

Kötü kesilmiş elmaslar neden daha pahalı? 53.940 satırlık bir veri setinin bana öğrettiği ders

Ortalamalara güvenmenin bedeli üzerine bir veri analizi hikâyesi

Veri analitiği öğrenirken en çok tekrarlanan cümlelerden biri şu: "Korelasyon nedensellik değildir." Herkes ezberler, kimse gerçekten hissetmez. Ta ki elinizdeki veri size tamamen mantıksız bir şey söyleyene kadar.

Bu yazıda `diamonds` veri setiyle yaptığım analizi anlatacağım. Amacım bir model kurmak değildi; sadece veriyi tanımaktı. Ama ilk beş dakikada karşıma çıkan sonuç o kadar saçmaydı ki, analizin geri kalanı "bu nasıl olabilir?" sorusunun peşinden gitmekle geçti.

Sonuç: elmasların en kaliteli kesime sahip olanları, en kötü kesime sahip olanlardan **daha ucuzdu**.

Veri seti

53.940 elmasın kaydı, 10 değişken. Değişkenlerin önemli olanları şunlar:

- carat** — taşın ağırlığı (0.2 – 5.01)
- cut** — kesim kalitesi: Fair < Good < Very Good < Premium < Ideal
- color** — renk: D (en iyi, renksiz) → J (en kötü, sarımsı)
- clarity** — berraklık
- price** — fiyat, dolar (326 – 18.823)
- x, y, z** — mm cinsinden boyutlar

Kesim, renk ve berraklık sıralı kategorik değişkenler. Bu detay sonradan çok önemli olacak, çünkü pandas bunları alfabetik sıralar ve grafikleriniz anlamsız çıkar. Baştan halletmek gerekiyor:

```
sira = ["Fair", "Good", "Very Good", "Premium", "Ideal"]
df["cut"] = pd.Categorical(df.cut, categories=sira, ordered=True)
```

Önce temizlik: veri "temiz" görünüyordu, değildi

`df.isnull().sum()` sıfır döndü. Eksik değer yok. Rahatlayıp geçmek üzereydim ki, boyut sütunlarına baktım:

```
(df[["x", "y", "z"]] == 0).any(axis=1).sum()    # 20
```

20 elmasın en az bir boyutu sıfır. Bunlar eksik değer olarak işaretlenmemişti çünkü teknik olarak eksik değildi — sıfırdı. Ama sıfır kalınlığında bir elmas fiziksel olarak yok. Bu, gerçek bir eksik veriyi sıfırla doldurmuş birinin izi.

Devamı geldi:

- 146 tam kopya satır** — `df.duplicated().sum()`
- y'nin maksimumu 58.9 mm, z'nin maksimumu 31.8 mm.** Ortalama bir elmasın boyutu 5-6 mm civarında. 58.9 mm'lik bir taş bir elmas değil, bir yumruk büyüklüğünde bir şey olurdu. Bunlar veri giriş hatası.

Kopyaları ve fiziksel olarak imkânsız satırları çıkardım: **53.940 → 53.772 satır**, toplam 168 kayıp. Veri setinin %0.3'ü. Küçük bir oran ama bulguları etkiliyor, çünkü çıkardıklarımın çoğu uç değerlerdi.

Buradaki ders şu: `isnull()` temiz döndü diye veri temiz değil. **Alan bilgisiyle bakmadan veri kalitesi kontrolü yapılmış sayılmaz.** Bir elmasın 0 mm olamayacağını pandas bilmiyor, sizin bilmeniz gerekiyor.

İlk bulgu ve ilk şok

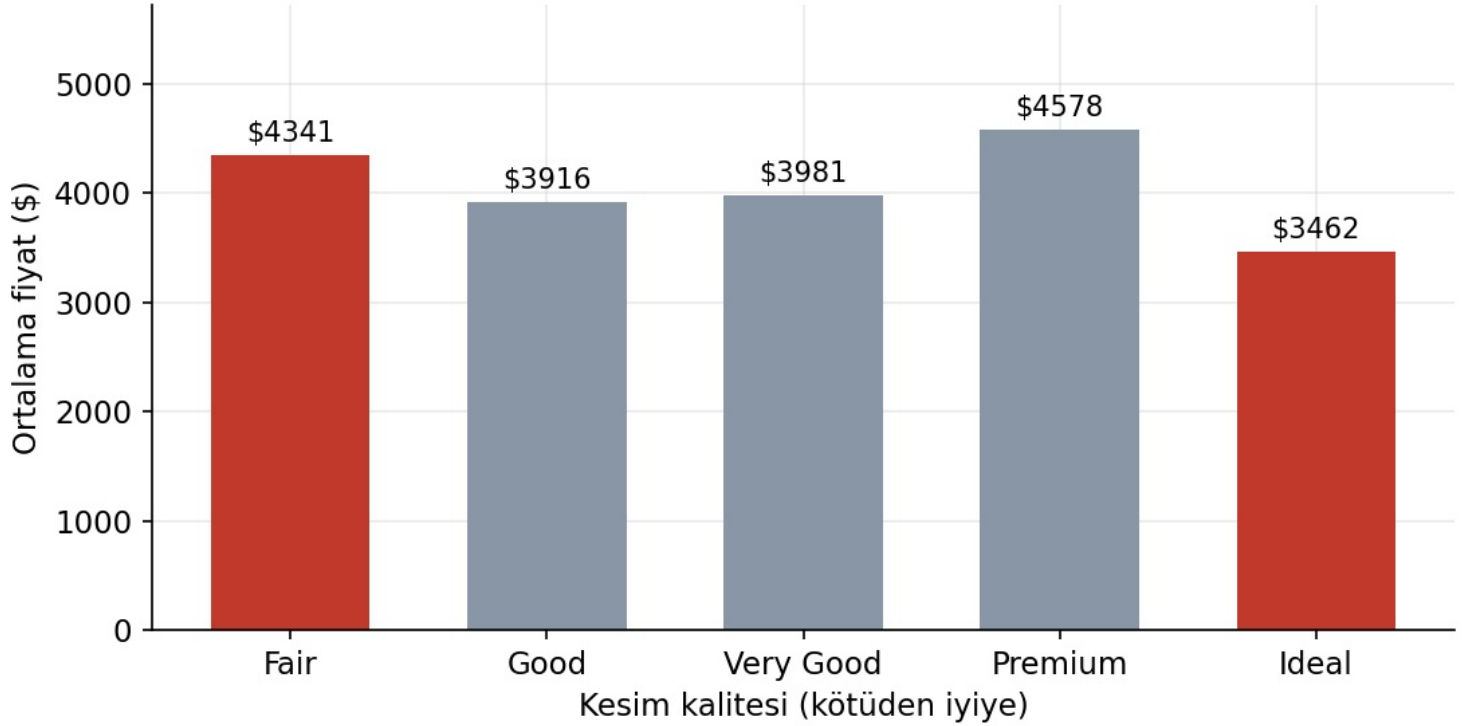
Temizlikten sonra en basit soruyu sordum: kesim kalitesi arttıkça fiyat artıyor mu?

```
df.groupby("cut", observed=True).agg(
    adet=("price", "size"),
    ort_fiyat=("price", "mean"),
    medyan_fiyat=("price", "median")
)
```

Kesim	Adet	Ortalama fiyat	Medyan fiyat
Fair	1.597	\$4.341	\$3.282

Good	4.888	\$3.916	\$3.038
Very Good	12.067	\$3.981	\$2.647
Premium	13.736	\$4.578	\$3.178
Ideal	21.484	\$3.462	\$1.813

Kesim kalitesi arttıkça fiyat... düşüyor?



En kötü kesim (Fair) ortalama **\$4.341**, en iyi kesim (Ideal) ortalama **\$3.462**. Aradaki fark %25. Medyanlara bakınca uçurum daha da büyük: \$3.282'ye karşı \$1.813, yani neredeyse iki katı.

Kod hatası aradım. Kategori sıralamasını iki kez kontrol ettim. Hata yoktu.

Sonra renge baktım ve durum daha da kötüleşti. D rengi renksiz ve en değerli olan, J ise sarımsı ve en değersiz olan:

Renk	Ortalama fiyat	Ortalama karat
D (en iyi)	\$3.173	0.66
E	\$3.080	0.66
F	\$3.727	0.74
G	\$3.999	0.77
H	\$4.476	0.91
I	\$5.080	1.02
J (en kötü)	\$5.326	1.16

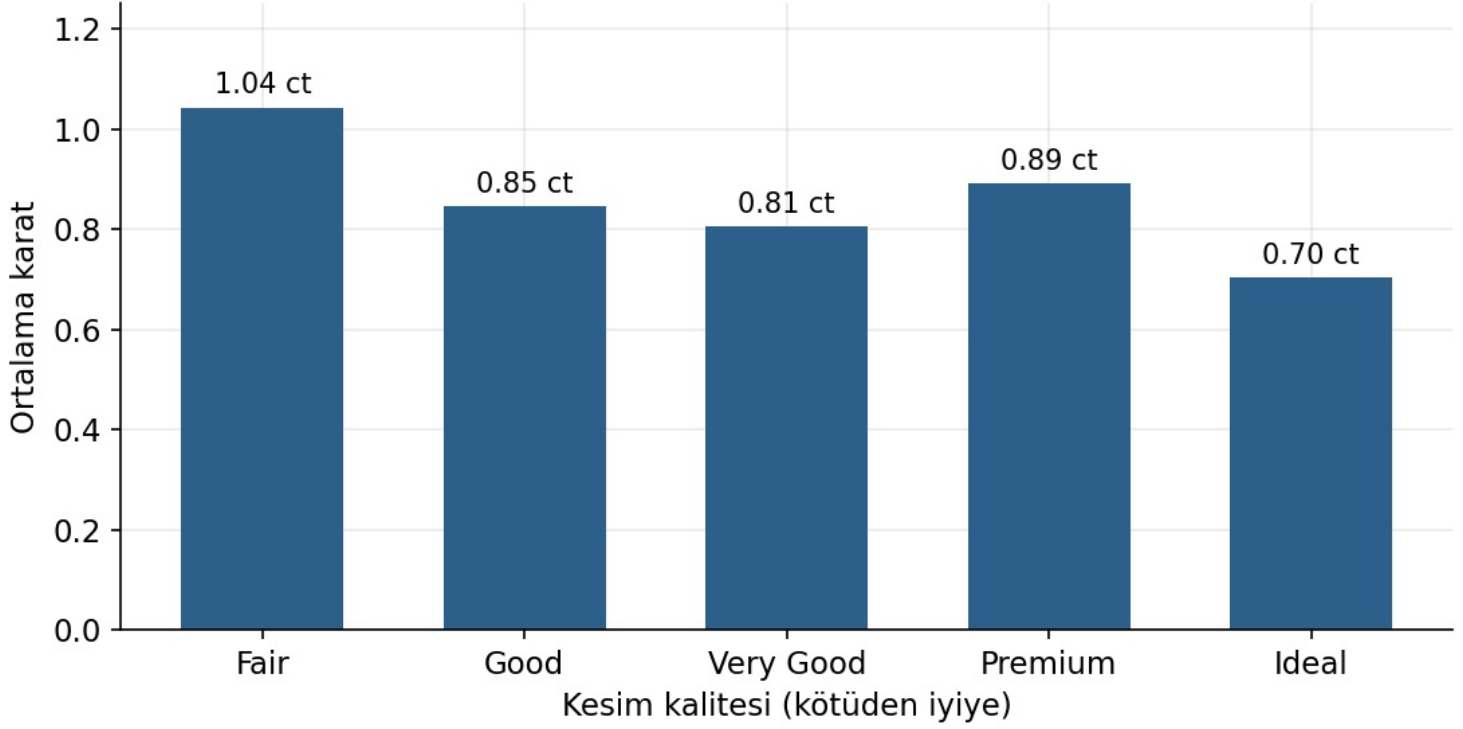
Kusursuz bir ters sıralama. En kötü renk, en iyi renkten %68 daha pahalı.

İki farklı kalite göstergesi de aynı yönde "yanlış" davranıyorsa, sorun kalite göstergelerinde değildir. Sorun benim sorumdadır.

Suçlu: karat

Yukarıdaki renk tablosunun sağ sütununa dikkat edin. Ortalama karat, tam olarak fiyatla aynı yönde artıyor. Aynı şeyi kesim için de yaptım:

Suçlu bulundu: iyi kesim taşlar daha küçük



Kesim	Ortalama karat
Fair	1.04
Good	0.85
Very Good	0.81
Premium	0.89
Ideal	0.70

İşte cevap. **İdeal kesim taşlar ortalama 0.70 karat, Fair kesim taşlar 1.04 karat** — yani %49 daha ağır.

Karatın fiyatla korelasyonu **0.92**. Veri setindeki en güçlü ilişki, açık ara. Kesim ve renk bu devin yanında toz zerresi.

Peki neden büyük taşlar kötü kesiliyor? Bu bir veri tuhaflığı değil, ekonomik bir gerçek. Elmas kesiminde ham taşın bir kısmı kaybedilir. Mükemmel oranlar için kesmek, ağırlığın önemli bir bölümünü feda etmek demektir. Ham taş büyük ve pahalıysa, kesici 1.04 karatlık "Fair" bir taşı, 0.85 karatlık "İdeal" bir taşı tercih eder — çünkü ağırlık, kesimden daha çok para eder. Yani **kesim kalitesi ile taş büyüklüğü arasında bilinçli bir takas var**.

Bu, istatistikte **karıştırıcı değişken** (confounding variable) denen şeyin ders kitabı örneği. Karat hem kesim kalitesini hem fiyatı etkiliyor. Karatı hesaba katmadan kesim-fiyat ilişkisine bakmak, yanlış cevabı garantiliyor.

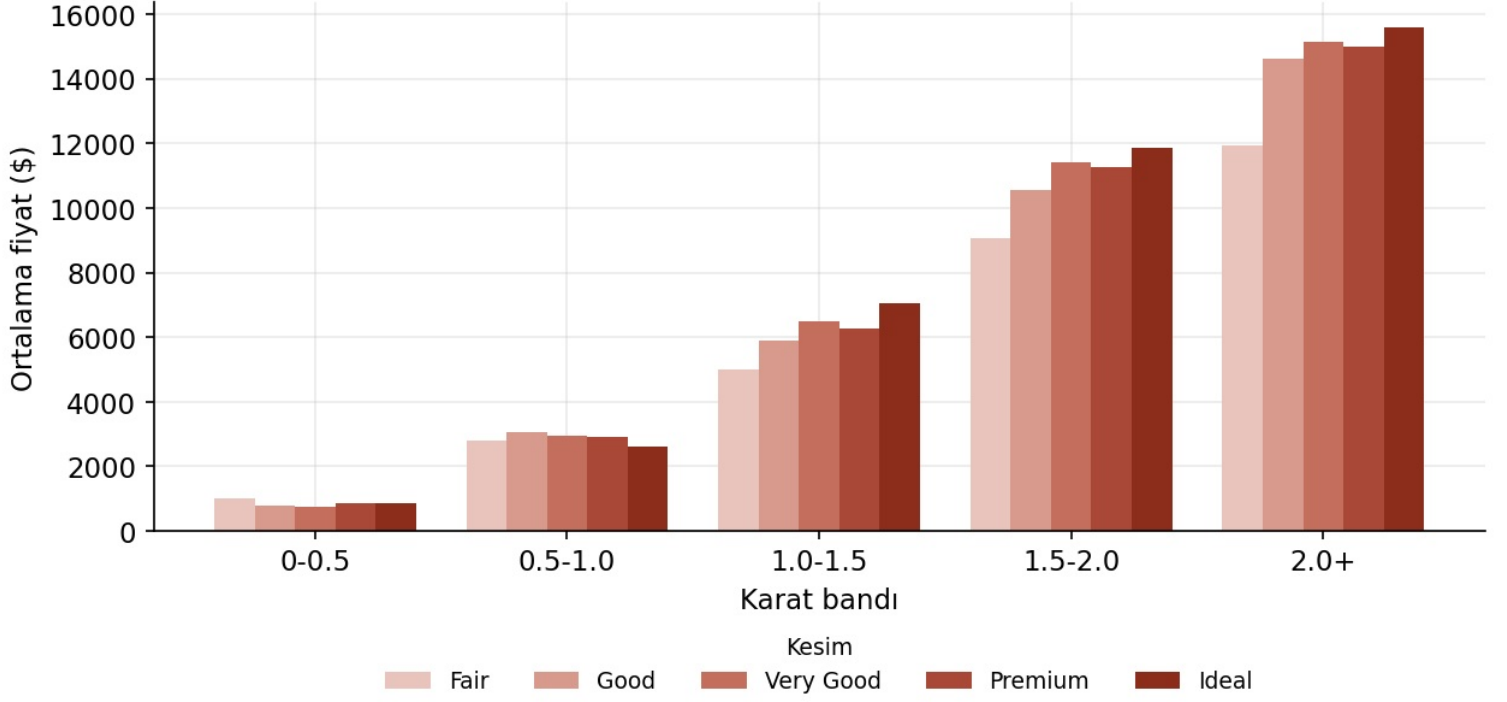
Karatı sabitleyince ne oluyor?

Çözüm: karşılaştırmayı benzer ağırlıktaki taşlar arasında yapmak. Karatı bantlara böldüm ve her bandın içinde kesime göre ortalama fiyata baktım:

```
df["karat_bandi"] = pd.cut(df.carat, [0, 0.5, 1.0, 1.5, 2.0, 6],
                             labels=["0-0.5", "0.5-1.0", "1.0-1.5", "1.5-2.0", "2.0+"])

df.pivot_table(index="karat_bandi", columns="cut",
                 values="price", aggfunc="mean", observed=True)
```

Aynı karat bandı içinde sıralama düzeliyor



Karat bandı	Fair	Good	Very Good	Premium	Ideal
0-0.5	1.018	786	767	863	864
0.5-1.0	2.800	3.061	2.958	2.903	2.597
1.0-1.5	4.996	5.896	6.503	6.264	7.058
1.5-2.0	9.059	10.556	11.437	11.261	11.867
2.0+	11.931	14.614	15.139	14.991	15.589

Sıralama düzeldi. 1 karat üstündeki her bantta Ideal kesim en pahalısı, Fair kesim en ucuzu. 2 karat üstünde aradaki fark **\$3.659**, yani %31.

Paradoks kayboldu. Aynı veri, aynı sütunlar — tek fark, doğru soruyu sormak.

İki not:

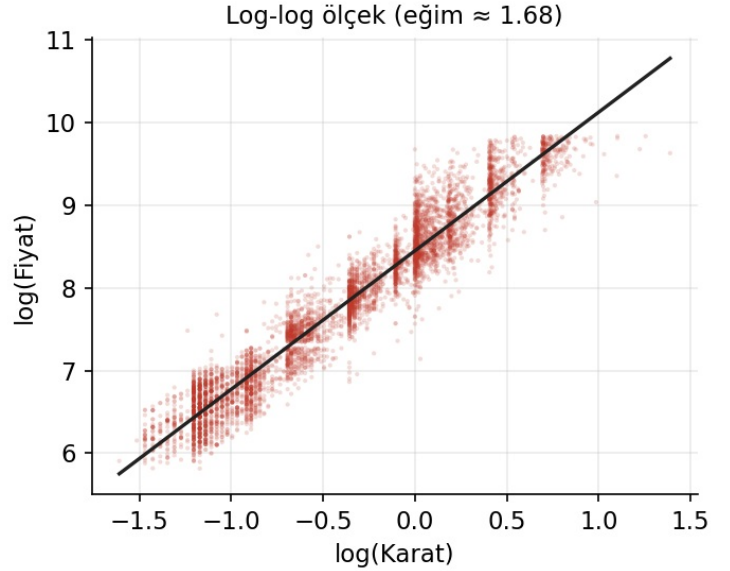
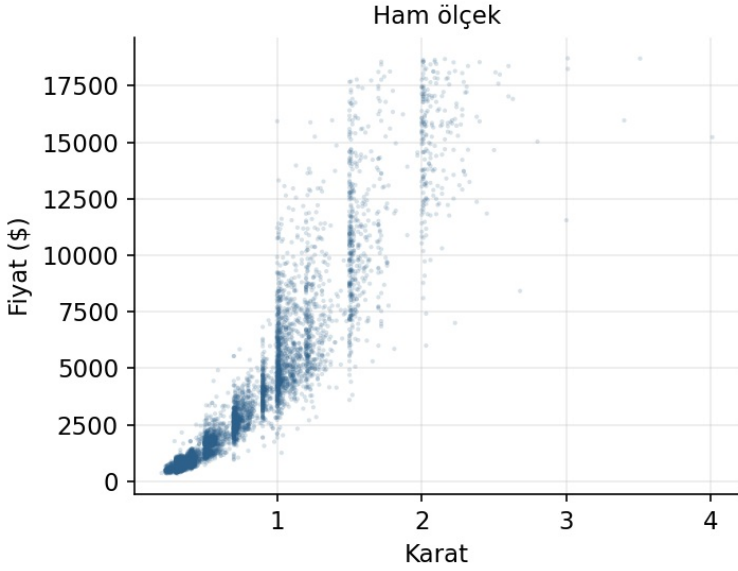
Küçük taşlarda (0-0.5) sıralama hâlâ tuhaf. Fair kesim burada en pahalı görünüyor. Ama o hücrede sadece 190 gözlem var, Ideal'de ise 9.104. Küçük örneklem + geniş bant = gürültü. Bu satıra bakıp "küçük taşlarda kesim önemsiz" demek, az önce düştüğüm tuzağa yeniden düşmek olurdu.

Bantlama mükemmel bir çözüm değil. 1.0-1.5 bandının içinde bile 1.02 karatlık bir taşla 1.48 karatlık bir taş aynı kefedede. Doğru yaklaşım karatı sürekli değişken olarak modele koymak — ama bu yazının konusu keşifçi analiz, model değil.

Bonus: ölçeği değiştirince ilişki düzleşiyor

Karat-fiyat ilişkisi doğrusal değil. Ham veride saçılım grafiği yukarı kıvrılan bir yay çiziyor; fiyat dağılımının çarpıklığı da 1.62, yani sağa hayli eğik. İki değişkenin de logaritmasını alınca:

Aynı ilişki, iki farklı ölçek



```
egim, kesisim = np.polyfit(np.log(df.carat), np.log(df.price), 1)
# egim  $\approx 1.68$ , korelasyon  $\approx 0.966$ 
```

Log-log ölçekte korelasyon 0.922'den **0.966**'ya çıkıyor ve ilişki neredeyse düz bir çizgi hâline geliyor.

Eğimin **1.68** olması ekonomik olarak anlamlı: karat %1 arttığında fiyat %1.68 artıyor. Yani 2 karatlık bir taş, 1 karatlık bir taşın iki katı değil, yaklaşık **üç katı** eder. Büyük taşlar doğada nadir olduğu için fiyat ağırlıktan hızlı artıyor.

Çıkardığım dersler

- Ortalama, gruplar dengesizse yalan söyler.** Buradaki her grup farklı bir karat dağılımına sahipti. Grup ortalamalarını karşılaştırmadan önce grupların birbirine benzeyip benzemediğini kontrol etmek gerekiyor.
- "Saçma" sonuç, analizin bittiği yer değil başladığı yer.** Fair kesim daha pahalı çıktığında iki seçeneğim vardı: sonucu olduğu gibi yazmak, ya da nedenini bulmak. Yazının tamamı ikinci seçeneğin ürünü.
- `isnull()` temizlik kontrolü değildir.** 20 tane sıfır boyutlu elmas hiçbir null testine takılmadı. Sayısal sütunlarda "bu değer fiziksel olarak mümkün mü?" diye sormak gerekiyor.
- Alan bilgisi olmadan veri analizi eksik kalıyor.** Kesim-ağırlık takasını bilmeseydim, karatın karıştırıcı değişken olduğunu bulur ama *neden* öyle olduğunu açıklayamazdım. Bulguyu rapora çeviren şey, o "neden".
- Hücre başına gözlem sayısına bakın.** 190 gözleme dayalı bir ortalama ile 9.104 gözleme dayalı bir ortalama yan yana koyup eşit muamele etmek, ilk hatanın küçük ölçekli bir tekrarı.

Bu analizin en değerli çıktısı bir grafik ya da bir sayı değildi. Kendi çıkardığım sonuca "bu doğru olamaz" diyebilme alışkanlığıydı. Modeller kurmayı öğrenmek görece kolay; kendi sonucunuzdan şüphe etmeyi öğrenmek asıl iş.

Analizin tamamı Python, pandas ve matplotlib ile yapıldı. Veri seti: `diamonds` (53.940 gözlem, açık kaynak). Kodun tamamı ve grafikleri üreten script burada: [GitHub deposu](#)