

BSM622 VERİ ANALİTİĞİ VE İSTATİSTİK

2. Hafta: Keşifçi Veri Analizi (EDA)

Doç. Dr. Onur Polat

Hacettepe Üniversitesi • Bilişim Enstitüsü

Tezsiz Yüksek Lisans • Güz 2026, Ders 2 / 14

Bu Hafta Neler Öğreneceksiniz?

Kaynak: Bruce, P., Bruce, A. & Gedeck, P. (2026). AI-Assisted Statistics for Data Scientists, 3. Baskı, Bölüm 1.

Konum Ölçüleri

- Aritmetik ortalama ve sınırlılıkları
- Kırpılmış ortalama (trimmed mean)
- Medyan ve dayanıklı tahminler
- Ağırlıklı ortalama ve medyan

Yayılım Ölçüleri

- Standart sapma ve varyans
- Ortalama mutlak sapma (MAD)
- Kartiller arası açıklık (IQR)
- Persantiller ve sıra istatistikleri

Dağılım Keşfi

- Frekans tabloları ve histogramlar
- Kutu grafikleri (boxplot)
- Yoğunluk tahmini (KDE)
- Çarpıklık ve basıklık kavramları

Çok Değişkenli Analiz

- Kategorik veri: mod, sütun grafik
- Korelasyon ve saçılım grafikleri
- Hexbin ve kontur grafikleri
- Keman grafikleri ve FacetGrid



BÖLÜM I

Konum Ölçüleri (Estimates of Location)

Neden Konum Tahminine İhtiyaç Duyarız?

Binlerce farklı değer içeren bir değişkeni tek bir sayıyla özetlemek istiyoruz:
bir "tipik değer" — verinin nerede yoğunlaştığını gösteren bir merkezi eğilim tahmini.

Ölçü	Formül	Aykırı Değerlere Duyarlılık	Kullanım Alanı
Aritmetik Ortalama	$\bar{x} = \sum x_i / n$	Yüksek	Simetrik dağılımlarda en verimli
Kırpılmış Ortalama	Uç %p çıkarılıp ortalama	Orta	Hafif çarpık verilerde uzlaşma
Medyan	Sıralı verinin ortası	Düşük (kırılma noktası %50)	Çarpık dağılımlarda güvenilir
Ağırlıklı Ortalama	$\bar{x}_w = \sum w_i x_i / \sum w_i$	Yüksek	Heterojen gözlem önemleri
Ağırlıklı Medyan	Ağırlık toplamalarının ortası	Düşük	Ağırlıklı + dayanıklı

Aritmetik Ortalama (Mean)

En yaygın merkezi eğilim tahmini. Tüm değerlerin toplamının gözlem sayısına bölümüdür.

Matematiksel Tanım

$$\bar{x} = (1/n) \times \sum_{i=1}^n x_i$$

Örnek: {3, 5, 1, 2} $\rightarrow (3+5+1+2)/4 = 2.75$

Kritik Özellik

Ortalama TÜM gözlemleri kullanır.

Bu, istatistiksel verimlilik sağlar ancak aykırı değerlere karşı savunmasızdır.

Bill Gates bir mahalleye taşınırsa ortalama gelir gerçekçi olmaktan çıkar.

Python Uygulaması:

```
import pandas as pd
from scipy.stats import trim_mean

state = pd.read_csv('state.csv')
print(f"Ortalama: {state['Population'].mean():,.0f}")      # 6,162,876
print(f"Medyan: {state['Population'].median():,.0f}")     # 4,436,370
```

Kırpılmış Ortalama (Trimmed Mean)

Sıralı verinin her iki ucundan sabit bir yüzdeyi çıkardıktan sonra kalan değerlerin ortalamasıdır.

Formül ve Mantık

$$\bar{x}_t = \sum_{i=p+1}^{n-p} x_{(i)} / (n - 2p)$$

p: Her uçtan çıkarılan gözlem sayısı

- %10 kırpma: En küçük ve en büyük %10'u çıkar, kalanın ortalamasını al
- Uluslararası dalış puanlamasında 5 hakemden en yüksek ve en düşük puan çıkarılır — aynı mantık

Neden Tercih Edilir?

Ortalama ve medyan arasında bir uzlaşma noktası sunar:

- ✓ Aykırı değerlere karşı dayanıklı
- ✓ Medyandan daha fazla veri kullanır
- ✓ İstatistiksel olarak daha verimli

Örnek (Eyalet nüfusları):

Ortalama: 6,162,876

Kırpılmış: 4,783,697 (%10)

Medyan: 4,436,370

```
from scipy.stats import trim_mean
print(f"Kırpılmış (%10): {trim_mean(state['Population'], 0.1):,.0f}") # 4,783,697
print(f"Kırpılmış (%20): {trim_mean(state['Population'], 0.2):,.0f}") # 4,459,825
```

Medyan (Median) — Dayanıklı Konum Tahmini

Sıralı verinin tam ortasındaki değer. Çift sayıda gözlem varsa, ortadaki iki değer ortalaması.

Kırılma Noktası: %50

Medyanın en güçlü özelliği:
verideki gözlemlerin %50'si değişse
bile medyan etkilenmez.

Bu, "kırılma noktası" (breakdown point)

Aykırı Değerler ve Anomali Tespiti

Aykırı değer (outlier): Diğer verilerden
çok farklı olan gözlem.

- Veri hatası olabilir (birim karışıklığı,
sensör arızası)

Metodolojik Not: Ortalama > Kırılmış Ortalama > Medyan ise dağılım sağa çarpıktır.

Bu ilişki, konum tahminlerinin dağılım şekli hakkında dolaylı bilgi verdiğini gösterir.

Hangi ölçünün "doğru" olduğu, araştırma sorusuna bağlıdır — tek doğru yoktur.

aykırı değerlerdir, diğer veri normal
tanımını oluşturur.

Ağırlıklı Ortalama ve Ağırlıklı Medyan

Bazı gözlemler diğerlerinden daha önemli olduğunda ağırlıklandırma kullanılır.

İki Temel Motivasyon

1. Değişkenlik farkı: Bazı ölçümler daha güvenilir — düşük doğruluklu sensöre düşük ağırlık.
2. Temsiliyet düzeltmesi: Örneklem kitleyi yansıtmıyorsa, eksik temsil edilen gruplara yüksek ağırlık vererek düzeltme yapılır.

Eyalet Cinayet Oranı Örneği

Basit ortalama: Wyoming (580K) ile California (37M) eşit ağırlıkta.

Ağırlıklı ortalama: Nüfusa göre →
bir ABD vatandaşının karşılaşacağı
tipik cinayet oranını tahmin eder.

Ağırlıklı ort: 4.45

Ağırlıklı med: 4.40

```
weighted_mean = np.average(state['Murder.Rate'], weights=state['Population'])  
weighted_median = wquantiles.median(state['Murder.Rate'], weights=state['Population'])  
print(f"Ağırlıklı Ortalama: {weighted_mean:.2f}") # 4.45  
print(f"Ağırlıklı Medyan: {weighted_median:.2f}") # 4.40
```

Temel Çıkarımlar — Konum Tahminleri

- Temel konum ölçüsü ortalamadır, ancak aykırı değerlere (outlier) karşı duyarlıdır.
- Medyan ve kırılmış ortalama, aykırı değerlere ve olağandışı dağılımlara karşı daha dayanıklıdır (robust).

Tahmin (Estimate) vs. Metrik (Metric)

İstatistikçiler "tahmin" terimini tercih eder — eldeki veriden hesaplanan değer ile gerçek kitle parametresi arasındaki farkı vurgulamak için. Veri bilimciler ise "metrik" der — odak noktası somut iş hedefleridir. İstatistikçiler tahmin eder, veri bilimciler ölçer.

— Bruce, Bruce & Gedeck (2026), s. 14



BÖLÜM II

Yayılm Ölçüleri (Estimates of Variability)

Değişkenliği Neden Ölçeriz?

"İstatistiğin kalbinde değişkenlik yatar: onu ölçmek, azaltmak, rassal olanı gerçek olandan ayırt etmek, gerçek değişkenliğin kaynaklarını belirlemek ve onun varlığında karar vermek."

— Bruce, Bruce & Gedeck (2026)

Ölçü	Tanım	Kırılma Noktası	Python
Standart Sapma (s)	$s = \sqrt{[\sum(x_i - \bar{x})^2 / (n-1)]}$	%0 — tek aykırı değer yeterli	<code>df['col'].std()</code>
Varyans (s^2)	$s^2 = \sum(x_i - \bar{x})^2 / (n-1)$	%0	<code>df['col'].var()</code>
Ort. Mutlak Sapma	$\sum x_i - \bar{x} / n$	~%0	<code>np.mean(np.abs(x - x.mean()))</code>
MAD (Medyandan)	Medyan($ x_i - m $)	%50 — maksimum dayanıklı	<code>robust.scale.mad(x)</code>
IQR	$Q_3 - Q_1$	%25	<code>df['col'].quantile(.75) - .quantile(.25)</code>
Aralık (Range)	max - min	%0	<code>df['col'].max() - df['col'].min()</code>

Standart Sapma ve Varyans

Neden Kare Alınır?

Sapmaların ($x_i - \bar{x}$) toplamı sıfırdır
— negatif ve pozitif sapmalar birbirini götürür.

İki çözüm:

1. Mutlak değer $\rightarrow$ MAD
2. Kare alma $\rightarrow$ Varyans

Kare alma matematiksel olarak daha kullanışlıdır (türevlenebilir, toplama kuralları). Bu yüzden istatistik teorisi varyans üzerine inşa edilmiştir.

Std sapma = $\sqrt{\text{Varyans}}$
(orijinal birime döndürür)

n vs. n-1: Serbestlik Derecesi

Örneklem varyansında n-1 kullanılır.
Neden?

Örneklem ortalaması ($\bar{x}$) hesaplandığında bir kısıt oluşur: son gözlem diğerlerinden türetilir. Bu, serbestlik derecesini n-1'e düşürür.

n kullanmak $\rightarrow$ yanlış (biased) tahmin
n-1 kullanmak $\rightarrow$ yanlış (unbiased) tahmin

Pratikte: n yeterince büyükse fark ihmal edilebilir. Ama ilkeyi bilin.

Python: `df.std()` $\rightarrow$ n-1 kullanır
`df.std(ddof=0)` $\rightarrow$ n kullanır

Medyan Mutlak Sapma (MAD) — Dayanıklı Yayılım Tahmini

$MAD = \text{Medyan}(|x_1 - m|, |x_2 - m|, \dots, |x_n - m|)$ burada $m = \text{Medyan}(x)$

Neden MAD?

- Medyan gibi, aykırı değerlerden etkilenmez (kırılma noktası %50)
- Standart sapmadan çok daha dayanıklıdır
- Normal dağılım varsayımı gerektirmez

Ölçek faktörü:

$MAD \times 1.4826 \approx \text{Standart Sapma}$

(Normal dağılım altında)

Eyalet Nüfusu Karşılaştırması

Standart Sapma: 6,848,235

IQR: 4,847,308

MAD: 3,849,870

Std sapma, MAD'ın neredeyse 2 katı!

Bu fark, dağılımdaki uç değerlerin (CA, TX, NY) std sapmayı şişirdiğini doğrular.

Sonuç: Çarpık verilerde MAD ve IQR, std sapmadan daha güvenilir yayılım tahminleri sunar.

```
from statsmodels import robust
print(f"Std: {state['Population'].std():,.0f}") # 6,848,235
print(f"IQR: {state['Population'].quantile(0.75) - state['Population'].quantile(0.25):,.0f}") # 4,847,308
print(f"MAD: {robust.scale.mad(state['Population']):,.0f}") # 3,849,870
```

Persantiller ve Sıra İstatistikleri

P. persantil: Verinin %P'sinin altında kaldığı değer. Medyan = 50. persantil.

Persantil	Diğer Adı	Cinayet Oranı	Yorumu
%5	Alt uç	1.60	Eyaletlerin %5'i bu değerin altında
%25	1. Kartil (Q_1)	2.42	Alt çeyrek
%50	Medyan	4.00	Tam orta
%75	3. Kartil (Q_3)	5.55	Üst çeyrek
%95	Üst uç	6.51	Eyaletlerin %95'i bu değerin altında

```
state[['Murder.Rate']].quantile([0.05, 0.25, 0.50, 0.75, 0.95])
```

Temel Çıkarımlar — Yayılım Tahminleri

- Varyans ve standart sapma en yaygın kullanılan yayılım ölçüleridir.
- Her ikisi de aykırı değerlere duyarlıdır (kare alma etkisi).
- Daha dayanıklı alternatifler: MAD, IQR ve kırpılmış standart sapma.
- Eşit boyutlu aralıklar (frekans tablosu): Her aralık aynı genişlikte, farklı sayıda gözlem.
- Eşit sayılı aralıklar (persantiller/kartiller): Her aralıkta aynı sayıda gözlem, farklı genişlik.

Bu ayırım, dağılımın keşfinde hangi özetleme yönteminin kullanılacağını belirler.



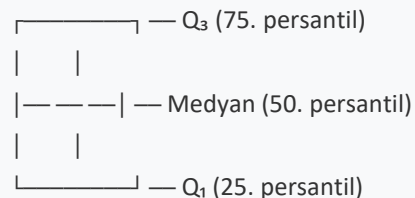
BÖLÜM III

Dağılımın Keşfi (Exploring the Data Distribution)

Kutu Grafiği (Boxplot) — Tukey'in Buluşu

Kutu Grafiğinin Anatomisi

Üst bıyık — $Q_3 + 1.5 \times IQR$ 'ye kadar
en uzak gözlem



Alt bıyık — $Q_1 - 1.5 \times IQR$ 'ye kadar
en uzak gözlem

○ ○ ○ Bıyıkların dışı:
potansiyel aykırı değerler

$IQR = Q_3 - Q_1$ (kutunun yüksekliği)

Neyi Gösterir?

Tek bir grafikte 5 bilgi:

1. Merkez (medyan çizgisi)
2. Yayılm (kutu = IQR)
3. Çarpıklık (medyanın kutu içindeki konumu)

```
ax = (state['Population'] / 1_000_000).plot.box(
    figsize=(3, 4))
ax.set_ylabel('Nüfus (milyon)')
plt.title('Eyalet Nüfus Dağılımı')
plt.show()
```

Frekans Tabloları

Değişkenin değer aralığını eşit genişlikli aralıklara bölerek her aralıktaki gözlem sayısını gösterir.

Aralık	Sayı	Eyaletler (örnek)
564K – 4.2M	24	WY, VT, ND, AK, SD, DE, MT, RI, NH...
4.2M – 7.9M	14	KY, LA, SC, AL, CO, MN, WI, MD...
7.9M – 11.6M	6	VA, NJ, NC, GA, MI, OH
11.6M – 15.2M	2	PA, IL
15.2M – 18.9M	1	FL
18.9M – 22.6M	1	NY
22.6M – 26.2M	1	TX
26.2M – 37.3M	1	CA (2 boş aralıktan sonra)

Önemli: Boş aralıkları grafikte dahil edin — boş olması da bir bilgidir. Aralık genişliği çok büyükse yapı kaybolur, çok küçükse gürültü artar.

Histogramlar

Histogram Kuralları

1. Boş aralıklar dahil edilir
2. Aralık genişlikleri eşittir
3. Aralık sayısı (bin) kullanıcı seçer
4. Sütunlar bitişiktir (sütun grafiğinden farklı!)

Histogram $\neq$ Sütun grafiği

- Histogram: sürekli değişken, bitişik sütunlar, alan $\propto$ frekans
- Sütun grafiği: kategorik değişken, ayırık sütunlar, yükseklik $\propto$ frekans

```
ax = (state['Population'] / 1_000_000).plot.hist(figsize=(4, 4))
ax.set_xlabel('Nüfus (milyon)')
plt.title('Eyalet Nüfus Histogramı')
plt.tight_layout()
plt.show()
```

Aralık Sayısı Seçimi (bins)

Çok az aralık $\rightarrow$ yapıyı gizler

Çok fazla aralık $\rightarrow$ gürültü yakalar

Pratik kurallar:

- Sturges: $k = 1 + 3.322 \times \log_{10}(n)$
- Freedman-Diaconis: $h = 2 \times \text{IQR} / n^{(1/3)}$
- Deneme-yanılma: 15-50 arası başlayın

Python'da:

```
df['col'].plot.hist(bins=30)
plt.hist(x, bins='auto') # otomatik
```

Yoğunluk Grafikleri ve KDE

Yoğunluk grafiği, histogramın düzgünleştirilmiş versiyonudur. Çekirdek Yoğunluk Tahmini (KDE) ile hesaplanır.

KDE Nasıl Çalışır?

Her veri noktasına bir "çekirdek" (kernel) fonksiyonu yerleştirilir (genellikle Gaussian). Tüm çekirdeklerin toplamı düzgün bir yoğunluk eğrisi verir.

Bant genişliği (bandwidth):

- Dar bant → gürültüyü yakalar
- Geniş bant → yapıyı gizler

Toplam alan = 1 olacak şekilde normalize edilir.

Histogram vs. KDE

Histogram:

- ✓ Sezgisel, kolay yorumlanır
- X Kesikli, aralık seçimine bağlı

KDE:

- ✓ Sürekli, pürüzsüz
- ✓ Dağılım şekli daha net
- X Bant genişliği seçimine bağlı
- X Küçük örneklerde yanıltıcı olabilir

En iyi uygulama:

Histogram + KDE üst üste çizin.

```
ax = state['Murder.Rate'].plot.hist(density=True, xlim=[0, 12], bins=range(1, 12))
state['Murder.Rate'].plot.density(ax=ax)
ax.set_xlabel('Cinayet Oranı (100,000 kişi başına)')
```

İstatistiksel Momentler — Çarpıklık ve Basıklık

Moment	Ölçtüğü Şey	Normal Dağılım	Yorumu
1. Moment: Konum	Merkezi eğilim	$\mu = 0$ (standart)	Verinin nerede toplandığı
2. Moment: Yayılım	Dağınıklık	$\sigma^2 = 1$ (standart)	Verinin ne kadar dağıldığı
3. Moment: Çarpıklık	Asimetri	$= 0$ (simetrik)	> 0 sağa çarpık, < 0 sola çarpık
4. Moment: Basıklık	Kuyruk ağırlığı	$= 3$ (mesokurtik)	> 3 ağır kuyruk (leptokurtik)

Pratikte: Çarpıklık ve basıklık genellikle metrik olarak değil, görsel olarak keşfedilir.

Histogramlar ve kutu grafikleri bu bilgiyi sezgisel olarak sunar — sayısal değerler nadiren raporlanır.

```
log_ret = np.log(prices / prices.shift(1)).dropna()
print(f"Çarpıklık: {log_ret.skew():.3f}") # scipy.stats.skew alternatif
print(f"Basıklık: {log_ret.kurtosis():.3f}") # Fazla basıklık (excess kurtosis)
```



BÖLÜM IV

Kategorik Veri ve Korelasyon Analizi

İkili ve Kategorik Veri Keşfi

Temel Kavramlar

Mod (Mode): En sık görülen kategori.

Beklenen değer: Kategorik sonuçların ağırlıklı ortalaması.

Oranlar ve yüzdeler: Kategorik verinin temel özetleme aracı.

Sütun grafiği (Bar chart): Kategoriler arasındaki frekans karşılaştırması.

Histogramdan farkı: sütunlar ayrıktır.

DFW Havalimanı — Gecikme Nedenleri

Carrier: 23.0%

ATC: 30.4%

Weather: 21.5%

Security: 0.1%

Inbound: 42.1%

Pareto ilkesi: Inbound (%42) ve ATC (%30) toplam gecikmelerin büyük çoğunluğunu oluşturur. İyileştirme çabası buraya yoğunlaştırılmalıdır.

```
dfw = pd.read_csv(f'{DATA_URL}/dfw_airline.csv')
print((100 * dfw / dfw.to_numpy().sum()).round(1))
```

```
ax = dfw.transpose().plot.bar(figsize=(5, 4), legend=False)
ax.set_xlabel('Gecikme Nedeni')
ax.set_ylabel('Sayı')
```

Korelasyon Katsayısı (Pearson r)

İki sayısal değişken arasındaki doğrusal ilişkinin yönünü ve gücünü ölçer.

Pearson r	İlişki Gücü	Yorumu
$r = 1$	Tam pozitif	Biri artarken diğeri de orantılı artar
$0.7 < r < 1$	Güçlü pozitif	Belirgin birlikte hareket
$0.3 < r < 0.7$	Orta düzey	Eğilim var ama dağınık
$-0.3 < r < 0.3$	Zayıf / Yok	Doğrusal ilişki kanıtı zayıf
$r = -1$	Tam negatif	Biri artarken diğeri orantılı azalır

KRİTİK UYARI: Korelasyon $\neq$ Nedensellik. İki değişken ortak bir karıştırıcı faktör, ters nedensellik veya saf tesadüf nedeniyle güçlü korelasyon gösterebilir. Nedensellik için kontrollü deney veya nedensel çıkarım yöntemleri gerekir.

Korelasyon Matrisi ve Isı Haritası

```
# ETF getirileri arasındaki korelasyon ısı haritası
etf_symbols = sp500_sym[sp500_sym['sector'] == 'etf']['symbol']
etfs = sp500_px.loc[sp500_px.index > '2012-07-01', etf_symbols]

plt.figure(figsize=(8, 6))
sns.heatmap(etfs.corr(), vmin=-1, vmax=1,
            cmap=sns.diverging_palette(20, 220, as_cmap=True),
            annot=False, square=True)
plt.title('ETF Getirileri Arasındaki Korelasyon Matrisi')
plt.tight_layout()
plt.show()
```

Isı Haritası Okuma Rehberi

- Koyu kırmızı → güçlü pozitif korelasyon (birlikte hareket)
- Koyu mavi → güçlü negatif korelasyon (ters yönde hareket)
- Beyaz/açık renk → zayıf veya yok
- Diyagonal her zaman 1.0'dır (kendisiyle korelasyon)
- Matris simetriktir: üst ve alt üçgen aynı bilgiyi taşır

Saçılım Grafikleri (Scatterplots)

Neyi Gösterir?

- İki değişken arasındaki ilişkinin şeklini, yönünü ve gücünü
- Doğrusal olmayan ilişkileri (korelasyon katsayısı yakalamaz!)
- Kümelenmeleri ve aykırı değerleri
- Heterojen alt grupları

Korelasyon katsayısı + saçılım grafiği birlikte yorumlanmalıdır. $r = 0$ olsa bile güçlü bir ilişki olabilir (ör. parabolik)

AT&T vs. Verizon Getirileri

Aynı sektördeki hisseler benzer makro faktörlere maruz kalır → pozitif korelasyon beklenir.

Saçılım grafiğinde:

- Eğilim çizgisi etrafındaki dağınıklık idiosinkratik (firmaya özgü) risk
- Eğilimin yönü sistematik risk

Bu ayrım portföy teorisinin temelini oluşturur.

```
telecom = sp500_px.loc[sp500_px.index >= '2012-07-01', :]  
ax = telecom.plot.scatter(x='T', y='VZ', figsize=(5, 5),  
                           marker='$\\u25EF$', alpha=0.5)  
ax.set_xlabel('ATT (T)'); ax.set_ylabel('Verizon (VZ)')  
ax.axhline(0, color='grey', lw=1); ax.axvline(0, color='grey', lw=1)
```



BÖLÜM V

Çok Değişkenli Görselleştirme

Altıgen Gruplama (Hexagonal Binning)

432,000+ gözlemle standart saçılım grafiği okunaksız hale gelir (overplotting). Hexbin, yoğunluğu renk kodlamasıyla çözer.

Neden Hexbin?

- Büyük veri setlerinde overplotting'i önler
- Altıgen hücreler kare hücrelerden daha az kenar paylaşır → daha pürüzsüz görünüm
- Renk yoğunluğu = gözlem yoğunluğu
- Kolay yorumlanır: koyu = kalabalık

King County Emlak Verisi

kc_tax veri seti: 432,000+ gözlem
SqFtTotLiving vs. TaxAssessedValue

Filtre sonrası (makul aralıklar):

SqFtTotLiving: 100–3500

TaxAssessedValue: < 750,000

Bu filtreleme EDA'nın önemli bir parçasıdır — aşırı değerler yapıyı gizler.

```
kc_tax0 = kc_tax.loc[(kc_tax.TaxAssessedValue < 750_000) &
                    (kc_tax.SqFtTotLiving > 100) &
                    (kc_tax.SqFtTotLiving < 3500), :]

ax = kc_tax0.plot.hexbin(x='SqFtTotLiving', y='TaxAssessedValue',
                        gridsize=30, sharex=False, figsize=(6, 4))
```

2B Yoğunluk Tahmini (Contour Plots)

2B KDE

1B'deki KDE'nin iki boyutlu versiyonu.

Her veri noktasına 2B Gaussian çekirdek yerleştirilir; toplam yüzey düzgün bir yoğunluk tahmini verir.

Kontur çizgileri = eşit yoğunluk eğrileri (haritadaki yükseklik eğrileri gibi).

Büyük veri setlerinde hesaplama yükü için örnekleme (sample) kullanılır.

```
# 10,000 örneklem ile 2B KDE
ax = sns.kdeplot(
    data=kc_tax0.sample(10_000,
                        random_state=42),
    x='SqFtTotLiving',
    y='TaxAssessedValue'
)
ax.set_xlabel('Yaşam Alanı (sq ft)')
ax.set_ylabel('Vergi Değeri ($)')
plt.title('2B Yoğunluk Tahmini')
```

Çapraz Tablolar (Contingency Tables)

İki kategorik değişkenin birlikte dağılımını gösterir — ki-kare testinin (Hafta 6) temelini oluşturur.

LendingClub Kredi Verileri

Kredi notu (A-G) vs. geri ödeme durumu

Bulgu: Not düştükçe temerrüt oranı

monoton artar:

A: %2.2 → G: %12.6

Bu monoton ilişki, not sisteminin geçerli bir risk sınıflandırıcısı olduğunu doğrular.

```
crosstab = lc_loans.pivot_table(
    index='grade',
    columns='status',
    aggfunc=len,
    margins=True
)
print(crosstab)

# Oransal dağılım
perc = df_ct.loc[:, 'Charged Off':'Late'].div(
    df_ct['All'], axis=0)
print(perc.round(3))
```

Kategorik × Sayısal: Kutu ve Keman Grafikleri

Kutu Grafiği (Gruplu)

Havayolu bazında gecikme dağılımı:

```
ax = airline_stats.boxplot(  
    by='airline',  
    column='pct_carrier_delay')
```

Gruplar arası farkları görmek için
yan yana kutu grafikleri idealdir.

→ ANOVA (Hafta 9) ön keşfi

Keman Grafiği (Violin Plot)

Kutu grafiğinden farkı: dağılımın
tam şeklini (KDE) gösterir.

```
sns.violinplot(  
    data=airline_stats,  
    x='airline',  
    y='pct_carrier_delay',  
    inner='quartile')
```

Bimodalite, ağır kuyruk gibi
özellikleri kutu grafiği gizler,
keman grafiği ortaya çıkarır.

FacetGrid — Koşullu Görselleştirme

Posta koduna göre çoklu hexbin grafikleri

Aynı grafik türünü farklı alt kümelere uygulayarak gruplar arası yapısal farkları ortaya çıkarır.

```
zip_codes = [98188, 98105, 98108, 98126]
kc_tax_zip = kc_tax0.loc[kc_tax0.ZipCode.isin(zip_codes), :]

def hexbin(x, y, color, **kwargs):
    cmap = sns.light_palette(color, as_cmap=True)
    plt.hexbin(x, y, gridsize=25, cmap=cmap, **kwargs)

g = sns.FacetGrid(kc_tax_zip, col='ZipCode', col_wrap=2, height=3.5)
g.map(hexbin, 'SqFtTotLiving', 'TaxAssessedValue',
      extent=[0, 3500, 0, 700_000])
g.set_axis_labels('Yaşam Alanı', 'Vergi Değeri')
g.set_titles('Posta Kodu: {col_name:.0f}')
```

Bulgu: Yaşam alanı–vergi ilişkisinin eğimi, yayılımı ve yoğunluk merkezi posta koduna göre değişir. Bu bölgesel heterojenlik, tek bir toplam grafikte görünmez.



BÖLÜM VI

Yapay Zekâ Destekli Keşif (3. Baskı)

Görselleştirme Sonuçlarını AI ile Yorumlama

Adım	Eylem	Örnek Prompt
1. Oluştur	Grafiği Python'da çizin	—
2. Gönder	AI aracına yapıştırın	"Bu grafiği bir paragrafta yorumla"
3. Derinleştir	Takip sorusu sorun	"Ana kümenin üzerindeki iki küme ne?"
4. Koşullandır	Strateji önerin	"Lokasyona göre ayrıştırsak ne görürüz?"
5. Özetle	Rapor taslağı isteyin	"Bulguları kısaca özetle"

AI ile Veri Sözlüğü Oluşturma

Geleneksel Yaklaşım

Manuel veri sözlüğü oluşturma:

1. Her sütunu tek tek inceleyin
2. Veri tipini belirleyin
3. Birim ve aralık tespit edin
4. Açıklama yazın
5. Eksik veri kodlamasını belgeleyin

Bu süreç büyük veri setlerinde saatler sürebilir ve hatalara açıktır.

AI Destekli Yaklaşım (3. Baskı)

LLM'ye DataFrame yapısını verin:

Prompt örneği:

"Bu DataFrame'in her sütunu için veri sözlüğü oluşturun: tip, birim, aralık, açıklama."

Avantajlar:

- ✓ Hızlı taslak üretimi
- ✓ Standart format
- ✓ Tutarlı terminoloji

Dezavantajlar:

- X Alan bilgisi gerektiren nüansları kaçırabilir
- X Her zaman doğrulama gerektirir

Uçtan Uca Keşifçi Analiz — 6 Adım

#	Adım	Python İşlemleri	Cevapladığı Soru
1	Veri Yükleme	<code>pd.read_csv()</code> , <code>pd.read_excel()</code>	Veri nerede ve nasıl?
2	İnceleme	<code>shape</code> , <code>dtypes</code> , <code>head()</code> , <code>describe()</code>	Veri neye benziyor?
3	Temizleme	<code>dropna()</code> , <code>fillna()</code> , <code>astype()</code>	Güvenebilir miyiz?
4	Öznitelik Mühendisliği	yeni sütunlar, <code>cut()</code> , <code>apply()</code>	Daha bilgilendirici değişkenler?
5	Özetleme	<code>groupby()</code> , <code>agg()</code> , <code>crosstab()</code>	Gruplar arası farklar?
6	Görselleştirme	<code>hist()</code> , <code>boxplot()</code> , <code>scatter()</code> , <code>heatmap()</code>	Dağılım ve ilişkiler?

Bu 6 adımlı iş akışını dersin her haftasında uygulayacaksınız. Adımlar doğrusal değildir — EDA sizi genellikle temizleme adımına geri gönderir, modelleme ek veri ihtiyacını ortaya çıkarabilir.

seaborn — İstatistiksel G rselleŐtirme K t phanesi

Fonksiyon	Ama	Kullanım
sns.histplot()	Histogram + opsiyonel KDE	sns.histplot(data=df, x='col', kde=True)
sns.boxplot()	Gruplu kutu grafiĐi	sns.boxplot(data=df, x='cat', y='num')
sns.violinplot()	Keman grafiĐi (tam daĐılım)	sns.violinplot(data=df, x='cat', y='num')
sns.scatterplot()	Saılım (renk/boyut kodlama)	sns.scatterplot(data=df, x='x', y='y', hue='cat')
sns.heatmap()	Korelasyon ısı haritası	sns.heatmap(df.corr(), annot=True)
sns.pairplot()	T�m iftler arası saılım	sns.pairplot(df[['a', 'b', 'c']])
sns.kdeplot()	1B veya 2B yoĐunluk tahmini	sns.kdeplot(data=df, x='a', y='b')
sns.FacetGrid()	KoŐullu oklu grafik	g = sns.FacetGrid(df, col='cat')

seaborn, matplotlib  zerine inŐa edilmiŐtir. İstatistiksel farkındalık ile tasarlanmıŐtır: otomatik KDE, g ven aralıkları, faceting.

Korelasyon Elips Grafiği

Okuma Rehberi

Dar elips = güçlü korelasyon

→ Sağa yatık: pozitif

→ Sola yatık: negatif

Daire = korelasyon yok

Isı haritasına kıyasla avantajı:
hem yön hem büyüklüğü geometrik
şekil ile kodlar — renk yerine
şekil algılanması daha hızlıdır.

Dezavantajı: Büyük matrislerde
okunması zorlaşır.

```
# Ellipse grafiği (detaylı kod Colab'da)
from matplotlib.collections import
    EllipseCollection
from matplotlib.colors import Normalize

m, ax = plot_corr_ellipses(
    etfs.corr(),
    figsize=(6, 5),
    cmap='bwr_r'
)
cb = plt.colorbar(m, ax=ax)
cb.set_label('Korelasyon Katsayısı')
```

Temel Çıkarımlar — Bölüm 1 Tamamı

Konum ve Yayılım

- ✓ Temel konum: ortalama; dayanıklı: medyan ve kırılmış ortalama
- ✓ Temel yayılım: standart sapma; dayanıklı: MAD ve IQR
- ✓ Hangi ölçünün "doğru" olduğu araştırma sorusuna bağlıdır
- ✓ Dayanıklı (robust) ölçüler çarpık ve aykırı değerli verilerde daha güvenilir tahminler sunar

Dağılım ve Çok Değişkenli

- ✓ Histogram: dağılım şeklinin ilk keşfi
- ✓ Kutu grafiği: 5 sayılık özet
- ✓ KDE: pürüzsüz yoğunluk tahmini
- ✓ Hexbin/kontur: büyük verilerde overplotting çözümü
- ✓ FacetGrid: gruplar arası yapısal farkların keşfi
- ✓ Korelasyon $\neq$ nedensellik
- ✓ AI yorumları doğrulama gerektirir

Uygulamalı Çalışma

Hafta 2 Colab defterini açın ve aşağıdaki adımları tamamlayın:

1. state.csv veri setini yükleyin. Murder.Rate değişkeni için ortalama, medyan ve kırılmış ortalama (%5 ve %20) hesaplayın. Sonuçları karşılaştırıp yorumlayın.
2. airline_stats veri setinde pct_carrier_delay değişkeninin her havayolu için standart sapma, IQR ve MAD değerlerini hesaplayın (groupby kullanın). Hangi havayolu en değişken gecikme performansına sahip?
3. kc_tax0 veri setinde SqFtTotLiving değişkeninin histogramını çizin (bins=50). Dağılım şeklini yorumlayın: simetrik mi, çarpık mı, unimodal mı? Üzerine KDE eğrisi ekleyin.
4. etfs veri setinden herhangi iki ETF seçin. Saçılım grafiği çizin ve Pearson korelasyonunu hesaplayın. 2-3 cümleyle yorumlayın.

Ödev 2 — Keşifçi Veri Analizi

1. tips veri setini (seaborn) yükleyin. total_bill ve tip değişkenleri için tüm konum ve yayılım ölçülerini hesaplayın. Tablo olarak raporlayın.
2. $\text{tip_orani} = \text{tip} / \text{total_bill} \times 100$ sütunu oluşturun. Histogramını çizin (bins=30, KDE dahil). Dağılım şeklini yorumlayın.
3. day sütununa göre gruplu kutu grafiği ve keman grafiği oluşturun. Hangi gün bahşış oranı daha yüksek ve daha değişken?
4. total_bill ile tip arasındaki saçılım grafiğini çizin. Pearson ve Spearman korelasyonlarını hesaplayın. Farkı yorumlayın.

Son Teslim: 3. Hafta ders saatinden önce • Teslim: Moodle Ödev modülü • Gecikme cezası: Günlük -%20 (azami 3 gün).

6. AI Görevi: Keman grafiğinizi AI'ya yorumlatın. 2 takip sorusu sorun. AI'nın yorumlarını eleştirel değerlendiren bir paragraf yazın.

Bu Haftanın Sık Yapılan Hataları

#	Hata	Çözüm
1	Aykırı değerleri incelemeden silmek	Önce neden aykırı olduğunu araştırın — veri hatası mı, gerçek gözlem mi?
2	Tek bir konum ölçüsüne güvenmek	Ortalama, medyan ve kırılmış ortalamayı birlikte raporlayın
3	Histogram aralık sayısını optimize etmemek	bins=15, 30, 50 ile deneyin; bins='auto' kullanın
4	Korelasyonu nedensellik olarak yorumlamak	Her zaman "korelasyon nedensellik değildir" uyarısı ekleyin
5	Büyük veri setinde scatter plot kullanmak	10K+ gözlemde hexbin veya 2D KDE kullanın
6	Dağılım şeklini görmezden gelmek	Her sayısal değişken için histogram + boxplot çizin

Bu Haftadan Sonra Bilmeniz Gerekenler

Kavramsal

- ✓ Ortalama, medyan, kırılmış ortalama ve ağırlıklı versiyonlarını tanımlama
- ✓ Standart sapma, MAD ve IQR'nin dayanıklılık farklarını açıklama
- ✓ Histogram, kutu grafiği ve KDE'nin hangi bilgiyi sunduğunu ayırt etme
- ✓ Korelasyon katsayısını yorumlama ve nedensellik yanılgısını tanıma
- ✓ Çarpıklık ve basıklığın ne anlama geldiğini sezgisel olarak bilme

Uygulamalı

- ✓ Python'da tüm konum ve yayılım ölçülerini hesaplama
- ✓ Histogram, KDE, boxplot, violin plot, scatter plot, heatmap oluşturma
- ✓ Frekans tablosu ve çapraz tablo oluşturma ve yorumlama
- ✓ Hexbin ve FacetGrid ile büyük veri setlerini keşfetme
- ✓ AI araçlarını görsel yorumlamada eleştirel biçimde kullanma

Hafta 3: Çıkarımsal İstatistik — Permütasyon ve p-değerleri

Hazırlık

- Bruce vd., Bölüm 2 (Data and Sampling Distributions) ve Bölüm 3 (Statistical Experiments) okuyun
- Ödev 2'yi tamamlayıp Moodle üzerinden teslim edin
- Rassal örnekleme ve istatistiksel anlamlılık kavramlarını düşünün:
"İki grup arasındaki fark gerçek mi, yoksa tesadüf mü?"

Gelecek haftanın sorusu: Bir A/B testinde gözlemlediğiniz farkın şans eseri olma olasılığını nasıl hesaplarsınız? Bu soruyu permütasyon testiyle cevaplayacağız.