1 분산분석

AB 검정 말고 여러그룹의 수치들을 서로 비교. 여러 그룹간의 통계적 유의마한 차이를 검정하는 통계적 절차를 분산분석 ANOVA라고 한다.

- 쌍별비교 : 여러 그룹 중 두 그룹간의 가설검정
- 총괄검정 : 여러 그룹 편균들의 전체 분산에 관한 단일 가설검정
- 분산분해 : 구성 요소 분리 예를들면 전체 평균, 처리 평균, 잔차 오차로부터 개별 값들에 대한 기여
- F 통계량 : 그룹 평균 간의 차이가 랜덤 모델에서 예상되는 것에서 벗어나는 정도를 츤정하는 표준화된 통계량
- SS: 어떤 평균으로부터의 편차들의 제곱합

1.1 데이터 설명

4개의 웹페이지 점착성, 즉 방문자가 페이지에서 보낸 시간을 의미. 네 페이지는 무작위로 전환되며 각 웹 방문자는 무작위로 그중 한곳에 접속. 페이지에는 총 5명의 방문자. 각 열든 독립적인 데이터 집합. 웹 테스트에서는 전통적인 임의표본추출 설계를 완전히 구현할 수 없다. 실험 결과를 검토할 때 이러한 요소들을 잠재적 편향 요인으로 고려.
4개의 평균에 대해서 총 6가지의 비교가 가능. 한쌍씩 비교하는 횟수가 증가할수록 우연히 일어난 일에 속을 가능성이 커짐. 개별 페이지 간의 가능한 모든 비교에 대해 걱정하는 대신 모든 페이지가 동일한 기본적인 점착성을 갖는가? 이들 사이의 차이는 우연에 의한 것이고 원래 4개의 페이지에 할당된 세션 시간 역시 무작위로 할당된 것인가? 이 질문을 다루는 전체적인 총괄검정을 할 수 있다.
ANOVA가 바로 이검정에 사용되는 방법.//

- 모든 데이터를 한 상자에 모은다.
- 5개의 값을 갖는 4개의 재표본을 섞어서 추출한다.
- 각 그룹의 평균을 기록한다.
- 네 그룹 평균 사이의 분산을 기록한다.
- 2 4단계를 여러번 반복한다.

재표본추출한 분산이 관찰된 변화를 초과한 시간이 p값.

ANOVA_oneway

July 15, 2022

[2]:
```
columns = ['page1','page2','page3','page4']
page1 = [164,172,177,156,195]
page2 = [178,191,182,185,177]
page3 = [175,193,171,163,176]
page4 = [155,166,164,170,168]
```

[10]:
```
import pandas as pd

data1 = pd.DataFrame(page1,columns = ['page1'])

data2 = pd.DataFrame(page2,columns = ['page2'])

data3 = pd.DataFrame(page3,columns = ['page3'])

data4 = pd.DataFrame(page4,columns = ['page4'])
```

[11]:
```
df = pd.concat([data1,data2,data3,data4],axis = 1, join = 'inner')
```

[12]:
```
df.head()
```

[12]:

	page1	page2	page3	page4
0	164	178	175	155
1	172	191	193	166
2	177	182	171	164
3	156	185	163	170
4	195	177	176	168

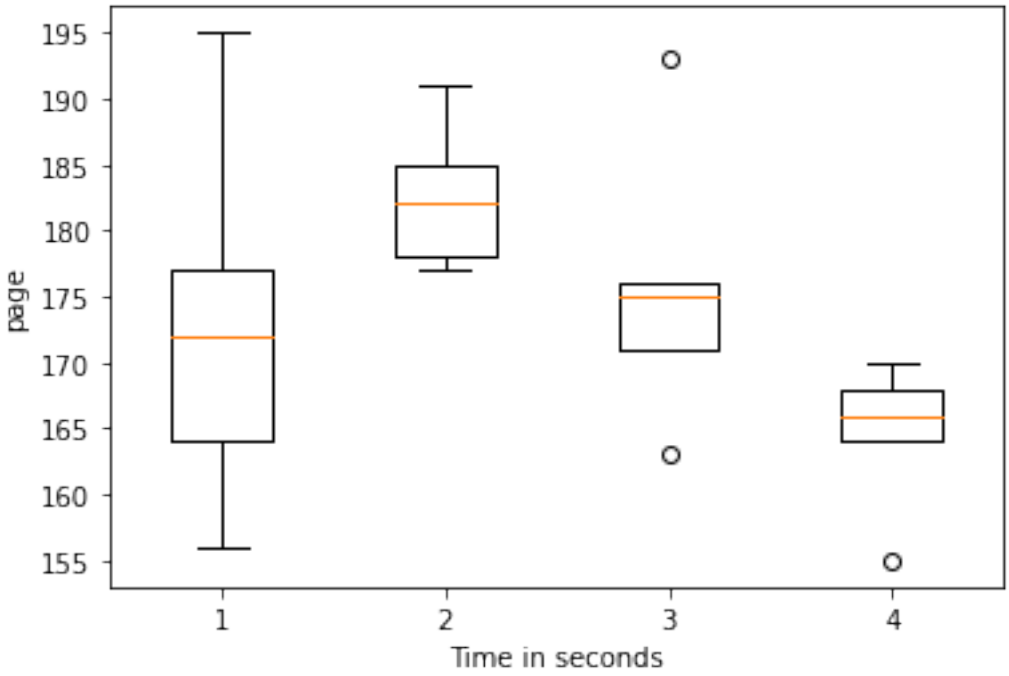
[13]:
```
df.describe()
```

[13]:

	page1	page2	page3	page4
count	5.00000	5.000000	5.000000	5.000000
mean	172.80000	182.600000	175.600000	164.600000
std	14.75466	5.683309	10.990905	5.813777
min	156.00000	177.000000	163.000000	155.000000
25%	164.00000	178.000000	171.000000	164.000000
50%	172.00000	182.000000	175.000000	166.000000
75%	177.00000	185.000000	176.000000	168.000000
max	195.00000	191.000000	193.000000	170.000000

[15]:
```
import matplotlib.pyplot as plt

plt.boxplot(df)
plt.xlabel('Time in seconds')
plt.ylabel('page')
plt.xticks(label = columns)
plt.show()
```



[107]:
```
data = page1+page2+page3+page4
data5 = pd.DataFrame(data, columns = ['page'])
```

[50]:
```
import random
```

[56]:
```
random.shuffle(data)
```

```
[193, 166, 170, 185, 172, 176, 195, 171, 156, 164, 177, 155, 168, 182, 164, 191,
178, 163, 175, 177]
```

[99]:
```
data1 = pd.DataFrame(data[:5],columns = ['page1'])

data2 = pd.DataFrame(data[5:10],columns = ['page2'])

data3 = pd.DataFrame(data[10:15],columns = ['page3'])

data4 = pd.DataFrame(data[15:20],columns = ['page4'])
```

[123]:
```
print(data1)
```

```
   page1
0    164
1    172
2    177
3    156
4    195
```

[101]:
```
print(data2)
```

```
   page2
0    178
1    191
2    182
3    185
4    177
```

[102]:
```
print(data3)
```

```
   page3
0    175
1    193
2    171
3    163
4    176
```

[103]:
```
print(data4)
```

```
   page4
0    155
1    166
2    164
3    170
4    168
```

[104]:
```
df2 = pd.concat([data1,data2,data3,data4],axis = 1, join = 'inner')
```

[126]:
```
print(' :',df2['page1'].mean(),'  :',math.sqrt(df2['page1'].var()))
print(' :',df2['page2'].mean(),'  :',math.sqrt(df2['page2'].var()))
print(' :',df2['page3'].mean(),'  :',math.sqrt(df2['page3'].var()))
print(' :',df2['page4'].mean(),'  :',math.sqrt(df2['page4'].var()))
print(' :', data5.mean(),' :',data5.var())
```

```
 : 172.8   : 14.75466028073842
 : 182.6   : 5.683308895353129
 : 175.6   : 10.99090533122727
 : 164.6   : 5.813776741499454
 : page    173.9
```

```
dtype: float64  : page    128.936842
dtype: float64
```

[110]:
```
import math
```

[111]:
```
def ssb(records,groups,g_avg,e_avg):
    res = 0
    for i in range(groups):
        res+=(g_avg[i]-e_avg)**2
        print(res)
    res=records*res
    return res
```

[135]:
```
g_avg = []
sum_ = 0
for k in range(1,5):
    g_avg.append(df2['page'+str(k)].mean())
print(g_avg)

for i in range(1,5):
    print(df2['page'+str(i)].var())
    sum_+=df2['page'+str(i)].var()
ssw = sum_*4
```

```
[172.8, 182.6, 175.6, 164.6]
217.7
32.300000000000004
120.80000000000003
33.800000000000004
```

[113]:
```
e_avg = data5.mean()
```

[136]:
```
ssb = ssb(5,4,g_avg,e_avg)
print(ssb)
```

```
page    1.21
dtype: float64
page    76.9
dtype: float64
page    79.79
dtype: float64
page    166.28
dtype: float64
page    831.4
dtype: float64
```

[115]:
```
from scipy.stats import f_oneway
F_statistic, pVal = f_oneway(data1, data2, data3,data4)
```

```
print('Altman 910           :',F_statistic, pVal)
if pVal < 0.05:
    print('P-value                              .')
```

```
Altman 910           : [2.73982534] [0.07758622]
```

[139]:
```
sst = ssw + ssb
msb = ssb/3
msw = ssw/ 16
F_stat = msb/msw
print(F_stat)
```

```
page    2.739825
dtype: float64
```