

VERİ BİLİMİ DERSİ

Veri Görselleştirme ve Hikayeleştirme

Hafta 3 · Modül 3 — Veriden Anlam Çıkarmak

Dr. Murat Altun

Veri Bilimi ve Yapay Zekâ Eğitimi · 2026

6

Saat

4

Notebook

5+

Grafik Türü

1

Matplotlib Temelleri

Line, bar, histogram, scatter, pie chart; grafik özelleştirme

Slayt 3-8

2

Seaborn ile İleri Görselleştirme

countplot, boxplot, violinplot, heatmap, pairplot

Slayt 9-13

3

Bar Chart Race & Google Trends

Animasyonlu grafikler, pytrends ile veri çekme

Slayt 14-15

4

Oto-EDA Araçları

Sweetviz, YData-Profiling ile otomatik keşifsel analiz

Slayt 16-17

Anscombe Dörtlüsü (1973)

Dört farklı veri seti aynı ortalama, varyans ve korelasyona sahiptir — ancak grafikleri birbirinden tamamen farklıdır.

Bu örnek, sadece istatistiksel özetlerin yeterli olmadığını ve görselleştirmenin neden kritik olduğunu kanıtlar.

7.5

Ortalama X

0.82

Korelasyon

4

Farklı Dağılım

Veri Görselleştirmenin Temel İlkeleri

- Doğru grafik türü seçimi: Karşılaştırma → Bar, Dağılım → Histogram, İlişki → Scatter
- Renk, etiket ve başlık ile hikaye anlatımı (data storytelling)
- Hedef kitleye uygun sadelik düzeyi — teknik vs yönetici raporu

matplotlib.pyplot Modülü

Python'un en yaygın görselleştirme kütüphanesidir. MATLAB benzeri bir arayüz sunar.

Temel bileşenler:

- Figure — tüm grafiğin taşıyıcısı
- Axes — tek bir grafik alanı
- Artist — çizgi, metin, etiket vb.

plt.show() ile grafiği ekrana basarsınız.

plt.savefig() ile dosyaya kaydedersiniz.

```
import matplotlib.pyplot as plt# Basit çizgi grafiği
x = [1, 2, 3, 4, 5]y = [10, 25, 18, 32, 28]plt.plot(x,
y, color='teal', marker='o',
linewidth=2)plt.xlabel('Gün')plt.ylabel('Satış')plt.ti
tle('Günlük Satış Trendi')plt.grid(True,
alpha=0.3)plt.show()
```

plt.bar() — Kategorik Karşılaştırma

Kategorik değişkenlerin büyüklük karşılaştırması için kullanılır.

Parametreler: x (kategoriler), height (değerler), color, edgecolor, width, label

```
# Bar Grafiği  
sehirler = ['İstanbul', 'Ankara', 'İzmir']  
nufus = [16, 5.7, 4.4]  
plt.bar(sehirler, nufus,  
color=['#2B7A78', '#C97B5A',  
'#E8913A'])
```

plt.hist() — Sürekli Dağılım

Sürekli bir değişkenin frekans dağılımını gösterir.

Parametreler: x (veri), bins (çubuk sayısı), alpha (şeffaflık), edgecolor, density

```
# Histogram  
import numpy as np  
veri = np.random.normal(170, 10, 500)  
plt.hist(veri,  
bins=25, color='teal', alpha=0.7,  
edgecolor='white')
```

plt.scatter() — İlişki Analizi

İki sürekli değişken arasındaki ilişkiyi gösterir. Noktaların dağılımı korelasyonu ortaya koyar.

- Pozitif korelasyon: noktalar sağ üste doğru
- Negatif korelasyon: noktalar sağ alta doğru
- s parametresi: nokta büyüklüğü (3. değişken)

```
# Scatter Plotplt.scatter(df['Gelir'],
df['Harcama'],          c=df['Yaş'],
cmap='viridis',        s=50,
alpha=0.6)plt.colorbar(label='Yaş')
```

plt.pie() — Oran Gösterimi

Bütünün parçalarını yüzdelik dilimler halinde gösterir. 5'ten fazla kategori varsa okunabilirlik düşer.

- explode: dilimi dışarı çıkar
- autopct: yüzde göster ('%1.1f%%')
- startangle: başlangıç açısı

```
# Pie Chartlabels =
['Python', 'R', 'SQL', 'Julia']sizes = [55, 20, 18,
7]plt.pie(sizes, labels=labels,
autopct='%1.1f%%',      startangle=90)
```

Figure ve Axes Yönetimi

Tek bir figure içinde birden fazla grafik oluşturmak için subplots kullanılır.

Avantajları:

- Farklı perspektifleri yan yana karşılaştırma
- Raporlarda alan tasarrufu
- Tutarlı eksen ve ölçek kontrolü

Önemli parametreler:

- nrows, ncols — satır ve sütun sayısı
- figsize — (genişlik, yükseklik) tuple
- sharex, sharey — ortak eksen paylaşımı
- tight_layout() — otomatik boşluk ayarı

```
# 2x2 çoklu grafik ızgarasıfig, axes = plt.subplots(2, 2,
figsize=(12, 8))# Sol üst: Çizgi
grafiğiaxes[0,0].plot(x, y, 'b-
o')axes[0,0].set_title('Trend')# Sağ üst: Bar
grafiğiaxes[0,1].bar(kategoriler, degerler)# Sol alt:
Histogramaxes[1,0].hist(veri, bins=20)# Sağ alt:
Scatteraxes[1,1].scatter(x, y,
c='teal')plt.tight_layout()plt.savefig('coklu_grafik.png'
, dpi=150)
```

Parametre	Açıklama	Örnek
color / c	Grafik rengi (isim, hex, RGB)	color='teal', c='#2B7A78'
linewidth / lw	Çizgi kalınlığı	lw=2.5
marker	Veri noktası işareti	marker='o', 's', '^', 'D'
alpha	Şeffaflık (0-1)	alpha=0.7
fontsize	Yazı boyutu (başlık, etiket)	fontsize=14
grid	Izgara çizgileri	plt.grid(True, alpha=0.3)
legend	Açıklama kutusu	plt.legend(loc='upper left')
annotate	Grafik üzerinde metin	plt.annotate('Max', xy=(3,32))

İpucu: plt.style.available ile tüm hazır stilleri listeleyebilirsiniz.

Matplotlib

- Düşük seviyeli kontrol
- Her detayı elle ayarlama
- Daha fazla kod satırı
- Esnek ama karmaşık
- Varsayılan stiller sade
- Pandas entegrasyonu sınırlı

Seaborn

- Yüksek seviyeli API
- Güzel varsayılan stiller
- Daha az kod, daha çok iş
- İstatistiksel grafikler hazır
- Pandas DataFrame ile doğal uyum
- Matplotlib üzerine kurulu

```
pip install seaborn # Kurulum
import seaborn as sns # Tema ayarla
sns.set_theme(style='whitegrid')
```

Kategorik Veri Görselleştirme

countplot(): Kategori frekanslarını otomatik sayar ve gösterir. barplot(): Sayısal bir değişkenin kategoriye göre ortalamasını (veya başka istatistiğini) gösterir.

```
# Frekans grafiği
sns.countplot(data=df,
              x='Departman', hue='Cinsiyet',
              palette='Set2')plt.xticks(rotation=45)
```

```
# Ortalama maaş (departmana göre)
sns.barplot(data=df, x='Departman',
            y='Maas', estimator='mean', ci=95, # güven
            aralığı palette='coolwarm')
```

sns.boxplot() — Aykırı Değer Tespiti

5 sayı özeti gösterir:

- Minimum (Q1 - 1.5xIQR)
- Q1 (1. çeyreklik — %25)
- Medyan (Q2 — %50)
- Q3 (3. çeyreklik — %75)
- Maksimum (Q3 + 1.5xIQR)
- Dışındaki noktalar → Aykırı değer (outlier)

```
sns.boxplot(data=df, x='Kategori',  
            y='Fiyat', hue='Bölge',  
            palette='pastel',  
            flierprops={'marker':'D'})
```

sns.violinplot() — Dağılım Şekli

Boxplot + KDE (yoğunluk tahmini) birleşimi.

Avantajları:

- Dağılımın bimodal olup olmadığını gösterir
- Veri yoğunluğunu genişlikle ifade eder
- inner='box' ile boxplot'u da içerir
- split=True ile iki grubu karşılaştırır

```
sns.violinplot(data=df, x='Kategori',  
               y='Fiyat', inner='box',  
               palette='muted')
```

Korelasyon Matrisi Nedir?

Her değişken çiftinin doğrusal ilişki gücünü -1 ile +1 arasında ölçer.

Yorumlama:

- +1.0 → Mükemmel pozitif ilişki
- +0.7 ~ +1.0 → Güçlü pozitif
- +0.3 ~ +0.7 → Orta pozitif
- -0.3 ~ +0.3 → Zayıf / yok
- -0.7 ~ -1.0 → Güçlü negatif

Renk haritaları (cmap):

- 'coolwarm' — mavi-kırmızı (en yaygın)
- 'RdYlGn' — kırmızı-yeşil
- 'viridis' — evrensel erişilebilir

```
# Korelasyon ısı haritasıcorr =  
df.select_dtypes('number').corr()  
plt.figure(figsize=(10, 8))sns.heatmap(  
corr, annot=True, # değerleri yaz  
    fmt='.2f', # 2 ondalık  
    cmap='coolwarm', # renk haritası  
    center=0, # sıfır merkez  
    linewidths=0.5,  
    square=True)plt.title('Değişkenler Arası  
Korelasyon')
```

Pairplot Nedir?

Tüm sayısal değişken çiftlerini tek seferde görselleştirir.

- Köşegen: Her değişkenin kendi dağılımı (histogram veya KDE)
- Üst/Alt üçgen: İkili scatter plotlar
- hue parametresi ile gruplara renk atanır
- Büyük veri setlerinde yavaş olabilir — sütun seçimi yapın

```
# Iris veri seti ile pairplotiris =  
sns.load_dataset('iris')sns.pairplot(    ir  
is,    hue='species',    diag_kind='kde',  
    palette='husl',    plot_kws={'alpha':  
0.6})
```

Performans İpucu

Çok sütunlu veri setlerinde vars=[...] ile sadece ilgilendiğiniz sütunları seçin. corner=True ile sadece alt üçgeni çizin.

Alternatif: sns.jointplot()

İki değişken arasındaki ilişkiyi scatter + kenar histogramları ile gösterir. kind='hex', 'kde', 'reg' seçenekleri.

bar_chart_race Kütüphanesi

Zaman serisi verilerini animasyonlu yarışan çubuk grafiğine dönüştürür.

- MP4, GIF veya HTML çıktısı
- Pandas DataFrame doğrudan kullanılır
- COVID-19, nüfus, GSYİH gibi veriler için ideal

Kullanım Alanları

- COVID-19 vaka sayıları (ülke bazlı)
- Ülkelerin GSYİH değişimi (1960-2024)
- YouTube en çok izlenen kanallar
- Programlama dilleri popülaritesi
- Şehir nüfus değişimleri

```
import bar_chart_race as bcrimport pandas as pd# COVID-19 veri seti (satırlar: tarih, sütunlar: ülke)df = pd.read_csv('covid_worldwide.csv', index_col='date', parse_dates=True)bcr.bar_chart_race(df=df, filename='covid_race.mp4', nBars=10, title='COVID-19 Toplam Vaka Sayıları', period_length=500, figsize=(8, 5), cmap='dark12', bar_size=0.8)
```

pytrends — Google Arama Trendlerini Python'a Getirin

Google Trends verilerini programatik olarak çekmenizi sağlar. Anahtar kelime popülaritesi, bölgesel ilgi, ilgili sorgular ve yükselen trendleri analiz edebilirsiniz. Veri bilimi projelerinde trend keşfi ve mevsimsel analiz için güçlü bir araçtır.

```
from pytrends.request import TrendReq
pytrends = TrendReq(hl='tr-TR', tz=180)
# Anahtar kelime karşılaştırmak
kw = ['yapay zeka', 'veri bilimi', 'makine öğrenmesi']
pytrends.build_payload(kw, timeframe='2020-01-01 2024-12-31', geo='TR')
df = pytrends.interest_over_time()
```

Kullanılabilir Veriler

- interest_over_time()
- interest_by_region()
- related_queries()

Görselleştirme

- df.plot() ile trend çizgisi
- sns.heatmap() ile bölgesel ısı haritası

Tek Satırda Tam EDA Raporu

Sweetviz, veri setinizi otomatik olarak analiz eder ve interaktif bir HTML raporu üretir.

Rapor içeriği:

- Her sütun için dağılım grafikleri
- Eksik değer yüzdeleri
- Korelasyon matrisi
- Hedef değişken analizi
- İki veri seti karşılaştırması (train vs test)

Avantaj: Saatlerce sürecektir EDA'yı saniyede bitirir. Dezavantaj: Büyük veri setlerinde RAM yoğun.

```
import sweetviz as sv
import pandas as pd
df = pd.read_csv('veri.csv')
# Tek satır EDA raporu
rapor = sv.analyze(df,
target_feat='Fiyat')
rapor.show_html('eda_rapor.html')
```

```
# Train vs Test karşılaştırma raporu
rapor = sv.compare([train, 'Eğitim'], [test, 'Test'])
```

Kurulum: `pip install sweetviz` — Minimum Python 3.7, Pandas >= 1.0 gerektirir.

ProfileReport ile Derin Analiz

Eski adıyla pandas-profiling. Sweetviz'den daha detaylı rapor üretir.

Ekstra özellikler:

- Değişken etkileşim grafikleri
- Duplikasyon analizi
- Metin ve kategorik sütun detayları
- Jupyter Notebook içinde gösterim

```
from ydata_profiling import ProfileReport
profil = ProfileReport(df, title='Veri Seti Profili',
explorative=True)
profil.to_file('profil_rapor.html')
```

Sweetviz vs YData-Profiling

Özellik	Sweetviz	YData-Profiling
Kurulum	pip install sweetviz	pip install ydata-profiling
Hız	Hızlı	Daha yavaş (detaylı)
Rapor Boyutu	Kompakt	Kapsamlı
Karşılaştırma	compare() ile kolay	compare() mevcut
Jupyter	show_notebook()	to_notebook_iframe()
Duplikasyon	Yok	Var
Etkileşim	Sınırlı	Detaylı

1

NB1: Matplotlib & Seaborn

Temel grafik türleri, özelleştirme, stil ayarları. 15+ farklı grafik uygulaması.

[matplotlib_seaborn.ipynb](#)

2

NB2: COVID Yarışan Çubuk

bar_chart_race ile COVID-19 verilerini animasyonlu görselleştirme.

[covid_yarisan.ipynb](#)

3

NB3: Supermarket EDA

Gerçek süpermarket veri seti üzerinde Sweetviz ve YData-Profiling uygulaması.

[supermarket_eda.ipynb](#)

4

NB4: Google Trends

pytrends ile Türkiye'deki AI trendlerini çekme ve görselleştirme.

[google_trends.ipynb](#)

Haftalık Ödev — 5 Grafik Türü

- 1 Çizgi grafiği: Hava sıcaklığı trendi (7 günlük)
- 2 Bar grafiği: Departman bazlı ortalama maaş karşılaştırması
- 3 Histogram + Boxplot: Müşteri yaş dağılımı analizi
- 4 Heatmap: Korelasyon matrisi ve önemli ilişkiler raporu
- 5 Bar Chart Race: Türkiye illeri nüfus değişimi (1990–2023)

Teslim: Notebook (.ipynb) + PDF rapor

Faydalı Kaynaklar

Matplotlib Gallery
matplotlib.org/gallery

Seaborn Tutorial
seaborn.pydata.org/tutorial

Python Graph Gallery
python-graph-gallery.com

bar_chart_race Docs
github.com/dexplo/bar_chart_race

Sweetviz Docs
github.com/fbdesignpro/sweetviz

YData-Profiling Docs
docs.profiling.ydata.ai

Hafta 3 — Öğrendiklerimiz

1 Matplotlib ile temel grafik türlerini (line, bar, hist, scatter, pie) oluşturmayı öğrendik

2 Seaborn ile istatistiksel görselleştirme: boxplot, heatmap, pairplot

3 bar_chart_race ile animasyonlu, etkileyici veri hikayesi anlatımı

4 Sweetviz ve YData-Profiling ile otomatik EDA raporu üretimi

5 Doğru grafik seçimi ve data storytelling ilkeleri

“Verinin en güçlü hikayesi, doğru grafikle anlatılandır.”