# *İstatistiksel Yazılımlar Ders Notları*

*Bir istatistikçiyi deneyi gerçekleştirdikten sonra çağırmak, ondan otopsi gerçekleştirmesini istemekten fazlası değildir: Size deneyinizin neden öldüğünü söyleyebilir.*

---

*1938*
PROFESOR SIR RONALD AYMLER FISHER

Tolga BERBER
İstatistik ve Bilgisayar Bilimleri Bölümü
tberber@ktu.edu.tr

# İçindekiler

BÖLÜM

# 1 İstatistiksel Yazılımlar

- Bu güne kadar gördüğünüz yöntemlerin uygulaması için kullanılırlar
- Uygulama amaçlarına göre;
  - Veri Madenciliği Yazılımları
  - Analiz Yazılımları
- Türlerine göre;
  - Mevcut Yazılımların İstatistiksel Eklentileri
  - Paket Programlar
  - Programlama Dili Eklentileri
  - Programlama Dilleri

## 1.1 Uygulama Amaçlarına Göre İstatistiksel Yazılımlar

### 1.1.1 Veri Madenciliği Yazılımları

- Büyük miktardaki verilerden bilgi ayıklamak
- Akıllı sistemler oluşturmak
- Kümeleme ve sınıflama gibi makine öğrenmesi yöntemlerini uygulamak
- Metin, görüntü gibi bilgilerden bilgi ayıklayıp anlamlandırmak

#### 1.1.1.1 Örnekler

- SPSS Modeler
- R
- S
- RapidMiner
- Weka
- SQL Server Analysis Services
- Oracle Data Mining

### 1.1.2 Veri Analizi Yazılımları

- Anket sonuçları değerlendirmek
- Hipotez testleri gerçekleştirmek
- Tanımlayıcı istatistikler ve grafikler oluşturmak

#### 1.1.2.1 Örnekler

- SPSS
- R
- SAS
- Excel
- Libre Office Calc
- MiniTab
- STATA

Bu ders kapsamında çoğunlukla Veri Analizi yazılımlarından bahsedilecektir.

## 1.2 Türlerine Göre İstatistiksel Yazılımlar

### 1.2.1 İstatistiksel Eklentiler

- Genellikle elektronik tablolama programlarında bulunurlar
- Formüller veya fonksiyonlar yardımıyla bazı istatistiksel işlemlerin yapılmasını sağlarlar

#### 1.2.1.1 Örnekler

- Microsoft Excel
- Libre Office Calc

### 1.2.2 Paket Programlar

- Tamamıyla istatistiksel analizler gerçekleştirmek amacıyla geliştirilmişlerdir
- Kullanımları eklentilere göre daha kolaydır (?)

#### 1.2.2.1 Örnekler

- SPSS
- SAS
- MiniTAB

### 1.2.3 Programlama Dili Eklentileri

- Mevcut yazılım geliştirme dillerine ek fonksiyon kütüphanesi olarak gelirler

#### 1.2.3.1 Örnekler

- Matlab: Statistics ToolBox
- C++: Boost İstatistik Kütüphanesi
- C#: IMSL .NET
- GNU Octave: Statistics ToolBox
- Java: IMSL .NET

### 1.2.4 İstatistiksel Programlama Dilleri

- Tamamıyla istatistiksel analiz ve veri madenciliği yapmak amacıyla geliştirilmişlerdir

#### 1.2.4.1 Örnekler

- GNU R
- S

## 1.3 R Programlama Dili

- Açık Kaynak
- Ücretsiz
- Platform Bağımsız
- Güncel İleri İstatistiksel Yöntemleri İçerir
- Güçlü Grafik Kabiliyetleri Vardır
- Yeni İstatistiksel Yöntemler Geliştirilmesine Olanak Sunar
- Bir İnteraktif Programlama Dili'dir

### 1.3.1 Tarihçesi

- Bell Laboratuvarlarında Geliştirilmeye Başlandı
  - İlk dilin adı S'tir
  - 1988 - S2: RA Becker, JM Chambers, A Wilks
  - 1992 - S3: JM Chambers, TJ Hastie
  - 1998 - S4: JM Chambers
- İlk R sürümü, S dilinden esinlenerek 1990 yılında geliştirildi
  - Ross Ishaka, Robert Gentleman
- 1997'den itibaren 15 Çekirdek R geliştiricisi yazılımı geliştirmektedir.
- R'ı kullanabilmek için önce R yorumlayıcısı kurmanız gerekiyor, bunun için
  - 32/64 bit işletim sistemleri:
    * https://cran.r-project.org/bin/windows/base/

    * Burada hem işletim sisteminize uygun R yorumlayıcısı hem de RTools paketini indirin.

- R'ın görsel olarak kullanabilmek için de Rstudio yazılımını kullanacağız
  - https://download1.rstudio.org/RStudio-1.0.136.exe

Yazılımlarını indirip kurun

## 1.4 İlk R Programınız

R Studio'yu açın ve aşağıdaki komutları girin. Her komuttan sonra Enter tuşuna basınız.

```
a <- "Merhaba Dünya"
print(a)
```

```
## [1] "Merhaba Dünya"
```

R Studio'dan çıkmak için iki yolunuz var.
1- Pencereyi kapatabilirsiniz. 2- Konsoldan **q()** komutunu girebilirsiniz.

BÖLÜM

# 2 R ile Çalışma

## 2.1 R Çalışma Ortamı

- R ile yaptığınız çalışmaların kaydedildiği klasördür
  - setwd() fonksiyonu ile değiştirilir
  - getwd() fonksiyonu ile öğrenilir
- Önereceğim yol;
  - Çalışmaya başlamadan önce bir klasör oluşturun
  - Çalışma ortamınızı ayarlayın
  - Analizinizi Gerçekleştirin

## 2.2 R Veri Türleri

- Her programlama dili veri tutmaya ihtiyaç duyar
- R istatistiksel olarak kullanılabilecek veri türleri tanımlar
- C/C++ aksine R da değişken tanımlarken tür belirtmek zorunda değilsiniz
- R Programlama dilinde temelde iki farklı veri türü yer alır
  - Atomik Vektörler
  - R Nesneleri

### 2.2.1 Atomik Vektörler

- Sadece bir değer saklayabilen veri türleridir
  - Vektörlerin Altı Sınıfı olarak ta adlandırılırlar
- Altı adet atomik vektör türü vardır
  - Logical (Mantıksal)

- Numeric (Sayısal)
- Integer (Tamsayı)
- Complex (Karmaşık Sayı)
- Character (Metin)
- Raw (Ham)

```
v <- TRUE
print(class(v))
```

```
## [1] "logical"
```

```
v <- 23.5
print(class(v))
```

```
## [1] "numeric"
```

```
v <- 2L
print(class(v))
```

```
## [1] "integer"
```

```
v <- 2+5i
print(class(v))
```

```
## [1] "complex"
```

```
v <- "TRUE"
print(class(v))
```

```
## [1] "character"
```

```
v <- charToRaw("Merhaba")
print(class(v))
```

```
## [1] "raw"
```

```
print(v)
```

```
## [1] 4d 65 72 68 61 62 61
```

## 2.2.2 R Nesneleri

- R programlama dilinde bir çok nesne bulunmaktadır
- En sık kullanılanları
    - Vektörler
    - Listeler
    - Matrisler
    - Diziler
    - Faktörler
    - Veri Çerçeveleri (Data Frames)

#### 2.2.2.1 Vektörler

- R programlama dilindeki en temel nesnedir
- Tek türde veri saklarlar
    - Atomik Türler
    - R Nesneleri
- Birden çok değer saklayabilirler
- c() fonksiyonu ile oluşturulurlar
- Bir yada birden fazla elemana sahip olabilirler

```r
v <- 5

print(v)
```

```
## [1] 5
```

```r
v <- c(1,2,3)

print(v)
```

```
## [1] 1 2 3
```

- Bir seri içerebilirler

```r
v <- 6.6:12.6

print(v)
```

```
## [1]  6.6  7.6  8.6  9.6 10.6 11.6 12.6
```

```r
v <- 3.8:11.4

print(v)
```

```
## [1]  3.8  4.8  5.8  6.8  7.8  8.8  9.8 10.8
```

```r
v <- seq(10.5, 2.7, by=-0.2)

print(v)
```

```
##  [1] 10.5 10.3 10.1  9.9  9.7  9.5  9.3  9.1  8.9  8.7  8.5  8.3  8.1  7.9
## [15]  7.7  7.5  7.3  7.1  6.9  6.7  6.5  6.3  6.1  5.9  5.7  5.5  5.3  5.1
## [29]  4.9  4.7  4.5  4.3  4.1  3.9  3.7  3.5  3.3  3.1  2.9  2.7
```

- Vektör elemanlarına aşağıdaki gibi erişilebilir

```r
t <- c("Pzt","Sal","Çar","Per","Cum","Cmt","Paz")

# 2. 3. ve 6. elemanları al
u <- t[c(2,3,6)]

print(u)
```

```
## [1] "Sal" "Çar" "Cmt"
```

```r
# Değeri True olan elemanları al
v <- t[c(TRUE,FALSE,FALSE,FALSE,FALSE,TRUE,FALSE)]

print(v)
```

```
## [1] "Pzt" "Cmt"
```

```r
# 2. ve 5. elemanları çıkart
x <- t[c(-2,-5)]

print(x)
```

```
## [1] "Pzt" "Çar" "Per" "Cmt" "Paz"
```

- Vektör elemanları aşağıdaki şekilde değiştirilir

```
t[1] <- "Pazartesi"

print(t)
```

```
## [1] "Pazartesi" "Sal"       "Çar"       "Per"       "Cum"       "Cmt"
## [7] "Paz"
```

```
t[c(2,3,5)] <- c("Salı", "Çarşamba", "Cuma")

print(t)
```

```
## [1] "Pazartesi" "Salı"      "Çarşamba"  "Per"       "Cuma"      "Cmt"
## [7] "Paz"
```

```
t[c(FALSE, FALSE, FALSE, TRUE, FALSE, TRUE, TRUE)] =
  c("Perşembe", "Cumartesi", "Pazar")

print(t)
```

```
## [1] "Pazartesi" "Salı"      "Çarşamba"  "Perşembe"  "Cuma"      "Cumartesi"
## [7] "Pazar"
```

- Vektör Aritmetiği

```
v1 <- c(3,8,4,5,0,11)
v2 <- c(4,11,0,8,1,2)

ekle.sonuc <- v1+v2

print(ekle.sonuc)
```

```
## [1]  7 19  4 13  1 13
```

```
fark.sonuc <- v1-v2
print(fark.sonuc)
```

```
## [1] -1 -3  4 -3 -1  9
```

```
carp.sonuc <- v1*v2
print(carp.sonuc)
```

```
## [1] 12 88  0 40  0 22
```

```
bol.sonuc <- v1/v2
print(bol.sonuc)
```

```
## [1] 0.7500000 0.7272727       Inf 0.6250000 0.0000000 5.5000000
```

- R Vektörleri Aritmetik işlemlere girerken, eğer bir vektörün eleman sayısı ötekine eşit değilse, R eşitsizliği eleman tekrar ederek tamamlar.

```
v2 <- c(4,11)

ekle.sonuc <- v1+v2
print(ekle.sonuc)
```

```
## [1]  7 19  8 16  4 22
```

```
fark.sonuc <- v1-v2
print(fark.sonuc)
```

```
## [1] -1 -3  0 -6 -4  0
```

```
carp.sonuc <- v1*v2
print(carp.sonuc)
```

```
## [1]  12  88  16  55   0 121
```

```
bol.sonuc <- v1/v2
print(bol.sonuc)
```

```
## [1] 0.7500000 0.7272727 1.0000000 0.4545455 0.0000000 1.0000000
```

- Elemanların sıralanması

```
v <- c(3,8,4,5,0,11, -9, 304)

sirala.sonuc <- sort(v)

print(sirala.sonuc)
```

```
## [1]  -9   0   3   4   5   8  11 304
```

```r
azalan.sirala.sonuc <- sort(v, decreasing = TRUE)

print(azalan.sirala.sonuc)
```

```
## [1] 304  11   8   5   4   3   0  -9
```

```r
t <- c("Pzt","Sal","Çar","Per","Cum","Cmt","Paz")

sirala.sonuc <- sort(t)

print(sirala.sonuc)
```

```
## [1] "Cmt" "Cum" "Çar" "Paz" "Per" "Pzt" "Sal"
```

```r
azalan.sirala.sonuc <- sort(t, decreasing = TRUE)
print(azalan.sirala.sonuc)
```

```
## [1] "Sal" "Pzt" "Per" "Paz" "Çar" "Cum" "Cmt"
```

#### 2.2.2.2 Listeler

- Vektörler gibi birden çok değer saklayabilirler
- Değerler aynı türde olmak zorunda değildir
- *list()* komutu ile oluşturulurlar
- Sakladıkları her eleman isimlendirilebilir

```r
list_data <- list(c("Oca","Şub","Mar"),
                  matrix(c(3,9,5,1,-2,8), nrow=2),
                  list("green",12.3))

names(list_data) <- c("İlk Çeyrek", "Bir Matris", "Bir İç Liste")

print(list_data)
```

```
## $`İlk Çeyrek`
## [1] "Oca" "Şub" "Mar"
##
## $`Bir Matris`
##      [,1] [,2] [,3]
## [1,]    3    5   -2
## [2,]    9    1    8
##
## $`Bir İç Liste`
```

```
## $`Bir İç Liste`[[1]]
## [1] "green"
##
## $`Bir İç Liste`[[2]]
## [1] 12.3
```

- Liste Elemanlarına Erişim Yöntemleri

```
print(list_data[1])
```

```
## $`İlk Çeyrek`
## [1] "Oca" "Şub" "Mar"
```

```
print(list_data$`Bir Matris`)
```

```
##      [,1] [,2] [,3]
## [1,]    3    5   -2
## [2,]    9    1    8
```

- Eleman değerlerinin değiştirilmesi

```
list_data[4] <- "Yeni Eleman"
print(list_data[4])
```

```
## [[1]]
## [1] "Yeni Eleman"
```

```
list_data[4] <- NULL
print(list_data[4])
```

```
## $<NA>
## NULL
```

```
list_data[3] <- "Artık Bir Metin"
print(list_data[3])
```

```
## $`Bir İç Liste`
## [1] "Artık Bir Metin"
```

- Listelerin birleştirilmesi

```
list1 <- list(1,2,3)
list2 <- list("Pzt","Sal","Çar")

bir.liste <- c(list1, list2)

print(bir.liste)
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] 3
##
## [[4]]
## [1] "Pzt"
##
## [[5]]
## [1] "Sal"
##
## [[6]]
## [1] "Çar"
```

- Listeleri vektöre dönüştürme

```
list1 <- list(1:5)
print(list1)
```

```
## [[1]]
## [1] 1 2 3 4 5
```

```
list2 <-list(10:14)
print(list2)
```

```
## [[1]]
## [1] 10 11 12 13 14
```

```
v1 <- unlist(list1)
v2 <- unlist(list2)
print(v1)
```

```
## [1] 1 2 3 4 5
```

```r
print(v2)
```

```
## [1] 10 11 12 13 14
```

#### 2.2.2.3 Matrisler

- Matris iki boyutlu dikdörtgensel veri kümeleridir
- Aynı türde veri saklarlar
- Bir vektörün satır ve sütunlara ayrılması ile oluşturulabilirler
- *matrix()* fonksiyonu ile oluşturulurlar
- Matrisler dikdörtgensel veri saklamaya yarar

```r
M <- matrix(c(3:14), nrow=4, byrow=TRUE)
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    3    4    5
## [2,]    6    7    8
## [3,]    9   10   11
## [4,]   12   13   14
```

```r
N <- matrix(c(3:14), nrow=4, byrow=FALSE)
print(N)
```

```
##      [,1] [,2] [,3]
## [1,]    3    7   11
## [2,]    4    8   12
## [3,]    5    9   13
## [4,]    6   10   14
```

- Satır ve sütunlarına isim verilebilir

```r
isim.satır = c("sat1", "sat2", "sat3", "sat4")
isim.sütun = c("süt1", "süt2", "süt3")

P <- matrix(c(3:14), nrow=4, byrow=TRUE,
            dimnames=list(isim.satır, isim.sütun))

print(P)
```

```
##      süt1 süt2 süt3
## sat1    3    4    5
## sat2    6    7    8
## sat3    9   10   11
## sat4   12   13   14
```

- Matrislere Erişim

```r
print(P[1,3])
```

```
## [1] 5
```

```r
P[1,3] = 255
print(P[1,3])
```

```
## [1] 255
```

```r
print(P[2,])
```

```
## süt1 süt2 süt3
##    6    7    8
```

```r
P[2,] = c(1,2,3)
print(P[2,])
```

```
## süt1 süt2 süt3
##    1    2    3
```

```r
print(P[,2])
```

```
## sat1 sat2 sat3 sat4
##    4    2   10   13
```

```r
P[,2] = c(1,2,3,4)
print(P[,2])
```

```
## sat1 sat2 sat3 sat4
##    1    2    3    4
```

#### 2.2.2.4 Diziler

- Çok boyutlu veri kümeleridir
    - Matrisler dizilerin 2 boyutlu halidir
- Aynı türde veri saklarlar
- Bir vektörden elde edilebilirler
- array() komutu ile oluşturulurlar
- N-Boyutlu veriler için oluşturulmuşlardır
- Boyutlara isim verilebilir

```r
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)

result <- array(c(vector1,vector2),dim=c(3,3,2))
print(result)
```

```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]    5   10   13
## [2,]    9   11   14
## [3,]    3   12   15
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]    5   10   13
## [2,]    9   11   14
## [3,]    3   12   15
```

```r
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)

isim.sütun <- c("SÜT1","SÜT2","SÜT3")
isim.satır <- c("SAT1","SAT2","SAT3")
isim.matris <- c("Matris1","Matris2")


result <- array(c(vector1, vector2), dim=c(3,3,2),
                dimnames = list(isim.sütun,
                                isim.satır,
                                isim.matris))
print(result)
```

```
## , , Matris1
##
##      SAT1 SAT2 SAT3
## SÜT1    5   10   13
## SÜT2    9   11   14
## SÜT3    3   12   15
##
## , , Matris2
##
##      SAT1 SAT2 SAT3
## SÜT1    5   10   13
## SÜT2    9   11   14
## SÜT3    3   12   15
```

- Fonksiyonlar dizilerin istenilen boyutuna uygulanabilir

```
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)

yeni.dizi <- array(c(vector1,vector2),dim=c(3,4,2))

print(yeni.dizi)
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    5   10   13    5
## [2,]    9   11   14    9
## [3,]    3   12   15    3
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]   10   13    5   10
## [2,]   11   14    9   11
## [3,]   12   15    3   12
```

```
# Bütün Satırların Toplamının Hesaplanması
result <- apply(yeni.dizi, c(1), sum)
print(result)
```

```
## [1] 71 88 75
```

```
# Bütün Sütunların Toplamının Hesaplanması
result <- apply(yeni.dizi, c(2), sum)
print(result)
```

```
## [1] 50 75 59 50
```

```
# Bütün Matrislerin Toplamının Hesaplanması
result <- apply(yeni.dizi, c(3), sum)
print(result)
```

```
## [1] 109 125
```

#### 2.2.2.5 Faktörler

- Bir vektör kullanılarak oluşturulurlar
- Bir vektördeki;
  - Tekrar etmeyen elemanları etiket olarak,
  - Vektörün elemanlarını tamsayı olarak saklarlar
- *factor()* fonksiyonu ile oluşturulurlar
- *nlevels()* fonksiyonu ile tekrar etmeyen elemanların sayısı elde edilebilir
- *levels()* fonksiyonu ile tekrar etmeyen elemanlar elde edilebilir
- Nominal Veri Saklarlar
- Bir veri çerçevesine koyduğunuz metinler faktöre çevirilir.

```
boy <- c(132,151,162,139,166,147,122)
kilo <- c(48,49,66,53,67,52,40)
cinsiyet <- c("erkek","erkek","kadın","kadın","erkek","kadın","erkek")

giriş.verisi <- data.frame(boy,kilo,cinsiyet)
print(giriş.verisi)
```

```
##   boy kilo cinsiyet
## 1 132   48    erkek
## 2 151   49    erkek
## 3 162   66    kadın
## 4 139   53    kadın
## 5 166   67    erkek
## 6 147   52    kadın
## 7 122   40    erkek
```

```
print(is.factor(giriş.verisi$cinsiyet))
```

```
## [1] TRUE
```

```
print(giriş.verisi$cinsiyet)
```

```
## [1] erkek erkek kadın kadın erkek kadın erkek
## Levels: erkek kadın
```

- Faktörlerin sırası faktör fonksiyonu ile değiştirilebilir

```
veri <- c("Doğu","Batı","Doğu","Kuzey","Kuzey",
          "Doğu","Batı","Batı","Batı","Doğu","Kuzey")

factor_veri <- factor(veri)
print(factor_veri)
```

```
##  [1] Doğu  Batı  Doğu  Kuzey Kuzey Doğu  Batı  Batı  Batı  Doğu  Kuzey
## Levels: Batı Doğu Kuzey
```

```
yeni.faktör.veri <- factor(factor_veri,
                           levels = c("Kuzey","Doğu","Batı"))
print(yeni.faktör.veri)
```

```
##  [1] Doğu  Batı  Doğu  Kuzey Kuzey Doğu  Batı  Batı  Batı  Doğu  Kuzey
## Levels: Kuzey Doğu Batı
```

- Yeni faktörler *gl()* fonksiyonu ile üretilebilir.

```
v <- gl(3, 4, labels = c("Kırmızı", "Beyaz","Yeşil"))
print(v)
```

```
##  [1] Kırmızı Kırmızı Kırmızı Kırmızı Beyaz   Beyaz   Beyaz   Beyaz
##  [9] Yeşil   Yeşil   Yeşil   Yeşil
## Levels: Kırmızı Beyaz Yeşil
```

#### 2.2.2.6 Veri Çerçeveleri

- Tablo şeklindeki veri nesneleridir
- Matrislere benzerler
  - Fakat sütunlar farklı türlerde olabilir
  - Örneğin;
    - İlk Sütun Karakter
    - İkinci Sütun Sayısal
  - Olabilir
- *data.frame()* fonksiyonu ile oluşturulurlar

- Birden fazla vektörü bir arada tutan yapılardır.
- Sütunların ismi olabilir

```
personel.verisi <- data.frame(
  personel.no = c (1:5),
  personel.adi = c("Tolga","Mustafa","Uğur","Erdinç","Yavuz"),
  personel.ucreti = c(623.3,515.2,611.0,729.0,843.25),
  personel.başlama_tarihi = as.Date(c("2012-01-01","2013-09-23",
                                      "2014-11-15","2014-05-11",
                                      "2015-03-27")),
  stringsAsFactors=FALSE
)

print(personel.verisi )
```

```
##   personel.no personel.adi personel.ucreti personel.başlama_tarihi
## 1           1        Tolga          623.30              2012-01-01
## 2           2      Mustafa          515.20              2013-09-23
## 3           3         Uğur          611.00              2014-11-15
## 4           4       Erdinç          729.00              2014-05-11
## 5           5        Yavuz          843.25              2015-03-27
```

- Bir veri çerçevesinin yapısı hakkında *str()* fonksiyonu ile bilgi alabilirsiniz.

```
str(personel.verisi)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ personel.no            : int  1 2 3 4 5
##  $ personel.adi           : chr  "Tolga" "Mustafa" "Uğur" "Erdinç" ...
##  $ personel.ucreti        : num  623 515 611 729 843
##  $ personel.başlama_tarihi: Date, format: "2012-01-01" "2013-09-23" ...
```

- Bir veri çerçevsi hakkında özet bilgileri summary fonksiyonu ile alabilirsiniz.

```
print(summary(personel.verisi))
```

```
##   personel.no personel.adi       personel.ucreti personel.başlama_tarihi
##  Min.   :1    Length:5           Min.   :515.2   Min.   :2012-01-01
##  1st Qu.:2    Class :character   1st Qu.:611.0   1st Qu.:2013-09-23
##  Median :3    Mode  :character   Median :623.3   Median :2014-05-11
##  Mean   :3                       Mean   :664.4   Mean   :2014-01-14
##  3rd Qu.:4                       3rd Qu.:729.0   3rd Qu.:2014-11-15
##  Max.   :5                       Max.   :843.2   Max.   :2015-03-27
```

- Bir çerçeveden istediğiniz sütunları ve satırları çıkarabilirsiniz

```
yeni <- data.frame(personel.verisi$personel.adi,
                   personel.verisi$personel.ucreti)
print(yeni)
```

```
##   personel.verisi.personel.adi personel.verisi.personel.ucreti
## 1                        Tolga                          623.30
## 2                      Mustafa                          515.20
## 3                         Uğur                          611.00
## 4                       Erdinç                          729.00
## 5                        Yavuz                          843.25
```

```
yeni <- personel.verisi[1:2,]
print(yeni)
```

```
##   personel.no personel.adi personel.ucreti personel.başlama_tarihi
## 1           1        Tolga           623.3              2012-01-01
## 2           2      Mustafa           515.2              2013-09-23
```

```
yeni <- personel.verisi[c(3,5),c(2,4)]
print(yeni)
```

```
##   personel.adi personel.başlama_tarihi
## 3         Uğur              2014-11-15
## 5        Yavuz              2015-03-27
```

- Veri çerçevelerine yeni sütunlar ekleyebilirsiniz

```
personel.verisi$bölüm <- c("BT","Satış","BT","İK","Finans")
print(personel.verisi)
```

```
##   personel.no personel.adi personel.ucreti personel.başlama_tarihi  bölüm
## 1           1        Tolga          623.30              2012-01-01     BT
## 2           2      Mustafa          515.20              2013-09-23  Satış
## 3           3         Uğur          611.00              2014-11-15     BT
## 4           4       Erdinç          729.00              2014-05-11     İK
## 5           5        Yavuz          843.25              2015-03-27 Finans
```

- Veri Çerçevelerine yeni satırlar ekleyebilirsiniz

```
yeni.personel <- data.frame(
  personel.no = c (6:8),
  personel.adi = c("Mehmet Ali","Buğra","Ali Hikmet"),
  personel.ucreti = c(623.3,515.2,611.0),
  personel.başlama_tarihi = as.Date(c("2012-01-01","2013-09-23",
                                     "2014-11-15")),
  bölüm = c("IT","Satış","Finans"),
  stringsAsFactors=FALSE
)

personel.birleştirilmiş <- rbind(personel.verisi, yeni.personel)

print(personel.birleştirilmiş)
```

```
##   personel.no personel.adi personel.ucreti personel.başlama_tarihi  bölüm
## 1           1        Tolga          623.30              2012-01-01     BT
## 2           2      Mustafa          515.20              2013-09-23  Satış
## 3           3         Uğur          611.00              2014-11-15     BT
## 4           4       Erdinç          729.00              2014-05-11     İK
## 5           5        Yavuz          843.25              2015-03-27 Finans
## 6           6   Mehmet Ali          623.30              2012-01-01     IT
## 7           7        Buğra          515.20              2013-09-23  Satış
## 8           8   Ali Hikmet          611.00              2014-11-15 Finans
```

#### 2.2.2.7 Uygulama

```
# Vektör Oluştur
elma <- c('Kırmızı', 'Yeşil', "Mavi")
print(elma)
```

```
## [1] "Kırmızı" "Yeşil"   "Mavi"
```

```
# Vektörün Elemanlarının Türü
print(class(elma))
```

```
## [1] "character"
```

```
# Liste oluştur
liste1 <- list(c(2,5,3), 21.3, sin)

# Liste yazdır
print(liste1)
```

```
## [[1]]
## [1] 2 5 3
##
## [[2]]
## [1] 21.3
##
## [[3]]
## function (x)  .Primitive("sin")
```

```r
# Matris oluştur Satırlara göre
M <- matrix(c('a','a','b','c','b',"a"), nrow = 2, ncol = 3, byrow = TRUE)

print(M)
```

```
##      [,1] [,2] [,3]
## [1,] "a"  "a"  "b"
## [2,] "c"  "b"  "a"
```

```r
# Matris oluştur Sütunlara göre
M <- matrix(c('a','a','b','c','b',"a"), nrow = 2, ncol = 3)

print(M)
```

```
##      [,1] [,2] [,3]
## [1,] "a"  "b"  "b"
## [2,] "a"  "c"  "a"
```

```r
# Dizi oluştur
a <- array(c('yeşil', 'sarı'), dim = c(3,3,2))

print(a)
```

```
## , , 1
##
##      [,1]    [,2]    [,3]
## [1,] "yeşil" "sarı"  "yeşil"
## [2,] "sarı"  "yeşil" "sarı"
## [3,] "yeşil" "sarı"  "yeşil"
##
## , , 2
##
##      [,1]    [,2]    [,3]
## [1,] "sarı"  "yeşil" "sarı"
## [2,] "yeşil" "sarı"  "yeşil"
## [3,] "sarı"  "yeşil" "sarı"
```

```
# Bir vektör oluştur

elma_renkleri <- c('yeşil','yeşil','sarı','kırmızı','kırmızı','kırmızı', 'yeşil')

# Faktör nesnesi oluştur

elma_faktorleri <- factor(elma_renkleri)

print(elma_faktorleri)
```

```
## [1] yeşil   yeşil   sarı    kırmızı kırmızı kırmızı yeşil
## Levels: kırmızı sarı yeşil
```

```
print(nlevels(elma_faktorleri))
```

```
## [1] 3
```

```
print(levels(elma_faktorleri))
```

```
## [1] "kırmızı" "sarı"    "yeşil"
```

```
# Veri Çerçevesi oluştur

bmi <- data.frame(
  cinsiyet <- c("Erkek","Erkek","Kadın"),
  boy      <- c(152, 171.5, 165),
  kilo     <- c(81, 93, 78),
  yaş      <- c(42, 38, 28)
)

print(bmi)
```

```
##   cinsiyet....c..Erkek....Erkek....Kadın.. boy....c.152..171.5..165.
## 1                                    Erkek                     152.0
## 2                                    Erkek                     171.5
## 3                                    Kadın                     165.0
##   kilo....c.81..93..78. yaş....c.42..38..28.
## 1                    81                   42
## 2                    93                   38
## 3                    78                   28
```

### 2.2.3 R Değişkenleri

- R programlama dili, sizin kullanmanız için isimlendirilmiş bir depolama mekanizması sunar
- Bu depolama mekanizmasında her bir eleman bir değişken olarak adlandırılır
- Değişkenlerin isimleri
  - Harfler, Sayılar, Nokta ve Alt Çizgiden Oluşabilir
  - Sayı ve alt çizgi ile başlayamazlar
  - Noktadan sonra sayı gelemez
- Bir değişkenin içeriği print() veya cat() fonksiyonu ile yazdırılabilir.
- cat() fonksiyonu birden fazla değişkeni ekrana yazdırabilir.

```
# Eşittir ile atama
var.1 = c(0,1,2,3)

# Sol operatörü ile atama
var.2 <- c("program","R")

# Sağ operatörü ile atama
c(TRUE,1) -> var.3

print(var.1)
```

```
## [1] 0 1 2 3
```

```
cat ("var.1 değeri, ", var.1 ," dir.\n")
```

```
## var.1 değeri,  0 1 2 3  dir.
```

```
cat ("var.2 değeri, ", var.2 ," dir.\n")
```

```
## var.2 değeri,  program R  dir.
```

```
cat ("var.3 değeri, ", var.3 ," dir.\n")
```

```
## var.3 değeri,  1 1  dir.
```

- Değişkenlerin türleri yoktur
- En son atanan değer değişken türünü belirler
- Değişken türü *class()* fonksiyonu ile öğrenilebilir

```
var_x <- "Merhaba"
cat("var_x'in türü ",class(var_x)," dir\n")
```

```
## var_x'in türü  character  dir
```

```
var_x <- 34.5
cat("var_x'in türü şimdi ",class(var_x)," dir\n")
```

```
## var_x'in türü şimdi  numeric  dir
```

```
var_x <- 27L
cat("var_x'in türü şimdi ",class(var_x)," oldu\n")
```

```
## var_x'in türü şimdi  integer  oldu
```

- Çalışma ortamınızdaki tüm değişkenleri öğrenmek için *ls()* fonksiyonu kullanılır

```
ls()
```

```
##  [1] "a"              "bmi"            "boy"
##  [4] "cinsiyet"       "elma"           "elma_faktorleri"
##  [7] "elma_renkleri"  "kilo"           "liste1"
## [10] "M"              "v"              "var.1"
## [13] "var.2"          "var.3"          "var_x"
## [16] "yaş"
```

- Çalışma ortamınızdaki değişkenleri silmek için *rm()* fonksiyonu kullanılır

```
rm(var_x)
```

### 2.2.4 R Operatörleri

- R programlama dilinde 5 farklı operatör grubu bulunmaktadır
  - Aritmetik Operatörler
  - İlişkisel Operatörler
  - Mantıksal Operatörler
  - Atama Operatörleri
  - Diğer Operatörler (Miscellaneous, Kategorize edilemeyen ?)
- Operatörler aksi belirtilmediği sürece eleman bazlı işlem yaparlar.

#### 2.2.4.1 Aritmetik İşlemler

- Aritmetik işlemleri gerçekleştirmek için kullanılırlar
  - +, -, *, /
  - %% Mod alma operatörü
  - %/% Bölüm bulma operatörü
  - ˆ Üs alma oprreratörü

```
v <- c( 2, 5.5, 6)
t <- c( 8, 3,   4)

cat("v=",v,"\n")
```

```
## v= 2 5.5 6
```

```
cat("t=",t,"\n")
```

```
## t= 8 3 4
```

```
cat("v+t=",v+t  ,"\n")
```

```
## v+t= 10 8.5 10
```

```
cat("v-t=",v-t  ,"\n")
```

```
## v-t= -6 2.5 2
```

```
cat("v*t=",v*t  ,"\n")
```

```
## v*t= 16 16.5 24
```

```
cat("v/t=",v/t  ,"\n")
```

```
## v/t= 0.25 1.833333 1.5
```

```
cat("v%%t=",v%%t ,"\n")
```

```
## v%%t= 2 2.5 2
```

```
cat("v%/%t=",v%/%t,"\n")
```

```
## v%/%t= 0 1 1
```

```
cat("v^t=",v^t  ,"\n")
```

```
## v^t= 256 166.375 1296
```

#### 2.2.4.2 İlişkisel Operatörler

- İki değişkenin arasındaki ilişkiyi belirlemek için kullanırlar

```
v <- c(2,5.5,6,9)
t <- c(8,2.5,14,9)

cat("v=",v,"\n")
```

```
## v= 2 5.5 6 9
```

```
cat("t=",t,"\n")
```

```
## t= 8 2.5 14 9
```

```
cat("v>t  =",v>t,"\n")
```

```
## v>t  = FALSE TRUE FALSE FALSE
```

```
cat("v<t  =",v<t,"\n")
```

```
## v<t  = TRUE FALSE TRUE FALSE
```

```
cat("v==t =",v==t,"\n")
```

```
## v==t = FALSE FALSE FALSE TRUE
```

```
cat("v<=t =",v<=t,"\n")
```

```
## v<=t = TRUE FALSE TRUE TRUE
```

```
cat("v>=t =",v>=t,"\n")
```

```
## v>=t = FALSE TRUE FALSE TRUE
```

```
cat("v!=t =",v!=t,"\n")
```

```
## v!=t = TRUE TRUE TRUE FALSE
```

#### 2.2.4.3 Mantıksal Operatörler

- Sadece Mantıksal (TRUE/FALSE) değişkenlere uygulanabilir
- R programlama dilinde, 0 ve dengi değerler FALSE, diğer değerler TRUE kabul edilir

```
v <- c(3,0,TRUE,2+3i)
t <- c(4,0,FALSE,2+3i)

cat("v=",v,"\n")
```

```
## v= 3+0i 0+0i 1+0i 2+3i
```

```
cat("t=",t,"\n")
```

```
## t= 4+0i 0+0i 0+0i 2+3i
```

```
# Eleman Bazlı Karşılaştırma
cat("v&t = ",v&t,"\n")
```

```
## v&t =  TRUE FALSE FALSE TRUE
```

```
cat("v|t = ",v|t,"\n")
```

```
## v|t =  TRUE FALSE TRUE TRUE
```

```
cat("!v  = ",!v ,"\n")
```

```
## !v  =  FALSE TRUE FALSE FALSE
```

```
# Sadece ilk elemanı karşılaştırma
cat("v&&t =", v&&t, "\n")
```

```
## v&&t = TRUE
```

```
cat("v||t =", v||t, "\n")
```

```
## v||t = TRUE
```

#### 2.2.4.4 Atama Operatörleri

```
v1 <- c(3,1,TRUE,2+3i)
v2 <<- c(3,1,TRUE,2+3i)
v3 = c(3,1,TRUE,2+3i)

cat("v1 = ",v1,"\n")
```

```
## v1 =  3+0i 1+0i 1+0i 2+3i
```

```
cat("v2 = ",v2,"\n")
```

```
## v2 =  3+0i 1+0i 1+0i 2+3i
```

```
cat("v3 = ",v3,"\n")
```

```
## v3 =  3+0i 1+0i 1+0i 2+3i
```

```
c(3,1,TRUE,2+3i) -> v1
c(3,1,TRUE,2+3i) ->> v2

cat("v1 = ",v1,"\n")
```

```
## v1 =  3+0i 1+0i 1+0i 2+3i
```

```
cat("v2 = ",v2,"\n")
```

```
## v2 =  3+0i 1+0i 1+0i 2+3i
```

#### 2.2.4.5 Diğer Operatörler

- : Operatörü
  - Sıra sayıları üretmek için kullanılır, seq() fonksiyonunun kısaltılmış halidir.
- %in% Operatörü
  - Bir elemanın vektörde olup olmadığını belirlemek için kullanılır
- %*% Operatörü
  - Matris çarpma için kullanılır

```
v <- 2:8
cat("v = ",v,"\n")
```

```
## v =  2 3 4 5 6 7 8
```

```
v1 <- 8
v2 <- 12
t <- 1:10
cat(v1," vektörde var mı ? ", v1 %in% t,"\n")
```

```
## 8  vektörde var mı ?  TRUE
```

```
cat(v2," vektörde var mı ? ", v2 %in% t,"\n")
```

```
## 12  vektörde var mı ?  FALSE
```

```
M = matrix( c(2,6,5,1,10,4), nrow=2,ncol=3,byrow = TRUE)
N = matrix( c(1,2,3,4,5,6),  nrow=3,ncol=2,byrow = TRUE)
t = M %*% N
print("Matris Çarpımı:")
```

```
## [1] "Matris Çarpımı:"
```

```
print(t)
```

```
##      [,1] [,2]
## [1,]   45   58
## [2,]   51   66
```

Bölüm

3

# Uygulama: R ile Veri Kümesi İşlemleri

Aşağıdaki komutları çalıştırınız.

```r
# Verinin Yüklenmesi
veri <- read.csv2("imports-85.csv")

str(veri)
```

```
## 'data.frame':    159 obs. of  26 variables:
##  $ SiraNo          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ RiskDerecesi    : int  2 2 1 1 2 0 0 0 2 1 ...
##  $ KayipDegeri     : int  164 164 158 158 192 192 188 188 121 98 ...
##  $ Uretici         : Factor w/ 18 levels "audi","bmw","chevrolet",..: 1 1 1 1 2 2 2 2
##  $ YakitTuru       : Factor w/ 2 levels "diesel","gas": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Enjeksiyon      : Factor w/ 2 levels "std","turbo": 1 1 1 2 1 1 1 1 1 1 ...
##  $ KapiSayisi      : Factor w/ 2 levels "four","two": 1 1 1 1 2 1 2 1 2 2 ...
##  $ GovdeTuru       : Factor w/ 5 levels "convertible",..: 4 4 4 4 4 4 4 4 3 3 ...
##  $ Cekis           : Factor w/ 3 levels "4wd","fwd","rwd": 2 1 2 2 3 3 3 3 2 2 ...
##  $ DingilAcikligi  : num  99.8 99.4 105.8 105.8 101.2 ...
##  $ Uzunluk         : num  177 177 193 193 177 ...
##  $ Genislik        : num  66.2 66.4 71.4 71.4 64.8 64.8 64.8 64.8 60.3 63.6 ...
##  $ Yukseklik       : num  54.3 54.3 55.7 55.9 54.3 54.3 54.3 54.3 53.2 52 ...
##  $ AzamiAgirlik    : int  2337 2824 2844 3086 2395 2395 2710 2765 1488 1874 ...
##  $ MotorTuru       : Factor w/ 5 levels "dohc","l","ohc",..: 3 3 3 3 3 3 3 3 2 3 ...
##  $ SilindirSayisi  : Factor w/ 5 levels "eight","five",..: 3 2 2 2 3 3 4 4 5 3 ...
##  $ MotorHacmi      : int  109 136 136 131 108 108 164 164 61 90 ...
##  $ YakitSistemi    : Factor w/ 6 levels "1bbl","2bbl",..: 5 5 5 5 5 5 5 5 2 2 ...
##  $ SilindirCapi    : num  3.19 3.19 3.19 3.13 3.5 3.5 3.31 3.31 2.91 3.03 ...
##  $ Zamanlama       : num  3.4 3.4 3.4 3.4 2.8 2.8 3.19 3.19 3.03 3.11 ...
##  $ SikistirmaOrani : num  10 8 8.5 8.3 8.8 8.8 9 9 9.5 9.6 ...
```

```
##  $ BeygirGucu      : int  102 115 110 140 101 101 121 121 48 70 ...
##  $ MaximumDevir    : int  5500 5500 5500 5500 5800 5800 4250 4250 5100 5400 ...
##  $ SehirIciHarcama : int  24 18 19 17 23 23 21 21 47 38 ...
##  $ SehirDisiHarcama: int  30 22 25 20 29 29 28 28 53 43 ...
##  $ Fiyat           : int  13950 17450 17710 23875 16430 16925 20970 21105 5151 6295 .
```

```
print(summary(veri))
```

```
##      SiraNo       RiskDerecesi      KayipDegeri        Uretici
##  Min.   :  1.0   Min.   :-2.0000   Min.   : 65.0   toyota :31
##  1st Qu.: 40.5   1st Qu.: 0.0000   1st Qu.: 94.0   nissan :18
##  Median : 80.0   Median : 1.0000   Median :113.0   honda  :13
##  Mean   : 80.0   Mean   : 0.7358   Mean   :121.1   subaru :12
##  3rd Qu.:119.5   3rd Qu.: 2.0000   3rd Qu.:148.0   mazda  :11
##  Max.   :159.0   Max.   : 3.0000   Max.   :256.0   volvo  :11
##                                                    (Other):63
##   YakitTuru   Enjeksiyon  KapiSayisi       GovdeTuru  Cekis
##  diesel: 15   std  :132   four:95    convertible: 2   4wd:  8
##  gas   :144   turbo: 27   two :64    hardtop    : 5   fwd:105
##                                      hatchback  :56   rwd: 46
##                                      sedan      :79
##                                      wagon      :17
##
##
##  DingilAcikligi      Uzunluk         Genislik       Yukseklik
##  Min.   : 86.60   Min.   :141.1   Min.   :60.30   Min.   :49.40
##  1st Qu.: 94.50   1st Qu.:165.7   1st Qu.:64.00   1st Qu.:52.25
##  Median : 96.90   Median :172.4   Median :65.40   Median :54.10
##  Mean   : 98.26   Mean   :172.4   Mean   :65.61   Mean   :53.90
##  3rd Qu.:100.80   3rd Qu.:177.8   3rd Qu.:66.50   3rd Qu.:55.50
##  Max.   :115.60   Max.   :202.6   Max.   :71.70   Max.   :59.80
##
##   AzamiAgirlik  MotorTuru  SilindirSayisi   MotorHacmi    YakitSistemi
##  Min.   :1488   dohc:  8   eight:  1      Min.   : 61.0   1bbl:11
##  1st Qu.:2066   l   :  8   five :  7      1st Qu.: 97.0   2bbl:63
##  Median :2340   ohc :123   four :136      Median :110.0   idi :15
##  Mean   :2461   ohcf: 12   six  : 14      Mean   :119.2   mfi : 1
##  3rd Qu.:2810   ohcv:  8   three:  1      3rd Qu.:135.0   mpfi:64
##  Max.   :4066                             Max.   :258.0   spdi: 5
##
##   SilindirCapi    Zamanlama     SikistirmaOrani   BeygirGucu
##  Min.   :2.54   Min.   :2.070   Min.   : 7.00   Min.   : 48.00
##  1st Qu.:3.05   1st Qu.:3.105   1st Qu.: 8.70   1st Qu.: 69.00
##  Median :3.27   Median :3.270   Median : 9.00   Median : 88.00
```

```
##  Mean   :3.30   Mean   :3.236   Mean   :10.16   Mean   : 95.84
##  3rd Qu.:3.56   3rd Qu.:3.410   3rd Qu.: 9.40   3rd Qu.:114.00
##  Max.   :3.94   Max.   :4.170   Max.   :23.00   Max.   :200.00
##
##   MaximumDevir  SehirIciHarcama SehirDisiHarcama     Fiyat
##  Min.   :4150   Min.   :15.00   Min.   :18.00    Min.   : 5118
##  1st Qu.:4800   1st Qu.:23.00   1st Qu.:28.00    1st Qu.: 7372
##  Median :5200   Median :26.00   Median :32.00    Median : 9233
##  Mean   :5114   Mean   :26.52   Mean   :32.08    Mean   :11446
##  3rd Qu.:5500   3rd Qu.:31.00   3rd Qu.:37.00    3rd Qu.:14720
##  Max.   :6600   Max.   :49.00   Max.   :54.00    Max.   :35056
##
```

```
# Toyota marka araçların bulunması

toyotalar <- veri$Uretici=="toyota"

toyota.veri <- veri[toyotalar,]

print(toyota.veri)
```

```
##     SiraNo RiskDerecesi KayipDegeri Uretici YakitTuru Enjeksiyon
## 110    110            1          87  toyota       gas        std
## 111    111            1          87  toyota       gas        std
## 112    112            1          74  toyota       gas        std
## 113    113            0          77  toyota       gas        std
## 114    114            0          81  toyota       gas        std
## 115    115            0          91  toyota       gas        std
## 116    116            0          91  toyota       gas        std
## 117    117            0          91  toyota       gas        std
## 118    118            0          91  toyota    diesel        std
## 119    119            0          91  toyota    diesel        std
## 120    120            0          91  toyota       gas        std
## 121    121            0          91  toyota       gas        std
## 122    122            0          91  toyota       gas        std
## 123    123            1         168  toyota       gas        std
## 124    124            1         168  toyota       gas        std
## 125    125            1         168  toyota       gas        std
## 126    126            1         168  toyota       gas        std
## 127    127            2         134  toyota       gas        std
## 128    128            2         134  toyota       gas        std
## 129    129            2         134  toyota       gas        std
## 130    130            2         134  toyota       gas        std
## 131    131            2         134  toyota       gas        std
```

```
## 132    132           2          134  toyota       gas        std
## 133    133          -1           65  toyota       gas        std
## 134    134          -1           65  toyota    diesel      turbo
## 135    135          -1           65  toyota       gas        std
## 136    136          -1           65  toyota       gas        std
## 137    137          -1           65  toyota       gas        std
## 138    138           3          197  toyota       gas        std
## 139    139           3          197  toyota       gas        std
## 140    140          -1           90  toyota       gas        std
##     KapiSayisi   GovdeTuru Cekis DingilAcikligi Uzunluk Genislik Yukseklik
## 110        two   hatchback   fwd           95.7   158.7     63.6      54.5
## 111        two   hatchback   fwd           95.7   158.7     63.6      54.5
## 112       four   hatchback   fwd           95.7   158.7     63.6      54.5
## 113       four       wagon   fwd           95.7   169.7     63.6      59.1
## 114       four       wagon   4wd           95.7   169.7     63.6      59.1
## 115       four       wagon   4wd           95.7   169.7     63.6      59.1
## 116       four       sedan   fwd           95.7   166.3     64.4      53.0
## 117       four   hatchback   fwd           95.7   166.3     64.4      52.8
## 118       four       sedan   fwd           95.7   166.3     64.4      53.0
## 119       four   hatchback   fwd           95.7   166.3     64.4      52.8
## 120       four       sedan   fwd           95.7   166.3     64.4      53.0
## 121       four   hatchback   fwd           95.7   166.3     64.4      52.8
## 122       four       sedan   fwd           95.7   166.3     64.4      52.8
## 123        two       sedan   rwd           94.5   168.7     64.0      52.6
## 124        two   hatchback   rwd           94.5   168.7     64.0      52.6
## 125        two       sedan   rwd           94.5   168.7     64.0      52.6
## 126        two   hatchback   rwd           94.5   168.7     64.0      52.6
## 127        two     hardtop   rwd           98.4   176.2     65.6      52.0
## 128        two     hardtop   rwd           98.4   176.2     65.6      52.0
## 129        two   hatchback   rwd           98.4   176.2     65.6      52.0
## 130        two     hardtop   rwd           98.4   176.2     65.6      52.0
## 131        two   hatchback   rwd           98.4   176.2     65.6      52.0
## 132        two convertible   rwd           98.4   176.2     65.6      53.0
## 133       four       sedan   fwd          102.4   175.6     66.5      54.9
## 134       four       sedan   fwd          102.4   175.6     66.5      54.9
## 135       four   hatchback   fwd          102.4   175.6     66.5      53.9
## 136       four       sedan   fwd          102.4   175.6     66.5      54.9
## 137       four   hatchback   fwd          102.4   175.6     66.5      53.9
## 138        two   hatchback   rwd          102.9   183.5     67.7      52.0
## 139        two   hatchback   rwd          102.9   183.5     67.7      52.0
## 140       four       sedan   rwd          104.5   187.8     66.5      54.1
##     AzamiAgirlik MotorTuru SilindirSayisi MotorHacmi YakitSistemi
## 110         1985       ohc           four         92         2bbl
## 111         2040       ohc           four         92         2bbl
## 112         2015       ohc           four         92         2bbl
```

```
## 113          2280      ohc          four       92         2bbl
## 114          2290      ohc          four       92         2bbl
## 115          3110      ohc          four       92         2bbl
## 116          2081      ohc          four       98         2bbl
## 117          2109      ohc          four       98         2bbl
## 118          2275      ohc          four      110          idi
## 119          2275      ohc          four      110          idi
## 120          2094      ohc          four       98         2bbl
## 121          2122      ohc          four       98         2bbl
## 122          2140      ohc          four       98         2bbl
## 123          2169      ohc          four       98         2bbl
## 124          2204      ohc          four       98         2bbl
## 125          2265     dohc          four       98         mpfi
## 126          2300     dohc          four       98         mpfi
## 127          2540      ohc          four      146         mpfi
## 128          2536      ohc          four      146         mpfi
## 129          2551      ohc          four      146         mpfi
## 130          2679      ohc          four      146         mpfi
## 131          2714      ohc          four      146         mpfi
## 132          2975      ohc          four      146         mpfi
## 133          2326      ohc          four      122         mpfi
## 134          2480      ohc          four      110          idi
## 135          2414      ohc          four      122         mpfi
## 136          2414      ohc          four      122         mpfi
## 137          2458      ohc          four      122         mpfi
## 138          2976     dohc           six      171         mpfi
## 139          3016     dohc           six      171         mpfi
## 140          3131     dohc           six      171         mpfi
##     SilindirCapi Zamanlama SikistirmaOrani BeygirGucu MaximumDevir
## 110         3.05      3.03             9.0         62         4800
## 111         3.05      3.03             9.0         62         4800
## 112         3.05      3.03             9.0         62         4800
## 113         3.05      3.03             9.0         62         4800
## 114         3.05      3.03             9.0         62         4800
## 115         3.05      3.03             9.0         62         4800
## 116         3.19      3.03             9.0         70         4800
## 117         3.19      3.03             9.0         70         4800
## 118         3.27      3.35            22.5         56         4500
## 119         3.27      3.35            22.5         56         4500
## 120         3.19      3.03             9.0         70         4800
## 121         3.19      3.03             9.0         70         4800
## 122         3.19      3.03             9.0         70         4800
## 123         3.19      3.03             9.0         70         4800
## 124         3.19      3.03             9.0         70         4800
## 125         3.24      3.08             9.4        112         6600
```

```
## 126          3.24      3.08              9.4        112          6600
## 127          3.62      3.50              9.3        116          4800
## 128          3.62      3.50              9.3        116          4800
## 129          3.62      3.50              9.3        116          4800
## 130          3.62      3.50              9.3        116          4800
## 131          3.62      3.50              9.3        116          4800
## 132          3.62      3.50              9.3        116          4800
## 133          3.31      3.54              8.7         92          4200
## 134          3.27      3.35             22.5         73          4500
## 135          3.31      3.54              8.7         92          4200
## 136          3.31      3.54              8.7         92          4200
## 137          3.31      3.54              8.7         92          4200
## 138          3.27      3.35              9.3        161          5200
## 139          3.27      3.35              9.3        161          5200
## 140          3.27      3.35              9.2        156          5200
##     SehirIciHarcama SehirDisiHarcama Fiyat
## 110              35               39  5348
## 111              31               38  6338
## 112              31               38  6488
## 113              31               37  6918
## 114              27               32  7898
## 115              27               32  8778
## 116              30               37  6938
## 117              30               37  7198
## 118              34               36  7898
## 119              38               47  7788
## 120              38               47  7738
## 121              28               34  8358
## 122              28               34  9258
## 123              29               34  8058
## 124              29               34  8238
## 125              26               29  9298
## 126              26               29  9538
## 127              24               30  8449
## 128              24               30  9639
## 129              24               30  9989
## 130              24               30 11199
## 131              24               30 11549
## 132              24               30 17669
## 133              29               34  8948
## 134              30               33 10698
## 135              27               32  9988
## 136              27               32 10898
## 137              27               32 11248
## 138              20               24 16558
```

```
## 139             19                 24 15998
## 140             20                 24 15690
```

```
# Toyota Marka Araçların Dizel yakıt kullananlarının bulunması yöntem 1

toyota.dizeller <- toyota.veri$YakitTuru=="diesel"

dizel.toyotalar <- toyota.veri[toyota.dizeller,]

print(dizel.toyotalar)
```

```
##     SiraNo RiskDerecesi KayipDegeri Uretici YakitTuru Enjeksiyon
## 118    118            0          91  toyota    diesel        std
## 119    119            0          91  toyota    diesel        std
## 134    134           -1          65  toyota    diesel      turbo
##     KapiSayisi GovdeTuru Cekis DingilAcikligi Uzunluk Genislik Yukseklik
## 118       four     sedan   fwd           95.7   166.3     64.4      53.0
## 119       four hatchback   fwd           95.7   166.3     64.4      52.8
## 134       four     sedan   fwd          102.4   175.6     66.5      54.9
##     AzamiAgirlik MotorTuru SilindirSayisi MotorHacmi YakitSistemi
## 118         2275       ohc           four        110          idi
## 119         2275       ohc           four        110          idi
## 134         2480       ohc           four        110          idi
##     SilindirCapi Zamanlama SikistirmaOrani BeygirGucu MaximumDevir
## 118         3.27      3.35            22.5         56         4500
## 119         3.27      3.35            22.5         56         4500
## 134         3.27      3.35            22.5         73         4500
##     SehirIciHarcama SehirDisiHarcama Fiyat
## 118              34               36  7898
## 119              38               47  7788
## 134              30               33 10698
```

```
# Toyota Marka Araçların Dizel yakıt kullananlarının bulunması yöntem 2

dizeller <- veri$YakitTuru == "diesel"

toyota.dizeller <- toyotalar & dizeller

dizel.toyotalar = veri[toyota.dizeller,]

print(dizel.toyotalar)
```

```
##     SiraNo RiskDerecesi KayipDegeri Uretici YakitTuru Enjeksiyon
## 118    118            0          91  toyota    diesel        std
```

```
## 119    119            0          91  toyota    diesel         std
## 134    134           -1          65  toyota    diesel       turbo
##     KapiSayisi GovdeTuru Cekis DingilAcikligi Uzunluk Genislik Yukseklik
## 118       four     sedan   fwd           95.7   166.3     64.4      53.0
## 119       four hatchback   fwd           95.7   166.3     64.4      52.8
## 134       four     sedan   fwd          102.4   175.6     66.5      54.9
##     AzamiAgirlik MotorTuru SilindirSayisi MotorHacmi YakitSistemi
## 118         2275       ohc           four        110          idi
## 119         2275       ohc           four        110          idi
## 134         2480       ohc           four        110          idi
##     SilindirCapi Zamanlama SikistirmaOrani BeygirGucu MaximumDevir
## 118         3.27      3.35            22.5         56         4500
## 119         3.27      3.35            22.5         56         4500
## 134         3.27      3.35            22.5         73         4500
##     SehirIciHarcama SehirDisiHarcama Fiyat
## 118              34               36  7898
## 119              38               47  7788
## 134              30               33 10698
```

```r
# Yeni bir sütun eklenmesi

veri$TLFiyat = veri$Fiyat * 3.6050

print(summary(veri$TLFiyat))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18450   26580   33280   41260   53060  126400
```

```r
# Toyota marka araçların ortalama fiyatının bulunması

toyota.ortalama.fiyat <- mean(veri$Fiyat[veri$Uretici=="toyota"])

print(toyota.ortalama.fiyat)
```

```
## [1] 9696.645
```

```r
# Kaç tane benzinli araç olduğunun bulunması

print(length(veri$SiraNo[veri$YakitTuru=='gas']))
```

```
## [1] 144
```

```r
print(nrow(veri[veri$YakitTuru=='gas',]))
```

```
## [1] 144
```

```r
# TL Fiyatı 100000 'den büyük araçların bulunması

print(veri[veri$TLFiyat>100000,])
```

```
##    SiraNo RiskDerecesi KayipDegeri       Uretici YakitTuru Enjeksiyon
## 33     33            0         145        jaguar       gas        std
## 46     46           -1          93 mercedes-benz    diesel      turbo
## 47     47            0          93 mercedes-benz    diesel      turbo
## 48     48           -1          93 mercedes-benz    diesel      turbo
## 49     49            3         142 mercedes-benz       gas        std
##    KapiSayisi   GovdeTuru Cekis DingilAcikligi Uzunluk Genislik Yukseklik
## 33       four       sedan   rwd          113.0   199.6     69.6      52.8
## 46       four       wagon   rwd          110.0   190.9     70.3      58.7
## 47        two     hardtop   rwd          106.7   187.5     70.3      54.9
## 48       four       sedan   rwd          115.6   202.6     71.7      56.3
## 49        two convertible   rwd           96.6   180.3     70.5      50.8
##    AzamiAgirlik MotorTuru SilindirSayisi MotorHacmi YakitSistemi
## 33         4066      dohc            six        258         mpfi
## 46         3750       ohc           five        183          idi
## 47         3495       ohc           five        183          idi
## 48         3770       ohc           five        183          idi
## 49         3685      ohcv          eight        234         mpfi
##    SilindirCapi Zamanlama SikistirmaOrani BeygirGucu MaximumDevir
## 33         3.63      4.17             8.1        176         4750
## 46         3.58      3.64            21.5        123         4350
## 47         3.58      3.64            21.5        123         4350
## 48         3.58      3.64            21.5        123         4350
## 49         3.46      3.10             8.3        155         4750
##    SehirIciHarcama SehirDisiHarcama Fiyat  TLFiyat
## 33              15               19 32250 116261.2
## 46              22               25 28248 101834.0
## 47              22               25 28176 101574.5
## 48              22               25 31600 113918.0
## 49              16               18 35056 126376.9
```

```r
# TL Fiyatı en pahalı ve en ucuz olan araçların bulunması

enBuyuk <- veri[veri$TLFiyat==max(veri$TLFiyat),]
enKucuk <- veri[veri$TLFiyat==min(veri$TLFiyat),]
```

```
sonuc <- rbind(enBuyuk,enKucuk)

print(sonuc)
```

```
##    SiraNo RiskDerecesi KayipDegeri       Uretici YakitTuru Enjeksiyon
## 49     49            3         142 mercedes-benz       gas        std
## 98     98            2          83        subaru       gas        std
##    KapiSayisi   GovdeTuru Cekis DingilAcikligi Uzunluk Genislik Yukseklik
## 49        two convertible   rwd           96.6   180.3     70.5      50.8
## 98        two   hatchback   fwd           93.7   156.9     63.4      53.7
##    AzamiAgirlik MotorTuru SilindirSayisi MotorHacmi YakitSistemi
## 49         3685      ohcv          eight        234         mpfi
## 98         2050      ohcf           four         97         2bbl
##    SilindirCapi Zamanlama SikistirmaOrani BeygirGucu MaximumDevir
## 49         3.46      3.10             8.3        155         4750
## 98         3.62      2.36             9.0         69         4900
##    SehirIciHarcama SehirDisiHarcama Fiyat    TLFiyat
## 49              16               18 35056 126376.88
## 98              31               36  5118  18450.39
```

```
# Araç üteticilerini listelenmesi

ureticiler <- levels(veri$Uretici)

print(ureticiler)
```

```
##  [1] "audi"          "bmw"           "chevrolet"     "dodge"
##  [5] "honda"         "jaguar"        "mazda"         "mercedes-benz"
##  [9] "mitsubishi"    "nissan"        "peugot"        "plymouth"
## [13] "porsche"       "saab"          "subaru"        "toyota"
## [17] "volkswagen"    "volvo"
```

```
# Araç üreticilerine göre ortalama fiyatların bulunması

ortalama.listesi = list()

length(ortalama.listesi) <- length(ureticiler)

names(ortalama.listesi) <- ureticiler

for(uretici in ureticiler) {
  ortalama.listesi[[uretici]] <- mean(veri$Fiyat[veri$Uretici==uretici])
}
```

```r
print(ortalama.listesi)
```

```
## $audi
## [1] 18246.25
##
## $bmw
## [1] 18857.5
##
## $chevrolet
## [1] 6007
##
## $dodge
## [1] 7790.125
##
## $honda
## [1] 8184.692
##
## $jaguar
## [1] 32250
##
## $mazda
## [1] 9080
##
## $`mercedes-benz`
## [1] 29726.4
##
## $mitsubishi
## [1] 7813
##
## $nissan
## [1] 10415.67
##
## $peugot
## [1] 15758.57
##
## $plymouth
## [1] 7163.333
##
## $porsche
## [1] 22018
##
## $saab
## [1] 15223.33
##
```

```
## $subaru
## [1] 8541.25
##
## $toyota
## [1] 9696.645
##
## $volkswagen
## [1] 8738.125
##
## $volvo
## [1] 18063.18
```

BÖLÜM

# 4 R ile Tanımlayıcı İstatistikler ve Nominal Değerlerin İncelenmesi

R Programlama dili ile verinize ait tanımlayıcı istatistikleri aşağıdaki şekilde elde edebilirsiniz.

## 4.1 Veri Yükleme

İlk olarak verinizi bir klasöre indirim ve onu çalışma klasürü olarak belirleyin. Daha sonra csv dosyanızı R içerisine alın.

- NOT: Burada read.csv2, Türkçe Excel ile oluşturulmuş CSV'leri yüklemek için kullanılmaktadır.

```
analiz_verisi <- read.csv2('imports-85.csv')
```

## 4.2 Veri Bölme

*summary()* Fonksiyonuyla verimizin yapısı hakkında bilgi edinelim. Fakat burada summary fonksiyonunu sadece sayısal değişkenlerimiz için kullanalım. Bunun için, ilk olarak veri içerisindeki numeric ve integer alanların hangileri olduklarını bulmalısınız. Eğer tek bir vektörün veri türünü öğrenmek isteseydik, o zaman class fonksiyonunu kullanabilirdik. Fakat burada, bir data frame'in içerisindeki bütün sütunların veri türünü öğrenmek istiyoruz. O yüzden *sapply()* fonksiyonunu kullanmalısınız.

- NOT: *sapply()* komutu belirttiğiniz fonksiyonu data frame içerisindeki bütün sütunlara uygular ve sonuçlarını içeren bir vektör oluşturur. Burada, bütün sütunlara *class()* fonksiyonunu uygulamak için kullanılmıştır. Elde edilen sonuçların numeric veya integer değer olanları bulunup filtrelenmiştir.

```
numeric_sutunlar <- sapply(analiz_verisi, class)=="numeric"
integer_sutunlar <- sapply(analiz_verisi, class)=="integer"

sayisal_sutunlar <- numeric_sutunlar | integer_sutunlar

sayisal_veri <- analiz_verisi[,sayisal_sutunlar]

print(summary(sayisal_veri))
```

```
##      SiraNo       RiskDerecesi      KayipDegeri    DingilAcikligi
##  Min.   :  1.0   Min.   :-2.0000   Min.   : 65.0   Min.   : 86.60
##  1st Qu.: 40.5   1st Qu.: 0.0000   1st Qu.: 94.0   1st Qu.: 94.50
##  Median : 80.0   Median : 1.0000   Median :113.0   Median : 96.90
##  Mean   : 80.0   Mean   : 0.7358   Mean   :121.1   Mean   : 98.26
##  3rd Qu.:119.5   3rd Qu.: 2.0000   3rd Qu.:148.0   3rd Qu.:100.80
##  Max.   :159.0   Max.   : 3.0000   Max.   :256.0   Max.   :115.60
##     Uzunluk         Genislik        Yukseklik      AzamiAgirlik
##  Min.   :141.1   Min.   :60.30   Min.   :49.40   Min.   :1488
##  1st Qu.:165.7   1st Qu.:64.00   1st Qu.:52.25   1st Qu.:2066
##  Median :172.4   Median :65.40   Median :54.10   Median :2340
##  Mean   :172.4   Mean   :65.61   Mean   :53.90   Mean   :2461
##  3rd Qu.:177.8   3rd Qu.:66.50   3rd Qu.:55.50   3rd Qu.:2810
##  Max.   :202.6   Max.   :71.70   Max.   :59.80   Max.   :4066
##    MotorHacmi     SilindirCapi    Zamanlama     SikistirmaOrani
##  Min.   : 61.0   Min.   :2.54   Min.   :2.070   Min.   : 7.00
##  1st Qu.: 97.0   1st Qu.:3.05   1st Qu.:3.105   1st Qu.: 8.70
##  Median :110.0   Median :3.27   Median :3.270   Median : 9.00
##  Mean   :119.2   Mean   :3.30   Mean   :3.236   Mean   :10.16
##  3rd Qu.:135.0   3rd Qu.:3.56   3rd Qu.:3.410   3rd Qu.: 9.40
##  Max.   :258.0   Max.   :3.94   Max.   :4.170   Max.   :23.00
##    BeygirGucu      MaximumDevir  SehirIciHarcama SehirDisiHarcama
##  Min.   : 48.00   Min.   :4150   Min.   :15.00   Min.   :18.00
##  1st Qu.: 69.00   1st Qu.:4800   1st Qu.:23.00   1st Qu.:28.00
##  Median : 88.00   Median :5200   Median :26.00   Median :32.00
##  Mean   : 95.84   Mean   :5114   Mean   :26.52   Mean   :32.08
##  3rd Qu.:114.00   3rd Qu.:5500   3rd Qu.:31.00   3rd Qu.:37.00
##  Max.   :200.00   Max.   :6600   Max.   :49.00   Max.   :54.00
##      Fiyat
##  Min.   : 5118
##  1st Qu.: 7372
##  Median : 9233
##  Mean   :11446
##  3rd Qu.:14720
##  Max.   :35056
```

Ayrıca R'ın genişletilebilir yapısı sayesinde sayısal değişkenleriniz için istediğiniz istatistiksel özeti, hazır paketler kullanarak alabilirsiniz. Bunları bilgisayarınıza yüklemek için bilgisayarınızda RTools yazılımının yüklü olması gerekmektedir. Şu aşamada kullanacağımız paket, *psych* paketidir. Bu paket, bilgiayarınıza aşağıdaki R komutu kullanılarak yüklenebilir.

```
install.packages('psych')
```

Bir paket bilgisayarınıza yüklendikten sonra *library* komutu ile kullanıma alınabilir.

```
library(psych)
```

Dikkat ederseniz paket bilgisayarınıza yüklenirken, tırnak işareti yazılmalıdır. Ondan sonra paketi kullanıma alırken doğrudan adını yazmanız yeterli olacaktır. *psych* paketini kullanıma aldıktan sonra, iki komut yardımıyla veri çerçevesinin sayısal değişkenleri hakkında özet bilgi alabilirsiniz. Bu komutlar *describe* ve *describeBy* komutlarıdır. *describe* komutu bütün veri çerçevesinin tanımlayısı istatistiklerini elde etmek için kullanılırken, *describeBy* komutu veriyi bir nominal değişkene bağlı olarak gruplayarak her gruba ait tanımlayıcı istatistikleri çıkartır.

```
describe(sayisal_veri)
```

```
##                  vars   n     mean      sd  median  trimmed     mad
## SiraNo              1 159    80.00   46.04   80.00    80.00   59.30
## RiskDerecesi        2 159     0.74    1.19    1.00     0.71    1.48
## KayipDegeri         3 159   121.13   35.65  113.00   118.43   32.62
## DingilAcikligi      4 159    98.26    5.17   96.90    97.72    3.56
## Uzunluk             5 159   172.41   11.52  172.40   172.49    9.04
## Genislik            6 159    65.61    1.95   65.40    65.39    1.78
## Yukseklik           7 159    53.90    2.27   54.10    53.85    2.37
## AzamiAgirlik        8 159  2461.14  481.94 2340.00  2422.71  492.22
## MotorHacmi          9 159   119.23   30.46  110.00   114.88   19.27
## SilindirCapi       10 159     3.30    0.27    3.27     3.29    0.36
## Zamanlama          11 159     3.24    0.29    3.27     3.27    0.21
## SikistirmaOrani    12 159    10.16    3.89    9.00     9.08    0.59
## BeygirGucu         13 159    95.84   30.72   88.00    92.43   29.65
## MaximumDevir       14 159  5113.84  465.75 5200.00  5114.34  444.78
## SehirIciHarcama    15 159    26.52    6.10   26.00    26.09    5.93
## SehirDisiHarcama   16 159    32.08    6.46   32.00    31.74    5.93
## Fiyat              17 159 11445.73 5877.86 9233.00 10543.25 3534.52
##                     min      max   range  skew kurtosis     se
## SiraNo             1.00   159.00   158.0  0.00    -1.22   3.65
## RiskDerecesi      -2.00     3.00     5.0  0.09    -0.58   0.09
## KayipDegeri       65.00   256.00   191.0  0.82     0.52   2.83
## DingilAcikligi    86.60   115.60    29.0  0.90     0.53   0.41
## Uzunluk          141.10   202.60    61.5 -0.06    -0.27   0.91
```

```
## Genislik           60.30    71.70    11.4  0.90     0.74   0.15
## Yukseklik          49.40    59.80    10.4  0.17    -0.34   0.18
## AzamiAgirlik     1488.00  4066.00  2578.0  0.77     0.07  38.22
## MotorHacmi         61.00   258.00   197.0  1.46     2.75   2.42
## SilindirCapi        2.54     3.94     1.4  0.15    -0.86   0.02
## Zamanlama           2.07     4.17     2.1 -0.97     2.35   0.02
## SikistirmaOrani     7.00    23.00    16.0  2.66     5.40   0.31
## BeygirGucu         48.00   200.00   152.0  0.90     0.21   2.44
## MaximumDevir     4150.00  6600.00  2450.0  0.15     0.31  36.94
## SehirIciHarcama    15.00    49.00    34.0  0.72     1.02   0.48
## SehirDisiHarcama   18.00    54.00    36.0  0.59     0.72   0.51
## Fiyat            5118.00 35056.00 29938.0  1.56     2.41 466.14
```

```
describeBy(sayisal_veri, group = analiz_verisi$KapiSayisi)
```

```
##
##  Descriptive statistics by group
## group: four
##                  vars  n     mean      sd  median  trimmed     mad     min
## SiraNo              1 95    86.13   46.78   88.00    87.29   59.30    1.00
## RiskDerecesi        2 95     0.13    0.97    0.00     0.08    1.48   -2.00
## KayipDegeri         3 95   109.65   30.36  102.00   107.57   19.27   65.00
## DingilAcikligi      4 95   100.07    5.34   97.30    99.61    4.15   93.10
## Uzunluk             5 95   176.30   10.16  174.60   176.37   11.56  157.10
## Genislik            6 95    66.00    1.99   65.40    65.80    1.63   62.50
## Yukseklik           7 95    54.77    2.11   54.90    54.75    1.78   50.60
## AzamiAgirlik        8 95  2563.35  477.51 2410.00  2526.99  515.94 1909.00
## MotorHacmi          9 95   121.35   28.83  110.00   117.14   19.27   90.00
## SilindirCapi       10 95     3.35    0.26    3.31     3.34    0.34    2.91
## Zamanlama          11 95     3.23    0.30    3.27     3.25    0.21    2.19
## SikistirmaOrani    12 95    10.53    4.43    9.00     9.49    0.59    7.00
## BeygirGucu         13 95    96.59   28.59   92.00    93.45   32.62   52.00
## MaximumDevir       14 95  5049.47  451.68 5100.00  5065.58  444.78 4150.00
## SehirIciHarcama    15 95    25.69    5.33   26.00    25.60    5.93   15.00
## SehirDisiHarcama   16 95    31.23    5.98   32.00    31.10    5.93   19.00
## Fiyat              17 95 12225.56 5811.54 9988.00 11335.73 4136.45 6229.00
##                       max    range  skew kurtosis     se
## SiraNo             159.00   158.00 -0.16    -1.14   4.80
## RiskDerecesi         2.00     4.00  0.23    -0.30   0.10
## KayipDegeri        192.00   127.00  0.77    -0.31   3.12
## DingilAcikligi     115.60    22.50  0.77    -0.46   0.55
## Uzunluk            202.60    45.50  0.18    -0.59   1.04
## Genislik            71.70     9.20  0.86     0.26   0.20
## Yukseklik           59.80     9.20  0.08    -0.14   0.22
```

```
## AzamiAgirlik      4066.00  2157.00  0.74    -0.07  48.99
## MotorHacmi         258.00   168.00  1.67     4.12   2.96
## SilindirCapi         3.78     0.87  0.13    -1.21   0.03
## Zamanlama            4.17     1.98 -0.65     1.40   0.03
## SikistirmaOrani     23.00    16.00  2.16     2.88   0.45
## BeygirGucu         176.00   124.00  0.82    -0.01   2.93
## MaximumDevir      6000.00  1850.00 -0.24    -0.59  46.34
## SehirIciHarcama     38.00    23.00  0.18    -0.46   0.55
## SehirDisiHarcama    47.00    28.00  0.29    -0.16   0.61
## Fiyat            32250.00 26021.00  1.36     1.53 596.25
## --------------------------------------------------------
## group: two
##                  vars  n     mean      sd  median trimmed     mad     min
## SiraNo              1 64    70.91   43.72   63.50   70.04   53.37    5.00
## RiskDerecesi        2 64     1.64    0.88    1.50    1.63    0.74    0.00
## KayipDegeri         3 64   138.17   36.31  134.00  135.73   40.03   83.00
## DingilAcikligi      4 64    95.58    3.48   94.50   95.49    2.08   86.60
## Uzunluk             5 64   166.64   11.06  167.50  166.74   13.05  141.10
## Genislik            6 64    65.02    1.74   64.30   64.82    1.04   60.30
## Yukseklik           7 64    52.61    1.85   52.60   52.57    2.59   49.40
## AzamiAgirlik        8 64  2309.42  450.74 2215.00 2262.52  440.33 1488.00
## MotorHacmi          9 64   116.08   32.71  100.50  111.50   14.08   61.00
## SilindirCapi       10 64     3.23    0.27    3.16    3.22    0.28    2.54
## Zamanlama          11 64     3.25    0.29    3.29    3.28    0.23    2.07
## SikistirmaOrani    12 64     9.62    2.86    9.20    9.13    0.30    7.00
## BeygirGucu         13 64    94.72   33.84   84.50   90.87   25.20   48.00
## MaximumDevir       14 64  5209.38  473.41 5200.00 5181.73  444.78 4250.00
## SehirIciHarcama    15 64    27.75    6.95   26.00   27.17    7.41   16.00
## SehirDisiHarcama   16 64    33.34    6.97   32.00   32.85    6.67   18.00
## Fiyat              17 64 10288.17 5828.90 8016.50 9279.33 2779.13 5118.00
##                       max    range  skew kurtosis     se
## SiraNo             148.00   143.00  0.20    -1.33   5.46
## RiskDerecesi         3.00     3.00  0.20    -0.95   0.11
## KayipDegeri        256.00   173.00  0.79     0.57   4.54
## DingilAcikligi     106.70    20.10  0.28     1.36   0.44
## Uzunluk            187.50    46.40 -0.08    -0.66   1.38
## Genislik            70.50    10.20  0.94     1.59   0.22
## Yukseklik           56.10     6.70  0.12    -1.00   0.23
## AzamiAgirlik      3685.00  2197.00  0.92     0.36  56.34
## MotorHacmi         234.00   173.00  1.30     1.34   4.09
## SilindirCapi         3.94     1.40  0.29    -0.52   0.03
## Zamanlama            3.90     1.83 -1.47     3.85   0.04
## SikistirmaOrani     23.00    16.00  3.92    14.52   0.36
## BeygirGucu         200.00   152.00  0.97     0.21   4.23
## MaximumDevir      6600.00  2350.00  0.60     0.69  59.18
```

```
## SehirIciHarcama      49.00    33.00  0.87     0.80   0.87
## SehirDisiHarcama     54.00    36.00  0.75     0.82   0.87
## Fiyat             35056.00 29938.00  1.98     4.38 728.61
```

## 4.3 Nominal Değerler İçin Tanımlayıcı İstatistikler

Şimdi de nominal değişkenler için tanımlayıcı istatistikleri elde etmeye çalışalım. Geriye nominal alanlar kaldığı için, bu değişkenlerin hangi değerden kaç adet içerdiğinin bulunması yeterli olacaktır. Bu işlemi gerçekleştirebilmek için **for** döngüsü ve *table()* fonksiyonundan yararlanacağız.

### 4.3.1 Tablo Oluşturma

*table* Fonksiyonu R'ın en esnek fonksiyonlarından biridir. *table()* fonksiyonu tek bir nominal değer ile kullanılırsa, size nominal değerin frekans tablosunu oluşturur. Şimdi bütün nominal değerlerin frekans tablolarını oluşturalım.

```
nominal_sutunlar <- !sayisal_sutunlar

nominal_veri <- analiz_verisi[,nominal_sutunlar]

for(i in 1:ncol(nominal_veri)) {
  print(table(nominal_veri[,i]))
}
```

```
##
##          audi           bmw     chevrolet         dodge         honda
##             4             4             3             8            13
##        jaguar         mazda mercedes-benz    mitsubishi        nissan
##             1            11             5            10            18
##        peugot      plymouth       porsche          saab        subaru
##             7             6             1             6            12
##        toyota    volkswagen         volvo
##            31             8            11
##
## diesel    gas
##     15    144
##
##   std turbo
##   132    27
##
## four  two
##   95   64
##
## convertible     hardtop   hatchback       sedan       wagon
```

```
##           2           5          56          79          17
##
## 4wd fwd rwd
##   8 105  46
##
## dohc    l  ohc ohcf ohcv
##    8    8  123   12    8
##
## eight  five  four   six three
##     1     7   136    14     1
##
## 1bbl 2bbl  idi  mfi mpfi spdi
##   11   63   15    1   64    5
```

### 4.3.2 Marjinal Değerleri Ekleme

*table()* fonksiyonu ile bir tablo oluşturdukran sonra *addmargins* fonksiyonu ile marjinal toplamları tablonuza ekleyebilirsiniz.

```
nominal_sutunlar <- !sayisal_sutunlar

nominal_veri <- analiz_verisi[,nominal_sutunlar]

for(i in 1:ncol(nominal_veri)) {
  frekans.tablosu <- table(nominal_veri[,i])
  marjinal.frekans.tablosu <- addmargins(frekans.tablosu)
  print(marjinal.frekans.tablosu)
}
```

```
##
##          audi           bmw     chevrolet         dodge         honda
##             4             4             3             8            13
##        jaguar         mazda mercedes-benz    mitsubishi        nissan
##             1            11             5            10            18
##        peugot      plymouth       porsche          saab        subaru
##             7             6             1             6            12
##        toyota    volkswagen         volvo           Sum
##            31             8            11           159
##
## diesel    gas    Sum
##     15    144    159
##
##   std turbo   Sum
##   132    27   159
##
```

```
## four  two  Sum
##   95   64  159
##
## convertible     hardtop   hatchback       sedan       wagon          Sum
##           2           5          56          79          17          159
##
## 4wd fwd rwd Sum
##   8 105  46 159
##
## dohc    l  ohc ohcf ohcv  Sum
##    8    8  123   12    8  159
##
## eight  five  four   six three   Sum
##     1     7   136    14     1   159
##
## 1bbl 2bbl  idi  mfi mpfi spdi  Sum
##   11   63   15    1   64    5  159
```

### 4.3.3 Oran Tabloları Oluşturma

*table()* fonksiyonu ile bir tablo oluşturduktan sonra *prop.table()* fonksiyonu ile bu tablonun yüzdelikli değerlerini hesaplattırabilirsiniz. Fakat dikkat etmeniz gereken nokta, *prop.table()* fonksiyonunun 0 ile 1 arasında değer ürettiğidir. Bu sebeple, bu değeri 100 ile çarpıp, virgülden sonra iki hane olacak şekilde yuvarlaryarak göstermeniz gerekebilir.

```
nominal_sutunlar <- !sayisal_sutunlar

nominal_veri <- analiz_verisi[,nominal_sutunlar]

for(i in 1:ncol(nominal_veri)) {
  frekans.tablosu <- table(nominal_veri[,i])
  yuzdelikli.frekans.tablosu <- prop.table(frekans.tablosu)
  # Yukarıda bahsedilen yuvarlamayı yapalım.
  print(round(100*yuzdelikli.frekans.tablosu,2))
}
```

```
##
##          audi            bmw      chevrolet          dodge          honda
##          2.52           2.52           1.89           5.03           8.18
##        jaguar          mazda mercedes-benz     mitsubishi         nissan
##          0.63           6.92           3.14           6.29          11.32
##        peugot       plymouth        porsche           saab         subaru
##          4.40           3.77           0.63           3.77           7.55
##        toyota     volkswagen          volvo
##         19.50           5.03           6.92
```

```
##
## diesel    gas
##   9.43  90.57
##
##   std turbo
## 83.02 16.98
##
##  four   two
## 59.75 40.25
##
## convertible     hardtop   hatchback       sedan       wagon
##        1.26        3.14       35.22       49.69       10.69
##
##   4wd   fwd   rwd
##  5.03 66.04 28.93
##
##  dohc     l   ohc  ohcf  ohcv
##  5.03  5.03 77.36  7.55  5.03
##
## eight  five  four   six three
##  0.63  4.40 85.53  8.81  0.63
##
##  1bbl  2bbl   idi   mfi  mpfi  spdi
##  6.92 39.62  9.43  0.63 40.25  3.14
```

### 4.3.4 Çapraz Tablo Oluşturma

*table()* fonksiyonuna bir yerine iki nominal değer verirseniz, bu iki değerin oluşturacağı Çapraz Tablo'yu elde edersiniz. Çapraz Tablo, satırları A değişeninin nominal değerlerini, sütunları B değişkeninin nominal değerlerini içeren, hücrelerinde ise, satır ve sütuna ait nominal değerlerin kesişiminin kaç kez geçtiğini gösteren bir tablodur. Örneğin, yakıt türü ve kapı sayısına ait bir çapraz tablo aşağıdaki gibi oluçturulabilir.

```
capraz.tablo <- table(nominal_veri$YakitTuru, nominal_veri$KapiSayisi)

print(capraz.tablo)
```

```
##
##          four two
##   diesel   12   3
##   gas      83  61
```

Tabloya göre, verimizde hem *diesel* hem de *dört* kapılı 12 araç bulunmaktadır. Yukarıda bahsedilen marjinal değerler ve oransal değerler çapraz tablolaya uygulanabilir.

```
marjinal.frekans.tablosu <- addmargins(capraz.tablo)

print("Marjinal Frekans Tablosu")
```

```
## [1] "Marjinal Frekans Tablosu"
```

```
print(marjinal.frekans.tablosu)
```

```
##
##          four two Sum
##   diesel   12   3  15
##   gas      83  61 144
##   Sum      95  64 159
```

```
yuzdelikli.frekans.tablosu <- prop.table(capraz.tablo)

print("Yüzdelikli Frekans Tablosu")
```

```
## [1] "Yüzdelikli Frekans Tablosu"
```

```
print(round(100*yuzdelikli.frekans.tablosu,2))
```

```
##
##           four   two
##   diesel  7.55  1.89
##   gas    52.20 38.36
```

Bu tablolara göre, *diesel* ve *dört* kapılı araçlar toplan araçların %7.55'ini oluşturmaktadır. Bir veri kümesinde yer alan nominal değ?erleerin ikili bütün çapraz tabloları ele alınmalıdır. Bu işlem nominal değerler arasındaki ilişkiyi tespit etmek için önemli bir bilgi arz etmektedir. Bu sebeple, bu işlemi bütün nominal değer ikilileri için yapmak isteyebilirsiniz. Bunun için ya bütün değerlere çapraz tablo oluşturan komutları el ile yazmalısınız veya iç içe iki adet *for* döngüsü kullanmalısınız.

- NOT: Nominal değişken çiftleri ile çalışmak istediğiniz için siz tek bir *for* yetmeyecektir. Örneği inceleyin.

```
# Burada bütün nominal değerlerin yer aldığı nominal_veri data.frame'i kullanılıyor.
# ncol fonksiyonu sütun sayısını öğrenmek için kullanılan fonksiyondur.

# İlk for döngümüzü, 1'den sütun sayısının 1 eksiğine kadar yapıyoruz
for(i in 1:(ncol(nominal_veri)-1)) {
  # Data Frame'den ilk değişkeni çıkartalım.
```

```
  a <- nominal_veri[,i]
  # İçteki for döngümüzüde i+1'den sütun sayısına kadar yapıyoruz
  for(j in (i+1):ncol(nominal_veri)) {
    # Data Frame'den ikinci değişkeni çıkartalım.
    b <- nominal_veri[,j]

    # Yukarıdaki komutları buraya yapıştırın !!!
    capraz.tablo <- table(a, b)

    marjinal.frekans.tablosu <- addmargins(capraz.tablo)

    cat("Marjinal Frekans Tablosu\n")
    print(marjinal.frekans.tablosu)

    yuzdelikli.frekans.tablosu <- prop.table(capraz.tablo)

    cat("Oransal Frekans Tablosu\n")
    print(round(100*yuzdelikli.frekans.tablosu,2))
  }
}
```

### 4.3.5 Çapraz Tablo Kullanarak Bağımsızlık Testleri

Nominal değişkenlerin arasındaki ilişkileri bulmak için çapraz tablo istatistikleri kullanılır, bunlar Ki-Kare Testi ve Fisher'ın exact testleridir. Burada, hangi testin kullanılacağına tablo değerlerine bakılarak karar verilir. İlk olarak, çapraz tablo için Ki-Kare testi uygulanır. Eğer ki-kare testine göre çapraz tablonun bazı hücrelerin beklenen değeri 5'ten küçük ise Fisher testi yapılır. Aksi durumda Ki-Kare testinin sonuçları gösterilir. Burada test edilen yokluk (null, H0) hipotezi değişkenlerin bağımsız olduğudur. İki test sonucunda da elde edilecek p değeri 0.05'ten küçük olursa yokluk hipotezi reddedilir ve değişkenler arasında ilişki olduğuna karar verilir.

#### 4.3.5.1 Ki-Kare Testi

Ki-Kare testi *chisq.test()* fonksiyonu ile gerçekleştirilir. Eğer sonucu bir değişkene aktarılırsa, değişken (htest türünde) liste olur. Fonksiyonun sonucunu bir değişkene aktarmazsanız, test sonucunu ekrana çıktı verecektir.

```
t <- table(nominal_veri$Enjeksiyon,nominal_veri$KapiSayisi)
chisq.test(t)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  t
## X-squared = 1.0401, df = 1, p-value = 0.3078
```

Burada elde edilen p değeri 0.3078 olduğundan Yokluk hipotezi reddedilemez ve değişkenlerin bağımsız olduğu söylenir.

#### 4.3.5.2 Fisher Exact Testi

Fisher Exact testi ise *fisher.test()* fonksiyonu ile yapılır. Eğer sonucu bir değişkene aktarılırsa, değişken (htest türünde) liste olur. Fonksiyonun sonucunu bir değişkene aktarmazsanız, test sonucunu ekrana çıktı verecektir.

```
t <- table(nominal_veri$Enjeksiyon, nominal_veri$Uretici)
fisher.test(t,workspace = 2e07)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  t
## p-value = 0.000117
## alternative hypothesis: two.sided
```

Burada elde edilen p değeri 0.000117 olduğundan Yokluk hipotezi reddedilir ve değişkenler arasında bir ilişki olduğu söylenir.

#### 4.3.5.3 İlişkinin Gücünün Belirlenmesi

Eğer, iki *nominal* değişken arasında ilişki bulduysanız bu ilişkinin ne kadar güçlü olduğunu belirlemek için Cramer değerini kullanabilirsiniz. Bu değer 2x2'lik tablolar için [-1, 1] aralığında, diğer tablolar için [0, 1] aralığında değer alır. Hesaplanan Cramer değeri 1'e ne kadar yakınsa ilişkinin o kadar güçlü olduğu söylenebilir. Cramer değerini elde edebilmek için R'ın *vcd* paketinde yer alan *assocstats()* fonksiyonuna ihtiyacınız olacaktır. Bu paket R kurulumu ile beraber gelmez, onun yerine internetten indirip kurmanız gerekecektir.

```
install.packages('vcd')
```

Komut tamamlandığında artık bilgisayarınızda vcd paket hazır olacaktır. Bu aşamayı tamamladıktan sonra *assocstats()* fonksiyonu ile Cramer değerini elde edebilirsiniz.

```
library(vcd)

istatistik <- assocstats(t)

print(istatistik$cramer)
```

```
## [1] 0.526055
```

#### 4.3.5.4 Karar Verme İşeminin Otomatize Edilmesi

R'ın bir programlama dili olmasının avantajlarından faydalanarak, test sonuçlarını sizin adınıza değerlendirmesini sağlayabilirsiniz. Bunun için *if* komutunu kullanmanız yeterli olacaktır. If komutunun yapısı aşağıdaki şekildedir.

```
if(Şart) {
  Şart Doğruysa Çalışacak Komutlar
} else {
  Şart Yanlışsa Çalışacak Komutlar
}
```

Bağımsızlık testi için karar şartına bakacak olursanız, ilk olarak *chisq.test()* ile ki-kare testini gerçekleştirip p değerine göre karar verilebilir. Bunun için, *chisq.test()* fonksiyonunun çıktısını bir değişkene aktarmanız yeterli olacaktır. Bu değişken bir liste olup ki-kare testinin bütün çıktılarını içerir. Özellikle, *p.value* elemanı ki kare testinin p değerini içerir ve sadece bu elemana göre karar vermeniz yeterli olacktır. Bu arada, bütün hipotez testlerinin sonucunda *p.value* değeri hesaplanır.

- NOT: Eğer vcd paketini kuramadıysanız, "library(vcd)", "istatistik <- assocstats(t)" ve sonrasında ki satırı yazmasanız da olur.

```
library(vcd)

t <- table(nominal_veri$Enjeksiyon,nominal_veri$KapiSayisi)
test<-chisq.test(t)

if(test$p.value<0.05) {
  cat("Değişkenler arasında ilişki vardır.\n")
  istatistik <- assocstats(t)
  cat("İlişkinin derecesi ",istatistik$cramer," dir\n")
} else {
  cat("Değişkenler bağımsızdır.\n")
}
```

```
## Değişkenler bağımsızdır.
```

#### 4.3.5.5 Hazır Paketler Kullanılması

Bütün bu adımları, *gmodels* paketinde yer alan *Crosstable* fonksiyonu ile gerçekleştirebilirsiniz. Eğer bilgisayarınızda *gmodels* paketi yüklü değilse, *install.packages* ile yükleyebilirsiniz. Aşağıdaki örneği inceleyin.

```r
library(gmodels)

CrossTable(nominal_veri$KapiSayisi, nominal_veri$Enjeksiyon,
           chisq = TRUE, fisher = TRUE, mcnemar = TRUE,
           resid = TRUE, format = "SPSS")
```

```
##
##    Cell Contents
## |-------------------------|
## |                   Count |
## | Chi-square contribution |
## |             Row Percent |
## |          Column Percent |
## |           Total Percent |
## |                Residual |
## |-------------------------|
##
## Total Observations in Table:  159
##
##                         | nominal_veri$Enjeksiyon
## nominal_veri$KapiSayisi |      std  |    turbo  | Row Total |
## ------------------------|-----------|-----------|-----------|
##                    four |       76  |       19  |       95  |
##                         |    0.104  |    0.510  |           |
##                         |  80.000%  |  20.000%  |  59.748%  |
##                         |  57.576%  |  70.370%  |           |
##                         |  47.799%  |  11.950%  |           |
##                         |   -2.868  |    2.868  |           |
## ------------------------|-----------|-----------|-----------|
##                     two |       56  |        8  |       64  |
##                         |    0.155  |    0.757  |           |
##                         |  87.500%  |  12.500%  |  40.252%  |
##                         |  42.424%  |  29.630%  |           |
##                         |  35.220%  |   5.031%  |           |
##                         |    2.868  |   -2.868  |           |
## ------------------------|-----------|-----------|-----------|
##            Column Total |      132  |       27  |      159  |
##                         |  83.019%  |  16.981%  |           |
## ------------------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
```

```
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  1.525758     d.f. =  1     p =  0.2167503
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  1.040126     d.f. =  1     p =  0.3077922
##
##
## McNemar's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  18.25333     d.f. =  1     p =  1.933872e-05
##
## McNemar's Chi-squared test with continuity correction
## ------------------------------------------------------------
## Chi^2 =  17.28     d.f. =  1     p =  3.225641e-05
##
##
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Sample estimate odds ratio:  0.5733641
##
## Alternative hypothesis: true odds ratio is not equal to 1
## p =  0.282677
## 95% confidence interval:  0.2019521 1.49281
##
## Alternative hypothesis: true odds ratio is less than 1
## p =  0.1538223
## 95% confidence interval:  0 1.306008
##
## Alternative hypothesis: true odds ratio is greater than 1
## p =  0.9285521
## 95% confidence interval:  0.2373441 Inf
##
##
##
##        Minimum expected frequency: 10.86792
```

Burada *mcnemar* testi sadece 2x2'lik çapraz tablolar için kullanılmalıdır.

## 4.4 R Grafikleri

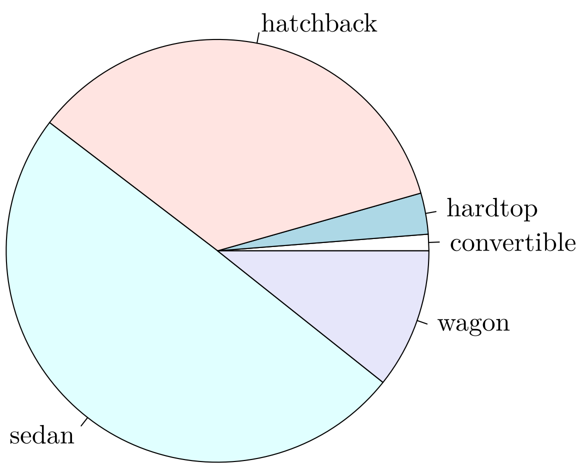
R programlama dili çok esnek bir grafik kütüphanesi ile birlikte gelmektedir. R'ın grafik kütüphanesini kullanabilmek için gerekli komutları inceleyelim.

### 4.4.1 Pasta Grafikleri

#### 4.4.1.1 Temel Pasta Grafiği

Pasta grafikleri verinin yüzde olarak genel dağılımı hakkında bilgi verirler. R'da oluşturmaları için *pie* komutu yeterli olacaktır. Örneğin, araçların gövde türlerine göre dağılımını göstermek için aşağıdaki komutlar kullanılabilir.

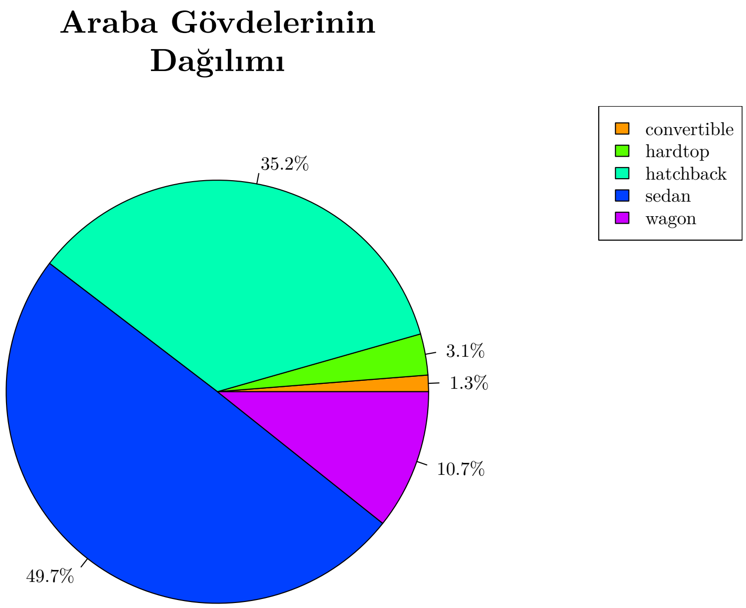
```
t <- table(analiz_verisi$GovdeTuru)

etiketler <- names(t)

pie(t, etiketler)
```




#### 4.4.1.2 Grafikte Görünecek Değerleri Belirlemek

Güzerinde yüzdeleri gösterip renklendirmeleri yan tarafta gösterelim.

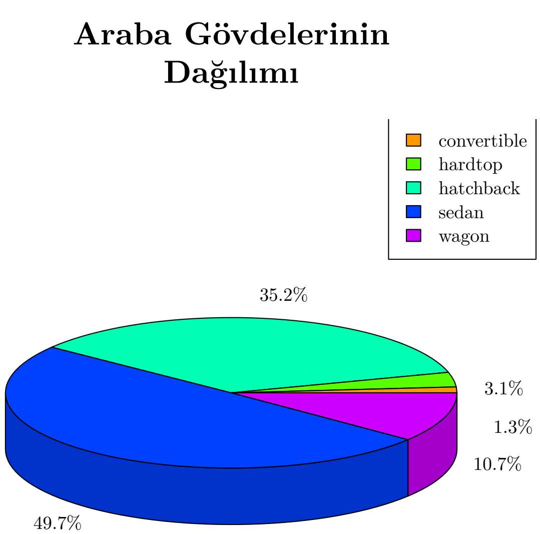
```
# Yüzdeleri Hesaplayalım
etiketler <- round(100*t/sum(t),1)

# Değerlerin sonuna % sembolü ekleyelim
etiketler <- paste(etiketler, "%", sep = "")

# Renkleri Belirleyelim
renk <- rainbow(length(etiketler), start = 0.1, end = 0.8)

pie(t, main="Araba Gövdelerinin\nDağılımı", labels=etiketler,
    cex=0.7, col = renk)

legend("topright", cex=0.7, names(t), fill = renk)
```




#### 4.4.1.3 3-Boyutlu Pasta Grafiği

R ile 3 Boyutlu pasta grafiği çizebilmek için *plotrix* kütüphanesini kullanmalısınız. Bu kütüphanenin içerisinde bulunan *pie3D* fonksiyonu 3 boyutlu pasta grafikleri çizdirebilmeniz için yeterli olacaktır. Aşağıdaki kod bloğunu inceleyin.

```
library(plotrix)

pie3D(t, labels=etiketler, main="Araba Gövdelerinin\nDağılımı",
      col = renk, labelcex=0.7, radius = 0.8)

legend("topright", cex=0.7, names(t), fill = renk)
```



#### 4.4.1.4 Grafikleri PNG Olarak Kaydetmek

R ile oluşturacağınız grafikleri png olarak dosyalara kaydedebilirsiniz. Bunun için bir PNG dosya aygıtı oluşturup, grafiğinizi çizdirmeli ve işlem sonunda bu aygıtı kapatmalısınız. Aşağıdaki örneği inceleyin.

```
# PNG Dosyasını Oluşturalım...
png(filename = "Grafik1.png")

pie3D(t, labels=etiketler,
      main="Araba Gövdelerinin\nDağılımı",
      col = renk, labelcex=0.7, radius = 0.8)

legend("topright", cex=0.7, names(t), fill = renk)

# Dosyayı Kaydedelim.
dev.off()
```

#### 4.4.1.5 Verideki Bütün Nominal Değerlerin 3B Pasta Grafiklerinin Oluşturulması

Bu işlem için basit bir for döngüsü yeterli olacaktır.

```
sutun_adlari <- names(nominal_veri)
for(i in 1:ncol(nominal_veri)) {
  t <- table(nominal_veri[,i])
  etiketler <- names(t)

  renk <- rainbow(length(etiketler), start = 0.1, end = 0.8)

  png(filename = paste("PastaGrafik_",sutun_adlari[i],".png", sep = ""))

  pie3D(t, labels = etiketler,
        main = paste(sutun_adlari[i],"\nDağılımı", sep = ""),
        col = renk, labelcex = 0.7, radius = 0.8)

  legend("topright", cex = 0.7, names(t), fill = renk)

  dev.off()
}
```

BÖLÜM

# 5 İstatistiksel Analize Hazırlık

## 5.1 Hipotez Testleri

**Tanım 1 (Hipotez)** *Bir konu ile ilgili varsayılan, ileri sürülen, doğruluğu kabul ediliyor olsa da ispat edilmemiş bilgiye* ***hipotez*** *denir.*

**Tanım 2 (İstatistiksel Hipotez)** *Karar verebilmek için öne sürülen, varsayılan ve geçerli olup olmadığına, belirlenen olasılıklarla karar verilecek sayısal bilgiye* ***İstatistiksel Hipotez*** *denir.*

İstatistiksel hipotez testleri ile yapılan için genel anlamı, anakütle ile ilgili doğruluğu kesin olmayan bir sayısal bilginin veya varsayımın, alınan örnekten elde edilen, tahmin edilen bilgi ile doğruluğunu incelemektir. Örnekten elde edilen bilgi ile anakütle hakkındaki varsayım arasında fark yoksa, test edilecek bir şey yoktur. Ancak, örnekten elde edilen bilgi ile anakütle hakkındaki varsayım arasında fark varsa, bu farkın anlamlı olup olmadığını belirleyebilmek için test işlemine gerek olacaktır. Yapılacak hipotez testi ile amaç, örnekten elde edilen bilgi ile anakütle hakkındaki varsayım arasındaki farkın, örnekten veya örneklemeden kaynaklanıp kaynaklanmadığını belirlemektir. Örneğin, bir sınıfın *İstatistiksel Yazılımlar* dersinden ortalamasının 80 olduğu iddia ediliyorsa ve bu sınıftan alınan 25 öğrencinin ortalaması da 78 olarak hesaplanmışsa, aradaki 2 birimlik farkın örneklemeden kaynaklanıp kaynaklanmadığını bulmak için hipotez testleri yapılmalıdır. Test sonucunda, aradaki farkın örneklemeden kaynaklanıp kaynaklanmadığına yani istatistiksel olarak anlamlı olup olmadığına karar verilir. Bu karar, $\alpha$ olasılık seviyesi ile verilir ve bu olasılığa *hata payı* veya *anlam seviyesi* adı verilir. Bu olasılığın, tümleyeni ise $(1-\alpha)$ *güven olasılığı* olarak adlandırılır.

## 5.2 Hipotez Testinin Aşamaları

Hipotez Testleri ister parametrik ister parametrik olmayan test olsun, izlenecek adımlar birbirlerine çok benzerdir ve şöyle sıralanabilir.

- *Hipotezlerin Oluşturulması:* Bu aşamada iki hipotez oluşturulur ve kullanılacak testin parametrik mi, yoksa parametrik olmayan test mi olacağına karar verilir.

- *Hipotezlerden Birinin Kabul Kararı için Tablo Değerlerinin Belirlenmesi:* Bu aşamada testin çift yada tek taraflı oluşu, parametrik veya parametrik olmayan test oluşu dikkate alınarak belirlenen hata payı ile tablo değerleri bulunur. Yazılımlardan elde edilen **p** değeri bu amaçla kullanılır.
- *Test İstatistiğinin Hesaplanması:* Uygun test türü seçildiğinde, eldeki verinin özellikleri de göz önüne alınarak test istatistiği hesaplanır. Genellikle paket programlar bu değeri otomatik olarak hesaplarlar.
- *Kabul Kararı Verilmesi:* Hesaplanan test istatistiği ile tablo değerleri karşılaştırılarak, kurulan hipotezlerden biri kabul diğeri reddedilir.

Hipotez testinin ilk aşaması iki adet hipotezin kurulması ile başlar, bunlar $H_0$ (yokluk hipotezi, temel hipotez, sıfır hipotezi) ve $H_1$ (alternatif hipotez, karşıt hipotez) hipotezleridir. Burada önemli olan $H_1$ hipotezinin yönüdür. $H_1$ hipotezi tek yanlı veya çift yanlı olarak kurulabilir. Çift yanlı $H_1$ eşit olmamayı, tek yanlı hipotez büyük veya küçük olmayı ifade eder. Yukarıdaki örnek ile ilgili kurulan örnek hipotezler aşağıdaki gibi olabilir.

- İki yanlı hipotez

$$H_0 : \mu = 80$$
$$H_1 : \mu \neq 80$$

- Tek yanlı hipotez

$$H_0 : \mu = 80$$
$$H_1 : \mu < 80$$

Hipotez testleri sonucunda 4 adet durum söz konusudur bunlar aşağıdaki tabloda belirtilmiştir.

| | $H_0$ Kabul | $H_1$ Kabul |
|---|---|---|
| $H_0$ Doğru | Doğru Karar | I. Tip Hata veya $\alpha$ hatası |
| $H_1$ Doğru | II. Tip Hata veya $\beta$ hatası | Doğru Karar |

## 5.3 Normallik Testleri

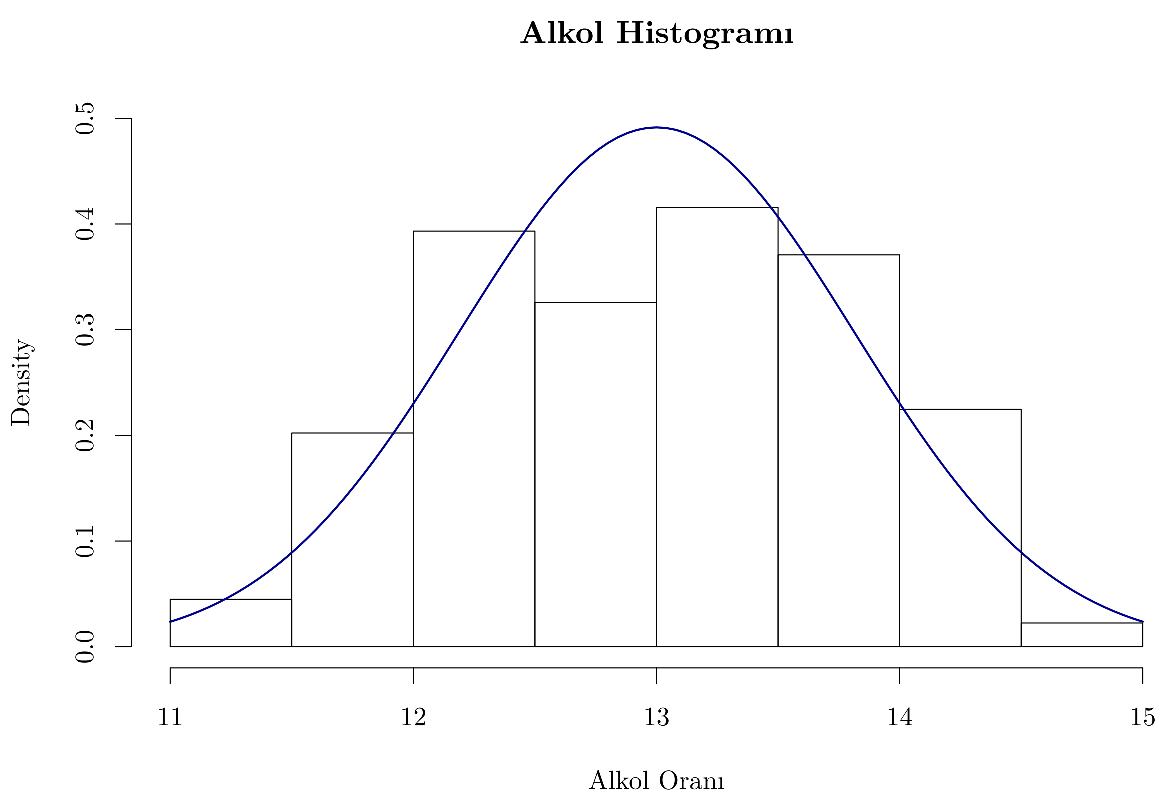
Hipotez testlerinin en önemli aşamalarından biri de kullanılacak teste karar verilmesi aşamasıdır. Burada iki alternatifiniz bulunmaktadır; Parametrik Testler ve Parametrik Olmayan Testler. Parametrik testlerin uygulanabilmesi için örnekleme dağılımının *Normal Dağılım, t-Dağılımı* olması veya *Normal Dağılım* olarak kabul edilebilmesi gereklidir. Bir örneklemin, normal olup olmadığı veya normal kabul edilip edilemeyeceğine, normal dağılımın özelliklerine bakılarak veya istatistiksel olarak bakılarak karar verilebilir. Bir örneklemin normal dağılıma sahip olup olmadığına karar verebilmek için aşağıdaki özelliklerine bakabilirsiniz.

- Verinin dağılımı çan eğrisine benzer

- Aritmetik ortalama, mod ve medyan birbirine eşittir
- Normal dağılımın belirli oranları, standart aralıklarda yer alır
- Basıklık ($\alpha_3$) ve çarpıklık ($\alpha_4$) ölçüleri sırasıyla 0 ve 3 değerini alır

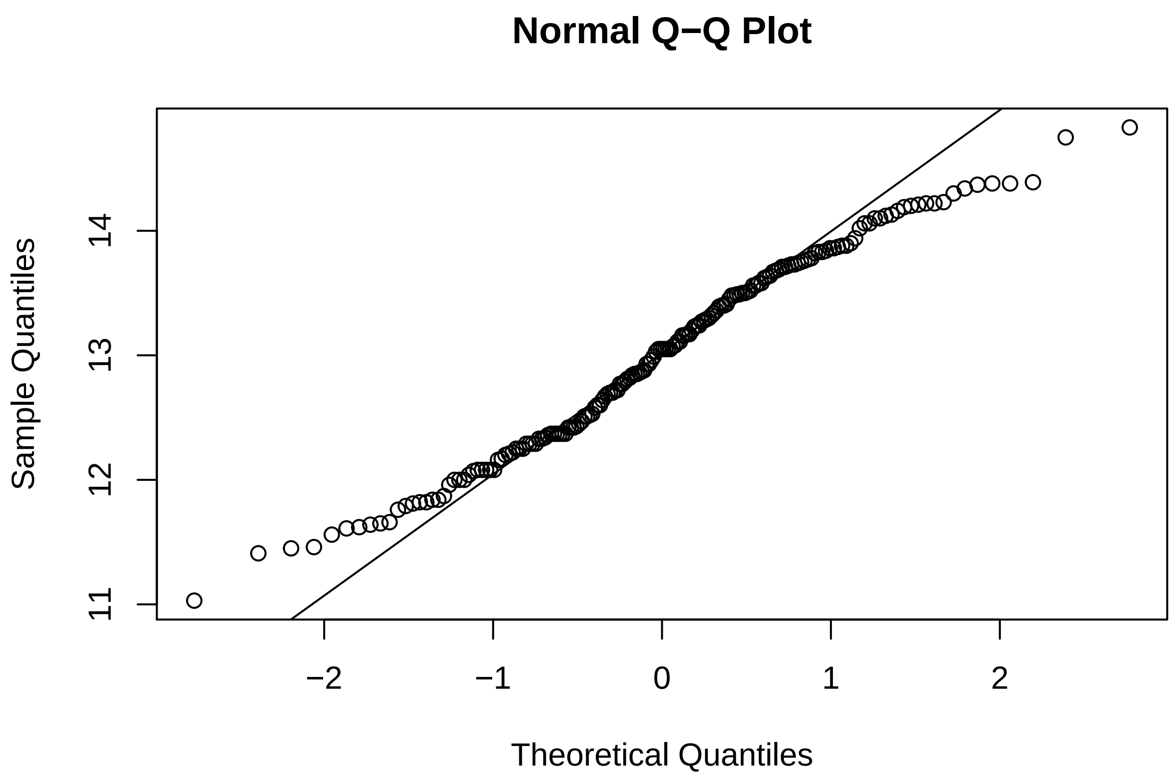
Bu özellikleri bir örnekte deneyelim. Önce örnek histogramına bakalım.

```
veri = read.csv2('wine.csv', row.names = NULL)
sayisal.sutunlar = sapply(veri, class) != 'factor'
sayisal.veri = veri[,sayisal.sutunlar]

v = sayisal.veri$Alcohol

ort = mean(v)
ssapma = sqrt(var(v))

hist(v, probability = TRUE, xlab = "Alkol Oranı",
     main = "Alkol Histogramı", ylim = c(0,0.5))
curve(dnorm(x, mean = ort, sd = ssapma),
      col = "darkblue", lwd=2, add = TRUE)
```




Şimdide QQ grafiğini inceleyelim.

```
qqnorm(v)
qqline(v)
```




Aritmetik ortalama, mod ve medyanın durumlarını inceleyelim.

```
ort<-mean(v)
medyan<-median(v)
t <- table(v)
mod<-t[which.max(t)]

cat("Ortalama=",ort,"\nMedyan=",medyan,"\nMod=",names(mod))
```

```
## Ortalama= 13.00062
## Medyan= 13.05
## Mod= 12.37
```

Basıklık ve Çarpıklık ölçülerini inceleyelim. (Moments paketi bilgisayarınızda olmayabilir, yüklemeniz gerekebilir)

```
library(moments)

carpiklik = skewness(v)
basiklik = kurtosis(v)

cat("Çarpıklık = ",carpiklik,"\nBasıklık = ",basiklik)
```

```
## Çarpıklık =  -0.05148233
## Basıklık =  -0.8524996
```

Bu değerlere göre, verinizin normal olduğu kanısına varabilirsiniz, fakat bu işlemin daha doğrusu verinizin normalliğini istatistiksel olarak test etmenizdir. Bunun için iki adet istatistiksel test kullanılır. Bunlar, Kolmogorov-Simirnov testi ve Shapiro-Wilk Testidir.

### 5.3.1 Kolmogorov-Smirnov Testi

Bu test, parametrik olmayan bir testtir ve temeli incelenen örneklem dağılımının birikimli olasılık yoğunluk fonksiyonunun, herhangi bir sürekli dağılımın olasılık yoğunluk foksiyonu ile karşılaştırılmasına dayanır. Bu testin yokluk ve alternatif hipotezleri aşağıdaki gibidir.

| | |
|---|---|
| $H_0$ : | Dağılım İstenilen Sürekli Dağılımdır |
| $H_1$ : | Dağılım İstenilen Sürekli Dağılım Değildir |

Kullanılan test istatistiği ise;

$$D = max_{1\leq i\leq N}\left|F(Y_i) - \frac{i}{N}\right|$$

Burada, F sürekli dağılımın olasılık yoğunluk fonksiyonu, $Y_i$ i. gözlem değeri, N eleman sayısıdır. Burada, teorik dağılımın parametreleri örnekleme dağılımından hesaplanarak elde edilir. Kolmogorov-Simirnov testi genel amaçlı bir test olmasına rağmen burada ilgilenilen dağılımın normal dağılım olup olmadığını test etmek amacı ile kullanılacaktır. Dolayısıyla, burada F fonksiyonu normal dağılımın olasılık yoğunluk fonksiyonu olacaktır. Şimdi, verinin normal dağılıma sahip olup olmadığına bakalım.

```
test <- ks.test(v, 'pnorm')
if(test$p.value<0.05) {
  cat('Dağılım Normal Dağılım Değildir (p=',test$p.value,")")
} else {
  cat('Dağılım Normal Dağılımdır (p=',test$p.value,")")
}
```

```
## Dağılım Normal Dağılım Değildir (p= 0 )
```

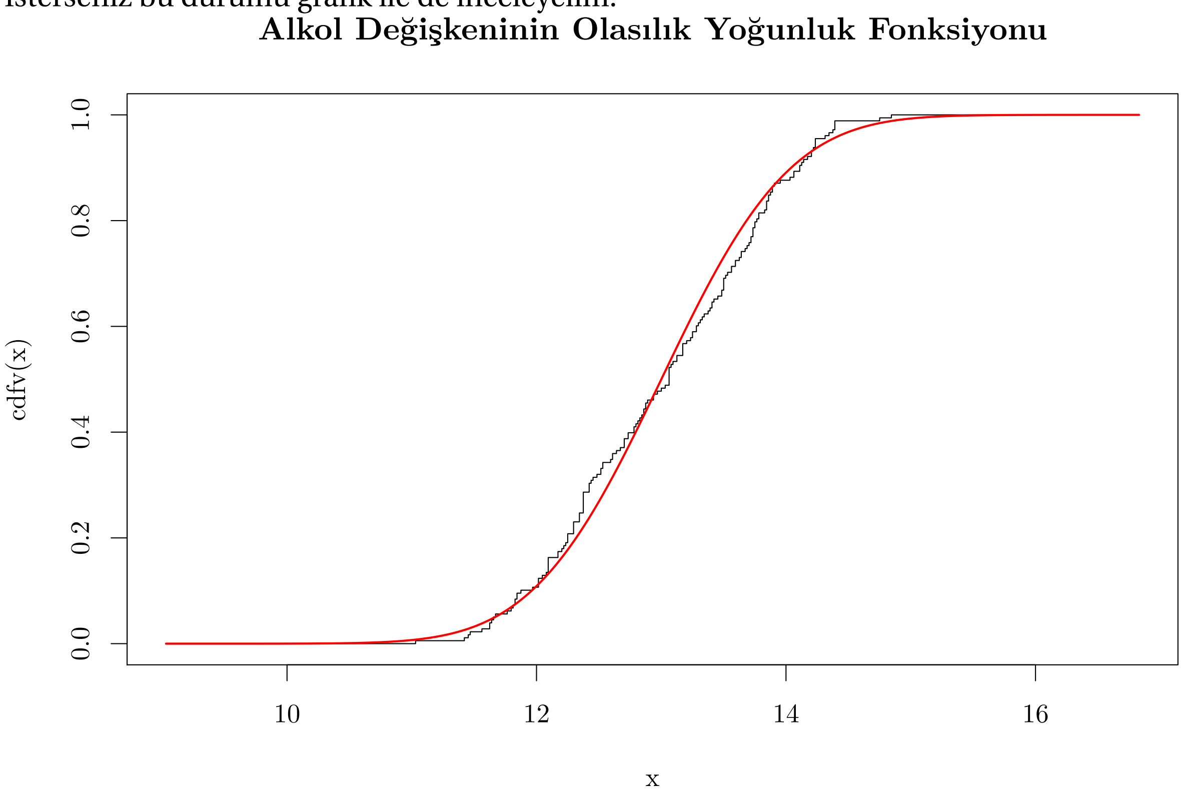
Buradaki verinin standart normal dağılıma sahip olduğu test edilmektedir, yani $v \sim N(0,1)$. Bu sebeple, Kolmogorov-Smirnov testinin teorik dağılımlarının parametre tahminlerini kullanan hali olan Lilliefors testinin kullanılması önerilmektedir (Eğer teorik dağılımın parametreleri biliniyorsa, Kolmogorov-Smirnov testinin kullanılmasında bir sorun bulunmamaktadır). Burada amaç, değişkenin normalliği olduğu için, Lilliefors testinin kullanılması daha doğru olacaktır. Aşağıdaki komutlar ile Lilliefors testi gerçekleştirilebilir.

```
library(nortest)
test <- lillie.test(v)
if(test$p.value<0.05) {
  cat('Dağılım Normal Dağılım Değildir (p=',
      test$p.value,")\n")
```

```
} else {
  cat('Dağılım Normal Dağılımdır (p=',
      test$p.value,")\n")
}
```

```
## Dağılım Normal Dağılım Değildir (p= 0.04404785 )
```

İsterseniz bu durumu grafik ile de inceleyelim.




Bu yaptıklarımızı, bütün değişkenlere uygulayalım.

```
degisken.isimleri = names(sayisal.veri)
for(i in 1:ncol(sayisal.veri)) {
  degisken = sayisal.veri[,i]
  degisken.adi = degisken.isimleri[i]

  test <- lillie.test(degisken)
  if(test$p.value<0.05) {
    cat(degisken.adi,' dağılımı Normal Dağılım Değildir (p=',
        test$p.value,")\n")
  } else {
    cat(degisken.adi,' dağılımı Normal Dağılımdır (p=',
        test$p.value,")\n")
  }
}
```

```
## Class  dağılımı Normal Dağılım Değildir (p= 3.47753e-23 )
## Alcohol  dağılımı Normal Dağılım Değildir (p= 0.04404785 )
## Malic.Acid  dağılımı Normal Dağılım Değildir (p= 1.555976e-17 )
## Ash  dağılımı Normal Dağılımdır (p= 0.1446116 )
## Aash  dağılımı Normal Dağılımdır (p= 0.07713065 )
## Magnesium  dağılımı Normal Dağılım Değildir (p= 0.001403261 )
## TotPhenol  dağılımı Normal Dağılım Değildir (p= 0.02491662 )
## Flavanoids  dağılımı Normal Dağılım Değildir (p= 0.003005415 )
## Nphenols  dağılımı Normal Dağılım Değildir (p= 5.543312e-06 )
## Proant  dağılımı Normal Dağılımdır (p= 0.174296 )
## Color  dağılımı Normal Dağılım Değildir (p= 0.001162927 )
## Hue  dağılımı Normal Dağılımdır (p= 0.08829705 )
## OD280  dağılımı Normal Dağılım Değildir (p= 3.304797e-06 )
## Proline  dağılımı Normal Dağılım Değildir (p= 1.205789e-07 )
```

### 5.3.2 Shapiro-Wilk Testi

Bu test de verinin normalliği için kullanılan testlerden biridir, hatta yapılan çalışmalar ışığında bu testin gücünün Kolmogorov-Smirnov testinin gücünden daha fazla olduğu bulunmuştur. Uygulamada genellikle iki testin sonucuna da bakılır. İki test sonucunun farklı olması durumunda, Shapiro-Wilk testinin sonucunun kullanılması tavsiye edilir. Bu testin hipotezleri de Kolmogorov-Smirnov testi ile aynıdır. Fakat hesaplanan istatistik aşağıdaki gibidir.

$$W = \frac{\left(\sum_{i=1}^{m} a_i \left(x_{n+1-i} - x_i\right)\right)^2}{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)^2}$$

Burada m eleman sayısı çift ise $\frac{n}{2}$, tek ise $\frac{(n-1)}{2}$; $a_i$ ise teorik dağılımdan elde edilen sıra istatistikleridir. Bu testin R ile uygulanması aşağıdaki gibi olur.

```
test<-shapiro.test(v)
if(test$p.value<0.05) {
  cat('Dağılım Normal Dağılım Değildir (p=',test$p.value,")\n")
} else {
  cat('Dağılım Normal Dağılımdır (p=',test$p.value,")\n")
}
```

```
## Dağılım Normal Dağılım Değildir (p= 0.02004798 )
```

Bu yaptıklarımızı, bütün değişkenlere uygulayalım.

```
degisken.isimleri = names(sayisal.veri)
for(i in 1:ncol(sayisal.veri)) {
  degisken = sayisal.veri[,i]
  degisken.adi = degisken.isimleri[i]
```

```
  test <- shapiro.test(degisken)
  if(test$p.value<0.05) {
    cat(degisken.adi,' dağılımı Normal Dağılım Değildir (p=',
        test$p.value,")\n")
  } else {
    cat(degisken.adi,' dağılımı Normal Dağılımdır (p=',
        test$p.value,")\n")
  }
}
```

```
## Class  dağılımı Normal Dağılım Değildir (p= 3.52658e-14 )
## Alcohol  dağılımı Normal Dağılım Değildir (p= 0.02004798 )
## Malic.Acid  dağılımı Normal Dağılım Değildir (p= 2.945801e-10 )
## Ash  dağılımı Normal Dağılım Değildir (p= 0.03868278 )
## Aash  dağılımı Normal Dağılımdır (p= 0.2638688 )
## Magnesium  dağılımı Normal Dağılım Değildir (p= 6.345694e-07 )
## TotPhenol  dağılımı Normal Dağılım Değildir (p= 0.004395305 )
## Flavanoids  dağılımı Normal Dağılım Değildir (p= 1.678853e-05 )
## Nphenols  dağılımı Normal Dağılım Değildir (p= 0.00010551 )
## Proant  dağılımı Normal Dağılım Değildir (p= 0.01445402 )
## Color  dağılımı Normal Dağılım Değildir (p= 9.22921e-07 )
## Hue  dağılımı Normal Dağılım Değildir (p= 0.0174255 )
## OD280  dağılımı Normal Dağılım Değildir (p= 2.316064e-06 )
## Proline  dağılımı Normal Dağılım Değildir (p= 1.74126e-07 )
```

### 5.3.3 Veri Düzeltmesi

Eğer verinizdeki bazı değişkenler normal dağılıma sahip değilse, bazu dönüşümler yardımı ile bu değişkenleri normalleştirmeyi deneyebilirsiniz. Buradaki kural, sağa çarpık değişkenlerin karekök veya logaritma fonksiyonları ile, sola çarpık değişkenlerin üstel ($e^{-x}$) veya $\frac{1}{x}$ fonksiyonu ile normal dağılıma dönüşebilmeleridir. Şimdi, bu ifadelerin R ile uygulamasına bakalım.

```
degisken.isimleri = names(sayisal.veri)
for(i in 1:ncol(sayisal.veri)) {
  degisken = sayisal.veri[,i]
  degisken.adi = degisken.isimleri[i]

  degisken = degisken[!is.na(degisken)]

  test <- shapiro.test(degisken)
  if(test$p.value<0.05) {

    log.d = log(degisken)
    log.test = shapiro.test(log.d)
```

```
    if(!is.nan(log.test$p.value)) {
      if(log.test$p.value>0.05) {
        cat(degisken.adi," log dönüşümü ile normal dağılıma sahiptir (p=",
            log.test$p.value,")\n")
        next
      }
    }

    sqrt.d = sqrt(degisken)
    sqrt.test = shapiro.test(sqrt.d)
    if(!is.nan(sqrt.test$p.value)) {
      if(sqrt.test$p.value>0.05) {
        cat(degisken.adi," karekök dönüşümü ile normal dağılıma sahiptir(p="
            ,sqrt.test$p.value,")\n")
        next
      }
    }

    ustel.d = exp(-degisken)
    ustel.test = shapiro.test(ustel.d)
    if(!is.nan(ustel.test$p.value)) {
      if(ustel.test$p.value>0.05) {
        cat(degisken.adi," üstel dönüşümü ile normal dağılıma sahiptir (p=",
            ustel.test$p.value,")\n")
        next
      }
    }

    bx.d = 1/degisken
    bx.test = shapiro.test(bx.d)
    if(!is.nan(bx.test$p.value)) {
      if(bx.test$p.value>0.05) {
        cat(degisken.adi," 1/x dönüşümü ile normal dağılıma sahiptir (p=",
            bx.test$p.value,")\n")
        next
      }
    }

    cat(degisken.adi,' dağılımı Normal Dağılım Değildir (p=',
        test$p.value,")\n")
  } else {
    cat(degisken.adi,' dağılımı Normal Dağılımdır (p=',
        test$p.value,")\n")
  }
```

```
}
```

```
## Class  dağılımı Normal Dağılım Değildir (p= 3.52658e-14 )
## Alcohol  dağılımı Normal Dağılım Değildir (p= 0.02004798 )
## Malic.Acid  dağılımı Normal Dağılım Değildir (p= 2.945801e-10 )
## Ash  dağılımı Normal Dağılım Değildir (p= 0.03868278 )
## Aash  dağılımı Normal Dağılımdır (p= 0.2638688 )
## Magnesium  1/x dönüşümü ile normal dağılıma sahiptir (p= 0.3556347 )
## TotPhenol  dağılımı Normal Dağılım Değildir (p= 0.004395305 )
## Flavanoids  dağılımı Normal Dağılım Değildir (p= 1.678853e-05 )
## Nphenols  dağılımı Normal Dağılım Değildir (p= 0.00010551 )
## Proant  karekök dönüşümü ile normal dağılıma sahiptir(p= 0.6160586 )
## Color  log dönüşümü ile normal dağılıma sahiptir (p= 0.2456765 )
## Hue  dağılımı Normal Dağılım Değildir (p= 0.0174255 )
## OD280  dağılımı Normal Dağılım Değildir (p= 2.316064e-06 )
## Proline  dağılımı Normal Dağılım Değildir (p= 1.74126e-07 )
```

## 5.4 Dağılım Testleri

### 5.4.1 Rastsallık Testi

Test edeceğiniz hipotez ister parametrik ister parametrik olmayan bir test olsun, verinizin rastsallığından emin olmalısınız. Bu durum için kullanabileceğiniz parametrik olmayan bir test mevcuttur. Wald-Wolfowitz Runs Testi bir verinin rastsallığını istatistiksel olarak belirlemek için kullanılabilir. Testi kullanabilmeniz için ilk olarak verinizi belirli bir sırada sıralamanız gerekmektedir. Örneğin, zamana göre bir anket yaptığınızda, elde ettiğiniz verilerin zaman alanına göre rastsal olması beklenmektedir. Bu sebeple, verinizin rastsallığını ölçmeden önce rastsallık için belirleyeceğiniz bir ölçüte göre sıralama yapıp, rastsallığı bu şekilde ölçmeniz gerekmektedir. Bu özelliği ile sadece kısıtlı miktarda verilere uygulanması söz konusudur.

Wald-Wolfowitz testi için test istatistiği aşağıdaki şekilde hesaplanır. Sıralı veri, bir değere göre (genellikle medyan veya ortalama) değerlendirilir ve bu değerden yüksek elemanlar +, düşük elemanlar - olarak işaretlenir. Aynı işaretlerin kümesine Run adı verilir. Örneğin, 22 elemanlık bir dizideki elemanların değerlendirmesi “++++- - -+++- -++++++- - - -” şeklindeyse, bu veride 6 adet run bulunmaktadır ve bunlardan üçü + üçü de - run dır. Testin varsayımı, verinin runlarının dağılımının aşağıdaki parametrelerle normal dağılıma sahip olmasıdır.

$$
\begin{aligned}
\mu &= \frac{2N_-N_+}{N} + 1 \\
\sigma^2 &= \frac{2N_-N_+\,(2N_-N_+ - N)}{N^2(N-1)} = \frac{\left(\mu-1\right)\left(\mu-2\right)}{N-1}
\end{aligned}
$$

Bu testin hipotezleri aşağıdaki gibidir.

| | |
|---|---|
| $H_0$ : | Değişken, mevcut sırasıyla rastsaldır. |
| $H_1$ : | Değişken, mevcut sırasıyla rastsal değildir. |

Runs testinin R ile gerçekleştirilebilmesi için, verinizin önce sıralanması ve randtest kütüphanesininde yüklenmesi gerekmektedir.

```
library(randtests)
# Önce Proline'ın sıra bilgisi oluşturulur
proline.sira = order(sayisal.veri$Proline)
# Sonra bu sıra bilgisi Alchol'e uygulanır
degisken = sayisal.veri$Alcohol[proline.sira]
# Ve test gerçekleştirilir
test = runs.test(degisken)

if(test$p.value<0.05) {
  cat("Alkol proline'a göre rastsal değildir (p=",test$p.value,")\n")
} else {
  cat("Alkol proline'a göre rastsaldır (p=",test$p.value,")\n")
}
```

```
## Alkol proline'a göre rastsal değildir (p= 2.135012e-06 )
```

Bütün sayısal değişkenlerin birbirlerine göre rastsallığı ise aşağıdaki gibi olacaktır.

| | Clas | Alco | Mali | Ash | Aash | Magn | TotP | Flav | Nphe | Proa | Colo | Hue | OD28 | Prol |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Class | | - | - | + | - | - | - | - | - | - | - | - | - | - |
| Alcohol | + | | + | + | + | + | + | + | + | - | - | + | + | - |
| Malic.Acid | - | - | | + | - | + | - | - | + | - | + | - | + | + |
| Ash | + | + | + | | + | + | - | + | + | + | - | + | + | + |
| Aash | - | - | + | + | | + | - | - | - | - | - | - | - | - |
| Magnesium | + | - | + | + | + | | + | + | + | + | - | + | + | - |
| TotPhenol | - | - | + | + | - | + | | - | - | - | - | - | - | - |
| Flavanoids | - | - | + | + | + | + | - | | - | - | - | - | - | - |
| Nphenols | - | - | - | + | - | + | - | - | | + | - | - | - | - |
| Proant | - | + | - | + | + | + | - | - | - | | - | - | - | - |
| Color | - | - | + | + | + | + | - | + | + | + | | - | + | - |
| Hue | - | + | - | + | - | + | - | - | + | + | - | | - | - |
| OD280 | - | + | + | + | + | + | - | - | - | - | - | - | | + |
| Proline | - | - | + | + | - | - | - | - | - | - | - | - | - | |

### 5.4.2 Varyans Homojenliği Testi

Araştırmanızda iki farklı örneklemin karşılaştırılması isteniyorsa, bu örneklerin dağılımlarının sadece normal dağılım olması yeterli değildir. Ayrıca, değişkenlerin varyanslarının Homojen (eşit) olmasıda beklenmektedir. Örneklerin varyanslarını test edebilmek için, Levene testini kullanmanız gerekmektedir. Test istatistiği, aşağıdaki gibidir.

$$W = \frac{(N-k)}{(k-1)} \frac{\sum_{i=1}^{k} N_i (Z_{i\cdot} - Z_{\cdot\cdot})^2}{\sum_{i=1}^{k} \sum_{j=1}^{N_i} \left(Z_{ij} - Z_{i\cdot}\right)}$$

Bu testin, hipotezleri aşağıdaki gibidir.

| | |
|---|---|
| $H_0$ : | Değişkenlerin varyansları homojendir. |
| $H_1$ : | Değişkenlerin varyansları homojen değildir. |

Bu testi, R ile kullanabilmeniz için iki ayrı verinin birleştirip tek veri haline getirmeniz ve bu verideki elemanların hangi örnekleme ait olduğunu gösteren bir factor değişkeni oluşturmanız gerekmektedir. Aşağıdaki örneği inceleyin.

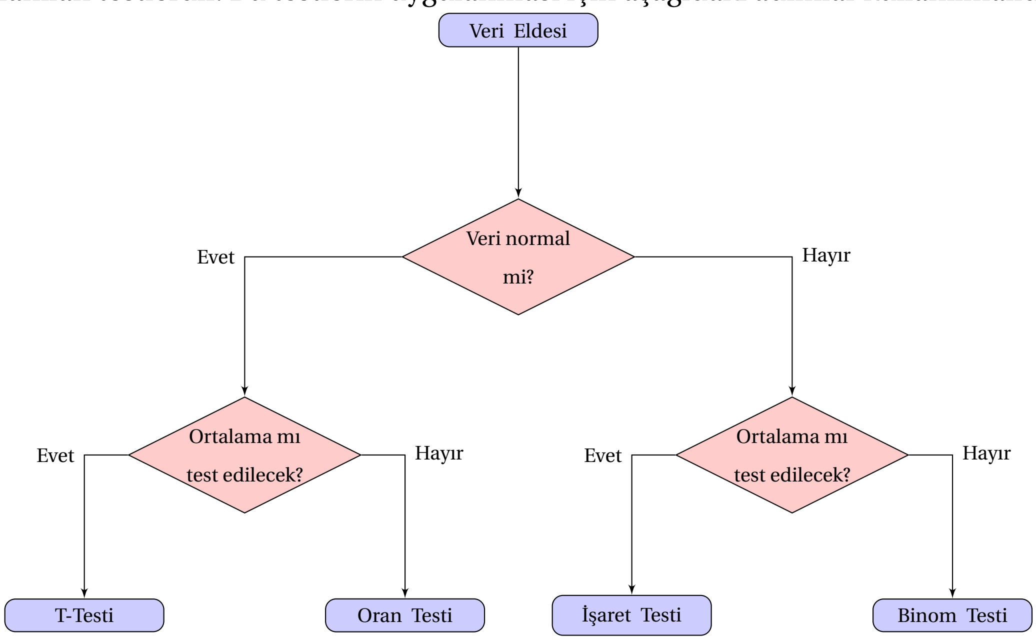
```
library(car)

degisken1 = sayisal.veri$Alcohol
degisken2 = sayisal.veri$Malic.Acid

birlestirilmis.degisken = c(degisken1, degisken2)

degisken1.uzunluk = length(degisken1)
degisken2.uzunluk = length(degisken2)

degisken1.belirtec = rep(1, degisken1.uzunluk)
degisken2.belirtec = rep(2, degisken2.uzunluk)

degisken.bileske.belirtec = c(degisken1.belirtec, degisken2.belirtec)

test = leveneTest(birlestirilmis.degisken, degisken.bileske.belirtec)
p.degeri = test$`Pr(>F)`[1]

if(p.degeri<0.05) {
  cat("Alkol ve Malik asitin varyansları homojen değildir (p=",
      p.degeri,").\n")
} else {
  cat("Alkol ve Malik asitin varyansları homojendir (p=",
      p.degeri,").\n")
}
```

```
## Alkol ve Malik asitin varyansları homojen değildir (p= 0.03568517 ).
```

BÖLÜM

# 6 Tek Örnek Testleri

Anakütle ortalaması veya oranı için ortaya atılan hipotezlerin gerçekliğinin incelenmesi için kullanılan testlerdir. Bu testlerin uygulanması için aşağıdaki adımlar kullanılmalıdır.




## 6.1 Parametrik Testler

### 6.1.1 Ortalamaların Testi

Bu test ile anakütle ortalaması $\mu$ ile ilgili bilinen veya ileri sürülen $\mu_0$ ile örneklemden elde edilen $\overline{X}$ arasındaki farkın istatistiksel olarak anlamlı olup olmadığı değerlendirilir.

Ortalaması $\mu$, varyansı $\sigma^2$ olduğu iddia edilen bir anakütleden $n$ elemanlı bir örneklem çekilmiş olsun. Çekilen örneklemin, ortalaması $\overline{X}$ ve varyansı da $S^2$ olarak bulunmuş olsun. Bu durumda aşağıdaki hipotezler kurulabilir.

| | |
|---|---|
| $H_0:$ | $\mu = \mu_0$ |
| $H_1:$ | $\mu \neq \mu_0$ (Çift Yanlı) |
| $H_1:$ | $\mu < \mu_0$ (Tek Yanlı) |
| $H_1:$ | $\mu > \mu_0$ (Tek Yanlı) |

Burara Anakütlenin varyansı bilinmediği için t-Dağılımı kullanılacaktır. Test istatistiği aşağıdaki gibi hesaplanır.

$$t = \frac{\overline{x} - \mu_0}{\sigma_{\overline{x}}}$$

Eğer anakütle ortalaması biliniyorsa - ki bu durumda normal dağılım testi de kullanabilirsiniz - $\sigma_{\overline{x}}$ aşağıdaki gibi hesaplanır.

$$\sigma_{\overline{x}} = \frac{\sigma}{\sqrt{n}}$$

Ancak anakütle standart sapması bilinemeyeceğinden, $\sigma$ nın S tamin edicisi kullanılır.

$$S^2 = \frac{\sum\left(c - \overline{x}\right)^2}{n-1}$$
$$S^2 = \frac{n\tilde{\sigma}^2}{n-1}$$

Burada elde edilecek olan **p** değeri $\alpha$ değerinden küçükse $H_0$ hipotezi, aksi durumda $H_1$ hipotezi reddedilir. Şimdi Bu testi R ile uygulayalım. Örnek olarak parkinson verisindeki Fo alanının ortalamasının 150 olup olmadığını test edelim.

```
veri = read.csv2('parkinsons.csv')
test = t.test(veri$Fo, alternative = "two.sided", mu = 150)
if(test$p.value<0.05) {
  cat("Fo alanının ortalaması 150 değildir p=",test$p.value)
} else {
  cat("Fo alanının ortalaması 150 dir p=",test$p.value)
}
```

```
## Fo alanının ortalaması 150 dir p= 0.1552838
```

Şimdi de Fo alanını ortalamasının 160'tan büyük veya küçük olması hipotezlerini test edelim.

```
test = t.test(veri$Fo, alternative = "greater", mu = 160)
if(test$p.value<0.05) {
  cat("Fo alanının ortalaması 160'tan büyüktür p=",test$p.value)
} else {
  cat("Fo alanının ortalaması 160'tan büyük değildir dir p=",test$p.value)
}
```

```
## Fo alanının ortalaması 160'tan büyük değildir dir p= 0.9735198
```

```
test = t.test(veri$Fo, alternative = "less", mu = 160)
if(test$p.value<0.05) {
  cat("Fo alanının ortalaması 160'tan küçüktür p=",test$p.value)
} else {
  cat("Fo alanının ortalaması 160'tan küçük değildir dir p=",test$p.value)
}
```

```
## Fo alanının ortalaması 160'tan küçüktür p= 0.02648018
```

### 6.1.2 Oranların Testi

Bu test aracılığı ile örnekleminizi kullanarak ana kütlede yer alan bilgilerin birbirlerine olan oranlarını test edebilirsiniz. Örneğin, bir doktorun trafik kazası geçirmiş hastaların hayatta halma oranının %75 olduğunu iddiasının test etmek için kullanabilirsiniz. Bu testin varsayımı, uğraştığınız veri kümesinin eleman sayısının yüksek olmasıdır (genel kanı bildiğiniz gibi N>30 olmasıdır). Bu durumda binom dağılımı normal dağılıma yaklaşacağından aşağıdaki test istatistiği kullanılabilir.

$$z = \frac{p_0 - p_e}{\sqrt{\frac{p_0 q}{n}}}$$

Oran testi için kurulacak yokluk hipotezi aşağıdaki gibidir.

$$H_0 : p_0 = p_e$$

Oran testi R ile aşağıdaki şekilde gerçekleştirilir.

```
kurtulanlar = c(3224)
toplam.hasta = c(3406)

test = prop.test(kurtulanlar, toplam.hasta, p=0.75)

if(test$p.value < 0.05) {
   cat("Trafik kazalarından kurtulanların oranı %75 değildir p=",test$p.value)
} else {
   cat("Trafik kazalarından kurtulanların oranı %75'tir p=", test$p.value)
}
```

```
## Trafik kazalarından kurtulanların oranı %75 değildir p= 1.983575e-154
```

```
test
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  kurtulanlar out of toplam.hasta, null probability 0.75
## X-squared = 700.82, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.75
## 95 percent confidence interval:
##  0.9383400 0.9537641
## sample estimates:
##         p
## 0.9465649
```

```
test = prop.test(kurtulanlar, toplam.hasta, p=0.75, alternative = "g")

if(test$p.value < 0.05) {
   cat("Trafik kazalarından kurtulanların oranı %75'ten büyüktür p=",test$p.value)
} else {
   cat("Trafik kazalarından kurtulanların oranı %75'tir p=", test$p.value)
}
```

```
## Trafik kazalarından kurtulanların oranı %75'ten büyüktür p= 9.917876e-155
```

```
test
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  kurtulanlar out of toplam.hasta, null probability 0.75
## X-squared = 700.82, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is greater than 0.75
## 95 percent confidence interval:
##  0.9397096 1.0000000
## sample estimates:
##         p
## 0.9465649
```

```
test = prop.test(kurtulanlar, toplam.hasta, p=0.75, alternative = "l")

if(test$p.value < 0.05) {
   cat("Trafik kazalarından kurtulanların oranı %75'ten küçüktür p=",test$p.value)
} else {
   cat("Trafik kazalarından kurtulanların oranı %75'tir p=", test$p.value)
}
```

```
## Trafik kazalarından kurtulanların oranı %75'tir p= 1
```

```r
test
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  kurtulanlar out of toplam.hasta, null probability 0.75
## X-squared = 700.82, df = 1, p-value = 1
## alternative hypothesis: true p is less than 0.75
## 95 percent confidence interval:
##  0.0000000 0.9526949
## sample estimates:
##         p
## 0.9465649
```

## 6.2 Parametrik Olmayan Testler

### 6.2.1 Wilcoxon İşaretli Sıra Sayıları Testi

Normal dağılıma sahip olmayan anakütlelerin ortalamaları ile ilgili hipotezlerin test edilmesi amacıyla kullanılır. Burada merkezi dağılım ölçüsü olarak ortalama yerine, dağılımın uç değerlerinden ve çarpıklığından daha az etkilendiği için **medyan** kullanılır. Medyan hesaplandıktan sonra bütün değerlerden medyan çıkartılarak oluşan farkların işaret sayıları belirlenir. Test istatistiği hesaplanırken, bulunan işaretlerin sayıları kullanılır. $H_0$ hipotezinin kabul edilebilmesi için işaret sayılarının p=0.5 olan Binom Dağılımına sahip olması beklenmektedir. Hipotezler, t-Testindeki gibi kurulur.

R ile Wilcoxon testinin gerçekleştirilmesi aşağıdaki şekilde yapılır (Örnek olması açısından, t-Testi ile aynı değişken kullanılmıştır, normalde uygun değişken seçimi yapılmalıdır).

```r
test = wilcox.test(veri$Fo, alternative = "two.sided", mu = 150)
if(test$p.value<0.05) {
  cat("Fo alanının ortalaması 150 değildir p=",test$p.value)
} else {
  cat("Fo alanının ortalaması 150 dir p=",test$p.value)
}
```

```
## Fo alanının ortalaması 150 dir p= 0.5459381
```

```r
test = wilcox.test(veri$Fo, alternative = "greater", mu = 160)
if(test$p.value<0.05) {
  cat("Fo alanının ortalaması 160'tan büyüktür p=",test$p.value)
} else {
```

```
  cat("Fo alanının ortalaması 160'tan büyük değildir dir p=",test$p.value)
}
```

```
## Fo alanının ortalaması 160'tan büyük değildir dir p= 0.9914166
```

```
test = wilcox.test(veri$Fo, alternative = "less", mu = 160)
if(test$p.value<0.05) {
  cat("Fo alanının ortalaması 160'tan küçüktür p=",test$p.value)
} else {
  cat("Fo alanının ortalaması 160'tan küçük değildir dir p=",test$p.value)
}
```

```
## Fo alanının ortalaması 160'tan küçüktür p= 0.008612979
```

### 6.2.2 Binom Testi

Binom testi, oran testinin parametrik olmayan halidir ve eleman sayısı küçük olduğunda kullanılması tavsiye edilir (N<30). Hipotezleri oran testi ile aynıdır. Bu testin R ile uygulanması aşağıdaki gibi olacaktır.

```
kurtulanlar = c(28)
toplam.hasta = c(30)

test = binom.test(kurtulanlar, toplam.hasta, p=0.75)

if(test$p.value < 0.05) {
   cat("Trafik kazalarından kurtulanların oranı %75 değildir p=",test$p.value)
} else {
   cat("Trafik kazalarından kurtulanların oranı %75'tir p=", test$p.value)
}
```

```
## Trafik kazalarından kurtulanların oranı %75 değildir p= 0.01877506
```

```
test
```

```
##
##  Exact binomial test
##
## data:  kurtulanlar and toplam.hasta
## number of successes = 28, number of trials = 30, p-value = 0.01878
## alternative hypothesis: true probability of success is not equal to 0.75
## 95 percent confidence interval:
##  0.7792646 0.9918219
## sample estimates:
```

```
## probability of success
##               0.9333333
```

```
test = binom.test(kurtulanlar, toplam.hasta, p=0.75, alternative = "g")

if(test$p.value < 0.05) {
   cat("Trafik kazalarından kurtulanların oranı %75'ten büyüktür p=",test$p.value)
} else {
   cat("Trafik kazalarından kurtulanların oranı %75'tir p=", test$p.value)
}
```

```
## Trafik kazalarından kurtulanların oranı %75'ten büyüktür p= 0.01059587
```

```
test
```

```
##
##  Exact binomial test
##
## data:  kurtulanlar and toplam.hasta
## number of successes = 28, number of trials = 30, p-value = 0.0106
## alternative hypothesis: true probability of success is greater than 0.75
## 95 percent confidence interval:
##  0.804674 1.000000
## sample estimates:
## probability of success
##               0.9333333
```

```
test = binom.test(kurtulanlar, toplam.hasta, p=0.75, alternative = "l")

if(test$p.value < 0.05) {
   cat("Trafik kazalarından kurtulanların oranı %75'ten küçüktür p=",test$p.value)
} else {
   cat("Trafik kazalarından kurtulanların oranı %75'tir p=", test$p.value)
}
```

```
## Trafik kazalarından kurtulanların oranı %75'tir p= 0.9980356
```

```
test
```

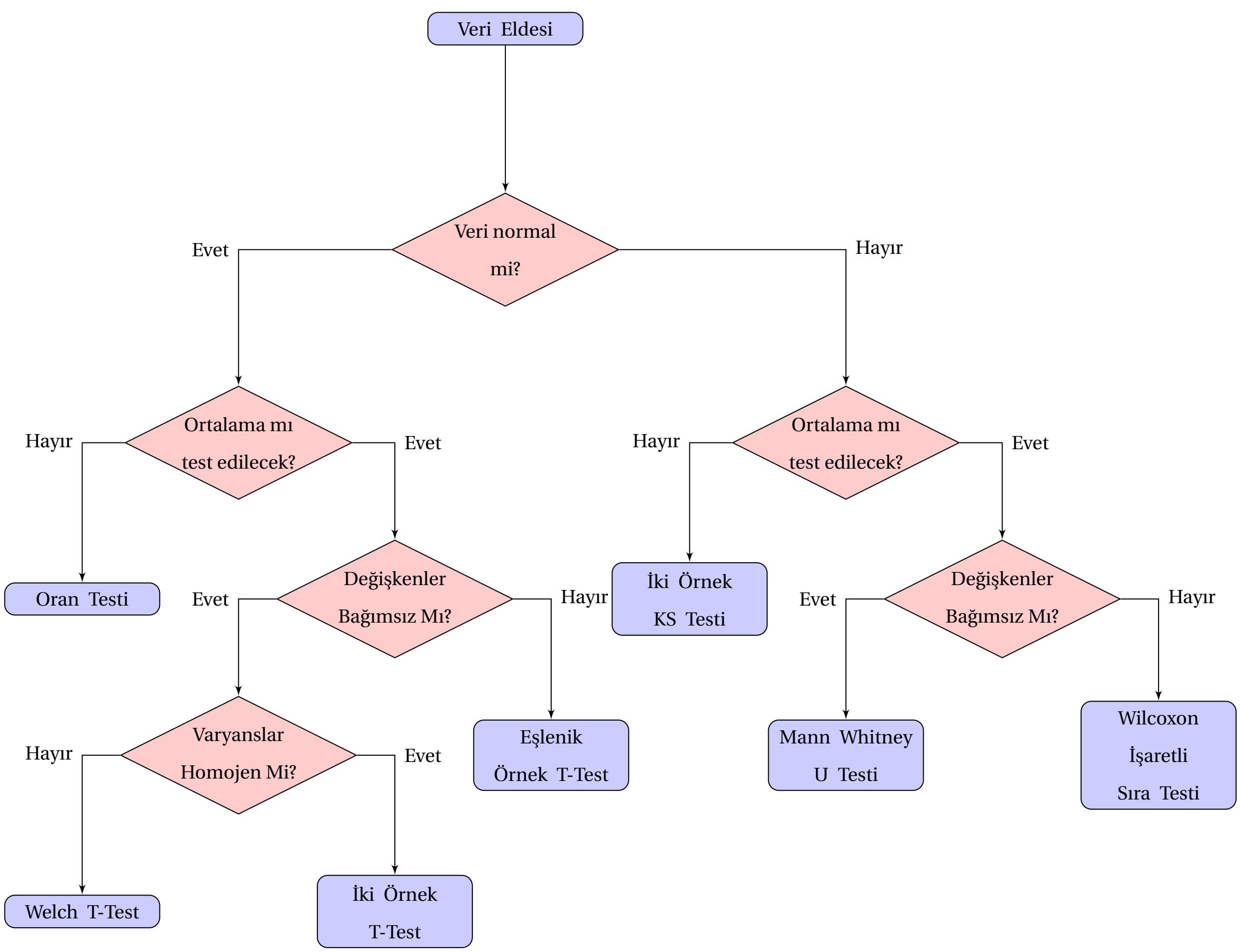
```
##
##  Exact binomial test
##
## data:  kurtulanlar and toplam.hasta
## number of successes = 28, number of trials = 30, p-value = 0.998
```

```
## alternative hypothesis: true probability of success is less than 0.75
## 95 percent confidence interval:
##  0.0000000 0.9880242
## sample estimates:
## probability of success
##              0.9333333
```

BÖLÜM

# 7 İki Örnek Testleri

İki anakütle ile ilgili parametrelerin karşılaştırılması için kullanılacak testlerdir. Aşağıdaki algoritmaya göre seçilirler. Ancak, duruma göre parametrik olmayan yöntemlerin doğrudan kullanılması söz konusu olabilir.



## 7.1 Parametrik Testler

Karşılaştırılacak anakütlelerin normal dağılıma sahip olduğu belirlenmişse, parametrelerinin karşılaştırılması amacıyla parametrik testler kullanılır. Bu parametreler anakütlelerin ortalamaları ve oranları olabilir.

### 7.1.1 İki Ortalamanın Testi

İki anakütle ortalaması, bu anakütlelerden alınacak örneklemelerin ortalamaları kullanılarak birbirleri ile karşılaştırılabilirler. Ancak, incelenen örneklerin bağımlı veya bağımsız olması durumuna göre farklı testler gerçekleştirilmelidir.

#### 7.1.1.1 İki Bağımsız Ortalamanın Testi

Farklı anakütlelerden alınan bağımsız iki $n_1$ ve $n_2$ elemanlı iki adet örneklem, ana kütle ortalamalarının test edilmesi için yeterli olacaktır. Bu test için yokluk hipotezi aşağıdaki gibi kurulur.

$$H_0 = \mu_1 = \mu_2$$

veya $\mu_1$ ve $\mu_2$ eşir ise aralarındaki fark 0 olacağından;

$$H_0 = \mu_1 - \mu_2 = 0$$

olur. Alternatif hipotez duruma göre oluşturulmalıdır.

Bu amaçla iki örnek t-Testi kullanılır. Bu testin iki farklı sürümü bulunmaktadır. İlk olarak varyansların homojen olması durumunu inceleyelim. Bu durumda test istatistiği aşağıdaki şekilde olacaktır.

$$t = \frac{\left(\overline{x}_1 - \overline{x}_2\right) - \left(\mu_1 - \mu_2\right)}{\sigma_{\overline{x}_1 - \overline{x}_2}}$$

Anakütle ortalamalarının aynı olduğu ($\mu_1 = \mu_2$) varsayımı altında, test istatistiği aşağıdaki gibi olacaktır.

$$t = \frac{\left(\overline{x}_1 - \overline{x}_2\right)}{\sigma_{\overline{x}_1 - \overline{x}_2}}$$

Burada, ortalama farklarının standart hatası $\sigma_{\overline{x}_1 - \overline{x}_2}$ varyansların homojen olduğu durumda;

- Anakütle varyansları ($\sigma_1 = \sigma_2 = \sigma$) biliniyorsa;

$$\sigma_{\overline{x}_1 - \overline{x}_2} = \sigma \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

- Anakütle varyansları bilinmiyorsa;

$$S_{\overline{x}_1-\overline{x}_2} = S\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$$
$$S = \sqrt{\frac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2}}$$

şeklinde olacaktır. Ancak varyans homojenliği her zaman sağlanabilen bir durum değildir. Bu sebeple, varyans homojenliğinin sağlanamadığı durumda kullanılabilecek bir alternatife ihtiyacınız olacaktır. Bu alternatif Welch tarafından önerilen t-Testidir. Bu test, varyans homojenliği sağlanamadığı durumda t-Dağılımının serbestlik derecesini, standart hatanın aşağıdaki yöntemle hesaplanması ile değiştirir ve t-Testinin gerçekleştirilmesini sağlar. Bu durumda ortalama farkının standart hatası;

- Anakütle varyansları $(\sigma_1, \sigma_2)$ biliniyorsa;

$$\sigma_{\overline{x}_1-\overline{x}_2} = \sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}$$

- Anakütle varyansları bilinmiyorsa;

$$S_{\overline{x}_1-\overline{x}_2} = \sqrt{\frac{S_1^2}{n_1}+\frac{S_2^2}{n_2}}$$

olur. Şimdi bu testi bu iki durumu inceleyerek R ile gerçekleştirelim. Yalnız unutulmaması gereken en önemli nokta, eğer incelediğiniz değişkenleri bir dönüşüm yardımıyla normalleştirmişseniz, ortalama farklarını inceleyeceğiniz değişkenlerin ikisininde **aynı** dönüşüm ile normal dağılıma dönmesidir.

```
# Parkinson verisinde D2 ve PPE değişkenleri logaritma dönüşümü
# ile normalleşmektedir.
d2.norm = log(veri$D2)
ppe.norm = log(veri$PPE)

# Normallik testlerini gerçekleştirelim.
d2.norm.test = shapiro.test(d2.norm)
ppe.norm.test = shapiro.test(ppe.norm)
```

```
if(d2.norm.test$p.value<0.05 || ppe.norm.test$p.value<0.05) {
  print("Değişkenler normal dağılmıyor, t-Testi uygulanamaz.")
} else {
  # Varyans Homojenliğini test edelim.
  birlestirilmis.degisken = c(d2.norm, ppe.norm)

  d2.uzunluk = length(d2.norm)
  ppe.uzunluk = length(ppe.norm)

  d2.belirtec = rep(1, d2.uzunluk)
  ppe.belirtec = rep(2, ppe.uzunluk)

  bileske.belirtec = c(d2.belirtec, ppe.belirtec)

  test = leveneTest(birlestirilmis.degisken, bileske.belirtec)

  p.degeri = test$`Pr(>F)`[1]

  if(p.degeri<0.05) {
    # Varyanslar Homojen Değil İki Yönlü Welch T-Test
    test = t.test(d2.norm, ppe.norm, var.equal = FALSE)

    # Testi Değerlendirelim
    if(test$p.value<0.05) {
      print("d2 ortalaması ppe'den farklıdır")
      print(test)
    } else {
      print("d2 ortalaması ppe ile aynıdır")
      print(test)
    }

    # Varyanslar Homojen Değil Tek Yönlü Welch T-Test
    test = t.test(d2.norm, ppe.norm, var.equal = FALSE, alternative = "g")

    # Testi Değerlendirelim
    if(test$p.value<0.05) {
      print("d2 ortalaması ppe'den büyüktür")
      print(test)
    } else {
      print("d2 ortalaması ppe ile aynıdır")
      print(test)
    }

    # Varyanslar Homojen Değil Tek Yönlü Welch T-Test
```

```
    test = t.test(d2.norm, ppe.norm, var.equal = FALSE, alternative = "l")

    # Testi Değerlendirelim
    if(test$p.value<0.05) {
      print("d2 ortalaması ppe'den küçüktür")
      print(test)
    } else {
      print("d2 ortalaması ppe ile aynıdır")
      print(test)
    }
  } else {
    # Varyanslar Homojen İki Yönlü Two Sample T-Test
    test = t.test(d2.norm, ppe.norm, var.equal = TRUE)

    # Testi Değerlendirelim
    if(test$p.value<0.05) {
      print("d2 ortalaması ppe'den farklıdır")
      print(test)
    } else {
      print("d2 ortalaması ppe ile aynıdır")
      print(test)
    }

    # Varyanslar Homojen Tek Yönlü Two Sample T-Test
    test = t.test(d2.norm, ppe.norm, var.equal = TRUE, alternative = "g")

    # Testi Değerlendirelim
    if(test$p.value<0.05) {
      print("d2 ortalaması ppe'den büyüktür")
      print(test)
    } else {
      print("d2 ortalaması ppe ile aynıdır")
      print(test)
    }

        # Varyanslar Homojen Tek Yönlü Two Sample T-Test
    test = t.test(d2.norm, ppe.norm, var.equal = TRUE, alternative = "l")

    # Testi Değerlendirelim
    if(test$p.value<0.05) {
      print("d2 ortalaması ppe'den küçüktür")
      print(test)
    } else {
      print("d2 ortalaması ppe ile aynıdır")
```

```
      print(test)
    }
  }
}
```

```
## [1] "d2 ortalaması ppe'den farklıdır"
##
##  Welch Two Sample t-test
##
## data:  d2.norm and ppe.norm
## t = 73.507, df = 241.87, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.461529 2.597088
## sample estimates:
##  mean of x  mean of y
##  0.8551138 -1.6741948
##
## [1] "d2 ortalaması ppe'den büyüktür"
##
##  Welch Two Sample t-test
##
## data:  d2.norm and ppe.norm
## t = 73.507, df = 241.87, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  2.472493      Inf
## sample estimates:
##  mean of x  mean of y
##  0.8551138 -1.6741948
##
## [1] "d2 ortalaması ppe ile aynıdır"
##
##  Welch Two Sample t-test
##
## data:  d2.norm and ppe.norm
## t = 73.507, df = 241.87, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 2.586124
## sample estimates:
##  mean of x  mean of y
##  0.8551138 -1.6741948
```

#### 7.1.1.2 İki Bağımlı Ortalamanın Testi

Bazı araştırmalarda, bir sürecin mevcut duruma etkisinin gözlenmesi gerekmektedir. Örneğin bir eğitim yönteminin öğrencilerin başarımı üzerine etkisi, bir tedavinin bir hastalığa olan etkisi gibi. Bu durumda Eşleştirilmiş veya Eşlenik Örnek t-Testi (paired sample T-Test) gerçekleştirilir. Bu test ile, iki bağımlı örneklemin ortalamaları arasındaki fark incelenmektedir. Buna göre, ölçümler arası farkı $D_i$ ile ifade edersek;

$$D_i = x_{1i} - x_{2i}$$

olacaktır. Bu testin hipotezleri iki bağımsız örnek testi hipotezleri ile aynıdır. Fakat, burada araştırılması gereken normallik değişkenlerin ayrı aryı normalliği değildir. **Onun yerine yeni oluşturulan farkın yani $D_i$'nin normalliğine bakılarak bu teste karar verilir**. Kullanılacak test istatistiği aşağıdaki gibi olacaktır.

$$t = \frac{\sum D_i}{\sqrt{\frac{n\sum D_i^2 - (\sum D_i)^2}{n-1}}}$$

Buradaki örnek veri, bir SPSS dosyası olacaktır. Bu durumda, R ile veriyi yüklemek için *foreign* kütüphanesini kullanmanız gerekmektedir. Örnek veriyi, dersin sayfasından indirebilirsiniz. İstenen bilgi, bir derse ait not ortalamasının, öğrencilere uygulamalı eğitim verildiğinde arttığının kanıtlanmasıdır. Aşağıdaki R kodlarını inceleyiniz.

```r
library(foreign)

veri = read.spss('egitim.sav')

eo = veri$egıtımoncesı
es = veri$egıtımsonrası

D = eo - es

test = shapiro.test(D)

if(test$p.value<0.05) {
  cat("Veri Normal Dağılıma sahip değildir, t-Testi Yapılamaz.\n")
} else {
  # Eşlenik Örnek İki Yönlü  T Testi
  test = t.test(eo, es, paired = TRUE)

  if(test$p.value<0.05) {
    cat("Eğitim öncesi ile eğitim sonrası arasında fark vardır p=",
        test$p.value,"\n")
```

```
  } else {
    cat("Eğitim öncesi ile eğitim sonrası arasında fark yoktur p=",
        test$p.value,"\n")
  }

  # Eşlenik Örnek Tek Yönlü T Testi
  test = t.test(eo, es, paired = TRUE, alternative = "g")

  if(test$p.value<0.05) {
    cat("Eğitim öncesi, eğitim sonrasından büyüktür p=",
        test$p.value,"\n")
  } else {
    cat("Eğitim öncesi ile eğitim sonrası arasında fark yoktur p=",
        test$p.value,"\n")
  }

  # Eşlenik Örnek Tek Yönlü T Testi
  test = t.test(eo, es, paired = TRUE, alternative = "l")

  if(test$p.value<0.05) {
    cat("Eğitim öncesi, eğitim sonrasından küçüktür p=",
        test$p.value,"\n")
  } else {
    cat("Eğitim öncesi ile eğitim sonrası arasında fark yoktur p=",
        test$p.value,"\n")
  }
}
```

```
## Eğitim öncesi ile eğitim sonrası arasında fark vardır p= 0.01639007
## Eğitim öncesi ile eğitim sonrası arasında fark yoktur p= 0.991805
## Eğitim öncesi, eğitim sonrasından küçüktür p= 0.008195035
```

### 7.1.2 İki Oranın Testi

Oran testi ayrıca iki oranın testi içinde kullanılan bir testtir. Örneğin, bir doktor emniyet kemeri kullanımının trafik kazalarındaki kurtulma oranını arttırdığını iddia etsin. Bu durumda kullanılacak test istatistiği aşağıdaki gibi olacaktır.

$$t = \frac{p_A - p_B}{\sqrt{\frac{pq}{n_A} + \frac{pq}{n_B}}}$$

Burada p ve q genel oranları ifade eder. Kurulacak yokluk hipotezi aşağıdaki gibi olacaktır. Alternatif hipotezler istenilen ifadeye uygun olarak kurulmalıdır.

$$H_0 : p_A = p_B$$

```
kurtulanlar = c(1443, 1781)
toplam.hasta = c(1490, 1916)

test = prop.test(kurtulanlar, toplam.hasta )
if(test$p.value<0.05) {
  cat("Emniyet kemeri kullananlar ile kullanmayanlar arasında fark vardır p=",
      test$p.value)
} else {
  cat("Emniyet kemeri kullananlar ile kullanmayanlar arasında fark yoktur p=",
      test$p.value)
}
```

```
## Emniyet kemeri kullananlar ile kullanmayanlar arasında fark vardır p= 8.104882e-07
```

```
test = prop.test(kurtulanlar, toplam.hasta, alternative = "greater")
if(test$p.value<0.05) {
  cat("Emniyet kemeri kullananların kurtulma oranı daha fazladır p=", test$p.value)
} else {
  cat("Emniyet kemeri kullananların kurtulma oranı aynıdır p=", test$p.value)
}
```

```
## Emniyet kemeri kullananların kurtulma oranı daha fazladır p= 4.052441e-07
```

```
test=prop.test(kurtulanlar, toplam.hasta, alternative = "less")
if(test$p.value<0.05) {
  cat("Emniyet kemeri kullananların kurtulma oranı daha azdır p=", test$p.value)
} else {
  cat("Emniyet kemeri kullananların kurtulma oranı aynıdır p=", test$p.value)
}
```

```
## Emniyet kemeri kullananların kurtulma oranı aynıdır p= 0.9999996
```

## 7.2 Parametrik Olmayan Testler

Parametrik testlerin uygulanamadığı durumlarda kullanılacak testlerdir. Verinin bağımlı veya bağımsız olmasına göre, Mann Whitney U, Wald Wolfowitz ve Wilcoxon testleri uygulanır. Bunu yanı sıra, bu testler normallik testleri gerçekleştirmeden araştırmacının isteğine göre veya veri türüne göre doğrudan uygulanabilirler.

### 7.2.1 Mann Whitney U Testi

Bağımsız örnek T-Testinin parametrik olmayan karşılığıdır. Bağımlı değişkenin aralıklı veya oransal ölçek olmadığı durumlarda, bağımlı değişkenin aralıklı veya oransal ölçek olduğu ancak örnek birim sayısının 30'dan küçük olduğu durumlarda, gözlem sayısı yeterli olduğu halde parametrik test varsayımlarının sağlanmadığı durumlarda kullanılmaktadır.

Testin hipotezleri bağımsız örnek t-Testi ile aynıdır. Test istatistiğinin iki farklı hesaplanması söz konusudur.

- $n_1 + n_2 < 20$ **olduğu durumda**

İki veri birleştirilir ve sıralanır. Bütün değerlere sıra sayıları verilir. Aynı değerli elemanların tümüne ortalama sıra sayısı değeri verilir. $U_1$ ve $U_2$ test istatistikleri aşağıdaki şekilde hesaplanır.

$$
\begin{aligned}
U_1 &= n_1 n_2 + \frac{n_1(n_1+1)}{2} - \sum_{i=0}^{n_1} S_{1i} \\
U_2 &= n_2 n_1 + \frac{n_2(n_2+1)}{2} - \sum_{i=0}^{n_2} S_{2i} \\
U &= \min\{U_1, U_2\}
\end{aligned}
$$

Burada $S_{ij}$ i. veriye ait j. sıra sayısını ifade eder.

- $n_1 + n_2 > 20$ **olduğu durumda**

Bu durumda birleştirilmiş verinin normal dağılıma sahip olması olasılığı artmaktadır. Bu sebeple test istatistiği aşağıdaki şekilde olacaktır.

$$
Z = \frac{U - \frac{n_1 n_2}{2}}{\sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}}
$$

Burada, U örneklerin birinden hesaplanacak olan U değeridir. Kullanılacak olan U değerinin hangi örnekten hesaplanacağı önemli değildir, çünkü yapılacak seçim sadece Z istatistiğinin işaretini değiştirecektir.

Yukarıda kullanıdığımız veriler için Mann-Whitney U testi gerçekleştirmek için aşağıdaki R kodunu inceleyebilirsiniz.

```r
# Parkinson verisinde yer alan d2 ve ppe'yi inceleyelim
veri = read.csv2('parkinsons.csv')

d2 = veri$D2
ppe = veri$PPE

# Mann-Whitney U testini Uygulayalım
test = wilcox.test(d2, ppe, paired = FALSE)
```

```
# Testi Yorumlayalım
if(test$p.value<0.05) {
  cat("d2 ve ppe ortalamaları aynı değidir p=",
      test$p.value,"\n")
} else {
  cat("d2 ve ppe ortalamaları aydır p=",
      test$p.value,"\n")
}
```

```
## d2 ve ppe ortalamaları aynı değidir p= 2.080542e-65
```

```
# Mann-Whitney U testini Uygulayalım
test = wilcox.test(d2, ppe, paired = FALSE, alternative = "g")

# Testi Yorumlayalım
if(test$p.value<0.05) {
  cat("d2 > ppe p=",
      test$p.value,"\n")
} else {
  cat("d2 ve ppe ortalamaları aydır p=",
      test$p.value,"\n")
}
```

```
## d2 > ppe p= 1.040271e-65
```

```
# Mann-Whitney U testini Uygulayalım
test = wilcox.test(d2, ppe, paired = FALSE, alternative = "l")

# Testi Yorumlayalım
if(test$p.value<0.05) {
  cat("d2 < ppe p=",
      test$p.value,"\n")
} else {
  cat("d2 ve ppe ortalamaları aydır p=",
      test$p.value,"\n")
}
```

```
## d2 ve ppe ortalamaları aydır p= 1
```

**NOT**

R koduna dikkat ederseniz, *wilcox.test* foksiyonunun kullanıldığını görürsünüz. Bu durum R programcıları tarafından aşağıdaki şekilde açıklanmaktadır:

The literature is not unanimous about the definitions of the Wilcoxon rank sum and Mann-Whitney tests.
The two most common definitions correspond to the sum of the ranks of the first sample with the minimum value subtracted or not: R subtracts and S-PLUS does not, giving a value which is larger by m(m+1)/2 for a first sample of size m. (It seems Wilcoxon's original paper used the unadjusted sum of the ranks but subsequent tables subtracted the minimum.)
R's value can also be computed as the number of all pairs (x[i], y[j]) for which y[j] is not greater than x[i], the most common definition of the Mann-Whitney test.

---

Burada, R programcıları literatürde de bu iki testin literatürdeki tanımları hakkında kabul görmüş bir tanım olmadığından bahsetmektedir. Temelde bu testi öneren Wilcoxon'dır ve önerdiği test eşit eleman sayısına sahip örnekler için geliştirmiştir. Mann ve Whitney ise bu testin eleman eşitliğini şartını kaldırarak genelleştirmiştir. Bu test aynı zamanda **Wilcoxon Sıralama Toplamı Testi** veya **Wilcoxon-Mann-Whitney Testi** olarakta adlandırılır.

### 7.2.2 Wilcoxon İşaretli Sıra Testi

Bu test bağımlı örnek t-Testinin alternatifidir. Burada ortalama yerine medyan değerlerinin farkından bahsedilmektedir. Test istatistiği aşağıdaki şekilde hesaplanır.

$$
\begin{aligned}
D_i^+ &= \sum_{j=0}^{n} \begin{cases} 1 & x_{1j} > x_{2j} \\ 0 & x_{1j} <= x_{2j} \end{cases} \\
D_i^- &= \sum_{j=0}^{n} \begin{cases} 1 & x_{1j} < x_{2j} \\ 0 & x_{1j} >= x_{2j} \end{cases} \\
t_w &= \min\left\{D_i^+, D_i^-\right\}
\end{aligned}
$$

Kurulacak olan hipotezler, bağımlı örnek t Testi ile aynıdır. Bağımlı örnek T-Testinde kullandığımız problemi Wilcoxon testi ile çözümü aşağıda verilmiştir.

```r
library(foreign)

veri = read.spss('egitim.sav')

eo = veri$egıtımoncesı
es = veri$egıtımsonrası

# İki Yönlü Wilcoxon İşaretli Sayı Testini Uygulayalım
```

```
test = wilcox.test(eo, es, paired = TRUE)

# Testi Değerlendirelim
if(test$p.value<0.05) {
  cat("Eğitim Öncesi Eğitim sonrasından farklıdır p=",
      test$p.value,"\n")
} else {
  cat("Eğitim Öncesi Eğitim sonrası ile aynıdır p=",
      test$p.value,"\n")
}
```

```
## Eğitim Öncesi Eğitim sonrasından farklıdır p= 0.01387757
```

```
# Tek Yönlü Wilcoxon İşaretli Sayı Testini Uygulayalım
test = wilcox.test(eo, es, paired = TRUE, alternative = "g")

# Testi Değerlendirelim
if(test$p.value<0.05) {
  cat("Eğitim Öncesi Eğitim sonrasından büyüktür p=",
      test$p.value,"\n")
} else {
  cat("Eğitim Öncesi Eğitim sonrası ile aynıdır p=",
      test$p.value,"\n")
}
```

```
## Eğitim Öncesi Eğitim sonrası ile aynıdır p= 0.9933858
```

```
# İki Yönlü Wilcoxon İşaretli Sayı Testini Uygulayalım
test = wilcox.test(eo, es, paired = TRUE, alternative = "l")

# Testi Değerlendirelim
if(test$p.value<0.05) {
  cat("Eğitim Öncesi Eğitim sonrasından küçüktür p=",
      test$p.value,"\n")
} else {
  cat("Eğitim Öncesi Eğitim sonrası ile aynıdır p=",
      test$p.value,"\n")
}
```

```
## Eğitim Öncesi Eğitim sonrasından küçüktür p= 0.006938787
```

### 7.2.3 Kolmogorov-Smirnov İki Örnek Testi

Kolmogorov-Smirnov testinin bir diğer kullanım alanı da iki değişkenin dağılımlarının parametreleriyle birlikte aynı olup olmadığının test edilmesidir. Burada, hipotezler aşağıdaki gibi kurulur.

| | |
|---|---|
| $H_0$ : | Dağılımlar Aynıdır |
| $H_1$ : | Dağılımlar Aynı Değildir |

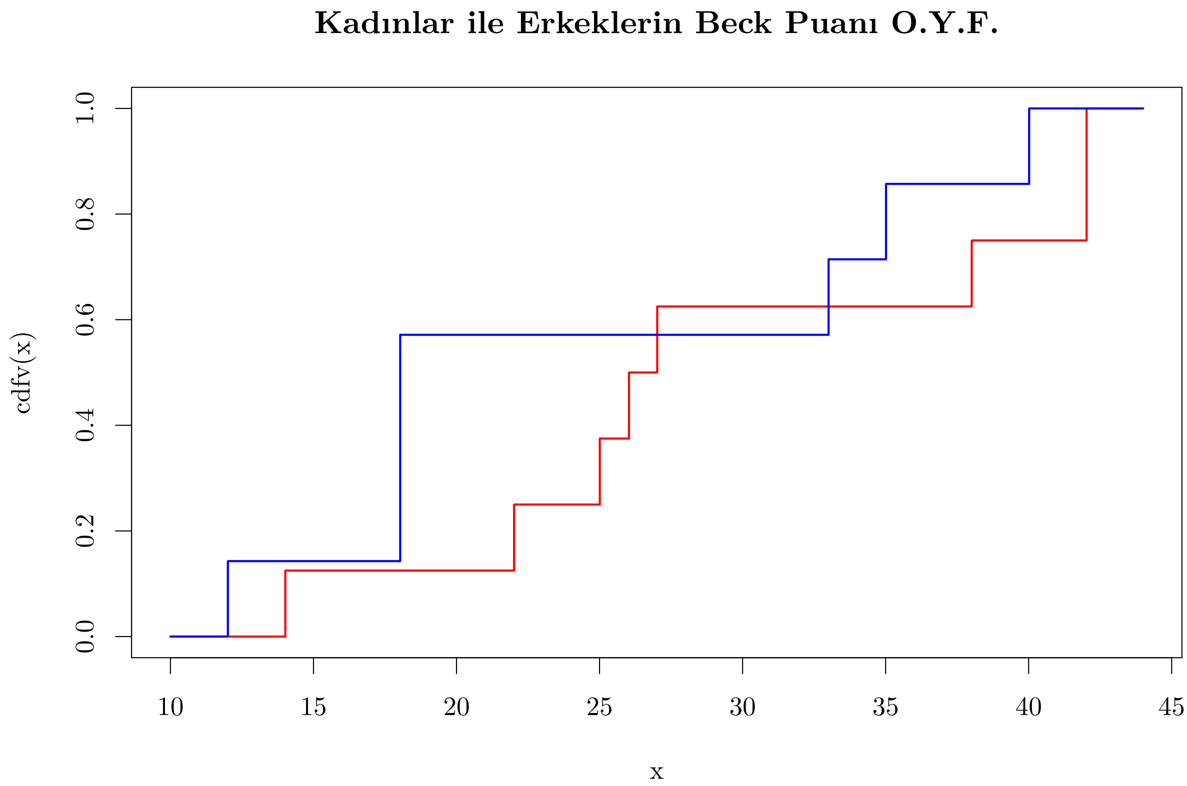
Örnek olarak, bir psikiyatri kliniğine başvuran hastalara uygulanan Beck Depresyon ölçeği değerlerinin kadın ve erkekler için aynı dağılıma sahip olup olmadığını inceleyelim. Veriler yine bir SPSS dosyasından yüklenmektedir.

```
veri = read.spss('beck_puan.sav')

cinsiyet = veri$cınsıyet[!is.na(veri$cınsıyet)]
beck_puan = veri$beckpuanı[!is.na(veri$beckpuanı)]

kadin_beck_puan = beck_puan[cinsiyet=="kadın"]
erkek_beck_puan = beck_puan[cinsiyet=="erkek"]

test = ks.test(erkek_beck_puan, kadin_beck_puan)

if(test$p.value<0.05) {
  cat("Erkek ve Kadınların depresyon puanları aynı dağılıma",
      "sahip değildir p=", test$p.value,"\n")
} else {
  cat("Erkek ve Kadınların depresyon puanları aynı dağılıma",
      "sahiptir p=", test$p.value,"\n")
}
```

```
## Erkek ve Kadınların depresyon puanları aynı dağılıma sahiptir p= 0.4464094
```

Grafiksel olarak incelendiğinde, aşağıdaki sonuçlar elde edilecektir.

```
xMin = min(min(kadin_beck_puan), min(erkek_beck_puan))-2
xMax = max(max(kadin_beck_puan), max(erkek_beck_puan))+2

x = seq(from=xMin, to=xMax, length.out = 1000)

cdfv = ecdf(kadin_beck_puan)
cdfy = ecdf(erkek_beck_puan)

plot(x, cdfv(x), main = "Kadınlar ile Erkeklerin Beck Puanı O.Y.F.",
     type="s", col="red", lwd=2)
lines(x, cdfy(x), type="s", col="blue", lwd=2)
```


Kadınlar ile Erkeklerin Beck Puanı O.Y.F.
cdfv(x)
0.0
0.2
0.4
0.6
0.8
1.0
10
15
20
25
30
35
40
45
x

BÖLÜM

# 8

# Uygulama: Uyku Anketi Değerlendirmesi

Bu uygulamada, sitedeki *Uyku Ölçeği* verisini indirip, aşağıdaki işlemleri gerçekleştiriniz. Aksi belirtilmediği sürece $\alpha = 0.05$ kullanınız.

- Nominal Değerli değişkenlerin frekans tablolarını oluşturunuz.
- Medeni hal değişkeni ile eğitim seviyesi değişkenleri için çapraz frekans tablosu, çapraz marjinal frekans tablosu ve oran tablolarını oluşturunuz.
- Medeni Hal ve Eğitim Seviyesi değişkenleri için pasta grafiği ve histogram oluşturunuz.
- Medeni Hal ile Uykuya Dalma Güçlüğü değişkenlerinin arasında anlamlı bir ilişki olup olmadığını belirleyiniz.
- Epworth Uyku Ölçeği, HADS Anksiyete ve HADS Depresyon ölçeği değerlerinin normalliğini belirleyiniz, mümkünse normalleştiriniz.
- Epworth Uyku Ölçeği değerinin ortalamasının 12'den, HADS Anksiyete ve HADS depresyon ölçeği skorlarının 10'dan büyük olup olmadığını belirleyiniz.
- Hafta İçi Uyku Süresinin, Hafta Sonu Uyku Süresi ile aynı olup olmadığını inceleyiniz.
- Erkeklerin uyku problemine sahip olmalarının oranının, kadınlarınkinden farklı olup olmadığını inceleyiniz.
- Erkekler ve kadınlar için epworth uyku olceği, HADS anksiyete ve HADS depresyon puanlarını farklarını inceleyiniz.

BÖLÜM

# 9

# Çok Örnek Testleri

İstatistiksel olarak 2'den fazla örneğin karşılaştırılması gerektiğinde kullanılması önerilen testlerdir.

Bağımsız örneklerin test edilmesi sözkonusu olduğunda, 2 adet örnek için t-Testinin kullanılması gerektiğini öğrenmiştik. Eğer 2'den fazla bağımsız örneğin karşılaştırılması gerekli olduğu durumlarda ise t-Testinin anlamlılık seviyesi düşecektir. Her bir ikili için t-Testi kullanılması durumunda testlerin hepsinden %5 hata oranı gelecektir. Örneğin, 3 adet ikili karşılaştırma için testin gücü binom dağılımına göre aşağıdaki gibi hesaplanır.

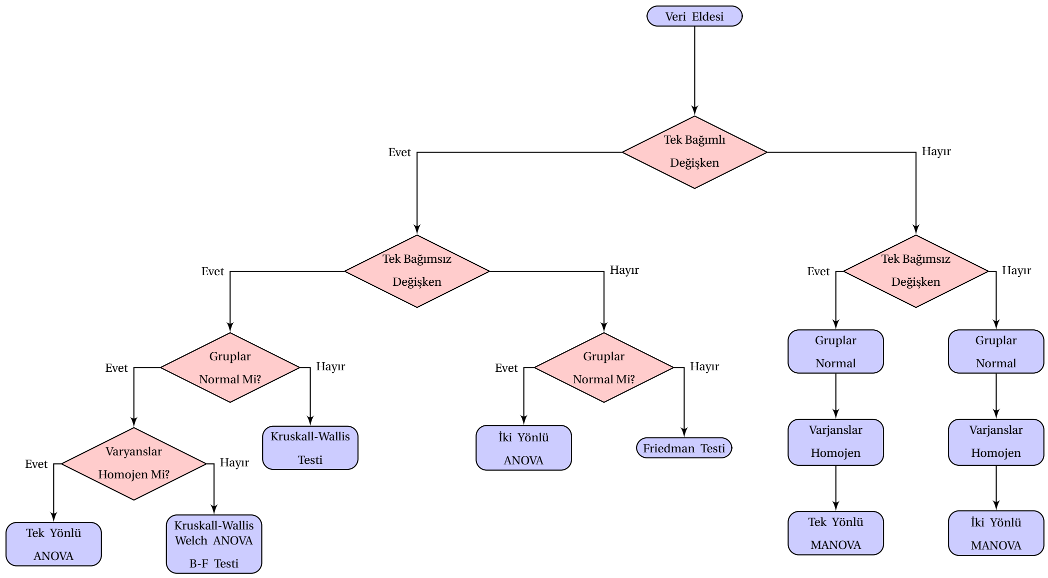
$$P(x=3) = C_3^3(0.95)^3(0.05)^0 = 0.95^3 = 0.86$$

Bu durumda, $\alpha = 1-\beta = 1-0.86 = 0.14$ olarak hesaplanır. Ancak, bütün bu işlemleri ANOVA (ANalysis Of VAriance) testi ile gerçekleştirmeniz durumunda $\alpha = 0.05$ olacak ve testinizin gücü artacaktır.

Anova testinin farklı varyasyonları bulunmaktadır. Bu varyasyonlar;

- Bir bağımsız değişkenin, bir bağımlı değişken üzerindeki etkisinin incelendiği, Tek Yönlü ANOVA
- İki bağımsız değişkenin, bir bağımlı değişken üzerindeki etkisinin incelendiği, İki Yönlü ANOVA
- İkiden fazla bağımlı örneğin karşılaştırılmasında kullanılan, Bağımlı Örnekler için ANOVA
- Bir bağımsız değişkenin, birden fazla bağımlı değişken üzerindeki etkisinin incelendiği, Tek Yönlü MANOVA
- İki bağımsız değişkenin, birden fazla bağımlı değişken üzerindeki etkisinin incelendiği, İki Yönlü MANOVA

Ayrıca ANOVA testinin parametrik olmayan karşılıklarıda bulunmaktadır. Genellikle ANOVA'nın varsayımlarının sağlanamadığı koşullarda, Kruskall-Wallis (Tek Yönlü ANOVA) veya Friedman (İki Yönlü ANOVA) testi kullanılmaktadır. Aşağıdaki akış şemasında bu durum özetlenmektedir.




Anova testini bir modelleme aracı olarak da kabul etmek mümkündür. Zaten R ile yapacağınız testlerde anova modeli oluşturmanız gerekmektedir.

## 9.1 Tek Yönlü ANOVA Testi

Tek yönlü anova testinde; bağımlı değişkenin bağımsız değişkenin düzeylerine göre farklılaşıp farklılaşmadığı, farklı gruplardaki varyansların karşılaştırılması temeline dayanan F testi ile incelenmektedir. Anova testi parametrik bir testtir ve bu testte kullanılacak bütün örnekler parametrik test varsayımlarını sağlamalıdır.

Bu testin hipotezi aşağıdaki şekilde kurulur.

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k$$
$$H_1 : \mu_1 \neq \mu_2 \neq \cdots \neq \mu_k$$

Kullanılacak test istatistiği aşağıdaki şekilde hesaplanır.

İlk olarak örneklerin ortalamaları hesaplanır. Bu ortalamalar kullanılarak genel ortalama aşağıdaki formül ile hesaplanır.

$$\overline{X} = \frac{\sum_i^{n_g} \overline{X}_i}{n_g}$$

Burada $n_g$ grup sayısı, $X_i$'de i. grup ortalamasını ifade eder. Sonra gruplar arası ortalama kareler farkı $\overline{D_{GA}}$ aşağıdaki formül ile hesaplanır.

$$D_{GA} = n\sum_{i}^{n_g}\left(\overline{X}_i - \overline{X}\right)^2$$
$$SD_{GA} = n_g - 1$$
$$\overline{D}_{GA} = \frac{D_{GA}}{SD_{GA}}$$

Grup içi ortalama kareler farkı $\overline{D}_{Gİ}$ da aşağıdaki şekilde hesaplanır.

$$D_{Gİ} = \sum_{i}^{n}\left(X_i - \overline{X}_i\right)^2$$
$$SD_{Gİ} = n_g(n-1)$$
$$\overline{D}_{Gİ} = \frac{D_{Gİ}}{SD_{Gİ}}$$

Burada n örneklerin eleman sayısıdır. Bu iki değer elde edildikten sonra kullanılacak olan F istatistiği aşağıdaki şekilde hesaplanır.

$$F = \frac{\overline{D}_{GA}}{\overline{D}_{Gİ}}$$

Şimdi R ile anova testinin nasıl gerçekleştirildiğine bakalım. Bu işlem için R'ın model (formül) yazımını kullanmalısınız. Bu yazımda, bağımlı değişken bağımsız değişkenlerle ifade edilir şeklinde bir ifade söz konusudur. Örneğin, $A \sim B$ şeklindeki bir ifade de A **bağımlı değişken**, B de **bağımsız** değişken olmaktadır. Örnek olarak, çeşitli bölgelerden elde edilmiş fil dişlerinin içerdiği kimyasal maddelerin farklarını incelemeniz istensin. Aşağıdaki R kodunu inceleyin.

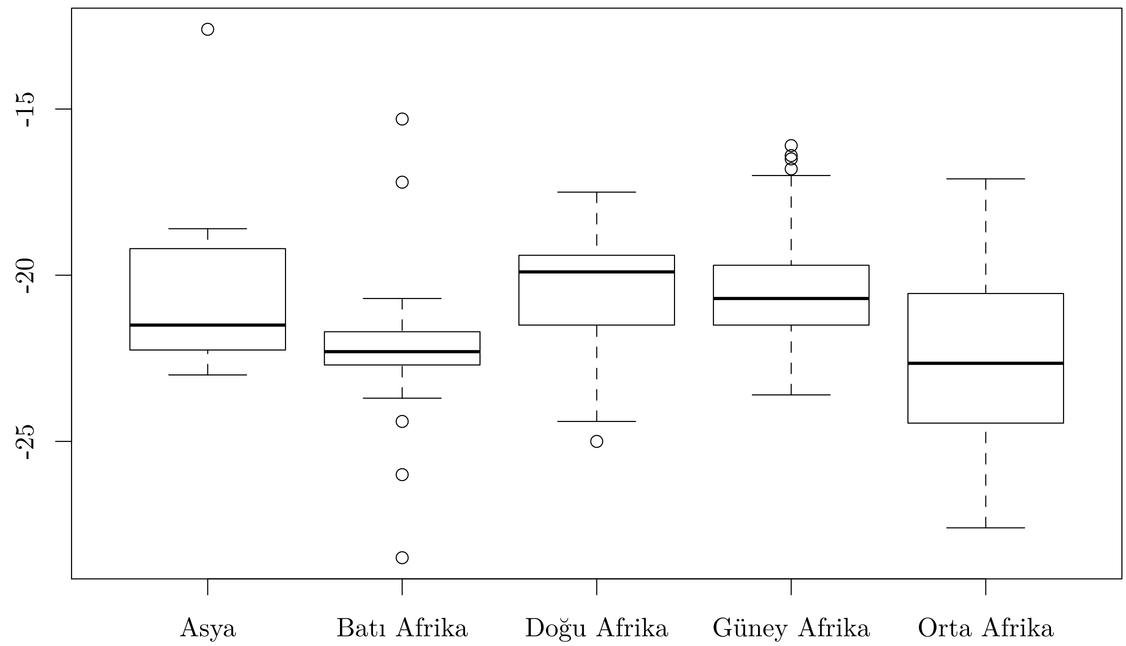
```
veri = read.csv2('fildisi.csv')

aov.ornek = aov(delta13C~bolge, data = veri)

summary(aov.ornek)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## bolge         4  413.6  103.41   31.65 <2e-16 ***
## Residuals   490 1600.9    3.27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Değişkenlerin ortalamaları arasında anlamlı bir fark çıktığından ötürü, hangi iki değişkenlerin arasında fark olduğunu öğrenmek için Post Hoc testlere başvurmalısınız.

Grafiksel olarak bu durumu incelendiğinde aşağıdaki durum ortaya çıkmaktadır.

```r
boxplot(delta13C~bolge, data=veri)
```




### 9.1.1 Eşit Varyans Varsayımı ile Kullanılacak Post Hoc Testler

Burada ortalamaların farklarından bahsedebilmek için ilk olarak varyans homojenliği şartı kontrol edilmelidir. Bu işlem yine levene testi ile gerçekleştirilebilir.

```r
library(car)

leveneTest(veri$delta13C, veri$bolge)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   4  17.943 9.23e-14 ***
##       490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Örnektende görüldüğü gibi eşit varyans varsayımı sağlanmamaktadır. Ancak, uygulama olması açısından tüm testler bu veriler üzerinde gerçekleştirilecektir.

#### 9.1.1.1 Tukey Testi

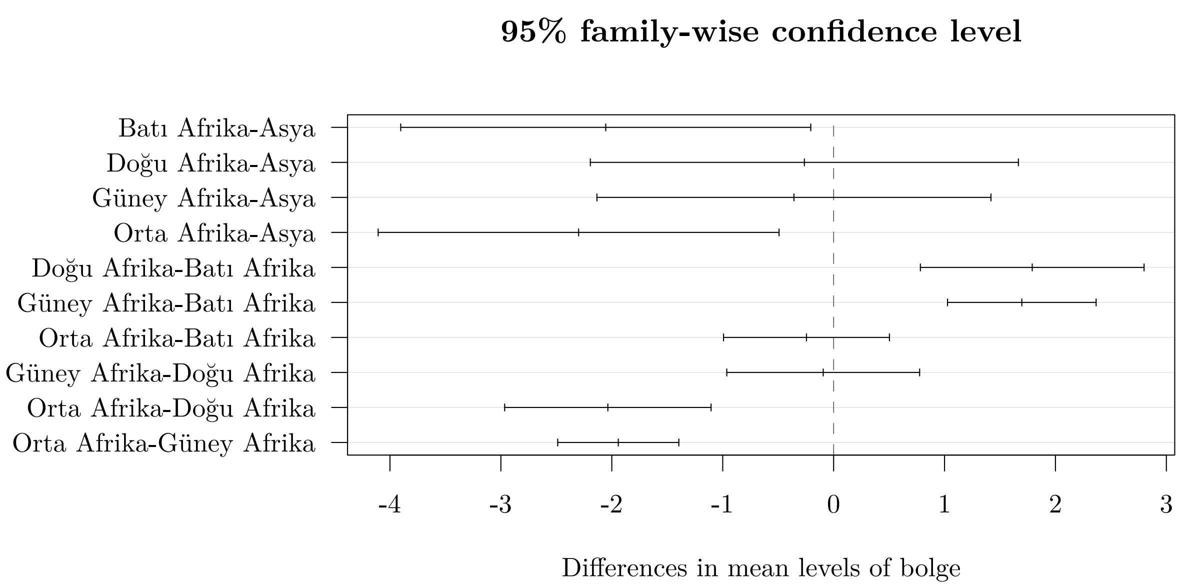
Bu test ortalamaların güven aralıkları gerekliyse veya grupların eleman sayıları aynı değilse kullanılmalıdır. Aksi durumlarda Scheffe testi veya Aile Çapı Hata Oranı yöntemleri denenmelidir. Aşağıdaki R kodunu inceleyin.

```r
TukeyHSD(x = aov.ornek, "bolge", conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = delta13C ~ bolge, data = veri)
##
## $bolge
##                                  diff        lwr        upr     p adj
## Batı Afrika-Asya          -2.05452899 -3.9029126 -0.2061454 0.0207243
## Doğu Afrika-Asya          -0.26385135 -2.1934931  1.6657904 0.9958149
## Güney Afrika-Asya         -0.35771073 -2.1340552  1.4186337 0.9817489
## Orta Afrika-Asya          -2.29916667 -4.1062811 -0.4920522 0.0048776
## Doğu Afrika-Batı Afrika    1.79067763  0.7822521  2.7991031 0.0000155
## Güney Afrika-Batı Afrika   1.69681826  1.0268898  2.3667468 0.0000000
## Orta Afrika-Batı Afrika   -0.24463768 -0.9923457  0.5030703 0.8984088
## Güney Afrika-Doğu Afrika -0.09385938 -0.9632267  0.7755079 0.9983329
## Orta Afrika-Doğu Afrika   -2.03531532 -2.9659401 -1.1046905 0.0000000
## Orta Afrika-Güney Afrika -1.94145594 -2.4872991 -1.3956127 0.0000000
```

Test sonuçlarına göre Doğu Afrika-Asya, Güney Afrika-Asya, Orta Afrika-Batı Afrika ve Güney Afrika-Doğu Afrika'da gerçekleştirilen ölçümler arasındaki farklar istatistiksel olarak anlamsızdır.

```r
par(mar=c(5,11,4,2)+0.1)
plot(TukeyHSD(x = aov.ornek, "bolge", conf.level = 0.95),
     las = 1, cex.axis=1)
```



#### 9.1.1.2 Scheffe Testi

Genel olarak, çok fazla sayıda karşılaştırma olduğu durumlarda bu test Tukey testinden daha iyi sonuçlar vermektedir. Aşağıdaki R kodunu inceleyin.

```
library(agricolae)

test = scheffe.test(aov.ornek, "bolge", group = TRUE, console = TRUE)
```

```
##
## Study: aov.ornek ~ "bolge"
##
## Scheffe Test for delta13C
##
## Mean Square Error  : 3.267213
##
## bolge,  means
##
##               delta13C      std   r   Min   Max
## Asya         -20.18750 3.385869   8 -23.0 -12.6
## Batı Afrika  -22.24203 1.549327  69 -28.5 -15.3
## Doğu Afrika  -20.45135 1.899798  37 -25.0 -17.5
## Güney Afrika -20.54521 1.413515 261 -23.6 -16.1
## Orta Afrika  -22.48667 2.439243 120 -27.6 -17.1
##
## alpha: 0.05 ; Df Error: 490
## Critical Value of F: 2.390132
##
## Harmonic Mean of Cell Sizes  27.98228
## Minimum Significant Difference: 1.494181
##
## Means with the same letter are not significantly different.
##
## Groups, Treatments and means
## a     Asya            -20.19
## a     Doğu Afrika     -20.45
## a     Güney Afrika    -20.55
## ab    Batı Afrika     -22.24
## b     Orta Afrika     -22.49
```

Sonuç tablosunda, aynı harf ile gösterilen ortalamalar arasındaki farklar istatistiksel olarak anlamlı değildir. Bu durumda Asya, Doğu Afrika, Güney Afrika ve Batı Afrika'da yapılan ölçümler arasındaki farklar ile Batı Afrika ve Orta Afrika'da yapılan ölçümler arasındaki farklar istatistiksel olarak anlamsızdır.

#### 9.1.1.3 Aile Çapı Hata Oranı (Family-Wise Error Rate, FWER)

Bu testler, gruplar arası farkları ikili t-Testi kullanarak bakılmasına dayalı olark yapılan testlerdir. Burada dikkat edilmesi gereken en önemli nokta, birden fazla ikili karşılaştırma işlemi gerçekleştirildiği için hesaplanan p değerlerinin düzeltilmesi gerekmesidir. Bu düzeltmeler için birden çok yöntem bulunmaktadır. R bu düzeltmeleri otomatik olarak yapmakta ve size karşılaştırma yapmanız gereken p değerlerini sunmaktadır. Ayrıca bu testlerin varyans homojenliği söz konusu olduğundan, hesaplanacak varyansın yerel olası gerekmektedir. Bu karşılaştırmaların, R ile gerçekleştirimleri aşağıdaki şekildedir.

- Düzeltme olmadan yapılan karşılaştırma

```
ikili.karsilastirma = pairwise.t.test(veri$delta13C, veri$bolge,
                                      p.adjust.method = "none",
                                      pool.sd = FALSE)
print(ikili.karsilastirma)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  veri$delta13C and veri$bolge
##
##              Asya  Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  0.132 -           -           -
## Doğu Afrika  0.836 6.6e-06     -           -
## Güney Afrika 0.774 7.1e-13     0.774       -
## Orta Afrika  0.098 0.401       1.1e-06     1.3e-13
##
## P value adjustment method: none
```

- Bonferroni Düzeltmesi ile karşılaştırma

```
ikili.karsilastirma = pairwise.t.test(veri$delta13C, veri$bolge,
                                      p.adjust.method = "bonferroni",
                                      pool.sd = FALSE)
print(ikili.karsilastirma)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  veri$delta13C and veri$bolge
##
##              Asya Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  1.00 -           -           -
## Doğu Afrika  1.00 6.6e-05     -           -
```

```
## Güney Afrika 1.00 7.1e-12     1.00         -
## Orta Afrika  0.98 1.00        1.1e-05      1.3e-12
##
## P value adjustment method: bonferroni
```

- Holm Düzeltmesi ile karşılaştırma

```
ikili.karsilastirma = pairwise.t.test(veri$delta13C, veri$bolge,
                                      p.adjust.method = "holm",
                                      pool.sd = FALSE)
print(ikili.karsilastirma)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  veri$delta13C and veri$bolge
##
##              Asya Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  0.66 -           -           -
## Doğu Afrika  1.00 4.6e-05     -           -
## Güney Afrika 1.00 6.3e-12     1.00        -
## Orta Afrika  0.59 1.00        8.6e-06     1.3e-12
##
## P value adjustment method: holm
```

- Hochberg Düzeltmesi ile karşılaştırma

```
ikili.karsilastirma = pairwise.t.test(veri$delta13C, veri$bolge,
                                      p.adjust.method = "hochberg",
                                      pool.sd = FALSE)
print(ikili.karsilastirma)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  veri$delta13C and veri$bolge
##
##              Asya Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  0.66 -           -           -
## Doğu Afrika  0.84 4.6e-05     -           -
## Güney Afrika 0.84 6.3e-12     0.84        -
## Orta Afrika  0.59 0.84        8.6e-06     1.3e-12
##
## P value adjustment method: hochberg
```

- Hommel Düzeltmesi ile karşılaştırma

```
ikili.karsilastirma = pairwise.t.test(veri$delta13C, veri$bolge,
                                      p.adjust.method = "hommel",
                                      pool.sd = FALSE)
print(ikili.karsilastirma)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  veri$delta13C and veri$bolge
##
##              Asya Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  0.66 -           -           -
## Doğu Afrika  0.84 4.6e-05     -           -
## Güney Afrika 0.84 6.3e-12     0.84        -
## Orta Afrika  0.49 0.84        8.6e-06     1.3e-12
##
## P value adjustment method: hommel
```

- Benjamini & Hochberg Düzeltmesi ile karşılaştırma

```
ikili.karsilastirma = pairwise.t.test(veri$delta13C, veri$bolge,
                                      p.adjust.method = "BH",
                                      pool.sd = FALSE)
print(ikili.karsilastirma)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  veri$delta13C and veri$bolge
##
##              Asya Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  0.22 -           -           -
## Doğu Afrika  0.84 1.7e-05     -           -
## Güney Afrika 0.84 3.5e-12     0.84        -
## Orta Afrika  0.20 0.57        3.6e-06     1.3e-12
##
## P value adjustment method: BH
```

- Benjamini & Yekutieli Düzeltmesi ile karşılaştırma

```
ikili.karsilastirma = pairwise.t.test(veri$delta13C, veri$bolge,
                                      p.adjust.method = "BY",
                                      pool.sd = FALSE)
print(ikili.karsilastirma)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  veri$delta13C and veri$bolge
##
##              Asya Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  0.64 -           -           -
## Doğu Afrika  1.00 4.9e-05     -           -
## Güney Afrika 1.00 1.0e-11     1.00        -
## Orta Afrika  0.58 1.00        1.0e-05     3.9e-12
##
## P value adjustment method: BY
```

### 9.1.2 Eşit Varyans Varsayımı Sağlanmadığında Kullanılan Post Hoc Testler

Eşit varyans varsayımı sağlanmadığında kullanılabilecek post-hoc testleri aşağıda listelenmiştir. Genellikle, eşit örnek sayısı sağlanması ve bu testlerin tercih edilmemesi önerilmektedir. Ayrıca, bu testler Welch ve Brown-Forsythe testlerininde varsayılan PostHoc testleridir. Ama elinizdeki veri sizi sınırlıyorsa bu durumda aşağıdaki testleri kullanmanız önerilmektedir.

#### 9.1.2.1 Dunnet C Testi

Bu test, varyans homojenliği sağlanamadığı durumlarda kullanılan PostHoc testlerden biridir. Örneklerin eşit sayıda olması ve varyans homojenliği sağlanmadığı durumunda tercih edilmesi önerilmektedir. Sadece bir kontrol değişkeni ile diğerlerinin farklarının anlamlılık düzeylerini inceler. Kullanımı aşağıdaki gibidir.

```
library(DescTools)

DunnettTest(delta13C~bolge, data = veri, control="Asya")
```

```
##
##   Dunnett's test for comparing several treatments with a control :
##     95% family-wise confidence level
##
## $Asya
##                        diff    lwr.ci     upr.ci   pval
## Batı Afrika-Asya  -2.0545290 -3.558167 -0.5508907 0.0054 **
## Doğu Afrika-Asya  -0.2638514 -1.833592  1.3058894 0.9214
```

```
## Güney Afrika-Asya -0.3577107 -1.802746  1.0873245 0.8020
## Orta Afrika-Asya  -2.2991667 -3.769233 -0.8291004 0.0011 **
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#### 9.1.2.2 Dunnet T3 Testi

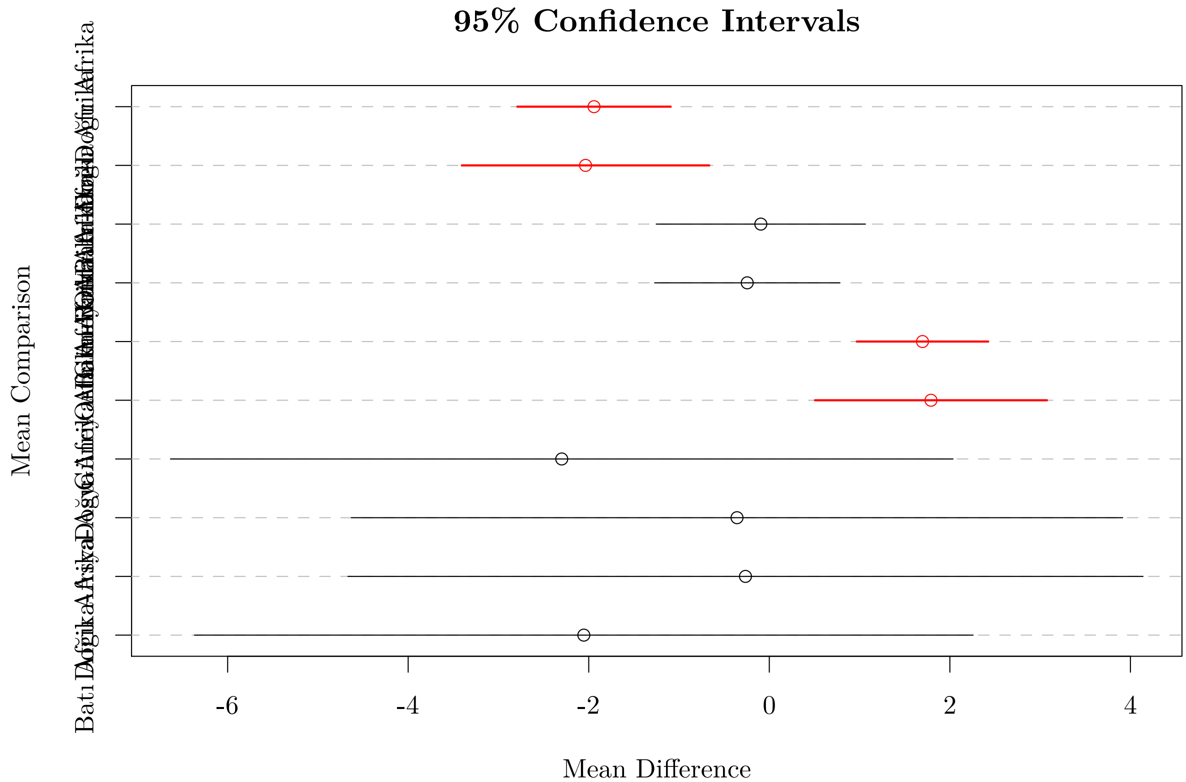
Bu test, Tukey-Cramer testinin varyans homojenliği sağlanmadığı ve/veya örnek sayısının eşit olmadığı durumda kullanılabilecek halidir. Tamhane T2 testinin gelişmiş halidir. Standart R kurulumunda bulunmamakla beraber, DTK paketi ile bilgisayarınızda kullanılabilir hale gelir. Aşağıdaki R kodunu inceleyin.

```
library(DTK)

ikili.karsilastirma = DTK.test(veri$delta13C, veri$bolge)
print(ikili.karsilastirma)
```

```
## [[1]]
## [1] 0.05
##
## [[2]]
##                                Diff   Lower CI   Upper CI
## Batı Afrika-Asya         -2.05452899 -6.3671207  2.2580627
## Doğu Afrika-Asya         -0.26385135 -4.6676740  4.1399713
## Güney Afrika-Asya        -0.35771073 -4.6302559  3.9148344
## Orta Afrika-Asya         -2.29916667 -6.6334371  2.0351038
## Doğu Afrika-Batı Afrika   1.79067763  0.5054660  3.0758893
## Güney Afrika-Batı Afrika  1.69681826  0.9689647  2.4246718
## Orta Afrika-Batı Afrika  -0.24463768 -1.2708412  0.7815658
## Güney Afrika-Doğu Afrika -0.09385938 -1.2529513  1.0652325
## Orta Afrika-Doğu Afrika  -2.03531532 -3.4060561 -0.6645745
## Orta Afrika-Güney Afrika -1.94145594 -2.7910103 -1.0919015
```

Grafiksel olarak aşağıdaki şekilde gösterilebilir.

```
DTK.plot(ikili.karsilastirma)
```




## 9.2 Bağımlı Örnekler için Tek Yönlü ANOVA Testi

Eşlenik örnek t-Testinin k Örnekteki karşılığıdır. Art arda ölçülen değerlerin ortalamaları arasında fark olup olmadığı ölçmek amacıyla kullanılır. Örnek olarak, beyin hasarı görmüş insanların, dikkat ağları ile ilgili yapılan bir araştırmada elde edilen ölçümlere göre farklı dikkat ağlarının tepki sürelerini karşılaştırılması istensin. Karşılaştırmanın yapılabilmesi için düzenlenen deneyde, denekler aynı testi iki oturumda tekrarlamışlar ve iki oturum arasında her deneğe bir renklerle ilgili 10 dakika süren bir alıştırma yapılmıştır. Burada, deneklerin ortalama cevap sürelerinin dikkat ağlarına verilen uyartılara bağımlı olup olmadığını ölçmek için bağımlı örnekler için tek yönlü ANOVA testi uygulanır. Aşağıdaki R kodunu inceleyebilirsiniz.

```
veri = read.csv2('sinir.agi.csv')

library(ez)
aov.ornek = ezANOVA(data=veri, dv=cevap.suresi, wid=sira.no,
                    within = ipucu)
print(aov.ornek)
```

```
## $ANOVA
##   Effect DFn DFd        F            p p<.05       ges
## 2  ipucu   3  57 452.3171 1.076345e-39     * 0.9401073
##
## $`Mauchly's Test for Sphericity`
```

```
##   Effect         W         p p<.05
## 2  ipucu 0.8765435 0.8013532
##
## $`Sphericity Corrections`
##   Effect       GGe        p[GG] p[GG]<.05      HFe        p[HF] p[HF]<.05
## 2  ipucu 0.9270926 5.722687e-37         * 1.102133 1.076345e-39         *
```

Çıktıya dikkatlice baktığınızda 3 farklı test sonucunu göreceksiniz. İlk olarak, size iki örnek bağımsız örnek için ANOVA tablosu gösterilmektedir. Bu tablo, varyans homojenliği şartı sağlandığında kullanacağınız değerleri içermektedir. İkinci tabloda, size varyans homojenliği hakkında bilgi vermektedir. Bağımlı örnekler için bir test gerçekleştirdiğimizden, burada sadece varyans homojenliği söz konusu değildir. Kovaryanslarında varyans homojenliği üzerine olan etkisinin karşılaştırılması gereklidir. Bu sebeple, burada levene testi yerine Mauchly'nin Küresellik testi kullanılır. Bu test size ilk tablodaki hangi satırları kullanacağınızı söyler. Örnekte, varyans homojenliği sağlandığı için ilk tablodaki satırlar kullanılabilir. Üçüncü tablo ise varyans homojenliği sağlanmadığı durumda, kullanmanız gereken değerleri sunmaktadır. Burada genellikle GG değerlerinin kullanılması tavsiye edilmekle beraber, GGe değerinin 0.75'ten büyük olduğu durumda, p[GG] veya p[HF] değerlerinden istediğinizi kullanabilirsiniz.

## 9.3 İki Yönlü ANOVA Testi

Bu test ile bağımlı değişkenin birdan fazla bağımsız değişkene göre gruplandırılması söz konusudur. Kullanımı, tek yönlü anova ile hemen hemen aynıdır. Örneğin, üniversite öğrencileri arasında gerçekleştirilen bir araştırmada, cinsiyet ve spor yapmanın kızgınlık ifadesine olan etkisi ölçülmek istensin. Aşağıdaki R Kodunu inceleyiniz.

```r
library(foreign)

veri = read.dbf('ikitablo.dbf')

aov.ornek = aov(Anger_Expr~Gender*Sports, data=veri)

summary(aov.ornek)
```

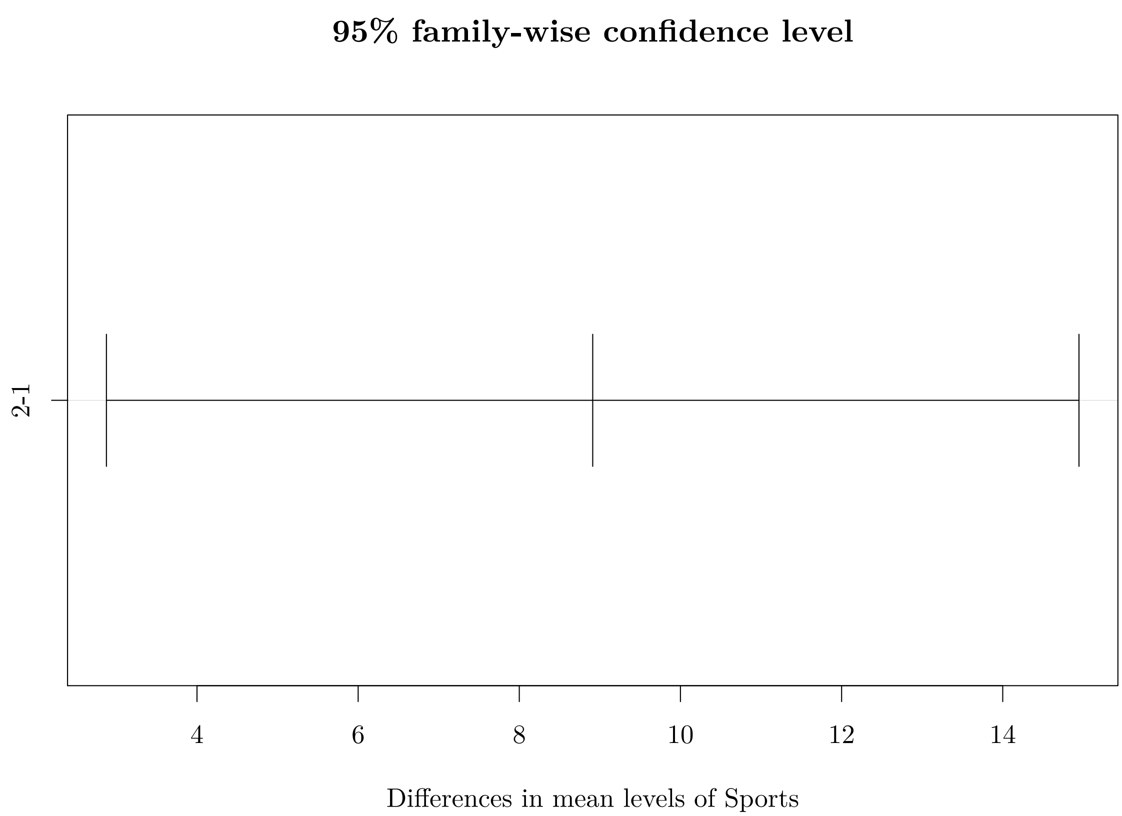
```
##               Df Sum Sq Mean Sq F value  Pr(>F)
## Gender         1      1     1.4   0.009 0.92598
## Sports         1   1357  1357.2   8.709 0.00424 **
## Gender:Sports  1      5     5.2   0.034 0.85505
## Residuals     74  11532   155.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Örneğe göre sadece spor faktörünün kızgınlık ifadesi üzerinde etkisi anlamlı farklılıklar göstermektedir. Bu durumda Post Hoc testlerini sadece Sport faktörüne göre gerçekleştirebilirsiniz.

```
TukeyHSD(x=aov.ornek, which="Sports")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Anger_Expr ~ Gender * Sports, data = veri)
##
## $Sports
##         diff      lwr      upr     p adj
## 2-1 8.911132 2.875994 14.94627 0.0043509
```

```
plot(TukeyHSD(x=aov.ornek, which="Sports"))
```




Burada da görüldüğü gibi 1 nolu spor grubu ile 2 nolu spor grubu arasındaki fark istatistiksel olarak anlamlıdır.

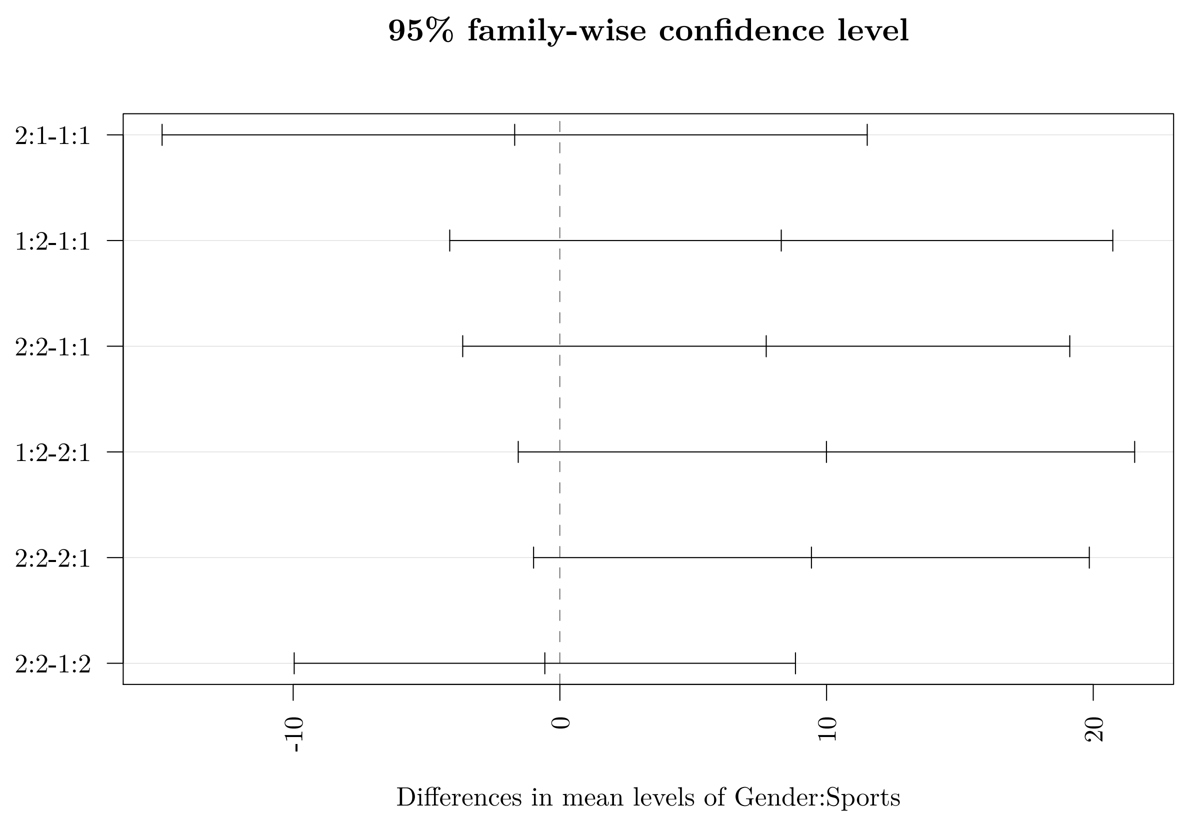
İlk çıktıya dikkat ettiyseniz, çıktıda belirtmediğiniz halde üç faktörün etkisi gözükmektedir. Bunlar, *Sports*, *Gender* ve *Sports:Gender* olarak sıralanmaktadır. Üçüncü faktör, iki değişkenin bileşke etki faktörüdür ve bu faktörde anlamlı bir fark bulunursa, bu faktör ile ilgili ikili fark testleri aşağıdaki şekilde gerçekleştirilebilir.

```
TukeyHSD(x=aov.ornek, which="Gender:Sports")
```

```
##   Tukey multiple comparisons of means
```

```
##      95% family-wise confidence level
##
## Fit: aov(formula = Anger_Expr ~ Gender * Sports, data = veri)
##
## $`Gender:Sports`
##                 diff         lwr       upr     p adj
## 2:1-1:1 -1.6948052 -14.9150197 11.525409 0.9867186
## 1:2-1:1  8.3014354  -4.1298588 20.732730 0.3031425
## 2:2-1:1  7.7379679  -3.6435272 19.119463 0.2877415
## 1:2-2:1  9.9962406  -1.5607567 21.553238 0.1136570
## 2:2-2:1  9.4327731  -0.9867058 19.852252 0.0900007
## 2:2-1:2 -0.5634675  -9.9617941  8.834859 0.9985940
```

```
plot(TukeyHSD(x=aov.ornek, which="Gender:Sports"), las=2)
```




## 9.4 Tek Yönlü MANOVA Testi

Tek yönlü MANOVA testi birden çok bağımlı değişkenin, bağımsız gruplara göre ortalamalarının farkını ölçmek amacıyla kullanılır. Varsayımları tek yönlü ANOVA testi ile aynıdır. Yalnız normallik testini gerçekleştirebilmeniz için, tek değişkenli normallik testleri yerine çok değişkenli normallik testlerini kullanmanız gerekmektedir. Hacettepe üniversitesi tarafından geliştirilmiş olan **MVN** kütüphanesini kullanabilirsiniz. Bu kütüphanede 3 adet çok değişkenli normal dağılım test fonksiyonu bulunmaktadır; bunlar: *mardiaTest*, *hzTest* ve *roystonTest* fonksiyonlarıdır.

Örnek olarak, bir tarım ürününün verimini, kuru ağırlığını ve optik yoğunluğunu ölçmek için yapılan deneylerde sıcaklık ve gübre etken maddesi ölçülmüştür. İlk olarak, değişkenkerin çok boyutlu normal dağılıma sahip olup olmadıklarına bakalım. Aşağıdaki R kodunu inceleyiniz.

```
library(MVN)
veri = read.dbf('manova.dbf')

t = mardiaTest(veri[c('Dry_weight', 'Optical_de', 'Product_yi')])
print(t)
```

```
##    Mardia's Multivariate Normality Test
## ---------------------------------------
##    data : veri[c("Dry_weight", "Optical_de", "Product_yi")]
##
##    g1p            : 4.359416
##    chi.skew       : 87.18832
##    p.value.skew   : 1.929952e-14
##
##    g2p            : 24.37664
##    z.kurtosis     : 9.376643
##    p.value.kurt   : 0
##
##    chi.small.skew : 90.4898
##    p.value.small  : 4.282393e-15
##
##    Result          : Data are not multivariate normal.
## ---------------------------------------
```

```
t = hzTest(veri[c('Dry_weight', 'Optical_de', 'Product_yi')])
print(t)
```

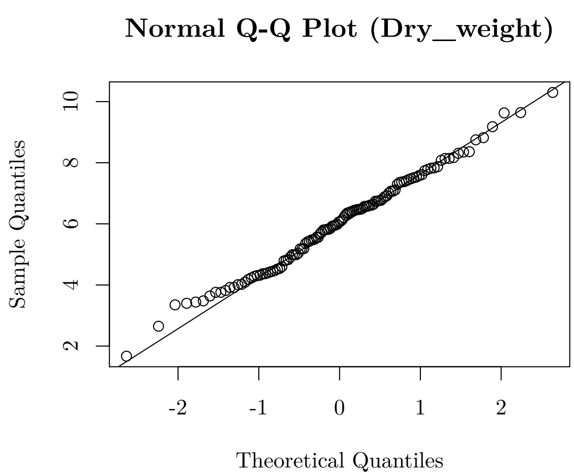
```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : veri[c("Dry_weight", "Optical_de", "Product_yi")]
##
##   HZ      : 1.164483
##   p-value : 0.007373855
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

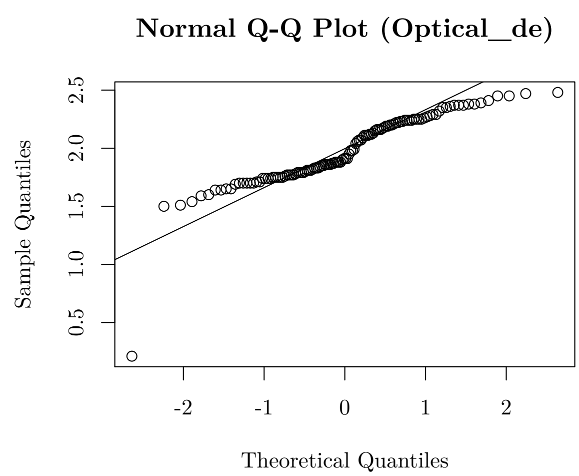
```r
t = roystonTest(veri[c('Dry_weight', 'Optical_de', 'Product_yi')])
print(t)
```

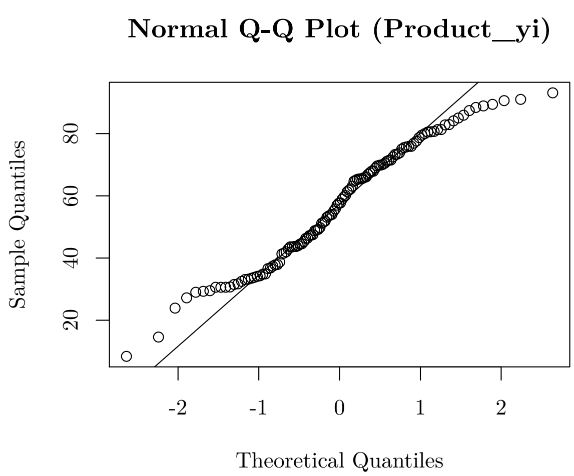
```
##   Royston's Multivariate Normality Test
## ---------------------------------------------
##   data : veri[c("Dry_weight", "Optical_de", "Product_yi")]
##
##   H        : 40.81487
##   p-value : 6.40649e-09
##
##   Result  : Data are not multivariate normal.
## ---------------------------------------------
```

Ayrıca, bu durum grafiksel olarakta incelenebilir.

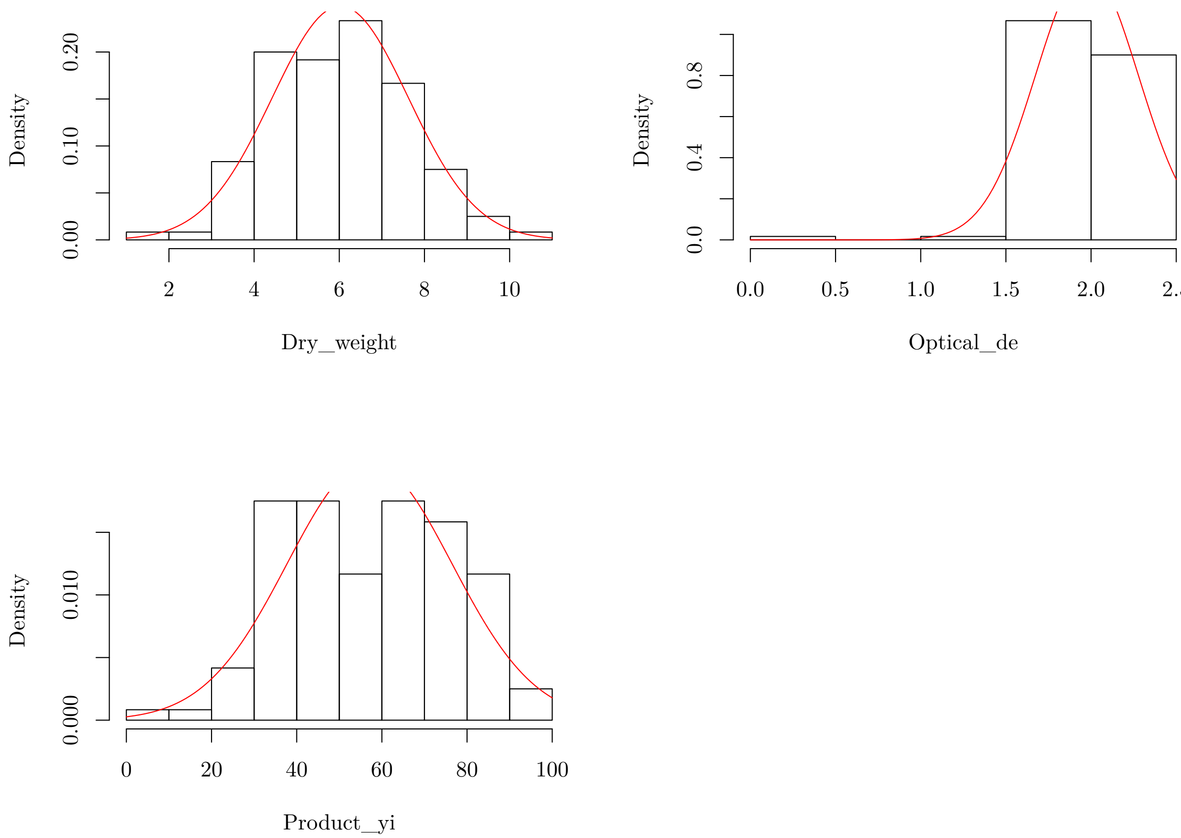
```r
uniPlot(veri[c('Dry_weight', 'Optical_de', 'Product_yi')],
        type = "qqplot")
```



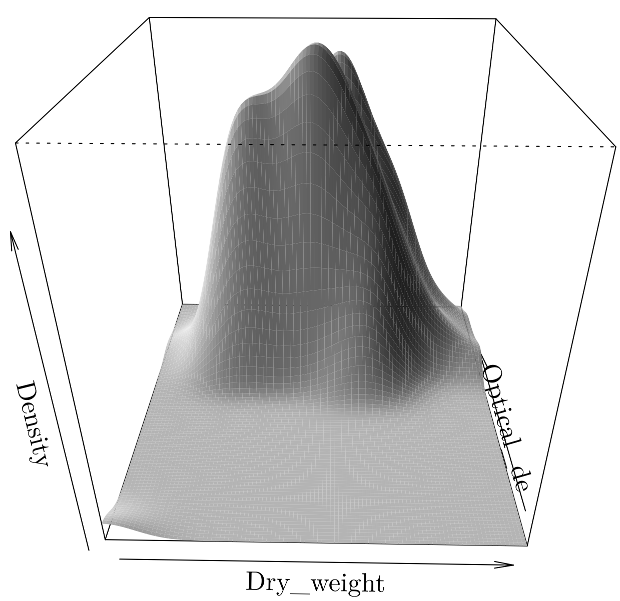



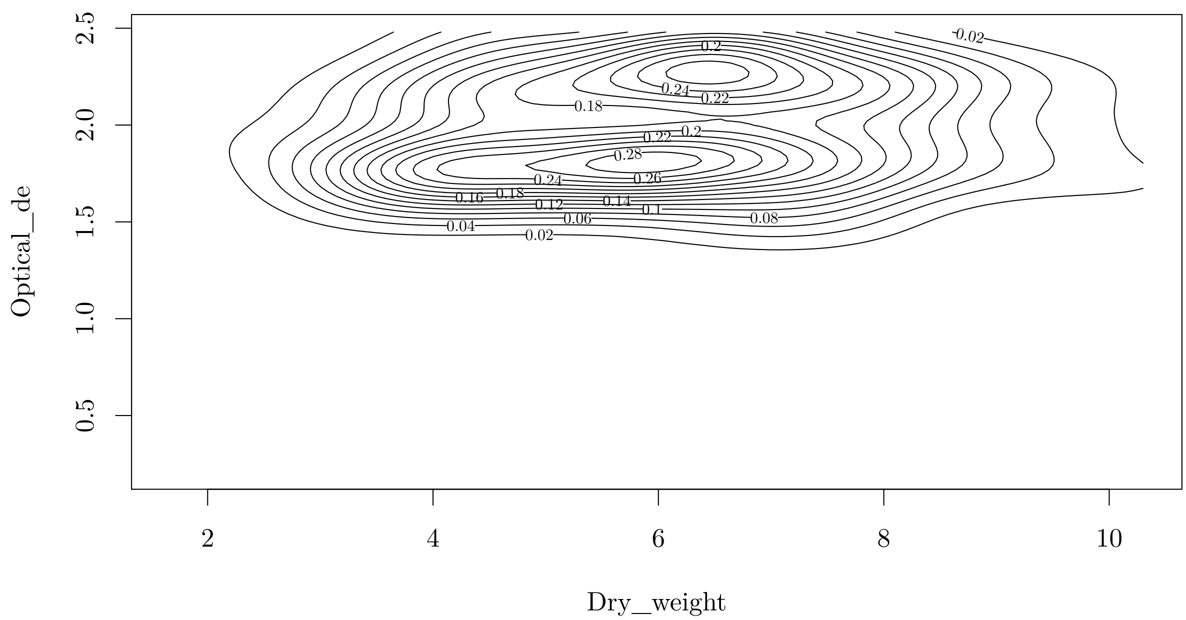
```
uniPlot(veri[c('Dry_weight', 'Optical_de', 'Product_yi')],
        type = "histogram")
```




```
result = hzTest(veri[c('Dry_weight', 'Optical_de')])
mvnPlot(result, type="persp", default = TRUE)
```

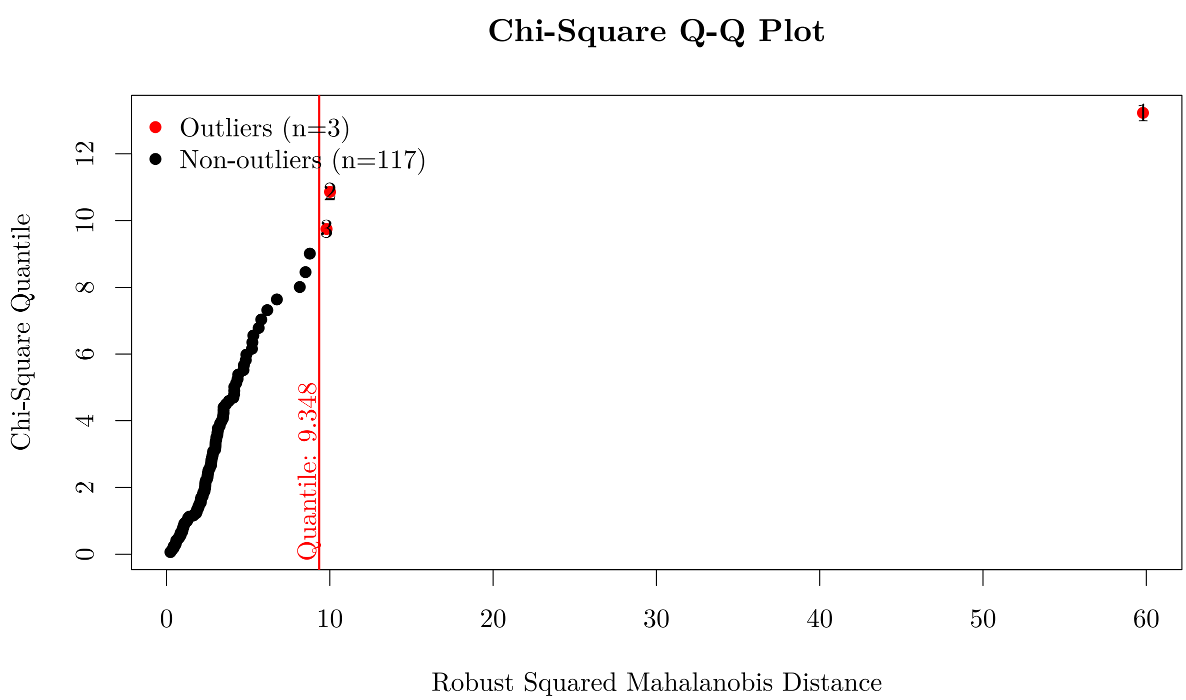


```r
result = hzTest(veri[c('Dry_weight', 'Optical_de')])
mvnPlot(result, type="contour", default = TRUE)
```




MANOVA aykırı değerlerden çok fazla etkilendiği için bu değerlerin bulunması ve gerekliyse veriden ayıklanması önemlidir. Bu sebeple yine **MVN** kütüphanesindeki fonksiyonlar kullanılabilir.

```r
result <- mvOutlier(veri[c('Dry_weight', 'Optical_de', 'Product_yi')],
                    qqplot = TRUE, method = "quan")
```



```
print(result)
```

Örneğe göre, verimiz normal olmamasına rağmen uygulama olması açısından manova testi gerçekleştirilecektir. Tek yönlü manova testinde birden çok bağımlı değişken ve tek faktör değişkeni mevcuttur. Aşağıdaki R kodunu inceleyiniz.

```
man.ornek = manova(cbind(Product_yi, Optical_de)~Temperatur, data=veri)

summary(man.ornek)
```

```
##              Df   Pillai approx F num Df den Df Pr(>F)
## Temperatur    2 0.025337  0.75063      4    234 0.5585
## Residuals   117
```

Bu çıktıda, MANOVA testi için *Pillai* düzeltmesi yapılmıştır. Alternatifleriniz aşağıda belirtilmiştir.

```
summary(man.ornek, test="Pillai")
```

```
##              Df   Pillai approx F num Df den Df Pr(>F)
## Temperatur    2 0.025337  0.75063      4    234 0.5585
## Residuals   117
```

```
summary(man.ornek, test="Wilks")
```

```
##              Df  Wilks approx F num Df den Df Pr(>F)
## Temperatur    2 0.9747  0.74801      4    232 0.5602
## Residuals   117
```

```
summary(man.ornek, test="Hotelling-Lawley")
```

```
##              Df Hotelling-Lawley approx F num Df den Df  Pr(>F)
## Temperatur    2         0.025924  0.74533      4    230   0.562
## Residuals   117
```

```
summary(man.ornek, test="Roy")
```

```
##              Df     Roy approx F num Df den Df Pr(>F)
## Temperatur    2 0.02448   1.4321      2    117  0.243
## Residuals   117
```

Örneğin detaylı çıktısı için *summary.aov* fonksiyonu kullanılmalıdır.

```
summary.aov(man.ornek)
```

```
##  Response Product_yi :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Temperatur    2     81   40.46   0.106 0.8995
## Residuals   117  44659  381.70
##
##  Response Optical_de :
##              Df  Sum Sq  Mean Sq F value Pr(>F)
## Temperatur    2  0.1656 0.082808  0.9035  0.408
## Residuals   117 10.7237 0.091656
```

## 9.5 İki Yönlü MANOVA Testi

İki yönlü ANOVA testinin, birden çok bağımlı değişkene sahip halidir. Bu sebeple, ANOVA'nın bütün varsayımlarına sahiptir. İki yönlü ANOVA testi için kullandığımız örneği, değiştirip bağımlı değişkenlerin sayısını 2'ye çıkartalım. Aşağıdaki R kodunu inceleyiniz.

```
library(foreign)

veri = read.dbf('ikitablo.dbf')

man.ornek = manova(cbind(Anger_In, Anger_Out)~Gender*Sports, data=veri)

summary(man.ornek)
```

```
##               Df   Pillai approx F num Df den Df Pr(>F)
## Gender         1 0.015597   0.5783      2     73 0.5634
## Sports         1 0.100906   4.0964      2     73 0.0206 *
## Gender:Sports  1 0.000101   0.0037      2     73 0.9963
## Residuals     74
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Detaylı farklarda aşağıdaki gibi olacaktır.

```
summary.aov(man.ornek)
```

```
##  Response Anger_In :
##               Df  Sum Sq Mean Sq F value  Pr(>F)
## Gender         1   10.16  10.155  0.4849 0.48839
## Sports         1  139.04 139.037  6.6389 0.01197 *
## Gender:Sports  1    0.10   0.096  0.0046 0.94631
```

```
## Residuals     74 1549.75  20.943
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response Anger_Out :
##               Df  Sum Sq Mean Sq F value Pr(>F)
## Gender         1   11.69 11.6926  0.6498 0.4228
## Sports         1   26.18 26.1827  1.4550 0.2316
## Gender:Sports  1    0.05  0.0478  0.0027 0.9590
## Residuals     74 1331.62 17.9948
```

Detaylı incelememize göre *Anger_In* değişkeninin *Sports* değişkenine farkı istatistiksel olarak anlamlıdır.

## 9.6 Kruskal Wallis Testi

Kruskall-Wallis testi tek yönlü ANOVA testinin parametrik olmayan halidir. Grupların Normal dağılması varsayımını göz ardı etmesine rağmen, gruplara ait dağılımların şekillerinin aynı olması varsayımı bulunmaktadır. Bu sebeple, bu testi yapmaya geçmeden önce grup dağılımlarının benzer şekilli olmasını test etmelisiniz. Aşağıdaki R kodlarını inceleyin.

```r
veri = read.csv2('fildisi.csv')
kw.ornek = kruskal.test(delta13C~bolge, data = veri)

print(kw.ornek)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  delta13C by bolge
## Kruskal-Wallis chi-squared = 108.33, df = 4, p-value < 2.2e-16
```

Örnekte de gördüğünüz gibi, gruplar arasında anlamlı farklılıklar bulunmaktadır. Bu durumda, Kruskal-Wallis PostHoc testleri ile farkın hangi gruplar arasında olduğunun incelenmesi gerekmektedir. Aşağıda, R ile kullanabileceğiniz Kruskal-Wallis Post Hoc testlerinin bir listesi bulunmaktadır.

- Dunn Testi (Burada method yerine FWER'deki bütün değerleri kullanabilirsiniz.)

```r
library(FSA)
t=dunnTest(delta13C~bolge, data = veri, method = "bonferroni")
print(t)
```

```
##                  Comparison          Z       P.unadj         P.adj
## 1          Asya - Batı Afrika  2.1374935 3.255788e-02 3.255788e-01
## 2          Asya - Doğu Afrika -0.8541105 3.930438e-01 1.000000e+00
## 3   Batı Afrika - Doğu Afrika -5.5522548 2.820082e-08 2.820082e-07
## 4         Asya - Güney Afrika -0.6928241 4.884199e-01 1.000000e+00
## 5  Batı Afrika - Güney Afrika -7.7345599 1.037614e-14 1.037614e-13
## 6  Doğu Afrika - Güney Afrika  0.4801572 6.311156e-01 1.000000e+00
## 7          Asya - Orta Afrika  1.7741102 7.604492e-02 7.604492e-01
## 8   Batı Afrika - Orta Afrika -0.9962281 3.191393e-01 1.000000e+00
## 9   Doğu Afrika - Orta Afrika  5.2160090 1.828193e-07 1.828193e-06
## 10 Güney Afrika - Orta Afrika  8.1281850 4.357658e-16 4.357658e-15
```

- Nemenyi Testi (Burada dist değişkenine "tukey" veya "chisq" değerlerini verebilirsiniz)

```r
library(DescTools)
t = NemenyiTest(x = veri$delta13C, g=veri$bolge, dist = "tukey")
print(t)
```

```
##
##  Nemenyi's test of multiple comparisons for independent samples (tukey)
##
##                          mean.rank.diff    pval
## Batı Afrika-Asya             -114.17120  0.2043
## Doğu Afrika-Asya               47.62669  0.9135
## Güney Afrika-Asya              35.56394  0.9581
## Orta Afrika-Asya              -92.64583  0.3889
## Doğu Afrika-Batı Afrika       161.79788 2.8e-07 ***
## Güney Afrika-Batı Afrika      149.73513 1.3e-13 ***
## Orta Afrika-Batı Afrika        21.52536  0.8573
## Güney Afrika-Doğu Afrika      -12.06275  0.9892
## Orta Afrika-Doğu Afrika      -140.27252 1.8e-06 ***
## Orta Afrika-Güney Afrika     -128.20977 6.6e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- İkili Mann-Whitney U Testleri

```r
t = pairwise.wilcox.test(veri$delta13C, veri$bolge,
                         p.adjust.method = "bonferroni")
print(t)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
```

```
## data:  veri$delta13C and veri$bolge
##
##              Asya Batı Afrika Doğu Afrika Güney Afrika
## Batı Afrika  0.43 -           -           -
## Doğu Afrika  1.00 4.0e-06     -           -
## Güney Afrika 1.00 < 2e-16     1.00        -
## Orta Afrika  0.57 1.00        9.1e-05     4.1e-13
##
## P value adjustment method: bonferroni
```

## 9.7 Friedman Testi

Bu test iki yönlü ANOVA testinin parametrik olmayan halidir. Kruskal-Wallis ile aynı varsayımlara sahiptir. Ayrıca, deneyin tam anlamıyla dengeli olmasını yani her grupta eşit eleman ve en az iki grup olmasını gerektirmektedir. Örneğin, öğrencilerin dersi sevmeleri ders veren öğretim elemanları ve sınavları okuyan kişilere göre değişip değişmediği ölçülmek istensin. Bu test aşağıdaki şekilde R ile gerçekleştirilebilir.

```
library(foreign)
veri = read.dbf('ikitablo.dengeli.dbf')
t=friedman.test(Likert~Instructor|Rater, data = veri)
print(t)
```

```
##
##  Friedman rank sum test
##
## data:  Likert and Instructor and Rater
## Friedman chi-squared = 23.139, df = 4, p-value = 0.0001188
```

Bu test için kullanacağınız PostHoc Testlerde aşağıdaki gibidir.

- İkili İşaret Testi (Her faktör için ayrı ayrı gerçekleştirilir.)

```
library(rcompanion)
t = pairwiseSignMatrix(Likert ~ Instructor,
                       data   = veri,
                       method = "bonferroni")
print(t)
```

```
## $Unadjusted
##                 Bob Belcher Gene Belcher Linda Belcher Louise Belcher
## Bob Belcher              NA       0.6875      0.015630       0.070310
## Gene Belcher             NA           NA      0.007813       0.007813
## Linda Belcher            NA           NA            NA       0.375000
```

```
## Louise Belcher           NA           NA             NA             NA
## Tina Belcher             NA           NA             NA             NA
##                  Tina Belcher
## Bob Belcher         0.031250
## Gene Belcher        0.007813
## Linda Belcher       0.625000
## Louise Belcher      0.218700
## Tina Belcher              NA
##
## $Method
## [1] "bonferroni"
##
## $Adjusted
##                  Bob Belcher Gene Belcher Linda Belcher Louise Belcher
## Bob Belcher           1.0000      1.00000       0.15630        0.70310
## Gene Belcher          1.0000      1.00000       0.07813        0.07813
## Linda Belcher         0.1563      0.07813       1.00000        1.00000
## Louise Belcher        0.7031      0.07813       1.00000        1.00000
## Tina Belcher          0.3125      0.07813       1.00000        1.00000
##                  Tina Belcher
## Bob Belcher          0.31250
## Gene Belcher         0.07813
## Linda Belcher        1.00000
## Louise Belcher       1.00000
## Tina Belcher         1.00000
```

```
t = pairwiseSignMatrix(Likert ~ Rater,
                       data   = veri,
                       method = "bonferroni")
print(t)
```

```
## $Unadjusted
##    a     b     c      d      e      f      g      h
## a NA 0.375 1.000 1.0000 0.2500 0.6250 0.6250 0.3750
## b NA    NA 0.375 0.0625 0.0625 0.0625 0.0625 0.0625
## c NA    NA    NA 0.5000 0.1250 0.3750 0.2500 0.1250
## d NA    NA    NA     NA 0.5000 0.6250 0.2500 0.2500
## e NA    NA    NA     NA     NA 1.0000 1.0000 0.6250
## f NA    NA    NA     NA     NA     NA 1.0000 1.0000
## g NA    NA    NA     NA     NA     NA     NA 0.6250
## h NA    NA    NA     NA     NA     NA     NA     NA
##
## $Method
## [1] "bonferroni"
```

```
##
## $Adjusted
##   a b c d e f g h
## a 1 1 1 1 1 1 1 1
## b 1 1 1 1 1 1 1 1
## c 1 1 1 1 1 1 1 1
## d 1 1 1 1 1 1 1 1
## e 1 1 1 1 1 1 1 1
## f 1 1 1 1 1 1 1 1
## g 1 1 1 1 1 1 1 1
## h 1 1 1 1 1 1 1 1
```

- Conover Testi

```
library(PMCMR)
t = posthoc.friedman.conover.test(y      = veri$Likert,
                                   groups = veri$Instructor,
                                   blocks = veri$Rater,
                                   p.adjust.method="bonferroni")

print(t)
```

```
##
##  Pairwise comparisons using Conover's test for a two-way
##                     balanced complete block design
##
## data:  veri$Likert , veri$Instructor and veri$Rater
##
##                Bob Belcher Gene Belcher Linda Belcher Louise Belcher
## Gene Belcher   1.00000     -            -             -
## Linda Belcher  0.00014     2.3e-06      -             -
## Louise Belcher 5.7e-06     1.1e-07      1.00000       -
## Tina Belcher   0.00223     3.5e-05      1.00000       0.36077
##
## P value adjustment method: bonferroni
```

## 9.8 Welch ve Brown-Forsythe Testleri

ANOVA testinin varyans homojenliği şartı sağlanmadığı durumlarda, kruskall-wallis testine alternatif güçlü (robust) bir alternatiftir. Welch testi, test istatistiğinin serbestlik derecesini değiştirirken, Brown-Forsythe ise farklı bir F istatistiği kullanmaktadır. Yapılan araştırmalarda Welch testinin istatistiksel gücünün fazla olduğu bulunmuştur. Bu testlerin R ile kullanımları aşağıdaki şekilde verilmiştir.

```r
library(onewaytests)
veri = read.csv2('fildisi.csv')
t = bf.test(veri$delta13C, veri$bolge)
```

```
##
##   Brown-Forsythe Test
## ---------------------------------------------------------
##   data : veri$delta13C and veri$bolge
##
##   statistic  : 18.68663
##   num df     : 4
##   denom df   : 26.17584
##   p.value    : 2.282741e-07
##
##   Result     : Difference is statistically significant.
## ---------------------------------------------------------
```

```r
t = welch.test(veri$delta13C, veri$bolge)
```

```
##
##   Welch's Heteroscedastic F Test
## ---------------------------------------------------------
##   data : veri$delta13C and veri$bolge
##
##   statistic  : 28.55471
##   num df     : 4
##   denom df   : 43.38833
##   p.value    : 1.175298e-11
##
##   Result     : Difference is statistically significant.
## ---------------------------------------------------------
```

Bu testleri gerçekleştirdikten sonra, kullanılacak post hoc testler, varyans homojenliği sağlanmadığı durumda kullanılacak ANOVA post hoc testleri ile aynıdır.

BÖLÜM

# 10

# Uygulama: New York Belediyesi Eğitim Durumu Değerlendirmesi

New York Belediyesinin, şehir genelinde düzenlenen İngilizce ve Sanat Sınavı ile ilgili istatistikleri içeren sınavına ait veriler *sitemizde* verilmiştir. Verinin yapısı aşağıdaki gibidir.

| Alan Adı | Açıklama |
|---|---|
| Okul.Adi | Okul Adı |
| Sinif | Sınıf |
| Yıl | Sınav Yılı |
| Okul.Tur | Okul Türü |
| OgrenciSayisi | Sınava Katılan Öğrenci Sayısı |
| Ortalama.Puan | Öğrencilerin Ortalama Başarı Puanları |
| Seviye.1.Sayi | Sınav Sonucu 1. Seviye Öğrenci Sayısı |
| Seviye.1.Yuzde | Sınav Sonucu 1. Seviye Öğrenci Yüzdesi |
| Seviye.2.Sayi | Sınav Sonucu 2. Seviye Öğrenci Sayısı |
| Seviye.2.Yuzde | Sınav Sonucu 2. Seviye Öğrenci Yüzdesi |
| Seviye.3.Sayi | Sınav Sonucu 3. Seviye Öğrenci Sayısı |
| Seviye.3.Yuzde | Sınav Sonucu 3. Seviye Öğrenci Yüzdesi |
| Seviye.4.Sayi | Sınav Sonucu 4. Seviye Öğrenci Sayısı |
| Seviye.4.Yuzde | Sınav Sonucu 4. Seviye Öğrenci Yüzdesi |
| Seviye.3.ve.4.Sayi | Sınav Sonucu 3. ve 4. Seviye Öğrenci Sayısı |
| Seviye.3.ve.4.Yuzde | Sınav Sonucu 3. ve 4. Seviye Öğrenci Yüzdesi |

Bu verileri indirip, aşağıdaki analizlerin sonuçlarını elde etmek için gerekli R komutlarını yazınız.

Aşağıdaki değişkenler için gerekli tabloları oluşturunuz. Veri bir RData dosyası olduğundan aşağıdaki gibi yüklenmelidir.

```
load('NewYorkAnalizUygulama.RData')
```

## 10.1 Uygulama 1: Tanımlayıcı İstatistikler

- Verinin Tanımlayıcı İstatistiklerini Bulan R Kodunu yazınız.
- Yıllara Göre Bütün Sayısal Değişkenlerin Tanımlayıcı İstatistiklerini Bulunuz.
- Sınıflara Göre Bütün Sayısal Değişkenlerin Tanımlayıcı İstatistiklerini Bulunuz.

## 10.2 Uygulama 2: Çapraz Tablolar

- Okul Türü ile Sınıf arasındaki çapraz tabloyu oluşturup, aralarında ilişki olup olmadığını bulunuz.
- Okul Türü ile Yıl arasındaki çapraz tabloyu oluşturup, aralarında ilişki olup olmadığını bulunuz.
- Sınıf ile Yıl arasındaki çapraz tabloyu oluşturup, aralarında ilişki olup olmadığını bulunuz.

## 10.3 Uygulama 3: Tek Örnek Testleri

- Öğrenci sayılarının ortalamasının veya medyanının (hangisi olduğunu siz seçmelisiniz) 20 olup olmadığını bulunuz. Gerekli ise 20'den büyüklük küçüklük hakkında bilgi veriniz.

- Öğrenci ortalamasının veya medyanının 300 olup olmadığını bulunuz. Gerekli ise 300'den büyüklük küçüklük hakkında bilgi veriniz.

- 1. Seviye öğrenci sayılarının ortalamasının veya medyanının 15 olup olmadığını bulunuz. Gerekli ise 15'ten büyüklük küçüklük hakkında bilgi veriniz.

- 1. Seviye öğrenci yüzdelerinin ortalamasının veya medyanının 50 olup olmadığını bulunuz. Gerekli ise 50'den büyüklük küçüklük hakkında bilgi veriniz.

- 2. Seviye öğrenci sayılarının ortalamasının veya medyanının 15 olup olmadığını bulunuz. Gerekli ise 15'ten büyüklük küçüklük hakkında bilgi veriniz.

- 2. Seviye öğrenci yüzdelerinin ortalamasının veya medyanının 50 olup olmadığını bulunuz. Gerekli ise 50'den büyüklük küçüklük hakkında bilgi veriniz.

- 3. Seviye öğrenci sayılarının ortalamasının veya medyanının 10 olup olmadığını bulunuz. Gerekli ise 10'ten büyüklük küçüklük hakkında bilgi veriniz.

- 3. Seviye öğrenci yüzdelerinin ortalamasının veya medyanının 10 olup olmadığını bulunuz. Gerekli ise 10'den büyüklük küçüklük hakkında bilgi veriniz.

- 4. Seviye öğrenci sayılarının ortalamasının veya medyanının 5 olup olmadığını bulunuz. Gerekli ise 5'ten büyüklük küçüklük hakkında bilgi veriniz.

- 4. Seviye öğrenci yüzdelerinin ortalamasının veya medyanının 10 olup olmadığını bulunuz. Gerekli ise 10'den büyüklük küçüklük hakkında bilgi veriniz.

## 10.4 Uygulama 4: İki Örnek Testleri

- 1. Seviye Öğrenci Sayılarının 2. Seviye Öğrenci Sayılarına eşit olup olmadığını bulunuz. Gerekli ise büyüklük ve küçüklük hakkında bilgi veriniz.

- 3. Seviye Öğrenci Sayılarının 4. Seviye Öğrenci Sayılarına eşit olup olmadığını bulunuz. Gerekli ise büyüklük ve küçüklük hakkında bilgi veriniz.

## 10.5 Uygulama 5: Çok Örnek Testeri

- Yıllara Göre Öğrenci ortalamaları arasında fark olup olmadığını inceleyiniz. Farklı olan yılları belirleyiniz.
- Okul Türlerine Göre Öğrenci ortalamaları arasında fark olup olmadığını inceleyiniz. Farklı olan okul türlerini belirleyiniz.
- Sınıflara Göre Öğrenci ortalamaları arasında fark olup olmadığını inceleyiniz. Farklı olan sınıfları belirleyiniz.
- Yıllara Göre 1. seviye öğrenci sayıları ve yüzdeleri arasında fark olup olmadığını inceleyiniz. Farklı olan yılları belirleyiniz.
- Okul Türlerine Göre 2. seviye öğrenci sayıları ve yüzdeleri arasında fark olup olmadığını inceleyiniz. Farklı olan okul türlerini belirleyiniz.
- Sınıflara Göre 3. seviye öğrenci sayıları ve yüzdeleri arasında fark olup olmadığını inceleyiniz. Farklı olan sınıfları belirleyiniz.
- Okul Türlerine Göre 4. seviye öğrenci sayıları ve yüzdeleri arasında fark olup olmadığını inceleyiniz. Farklı olan sınıfları belirleyiniz.

Aynı istatistiksel işlemleri New York Belediyesinin matematik sınavlarınada uygulayınız.

BÖLÜM

# 11

# Korelasyon ve Regresyon Analizi

İstatistik hemen hemen bütün bilim dallarında kullanılmaktadır. Bazı araştırmalarda, birden fazla değişkenin arasındaki ilişkinin incelenmesi gerekmektedir. Bu inceleme sonucunda beklenen değişkenler arasındaki ilişkinin yönü, kuvveti ve yapısının ortaya çıkarılmasıdır. Bu amaçla kullanılması gereken istastistiksel yöntemler regresyon ve korelasyon analizidir.

## 11.1 Korelasyon

Korelasyon ile iki değişkenin değişimlerinin birbirleri ile bağlantılı olup olmadığı araştırılır. Bu bağlantının yönü ve kuvveti *Korelasyon Katsayısı* ile belirlenir. Burada dikkat edilmesi gereken en önemli nokta, değişkenler arasındaki ilişkinin doğrusal olması gerektiğidir. Doğrusal olmayan bir ilişkinin korelasyonundan bahsedilemez.

### 11.1.1 Korelasyon Katsayısının Hesaplanması

Korelasyon iki değişkenin değişiminin bağlantısını ifade ettiğinden, ilk olarak bir değişkenin değişiminin nasıl incelendiğine bakalım. Burada, bir değişkenin değişiminin incelenmesi amacıyla, değişkenin aldığı değerlerin ortalama etrafındaki dağılımını ölçen varyans ölçüsü kullanılır. Bu tanımdan yola çıkarak, iki değişkene ait değerlerin ortalamarı etrafındaki dağılımın incelenmesi için kovaryans (covariance) ölçüsü kullanılmaktadır. Bu ölçü, *Cov(X,Y)* şeklinde veya $\sigma_{xy}$ şeklinde gösterilir ve aşağıdaki şekilde hesaplanır.

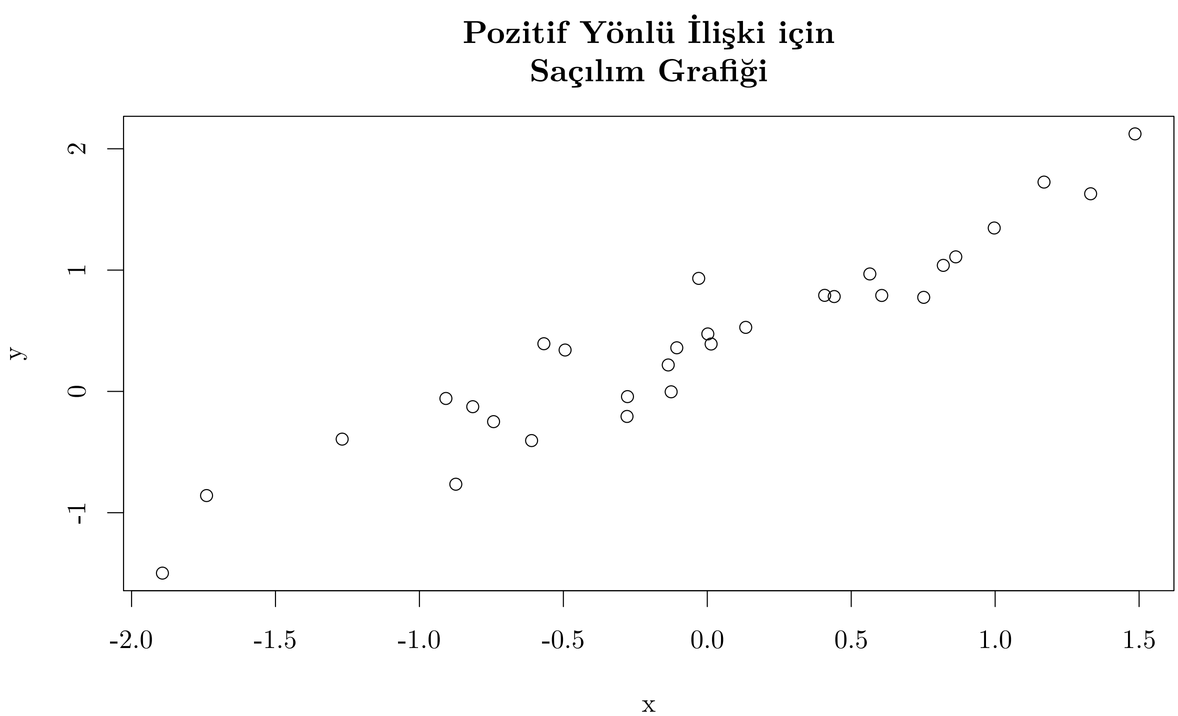
$$Cov(X,Y) = \sigma_{XY} = \frac{\sum_{i=1}^{N} \left(X_i - \mu_x\right)\left(Y_i - \mu_y\right)}{N}$$

Burada hesaplanan kovaryans anakütle kovaryansı olup, örneklemden hesaplanan yansız kovaryans formülü aşağıdaki gibi olmaktadır.

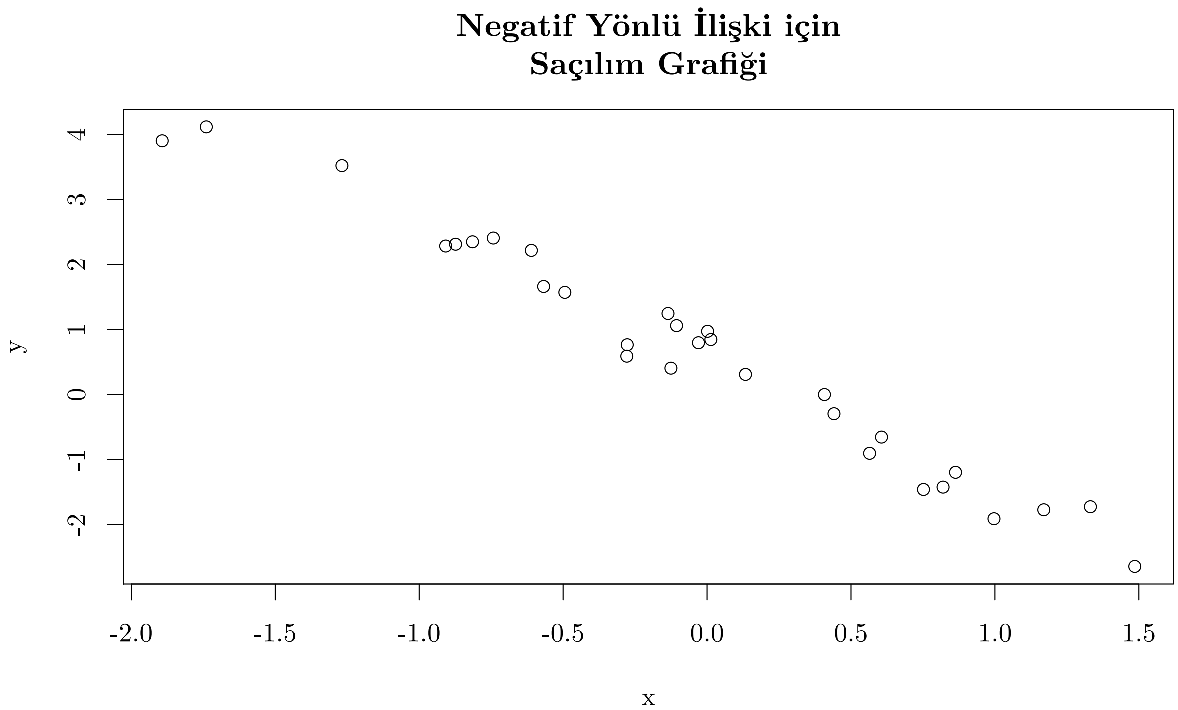
$$cov(x,y) = S_{xy} = \frac{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{n-1}$$

Ancak kovaryans sadece değişken değişimlerinin yönü hakkında bilgi verir. Eğer iki değişken arasındaki kovaryans 0 ise, *değişkenlerin değişimi birbirlerinden bağımsızdır*. Aksi durumlarda, iki değişken arasındaki kovaryans pozitif ise, *değişkenlerin değişimi birbirlerini doğru yönlü olarak (yani bir artarken öbürü de artar)*, veya iki değişken arasındaki kovaryans negatif ise, *değişkenlerin değişimi birbirlerini test yönlü olarak (yani biri artarken öbürü azalır)* etkilemektedir. Bu durumu en iyi şekilde saçılım grafiği ile görselleştirebilirsiniz. Aşağıda R ile iki değişken arasındaki saçılım grafiğinin oluşturulması için gerekli kodlar verilmiştir.

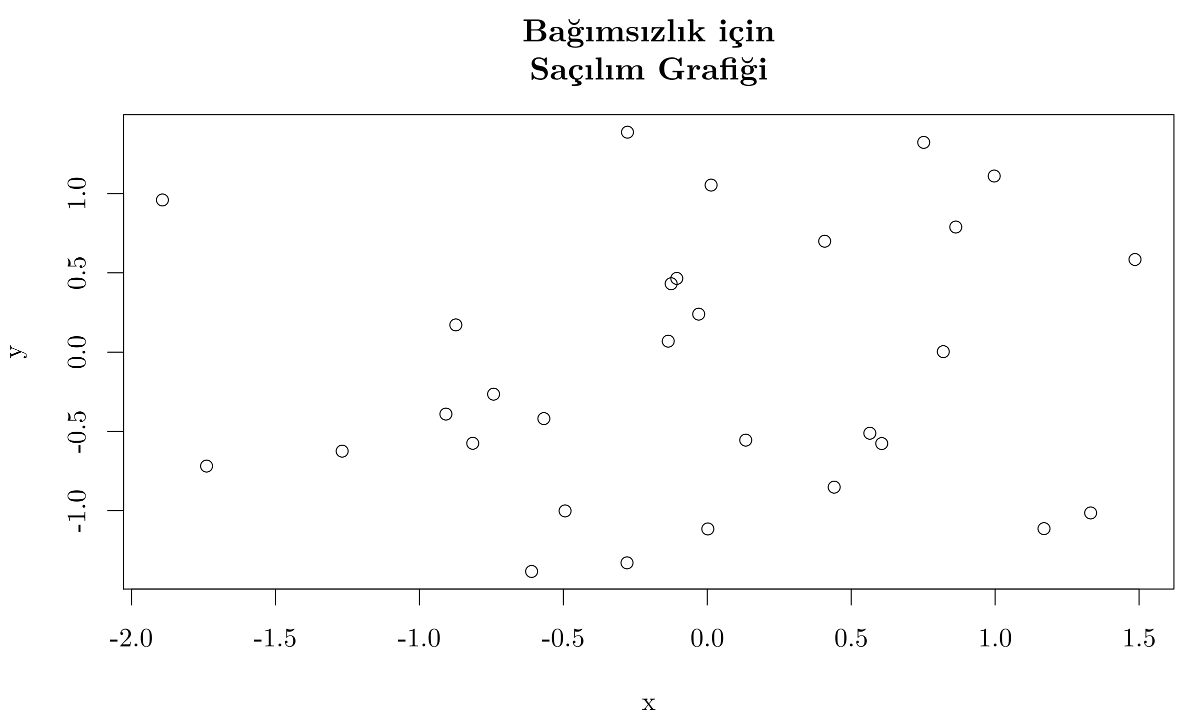
```r
set.seed(127)

x = rnorm(30)
e = runif(30)

y = x+e

plot(x,y, main = "Pozitif Yönlü İlişki için\nSaçılım Grafiği")
```




```r
y = -2*x+runif(30)

plot(x,y, main="Negatif Yönlü İlişki için\nSaçılım Grafiği")
```




```r
y = rnorm(30)

plot(x,y, main="Bağımsızlık için\nSaçılım Grafiği")
```




Ancak kovaryans değişkenler arası ilişkinin gücünü ölçmek amacıyla kullanılabilecek bir **ölçü** değildir. Bu amaçla kullanılmak üzere korelasyon katsayısı önerilmiştir. Korelasyon katsayısı, kovaryans ifadesinin normalleştirilmesi sonucu elde edilen bir ifadedir ve aşağıdaki şekilde hesaplanır.

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

Bu ifadede, $\rho_{XY}$ $X$ ve $Y$ değişkenleri arasındaki korelasyon katsayısını ifade etmektedir. Bu ifade, ana kütle korelasyonu için kullanılacak bir ifade olup, örneklem için hesaplanacak korelasyon katsayısı için $\sigma_{XY}$ yerine $S_{xy}$, $\sigma_X$ yerine $S_x$ ve $\sigma_Y$ yerine de $S_y$ konulur.

$$S_{xy} = \frac{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{n-1}$$

$$S_x = \sqrt{\frac{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)^2}{n-1}}$$

$$S_y = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \overline{y}\right)^2}{n-1}}$$

$$r = \frac{S_{xy}}{S_x S_y} = \frac{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{\sqrt{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)^2 \sum_{i=1}^{n} \left(y_i - \overline{y}\right)^2}}$$

Son formülde yer alan $r$ değişkeni örneklem için hesaplanan korelasyon katsayısıdır. Gerekli düzenlemeler yapıldığı takdirde $r$ kaysayısı aşağıdaki şekilde hesaplanabilir duruma gelecektir.

$$r = \frac{\sum_{i=1}^{n} x_i y_i - n\overline{x}\overline{y}}{\sqrt{\left(\sum_{i=1}^{n} x_i^2 - n\overline{x}^2\right)\left(\sum_{i=1}^{n} y_i^2 - n\overline{y}^2\right)}}$$

Korelasyon katsayısı -1 ile +1 arasında değer alır ($-1 \le r \le +1$) ve aşağıdaki şekilde yorumlanır.

- r=0 ise, değişkenler arasında ilişki bulunmamaktadır.
- r<0 ise, değişkenler arasında negatif yönlü bir ilişki bulunmaktadır.
- r>0 ise, değişkenler arasında pozitif yönlü bir ilişki bulunmaktadır.

denir. İlişkinin gücü hakkında ise aşağıdaki yorumlarda bulunulabilir.

- $|r| \approx 1$ ise, değişkenler arasında güçlü bir ilişki bulunmaktadır.
- $|r| \approx 0.5$ ise, değişkenler arasında orta dereceli bir ilişki bulunmaktadır.
- $|r| \approx 0$ ise, değişkenler arasında zayıf bir ilişki bulunmaktadır.

### 11.1.2 İki Değişken Arasındaki Korelasyonun Test Edilmesi

Örneklemden hesaplanan iki değişkenin korelasyon katsayısının istatistiksel olarak anlamlı olup olmadığı test edilmelidir. Bu amaçla kurulacak hipotez testi aşağıdaki gibidir.

$$H_0 : \rho = 0$$
$$H_1 : \rho \neq 0$$
$$H_1 : \rho > 0$$
$$H_1 : \rho < 0$$

Bu testin gerçekleştirilebilmesi için gerekli test istatistiği aşağıdaki şekildedir.

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$

Test istatistiğinden de anlaşılacağı gibi kullanılan dağılım n-2 serbestlik dereceli t-dağılımıdır. Şimdi biraz önce grafiklerini cizdiğimiz değişkenler arasındaki korelasyon katsayısının anlamlılığını test edelim. Bunun için r ile beraber gelen *cor.test* fonksiyonunu kullanacağız. Bu fonksiyon hem parametrik hemde parametrik olmayan korelasyon katsayısı testini içermektedir. Değişkenlerimizin ikisi de normal dağılıma sahip olduğu için parametrik korelasyon katsayısı testi olan *"Pearson Korelasyon Testini"* gerçekleştirelim.

```
set.seed(127)

x = rnorm(30)
e = runif(30)

y = x+e

t = cor.test(x,y,method = "pearson")

if(t$p.value<0.05) {
  # Korelasyon katsayısı 0'dan farklı
  # Acaba sıfırdan büyük mü?
  t = cor.test(x,y,method = "pearson", alternative = "g")
  if(t$p.value<0.05) {
    # Evet Büyük
    cat("Korelasyon katsayısı 0'dan büyüktür r=",
        t$estimate," p=",t$p.value,"\n")
  } else {
    # Hayır Küçük
    cat("Korelasyon katsayısı 0'dan küçüktür r=",
        t$estimate," p=",1-t$p.value,"\n")
  }
} else {
  # Korelasyon katsayısı 0'dan farklı değil!
```

```
  cat("Korelasyon katsayısı 0'dır r=",
      t$estimate," p=", t$p.value,"\n")
}
```

```
## Korelasyon katsayısı 0'dan büyüktür r= 0.9476611  p= 1.027784e-15
```

```
y = -2*x+runif(30)

t = cor.test(x,y,method = "pearson")

if(t$p.value<0.05) {
  # Korelasyon katsayısı 0'dan farklı
  # Acaba sıfırdan büyük mü?
  t = cor.test(x,y,method = "pearson", alternative = "g")
  if(t$p.value<0.05) {
    # Evet Büyük
    cat("Korelasyon katsayısı 0'dan büyüktür r=",
        t$estimate," p=",t$p.value,"\n")
  } else {
    # Hayır Küçük
    cat("Korelasyon katsayısı 0'dan küçüktür r=",
        t$estimate," p=",1-t$p.value,"\n")
  }
} else {
  # Korelasyon katsayısı 0'dan farklı değil!
  cat("Korelasyon katsayısı 0'dır r=",
      t$estimate," p=", t$p.value,"\n")
}
```

```
## Korelasyon katsayısı 0'dan küçüktür r= -0.9851325  p= 0
```

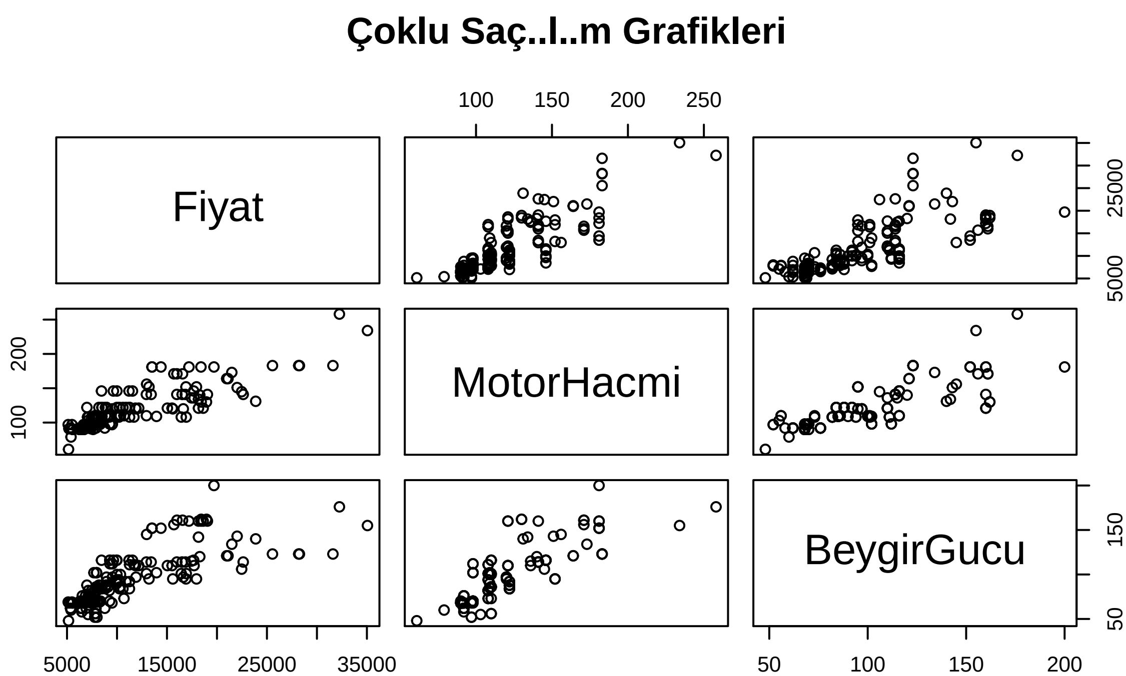
```
y = rnorm(30)

t = cor.test(x,y,method = "pearson")

if(t$p.value<0.05) {
  # Korelasyon katsayısı 0'dan farklı
  # Acaba sıfırdan büyük mü?
  t = cor.test(x,y,method = "pearson", alternative = "g")
  if(t$p.value<0.05) {
    # Evet Büyük
    cat("Korelasyon katsayısı 0'dan büyüktür r=",
        t$estimate," p=",t$p.value,"\n")
  } else {
```

```
    # Hayır Küçük
    cat("Korelasyon katsayısı 0'dan küçüktür r=",
        t$estimate," p=",1-t$p.value,"\n")
  }
} else {
  # Korelasyon katsayısı 0'dan farklı değil!
  cat("Korelasyon katsayısı 0'dır r=",
      t$estimate," p=", t$p.value,"\n")
}
```

```
## Korelasyon katsayısı 0'dır r= 0.1260301  p= 0.5069351
```

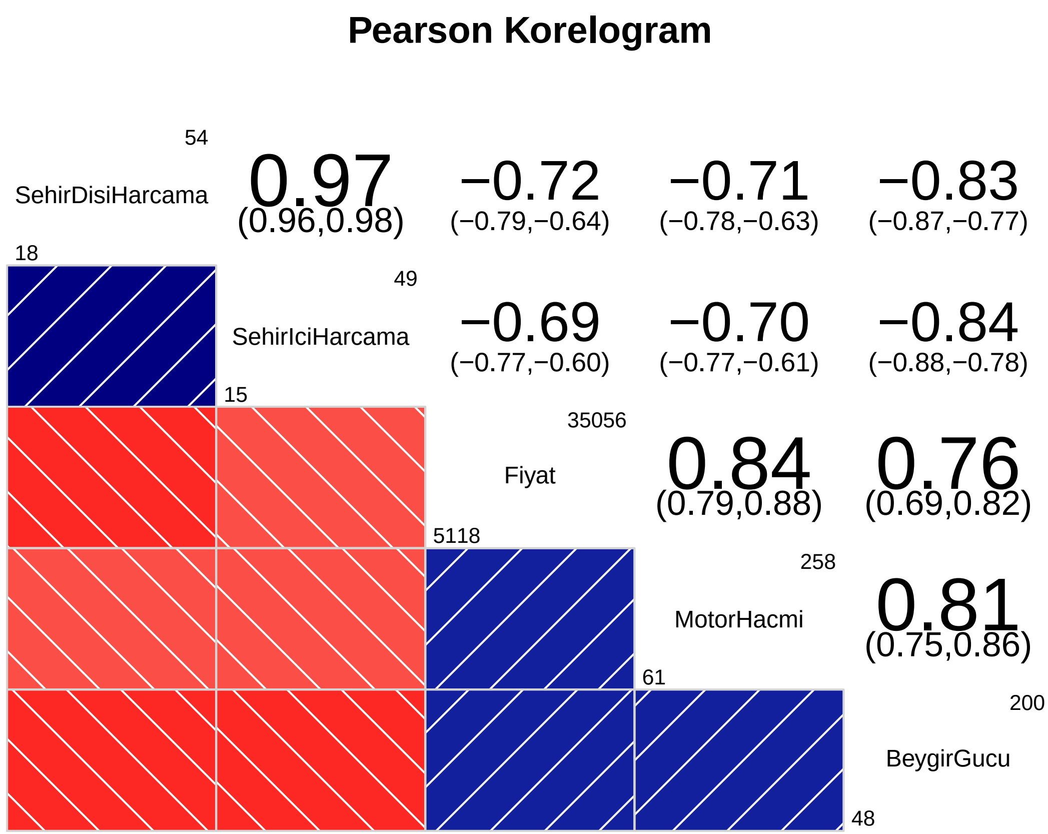
Gördüğünüz gibi rastgele sayılar ile oluşturduğumuz test ortamında korelasyon katsayıları ile ilgili test *x* ve *y* değişkeni arasındaki ilişkinin yönünü doğru olarak buldu. Şimdi gerçek bir veride bu testleri gerçekleştirelim. Ancak burada bütün değişkenler arasındaki ilişkinin bulunması işlemini grafiksel olarak yapmaya çalışalım. İlk olarak, saçılım grafiğinin bütün değişkenler için yapılan hali olan *Çoklu Saçılım Grafiğini* inceleyelim. Çoklu saçılım grafiği *pairs* fonksiyonu le yapılır. Aşağıdaki R kodlarını inceleyiniz.

```
veri = read.csv2('imports-85.csv')
sVeri = veri[c("Fiyat","MotorHacmi","BeygirGucu",
               "SehirIciHarcama", "SehirDisiHarcama")]
# Fiyat, Motor Hacmi ve Beygir Gucu arasındaki ilişkiyi
# Grafiksel olarak çizelim.
pairs(~Fiyat+MotorHacmi+BeygirGucu, data=sVeri,
      main="Çoklu Saçılım Grafikleri")
```



Çoklu saçılım grafiği değişkenler arasındaki korelasyon hakkında bilgi vermemektedir. Bu sebeple, *corrgram* paketinde yer alan **corrgram** fonksiyonunu kullanmanız yeterlidir. Aşağıdaki R kodlarını inceleyiniz.

```
library(corrgram)

corrgram(sVeri, order = TRUE, lower.panel = panel.shade,
         upper.panel = panel.conf, diag.panel = panel.minmax,
         main = "Pearson Korelogram", cor.method = "pearson")
```




Pearson korelasyon testi *normal dağılıma* sahip değişkenler arasındaki farkları incelemek için kullanılmaktadır. **Buradaki normallikten kasıt, incelenecek olan değişkenlerin ortak olarak normal dağılıma yani çok değişkenli normal dağılıma sahip olmalarıdır.** Ancak pratikte, iki değişkende tek başlarına normalse, çok boyutlu normallik varsayımının sağlandığı kabul edilebilir. Aksi durumlarda, Pearson korelasyon yerine, *Spearman Sıra Sayı Korelasyonu* yöntemi kullanılmalıdır. R ile Spearman korelasyon testi yine *cor.test* fonksiyonu ile gerçekleştirilir. Aşağıda Pearson korelasyon için gerçekleştirilen örneklerin Spearman korelasyon ile gerçekleştirilmiş durumları yer almaktadır. (NOT: Örneklerin normal dağılıma sahip olduğu görülmektedir. Aşağıda amaç sadece Spearman korelasyon testinin kullanımını **anlatmaktır**.)

```
set.seed(127)

x = rnorm(30)
```

```
e = runif(30)

y = x+e

t = cor.test(x,y,method = "spearman")

if(t$p.value<0.05) {
  # Korelasyon katsayısı 0'dan farklı
  # Acaba sıfırdan büyük mü?
  t = cor.test(x,y,method = "spearman", alternative = "g")
  if(t$p.value<0.05) {
    # Evet Büyük
    cat("Korelasyon katsayısı 0'dan büyüktür r=",
        t$estimate," p=",t$p.value,"\n")
  } else {
    # Hayır Küçük
    cat("Korelasyon katsayısı 0'dan küçüktür r=",
        t$estimate," p=",1-t$p.value,"\n")
  }
} else {
  # Korelasyon katsayısı 0'dan farklı değil!
  cat("Korelasyon katsayısı 0'dır r=",
      t$estimate," p=", t$p.value,"\n")
}
```

```
## Korelasyon katsayısı 0'dan büyüktür r= 0.9461624  p= 0
```

```
y = -2*x+runif(30)

t = cor.test(x,y,method = "spearman")

if(t$p.value<0.05) {
  # Korelasyon katsayısı 0'dan farklı
  # Acaba sıfırdan büyük mü?
  t = cor.test(x,y,method = "spearman", alternative = "g")
  if(t$p.value<0.05) {
    # Evet Büyük
    cat("Korelasyon katsayısı 0'dan büyüktür r=",
        t$estimate," p=",t$p.value,"\n")
  } else {
    # Hayır Küçük
    cat("Korelasyon katsayısı 0'dan küçüktür r=",
        t$estimate," p=",1-t$p.value,"\n")
  }
```

```
} else {
  # Korelasyon katsayısı 0'dan farklı değil!
  cat("Korelasyon katsayısı 0'dır r=",
      t$estimate," p=", t$p.value,"\n")
}
```

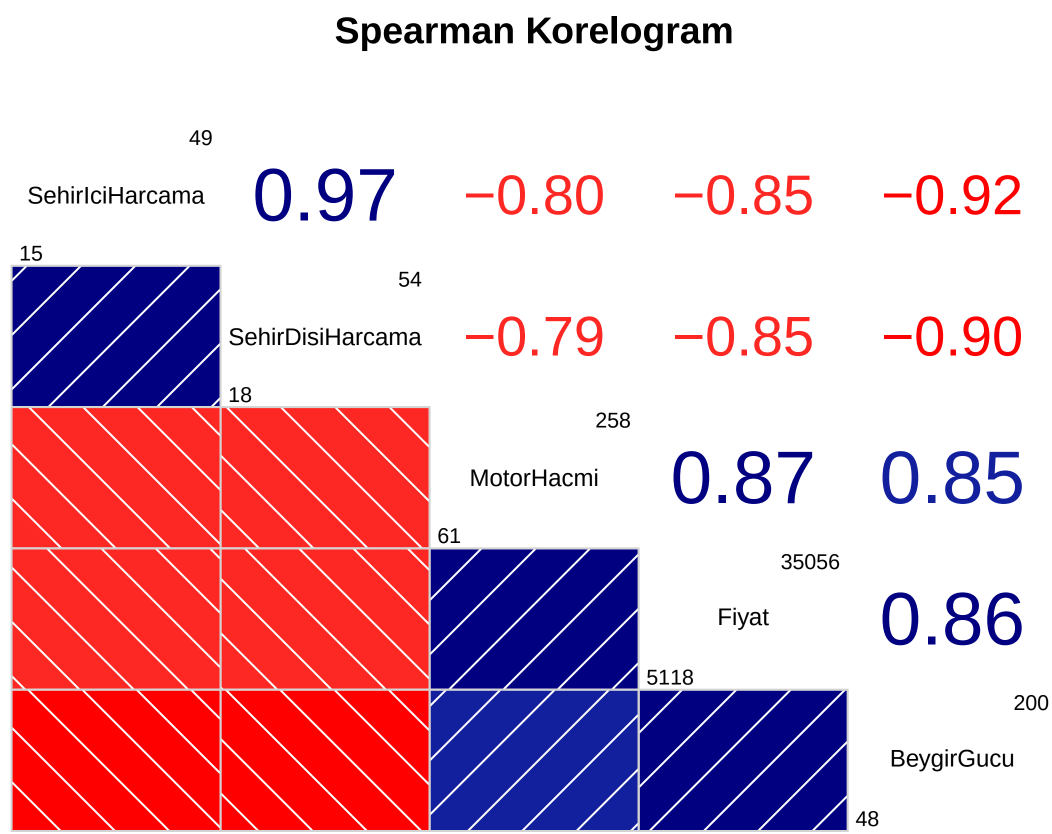
```
## Korelasyon katsayısı 0'dan küçüktür r= -0.9657397  p= 0
```

```
y = rnorm(30)

t = cor.test(x,y,method = "spearman")

if(t$p.value<0.05) {
  # Korelasyon katsayısı 0'dan farklı
  # Acaba sıfırdan büyük mü?
  t = cor.test(x,y,method = "spearman", alternative = "g")
  if(t$p.value<0.05) {
    # Evet Büyük
    cat("Korelasyon katsayısı 0'dan büyüktür r=",
        t$estimate," p=",t$p.value,"\n")
  } else {
    # Hayır Küçük
    cat("Korelasyon katsayısı 0'dan küçüktür r=",
        t$estimate," p=",1-t$p.value,"\n")
  }
} else {
  # Korelasyon katsayısı 0'dan farklı değil!
  cat("Korelasyon katsayısı 0'dır r=",
      t$estimate," p=", t$p.value,"\n")
}
```

```
## Korelasyon katsayısı 0'dır r= 0.1470523  p= 0.4364424
```

Şimdide spearman korelasyon değerleri için korelogram grafiği oluşturalım.

```
library(corrgram)

corrgram(sVeri, order = TRUE, lower.panel = panel.shade,
         upper.panel = panel.cor, diag.panel = panel.minmax,
         main = "Spearman Korelogram", cor.method = "spearman")
```




## 11.2 Regresyon Analizi

Korelasyon katsayısı iki değişkenin değişimlerinin birbirleri ile etkileşimi hakkında bilgi verir, fakat aralarındaki ilişkinin matematiksel yapısı hakkında bilgi vermez. Regresyon analizinde ise iki değişken arasındaki ilişki matematiksel bir formül veya *model* aracılığı ile açıklanmaya çalışılır. Bu denklemin yapısı basit doğrusal bir yapıda olabileceği gibi doğrusal olmayan bir yapıda da olabilir. Doğrusal yapıya sahip regresyon modeline *Doğrusal Regresyon*, doğrusal yapıya sahip olmayan regresyon modeline de *Doğrusal Olmayan Regresyon* adı verilir.

### 11.2.1 Basit Doğrusal Regresyon

Bir adet bağımlı değişken ve bir adet bağımsız değişkenden oluşan regresyon modeline **Basit Doğrusal Regresyon** modeli denir. Bu model aşağıdaki şekilde ifade edilir.

$$Y_i = \beta_0 + \beta_1 X_i$$

Bu modelde, X değişkeninin alabileceği bütün değerler için bir Y değeri bulunabilir ve bu değerlerde aynı doğru üzerindedir. Uygulamada bu şekilde bir model ortaya çıkması mümkün değildir. Bu sebeple yukarıdaki modele bir hata terimi eklenir ve aşağıdaki hale gelir.

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

Bu ifade regresyon modeli plarak adlandırılır. Burada, $Y_i$ değişkenine bağımlı veya etkilenen, $X_i$ değişkenine de bağımsız veya etkileyen değişken adı verilir. Burada yer alan regresyon modeli anakütle için kullanılacak olan regresyon modelidir. Örneklemden hesaplanan regresyon modeli ise aşağıdaki şekilde hesaplanır.

$$\widehat{Y_i} = \widehat{\beta_0} + \widehat{\beta_1} X_i + e_i$$

Burada $\widehat{\beta_i}$, $\beta_i$ nin $e_i$ ise $\epsilon_i$ için kullanılan tahmin edicilerdir. Bu tamin edicilerin hesaplanması aşağıdaki şekilde gerçekleştirilir.

$$\sum_{i=1}^{n} Y_i = n\widehat{\beta_0} + \widehat{\beta_1} \sum_{i=1}^{n} X_i$$
$$\sum_{i=1}^{n} X_i Y_i = \widehat{\beta_0} \sum_{i=1}^{n} X_i + \widehat{\beta_1} \sum_{i=1}^{n} X_i^2$$

Bu denklemlerden $\widehat{\beta}_0$ çekilip yerine yazılırsa $\widehat{\beta}_1$;

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n} X_i Y_i - n\overline{XY}}{\sum_{i=1}^{n} X_i^2 - n\overline{X}^2}$$

olarak ve $\beta_0$'da,

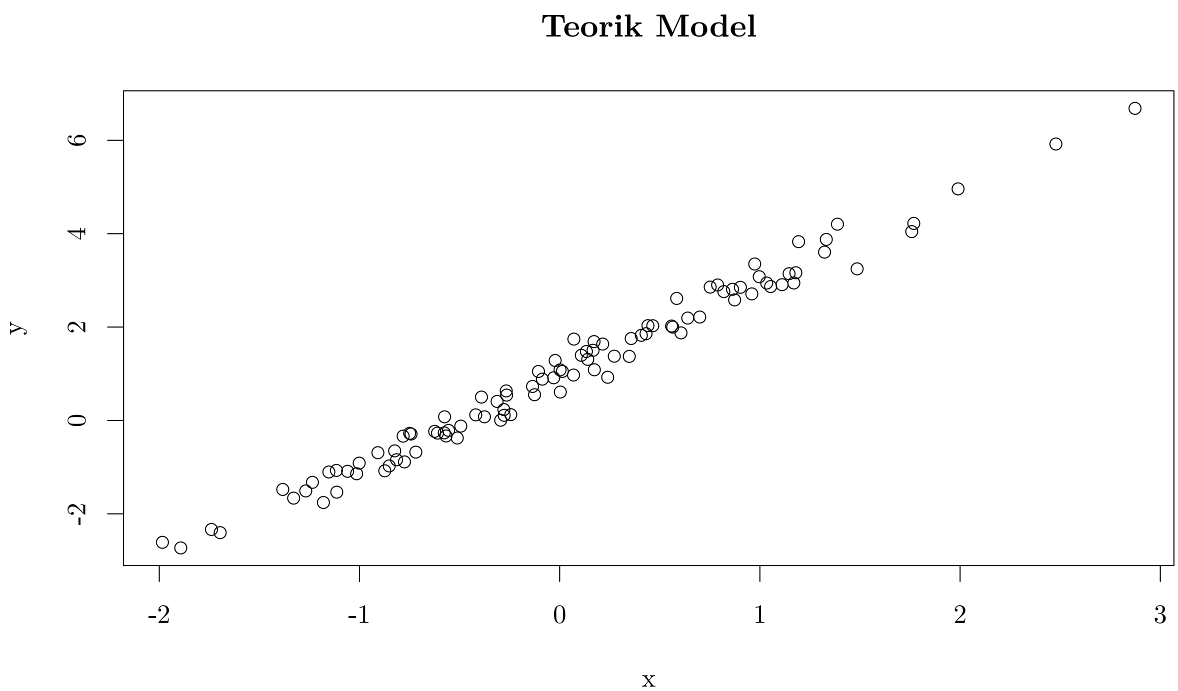
$$\widehat{\beta}_0 = \overline{Y} - \beta_1 \overline{X}$$

şeklinde hesaplanır.

R ile basit doğrusal regresyon modeli *lm* fonksiyonu ile oluşturulur. Korelasyon için oluşturduğumuz örnek sanal değişkenlerin regresyon modeli ile test edelim.

```
set.seed(127)

# Bağımsız Değişkeni Oluşturalım
x = rnorm(100)
# Bağımsız değişkeni sıralayalım
x = sort(x)

# Rastgele Hata Değişkenimizi Oluşturalım
```

```
e = rnorm(100, mean = 0, sd = 0.25)

B0 = 1
B1 = 2
# Teorik modeli oluşturalım B0=1 B1=2
y = B0 + B1*x + e

# Saçılım Grafiğine Bakalım
plot(x,y,main="Teorik Model")
```

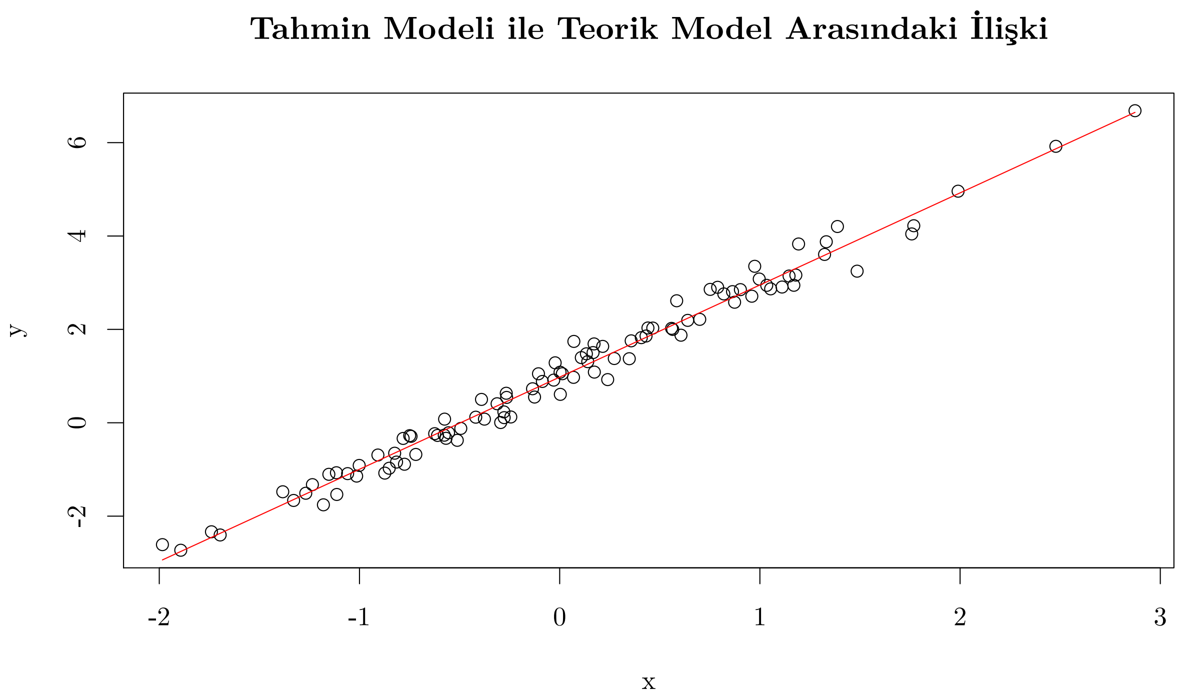



```
# Regresyon Modelini oluşturalım
regresyon.modeli = lm(y~x)

# Regresyon modelinden hesaplanan katsayıları öğrenelim
katsayilar = coefficients(regresyon.modeli)

print(katsayilar)
```

```
## (Intercept)           x
##   0.9776659   1.9731948
```

```
# Tahmin edilen katsayıları değişkenlere aktaralım
B0Sapka = katsayilar[1]
B1Sapka = katsayilar[2]
```

```
# Tahmin modelini çalıştıralım
ySapka = B0Sapka+B1Sapka*x

# Tahmin modeli ile Teorik modeli grafiksel olarak inceleyelim
plot(x,y,main="Tahmin Modeli ile Teorik Model Arasındaki İlişki")
lines(x,ySapka,col="red")
```




Örnekten de gördüğünüz gibi doğrusal regresyon modelimizden elde edilen parametre tahmin edicileri, teorik modelde kullandığımız parametreler ile benzerlik göstermektedir.

### 11.2.2 Belirlilik Katsayısı

Regresyon modeli bağımlı değişkende oluşan değişimin bir kısmını açıklar, geri kalan kısmı ise açıklanamaz. Aşağıda bu değişimlerin matematiksel ifadeleri yer almaktadır.

$$TD = \sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2$$

$$AD = \sum_{i=1}^{n} \left(\widehat{Y}_i - \overline{Y}\right)^2$$

$$AD_1 = \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i\right)^2$$

Burada, $TD$ toplam değişim miktarını, $AD$ açıklanan değişim miktarını ve $AD_1$'de açıklanamayan değişim miktarını göstermektedir. Belirlilik katsayısı, açıklanan değişim miktarının toplam değişim miktarına oranlanması ile bulunur. $0 \leq R^2 \leq 1$ dir. Formülü aşağıdaki gibidir.

$$R^2 = \frac{AD}{TD} = \frac{\sum_{i=1}^{n} \left(\widehat{Y}_i - \overline{Y}\right)^2}{\sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2}$$

$R^2$ değeri regresyon modelinin açıklayıcılığının ölçüsüdür.

R ile oluşturduğunuz modellerde belirleyicilik katsayısı otomatik olarak hesaplanır. Bir önceki başlıkta oluşturduğumuz model için belirleyicilik katsayısını bulan R kodu aşağıda verilmiştir.

```
model.ozeti = summary(regresyon.modeli)

r2 = model.ozeti$r.squared

print(r2)
```

```
## [1] 0.7166659
```

### 11.2.3 Katsayıların Testi

Regresyon modelinde hesaplanan katsayıların ($\beta_0$ ve $\beta_1$) istatistiksel olarak anlamlı olup olmadığını ölçmek gereklidir. Bu amaçla oluşturacağınız hipotez aşağıdaki gibidir.

$$H_0 : \beta_i = 0$$
$$H_1 : \beta_i \neq 0$$

R, model özetinde size hesapladığı katsayılar için gerekli test istatistiğini ve p değerini hesaplar. Aşağıdaki R kodunu inceleyin.

```
print(model.ozeti$coefficients)
```

```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 1.155797  0.2086572 5.539214 6.367339e-06
## x           2.071917  0.2461979 8.415656 3.744993e-09
```

Bu tabloda *Pr(>|t|)* sütunu kullanacağınız p değerlerini gösterir. Burada $\widehat{\beta}_0$ *(Intercept)* olarak, $\widehat{\beta}_1$ de $x$ ile ifade edilmektedir.

### 11.2.4 Regresyon Modeli ile Tahmin

Regresyon modelinizi oluşturduktan sonra bilinmeyen bir x değeri için y değerini tahmin etmeye çalışalım. Aşağıdaki R kodunu inceleyiniz.

```
# Bilinmeyen bir x değeri alalım
bilinmeyen.x = 35

# Modelimizi çalıştıralım
tahmin.y = B0Sapka+B1Sapka*bilinmeyen.x

# Tahmin Edilen Y Değeri
print(tahmin.y)
```

```
## (Intercept)
##    73.67288
```

### 11.2.5 Gerçek Veri ile Uygulama

Dersin sitesinden TÜFE verisini indirin. Bu veride, yıllara göre tüketici fiyat endeksine göre hesaplanan enflasyon değerleri ve nüfus bilgisi bulunmaktadır. Yıllara göre nüfus için bir regresyon modeli oluşturup, 2016, 2017 ve 2018 için nüfusumuzu tahmin etmeye çalışalım. Aşağıdaki R kodlarını çalıştırın.

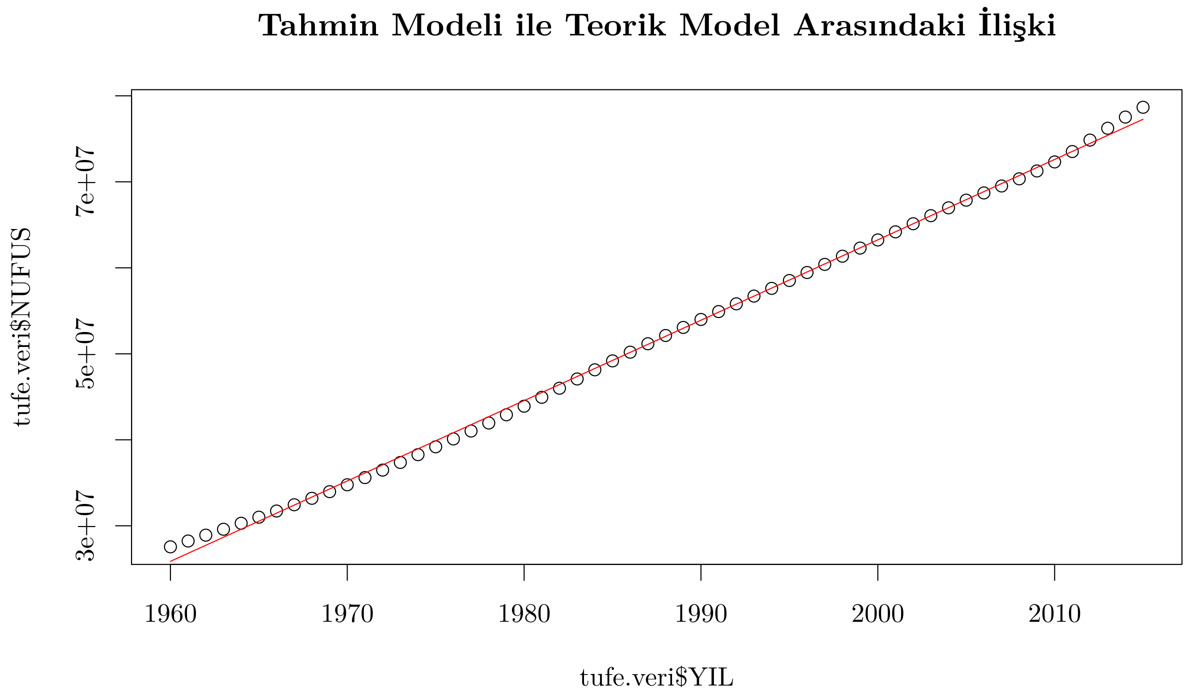
```
tufe.veri = read.csv2("TUFE.csv")

nufus.model = lm(NUFUS~YIL, data=tufe.veri)

katsayilar = coefficients(nufus.model)

B0Sapka = katsayilar[1]
B1Sapka = katsayilar[2]

nufus.ozet = summary(nufus.model)

r2 = nufus.ozet$r.squared

print(r2)
```

```
## [1] 0.9985873
```

```
print(nufus.ozet$coefficients)
```

```
##                  Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) -1805989694.9 9508066.363 -189.9429 5.565373e-78
## YIL              934619.8    4783.775  195.3729 1.217343e-78
```

```
nufusSapka = B0Sapka+B1Sapka*tufe.veri$YIL

# Tahmin modeli ile Teorik modeli grafiksel olarak inceleyelim
plot(tufe.veri$YIL,tufe.veri$NUFUS,
     main="Tahmin Modeli ile Teorik Model Arasındaki İlişki")
lines(tufe.veri$YIL,nufusSapka,col="red")
```




```
# Tahmin için gerekli değerleri oluşturalım

yillar = c(2016, 2017, 2018)

nufus.tahmin = B0Sapka+B1Sapka*yillar

print(nufus.tahmin)
```

```
## [1] 78203788 79138408 80073028
```

Gördüğünüz üzere modelimiz 2016, 2017 ve 2018 yılları için nüfusumuzu sırasıyla 78.203.788, 79.138.408 ve 80.073.028 olarak tahmin etmiştir.

### 11.2.6 Doğrusal Olmayan Modeller

Bu modeller gerçekte doğrusal olan ve gerçekte doğrusal olmayan modeller olmak üzere ikiye ayrılırlar. Gerçekte doğrusal olan modeller, değişken dönüşümleri yardımıyla doğrusallaştırılabilen modellerdir. Bu bölümde sadece bu modellerden bahsedilecektir.

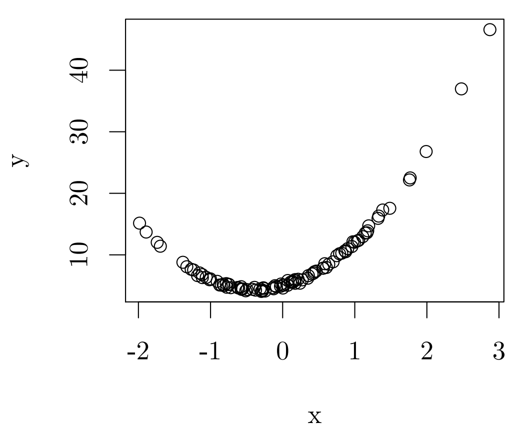
#### 11.2.6.1 Kareli Model

Değişkenlerin saçılım grafiği incelendiğinde tek maksimum veya tek minimum yapması durumunda kareli model doğrusal modelden daha fazla açıklayıcılığa sahip olabilir. Bu durumda teorik model aşağıdaki gibi olur.

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i$$

Bu durumu R ile inceleyelim.

```
set.seed(127)

# Bağımsız değişkenimizi oluşturalım.
x = rnorm(100)
# Bağımsız değişkeni sıralayalım
x = sort(x)

# Kare değişkenimizi oluşturalım
x2 = x * x

# Hata oranımızı oluşturalım
e = rnorm(100, mean = 0, sd = 0.25)

# Model parametrelerini oluşturalım
B0 = 5
B1 = 3
B2 = 4

# Modeli kurup y değişkenimizi oluşturalım
y = B0+B1*x+B2*x2+e

# Saçılım Grafiğine bakalım
plot(x,y)
```



```r
# Regresyon modelimizi oluşturalım
polinom.model = lm(y~x+x2)

# Regresyon Modeli Katsayılarına Bakalım
katsayilar = coefficients(polinom.model)

print(katsayilar)
```

```
## (Intercept)           x          x2
##    4.975865    2.972448    4.002021
```

```r
# Belirleyicilik Katsayısını Bulalım
polinom.ozet = summary(polinom.model)

r2 = polinom.ozet$r.squared

print(r2)
```

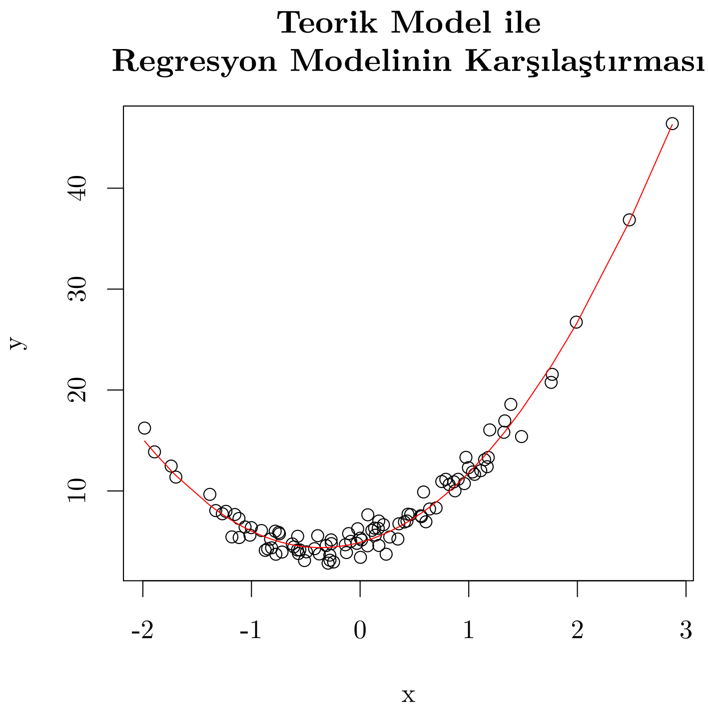
```
## [1] 0.998577
```

```r
# Katsayıların anlamlılığını ölçelim
print(polinom.ozet$coefficients)
```

```
##             Estimate Std. Error  t value      Pr(>|t|)
## (Intercept) 4.975865 0.03076840 161.7200 8.681556e-120
## x           2.972448 0.02728135 108.9553 3.041549e-103
## x2          4.002021 0.02019247 198.1937 2.492997e-128
```

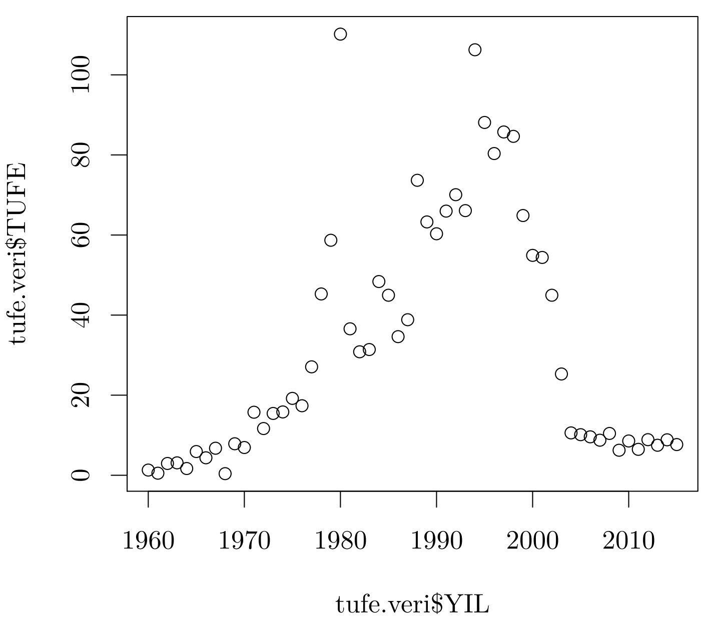
```r
# Tahmin Modelini Çalıştıralım
ySapka = katsayilar[1]+katsayilar[2]*x+katsayilar[3]*x2
```

```r
# Teorik model ile regresyon modelinin karşılaştıralım
plot(x, y, main = "Teorik Model ile\nRegresyon Modelinin Karşılaştırması")
lines(x, ySapka, col="red")
```




Gördüğünüz üzere regresyon modeli teorik modele yakın ve istatistiksel olarak anlamlı sonuçlar verdi. Bir önceki verimizde yer alan TÜFE verilerini incelerseniz de kareli regresyonun doğrusal regresyondan daha uygun bir açıklayıcılığa sahip olacağını görürsünüz. Şimdi elimizdeki verileri kullanıp gelecek 3 yıl için TÜFE değerlerini tahmin etmeye çalışalım.

```
tufe.veri = read.csv2("TUFE.csv")

plot(tufe.veri$YIL, tufe.veri$TUFE)
```



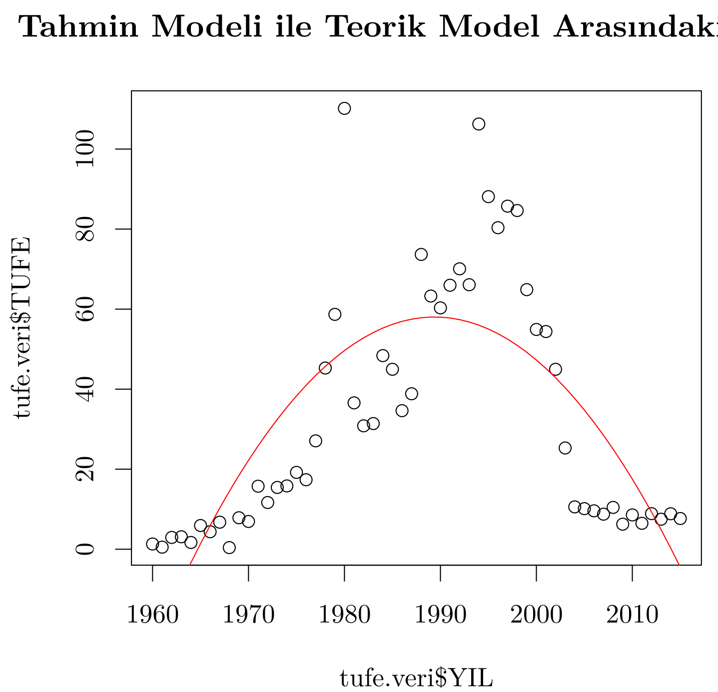
```r
tufe.veri$YIL2 = tufe.veri$YIL*tufe.veri$YIL

tufe.model = lm(TUFE~YIL+YIL2, data=tufe.veri)

katsayilar = coefficients(tufe.model)

B0Sapka = katsayilar[1]
B1Sapka = katsayilar[2]
B2Sapka = katsayilar[3]

tufe.ozet = summary(tufe.model)

r2 = tufe.ozet$r.squared

print(r2)
```

```
## [1] 0.5837935
```

```r
print(tufe.ozet$coefficients)
```

```
##                  Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept) -3.770924e+05 4.511567e+04 -8.358346 3.008068e-11
## YIL          3.791632e+02 4.540152e+01  8.351334 3.086142e-11
## YIL2        -9.529671e-02 1.142169e-02 -8.343486 3.175948e-11
```

```r
tufeSapka = B0Sapka+B1Sapka*tufe.veri$YIL+B2Sapka*tufe.veri$YIL2

# Tahmin modeli ile Teorik modeli grafiksel olarak inceleyelim
plot(tufe.veri$YIL,tufe.veri$TUFE,
     main="Tahmin Modeli ile Teorik Model Arasındaki İlişki")
lines(tufe.veri$YIL,tufeSapka,col="red")
```




```
# Tahmin için gerekli değerleri oluşturalım

yillar = c(2016, 2017, 2018)
yillar2 = yillar*yillar

tufe.tahmin = B0Sapka+B1Sapka*yillar+B2Sapka*yillar2

print(tufe.tahmin)
```

```
## [1]  -9.487104 -14.655488 -20.014465
```

Modelimizin tahminlerine göre önümüzdeki üç yıl TÜFE endeksi negatif olacak :).

#### 11.2.6.2 Tam Logaritmik Model

Bu modelin teorik yapısı aşağıdaki gibidir.

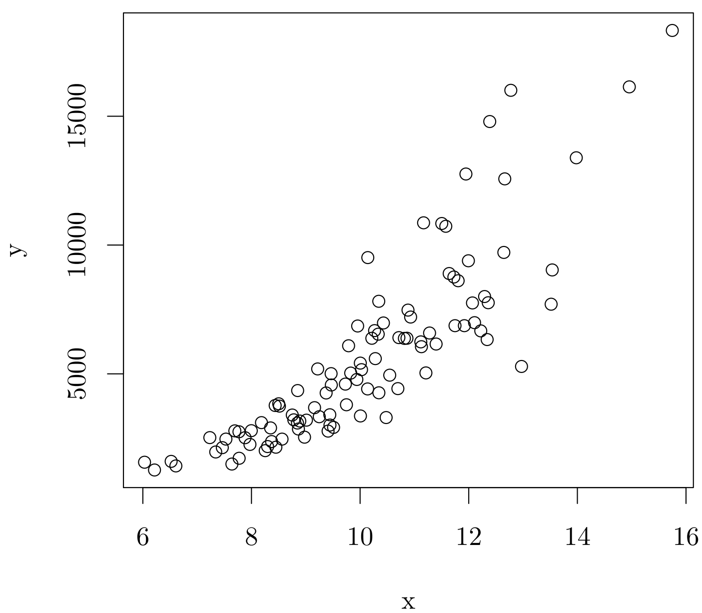
$$Y_i = \beta_0 X_i^{\beta_1} e^{e_i}$$

Teorik modelin her iki tarafının logaritması alınırsa, model aşağıdaki hale gelecektir.

$$lnY_i = ln\beta_0 + \beta_1 lnX_i + e_i$$

Bu modeli R ile inceleyelim.

```r
set.seed(127)

# Bağımsız değişkenimizi oluşturalım.
x = rnorm(100, mean = 10, sd=2)
# Bağımsız değişkeni sıralayalım
x = sort(x)

# Hata oranımızı oluşturalım
e = rnorm(100, mean = 0, sd = 0.25)

# Model parametrelerini oluşturalım
B0 = 5
B1 = 3

# Modeli kurup y değişkenimizi oluşturalım
y = B0*(x^B1)*exp(e)

# Saçılım Grafiğine bakalım
plot(x,y)
```




```r
# Regresyon modelimizi oluşturalım
log.model = lm(log(y)~log(x))

# Regresyon Modeli Katsayılarına Bakalım
katsayilar = coefficients(log.model)

print(katsayilar)
```

```
## (Intercept)      log(x)
##    1.902981    2.861748
```

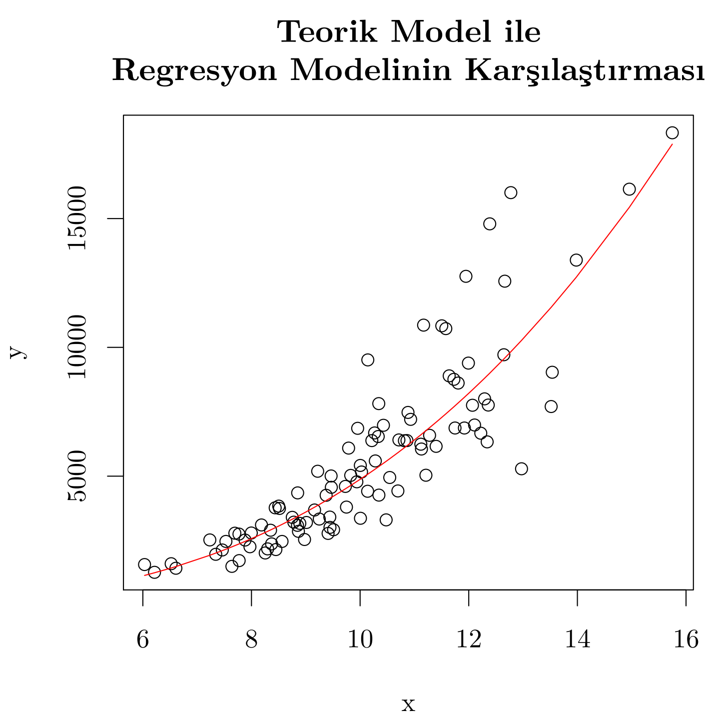
```
# Belirleyicilik Katsayısını Bulalım
log.ozet = summary(log.model)

r2 = log.ozet$r.squared

print(r2)
```

```
## [1] 0.8320554
```

```
# Katsayıların anlamlılığını ölçelim
print(log.ozet$coefficients)
```

```
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 1.902981  0.2987114  6.370633 6.136969e-09
## log(x)      2.861748  0.1298749 22.034652 9.492986e-40
```

```
# Tahmin Modelini Çalıştıralım
ySapka = exp(katsayilar[1])*x^katsayilar[2]

# Teorik model ile regresyon modelinin karşılaştıralım
plot(x, y, main = "Teorik Model ile\nRegresyon Modelinin Karşılaştırması")
lines(x, ySapka, col="red")
```



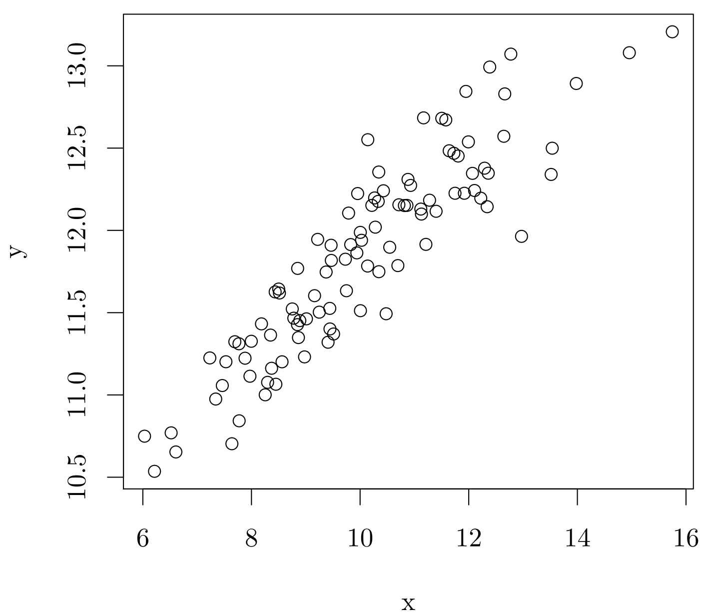
#### 11.2.6.3 Doğrusal Logaritmik Model

Bu modelin teorik yapısı aşağıdaki gibidir.

$$Y_i = \beta_0 + \beta_1 lnX_i + e_i$$

Bu modeli R ile inceleyelim.

```
set.seed(127)

# Bağımsız değişkenimizi oluşturalım.
x = rnorm(100, mean = 10, sd=2)
# Bağımsız değişkeni sıralayalım
x = sort(x)

# Hata oranımızı oluşturalım
e = rnorm(100, mean = 0, sd = 0.25)

# Model parametrelerini oluşturalım
B0 = 5
B1 = 3

# Modeli kurup y değişkenimizi oluşturalım
y = B0+B1*log(x)+e

# Saçılım Grafiğine bakalım
plot(x,y)
```



```
# Regresyon modelimizi oluşturalım
log.model = lm(y~log(x))

# Regresyon Modeli Katsayılarına Bakalım
katsayilar = coefficients(log.model)

print(katsayilar)
```

```
## (Intercept)      log(x)
##    5.293543    2.861748
```

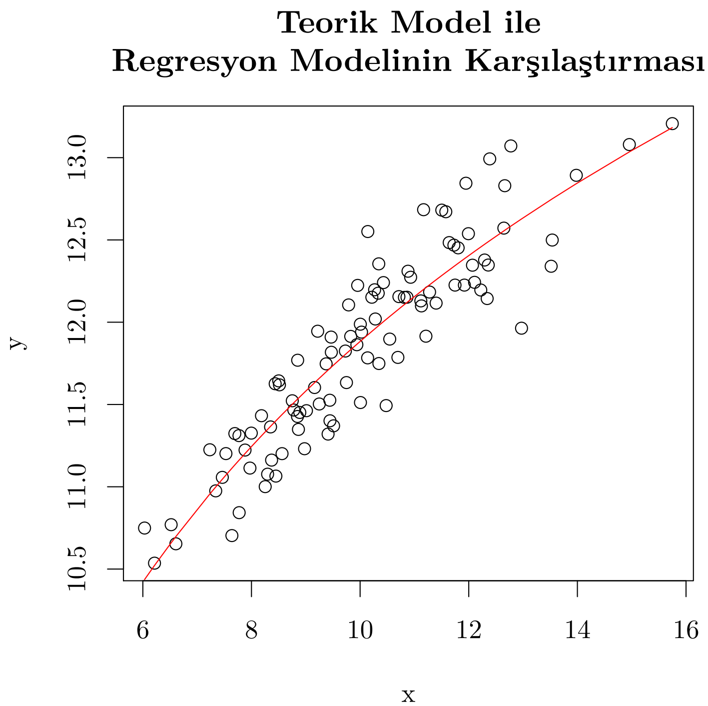
```
# Belirleyicilik Katsayısını Bulalım
log.ozet = summary(log.model)

r2 = log.ozet$r.squared

print(r2)
```

```
## [1] 0.8320554
```

```
# Katsayıların anlamlılığını ölçelim
print(log.ozet$coefficients)
```

```
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 5.293543  0.2987114 17.72126 2.516169e-32
## log(x)      2.861748  0.1298749 22.03465 9.492986e-40
```

```
# Tahmin Modelini Çalıştıralım
ySapka = katsayilar[1]+katsayilar[2]*log(x)

# Teorik model ile regresyon modelinin karşılaştıralım
plot(x, y, main = "Teorik Model ile\nRegresyon Modelinin Karşılaştırması")
lines(x, ySapka, col="red")
```




#### 11.2.6.4 Logaritmik Doğrusal Model

Bu modelin teorik yapısı aşağıdaki gibidir.

$$Y_i = e^{\beta_0 + \beta_1 X_i + e_i}$$

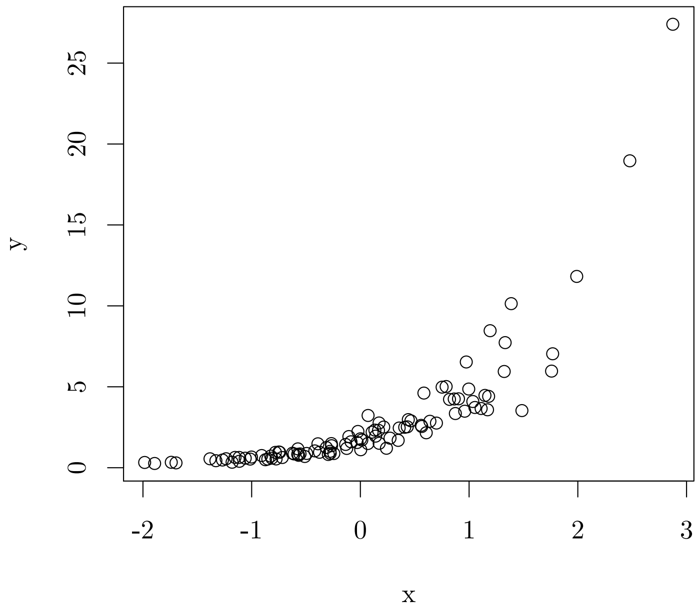
Teorik modelin her iki tarafının logaritması alınırsa model aşağıdaki şekle dönüşecektir.

$$lnY_i = \beta_0 + \beta_1 X_i + e_i$$

Bu modeli R ile inceleyelim.

```
set.seed(127)

# Bağımsız değişkenimizi oluşturalım.
x = rnorm(100)
# Bağımsız değişkeni sıralayalım
x = sort(x)

# Hata oranımızı oluşturalım
e = rnorm(100, mean = 0, sd = 0.25)

# Model parametrelerini oluşturalım
B0 = 0.5
B1 = 1
```

```
# Modeli kurup y değişkenimizi oluşturalım
y = exp(B0+B1*x+e)

# Saçılım Grafiğine bakalım
plot(x,y)
```




```
# Regresyon modelimizi oluşturalım
log.model = lm(log(y)~x)

# Regresyon Modeli Katsayılarına Bakalım
katsayilar = coefficients(log.model)

print(katsayilar)
```

```
## (Intercept)           x
##   0.4776659   0.9731948
```

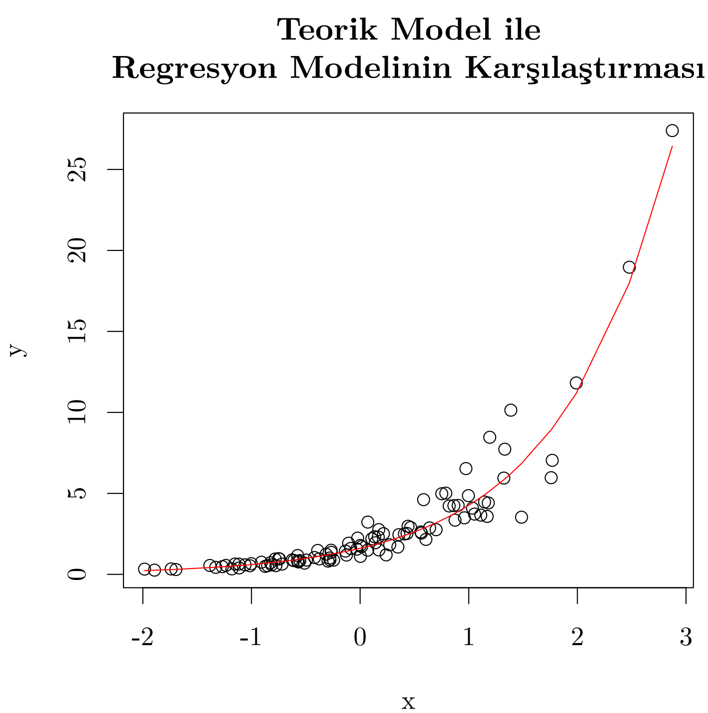
```
# Belirleyicilik Katsayısını Bulalım
log.ozet = summary(log.model)

r2 = log.ozet$r.squared

print(r2)
```

```
## [1] 0.934121
```

```r
# Katsayıların anlamlılığını ölçelim
print(log.ozet$coefficients)
```

```
##              Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 0.4776659 0.02483380 19.23451 4.838042e-35
## x           0.9731948 0.02610711 37.27700 1.092270e-59
```

```r
# Tahmin Modelini Çalıştıralım
ySapka = exp(katsayilar[1]+katsayilar[2]*x)

# Teorik model ile regresyon modelinin karşılaştıralım
plot(x, y, main = "Teorik Model ile\nRegresyon Modelinin Karşılaştırması")
lines(x, ySapka, col="red")
```



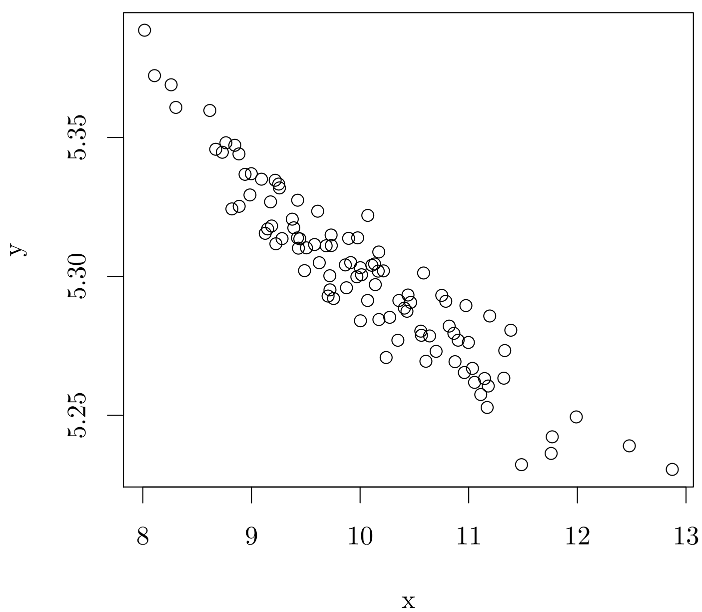

#### 11.2.6.5 Ters Model

Bu modelin teorik yapısı aşağıdaki gibidir.

$$Y_i = \beta_0 + \beta_1 \frac{1}{X_i} + e_i$$

Bu modeli R ile inceleyelim.

```
set.seed(127)

# Bağımsız değişkenimizi oluşturalım.
x = rnorm(100, mean = 10, sd = 1)
# Bağımsız değişkeni sıralayalım
x = sort(x)

# Hata oranımızı oluşturalım
e = rnorm(100, mean = 0, sd = 0.01)

# Model parametrelerini oluşturalım
B0 = 5
B1 = 3

# Modeli kurup y değişkenimizi oluşturalım
y = B0+B1*(1/x)+e

# Saçılım Grafiğine bakalım
plot(x,y)
```




```
# Dönüşüm Değişkeni oluşturalım
invX = 1/x;

# Regresyon modelimizi oluşturalım
ters.model = lm(y~invX)

# Regresyon Modeli Katsayılarına Bakalım
katsayilar = coefficients(ters.model)
```

```r
print(katsayilar)
```

```
## (Intercept)        invX
##    4.987958    3.110511
```

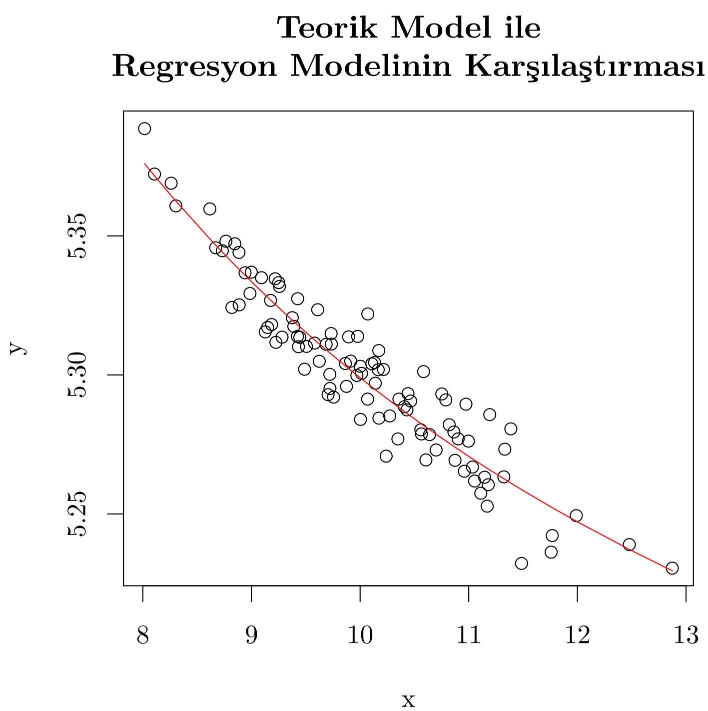
```r
# Belirleyicilik Katsayısını Bulalım
ters.ozet = summary(ters.model)

r2 = ters.ozet$r.squared

print(r2)
```

```
## [1] 0.8998785
```

```r
# Katsayıların anlamlılığını ölçelim
print(ters.ozet$coefficients)
```

```
##             Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 4.987958 0.01058146 471.38674 2.985563e-166
## invX        3.110511 0.10480706  29.67845  8.984535e-51
```

```r
# Tahmin Modelini Çalıştıralım
ySapka = katsayilar[1]+katsayilar[2]*(1/x)

# Teorik model ile regresyon modelinin karşılaştıralım
plot(x, y, main = "Teorik Model ile\nRegresyon Modelinin Karşılaştırması")
lines(x, ySapka, col="red")
```


Teorik Model ile
Regresyon Modelinin Karşılaştırması
y
5.25
5.30
5.35
x
8
9
10
11
12
13