.

ANOVA

Prob > F

- The p-value for the test. The Prob > F value measures the probability of obtaining an F Ratio as large as what is observed, given that all parameters except the intercept are zero. Small values of Prob > F indicate that the observed F Ratio is unlikely. Such values are considered evidence that there is at least one significant effect in the model.[1]

Parameter Estimates

Estimate

- The parameter estimates for each term. These are the estimates of the model coefficients (the Beta's).

[1] https://www.jmp.com/support/help/en/15.0/#page/jmp/analysis-of-variance.shtml#ww137405

5. Plot Residuals

Residuals help in checking model quality; these plots help in discovering errors, outliers, or correlations in the model or data.

```
plotResiduals(mdl) % Original model
```

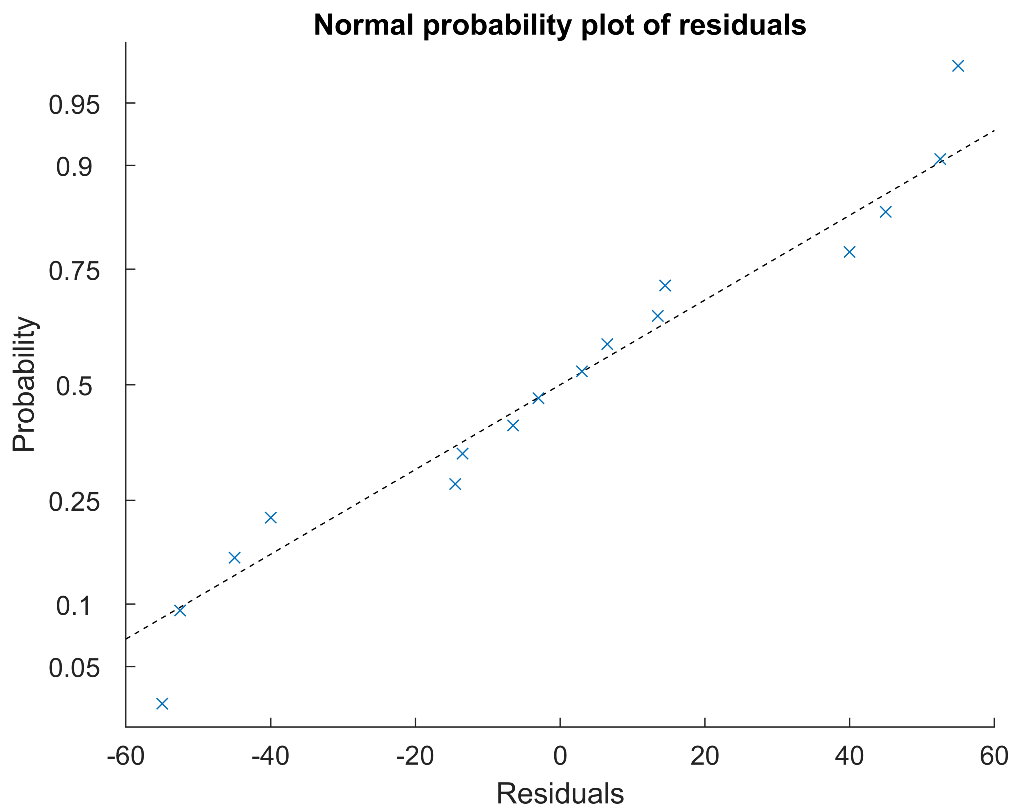
```
plotResiduals(mdl2) % New model with statistically significant terms only
```

Residuals histogram plot: shows the range of the residuals and their frequencies. The area of each bar is the relative number of observations. The sum of the bar areas is equal to 1.

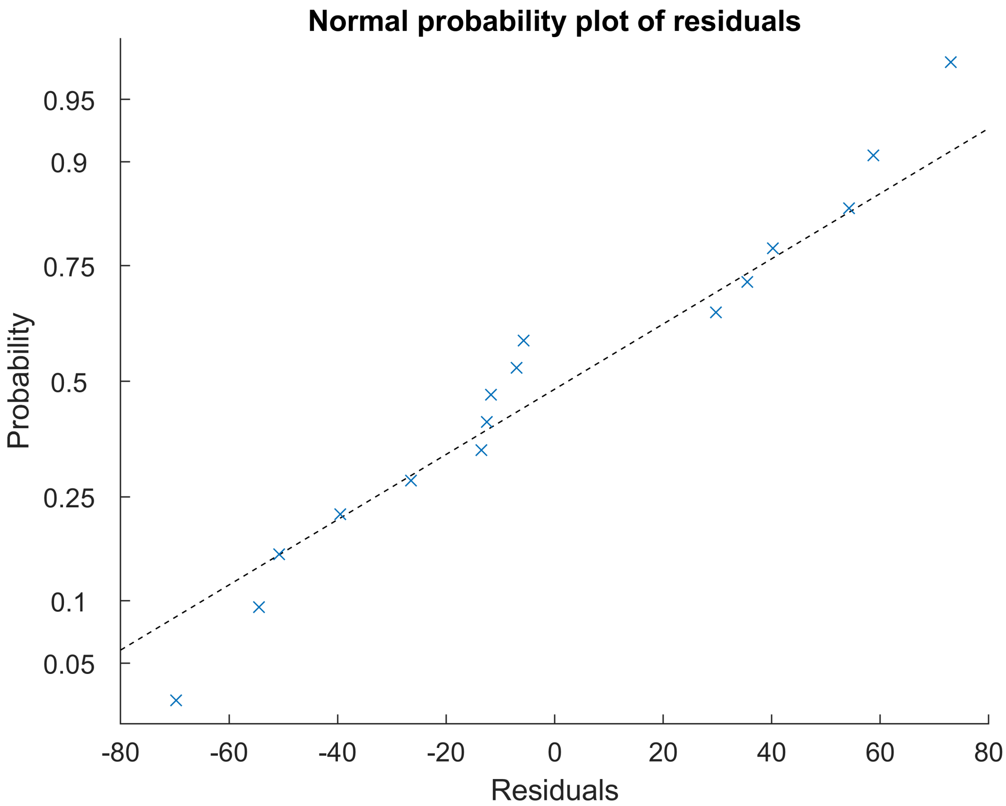
The generated histogram plot of the new model does not suggest any problems with model assumptions.

Notice the comparison of the original model and the new model; the original model contains a higher frequency of residuals in the -50 to +50 range as compared to the new model.

```
plotResiduals(mdl,'probability') % Original
```




```
plotResiduals(mdl2,'probability') % New
```

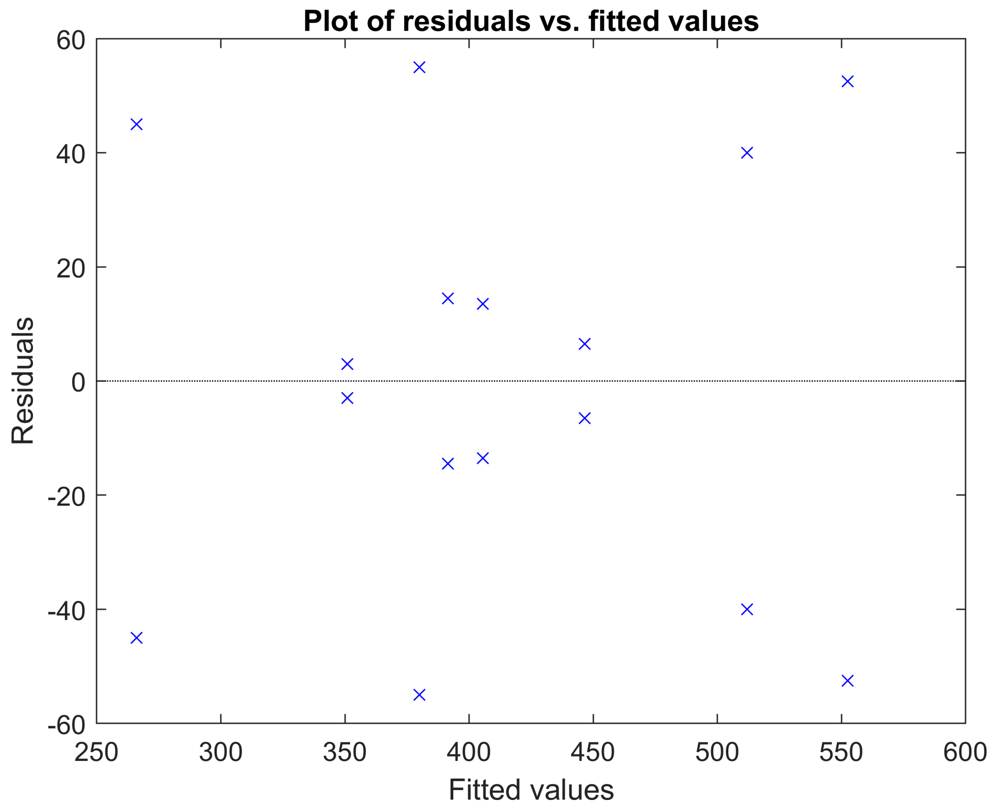


Normal probability plot of residuals: shows how the distribution of the residuals compares to a normal distribution with matched variance. The probability plot seems reasonably straight, meaning a reasonable fit to normally distributed residuals.

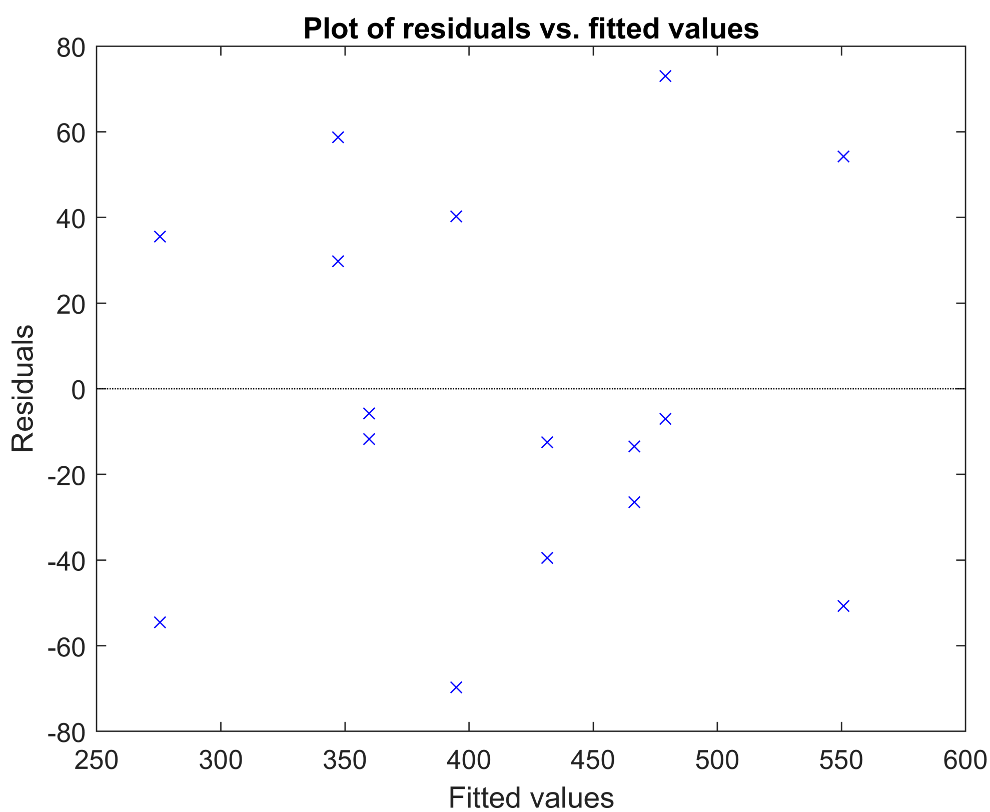
- Small deviations from the straight line in a normal probability plot are common, but an “S” shaped curve on this graph suggests a bimodal distribution of residuals. *Breaks near the middle of this graph are also indications of abnormalities in the residual distribution.*

Both probability plots are very similar as indicated by the adjusted R-square value for both models.

```
plotResiduals(mdl,'fitted') % Original
```



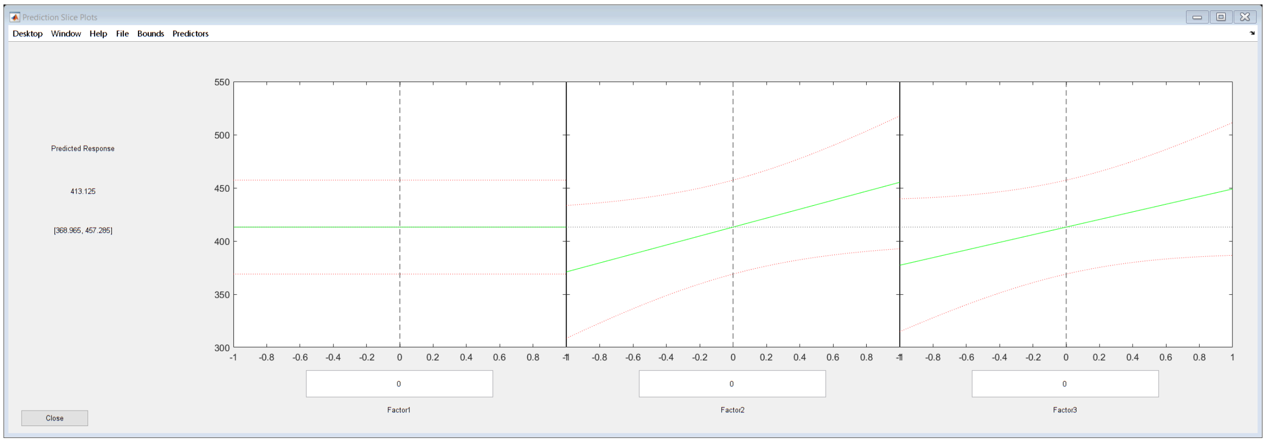

```
plotResiduals(mdl2,'fitted') % New
```



Plot of residuals vs. fitted values: the center-line is the fitted model and data points are residuals. The spread among the residuals seems to be consistent across the domain of the model, so we can again conclude that this plot does not suggest any problems with model assumptions.

6. Prediction Profiler

```
% Plot a prediction profiler
plotSlice(mdl2)
```



Plot of slices through fitted linear regression surface. Also displays the 95% confidence bounds for the response values.

7. Response by Adjusted Whole Model (Predicted) Plot

```
plot(mdl2)
```

The plot shows the whole model except the constant (intercept) term.

Each 'x' is an observation, the red line is the linear regression line, and the red-dashed lines on either side is the 95% confidence bounds. The slope of the linear regression line is the slope of a fit to the predictors projected onto their best-fitting direction, i.e. the norm of the coefficient vector.

The model as a whole is significant because a 'horizontal line' (the mean the response) does not fit between the confidence bounds.

8. Surface Plot

```
% Create grid
% Nodes
res = 0.1; % Resolution
x1 = [min(f1):res:max(f1)]; % Column vector from low level to high level in increments of resol
x2 = [min(f2):res:max(f2)];
x3 = [min(f3):res:max(f3)];
```

```
nx1 = length(x1); % Returns the number of elements in the vector (nodes)
nx2 = length(x2);
nx3 = length(x3);

% Mesh
[X2,X3] = meshgrid(x2,x3); % meshgrid function returns 2D grid coordinates

% Extract model parameter estimates, known as the regression coefficients
% beta's
betas = mdl2.Coefficients.Estimate;
mu = betas(1,1);
beta2 = betas(2,1);
beta3 = betas(3,1);
beta13 = betas(4,1);

% Evaluate the regression equation at every coordinate within the grid.
% Initalize variable to hold the evaluated equation.
% Main effects only
Zm = nan(nx2,nx3); % This is the variable to hold the value of the response.
for i = 1:nx2 % First loop; i is the index of the grid's row
    for j = 1:nx3 % Second loop; j is the index of the grid's column
        Zm(i,j) = mu + beta2*X2(i,j) + beta3*X3(i,j);
    end
end


% Surface plot
figure;
surfc(X2,X3,Zm); % surfc creates a surface and contour plot
c1 = colorbar;
c1.Label.String = 'Cutting Tool Life';
title('Response Surface for Model');
xlabel('Metal Hardness');
ylabel('Cutting Angle');
zlabel('Tool Life');
```

```
% Contour plot
cl = 40; % Number of contour levels (for increased resolution)
figure;
contour(X2,X3,Zm,cl) % 2D contour plot
title('Contour Plot for Model');
xlabel('Metal Hardness');
ylabel('Cutting Angle');
zlabel('Tool Life');
c2 = colorbar;
c2.Label.String = 'Cutting Tool Life';
```

```
% Mesh
[X1,X3] = meshgrid(x1,x3);

% Evaluate the regression equation at every coordinate within the grid.
% Interaction effects
Zi = nan(nx1,nx3); % This is the variable to hold the value of the response.
for i = 1:nx1 % First loop; i is the index of the grid's row
    for j = 1:nx3 % Second loop; j is the index of the grid's column
        Zi(i,j) = mu + beta3*X3(i,j) + beta13*X1(i,j)*X3(i,j);
    end
end

% Surface plot
figure;
surfc(X1,X3,Zi); % surfc creates a surface and contour plot
c3 = colorbar;
c3.Label.String = 'Cutting Tool Life';
title('Response Surface for Model');
xlabel('Cutting Speed');
ylabel('Cutting Angle');
zlabel('Tool Life');
```

```
% Contour plot
cl = 40; % Number of contour levels (for increased resolution)
figure;
contour(X1,X3,Zi,cl) % 2D contour plot
title('Contour Plot for Model');
xlabel('Cutting Speed');
ylabel('Cutting Angle');
zlabel('Tool Life');
c4 = colorbar;
c4.Label.String = 'Cutting Tool Life';
```

Problem 8.4 - Taste of a soft-drink beverage, a 2^4 full-factorial design

1. Enter Data

```
dFF4 = fullfact([2 2 2 2])
```

```
dFF4 = 16×4
     1     1     1     1
     2     1     1     1
     1     2     1     1
     2     2     1     1
     1     1     2     1
     2     1     2     1
     1     2     2     1
     2     2     2     1
     1     1     1     2
```

```
     2     1     1     2
     ⋮
```

```
% Replace 1s and 2s with -1s and +1s, respectively, to code the design.
% Initialize array to hold coded variables
dFF4c = nan;
% Repetition structure
for i = 1:length(dFF4(:,1)) % From one to number of number of rows in column 1
    for j = 1:length(dFF4(1,:))
        if dFF4(i,j) == 1
            dFF4c(i,j) = -1;
        elseif dFF4(i,j) == 2
            dFF4c(i,j) = 1;
        end
    end
end
dFF4c
```

```
dFF4c = 16×4
    -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
     ⋮
```

```
% Factors
sw = dFF4c(:,1); % Type of sweetner
ra = dFF4c(:,2); % Ratio of syrup to water
cb = dFF4c(:,3); % Carbonation level
tp = dFF4c(:,4); % Temperature
```

2. Response columns

```
% Replicates vectors
score1 = [188;172;179;185;175;183;190;175;200;170;189;183;201;181;189;178]
```

```
score1 = 16×1
   188
   172
   179
   185
   175
   183
   190
   175
   200
   170
     ⋮
```

```
score2 = [195;180;187;178;180;178;180;168;193;178;181;188;188;173;182;182]
```

```
score2 = 16×1
   195
   180
   187
   178
   180
   178
   180
   168
   193
   178
     ⋮
```

```
% Combined replicates column vector
score = [score1;score2];
```

```
score = 32×1
   188
   172
   179
   185
   175
   183
   190
   175
   200
   170
     ⋮
```

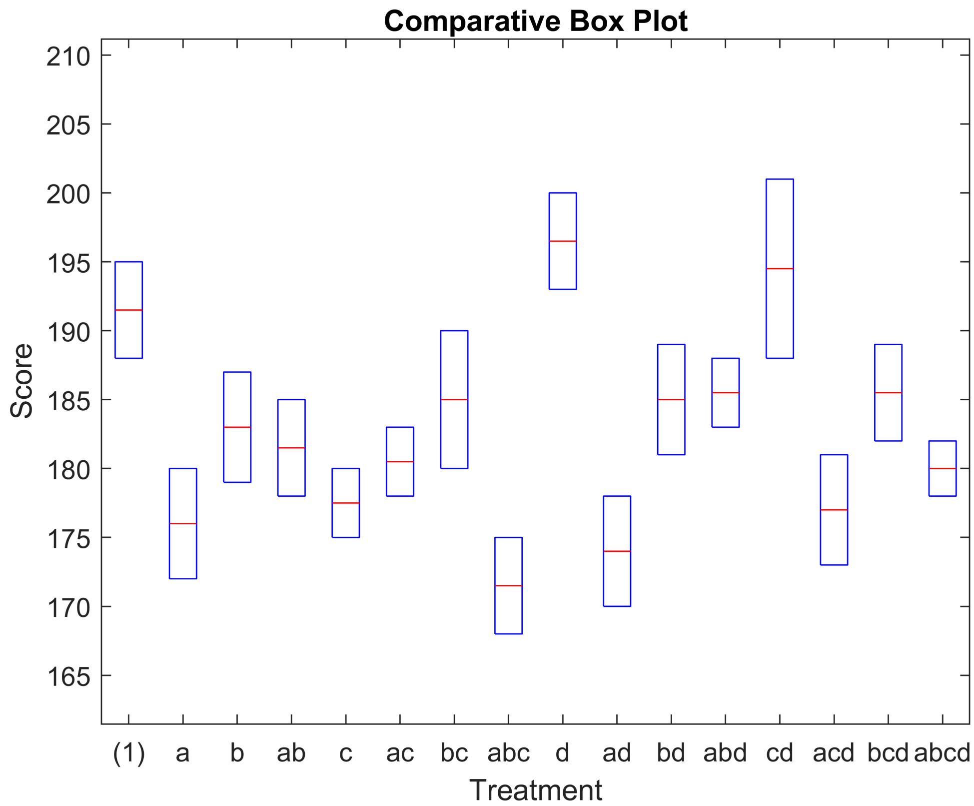
```
% Average
scoreAvg = mean([score1,score2],2)
```

```
scoreAvg = 16×1
  191.5000
  176.0000
  183.0000
  181.5000
  177.5000
  180.5000
  185.0000
  171.5000
  196.5000
  174.0000
     ⋮
```

3. Examine data graphically

```
% Treatment combinations
treat = {'(1)';'a';'b';'ab';'c';'ac';'bc';'abc';'d';'ad';'bd';'abd';'cd';'acd';'bcd';'abcd'}; %
% Create a grouping variable
gtreat = [treat; treat];

boxplot(score,gtreat)
```

```
ylabel('Score');
xlabel('Treatment');
title('Comparative Box Plot');
```



Create main effects plots for the treatment means to visualize the magnitude effect of each factor.

The main effect tells us how much the response will change per unit level change in the factor.

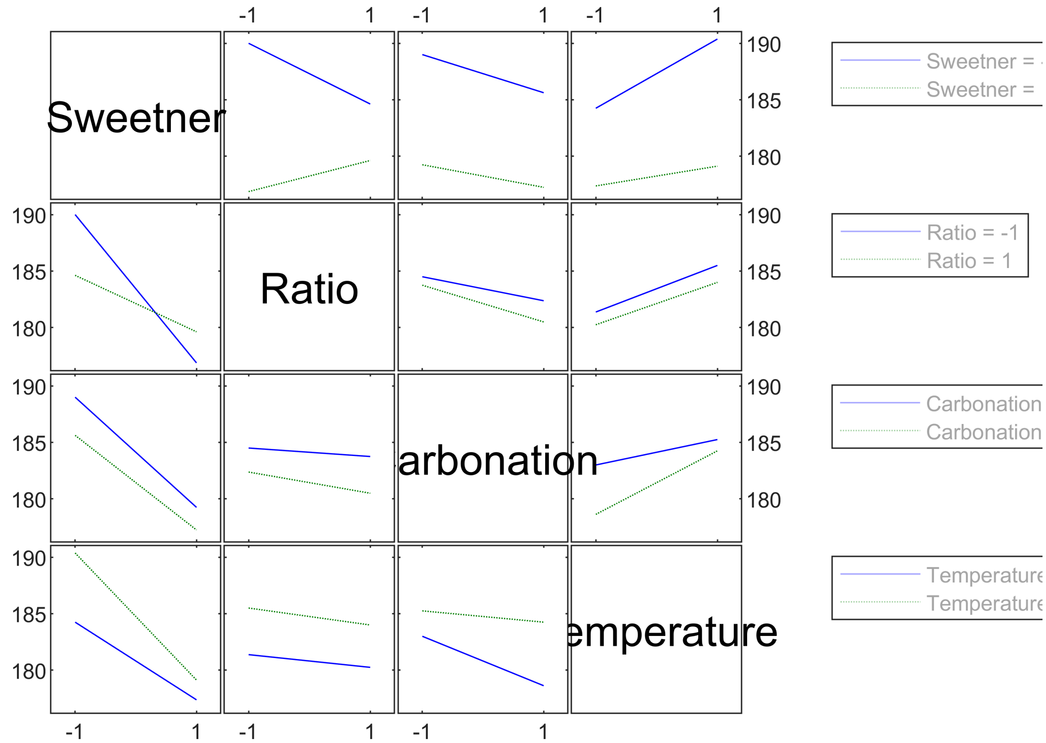
```
maineffectsplot(scoreAvg,{sw,ra,cb,tp},'varnames',{'Sweetner','Ratio','Carbonation','Temperatur
```

From a visual inspection, clearly sweetner has the largest effect on the response as seen by the slope of the line. Temperature also seems to have a moderate effect.

Create interaction plots to viusalize interaction effects.

An interaction occurs when the difference in response between the levels of one factor is not the same at all levels of the other factors.

```
interactionplot(scoreAvg,{sw,ra,cb,tp},'varnames',{'Sweetner','Ratio','Carbonation','Temperatur
```



Clearly there is an interaction between sweetner and ratio. Other less apparent interactions include: sweetner and temperature, carbonation and temperature.

4. Create Fitted Model

The regression model

Create a table to hold the DOE test matrix and response variables.

```
tbl841 = table([sw;sw],[ra;ra],[cb;cb],[tp;tp],[score1;score2],'VariableNames',{'Sweetner','Rat
```

tbl841 = 32×5 table

	Sweetner	Ratio	Carbonation	Temperature	Score
1	-1	-1	-1	-1	188
2	1	-1	-1	-1	172
3	-1	1	-1	-1	179
4	1	1	-1	-1	185
5	-1	-1	1	-1	175
6	1	-1	1	-1	183
7	-1	1	1	-1	190
8	1	1	1	-1	175
9	-1	-1	-1	1	200

	Sweetner	Ratio	Carbonation	Temperature	Score
10	1	-1	-1	1	170
11	-1	1	-1	1	189
12	1	1	-1	1	183
13	-1	-1	1	1	201
14	1	-1	1	1	181
15	-1	1	1	1	189
16	1	1	1	1	178
17	-1	-1	-1	-1	195
18	1	-1	-1	-1	180
19	-1	1	-1	-1	187
20	1	1	-1	-1	178
21	-1	-1	1	-1	180
22	1	-1	1	-1	178
23	-1	1	1	-1	180
24	1	1	1	-1	168
25	-1	-1	-1	1	193
26	1	-1	-1	1	178
27	-1	1	-1	1	181
28	1	1	-1	1	188
29	-1	-1	1	1	188
30	1	-1	1	1	173
31	-1	1	1	1	182
32	1	1	1	1	182

Linear regression model

Recall: Regression models describe the relationship between a response (output) variable, and one or more predictor (input variables).

```
% Perform a linear fit of main effects only
mdl841 = fitlm(tbl841, 'linear','CategoricalVars','Sweetner') % Indicate which factors are cate
```

```
mdl841 = 
Linear regression model:
    Score ~ 1 + Sweetner + Ratio + Carbonation + Temperature

Estimated Coefficients:
                   Estimate      SE       tStat       pValue   
                   ________    ______    _______    __________

    (Intercept)      187.31    1.6064      116.6    4.9371e-38
    Sweetner_1      -9.0625    2.2719     -3.989    0.00045559
```

```
    Ratio          -0.65625    1.1359    -0.57772       0.56824
    Carbonation     -1.3438    1.1359      -1.183       0.24714
    Temperature      1.9688    1.1359      1.7332      0.094474


Number of observations: 32, Error degrees of freedom: 27
Root Mean Squared Error: 6.43
R-squared: 0.433,  Adjusted R-Squared: 0.349
F-statistic vs. constant model: 5.16, p-value = 0.0032
```

```
% Perform a linear fit of main effects and two-way interactions
mdl842 = fitlm(tbl841,'interactions','CategoricalVars','Sweetner')
```

```
mdl842 = 
Linear regression model:
    Score ~ 1 + Sweetner*Ratio + Sweetner*Carbonation + Sweetner*Temperature + Ratio*Carbonation + Ratio*Temperature

Estimated Coefficients:
                               Estimate      SE        tStat        pValue  
                               ________    ______    _________    __________

    (Intercept)                  187.31    1.6504        113.5    9.0593e-31
    Sweetner_1                  -9.0625     2.334      -3.8829     0.0008593
    Ratio                       -2.6875    1.6504      -1.6284       0.11834
    Carbonation                 -1.6875    1.6504      -1.0225       0.31818
    Temperature                  3.0625    1.6504       1.8557      0.077593
    Sweetner_1:Ratio             4.0625     2.334       1.7406      0.096387
    Sweetner_1:Carbonation       0.6875     2.334      0.29456       0.77122
    Sweetner_1:Temperature      -2.1875     2.334     -0.93725       0.35929
    Ratio:Carbonation          -0.28125     1.167     -0.24101       0.81189
    Ratio:Temperature          -0.09375     1.167    -0.080336       0.93673
    Carbonation:Temperature     0.84375     1.167      0.72302       0.47764


Number of observations: 32, Error degrees of freedom: 21
Root Mean Squared Error: 6.6
R-squared: 0.535,  Adjusted R-Squared: 0.313
F-statistic vs. constant model: 2.41, p-value = 0.0426
```

ANOVA

To examine the quality of the fitted model, consult an ANOVA table.

Recall: ANOVA is used to test H0 (null hypothesis): mu1 = mu2 = mu3 = mu4 = mu5 = mu6 = mu7 = mu8; meaning the mean of each treatment is equal, i.e. there is no difference among the treatments and therefore none of the factors are having an effect on the response.

The null hypothesis is tested against the alternative hypothesis H1: mu1 dne mu2 dne mu3 dne mu4 dne mu5 dne mu6 dne mu7 dne mu8 (dne means 'does not equal'); i.e. some means are different, and therefore some factors ARE having an effect on the response.

```
tbl842 = anova(mdl842)
```

tbl842 = 11×5 table

	SumSq	DF	MeanSq	F	pValue
1 Sweetner	657.0313	1	657.0313	15.0768	0.0009
2 Ratio	13.7813	1	13.7813	0.3162	0.5798

	SumSq	DF	MeanSq	F	pValue
3 Carbonation	57.7813	1	57.7813	1.3259	0.2625
4 Temperature	124.0312	1	124.0312	2.8461	0.1064
5 Sweetner:Ratio	132.0313	1	132.0313	3.0297	0.0964
6 Sweetner:Carbonation	3.7813	1	3.7813	0.0868	0.7712
7 Sweetner:Temperature	38.2812	1	38.2812	0.8784	0.3593
8 Ratio:Carbonation	2.5312	1	2.5312	0.0581	0.8119
9 Ratio:Temperature	0.2812	1	0.2812	0.0065	0.9367
10 Carbonation:Temperature	22.7812	1	22.7812	0.5228	0.4776
11 Error	915.1562	21	43.5789	1.0000	0.5000

ANOVA gives us two important estimates of variance (sigma-squared):

1. The inherent variability within factors is given by MSE (Mean sum of squares due to error within factors).
2. The variability between factors is given by MS factor (Mean sum of squares due to factor).

If there are no differences in the factor means (H0), the two estimates MSE and MS factor should be very similar.

*** See handwritten notes dated 10/12/20 - 10/13/20 ***

If MSE and MS factor are not close in value (the difference is large), then we suspect that the observed difference must be caused by differences in the factor means (H1).

Or,

If MS factor >> MSE, we can reject the null hypothesis (H0), and therefore inferring the factor is statistically significant (having an effect on the response).

From the ANOVA table we conclude:

Sweetner, Temperature, and Sweetner*Ratio are statistically significant.

Regression on statistically significant terms only.

To obtain a more robust model we iterate the regression analysis procedure on only the statistically significant terms.

```
% Or create a new model with only statistically significant terms
mdl843 = fitlm(tbl841,'Score ~ Sweetner + Temperature + Sweetner:Ratio','CategoricalVars','Swee
```

```
mdl843 =
Linear regression model:
    Score ~ 1 + Sweetner + Temperature + Sweetner:Ratio

Estimated Coefficients:
                   Estimate     SE       tStat       pValue
                   ________   ______    _______    __________

    (Intercept)     187.31    1.6065     116.6     3.5898e-39
```

```
Sweetner_1           -9.0625     2.2719     -3.989    0.00043278
Temperature           1.9687     1.1359     1.7332      0.094071
Sweetner_1:Ratio       1.375     1.6065    0.85592       0.39931


Number of observations: 32, Error degrees of freedom: 28
Root Mean Squared Error: 6.43
R-squared: 0.412,  Adjusted R-Squared: 0.349
F-statistic vs. constant model: 6.55, p-value = 0.0017
```

NOTE: The adjusted R-squared value for mdl843 is higher than the earlier models created, indicating a better fit to the data.

Summary of Fit

R-Square

- Estimates the proportion of variation in the response that can be attributed to the model rather than to random error. Calculated:
- SSmodel / SStotal

R-Square Adj

- Adjusts the R-Square statistic for the number of parameters in the model. R-Square Adj facilitates comparisons among models with different numbers of parameters. Calculated:
- 1 - MSerror / (SStotal / DFtotal)

Root Mean Square Error

- Estimates the standard deviation of the random error.

ANOVA

Prob > F

- The p-value for the test. The Prob > F value measures the probability of obtaining an F Ratio as large as what is observed, given that all parameters except the intercept are zero. Small values of Prob > F indicate that the observed F Ratio is unlikely. Such values are considered evidence that there is at least one significant effect in the model.[1]

Parameter Estimates

Estimate

- The parameter estimates for each term. These are the estimates of the model coefficients (the Beta's).

[1] https://www.jmp.com/support/help/en/15.0/#page/jmp/analysis-of-variance.shtml#ww137405

Problem 8.5.

5. Plot Residuals

Residuals help in checking model quality; these plots help in discovering errors, outliers, or correlations in the model or data.

```
plotResiduals(mdl843)
```

```
plotResiduals(mdl843,'probability')
```

```
plotResiduals(mdl843,'fitted')
```

Residuals histogram plot: shows the range of the residuals and their frequencies. The area of each bar is the relative number of observations. The sum of the bar areas is equal to 1.

The generated histogram plot of the new model does not suggest any problems with model assumptions.

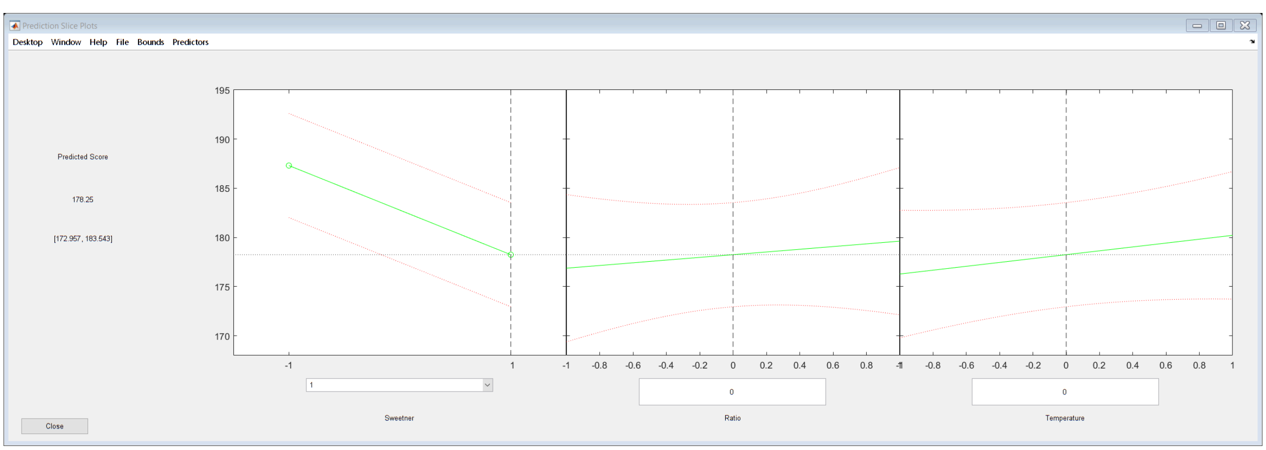
Normal probability plot of residuals: shows how the distribution of the residuals compares to a normal distribution with matched variance. The probability plot seems reasonably straight, meaning a reasonable fit to normally distributed residuals.

- Small deviations from the straight line in a normal probability plot are common, but an "S" shaped curve on this graph suggests a bimodal distribution of residuals. *Breaks near the middle of this graph are also indications of abnormalities in the residual distribution.*

Plot of residuals vs. fitted values: the center-line is the fitted model and the data points are residuals. The spread among the residuals seems to be consistent across the domain of the model, so we can again conclude that this plot does not suggest any problems with model assumptions.

6. Prediction Profiler

```
% Plot a prediction profiler
plotSlice(mdl843)
```



Plot of slices through fitted linear regression surface. Also displays the 95% confidence bounds for the response values.

7. Response by Adjusted Whole Model (Predicted) Plot

```
plot(mdl843)
```

The plot shows the whole model except the constant (intercept) term.

Each 'x' is an observation, the red line is the linear regression line, and the red-dashed lines on either side are the 95% confidence bounds. The slope of the linear regression line is the slope of a fit to the predictors projected onto their best-fitting direction, i.e. the norm of the coefficient vector.

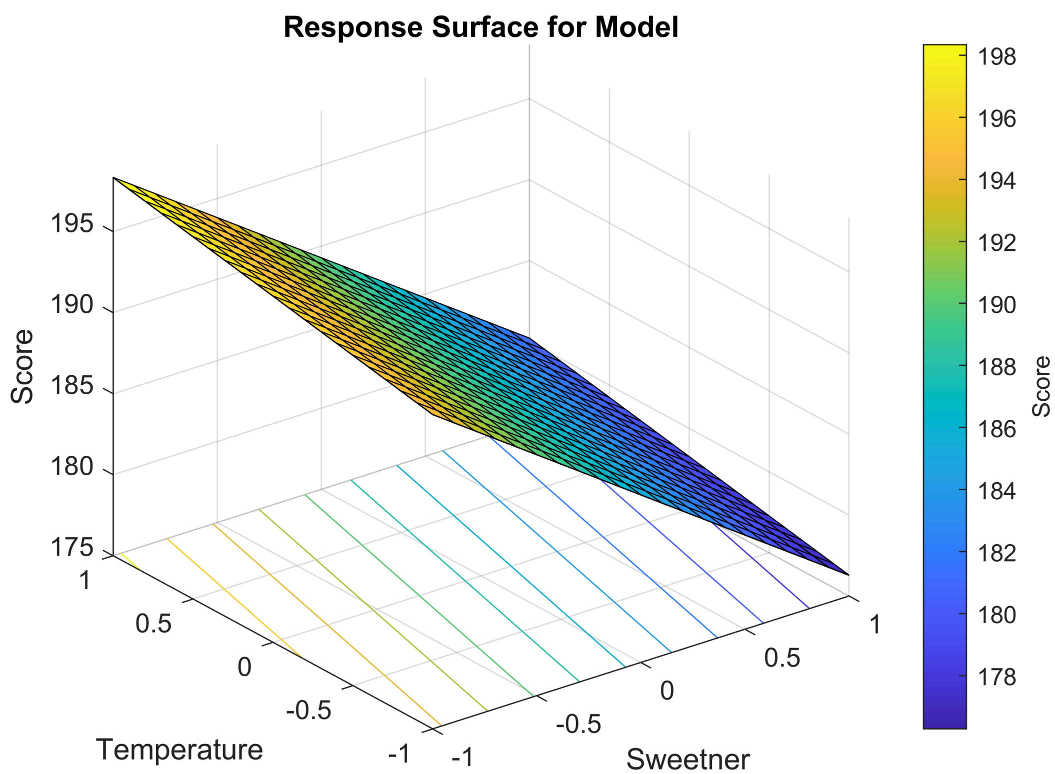
The model as a whole is significant because a 'horizontal line' (the mean the response) does not fit between the confidence bounds.

8. Surface Plot

Main effects only

```
% Create grid
% Nodes
res = 0.1; % Resolution
xsw = [min(sw):res:max(sw)]; % Column vector from low level to high level in increments of reso
xtp = [min(tp):res:max(tp)];
xra = [min(ra):res:max(ra)];

nxsw = length(xsw); % Returns the number of elements in the vector - this becomes
% the nodes on our mesh
nxtp = length(xtp);
nxra = length(xra);

[Xsw,Xtp] = meshgrid(xsw,xtp); % meshgrid function returns 2D grid coordinates

% Extract model parameter estimates, known as the regression coefficients
% beta's
betas = mdl843.Coefficients.Estimate;
mu = betas(1,1); % Mean
betasw = betas(2,1); % Regression coefficient for sweetner factor, beta
betatp = betas(3,1);
betaswra = betas(4,1);

% Evaluate the regression equation at every coordinate within the grid.
% Initalize variable to hold the evaluated equation.
% Main effects only
Zm = nan; % This is the variable to hold the value of the response.
for i = 1:nxsw % First loop; i is the index of the grid's row
    for j = 1:nxtp % Second loop; j is the index of the grid's column
        Zm(i,j) = mu + betasw*Xsw(i,j) + betatp*Xtp(i,j);
```

```
    end
end


% Surface plot
figure;
surfc(Xsw,Xtp,Zm); % surfc creates a surface and contour plot
c1 = colorbar;
c1.Label.String = 'Score';
title('Response Surface for Model');
xlabel('Sweetner');
ylabel('Temperature');
zlabel('Score');
```

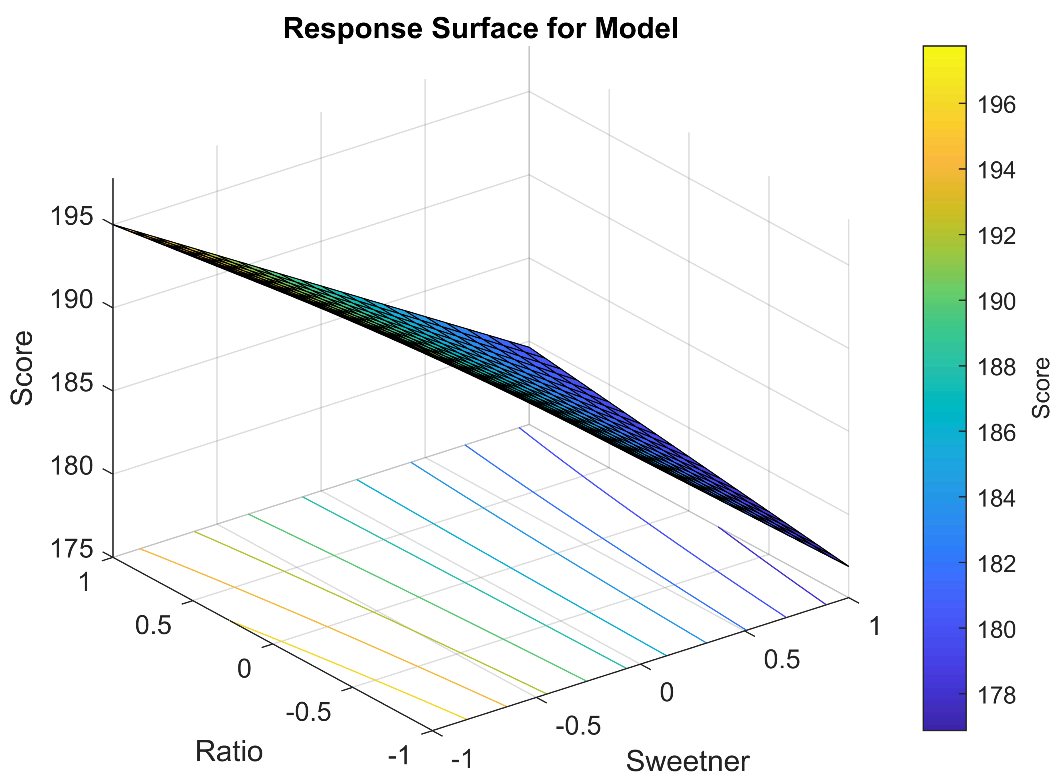



```

% Contour plot
cl = 40; % Number of contour levels (for increased resolution)
figure;
contour(Xsw,Xtp,Zm,cl) % 2D contour plot
title('Contour Plot for Model');
xlabel('Sweetner');
ylabel('Temperature');
zlabel('Score');
c2 = colorbar;
c2.Label.String = 'Score';
```

The surface plot helps in drawing the conclusion that in order to maximize score, the -1 type of sweetner should be used and the temperature should be set to the high level.

Interaction effects

```
% Mesh
[Xsw,Xra] = meshgrid(xsw,xra);

% Evaluate the regression equation at every coordinate within the grid.
% Interaction effects
Zi = nan; % This is the variable to hold the value of the response.
for i = 1:nxsw % First loop; i is the index of the grid's row
    for j = 1:nxra % Second loop; j is the index of the grid's column
        Zi(i,j) = mu + betasw*Xsw(i,j) + betaswra*Xsw(i,j)*Xra(i,j);
    end
end

% Surface plot
figure;
surfc(Xsw,Xra,Zi); % surfc creates a surface and contour plot
c3 = colorbar;
c3.Label.String = 'Score';
title('Response Surface for Model');
xlabel('Sweetner');
ylabel('Ratio');
zlabel('Score');
```



```
% Contour plot
cl = 40; % Number of contour levels (for increased resolution)
figure;
contour(Xsw,Xra,Zi,cl) % 2D contour plot
title('Response Surface for Model');
xlabel('Sweetner');
ylabel('Ratio');
zlabel('Score');
c4 = colorbar;
c4.Label.String = 'Score';
```

Additionally, the ratio of syrup to water can be set to anywhere between and including the low and high settings without having an effect on the score, so as long the low sweetner type is used.

Problem 8.17 - 3^3 Full-factorial design

1. Enter Data

```
dFF33 = fullfact([3 3 3])
```

```
dFF33 = 27×3
     1     1     1
     2     1     1
     3     1     1
     1     2     1
     2     2     1
     3     2     1
     1     3     1
     2     3     1
     3     3     1
     1     1     2
     ⋮
```

```
% Replace 1s and 2s with -1s and +1s, respectively, to code the design.
% Initialize array to hold coded variables
dFF33c = nan;
% Repetition structure
for i = 1:length(dFF33(:,1)) % From one to number of number of rows in column 1
    for j = 1:length(dFF33(1,:))
        if dFF33(i,j) == 1
            dFF33c(i,j) = -1;
        elseif dFF33(i,j) == 2
            dFF33c(i,j) = 0;
        elseif dFF33(i,j) == 3
            dFF33c(i,j) = 1;
        end
    end
end
dFF33c
```

```
% Factors
fx1 = dFF33c(:,1); % Factor x1
fx2 = dFF33c(:,2); % Factor x2
fx3 = dFF33c(:,3); % Factor x3
```

```
dFF33c = 25×3
     1     1     1
     0     1     1
     0     1     1
     1     0     1
     0     0     1
     0     0     1
     1     0     1
     0     0     1
     0     0     1
     1     1     0
     ⋮
```

2. Response columns

```
% Response vectors
avgResponse = [24;120.33;213.67;86;136.63;340.67;112.33;256.33;271.67;81;101.67;357;171.33;372;
stdDev = [12.49;8.39;42.83;3.46;80.41;16.17;27.57;4.62;23.63;0;17.67;32.91;15.01;0;92.50;63.50;
```

3. Examine data graphically

```
% Create a vector to hold the treatment number
% Initialize
treatment = zeros(length(dFF33c),1);
iterCount = 0;
for k = 1:length(dFF33c(:,1))
    iterCount = iterCount + 1;
    treatment(k,1) = iterCount;
end

% Box plot
boxplot(avgResponse,treatment)
ylabel('AverageResponse');
xlabel('Treatment');
title('Comparative Box Plot');
```

```
% Scatter Plots
sp1 = scatter(treatment,avgResponse)
```

```
sp1 =
  Scatter with properties:

           Marker: 'o'
  MarkerEdgeColor: 'flat'
  MarkerFaceColor: 'none'
         SizeData: 36
        LineWidth: 0.5000
```

```
           XData: [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27]
           YData: [24 120.3300 213.6700 86 136.6300 340.6700 112.3300 256.3300 271.6700 81 101.6700 357 171.3300
           ZData: [1×0 double]
           CData: [0 0.4470 0.7410]

  Show all properties
```

```
ylabel('Average Response');
xlabel('Treatment');
title('Scatter Plot of Average Response');
```

```
sp2 = scatter(treatment,stdDev)
```

```
sp2 = 
  Scatter with properties:

             Marker: 'o'
    MarkerEdgeColor: 'flat'
    MarkerFaceColor: 'none'
           SizeData: 36
          LineWidth: 0.5000
              XData: [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27]
              YData: [12.4900 8.3900 42.8300 3.4600 80.4100 16.1700 27.5700 4.6200 23.6300 0 17.6700 32.9100 15.0100
              ZData: [1×0 double]
              CData: [0 0.4470 0.7410]

  Show all properties
```

```
ylabel('Standard Deviation');
xlabel('Treatment');
title('Scatter Plot of Standard Deviation');
```

Create main effects plots for the treatment means to visualize the magnitude effect of each factor.

The main effect tells us how much the response will change per unit level change in the factor.

```
maineffectsplot(avgResponse,{fx1,fx2,fx3},'varnames',{'Factorx1','Factorx2','Factorx3'})
```

From visual inspection, all factors have an effect, with Factor x1 appearing to have the largest effect on the average response.

Create interaction plots to viusalize interaction effects.

An interaction occurs when the difference in response between the levels of one factor is not the same at all levels of the other factors.

```
interactionplot(avgResponse,{fx1,fx2,fx3},'varnames',{'Factorx1','Factorx2','Factorx3'});
```

Interactions appear to minimal across all factors.

4. Create Fitted Model

The regression model

Create a table to hold the DOE test matrix and response variables.

```
tbl8171 = table(fx1,fx2,fx3,avgResponse,'VariableNames',{'Factorx1','Factorx2','Factorx3','AvgR
```

tbl8171 = 27×4 table

	Factorx1	Factorx2	Factorx3	AvgResponse
1	-1	-1	-1	24.0000
2	0	-1	-1	120.3300
3	1	-1	-1	213.6700
4	-1	0	-1	86.0000
5	0	0	-1	136.6300
6	1	0	-1	340.6700
7	-1	1	-1	112.3300
8	0	1	-1	256.3300
9	1	1	-1	271.6700
10	-1	-1	0	81.0000
11	0	-1	0	101.6700
12	1	-1	0	357.0000
13	-1	0	0	171.3300
14	0	0	0	372
15	1	0	0	501.6700
16	-1	1	0	264
17	0	1	0	427
18	1	1	0	730.6700
19	-1	-1	1	220.6700
20	0	-1	1	239.6700
21	1	-1	1	422
22	-1	0	1	199
23	0	0	1	485.3300
24	1	0	1	673.6700
25	-1	1	1	176.6700
26	0	1	1	501

	Factorx1	Factorx2	Factorx3	AvgResponse
27	1	1	1	1010

Linear regression model

Recall: Regression models describe the relationship between a response (output) variable, and one or more predictor (input variables).

Part a.

```
% Perform a quadratic fit to the response
 mdl8171 = fitlm(tbl8171, 'quadratic')
```

```
mdl8171 = 
Linear regression model:
    AvgResponse ~ 1 + Factorx1*Factorx2 + Factorx1*Factorx3 + Factorx2*Factorx3 + Factorx1^2 + Factorx2^2 + Factorx3

Estimated Coefficients:
                         Estimate      SE       tStat       pValue  
                         ________    ______    ________    _________

    (Intercept)           327.62     38.758      8.4532    1.7101e-07
    Factorx1                 177     17.941      9.8656    1.8872e-08
    Factorx2              109.43     17.941      6.0991    1.1803e-05
    Factorx3              131.47     17.941      7.3276    1.1808e-06
    Factorx1:Factorx2     66.028     21.973      3.0049     0.0079707
    Factorx1:Factorx3     75.471     21.973      3.4346     0.0031616
    Factorx2:Factorx3     43.583     21.973      1.9835      0.063709
    Factorx1^2            32.006     31.075      1.0299       0.31747
    Factorx2^2           -22.384     31.075    -0.72033       0.48111
    Factorx3^2           -29.058     31.075    -0.93508       0.36284


Number of observations: 27, Error degrees of freedom: 17
Root Mean Squared Error: 76.1
R-squared: 0.927,  Adjusted R-Squared: 0.888
F-statistic vs. constant model: 23.9, p-value = 6.03e-08
```

Part b.

```
tbl8172 = table(fx1,fx2,fx3,stdDev,'VariableNames',{'Factorx1','Factorx2','Factorx3','StdDeviat
```

tbl8172 = 27×4 table

	Factorx1	Factorx2	Factorx3	StdDeviation
1	-1	-1	-1	12.4900
2	0	-1	-1	8.3900
3	1	-1	-1	42.8300
4	-1	0	-1	3.4600
5	0	0	-1	80.4100
6	1	0	-1	16.1700
7	-1	1	-1	27.5700
8	0	1	-1	4.6200
9	1	1	-1	23.6300

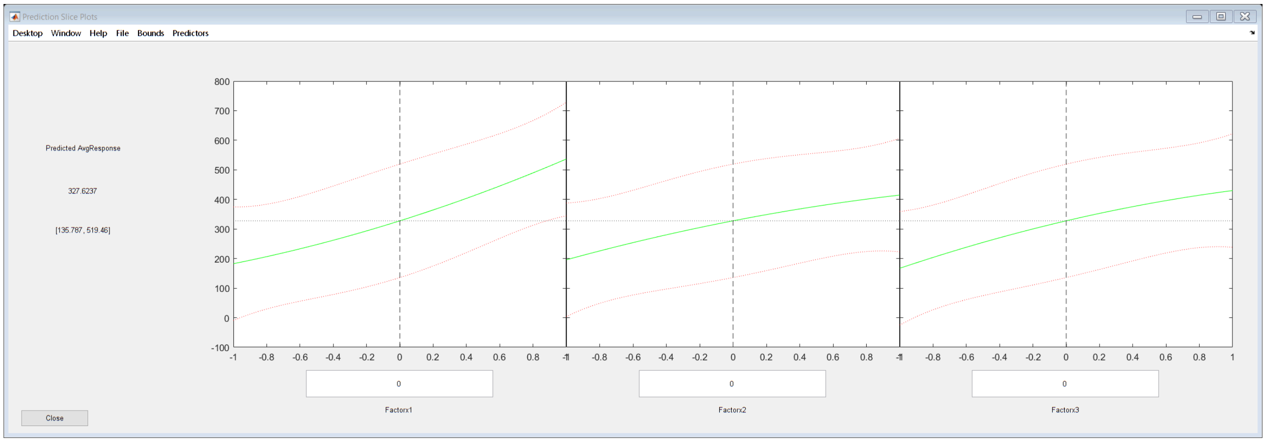
	Factorx1	Factorx2	Factorx3	StdDeviation
10	-1	-1	0	0
11	0	-1	0	17.6700
12	1	-1	0	32.9100
13	-1	0	0	15.0100
14	0	0	0	0
15	1	0	0	92.5000
16	-1	1	0	63.5000
17	0	1	0	88.6100
18	1	1	0	21.0800
19	-1	-1	1	133.8200
20	0	-1	1	23.4600
21	1	-1	1	18.5200
22	-1	0	1	29.4400
23	0	0	1	44.6700
24	1	0	1	158.2100
25	-1	1	1	55.5100
26	0	1	1	138.9400
27	1	1	1	142.4500

```
mdl8172 = fitlm(tbl8172,'linear')
```

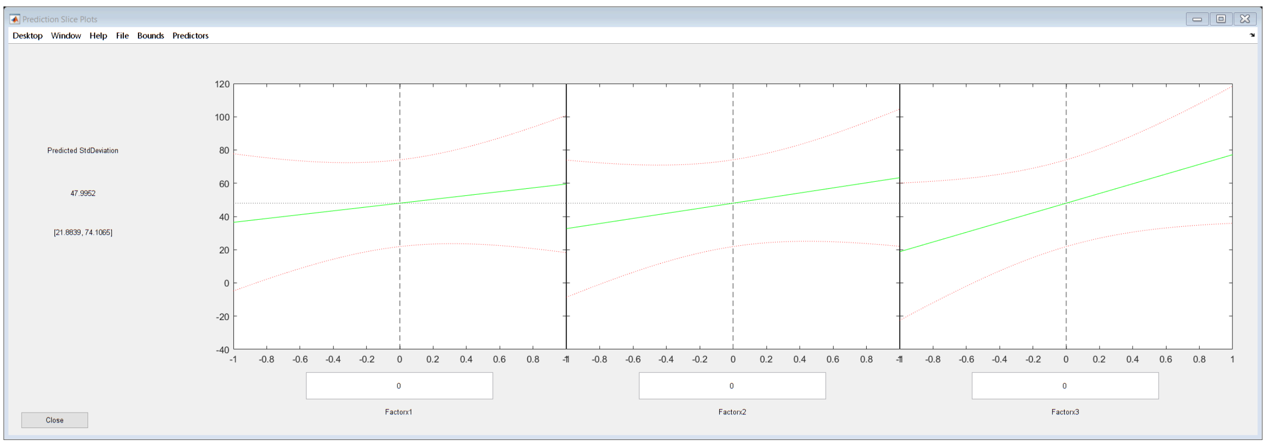
```
mdl8172 =
Linear regression model:
    StdDeviation ~ 1 + Factorx1 + Factorx2 + Factorx3

Estimated Coefficients:
                   Estimate      SE       tStat       pValue
                   ________    ______    ______    __________

    (Intercept)     47.995     7.8085    6.1466    2.8587e-06
    Factorx1        11.528     9.5634    1.2054        0.2403
    Factorx2        15.323     9.5634    1.6023       0.12274
    Factorx3        29.192     9.5634    3.0524     0.0056483


Number of observations: 27, Error degrees of freedom: 23
Root Mean Squared Error: 40.6
R-squared: 0.367,  Adjusted R-Squared: 0.284
F-statistic vs. constant model: 4.45, p-value = 0.0132
```

Part c.

```
plotSlice(mdl8171)
```




```
plotSlice(mdl8172)
```



Based on the analysis of the plot slice profiles, setting x1 = ~ 0.90, x2 = ~ 0, x3 = -0.07 will produce an avg. response equal to 500 while keeping the std. dev at a minimum.