hw5

March 30, 2015

1 STAT 541 HW5

1.1 Problem 2 (Problem 5.4 page 225)

```
In [19]: # setup
         import pandas as pd
         import numpy as np
         import statsmodels.api as sm
         from statsmodels.formula.api import ols
         import matplotlib.pyplot as plt
         # from scipy import stats

         depth_levels = ['0.15', '0.18', '0.20', '0.25']
         feed_levels = ['0.20', '0.25', '0.30']
         metal_data = pd.DataFrame({'depth' : np.tile(np.repeat(depth_levels,3),3),
                                    'feed_rate' : np.repeat(feed_levels, 12),
                            'finish' : [74, 64, 60, 79, 68, 73, 82, 88, 92, 99, 104, 96,
                                        92, 86, 88, 98, 104, 88, 99, 108, 95, 104, 110, 99,
                                        99, 98, 102, 104, 99, 95, 108, 110, 99, 114, 111, 107]})
```

1.1.1 Part a) initial analysis

For the model

$$y_{ijk} = \beta_i + \tau_j + \gamma_{ij} + \epsilon_{ijk} \qquad 1 \le i \le 4,\ 1 \le j \le 3,\ 1 \le k \le 3$$

$$\epsilon_{ijk} \overset{iid}{\sim} N(0, \sigma^2)$$

the hypotheses being tested are:

$$H_0 : \gamma_{ij} = \gamma_{kl} \quad \text{for all} \quad i, j, k, l$$

$$H_a : \gamma_{ij} \neq \gamma_{kl} \quad \text{for some} \quad i, j, k, l$$

And ultimately for main effects:

$$H_0 : \beta_i = \beta_j \quad \text{for all} \quad i, j$$

$$H_a : \beta_i \neq \beta_j \quad \text{for some} \quad i, j$$

$$H_0 : \tau_i = \tau_j \quad \text{for all} \quad i, j$$

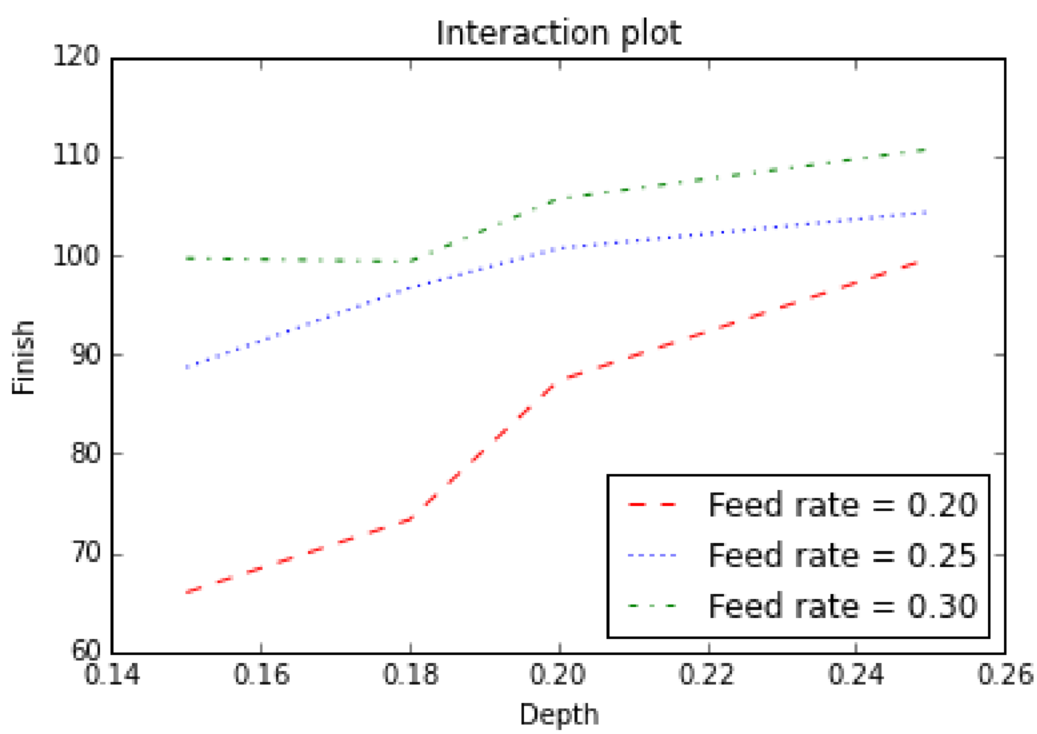
$$H_a : \tau_i \neq \tau_j \quad \text{for some} \quad i, j$$

```
In [20]: metal_model = ols('finish ~ C(depth, Sum) * C(feed_rate, Sum)', metal_data).fit()
         sm.stats.anova_lm(metal_model, typ=2)
```

```
Out[20]:                                    sum_sq  df          F        PR(>F)
         C(depth, Sum)                 2125.111111   3  24.662798  1.652000e-07
         C(feed_rate, Sum)             3160.500000   2  55.018375  1.086046e-09
         C(depth, Sum):C(feed_rate, Sum)  557.055556   6   3.232431  1.797302e-02
         Residual                       689.333333  24        NaN           NaN
```

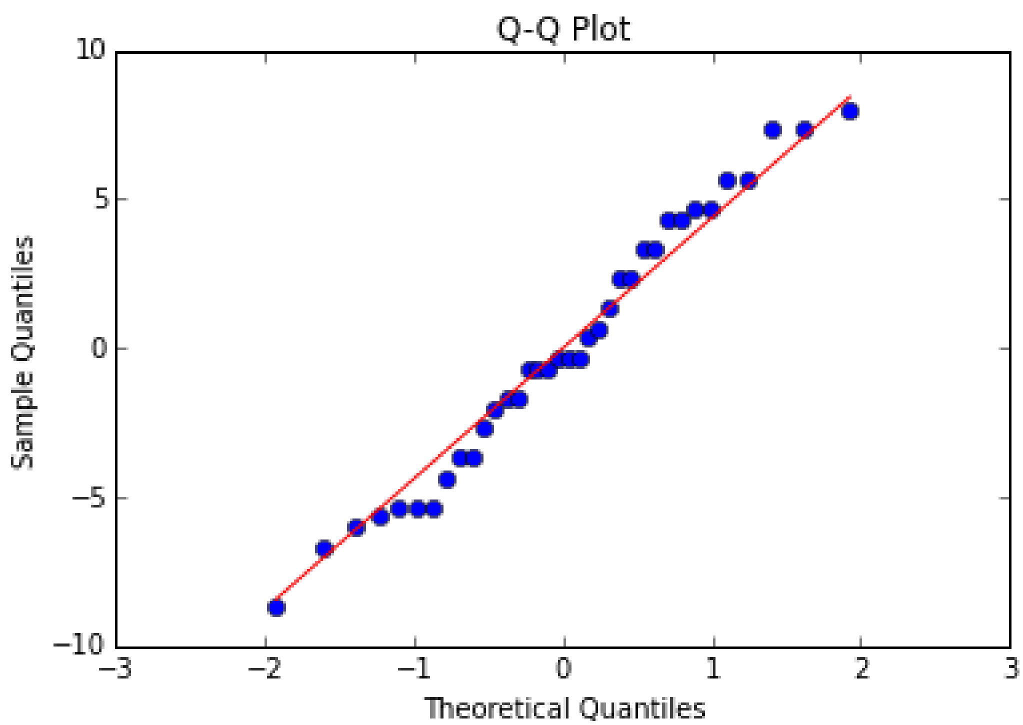
It looks like both of the main effects and the interactions are significant at the $\alpha = .05$ level as all p-values are pretty small. The p-value for the interaction, which we inspect first, is 0.01797. The p-values for the depth and feed rate main effect terms are both less than .0001 indicating that all terms are significant. This is justified by the interaction plot generated below. We can see that as depth increases (along the x-axis) there is a general increase in finish, regardless of feed rate. As feed rate increases, the line representing the data goes up. Thus it seems that both main effects are significant. Moreover, as the lines are not parallel, we can assume that there is an interaction. It appears that as depth increases, feed rate's effect on finish isn't as large.

```
In [21]: by_feed_depth = metal_data.groupby(['feed_rate', 'depth']).mean()
         test1 = by_feed_depth.xs('0.20', level='feed_rate')
         test2 = by_feed_depth.xs('0.25', level='feed_rate')
         test3 = by_feed_depth.xs('0.30', level='feed_rate')

         plt.plot(test1.index.values, test1, 'r--', test2.index.values, test2, 'b:', test2.index.values

         import matplotlib.lines as mlines
         red_line = mlines.Line2D([], [], color='red', linestyle = '--', label='Feed rate = 0.20')
         blue_line = mlines.Line2D([], [], color='blue', linestyle = ':', label='Feed rate = 0.25')
         green_line = mlines.Line2D([], [], color='green', linestyle = '-.', label='Feed rate = 0.30')
         plt.legend(handles=[red_line, blue_line, green_line], loc=0)
         plt.title('Interaction plot')
         plt.xlabel('Depth')
         plt.ylabel('Finish')
         plt.show()
```



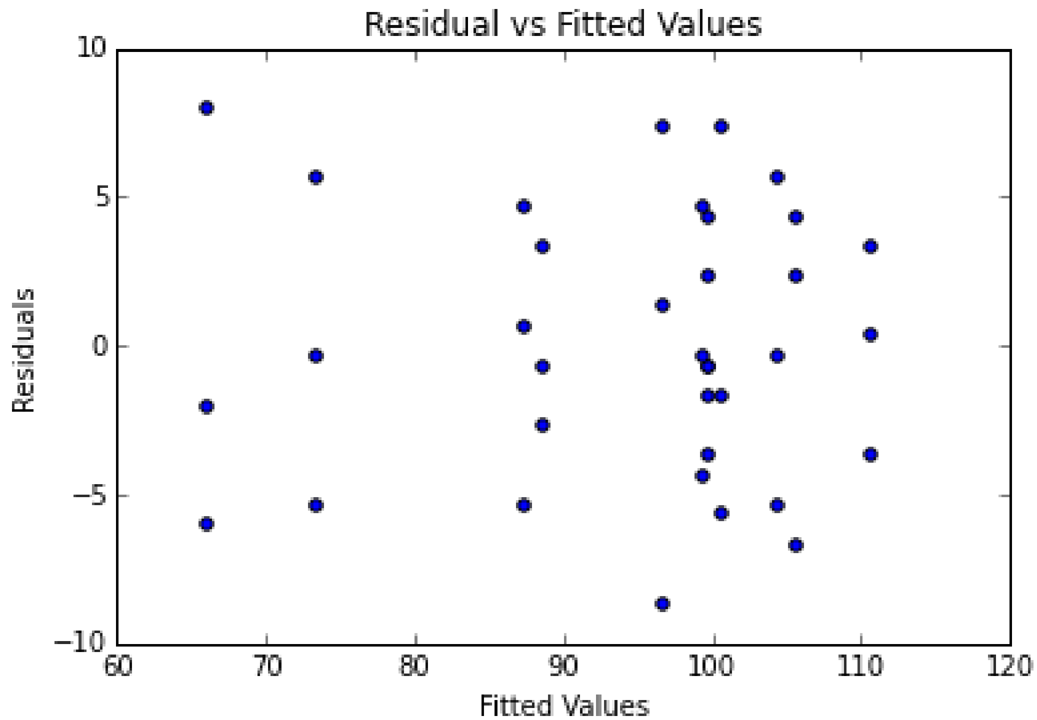
1.1.2 Part b) Residual Plots

In [22]:
```
import matplotlib.pyplot as plt
%matplotlib inline

resid = metal_model.resid
fitted = metal_model.fittedvalues

sm.qqplot(resid, line='s')
plt.title('Q-Q Plot')
plt.show()
```



Because all the points appear to lie close to the diagonal line, there is no reason to suspect a problem with the residuals being normally distributed.

In [23]:
```
plt.scatter(fitted, resid)
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Residual vs Fitted Values')
plt.show()
```



It appears that there is no problem with homogeneity of variance as the residuals appear to be consistently spread out across the different fitted values.

1.1.3 Part c) Estimates

```
In [44]: pd.DataFrame({'Estimate': metal_model.params})
```

```
Out[44]:
                                                          Estimate
         Intercept                                       94.333333
         C(depth, Sum)[S.0.15]                           -9.555556
         C(depth, Sum)[S.0.18]                           -4.555556
         C(depth, Sum)[S.0.20]                            3.555556
         C(feed_rate, Sum)[S.0.20]                      -12.750000
         C(feed_rate, Sum)[S.0.25]                        3.250000
         C(depth, Sum)[S.0.15]:C(feed_rate, Sum)[S.0.20]  -6.027778
         C(depth, Sum)[S.0.18]:C(feed_rate, Sum)[S.0.20]  -3.694444
         C(depth, Sum)[S.0.20]:C(feed_rate, Sum)[S.0.20]   2.194444
         C(depth, Sum)[S.0.15]:C(feed_rate, Sum)[S.0.25]   0.638889
         C(depth, Sum)[S.0.18]:C(feed_rate, Sum)[S.0.25]   3.638889
         C(depth, Sum)[S.0.20]:C(feed_rate, Sum)[S.0.25]  -0.472222
```

It doesn't print out every coefficient estimate, but because it is assumed that the sum of effects is 0, it can be easily computed that the - depth = 0.25 estimate is

$$\beta_4 = -(-9.56 + -4.56 + 3.56) = 10.54$$

- feed rate = 0.30 estimate is

$$\tau_3 = -(-12.75 + 3.25) = 9.5$$

- depth = 0.25, feed rate = 0.20 interaction estimate is

$$\gamma_{4,1} = -(-6.03 + -3.69 + 2.19) = 7.53$$

- depth = 0.25, feed rate = 0.25 interaction estimate is

$$\gamma_{4,2} = -(0.64 + 3.64 + -0.47) = -3.81$$

- depth = 0.15, feed rate = 0.30 interaction estimate is

$$\gamma_{1,3} = -(-6.03 + 0.64) = 5.39$$

- depth = 0.18, feed rate = 0.30 interaction estimate is

$$\gamma_{2,3} = -(-3.69 + 3.64) = 0.05$$

- depth = 0.20, feed rate = 0.30 interaction estimate is

$$\gamma_{3,3} = -(2.19 + -0.47) = -1.72$$

- depth = 0.25, feed rate = 0.30 interaction estimate is

$$\gamma_{4,3} = -(7.53 + -3.81) = -3.72$$

1.1.4 Part d) p-values

P-values for the tests are given in Part a) above.

1.2 Problem 3

In [28]:
```
tablet_data = pd.DataFrame({'tablet_type' : np.tile((np.repeat(['MS', 'LP'], 10)), 3),
                            'mesh_granule_size' : np.repeat(['12', '16', '20'],20),
                            'disintegration_time' :
                  [56.3, 61.1, 60.9, 53.8, 59.3, 56.7, 60.8, 55.9, 60.9, 55.1,
                   57.3, 61.8, 60.8, 63.5, 60.6, 58.7, 56.5, 54.1, 64.2, 60.8,
                   62.1, 63.9, 67.5, 65.7, 65.9, 61.9, 62.2, 65.2, 70.2, 65.7,
                   63.6, 62.0, 64.4, 63.1, 69.5, 68.6, 61.8, 72.1, 60.7, 67.1,
                   69.5, 69.8, 70.6, 68.6, 66.3, 64.5, 66.8, 66.1, 71.7, 66.5,
                   70.8, 74.7, 72.3, 73.6, 73.0, 67.1, 75.8, 72.7, 70.1, 68.0]})

tablet_model = ols('disintegration_time ~ tablet_type * mesh_granule_size', tablet_data).fit()
anova_table = sm.stats.anova_lm(tablet_model, typ=2)
anova_table
```

Out[28]:

	sum_sq	df	F	PR(>F)
tablet_type	55.680667	1	6.346313	1.475516e-02
mesh_granule_size	1210.321000	2	68.974349	1.344720e-15
tablet_type:mesh_granule_size	31.034333	2	1.768599	1.803085e-01
Residual	473.780000	54	NaN	NaN

1.3 Problem 4

We have that

$$SS_A + SS_B + SS_{AB} = 55.68 + 1210.32 + 31.03 = 1297.03$$

which is the sum of squares for the one-way ANOVA on the midterm. This makes sense as the amount of variability between different groups is still the same as in the single factor analysis. What has changes is the partitioning of the variability.

1.4 Problem 5

1. None of the sum of squares is zero
2. $SS_A = 0$
3. $SS_{AB} = 0$
4. $SS_A = SS_{AB} = 0$
5. $SS_A = SS_B = 0$