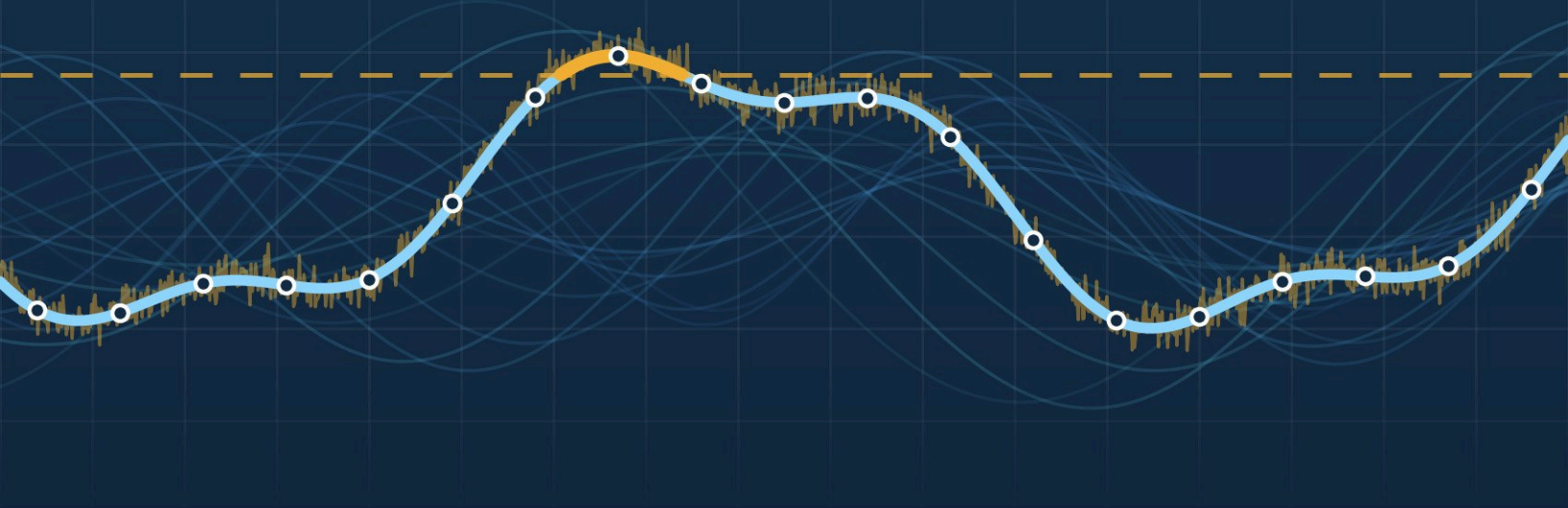


Python Uygulamalarıyla Sensör Teknolojileri ve Veri Toplama

Sensör simülasyonu, veri analizi ve görselleştirme

NumPy • Pandas • Matplotlib • Jupyter



Dr. Selim Ünal

Kastamonu Üniversitesi

Metalurji ve Malzeme Mühendisliği Bölümü

2026–2027 Güz

Python Uygulamalarıyla Sensör Teknolojileri ve Veri Toplama

Yazar	Dr. Selim Ünal
Kurum	Kastamonu Üniversitesi Metalurji ve Malzeme Mühendisliği Bölümü
ISBN	978-625-00-5106-1
Basım	1. Basım, 2026

© 2026 Dr. Selim Ünal. Tüm hakları saklıdır.

Bu kitabın yayın hakları yazarına aittir. Kaynak gösterilerek yapılan kısa alıntılar dışında, kitabın tamamı veya herhangi bir bölümü yazarın yazılı izni olmadan elektronik, mekanik, fotokopi ya da başka bir yöntemle çoğaltılamaz, yayımlanamaz ve dağıtılamaz.

LABORATUVAR ORTAMININ HAZIRLANMASI

Python + VS Code + Jupyter

Bu föydeki uygulamaları bilgisayarda çalıştırmak için önce Python ve VS Code kurulacaktır.

Windows 10/11 için önerilen sıra:

1. Python kurulumu → 2. VS Code kurulumu → 3. Python ve Jupyter eklentileri → 4. Çalışma klasörü → 5. İlk Jupyter Notebook

Kurulum sırasında öğretmenin verdiği bilgisayar hesabını kullan. Yönetici yetkisi yoksa okulun BT sorumlusundan destek al.

Gerekli programlar

Python: <https://www.python.org/downloads/>

Visual Studio Code: <https://code.visualstudio.com/>

İnternet bağlantısı varsa kurulumdan sonra kütüphaneler terminalden kurulacaktır.

1. Python Kurulumu

Adım 1 — Python indirme

Tarayıcıdan python.org/downloads adresini aç. Windows için güncel Python sürümünü indir.

Adım 2 — Kurulum ekranı

Kurulum penceresinin ilk ekranında Python komutunun terminalden çalışabilmesi için PATH seçeneğini işaretlendiğinden emin ol. Sonra kurulumu başlat.

Adım 3 — Kurulumu test et

```
python --version  
py --version
```

Bunlardan biri Python sürümünü göstermelidir.

Adım 4 — Python yorumlayıcısını test et

```
python  
print("Merhaba Sensör Laboratuvarı")  
2 + 3  
exit()
```

GÖREV: Terminalde kendi adını ve bir sıcaklık değeri yazdır.

```
ad = "Selim"  
sicaklik = 24.5  
print(ad)  
print(sicaklik, "°C")
```

2. Visual Studio Code Kurulumu

Adım 1 — VS Code indir

<https://code.visualstudio.com/> adresini aç ve Windows sürümünü indir.

Adım 2 — Kur

Kurulumda masaüstü kısayolu isteğe bağlıdır. VS Code kurulduktan sonra programı aç.

Adım 3 — Sensör Laboratuvarı klasörü oluştur

```
C:\Sensor_Laboratuvari
```

VS Code → File → Open Folder → Sensor_Laboratuvari klasörünü seç.

Adım 4 — İlk Python dosyası

Explorer → New File → sensor_01.py

```
print("Sensör laboratuvarı başladı")
sicaklik = 25.0
nem = 45.0
print("Sıcaklık:", sicaklik, "°C")
print("Nem:", nem, "%")
```

Çalıştırma: Dosyanın sağ üstündeki ► Run Python File düğmesine bas.

GÖREV: Değerleri değiştir, dosyayı kaydet ve tekrar çalıştır.

3. VS Code: Python ve Jupyter Eklentileri

Sol taraftaki Extensions simgesine tıkla ve aşağıdaki eklentileri kur:

- Python — Microsoft
- Jupyter — Microsoft

Kurulumdan sonra VS Code'u kapatıp açmak gerekebilir.

Python yorumlayıcısını seç

Ctrl + Shift + P → Python: Select Interpreter → kurduğun Python sürümünü seç.

Terminal aç

Terminal → New Terminal

```
python --version  
pip --version
```

Gerekli kütüphaneleri kur

```
python -m pip install --upgrade pip  
python -m pip install numpy pandas matplotlib  
jupyter
```

Kurulumu kontrol et

```
python -c "import numpy, pandas, matplotlib;  
print('Kütüphaneler hazır')"
```

GÖREV: Terminalde yukarıdaki test kodunu çalıştır ve ekrana gelen sonucu kontrol et.

4. Proje İçin Ortam Oluşturma

paketleri kur:

```
python -m pip install --upgrade pip  
python -m pip install numpy pandas matplotlib  
jupyter ipykernel
```

5. VS Code İçinde Jupyter Notebook Kullanımı

Yeni notebook oluştur

Explorer → New File → sensor_deneme.ipynb

Notebook açıldığında sağ üstten Kernel / Select Kernel seçeneğine tıkla ve Python ortamını seç.

İlk hücre

```
print("Jupyter çalışıyor")
```

Hücreyi çalıştır: Shift + Enter

NumPy

```
import numpy as np

x = np.array([20, 22, 24, 25, 27])
print(x)
print("Ortalama:", x.mean())
```

Matplotlib

```
import matplotlib.pyplot as plt

t = np.arange(5)
plt.plot(t, x, marker="o")
plt.xlabel("Ölçüm")
plt.ylabel("Sıcaklık (°C)")
plt.grid()
plt.show()
```

GÖREV: Beş sıcaklık değerini değiştir ve grafiği tekrar çalıştır.

6. İlk Sensör Veri Analizi: Jupyter

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

rng = np.random.default_rng(42)
t = np.arange(0, 60)
temp = 24 + 2*np.sin(t/8) + rng.normal(0, 0.35,
len(t))
nem = 50 - 5*np.sin(t/10) + rng.normal(0, 1.2,
len(t))

df = pd.DataFrame({"zaman": t, "sicaklik": temp,
"nem": nem})
display(df.head())
print(df.describe())

plt.figure(figsize=(9,4))
plt.plot(df["zaman"], df["sicaklik"],
label="Sıcaklık")
plt.xlabel("Zaman (s)")
plt.ylabel("Sıcaklık (°C)")
plt.legend()
plt.grid()
plt.show()

plt.figure(figsize=(9,4))
plt.plot(df["zaman"], df["nem"], label="Nem")
plt.xlabel("Zaman (s)")
plt.ylabel("Nem (%)")
plt.legend()
plt.grid()
plt.show()
```

GÖREV: Sıcaklık için ortalama, minimum ve maksimum değerleri yazdır.

```
print("Ortalama:", df["sicaklik"].mean())
print("Minimum:", df["sicaklik"].min())
print("Maksimum:", df["sicaklik"].max())
```

7. Her Sensör Uygulamasında İzlenecek Yol

```
# 1. Kütüphaneler
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# 2. Veri üret / oku
t = np.arange(100)
sensor = 25 + np.random.normal(0, 0.5, 100)

# 3. DataFrame oluştur
df = pd.DataFrame({"zaman": t, "sensor": sensor})

# 4. Veriyi kontrol et
print(df.head())
print(df.describe())

# 5. Grafik çiz
plt.plot(df["zaman"], df["sensor"])
plt.xlabel("Zaman")
plt.ylabel("Sensör")
plt.grid()
plt.show()
```

Sık karşılaşılan sorunlar

ModuleNotFoundError → doğru Python ortamında pip install çalıştır.

Kernel görünmüyor → Select Kernel ile ortamını seç.

python komutu bulunamadı → Python kurulumu ve PATH ayarını kontrol et.

Grafik görünmüyor → hücreyi Shift + Enter ile çalıştır ve plt.show() kullan.

Dosya bulunamadı → VS Code ile açılan klasörün doğru klasör olduğunu kontrol et.

Laboratuvar kuralı

Kodu önce aynen çalıştır. Sonra bir satırı değiştir. Sonucu gözle. Ardından ikinci değişikliği yap. Her uygulamada kodu çalıştırmadan önce tahmin yap.

SENSÖR TEKNOLOJİLERİ VE VERİ TOPLAMA PYTHON UYGULAMA FÖYÜ

Kod • Simülasyon • Görselleştirme • Veri Analizi

Ad Soyad: _____
Sınıf: _____ Tarih: _____

Her uygulama: çalıştır → değiştir → gözle → tekrar çalıştır

1. Laboratuvar Ortamı ve İlk Sensör Verisi

Hazırlık

Python sürümü

```
import sys
print(sys.version)
```

Görev: Sürümü not et.

Kütüphaneler

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
print(np.__version__)
print(pd.__version__)
```

Görev: Kodu çalıştır.

İlk ölçümler

```
T=24.6
RH=48
mesafe=37.5
print(T,"°C",RH,"%",mesafe,"cm")
```

Görev: Değerleri değiştir.

İlk grafik

```
t=np.arange(10)
y=24+np.sin(t)
plt.plot(t,y,marker="o")
plt.xlabel("Zaman"); plt.ylabel("°C"); plt.grid();
plt.show()
```

Görev: Başlığı değiştir.

2. Python Temelleri: Sensör Değeri

1. Hafta

Değişken ve hesaplama

```
volt = 2.35
akim = 0.18
sure = 12

print("Güç =", volt * akim, "W")
print("Toplam yük =", akim * sure, "C")
```

Görev: Gerilim ve akımı değiştir.

Kullanıcıdan veri al

```
sicaklik = float(input("Sıcaklık (°C): "))
nem = float(input("Nem (%): "))
print(f"T={sicaklik:.1f} °C RH={nem:.1f} %")
```

Görev: 50 °C üstünde alarm yazdır.

Koşul ile sensör alarmı

```
x = float(input("Sensör değeri: "))
if x > 80:
    print("ALARM")
elif x > 60:
    print("UYARI")
else:
    print("NORMAL")
```

Görev: Eşikleri değiştir.

3. Veri Toplama: Listeler ve Döngüler

1-2. Hafta

Liste oluştur

```
olcum = [23.4, 23.8, 24.1, 24.0, 24.6, 24.9]
print("Ölçüm sayısı:", len(olcum))
print("İlk:", olcum[0])
print("Son:", olcum[-1])
```

Görev: Listeye 3 ölçüm ekle.

100 sensör okuması

```
import random
olcum = []
for i in range(100):
    olcum.append(random.uniform(20, 30))
print(olcum[:5])
print("Ortalama:", sum(olcum) / len(olcum))
```

Görev: Aralığı 10-40 °C yap.

Minimum, maksimum, ortalama

```
import numpy as np
x = np.array([22.1, 22.8, 23.0, 24.4, 25.1, 24.7])
print("Min:", x.min())
print("Maks:", x.max())
print("Ortalama:", x.mean())
print("Std:", x.std())
```

Görev: Standart sapmayı gözlemle.

4. Fiziksel Büyüklükler ve Birim Dönüşümü

2. Hafta

Sıcaklık dönüşümü

```
c = float(input("°C: "))
f = c * 9 / 5 + 32
k = c + 273.15
print("°F =", round(f,2))
print("K   =", round(k,2))
```

Görev: 25 °C ve 40 °C için dene.

Mesafe ve hız

```
mesafe = float(input("Mesafe (m): "))
sure = float(input("Süre (s): "))
hiz = mesafe / sure
print("Hız =", round(hiz,2), "m/s")
```

Görev: 20 m ve 4 s kullan.

Akış ve hacim

```
debi_Lmin = 2.5
sure_min = 8
hacim = debi_Lmin * sure_min
print("Hacim =", hacim, "L")
print("mL =", hacim * 1000)
```

Görev: Debiyi 0.5, 1, 2.5 ve 5 L/dk yap.

5. NumPy ile Sensör Dizileri

2. Hafta

Dizi işlemleri

```
import numpy as np
x = np.array([10, 12, 15, 14, 18, 20])
print(x + 2)
print(x * 0.5)
print(x.mean())
print(x.std())
```

Görev: Kendi 8 ölçümünü yaz.

Zaman dizisi

```
import numpy as np
t = np.arange(0, 10, 0.5)
print(t)
print("Adet:", len(t))
```

Görev: 0.1 saniyelik örnekleme yap.

Sensör sinyali üret

```
import numpy as np
rng = np.random.default_rng(3)
t = np.linspace(0, 10, 500)
```

```
sinyal = 25 + 2*np.sin(t) + rng.normal(0, 0.5,  
len(t))  
print(sinyal[:10])
```

Görev: Gürültüyü 0.1 ve 2.0 yapıp karşılaştır.

6. İlk Görselleştirme Uygulamaları

2. Hafta

Çizgi grafiği

```
import numpy as np
import matplotlib.pyplot as plt

t = np.arange(20)
temperature = 22 + 0.4*t + np.sin(t/2)
plt.plot(t, temperature, marker="o")
plt.xlabel("Zaman (dk)")
plt.ylabel("Sıcaklık (°C)")
plt.title("Sıcaklık Sensörü")
plt.grid()
plt.show()
```

Görev: Grafiğe çizgi kalınlığı ekle.

Çubuk grafik

```
import matplotlib.pyplot as plt
sensor = ["T", "RH", "LDR", "PIR"]
value = [24, 52, 680, 1]
plt.bar(sensor, value)
plt.title("Anlık Sensör Değerleri")
plt.ylabel("Değer")
plt.show()
```

Görev: Beşinci bir sensör ekle.

Histogram

```
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.default_rng(5)
x = rng.normal(25, 1.8, 500)
plt.hist(x, bins=25, edgecolor="black")
plt.xlabel("Sıcaklık (°C)")
plt.ylabel("Frekans")
plt.title("Sensör Ölçüm Dağılımı")
plt.show()
```

Görev: Ortalama ve std değerini grafiğe yaz.

7. Sensör Özellikleri: Aralık, Hassasiyet, Çözünürlük

3. Hafta

Sensör aralığı

```
min_deger = 0
max_deger = 100
olcumlər = [2, 20, 50, 75, 101, -3]
for x in olcumlər:
    uygun = min_deger <= x <= max_deger
    print(x, uygun)
```

Görev: Sınırları -20 ve 80 yap.**Çözünürlük simülasyonu**

```
deger = 24.678
for adim in [1, 0.1, 0.01]:
    yuvarlanmis = round(deger / adim) * adim
    print(adim, yuvarlanmis)
```

Görev: 0.001 ekle.**Duyarlılık / eğim**

```
import numpy as np
x = np.array([0, 1, 2, 3, 4])
y = np.array([0.0, 2.1, 4.1, 6.0, 8.2])
egim = np.polyfit(x, y, 1)[0]
print("Duyarlılık ≈", egim)
```

Görev: Yeni y değerleri dene.

8. Sensör Kalibrasyonu

3. Hafta

Doğrusal kalibrasyon

```
import numpy as np
volt = np.array([0.5, 1, 1.5, 2, 2.5])
referans = np.array([10, 20, 30, 40, 50])
k, b = np.polyfit(volt, referans, 1)
print("T =", k, "* V +", b)
```

Görev: Beş yeni kalibrasyon noktası seç.

Kalibrasyon sonrası grafik

```
import numpy as np
import matplotlib.pyplot as plt
volt = np.linspace(0.2, 4.8, 30)
gercek = 12*volt + 2
olculen = (gercek +
np.random.default_rng(2).normal(0, 1.2, len(volt)))
k,b=np.polyfit(volt,olculen,1)
tahmin=k*volt+b
plt.scatter(volt, olculen, label="Ölçüm")
plt.plot(volt, tahmin, label="Kalibrasyon")
plt.legend(); plt.grid(); plt.show()
```

Görev: Grafiğe gerçek doğruyu da ekle.

Kalibrasyon hatası

```
hata = olculen - gercek
print("MAE:", np.mean(np.abs(hata)))
print("RMSE:", np.sqrt(np.mean(hata**2)))
print("Ortalama hata:", np.mean(hata))
```

Görev: MAE ve RMSE sonuçlarını not et.

9. Elektronik Devre ve ADC Simülasyonu

4-5. Hafta

ADC dönüşümü

```
import numpy as np
Vref = 5.0
bit = 10
adc = np.array([0, 128, 256, 512, 768, 1023])
volt = adc * Vref / (2**bit - 1)
print(volt)
```

Görev: 12 bit çözünürlük kullan.

Gerilim bölücü

```
R1 = 10000
R2 = 4700
Vin = 5
```

```
Vout = Vin * R2/(R1+R2)
print("Vout =", Vout, "V")
```

Görev: R2 değerini değiştir.

LDR simülasyonu

```
import numpy as np
lux = np.linspace(1, 1000, 100)
R = 10000/(lux**0.7)
print("R(1 lux)=", R[0])
print("R(1000 lux)=", R[-1])
```

Görev: LDR direncinin ışıkla nasıl değiştiğini yorumla.

10. Elektronik Sensör Eğrisi Çiz

4-5. Hafta

LDR eğrisi

```
import numpy as np
import matplotlib.pyplot as plt
lux=np.logspace(0,3,200)
R=20000/(lux**0.65)
plt.semilogx(lux,R)
plt.xlabel("Işık (lux)"); plt.ylabel("Direnç (ohm)")
plt.title("LDR Sensör Simülasyonu")
plt.grid(); plt.show()
```

Görev: Y eksenini log ölçeğinde de göster.

NTC termistör

```
import numpy as np
import matplotlib.pyplot as plt
T=np.linspace(0,60,200)+273.15
R25=10000; B=3950; T25=298.15
R=R25*np.exp(B*(1/T-1/T25))
plt.plot(T-273.15,R)
plt.xlabel("Sıcaklık (°C)"); plt.ylabel("R (ohm)")
plt.grid(); plt.show()
```

Görev: B katsayısını 3500 yap.

NTC ile gerilim

```
Vcc=5
Rfixed=10000
Vout=Vcc*R/(Rfixed+R)
print(Vout[:5])
print(Vout[-5:])
```

Görev: Devreyi ters kurup formülü değiştir.

11. Veri Filtreleme: Hareketli Ortalama

1-5. Hafta

Gürültülü veri

```
import numpy as np
import matplotlib.pyplot as plt
rng=np.random.default_rng(1)
t=np.linspace(0,10,400)
gercek=25+2*np.sin(t)
olcum=gercek+rng.normal(0,0.8,len(t))
plt.plot(t,olcum,label="Ölçüm")
plt.plot(t,gercek,label="Gerçek")
plt.legend(); plt.grid(); plt.show()
```

Görev: Gürültüyü 0.2 ve 2.0 yap.

Hareketli ortalama

```
import pandas as pd
s=pd.Series(olcum)
filtre=s.rolling(15,center=True).mean()
plt.plot(t,olcum,alpha=0.35)
plt.plot(t,filtre,lw=2)
plt.grid(); plt.show()
```

Görev: 15 yerine 5 ve 41 kullan.

Filtreleme hatası

```
valid=~filtre.isna()
mae=np.mean(np.abs(gercek[valid]-filtre[valid]))
rmse=np.sqrt(np.mean((gercek[valid]-
filtre[valid])**2))
print("MAE:",mae)
print("RMSE:",rmse)
```

Görev: Üç farklı pencereyi karşılaştır.

12. Veri Filtreleme: Medyan ve Aykırı Değer

1-5. Hafta

Medyan filtre

```
%pip install scipy
import numpy as np
from scipy.signal import medfilt
x=np.array([10,11,10,12,10,80,11,10,12,11],float)
print(medfilt(x, kernel_size=3))
```

Görev: 80 değerini 500 yap.

IQR ile aykırı değer

```
import numpy as np
x=np.array([24,25,24.5,25.1,24.8,25.0,25.2,40,24.9])
q1,q3=np.percentile(x,[25,75])
iqr=q3-q1
alt=q1-1.5*iqr
ust=q3+1.5*iqr
print("Alt/Üst:",alt,ust)
print("Aykırıları:",x[(x<alt)|(x>ust)])
```

Görev: 40 değerini 30 yap ve tekrar çalıştır.

Kutu grafiği

```
import matplotlib.pyplot as plt
plt.boxplot(x)
plt.ylabel("Sıcaklık (°C)")
plt.title("Sensör Verisi")
plt.grid(axis="y")
plt.show()
```

Görev: İki sensör için iki kutu çiz.

13. CSV ile Veri Toplama

1-2. Hafta

CSV oluştur

```
import pandas as pd
veri=pd.DataFrame({
    "zaman":range(10),
    "sicaklik":
    [22,22.4,22.8,23.0,23.5,24,24.2,24.6,24.4,24.8],
    "nem":[55,54,53,52,51,50,49,48,49,47]
})
veri.to_csv("sensor.csv",index=False)
print(veri)
```

Görev: Dosyayı bilgisayarda bul ve aç.

CSV oku

```
df=pd.read_csv("sensor.csv")
print(df.head())
```

```
print(df.info())  
print(df.describe())
```

Görev: describe çıktısını incele.

Eksik veri

```
df.loc[3,"sicaklik"]=None  
df.loc[7,"nem"]=None  
print(df.isna().sum())  
df2=df.interpolate()  
print(df2)
```

Görev: interpolate yerine fillna kullanarak deney yap.

14. Pandas ile Veri Analizi

2-3. Hafta

Temel istatistikler

```
print("Ortalama sıcaklık:",df["sicaklik"].mean())
print("Maks sıcaklık:",df["sicaklik"].max())
print("Min sıcaklık:",df["sicaklik"].min())
print("Std:",df["sicaklik"].std())
```

Görev: Aynı işlemleri nem için yap.

Filtreleme

```
print(df[df["sicaklik"]>24])
print(df[df["nem"]<50])
```

Görev: İki koşulu & ile birleştir.

Sıralama

```
print(df.sort_values("sicaklik",ascending=False))
```

Görev: En düşük üç ölçümü yazdır.

15. Pandas: Gruplama ve Yeniden Örnekleme

2-3. Hafta

Sensör ve bölge verisi

```
import pandas as pd
df=pd.DataFrame({
    "bolge":["A","A","B","B","A","B","A","B"],
    "sicaklik":[20,22,24,23,25,27,21,26],
    "nem":[50,52,48,49,51,45,53,46]
})
print(df.groupby("bolge").mean(numeric_only=True))
```

Görev: Üçüncü bölgeyi ekle.

Saatlik veri

```
import numpy as np
zaman=pd.date_range("2026-01-01",
periods=48,freq="30min")
T=24+3*np.sin(np.arange(48)/8)
df2=pd.DataFrame({"sicaklik":T},index=zaman)
print(df2.resample("2h").mean())
```

Görev: 4 saatlik ortalama yap.

Günlük min-max

```
gunluk=df2.resample("1D").agg(["min","max","mean"])
print(gunluk)
```

Görev: Standart sapmayı da ekle.

16. Zaman Serisi Grafikleri

2-3. Hafta

Zaman eksenini

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,3.5))
plt.plot(df2.index,df2["sicaklik"])
plt.xlabel("Zaman"); plt.ylabel("°C")
plt.title("Sıcaklık Zaman Serisi")
plt.grid(); plt.tight_layout(); plt.show()
```

Görev: Şekil boyutunu değiştir.

Hareketli ortalama çiz

```
df2["MA"] = df2["sicaklik"].rolling(5).mean()
plt.plot(df2.index,df2["sicaklik"],label="Ölçüm",
alpha=.5)
plt.plot(df2.index,df2["MA"],label="MA(5)")
plt.legend(); plt.grid(); plt.show()
```

Görev: MA(10) ekle.

Alt grafik olmadan karşılaştır

```
plt.figure(figsize=(8,3.5))
plt.plot(df2.index,df2["sicaklik"],label="T")
plt.plot(df2.index,20+2*np.cos(np.arange(48)/6),
label="Referans")
plt.legend(); plt.grid(); plt.show()
```

Görev: Renk vermeden farklı çizgi stilleri kullan.

17. Dağılım Analizi

3. Hafta

Histogram ve bin sayısı

```
import numpy as np
import matplotlib.pyplot as plt
x=np.random.default_rng(4).normal(25,2,1000)
for bins in [10,20,40]:
    plt.figure(figsize=(6,3))
    plt.hist(x,bins=bins,edgecolor="black")
    plt.title(f"bins={bins}"); plt.show()
```

Görev: Üç grafiğin farkını gözlemle.

Ortalama, medyan ve çeyrekler

```
print("Ortalama:",x.mean())
print("Medyan:",np.median(x))
print("Q1:",np.percentile(x,25))
print("Q3:",np.percentile(x,75))
```

Görev: Sonuçları deftere yaz.

İki sensörü karşılaştır

```
a=np.random.default_rng(1).normal(25,1,500)
b=np.random.default_rng(2).normal(25,3,500)
plt.boxplot([a,b],
tick_labels=["Sensör A","Sensör B"])
plt.ylabel("°C"); plt.grid(axis="y"); plt.show()
```

Görev: Hangi sensörün yayılımı daha fazla? Sayısal değerle göster.

18. Hareket ve Pozisyon: Ultrasonik Sensör

10-11. Hafta

Mesafe simülasyonu

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,20,300)
d=50+20*np.sin(t/3)
plt.plot(t,d)
plt.xlabel("Zaman (s)"); plt.ylabel("Mesafe (cm)")
plt.grid(); plt.show()
```

Görev: Mesafeye gürültü ekle.

Gürültülü mesafe

```
rng=np.random.default_rng(8)
olcum=d+rng.normal(0,2,len(d))
plt.plot(t,d,label="Gerçek")
plt.plot(t,olcum,label="Sensör",alpha=.5)
plt.legend(); plt.grid(); plt.show()
```

Görev: Gürültü std=5 yap.

Mesafe alarmı

```
for x in olcum[:,20]:
    print(f"{x:5.1f} cm", "YAKIN" if x<35 else
          "NORMAL")
```

Görev: Eşiği 60 cm yap.

19. PIR Hareket Sensörü

10-11. Hafta

Dijital PIR

```
import numpy as np
rng=np.random.default_rng(9)
t=np.arange(60)
pir=(rng.random(60)<0.15).astype(int)
print(pir)
```

Görev: Hareket olasılığını 0.30 yap.

Hareket grafiği

```
import matplotlib.pyplot as plt
plt.step(t,pir,where="post")
plt.ylim(-0.1,1.1)
plt.xlabel("Zaman (s)"); plt.ylabel("PIR")
plt.grid(); plt.show()
```

Görev: Y eksenine 0=Yok, 1=Var düşün.

Hareket sayacı

```
print("Hareket sayısı:", pir.sum())
print("Hareket oranı:", pir.mean())
```

Görev: 60 saniye yerine 600 saniye üret.

20. İvmeölçer ve Hareket

10-11. Hafta

3 eksenli ivme

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,10,500)
ax=(np.sin(2*t)
+np.random.default_rng(1).normal(0,.1,len(t)))
ay=.5*np.cos(1.5*t)
az=9.81+.2*np.sin(4*t)
plt.plot(t,ax,label="ax")
plt.plot(t,ay,label="ay")
plt.plot(t,az,label="az")
plt.legend(); plt.grid(); plt.show()
```

Görev: Her eksenin genliğini değiştir.

Toplam ivme

```
a=np.sqrt(ax**2+ay**2+az**2)
print("Ortalama toplam ivme:",a.mean())
print("Maksimum:",a.max())
```

Görev: Sonucu 10 ölçüm için yazdır.

Hareket eşiği

```
hareket=a>10.0
print("Hareket örnekleri:",hareket.sum())
```

Görev: Eşiği 10.5 yap.

21. Pozisyon ve Hız

10-11. Hafta

Pozisyon üret

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,10,200)
x=t**2
plt.plot(t,x)
plt.xlabel("t (s)"); plt.ylabel("x (m)")
plt.grid(); plt.show()
```

Görev: $x=t^2$ yerine $x=5t$ yap.

Sayısal hız

```
v=np.gradient(x,t)
print(v[:10])
```

Görev: Hız grafiğini çiz.

İvme

```
a=np.gradient(v,t)
print("Ortalama ivme:",a.mean())
```

```
plt.plot(t,a); plt.xlabel("t (s)");  
plt.ylabel("m/s2"); plt.grid(); plt.show()
```

Görev: $x=t^3$ kullan.

22. Akış Sensörü: Debi ve Hacim

6-7. Hafta

Debi sensörü simülasyonu

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,30,300)
q=2+0.5*np.sin(t/4)
plt.plot(t,q)
plt.xlabel("Zaman (s)"); plt.ylabel("L/dk")
plt.grid(); plt.show()
```

Görev: Gürültü ekle.

Debiden hacim

```
dt=t[1]-t[0]
hacim=np.cumsum(q)*dt/60
print("Son hacim:",hacim[-1],"L")
plt.plot(t,hacim); plt.xlabel("Zaman");
plt.ylabel("L"); plt.grid(); plt.show()
```

Görev: Debiyi 3 L/dk yap.

Debi alarmı

```
alarm=q>2.4
print("Alarm örnekleri:",alarm.sum())
```

Görev: Eşiği 2.7 yap.

23. Tank Seviye Sensörü

6-7. Hafta

Tank dolum simülasyonu

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,20,200)
seviye=10+4*t
seviye=np.minimum(seviye,100)
plt.plot(t,seviye)
plt.xlabel("Zaman (dk)"); plt.ylabel("Seviye (%)")
plt.grid(); plt.show()
```

Görev: Başlangıç seviyesini 20 yap.

Taşma alarmı

```
for s in [30,60,80,95,101]:
    if s>=90: print(s,"TAŞMA RİSKİ")
    else: print(s,"NORMAL")
```

Görev: Alarm eşiğini 80 yap.

Pompa kontrolü

```
pompa=seviye<30
print("Pompa açık örnekleri:",pompa.sum())
```

Görev: Pompa açılma seviyesini 40 yap.

24. Sıcaklık Sensörü

12-13. Hafta

Sıcaklık simülasyonu

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,24,288)
T=22+4*np.sin(2*np.pi*t/24)
rng=np.random.default_rng(12)
olcum=T+rng.normal(0,.4,len(t))
plt.plot(t,olcum,label="Ölçüm")
plt.plot(t,T,label="Gerçek")
plt.xlabel("Saat"); plt.ylabel("°C")
plt.legend(); plt.grid(); plt.show()
```

Görev: 24 saat yerine 48 saat yap.

Min-max saatleri

```
i_min=np.argmin(olcum); i_max=np.argmax(olcum)
print("Min saat:",t[i_min],"değer:",olcum[i_min])
print("Maks saat:",t[i_max],"değer:",olcum[i_max])
```

Görev: Sonuçları yorumla.

Sıcaklık alarmı

```
print(">25 °C:", (olcum>25).sum())
print("<18 °C:", (olcum<18).sum())
```

Görev: Eşikleri değiştir.

25. Nem Sensörü

12-13. Hafta

Nem simülasyonu

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,24,288)
RH=55+12*np.cos(2*np.pi*t/24)
RH+=np.random.default_rng(4).normal(0,1.2,len(t))
plt.plot(t,RH)
plt.xlabel("Saat"); plt.ylabel("RH (%)")
plt.grid(); plt.show()
```

Görev: Gürültü std değerini değiştir.

Kuru / normal / nemli

```
for x in [35,45,55,70,85]:
    if x<40: d="KURU"
    elif x>70: d="NEMLİ"
    else: d="NORMAL"
    print(x,d)
```

Görev: Sınırları 30 ve 80 yap.

Sıcaklık-nem dağılımı

```
T=np.random.default_rng(1).normal(24,2,200)
RH=np.random.default_rng(2).normal(55,8,200)
plt.scatter(T,RH)
plt.xlabel("T (°C)"); plt.ylabel("RH (%)")
plt.grid(); plt.show()
```

Görev: Nokta sayısını 1000 yap.

26. Sensörler Arası İlişki

12-13. Hafta

Korelasyon

```
import pandas as pd
import numpy as np
df=pd.DataFrame({
    "T":np.random.default_rng(1).normal(25,2,200),
    "RH":np.random.default_rng(2).normal(55,8,200)
})
df["V"] = (0.5*df["T"] +
np.random.default_rng(3).normal(0,1,200))
print(df.corr(numeric_only=True))
```

Görev: T ile V korelasyonunu yorumla.

Saçılım grafiği

```
import matplotlib.pyplot as plt
plt.scatter(df["T"],df["V"])
plt.xlabel("T (°C)"); plt.ylabel("V (V)")
plt.grid(); plt.show()
```

Görev: Üzerine doğrusal trend çiz.

Basit regresyon

```
m,b=np.polyfit(df["T"],df["V"],1)
print("V =",m,"T +",b)
yhat=m*df["T"]+b
r2=1-np.sum((df["V"]-yhat)**2)/np.sum((df["V"]-
df["V"].mean())**2)
print("R²:",r2)
```

Görev: Gürültüyü azalt ve R² değişimini gözle.

27. Kimyasal Sensör: Gaz Konsantrasyonu

14-15. Hafta

Gaz sensörü

```
import numpy as np
import matplotlib.pyplot as plt
ppm=np.linspace(10,500,200)
R0=10000
R=R0/(1+ppm**0.8/30)
plt.plot(ppm,R)
plt.xlabel("Gaz (ppm)"); plt.ylabel("Direnç (ohm)")
plt.grid(); plt.show()
```

Görev: Üst sınırı 1000 ppm yap.

ADC ile gaz okuma

```
V=5*R0/(R0+R)
adc=np.round(V/5*1023).astype(int)
```

```
print adc[:10])  
print("Maks ADC:", adc.max())
```

Görev: 12 bit ADC kullan.

Gaz alarmı

```
alarm=ppm>300  
print("Alarm oranı:", alarm.mean()*100, "%")
```

Görev: Alarm eşiğini 150 ppm yap.

28. Kimyasal Sensör: pH

14-15. Hafta

pH verisi

```
import numpy as np
import matplotlib.pyplot as plt
t=np.arange(30)
pH=(7+0.4*np.sin(t/3)
+np.random.default_rng(5).normal(0,.08,30))
plt.plot(t,pH,marker="o")
plt.axhline(6.5,linestyle="--")
plt.axhline(8.5,linestyle="--")
plt.xlabel("Ölçüm"); plt.ylabel("pH")
plt.grid(); plt.show()
```

Görev: Eşik değerlerini değiştir.

pH sınıflandırma

```
for x in pH[:10]:
    if x<6.5: s="ASİDİK"
    elif x>8.5: s="BAZİK"
    else: s="NORMAL"
    print(round(x,2),s)
```

Görev: Her sınıfın sayısını hesapla.

pH istatistikleri

```
print("Ortalama:",pH.mean())
print("Medyan:",np.median(pH))
print("Min:",pH.min())
print("Maks:",pH.max())
```

Görev: Ölçüm sayısını 300 yap.

29. Veri Temizleme Laboratuvarı

9-13. Hafta

Eksik + aykırı + gürültü

```
import numpy as np
import pandas as pd
rng=np.random.default_rng(7)
t=np.arange(100)
x=25+np.sin(t/8)+rng.normal(0,.5,100)
x[[10,30,70]]=np.nan
x[55]=50
df=pd.DataFrame({"zaman":t,"T":x})
print(df.isna().sum())
```

Görev: İkinci bir aykırı değer ekle.

Eksikleri doldur

```
df["T_fill"]=df["T"].interpolate()
print(df.head(15))
```

Görev: ffill ve bfill deneyin.

Aykırıyı sınırla

```
q1=df["T_fill"].quantile(.25)
q3=df["T_fill"].quantile(.75)
iqr=q3-q1
low=q1-1.5*iqr; high=q3+1.5*iqr
df["T_clean"]=df["T_fill"].clip(low,high)
print(df["T_clean"].describe())
```

Görev: low/high değerlerini yazdır.

30. Veri Analizi: Özet Rapor

9-13. Hafta

Bir sensör raporu

```
rapor=pd.Series({
    "ölçüm_sayısı":len(df),
    "ortalama":df["T_clean"].mean(),
    "std":df["T_clean"].std(),
    "min":df["T_clean"].min(),
    "max":df["T_clean"].max()
})
print(rapor)
```

Görev: Rapora medyan ve yüzde 95 ekle.

CSV raporu

```
rapor.to_csv("sensor_raporu.csv",header=["deger"])
print("Rapor kaydedildi")
```

Görev: Dosyayı kontrol et.

Grafikli rapor

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,3.5))
plt.plot(df["zaman"],df["T"],label="Ham",alpha=.4)
plt.plot(df["zaman"],df["T_clean"],label="Temiz")
plt.legend(); plt.xlabel("Ölçüm"); plt.ylabel("°C")
plt.grid(); plt.tight_layout();
plt.savefig("sensor_raporu.png",dpi=150)
plt.show()
```

Görev: PNG dosyasını aç.

31. Çoklu Sensör Panosu

1-15. Hafta

4 sensörü tek tabloda üret

```
import numpy as np
import pandas as pd
n=200
t=np.arange(n)
rng=np.random.default_rng(11)
df=pd.DataFrame({
    "zaman":t,
    "T":24+2*np.sin(t/20)+rng.normal(0,.3,n),
    "RH":55+8*np.cos(t/25)+rng.normal(0,1,n),
    "mesafe":50+15*np.sin(t/15)+rng.normal(0,1.5,n),
    "debi":2+rng.normal(0,.2,n)
})
print(df.head())
```

Görev: Beşinci sensörü ekle.

4 grafik

```
import matplotlib.pyplot as plt
fig,ax=plt.subplots(2,2,figsize=(9,6))
ax[0,0].plot(t,df["T"]);
ax[0,0].set_title("Sıcaklık")
ax[0,1].plot(t,df.RH); ax[0,1].set_title("Nem")
ax[1,0].plot(t,df.mesafe);
ax[1,0].set_title("Mesafe")
ax[1,1].plot(t,df.debi); ax[1,1].set_title("Debi")
for a in ax.ravel(): a.grid()
plt.tight_layout(); plt.show()
```

Görev: Başlık ve birim ekle.

Anlık değer kutusu

```
print(df.iloc[-1])
print("T alarm:",df["T"].iloc[-1]>27)
print("Mesafe alarm:",df.mesafe.iloc[-1]<35)
```

Görev: Alarm eşiklerini değiştir.

32. Görselleştirme: Isı Haritası

9-13. Hafta

Korelasyon matrisi

```
import numpy as np
import matplotlib.pyplot as plt
C=df.corr(numeric_only=True)
plt.imshow(C,aspect="auto")
plt.xticks(range(len(C.columns)),C.columns,rotation=45)
plt.yticks(range(len(C.columns)),C.columns)
plt.colorbar(label="Korelasyon")
plt.title("Sensör Korelasyon Matrisi")
plt.tight_layout(); plt.show()
```

Görev: Kendi veri setin için çalıştır.

Değerleri hücelere yaz

```
fig,ax=plt.subplots(figsize=(6,5))
im=ax.imshow(C, vmin=-1, vmax=1)
ax.set_xticks(range(len(C.columns)));
ax.set_xticklabels(C.columns,rotation=45)
ax.set_yticks(range(len(C.columns)));
ax.set_yticklabels(C.columns)
for i in range(len(C)):
    for j in range(len(C)):
        ax.text(j,i,f"{C.iloc[i,j]:.2f}",
                ha="center",va="center")
fig.colorbar(im,ax=ax); plt.tight_layout();
plt.show()
```

Görev: Hücre yazı boyutunu değiştir.

33. Görselleştirme: Histogram + Kutu Grafiği + ECDF

9-13. Hafta

Histogram + medyan

```
x=df["T"].dropna()
import matplotlib.pyplot as plt
plt.hist(x,bins=20,edgecolor="black")
plt.axvline(x.mean(),linestyle="--",label="Ort.")
plt.axvline(x.median(),linestyle=":",label="Medyan")
plt.legend(); plt.grid(); plt.show()
```

Görev: 95. persentili ekle.

Kutu grafiği

```
plt.boxplot(x,orientation="horizontal")
plt.xlabel("Sıcaklık (°C)"); plt.grid(axis="x");
plt.show()
```

Görev: Grafiğin başlığını değiştir.

ECDF

```
xs=np.sort(x.to_numpy())
y=np.arange(1,len(xs)+1)/len(xs)
plt.plot(xs,y)
plt.xlabel("Sıcaklık (°C)");
plt.ylabel("Kümülatif oran")
plt.grid(); plt.show()
```

Görev: 0.5 yatay çizgisi ekle.

34. Basit Veri Analizi: Trend ve Tahmin

9-13. Hafta

Trend doğrusu

```
import numpy as np
import matplotlib.pyplot as plt
y=df["T"].to_numpy()
x=np.arange(len(y))
m,b=np.polyfit(x,y,1)
trend=m*x+b
plt.plot(x,y,label="Sensör")
plt.plot(x,trend,label="Trend")
plt.legend(); plt.grid(); plt.show()
print("Eğim: ",m)
```

Görev: Trendin pozitif/negatif oluşunu gözlemle.

Gelecek 20 nokta

```
xf=np.arange(len(y),len(y)+20)
yf=m*xf+b
plt.plot(x,y)
plt.plot(xf,yf,linestyle="--")
plt.grid(); plt.show()
```

Görev: 20 yerine 50 yap.

Hata ölçümü

```
print("MAE:",np.mean(np.abs(y-trend)))
print("RMSE:",np.sqrt(np.mean((y-trend)**2)))
```

Görev: Trend modelini gürültülü veri ile tekrar dene.

35. Final Proje 1: Akıllı Oda Simülasyonu

Hafta 10-13

Veriyi üret

```
import numpy as np
import pandas as pd
rng=np.random.default_rng(20)
t=np.arange(0,3600,10)
T=23+2*np.sin(t/700)+rng.normal(0,.3,len(t))
RH=50+8*np.cos(t/900)+rng.normal(0,1,len(t))
PIR=(rng.random(len(t))<.08).astype(int)
light=450+150*np.sin(t/500)+rng.normal(0,25,len(t))
df=pd.DataFrame({"t":t,"T":T,"RH":RH,"PIR":PIR,"lux":light})
print(df.head())
```

Görev: Kodun çalışma sonucunu incele.

Kural tabanlı karar

```
df["klima"]=(df["T"]>25)&(df.PIR==1)
df["havalandırma"]=(df.RH>65)
```

```
df["isik"]=(df.lux<250)&(df.PIR==1)
print(df[["klima","havalandirma","isik"]].mean()*100
)
```

Görev: Kuralları değiştir.

Paneli çiz

```
fig,ax=plt.subplots(3,1,figsize=(9,7),sharex=True)
ax[0].plot(df.t/60,df["t"]);
ax[0].set_ylabel("%C"); ax[0].grid()
ax[1].plot(df.t/60,df.RH); ax[1].set_ylabel("RH %");
ax[1].grid()
ax[2].step(df.t/60,df.PIR,where="post");
ax[2].set_ylabel("PIR"); ax[2].set_xlabel("Dakika");
ax[2].grid()
plt.tight_layout(); plt.show()
```

Görev: Grafik başlıklarını ekle.

36. Final Proje 2: Akıllı Su Tankı

Hafta 6-7

Tank verisi

```
import numpy as np
import pandas as pd
rng=np.random.default_rng(30)
t=np.arange(0,120,1)
debi_giris=2+rng.normal(0,.2,len(t))
debi_cikis=1.7+rng.normal(0,.2,len(t))
seviye=np.cumsum(debi_giris-debi_cikis)
seviye=50+seviye
seviye=np.clip(seviye,0,100)
tank=pd.DataFrame({"t":t,"giris":debi_giris,
"cikis":debi_cikis,"seviye":seviye})
print(tank.tail())
```

Görev: Çıkış debisini 2.5 yap.

Pompa kuralı

```
tank["pompa"]=(tank.seviye<30)
tank.loc[tank.seviye>80,"pompa"]=False
print(tank[tank.pompa].head())
```

Görev: Alt ve üst eşikleri değiştir.

Tank grafiği

```
fig,ax=plt.subplots(figsize=(9,4))
ax.plot(tank.t,tank.seviye,label="Seviye (%)")
ax.axhline(30,linestyle="--")
ax.axhline(80,linestyle="--")
ax.set_xlabel("Zaman (dk)");
ax.set_ylabel("Seviye (%)")
ax.grid(); ax.legend(); plt.tight_layout();
plt.show()
```

Görev: Alt ve üst çizgilere açıklama ekle.

37. Final Proje 3: Sensör Veri Analiz Raporu

Hafta 9-15

Ham veriyi kaydet

```
df.to_csv("final_sensor_data.csv",index=False)
print("Kaydedildi:",df.shape)
```

Görev: CSV dosyasını aç.

Özet tablo

```
ozet=df.describe(include="all")
print(ozet)
ozet.to_csv("final_ozet.csv")
```

Görev: Özet dosyasını kontrol et.

3 temel grafik

```
fig,ax=plt.subplots(1,3,figsize=(12,3.5))
ax[0].hist(df["T"].dropna(),bins=20,
edgecolor="black")
ax[0].set_title("T Dağılımı")
ax[1].boxplot(df["RH"].dropna())
ax[1].set_title("RH")
ax[2].scatter(df["T"],df["RH"],s=10)
ax[2].set_xlabel("T"); ax[2].set_ylabel("RH")
for a in ax: a.grid()
plt.tight_layout(); plt.show()
```

Görev: Üç grafiği açıklayacak üç cümle yaz.

38. Final Proje 4: Çoklu Sensör Analiz Uygulaması

Hafta 1-15

Birleştir

```
import numpy as np
import pandas as pd
rng=np.random.default_rng(40)
n=500
t=np.arange(n)
df=pd.DataFrame({
    "T":24+2*np.sin(t/50)+rng.normal(0,.5,n),
    "RH":55+10*np.cos(t/70)+rng.normal(0,1.5,n),
    "mesafe":60+20*np.sin(t/35)+rng.normal(0,3,n),
    "debi":2+rng.normal(0,.25,n),
    "gaz":100+20*np.sin(t/40)+rng.normal(0,8,n),
    "hareket":(rng.random(n)<.12).astype(int)
})
print(df.describe())
```

Görev: Yeni bir sensör ekle.

Temizle + filtrele

```
for c in ["T", "RH", "mesafe", "debi", "gaz"]:
    df[c]=df[c].interpolate()
    df[c]=df[c].rolling(7,min_periods=1).median()
print(df.head())
```

Görev: rolling penceresini 15 yap.

Korelasyon ve rapor

```
print(df.corr(numeric_only=True).round(2))
df.to_csv("coklu_sensör_temiz.csv",index=False)
print("Temiz veri kaydedildi")
```

Görev: Korelasyon matrisini ısı haritasına çevir.

39. Laboratuvar Görevleri: Kodu Değiştir

Hafta 1-15

Görev A • Gürültü

```
import numpy as np
import matplotlib.pyplot as plt
rng=np.random.default_rng(50)
t=np.linspace(0,20,500)
true=30+3*np.sin(t)
noise=rng.normal(0,1.5,len(t))
measured=true+noise
filtered=np.convolve(measured,np.ones(15)/15,
mode="same")
plt.plot(t,measured,alpha=.35,label="Ölçüm")
plt.plot(t,filtered,label="Filtre")
```

```
plt.plot(t,true,label="Gerçek")
plt.legend(); plt.grid(); plt.show()
```

Görev: Filtre penceresini 5, 25 ve 51 yap. RMSE hesapla.

Görev B • Sensör karşılaştırma

```
import numpy as np
import matplotlib.pyplot as plt
rng=np.random.default_rng(51)
ref=np.linspace(0,100,100)
a=ref+rng.normal(0,2,100)
b=1.03*ref+2+rng.normal(0,1,100)
plt.scatter(ref,a,label="A")
plt.scatter(ref,b,label="B")
plt.plot(ref,ref,linestyle="--",label="İdeal")
plt.xlabel("Referans"); plt.ylabel("Sensör")
plt.legend(); plt.grid(); plt.show()
```

Görev: A ve B için MAE ve bias hesapla.

Görev C • Eşik alarmı

```
alarm=(measured>31)
print("Alarm sayısı:",alarm.sum())
print("Alarm oranı %:",100*alarm.mean())
```

Görev: Eşiği 29 ve 33 ile karşılaştır.

40. Final Alıştırma: Baştan Sona Sensör Sistemi

Hafta 1-15

1) Veri üret

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
rng=np.random.default_rng(60)
n=600
t=np.arange(n)
T=24+3*np.sin(t/60)+rng.normal(0,.6,n)
RH=50+9*np.cos(t/80)+rng.normal(0,1.5,n)
mesafe=55+18*np.sin(t/45)+rng.normal(0,2,n)
PIR=(rng.random(n)<.1).astype(int)
debi=2+rng.normal(0,.25,n)
df=pd.DataFrame({"t":t,"T":T,"RH":RH,"mesafe":mesafe,
"PIR":PIR,"debi":debi})
print(df.head())
```

Görev: Veri sayısını 1000 yap.

2) Analiz et

```
print(df.describe())
print("Eksik değerler")
print(df.isna().sum())
print("Korelasyon")
print(df.corr(numeric_only=True).round(2))
```

Görev: En büyük korelasyonu bul.

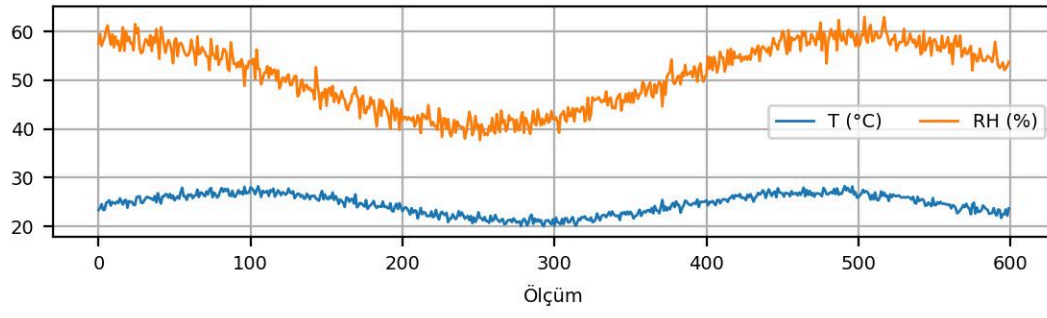
3) Temizle, filtrele, görselleştir

```
for c in ["T","RH","mesafe","debi"]:
    df[c]=df[c].rolling(9,min_periods=1).mean()

fig,ax=plt.subplots(4,1,figsize=(9,8),sharex=True)
ax[0].plot(df.t,df["T"]); ax[0].set_ylabel("°C")
ax[1].plot(df.t,df.RH); ax[1].set_ylabel("RH %")
ax[2].plot(df.t,df.mesafe); ax[2].set_ylabel("cm")
ax[3].plot(df.t,df.debi); ax[3].set_ylabel("L/dk")
for a in ax: a.grid()
plt.tight_layout(); plt.show()
df.to_csv("final_sensor_sistemi.csv",index=False)
```

Görev: Sonuçlarını kaydet. Kendi sensörünü sisteme ekle.

Örnek çıktı • Çoklu sensör grafiği



41. Veri Analizi • Standartlaştırma ve Normalizasyon

Hafta 3-9

Z-skoru

```
x=np.array([20,22,24,26,28,30],float)
z=(x-x.mean())/x.std()
print(z)
```

Görev: Ortalaması 0 ve std değeri 1 olacak mı kontrol et.

Min-max normalizasyon

```
xmin,xmax=x.min(),x.max()
x01=(x-xmin)/(xmax-xmin)
print(x01)
```

Görev: Sonucu 0-1 aralığında gözle.

Robust ölçek

```
med=np.median(x)
iqr=np.percentile(x,75)-np.percentile(x,25)
print((x-med)/iqr)
```

Görev: Aykırı bir değer ekle.

Dağılım grafiği

```
plt.hist(x,bins=6,edgecolor="black")
plt.axvline(x.mean(),linestyle="--")
plt.grid(); plt.show()
```

Görev: Grafiği genişlet.

42. Veri Analizi • Hata Ölçümleri

Hafta 3-15

MAE

```
y=np.array([10,12,15,18,20],float)
yh=np.array([11,11,14,20,19],float)
mae=np.mean(np.abs(y-yh))
print(mae)
```

Görev: Bir tahmini değiştir.

RMSE

```
rmse=np.sqrt(np.mean((y-yh)**2))
print(rmse)
```

Görev: MAE ile karşılaştır.

Bias

```
bias=np.mean(yh-y)
print("bias=",bias)
```

Görev: Tahminleri hep 2 artır.

MAPE

```
mape=np.mean(np.abs((y-yh)/y))*100  
print("MAPE % =",mape)
```

Görev: Sıfıra yakın gerçek değer ekleyip sonucu gözle.

43. Görselleştirme • Çizgi, Saçılım ve Çubuk

Hafta 2-15

Üç grafik

```
x=np.arange(1,8)
y=[12,15,14,18,20,19,22]
plt.plot(x,y,marker="o"); plt.grid(); plt.show()
plt.scatter(x,y); plt.grid(); plt.show()
plt.bar(x,y); plt.grid(axis="y"); plt.show()
```

Görev: Üç grafiği ayrı ayrı karşılaştır.

Çoklu seri

```
a=20+0.5*x
b=25-0.3*x
plt.plot(x,a,label="A")
plt.plot(x,b,label="B")
plt.legend(); plt.grid(); plt.show()
```

Görev: A ve B eğimlerini değiştir.

Hareketli ortalama

```
s=pd.Series(y)
plt.plot(x,s,label="ham")
plt.plot(x,s.rolling(3).mean(),label="MA")
plt.legend(); plt.grid(); plt.show()
```

Görev: Pencereyi 5 yap.

44. Görselleştirme • Dağılım ve Aykırı Değer

Hafta 3-13

Histogram

```
rng=np.random.default_rng(80)
x=rng.normal(25,2,500)
plt.hist(x,bins=20,edgecolor="black")
plt.grid(); plt.show()
```

Görev: 500 yerine 2000 üret.

Kutu grafiği

```
plt.boxplot(x,orientation="horizontal")
plt.xlabel("°C"); plt.grid(axis="x"); plt.show()
```

Görev: 40 ve 50 değerlerini ekle.

ECDF

```
xs=np.sort(x)
y=np.arange(1,len(xs)+1)/len(xs)
plt.plot(xs,y)
plt.grid(); plt.show()
```

Görev: 0.5 yatay çizgisi ekle.

Persentil çizgileri

```
for q in [5, 50, 95]:  
    plt.axvline(np.percentile(x, q), linestyle="--")  
plt.hist(x, bins=20, alpha=.5); plt.grid(); plt.show()
```

Görev: Q değerlerini ekrana yazdır.

45. Zaman Serisi • Tepe ve Trend

Hafta 9-13

Tepe değeri bul

```
t=np.linspace(0,20,400)
y=(25+3*np.sin(t)
+np.random.default_rng(4).normal(0,.4,400))
i=np.argmax(y)
print(t[i],y[i])
```

Görev: En düşük değeri de bul.

Değişim hızı

```
dy=np.gradient(y,t)
plt.plot(t,dy)
plt.xlabel("Zaman"); plt.ylabel("dy/dt");
plt.grid(); plt.show()
```

Görev: Ani değişim bölgesini bul.

Trend

```
m,b=np.polyfit(t,y,1)
plt.plot(t,y,alpha=.5)
plt.plot(t,m*t+b,linestyle="--")
plt.grid(); plt.show()
```

Görev: Trend eğimini yazdır.

Filtre + trend

```
f=pd.Series(y).rolling(15,center=True).mean()
plt.plot(t,y,alpha=.3)
plt.plot(t,f); plt.plot(t,m*t+b,linestyle=":")
plt.grid(); plt.show()
```

Görev: Pencereyi 31 yap.

46. Sensör Karşılaştırma Laboratuvarı

Hafta 3-15

İki sensör

```
rng=np.random.default_rng(81)
ref=np.linspace(0,100,100)
A=ref+rng.normal(0,2,100)
B=1.02*ref+1+rng.normal(0,1,100)
plt.scatter(ref,A,label="A",s=10)
plt.scatter(ref,B,label="B",s=10)
plt.plot(ref,ref,linestyle="--")
plt.legend(); plt.grid(); plt.show()
```

Görev: Grafikte ideal doğruyu incele.

Hata tablosu

```
for name,p in [("A",A),("B",B)]:
    e=p-ref
```

```
print(name, "MAE", np.mean(np.abs(e)), "RMSE",  
      np.sqrt(np.mean(e**2)), "bias", np.mean(e))
```

Görev: Çıktıları tablo haline getir.

Hata histogramı

```
plt.hist(A-ref, bins=20, alpha=.5, label="A")  
plt.hist(B-ref, bins=20, alpha=.5, label="B")  
plt.legend(); plt.grid(); plt.show()
```

Görev: Bin sayısını 30 yap.

Bias düzeltme

```
B2=B-np.mean(B-ref)  
print("Yeni bias:", np.mean(B2-ref))
```

Görev: Kalibrasyon sonrası MAE hesapla.

47. Kalibrasyon • Doğrusal ve Polinom

Hafta 3-5

Doğrusal

```
v=np.linspace(.5,4.5,30)
y=10*v+3+np.random.default_rng(82).normal(0,.8,30)
k,b=np.polyfit(v,y,1)
print(k,b)
```

Görev: Katsayıları not et.

Polinom 2. derece

```
p=np.polyfit(v,y,2)
y2=np.polyval(p,v)
print(p)
```

Görev: Polinom katsayılarını incele.

Karşılaştır

```
plt.scatter(v,y,s=12,label="ölçüm")
plt.plot(v,k*v+b,label="doğrusal")
plt.plot(v,y2,label="poly2")
plt.legend(); plt.grid(); plt.show()
```

Görev: Grafiğe gerçek modeli ekle.

RMSE karşılaştır

```
r1=np.sqrt(np.mean((y-(k*v+b))**2))
r2=np.sqrt(np.mean((y-y2)**2))
print("doğrusal",r1,"poly2",r2)
```

Görev: Her model için hata yazdır.

48. Pandas • Temizleme ve Veri Kalitesi

Hafta 1-15

Aralık kontrolü

```
df=pd.DataFrame({"T":[24,25,99,23,-10,24],"RH":
[50,55,120,48,52,51]})
df.loc[(df["T"]<0)|(df["T"]>50),"T"]=np.nan
df.loc[(df.RH<0)|(df.RH>100),"RH"]=np.nan
print(df)
```

Görev: Sınırları değiştir.

Tekrarlı kayıt

```
df=pd.concat([df,df.iloc[[0]]],ignore_index=True)
print(df.duplicated().sum())
df=df.drop_duplicates()
print(df)
```

Görev: İki tekrar daha ekle.

Eksik doldurma

```
df["T"]=df["T"].interpolate()  
df["RH"]=df["RH"].fillna(df["RH"].median())  
print(df)
```

Görev: mean ile doldurmayı dene.

Kalite raporu

```
print("boyut", df.shape)  
print("eksik")  
print(df.isna().sum())  
print(df.describe())
```

Görev: Raporu CSV olarak kaydet.

49. Mini Proje • Sensör Veri Günlüğü

Hafta 1-15

1. Veri üret

```
import numpy as np
import pandas as pd
rng=np.random.default_rng(90)
n=300
t=pd.date_range("2026-01-01",periods=n,freq="min")
T=24+2*np.sin(np.arange(n)/30)+rng.normal(0,.4,n)
RH=55+7*np.cos(np.arange(n)/45)+rng.normal(0,1,n)
df=pd.DataFrame({"zaman":t,"T":T,"RH":RH})
print(df.head())
```

Görev: n değerini 1000 yap.

2. Temizle

```
df.loc[50,"T"]=60
df.loc[80,"RH"]=150
df.loc[120,"T"]=np.nan
df.loc[df["T"]>50,"T"]=np.nan
df.loc[df.RH>100,"RH"]=np.nan
df=df.interpolate()
print(df.describe())
```

Görev: Başka bir aykırı değer ekle.

3. Filtrele

```
df["T_f"] = df["T"].rolling(9,min_periods=1).mean()
df["RH_f"] = df["RH"].rolling(9,min_periods=1).mean()
print(df[["T","T_f","RH","RH_f"]].head(12))
```

Görev: Pencereyi 21 yap.

4. Görselleştir ve kaydet

```
fig,ax=plt.subplots(2,1,figsize=(9,5),sharex=True)
ax[0].plot(df.zaman,df["T"],alpha=.4);
ax[0].plot(df.zaman,df.T_f)
ax[1].plot(df.zaman,df.RH,alpha=.4);
ax[1].plot(df.zaman,df.RH_f)
for a in ax: a.grid()
plt.tight_layout();
plt.savefig("veri_gunlugu.png",dpi=140); plt.show()
df.to_csv("veri_gunlugu.csv",index=False)
```

Görev: PNG ve CSV dosyalarını kontrol et.

50. Sinyal Kalitesi • Gürültü, SNR ve Değişim

Hafta 1-9

Sinyal + gürültü

```
t=np.linspace(0,10,500)
rng=np.random.default_rng(100)
true=25+2*np.sin(t)
noise=rng.normal(0,1,500)
y=true+noise
print("gürültü std=",noise.std())
```

Görev: Gürültü std değerini 0.2, 1 ve 3 yap.

SNR

```
signal_power=np.mean(true**2)
noise_power=np.mean(noise**2)
snr=10*np.log10(signal_power/noise_power)
print("SNR(dB)=",snr)
```

Görev: Gürültü azaldığında SNR değerini gözle.

Tepe-tepe değer

```
print("Tepe-tepe:",np.ptp(y))
print("RMS:",np.sqrt(np.mean(y**2)))
```

Görev: RMS ve tepe-tepe değerlerini not et.

Ani değişim grafiği

```
dy=np.gradient(y,t)
plt.plot(t,dy)
plt.xlabel("Zaman"); plt.ylabel("Değişim hızı");
plt.grid(); plt.show()
```

Görev: Ani değişim bölgelerini bul.

51. Görselleştirme • Sensör Gösterge Paneli

Hafta 9-15

Gösterge paneli

```
fig,ax=plt.subplots(2,2,figsize=(9,5))
ax[0,0].plot(df["T"]); ax[0,0].set_title("Sıcaklık")
ax[0,1].plot(df.RH); ax[0,1].set_title("Nem")
ax[1,0].hist(df["T"],bins=20,edgecolor="black");
ax[1,0].set_title("T dağılımı")
ax[1,1].scatter(df["T"],df.RH,s=10);
ax[1,1].set_title("T-RH")
for a in ax.ravel(): a.grid()
plt.tight_layout(); plt.show()
```

Görev: Grafiklere birim ekle.

Eşik çizgileri

```
plt.figure(figsize=(8,3))
plt.plot(df["T"])
plt.axhline(27,linestyle="--")
plt.axhline(20,linestyle=":")
plt.grid(); plt.show()
```

Görev: İki eşik değerini değiştir.

Anlık bilgi

```
last=df.iloc[-1]
print(last)
print("T alarm:",last["T"]>27)
print("Nem alarm:",last.RH>70)
```

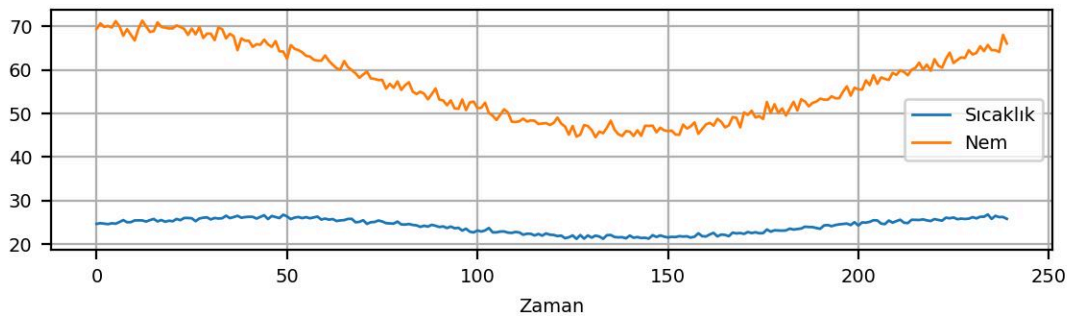
Görev: Üçüncü alarm kuralı ekle.

Dosya çıktısı

```
fig.savefig("dashboard.png",dpi=150)
df.to_csv("dashboard.csv",index=False)
print("Dosyalar kaydedildi")
```

Görev: Dosyaları kontrol et.

Örnek çıktı • Gösterge paneli



52. Serbest Proje • Kendi Sensörünü Tasarla

Hafta 1-15

1. Sensör modeli

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
rng=np.random.default_rng(200)
n=400
t=np.arange(n)
gercek=20+5*np.sin(t/30)
olcum=gercek+rng.normal(0,1,n)
df=pd.DataFrame({"t":t,"gercek":gercek,"olcum":olcum})
print(df.head())
```

Görev: Gerçek sensör büyüklüğünü seç ve isimleri değiştir.

2. Temizleme

```
df.loc[50,"olcum"]=80
df.loc[100,"olcum"]=np.nan
df["olcum"]=df["olcum"].interpolate()
q1=df.olcum.quantile(.25);
q3=df.olcum.quantile(.75); iqr=q3-q1
df["temiz"]=df.olcum.clip(q1-1.5*iqr,q3+1.5*iqr)
```

Görev: Aykırı değeri kaldırmak yerine medyan filtresi dene.

3. Analiz

```
e=df.temiz-df.gercek
print("MAE",np.mean(np.abs(e)))
print("RMSE",np.sqrt(np.mean(e**2)))
print("Bias",np.mean(e))
```

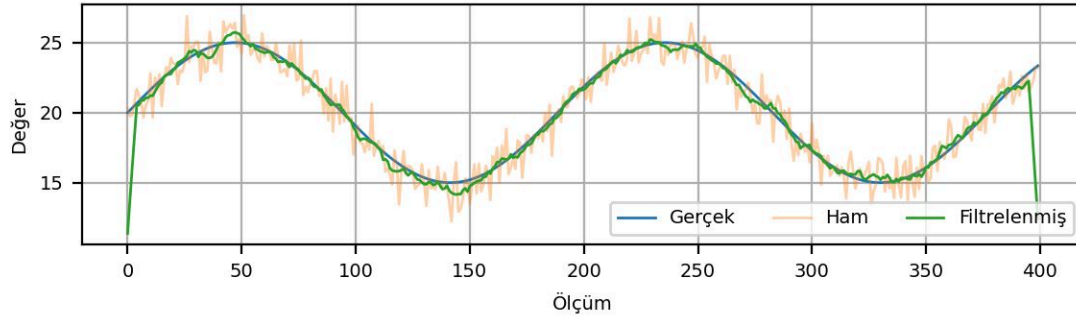
Görev: Üç metriği karşılaştır.

4. Sonuç grafiği

```
plt.plot(df.t,df.gercek,label="Gerçek")
plt.plot(df.t,df.olcum,alpha=.35,label="Ham")
plt.plot(df.t,df.temiz,label="Temiz")
plt.legend(); plt.grid(); plt.show()
df.to_csv("kendi_sensorum.csv",index=False)
```

Görev: Kendi sensörünü ekle ve raporu teslim et.

Örnek çıktı • Ham / temiz sensör verisi



Ham veriyi, filtrelenmiş veriyi ve referans sinyali aynı grafikte incele.

Python Uygulamalarıyla Sensör Teknolojileri ve Veri Toplama

Bu kitap, Sensör Teknolojileri ve Veri Toplama dersi için hazırlanmış uygulamalı bir Python çalışma kitabıdır. Python kurulumundan başlayarak NumPy, Pandas ve Matplotlib ile sensör verisinin nasıl üretileceğini, toplanacağını, temizleneceğini, filtreleneceğini ve analiz edileceğini adım adım gösterir.

Her bölümde öğrencinin kendisi yazarak çalıştıracağı kısa kodlar ve kodu değiştirip sonucu gözlemlemesini isteyen görevler yer alır.

Kitapta

- Veri toplama, gürültü ve filtreleme
- Kalibrasyon, ADC ve sensör devreleri
- Hareket, mesafe, akış ve seviye sensörleri
- Sıcaklık, nem ve kimyasal (gaz, pH) sensörler
- İstatistik, hata ölçümleri ve görselleştirme
- Akıllı oda, akıllı su tankı ve çoklu sensör projeleri

Dr. Selim Ünal

Kastamonu Üniversitesi

Metalurji ve Malzeme Mühendisliği Bölümü

ISBN 978-625-00-5106-1



9 786250 051061