

İçindekiler Tablosu

<i>Kullanılan Veri Seti:</i>	4
<i>Değişkenler ve Türleri:</i>	4
<i>Kullanım Amaçları:</i>	5
BÖLÜM 5. BETİMLEYİCİ İSTATİSTİKLER	6
5.1 Merkezi Eğilim Ölçüleri	7
5.1.1 Aritmetik Ortalama	7
5.1.2 Python'da Ortalamanın Hesaplanması	7
5.1.3 Ortanca (Medyan, Median)	8
5.1.4 Ortalama mı yoksa Medyan mı?	8
5.1.5 Gerçek Hayattan Bir Örnek	8
5.1.6 Kırılmış Ortalama	8
5.1.7 Mod	9
5.2. Değişkenlik Ölçüleri	10
5.2.1 Değişim Aralığı (Range)	11
5.2.2 Çeyreklik Aralığı	12
5.2.3 Ortalamadan Mutlak Sapma	13
5.2.4 Varyans	14
5.2.5. Standart Sapma	14
5.2.6 Medyandan Mutlak Sapma	15
5.2.7 Hangi Ölçüyü Kullanmalıyız?	15
5.3 Çarpıklık ve Basıklık	16
5.4 Bir Değişkenin Genel Özétini Alma	19
5.4.1 Bir Değişkeni Tanımak	19
5.4.2 Bir Veri Çerçeveini Tanımlamak	19
5.5 Standart Puanlar	19
5.6 Korelasyonlar	20
5.6.1 Korelasyon Katsayıtı	20
5.6.2 Python'da Korelasyonun Hesaplanması	21
BÖLÜM 12. HİPOTEZ TESTİ	23
12.1 Hipotezler Topluluğu	23
12.1.1 Araştırma Hipotezleri ve İstatistiksel Hipotezler	23
12.1.2 Yokluk Hipotezi ve Alternatif Hipotez	25

12.2 İki Tip Hata	27
12.3 Test İstatistikleri ve Örnekleme Dağılımları	27
12.4 Karar Alma	30
12.4.1 Kritik Bölgeler ve Kritik Değerler	30
12.4.2 Tek Yanlı ve İki Yanlı Testler Arasındaki Fark	30
12.5 <i>p-değeri</i>	31
12.5.1 Karar Alma Süreci	31
12.5.2. Yaygın Bir Hata	32
12.6 Etki Büyüklüğü, Örneklem Büyüklüğü ve Güç	33
12.6.1 Etki Büyüklüğü	33
12.6.2 Güç Fonksiyonu	35
12.6.3 Çalışmanın Gücünü Artırmak	35
BÖLÜM 13 KATEGORİK VERİ ANALİZİ	36
13.1 Ki-Kare Uyum İyiliği Testi:	36
13.1.1. Spor Salonu Verileri:.....	36
13.1.2. Sıfır Hipotezi ve Alternatif Hipotez:	37
13.1.3. “Uyumun İyiliği” Test İstatistiği:	38
13.1.4. Gof İstatistiğinin Örnekleme Dağılımı (Gelişmiş):	39
13.1.5. Sıfır Hipotezini Test Etme:	39
13.1.6. Python'da Test Yapmak:.....	40
13.1.8. Test Sonuçlarının Nasıl Raporlanacağı:	44
13.1.9. İstatistiksel Gösterime İlişkin Bir Yorum (Gelişmiş):	44
13.2. χ^2 Bağımsızlık (Veya İlişki) Testi:	44
13.2.1. Hipotez Testimizi Oluşturma:	46
13.2.2. Python'da Test Yapmak:.....	47
13.3. Sürekliklilik Düzeltmesi:	52
13.4. Etki Büyüklüğü:	52
13.5. Testlerin Varsayımları:	52
13.6. Fisher Kesin Testi:	53
13.7. McNemar Testi:	56
13.7.1. Python'da McNemar Testinin Yapılması:	56
KAYNAKÇA VE GÖREV DAĞILIM ÇİZELGESİ:.....	59

Kullanılan Veri Seti:

Değişkenler ve Türleri:

1. **Yas (int)**: Kişiin yaşı. Analizlerde yaş gruplarını karşılaştırmak için kullanılabilir.
2. **Cinsiyet (object)**: Kişiin cinsiyeti. Kategorik bir değişken; demografik analizler için uygundur.
3. **Kilo (kg) (float)**: Kişiin kilogram cinsinden kilosu. Sağlık, fitness ve BMI hesaplamalarında kullanılabilir.
4. **Boy(m) (float)**: Kişiin metre cinsinden boyu. BMI hesaplaması ve antropometrik analizler için uygundur.
5. **Max_BPM (int)**: Egzersiz sırasında kaydedilen maksimum kalp atış hızı. Egzersiz yoğunluğu analizleri için kullanılabilir.
6. **Ort_BPM (int)**: Egzersiz sırasında kaydedilen ortalama kalp atış hızı. Fitness seviyesini ve egzersiz etkisini değerlendirmede yararlı.
7. **Dinlenme_BPM (int)**: Kişiin dinlenme sırasında ölçülen kalp atış hızı. Kardiyovasküler sağlık için önemli bir göstergedir.
8. **Oturum_Süresi(saat) (float)**: Egzersiz oturumunun süresi (saat olarak). Egzersiz alışkanlıklarının değerlendirilmesinde kullanılır.
9. **Yakilan_Kalori (int)**: Egzersiz sırasında yakılan kalori miktarı. Enerji harcamasını ölçmek için kullanılabilir.
10. **Egzersiz_Türü (object)**: Egzersiz türü (ör. Yoga, HIIT). Egzersiz tercihlerini ve etkilerini analiz etmek için uygun.
11. **Yag_Yüzdesi (float)**: Kişiin vücut yağ yüzdesi. Fiziksel sağlık değerlendirmeleri için kritik bir ölçü.
12. **Tüketilen_Su(Litre) (float)**: Egzersiz sırasında tüketilen su miktarı. Hidratasyon alışkanlıklarını analiz etmek için kullanılabilir.
13. **Antreman_Sýklýgý (gün/hafta) (int)**: Haftada kaç gün egzersiz yapıldığı. Egzersiz düzeninin değerlendirilmesinde kullanılabilir.
14. **Deneyim_Seviyesi (int)**: Kişiin egzersiz deneyim seviyesi (ör. 1: Başlangıç, 3: İleri seviye). Egzersiz alışkanlıkları ile ilişkili analizlerde önemli.
15. **BKE (float)**: Beden Kitle Endeksi (BMI). Sağlık ve obezite analizlerinde kritik bir değişken.

Kullanım Amaçları:

- **Sağlık ve Fitness Analizi:** Kilo, boy, BKE ve yağ yüzdesi gibi değişkenler sağlık değerlendirmelerinde kullanılabilir.
- **Demografik Karşılaştırmalar:** Yaş ve cinsiyet ile egzersiz alışkanlıkları karşılaştırılabilir.
- **Egzersiz Alışkanlıkları:** Egzersiz süresi, sıklığı ve türü gibi değişkenlerle bireylerin spor yapma alışkanlıkları incelenebilir.

BÖLÜM 5. BETİMLEYİCİ İSTATİSTİKLER

İncelemek için bir veri seti aldığımızda veriyi anlaşılabilir şekilde özetlememiz gereklidir. Bu özetlemeyi yapmak için betimleyici istatistikleri kullanırız. Betimleyici istatistik; örneklem ya da kitleden türetilen veri kümelerinin analizi, özeti ve sunumudur. Bu bölümde betimleyici istatistikler incelenecaktır.

*Betimleyici istatistikleri daha iyi anlamak için yükleyeceğimiz veri setini **data** ismiyle tanımladık. Bu veri setini yüklemek için **Pandas** kütüphanesinden yararlanıyoruz. Pandas veri işleme ve analizi için Python üzerine inşa edilmiş açık kaynaklı bir yazılım kütüphanesidir. Pandas, popüler programlama diline elektronik tablo benzeri verilerle çalışma yeteneği kazandırır.*

```
( import pandas as pd  
data = pd.read_csv("istyazilim_odevi/gym_members_exercise_tracking.csv")
```

*Bir spor salonu üyelerinin egzersizlerinden oluşan veri seti. Verinin ilk beş satırını gözlemlmek için **head()** ve son beş satırını gözlemlmek için **tail()** fonksiyonlarını kullanıyoruz.*

6

```
( data.head()  
data.tail()
```

	Age	Gender	Weight (kg)	Height (m)	Max_BPM	Avg_BPM	Resting_BPM	Session_Duration (hours)	Calories_Burned	Workout_Type	Fat_Percentage	Water_Intake (liters)	Workout_Frequency (days/week)	Experience_Level	BMI
0	56	Male	88.3	1.71	180	157	60	1.69	1313.0	Yoga	12.6	3.5	4	3	30.20
1	46	Female	74.9	1.53	179	151	66	1.30	883.0	HIIT	33.9	2.1	4	2	32.00
2	32	Female	68.1	1.66	167	122	54	1.11	677.0	Cardio	33.4	2.3	4	2	24.71
3	25	Male	53.2	1.70	190	164	56	0.59	532.0	Strength	28.8	2.1	3	1	18.41
4	38	Male	46.1	1.79	188	158	68	0.64	556.0	Strength	29.2	2.8	3	1	14.39

	Age	Gender	Weight (kg)	Height (m)	Max_BPM	Avg_BPM	Resting_BPM	Session_Duration (hours)	Calories_Burned	Workout_Type	Fat_Percentage	Water_Intake (liters)	Workout_Frequency (days/week)	Experience_Level	BMI
968	24	Male	87.1	1.74	187	158	67	1.57	1364.0	Strength	10.0	3.5	4	3	28.77
969	25	Male	66.6	1.61	184	166	56	1.38	1260.0	Strength	25.0	3.0	2	1	25.69
970	59	Female	60.4	1.76	194	120	53	1.72	929.0	Cardio	18.8	2.7	5	3	19.50
971	32	Male	126.4	1.83	198	146	62	1.10	883.0	HIIT	28.2	2.1	3	2	37.74
972	46	Male	88.7	1.63	166	146	66	0.75	542.0	Strength	28.8	3.5	2	1	33.38

Sütun başlıklarları spor salonundaki üyelerden alınan bilgilerin içeriğini gösteriyor. Sütunlar değişkenlerimizdir. Bunlar sırasıyla yaş, cinsiyet, kilo, boy, egzersiz sırasında dakikada maksimum kalp atış hızı, egzersiz sırasında dakikada ortalama kalp atış hızı, üyenin egzersize başlamadan önceki (dinlenme halindeki) kalp atış hızı, egzersiz seansı sırasında harcanan toplam süre (saat), antrenman sırasında yakılan kalori miktarı, yapılan egzersiz türü, üyelerin yağ yüzdesi, gün içerisinde tüketilen su miktarı, haftada kaç antrenman seansı yapacağı, üyenin deneyim seviyesi ve vücut kitle indeksidir.

973 üyenin tamamının verilerine tek tek bakabiliyoruz fakat verilere bakmak onu anlamak için yeterli bir yol değildir. Veriler hakkında daha fazla fikir sahibi olmak için betimleyici istatistikler üzerinden ilerleyeceğiz.

5.1 Merkezi Eğilim Ölçüleri

Merkezi eğilim ölçüsü, bir veri kümesindeki merkezi konumu belirleyerek bir veri kümesi tanımlamaya çalışan tek bir değerdir. Ortalama, medyan ve mod, hepsi geçerli merkezi eğilim ölçüleridir, ancak farklı koşullar altında, bazı merkezi eğilim ölçülerini diğerlerinden daha uygun hale gelir.

5.1.1 Aritmetik Ortalama

Veri setindeki tüm değerlerin, içindeki gözlem sayısına bölünmesiyle ortalama hesaplanır. Hem kesikli hem de sürekli verilerde kullanılabilir. \bar{x} , örneklem ortalamasını temsil ediyorken; μ , kitle ortalamasını temsil eder.

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} \quad \mu = \frac{\sum x}{n}$$

5.1.2 Python'da Ortalamanın Hesaplanması

Python'da ortalamayı tipki bir hesap makinesinde yapıyor gibi tüm değerleri toplayıp gözlem sayısına bölebiliriz fakat bu işlem büyük bir veri setiyle çalıştığımızdan kolaylık sağlamaz. Bundan dolayı python'in pandas kütüphanesindeki fonksiyonları kullanırız.

Ortalama hesaplaması için python'da `mean()` fonksiyonunu kullanabiliriz fakat bize daha gözlemlenebilir bir tablo veren ve diğer tanımlayıcı istatistikleri de hesaplayan `describe()` fonksiyonunu kullandık.

Numeric_data adıyla sayısal değişkenleri seçtik, alttaki satırda ise sayısal değişkenlerin betimleyici istatistiklerini hesaplayacak kodu yazıp descriptive_statistics ismiyle atama yaptık.

```
( numeric_data = data.select_dtypes(include=['number'])  
descriptive_statistics = numeric_data.describe()
```

5.1.3 Ortanca (Medyan, Median)

Medyan, büyülüklük sırasına göre düzenlenmiş bir veri kümесinin orta puanıdır. Tek sayılı bir veri kümесini iki eşit parçaya ayıran değer iken, çift sayılı bir veri kümeseinde ortada kalan iki değerin aritmetik ortalamasıdır.

Veri setimizin ortancasını hesaplamak için aşağıdaki kodu kullandık.

```
( descriptive_statistics.loc['Medyan'] = numeric_data.median()
```

5.1.4 Ortalama mı yoksa Medyan mı?

Ortalama, veri kümесinin bir nevi ağırlık merkezi gibidir, medyan ise verilerdeki orta değerdir. Ortalama aykırı değerlere karşı duyarlıken, medyan aykırı değerlere duyarlı değildir. Ortalama normal dağılım için uygulanabilirken, medyan çarpık dağılımlar için uygulanabilir. Bu ifadelerden hangisi veri setimiz için daha anlamlı ise onu tercih etmeliyiz.

8

5.1.5 Gerçek Hayattan Bir Örnek

Gerçek bir veri seti ile ilgiliğimizde ortalama mı yoksa medyan mı seçeceğimize nasıl karar vereceğimizi kendi veri setimizde inceleyelim. Verimizde aykırı değerler yok. Verimiz büyük olduğu için merkezi limit teoremindeki normal dağılığını söyleyebiliriz. Spor salonu üyelerinin egzersizleri verisi hakkında yorum yapacak olursak ortalamayı seçmemiz daha doğru bir tercihtir.

5.1.6 Kırılmış Ortalama

Kırılmış ortalama, dağılımin her iki ucundan belirli bir yüzdelik üç değer çıkarıldıkten sonra bir veri kümесinin ortalamasını hesaplayan istatistiksel bir ölçüddür. Aykırı değerleri hariç tutarak, bu istatistik bir veri kümесinin tipik veya merkezi değerlerinin daha doğru bir temsilini sağlayabilir. Genellikle, %10 veya %20 gibi değer yüzdeleri kırılır. Kırılmış ortalama, aykırı değerlerden kurtulup sağlam bir merkezî eğilim ölçüsü elde etmemize yardımcı olur. %0 kırılmış ortalama sadece normal ortalamadır. %50 kırılmış ortalama ise medyandır.

Kırılmış ortalamayı bulmak için sırasıyla; veri seti sıralanır, kırılmak üzere değerlerin yüzdesi belirlenir, kırılmak üzere gözlem sayısı hesaplanır, veri seti kırılır ve kırılmış ortalama hesaplanır. Elimizdeki veri setinin %5 ve %10 olacak şekilde kırılmış ortalamalarını python'da hesaplayalım.

```
trimmed_means_10 = numeric_data.apply(lambda col: stats.trim_mean(col, 0.1))
trimmed_means_5 = numeric_data.apply(lambda col: stats.trim_mean(col, 0.05))
trimmed_means_df = pd.DataFrame({'%10 Kırılmış Ortalama': trimmed_means_10,
                                  '%5 Kırılmış Ortalama': trimmed_means_5})

print("\nKırılmış Ortalamalar:")
print(trimmed_means_df)

Kırılmış Ortalamalar:
          %10 Kırılmış Ortalama  %5 Kırılmış Ortalama
Yas           38.792041           38.724059
Kilo (kg)      71.853530           72.763740
Boy(m)         1.717125           1.719795
Max_BPM        179.939666          179.916762
Ort_BPM        143.616175          143.710376
Dinlenme_BPM    62.287548           62.250855
Oturum_Suresi(saat)  1.255379           1.256568
Yakilan_Kalori     898.763800          899.834664
Yag_Yuzdesi       25.460591           25.221209
Tuketilen_Su(Litre)  2.632734           2.631813
Antreman_Sikligi (gun/hafta)  3.277279           3.302166
Deneyim_Seviyesi     1.762516           1.789054
BKE              24.425404           24.559122
```

9

5.1.7 Mod

Mod, veri kümesinde en sık görülen değerdir. Modu bazı durumlarda en popüler seçenek olarak düşünebiliriz. Hiçbir modun olmaması, bir modun olması veya birden fazla modun olması mümkündür. Modu bulmak için veri kümesini sayısal veya kategorik olarak sıraladığımızda ve en sık oluşan yanıttır. Mod, nominal ölçüm seviyesindeki veriler için en uygun olanıdır. Nominal veriler karşılıklı olarak birbirini dışlayan kategorilere ayrıılır, bu nedenle mod en popüler kategoriyi söyler. Sürekli değişkenler veya ölçüm oranı düzeyleri için mod, merkezi eğilimin yararlı bir ölçüsü olmayabilir. Bir değerin ölçüm oranı düzeyinde tekrar etmesi olası değildir.

Elimizdeki veri setinin modunu hesaplamak için mode() fonksiyonunu kullanırız.

```

mod_values = numeric_data.mode().iloc[0]
descriptive_statistics.loc['Mod'] = mod_values

```

5.2. Değişkenlik Ölçüleri

Merkezi eğilim ölçülerini hangi değerin ortada veya popüler olduğunu ölçerken, dağılım ölçülerinde verilerin değişkenliğini ölçeriz. Veriler ne kadar yayılmış? Gözlemlenen değerler ortalamadan veya medyandan ne kadar uzak? Sorularının cevaplarını ararız.

Şimdiye kadar degindigimiz tüm başlıkların python'da çıktısını yazdırma için print() fonksiyonunu kullandık.

```

print("\nMerkezi Eğilim Ölçüleri ve Dağılım:")
print(descriptive_statistics)

```

Merkezi Eğilim Ölçüleri ve Dağılım:					
	Yas	Kilo (kg)	Boy(m)	Max_BPM	Ort_BPM
count	973.000000	973.000000	973.000000	973.000000	973.000000
mean	38.683453	73.854676	1.722580	179.883864	143.766701
std	12.180928	21.207500	0.127720	11.525686	14.345101
min	18.000000	40.000000	1.500000	160.000000	120.000000
25%	28.000000	58.100000	1.620000	170.000000	131.000000
50%	40.000000	70.000000	1.710000	180.000000	143.000000
75%	49.000000	86.000000	1.800000	190.000000	156.000000
max	59.000000	129.900000	2.000000	199.000000	169.000000
Varyans	148.375004	449.758077	0.016312	132.841437	205.781935
Medyan	40.000000	70.000000	1.710000	180.000000	143.000000
MAD	10.566815	16.971618	0.105898	9.982356	12.473965
Mod	43.000000	57.700000	1.620000	198.000000	132.000000

Dinlenme_BPM Oturum_Suresi(saat) Yakilan_Kalori Yag_Yuzdesi				
	Dinlenme_BPM	Oturum_Suresi(saat)	Yakilan_Kalori	Yag_Yuzdesi
count	973.000000	973.000000	973.000000	973.000000
mean	62.223022	1.256423	905.422405	24.976773
std	7.327060	0.343033	272.641516	6.259419
min	50.000000	0.500000	303.000000	10.000000
25%	56.000000	1.040000	720.000000	21.300000
50%	62.000000	1.260000	893.000000	26.200000
75%	68.000000	1.460000	1076.000000	29.300000
max	74.000000	2.000000	1783.000000	35.000000
Varyans	53.685807	0.117672	74333.396493	39.180324
Medyan	62.000000	1.260000	893.000000	26.200000
MAD	6.326026	0.270896	215.843011	5.038284
Mod	50.000000	1.030000	883.000000	28.100000

	Tuketilen_Su(Litre)	Antreman_Sikligi (gun/hafta)	Deneyim_Seviyesi \
count	973.000000	973.000000	973.000000
mean	2.626619	3.321686	1.809866
std	0.600172	0.913047	0.739693
min	1.500000	2.000000	1.000000
25%	2.200000	3.000000	1.000000
50%	2.600000	3.000000	2.000000
75%	3.100000	4.000000	2.000000
max	3.700000	5.000000	3.000000
Varyans	0.360206	0.833655	0.547145
Medyan	2.600000	3.000000	2.000000
MAD	0.500273	0.778525	0.625919
Mod	3.500000	3.000000	2.000000
	BKE		
count	973.000000		
mean	24.912127		
std	6.660879		
min	12.320000		
25%	20.110000		
50%	24.160000		
75%	28.560000		
max	49.840000		
Varyans	44.367314		
Medyan	24.160000		
MAD	5.171984		
Mod	23.530000		

5.2.1 Değişim Aralığı (Range)

Değişim aralığı, bir veri kümесinin maksimum ve minimum değerleri arasındaki farktır. Maksimum, veri kümесindeki en büyük değer; minimum ise, en küçük değerdir. Değişim aralığı hesaplaması kolaydır ancak aşırı değerlerden çok etkilenir.

11

`min()` ve `max()` fonksiyonları ile değişkenlerimizin maksimum ve minimum değerlerini elde edebiliriz. Fakat Bölüm 5.2 'de `describe()` fonksiyonu ile verimizin çıktısından minimum ve maksimum değerlerine zaten ulaşmıştık. Dolayısıyla elde ettiğimiz verileri bu çıktı üzerinden yorumladığımızda sayısal değişkenlerimizin değişim aralığını aşağıdaki gibidir:

- Age (yaş) değişkenine baktığımızda spor salonu üyelerinin maksimum 59, minimum 18 yaşında olduğunu görüyoruz. Değişim aralığı ise $59 - 18 = 41$ 'dir.
- Weight (kilo) değişkenine baktığımızda spor salonu üyelerinin maksimum 129, minimum 40 kilo olduğunu söyleriz. Değişim aralığı ise $129 - 40 = 89$ 'dur.
- Height (boy) değişkenine baktığımızda spor salonu üyelerinin maksimum 200 santimetre, minimum 150 santimetre olduğunu görüyoruz. Değişim aralığı ise $200 - 150 = 50$ santimetredir.
- Max BPM (dakikada maksimum kalp atış hızı) değişkenine baktığımızda spor salonu üyelerinin dakikada maksimum kalp atış hızı 199, minimum ise 160 olduğunu görüyoruz. Değişim aralığı ise $199 - 160 = 39$ 'dur.
- Avg BPM (dakikada ortalama kalp atış hızı) değişkenine baktığımızda spor salonu üyelerinin maksimum dakikada ortalama kalp atış hızlarının 169, minimum ise 120 olduğunu görüyoruz. Değişim aralığı ise $169 - 120 = 49$ 'dur.

- Resting BPM değişkenine baktığımızda spor salonu üyelerinin dinlenme esnasında maksimum kalp atış hızı 74, minimum ise 50 olduğunu görüyoruz. Değişim aralığı ise $74 - 50 = 24$ 'tır.
- Session Duration değişkenine baktığımızda spor salonu üyelerinin maksimum oturum süresi 2 saat, minimum ise 0.5 saat olduğunu görüyoruz. Değişim aralığı ise $2 - 0.5 = 1.5$ saatdir.
- Calories Burned değişkenine baktığımızda spor salonu üyelerinin maksimum 1783 kalori yaktığını, minimum ise 303 olduğunu görüyoruz. Değişim aralığı ise $1783 - 303 = 1480$ 'dır.
- Fat Percentage değişkenine baktığımızda spor salonu üyelerinin maksimum yağ yüzdesinin 35, minimum ise 10 olduğunu görüyoruz. Değişim aralığı ise $35 - 10 = 25$ 'dir.
- Water Intake değişkenine baktığımızda spor salonu üyelerinin maksimum 3.70 litre su tüketliğini, minimum ise 1.50 litre olduğunu görüyoruz. Değişim aralığı ise $3.70 - 1.50 = 2.20$ litredir.
- Workout Frequency değişkenine baktığımızda spor salonu üyelerinin maksimum antrenman sıklığı 5 seansken, minimum ise 2 seans olduğunu görüyoruz. Değişim aralığı ise $5 - 2 = 3$ 'tür.
- Experience Level değişkenine baktığımızda spor salonu üyelerinin maksimum deneyim seviyesi 3 iken, minimum ise 1 olduğunu görüyoruz. Değişim aralığı ise $3 - 1 = 2$ 'dir.
- BMI değişkenine baktığımızda spor salonu üyelerinin maksimum vücut kitle indeksi 49.84 çıkmışken, minimum ise 12.32 olduğunu görüyoruz. Değişim aralığı ise $49.84 - 12.32 = 37.52$ 'dir.

12

5.2.2 Çeyreklik Aralığı

Çeyreklik aralığı, bir dağılımda ortadaki %50 puanlarının aralığıdır. En büyük ve en küçük değer arasındaki farkı hesaplamak yerine, %25 ile %75 arasındaki farkı hesaplar.

`quantile()` fonksiyonu ile verilerin çeyrekliğini hesaplayabiliriz. Fakat Bölüm 5.2 'de `describe()` fonksiyonu çıktısı ile verimizin %25 ve %75 değerlerine ulaşmıştık. Dolayısıyla elde ettiğimiz verileri bu çıktı üzerinden yorumladığımızda sırasıyla değişkenlerimizin çeyreklik aralıkları aşağıdaki gibidir.

- Age (yaş) değişkenine baktığımızda spor salonu üyelerinin çeyreklik aralığı 24'tür.
- Weight (kilo) değişkenine baktığımızda spor salonu üyelerinin çeyreklik aralığı 27.9'dur.
- Height (boy) değişkenine baktığımızda spor salonu üyelerinin çeyreklik aralığı 1.18' dir.

- Max BPM (dakikada maksimum kalp atış hızı) değişkenine baktığımızda spor salonu üyelerinin dakikada maksimum kalp atış hızı çeyreklik aralığı 20'dir.
- Avg BPM (dakikada ortalama kalp atış hızı) değişkenine baktığımızda spor salonu üyelerinin dakikada ortalama kalp atış hızlarının çeyreklik aralığı 25'tir.
- Resting BPM değişkenine baktığımızda spor salonu üyelerinin dinlenme esnasında kalp atış hızı çeyreklik aralığı 12'dir.
- Session Duration değişkenine baktığımızda spor salonu üyelerinin oturum süresi çeyreklik aralığı 0.42'dir.
- Calories Burned değişkenine baktığımızda spor salonu üyelerinin yaktığı kalori miktarının çeyreklik aralığı 356'dır.
- Fat Percentage değişkenine baktığımızda spor salonu üyelerinin yağ yüzdesinin çeyreklik aralığı 8'dir.
- Water Intake değişkenine baktığımızda spor salonu üyelerinin tükettiği su miktarının çeyreklik aralığı 0.9'dur.
- Workout Frequency değişkenine baktığımızda spor salonu üyelerinin antrenman sikliğinin çeyreklik aralığı 1'dir.
- Experience Level değişkenine baktığımızda spor salonu üyelerinin deneyim seviyesinin çeyreklik aralığı 1'dir.
- BMI değişkenine baktığımızda spor salonu üyelerinin viçut kitle indekslerinin çeyreklik aralığı 8.45'tir.

13

5.2.3 Ortalamadan Mutlak Sapma

Ortalamadan mutlak sapma, gözlemler ile ortalamaları arasındaki ortalama mesafeyi belirten bir değişkenlik ölçüsüdür. Daha büyük değerler, veri noktalarının ortalamadan daha uzağa yayıldığını gösterir. Tersine, daha düşük değerler, veri noktalarının ona daha yakın bir şekilde kümelenmesine karşılık gelir. Ortalamadan mutlak sapmanın bu tanımı, standart sapmaya benzer geliyor. Her ikisi de değişkenliği ölçse de farklı hesaplamaları vardır.

$$\frac{1}{n} \sum_{i=1}^n |x_i - m(X)|$$

5.2.4 Varyans

Varyans, puanların ortalamadan karesel farkının ortalamasıdır. Bir popülasyondaki varyansı hesaplamak için:

1. Ortalama hesaplanır.
2. Ortalama değerden sapmayı hesaplamak için her değerden ortalama çıkarılır.
3. Her sapma değerinin karesini alınır.
4. Toplami bulmak için sapma değerlerinin kareleri toplanır.
5. Toplami değer sayısına bölünür.

$$\sigma^2 = \frac{\sum(x - \mu)^2}{N} \quad S^2 = \frac{\sum(x - \bar{x})^2}{N - 1}$$

σ^2 değeri kitle varyansının formülüünü gösterirken, S^2 örneklem varyansının formülüünü gösterir. Aşağıdaki kod dizisi ile veri setimizin `var()` fonksiyonu ile varyansını hesapladık. Bu kodun çıktısı Bölüm 5.2'nin altındadır.

```
( descriptive_statistics.loc['Varyans'] = numeric_data.var() )
```

5.2.5. Standart Sapma

14

Standart sapma varyanstan türetilir ve ortalama olarak her değerin ortalamadan ne kadar uzakta olduğunu söyler. Varyansın kareköküdür ve dağılım normal veya yaklaşıktır olarak normal olduğunda değişkenliğin kullanışlı bir ölçüüsüdür.

Küçük bir standart sapma katsayısı, küçük bir değişkenlik derecesini (yani değerlerin birbirine yakın olduğunu) gösterir; daha büyük standart sapma katsayıları ise büyük değişkenliği (yani değerlerin birbirinden uzak olduğunu) gösterir.

Standart sapmanın formülü aşağıdaki gibidir.

$$s = \sqrt{\frac{\sum(x - \bar{x})^2}{n - 1}}$$

Spor salonu üyelerinin bilgilerinden oluşan veri setindeki değişkenlerimizin standart sapmalarını Bölüm 5.2'deki kod dizisi ile elde ettiğimiz çıktıdaki `std` sütunu ile gözlemleyebiliriz. Aynı zamanda `stdev()` fonksiyonuyla da standart sapmayı `python`'da hesaplayabiliriz.

Örneğin; üyelerin kilolarının standart sapması 0.127720 iken, viçut kitle indekslerinin standart sapmaları 6.660879'dur.

	count	mean	std	min	25%	50%	75%	max
Age	973.0	38.683453	12.180928	18.00	28.00	40.00	49.00	59.00
Weight (kg)	973.0	73.854676	21.207500	40.00	58.10	70.00	86.00	129.90
Height (m)	973.0	1.722580	0.127720	1.50	1.62	1.71	1.80	2.00
Max_BPM	973.0	179.883864	11.525686	160.00	170.00	180.00	190.00	199.00
Avg_BPM	973.0	143.766701	14.345101	120.00	131.00	143.00	156.00	169.00
Resting_BPM	973.0	62.223022	7.327060	50.00	56.00	62.00	68.00	74.00
Session_Duration (hours)	973.0	1.256423	0.343033	0.50	1.04	1.26	1.46	2.00
Calories_Burned	973.0	905.422405	272.641516	303.00	720.00	893.00	1076.00	1783.00
Fat_Percentage	973.0	24.976773	6.259419	10.00	21.30	26.20	29.30	35.00
Water_Intake (liters)	973.0	2.626619	0.600172	1.50	2.20	2.60	3.10	3.70
Workout_Frequency (days/week)	973.0	3.321686	0.913047	2.00	3.00	3.00	4.00	5.00
Experience_Level	973.0	1.809866	0.739693	1.00	1.00	2.00	2.00	3.00
BMI	973.0	24.912127	6.660879	12.32	20.11	24.16	28.56	49.84

5.2.6 Medyandan Mutlak Sapma

15

Verilerin medyanlarından mutlak sapmalarının medyanıdır. Her veri noktasının tüm veri noktalarının medyanından uzaklığı bulunur (işaretleri göz ardı ederek) ve ardından bulunan bu değerlerin medyanı alınır.

Elimizdeki veri setinin medyandan sapmalarını bulmak için aşağıdaki kod dizisini kullandık. Bölüm 5.2'de çıktısını MAD isimli satır ile gözlemleyebiliriz.

```
( descriptive_statistics.loc['MAD'] = numeric_data.apply(lambda x:(x-x.mean()).abs().mean()) )
```

5.2.7 Hangi Ölçüyü Kullanmalıyız?

- *Değişim aralığı; bize verilerin genel yayılımını verir. Aykırı değerlere karşı hassastır. Standart sapması küçük olan verilerde daha doğru yorumlanabilir.*
- *Ceyreklik aralığı; verinin orta yarısının nerede olduğunu söyler. Medyani tamamlar.*
- *Ortalama mutlak sapma; gözlemlerin ortalamadan ortalama olarak ne kadar uzakta olduğunu söyler. Standart sapmaya göre hatalı sonuçlar verebilir. Çok tercih edilmez.*

- *Varyans; ortalamadan ortalama karesel sapmayı söyler. Ortalamanın etrafındaki varyasyonu tanımlamanın "doğru" yoludur ancak verilerle aynı birimleri kullanmadığı için tamamen yorumlanamaz.*
- *Standart sapma; varyansın kareköküdür. Matematiksel olarak verilerle aynı birimlerde ifade edilir, bu nedenle oldukça iyi yorumlanabilir. Ortalamanın merkezi eğilimin ölçüsü olduğu durumlarda, bu varsayılan değerdir.*
- *Ortanca mutlak sapma; ortanca değerden sapma. Ham haliyle basit ve yorumlanabilirdir. Düzeltilmiş haliyle bazı veri kümeleri için standart sapmayı tahmin etmenin iyi bir yoludur.*

5.3 Çarpıklık ve Basıklık

Bir dağılım veya veri kümesi, merkez noktasının solunda ve sağında aynı görünüyorrsa simetriktir.

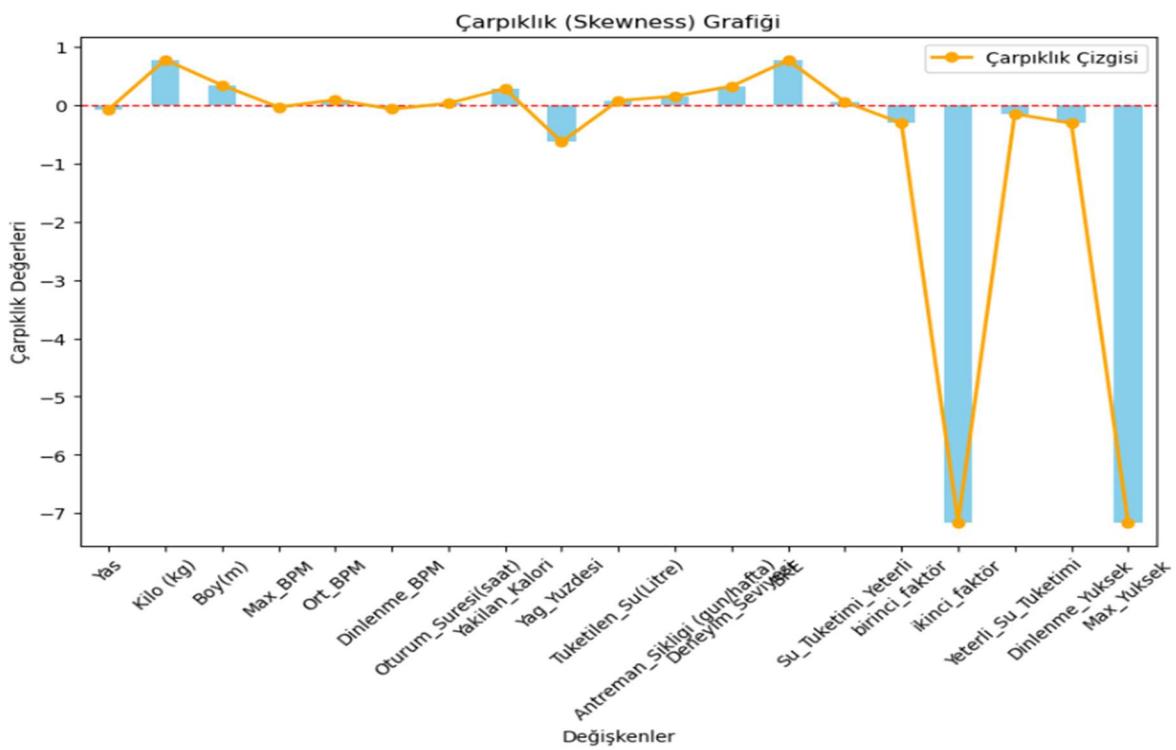
Çarpıklık, simetri eksikliğidir. Uzun kuyruğu pozitif yönde uzanan bir dağılımin pozitif çarpıklığa sahip olduğu söylenir yani sağa çarpiktir. Uzun kuyruğu sola doğru uzanan bir dağılım negatif çarpiktir, yani sola doğru çarpiktir. Normal dağılım için çarpıklık sıfırdır ve herhangi bir simetrik verinin çarpıklığı sıfıra yakın olmalıdır. Veriler çok modluysa, bu çarpıklığın işaretini etkileyebilir. Çarpıklık formülü:

16

$$\tilde{\mu} = \frac{\sum_{i=1}^N (x_i - \bar{x})^3}{(N - 1) \cdot \sigma^3}$$

```
skewness = data.skew(numeric_only=True)
print("Çarpıklık (Skewness) Değerleri:")
print(skewness)

Çarpıklık (Skewness) Değerleri:
Yas                      -0.077864
Kilo (kg)                 0.772384
Boy(m)                   0.338858
Max_BPM                  -0.037950
Ort_BPM                   0.086361
Dinlenme_BPM              -0.071636
Oturum_Suresi(saat)       0.025761
Yakilan_Kalori            0.278321
Yag_Yuzdesi               -0.635225
Tuketilen_Su(Litre)       0.071480
Antreman_Sikligi (gun/hafta) 0.149815
Deneyim_Seviyesi          0.318513
BKE                       0.763648
Su_Tuketimi_Yeterli        0.051484
birinci_faktör            -0.314672
ikinci_faktör             -7.157675
Yeterli_Su_Tuketimi        -0.154861
Dinlenme_Yuksek            -0.314672
Max_Yuksek                -7.157675
dtype: float64
```



Basıklık, bir dağılımin kuyruklarının normal bir dağılımin kuyruklarından ne kadar farklı olduğunu tanımlayan istatistiksel bir ölçütür. Başka bir deyişle belirli bir dağılımin kuyruklarının uç değerler içerip içermediğini belirler.

17

- *Orta basıklığa sahip dağılımlar orta kalınlıkta kuyruklara,*
- *Düşük basıklığa sahip dağılımlar ince kuyruklara,*
- *Yüksek basıklığa sahip dağılımlar kalın kuyruklara sahiptir.*

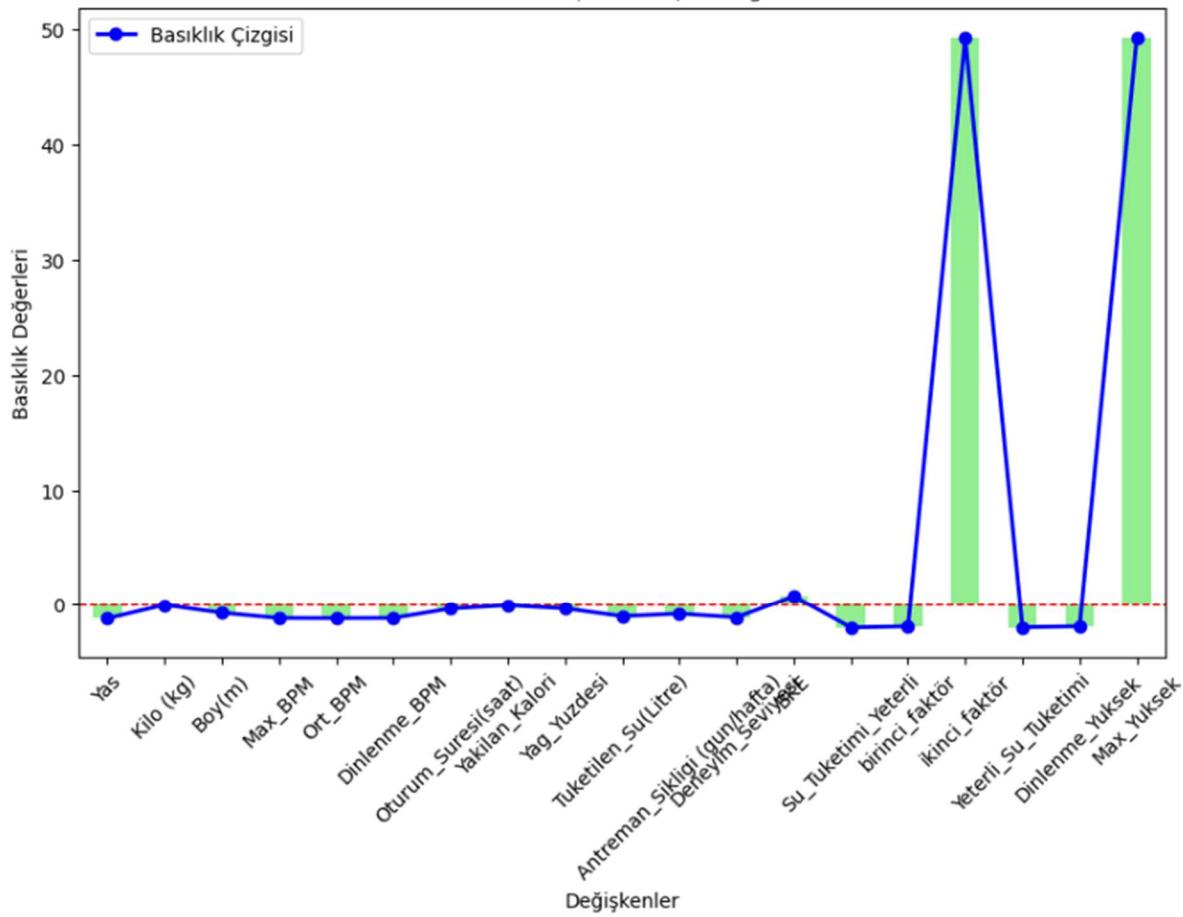
$$kurt(x) = \frac{\sum_{i=1}^N (x_i - \bar{x})^4}{N \cdot \sigma^4} - 3$$

```
( kurtosis = data.kurtosis(numeric_only=True)
  print("Basıklık (Kurtosis) Değerleri:")
  print(kurtosis)
```

Basıklık (Kurtosis) Değerleri:

Yas	-1.215076
Kilo (kg)	-0.023969
Boy(m)	-0.723633
Max_BPM	-1.187916
Ort_BPM	-1.198724
Dinlenme_BPM	-1.181466
Oturum_Suresi(saat)	-0.350805
Yakilan_Kalori	-0.056050
Yag_Yuzdesi	-0.339019
Tuketilen_Su(Litre)	-1.020298
Antreman_Sikligi (gun/hafta)	-0.806101
Deneyim_Seviyesi	-1.123111
BKE	0.743240
Su_Tuketimi_Yeterli	-2.001468
birinci_faktör	-1.904901
ikinci_faktör	49.333713
Yeterli_Su_Tuketimi	-1.980092
Dinlenme_Yuksek	-1.904901
Max_Yuksek	49.333713
dtype: float64	

Basıklık (Kurtosis) Grafiği



5.4 Bir Değişkenin Genel Özeti Alma

Bölümün bu kısmına kadar farklı fonksiyonlarla değişkenlerin ortalamalarını, medyanlarını, standart sapmalarını, çeyrekliklerini, maksimum ve minimum değerlerinin nasıl hesaplanacağından bahsettiğimizde. Sadece belirli bir betimleyici istatistiğe ihtiyacımız olduğunda bu fonksiyonların kullanımı bize kolaylık sağlayabilir fakat betimleyici istatistiklerin tamamı için genel bir gösterim ariyorsak `describe()` fonksiyonu tipki Bölüm 5.1.2'deki kod dizisinde olduğu gibi bize tüm değişkenler için betimleyici istatistikleri özetler.

5.4.1 Bir Değişkeni Tanımak

Tek bir değişkene ait veri çerçevesini gözlemelemek için aşağıdaki kod dizisini kullanırız. `data` isimli veri setimizden `age` isimli yaş değişkeninin betimleyici istatistiklerini hesaplarız. Bu hesaplamayı `age_describe` ismiyle atadık ve ekrana çıktısını yazdırmak için `print()` fonksiyonunu kullandık.

```
( age_describe = data[“Age”].describe()  
print(age_describe)
```

5.4.2 Bir Veri Çerçevesini Tanımlamak

Bir değişkenin genel özeti alma bölümünde `describe()` fonksiyonunu kullandık. Bize tüm değişkenler hakkında bilgi verir ancak bazı durumlarda veri çerçevesi için sınırlı kalabilir burada `include = “all”` ifadesini kullanarak bu sınırlılığı ortadan kaldırılmaya çalışıyoruz. Bu sayede değişkenlerin sikliklarına, tepe değerlerine ulaşabiliyoruz.

```
( data.describe(include = “all”)
```

5.5 Standart Puanlar

Standart puan ya da z puanı olarak adlandırılır. Bir değerin normal dağılım içinde oluşu olasılığını hesaplamayı ve farklı normal dağılımlardan gelen iki değeri karşılaştırmayı sağlar. Standart puan bunu, normal dağılımdaki puanları standart normal dağılım haline gelen z puanlarına dönüştürerek başka bir deyişle, standartlaştırarak yapar. Bir frekans dağılımı normal dağılımlı olduğunda, standart puanlar (veya z puanları) olarak bilinen puanları standartlaştırarak bir puanın oluşma olasılığını bulabiliyoruz. Frekans dağılımındaki veri grubunu, ortalama 0 ve standart sapma 1 olacak şekilde basitçe

dönüştürür. z puanlarının, ortalaması 0 ve standart sapması 1 olan bir dağılımı vardır bu dağılıma standart normal dağılım denir. Aşağıdaki formül ile standart puanlar hesaplanır.

$$z = \frac{x - \mu}{\sigma}$$

Bir veri kümesinin python'da standart puanını hesaplamak için **scipy** kütüphanesinden **stats** bölümünü içe aktarmak gereklidir. `data` isimli veriyi **stats.zscore(data)** ile z puanlarını hesaplar.

```
from scipy.stats import zscore  
stats.zscore(data)
```

5.6 Korelasyonlar

Korelasyon, iki değişkenin doğrusal olarak ilişkili olma derecesini ifade eden istatistiksel bir ölçütür yani sabit bir oranda birlikte değişirler. Neden ve sonuç hakkında bir açıklama yapmadan basit ilişkileri tanımlamak için yaygın bir araçtır. Korelasyonlar istatistiksel anlamlılık açısından da test edilir.

5.6.1 Korelasyon Katsayısı

Korelasyonlar, -1 ile $+1$ arasında değişen ve r ile gösterilen **korelasyon katsayısı** adı verilen birimsiz bir ölçü ile tanımlanır. Korelasyon katsayısı r , ilişkinin gücünü niceliksel olarak belirler. Korelasyon katsayısı hakkında:

- r sıfıra ne kadar yakınsa doğrusal ilişki o kadar zayıftır.
- Pozitif r değerleri pozitif korelasyonu gösterir, yani her iki değişkenin değerleri birlikte artma eğilimindedir.
- Negatif r değerleri negatif korelasyonu gösterir; yani bir değişkenin değerleri azalırken diğer değişkenin değerleri artma eğilimindedir.

söylediğimiz gibi hesaplanır

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Bir korelasyon matrisi, farklı değişkenler için korelasyon katsayılarını gösteren basit bir tablodur. Matris, bir tablodaki tüm olası değer çiftleri arasındaki korelasyonu tasvir eder. Büyük bir veri setini özetlemek ve verilen verilerdeki kalıpları belirlemek ve görselleştirmek için güçlü bir araçtır. Bir korelasyon matrisi, değişkenleri gösteren satır ve sütunlardan oluşur. Tablodaki her hücre korelasyon katsayısını içerir.

Spor salonu üyelerinin veri setinde korelasyon değerlerini korelasyon matrisi biçiminde inceleyeceğiz.

5.6.2 Python'da Korelasyonun Hesaplanması

Python'da korelasyon hesaplaması corr() ifadesi ile yapılabilir. Aşağıda veri setimizin korelasyon matrisinin kod dizisini ve çıktısını inceleyelim.

```
correlation_matrix = numeric_data.corr()  
print("\nKorelasyon Matrisi:")  
print(correlation_matrix)
```

Korelasyon Matrisi:

	Yas	Kilo (kg)	Boy(m)	Max_BPM	\
Yas	1.000000	-0.036340	-0.027837	-0.017073	
Kilo (kg)	-0.036340	1.000000	0.365321	0.057061	
Boy(m)	-0.027837	0.365321	1.000000	-0.017660	
Max_BPM	-0.017073	0.057061	-0.017660	1.000000	
Ort_BPM	0.035969	0.009717	-0.014776	-0.039751	
Dinlenme_BPM	0.004354	-0.032138	-0.005090	0.036647	
Oturum_Suresi(saat)	-0.019912	-0.013666	-0.010206	0.010051	
Yakilan_Kalori	-0.154679	0.095443	0.086348	0.002090	
Yag_Yuzdesi	0.002370	-0.225512	-0.235521	-0.009056	
Tuketilen_Su(Litre)	0.041528	0.394276	0.393533	0.031621	
Antreman_Sikligi (gun/hafta)	0.008055	-0.011769	-0.011270	-0.029099	
Deneyim_Seviyesi	-0.018676	0.003379	-0.010267	0.000545	
BKE	-0.013691	0.853158	-0.159469	0.067105	
	Ort_BPM	Dinlenme_BPM	Oturum_Suresi(saat)		\
Yas	0.035969	0.004354		-0.019912	
Kilo (kg)	0.009717	-0.032138		-0.013666	
Boy(m)	-0.014776	-0.005090		-0.010206	
Max_BPM	-0.039751	0.036647		0.010051	
Ort_BPM	1.000000	0.059636		0.016014	
Dinlenme_BPM	0.059636	1.000000		-0.016649	
Oturum_Suresi(saat)	0.016014	-0.016649		1.000000	
Yakilan_Kalori	0.339659	0.016518		0.908140	
Yag_Yuzdesi	-0.007302	-0.016834		-0.581520	
Tuketilen_Su(Litre)	-0.002911	0.007726		0.283411	
Antreman_Sikligi (gun/hafta)	-0.010681	-0.007967		0.644140	
Deneyim_Seviyesi	-0.000888	0.001758		0.764768	
BKE	0.021605	-0.032543		-0.006493	

	<i>Yakilan_Kalori</i>	<i>Yag_Yuzdesi</i>	\
<i>Yas</i>	-0.154679	0.002370	
<i>Kilo (kg)</i>	0.095443	-0.225512	
<i>Boy(m)</i>	0.086348	-0.235521	
<i>Max_BPM</i>	0.002090	-0.009056	
<i>Ort_BPM</i>	0.339659	-0.007302	
<i>Dinlenme_BPM</i>	0.016518	-0.016834	
<i>Oturum_Suresi(saat)</i>	0.908140	-0.581520	
<i>Yakilan_Kalori</i>	1.000000	-0.597615	
<i>Yag_Yuzdesi</i>	-0.597615	1.000000	
<i>Tuketilen_Su(Litre)</i>	0.356931	-0.588683	
<i>Antreman_Sikligi (gun/hafta)</i>	0.576150	-0.537060	
<i>Deneyim_Seviyesi</i>	0.694129	-0.654363	
<i>BKE</i>	0.059761	-0.119258	

	<i>Tuketilen_Su(Litre)</i>	\
<i>Yas</i>	0.041528	
<i>Kilo (kg)</i>	0.394276	
<i>Boy(m)</i>	0.393533	
<i>Max_BPM</i>	0.031621	
<i>Ort_BPM</i>	-0.002911	
<i>Dinlenme_BPM</i>	0.007726	
<i>Oturum_Suresi(saat)</i>	0.283411	
<i>Yakilan_Kalori</i>	0.356931	
<i>Yag_Yuzdesi</i>	-0.588683	
<i>Tuketilen_Su(Litre)</i>	1.000000	
<i>Antreman_Sikligi (gun/hafta)</i>	0.238563	
<i>Deneyim_Seviyesi</i>	0.304104	
<i>BKE</i>	0.213697	

	<i>Antreman_Sikligi (gun/hafta)</i>	<i>Deneyim_Seviyesi</i>	\
<i>Yas</i>	0.008055	-0.018676	
<i>Kilo (kg)</i>	-0.011769	0.003379	
<i>Boy(m)</i>	-0.011270	-0.010267	
<i>Max_BPM</i>	-0.029099	0.000545	
<i>Ort_BPM</i>	-0.010681	-0.000888	
<i>Dinlenme_BPM</i>	-0.007967	0.001758	
<i>Oturum_Suresi(saat)</i>	0.644140	0.764768	
<i>Yakilan_Kalori</i>	0.576150	0.694129	
<i>Yag_Yuzdesi</i>	-0.537060	-0.654363	
<i>Tuketilen_Su(Litre)</i>	0.238563	0.304104	
<i>Antreman_Sikligi (gun/hafta)</i>	1.000000	0.837079	
<i>Deneyim_Seviyesi</i>	0.837079	1.000000	
<i>BKE</i>	0.001645	0.016031	

	<i>BKE</i>
<i>Yas</i>	-0.013691
<i>Kilo (kg)</i>	0.853158
<i>Boy(m)</i>	-0.159469
<i>Max_BPM</i>	0.067105
<i>Ort_BPM</i>	0.021605
<i>Dinlenme_BPM</i>	-0.032543
<i>Oturum_Suresi(saat)</i>	-0.006493
<i>Yakilan_Kalori</i>	0.059761
<i>Yag_Yuzdesi</i>	-0.119258
<i>Tuketilen_Su(Litre)</i>	0.213697
<i>Antreman_Sikligi (gun/hafta)</i>	0.001645
<i>Deneyim_Seviyesi</i>	0.016031
<i>BKE</i>	1.000000

BÖLÜM 12. HİPOTEZ TESTİ

Bir hipotez, belirli koşullar altında bir şeyin neden gerçekleştiğini veya ne olabileceğini açıklamayı amaçlar. Ayrıca farklı değişkenlerin birbirleriyle nasıl ilişkili olduğunu anlamak için de kullanılabilir. Bir hipotez sadece bir tahmin değildir mevcut teorilere ve bilgiye dayanmalıdır. Ayrıca test edilebilir olmalıdır, yani bilimsel araştırma yöntemleri (deneyler, gözlemler ve verilerin istatistiksel analizi gibi) aracılığıyla onu destekleyebilir veya çürütebiliriz.

Hipotez testi, bir popülasyon hakkında sonuçlar çıkarmak için bir örneklemden alınan verileri kullanan bir istatistiksel çıkarım biçimidir. Aynı zamanda hipotezde belirtilen bir varsayımlı test etmenin istatistiksel bir yoludur. Hipotez testinde birden fazla parametrenin karşılaştırılacağı durumda değişkenlerin bazı özellikleri sağlanması gereklidir. Bu özellikler:

- *Ana kütleye alınan birimlerin birbirlerinden bağımsızlığı,*
- *Ana kütlenin normal dağılıma sahip olmasıdır.*

12.1 Hipotezler Topluluğu

12.1.1 Araştırma Hipotezleri ve İstatistiksel Hipotezler

Araştırma hipotezleri, araştırma sorusuna önerilen cevaptır. Araştırma hipotezi genellikle bir açıklama içerir. Örnek verecek olursak;

- Yapılan spor türüyle cinsiyet birbirleriyle ilişkilidir.*
- Kahvaltı yapan öğrenciler, kahvaltı yapamayan öğrencilere göre derse daha iyi odaklıdır.*
- Yüksek şekerle beslenen ve hareketsiz aktivite düzeyine sahip kişilerde depresyon gelişme olasılığı daha yüksektir.*
- Müzik dinlemek, diğer şeylere dikkat etme yeteneğini azaltır.*

Örneklerden de görüleceği üzere araştırma hipotezi sayısal iddialardan çok ilişkisel iddialarla ilgilenir.

İstatistiksel hipotezler, bir popülasyon parametresilarındaki matematiksel ifadedir. İstatistiksel hipotezler her zaman çiftler halinde gelir. Bunlar; sıfır ve alternatif hipotezlerdir. İyi tasarlanmış bir çalışmada, istatistiksel hipotezler mantıksal olarak araştırma hipotezine karşılık gelir. İstatistiksel hipotezlere veri setimiz üzerinden örnek verecek olursak;

- Cinsiyete göre yakılan ortalama kalori miktarları arasında fark yoktur.*

- ii. 73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark yoktur.
- iii. 30 yaşından küçük üyelerin ortalama antrenman sıklığı, 30 yaşından büyük üyelerin ortalama antrenman sıklığından daha fazladır.

İstatistiksel hipotezleri incelediğimizde ise ortalama gibi parametreler üzerinden kurulduğunu gördük.

Bölüm 12 başlığında istatistiksel hipotez örneklerini spor salonu üyelerinden alınan bilgilerle hazırlanmış veri setini python ile bölüm alt başlıkları altında inceleyeceğiz.

```
import pandas as pd

data = pd.read_csv("istyazilim_odevi/gym_members_exercise_tracking.csv")
```

- Kullanacağımız veri setini python'a pandas kütüphanesi ile yükliyoruz.
- Veri setini "data" ismi ile python'a atadık.

Pandas, veri işleme ve analizi için Python üzerine inşa edilmiş açık kaynaklı bir yazılım kütüphanesidir. Pandas, popüler programlama diline elektronik tablo benzeri verilerle çalışma yeteneği kazandırır.

24

Araştırmaya geçmeden önce veri yapısı hakkında bilgi sahibi olmak için info() fonksiyonunu kullanıyoruz. Bu işlemi "veri_yapisi" olarak tanımladık.

```
veri_yapisi = data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 973 entries, 0 to 972
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Age              973 non-null    int64  
 1   Gender            973 non-null    object  
 2   Weight (kg)       973 non-null    float64 
 3   Height (m)        973 non-null    float64 
 4   Max_BPM           973 non-null    int64  
 5   Avg_BPM            973 non-null    int64  
 6   Resting_BPM        973 non-null    int64  
 7   Session_Duration (hours) 973 non-null    float64 
 8   Calories_Burned     973 non-null    float64 
 9   Workout_Type         973 non-null    object  
 10  Fat_Percentage      973 non-null    float64 
 11  Water_Intake (liters) 973 non-null    float64 
 12  Workout_Frequency (days/week) 973 non-null    int64  
 13  Experience_Level      973 non-null    int64  
 14  BMI                 973 non-null    float64 

dtypes: float64(7), int64(6), object(2)
memory usage: 114.2+ KB
```

- Veri setinde toplamda 15 değişken mevcut. Bu değişkenlerden 7 tanesinin float (sürekli), 6 tanesinin integer (kesikli) ve 2 tanesinin object (kategorik) olduğunu gözlemliyoruz. Non-null ifadesi değişkenlerin her birinin içinde eksik gözlem değeri olmadığını gösterir. Eksik değerlerimizin olmaması analiz yaparken ekstra bir işlem uygulamamız gerektiğini gösterir.

12.1.2 Yokluk Hipotezi ve Alternatif Hipotez

Yokluk hipotezi, sıfır hipotezi olarak da bilinir. H_0 olarak gösterilir. Sıfır hiçbir şey anlamına gelir. Bu hipotez, gruplar arasında hiçbir fark olmadığını veya değişkenler arasında hiçbir ilişki olmadığını belirtir. Test edilmesi gereken istatistiksel hipotezdir. Bölüm 12.1.1'de verilen istatistiksel hipotez örnekleri yokluk hipotezleri gösterir. Bu yokluk hipotezlerini matematiksel olarak gösterimlerini aşağıdaki gibidir.

- Cinsiyete göre yakılan ortalama kalori miktarları arasında fark yoktur.
 μ_K : Kadın üyelerin yaktıkları ortalama kalori miktarı
 μ_E : Erkek üyelerin yaktıkları ortalama kalori miktarını olarak gösterdiğimizde yokluk hipotezi;

$H_0: \mu_K - \mu_E = 0$ şeklindedir. Bu hipotezi python'da aşağıdaki gibi yaptık.

- Cinsiyet birer kategorik değişken olduğundan daha doğru sonuçlar elde etmek için erkek üyelere 0, kadın üyelere ise 1 değerlerini verdik. `unique()` ile yaptığımız işlemin doğruluğunu test ettik. Çıktı ise işlemin doğruluğunu gösteriyor.

```
data["Gender"] = data["Gender"].map({"Male": 0, "Female": 1})
print(data["Gender"].unique())
[0 1]
```

- Kadın ve erkek üyeleri yaktıkları kalori miktarlarına göre iki gruba ayırdık.

```
male_calories = data[data["Gender"] == 0]["Calories_Burned"]
female_calories = data[data["Gender"] == 1]["Calories_Burned"]
```

- 73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark yoktur.
 μ_1 : 73'ten az kiloya sahip spor salonu üyelerinin ortalama oturum süreleri
 μ_2 : 73'ten fazla kiloya sahip spor salonu üyelerinin ortalama oturum süreleri olarak gösterildiğinde yokluk hipotezi;

$H_0: \mu_1 - \mu_2 = 0$ şeklindedir.

- Kilo ve oturum süresi değişkenleri sayısal değerlerden oluşan ekstra bir işlem yapmadan 73'ten az kiloya sahip üyeleri group1, 73'ten fazla kiloya sahip üyeleri group2 olarak ikiye ayırdık.

```
group1 = data[data["Weight (kg)"] < 73]["Session_Duration (hours)"]
group2 = data[data["Weight (kg)"] >= 73]["Session_Duration (hours)"]
```

- iii. 30 yaşından küçük üyelerin ortalama antrenman sıklığı, 30 yaşından büyük üyelerin ortalama antrenman sıklığından daha fazladır fakat bu bir iddia olduğu için bunu alternatif hipotez olarak kabul ederiz. Bu durumda yokluk hipotezini: “30 yaşından küçük üyelerin ortalama antrenman sıklığı ile 30 yaşından büyük üyelerin ortalama antrenman sıklığından fark yoktur.” olarak kurarız.
 μ_A : 30 yaşından küçük üyelerin ortalama antrenman sıklığı
 μ_B : 30 yaşından büyük üyelerin ortalama antrenman sıklığı olarak gösterildiğinde yokluk hipotezi;

$H_0: \mu_A - \mu_B = 0$ şeklindedir.

26

- 30 yaşından küçük olan üyeleri “kucuk_yas”, 30 yaşından büyük olacak üyeleri “buyuk_yas” olacak şekilde iki gruba ayırdık.

```
kucuk_yas = data[data["Age"] < 30]["Workout_Frequency (days/week)"]
buyuk_yas = data[data["Age"] >= 30]["Workout_Frequency (days/week)"]
```

Alternatif hipotez, seçenek hipotezi olarak da bilinir. Aynı zamanda iddia edilendir. Hipotez testi aslında bir nevi karşılaştırma ve seçim işlemi olduğu için birden fazla hipoteze ihtiyaç duyulur. Bu hipotezler alternatif hipotezlerdir. Yokluk hipotezi adı altında yazdığımız tüm hipotezler test edilebilmek için alternatif hipoteze ihtiyaç duyur. Bölüm 12.1.1'deki hipotezlerin alternatif hipotezleri sırasıyla aşağıdaki gibidir:

- i. $H_1: \mu_K - \mu_E \neq 0, \quad (\mu_K \neq \mu_E)$
- ii. $H_1: \mu_1 - \mu_2 \neq 0, \quad (\mu_1 \neq \mu_2)$
- iii. $H_1: \mu_A - \mu_B > 0, \quad (\mu_A > \mu_B)$

olarak gösterilirler.

12.2 İki Tip Hata

Hipotez testini hiçbir zaman hata yapmayacak şekilde kurmak isteriz fakat bu mümkün değildir. İstatistiksel hipotezlerde iki tip hata vardır bunlar 1.tip hata ve 2.tip hata olarak geçer.

1.Tip hata, diğer adıyla anlamlılık düzeyi. Genellikle α sembolü ile gösterilir. H_0 yani yokluk hipotezi doğru iken reddedildiğinde 1.tip hata ortaya çıkar. Araştırmalarda bu durum kontrol altına alınmak istenir bu nedenle uygulamada genellikle $\alpha = 0.05, 0.01, 0.10$ değerlerinden biri olarak kabul edilir.

2.Tip hata, genellikle β sembolü ile gösterilir. H_0 hipotezi yanlış iken doğru kabul edildiğinde ortaya 2.tip hata çıkar.

Aşağıdaki tablo ile 1.tip ve 2.tip hataları daha net bir şekilde gözlemleneyebiliriz.

HİPOTEZ

KARAR	Kitledeki Durum	
Test Sonucu	H_0 doğru	H_0 yanlış
H_0 kabul	Karar doğru	II. tip hata(β)
H_0 ret	I.tip hata(α)	Karar doğru

Spor salonu üyelerinin veri seti için 1.tip hata yani $\alpha = 0.05$ olarak kabul ettik.

12.3 Test İstatistikleri ve Örneklemme Dağılımları

Yokluk ve alternatif hipotezlerimizi belirledikten sonra yokluk hipotezini kabul etmek mi yoksa reddetmek mi arasında bir karar vermemiz gereklidir. Bu karara rehberlik etmesi için yaptığımız hesaplamaya **test istatistiği** denir. İstatistiksel hipotezlerde z-testi ve t-testi kullanılır. Bu testlerden z-testi standart normal dağılımdan gelirken t-testi t dağılımdan gelir. Bu testlerden hangisini kullanacağımız ise elimizdeki veri setine bağlıdır.

Standart normal dağılım, ortalaması 0, standart sapması 1 olan dağılıma denir. Normal dağılımdan türetilmiştir. Normal dağılan verilerin bir dizi standartlaşma işleminden geçirilerek standartlaştırılmıştır. Büyüük bir veriye sahipsek (Bu büyülük $n > 30$ olarak kabul edilir.) ve standart sapma hakkında iyi bir tahminimiz varsa z-testi kullanılır. Çünkü verinin büyümesiyle elimizdeki dağılım normal dağılıma yakınsar. Bu yakınsama istatistikte **merkezi limit teoremi** olarak bilinir.

T dağılımı, normal dağılıma sahip fakat örneklem büyüklüğü küçükse ($n < 30$) ve standart sapma hakkında iyi bir tahmine sahip olmadığımız durumlarda kullanılır. T-testi ise bu örneklenin ($n-1$) serbestlik derecesiyle hesaplanmasıdır.

Elimizdeki veri seti üzerine konuştuğumuzda her hipotezde 973 veriyi kendi içinde test ediyoruz. Bundan dolayı verilerimizin merkezi limit teoreminden normal dağılığını söyleyebiliriz.

- i. Cinsiyete göre yakılan ortalama kalori miktarları arasında fark yoktur hipotezinde değişkenlerimizin birbirinden bağımsız olduğunu biliyoruz. Merkezi limit teoremini görmezden gelerek test ettiğimizde veri setinin normal dağılıma uyup uymadığını test edeceğiz.
- Veri setinin büyüklüğü $n > 50$ olduğu için normalliği test etmek için Kolmogorov-Smirnov testini kullanırız. Bu test python'da scipy kütüphanesinin istatistik kısmında bulunur. Aşağıda scipy kütüphanesinden Kolmogorov-Smirnov testini çağırıldık.

```
from scipy.stats import ks_2samp
```

- Aşağıdaki kod dizisi ile erkeklerin ortalama kalori miktarı ile kadın üyelerin ortalama kalori miktarlarının KS test sonucunu ve p-değerini bulduk. Bulduğumuz değerleri çıktıda gözlemlayabiliriz.

```
ks_statistic, p_value = ks_2samp(male_calories, female_calories)

print("Kolmogorov-Smirnov test istatistiği: ", ks_statistic)
print("p-değeri: ", p_value)
```

Kolmogorov-Smirnov test istatistiği: 0.14617802289035164
p-değeri: 5.3686636227162995e-05

- If döngüsü ile Bölüm 12.2'de belirlediğimiz $\alpha = 0.05$ değeri ile p-değerini karşılaştırıp iki örneklenin aynı dağılıma sahip olma durumlarını gözlemledik. Buradaki hipotezleri dağılım için kurduk. Asıl test hipotezlerinden bağımsızdır. Kod çıktısına göre örneklem dağılımları aynı değildir.

```
alpha = 0.05
if p_value < alpha:
    print("İki örneklem aynı dağılıma sahip değildir.")
else:
    print("İki örneklem aynı dağılıma sahiptir.")
```

İki örneklem aynı dağılıma sahip değildir.

- Örneklenler normal dağılmadığı için t testi yerine parametrik olmayan testlerden Mann-Whitney testini kullanacağız. Mann-Whitney testini scipy kütüphanesinden çağırıldık ve ana hipotezimize ait p-değerini elde ettik.

```

from scipy.stats import mannwhitneyu

stat, p = mannwhitneyu(male_calories, female_calories)
print("Mann-Whitney U testi p-değeri:" ,p)

Mann-Whitney U testi p-değeri: 1.1387859626833666e-05

```

- ii. 73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark yoktur hipotezinde örneklemler birbirlerinden bağımsızdır. Örneklem 30'dan büyük olduğu için merkezi limit teoreminden örneklem dağılımlarının normal dağıldığı varsayımlı altında çözüyoruz.
- t-testini scipy kütüphanesinden import ettik. p-değerinin ve t-testi sonuçlarını çıktıda gözlemledik.

```

t_stat, p_value2 = ttest_ind(group1, group2)

print("t-test istatistiği: ",t_stat)
print("p-değeri: ",p_value2)

t-test istatistiği: -2.2810119499733217
p-değeri: 0.02276360451356769

```

29

- iii. 30 yaşından küçük üyelerin ortalama antrenman sıklığı, 30 yaşından büyük üyelerin ortalama antrenman sıklığından daha fazladır hipotezinde örneklemler birbirlerinden bağımsızdır ve 30'dan büyük oldukları için merkezi limit teoreminden normal dağılıklarını varsayıarak test ediyoruz.
- T-testini uyguladık fakat bu hipotez tek yönlü olduğu için p-değerini ikiye böldük. p-değerinin ve t-testi sonuçlarını çıktıda gözlemledik.

```

t_stat2, p_value3 = ttest_ind(kucuk_yas, buyuk_yas)

p_value_one_tailed = p_value3/2

print("t-istatistiği sonucu: ", t_stat2)
print("p-değeri: ", p_value3)

t-istatistiği sonucu: -0.8399526268813662
p-değeri: 0.40114171156457035

```

12.4 Karar Alma

12.4.1 Kritik Bölgeler ve Kritik Değerler

Kritik bölge denen kavram bizi yokluk hipotezini reddetmeye götürür. Bu yüzden kritik bölge red bölgesi olarak da geçer. α değeri kritik bölgenin örneklemeye dağılıminin yüzde kaçını kapsayacağını verir. Yani $\alpha = 0.05$ ise kritik bölge örneklemeye dağılıminin %5'ini kapsar.

Kritik değerler α 'nın Bölüm 12.3'te bahsettiğimiz z ve t dağılımlarının tablo değerleri, standart sapmaları ve verinin büyütüğü yardımı ile hesaplanır. Kisaca güven aralığının sınırlarından oluşur.

Veri setimizde kullanacağımız $\alpha = 0.05$ olduğundan kritik bölge örneklemeye dağılımımızın %5'ini kapsar.

12.4.2 Tek Yanlı ve İki Yanlı Testler Arasındaki Fark

Bölüm 12.1.1'de oluşturduğumuz istatistiksel hipotez örneklerinin Bölüm 12.1.2'de alternatif hipotezlerini belirtmiştık. Bu örneklerin bu başlık altında sırasıyla alternatif hipotezlerinin tek yanlı ve çift yanlı olmalarını inceleyeceğiz.

30

- i. Cinsiyete göre yakılan ortalama kalori miktarları arasında fark yoktur hipotezi için;

$$H_1: \mu_K - \mu_E \neq 0, \quad (\mu_K \neq \mu_E)$$

yokluk hipotezini kurmuştuk. Burada alternatif hipotezinin $\mu_K < \mu_E$ ve $\mu_K > \mu_E$ her iki olasılığını da kapsadığını gözlemliyoruz. İlgilendiğimiz tek durum ortalamalarının birbirine eşit olup olmadığı. İstatistikte iki olasılığı barındıran testlere **iki yanlı test** denir. İki yanlı test olduğundan Bölüm 12.3'te p -değerini buna göre hesapladık.

- ii. 73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark yoktur hipotezi için;

$$H_1: \mu_1 - \mu_2 \neq 0, \quad (\mu_1 \neq \mu_2)$$

yokluk hipotezini kurduk. Tıpkı i. örnekteki gibi bu hipotezde iki yanlı testtir. İki yanlı test olduğundan Bölüm 12.3'te p -değerini buna göre hesapladık.

- iii. 30 yaşından küçük üyelerin ortalama antrenman sıklığı, 30 yaşından büyük üyelerin ortalama antrenman sıklığından daha fazladır hipotezi için;

$$H_1: \mu_A - \mu_B > 0, \quad (\mu_A > \mu_B)$$

yokluk hipotezini kurduk. Burada alternatif hipotez yalnızca $\mu_A < \mu_B$ olasılığını kapsar. İstatistikte tek olasılığı barındıran testlere **tek yanlı test** denir. Tek yanlı test olduğundan Bölüm 12.3'te p -değerini ikiye bölgerek hesapladık.

12.5 p -değeri

p -değeri, istatistiksel bir testten hesaplanan ve sıfır hipotezi doğruya belirli bir gözlem kümesini bulma olasılığını tanımlayan bir sayıdır. p -değeri, sıfır hipotezini reddedip reddetmemeye karar vermeye yardımcı olması için hipotez testinde kullanılır.

p -değerleri genellikle istatistik programları tarafından otomatik olarak hesaplanır. Ayrıca test istatistiğinin p değerini tahmin etmek için çevrimiçi tablolar da kullanılabilir. Bu tablolar, test istatistiğine ve testin serbestlik derecesine (gözlem sayısı eksi bağımsız değişken sayısı) dayanarak, sıfır hipotezi altında o test istatistiğinin ne sıklıkta görmeyi beklenildiğini gösterir. α ile karşılaşıldığında yokluk hipotezi hakkında karar alınır.

12.5.1 Karar Alma Süreci

Yokluk hipotezimizin kabul ya da reddedileceğinin kararını vermek için p -değeri ve anlamlılık düzeyini karşılaştırırız. İki ihtimal vardır. Bunlar;

- p -değeri $< \alpha$ ise H_0 hipotezi reddedilir.
- p -değer $\geq \alpha$ ise H_0 hipotezi kabul edilir.

31

Spor salonu üyelerinin veri setine dönecek olursak kurduğumuz hipotezlerin kabul ya da reddedilme durumlarını inceleyelim.

- i. Cinsiyete göre yakılan ortalama kalori miktarları arasında fark yoktur hipotezi için;
 - Karar alma süreci için if döngüsü kurduk. Bu döngü alfa değeri p -değerinden küçükse cinsiyete göre yakılan kalori miktarları arasında fark olduğunu, alfa değeri p -değerinden büyükse cinsiyete göre yakılan kalori miktarları arasında fark olmadığını gösterir. Elde ettiğimiz Python çıktısına göre H_0 hipotezi reddedilir.

```
if p < alpha:  
    print("H0 hipotezi reddedilir. Cinsiyete göre yakılan kalori miktarları arasında fark vardır.")  
else:  
    print("H0 hipotezi kabul edilir. Cinsiyete göre yakılan kalori miktarları arasında fark yoktur.")
```

H_0 hipotezi reddedilir. Cinsiyete göre yakılan kalori miktarları arasında fark vardır.

- ii. *73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark yoktur hipotezi için;*
- Karar alma süreci için if döngüsü kurduk. Bu döngü alfa değeri p-değerinden küçükse 73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark olduğunu, alfa değeri p-değerinden büyükse 73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark olmadığını gösterir. Elde ettiğimiz Python çıktısına göre H_0 hipotezi reddedilir.

```
if p_value2 < alpha:
    print("H0 hipotezi reddedilir. 73'ten az ve fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark vardır.")
else:
    print("H0 hipotezi kabul edilir. 73'ten az ve fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark yoktur.")

H0 hipotezi reddedilir. 73'ten az ve fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark vardır.
```

- iii. *30 yaşından küçük üyelerin ortalama antrenman sıklığı, 30 yaşından büyük üyelerin ortalama antrenman sıklığından daha fazladır hipotezi için;*

- Karar alma süreci için if döngüsü kurduk. Bu döngü alfa değeri p-değerinden küçükse; 30 yaşından küçük üyelerin ortalama antrenman sıklığının, 30 yaşından büyük üyelerin ortalama antrenman sıklığından daha fazla olduğunu, 30 yaşından küçük üyelerin ortalama antrenman sıklığı, 30 yaşından büyük üyelerin ortalama antrenman sıklığından daha fazla olmadığını gösterir. Elde ettiğimiz Python çıktısına göre H_0 hipotezi kabul edilir.

32

```
if p_value_one_tailed < alpha and t_stat2 > 0:
    print("H0 reddedilir. 30 yaşından küçük üyelerin ortalama antrenman sıklığı daha fazladır.")
else:
    print("H0 kabul edilir. 30 yaşından küçük üyelerin antrenman sıklığı, 30 yaşından büyük üyelerin antrenman sıklığından fazla değildir.")

H0 kabul edilir. 30 yaşından küçük üyelerin antrenman sıklığı, 30 yaşından büyük üyelerin antrenman sıklığından fazla değildir.
```

12.5.2. Yaygın Bir Hata

p-değeri, sıfır hipotezinin doğru olma olasılığı olarak düşünülebilir. Bu ifade iki önemli açıdan yanlıştır:

- a) *Sıfır hipotezi sıklıkçı bir araçtır ve olasılığa yönelik sıklıkçı yaklaşım sıfır hipotezine değerler atamamıza izin vermez. Bu olasılık görüşüne göre sıfır hipotezi ya doğrudur ya da değildir. Doğru olma şansı %5 olamaz.*
- b) *Hipotezlere olasılıklar atanın Bayes yaklaşımında bile p-değeri yokluk hipotezinin doğru olma olasılığına karşılık gelmez.*

Bu yorum matematikte tamamen tutarsızdır.

12.6 Etki Büyüklüğü, Örneklem Büyüklüğü ve Güç

Bölüm 12.2'de 1.tip hatanın oranının araştırmacılar tarafından kontrol altına alınmak istendiğinden bahsetmiştik. $\alpha = 0.05$ olarak belirlemiştik. Sıfır hipotezlerinin yalnızca %5'inin yanlış bir şekilde reddedilmesini sağlamaya çalışıyoruz. Hipotez testinin ikincil bir amacı 2.tip hata yani β oranıdır. Ancak genellikle 2.tip hatayı en aza indirmekten bahsedilmez. Bunun yerine testin gücünü en üst düzeye çıkarmaktır.

12.6.1 Etki Büyüklüğü

Etki büyülüğu, iki değişken arasındaki ilişkinin gücünü sayısal bir ölçekte ölçen istatistiksel bir kavramdır. Başka bir tanımla yokluk hipotezinin uygulamaya ne kadar benzer olduğunu nicelleştirilmesine etki büyülüğünün ölçüsü denir. Hipotez testi; gözlemlediğimiz etkinin gerçek olduğuna inanıp inanmamamız gerektiğini söyler, etki büyülüğü ise bu durumu dikkate alıp almamamızla ilgilenir.

- i. Cinsiyete göre yakılan ortalama kalori miktarları arasında fark yoktur hipotezinin etki büyülüğü python'da aşağıdaki gibi hesaplanır.

➤ Erkek ve kadın üyelerin ortalama kalori miktarlarını `mean()` ile hesapladık.

33

```
mean_male = male_calories.mean()
print("Erkeklerin yaktığı ortalama kalori miktarı: ", mean_male)

mean_female = female_calories.mean()
print("Kadınların yaktığı ortalama kalori miktarı: ", mean_female)
```

Erkeklerin yaktığı ortalama kalori miktarı: 944.4559686888454
Kadınların yaktığı ortalama kalori miktarı: 862.2489177489177

➤ Erkek ve kadın üyelerin kalori miktarlarının standart sapmaları hesapladık.

```
std_male = np.std(male_calories, ddof = 1)
std_female = np.std(female_calories, ddof = 1)

print("Erkek üyelerin yaktıkları ortalama kalori miktarının standart sapması: ", std_male)
print("Kadın üyelerin yaktıkları ortalama kalori miktarının standart sapması: ", std_female)
```

Erkek üyelerin yaktıkları ortalama kalori miktarının standart sapması: 286.59098667906056
Kadın üyelerin yaktıkları ortalama kalori miktarının standart sapması: 249.6149154564867

- Etki büyüğünü ise; erkek ve kadın üyelerin ortalama kalori miktarlarının farkının, erkek ve kadın üyelerin kalori miktarlarının standart sapmalarının karelerinin toplamının karekökünün ikiye bölünmesi ile bulunur.

```
effect_size1 = (mean_male - mean_female) / np.sqrt((std_male**2 + std_female**2)/2)
print("Etki büyüğü: ", effect_size1)

Etki büyüğü:  0.3058984810793362
```

- ii. 73'ten az kiloya sahip üyelerin ortalama oturum süreleri ile 73'ten fazla kiloya sahip üyelerin ortalama oturum süreleri arasında fark yoktur hipotezinin etki büyüğü python'da aşağıdaki gibi hesaplanır.

- Grup1 ve grup2 olarak ikiye ayırdığımız grupların ortalamalarını hesapladık.

```
mean_weight1 = group1.mean()
print("73'ten az kiloya sahip üyelerin ortalama oturum süresi: ", mean_weight1)

mean_weight2 = group2.mean()
print("73'ten fazla kiloya sahip üyelerin ortalama oturum süresi: ", mean_weight2)

73'ten az kiloya sahip üyelerin ortalama oturum süresi:  1.234426229508197
73'ten fazla kiloya sahip üyelerin ortalama oturum süresi:  1.2849056603773583
```

34

- Grup1 ve grup2'nin standart sapmalarını hesaplıyoruz.

```
mean_male = male_calories.mean()
print("Erkeklerin yaktığı ortalama kalori miktarı: ", mean_male)

mean_female = female_calories.mean()
print("Kadınların yaktığı ortalama kalori miktarı: ", mean_female)

Erkeklerin yaktığı ortalama kalori miktarı:  944.4559686888454
Kadınların yaktığı ortalama kalori miktarı:  862.2489177489177
```

- 2.hipotezimizin etki büyüğü aşağıdaki gibidir.

```
effect_size2 = (mean_weight1 - mean_weight2) / np.sqrt((std_group1**2 + std_group2**2) / 2)
print("Etki büyüğü: ", effect_size2)

Etki büyüğü:  -0.1472326180175729
```

12.6.2 Güç Fonksiyonu

p-değeri gibi, güç de koşullu bir olasılıktır. Bir hipotez testinde alternatif hipotez, sıfır hipotezinin yanlış olduğu ifadesidir. Alternatif hipotez gerçekten doğruysa birinin sıfır hipotezini doğru bir şekilde reddetme olasılığına güç denir. $(1-\beta)$ olarak gösterilir.

- i. Cinsiyete göre yakılan ortalama kalori miktarları arasında fark yoktur hipotezinin gücü python'da aşağıdaki gibi hesaplanır.
 - Güç hesaplanırken python'in statsmodels kütüphanesinden yararlanılır. 2.tip hatayı ve testin gücünü değerlerine bu kod dizisi ile ulaşırız.

```
from statsmodels.stats.power import TTestIndPower
```

```
power_analysis = TTestIndPower()  
  
power = power_analysis.solve_power(effect_size1, power = None, nobs1 = len(male_calories), ratio = 1.0, alpha = alpha)  
beta = 1 - power  
  
print("Testin gücü: ", power)  
print("2.tip hata: ", beta)  
  
Testin gücü:  0.9982778243508986  
2.tip hata:  0.0017221756491013762
```

35

12.6.3 Çalışmanın Gücünü Artırmak

Bölüm 12.8.1'de gördüğümüz gibi gücü etkileyen faktörlerden biri etki büyülüğüdür. Bundan dolayı gücün artırmak için yapacağımız ilk şey etki büyülüğünü artırmaktır. Pratikte ise çalışmayı etki büyülüğünü artıracak şekilde tasarlamak daha doğru olur.

Bazı deney tasarımlarında bile küçük bir etki elde edilebilir. Bu koşullar altında en iyi seçenek örneklem büyülüğünü artırmak olur. Genel olarak elimizde ne kadar fazla gözlem olursa iki hipotez arasında ayrılmama olasılığı o kadar artar. Başka bir deyişle, güç örneklem büyülüğü arttıkça artar.

BÖLÜM 13 KATEGORİK VERİ ANALİZİ

Verilerin nicel olmayan tamamen nitel verilerden oluşan verilere kategorik veri (cinsiyet, eğitim, yaş grubu, medeni durum) denir. Hatta nominal ölçekli veri analizi diye de adlandırılır. Kategorik değişkenler arasındaki ilişkileri test etmek veya kategorik bağımlı değişkenli bir regresyon modeli kurmak için **kategorik veri analizi** teknikleri kullanılır. Burada kategorik veri analizi teknikleri kullanarak analizler yapılacaktır.

13.1 Ki-Kare Uyum İyiliği Testi:

Ki-kare testinin ne olduğundan bahsedilecek olursa: iki kategorik değişken arasındaki ilişkiyi anlamayı ve analiz etmeyi ve gözlenen değer ile beklenen değerin herhangi bir şekilde farklılık gösterip göstermediğini belirlemek için kullanılır. Ki-karenin türleri vardır ve bunlardan bir tanesi **ki- karenin uyum iyiliği testi**: bir değişkenin belirli bir dağılımdan gelip gelmediğini tespit etmek için yapılan istatistiksel bir testtir.

13.1.1. Spor Salonu Verileri:

Bu kısım bölüm 5 de betimleyici istatistiklere kısmında ilk başlarken gösterilmiştir trkar kısaca gösterilecek olursa. Yüklenen veri setini **data** ismiyle tanımladık. Burada da tekrar bakalım. Bu veri setini yüklemek için **Pandas** kütüphanesinden yararlanıyoruz. Pandas kütüphanesinin ne işe yaradığına diğer bölümlerde bahsedilmiştir.

```
import pandas as pd  
data = pd.read_csv(r"C:\Users\bilge\Downloads\data.csv")  
data
```

36

Verimizi `pd.read.csv` kodu ile yüklandı ve `data` değişkenine atadı verinin içeriğini görüntülemek için `data` yazılarak çalıştırılması yeterli olur.

	Yas	Cinsiyet	Kilo (kg)	Boy(m)	Max_BPM	Ort_BPM	Dinlenme_BPM	Oturum_Suresi(saat)	Yakilan_Kalori	Egzersiz_Turu	Yag_Yuzdesi	Tuketilen_Su(Litre)	Antreman_Sıklığı (gün/hafta)	Deneyim_Seviyesi	BKE
0	56	Male	88.3	1.71	180	157	60	1.69	1313	Yoga	12.6	3.5	4	3	30.20
1	46	Female	74.9	1.53	179	151	66	1.30	883	HIT	33.9	2.1	4	2	32.00
2	32	Female	68.1	1.66	167	122	54	1.11	677	Cardio	33.4	2.3	4	2	24.71
3	25	Male	53.2	1.70	190	164	56	0.59	532	Strength	28.8	2.1	3	1	18.41
4	38	Male	46.1	1.79	188	158	68	0.64	556	Strength	29.2	2.8	3	1	14.39
...
968	24	Male	87.1	1.74	187	158	67	1.57	1364	Strength	10.0	3.5	4	3	28.77
969	25	Male	66.6	1.61	184	166	56	1.38	1260	Strength	25.0	3.0	2	1	25.69
970	59	Female	60.4	1.76	194	120	53	1.72	929	Cardio	18.8	2.7	5	3	19.50
971	32	Male	126.4	1.83	198	146	62	1.10	883	HIT	28.2	2.1	3	2	37.74
972	46	Male	88.7	1.63	166	146	66	0.75	542	Strength	28.8	3.5	2	1	33.38

973 rows × 15 columns

Burada ki-kare uyum iyiliği testi yapacağım için gözlenen frekanslara atanacak olan değişkeni **Egzersiz_Turu** seçelim ve Python da **observed** değişkenine atayarak yapılım.

```
# 'Egzersiz_Turu' değişkeni için gözlemlenen frekansları hesapla  
observed = data['Egzersiz_Turu'].value_counts()  
print(observed)
```

Egzersiz_Turu	count
Strength	258
Cardio	255
Yoga	239
HIT	221
Name: count, dtype: int64	

Buna göre frekans tablosu:

<i>Etiket</i>	<i>Dizin</i>	<i>Matematiksel Notasyon</i>	<i>Python Komutu</i>	<i>Değer</i>
<i>Strength</i>	0	O_1	<code>observed[0]</code>	258
<i>Cardio</i>	1	O_2	<code>observed[2]</code>	255
<i>Yoga</i>	2	O_3	<code>observed[3]</code>	239
<i>HIIT</i>	3	O_4	<code>observed[4]</code>	221

O_i : Gözlemlenen frekans (*observed*).

E_i : Beklenen frekans (*expected*).

Matematiksel Gösterim

Eğer *Egzersiz_Turu* sütunundaki tüm gözlemlenen frekanslara bir vektör olarak atıfta bulunmak istersek:

$$O = [O_1, O_2, O_3, O_4] = [258, 255, 239, 221]$$

Burada:

- O : Tüm gözlenen frekansların vektörü.
- O_1 : *Strength* frekansı.
- O_2 : *Cardio* frekansı
- O_3 : *Yoga* frekansı
- O_4 : *HIIT* frekansı

13.1.2. Sıfır Hipotezi ve Alternatif Hipotez:

H_0 (*Sıfır Hipotezi*): Dört egzersiz türü hepsi eşit olasılıkla seçilir.

H_1 (*Alternatif Hipotez*): dört egzersiz türü arasında en az birini farklı olasılıkla seçilir.

Bu değişkende 4 grup olduğu için bütün türler 0.25 olasılıkla seçilmesi beklenir. Buna da P_i -inci egzersiz türü olasılığını verir. Bunun içinde Python da listelemeyi kullanarak yapılır ve *probabilities* değişkenine atanır.

```
dict = {'Strength' : .25,
        'Cardio' : .25,
        'Yoga' : .25,
        'HIIT' : .25}

probabilities = pd.Series(dict)
probabilities
```

```
Strength      0.25
Cardio       0.25
Yoga        0.25
HIIT        0.25
dtype: float64
```

Burada da *dict* olarak tanımlanan listeyi okuması için pandas kütüphanesinin *pd.Series()* yararlanılmıştır.

13.1.3. "Uyumun İyiliği" Test İstatistiği:

Ki-kare istatistiği, gözlemlenen frekansların (*observed*) beklenen frekanslardan (*expected*) ne kadar saplığını ölçer. Önceki kısımda gözlemlenen frekansların (*observed*) bulunmuştu şimdi de beklenen frekansları (*expected*) bulalım bunu hesaplamak için de;

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Formülü kullanılır şimdi baradaki işlemleri tek tek Python komutlarıyla hesaplayıp ki-kare test istatistiğine ulaşalım.

Gözlemlenen frekansların (*observed*) değerini zaten diğer kısımda Egzersiz_Turu olarak belirlemiştik şimdi beklenen frekanslardan (*expected*) değerini bulalım.

$$E_i = N \times P_i$$

Formülü kullanılır P_i zaten bulmuştuk. N ise veri setindeki Egzersiz_Turu değişkenin gözlem sayısını vermektedir. Bu da $N=973$ tür. Beklenen frekansı Python'da bulunması için;

```
N = 973
expected = N * probabilities
expected
```

```
Strength      243.25
Cardio        243.25
Yoga          243.25
HIIT          243.25
dtype: float64
```

Kodu ile beklenen frekanslar bulunur ve ekrana yazdırılır. Eğer 973 gözlem varsa ve biz dört kategorinin de eşit derecede olası olduğunu düşünüyorsak, o zaman ortalama olarak 243.25 gözlem görmeyi bekleriz.

```
observed - expected
```

```
Egzersiz_Turu
Strength      14.75
Cardio        11.75
Yoga          -4.25
HIIT          -22.25
dtype: float64
```

Ki-kare test istatistiği formülüne ulaşmak gözlemlenen frekansların (*observed*) ve beklenen frekanslardan (*expected*) farklarının alınması gereklidir. Bunu da Python'da yandaki gibi yapabiliriz.

```
(observed - expected)**2
```

```
Egzersiz_Turu
Strength      217.5625
Cardio        138.0625
Yoga          18.0625
HIIT          495.0625
dtype: float64
```

Şimdi de bu farkında karesini alalım. Python'da $**2$ üs almaya yarar.

```
(observed - expected)**2 / expected
```

```
Egzersiz_Turu
Strength      0.894399
Cardio        0.567575
Yoga          0.074255
HIIT          2.035200
dtype: float64
```

Tamamen test istatistiğine göre devam edelim ve bu farkların karelini beklenen frekansa (expected) bölelim

```
sum((observed - expected)**2/expected)
```

```
3.5714285714285716
```

Son olarak da bütün bulduğumuz bu kategorileri toplayıp. Ki-kare test istatistiğine ulaşalım. Sonuç: **Ki-Kare= 3,57** olarak bulunur.

13.1.4. Gof İstatistiğinin Örnekleme Dağılımı (Gelişmiş):

Burada eğer sıfır hipotez gerçekten doğru ise yani sıfır hipotezini kabul edersek ne yapmamız gerekiğinden ve sıfır hipotezinin doğru olması durumunda örnekleme dağılıminın nasıl oluşturulacağı hakkında bilgi verir. Sıfır hipotezi, bir deneyde belirli kategorilere düşme olasılıklarının önceden belirlenmiş olasılıklarla uyumlu olduğunu ifade eder. "Gözlemler" sanki **binom dağılımı** gibi davranışır. Eğer örneklem büyülüğu (n) yeterince büyükse ve olasılıklar 0 veya 1'e çok yakın değilse, bu binom dağılımı **normal dağılıma** yaklaşır (Merkezi Limit Teoremi sayesinde) ve bu durum ki-kare dağılımını ortaya çıkarır. Burada diğer önemli konu **Serbestlik Dereceleri Nedir? (DOF)** bir veri kümesinde özgürce değişim olan bağımsız değerlerin sayısını ifade eder. Ki-kare uyum iyiliği testinde serbestlik derecesi, toplam k kategori için $SD = k - 1$ olarak hesaplanır. Eğer parametre tahminleri eklenirse, her tahmin bir kısıtlama getirir ve serbestlik derecesi $SD = k - m - 1$ olur. Örneğin, binom dağılımında m tahmin ediliyorsa $m=1$, $m=1$, normal dağılımda ortalama ve standart sapma tahmin ediliyorsa $m=2$, $m=2$ olarak alınır.

13.1.5. Sıfır Hipotezini Test Etme:

Bu hipotezlerin kabul veya reddi için gerekli koşullar; **p-değeri < 0.05**: H_0 reddedilir. **p-değeri >= 0.05**: H_0 kabul edilir. Şeklinde hipotezi test ederiz. Ayrıca Ki-Kare değeri büyürse, gözlenen ve beklenen değerler arasında daha büyük bir fark olduğunu gösterir. Bu da Egzersiz_Turu değişkeninden gelen verileri tahmin etmede zayıf bir iş çıkardığı söylenebilir. Ki-kare uyum iyiliği testi her zaman tek taraflı bir testtir. Testimizin anlamlılık düzeyine $\alpha=0.05$ olarak alalım bu da örnekleme dağılıminın 0.95 yüzdelik dilimini istediğimiz anlamına gelir. Python'da 0.95 yüzdelik dilimini yani grafiğin ki-kare yüzdesini gösterdiği değeri hesaplamak için **chi2.ppf()** fonksiyonu kullanılır.

```
from scipy.stats import chi2
round(chi2.ppf(0.95, 3),2)
```

7.81

Python'da **chi2.ppf()** fonksiyonu, **SciPy** kütüphanesinin **scipy.stats** modülü altında bulunan bir fonksiyondur. Bu fonksiyon, **ki-kare dağılımı** için **ters** olasılık yoğunluk fonksiyonunu hesaplar. **0.95**: Kümülatif olasılık değeri. Bu, toplam olasılığın **%95'ine** karşılık gelen kritik değeri ifade eder. **3**: Serbestlik derecesi ($df=3$). **round()**: Sonucu **2 ondalık basamağa** yuvarlar. Kritik değer=7.81'dir. χ^2 değeri 7.81'den daha büyükse, o zaman sıfır hipotezini reddedebiliriz. Önceki bölümde hesaplanan χ^2 değeri 3,57 olduğu için sıfırın hipotezini reddedemeyiz.

Eğer bir *p_value* değeri hesaplayarak hipoteze karar vermek istenirse *SciPy* kütüphanesinin *scipy.stats* modülü altında bulunan *chisquare()* fonksiyonu ile hesaplanarak bulunabilir. Bu fonksiyon *ki-kare uyum iyiliği testi* yapmak için kullanılan bir fonksiyondur. Bu test, gözlenen değerlerin (*f_obs*) beklenen değerlerle (*f_exp*) ne kadar uyumlu olduğunu değerlendirir. Şimdi bunu yapalım

```
from scipy.stats import chisquare
chisquare(f_obs = observed, f_exp = expected)

Power_divergenceResult(statistic=3.5714285714285716, pvalue=0.3116155523686131)
```

Bu kodda *f_obs*: Gözlenen frekanslar (*observed*). *f_exp*: Beklenen frekanslar (*expected*) vermektedir. Burada test istatistiği yine bekendiği gibi 3.57 *p_değeri*= 0.31 çıkmıştır. *p-değeri* ≥ 0.05 olduğunda H_0 kabul edilir olduğunu biliyoruz. Sonuç olarak hem test istatistiğinde hem de *p_değerinde* hipotez kabul edilmiştir. Yani bunu da egzersiz türünde bütün kategoriler eşit olasılıkla seçilir yorumunu yaparız.

13.1.6. Python'da Test Yapmak:

Diğer bölümle *ki-kare test istatistiğini* tek tek hesaplamak yerine *SciPy* kütüphanesinin *scipy.stats* modülü altında bulunan *chisquare()* fonksiyonu ile diğer test istatistiğini ve *p-value* değerlerini kısa bir şekilde bulabiliriz şimdi bu şekilde bulalım:

```
#13.1 (χ²) Ki-Kare Test İstatistiği:

import pandas as pd
from scipy.stats import chisquare

# Gözlenen frekanslar (Cinsiyet dağılımı)
observed = data['Egzersiz_Turu'].value_counts()

# Beklenen frekanslar (eşit dağılım varsayımlı)
expected= [len(data) / len(observed)] * len(observed)

# Ki-Kare uyum iyiliği testi
chi2_stat, p_value = chisquare(f_obs=observed, f_exp=expected)

# Sonuçları yazdırma
print("Ki-Kare Test İstatistiği (χ²): {chi2_stat}")
print("p-Değeri: {p_value}")

# Hipotez testi sonucu
if p_value < 0.05:
    print("Sonuç: Gözlenen sıklık ile beklenen sıklık arasında istatistiksel olarak anlamlı bir fark vardır.")
else:
    print("Sonuç: Gözlenen sıklık ile beklenen sıklık arasında istatistiksel olarak anlamlı bir fark yoktur.")
```

Bu kod, "*Egzersiz_Turu*" değişkenindeki grupların dağılımin eşit olup olmadığını test eder. İlk olarak gerekli kütüphaneler eğer bilgisayar da yükli değilse yüklenir ve çalışma ortamına entegre edilir. Burada kullanılan kütüphaneler **Pandas Kütüphanesi**: Verileri analiz etmek ve işlemek için kullanılır, **SciPy Kütüphanesi**: Bilimsel hesaplama ve teknik hesaplama için kullanılan bir kütüphanedir. Buradaki kütüphaneleri indirmek için **pip install pandas** ve **pip install SciPy** kodu yazılarak kütüphaneler indirilir. **import pandas as pd** kodu pandas kütüphanesini çalışma ortamına entegre etmeye ve kodlarda kolaylık olması açısından **pd** kısaltılmışına çevrilmesine yarar. *Ki-Kare uyum iyiliği testi* yapmak için *SciPy* kütüphanesinden *scipy.stats.chisquare* fonksiyonu kullanılır bu fonksiyonu anlatılacak olursa. **Kütüphane**:

*SciPy (scipy), Alt modül: stats (istatistiksel analiz için kullanılır) ve Fonksiyon: chisquare (Ki-Kare uyum iyiliği testi yapar). Ki-Kare uyum iyiliği testi için Gözlenen frekanslara (Cinsiyet dağılımı) ihtiyaç vardır **observed = data['Egzersiz_Turu'].value_counts()** kodu kullanılır. Bu koddaki **data[' Egzersiz_Turu ']**: data verisindeki egzersiz türü sütununu seçmeye yarar. **value_counts()**: 'Egzersiz_Turu' sütunundaki farklı kategorilerin (Strength, Cardio, Yoga, HITT) frekanslarını hesaplar ve gözlenen egzersiz türü dağılımlarını içeren bir tabloyu (Strength=258, Cardio=255, Yoga=239, HITT=221) **observed** değişkenine atar. Ki-Kare uyum iyiliği testi için diğer gerekli olan etken beklenen frekanslardır (eşit dağılım varsayımları) bunun içinde **expected = [len(data) / len(observed_counts)] * len(observed_counts)** kodu kullanılır. Bu kodda **len(data)**: Veri kümesindeki toplam gözlem sayısını döndürür. **len(observed)**: "Egzersiz_Turu" sütunundaki kategorilerin sayısını döndürür. **len(data) / len(observed)**: Her bir kategori için eşit sayıda gözlem varsayar (Bütün egzersiz türleri için eşit beklenen dağılım). *** len(observed)**: Her kategori için beklenen frekansların bir listesini oluşturur. (973 veri var ve 4 kategori varsa, [243.25, 243.25, 243.25, 243.25]) ve bu işlemleri **expected** değişkenine atar. Beklenen ve gözlenen sıklıklar hesaplandıktan sonra Ki-Kare uyum iyiliği testi yapılabılır bunun için SciPy kütüphanesinin **chisquare** fonksiyonu kullanılır ve **chi2_stat, p_value = chisquare(f_obs=observed, f_exp=expected)** kodu ile yapılır. **f_obs**: Gözlenen frekanslar (Strength=258, Cardio=255, Yoga=239, HITT=221). **f_exp**: Beklenen frekanslar ([243.25, 243.25, 243.25, 243.25]). **Dönen Değerler**: **chi2_stat**: Ki-Kare test istatistiği. Gözlenen ve beklenen frekanslar arasındaki farkın büyüklüğünü ölçer. **p_value**: Bu farkın tesadüfen oluşma olasılığını ifade eden p-değeri. Ki-kare testi bulunur ve sonuçlar **print()** fonksiyonuyla yazdırılır. **print(f"Ki-Kare Test İstatistiği (χ^2): {chi2_stat}")**: Ki-Kare test istatistiğini yazdırır. (buradaki **f**': string içeriğine dinamik veri yerleştirme için kullanılır. Bu, değişken veya ifade ifade değerlerini doğrudan bir stringin içine yerleştirmenin basit bir yoludur. Yani buradaki **{chi2_stat}** un sayısal değerini yazar Ki-Kare Test İstatistiği (χ^2) kısmının karşısına **chi2_stat** değerini yazar). **print(f"p-Değeri: {p_value}")**: p-değerini yazdırır. Hipotez sonucunu reddini ve kabulüne de çıktı olarak bakılacak olursa **if else** koşulu kurularak bakılır;*

```

if p_value < 0.05:
    print("Sonuç: Gözlenen sıklık ile beklenen sıklık arasında istatistiksel olarak anlamlı bir fark vardır.")
else:
    print("Sonuç: Gözlenen sıklık ile beklenen sıklık arasında istatistiksel olarak anlamlı bir fark yoktur.")

```

if koşulu eğer p değeri 0.05 den küçükçe if koşulunu yazdırır yani gözlenen sıklık ile beklenen sıklık arasında istatistiksel olarak anlamlı bir fark vardır çıktısını verir. Else koşulu ise if koşulunun olmadığı diğer her durumda geçerli olur ve gözlenen sıklık ile beklenen sıklık arasında istatistiksel olarak anlamlı bir fark yoktur çıktısını verir.

Hipotezler kurulursa: **H_0 (Sıfır Hipotezi)**: Gözlenen frekanslar beklenen frekanslarla uyumludur (anlamlı fark yok). **H_1 (Alternatif Hipotez)**: Gözlenen frekanslar beklenen frekanslarla uyumlu değildir (anlamlı fark var). Bu hipotezlerin kabul veya redi için gerekli koşullar; **p-değeri < 0.05**: H_0 reddedilir. Gözlenen ve beklenen frekanslar arasında anlamlı bir fark vardır. **p-değeri >= 0.05**: H_0 kabul edilir. Gözlenen ve beklenen frekanslar arasında anlamlı bir fark yoktur. Şeklindir.

Yorum (Çıktı Değerleri):

Ki-Kare Test İstatistiği (χ^2): 3.5714285714285716
p-Değeri: 0.3116155523686131
Sonuç: Gözlenen sıklık ile beklenen sıklık arasında istatistiksel olarak anlamlı bir fark yoktur.

1. Ki-Kare Test İstatistiği (χ^2): 3.57

Bu istatistik, gözlenen değerler ile beklenen değerler (eşit dağılım varsayımlı) arasındaki farkın büyüklüğünü ölçer. Yani gözlenen değerlerle beklenen değerler arasındaki farkın büyüklüğü **3.57'dir**. Ki-Kare değeri büyükse, gözlenen ve beklenen değerler arasında daha büyük bir fark olmadığını gösterir.

2. p-Değeri: 0.3116

p-değeri, bu farkın tesadüfen oluşma olasılığını ifade eder. Eğer p-değeri **0.05'ten küçük** Burada p-değeri **0.3116**, yani 0.05'ten büyüktür ve H_0 kabul edilir. Gözlenen ve beklenen frekanslar arasında anlamlı bir fark yoktur şeklinde yorumlanır.

3. Sonuç:

"Gözlenen sıklik ile beklenen sıklik arasında istatistiksel olarak anlamlı bir fark yoktur." Bu sonuç, "Egzersiz_Turu" dağılıminındaki ki grupların eşit dağılım varsayımlına uygun olduğunu gösterir. Yani, grup türleri arasındaki frekansları arasında beklenenden daha büyük bir farklılık bulunmamaktadır.

42

13.1.7. Farklı bir Sıfır Hipotezi Belirleme:

Burada olasılık değerlerini 4 grup içinde eşit olmadığını ve hepsi için farklı farklı olasılıklar belirlersek ne olacağına örnek verilmek isteniyor.

```
dict = {'Strength' : .20,
        'Cardio' : .15,
        'Yoga' : .25,
        'HIIT': .40}

probabilities = pd.Series(dict)
probabilities
```



```
Strength    0.20
Cardio      0.15
Yoga        0.25
HIIT        0.40
dtype: float64
```

Burada (Strength=0.20, Cardio=0.15, Yoga=0.25, HIIT=0.40) olasılıkları alınmıştır. Şimdi bu olasılıklar üzerinden test istatistiğini ve p-value değerlerini hesaplayalım buna göre olan değişiklikleri inceleyelim

```

import pandas as pd
from scipy.stats import chisquare

# Gözlenen frekanslar (Cinsiyet dağılımı)
observed = data['Egzersiz_Turu'].value_counts()

#beklenen frekansı yeni olasılıklara göre hesapla
expected = probabilities * sum(observed)

# Ki-Kare uyum iyiliği testi
chi2_stat, p_value = chisquare(f_obs=observed, f_exp=expected)

# Sonuçları yazdırma
print(f"Ki-Kare Test İstatistiği ( $\chi^2$ ): {chi2_stat}")
print(f"p-Degeri: {p_value}")

```

Gerekli kütüphaneler ve fonksiyonlar aynıdır. Gözlenen frekanslar da bir değişiklik olmaz (`observed`) değişkeni aynı şekilde "Egzersiz_Turu" sütununu seçer. Beklenen frekanslarda (`expected`) olasılık değeri olduğu ve olasılık değerleri değiştiği için beklenen `expected` değeri değişir. **probabilities**: Daha önceden tanımlanmış olasılık değerleri `sum(observed)`: Gözlenen değerlerin(`observed`) toplamını hesaplar. **probabilities * sum(observed)**: Beklenen frekansları hesaplar. Her bir olasılık değeri, toplam gözlem sayısıyla çarpılır ve böylece **beklenen frekanslar** elde edilir. **chi2_stat, p_value = chisquare(f_obs=observed , f_exp=expected)** kodu ile yapılır. **f_obs**: Gözlenen frekanslar (`Strength=258, Cardio=255, Yoga=239, HITT=221`). **f_exp**: Beklenen frekanslar (`[Strength=194.60, Cardio =145.95, Yoga=243.25, HIIT=389.20]`). **Dönen Değerler**; **chi2_stat**: Ki-Kare test istatistiği. Gözlenen ve beklenen frekanslar arasındaki farkın büyüklüğünü ölçer. **p_value**: Bu farkın tesadüfen oluşma olasılığını ifade eden p-değeri. Ki-kare testi bulunur ve sonuçlar `print()` fonksiyonuyla yazdırılır. `print(f"Ki-Kare Test İstatistiği (χ^2): {chi2_stat}")`: Ki-Kare test istatistiğini yazdırır. `print(f"p-Degeri: {p_value}")`: p-değeri yazdırır.

43

Yorum (Çıktıdaki Değerler):

Ki-Kare Test İstatistiği (χ^2): 174.89979445015422
 p-Degeri: 1.1137534304716296e-37

1. Ki-Kare Test İstatistiği (χ^2): 174.89

Gözlenen değerlerle beklenen değerler arasındaki farkın büyüğü **174.89'dur**. Ki-Kare değeri eşit olasılık varsayımlı yapılan teste istatistiğine göre epeyce bir fark olarak büyümüştür. Buna bakarak gözlenen ve beklenen değerler arasında daha büyük bir fark olduğunu gösterir.

2. p-Degeri: 1.1137534304716296e-37

Eşit olasılıkla yapılan varsayımda $p\text{-değer}>0.05$ çıkmıştı ve sıfır hipotezi kabul edilmişti. Burada ise $p\text{-değer}<0.05$ olarak görülür ve H_0 hipotezi reddedilir.

3. Sonuç:

"Gözlenen sıkılık ile beklenen sıkılık arasında istatistiksel olarak anlamlı bir fark vardır." Bu sonuç, "Egzersiz_Turu" grup türleri arasındaki frekansları arasında beklenenden daha büyük bir farklılık bulunmaktadır yorumu yapılır.

13.1.8. Test Sonuçlarının Nasıl Raporlanacağı:

Burada yapılan istatistiklerin sonuçlarının nasıl karşı tarafa aktarılacağı hakkında bilgi verir. Egzersiz_Turu değişkeni için yaptığımız analizinden yararlanarak bir rapor oluşturalım. Bunun için ilk olarak istatistiksel test, tanımlayıcı istatistiklerle başlar ki bunu bölüm 5 de detaylıca yapmıştır. Daha sonra açıklama yani hipotezler kurulur ve hipotezlerin neyi ifade ettiğine bakılır. **H₀ (Sıfır Hipotezi):** Dört egzersiz türü hepsi eşit olasılıkla seçilir.

H₁ (Alternatif Hipotez): dört egzersiz türü arasında en az birini farklı olasılıkla seçilir.

Bunu da bu şekilde yaptı. Daha sonra hipotezin χ^2 değerine ve p-değerine göre hipotezin kabul veya reddedilme kararı verilir. Daha sonra bütün bu sonuçlar yorumlanır.

13.1.9. İstatistiksel Gösterime İlişkin Bir Yorum (Gelişmiş):

Bu kısımda **test istatistikleri** ve **örnekleme dağılımları** arasındaki isimlendirme karmaşasını anlatıyor. Ki-kare testleri için kullanılan test istatistikleri bazen örnekleme dağılımı ile aynı isimle anılır (örneğin, "ki-kare istatistiği"). Aynı örnekleme dağılımını (ki-kare dağılımı gibi) paylaşan farklı testler olabilir, ancak test istatistikleri farklı hesaplanır. Örneğin: Pearson'un uyum iyiliği testi ve başka bir test aynı dağılıma dayanabilir ama farklı değerler üretir. Farklı araştırmacılar ve yöntemler zamanla birleşmiş, bu da aynı dağılımı paylaşan testlerin isimlendirilmesinde tutarsızlıklara yol açmıştır. Sonuç olarak, bir test istatistiğinin ardından gelen örnekleme dağılımı aynı olsa bile, hesaplama yöntemi farklı olabilir. Bu nedenle, aynı dağılımla ilgili farklı testler arasında net bir ayrim her zaman yapılamaz.

13.2. χ^2 Bağımsızlık (Veya İlişki) Testi:

Ki-Kare bağımsızlık testi Ki-Kare testinin diğer bir türüdür. İki kategorik değişkenin ilişkili olup olmadığını belirlemek için bağımsızlığın ki-kare testini, aynı zamanda ilişki ki-kare testi olarak da kullanabilir. İki değişken ilişkiliyse, bir değişkenin belirli bir değere sahip olma olasılığı diğer değişkenin değerine bağlıdır. Ki-kare bağımsızlık testi hesaplamaları, her bir birleştirilmiş gruptaki gözlem sayısı olan gözlenen frekanslara dayanmaktadır. Test, çapraz tablolar oluşturularak gözlenen frekansları, iki değişkenin ilişkisiz olması durumunda bekleneceği frekanslarla karşılaştırılır. Değişkenler ilişkisiz olduğunda, gözlenen ve beklenen frekanslar benzer olacaktır. Aynı veri setinden devam edileceği için `read.csv()` yazarak tekrar göstermiyoruz. Veri setinden seçtiğimiz "**Antreman_Sıklığı (gun/hafta)**" değişkeni ve "**Egzersiz_Turu**" değişkenleri arasında çapraz tablo oluşturularak bakılacaktır. Şimdi çapraz tablo oluşturalım.

```
pd.crosstab(index=data["Egzersiz_Turu"], columns=data["Antreman_Sikligi (gun/hafta)"], margins=False)
```

Antreman_Sikligi (gun/hafta)	2	3	4	5
Egzersiz_Turu				
Cardio	63	92	77	23
HIIT	42	86	67	26
Strength	45	102	85	26
Yoga	47	88	77	27

Burada data olarak tanımladığımız veri setinden **Pandas** kütüphanesindeki **crosstab()** fonksiyonunu kullanarak iki değişken arasındaki **çapraz tabloyu** oluşturur: **pd.crosstab()**: **Çapraz tablo** oluşturmak için kullanılır. İki değişkenin kategori bazında frekanslarını (sayımını) gösterir.

Parametreler:

index=data["Egzersiz_Turu"]: Tablonun **satırlarını** belirler. Burada **data["Egzersiz_Turu"]** sütunundaki kategoriler satır etiketlerini oluşturur.

columns=data["Antreman_Sikligi (gun/hafta)"]: Tablonun **sütunlarını** belirler. Burada **data["Antreman_Sikligi (gun/hafta)"]** sütunundaki kategoriler sütun etiketlerini oluşturur.

margins=False: Toplamları hesaplamayı kapatır. Eğer **margins=True** yazılısaydı, satır ve sütun toplamları da tabloya eklenirdi.

Şimdi bir de **margins=True** yaparak bakalım.

45

```
pd.crosstab(index=data["Egzersiz_Turu"], columns=data["Antreman_Sikligi (gun/hafta)"], margins=True)
```

Antreman_Sikligi (gun/hafta)	2	3	4	5	All
Egzersiz_Turu					
Cardio	63	92	77	23	255
HIIT	42	86	67	26	221
Strength	45	102	85	26	258
Yoga	47	88	77	27	239
All	197	368	306	102	973

Burada da görüldüğü gibi satır ve sütun toplamları yeni satır ve sütunlar oluşturarak **All** kısmında yazıldı. Buna göre toplam **973 kişi**, dört farklı egzersiz türü ve haftalık 2, 3, 4 veya 5 gün antrenman yaparak dağılım göstermektedir. **Strength** egzersizi, toplamda en fazla tercih edilen türdür. Haftada **3 gün** antrenman yapmak, en popüler sikluktur. **Cardio** ve **Strength** türleri özellikle daha düzenli haftalık dağılımlar gösterirken, **HIIT** ve **Yoga** daha dengeli ama nispeten düşük tercihlere sahiptir.

13.2.1. Hipotez Testimizi Oluşturma:

Hipotezleri oluşturmak için notasyonları içeren bir tablo yapalım.

	2	3	4	5	Toplam
<i>Strength</i>	O_{11}	O_{12}	O_{13}	O_{14}	R_1
<i>Cardio</i>	O_{21}	O_{22}	O_{23}	O_{24}	R_2
<i>Yoga</i>	O_{31}	O_{32}	O_{33}	O_{34}	R_3
<i>HIIT</i>	O_{41}	O_{42}	O_{43}	O_{44}	R_4
Toplam	C_1	C_2	C_3	C_4	N

Satırlar: Egzersiz türleri (*Cardio*, *HIIT*, *Strength*, *Yoga*).

Sütunlar: Haftalık antrenman sıklığı (2, 3, 4, 5 gün/hafta).

Hücreler (O_{ij}): Her hücre, *i*. egzersiz türü ve *j*. antrenman sıklığı kombinasyonu için gözlemlenen frekansı temsil eder.

Toplamlar: R_i : Her satırındaki toplam (egzersiz türüne göre toplam kişi sayısı). C_j : Her sütundaki toplam (antrenman sıklığına göre toplam kişi sayısı). N : Genel toplam (tüm gözlemler toplamı).

Hipotez:

H_0 (**Sıfır Hipotezi**): Değişkenler bağımsızdır (aralarında ilişki yoktur).

H_1 (**Alternatif Hipotez**): Değişkenler bağımsız değildir (aralarında ilişki vardır).

Beklenen Frekans: Bir değişkenin oranlarının diğer değişkenin tüm değerleri için aynı olduğu şekildedir. Beklenen frekansları, olasılık tablosunu kullanarak hesaplayabilirsiniz. **Satır R** ve **sütun C** için beklenen frekans şudur:

$$\hat{E}_{ij} = \frac{(R_i \times C_j)}{N}$$

Eğer P_i olasılıklar bulunursa beklenen frekanslar; $E_{ij} = C_j \cdot P_i$ formülü ile bulunur. Olasılıklar verilerden tahmin etmemiz gereken bir şey bu satır R değerlerinin kitle büyüğününe (N) bölünerek bulunur. $P_i = \frac{R_i}{N}$ Formülü kullanılır.

Beklenen frekansların ve gözlenen frekansların nasıl hesaplandığını gördük şimdi ise χ^2 bağımsızlık test istatistiğinin nasıl hesaplandığına bakalım:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(E_{ij} - O_{ij})^2}{E_{ij}}$$

O_{ij} : *i*. satırında ve *j*. sütununda gözlenen hücre sayısıdır.

E_{ij} : *i*. satır ve *j*. sütunundaki beklenen hücre sayısıdır.

Serbestlik derecenin ne olduğundan ki-kare uyum iyiliği testinde bahsetmiştik. Peki ki-kare bağımsızlık testinin serbestlik derecesi nedir?

$$df = (r-1) \times (c-1)$$

r : satır sayısı

c : sütun sayısı

13.2.2. Python'da Test Yapmak:

Python da yukarıda verilen işlemleri bazı kütüphaneler ve fonksiyonlar kullanılarak kısaca bu kadar formül olmadan da ki-kare test istatistiği hesaplanabilir. Bu kütüphaneler SciPy kütüphanesinin `scipy.stats` modülünün `chi2_contingency` fonksiyonu ve `pingouin` kütüphaneleri kullanılarak yapılabilir.

13.2.2.1. `scipy.stats` İle Test Yapma:

```
# 13.2 (χ²) Ki-Kare Bağımsızlık veya (İlişki) Testi:

import pandas as pd
from scipy.stats import chi2_contingency

# Çapraz tablo oluşturma
contingency_table = pd.crosstab(index=data['Antreman_Sikligi (gun/hafta)'], columns=data['Egzersiz_Turu'])

# Bağımsızlık testi için ki-kare testi
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Sonuçları yazdırma
print("Çapraz Tablo:")
print(contingency_table)

print("\nChi2 Test İstatistiği:", chi2)
print("p-Degeri:", p)
print("Serbestlik Dereceleri:", dof)

print("\nBeklenen Frekanslar:")
print(pd.DataFrame(expected, index=contingency_table.index, columns=contingency_table.columns))

# Hipotez sonucu
if p < 0.05:
    print("Sonuç: Antreman Sıklığı ve egzersiz türü birbirinden bağımsızdır.")
else:
    print("Sonuç: Antreman Sıklığı ve egzersiz türü birbirinden bağımsız değildir.")
```

47

Kullanılan data verisinden kategorik değişken olan “`Antreman_Sikligi (gun/hafta)`” ve “`Egzersiz_Turu`” değişkenleri arasındaki ki-kare bağımsızlık testi yapılacaktır. Bağımsızlığın ki-kare testi, gözlenen ve beklenen frekansları karşılaştırarak çalışır. Beklenen frekanslar hesaplanır (formülü: $(satır toplamı \times sütun toplamı) / \text{genel toplam}$), ki kare testi yapılır ve çapraz tablo oluşturulur. Bunlara aşağıdaki Python kodlarıyla nasıl yapıldığını bakılacaktır.

İlk olarak gerekli kütüphaneler eğer bilgisayar da yüklü değilse yüklenir ve çalışma ortamına entegre edilir. Bunlar Bölüm 13.1 de de kullanılan **Pandas** ve **SciPy** kütüphaneleridir. Panda kütüphanesinde bir değişkil olmaz yine aynı şeilde. `import pandas as pd` kodu pandas kütüphanesini çalışma ortamına entegre edilir. Burada sadece SciPy kütüphanesinde değişiklik olur. Bu sefer `scipy.stats.chi2_contingency` fonksiyonu kullanılır. Burada kütüphane ve alt modül(`stats`) aynı uyum iyiliği testiyle aynı olur sadece fonksiyon değişir ve `chi2_contingency`: Ki-Kare bağımsızlık testi için kullanılan fonksiyon olur. `import pandas as pd` ve `from scipy.stats import chi2_contingency` kodlarıyla entegre edilir. İlk çapraz tablo oluşturalım `contingency_table = pd.crosstab(index=data['Antreman_Sikligi (gun/hafta)'], columns=data['Egzersiz_Turu'])` bu kodda; `index=data['Antreman_Sikligi (gun/hafta)']`: Satır değişkeni olarak “`Antrenman Sıklığı`” seçilir. `columns=data['Egzersiz_Turu']`: Sütun

değişkeni olarak "Egzersiz_Turu" seçilir. Ve bu değişkenlerle çapraz tablo oluşturmak için pandas kütüphanesinin fonksiyonu olan **pd.crosstab** kullanılır bu da iki kategorik değişken arasındaki ilişkiyi çapraz tablo oluşturur ve çapraz tabloyu **contingency_table** değişkenine atar. Çapraz tablo her kategori kombinasyonu için gözlenen frekansları içerir. Daha sonra **chi2, p, dof, expected = chi2_contingency(contingency_table)** kodu ile ki kare bağımsızlık testi yapılır. **chi2_contingency**: Çapraz tabloyu alır ve bağımsızlık testini gerçekleştirir(Eğer iki değişken arasında bir ilişki yoksa (bağımsızlık durumu), belirli bir kategoriye ait bireylerin nasıl dağılacığını tahmin etmek için toplam satır ve toplam sütun değerlerinden hesaplanır). **chi2**: Ki-Kare test istatistiği, gözlenen ve beklenen frekanslar arasındaki farkın büyülüğünü ölçer. **p**: *p*-değeri, bu farkın tesadüfen oluşma olasılığını ifade eder **dof**: Serbestlik dereceleri. **expected**: Beklenen frekanslar, değişkenler bağımsız olsaydı her hücrede beklenen değerleri içerir (Bu, bir numpy.ndarray olarak gelir ve çapraz tablonun boyutuna (satır ve sütun sayısına) uygun şekilde düzenlenmiştir.). Sonuçlara bakılacak olursa **print("Çapraz Tablo:")** ve **print(contingency_table)**: çapraz tablo sonucunu çapraz tablo yazısını yazarak gösterir. **print("\nChi2 Test İstatistiği:", chi2)**: ki-kare test istatistiğinin sonucunu verir. **print("p-Degeri:", p)**: *p* değerinin sonucunu verir. **print("Serbestlik Dereceleri:", dof)**: serbestlik derecesinin sonucunu verir. **print("\nBeklenen Frekanslar:")**: Beklenen Frekanslar yazısını yazdırır. **print(pd.DataFrame(expected, index=contingency_table.index, columns=contingency_table.columns))**: pd.DataFrame: expected verilerini bir pandas veri çerçevesine dönüştürür. Beklenen frekansları tablo olarak düzenler. **index=contingency_table.index: index**: Orijinal çapraz tablodaki satır isimlerini (contingency_table.index) alır ve yeni DataFrame'in satır etiketleri olarak ayarlar. Bu, beklenen değerlerin satırlarının gözlenen tabloyla aynı olmasını sağlar. **columns=contingency_table.columns: columns**: Orijinal çapraz tablodaki sütun isimlerini (contingency_table.columns) alır ve yeni DataFrame'in sütun etiketleri olarak ayarlar. Bu, beklenen değerlerin sütunlarının gözlenen tabloyla aynı olmasını sağlar.

```
if p < 0.05:
    print("Sonuç: Antreman Sıklığı ve egzersiz türü birbirinden bağımsızdır.")
else:
    print("Sonuç: Antreman Sıklığı ve egzersiz türü birbirinden bağımsız değildir.")
```

if koşulu eğer *p* değeri 0.05 den küçükçe *if* koşulunu yazdırır yani değişkenler birbirinden bağımsızdır, Else koşulu ise *if* koşulunun olmadığı diğer her durumda geçerli olur ve değişkenler birbirinden bağımsız değildir.

Hipotezler kurulursa: H_0 (Sıfır Hipotezi): Değişkenler bağımsızdır (aralarında ilişki yoktur) yani Antrenman Sıklığı ve egzersiz türü birbirinden bağımsızdır. **H_1 (Alternatif Hipotez):** Değişkenler bağımsız değildir (aralarında ilişki vardır) yani Antrenman Sıklığı ve egzersiz türü birbirinden bağımsız değildir. Bu hipotezlerin kabul veya redi için gerekli koşullar; **p-değeri < 0.05**: H_0 reddedilir. Değişkenler bağımsız değildir. **p-değeri >= 0.05**: H_0 kabul edilir. Değişkenler bağımsızdır. Sonuçları çıkarılır.

Yorum (Çıktı Değerleri):

Çapraz Tablo:

Egzersiz_Turu	Cardio	HIIT	Strength	Yoga
Antreman_Sıklığı (gün/hafta)				
2	63	42	45	47
3	92	86	102	88
4	77	67	85	77
5	23	26	26	27

Chi2 Test İstatistiği: 5.758264072995584

p-Degeri: 0.763847231607104

Serbestlik Dereceleri: 9

Beklenen Frekanslar:

Egzersiz_Turu	Cardio	HIIT	Strength	Yoga
Antreman_Sıklığı (gün/hafta)				
2	51.628983	44.745118	52.236382	48.389517
3	96.443988	83.584789	97.578623	90.392600
4	80.195272	69.502569	81.138746	75.163412
5	26.731757	23.167523	27.046249	25.054471

Sonuç: Antreman Sıklığı ve egzersiz türü birbirinden bağımsız değildir.

- 1- **Çapraz Tablo:** Bu tablo, gözlenen frekansları ve farklı egzersiz türleri için antrenman sıklığını göstermektedir. **2 gün/hafta antrenman yapanlar:** Cardio: 63 kişi, HIIT: 42 kişi, Strength: 45 kişi, Yoga: 47 kişi **Toplamda 197 kişi**, haftada 2 gün antrenman yapıyor. **3 gün/hafta antrenman yapanlar:** Cardio: 92 kişi, HIIT: 86 kişi, Strength: 102 kişi, Yoga: 88 kişi **toplamda 368 kişi**, haftada 3 gün antrenman yapıyor. **4 gün/hafta antrenman yapanlar:** Cardio: 77 kişi, HIIT: 67 kişi, Strength: 85 kişi, Yoga: 77 kişi **toplamda 306 kişi**, haftada 4 gün antrenman yapıyor. **5 gün/hafta antrenman yapanlar:** Cardio: 23 kişi, HIIT: 26 kişi, Strength: 26 kişi, Yoga: 27 kişi **toplamda 102 kişi**, haftada 5 gün antrenman yapıyor.
- 2- **Chi2 Test İstatistiği (Ki-Kare Test Sonuçları):** 5.75'tir. Gözlenen ve beklenen frekanslar arasındaki farkın büyüklüğünü ölçer. Düşük bir Chi2 değeri, gözlenen frekansların beklenen değerlere oldukça yakın olduğunu gösterir.
- 3- **p-Degeri:** 0.763'tür. p-değeri, bu farkın tesadüfen oluşma olasılığını ifade eder. $p(0.763) > 0.05$ olduğunda H_0 ipotez kabul edilir. Bu nedenle **Antrenman Sıklığı** ve **Egzersiz Türü** değişkenleri bağımsızdır.
- 4- **Serbestlik Dereceleri (dof):** 9'dur Çapraz tablodaki serbestlik derecesini ifade eder. Formül: $(4-1) \times (4-1) = 9$
- 5- **Beklenen Frekanslar:** Bu tablo, Gözlenen Frekanslar ile Beklenen Frekanslar karşılaştırılarak Ki-kare değeri hesaplanır. **2 gün/hafta antrenman yapanlar için:** Cardio: 51.63 kişi, HIIT: 44.75 kişi, Strength: 52.24 kişi, Yoga: 48.39 kişi. **3 gün/hafta antrenman yapanlar için:** Cardio: 96.44 kişi, HIIT: 83.58 kişi, Strength: 97.58 kişi, Yoga: 90.39 kişi. **4 gün/hafta antrenman yapanlar için:** Cardio: 80.19 kişi, HIIT: 69.50 kişi, Strength: 81.83 kişi, Yoga: 75.16 kişi. **5 gün/hafta antrenman yapanlar için:** Cardio: 26.73 kişi, HIIT: 23.17 kişi, Strength: 27.05 kişi, Yoga: 25.05 kişi.
- 6- **Sonuç:** $p > 0.05$ olduğundan: "Antrenman Sıklığı" ve "Egzersiz Türü" arasında. Bu, kadınların ve erkeklerin egzersiz türü seçimlerinin bağımsız olduğunu gösterir. Antrenman sıklığı, egzersiz türü seçimini etkilememektedir.

13.2.2.2. pingouin ile Test Yapma:

Pingouin kütüphanesi `scipy.stats` kütüphanesiyle yaptığımız testten farklı olarak tabloya yapılan testleri de ekler.

```
import pandas as pd
import pingouin as pg

# Çapraz tabloyu manuel oluşturma
observed_manual = pd.crosstab(index=data["Antreman_Sikligi (gun/hafta)"], columns=data["Egzersiz_Turu"], margins=False)

# Pingouin kullanarak bağımsızlık testi
expected, observed, stats = pg.chi2_independence(data, x='Antreman_Sikligi (gun/hafta)', y='Egzersiz_Turu')

# Sonuçların yazdırılması
print("Pingouin ile Hesaplanan Gözlenen Frekanslar:")
print(observed)

print("\nPingouin ile Hesaplanan Beklenen Frekanslar:")
print(expected)

print("\nPingouin Test İstatistikleri:")
print(stats)
```

Pingouin ile Hesaplanan Gözlenen Frekanslar:

Egzersiz_Turu	Cardio	HIIT	Strength	Yoga
Antreman_Sikligi (gun/hafta)				
2	63	42	45	47
3	92	86	102	88
4	77	67	85	77
5	23	26	26	27

Pingouin ile Hesaplanan Beklenen Frekanslar:

Egzersiz_Turu	Cardio	HIIT	Strength	Yoga
Antreman_Sikligi (gun/hafta)				
2	51.628983	44.745118	52.236382	48.389517
3	96.443988	83.584789	97.578623	90.392600
4	80.195272	69.502569	81.138746	75.163412
5	26.731757	23.167523	27.046249	25.054471

Pingouin Test İstatistikleri:

	test	lambda	chi2	dof	pval	cramer	power
0	pearson	1.000000	5.758264	9.0	0.763847	0.044415	0.121893
1	cressie-read	0.666667	5.721319	9.0	0.767453	0.044272	0.121350
2	log-likelihood	0.000000	5.655378	9.0	0.773852	0.044016	0.120382
3	freeman-tukey	-0.500000	5.612654	9.0	0.777972	0.043850	0.119757
4	mod-log-likelihood	-1.000000	5.575510	9.0	0.781536	0.043704	0.119214
5	neyman	-2.000000	5.517372	9.0	0.787080	0.043476	0.118367

Gördüğü gibi aslında gözlenen ve beklenen değerleri bulmada bir `scipy.stats` kütüphanesinde yapılan kodlarda pek bir değişiklik olmamıştır sadece `pingouin` test istatistikleri çıktısında bazı satırlar eklenmiştir Şimdi sadece bu satırların ne olduğunu ve onu yorumlayamaya bakalım.

test: Kullanılan test çeşidini belirtir. (Pearson, Cressie-Read vb.)

lambda: Testin uygulandığı lambda parametresidir. Bu değer, farklı test türlerini ayırt etmek için kullanılır. Örneğin, Pearson için lambda 1.0, log-likelihood için 0.0'dır.

chi2: Ki-kare test istatistiğidir.

dof: Serbestlik derecesidir

pval: p-değeri, hipotez testinin sonucunu değerlendirmek için kullanılır.

cramer: Cramér's V değeri, etki büyüklüğünü ölçer. Bu değer 0 ile 1 arasında değişir. 0'a yakın değerler ilişkinin zayıf olduğunu, 1'e yakın değerler ise güçlü bir ilişki olduğunu gösterir.

power: Testin istatistiksel gücüdür. Güç, alternatif hipotezin doğru kabul edilme olasılığını gösterir. Güç ne kadar yüksekse, testin anlamlı bir farkı tespit etme yeteneği o kadar iyidir.

Test Türleri

1. Pearson ($\lambda = 1.0$):

- Bu, standart Pearson Ki-kare testidir.
- Test istatistiği (χ^2) = 5.758264
- p -değeri = 0.763847
- Cramér's V = 0.044415
- Yorum: p -değeri > 0.05, bu nedenle sıfır hipotezi kabul edilir (değişkenler bağımsızdır).

2. Cressie-Read ($\lambda = 0.6667$):

- Bir modifiye Ki-kare testidir.
- Test istatistiği: 5.723119
- p -değeri: 0.767453
- Cramér's V: 0.044272

3. Log-Likelihood ($\lambda = 0.0$):

- Bu test logaritmik olasılık oranını kullanır.
- Test istatistiği: 5.655378
- p -değeri: 0.773852
- Cramér's V: 0.044016

4. Freeman-Tukey ($\lambda = -0.5$):

- Bu test istatistiği, ara değerli bir modifikasyon sunar.
- Test istatistiği: 5.612654
- p -değeri: 0.777972
- Cramér's V: 0.043850

51

5. Modified Log-Likelihood ($\lambda = -1.0$):

- Log-likelihood testinin bir türevi.
- Test istatistiği: 5.575510
- p -değeri: 0.781536
- Cramér's V: 0.043704

6. Neyman ($\lambda = -2.0$):

- Neyman testi, Ki-kare testinin özel bir modifikasyonudur.
- Test istatistiği: 5.517372
- p -değeri: 0.787080
- Cramér's V: 0.043476

Yorum: p-değeri: Tüm testlerde p -değeri > 0.05 olduğu için sıfır hipotezi kabul edilir. "Antreman Sıklığı" ve "Egzersiz Türü" birbirinden bağımsızdır. **Cramér's V:** Tüm testler için Cramér's V değeri çok küçük (~0.04). Bu değer, değişkenler arasındaki ilişkinin etki büyüklüğünün çok zayıf olduğunu gösterir. **Istatistiksel Güç (power):** Testlerin gücü ~ 0.12'dir., bu da testin anlamlı bir farkı tespit etme olasılığının düşük olduğunu gösterir.

13.3. Sürekliklilik Düzeltmesi:

Yates düzeltmesi, 2×2 olasılık tablosu analizlerinden (örneğin, Pearson ki-kare ve McNemara ki-kare) elde edilen ki-kare değerlerine ve serbestlik derecesi 1 olan tablolara yapılan bir ayarlamadır. Yates düzeltmesinin mantığı, olasılık tablosu analizlerinin ikili verilere dayanması ve istatistiksel ki-kare dağılıminin sürekli (ikili olmaktan ziyade) olması nedeniyle, daha doğru sonuçlar elde etmek için olasılık tablosu analizlerine bir ayarlama uygulanması gereği gerçeğine dayanır. Düzeltme, gözlenen ve beklenen hücre frekansları arasındaki her mutlak farktan 0.5 çıkarılmasından oluşur. Yates süreklilik düzeltmesinin toplam gözlem sayısı 40'dan düşük olduğu ($N < 40$) hallerde kullanılması tavsiye edilmektedir

$$\chi^2 = \sum_{i=1}^r \frac{(|E_i - O_i| - 0.5)^2}{E_i}$$

Eğer Python'da 2×2 lik bir tabloda Yates düzeltmesi yapmak istenirse;

Scipy ile Yates Düzeltmesi:

`chi2, p, dof, ex = chi2_contingency(observations, correction=True)`

Pingouin ile Yates Düzeltmesi:

`expected, observed, stats = pg.chi2_independence(df, x='species', y='choice', correction=True)`

`correction=True` parametresi süreklilik düzeltmesinin aktif olmasını sağlar.

52

13.4. Etki Büyüklüğü:

Bir değişkenin ya da bir etkinin istatistiksel olarak ne kadar önemli veya güçlü olduğunu belirtir. Bu terim, özellikle hipotez testi sonuçlarını değerlendirirken istatistiksel anlamlılığın ötesine geçerek, araştırma sonuçlarının pratik önemini değerlendirmeye yardımcı olan bir ölçüdür.

Istatistiksel bir teste, p değeri, sıfır hipotezinin reddedilip edilmemesi konusunda bir karar vermenize yardımcı olabilir. Ancak, p değeri tek başına yeterli değildir. Etki büyülüğu, gerçek dünyada ne kadar önemli bir fark olduğunu değerlendirmek için kullanılır. Etki büyülüğu türleri vardır (Örn: Pearson r, Cramer's V vs.) bunu da 13.2.2.2ç bölümde **pingouin** kütüphanesiyle yapılan teste çıktıının verdiği test türleridir.

13.5. Testlerin Varsayımları:

Ki-kare testinin doğru kullanılabilmesi için çok önemli iki temel varsayımin kullanıcılarda iyi bilinmesi gereklidir. Bu varsayımlar şunlardır;

i) Gruplar birbirinden bağımsız olmalıdır. Bağımlı gruplara, normal ki-kare testi uygulanamaz. Bu gruplar için, ki-kare testi ayrı bir yöntemle yapılır.

ii) Ki-kare dağılımı, sürekli bir dağılımdır. B'lerden herhangi biri 5'den küçük ise dağılım kesikli ve çarpık olur. Bu yüzden, test sonucunda elde edilen ki-kare değeri, ki-kare dağılımına uygunluk göstermez. Böyle bir durumda;

- a) 2×2 düzenlerinde, Fisher kesin ki-kare testi uygulanır.
- b) $2 \times c$ veya $r \times 2$ düzenlerinde, ki-kare testi uygulanmak isteniyorsa satır veya sütunlar birleştirilerek 5'ten küçük değerin ortadan kaldırılmasına çalışılır. Bu mümkün olmazsa Kolmogorov-Smirnov testi uygulanır.
- c) İlgili yazarlar $r \times c$ düzenlerinde, 5'ten küçük O_i 'ler testin sonucunu büyük oranda etkilemeyeceğini belirtmektedirler. Satır veya sütunların birleştirilerek 5'ten küçük değerin ortadan kaldırılması daha uygun olur.

13.6. Fisher Kesin Testi:

Fisher'in kesin testi, iki kategorik değişken arasında istatistiksel olarak anlamlı bir ilişkinin olup olmadığını belirler. Fisher'in kesin testi ki-kare istatistiğini ve örneklem dağılımını kullanmaz. Bunun yerine, gözlemlenen tabloyla aynı satır ve sütun toplamlarına (yani marginal_dağılımlara) sahip tüm olası olasılık tablolarının sayısını hesaplar. Daha sonra, gözlemlenen tablodan daha uç olan olası tabloların oranını bularak p-değeri için olasılığı hesaplar. Teknik olarak, Fisher'in kesin testi tüm örneklem büyüklükleri için uygundur. Ancak, olası tabloların sayısı üstel bir oranda artar ve kısa sürede kullanışsız hale gelir. Bu nedenle, bu test daha küçük örneklem büyüklükleri için kullanılır. Fisher'in kesin testi; hücre sayısı 20'den küçük olduğunda ve bir hücrenin beklenen değeri 5 veya daha az olduğunda kullanılır ayrıca sütun veya satır marginal değerleri aşırı derecede dengesizdir. Yine data verisinden kategorik değişken olan “Cinsiyet” ve “Egzersiz_Turu” değişkenleri arasında Fisher'in kesin testi yapılacaktır bu testte 2×2 lik tablo kullanılacağı için “Egzersiz_Turu” değişkeninden “Cardio” ve “HITT” kategorileri üzerinden Python kodlarıyla yapılacaktır.

```
#13.6 Fisher Kesin Testi

import pandas as pd
from scipy.stats import fisher_exact

# Çapraz tablo oluşturma
contingency_table = pd.crosstab(index=data['Cinsiyet'], columns=data['Egzersiz_Turu'])

# Cardio ve HITT Kategorilerini seçip 2x2 lik tabloya indirgeme
contingency_table_2x2 = contingency_table.loc[['Male', 'Female'], ['Cardio', 'HITT']]

# Fisher Kesin Testi
oddsratio, p_value = fisher_exact(contingency_table_2x2)

# Tabloyu ve sonuçları yazdırma
print("2x2 Çapraz Tablo:")
print(contingency_table_2x2)
print("\nFisher Kesin Testi Sonuçları:")
print("Odds Oranı:", oddsratio)
print("p-Değeri:", p_value)

# Hipotez sonucu
if p_value < 0.05:
    print("Sonuç: İki değişken arasında istatistiksel olarak anlamlı bir ilişki vardır.")
else:
    print("Sonuç: İki değişken arasında istatistiksel olarak anlamlı bir ilişki yoktur.")
```

*İlk önce gerekli kütüphaneler eğer bilgisayar da yüklü değilse yüklenir diğer bölmelerde yükleniği gibi yüklenir ve çalışma ortamına entegre edilir Bunlar diğer bölmelerde de kullanılan Pandas ve SciPy kütüphanelerdir. Panda kütüphanesinde bir değişiklik olmaz yine aynı şekilde. `import pandas as pd` kodu pandas kütüphanesini çalışma ortamına entegre edilir. SciPy kütüphanesi içinde alt modül(stats) diğer bölmelerdeki gibi stats olur fakat test değiştiği için fonksiyon değişir ve `fisher_exact`: (2x2 tablolar için Fisher Kesin Testi yapar.) fonksiyonu olur. Daha sonra Fisher kesin testi için gerekli olan çapraz tablo `contingency_table = pd.crosstab(index=data['Cinsiyet'], columns=data['Egzersiz_Turu'])` kodu ile oluşturulur. `pd.crosstab`: İki kategorik değişkenin (Cinsiyet ve Egzersiz_Turu) frekanslarını içeren bir çapraz tablo (contingency table) oluşturur. `index=data['Cinsiyet']`: Satır değişkeni olarak "Cinsiyet" seçilir (Male, Female).`columns=data['Egzersiz_Turu']`: Sütun değişkeni olarak "Egzersiz_Turu" seçilir (Cardio, HIIT, Strength, Yoga). Bu tablo, her bir "Cinsiyet" için farklı egzersiz türlerine katılım sayısını gösterir ve bu oluşan çapraz tablo `contingency_table` değişkenine atanır. Şimdi de Fisher kesin testinde 2x2'lik tablo olması gereği için "Egzersiz_Turu" değişkenindeki sadece iki kategori almamız gerekecek bunu da `contingency_table_2x2 = contingency_table.loc[['Male', 'Female'], ['Cardio', 'HIIT']]` kodu ile yapılır. `loc`: Satır ve sütun isimlerine göre belirli kategorileri seçmek için kullanılır. `[['Male', 'Female']]`: "Male" ve "Female" satırlarını seçer. `[['Cardio', 'HIIT']]`: Sadece "Cardio" ve "HIIT" sütunlarını seçer. Ve Çapraz tablo, sadece "Male" ve "Female" satırları ile "Cardio" ve "HIIT" sütunlarını içeren bir 2x2 tabloya indirgenir ve bu indirgenen çapraz tablo `contingency_table_2x2 değişkenine atanır`. Daha sonra Fisher kesin testi `oddsratio, p_value = fisher_exact(contingency_table_2x2)` kodu ile uygulanır. `fisher_exact`: 2x2 çapraz tabloyu alır ve Fisher Kesin Testi uygular. **Odds Oranı (oddsratio)**: Bir olayın diğerine göre gerçekleşme olasılığını ifade eder. Örneğin, erkeklerin "Cardio" egzersizine katılma olasılığı kadınlarinkine göre ne kadar fazla veya azdır. **p-Degeri (p_value)**: Bu, gözlenen tablo ile bağımsızlık varsayımlı altındaki tablo arasındaki farkın anlamlı olup olmadığını gösterir. Eğer $p < 0.05$ ise, değişkenler arasında anlamlı bir ilişki vardır. Tablolari ve sonuçları yazdırma; `print("2x2 Çapraz Tablo:")`: 2x2 Çapraz Tablo yazısının yazdırır. `print(contingency_table_2x2)`: `contingency_table_2x2` değişkenin sonucunu atar. `print("\nFisher Kesin Testi Sonuçları:")`: Fisher Kesin Testi Sonuçları: yazısını yazdırır (buradaki \n yazının bir alt satıra inmesi için kullanılır.) `print("Odds Oranı:", oddsratio)`: Odds Oranını ve değerini = **0.9609** yazdırır. `print("p-Degeri:", p_value)`: p-Degerini yazısını ve değerini= **0.8544** yazdırır. Son olarak Hipotez test sonuçlarını yazar ve testin sonucunu verir.*

`if p_value < 0.05:`

```
    print("Sonuç: İki değişken arasında istatistiksel olarak anlamlı bir ilişki vardır.")
else:
    print("Sonuç: İki değişken arasında istatistiksel olarak anlamlı bir ilişki yoktur.")
```

if koşulu eğer p değeri 0.05 den küçükçe if koşulunu yazdırır yani İki değişken arasında istatistiksel olarak anlamlı bir ilişki vardır çıktımasını verir. Else koşulu ise if koşulunun olmadığı diğer her durumda geçerli olur ve iki değişken arasında istatistiksel olarak anlamlı bir ilişki yoktur çıktımasını verir.

Hipotezler kurulursa: H_0 (Null Hipotezi): İki değişken arasında istatistiksel olarak anlamlı bir ilişki vardır. H_1 (Alternatif Hipotez): İki değişken arasında istatistiksel olarak anlamlı bir ilişki yoktur. Bu hipotezlerin kabul veya reddi için gerekli koşullar; **p-değeri <0.05**: H_0 reddedilir. İki değişken arasında istatistiksel olarak anlamlı bir ilişki yoktur ve cinsiyet ve egzersiz türü bağımsızdır (ilişki yoktur). **p-değeri >= 0.05**: H_0 kabul edilir. İki değişken arasında istatistiksel olarak anlamlı bir ilişki vardır ve cinsiyet ve egzersiz türü bağımsız değildir (ilişki vardır). Sonuçları çıkarılır.

Yorum (Çıktıdaki Değerler):

2x2 Çapraz Tablo:

Egzersiz_Turu	Cardio	HIIT
Cinsiyet		
Male	129	114
Female	126	107

Fisher Kesin Testi Sonuçları:

Odds Oranı: 0.9609440267335004

p-Degreri: 0.8544090398364391

Sonuç: İki değişken arasında istatistiksel olarak anlamlı bir ilişki yoktur.

- 1- 2x2 Çapraz Tablo:** Bu çapraz tablonun amacı Erkek ve kadınların "Cardio" ve "HIIT" egzersiz türlerini tercih etme oranları arasında bir ilişki olup olmadığını test etmek. Bu tablo, gözlenen frekansları içeriyor. Male (Erkek): "Cardio" yapan erkek sayısı: 129 ve "HIIT" yapan erkek sayısı: 114. Female (Kadın): "Cardio" yapan kadın sayısı: 126 ve "HIIT" yapan kadın sayısı: 107.

2- Fisher Kesin Testi Sonuçları:

Odds Oranı: 0.9609 Burada, erkeklerin "Cardio" yapma olasılığının kadınların "Cardio" yapma olasılığına göre oranı hesaplanmıştır. **Odds oranı 1 ise:** Hiçbir fark yoktur. **Odds oranı <1 ise:** İlgili olay kadınarda daha olasıdır. **Odds oranı > 1 ise:** İlgili olay erkeklerde daha olasıdır şeklinde yorumlanır. Burada değer 1'e çok yakın olduğu için bu, erkeklerin "Cardio" yapma olasılığı ile kadınların "Cardio" yapma olasılığının neredeyse eşit olduğunu gösterir.

p-Degreri: 0.8544 p-değeri, gözlenen farkın tesadüfen oluşma olasılığını ifade eder. Burada, $p > 0.05$ olduğu için değişkenler arasında anlamlı bir ilişki yoktur.

- 3- Sonuç:** İki değişken arasında istatistiksel olarak anlamlı bir ilişki yoktur. p-değeri(0.8544) 0.05'ten büyük olduğu için, H_0 hipotezini reddetmiyoruz. Bu, Cinsiyet ve Egzersiz Türü arasında anlamlı bir ilişki olmadığını gösterir.

NOT: Burada ki-kare bağımsızlık testindeki pingouin kütüphanesiyle de hesaplanabilir.

13.7. McNemar Testi:

McNemar testi, eşleştirilmiş nominal verileri analiz etmek için kullanılan parametrik olmayan bir testtir. 2×2 'lik bir olasılık tablosunda yapılan bir testtir ve iki değişkenin marginal homojenliğini kontrol eder. Test, iki kategoriye sahip bir nominal değişken ve iki bağımlı gruba sahip bir bağımsız değişken gerektirir. Ayrıca, bağımlı değişkendeki iki grup karşılıklı olarak birbirini dışlayan olmalıdır, yani birden fazla grupta olamaz. McNemar testi için gereken minimum örneklem büyüklüğü en az on uyumsuz çifttir. McNemar testi, aynı popülasyonda iki zaman noktasında ikili bir özellikte oranlarda istatistiksel olarak anlamlı bir değişiklik olup olmadığını değerlendirir.

13.7.1. Python'da McNemar Testinin Yapılması:

Burada kullanılan data verisinden “Dinlenme_BPM” ve “Max_BPM” değişkenlerinden spordan önceki nabız ve spordan sonraki nabızı için eşik değer belirlenir ve bu eşik değerin altında veya üstünde kalmasında göre 0-1 değerleri ve cinsiyetlere de kadın erkeğe 0-1 değerleri verilir ve ikili (binary) bir kategorik değişkene dönüştürüp çapraz tablo yapılır ve McNemar testi buna göre uygulanır. Bu anlatılanları şimdilik kodda gösterilecek olursa;

```
#13.7 McNemar Testi

import pandas as pd
import pingouin as pg

# Dinlenme BPM ve Max BPM için eşik değer belirleme
dinlenme_threshold = 60
max_threshold = 160

# ikili bir değişken oluşturma: Dinlenme ve Max BPM eşik üzerinde mi?
data['Dinlenme_Yuksek'] = (data['Dinlenme_BPM'] > dinlenme_threshold).astype(int)
data['Max_Yuksek'] = (data['Max_BPM'] > max_threshold).astype(int)

# "birinci_faktör" ve "ikinci_faktör" sütunlarını oluşturma
data['birinci_faktör'] = data['Dinlenme_Yuksek']
data['ikinci_faktör'] = data['Max_Yuksek']

# Pingouin McNemar testi
observed, stats = pg.chi2_mcneumar(data, 'birinci_faktör', 'ikinci_faktör')

# Sonuçları yazdırma
print(f"\nBelirlenen eşik değerler:\nDinlenme_BPM: {dinlenme_threshold}, Max_BPM: {max_threshold}")
print("\nÇapraz Tablo (Observed):")
print(observed)

print("\nMcNemar Testi Sonuçları:")
print(stats)
```

Burada ilk olarak yine gerekli kütüphaneleri yükleme ve entegre etme ile işe başlanır. Kullanılacak kütüphaneler pandas ve pingouin kütüphaneleridir. Pandas kütüphanesi zaten diğer bölmelerde yüklü olduğu için yükleme yapılmaz sadece **import pandas as pd** kodu ile ortama entegre edilir. Fakat pingouin verisi daha önce yüklü olmadığı için **pip install pingouin** (conda kullanılıyorsa **conda install pingouin**, ama kod kısmına yazılmıyor kullanılan arayüzde indiriliyor o yüzden kodlarda yok.) yazarak indirilir. Daha sonra **import pingouin as pg** kodu ile ortama entegre edilir. **pandas**: Veri işleme ve manipülasyon için kullanılır. **pingouin**:

*İstatistiksel testler ve analizler için kullanılan bir Python kütüphanesidir. McNemar testi bu kütüphane ile gerçekleştirilir. "Dinlenme_BPM" değişkeni için `dinlenme_threshold = 60` kodu ile eşik değeri 60 atanır. Bu değer, dinlenme durumundaki nabız için eşik olarak kabul edilir. Daha sonra "Max_BPM" için `max_threshold = 160` kodu yazılarak 160 olarak eşik atanır. Bu eşik değerleri için yeni bir ikili değişken oluşturulur. İkili bir değişken oluşturmak için `data['Dinlenme_Yuksek'] = (data['Dinlenme_BPM'] > dinlenme_threshold).astype(int)` kodu kullanılır ve bu Dinlenme_BPM eşik değerden büyükse, bu sütunda 1 atanır. Küçükse 0 atanır demektir. `data['Max_Yuksek'] = (data['Max_BPM'] > max_threshold).astype(int)` Max_BPM değişkeni içinde Max_BPM eşik değerden büyükse, bu sütunda 1 atanır. Küçükse 0 atanır. Burada eşik değerleri belirlenenenden fazlaysa `True`, aksi halde `False` döner. `.astype(int)`: Bu `True` ve `False` değerlerini sırasıyla 1 ve 0 olarak kodlar. Mcnemar testi için ikili sütunlar oluşturmak `data['birinci_faktör'] = data['Dinlenme_Yuksek']` kodunu bakılırsa `data['birinci_faktör']`: Veri çerçevesindeki Bireylerin dinlenme BPM durumunu temsil eder ve yeni bir sütun oluştur (0: düşük, 1: yüksek). "Dinlenme" sütundaki değerleri yeniden kodlar `data['ikinci_faktör'] = data['Max_Yuksek']` koduna bakılırsa. `data['ikinci_faktör']`: Bu işlem sonucunda yeni bir sütun oluşturulur: `ikinci_faktör` Bireylerin maksimum BPM durumunu temsil eder (0: düşük, 1: yüksek). McNemar testi uygulaması `observed`, `stats = pg.chi2_mcnemar(data, 'birinci_faktör', 'ikinci_faktör')` kodu ile yapılır. `pg.chi2_mcnemar`: İki ikili değişken arasında McNemar testi uygular. Parametreler: `data`: Veri çerçevesi. `birinci_faktör`: İlk değişken (`Dinlenme_BPM`). `ikinci_faktör`: İkinci değişken (`Max_BPM`). **Dönen Değerler:** `observed`: Gözlenen frekansların bulunduğu çapraz tablo. `stats`: Test istatistikleri (`chi2`, `p-değeri` vb.). Eşik değer yazdırılması için: `print(f"\nBelirlenen eşik değerler:\nDinlenme_BPM: {dinlenme_threshold}, Max_BPM: {max_threshold}")` kodu yazılır ve kalp atışları için belirlenen eşik değerleri ekrana yazdırılır. Çapraz tablo ve test sonuçlarının yazdırılması: `print("\nÇapraz Tablo (Observed):")` ve `print(observed)` `observed`: "birinci_faktör" ve "ikinci_faktör" arasındaki gözlenen frekansları gösteren bir çapraz tablo yazdırır. `print("\nMcNemar Testi Sonuçları:")` ve `print(stats)`: `stats` McNemar testi sonuçlarını içerir: `chi2`: Chi-Kare test istatistiği. `p-approx`: Yaklaşık `p-değeri`. `p-exact`: Tam `p-değeri`.*

Hipotezler kurulursa: **H_0 (Sıfır Hipotezi):** İki değişken arasında istatistiksel olarak anlamlı bir fark vardır. **H_1 (Alternatif Hipotez):** İki değişken arasında istatistiksel olarak anlamlı bir fark yoktur. Bu hipotezlerin kabul veya reddi için gerekli koşullar; **$p-değeri < 0.05$** : H_0 reddedilir. Cinsiyet (`birinci_faktör`) ve su tüketimi durumu (`ikinci_faktör`) arasında fark yoktur. Değişkenler bağımsızdır. **$p-değeri \geq 0.05$** : H_0 kabul edilir. Cinsiyet ve su tüketimi durumu arasında anlamlı bir fark vardır. Değişkenler bağımsız değildir.

Yorum (Çıktı Değerleri):

Belirlenen eşik değerler:
Dinlenme_BPM: 60, Max_BPM: 160

Çapraz Tablo (Observed):

	ikinci_faktör	0	1
birinci_faktör			
0		9	402
1		9	553

McNemar Testi Sonuçları:

	chi2	dof	p-approx	p-exact
mcnemar	373.878345	1	2.677749e-83	3.265826e-106

İlk olarak "Dinlenme_BPM" değişkenini iki kategoriye ayırmak için belirlenen eşik değer Dinlenme_BPM: 60 olarak alınmıştır ve "Max_BPM" için belirlenen eşik değer: 160 ekrana yazdırılmış. Çapraz Tablo (Observed): Bu tablo, **birinci_faktör** (dinlenme durumunda kalp atışı) ve **ikinci_faktör** (maksimum kalp atışı) arasındaki gözlenen frekansları gösterir. **9 kişi** hem Dinlenme_BPM hem de Max_BPM eşik değerinin altında (0, 0). **402 kişi** Dinlenme_BPM düşük (≤ 60), ancak Max_BPM yüksek (> 160) (0, 1). **9 kişi** Dinlenme_BPM yüksek (> 60), ancak Max_BPM düşük (≤ 160) (1, 0). **553 kişi** her iki durumda da eşik değerinin üzerinde (1, 1). Olarak bulunur ve çapraz tabloya yazdırılır. McNemar Testi Sonuçları:

- 1- **Test İstatistiği (chi2):** $\text{chi2} (\text{Chi-Kare Test İstatistiği}) = 373.878345$ bu, gözlenen frekanslar ile bağımsızlık varsayımlı altındaki beklenen frekanslar arasındaki farkın büyülüüğünü ölçer. Bu kadar yüksek bir değer, iki durum arasında anlamlı bir fark olduğunu güçlü bir şekilde gösterir.
- 2- **Serbestlik Derecesi (dof):** $\text{dof} (\text{Degrees of Freedom}) = 1$ çapraz tablodaki serbestlik derecesidir. McNemar testi için daima 1'dir. Çünkü serbestlik derecesi (satır sayısı – 1) x (sutun sayısı – 1) = 1 formülü ile bulunur. McNemar testi için de her zaman 2×2 'lik tablo oluşturulduğu için serbestlik derecesi her zaman 1'dir.
- 3- **Yaklaşık p-Değeri (p-approx):** $p\text{-approx} = 2.68 \times 10^{-83}$ Chi-Kare testinin yaklaşık p-değeri. Bu değer, gözlenen frekanslarla beklenen frekanslar arasındaki farkın tesadüfen oluşma olasılığını ifade eder. $p < 0.05$ olduğundan, fark istatistiksel olarak anlamlı olduğunu söyler.
- 4- **Tam p-Değeri (p-exact):** $p\text{-exact} = 3.27 \times 10^{-106}$ McNemar testinin tam p-değeri. (daha hassas bir hesaplama). $p < 0.05$, gözlenen farkın anlamlı olduğunu doğrular.
- 5- **Sonuç: $p > 0.05$ (yaklaşık ve tam p-değerler):** H_0 reddedilir. Dinlenme_BPM ve Max_BPM arasında istatistiksel olarak anlamlı bir fark olduğunu gösteriyor. Yani, bireylerin dinlenme ve maksimum kalp atış hızı durumları birbirine bağımlıdır. U, bireylerin dinlenme hallerine göre maksimum kalp atış hızında önemli bir farklılığı gösterdiğini işaret eder.