

Bağımlı iki örneklem testi burdaki amaç bir kitle üzerinde yapılan iki uygulama arasında bir fark olup olmadığını anlamaktır

Örneğin bir firmada yapılan bir eğitimin öncesi ve sonrası karşılaştırılabilir

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
```

```
In [2]: oncesi=pd.DataFrame([123,119,119,116,123,121,120,117,
                             118,121,121,123,119,121,118,124,121,
                             125,115,115,119,118,121,117,120,120,
                             121,117,118,117,123,118,124,121,115,118,125,115])
```

```
In [3]: sonrasi=pd.DataFrame([118,127,122,132,129,123,129,132,128,130,
                               128,138,140,130,134,134,124,140,134,129,129,138,134,124,122,126,133
                               ,127,130,130,130,132,117,130,125,129,133,120,127,123])
```

```
In [7]: #birinci Veri Seti
birlikte=pd.concat([oncesi,sonrasi],axis=1)
birlikte.head()
birlikte.columns=["Öncesi","Sonrası"]
```

```
In [10]: print("Birlikte Veri Seti \n",birlikte.head())
```

```
Birlikte Veri Seti
   Öncesi  Sonrası
0    123.0     118
1    119.0     127
2    119.0     122
3    116.0     132
4    123.0     129
```

```
In [16]: #ikinci veri seti
grup_once=np.arange(len(oncesi))
grup_once=pd.DataFrame(grup_once)
grup_once[:]="Öncesi"
A=pd.concat([oncesi,grup_once],axis=1)

grup_sonra=np.arange(len(sonrasi))
grup_sonra=pd.DataFrame(grup_sonra)
grup_sonra[:]="Sonrası"
B=pd.concat([sonrasi,grup_sonra],axis=1)

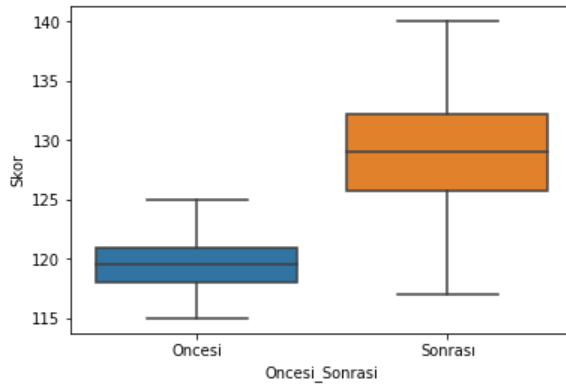
#TümVeriyi bir araya getirme
ab=pd.concat([A,B])
ab.columns=["Skor","Öncesi_Sonrası"]
```

```
In [17]: ab.head()
```

	Skor	Oncesi_Sonrasi
0	123	Oncesi
1	119	Oncesi
2	119	Oncesi
3	116	Oncesi
4	123	Oncesi

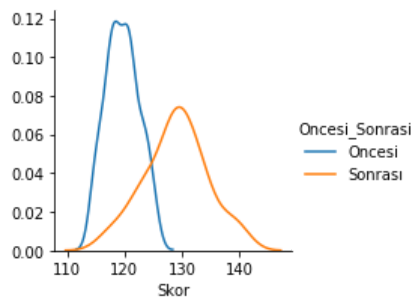
```
In [20]: sns.boxplot(x="Oncesi_Sonrasi",y="Skor",data=ab)
```

<matplotlib.axes._subplots.AxesSubplot at 0x284140f88d0>



```
In [27]: sns.FacetGrid(data=ab,hue="Oncesi_Sonrasi").map(sns.kdeplot,"Skor").add_legend()  
#Oncesi daha homojen dağılım ama sonrasi daha yayıllımlı
```

<seaborn.axisgrid.FacetGrid at 0x28414a17fd0>



```
In [28]: #Varsayım kontrolü 1-Normallik 2-Varyans Homojelği
```

```
In [29]: from scipy.stats import shapiro
```

```
In [31]: shapiro(oncesi)
```

(0.9558944702148438, 0.13929924368858337)

```
In [33]: shapiro(sonrasi)
```

(0.9780089259147644, 0.6159515380859375)

```
In [34]: from scipy.stats import levene
```

```
In [37]: levene(öncesi[0],sonrası[0])
#Varyans homojenliği reddeilir
#Ho varyanslar homojen dağılmıştır yani a ve b kitlesi
#arasında fark yoktu der
#Ama reddedilir
#Ama bunu göz ardı edeceğiz sonra non parametrik te deniyeceğiz
#Varyans homojenliğinde biraz esnek davranabiliriz
#ama normallikte sorun varsa direk nonparametrik yaklaşım kullanılır

LeveneResult(statistic=8.197532433118097, pvalue=0.005418326309480423)
```

```
In [55]: import scipy.stats as stats
birlikte.fillna(value=120,inplace=True)
```

```
In [59]: stats.ttest_rel(birlikte.Öncesi,birlikte.Sonrası)
#H0:  $\mu_0 = \mu_1$  yani bir değişme olmamıştır bu hipotez reddeilir yani öncesi ve sonrası arasında bir
değişiklik olmuştur
#eğitim başarılı olmutur %95 olasılık

Ttest_relResult(statistic=-9.922779606661322, pvalue=3.1856859759657504e-12)
```

Yukarıda Varsayımın sağlanmadığı görüldü bu varsayım varyansın homojen olmadığı içindi bu yüzden nonparametrik iki bağımlı örneklem testinide yapabiliriz

H0 : $\mu_0 = \mu_1$ H1 : $\mu_0 \neq \mu_1$

```
In [60]: stats.wilcoxon(birlikte.Öncesi,birlikte.Sonrası)

WilcoxonResult(statistic=18.5, pvalue=1.392810757723373e-07)
```

Hipotez reddeilir yani bir fark olmuştur