일원 one-way 분산분석 analysis of variance

선형모형 : y_ij = mu + A_i +e_ij

가정 : e_ij ~ N(0, sigma^2)

> y : 반응 response 변수, 측정형

> A : 요인 factor,

> i : 수준(범주) i=1,2,..,k

> j : 수준 내 반복 실험 수 j=1,2,...,n_i

분꽃 데이터

반응변수 : Sepal(길이, 넓이), Petal(길이, 넓이)

요인-품종 : setosa, virginica, versicolor

[연구문제1] 분꽃 품종에 따라 꽃받침(sepal) 길이는 차이가 있나?

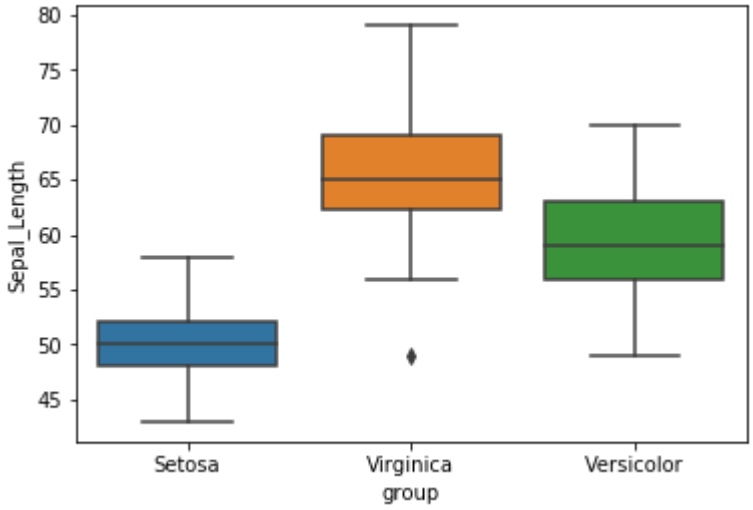
```
1  import pandas as pd
2  df=pd.read_csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/iris.csv')
3  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
Sepal_Length    150 non-null int64
Sepal_Width     150 non-null int64
Petal_Length    150 non-null int64
Petal_Width     150 non-null int64
group           150 non-null object
dtypes: int64(4), object(1)
memory usage: 5.9+ KB
```

그래프 표현

```
1  import seaborn as sns
2  sns.boxplot(x='group', y='Sepal_Length',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f680d3f6fd0>
```



이상치 제거

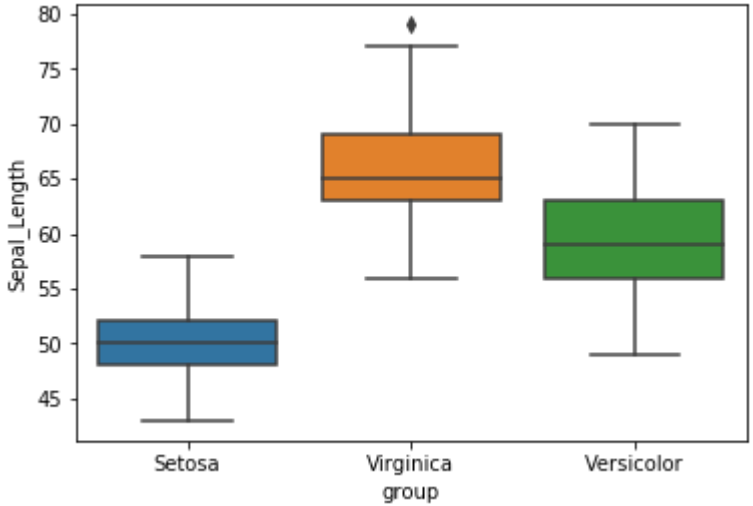
```
1  indexNames=df[(df['group']=='Virginica') & (df['Sepal_Length']<55)].index
2  df.drop(indexNames,inplace=True)
3  df.shape
```

```
(149, 5)
```

> 이상치 1개 제거

```
1  import seaborn as sns
2  sns.boxplot(x='group', y='Sepal_Length',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f680aa54c18>
```



> 다시 한 개 더 발생하였지만... 그냥 계속 간다.

숫자요약 (평균, 표준편차)

```
1  !pip install researchpy
```

```
1  import researchpy as rp
2  rp.summary_cont(df['Sepal_Length'].groupby(df['group']))
```

group	N	Mean	SD	SE	95% Conf.	Interval
Setosa	50	50.06000	3.524897	0.498496	49.073029	51.046971
Versicolor	50	59.36000	5.161711	0.729976	57.914721	60.805279
Virginica	49	66.22449	5.934592	0.847799	64.545584	67.903396

> 꽃받침 길이는 Virginica > Versicolor > Setosa 순이다.

분산분석

귀무가설 : 모든 종의 꽃받침 길이는 동일하다. mu1=mu2=mu3

대립가설 : 모든 종의 꽃받침 길이는 동일한 것은 아니다. 적어도 한 종은 다르다.

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
results = ols('Sepal_Length~group',data=df).fit()
results.summary()
```

OLS Regression Results

Dep. Variable:	Sepal_Length	**R-squared:**	0.644
Model:	OLS	**Adj. R-squared:**	0.639
Method:	Least Squares	**F-statistic:**	132.1
Date:	Sun, 13 Oct 2019	**Prob (F-statistic):**	1.79e-33
Time:	01:29:12	**Log-Likelihood:**	-448.79
No. Observations:	149	**AIC:**	903.6
Df Residuals:	146	**BIC:**	912.6
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	50.0600	0.703	71.237	0.000	48.671	51.449
group[T.Versicolor]	9.3000	0.994	9.358	0.000	7.336	11.264
group[T.Virginica]	16.1645	0.999	16.183	0.000	14.190	18.139

Omnibus:	2.598	**Durbin-Watson:**	2.042
Prob(Omnibus):	0.273	**Jarque-Bera (JB):**	2.647
Skew:	0.306	**Prob(JB):**	0.266
Kurtosis:	2.770	**Cond. No.**	3.72

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
1  aov_table=sm.stats.anova_lm(results, typ=2)
2  aov_table
```

	sum_sq	df	F	PR(>F)
group	6522.377710	2.0	132.080628	1.786776e-33
Residual	3604.870612	146.0	NaN	NaN

다중비교 Tukey 방법

요인의 각 수준간 쌍체 평균 비교

쌍체비교는 Post Hoc 검정으로 분산분석 결과와 관계 없이 실행

> 분산분석 결과 요인 수준에 따른 평균 차이가 있어도 쌍체 비교에서는 유의한 쌍체 pairwise 없을 수 있음.

> 물론 그 반대도 존재함

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison
mc=MultiComparison(df['Sepal_Length'],df['group'])
print(mc.tukeyhsd())
```

```
     Multiple Comparison of Means - Tukey HSD, FWER=0.05
=================================================================
  group1     group2   meandiff p-adj  lower   upper  reject
-----------------------------------------------------------------
    Setosa Versicolor      9.3 0.001  6.9466 11.6534   True
    Setosa  Virginica  16.1645 0.001 13.7992 18.5298   True
Versicolor  Virginica   6.8645 0.001  4.4992  9.2298   True
-----------------------------------------------------------------
```

첫 행 : Setosa(50.1) - Versicolor (59.4) 차이는 9.3이고 차이는 유의(True)

결과적으로 모든 쌍체 유의함 : Setosa < Versicolor < Virginica

오차 정규성 검정

귀무가설 : 오차항은 정규분포를 따른다.

```
stats.shapiro(results.resid)
```

```
(0.9836390614509583, 0.0740874856710434)
```

> > 유의확률 0.07로 귀무가설 채택 -> 최종 모형 문제 없음

분산 동질성 검정

해결 방법이 없음 - 하여, 굳이 할 필요 없음

```
import scipy.stats as stats
stats.levene(df['Sepal_Length'][df['group'] == 'Setosa'],
             df['Sepal_Length'][df['group'] == 'Virginica'],
             df['Sepal_Length'][df['group'] == 'Versicolor'])
```

```
LeveneResult(statistic=5.755093604085759, pvalue=0.003928503925436201)
```

> 유의확률이 0.003이므로 귀무가설(집단간 분산 동일성) 기각

Two-way ANOVA

모형 : Y_ijk = mu+A_i+B_j+(AB)_ij+e_ijk

주효과 : A, B 교호효과 : A, B

데이터

직장을갖는데걸리는시간을학력(E1=고졸,E2=전문대졸,E3=대졸,E4=대학원졸),성별2개요인에따라차이가 있는지 분석하시오.

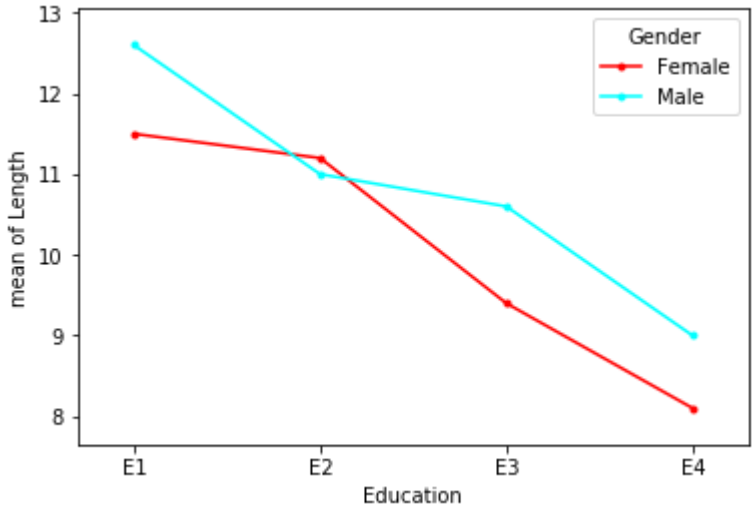
http://wolfpack.hnu.ac.kr/Stat_Notes/elem_stat/Stat_methods/Jobs.csv

```
import pandas as pd
df=pd.read_csv('http://wolfpack.hnu.ac.kr/Stat_Notes/elem_stat/Stat_methods/Jobs.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 80 entries, 0 to 79
Data columns (total 3 columns):
Gender       80 non-null object
Education    80 non-null object
Length       80 non-null int64
dtypes: int64(1), object(2)
memory usage: 2.0+ KB
```

그래프 요약

```
from statsmodels.graphics.factorplots import interaction_plot
fig=interaction_plot(df.Education,df.Gender,df.Length)
```



숫자요약

```
import researchpy as rp
rp.summary_cont(df.groupby(['Education','Gender']))['Length']
```

		N	Mean	SD	SE	95% Conf.	Interval
Education	Gender						
E1	Female	10	11.5	2.877113	0.909823	9.716747	13.283253
	Male	10	12.6	2.875181	0.909212	10.817944	14.382056
E2	Female	10	11.2	3.119829	0.986577	9.266310	13.133690
	Male	10	11.0	2.943920	0.930949	9.175339	12.824661
E3	Female	10	9.4	4.060651	1.284091	6.883182	11.916818
	Male	10	10.6	3.405877	1.077033	8.489015	12.710985
E4	Female	10	8.1	3.510302	1.110055	5.924292	10.275708
	Male	10	9.0	2.309401	0.730297	7.568618	10.431382

```
1  rp.summary_cont(df.groupby(['Education']))['Length']
```

	N	Mean	SD	SE	95% Conf.	Interval
Education						
E1	20	12.05	2.855742	0.638563	10.798416	13.301584
E2	20	11.10	2.954034	0.660542	9.805338	12.394662
E3	20	10.00	3.699218	0.827170	8.378746	11.621254
E4	20	8.55	2.928535	0.654840	7.266513	9.833487

```
1  rp.summary_cont(df.groupby(['Gender']))['Length']
```

	N	Mean	SD	SE	95% Conf.	Interval
Gender						
Female	40	10.05	3.573047	0.564948	8.942701	11.157299
Male	40	10.80	3.081791	0.487274	9.844943	11.755057

분산분석

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
results=ols('Length ~ Gender*Education',df).fit()
results.summary()
```

OLS Regression Results

Dep. Variable:	Length	**R-squared:**	0.174
Model:	OLS	**Adj. R-squared:**	0.094
Method:	Least Squares	**F-statistic:**	2.172
Date:	Sun, 13 Oct 2019	**Prob (F-statistic):**	0.0467
Time:	02:15:22	**Log-Likelihood:**	-201.75
No. Observations:	80	**AIC:**	419.5
Df Residuals:	72	**BIC:**	438.6
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	11.5000	1.004	11.451	0.000	9.498	13.502
Gender[T.Male]	1.1000	1.420	0.774	0.441	-1.731	3.931
Education[T.E2]	-0.3000	1.420	-0.211	0.833	-3.131	2.531
Education[T.E3]	-2.1000	1.420	-1.479	0.144	-4.931	0.731
Education[T.E4]	-3.4000	1.420	-2.394	0.019	-6.231	-0.569
Gender[T.Male]:Education[T.E2]	-1.3000	2.009	-0.647	0.520	-5.304	2.704
Gender[T.Male]:Education[T.E3]	0.1000	2.009	0.050	0.960	-3.904	4.104
Gender[T.Male]:Education[T.E4]	-0.2000	2.009	-0.100	0.921	-4.204	3.804

Omnibus:	3.100	**Durbin-Watson:**	2.206
Prob(Omnibus):	0.212	**Jarque-Bera (JB):**	1.910
Skew:	-0.130	**Prob(JB):**	0.385
Kurtosis:	2.289	**Cond. No.**	12.5

Warnings:

```
1  aov_table=sm.stats.anova_lm(results, typ= 2)
2  aov_table
```

	sum_sq	df	F	PR(>F)
Gender	11.25	1.0	1.115395	0.294443
Education	135.85	3.0	4.489672	0.006043
Gender:Education	6.25	3.0	0.206555	0.891546
Residual	726.20	72.0	NaN	NaN

> Education 요인만 유의함

Education 요인 쌍체 Tukey HSD

```
1  from statsmodels.stats.multicomp import pairwise_tukeyhsd
2  from statsmodels.stats.multicomp import MultiComparison
3  mc=MultiComparison(df['Length'],df['Education'])
4  print(mc.tukeyhsd())
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
===================================================
group1 group2 meandiff p-adj   lower   upper  reject
---------------------------------------------------
    E1     E2    -0.95 0.7472 -3.5486  1.6486  False
    E1     E3    -2.05 0.1717 -4.6486  0.5486  False
    E1     E4     -3.5 0.0038 -6.0986 -0.9014   True
    E2     E3     -1.1 0.6629 -3.6986  1.4986  False
    E2     E4    -2.55 0.0564 -5.1486  0.0486  False
    E3     E4    -1.45 0.4642 -4.0486  1.1486  False
---------------------------------------------------
```

> E1,고졸 E4- 대학원졸 잡 잡는 기간만 유의적 차이

ANCOVA 공분산분석 Analysis of Covariance

스텝업운동후힘든정도를파악하기위하여운동후맥박과일반상태에서맥박을측정하였다.

> 계단높이수준=2:Height:0ifstepatthelow(5.75")height,1ifatthehigh(11.5")height

> 운동빈도수준=3:Frequency:therateofstepping.0ifslow(14steps/min),1ifmedium(21 steps/min), 2 if high (28 steps/min)

> 쉬는상태의맥박:Rest_HR:therestingheartrateofthesubjectbeforeatrial,inbeatsper minute

> 운동후맥박HR:thefinalheartrateofthesubjectafteratrial,inbeatsperminute

쉬는상태의맥박은 공변량, 계단높이 요인과 운동빈도 요인이 운동후 맥박에 영향을주는지 공변량 이원분산분석 하시오.

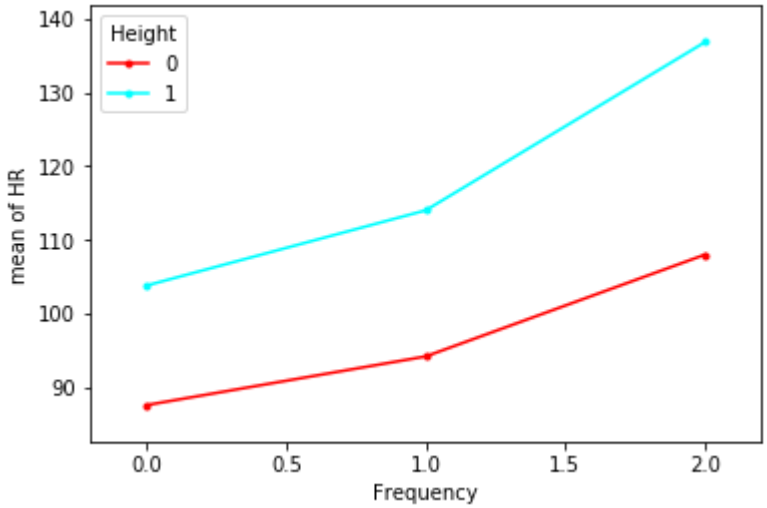
http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/Heart.csv

```
import pandas as pd
df=pd.read_csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/Heart.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 4 columns):
Height       30 non-null int64
Frequency    30 non-null int64
RestHR       30 non-null int64
HR           30 non-null int64
dtypes: int64(4)
memory usage: 1.0 KB
```

그래프 요약

```
from statsmodels.graphics.factorplots import interaction_plot
fig=interaction_plot(df.Frequency,df.Height,df.HR)
```



숫자요약

```
1  import researchpy as rp
2  rp.summary_cont(df.groupby(['Height','Frequency']))['HR']
```

Height	Frequency	N	Mean	SD	SE	95% Conf.	Interval
0	0	5	87.6	9.099451	4.069398	79.623980	95.576020
	1	5	94.2	8.899438	3.979950	86.399298	102.000702
	2	5	108.0	16.703293	7.469940	93.358918	122.641082
1	0	5	103.8	10.521407	4.705316	94.577580	113.022420
	1	5	114.0	21.000000	9.391486	95.592688	132.407312
	2	5	136.8	13.349157	5.969925	125.098948	148.501052

공분산분석

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
results=ols('HR~Frequency*Height+RestHR',df).fit()
aov_table=sm.stats.anova_lm(results, typ= 2)
aov_table
```

	sum_sq	df	F	PR(>F)
Frequency	3784.202188	1.0	30.775449	0.000009
Height	1823.321944	1.0	14.828370	0.000726
Frequency:Height	73.113792	1.0	0.594606	0.447871
RestHR	1785.056976	1.0	14.517176	0.000805
Residual	3074.043024	25.0	NaN	NaN

> 운동빈도, 높이 주효과만 유의함

> 교호 효과 유의하지 않음

운동빈도 주효과

```
rp.summary_cont(df.groupby(['Frequency']))['HR']
```

Frequency	N	Mean	SD	SE	95% Conf.	Interval
0	10	95.7	12.605554	3.986226	87.886996	103.513004
1	10	104.1	18.441800	5.831809	92.669654	115.530346
2	10	122.4	20.823064	6.584831	109.493731	135.306269

[참고]이원분산분석

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
results=ols('HR~Frequency*Height',df).fit()
aov_table=sm.stats.anova_lm(results, typ= 2)
aov_table
```

	sum_sq	df	F	PR(>F)
Frequency	3564.45	1.0	19.072606	0.000179
Height	3499.20	1.0	18.723467	0.000198
Frequency:Height	198.45	1.0	1.061863	0.312282
Residual	4859.10	26.0	NaN	NaN