

---

# İST377 PARAMETRİK OLMAYAN İSTATİSTİKSEL YÖNTEMLER

---

**HAZIRLAYAN: HANDE NUR BANUŞ 2210329067**

03 OCAK 2024

1.En az 3 bağımsız gruptan oluşan bir veri kümesi bulunuz. Bulduğunuz veri kümesine dair özetleyici istatistikleri bularak yorumlayınız. Uygun grafiksel gösterimleri veriniz (Box-plot vb.). Ayrıca her bir grup için verilerin normal dağılmadığını, dolayısıyla parametrik testlerin uygulanacağını gerekli testler ve grafikler ile gösteriniz.

```
#gerekli kütüphaneleri çağıralım
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import shapiro

# Veri seti oluşturalım
data = {
    'ülke': ['BRA']*10 + ['LUX']*10 + ['IDN']*10,
    'puan': [385, 388, 348, 361, 379, 382.925, 370, 379, 448.984, 454,
            530, 480, 502.193, 491, 487, 485, 482, 479, 502, 480,
            362, 365, 351, 410.899, 430, 425, 348, 368, 372, 432.140]
}

# Veri setini DataFrame'e çevirelim
df = pd.DataFrame(data)
```

Önce gerekli kütüphaneleri yükleriz. pandas veri çerçevesi oluşturmak ve manipüle etmek için, seaborn ve matplotlib.pyplot görselleştirmeler için, scipy.stats kütüphanesi ise shapiro fonksiyonlarına erişmek için kullanılan istatistiksel pakettir.

Veri setini oluştururken her ülkeden onar tane olduğunu belirten kod satırını azıp ismine ülke dedim. sonra sırasıyla puan değerlerini girdim ve data isminde değişkene atadım. Sonra data verimi data frame yapısına çevirdim yani veri çerçevesi haline getirdim ve df ismini verdim.

```
summary_stats = df.groupby('ülke')['puan'].describe()
print("Özetleyici İstatistikler:")
print(summary_stats)
```

Özetleyici İstatistikler:

	count	mean	std	min	25%	50%	75%	max
ülke								
BRA	10.0	389.5909	34.800214	348.0	372.25	380.9625	387.25000	454.00
IDN	10.0	386.4039	34.010176	348.0	362.75	370.0000	421.47475	432.14
LUX	10.0	491.8193	15.914950	479.0	480.50	486.0000	499.25000	530.00

Özetleyici istatistikler bilgisi için describe() fonksiyonu kullanılır. Çıktıya göre brazilyalı öğrencilerin en yüksek puanı 454 , Endonezya için 432.14 , Luxemburg için 530 olduğunu ve her bir ülkenin minimum puan değerlerini , ortalamalarını , standart sapmalarını ve çeyrekliklerini görebiliyoruz.

## Normallik incelenmesi.

Verilerin normal dağılıma uyup uymadığını görmek için shapiro testi yaparız. Burda yokluk hipotezi :

$$H_0 = \text{veriler normal dağılıma uygundur}$$

Eğer p değeri  $\alpha=0.05$ 'ten küçükse hipotez reddedilir ve verilerin normal dağılıma uymadığı söylenir.

```
# Shapiro-Wilk testini uygulayalım
shapiro_test = df.groupby('ülke')['puan'].apply(lambda x: shapiro(x).pvalue)

# İstatistiksel olarak anlamlı bir düzey belirle
alpha = 0.05

# Shapiro-Wilk testi p değerlerini yazdıralım
print("Shapiro-Wilk Testi P Değerleri:")
print(shapiro_test)

# Hesaplanan p değerlerini alıp ve normal dağılıma uygun olanları belirleyelim
normal_d = shapiro_test > alpha

# Normal dağılıma uymayan ülkeleri buluruz
non_normal_d = normal_d[normal_d == False].index

# Normal dağılıma uymayan ülkeleri yazdıralım
print("\nNormal dağılıma uymayan ülkeler:", non_normal_d)
```

Shapiro-Wilk Testi P Değerleri:

ülke

BRA 0.024995

IDN 0.045204

LUX 0.011244

Name: puan, dtype: float64

Normal dağılıma uymayan ülkeler: Index(['BRA', 'IDN', 'LUX'], dtype='object', name='ülke')

P değerleri 0.05'ten küçük olduğu için veriler normal dağılmıyordur. O yüzden parametrik olmayan testler uygulanacaktır.

1. **shapiro\_test = df.groupby('ülke')['puan'].apply(lambda x: shapiro(x).pvalue):** Bu satır, her bir ülke için Shapiro-Wilk testini uygular ve elde edilen p değerlerini **shapiro\_test** değişkenine kaydeder. **lambda x: shapiro(x).pvalue** ifadesi, her bir ülkenin puan verileri için Shapiro-Wilk testini uygulayan bir lambda fonksiyonudur.
2. **alpha = 0.05:** Bu satır, istatistiksel olarak anlamlı bir düzey belirler. Genellikle 0.05 kullanılır.

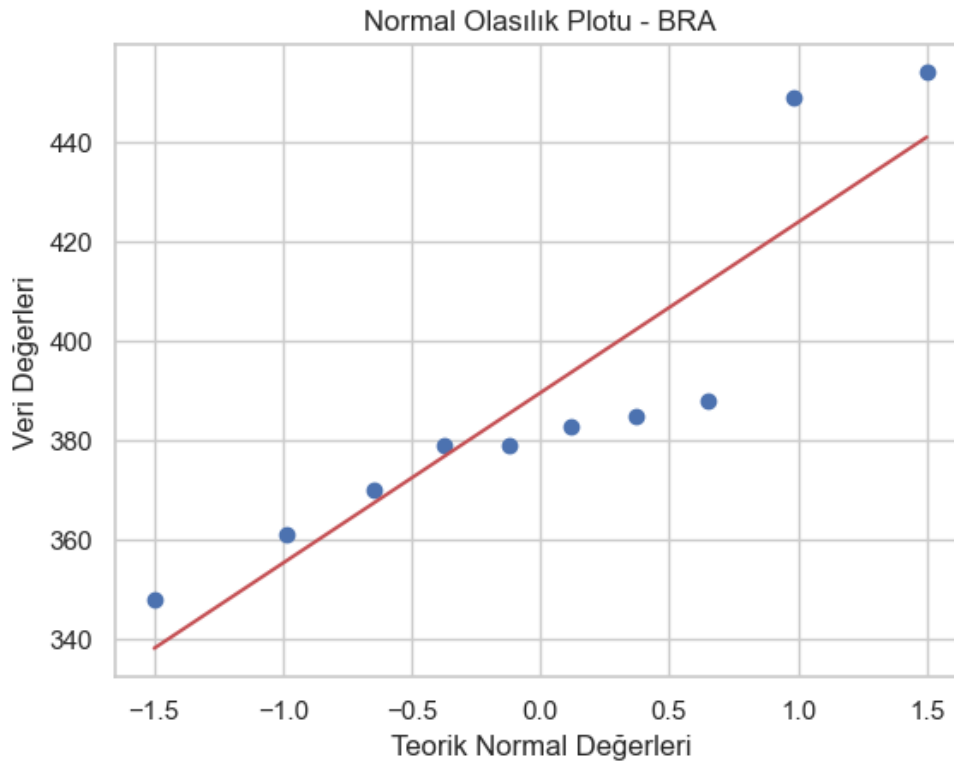
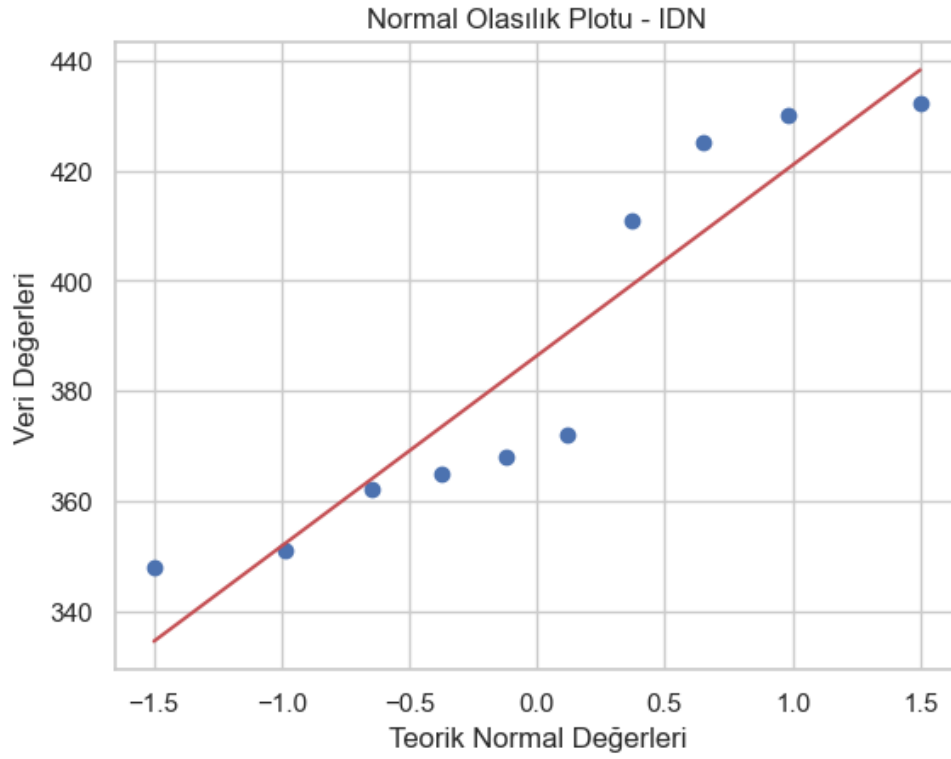
3. **print("Shapiro-Wilk Testi P Değerleri:")** ve **print(shapiro\_test)**: Bu satırlar, her bir ülke için elde edilen Shapiro-Wilk testi p değerlerini yazdırır.
4. **normal\_d = shapiro\_test > alpha**: Bu satır, her bir ülke için elde edilen p değerlerini, belirlenen anlamlılık düzeyi (**alpha**) ile karşılaştırarak normal dağılıma uyanları belirler. Bu, **normal\_distribution** adlı bir boolean serisi oluşturur.
5. **non\_normal\_d = normal\_d[normal\_d == False].index**: Bu satır, normal dağılıma uymayan ülkelerin indekslerini seçer ve **non\_normal\_countries** adlı bir liste oluşturur.
6. **print("\nNormal dağılıma uymayan ülkeler:", non\_normal\_d)**: Bu satır, normal dağılıma uymayan ülkelerin listesini yazdırır.

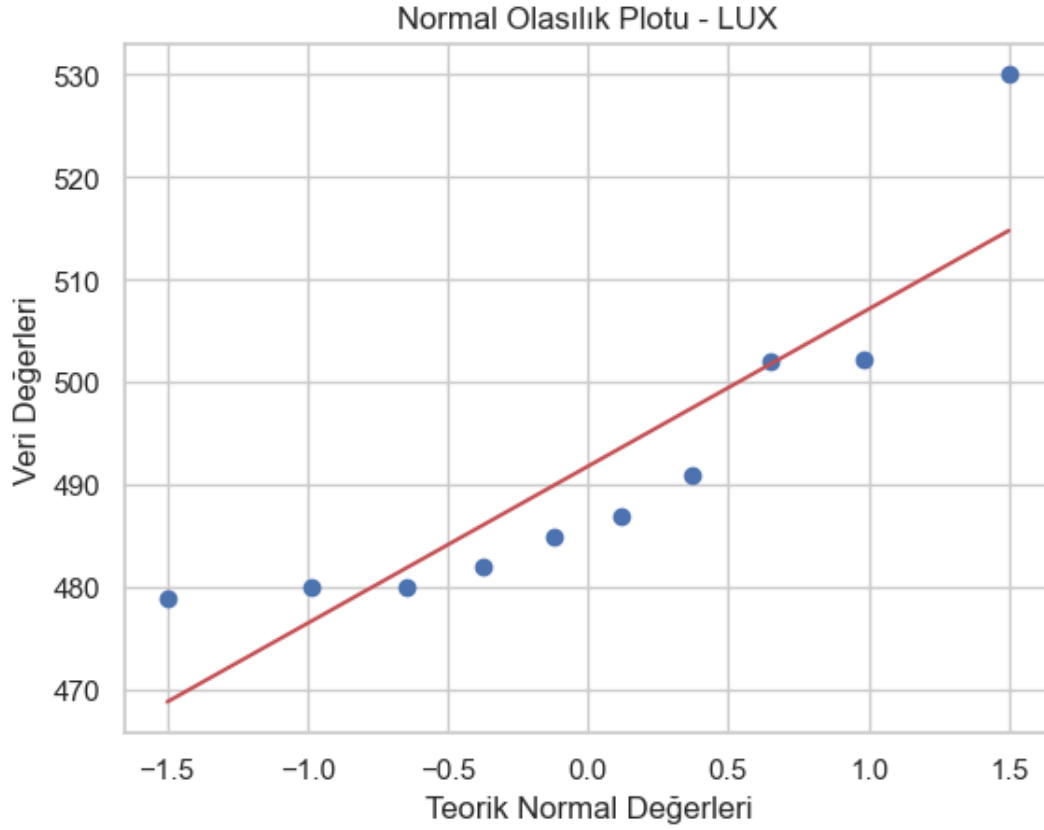
```
plt.figure(figsize=(12, 8))
sns.set(style="whitegrid")

for country in df['ülke'].unique():
    subset = df[df['ülke'] == country]['puan']
    stats.probplot(subset, dist="norm", plot=plt)
    plt.title(f'Normal Olasılık Plotu - {country}')
    plt.xlabel('Teorik Normal Değerleri')
    plt.ylabel('Veri Değerleri')
    plt.show()
```

1. **plt.figure(figsize=(12, 8))**: Bu satır, matplotlib kütüphanesinde bir figür oluşturur ve bu figürün boyutunu belirler (genişlik ve yükseklik).
2. **sns.set(style="whitegrid")**: Bu satır, seaborn kütüphanesinin görsel stilini belirler. "whitegrid" stili, bir beyaz arkaplan üzerine bir ızgara ekler.
3. **for country in df['ülke'].unique():**: Bu satır, veri çerçevesindeki 'ülke' sütunundaki unique (benzersiz) değerler (BRA, LUX, IDN) üzerinde bir döngü oluşturur.
4. **subset = df[df['ülke'] == country]['puan']**: Bu satır, her bir ülkenin puanlarını içeren bir alt küme (subset) oluşturur.
5. **stats.probplot(subset, dist="norm", plot=plt)**: Bu satır, scipy kütüphanesinin **probplot** fonksiyonunu kullanarak normal olasılık plotunu çizer. 'dist="norm"' ifadesi, teorik normal dağılımı temsil eder ve 'plot=plt' ifadesi, çizimin matplotlib figürüne eklenmesini sağlar.
6. **plt.title(f'Normal Olasılık Plotu - {country}')**: Bu satır, plotun başlığını belirler. Başlık, her bir ülkenin adını içerir.
7. **plt.xlabel('Teorik Normal Değerleri')**: Bu satır, x-ekseninin etiketini belirler. X-ekseni, teorik normal değerleri temsil eder.
8. **plt.ylabel('Veri Değerleri')**: Bu satır, y-ekseninin etiketini belirler. Y-ekseni, gerçek veri değerlerini temsil eder.

9. **plt.show()**: Bu satır, matplotlib figürünü görüntüler. Her bir ülke için ayrı bir normal olasılık plotu çizilir ve gösterilir.





Grafiklerden de anlaşılacağı üzere veriler doğrudan sapmış oldukları için normal dağılmadığını görüyoruz.

**2. En az 3 bağımlı gruptan oluşan bir veri kümesi bulunuz. Bulduğunuz veri kümesine dair özetleyici istatistikleri bularak yorumlayınız. Uygun grafiksel gösterimleri veriniz (Box-plot vb.). Ayrıca her bir grup için verilerin normal dağılmadığını, dolayısıyla parametrik testlerin uygulanacağını gerekli testler ve grafikler ile gösteriniz**

```
# Excel dosyasının yolu ve adı
import pandas as pd
bagımlı = 'C:\\Users\\HANDENUR\\OneDrive\\Masaüstü\\bağımlıveri.xlsx'

df2 = pd.read_excel(bağımlı)
```

Bağımlı veriyi excelden çağırdım ve data frame yapısına çevirip df2 adını verdim.

```
summary_stats = df2.describe()
print(summary_stats)
```

	vuruşdoğruluğu	saldırısavunması	altetmesavunması
count	40.000000	40.000000	40.000000
mean	39.400000	44.225000	29.250000
std	27.575351	21.438627	42.932117
min	0.000000	0.000000	0.000000
25%	19.000000	36.500000	0.000000
50%	42.000000	45.500000	0.000000
75%	60.500000	58.000000	69.750000
max	100.000000	80.000000	100.000000

Verilerin çeyrekliklerini, ortalamalarını ve standart sapmalarını görüyoruz. Vuruş doğruluğu sağa çarpıktır.

### Normallik incelenmesi.

Verilerin normal dağılıma uyup uymadığını görmek için shapiro testi yaparız.

### Hipotez:

$H_0 = \text{veriler normal dağılıma uygundur.}$

$H_1 = \text{veriler normal dağılıma uygun değildir.}$

Eğer p değeri  $\alpha=0.05$ 'ten küçükse hipotez reddedilir ve verilerin normal dağılıma uymadığı söylenir.

```
from scipy.stats import shapiro
for column in df2.columns:
    stat, p_value = shapiro(df2[column])
    print(f'{column}: p-value = {p_value}')
```

vuruşdoğruluğu: p-value = 0.04871165752410889  
saldırısavunması: p-value = 0.0066928984597325325  
altetmesavunması: p-value = 1.3604159576630082e-08

1. **from scipy.stats import shapiro:** Bu satır, SciPy kütüphanesinden Shapiro-Wilk testi fonksiyonunu içe aktarır. Shapiro-Wilk testi, bir veri setinin normal dağılıma uygun olup olmadığını değerlendirmek için kullanılır.
2. **for column in df2.columns::** Bu satır, **df2** veri çerçevesinin sütunlarında döngü yapar. Yani, her bir sütunun üzerinde tek tek işlem yapmak üzere bir döngü başlatılır.
3. **stat, p\_value = shapiro(df2[column]):** Bu satır, her sütunun verisini alır ve bu veri üzerinde Shapiro-Wilk testini uygular. **shapiro** fonksiyonu, test istatistiği (**stat**) ve p değeri (**p\_value**) olmak üzere iki değer döndürür.

4. **print(f'{column}: p-value = {p\_value}')**: Bu satır, her bir sütun için elde edilen p değerini ekrana yazdırır. Bu değer, testin null hipotezinin (verinin normal dağılıma sahip olduğu hipotezi) geçerliliğini gösterir. Eğer p değeri belirli bir anlamlılık düzeyinden küçükse (genellikle 0.05), yokluk hipotezi reddedilir ve veri setinin normal dağılıma uymadığı düşünülür.

Shapiro test sonucu p değeri verilerin normal dağılıma uygun olmadığını böylece parametrik olmayan testler yapılacağını görmüş olduk.

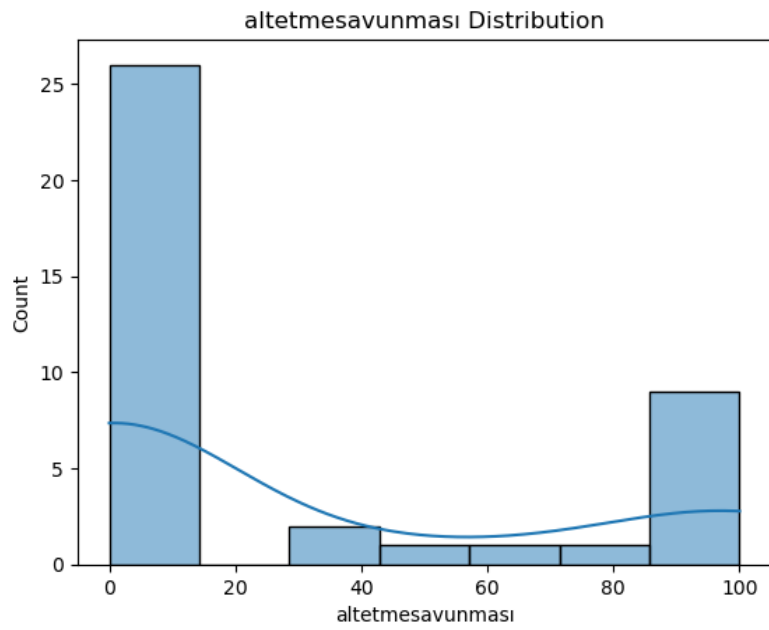
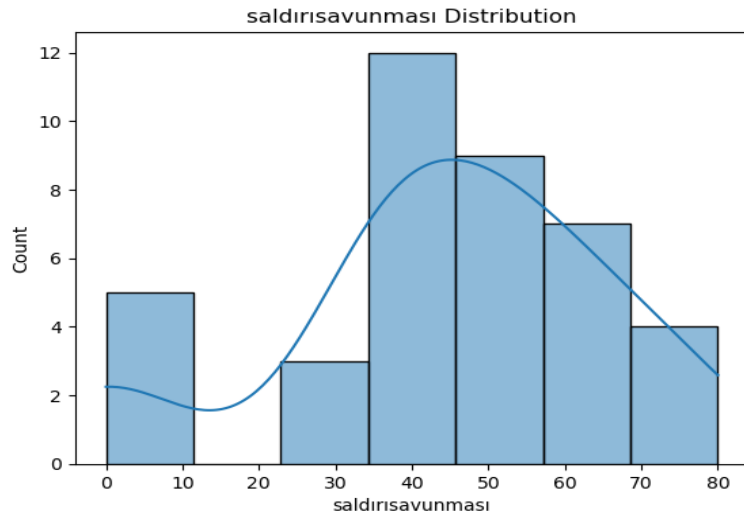
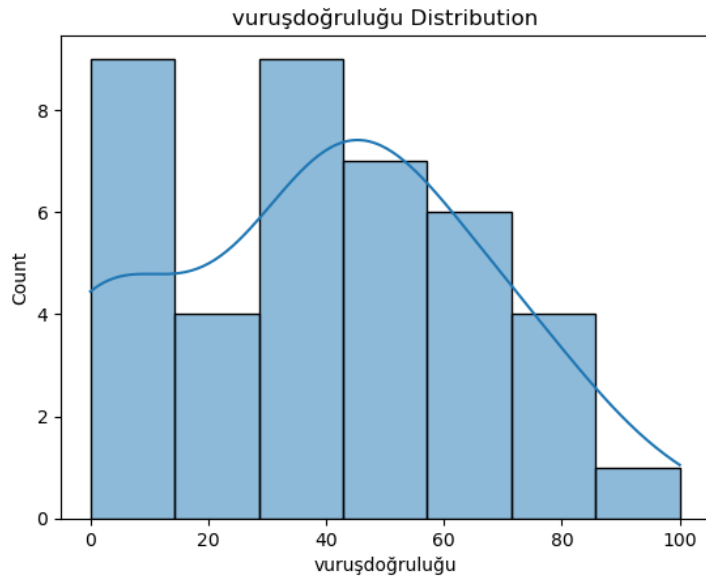
Grafik ile inceleyelim:

```
import seaborn as sns
import matplotlib.pyplot as plt
for column in df2.columns:
    sns.histplot(df2[column], kde=True)
    plt.title(f'{column} Distribution')
    plt.show()
```

Kütüphaneleri çalıştırdıktan sonra :

1. **for column in df2.columns::** Bu satır, **df2** veri çerçevesindeki her bir sütun üzerinde döngü yapar.
2. **sns.histplot(df2[column], kde=True)**: Bu satır, Seaborn kütüphanesinin **histplot** fonksiyonunu kullanarak, belirli bir sütunun histogramını çizer. **kde** (kernel density estimate veya çekirdek yoğunluk tahmini), bir veri setinin olasılık yoğunluk fonksiyonunu (PDF) yumuşatılmış bir tahminini sağlayan bir istatistiksel yöntemdir. **kde=True** parametresi kullanıldığında, histogramın üzerine bu yoğunluk tahmini eklenir.
3. **plt.title(f'{column} Distribution')**: Bu satır, her bir histogramın başlığını belirler. Başlık, o sütundaki verilerin dağılımını temsil eder.
4. **plt.show()**: Bu satır, oluşturulan histogramı ekrana gösterir. Her bir sütun için bir histogram oluşturulur ve ekrana sırayla gösterilir.





Görüldüğü gibi grafikler normal dağılım grafiği gibi dağılmamış.

3. 1’de bulduğunuz veri kümesinden herhangi bir grubu seçiniz. Veriye dair gerekli hipotezleri oluşturduktan sonra, tek örneklem konum testlerini (*İşaret Testi* ve *Wilcoxon İşaret Sıra Sayıları Testi*) uygulayınız ve elde ettiğiniz sonuçları yorumlayınız.

### Hipotezleri kuralım:

$$H_0: \theta = 370$$

$$H_A: \theta \neq 370$$

### İşaret testi:

Python’da işaret testi için kullanılacak kütüphane yok. Ama binom test ile işaret testini hesaplayabiliyoruz.

```
import numpy as np
from scipy.stats import binomtest

data_bra=np.array([385, 388, 348, 361, 379, 382.925, 370, 379, 448.984, 454])
medyan=370

nPlus=len(data_bra[data_bra>medyan])
nNeg=len(data_bra[data_bra<medyan])
n=nPlus+nNeg

binomtest(nPlus,n,p=0.5,alternative='two-sided')

BinomTestResult(k=7, n=9, alternative='two-sided', statistic=0.7777777777777778, pvalue=0.1796875)
```

NumPy kütüphanesi sayısal işlemler için kullanılır ve SciPy kütüphanesi istatistiksel fonksiyonlar içerir. Scipy.stats kütüphanesi içinden binom testi çağırdık. Bu iki kütüphaneyi aktifleştirdikten sonra Brezilya(BRA) verilerini numpy dizisine (np.array) dönüştürüp data\_bra adını verdim .

NumPy dizileri, Python listelerinden daha hızlı ve bellek açısından daha etkili bir veri yapısıdır. Ayrıca, NumPy, matematiksel ve istatistiksel işlemleri kolaylaştıran bir dizi fonksiyon ve metod içerir.

**nPlus** pozitif gözlem sayısı, **n** toplam gözlem sayısı, **p** olasılık başarı için (burada 0.5 çünkü iki taraflı test yapılıyor,binom ) ve **alternative** testin iki yönlü ('two-sided') olduğunu belirtir. Nplus değerleini alırsız çünkü işaret testinde “+” işaretli gözlem sayıları sayılır.

Çıktıya baktığımızda p değerinin 0.17 olduğunu görüyoruz.  $p > \alpha = 0.05$  olduğu için  $H_0$  reddedilemez. Yani Brezilyalı öğrencilerin sınav puanlarının medyanının 370’ den farklı olmadığı söylenir.

## Rstudio ile çözümü :

```
install.packages("DescTools")
library(DescTools)
bra=c( 385, 388, 348, 361, 379, 382.925, 370, 379, 448.984, 454)
SignTest(x=bra,mu=370,alternative="two.sided",conf.level=0.95,paired=TRUE)
```

### One-sample sign-Test

```
data:  bra
S = 7, number of differences = 9, p-value = 0.1797
alternative hypothesis: true median is not equal to 370
97.9 percent confidence interval:
 361.000 448.984
sample estimates:
median of the differences
          380.9625
```

Rstudioda işaret testi yapmak için önce descTools paketini çağırmalıyız.

p değerinin 0.17 olduğunu görüyoruz.  $p > \alpha = 0.05$  olduğu için  $H_0$  reddedilemez. Yani Brezilyalı öğrencilerin sınav puanlarının medyanının 370' den farklı olmadığı söylenir.

## Wilcoxon testi:

### Hipotezleri kuralım:

$$H_0: \theta = 400$$

$$H_A: \theta \neq 400$$

Wilcoxon için pythonda kütüphane vardır. Scipy.stats kütüphanesinden wilcoxon testini çağırdık.

```
from scipy.stats import wilcoxon
data_bra=np.array([385, 388, 348, 361, 379, 382.925, 370, 379, 448.984, 454])
medyan=400
wilcoxon(data_bra-medyan,alternative= 'two-sided',method='exact',correction=True)

WilcoxonResult(statistic=18.0, pvalue=0.375)
```

Wilcoxon testinde gözlem değerlerinden medyan değerlerini çıkarıp rank değerlerini buluruz. Alternatif hipotez eşitsizlik şeklinde olduğu için two-sided yazılır.

Çıktıda p değeri 0.375 çıkmış.  $p > \alpha = 0.05$  olduğu için  $H_0$  reddedilemez. Yani Brezilyalı öğrencilerin sınav puanlarının medyanı 400' den farklı değildir.

### Rstudio ile çözümü:

```
install.packages("stats")
library(stats)
bra=c( 385, 388, 348, 361, 379, 382.925, 370, 379, 448.984, 454)
wilcox.test(bra,mu=400,alternative="two.sided",exact=TRUE)
```

```
      wilcoxon signed rank test with continuity correction

data:  bra
V = 18, p-value = 0.3586
alternative hypothesis: true location is not equal to 400
```

Çıktıda p değeri 0.35 çıkmış.  $p > \alpha = 0.05$  olduğu için  $H_0$  reddedilemez. Yani Brezilyalı öğrencilerin sınav puanlarının medyanı 400' den farklı değildir.

**4)1'de bulduğunuz veri kümesinden herhangi iki grubu seçiniz. Veriye dair gerekli hipotezleri oluşturduktan sonra gruplar arasında konum yönünden fark olup olmadığını bağımsız iki örneklem testi (*Mann-Whitney Testi*) ile araştırınız.**

Lüksemburglu öğrencilerin sınav puanları ile Endonezyalı öğrencilerin sınav puanları arasında anlamlı bir fark olduğu hipotezi test edilsin.

### Hipotezleri kuralım:

$$H_0: \theta_{lux} = \theta_{idn}$$

$$H_A: \theta_{lux} \neq \theta_{idn}$$

```
import numpy as np
from scipy import stats

lux=np.array([530,480,502.193,491,487,485,482,479,502,480])
idn=np.array([362,365,351,410.899,430,425,348,368,372,432.140])

stats.mannwhitneyu(lux,idn, alternative= 'two-sided')

MannwhitneyuResult(statistic=100.0, pvalue=0.00018165114609146497)
```

P değeri 0.00018165114609146497 çıkmış.

$P < \alpha = 0.05$  olduğundan yokluk hipotezi reddedilir. Yani Lüksemburglu öğrencilerin sınav puanları ile Endonezyalı öğrencilerin sınav puanları arasında anlamlı bir fark olduğu %5 anlamlılık düzeyinde söylenebilir.

5)2’de bulduğunuz veri kümesinden herhangi iki grubu seçiniz. Veriye dair gerekli hipotezleri oluşturduktan sonra gruplar arasında konum yönünden fark olup olmadığını bağımlı iki örneklem testi (*Wilcoxon İşaret Sıra Sayıları Testi*) ile araştırınız.

**İşaret testi:**

**Hipotezleri kuralım:**

$$H_0: \text{vurus} - \text{saldırı} = 0$$

$$H_A: \text{vurus} - \text{saldırı} \neq 0$$

UFC dövüş sporcularının vurus sayısı ve saldırı sayısı arasında fark olup olmadığını inceleyelim.

```
import numpy as np
from scipy.stats import binomtest

vurus=np.array([0, 77, 0, 33, 60, 0, 50, 66, 42, 0, 45, 62, 7, 42, 62, 79, 32, 50, 100, 33, 0, 40, 50, 66, 23, 40,
                50, 33, 68, 25, 0, 20, 50, 0, 16, 80, 50, 42, 0, 83])
saldırı=np.array([54, 43, 0, 0, 60, 75, 42, 38, 80, 0, 33, 0, 27, 35, 40, 75, 0, 35, 37, 46, 51, 58,
                  37, 37, 50, 37, 48, 55, 63, 74, 67, 51, 34, 55, 41, 54, 68, 45, 58, 66])

D=vurus-saldırı
nPlus=len(D[D>0])
nNeg=len(D[D<0])
n=nPlus+nNeg
binomtest(nPlus,n,p=0.5,alternative='two-sided')
```

```
BinomTestResult(k=20, n=37, alternative='two-sided', statistic=0.5405405405405406, pvalue=0.7428293587290682)
```

P değeri=0.74 çıkmış.  $P > \alpha = 0.05$  olduğundan yokluk hipotezi reddedilemez. Yani UFC dövüş sporcularının vurus sayısı ve saldırı sayısı arasında fark olmadığı %5 anlamlılık düzeyinde söylenebilir.

## Rstudio ile çözümü:

```
library(readxl)
bağımlıveri <- read_excel("C:/Users/HANDENUR/OneDrive/Masaüstü/NONPARÖDEV/bağımlıveri.xlsx")
x=bağımlıveri$vuruşdoğruluğu
y=bağımlıveri$saldırısavunması
library(DescTools)
signTest(x,y,alternative="two.sided",conf.level=0.95,paired=TRUE)
```

### Dependent-samples Sign-Test

```
data: x and y
S = 20, number of differences = 37, p-value = 0.7428
alternative hypothesis: true median difference is not equal to 0
96.2 percent confidence interval:
 -18 12
sample estimates:
median of the differences
1
```

$P > \alpha = 0.05$  olduğundan yokluk hipotezi reddedilemez. Yani UFC dövüş sporcularının vurus sayısı ve saldırı sayısı arasında fark olmadığı %5 anlamlılık düzeyinde söylenebilir.

## Wilcoxon testi:

$$H_0: \text{vurus} - \text{saldırı} = 0$$

$$H_A: \text{vurus} - \text{saldırı} \neq 0$$

```
import numpy as np
from scipy import stats

vurus=np.array([0, 77, 0, 33, 60, 0, 50, 66, 42, 0, 45, 62, 7, 42, 62, 79, 32, 50, 100, 33, 0, 40, 50, 66, 23, 40,
50, 33, 68, 25, 0, 20, 50, 0, 16, 80, 50, 42, 0, 83])
saldırı=np.array([54, 43, 0, 0, 60, 75, 42, 38, 80, 0, 33, 0, 27, 35, 40, 75, 0, 35, 37, 46, 51, 58,
37, 37, 50, 37, 48, 55, 63, 74, 67, 51, 34, 55, 41, 54, 68, 45, 58, 66])

stats.wilcoxon(vurus,saldırı, alternative='two-sided', method='approx', zero_method='zsplit', correction=True)

WilcoxonResult(statistic=371.5, pvalue=0.6094802938457935)
```

**n**Plus pozitif gözlem sayısı, **n** toplam gözlem sayısı, **p** olasılık başarı için (burada 0.5 çünkü iki taraflı test yapılıyor, binom ) ve **alternative** testin iki yönlü ('two-sided') olduğunu belirtir. Nplus değerleini alırız çünkü işaret testinde "+" işaretli gözlem sayıları sayılır.

P değeri 0.6 çıkmış.  $P > \alpha = 0.05$  olduğundan yokluk hipotezi reddedilemez. Yani vuruş ve saldırı sayılarının dövüş başarısını etkilemekte aralarında anlamlı bir farklılık olmadığı %5 anlamlılık düzeyinde söylenebilir.

### Rstudio ile çözümü:

```
library(readxl)
bağımlıveri <- read_excel("C:/Users/HANDENUR/OneDrive/Masaüstü/NONPARÖDEV/bağımlıveri.xlsx")
x=bağımlıveri$vuruşdoğruluğu
y=bağımlıveri$saldırısavunması
library(descTools)
wilcox.test(x,y,alternative="two.sided",conf.level=0.95,paired=TRUE)
```

wilcoxon signed rank test with continuity correction

```
data: x and y
V = 308.5, p-value = 0.5214
alternative hypothesis: true location shift is not equal to 0
```

Rstudioda wilcoxon testi yapmak için önce descTools paketini çağırmalıyız.

P değeri 0.52 çıkmış.  $P > \alpha = 0.05$  olduğundan yokluk hipotezi reddedilemez. Yani vuruş ve saldırı sayılarının dövüş başarısını etkilemekte aralarında anlamlı bir farklılık olmadığı %5 anlamlılık düzeyinde söylenebilir.

6) 1'deki veri kümesi için gruplar arasında konum yönünden fark olup olmadığını gerekli hipotezleri oluşturduktan sonra k örneklem konum testi (*Kruskal-Wallis Testi*) ile araştırınız. Gruplar arasında fark bulduğunuz takdirde, farkın hangi gruplardan kaynaklandığını bulmak için Post-Hoc testlerinden birini uygulayınız.

### Hipotezleri kuralım:

$$H_0: \theta_1 = \theta_2 = \theta_3$$
$$H_1: \text{en az bir } \theta_j \text{ farklıdır.}$$

```
import numpy as np
from scipy import stats
```

```
bra=[385, 388, 348, 361, 379, 382.925, 370, 379, 448.984, 454]
lux=[530,480,502.193,491,487,485,482,479,502,480]
idn=[362,365,351,410.899,430,425,348,368,372,432.140]
```

```
stats.kruskal(bra,lux,idn)
```

```
KruskalResult(statistic=19.513023152270705, pvalue=5.7916306776637384e-05)
```

P değeri 5.7916306776637384e-05 çıkmış.  $P < \alpha = 0,05$  olduğu için  $H_0$  reddedilir. En az bir  $\theta_j$ 'nin farklı olduğu %5 anlamlılık düzeyinde söylenebilir. Farklılığın nereden kaynaklandığını bulmak için post-hoc testi yaparız.

### Rstudio ile çözüm:

```
library(readxl)
library(PMCMRplus)

bağımsızveri1 <- read_excel("C:/Users/HANDENUR/OneDrive/Masaüstü/bağımsızveri1.xlsx")
bağımsızveri1$ülke <- as.factor(bağımsızveri1$ülke)

kruskal.test(puan~ülke,data=bağımsızveri1)

> kruskal.test(puan~ülke,data=bağımsızveri1)
```

```
kruskal-wallis rank sum test

data: puan by ülke
kruskal-wallis chi-squared = 19.513, df = 2, p-value = 5.792e-05
```

P-value=0.000005792 <  $\alpha = 0,05$  olduğu için  $H_0$  reddedilir. En az bir  $\theta_j$ 'nin farklı olduğu %5 anlamlılık düzeyinde söylenebilir. Farklılığın nereden kaynaklandığını bulmak için post-hoc testi yaparız.

$H_0$ : brezilya ve endonezya arasında puan bakımından fark yoktur.

$H_1$ : brezilya ve endonezya arasında puan bakımından fark vardır.

$H_0$ : brezilya ve lüksemburg arasında puan bakımından fark yoktur.

$H_1$ : brezilya ve lüksemburg arasında puan bakımından fark vardır.



$H_0$ : endonezya ve lüksemburg arasında puan bakımından fark yoktur.

$H_1$ : endonezya ve lüksemburg arasında puan bakımından fark vardır.

```
library(PMCMRplus)
PMCMRplus::kwAllPairsNemenyiTest(x=bağımsızveri1$puan,bağımsızveri1$ülke, method="chisq")
```

```
      BRA      IDN
IDN 0.92311 -
LUX 0.00086 0.00019
```

PMCMRplus paketi, parametrik olmayan çoklu karşılaştırma testleri için çeşitli işlevler sağlar.

Brezilya ve Lüksemburg p değeri  $0.00086 < 0.05$  olduğu için  $H_0$  reddedilir. Aralarında anlamlı bir fark olduğu %5 anlamlılık düzeyinde söylenebilir.

Endonezya ve Lüksemburg p değeri  $0.00019 < 0.05$  olduğu için  $H_0$  reddedilir. Aralarında anlamlı bir fark olduğu %5 anlamlılık düzeyinde söylenebilir.

Farkın Lüksemburg sınav puanlarından kaynaklandığını %95 güven düzeyinde söyleyebiliriz.

**7) 2'deki veri kümesi için konum yönünden gruplar arasında fark olup olmadığını gerekli hipotezleri oluşturduktan sonra k örneklem konum testi (*Friedman Testi*) ile araştırınız. Gruplar arasında fark bulduğunuz takdirde, farkın hangi gruplardan kaynaklandığını bulmak için Post-Hoc testlerinden birini uygulayınız.**

**Hipotezleri kuralım:**

$$H_0: \theta_1 = \theta_2 = \theta_3$$
$$H_1: \text{en az bir } \theta_j \text{ farklıdır.}$$

```
from scipy import stats
```

```
vurus=[0, 77, 0, 33, 60, 0, 50, 66, 42, 0, 45, 62, 7, 42, 62, 79, 32, 50, 100, 33, 0, 40, 50, 66, 23, 40,
        50, 33, 68, 25, 0, 20, 50, 0, 16, 80, 50, 42, 0, 83]
saldırı=[54, 43, 0, 0, 60, 75, 42, 38, 80, 0, 33, 0, 27, 35, 40, 75, 0, 35, 37, 46, 51, 58,
        37, 37, 50, 37, 48, 55, 63, 74, 67, 51, 34, 55, 41, 54, 68, 45, 58, 66]
savunma=[0, 100, 0, 100, 0, 0, 66, 40, 0, 100, 100, 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 100, 50, 0, 81, 0, 0, 0,
        100, 0, 0, 0, 100, 0, 100, 33]
```

```
stats.friedmanchisquare(vurus,saldırı,savunma)
```

```
FriedmanchisquareResult(statistic=5.482993197278937, pvalue=0.06447378327637059)
```

P değeri 0.06447378327637059 .  $P > \alpha = 0,05$  olduğu için  $H_0$  reddedilemez. Yani gruplar arasında anlamlı bir fark olmadığı %5 anlamlılık düzeyinde söylenebilir. Yokluk hipotezi reddedilemediği için post-hoc testlerine bakılmaz.

### 8) Eğilim testi için uygun bir veri kümesi bulup Mann-Kendall testini uygulayınız.

Eğilim testi veri seti, Microsoft'un 2015-2021 yıllarındaki hisse senedi bilgileri günlük olarak kaydedilmiş verilerden oluşuyor. Ben sadece 2015 yılından 60 günlük zaman serisini ve bu günlerdeki en yüksek fiyat miktarlarını (tavan) aldım.

### Hipotez:

$H_0$ : trend yoktur.

$H_1$ : trend vardır.

### Rstudio çözüm:

```
library(readxl)
eğilimtesti <- read_excel("C:/Users/HANDENUR/OneDrive/Masaüstü/NONPARÖDEV/eğilimtesti.xlsx")
install.packages("trend")
library(trend)
mk.test(eğilimtesti$tavan fiyat)
```

```

> mk.test(eğilimtesti$tavan fiyat`)

      Mann-Kendall trend test

data:  eğilimtesti$tavan fiyat`
z = 0.91853, n = 60, p-value = 0.3583
alternative hypothesis: true S is not equal to 0
sample estimates:
      S      varS      tau
1.450000e+02 2.457767e+04 8.203686e-02

```

Trend paketi zaman serisi verilerinde trend varlığı olup olmadığını test etmek için kullanılan çeşitli istatistiksel testler sağlar. mk.test işlevi, Mann-Kendall testi olarak da bilinen bir nonparametrik testtir. Bu test, zaman serisi verilerinde monoton bir trend olup olmadığını belirlemek için kullanılır.

P-value = 0.3583 >  $\alpha = 0,05$  olduğu için  $H_0$  reddedilemez. Yani tavan fiyatları verisinde trend olmadığını %5 anlamlılık düzeyinde söyleyebiliriz.

### Kaynakça :

- Gamgam, H., & Altunkaynak, B. (2013). Parametrik Olmayan Yöntemler. Ankara: Seçkin Yayıncılık.
- Tatlıdil, H. (1996). Parametrik Olmayan İstatistiksel Yöntemler. Yayımlanmamış ders notu. İstatistik Bölümü, Hacettepe Üniversitesi, Ankara.
- <https://rpubs.com/esmacalisir/rilenonparametrik> 20 aralık 2023 tarihinde erişildi.