ANOVA Testing: Visualization with Python

Tiep M. H.

I. DEFINITIONS: SST, SSB, SSW

TABLE I

Group ($M = 4$)	Observations (There are $N = 6$ observations for each group)						Mean	Grand mean
(A)	$a_1 = 7$	$a_2 = 8$	$a_3 = 15$	$a_4 = 11$	$a_5 = 9$	$a_6 = 10$	$\overline{a} = 10$	$\widehat{\mu} = \frac{\overline{a}+\overline{b}+\overline{c}+\overline{d}}{M} \approx 15.9583$
(B)	$b_1 = 12$	$b_2 = 17$	$b_3 = 13$	$b_4 = 18$	$b_5 = 19$	$b_6 = 15$	$\overline{b} = 15.667$	
(C)	$c_1 = 14$	$c_2 = 18$	$c_3 = 19$	$c_4 = 17$	$c_5 = 16$	$c_6 = 18$	$\overline{c} = 17$	
(D)	$d_1 = 19$	$d_2 = 25$	$d_3 = 22$	$d_4 = 23$	$d_5 = 18$	$d_6 = 20$	$\overline{d} = 21.167$	

Denote SST as the *total sum of squares*. Denote SSB as the *between-group sum of squares*. Denote SSW as the *within-group sum of squares*. We have

$$\text{SST} = \text{SSB} + \text{SSW}. \tag{1}$$

Using the numbers provided in the table, we can calculate the SSB as follows:

$$\begin{aligned}
\text{SSB} &= N \sum_{m=1}^{M} (\widehat{\mu} - \mu_m)^2 \\
&= 6 \times \left[(15.9583 - 10)^2 + (15.9583 - 15.667)^2 + (15.9583 - 17)^2 + (15.9583 - 21.167)^2\right] \\
&\approx 382.8
\end{aligned} \tag{2}$$

where $\mu_m \in \{\overline{a}, \overline{b}, \overline{c}, \overline{d}\}$, denotes the group mean. Meanwhile, the SSW can be calculated as

$$\begin{aligned}
\text{SSW} &= \sum_{n=1}^{N} (a_n - \overline{a})^2 + \sum_{n=1}^{N} (b_n - \overline{b})^2 + \sum_{n=1}^{N} (c_n - \overline{c})^2 + \sum_{n=1}^{N} (d_n - \overline{d})^2 \\
&= 40 + 39.33 + 16 + 34.83 \\
&\approx 130.2
\end{aligned} \tag{3}$$

As a result, the SST is equal to $\text{SST} = 382.8 + 130.2 = 513$.

II. One-Way ANOVA Test

Let us define the two following hypotheses:

Null hypothesis $(H_0) : \mu_1 = \mu_2 = \ldots = \mu_M$

Alternative hypothesis (H_1) : At least one group mean is different from the others.

In order to accept/reject (H_0), we will compare F_0 to $F_{\alpha,M-1,M(N-1)}$. What is F_0 and what is $F_{\alpha,M-1,M(N-1)}$? In the ANOVA, F_0 is the F-test statistic, which can be derived from the SSB and SSW as follows:

$$F_0 = \frac{\text{SSB}/(M-1)}{\text{SSW}/(M(N-1))} = \frac{382.8/(4-1)}{130.2/(4(6-1))} \approx 20 \tag{4}$$

On the other hand, $F_{\alpha,M-1,M(N-1)}$ is deduced from the F distribution that has a confidence level of α and degrees of freedom $(M-1)$ and $M(N-1)$. Note that the p-value is

$$p = 1 - \alpha = \Pr\{F_0 \leq F_{\alpha,M-1,M(N-1)}\} = \text{probability of accepting } H_0,$$

which can be understood as the probability of rejecting (H_0). In short, we use the F distribution to find the value $F_{\alpha,M-1,M(N-1)}$ and then compare it to F_0.

If $F_0 > F_{\alpha,M-1,M(N-1)}$, we can reject **null hypothesis** (H_0) and accept the **alternative hypothesis** (H_1). In the case of $F_0 \leq F_{\alpha,M-1,M(N-1)}$, we can accept (H_0) and reject (H_1).

III. Data Visualization with Python

A. DataFrames and Table Styles

In Python, we use Pandas to store the data in a dataframe. The dataframe can be displayed in the form of a table. But, a question arises: what table style should we use to display the dataframe? There are many table styles in Python, thus it depends on our specific purposes. Since we will apply the one-way ANOVA on the data, we want to display the dataframe in the following table style:

0	A	7
1	A	8
2	A	15
3	A	11
4	A	9
5	A	10
6	B	12
7	B	17
8	B	13
⋮		

Group ($M = 4$)	Observations (There are $N = 6$ observations for each group)					
(A)	$a_1 = 7$	$a_2 = 8$	$a_3 = 15$	$a_4 = 11$	$a_5 = 9$	$a_6 = 10$
(B)	$b_1 = 12$	$b_2 = 17$	$b_3 = 13$	$b_4 = 18$	$b_5 = 19$	$b_6 = 15$
(C)	$c_1 = 14$	$c_2 = 18$	$c_3 = 19$	$c_4 = 17$	$c_5 = 16$	$c_6 = 18$
(D)	$d_1 = 19$	$d_2 = 25$	$d_3 = 22$	$d_4 = 23$	$d_5 = 18$	$d_6 = 20$

Fig. 1. (On the left side): The table style in which the dataframe is displayed looks like this.

B. Python Code

In the following, I will present two ways to enter the data into a list and then convert the list into a dataframe (using `pandas`). Depending on whether the dataframe is displayed in a **suitable** table style or not, I will employ the `pandas.melt` method to re-arrange the dataframe into the suitable table style (as seen in the figure above).

1) Way 1: If I create the following dataframe, then its table style will not be the same as the required table style.

In [1]:
```python
import pandas as pd
data = [{'A': 7, 'B': 12, 'C':14, 'D': 19},
        {'A': 8, 'B': 17, 'C':18, 'D': 25},
        {'A': 15, 'B': 13, 'C':19, 'D': 22},
        {'A': 11, 'B': 18, 'C':17, 'D': 23},
        {'A': 9, 'B': 19, 'C':16, 'D': 18},
        {'A': 10, 'B': 15, 'C':18, 'D': 20}
        ]
df_ = pd.DataFrame(data)
df_
```

Out[1]:

	A	B	C	D
0	7	12	14	19
1	8	17	18	25
2	15	13	19	22
3	11	18	17	23
4	9	19	16	18
5	10	15	18	20

Fig. 2. This dataframe is **not** in the table style that is required.

To obtain a suitable dataframe that has the same table style as Fig. 1, I have to use `pandas.melt` to create the following dataframe.

In [2]:
```python
df = pd.melt(df_.reset_index(),
             value_vars=['A', 'B', 'C', 'D'])
# replace column names
df.columns = ['Group', 'Value']
df
```

Out[2]:

	Group	Value
0	A	7
1	A	8
2	A	15
3	A	11
4	A	9
5	A	10
6	B	12
7	B	17

Fig. 3. Now, the new dataframe is in the suitable table style.

2) Way 2: If I create the following dataframe, then its table style will be directly the same as the required table style.

```
In [1]:  import pandas as pd
         data = [{'Group': 'A', 'Value': 7},
                 {'Group': 'A', 'Value': 8},
                 {'Group': 'A', 'Value': 15},
                 {'Group': 'A', 'Value': 11},
                 {'Group': 'A', 'Value': 9},
                 {'Group': 'A', 'Value': 10},
                 #
                 {'Group': 'B', 'Value': 12},
                 {'Group': 'B', 'Value': 17},
                 {'Group': 'B', 'Value': 13},
                 {'Group': 'B', 'Value': 18},
                 {'Group': 'B', 'Value': 19},
                 {'Group': 'B', 'Value': 15},
                 #
                 {'Group': 'C', 'Value': 14},
                 {'Group': 'C', 'Value': 18},
                 {'Group': 'C', 'Value': 18},
                 {'Group': 'C', 'Value': 17},
                 {'Group': 'C', 'Value': 16},
                 {'Group': 'C', 'Value': 18},
                 #
                 {'Group': 'D', 'Value': 19},
                 {'Group': 'D', 'Value': 25},
                 {'Group': 'D', 'Value': 22},
                 {'Group': 'D', 'Value': 23},
                 {'Group': 'D', 'Value': 18},
                 {'Group': 'D', 'Value': 20}]

         df = pd.DataFrame(data)
         df
```

Fig. 4. This is a direct way to create the required dataframe.

As a result, it is seen that the dataframe has the required table style.

Out[1]:

	Group	Value
0	A	7
1	A	8
2	A	15
3	A	11
4	A	9
5	A	10
6	B	12

Fig. 5. The required table style that is exactly the same as the one in Figure 1.

3) Data Visualization: Now, we can portray the data distribution using `seaborn.boxplot` as can be seen below.

In [2]:
```
# Use boxplot to see the data distribution

import seaborn as sns
ax = sns.boxplot(x='Group', y='Value', data=df, color='#9cd000')
ax = sns.swarmplot(x="Group", y="Value", data=df, color='#7d0000')
```

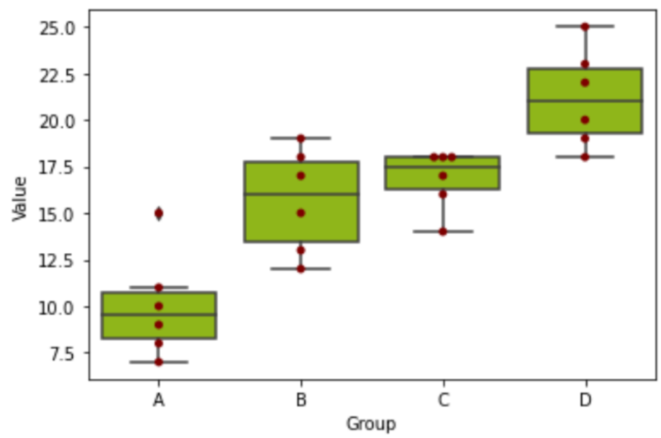

Fig. 6. The data distribution.

IV. One-Way ANOVA Test with Python

To calculate F_0 and p-value from the data, I call the `statsmodels` library and use the following syntax:

In [3]:
```
from statsmodels.formula.api import ols
import statsmodels.api as sm

model = ols('Value ~ Group', data=df).fit()
df_anova = sm.stats.anova_lm(model, typ=2)
df_anova
```

F_0

Out[3]:

	sum_sq	df	F	PR(>F)
Group	380.833333	3.0	19.991251	0.000003
Residual	127.000000	20.0	NaN	NaN

p-value

Before making a decision, I continue to calculate $F_{\alpha,M-1,M(N-1)}$. Assume that $\alpha = 0.01$, then I obtain $F_{\alpha,M-1,M(N-1)} \approx 5$.

In [4]:
```
import scipy.stats
M = 4
N = 6
scipy.stats.f.ppf(q=1-0.01, dfn=M-1, dfd=M*(N-1))
```

$F_{\alpha,M-1,M(N-1)}$

Out[4]:
```
4.938193382310539
```

It is clear that $F_0 \approx 20 > F_{\alpha,M-1,M(N-1)} \approx 5$, I reject the null hypothesis (H_0). Looking at the p-value, it is readily seen that $p \approx 3 \times 10^{-6}$ is too small, which means that the probability of accepting (H_0) is too small. Conclusion is reached! (H_0) is rejected.