ANOVA-One Way Analysis of Variance

One-Way Analysis of Variance (ANOVA) is an extension of hypothesis testing for two population means using the t-distribution. The ANOVA allows us to compare more than two populations means, if the following two conditions are satisfied.

1. The populations are normally distributed.
2. The populations' variances are all equal $\left(\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = ... = \sigma_n^2\right)$.

When you do your project on ANOVA you will have to use Minitab to check condition 1 and the rule of thumb for the equality of variance to check condition 2.

The hypothesis testing for the equality of the means is given below.

Step 1. $H_0 : \mu_1 = \mu_2 = \mu_3 = ... = \mu_n$
$H_a : At\ least\ one\ mean\ is\ different$

Step 2. Test statistic: F (Note: This value comes from the ANOVA table)

Step 3. Reject Ho if $F > F_{v_1, v_2; \alpha}$.
OR If the p-value < α . (Note: we are going to use the p-value)

Step 4. Conclusion.

Note: If we fail to reject H_o we conclude that all the means are equal.
If we reject H_o we have to find out which mean or means are different. We have to do pair-wise comparison on the means. This work will be done in Minitab.

Example: We would like to compare the average time that it takes a fire station to respond (the time it takes the fire truck to leave the station) to a phone call. There are four different fire stations in town. The following data was provided over a week's period. Is there a significant difference among the average response time of these four fire stations? Test it at $\alpha = 0.05$.

Station 1	Station 2	Station 3	Station 4
12 min	14 min	19 min	24 min
18 min	12 min	17 min	34 min
	13 min	21 min	

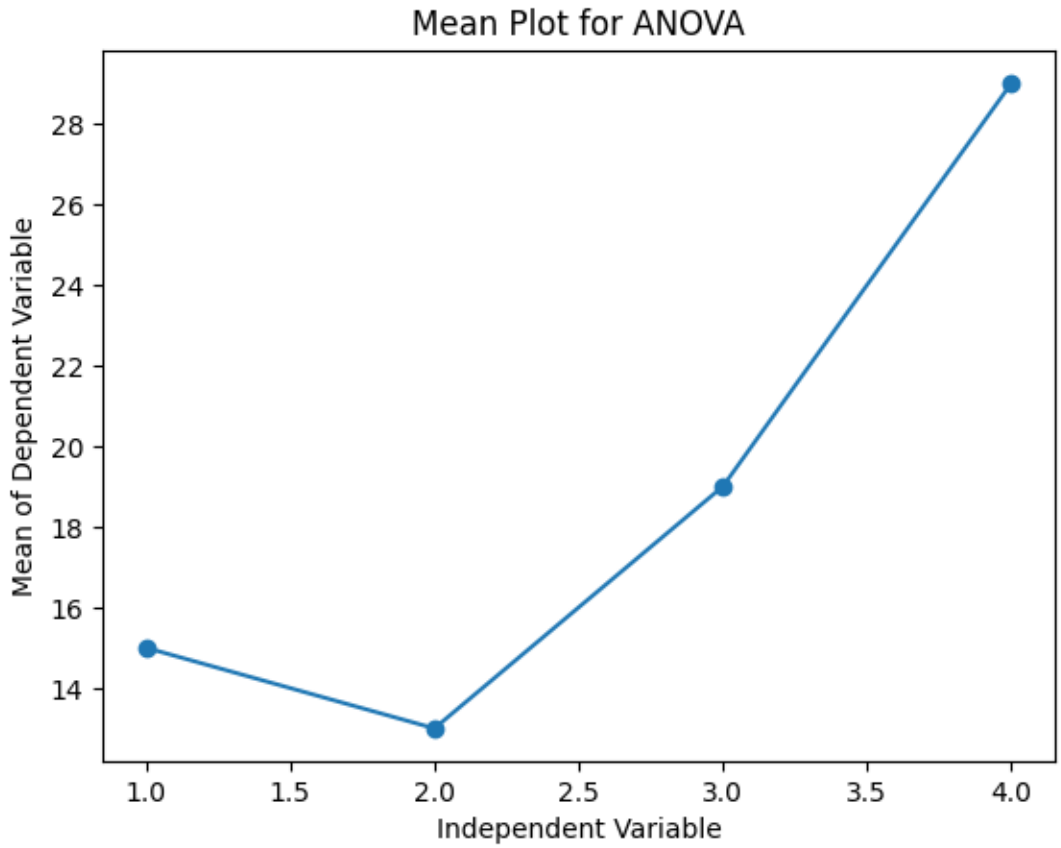
Use Python to do the Hypothesis Testing.

Python Code To Do Analysis of Variance

```
import pandas as pd
print()
print()

#Read the file (replace 'your_file.csv' with the actual filename)
df = pd.read_excel('Firestations.xlsx')
print(df)

print()
print()

from statsmodels.formula.api import ols
import statsmodels.api as sm

# Perform ANOVA
model = ols('Stations ~ Time', data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

# Print the ANOVA table
print(anova_table)
print()
print()

# Assuming 'df' is your DataFrame with 'dependent_variable' and
'independent_variable' columns
means = df.groupby('Stations')['Time'].mean()

import matplotlib.pyplot as plt
plt.plot(means.index, means.values, marker='o', linestyle='-')
plt.xlabel('Independent Variable')
plt.ylabel('Mean of Dependent Variable')
plt.title('Mean Plot for ANOVA')
plt.show()

print()
print()
print(means)
print()

# Perform Tukey's HSD post-hoc test
from statsmodels.stats.multicomp import pairwise_tukeyhsd
tukey_result = pairwise_tukeyhsd(df['Time'], df['Stations'], alpha=0.05)
print(tukey_result)
```

Time	Stations	
0	12	1
1	18	1
2	14	2
3	12	2
4	13	2
5	19	3
6	17	3
7	21	3
8	24	4
9	34	4

	sum_sq	df	F	PR(>F)
Time	6.276544	1.0	11.888927	0.00872
Residual	4.223456	8.0	NaN	NaN



```
Stations

1    15.0
2    13.0
3    19.0
4    29.0
Name: Time, dtype: float64

Please Note: The pair-comparison below are done on
group2 group1
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1group2 meandiff p-adj   lower      upper   reject
-----------------------------------------------------
1      2     -2.0   0.926 -13.3939  9.3939  False ⇒ μ2 = μ1
1      3      4.0  0.6404  -7.3939 15.3939  False ⇒ μ3 = μ1
1      4     14.0  0.0311   1.5186 26.4814   True ⇒ μ4 > μ1
2      3      6.0  0.2726   -4.191  16.191  False ⇒ μ3 = μ2
2      4     16.0  0.0112   4.6061 27.3939   True ⇒ μ4 > μ2
3      4     10.0  0.0822  -1.3939 21.3939  False ⇒ μ4 = μ3
-----------------------------------------------------
```

Homework: 12.63, and 12.65 pages 693-694.

Note: Use Python to do Hypothesis testing and draw a conclusion only.