



HACETTEPE
ÜNİVERSİTESİ

FEN FAKÜLTESİ İSTATİSTİK BÖLÜMÜ
2023-2024 YILI BAHAR DÖNEMİ
İST377 PARAMETRİK OLMAYAN
İSTATİSTİKSEL YÖNTEMLER ÖDEVİ

Öğretim Üyesi:
Doç. Dr. Semra TÜRKAN

Hazırlayan:
Zehra KELEŞ
2210329063

İçindekiler

1. 3 Bağımsız Gruptan Oluşan Veri Kümesi:	2
ÖRNEK 1:.....	2
Özetleyici istatistikler:.....	3
Normallik Testi:.....	4
➔ Histogram Görselleri:.....	4
➔ Q-Q Plots Görseli:	6
➔ Boxplot Görseli:	6
3. Tek Örneklem Konum Testleri:.....	7
➔ İşaret Testi	7
➔ Wilcoxon Testi	8
4. Bağımsız İki Örneklem Testi:	9
➔ Mann-Whitney U Testi.....	9
6. k Örneklem Konum Testi:	10
➔ Kruskal-Wallis Testi:.....	10
2. 3 Bağımlı Gruptan Oluşan Veri Kümesi:	11
ÖRNEK 2:.....	11
Özetleyici istatistikler:.....	12
Normallik Testi:.....	13
➔ Histogram Görselleri:.....	13
➔ Q-Q Plots Görseli:	15
➔ Boxplot Görseli:	15
5. Bağımlı İki Örneklem Testi.....	16
➔ Wilcoxon Testi:	16
7. k Örneklem Konum Testi	17
➔ Friedman Testi	17
8. Eğilim Testi	18
ÖRNEK 3:.....	18
Python Kodları:.....	19
1.soru:.....	19
2.soru:.....	22
3.soru:.....	24
RSTUDIO KODLARI.....	26
Soru1.....	26
Soru2.....	27
Soru3.....	27
Kaynakça	28

1. 3 Bağımsız Gruptan Oluşan Veri Kümesi:

ÖRNEK 1:

24 buzağıya verilen 4 farklı diyet sonucu oluşan ağırlık artışları aşağıda verilmiştir. Bu amaçla her grupta 6 gözlem olmak üzere 24 gözlemlik bir örneklem üzerinde çalışmıştır. Diyetlerin buzağılar üzerinde anlamlı bir fark olup olmadığını araştırınız.

Gözlem No	diyet1	Gözlem No	diyet2	Gözlem No	diyet3	Gözlem No	diyet4
1	12	7	18	13	10	19	19
2	10	8	19	14	12	20	20
3	13	9	18	15	13	21	18
4	11	10	18	16	16	22	19
5	12	11	19	17	14	23	18
6	9	12	19	18	13	24	19

H_0 : Diyetlerin arasında fark yoktur.

H_1 : Diyetlerin arasında fark vardır.

Özetleyici istatistikler:

```
In [1]: import pandas as pd
```

```
In [2]: veri = pd.read_excel("Kitap1.xlsx") #excel den aktarılan veri okunur
print(veri)
```

	Gözlem No	diyet1	Gözlem No.1	diyet2	Gözlem No.2	diyet3	\
0	1	12	7	18	13	10	
1	2	10	8	19	14	12	
2	3	13	9	18	15	13	
3	4	11	10	18	16	16	
4	5	12	11	19	17	14	
5	6	9	12	19	18	13	

	Gözlem No.3	diyet4
0	19	19
1	20	20
2	21	18
3	22	19
4	23	18
5	24	19

```
In [3]: df = pd.DataFrame({'diyet1':[12,10,13,11,12,9],
                           'diyet2':[18,19,18,18,19,19],
                           'diyet3':[10,12,13,16,14,13],
                           'diyet4':[19,20,18,19,18,19]})
print(df["diyet1"].describe())
print(df["diyet2"].describe())
print(df["diyet3"].describe())
print(df["diyet4"].describe())
```

```
count      6.000000
mean       11.166667
std        1.471960
min        9.000000
25%       10.250000
50%       11.500000
75%       12.000000
max       13.000000
Name: diyet1, dtype: float64
count      6.000000
mean       18.500000
std        0.547723
min       18.000000
25%       18.000000
50%       18.500000
75%       19.000000
max       19.000000
Name: diyet2, dtype: float64
count      6.00
mean       13.00
std        2.00
min       10.00
25%       12.25
50%       13.00
75%       13.75
max       16.00
Name: diyet3, dtype: float64
count      6.000000
mean       18.833333
std        0.752773
min       18.000000
25%       18.250000
50%       19.000000
75%       19.000000
max       20.000000
Name: diyet4, dtype: float64
```

Normallik Testi:

```
In [4]: from scipy.stats import shapiro  
shapiro(df["diyet1"])
```

```
Out[4]: ShapiroResult(statistic=0.958012044429779, pvalue=0.8042958378791809)
```

```
In [5]: from scipy.stats import shapiro  
shapiro(df["diyet2"])
```

```
Out[5]: ShapiroResult(statistic=0.6826766729354858, pvalue=0.004039337392896414)
```

```
In [6]: from scipy.stats import shapiro  
shapiro(df["diyet3"])
```

```
Out[6]: ShapiroResult(statistic=0.9764894247055054, pvalue=0.9328367710113525)
```

```
In [7]: from scipy.stats import shapiro  
shapiro(df["diyet4"])
```

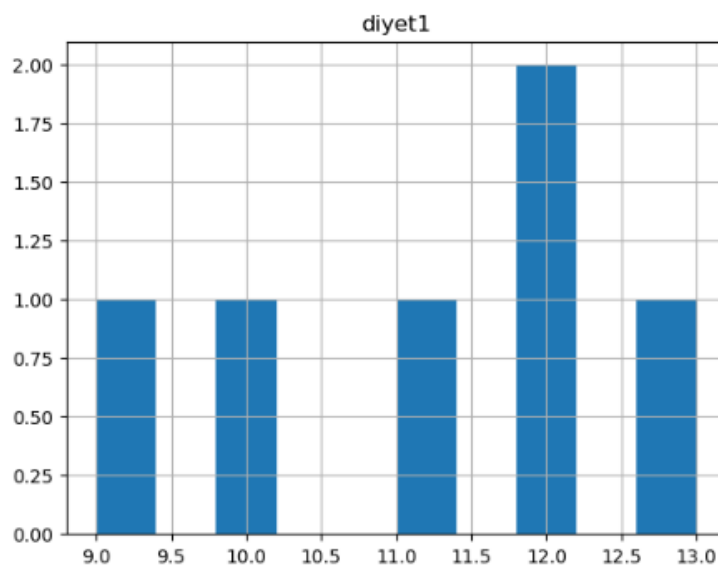
```
Out[7]: ShapiroResult(statistic=0.8662615418434143, pvalue=0.2117055505514145)
```

- ➔ diyet2'nin p değeri ($0,0040 < 0,05$) olduğu için normallik varsayımının sağlanmadığı %5 anlamlılık düzeyinde söylenebilir.
- ➔ diyet1 ve diyet3 p değeri 0,05'ten büyük olduğu için normallik varsayımının sağlandığı %5 anlamlılık düzeyinde söylenebilir.

➔ Histogram Görselleri:

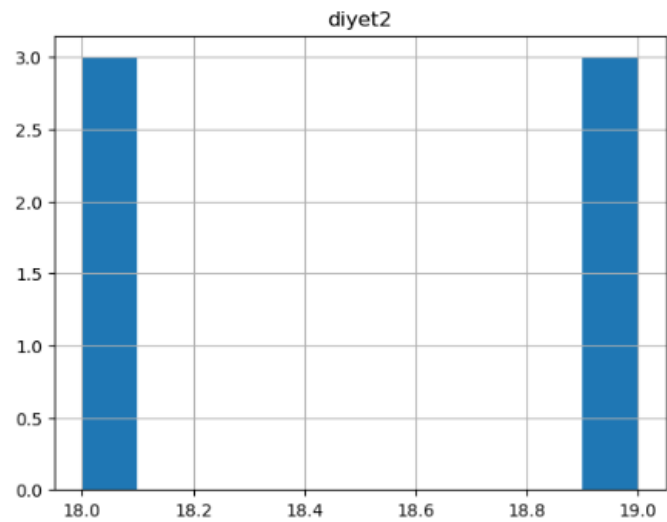
```
In [10]: df.hist('diyet1')
```

```
Out[10]: array([[<Axes: title={'center': 'diyet1'}>]], dtype=object)
```



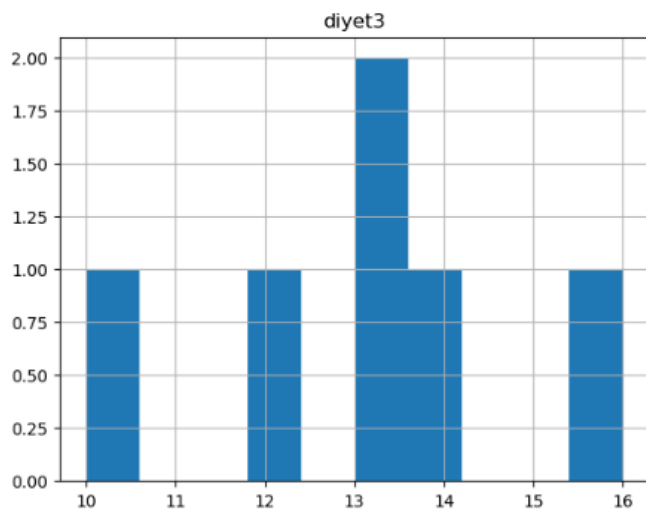
```
In [11]: df.hist('diyet2')
```

```
Out[11]: array([[<Axes: title={'center': 'diyet2'}>]], dtype=object)
```



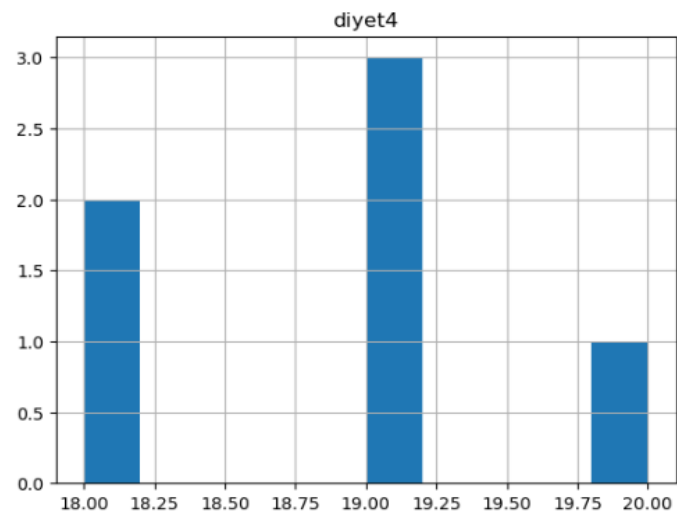
```
In [12]: df.hist('diyet3')
```

```
Out[12]: array([[<Axes: title={'center': 'diyet3'}>]], dtype=object)
```



```
In [13]: df.hist('diyet4')
```

```
Out[13]: array([[<Axes: title={'center': 'diyet4'}>]], dtype=object)
```

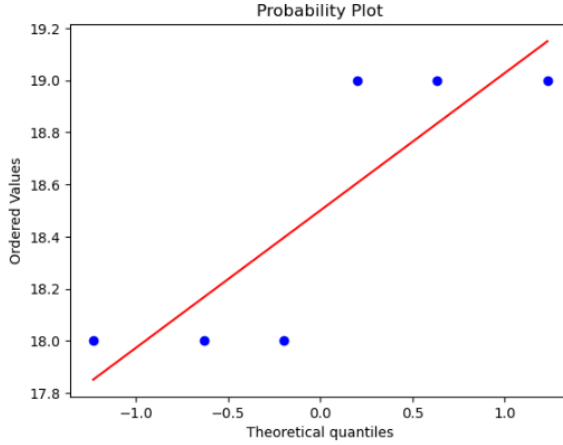


→ Q-Q Plots Görseli:

```
In [14]: # Q-Q Plots Görseli
import scipy.stats as stats
import matplotlib.pyplot as plt

In [17]: df = df[['diyet2']]
stats.probplot(df['diyet2'], dist="norm", plot=plt)

Out[17]: ((array([-1.23132171, -0.63003387, -0.19819716,  0.19819716,  0.63003387,
        1.23132171]),
        array([18, 18, 18, 19, 19, 19], dtype=int64)),
        (0.5274472446069902, 18.5, 0.851001534616253))
```



**Bu grafiğe bakarak noktaların normal dağılım çizgisinden uzak olduğunu yani verilerimizin normal dağılmadığını söyleyebiliriz.*

→ Boxplot Görseli:

```
In [12]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

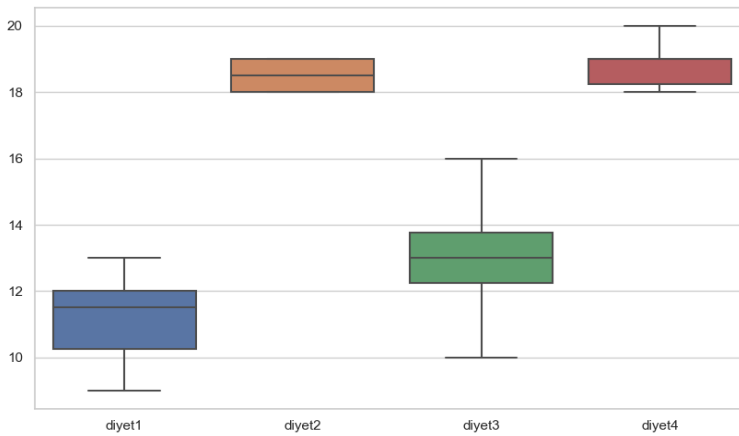
veriler = {'diyet1': [12, 10, 13, 11, 12, 9],
          'diyet2': [18, 19, 18, 18, 19, 19],
          'diyet3': [10, 12, 13, 16, 14, 13],
          'diyet4': [19, 20, 18, 19, 18, 19]}

df = pd.DataFrame(veriler)

# Boxplot çizimi
sns.set(style="whitegrid") # Stil ayarı isteğe bağlıdır
plt.figure(figsize=(10, 6)) # Grafiğin boyutu isteğe bağlıdır

sns.boxplot(data=df)

# Grafiği gösterme
plt.show()
```



3. Tek Örneklem Konum Testleri:

→ İşaret Testi

H_0 : 4. diyetle kilo artışlarının ortancası 15'a eşit veya daha küçüktür.

H_5 : 4. diyetle kilo artışlarının ortancası 15'ten fazladır.

Python:

```
In [3]: diyet4 = np.array([19, 20, 18, 19, 18, 19])
        median = 15
        n = len(diyet4)
        nPlus = len(diyet4[diyet4>median])
        nNeg = len(diyet4[diyet4<median])
        n = nPlus + nNeg

        stats.binomtest(nPlus, n, p=0.5, alternative="greater")

Out[3]: BinomTestResult(k=6, n=6, alternative='greater', statistic=1.0, pvalue=0.015625)
```

RStudio:

```
> library(DescTools)
> diyet4=c(19,20,18,19,18,19)
> SignTest(x=diyet4,mu=15,alternative="greater")

One-sample Sign-Test

data: diyet4
S = 6, number of differences = 6, p-value = 0.01563
alternative hypothesis: true median is greater than 15
98.4 percent confidence interval:
 18 Inf
sample estimates:
median of the differences
      19
```

**p-değeri (0.015<0.05) H_0 reddedilir.*

Yorum: 4. diyetin sonunda kilo artışlarının ortancasının 15'ten fazla olduğunu %95 güven düzeyinde söyleyebiliriz.

→ Wilcoxon Testi

H_0 : 1. diyetle kilo artışlarının ortancası 17'ye eşit veya daha büyüktür.

H_5 : 1. diyetle kilo artışlarının ortancası 17'den küçüktür.

Python:

```
In [3]: import numpy as np
        from scipy import stats
        diyet1=np.array([12,10,13,11,12,9])
        median = 17
        stats.wilcoxon(diyet1-median,
                        alternative='less',
                        method='approx',
                        zero_method='zsplit',
                        correction=True)

Out[3]: WilcoxonResult(statistic=0.0, pvalue=0.01776116831170729)
```

RStudio:

```
> library(stats)
> diyet1=c(12,10,13,11,12,9)
> wilcox.test(x,mu=17,alternative="less")

wilcoxon signed rank test with continuity correction

data:  x
V = 7.5, p-value = 0.02328
alternative hypothesis: true location is less than 17
```

**p-değeri (0.017<0.05) H_0 reddedilir.*

Yorum: 1. diyetin sonundaki kilo artışlarının ortancasının 17'den küçük olduğunu %95 güven düzeyinde söyleyebiliriz.

4. Bağımsız İki Örneklem Testi:

→ Mann-Whitney U Testi

H_0 : 1. ve 2. diyet sonrası kilo artışları arasında fark yoktur.

H_s : 1. ve 2. diyet sonrası kilo artışları arasında fark vardır.

Python:

```
In [20]: from scipy.stats import mannwhitneyu

stat, p_value = mannwhitneyu(df['diyet1'], df['diyet2'])
print(f"Mann-Whitney U Testi İstatistiği: {stat}")
print(f"P-değeri: {p_value}")

Mann-Whitney U Testi İstatistiği: 0.0
P-değeri: 0.004407757995697681
```

RStudio:

```
> ##MANN-WHITNEY U TESTİ
> diyet1 <- c(12,10,13,11,12,9)
> diyet2 <- c(18,19,18,18,19,19)
> wilcox.test(diyet1,diyet2)

      wilcoxon rank sum test with continuity correction

data:  diyet1 and diyet2
W = 0, p-value = 0.004408
alternative hypothesis: true location shift is not equal to 0
```

Yorum: p-değeri 0,05'ten küçük olduğu için H_0 reddedilir. Yani 1. ve 2.diyet sonrası kilo artışları arasında %95 güven düzeyinde fark olduğu söylenebilir.

H_0 : 1. ve 4. diyet sonrası kilo artışları arasında fark yoktur.

H_s : 1. diyet sonrası kilo artışlarının 4. diyet sonrası kilo artışından daha küçüktür.

Python:

```
In [11]: import numpy as np
from scipy import stats
diyet1 = np.array([12, 10, 13, 11, 12, 9])
diyet4 = np.array([19, 20, 18, 19, 18, 19])
stats.mannwhitneyu(diyet2, diyet4, alternative = 'less')

Out[11]: MannwhitneyuResult(statistic=13.5, pvalue=0.2380167464712148)
```

RStudio:

```
> ##MANN-WHITNEY U TESTİ
> diyet1 <- c(12,10,13,11,12,9)
> diyet4 <- c(19,20,18,19,18,19)
> wilcox.test(diyet1,diyet4)

      wilcoxon rank sum test with continuity correction

data:  diyet1 and diyet4
W = 0, p-value = 0.004624
alternative hypothesis: true location shift is not equal to 0
```

Yorum: p-değeri 0,05'ten büyük olduğu için H_0 kabul edilebilir. Yani 1. diyet sonrası kilo artışlarının 4.diyet sonrası kilo artışlarından küçük olduğunu %95 güven düzeyinde söyleyebiliriz.

6. k Örneklem Konum Testi:

→ Kruskal-Wallis Testi:

H_0 : Diyet sonrası kilo artışları arasında fark yoktur.

H_5 : Diyet sonrası kilo artışları arasında fark vardır.

Python:

```
In [22]: from scipy.stats import kruskal

stat, p_value = kruskal(df['diyet1'], df['diyet2'], df['diyet3'], df['diyet4'])
print(f"Kruskal-Wallis Testi İstatistiği: {stat}")
print(f"P-değeri: {p_value}")

Kruskal-Wallis Testi İstatistiği: 18.669499105545622
P-değeri: 0.00031996703065528

In [23]: import numpy as np
from scipy import stats

diyet1=[12, 10, 13, 11, 12, 9]
diyet2=[18, 19, 18, 18, 19, 19]
diyet3=[10, 12, 13, 16, 14, 13]
diyet4=[19, 20, 18, 19, 18, 19]

stats.kruskal(diyet1, diyet2, diyet3, diyet4)

Out[23]: KruskalResult(statistic=18.669499105545622, pvalue=0.00031996703065528)
```

RStudio:

```
> # Veriyi dosyadan okuma
> veri1 <- read_excel("c:/Users/Lenovo/Desktop/soru1.xlsx")
> # Grup tanımları
> grup <- rep(1:6, each = 8)
> veri <- unlist(veri1)
> grup <- unlist(grup)
> # Veri çerçevesi oluşturma
> df <- data.frame(
+   value = as.numeric(veri),
+   group = as.factor(grup)
+ )
> # Kruskal-Wallis testi
> result <- kruskal.test(value ~ group, data = df)
> # Test sonuçlarını görüntüleme
> print(result)

Kruskal-Wallis rank sum test

data: value by group
Kruskal-Wallis chi-squared = 13.434, df = 5, p-value = 0.01963
```

Yorum: p-değeri 0,05'ten küçük olduğu için H_0 reddedilir. %95 güven düzeyinde diyet sonrası kilo artışları arasında fark olduğu söylenebilir. Bu durumda Post-Hoc Testi yapılmalıdır.

**Gruplar arasındaki farklılığın hangi gruplar arasında olduğunu bulmak için, Mann-Whitney U Testini uygulayabiliriz.*

Python:

```
In [14]: diyet1=[12,10,13,11,12,9]
diyet2=[18,19,18,18,19,19]
diyet3=[10,12,13,16,14,13]
diyet4=[19,20,18,19,18,19]

stats.mannwhitneyu(diyet1, diyet2, alternative = 'greater')

Out[14]: MannwhitneyUResult(statistic=0.0, pvalue=0.9986942402179811)

In [15]: stats.mannwhitneyu(diyet2, diyet3, alternative = 'greater')

Out[15]: MannwhitneyUResult(statistic=36.0, pvalue=0.0022038789978488407)

In [16]: stats.mannwhitneyu(diyet3, diyet4, alternative = 'greater')

Out[16]: MannwhitneyUResult(statistic=0.0, pvalue=0.9986230098826184)

In [17]: stats.mannwhitneyu(diyet1, diyet4, alternative = 'greater')

Out[17]: MannwhitneyUResult(statistic=0.0, pvalue=0.9986230098826184)
```

RStudio:

```
> library(tidyverse)
> # Verileri tanımlayın
> diyet1 <- c(12, 10, 13, 11, 12, 9)
> diyet2 <- c(18, 19, 18, 18, 19, 19)
> diyet3 <- c(10, 12, 13, 16, 14, 13)
> diyet4 <- c(19, 20, 18, 19, 18, 19)
> diyet1 %>% wilcox.test(diyet3, p.adjust.method = "BH")
Wilcoxon rank sum test with continuity correction

data: . and diyet3
W = 7.5, p-value = 0.1037
alternative hypothesis: true location shift is not equal to 0
> diyet2 %>% wilcox.test(diyet3, p.adjust.method = "BH")
Wilcoxon rank sum test with continuity correction

data: . and diyet3
W = 36, p-value = 0.004408
alternative hypothesis: true location shift is not equal to 0
> diyet3 %>% wilcox.test(diyet4, p.adjust.method = "BH")
Wilcoxon rank sum test with continuity correction

data: . and diyet4
W = 13.5, p-value = 0.476
alternative hypothesis: true location shift is not equal to 0

> # Post hoc test olarak Mann-Whitney U testini uygulayın
> diyet1 %>% wilcox.test(diyet2, p.adjust.method = "BH")
Wilcoxon rank sum test with continuity correction

data: . and diyet2
W = 0, p-value = 0.004408
alternative hypothesis: true location shift is not equal to 0
> diyet1 %>% wilcox.test(diyet4, p.adjust.method = "BH")
Wilcoxon rank sum test with continuity correction

data: . and diyet4
W = 0, p-value = 0.004624
alternative hypothesis: true location shift is not equal to 0
> diyet2 %>% wilcox.test(diyet4, p.adjust.method = "BH")
Wilcoxon rank sum test with continuity correction

data: . and diyet4
W = 13.5, p-value = 0.476
alternative hypothesis: true location shift is not equal to 0
```

Yorum: p-değeri (0.0022<0.05) olduğu için H_0 reddedilir ve diyet2 ve diyet3 gruplarının arasında anlamlı bir fark olduğunu %95 güven düzeyinde söyleyebiliriz. Diğer grup karşılaştırmalarında p-değeri 0.05'ten büyük olduğu için H_0 kabul edilir ve aralarında anlamlı bir fark olmadığını %95 güven düzeyinde söyleyebiliriz. Yani farklılığı oluşturan grubun diyet2 olduğunu post-hoc testinden görebiliriz.

2. 3 Bağımlı Gruptan Oluşan Veri Kümesi:

ÖRNEK 2:

Üniversite öğrencileri arasında yapılan bir araştırmada, öğrencilerin Vize1, Vize2 ve Final notları arasındaki ilişkiyi anlamak amaçlanmaktadır. Veri seti, 10 öğrencinin bu üç sınavdaki notlarını içermektedir. Sınav notları arasında anlamlı fark olup olmadığını araştırınız.

NO	Vize1	Vize2	Final
1	56	54	90
2	60	62	74
3	50	55	50
4	72	55	89
5	72	70	76
6	50	71	89
7	60	49	50
8	76	61	80
9	60	77	70
10	40	60	87

H_0 : Sınav notları arasında fark yoktur.

H_5 : Sınav notları arasında fark vardır.

Özetleyici istatistikler:

```
In [2]: import numpy as np
import pandas as pd
from scipy import stats
df = pd.DataFrame({'vize1': [56,60,50,72,72,50,60,76,60,40],
                    'vize2': [54,62,55,55,70,71,49,61,77,60],
                    'final': [90,74,50,89,76,89,50,80,70,87]})
print(df["vize1"].describe())
print(df["vize2"].describe())
print(df["final"].describe())
```

```
count    10.000000
mean      59.600000
std       11.345092
min       40.000000
25%       51.500000
50%       60.000000
75%       69.000000
max       76.000000
Name: vize1, dtype: float64
count    10.000000
mean      61.400000
std        8.834277
min       49.000000
25%       55.000000
50%       60.500000
75%       68.000000
max       77.000000
Name: vize2, dtype: float64
count    10.000000
mean      75.500000
std       15.13091
min       50.000000
25%       71.000000
50%       78.000000
75%       88.500000
max       90.000000
Name: final, dtype: float64
```

Normallik Testi:

```
In [8]: from scipy.stats import shapiro

# Veriler
vize1 = [56, 60, 50, 72, 72, 50, 60, 76, 60, 40]
vize2 = [54, 62, 55, 55, 70, 71, 49, 61, 77, 60]
final = [90, 74, 50, 89, 76, 89, 50, 80, 70, 87]

# Shapiro-Wilk testi
stat_vize1, p_value_vize1 = shapiro(vize1)
stat_vize2, p_value_vize2 = shapiro(vize2)
stat_final, p_value_final = shapiro(final)

print(f"Vize1 - Shapiro-Wilk Testi İstatistiği: {stat_vize1}, p-değeri: {p_value_vize1}")
print(f"Vize2 - Shapiro-Wilk Testi İstatistiği: {stat_vize2}, p-değeri: {p_value_vize2}")
print(f"Final - Shapiro-Wilk Testi İstatistiği: {stat_final}, p-değeri: {p_value_final}")

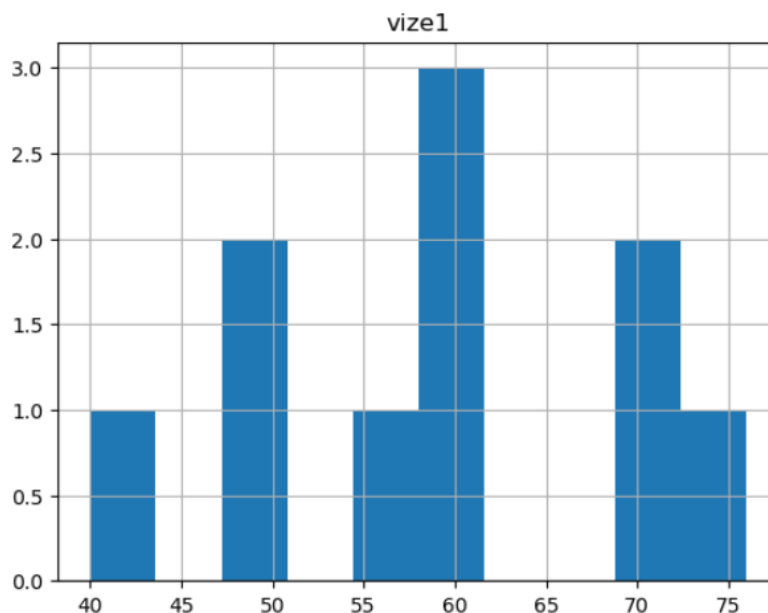
Vize1 - Shapiro-Wilk Testi İstatistiği: 0.9456766247749329, p-değeri: 0.6177170276641846
Vize2 - Shapiro-Wilk Testi İstatistiği: 0.945562481880188, p-değeri: 0.6163893342018127
Final - Shapiro-Wilk Testi İstatistiği: 0.8401842713356018, p-değeri: 0.04434916377067566
```

- ➔ Final'in p değeri ($0,0443 < 0,05$) olduğu için normallik varsayımının sağlanmadığı %5 anlamlılık düzeyinde söylenebilir.
- ➔ Vize1 ve vize2'nin p değeri 0,05'ten büyük olduğu için normallik varsayımının sağlandığı %5 anlamlılık düzeyinde söylenebilir.

➔ Histogram Görselleri:

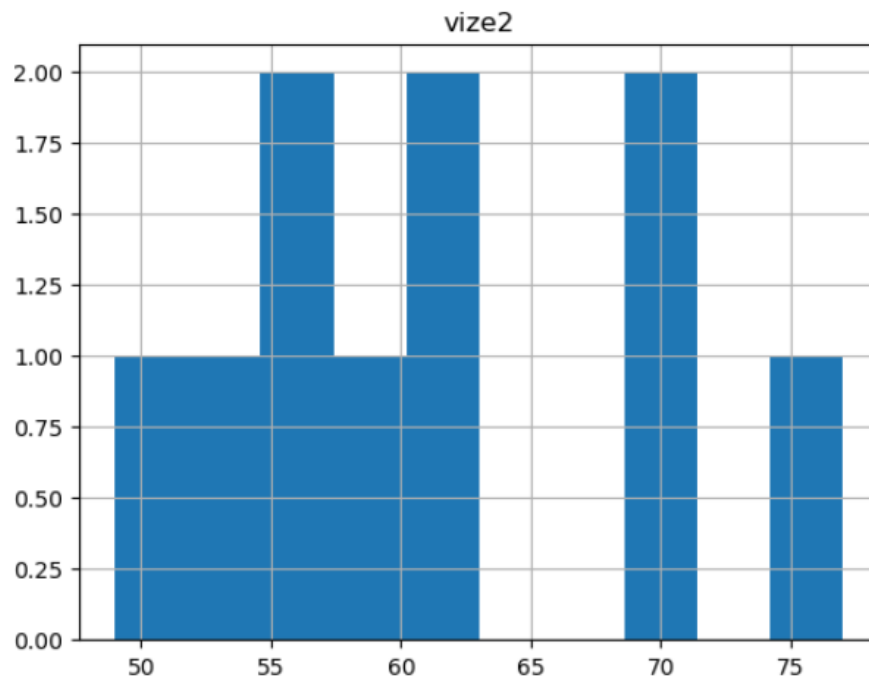
```
In [9]: df.hist('vize1')

Out[9]: array([[<Axes: title='{center': 'vize1'}>]], dtype=object)
```



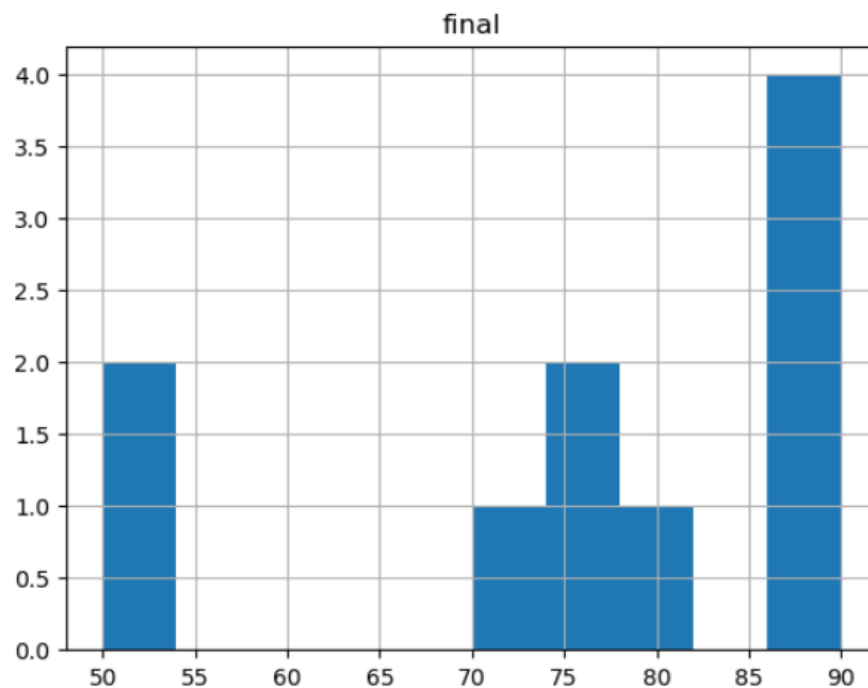
```
In [10]: df.hist('vize2')
```

```
Out[10]: array([[<Axes: title={'center': 'vize2'}>]], dtype=object)
```



```
In [11]: df.hist('final')
```

```
Out[11]: array([[<Axes: title={'center': 'final'}>]], dtype=object)
```

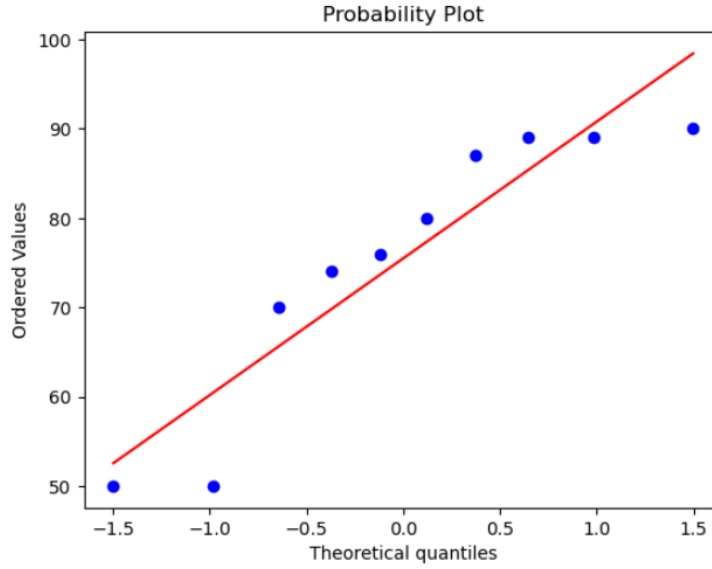


→ Q-Q Plots Görseli:

```
In [13]: import scipy.stats as stats
import matplotlib.pyplot as plt
```

```
In [14]: df = df[['final']]
stats.probplot(df['final'], dist="norm", plot=plt)
```

```
Out[14]: ((array([-1.49876728, -0.98494667, -0.64703803, -0.37109819, -0.12121409,
0.12121409, 0.37109819, 0.64703803, 0.98494667, 1.49876728])),
array([50, 50, 70, 74, 76, 80, 87, 89, 89, 90], dtype=int64)),
(15.30915209481996, 75.5, 0.9282297250967176))
```



→ Boxplot Görseli:

```
In [15]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

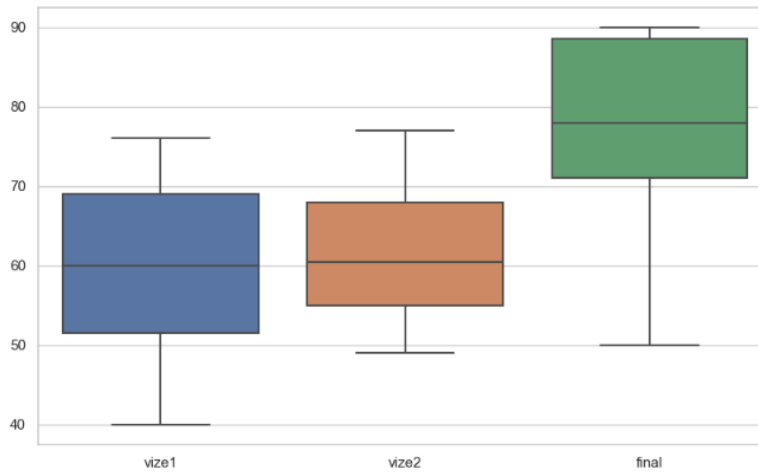
veriler = {'vize1': [56,60,50,72,72,50,60,76,60,40],
           'vize2': [54,62,55,55,70,71,49,61,77,60],
           'final': [90,74,50,89,76,89,50,80,70,87]}

df = pd.DataFrame(veriler)

# Boxplot çizimi
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))

sns.boxplot(data=df)

# Grafiği gösterme
plt.show()
```



5. Bağımlı İki Örneklem Testi

→ Wilcoxon Testi:

H_0 : 1.vize ile 2.vize notlarının arasında fark yoktur.

H_5 : 1.vize ile 2.vize notlarının arasında fark vardır.

Python:

```
In [20]: from scipy.stats import wilcoxon

In [21]: # Wilcoxon testi
stat, p_value = wilcoxon(vize1, vize2)

print(f"Wilcoxon İşaretli Sıralar Testi İstatistiği: {stat}")
print(f"P-değeri: {p_value}")

Wilcoxon İşaretli Sıralar Testi İstatistiği: 22.5
P-değeri: 0.625
```

Yorum: p-değeri 0,05'ten büyük olduğu için H_0 reddedilemez. %95 güven düzeyinde vize notları arasında fark olduğu söylenemez.

H_0 : 1.vize ile final notlarının arasında fark yoktur.

H_5 : 1.vize ile final notlarının arasında fark vardır.

Python:

```
In [21]: stat_vize1_final, p_value_vize1_final = wilcoxon(vize1, final)
print(f"Wilcoxon İşaretli Sıralar Testi (Vize1 ve Final) İstatistiği: {stat_vize1_final}")
print(f"P-değeri: {p_value_vize1_final}")

Wilcoxon İşaretli Sıralar Testi (Vize1 ve Final) İstatistiği: 3.5
P-değeri: 0.02414053862843877
```

Yorum: p-değeri 0,05'ten küçük olduğu için H_0 reddedilebilir. %95 güven düzeyinde 1.vize notları ile final notları arasında fark olduğu söylenebilir.

H_0 : 2.vize ile final notlarının arasında fark yoktur.

H_5 : 2.vize notları final notlarından daha düşüktür.

Python:

```
In [20]: stat_vize2_final, p_value_vize2_final = wilcoxon(vize2, final)
print(f"Wilcoxon İşaretli Sıralar Testi (Vize2 ve Final) İstatistiği: {stat_vize2_final}")
print(f"P-değeri: {p_value_vize2_final}")

Wilcoxon İşaretli Sıralar Testi (Vize2 ve Final) İstatistiği: 6.0
P-değeri: 0.02734375
```

Yorum: p-değeri 0,05'ten küçük olduğu için H_0 reddedilebilir. %95 güven düzeyinde 2.vize notları ile final notları arasında fark olduğu söylenebilir.

RStudio:

```
> # Wilcoxon işaret testini uygulayın (yaklaşık p-değeri ile) > print(vize1_final)
> vize1_vize2 <- wilcox.test(vize1, vize2, exact = FALSE) wilcoxon rank sum test with continuity correction
> vize1_final <- wilcox.test(vize1, final, exact = FALSE) data: vize1 and final
> vize2_final <- wilcox.test(vize2, final, exact = FALSE) w = 20.5, p-value = 0.02736
> # Sonuçları yazdırın alternative hypothesis: true location shift is not equal to 0
> print(vize1_vize2) > print(vize2_final)
wilcoxon rank sum test with continuity correction data: vize2 and final
data: vize1 and vize2 w = 22.5, p-value = 0.04095
w = 46.5, p-value = 0.8197 alternative hypothesis: true location shift is not equal to 0 alternative hypothesis: true location shift is not equal to 0
```

7. k Örneklem Konum Testi

→ Friedman Testi

H_0 : Sınav notları arasında farklılık yoktur.

H_1 : Sınav notları arasında farklılık vardır.

Python:

```
In [18]: from scipy import stats
vize1 = [56, 60, 50, 72, 72, 50, 60, 76, 60, 40]
vize2 = [54, 62, 55, 55, 70, 71, 49, 61, 77, 60]
final = [90, 74, 50, 89, 76, 89, 50, 80, 70, 87]
stats.friedmanchisquare(vize1, vize2, final)
```

```
Out[18]: FriedmanchisquareResult(statistic=6.512820512820522, pvalue=0.03852645016090633)
```

Yorum: p-değeri 0,05'ten küçük olduğu için H_0 reddedilebilir. %95 güven düzeyinde vize notları arasında fark olduğu söylenebilir.

*Bu durumda Post-Hoc testlerinden birini uygulamalıyız.

Python:

```
In [35]: from itertools import combinations
from scipy.stats import ranksums

# Veriler
vize1 = [56, 60, 50, 72, 72, 50, 60, 76, 60, 40]
vize2 = [54, 62, 55, 55, 70, 71, 49, 61, 77, 60]
final = [90, 74, 50, 89, 76, 89, 50, 80, 70, 87]

data = [vize1, vize2, final]

# Tüm kombinasyonları al
combinations_list = list(combinations(range(len(data)), 2))

# Herhangi post-hoc testi
for combo in combinations_list:
    group1, group2 = combo
    stat, p_value = ranksums(data[group1], data[group2])
    print(f"Grup {group1 + 1} ve Grup {group2 + 1} arasındaki p-değeri: {p_value}")

Grup 1 ve Grup 2 arasındaki p-değeri: 0.7913367801006604
Grup 1 ve Grup 3 arasındaki p-değeri: 0.025748080821108063
Grup 2 ve Grup 3 arasındaki p-değeri: 0.03763531378731424
```

Yorum: Grup1 ve grup2 arasında yapılan karşılaştırmada p-değeri 0,05'ten büyük olduğu için H_0 kabul edilebilir, yani grup1 ve grup2 arasında fark olmadığını %95 güven düzeyinde söyleyebiliriz. Diğer grup karşılaştırmalarında p-değeri 0,05'ten küçük olduğu için H_0 reddedilir, yani gruplar arasında fark olduğunu %95 güven düzeyinde söyleyebiliriz.

8. Eğilim Testi

ÖRNEK 3:

2007-2022 yılları arasında Genç nüfusun toplam nüfus içindeki oranları aşağıda verilmiştir.

17.6, 17.4, 17.2, 17.0, 16.8, 16.6, 16.6, 16.5, 16.4, 16.3, 16.1, 15.8, 15.6, 15.4, 15.3, 15.2

Python:

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

# Veri seti
genclik_oranlari = [17.6, 17.4, 17.2, 17.0, 16.8, 16.6, 16.6, 16.5, 16.4, 16.3, 16.1, 15.8, 15.6, 15.4, 15.3, 15.2]

# Yıllar (sıralı bir değişken olarak düşünebiliriz)
yillar = list(range(2007, 2023))

# Kendall Tau Testi
statistic, p_value = kendalltau(yillar, genclik_oranlari)
print(f"Kendall Tau İstatistiği: {statistic}")
print(f"P-değer: {p_value}")

Kendall Tau İstatistiği: -0.9958246164193105
P-değer: 8.179759546382855e-08
```

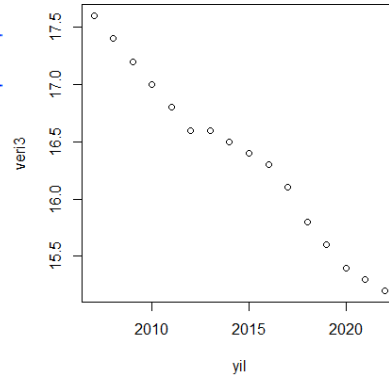
Yorum: Kendall Tau istatistiği -0.99 olduğu için genç sağlıklı nüfus oranı negatif yönde azalmıştır. p-değeri 0,05'ten küçük olduğu için H0 reddedilir. Yani %95 güven düzeyinde trendin olduğunu söyleyebiliriz.

RStudio:

```
> veri3 <- c(17.6, 17.4, 17.2, 17.0, 16.8, 16.6, 16.6, 16.5, 16.4, 16.3, 16.1, 15.8, 15.6, 15.4, 15.3, 15.2)
> yıl <- c(2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022)
> plot(veri3 ~ yıl)
> model <- lm(veri3 ~ yıl)
> print(model)
```

```
Call:
lm(formula = veri3 ~ yıl)
```

```
Coefficients:
(Intercept)      yıl
  330.9800      -0.1562
```



*Verilen modelde genç nüfusun toplum içindeki oranlarının yıllara göre dağılımının grafiği verilmiştir. Grafiğe baktığımızda oranın yıllara göre azaldığını söyleyebiliriz.

Python Kodları:

1.soru:

```
import pandas as pd

veri = pd.read_excel("Kitap1.xlsx") #excel den aktarılan veri okunur

print(veri)

df = pd.DataFrame({'diyet1':[12,10,13,11,12,9],
                   'diyet2':[18,19,18,18,19,19],
                   'diyet3':[10,12,13,16,14,13],
                   'diyet4':[19,20,18,19,18,19]})

print(df["diyet1"].describe())
print(df["diyet2"].describe())
print(df["diyet3"].describe())
print(df["diyet4"].describe())

from scipy.stats import shapiro
shapiro(df["diyet1"])
from scipy.stats import shapiro
shapiro(df["diyet2"])
from scipy.stats import shapiro
shapiro(df["diyet3"])
from scipy.stats import shapiro
shapiro(df["diyet4"])

df.hist('diyet1')
df.hist('diyet2')
df.hist('diyet3')
df.hist('diyet4')

# Q-Q Plots Görseli
import scipy.stats as stats
import matplotlib.pyplot as plt
df = df[['diyet2']]
```

```
stats.probplot(df['diyet2'],dist="norm", plot=plt)
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import pandas as pd
```

```
veriler = {'diyet1': [12,10,13,11,12,9],  
          'diyet2': [18,19,18,18,19,19],  
          'diyet3': [10,12,13,16,14,13],  
          'diyet4': [19,20,18,19,18,19]}
```

```
df = pd.DataFrame(veriler)
```

```
# Boxplot çizimi
```

```
sns.set(style="whitegrid")
```

```
plt.figure(figsize=(10, 6))
```

```
sns.boxplot(data=df)
```

```
# Grafiği gösterme
```

```
plt.show()
```

```
diyet4 = np.array([19, 20, 18, 19, 18, 19])
```

```
median = 15
```

```
n = len(diyet4)
```

```
nPlus = len(diyet4[diyet4>median])
```

```
nNeg = len(diyet4[diyet4<median])
```

```
n = nPlus + nNeg
```

```
stats.binomtest(nPlus, n, p=0.5, alternative="greater")
```

```
import numpy as np
```

```
from scipy import stats
```

```
diyet1=np.array([12,10,13,11,12,9])
```

```
median = 17
```

```
stats.wilcoxon(diyet1-median,
```

```
                alternative='less',
```

```
                method='approx',
```

```
                zero_method='zsplit',
```

```

        correction=True)

from scipy.stats import mannwhitneyu

stat, p_value = mannwhitneyu(df['diyet1'], df['diyet2'])

print(f"Mann-Whitney U Testi İstatistiği: {stat}")

print(f"P-değeri: {p_value}")

import numpy as np

from scipy import stats

diyet1 = np.array([12, 10, 13, 11, 12, 9])

diyet4 = np.array([19, 20, 18, 19, 18, 19])

stats.mannwhitneyu(diyet2, diyet4, alternative = 'less')


from scipy.stats import kruskal

stat, p_value = kruskal(df['diyet1'], df['diyet2'], df['diyet3'], df['diyet4'])

print(f"Kruskal-Wallis Testi İstatistiği: {stat}")

print(f"P-değeri: {p_value}")


import numpy as np

from scipy import stats

diyet1=[12, 10, 13, 11, 12, 9]

diyet2=[18, 19, 18, 18, 19, 19]

diyet3=[10, 12, 13, 16, 14, 13]

diyet4=[19, 20, 18, 19, 18, 19]

stats.kruskal(diyet1,diyet2,diyet3,diyet4)


import numpy as np

from scipy.stats import mannwhitneyu

diyet1 = np.array([12, 10, 13, 11, 12, 9])

diyet2 = np.array([15, 16, 17, 18, 19, 14])

u, p = mannwhitneyu(diyet1, diyet2)

print("Mann-Whitney U testi sonuçları:")

print("u:", u)

print("p-değeri:", p)

```

2.soru:

```
import numpy as np

import pandas as pd

from scipy import stats

df = pd.DataFrame({'vize1': [56,60,50,72,72,50,60,76,60,40],
                    'vize2': [54,62,55,55,70,71,49,61,77,60],
                    'final':[90,74,50,89,76,89,50,80,70,87]})

print(df["vize1"].describe())

print(df["vize2"].describe())

print(df["final"].describe())


from scipy.stats import shapiro

# Veriler

vize1 = [56, 60, 50, 72, 72, 50, 60, 76, 60, 40]

vize2 = [54, 62, 55, 55, 70, 71, 49, 61, 77, 60]

final = [90, 74, 50, 89, 76, 89, 50, 80, 70, 87]

# Shapiro-Wilk testi

stat_vize1, p_value_vize1 = shapiro(vize1)

stat_vize2, p_value_vize2 = shapiro(vize2)

stat_final, p_value_final = shapiro(final)

print(f"Vize1 - Shapiro-Wilk Testi İstatistiği: {stat_vize1}, p-değeri: {p_value_vize1}")

print(f"Vize2 - Shapiro-Wilk Testi İstatistiği: {stat_vize2}, p-değeri: {p_value_vize2}")

print(f"Final - Shapiro-Wilk Testi İstatistiği: {stat_final}, p-değeri: {p_value_final}")


df.hist('vize1')

df.hist('vize2')

df.hist('final')


import scipy.stats as stats
```

```
import matplotlib.pyplot as plt

df = df[['final']]

stats.probplot(df['final'], dist="norm", plot=plt)
```

```
import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

veriler = {'vize1': [56,60,50,72,72,50,60,76,60,40],
           'vize2': [54,62,55,55,70,71,49,61,77,60],
           'final':[90,74,50,89,76,89,50,80,70,87]}

df = pd.DataFrame(veriler)

# Boxplot çizimi

sns.set(style="whitegrid")

plt.figure(figsize=(10, 6))

sns.boxplot(data=df)

# Grafiği gösterme

plt.show()
```

```
from scipy.stats import wilcoxon

# Wilcoxon testi

stat, p_value = wilcoxon(vize1, vize2)

print(f"Wilcoxon İşaretli Sıralar Testi İstatistiği: {stat}")

print(f"P-değeri: {p_value}")
```

```
stat_vize2_final, p_value_vize2_final = wilcoxon(vize2, final)

print(f"Wilcoxon İşaretli Sıralar Testi (Vize2 ve Final) İstatistiği: {stat_vize2_final}")

print(f"P-değeri: {p_value_vize2_final}")
```

```
stat_vize1_final, p_value_vize1_final = wilcoxon(vize1, final)

print(f"Wilcoxon İşaretli Sıralar Testi (Vize1 ve Final) İstatistiği: {stat_vize1_final}")

print(f"P-değeri: {p_value_vize1_final}")
```

```
# Friedman testi
```



```

from scipy import stats

vize1 = [56, 60, 50, 72, 72, 50, 60, 76, 60, 40]
vize2 = [54, 62, 55, 55, 70, 71, 49, 61, 77, 60]
final = [90, 74, 50, 89, 76, 89, 50, 80, 70, 87]
stats.friedmanchisquare(vize1, vize2, final)

```

```

from itertools import combinations

from scipy.stats import ranksums

# Veriler
vize1 = [56, 60, 50, 72, 72, 50, 60, 76, 60, 40]
vize2 = [54, 62, 55, 55, 70, 71, 49, 61, 77, 60]
final = [90, 74, 50, 89, 76, 89, 50, 80, 70, 87]

data = [vize1, vize2, final]

# Tüm kombinasyonları al
combinations_list = list(combinations(range(len(data)), 2))

# Nemenyi post-hoc testi
for combo in combinations_list:
    group1, group2 = combo

    stat, p_value = ranksums(data[group1], data[group2])

    print(f"Grup {group1 + 1} ve Grup {group2 + 1} arasındaki p-değeri: {p_value}")

```

3.soru:

```

from scipy.stats import kendalltau

# Veri seti
genclik_oranlari = [17.6, 17.4, 17.2, 17.0, 16.8, 16.6, 16.6, 16.5, 16.4, 16.3, 16.1, 15.8, 15.6, 15.4, 15.3, 15.2]

# Yıllar (sıralı bir değişken olarak düşünebiliriz)
yillar = list(range(2007, 2023))

# Kendall Tau Testi
statistic, p_value = kendalltau(yillar, genclik_oranlari)

```

```
print(f"Kendall Tau İstatistiği: {statistic}")  
print(f"P-değer: {p_value}")
```

RSTUDIO KODLARI

Soru1

```
library(readxl)

soru1 <- read_excel("C:/Users/Lenovo/Desktop/soru1.xlsx")

View(soru1)
```

##İŞARET TESTİ

```
install.packages("DescTools")

library(DescTools)

diyet4=c(19,20,18,19,18,19)

SignTest(x=diyet4,mu=15,alternative="greater")
```

##WILCOXON TESTİ

```
install.packages("stats")

library(stats)

diyet1=c(12,10,13,11,12,9)

wilcox.test(x,mu=17,alternative="less")
```

##MANN-WHITNEY U TESTİ

```
diyet1 <- c(12,10,13,11,12,9)

diyet2 <- c(18,19,18,18,19,19)

wilcox.test(diyet1,diyet2)
```

##MANN-WHITNEY U TESTİ

```
diyet1 <- c(12,10,13,11,12,9)

diyet4 <- c(19,20,18,19,18,19)

wilcox.test(diyet1,diyet4)
```

##KRUSKAL-WALLİS

```
veri <- read_excel("C:/Users/Lenovo/Desktop/soru1.xlsx")

grup <- rep(1:6, each = 8)

veri <- unlist(veri)

grup <- unlist(grup)
```

```
df <- data.frame(
  value = as.numeric(veri),
  group = as.factor(grup)
)
result <- kruskal.test(value ~ group, data = df)
print(result)
```

Soru2

```
library(readxl)
soru2 <- read_excel("C:/Users/Lenovo/Desktop/soru2.xlsx")
View(soru2)
```

##WILCOXON TESTİ

```
vize1 <- c(56, 60, 50, 72, 72, 50, 60, 76, 60, 40)
vize2 <- c(54, 62, 55, 55, 70, 71, 49, 61, 77, 60)
final <- c(90, 74, 50, 89, 76, 89, 50, 80, 70, 87)
# Wilcoxon işaret testini uygulayın (yaklaşık p-değeri ile)
vize1_vize2 <- wilcox.test(vize1, vize2, exact = FALSE)
vize1_final <- wilcox.test(vize1, final, exact = FALSE)
vize2_final <- wilcox.test(vize2, final, exact = FALSE)
print(vize1_vize2)
print(vize1_final)
print(vize2_final)
```

Soru3

##EĞİLİM TESTİ

```
veri3 <- c(17.6, 17.4, 17.2, 17.0, 16.8, 16.6, 16.6, 16.5, 16.4, 16.3, 16.1, 15.8, 15.6, 15.4, 15.3, 15.2)
yil <- c(2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022)
plot(veri3 ~ yil)
model <- lm(veri3 ~ yil)
print(model)
```

Kaynakça

<https://sedatsen.files.wordpress.com/2016/11/3-sunum.pdf>

https://online.stat.psu.edu/stat502_fa21/lesson/3/3.8

<https://data.tuik.gov.tr/Bulten/Index?p=Istatistiklerle-Genclik-2022-49670> (tablo-1)

adreslerinden 12 Aralık 2023 ve 17 Aralık 2023 tarihlerinde erişildi.

<https://stackoverflow.com/>