

# İÇİNDEKİLER

1. GİRİŞ .....	3
2. YAZILIM BİLGİSİ .....	4
2.1 Temel Özellikler .....	4
2.2 Başlıca Kullanım Alanları .....	5
2.3 Spyder'ın Özellikleri .....	5
3. VERİ KÜMESİ .....	6
3.1 Bike Sharing Background.....	6
3.2 Veri Seti Hakkında.....	6
3.3 Referans .....	6
3.4 Değişkenler .....	7
3.4 Sonuç .....	7
4. ANALİZ .....	8
4.1 TANIMLAYICI İSTATİSTİKLER.....	8
Kullanılan Kütüphaneler.....	8
Eksik Değerlerin Kontrolü.....	9
Sayısal Değişkenler İçin Temel İstatistikler .....	9
Tanımlayıcı İstatistik Fonksiyonu ve İstatistiklerin Hesaplanması .....	10
Korelasyon Analizi .....	12
4.2 GRAFİK ÇİZİMİ .....	13
Kullanılan Kütüphaneler .....	13
Grafikler.....	13
Sağ Yönlü Scatter Plot: rüzgar_hızı vs. nem .....	13
Swarm Plot: sıcaklık vs. haftanın_günü .....	14
Pairplot: sıcaklık, rüzgar_hızı ve çalışma_günü.....	15
Pairplot: sıcaklık, rüzgar_hızı ve ay .....	15
4.3 BİR ÖRNEKLEMDEN BİLİNMEYEN MİKTARLARI TAHMİN ETME.....	16
Kullanılan Kütüphaneler.....	16
Popülasyon ve Örneklem.....	17
Basit Rastgele Örneklem Seçimi .....	17

Merkezi Limit Teoremi.....	18
Parametrelerin Tahmini ve Güven Aralığı.....	19
4.4 İKİ ORTALAMAYI KARŞILAŞTIRMA .....	20
Kullanılan Kütüphaneler.....	20
Tek Örneklem Z-Testi.....	20
Tek Örneklem T-Testi .....	21
Bağımsız Örneklem T-Testi (Student Testi).....	21
Bağımsız Örneklem T-Testi (Welch Testi).....	21
Eşleştirilmiş Örneklem T-Testi .....	21
Tek Yönlü Testler (One-Sided Tests).....	22
Etki Büyüklüğü (Effect Size).....	22
Normallik Kontrolü .....	22
Normale Uymayan Veriler için Wilcoxon Testleri .....	22
5. SONUÇ .....	23
6. Kaynakça.....	24

# 1. GİRİŞ

Bu rapor, Spyder ile ilgili genel bilgilendirmeleri ve analizleri içermektedir. Raporun amacı, yazılımın özelliklerini, kullanım alanlarını ve sunduğu istatistiksel yöntemleri detaylı bir şekilde incelemektir. Ayrıca, yazılımın kullanıcı arayüzünü ve gerekli paketler ile fonksiyonları da açıklayarak, yazılımın kullanımını daha anlaşılır hale getirmek hedeflenmiştir.

İlk bölümde, yazılımın genel özellikleri ve başlıca kullanım alanları üzerinde durulacaktır. Yazılımın temel işlevlerini ve kullanım amacını açıklayarak, bu araçların veri analizi ve istatistiksel yöntemlerde nasıl kullanılabileceği tartışılacaktır.

Bir sonraki bölümde, yazılımın ekran görüntüleri ve kullanıcı arayüzüne dair bilgiler verilecektir. Yazılımın çalışma ortamı ve görsel bileşenleri üzerinde durulacak, ekran pencereleri hakkında bilgiler sağlanacaktır. Ayrıca, yazılımın gerektirdiği paketler ve fonksiyonlar hakkında bilgi verilecektir.

Veri bölümünde, kullanılan örnek veriler açıklanacak, bu verilerin yapılarına, değişken sayılarına ve gözlem sayılarına dair bilgiler sunulacaktır. Kullanılan verilerin niteliği, yapılan analizlerin doğruluğu açısından önemli bir yer tutmaktadır.

Raporda, her bölümde kullanılan istatistiksel yöntemler; Tanımlayıcı İstatistikler, Grafik Çizme, Bilinmeyen Miktarların Tahmini, İki Ortalamanın Karşılaştırılması ve bu yöntemlerin analizlerdeki rolü üzerinde durulacaktır. Özellikle temel istatistiksel analizlerin dışında, yazılımın sunduğu ileri düzey yöntemlerin kullanımı ve bu yöntemlerin gerçek dünya problemleriyle nasıl ilişkilendirilebileceği tartışılacaktır.

Bu rapor, yazılımın kullanımını detaylı bir şekilde anlatmayı ve kullanıcıların yazılımı daha etkili kullanmalarını sağlamayı amaçlamaktadır.

## 2. YAZILIM BİLGİSİ

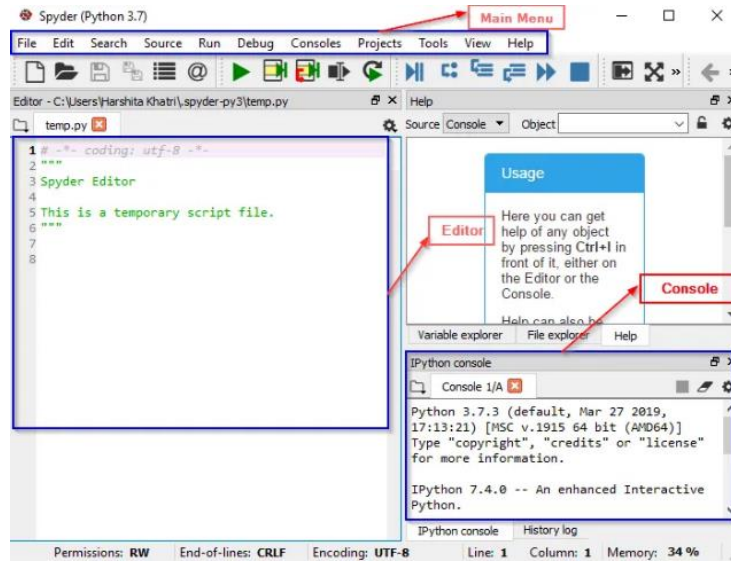
Spyder, Python Dili için geliştirilmiş açık kaynaklı ve platformlar arası çalışan bir IDE'dir. Spyder, "Scientific Python Development IDE" olarak da bilinir ve birçok güçlü özelliğe sahiptir. Spyder, veri analizi, bilimsel hesaplamalar ve makine öğrenimi gibi alanlarda yoğun bir şekilde kullanılır. Python Spyder IDE, Anaconda Python dağıtımıyla birlikte varsayılan bir uygulama olarak gelir. Bu yalnızca önerilen yöntem değil, aynı zamanda en kolay olanıdır.

### 2.1 Temel Özellikler

#### Dosya Oluşturma/Proje Başlatma:

Yeni bir dosya başlatmak için aşağıdaki şekilde ilerleyin: *Dosya->Yeni Dosya*

Yeni bir proje oluşturmak için: *Projeler->Yeni Proje*



#### Spyder'ın Arayüzü Genellikle 4 Ana Pencereden Oluşur:

- **Editor:** Python kodu yazacağınız alan.
- **IPython Console:** Kodu çalıştırıp çıktı alacağınız yer.
- **Variable Explorer:** Değişkenleri görsel olarak inceleyebilirsiniz.
- **File Explorer:** Proje dosyalarınızı düzenlemenize yardımcı olur.

## 2.2 Başlıca Kullanım Alanları

- Veri Bilimi ve Analizi
- Makine Öğrenimi
- Bilimsel Hesaplamalar
- Eğitim

## 2.3 Spyder'ın Özellikleri

### 1) Özelleştirilebilir Sözdizimi Vurgulama

```
7
8 import numpy as np
9 data = np.array([1, 2, 3, 4])
10 print(np.mean(data))
11
```

Import ve print gibi anahtar kelimeler ile np ve data gibi değişkenler farklı renklerde vurgulanır.

### 2) Kesme Noktaları ve Hata Ayıklama

```
12
13 import numpy as np
14 data = np.array([1, 2, 3, 4])
15 breakpoint() # Bu satırda kesme noktası eklenir
16 print(np.mean(data))
17
```

Bu kodda, breakpoint() fonksiyonu kesme noktası ekler ve hata ayıklama başlatılır.

### 3) Etkileşimli Yürütme

Spyder, kullanıcıların satır, dosya ve hücre bazında kod çalıştırmalarına olanak tanır. Bu, hızlı testler yaparken ve büyük projelerde verimli çalışırken oldukça kullanışlıdır.

### 4) Gerçek Zamanlı Kod İncelemesi

Kodunuzda hata, stil hatası veya eksik parametreler varsa, Spyder bunları hemen bildirir.

### 5) Matplotlib İle Inline Grafik Görselleştirme

Spyder, veri görselleştirmeleri için Matplotlib ve benzeri kütüphaneleri destekler. Grafikler, kodun hemen altında satır içi (inline) olarak görüntülenebilir.

### 3. VERİ KÜMESİ

#### 3.1 Bike Sharing Background

Bisiklet paylaşım sistemleri, üyelik, kiralama ve iade işlemlerinin otomatik olarak gerçekleştirildiği modern bisiklet kiralama yöntemleridir. Kullanıcılar, bir noktadan bisiklet kiralar ve başka bir noktaya bırakabilir. Dünya genelinde 500'den fazla program ve 500 binin üzerinde bisiklet bulunmaktadır. Bu sistemler, trafik, çevre ve sağlık alanlarında önemli rollere sahiptir ve oluşturdukları veri yapısıyla araştırmalar için cazip bir kaynak sunar. Seyahat süresi, başlangıç ve bitiş konumlarını kaydeden bu sistemler, şehir hareketliliğini analiz etmek ve önemli olayları algılamak için bir sensör ağı işlevi görür.

#### 3.2 Veri Seti Hakkında

Veri seti, bisiklet kiralama süreçlerinin çevresel ve mevsimsel faktörlerle güçlü bir şekilde ilişkili olduğunu göstermektedir. Örneğin, hava durumu, yağış, haftanın günü, mevsim ve günün saati gibi faktörler kiralama davranışlarını etkileyebilir.

Bu veri seti, ABD'nin Washington D.C. şehrinde bulunan **Capital Bikeshare** sistemine ait 2011 ve 2012 yıllarındaki iki yıllık geçmiş log verilerini içermektedir.

Veri seti, bisiklet kiralama süreçlerinin çevresel ve mevsimsel faktörlerle güçlü bir şekilde ilişkili olduğunu göstermektedir. Örneğin, hava durumu, yağış, haftanın günü, mevsim ve günün saati gibi faktörler kiralama davranışlarını etkileyebilir.

Bu veri seti, ABD'nin Washington D.C. şehrinde bulunan **Capital Bikeshare** sistemine ait 2011 ve 2012 yıllarındaki iki yıllık geçmiş log verilerini içermektedir.

#### 3.3 Referans

Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", *Progress in Artificial Intelligence* (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3.

### 3.4 Değişkenler

Değişken	Açıklama
instant	Kayıt dizini.
dteday	Tarih.
season	Mevsim (1: İlkbahar, 2: Yaz, 3: Sonbahar, 4: Kış).
yr	Yıl (0: 2011, 1: 2012).
mnth	Ay (1: Ocak, 2: Şubat, ..., 12: Aralık).
holiday	Günün tatil olup olmadığı (1: Tatil, 0: Tatil değil).
weekday	Haftanın günü (0: Pazar, 1: Pazartesi, ..., 6: Cumartesi).
workingday	Çalışma günü (1: Çalışma günü, 0: Hafta sonu veya tatil).
weathersit	Hava durumu kategorisi (1: Açık, az bulutlu, 2: Sisli, bulutlu, 3: Hafif yağmur/kar, 4: Şiddetli yağmur/kar/sis).
temp	Normalleştirilmiş sıcaklık (maks. sıcaklık 41'e bölünmüş).
atemp	Normalleştirilmiş hissedilen sıcaklık (maks. sıcaklık 50'ye bölünmüş).
hum	Normalleştirilmiş nem oranı (maks. 100'e bölünmüş).
windspeed	Normalleştirilmiş rüzgar hızı (maks. 67'ye bölünmüş).
casual	Gündelik kullanıcıların sayısı.
registered	Kayıtlı kullanıcıların sayısı.
cnt	Toplam bisiklet kiralama sayısı (gündelik + kayıtlı kullanıcılar).

### 3.4 Sonuç

Bu veri seti, bisiklet paylaşım sistemlerinin hem gerçek dünya uygulamalarını analiz etmek hem de araştırmalar için geniş bir olanak sunmaktadır. Çevresel faktörler ile bisiklet kiralama davranışları arasındaki ilişkiyi analiz ederek ulaşım, çevre ve sağlık konularında önemli çıkarımlar yapılabilir. Ayrıca, anomali tespiti gibi gelişmiş analizler için de ideal bir veri kaynağıdır.

## 4. ANALİZ

### 4.1 TANIMLAYICI İSTATİSTİKLER

- **Kategorik Değişkenler:** Mevsim, yıl, ay, tatil durumu, haftanın günü, çalışma günü, hava durumu.
- **Sayılarla İlgili Değişkenler:** Sıcaklık, hissedilen sıcaklık, nem, rüzgar hızı, gündelik kullanıcı sayısı, kayıtlı kullanıcı sayısı, toplam kullanıcı sayısı.

### Kullanılan Kütüphaneler

**NumPy (import numpy as np):** Sayısal hesaplamalar ve diziler için kullanılır.

**Pandas (import pandas as pd):** Veri analizi ve manipülasyonu için kullanılır, özellikle tabular veri üzerinde işlem yapar.

**SciPy (from scipy.stats import trim\_mean, skew, kurtosis, mode):** İstatistiksel fonksiyonlar sağlar.

- **trim\_mean:** Kırpılmış ortalama hesaplar.
- **skew:** Verinin eğimini (asimetrisini) ölçer.
- **kurtosis:** Basıklık ölçüsü.
- **mode:** En sık görülen değeri (mod) hesaplar.

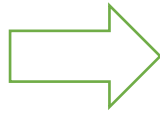
**StatsModels (from statsmodels import robust):** İstatistiksel hesaplamalar ve modeller için kullanılır.

- **robust.mad:** Medyan mutlak sapma (MAD) hesaplar.

**Seaborn (import seaborn as sns):** İstatistiksel veri görselleştirmeleri için kullanılır.

- **heatmap:** Korelasyon gibi verileri renkli harita ile görselleştirir.

```
In [2]: veriSeti.info()
...: veriSeti.dtypes
...: print(veriSeti.shape)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   instant     731 non-null   int64
1   dteday      731 non-null   object
2   season      731 non-null   int64
3   yr          731 non-null   int64
4   mnth       731 non-null   int64
5   holiday     731 non-null   int64
6   weekday    731 non-null   int64
7   workingday  731 non-null   int64
8   weathersit   731 non-null   int64
9   temp       731 non-null   float64
10  atemp      731 non-null   float64
11  hum        731 non-null   float64
12  windspeed  731 non-null   float64
13  casual     731 non-null   int64
14  registered  731 non-null   int64
15  cnt        731 non-null   int64
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB
(731, 16)
```



```
In [13]: print(veriSeti.dtypes)
kayıt_dizini      int64
tarih             object
mevsim            category
yıl               category
ay               category
tatil            category
haftanın_günü     category
çalışma_günü      category
hava_durumu       category
sıcaklık         float64
hissedilen_sıcaklık float64
nem              float64
rüzgar_hızı       float64
gündelik_kullanıcı int64
kayıtlı_kullanıcı int64
toplam_kullanıcı  int64
dtype: object
```



## Eksik Değerlerin Kontrolü

Veri setimizde eksik değer (NaN) bulunmamaktadır. Çünkü:

**veriSeti.isnull().any():** False döndürüyorsa, bu sütunda hiç eksik değer olmadığını ifade eder.

**veriSeti.isnull().sum():** Tüm sütunlar için eksik değer sayısı sıfır (0) olarak görünmekte. Bu da her bir sütunun eksiksiz olduğunu gösterir.

```
In [19]: veriSeti.isnull().any()
Out[19]:
kayıt_dizini      False
tarih             False
mevsim            False
yıl               False
ay                False
tatil             False
haftanın_günü     False
çalışma_günü      False
hava_durumu       False
sıcaklık         False
hissedilen_sıcaklık False
nem               False
rüzgar_hızı       False
gündelik_kullanıcı False
kayıtlı_kullanıcı False
toplam_kullanıcı  False
dtype: bool
```

```
In [20]: veriSeti.isnull().sum()
Out[20]:
kayıt_dizini      0
tarih             0
mevsim            0
yıl               0
ay                0
tatil             0
haftanın_günü     0
çalışma_günü      0
hava_durumu       0
sıcaklık         0
hissedilen_sıcaklık 0
nem               0
rüzgar_hızı       0
gündelik_kullanıcı 0
kayıtlı_kullanıcı  0
toplam_kullanıcı  0
dtype: int64
```

## Sayısal Değişkenler İçin Temel İstatistikler

Değişken	25%	50%	75%	count	max	mean	min	std
Kayıt Dizini	183.5	366.0	548.5	731	731.0	366.0	1.0	211.166
Sıcaklık	0.3378	0.4983	0.6554	731	0.8617	0.4954	0.0591	0.1831
Hissedilen Sıcaklık	0.3378	0.4867	0.6869	731	0.8409	0.4745	0.0797	0.163
Nem	0.52	0.6267	0.7302	731	0.9725	0.6279	0.0	0.1424
Rüzgar Hızı	0.1349	0.181	0.2332	731	0.5075	0.1985	0.0224	0.0775
Gündelik Kullanıcı	315.5	713.0	1096.0	731	3410.0	848.1765	2.0	686.622
Kayıtlı Kullanıcı	2497.0	3662.0	4776.5	731	6946.0	3656.17	20.0	1560.26
Toplam Kullanıcı	3152.0	4548.0	5956.0	731	8714.0	4504.35	22.0	1937.21

**Kayıt dizini**, veri setinin 731 gözlem içerdiğini gösteriyor. **Sıcaklık** verisi çoğunlukla 0 ile 1 arasında yoğunlaşırken, uç değerlerin geniş bir değişim gösterdiği dikkat çekiyor. **Hissedilen sıcaklık** genellikle gerçek sıcaklığa yakın değerler alırken, çevresel faktörlerin etkisiyle daha

yüksek değerlerde yoğunlaşma gözlemleniyor. **Nem** verisi, yüksek seviyelerde yoğunlaşırken geniş bir aralıkta dağılmakta, düşük nem oranları ise daha sık görülüyor. **Rüzgar hızı** verisi çoğunlukla düşük seviyelerde yoğunlaşmış, fakat bazı günlerde yüksek hızlar da gözlemleniyor. **Gündelik kullanıcı** sayısı, büyük değişkenlik gösteriyor; bazı günlerde yoğun kullanım varken, diğer günlerde düşük taleple karşılaşılıyor. **Kayıtlı kullanıcı** sayısının ise yüksek ve genellikle dengeli bir dağılım olduğu görülüyor. Son olarak, **toplam kullanıcı** çoğu zaman ortalama etrafında yoğunlaşmış durumda.

## Tanımlayıcı İstatistik Fonksiyonu ve İstatistiklerin Hesaplanması

Değişken	Ortalama	Ortanca	Kırpmış Ortalama	Mod	Aralık	Çeyreklik Aralık
Sıcaklık	0.4954	0.4983	0.4972	0.2658	0.8025	0.3183
Hissedilen Sıcaklık	0.4744	0.4867	0.4769	0.6547	0.7618	0.2708
Nem	0.6279	0.6267	0.6271	0.6133	0.9725	0.2102
Rüzgar Hızı	0.1905	0.181	0.1851	0.1064	0.4851	0.0983
Gündelik Kullanıcı	848.1765	713.0	744.9504	120.0	3408.0	780.5
Kayıtlı Kullanıcı	3656.1724	3662.0	3641.7231	1707.0	6926.0	2279.5
Toplam Kullanıcı	4504.3488	4548.0	4517.1897	1096.0	8692.0	2804.0

Ortalama Mutlak Sapma	Varyans	Standart Sapma	Medyan Mutlak Sapma	Eğim (Skewness)	Basıklık (Kurtosis)
0.1593	0.0335	0.1831	0.2347	-0.0544	-1.1194
0.1405	0.0266	0.163	0.2011	-0.1308	-0.9866
0.1164	0.0203	0.1424	0.1551	-0.0696	-0.0723
0.0612	0.006	0.0775	0.0728	0.676	0.3999
520.8103	471450.4414	686.6225	587.1105	1.2639	1.3049
1263.723	2434399.962	1560.2564	1712.4056	0.0436	-0.7164
1581.7927	3752788.2083	1937.2115	2086.0213	-0.0473	-0.8146

Bisiklet paylaşım sisteminin kullanıcı sayısının oldukça değişken olduğunu, ancak genellikle yoğun talep gördüğünü ve çevresel faktörlerin, özellikle sıcaklık ve nemin, kullanıcı sayısındaki değişimlere etki ettiğini göstermektedir.

## 1. Sıcaklık

- **Ortalama:** 0.4954 — Ortalama sıcaklık değeri, verilerin çoğunluğunun 0 ile 1 arasında yoğunlaştığını gösteriyor.
- **Aralık:** 0.8025 — Sıcaklıkta geniş bir varyasyon bulunuyor, bu da sıcaklık verilerinin büyük bir değişkenliğe sahip olduğunu gösteriyor.
- **Eğim (Skewness):** -0.0544 — Eğim negatif, bu da sağa çarpık bir dağılımı işaret eder. Sıcaklık verisi genellikle düşük değerlere daha yakın.

Sıcaklık verisi, çoğunlukla ortalama civarında yoğunlaşmış ve uç değerlere sahip olmasına rağmen genellikle dengeli bir dağılım sergiliyor.

## 2. Hissedilen Sıcaklık

- **Ortalama:** 0.4745 — Ortalama değeri, hissedilen sıcaklık ile sıcaklık arasındaki farkı anlamamıza yardımcı olur.
- **Mod:** 0.6547 — Hissedilen sıcaklık için en sık görülen değer, daha sıcak günlerde bisiklet kullanımının arttığını gösteriyor.
- **Aralık:** 0.7618 — Hissedilen sıcaklıkta da oldukça geniş bir dağılım gözlemleniyor.

Hissedilen sıcaklık değeri genellikle yüksek değerlerde yoğunlaşıyor, çevresel faktörlerin etkisiyle kullanıcı sayısı artabiliyor.

## 3. Gündelik Kullanıcı

- **Ortalama:** 848.1765 — Günlük kullanıcı sayısı ortalaması oldukça yüksek, bisiklet paylaşım sistemi popüler.
- **Aralık:** 3408 — Gündelik kullanıcı sayısı büyük değişkenlik gösteriyor, bazı günlerde çok fazla kullanıcı varken, diğer günlerde düşük kullanım gözlemleniyor.
- **Eğim (Skewness):** 1.2639 — Yüksek pozitif eğim, kullanıcı sayısının çoğu zaman daha yüksek olduğunu gösteriyor.

Gündelik kullanıcı sayısı büyük değişkenlik gösteriyor, bazı günlerde yoğun kullanım, diğerlerinde düşük kullanım var.

#### 4. Toplam Kullanıcı

- **Ortalama:** 4504.3488 — Toplam kullanıcı sayısı da oldukça yüksek, sistem yoğun şekilde kullanılıyor.
- **Aralık:** 8692 — Kullanıcı sayısındaki büyük değişkenlik, bazı günlerde aşırı yoğun kullanım, diğer günlerde düşük kullanım olduğunu gösteriyor.
- **Basıklık (Kurtosis):** -0.8146 — Düşük basıklık, verilerin daha düz bir şekilde dağıldığını ve uç noktaların daha seyrek olduğunu gösteriyor.

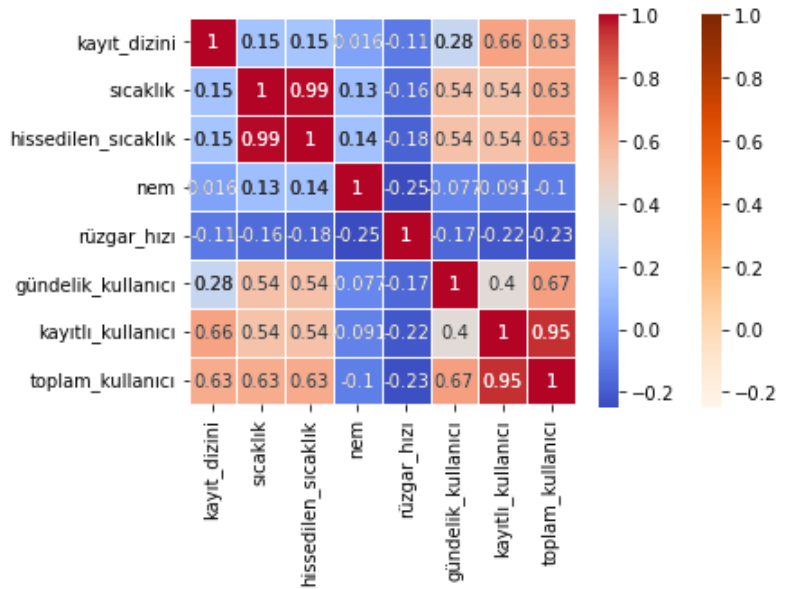
Toplam kullanıcı sayısındaki değişkenlik oldukça fazla, ancak veriler çoğu zaman ortalama etrafında yoğunlaşıyor.

#### Korelasyon Analizi

Görselde genel olarak sıcaklık ve hissedilen sıcaklık gibi değişkenler arasında çok güçlü bir ilişki gözlemleniyor, çünkü sıcaklık genellikle insanların hissettiği sıcaklıkla doğrudan bağlantılıdır. Diğer taraftan, rüzgar hızı ve nem arasındaki zayıf negatif korelasyon, bu iki değişkenin birbirine ters bir ilişki gösterdiğini ancak çok güçlü olmadığını belirtir.

```
In [32]: veriSeti.corr().unstack().min()
C:\Users\aktas\AppData\Local\Temp\ipykernel
numeric_only in DataFrame.corr is deprecated
valid columns or specify the value of numeric_only
veriSeti.corr().unstack().min()
Out[32]: -0.24848909864371393

In [33]: veriSeti.corr().unstack().idxmin()
C:\Users\aktas\AppData\Local\Temp\ipykernel
numeric_only in DataFrame.corr is deprecated
valid columns or specify the value of numeric_only
veriSeti.corr().unstack().idxmin()
Out[33]: ('nem', 'rüzgar_hızı')
```



**sıcaklık ve hissedilen\_sıcaklık:** Bu ikisi arasında 0.99 gibi çok yüksek bir pozitif korelasyon gözlemleniyor. Yani, sıcaklık ile hissedilen sıcaklık arasındaki ilişki neredeyse mükemmel bir şekilde paralel. Bu, sıcaklık arttığında hissedilen sıcaklığın da büyük ölçüde arttığını gösteriyor.

**kayıt\_dizini ve toplam\_kullanıcı:** Bu iki değişken arasında 0.66'lık bir pozitif korelasyon var. Kayıt dizini arttıkça toplam kullanıcı sayısının da artma eğiliminde olduğu söylenebilir.

**'nem' ve 'rüzgar\_hızı':** Aralarındaki negatif korelasyon -0.2485 civarında. Bu zayıf bir negatif ilişkiyi ifade eder. Yani, rüzgar hızı arttıkça nemin biraz azaldığı ancak bu ilişkinin çok güçlü olmadığı söylenebilir.

**'sıcaklık' ve 'günlük\_kullanıcı':** 0.28'lik bir pozitif korelasyon. Sıcaklık ile günlük kullanıcı sayısı arasında da zayıf bir ilişki var. Yani sıcaklık arttıkça günlük kullanıcı sayısının arttığı bir eğilim gözlemleniyor ancak bu da zayıf bir ilişki.

## 4.2 GRAFİK ÇİZİMİ

### Kullanılan Kütüphaneler

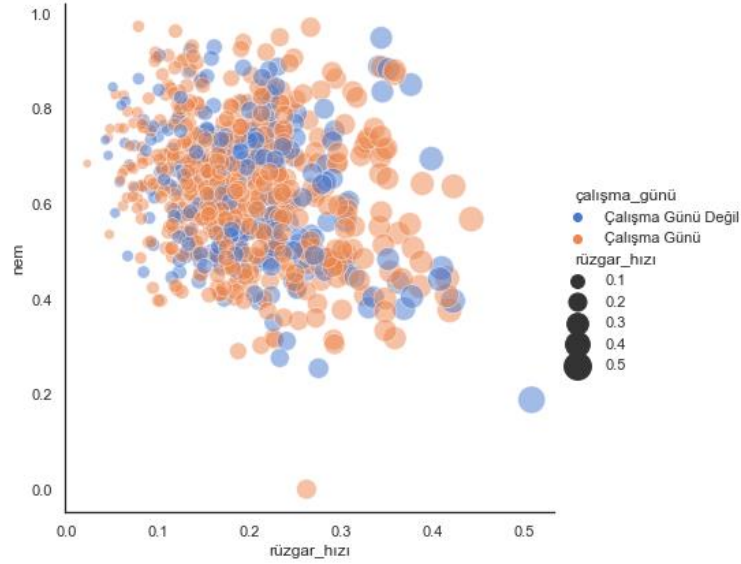
**Matplotlib (import matplotlib.pyplot as plt):** Veriyi görselleştirmek için grafikler oluşturmak amacıyla kullanılır. Bu kütüphane, scatter, show, xlabel, ylabel, title gibi fonksiyonlarla grafikler üzerinde işlem yapmanızı sağlar.

**Seaborn (import seaborn as sns):** Daha gelişmiş ve estetik görselleştirmeler oluşturmak için kullanılan bir Python kütüphanesidir. relplot, swarmplot, ve pairplot gibi fonksiyonlarla veri analizi ve görselleştirmeleri yapılır. sns.set\_theme ile grafiklerin genel stilini belirler.

### Grafikler

#### Sağ Yönlü Scatter Plot: rüzgar\_hızı vs. nem

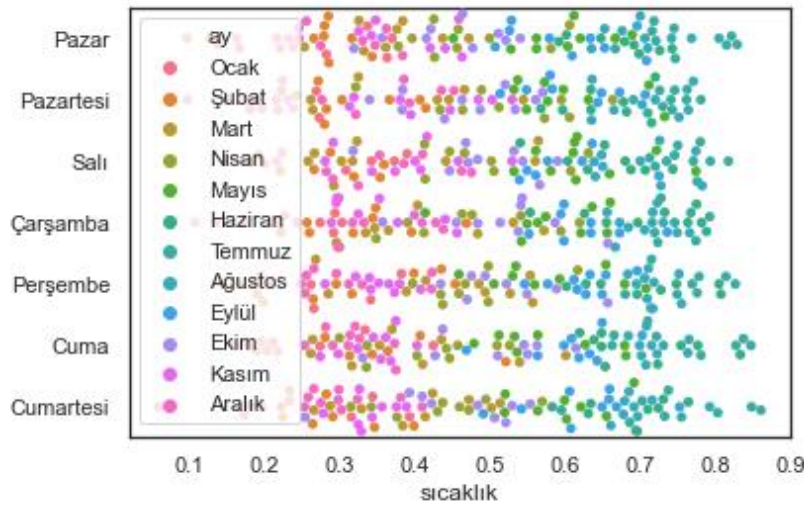
Rüzgar hızı ile nem arasında belirgin bir ilişki görünmüyor. Çalışma günlerinde nem oranının, rüzgar hızına göre daha düşük olduğu veya daha homojen dağıldığı gözlemlenebilir. Çalışma günü olan ve olmayan günler arasında rüzgar hızı ve nemin farklı seyirler izlediği söylenebilir.



### Swarm Plot: sıcaklık vs. haftanın\_günü

Sıcaklıkların haftanın günlerine göre değişiklik gösterdiğini, hafta ortasında Çarşamba, Perşembe gibi sıcaklıklar daha yüksek veya daha düşük olabilir. Bu durum, haftanın bazı günlerinde belirli aktivitelerin veya çevresel koşulların (örneğin iş günleri ve hafta sonları arasındaki farklılıklar) sıcaklık üzerinde etkisi olduğunu gösterebilir.

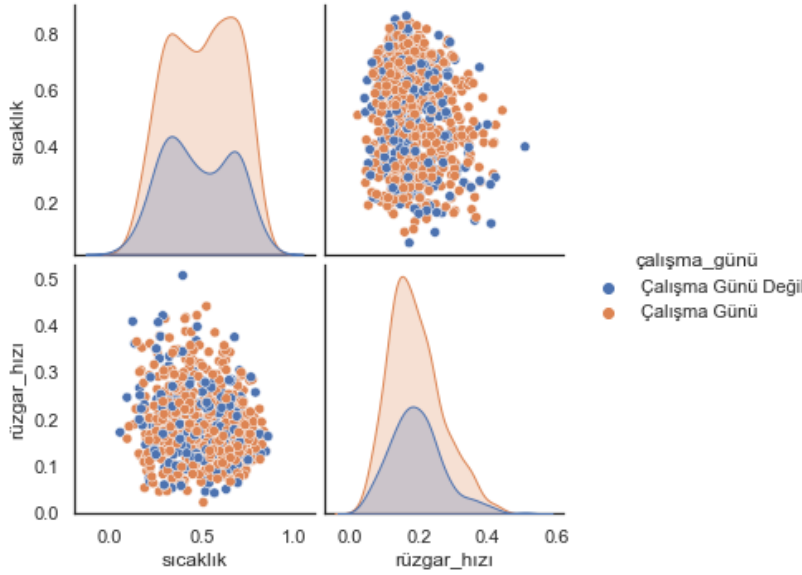
Her gün için ayların renkleri gösteriyor, bu da sıcaklıkların aylar arasında değiştiğini ve bu değişimlerin haftanın günlerine göre nasıl farklılaştığını gösteriyor.



### Pairplot: sıcaklık, rüzgar\_hızı ve çalışma\_günü

Sıcaklıkların haftanın günlerine göre değişiklik gösterdiğini, hafta ortasında Çarşamba, Perşembe gibi sıcaklıklar daha yüksek veya daha düşük olabilir. Bu durum, haftanın bazı günlerinde belirli aktivitelerin veya çevresel koşulların (örneğin iş günleri ve hafta sonları arasındaki farklılıklar) sıcaklık üzerinde etkisi olduğunu gösterebilir.

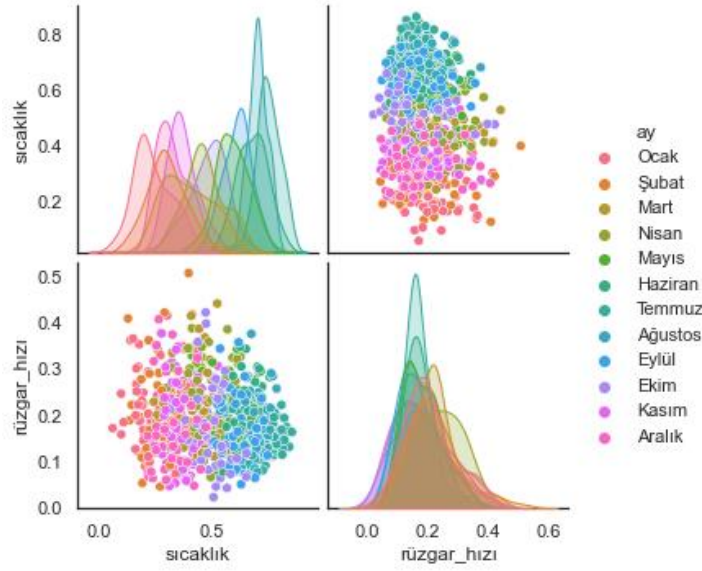
Her gün için ayların renkleri gösteriyor, bu da sıcaklıkların aylar arasında değiştiğini ve bu değişimlerin haftanın günlerine göre nasıl farklılaştığını gösteriyor.



### Pairplot: sıcaklık, rüzgar\_hızı ve ay

Sıcaklıkların haftanın günlerine göre değişiklik gösterdiğini, hafta ortasında Çarşamba, Perşembe gibi sıcaklıklar daha yüksek veya daha düşük olabilir. Bu durum, haftanın bazı günlerinde belirli aktivitelerin veya çevresel koşulların (örneğin iş günleri ve hafta sonları arasındaki farklılıklar) sıcaklık üzerinde etkisi olduğunu gösterebilir.

Her gün için ayların renkleri gösteriyor, bu da sıcaklıkların aylar arasında değiştiğini ve bu değişimlerin haftanın günlerine göre nasıl farklılaştığını gösteriyor.



## 4.3 BİR ÖRNEKLEMDEN BİLİNMEYEN MİKTARLARI TAHMİN ETME

### Kullanılan Kütüphaneler

**NumPy (import numpy as np):** Sayısal hesaplamalar yapmak için kullanılan temel kütüphane. Özellikle örneklem alma, ortalama hesaplama, standart sapma hesaplama gibi işlemler için kullanılmıştır.

**SciPy (import scipy.stats as stats):** İstatistiksel hesaplamalar için kullanılan kütüphane. norm.ppf() fonksiyonu gibi istatistiksel hesaplamalar ve güven aralığı hesaplamaları için kullanılmıştır.

**Matplotlib (import matplotlib.pyplot as plt):** Veri görselleştirmeleri için kullanılan kütüphane. Grafikler, histogramlar ve plotlar çizmek için kullanılmıştır.

**Seaborn (import seaborn as sns):** Matplotlib'in üstünde bir kütüphane olan Seaborn, görselleştirmeleri daha estetik hale getirmek için kullanılır. Örneklem dağılımını görselleştirmek amacıyla sns.histplot() fonksiyonu ile histogramlar oluşturulmuştur.

**Pandas (import pandas as pd):** Veri analizi için kullanılan kütüphane. Veri seti (veriSeti) üzerinde işlemler yapabilmek, popülasyon ortalaması ve standart sapmayı hesaplamak gibi işlemler için kullanılmıştır.



## Popülasyon ve Örneklem

**Popülasyon** tüm veri setini, yani tüm gözlemleri içerirken, **örneklem** bu veri setinin bir kısmıdır.

**Örneklem**, popülasyondan seçildiği için genellikle daha yönetilebilir ve analiz için daha uygun bir büyüklüktedir.

Belirli bir yıl (örneğin 2023) ve belirli bir mevsim (örneğin yaz) için hava durumu verileri.

Sadece tatil günlerindeki veriler (örneğin yalnızca 2024 yılı tatilleri) alınabilir.

Belirli bir haftanın günü (örneğin, sadece pazartesi günü) seçilerek bu günlere ait veriler analiz edilebilir.

Örneklem üzerinde yapılan analizlerle, popülasyon hakkında genellemeler yapılabilir, ancak genellemelerin doğruluğu örneklemin doğru şekilde seçilmesine bağlıdır.

## Basit Rastgele Örneklem Seçimi

Kullandığımız kod, veri setinizden **100 rastgele örnek seçer** ve bu örnekleri ekrana yazdırır. Bu işlem, **Basit Rastgele Örneklem** yöntemini kullanarak yapılır. Yani, her bir gözlem, veri setinden **eşit olasılıkla** seçilir. Bu sayede veri setindeki her bir gözlem aynı şansa sahip olur. Rastgele örneklem yöntemi, genellikle örneklem dağılımlarının çıkarılması, tahmin modelleri kurma ya da veri analizleri için kullanılır.

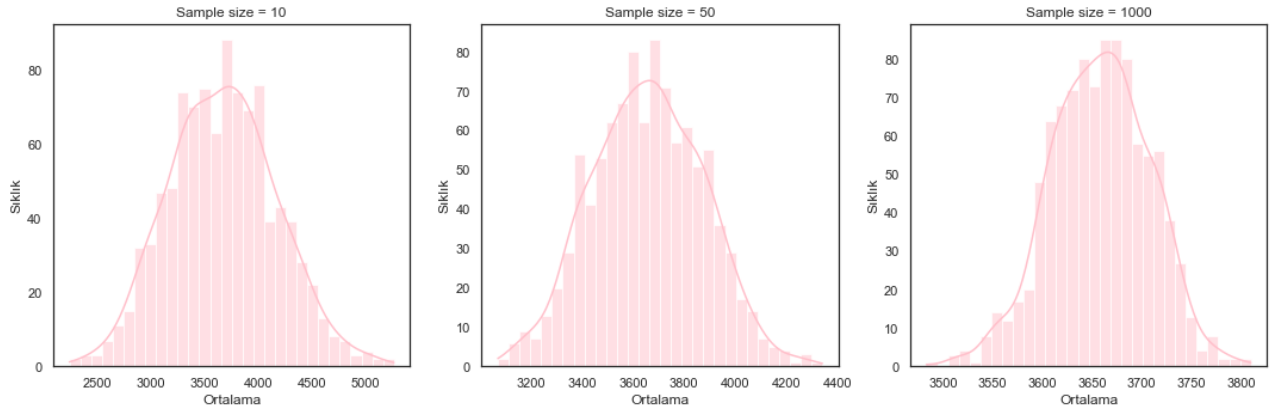


```
In [66]: rasgele_orneklem = veriSeti.sample(n=100, random_state=42)

In [67]: print(rasgele_orneklem)
   kayıt_dizini  tarih  ...  kayıtlı_kullanıcı  toplam_kullanıcı
703          704  2012-12-04  ...             6055             6606
33           34  2011-02-03  ...             1489             1550
300          301  2011-10-28  ...             3291             3747
456          457  2012-04-01  ...             3694             6041
633          634  2012-09-25  ...             6693             7538
..          ...  ...  ...  ...  ...
399          400  2012-02-04  ...             2448             2832
254          255  2011-09-12  ...             4023             4713
368          369  2012-01-04  ...             2273             2368
133          134  2011-05-14  ...             2507             3409
174          175  2011-06-24  ...             4022             4991
```

## Merkezi Limit Teoremi

Merkezi Limit Teoremi'nin temel fikri, örneklem büyüklüğü arttıkça, örneklem ortalamalarının dağılımının normal dağılıma yaklaşmasıdır. belirli örneklem büyüklükleri için (örneğin, 10, 50, 1000) **1000 deneme** gerçekleştirir ve her bir denemede örneklem ortalamalarını hesaplar. Bu işlem sonucunda elde edilen örneklem ortalamalarının dağılımları, üç farklı örneklem büyüklüğü için histogramlar şeklinde gösterilir.



Örneklem büyüklüğünün artırılmasının örneklem ortalamalarının dağılımı üzerindeki etkisini net bir şekilde ortaya koymaktadır. Küçük örneklem büyüklüklerinde (10) dağılım geniş ve düzensizdir, ancak örneklem büyüklüğü arttıkça (50 ve 1000), örneklem ortalamalarının dağılımı daralmakta ve normal dağılıma daha yakın bir hal almaktadır. Merkezi Limit Teoremi'nin etkisi, örneklem büyüklüğü büyüdükçe daha belirgin hale gelir. Bu durum, istatistiksel analizlerde büyük örneklem boyutlarının daha güvenilir sonuçlar ve daha doğru tahminler elde etmek için önemli olduğunu göstermektedir.

## Parametrelerin Tahmini ve Güven Aralığı

Popülasyon ortalaması, tüm veri setindeki değerlerin aritmetik ortalamasını temsil eder. Bu, veri seti hakkında genel bir bilgi verir ve tüm değerlerin ortalamasını belirler. Popülasyon standart sapması, veri setindeki değerlerin ortalama ne kadar sapma gösterdiğini ölçen bir parametredir. Yüksek bir standart sapma, verilerin geniş bir aralığa yayılmadığını gösterirken, düşük bir standart sapma ise verilerin daha dar bir aralıkta toplandığını gösterir.

```
In [98]: print(f"Popülasyon Ortalaması: {populasyon_ortalama}")
Popülasyon Ortalaması: 3656.172366621067

In [99]: print(f"Popülasyon Standart Sapması: {populasyon_std}")
Popülasyon Standart Sapması: 1560.2563770194527
```

Güven aralığı, popülasyon parametreleri hakkında bir tahmin aralığı sunar.

**Burada kullanılan formül:**

$$\text{Güven Aralığı} = \left[ \mu - Z_{\alpha/2} \times \frac{\sigma}{\sqrt{n}}, \mu + Z_{\alpha/2} \times \frac{\sigma}{\sqrt{n}} \right]$$

Hesaplanan **%95 Güven Aralığı** şu şekilde: [3543.07,3769.28]

Bu güven aralığı, popülasyon ortalamasının %95 olasılıkla bu değerler arasında olduğunu göstermektedir. Yani, bu aralıkta yer alan her değer popülasyon ortalaması olma olasılığı %95'tir.

```
In [107]:
...: margin_of_error = z_degeri * (populasyon_std / np.sqrt(n))
...: lower_limit = populasyon_ortalama - margin_of_error
...: upper_limit = populasyon_ortalama + margin_of_error
...:
...: print(f"%95 Güven Aralığı: [{lower_limit}, {upper_limit}]")
%95 Güven Aralığı: [3543.066438235805, 3769.278295006329]
```

## 4.4 İKİ ORTALAMAYI KARŞILAŞTIRMA

### Kullanılan Kütüphaneler

**StatsModels (from statsmodels.stats.weightstats):** Tek örneklem veya iki bağımsız grup için Z-testi gerçekleştirmek amacıyla kullanılır. Özellikle büyük örneklerde veya popülasyon varyansı bilindiğinde parametrik testlerde kullanımı uygundur. ztest fonksiyonu ile Z-testleri uygulanır.

**SciPy (import scipy.stats as stats):** İstatistiksel hesaplamalar için kullanılan bir kütüphane. Normallik kontrolü, parametrik ve parametrik olmayan testler gibi birçok istatistiksel analiz sağlar.

- `ttest_1samp`: Tek örneklem t-testi gerçekleştirir.
- `ttest_ind`: İki bağımsız grup arasında t-testi yapar. (Eşit varyans varsayımı altında veya olmadan kullanılabilir.)
- `ttest_rel`: Aynı grup üzerindeki iki durum için eşleştirilmiş t-testi yapar.
- `shapiro`: Verinin normal dağılıma uyup uymadığını kontrol eder.
- `mannwhitneyu`: Bağımsız iki grup arasında parametrik olmayan bir test gerçekleştirir.

**NumPy (import numpy as np):** Sayısal işlemler yapmak için kullanılan temel bir kütüphanedir. Ortalama farkı, standart sapma, karekök gibi işlemleri kolayca hesaplamayı sağlar.

- `np.mean()`: Ortalama hesaplama.
- `np.std()`: Standart sapma hesaplama.
- `np.sqrt()`: Karekök hesaplama.

### Tek Örneklem Z-Testi

Ortalama sıcaklığın popülasyon sıcaklık ortalamasına eşit olup olmadığını test etmek istiyoruz.

***H0 (Null Hipotezi): Örneklem ortalaması, popülasyon ortalamasına eşittir***

***H1 (Alternatif Hipotez): Örneklem ortalaması, popülasyon ortalamasından farklıdır***

- **Z-İstatistiği:** 36.24, P-Değeri: 1.24e-287
- **Yorum:** Sıcaklık ortalaması popülasyon ortalamasından anlamlı bir şekilde farklıdır ( $p < 0.05$ )

## Tek Örneklem T-Testi

Nem ortalamasının popülasyon nem ortalamasına eşit olup olmadığını test etmek.

*H0: Nem ortalaması , popülasyon nem ortalamasına eşittir*

*H1: Nem ortalaması popülasyon nem ortalamasından farklıdır*

- **T-İstatistiği:** 24.28, **P-Değeri:** 6.58e-96
- **Yorum:** Nem ortalaması popülasyon ortalamasından (0.50.50.5) anlamlı bir şekilde farklıdır ( $p < 0.05$ )

## Bağımsız Örneklem T-Testi (Student Testi)

Çalışma günlerinde (çalışma\_günü) ve tatil günlerinde (tatil) sıcaklık ortalamaları arasında fark var mı?

*H0: Sıcaklık ortalamaları eşittir*

*H1: Sıcaklık ortalamaları eşit değildir*

- **T-İstatistiği:** 1.42, **P-Değeri:** 0.155
- **Yorum:** Çalışma günleri ve tatil günlerindeki sıcaklık ortalamaları arasında anlamlı bir fark yoktur ( $p > 0.05$ )

## Bağımsız Örneklem T-Testi (Welch Testi)

Aynı senaryo (14.3), ancak grupların varyanslarının eşit olmadığı varsayımıyla.

- **T-İstatistiği:** 1.41, **P-Değeri:** 0.161
- **Yorum:** Eş varyans varsayımı yapılmadığında da gruplar arasında anlamlı bir fark bulunmamıştır ( $p > 0.05$ )

## Eşleştirilmiş Örneklem T-Testi

Belirli bir dönemin öncesi ve sonrası sıcaklık değişimlerini karşılaştırıyoruz (örneğin, 2011-01-01 öncesi ve sonrası).

- **T-İstatistiği:** 0.68, **P-Değeri:** 0.508
- **Yorum:** 2011-01-15 öncesi ve sonrası sıcaklıklar arasında anlamlı bir fark bulunmamaktadır ( $p > 0.05$ )

## Tek Yönlü Testler (One-Sided Tests)

Tatil günlerinde sıcaklık ortalamasının çalışma günlerinden daha yüksek olduğunu test etmek istiyoruz.

*H0: Tatil günlerinde sıcaklık  $\leq$  Çalışma günü sıcaklığı.*

*H1: Tatil günlerinde sıcaklık  $>$  Çalışma günü sıcaklığı.*

- **T-İstatistiği:** 1.42, **P-Değeri:** 0.077
- **Yorum:** Tatil günlerinde sıcaklığın çalışma günlerinden daha yüksek olduğu yönündeki tek taraflı hipotez desteklenmemektedir ( $p > 0.05$ )

## Etki Büyüklüğü (Effect Size)

Çalışma ve tatil günlerindeki sıcaklık ortalamaları arasındaki farkın büyüklüğünü ölçmek.

- **Cohen's d:** 0.113
- **Yorum:** Çalışma ve tatil günleri arasındaki sıcaklık farkı çok küçük bir etki büyüklüğüne sahiptir ( $d < 0.2$ )

## Normallik Kontrolü

Sıcaklık değerlerinin normal dağılıma uyup uymadığını kontrol etmek.

- **Shapiro-Wilk Testi:** 0.97, **P-Değeri:** 5.15e-12
- **Yorum:** Sıcaklık verisi normal dağılıma uymamaktadır ( $p < 0.05$ )

## Normale Uymayan Veriler için Wilcoxon Testleri

Çalışma ve tatil günlerindeki sıcaklık ortalamalarını parametrik olmayan bir testle karşılaştırmak.

- **Mann-Whitney U Testi:** 61512.0, **P-Değeri:** 0.156
- **Yorum:** Çalışma ve tatil günlerindeki sıcaklıklar arasında anlamlı bir fark bulunmamaktadır ( $p > 0.05$ )

## 5. SONUÇ

Bu proje, bisiklet paylaşım sistemleri verisinin analizi için istatistiksel yöntemlerin nasıl uygulanabileceğini göstermeyi amaçlamaktadır. Çalışma kapsamında tanımlayıcı istatistiklerden grafiksel analizlere, örnekleme yöntemlerinden hipotez testlerine kadar çeşitli analitik yaklaşımlar incelenmiştir.

Veri seti üzerinde yapılan analizler, bisiklet paylaşım sistemlerinin kullanımını etkileyen faktörleri belirlemek ve kullanıcı davranışlarını anlamak için önemli ipuçları sağlamıştır. Ayrıca, hipotez testleri ile elde edilen sonuçlar, popülasyon parametreleri hakkında güvenilir tahminler yapılmasına olanak tanımıştır.

Grafik çizimi bölümünde ise veri görselleştirme teknikleri kullanılmıştır. Bu görselleştirmeler, hem veri setinin yapısını daha iyi anlamaya hem de çıkarımları desteklemeye olanak tanımıştır.

Bir örneklemden bilinmeyen miktarların tahmini bölümü, istatistiksel yöntemlerin uygulamalı bir şekilde ele alındığı önemli bir aşamadır. Bu bölümde, merkezi limit teoremi kullanılarak örneklem ve popülasyon ilişkisi incelenmiş, parametre tahmini ve güven aralıkları hesaplanmıştır. Özellikle güven aralıkları, popülasyonun genel özelliklerini örneklem yoluyla tahmin etme konusunda güçlü bir araç olarak kullanılmıştır.

İki ortalamayı karşılaştırma bölümünde ise farklı hipotez testleri uygulanmıştır. Tek örneklem z-testi ve t-testi, bağımsız ve eşleştirilmiş örneklem t-testleri gibi yöntemlerle, belirli koşullar altında değişkenlerin anlamlı farklılıklar gösterip göstermediği değerlendirilmiştir. Ayrıca normallik kontrolleri ve normallik varsayımına uymayan durumlar için Wilcoxon testleri gibi parametrik olmayan testler de gerçekleştirilmiştir.

Sonuç olarak, bu proje bisiklet paylaşım sistemlerinin kullanımını etkileyen faktörlerin daha iyi anlaşılmasına yönelik somut veriler ve çıkarımlar sunmuştur. Ayrıca, analiz sürecinde kullanılan yöntemler ve araçlar, istatistiksel ve veri bilimi projelerinde sistematik bir yaklaşımın önemini ortaya koymaktadır.

## 6. Kaynakça

*Builtin.* (2024). <https://builtin.com/data-science/t-test-python> adresinden alındı

*CAMIS.* (2024). [https://psiaims.github.io/CAMIS/python/two\\_samples\\_t\\_test.html](https://psiaims.github.io/CAMIS/python/two_samples_t_test.html) adresinden alındı

*ChatGPT.* (2024). <https://chatgpt.com/> adresinden alındı

*ethanweed.* (2024). <https://ethanweed.github.io/pythonbook/landingpage.html> adresinden alındı

*GitHub.* (2024). <https://github.com/> adresinden alındı