In [33]:
```
'''
Thống kê suy diễn
'''
```

Out[33]: '\nThống kê suy diễn\n'

In [34]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [35]:
```
#Đọc File dulieutuyensinh
df = pd.read_csv('../data/dulieuxettuyendaihoc.csv',header=0,delimiter=',',encoding='utf-8')
```

In [36]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 16 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   MSSV               100 non-null    object 
 1   T1                 100 non-null    float64
 2   T2                 100 non-null    float64
 3   T3                 100 non-null    float64
 4   T4                 100 non-null    float64
 5   T5                 100 non-null    float64
 6   T6                 100 non-null    float64
 7   GT                 100 non-null    object 
 8   DT                 3 non-null      float64
 9   KV                 100 non-null    object 
 10  NGONNGU            100 non-null    float64
 11  TOANLOGICPHANTICH  100 non-null    float64
 12  GIAIQUYETVANDE     100 non-null    float64
 13  KT                 100 non-null    object 
 14  NGAYTHI            100 non-null    object 
 15  DINHHUONGNGHENGHIEP  100 non-null    object 
dtypes: float64(10), object(6)
memory usage: 12.6+ KB
```

In [37]:
```
df.head(5)
```

Out[37]:

	MSSV	T1	T2	T3	T4	T5	T6	GT	DT	KV	NGONNGU	TOANLOGICPHANTICH	GIAIQU
0	SV001	7.2	8.4	7.4	7.2	7.4	6.9	F	NaN	2NT	3.25	3.25	
1	SV002	5.4	6.3	4.3	4.9	3.0	4.0	M	NaN	1	6.00	4.00	
2	SV003	5.6	5.0	2.8	6.1	4.8	5.7	M	NaN	1	5.00	6.75	
3	SV004	6.6	5.1	5.9	4.1	6.1	7.4	M	NaN	1	4.25	4.25	
4	SV005	6.0	5.4	7.6	4.4	6.8	8.0	M	NaN	2NT	4.25	4.50	

In [38]:
```
df.describe()
```

Out[38]:

	T1	T2	T3	T4	T5	T6	DT	NGONNG
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.0000	3.000000	100.0000(
mean	5.946000	6.374000	6.383000	6.291000	6.717000	6.9370	2.666667	3.7400(
std	1.608338	1.561443	1.574484	1.469563	1.478059	1.3632	2.886751	1.4244(
min	2.400000	2.800000	2.300000	2.900000	3.000000	3.7000	1.000000	1.0000(
25%	5.000000	5.300000	5.175000	5.300000	5.800000	6.0000	1.000000	2.5000(
50%	5.850000	6.250000	6.650000	6.350000	6.800000	7.1000	1.000000	3.6250(
75%	7.200000	7.525000	7.500000	7.600000	7.800000	8.0000	3.500000	4.7500(
max	9.300000	9.600000	9.500000	9.400000	9.500000	9.5000	6.000000	7.0000(

In [39]:
```
#Đổi tên cột
df = df [['T5','T6','GT','DT','KV','KT','NGONNGU','TOANLOGICPHANTICH','GIAIQUYETVANDE','NGA
df.rename(columns={'TOANLOGICPHANTICH':'LOGIC',
                   'GIAIQUYETVANDE':'UNGXU',
                   'DINHHUONGNGHENGHIEP':'HUONGNGHIEP'},inplace=True)
```

In [40]:
```
df.head(5)
```

Out[40]:

	T5	T6	GT	DT	KV	KT	NGONNGU	LOGIC	UNGXU	NGAYTHI	HUONGNGHIEP
0	7.4	6.9	F	NaN	2NT	A1	3.25	3.25	4.50	12/7/2018	No
1	3.0	4.0	M	NaN	1	C	6.00	4.00	3.50	12/7/2018	Yes
2	4.8	5.7	M	NaN	1	C	5.00	6.75	4.00	12/7/2018	No
3	6.1	7.4	M	NaN	1	D1	4.25	4.25	5.25	12/7/2018	No
4	6.8	8.0	M	NaN	2NT	A	4.25	4.50	5.00	12/7/2018	No

In [41]:
```
'''
Phân tích suy diễn (inferrential statistics)
Lý do tại sao cần suy diễn: Kết luận dựa trên dữ liệu sample (mẫu) nhưng kết luận thì được hiểu là áp dụng cl
Câu hỏi đặt ra: Kết luận trên mẫu đó có phù hợp với tổng thể hay không

Khi phân tích suy diễn cần lưu ý:
1. Xác định giả thuyết H0
2. Các giả định hay các điều kiện về dữ liệu và môi trường để áp dụng kiểm điểm
- Lưu ý về giả định phân phối của biến số: normal, student, poison, chi-square...
3. Thiết lập mức tin cậy và sai lầm (alpha): 90% - 10%, 95% - 5% và 99% - 1%
4. Quy tắc suy diễn:
4.1 Nếu p-value < alpha => reject H0
4.2 Nếu p-value > alpha => accecpt H0
5. Kết luận của suy diễn chỉ cho chúng ta biết là có đủ dữ kiện để kết luận cho tổng thể hay không
 (còn gọi là ý nghĩa thống kê)
'''
```

Out[41]:
```
'\nPhân tích suy diễn (inferrential statistics)\nLý do tại sao cần suy diễn: Kết luận dựa trên dữ liệu sample
(mẫu) nhưng kết luận thì được hiểu là áp dụng cho tổng thể\nCâu hỏi đặt ra: Kết luận trên mẫu đó có phù
hợp với tổng thể hay không\n\nKhi phân tích suy diễn cần lưu ý:\n1. Xác định giả thuyết H0\n2. Các giả đ
ịnh hay các điều kiện về dữ liệu và môi trường để áp dụng kiểm điểm \n- Lưu ý về giả định phân phối của
biến số: normal, student, poison, chi-square...\n3. Thiết lập mức tin cậy và sai lầm (alpha): 90% - 10%, 9
5% - 5% và 99% - 1%\n4. Quy tắc suy diễn: \n4.1 Nếu p-value < alpha => reject H0\n4.2 Nếu p-value > al
pha => accecpt H0\n5. Kết luận của suy diễn chỉ cho chúng ta biết là có đủ dữ kiện để kết luận cho tổng th
ể hay không\n (còn gọi là ý nghĩa thống kê)\n'
```

In [42]:
```
'''
ONE SAMPLE T-TEST
Mục đích: Kiểm định trung bình của 1 biến số (định lượng) co bằng một giá trị
n < 30

'''
# H0: mean = X
```

Out[42]:
```
'\nONE SAMPLE T-TEST\nMục đích: Kiểm định trung bình của 1 biến số (định lượng) co bằng một giá tr
ị\nn < 30\n\n'
```

In [43]:
```
#Là phần mềm nguồn mở miễn phí của Python cho toán học, khoa học và kỹ thuật . Thư viện SciPy được xây

#SciPy là thư viện cơ sở. Nó được xây dựng trên phần mở rộng NumPy. Không cần nhập NumPy nếu bạn đã
# ho hoạt động của các hàm NumPy. SciPy và NumPy cùng là sự lựa chọn tốt nhất cho các hoạt động khoa h
#stats:Các hàm và phân phối thống kê
import scipy.stats as stats
```

Loading [MathJax]/extensions/Safe.js

In [44]:
```
# câu 1: Hãy kiểm ta xem LOGIC thí sinh khối C có bằng 4.0
dfKhoiC = df.loc[df['KT'] == 'C']
dfKhoiC['LOGIC']
```

Out[44]:
```
1     4.00
2     6.75
6     6.75
22    3.50
23    5.25
24    2.25
25    2.00
26    4.50
27    5.00
95    1.50
96    3.75
97    8.00
98    3.50
99    2.50
Name: LOGIC, dtype: float64
```

In [45]:
```
# Với one sample T-test thì giả thiết G0: muy = 4.0
# Mặc định mức tin cậy là 95% và mức sai lầm là 5%
stats.ttest_1samp(dfKhoiC['LOGIC'],popmean=4.0)
```

Out[45]: TtestResult(statistic=0.44599723713991907, pvalue=0.6629370899710998, df=13)

In [46]:
```
#4. Quy tắc suy diễn:
#4.1 Nếu p-value < alpha => reject H0
#4.2 Nếu p-value > alpha => accecpt H0
#5. Kết luận của suy diễn chỉ cho chúng ta biết là có đủ dữ kiện để kết luận cho tổng thể hay không (còn gọi l
'''
# Kết luận:
# Do alpha = 0.05 và p-value = 0.66...
# Suy ra đủ dữ liệu để nói rằng trung bình môn thi LOGIC bằng 4.0
# hay nói cách khác là chấp nhận H0 ở mức sai lầm 5%
'''
```

Out[46]: '\n# Kết luận:\n# Do alpha = 0.05 và p-value = 0.66...\n# Suy ra đủ dữ liệu để nói rằng trung bình môn thi LOGIC bằng 4.0\n# hay nói cách khác là chấp nhận H0 ở mức sai lầm 5%\n'

In [47]:
```
# Điểm trung bình có môn thi UNGXU của khối thi C có bằng 5.0 hay không


stats.ttest_1samp(dfKhoiC['UNGXU'],popmean=5)

# KẾt luận :
# do anpha = 0.05 và p-value =0.4581953822944209
# Suy ra đủ dữ liệu để nói rằng trung bình môn thi UNGXU = 5.0
# hay nói cách khác là chấp nhận G0 ở mức sai lầmm 5%
```

Out[47]: TtestResult(statistic=-0.7645471693105148, pvalue=0.4581953822944209, df=13)

In [48]:
```
# Tự nghiên cứu cách thiết lập mức tin cậy hoặc sai lầm trong đoạn code trên
```

In [49]:
```
# Bài tập tương tự: HẫY kiểm tra xem có phải điểm NGONNGU của thí sinh thi khối C
# là 5.5 hay không với mức sai lầm là 10%


dfKhoiC = df.loc[df['KT'] == 'C']
dfKhoiC['NGONNGU']
```

Out[49]:
```
1     6.00
2     5.00
6     6.50
22    5.00
23    6.75
24    7.00
25    4.75
26    5.25
27    5.25
95    5.25
96    5.25
97    7.00
98    5.00
99    5.25
Name: NGONNGU, dtype: float64
```

In [50]:
```
# Với one sample T-test thì giả thiết G0: muy = 5.5
# Mặc định mức tin cậy là 90% và mức sai lầm là 10%
stats.ttest_1samp(dfKhoiC['NGONNGU'],popmean=5.5)
#statistic = 7403 cho chúng ta biết mức sai lệch của trung bình mẫu từ giả thuyết.
```

Out[50]:
```
TtestResult(statistic=0.7403728818402906, pvalue=0.47223461312805337, df=13)
```

In [51]:
```
# Điểm trung bình có môn thi UNGXU của khối thi C có bằng 7.5 hay không, mức tin cậy 95%

stats.ttest_1samp(dfKhoiC['UNGXU'],popmean=7.5)
```

Out[51]:
```
TtestResult(statistic=-11.468207539657708, pvalue=3.5923185367668785e-08, df=13)
```

In [52]:
```
#KẾt luận :
# do anpha = 0.05 và p-value =3.5923185367668785e-08 (P-VALUE>ANPHA)
# Suy ra đủ dữ liệu để nói rằng trung bình môn thi ngonngu = 5.5
# hay nói cách khác là chấp nhận G0 ở mức sai lầmm 10%
#Nếu p-value < alpha => reject H0
# Nếu p-value > alpha => accecpt H0
#Do p-value=0.47 >alpha -->Chấp nhận H0 nghĩa là trung bình môn thi NGONNGU=5.5
```

In [53]:
```
# TWO SAMPLE T-TEST
# Mục đích: Kiểm tra xem trung bình của 2 biến số (định lượng)có bằng nhau không

#The sample size < 30 :
#H0 :delta_mean = 0
```

In [54]:
```
# Câu 2: Kiểm tra xem trung bình điểm thi LOGIC và trung bình điểm thi UNGXU của thí sinh thi khối C có
# H0: mean_LOGIC - mean_UNGXU = 0
dfKhoiC = df.loc[df['KT'] == 'C']
stats.ttest_ind(dfKhoiC['LOGIC'],dfKhoiC['UNGXU'],equal_var=True)
```

Out[54]:
```
TtestResult(statistic=-1.0329196014245297, pvalue=0.3111543826061086, df=26.0)
```

In [55]:
```
# Kết luận: Do alpha = 0.05 và p-value = 0.3111543826061086
# Suy ra đủ dữ liệu để nói rằng trung bình LOGIC bằng trung bình UNGXU hay nói cách khác là chấp nhận
```

In [56]:
```
# Kiểm tra xem trung bình ĐIỂM UNGXU có bằng trung bình NGONNGU cho thí sinh thi khối C hay không

dfKhoiC1 = df.loc[df['KT'] == 'C']
stats.ttest_ind(dfKhoiC1['UNGXU'],dfKhoiC1['NGONNGU'],equal_var=True)
```

Out[56]:
```
TtestResult(statistic=-2.6321295849540447, pvalue=0.014085909192033959, df=26.0)
```

In [57]:
```
# Kết luận: Do alpha = 0.01 và p-value = 0.014
# Suy ra đủ dữ liệu để nói rằng trung bình UNGXU bằng trung bình NGONNGU hay nói cách khác là chấp n
```

In [58]:
```
# ONE SAMPLE Z-TEST
# Mục đích: kiểm định trung bình của một biến số (định lượng)
# H0L mean = x
# n > 30 DÙNG Z TEST, N LÀ SỐ MẪU
```

In [59]:
```
from statsmodels.stats.weightstats import ztest as ztest
#PIP INSTALL STATSMODELS
```

In [60]:
```
# Hãy kiểm tra xem trung bình điểm toán học kì 2 lớp 12 có bằng 8.0
#Kiểm định Z (tiếng Anh: Z-Test) là một hình thức kiểm định thống kê được sử dụng để xác định xem hai giá
#Z-Test là một kiểm định thống kê được dùng để xác định xem liệu hai số bình quân của hai tổng thể có khác
#Nó chỉ được sử dụng khi có độ lệch chuẩn đã biết và cỡ mẫu lớn (n>30).
ztest(df['T6'],value=8.0)
```

Out[60]:
```
(-7.797828845339864, 6.298135014120743e-15)
```

In [61]:
```
# Kết luận: điểm trung bình không bằng 8 VÌ P-VALUE=6.298135014120743e-15<0
```

In [62]:
```
# TWO SAMPLE Z-TEST
# Tương tự n > 30
# H0: u1 = u2 both population means are equal
```

Loading [MathJax]/extensions/Safe.js

In [63]:
```
# Câu 4: Hãy kiểm tra xem điểm trung bình toán học kì 1 và học kì 2 năm lớp 12 có bằng nhau không
ztest(df['T5'],df['T6'],value=0)
```

Out[63]: (-1.094138573502891, 0.273894207026412)

In [64]:
```
# Đủ dữ kiện kết luận trung bình 2 học kì có bằng nhau
```

In [65]:
```
# Tương tự kiểm tra trung bình LOGIC và UNGXU có bằng nhau không
ztest(df['LOGIC'],df['UNGXU'],value=0)
# Nhỏ hơn => bác bỏ
```

Out[65]: (-4.172765180703833, 3.009250404643791e-05)

In [66]:
```
# Tương tự kiểm tra trung bình LOGIC và NGONNGU có bằng nhau không với mức tin cậy 95%
ztest(df['LOGIC'],df['NGONNGU'],value=0)
#P-VALUE =0.0037259661678783573>0 CHẤP NHẬN GIẢ THIẾT
```

Out[66]: (2.9004757923795, 0.0037259661678783573)

In [67]:
```
ztest(df['NGONNGU'],df['UNGXU'],value=0)
#BÁC BỎ VÌ P-VALUE =1.2511643846506845e-10<0
```

Out[67]: (-6.432991757573295, 1.2511643846506845e-10)

In [68]:
```
'''
# Kiểm định tương quan giữa 2 biến định lượng
# H0: r = 0
'''
from scipy.stats.stats import pearsonr
```

```
C:\Users\Lan Anh\AppData\Local\Temp\ipykernel_404\242169006.py:5: DeprecationWarning: Please use
`pearsonr` from the `scipy.stats` namespace, the `scipy.stats.stats` namespace is deprecated.
  from scipy.stats.stats import pearsonr
```

In [69]:
```
# Câu 5: Kiểm tra xem điểm toán hk1 và hk2 năm lớp 12 có tương quan không
pearsonr(df['T5'],df['T6'])
# Bác bỏ H0: r= 0 do pvalue < 5% => r =! 0 => 2 biến không tương quan (pvalue thì tương quan trên tổng
# statistic là gtri tương quan trên mẫu, nhìn lên sẽ biết tương quan mạnh hay YẾU
```

Out[69]: PearsonRResult(statistic=0.7786831657869808, pvalue=1.4846407216274206e-21)

In [70]:
```
# Kết luận: đủ dữ liệu để nói rằng T5 và T6 có tương quan với nhau với mức sai lầm là 5%
'''
Kết luận trên mẫu: tương quan thuận, mức độ tương quan rất cao: r = 0.7786831657869809
Suy diễn trên tổng thể: Đủ dữ liệu để nói rằng T5 và T6 có tương quan với nhau với mức sai lầm là 5%
Với p-value = 1.4846407216273482e
'''
```

Out[70]: '\nKết luận trên mẫu: tương quan thuận, mức độ tương quan rất cao: r = 0.7786831657869809\nSuy diễn tr
ên tổng thể: Đủ dữ liệu để nói rằng T5 và T6 có tương quan với nhau với mức sai lầm là 5%\nVới p-value
= 1.4846407216273482e\n'

In [71]:
```
'''
Định tính: Fisher (<30)
Chi square (>30)
Cả 2 H0: độc lập
'''

'''
Định lượng:
one sample t-test và one sample z-test => biến định lượng (H0; muy = alpha)
two sample t-test và two sample z-test => kiểm tra 2 biến định lượng có bằng nhau không (H0: denta muy = (

kiểm định Pearson kiểm tra xem 2 biến định lượng có tương quan không (H0: n=0)
'''
```

Out[71]:
```
'\nĐịnh lượng:\none sample t-test và one sample z-test => biến định lượng (H0; muy = alpha)\ntwo sample t-test và two sample z-test => kiểm tra 2 biến định lượng có bằng nhau không (H0: denta muy = 0)\n\nkiể
m định Pearson kiểm tra xem 2 biến định lượng có tương quan không (H0: n=0)\n'
```

In [72]:
```
# Sinh viên làm tương tự cho T5 và LOGIC có tương quan hay không
pearsonr(df['T5'],df['LOGIC'])
# statis: quá thấp, còn pvalue > 0.05 nên chấp nhận H0: r=0 => trên tổng thể không tương quan
```

Out[72]:
```
PearsonRResult(statistic=0.1846466122601273, pvalue=0.06590059130545516)
```

In [73]:
```
# Fisher Test
# Mục đích: Kiểm tra sự độc lập của 2 biến định tính dạng nhị phân 2x2
# Ho: Không có sự khác biệt giữa 2 biến định tính
import scipy.stats as stats
```

In [74]:
```
# Hãy kiểm tra xem có sự phụ thuộc nào giữa việc sinh viên có định hướng nghề nghiệp và giới tính
# khi thí sinh đăng ký dự thi hay không
crosdata = pd.crosstab(df['GT'],[df['HUONGNGHIEP']],rownames=['GT'],colnames=['HUONGNGHIEP'])
crosdata
```

Out[74]:

HUONGNGHIEP	No	Yes
GT		
F	23	25
M	32	20

In [75]:
```
odd_ratio, p_value = stats.fisher_exact(crosdata)
print('odd ratio is: ' + str(odd_ratio))
print('p_value is: ' + str(p_value))
```

```
odd ratio is: 0.575
p_value is: 0.22763927303454412
```

In [76]:
```
# Kết luận: Chấp nhận Ho vì p_value = 0.22763 > alpha = 0.05
# Tức là, đủ dữ liệu để nói rằng giới tính và việc định hướng nghề nghiệp là không có quan hệ gì cả ở mức sa
```

In [77]:
```
# Chi-Square Test
# Mục đích: Kiểm tra sự độc lập của 2 biến định tính
# Ho: Không có sự khác biệt giữa 2 biến định tính
from scipy.stats import chi2_contingency
```

In [78]:
```
# Hãy kiểm tra xem có sự phụ thuộc nào giữa khối thi và khu vực thi đăng ký dự thi hay không
crosdata = pd.crosstab(df['KV'],[df['KT']],rownames=['KV'],colnames=['KT'])
crosdata
```

Out[78]:

KT	A	A1	B	C	D1
KV					
1	29	2	8	8	13
2	9	0	0	2	8
2NT	11	4	1	4	1

In [79]:
```
stat, p, dof, excepted = chi2_contingency(crosdata)

# interpret p-value
alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print ('Dependent (Reject H0)')
else:
    print ('Independent (H0 holds true)')
# Có quan hệ giữa khối thi và khu vực thi đăng ký dự thi
# Kết luận: p-value = 0.02 < alpha = 0.05
# Tức là: không đủ dữ liệu để nói rằng KV và KT là độc lập hay có sự quan hệ giữa KT và KV
```

```
p value is 0.02012461887796485
Dependent (Reject H0)
```

In [80]:
```
# GT và KT có mối quan hệ hay không
crosdata = pd.crosstab(df['GT'],[df['KT']],rownames=['GT'],colnames=['KT'])
crosdata
```

Out[80]:

KT	A	A1	B	C	D1
GT					
F	15	5	4	8	16
M	34	1	5	6	6

In [81]:
```
stat, p, dof, excepted = chi2_contingency(crosdata)

# interpret p-value
alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print ('Dependent (Reject H0) => có quan hệ với nhau')
else:
    print ('Independent (H0 holds true) => Độc lập với nhau')
```

```
p value is 0.005044752209452435
Dependent (Reject H0) => có quan hệ với nhau
```

In [82]:
```
# ONE WAY ANOVA
# Kiểm định ANOVA ONE WAY
# Yêu cầu:
# 1. Biến định lượng trên nhóm định tính
# 2. Các biến định lượng trên từng nhóm theo phân phối chuẩn
# 3. H0: Giá trị trung bình dữ liệu định lượng trên từng nhóm định tính là bằng nhau
```

In [83]:
```
# Điểm toán học kì 2 lớp 12 có phụ thuộc vào giới tính hay không
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [84]:
```
model = ols('T6 ~ GT', data=df).fit()
aov_table = sm.stats.anova_lm(model,typ = 1)
aov_table
```

Out[84]:

	df	sum_sq	mean_sq	F	PR(>F)
GT	1.0	1.55201	1.55201	0.833769	0.363426
Residual	98.0	182.42109	1.86144	NaN	NaN

In [85]:
```
'''
Trả lời:
    H0: mean (nhóm GT) bằng nhau
    p-value = 0.363426 > 0.05 => chấp thuận
    Không phụ thuộc
'''
```

Out[85]: '\nTrả lời:\n H0: mean (nhóm GT) bằng nhau\n p-value = 0.363426 > 0.05 => chấp thuận\n Không phụ thuộc\n'

In [86]:
```
# Điểm LOGIC có phụ thuộc vào KV hay không
model = ols('LOGIC ~ KV', data=df).fit()
aov_table = sm.stats.anova_lm(model,typ = 1)
aov_table
# Ko phụ thuộc
```

Out[86]:

	df	sum_sq	mean_sq	F	PR(>F)
KV	2.0	6.053398	3.026699	2.790934	0.066299
Residual	97.0	105.194102	1.084475	NaN	NaN

In [87]:
```
# Điểm UNGXU có phụ thuộc khối thi hay không
model = ols('UNGXU ~ KT', data=df).fit()
aov_table = sm.stats.anova_lm(model,typ = 1)
aov_table
# p_value = 0.46041 => chấp nhận => không phụ thuộc
```

Out[87]:

	df	sum_sq	mean_sq	F	PR(>F)
KT	4.0	3.967636	0.991909	0.911814	0.46041
Residual	95.0	103.344864	1.087841	NaN	NaN

In [88]:
```
'''
Two way anova
# Kiểm định ANOVA two WAY
# Yêu cầu:
# 1. Biến định lượng trên nhóm định tính
# 2. Các biến định lượng trên từng nhóm theo phân phối chuẩn
# 3. H0: Trung bình các cột dữ liệu bằng nhau
'''
```

Out[88]:
```
'\nTwo way anova \n# Kiểm định ANOVA two WAY\n# Yêu cầu:\n# 1. Biến định lượng trên nhóm định tí
nh\n# 2. Các biến định lượng trên từng nhóm theo phân phối chuẩn\n# 3. H0: Trung bình các cột dữ liệu b
ằng nhau\n'
```

In [89]:
```
# Hãy cho biết điểm LOGIC có phụ thuộc vào loại GT trên từng nhóm KV hay không
# Performing two-way anova
model = ols('LOGIC ~ GT + KV + GT:KV', data=df).fit()
result = sm.stats.anova_lm(model,type = 2)

# Print the result
print(result)
# Chấp nhận Ho=> k phụ thuộc (p-value = 0.052602)
# Bác bỏ H0 => phụ thuộc (pvalue = 0.019173)
# GT:KV bác bỏ -> k phụ thuộc -> tuy nhiên điểm LOGIC trên từng nhóm GT xét theo từng KV nữa thì nó độ

'''
Kết luận khác
p-value = 0.052602 -> LOGIC độc lập theo nhóm GT
p-value = 0.0.019173 -> LOGIC phụ thuộc theo nhóm KV
p-value = 0.576510 -> LOGIC độc lập theo nhóm GT trên từng loại KV
'''
```

```
             df     sum_sq   mean_sq         F    PR(>F)
GT          1.0   3.998401  3.998401  3.853364  0.052602
KV          2.0   8.561314  4.280657  4.125382  0.019173
GT:KV       2.0   1.149707  0.574854  0.554002  0.576510
Residual   94.0  97.538077  1.037639       NaN       NaN
```

Out[89]:
```
'\nKết luận khác \np-value = 0.052602 -> LOGIC độc lập theo nhóm GT\np-value = 0.0.019173 -> LOGI
C phụ thuộc theo nhóm KV\np-value = 0.576510 -> LOGIC độc lập theo nhóm GT trên từng loại KV\n'
```

In [90]:
```
# Phân tích xem NGONNGU có phụ thuộc theo nhóm KV trên từng nhóm KT hay không
from statsmodels.formula.api import ols
model = ols('NGONNGU ~ KV + KT + KV:KT', data=df).fit()
result = sm.stats.anova_lm(model,type = 2)
print(result)
'''
Kết luận
Tất cả đều phụ thuộc
'''
```

```
             df      sum_sq    mean_sq         F        PR(>F)
KV          2.0    4.237274   2.118637   1.416009  2.482173e-01
KT          4.0   65.431143  16.357786  10.932867  3.150065e-07
KV:KT       8.0    1.486487   0.185811   0.124188  9.981082e-01
Residual   87.0  130.169640   1.496203        NaN           NaN
```

Out[90]:
```
'\nKết luận  \nTất cả đều phụ thuộc\n'
```