In [4]:
```
%reset
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
from scipy import stats
import statsmodels.api as sm
```

3J03 MATLS Assign 3 Q1: ANOVA In "Orthogonal Design for Process Optimization and Its Application to Plasma Etching" (Solid State Technology, 1987), G. Z. Yin and D. W. Jillie described an experiment to determine the effect of C2 F6 flow rate on the uniformity of the etch on a silicon wafer used in integrated circuit manufacturing. Three flow rates are used in the experiment, and the resulting uniformity (in percent) for six replicates follows.

C_2F_6 Flow	Uniformity (%)					
125	2.7	3.8	2.6	3.0	3.2	3.8
160	4.9	4.6	5.0	4.2	3.6	4.2
200	4.6	3.4	2.9	3.5	4.1	5.1

Does C2 F6 flow rate affect etch uniformity?

Step 1:

Construct box plots to compare the factor levels and perform the anal ysis of variance. Use α=0.05.

In [7]:
```
# write in the data

OneTwentyFive=pd.Series([],name="OneTwentyFive")
OneSixty=pd.Series([],name="OneSixty")
TwoHundred=pd.Series([],name="TwoHundred")

Total=pd.Series([2.7,3.8,2.6,3.0,3.2,3.8,4.9,4.6,5.0,4.2,3.6,4.2,4.6,3.4,2.9,3.5,4.1,5.1],name
```

In [8]:
```
# create a pandas dataframe of the name of df using pd.concat; axis=1 is for columns

df=pd.concat([OneTwentyFive,OneSixty,TwoHundred],axis='columns')
df
```

Out[8]:

	OneTwentyFive	OneSixty	TwoHundred

In [9]:
```
# Plot the data to visualize: Create a BoxPlot
```

In [10]:
```
# describe the summary data
```

Step 2

Perform an ANOVA and provide an ANOVA table

The ANOVA table decomposes the variance into the following component sum of squares:

- Total sum of squares. The degrees of freedom for this entry is the number of observations minus one.
- Sum of squares for the factor. The degrees of freedom for this entry is the number of levels minus one. The mean square is the sum of squares divided by the number of degrees of freedom.
- Residual sum of squares. The degrees of freedom is the total degrees of freedom minus the factor degrees of freedom. The mean square is the sum of squares divided by the number of degrees of freedom.

The sums of squares summarize how much of the variance in the data (total sum of squares) is accounted for by the factor effect (batch sum of squares) and how much is random error (residual sum of squares). Ideally, we would like most of the variance to be explained by the factor effect.

The ANOVA table provides a formal F test for the factor effect. For our example, we are testing the following hypothesis.

```
H0: All individual batch means are equal
```

$\mu_{1} = \mu_{2} = \mu_{3}$ OR $\tau_{1} = \tau_{2} = \tau_{3} = 0$

```
Ha: At least one batch mean is not equal to the others
```

μ_{1} and/or μ_{2} and/or μ_{3} $\neq 0$ OR $\tau_{i} \neq 0$ for at least one i

Thus, if the null hypothesis is true, each observation consists of the overall mean μ plus a realization of the random error component εij . This is equivalent to saying that all N observations are taken from a normal distribution with mean μ and variance σ2 . Therefore, if the null hypothesis is true, changing the levels of the factor (treatment effects are defined as deviations form the overall mean) has no effect on the mean response. e.

The linear statistical model is:

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij} \begin{cases} i = 1, 2, \ldots, a \\ j = 1, 2, \ldots, n \end{cases}$$

- Yij is a random variable denoting the (ij)th observation
- u is the overall mean (the mean parameter common to all treatments)
- τ_i is the treatment effect (the parameter associated with the ith treatment)
- ε_{ij} is a random error component

Assume that the errors are normally and independently distributed with mean=0 and variance = σ^2.

TABLE 13.2 Typical Data for a Single-Factor Experiment

Treatment	Observations				Totals	Averages
1	y_{11}	y_{12}	$\cdots$	y_{1n}	$y_{1\cdot}$	$\bar{y}_{1\cdot}$
2	y_{21}	y_{22}	$\cdots$	y_{2n}	$y_{2\cdot}$	$\bar{y}_{2\cdot}$
⋮	⋮	⋮	⋮⋮⋮	⋮	⋮	⋮
a	y_{a1}	y_{a2}	$\cdots$	y_{an}	$y_{a\cdot}$	$\bar{y}_{a\cdot}$
					$y_{\cdot\cdot}$	$\bar{y}_{\cdot\cdot}$

Use Python to run an ANOVA

In python: One-Way ANOVA

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.f_oneway.html

In [ ]:

In [ ]:

Run this ANOVA analysis by hand

Fill out the following ANOVA table:

ANOVA				
SOURCE	**SS**	**dof**	**MS**	**F**
Treatment	$SS_{Treatment} = n\sum_{i=1}^{a}(\bar{y}_i - \bar{y})^2$	a - 1	$MS_{Treatment} = \frac{SS\ Treatment}{dof\ Treatment}$	$\frac{MS_{Treatment}}{MS_{Error}}$
Error	$SS_{Error} = \sum_{i=1}^{a}\sum_{j=1}^{n}(y_{ij} - \bar{y}_i)^2$	a(n-1)	$MS_{Error} = \frac{SS\ Error}{dof\ Error}$	
Total	$SS_{Total} = \sum_{i=1}^{a}\sum_{j=1}^{n}(y_{ij} - \bar{y})^2$	an - 1		

What is a?

What is n?

What is N?

In [21]:
```
# Find the Total SS
```

In [22]:
```
# Find the Tretament SS (the Between SS)
```

In [23]:
```
# Find the Treatment MS
```

In [24]:
```
# Find the Error SS (the Within SS)
```

In [25]:
```
# Find the MS Error
```

The test statistic is the F-statistic, calculate this F-statistic and test it

In [27]:
```
# Find the F statistic
```

F(A=.95)	df1								
df2	1	2	3	4	5	6	7	8	9
1	161.448	199.500	215.707	224.583	230.162	233.986	236.768	238.883	240.543
2	18.513	19.000	19.164	19.247	19.296	19.330	19.353	19.371	19.385
3	10.128	9.552	9.277	9.117	9.013	8.941	8.887	8.845	8.812
4	7.709	6.944	6.591	6.388	6.256	6.163	6.094	6.041	5.999
5	6.608	5.786	5.409	5.192	5.050	4.950	4.876	4.818	4.772
6	5.987	5.143	4.757	4.534	4.387	4.284	4.207	4.147	4.099
7	5.591	4.737	4.347	4.120	3.972	3.866	3.787	3.726	3.677
8	5.318	4.459	4.066	3.838	3.687	3.581	3.500	3.438	3.388
9	5.117	4.256	3.863	3.633	3.482	3.374	3.293	3.230	3.179
10	4.965	4.103	3.708	3.478	3.326	3.217	3.135	3.072	3.020
11	4.844	3.982	3.587	3.357	3.204	3.095	3.012	2.948	2.896
12	4.747	3.885	3.490	3.259	3.106	2.996	2.913	2.849	2.796
13	4.667	3.806	3.411	3.179	3.025	2.915	2.832	2.767	2.714
14	4.600	3.739	3.344	3.112	2.958	2.848	2.764	2.699	2.646
15	4.543	3.682	3.287	3.056	2.901	2.790	2.707	2.641	2.588
16	4.494	3.634	3.239	3.007	2.852	2.741	2.657	2.591	2.538
17	4.451	3.592	3.197	2.965	2.810	2.699	2.614	2.548	2.494
18	4.414	3.555	3.160	2.928	2.773	2.661	2.577	2.510	2.456
19	4.381	3.522	3.127	2.895	2.740	2.628	2.544	2.477	2.423
20	4.351	3.493	3.098	2.866	2.711	2.599	2.514	2.447	2.393
21	4.325	3.467	3.072	2.840	2.685	2.573	2.488	2.420	2.366
22	4.301	3.443	3.049	2.817	2.661	2.549	2.464	2.397	2.342
23	4.279	3.422	3.028	2.796	2.640	2.528	2.442	2.375	2.320

The F statistic is the group mean square divided by the error mean square.

This statistic follows an F distribution with (a-1) and a(n-1) degrees of freedom. For our example, the critical F value (upper tail) for $\alpha = 0.05$, (a-1) = 2, and a(n-1) = 15 is 3.682.

You can find the critical F value in python as well

In [28]:
```
from scipy.stats import f
```

In [29]:
```
# critical F to compare against
```

In [30]:
```
# Find the exact p-value
```

In [31]:
```
# Make a conclusion
```

In [ ]:

In [ ]:

In [ ]: