VERİ BİLİMİ RPOGRAMI

HESAPLAMALI İSTATİSTİKLER I DERSİ

UYGULAMA

14. DERS

Doç.Dr. Erol TERZİ

R PROGRAMLAMA

R, istatistiksel analiz, grafik gösterimi ve raporlama için bir programlama dili ve yazılım ortamıdır. R'nin özü, işlevler kullanarak modüler programlamanın yanı sıra dallanma ve döngüye izin veren yorumlanmış bir bilgisayar dilidir. R, verimlilik için C, C ++, .Net, Python veya FORTRAN dillerinde yazılmış prosedürlerle entegrasyon sağlar.

R'nin Özellikleri

R; istatistiksel analiz, grafik gösterimi ve raporlama için bir programlama dili ve yazılım ortamıdır. Aşağıdakiler R'nin önemli özellikleridir.

- R, koşullu ifadeler, döngüler, kullanıcı tanımlı özyinelemeli işlevler ve giriş ve çıkış olanaklarını içeren iyi geliştirilmiş, basit ve etkili bir programlama dilidir.
- R'nin etkili bir veri işleme ve depolama alanı vardır.
- R, diziler, listeler, vektörler ve matrisler üzerindeki hesaplamalar için bir operatör paketi sağlar.
- R, veri analizi için geniş, tutarlı ve entegre bir araç koleksiyonu sağlar.
- R, veri analizi ve doğrudan bilgisayarda veya kağıtlara yazdırılması için grafiksel olanaklar sağlar.

Web : https://www.tutorialspoint.com/r/index.htm

R - Ortalama, Medyan ve Mod

R'deki istatistiksel analiz, birçok yerleşik işlev kullanılarak gerçekleştirilir. Bu işlevlerin çoğu R temel paketinin parçasıdır. Bu fonksiyonlar R vektörünü argümanlarla birlikte girdi olarak alır ve sonucu verir.

Bu bölümde tartıştığımız işlevler ortalama, medyan ve moddur.

Anlamına gelmek

Değerlerin toplamı alınarak ve bir veri serisindeki değerlerin sayısına bölünerek hesaplanır.

Ortalama () fonksiyonu bunu R'de hesaplamak için kullanılır.

Sözdizimi

R'deki ortalamayı hesaplamak için temel sözdizimi -

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Aşağıda kullanılan parametrelerin açıklaması verilmiştir -

- **x** giriş vektörüdür.
- **trim** , sıralanmış vektörün her iki ucundan bazı gözlemleri bırakmak için kullanılır.
- **na.rm** , eksik değerleri giriş vektöründen çıkarmak için kullanılır.

Misal

Canlı Demo

```
# Create a vector.
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find Mean.
result.mean <- mean(x)
print(result.mean)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
[1] 8.22
```

Kırpma Seçeneğini Uygulama

Kırpma parametresi verildiğinde, vektördeki değerler sıralanır ve ardından gerekli gözlem sayısı ortalamanın hesaplanmasından çıkarılır.

Trim = 0.3 olduğunda, ortalamayı bulmak için her uçtan 3 değer hesaplamalardan çıkarılır.

Bu durumda sıralanmış vektör (−21, −5, 2, 3, 4.2, 7, 8, 12, 18, 54) ve ortalamanın hesaplanması için vektörden çıkarılan değerler (−21, −5,2) soldan ve (12,18,54) sağdan.

Canlı Demo

```
# Create a vector.
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find Mean.
result.mean <-  mean(x,trim = 0.3)
print(result.mean)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
[1] 5.55
```

NA Seçeneğini Uygulama

Eksik değerler varsa, ortalama işlev NA değerini döndürür.

Eksik değerleri hesaplamadan çıkarmak için na.rm = TRUE kullanın. bu, NA değerlerini kaldırmak anlamına gelir.

Canlı Demo

```
# Create a vector.
x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)

# Find mean.
result.mean <-  mean(x)
print(result.mean)

# Find mean dropping NA values.
result.mean <-  mean(x,na.rm = TRUE)
print(result.mean)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
[1] NA
[1] 8.22
```

Medyan

Bir veri serisindeki en ortadaki değere medyan denir. **Ortanca ()** işlevi, bu değeri hesaplamak için R kullanılır.

Sözdizimi

R'de medyanı hesaplamak için temel sözdizimi -

```
median(x, na.rm = FALSE)
```

Aşağıda kullanılan parametrelerin açıklaması verilmiştir -

- **x** giriş vektörüdür.
- **na.rm** , eksik değerleri giriş vektöründen çıkarmak için kullanılır.

Misal

Canlı Demo

```
# Create the vector.
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find the median.
median.result <- median(x)
print(median.result)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
[1] 5.6
```

Mod

Mod, bir veri kümesinde en yüksek tekrar sayısına sahip olan değerdir. Ortalama ve medyandan farklı olarak, mod hem sayısal hem de karakter verilerine sahip olabilir.

R, modu hesaplamak için standart yerleşik bir işleve sahip değildir. Bu nedenle, R'deki bir veri kümesinin modunu hesaplamak için bir kullanıcı işlevi oluştururuz. Bu işlev, vektörü girdi olarak alır ve mod değerini çıktı olarak verir.

Misal

Canlı Demo

```
# Create the function.
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

# Create the vector with numbers.
v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)

# Calculate the mode using the user function.
result <- getmode(v)
print(result)

# Create the vector with characters.
charv <- c("o","it","the","it","it")
```

```
# Calculate the mode using the user function.
result <- getmode(charv)
print(result)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
[1] 2
[1] "it"
```

R - Doğrusal Regresyon

Regresyon analizi, iki değişken arasında bir ilişki modeli oluşturmak için çok yaygın olarak kullanılan bir istatistiksel araçtır. Bu değişkenlerden biri, değeri deneylerle elde edilen yordayıcı değişken olarak adlandırılır. Diğer değişken, değeri yordayıcı değişkenden türetilen yanıt değişkeni olarak adlandırılır.

Doğrusal Regresyonda bu iki değişken, her iki değişkenin üssünün (kuvvetinin) 1 olduğu bir denklem yoluyla ilişkilidir. Matematiksel olarak doğrusal bir ilişki, grafik olarak çizildiğinde düz bir çizgiyi temsil eder. Herhangi bir değişkenin üssünün 1'e eşit olmadığı doğrusal olmayan bir ilişki bir eğri oluşturur.

Doğrusal regresyon için genel matematiksel denklem -

```
y = ax + b
```

Aşağıda kullanılan parametrelerin açıklaması verilmiştir -

- **y** yanıt değişkenidir.
- **x** yordayıcı değişkendir.
- **a** ve **b** , katsayılar olarak adlandırılan sabitlerdir.

Bir Regresyon Oluşturma Adımları

Basit bir regresyon örneği, bir kişinin boyu bilindiği zaman kilosunu tahmin etmektir. Bunu yapmak için bir kişinin boyu ile kilosu arasındaki ilişkiye sahip olmamız gerekir.

İlişkiyi yaratmanın adımları -

- Gözlenen boy ve buna karşılık gelen ağırlık değerlerinden bir örnek toplama deneyini gerçekleştirin.
- R'deki **lm ()** işlevlerini kullanarak bir ilişki modeli oluşturun.
- Oluşturulan modelden katsayıları bulun ve bunları kullanarak matematiksel denklemi oluşturun
- Tahmindeki ortalama hatayı bilmek için ilişki modelinin bir özetini alın. **Kalıntılar** da denir .
- Yeni kişilerin ağırlığını tahmin etmek için , R'deki **tahmin ()** işlevini kullanın.

Giriş Verileri

Gözlemleri temsil eden örnek veriler aşağıdadır -

```
# Values of height
151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# Values of weight.
63, 81, 56, 91, 47, 57, 76, 72, 62, 48
```

lm () İşlevi

Bu işlev, yordayıcı ve yanıt değişkeni arasındaki ilişki modelini oluşturur.

Sözdizimi

Doğrusal regresyonda **lm ()** işlevinin temel sözdizimi -

```
lm(formula,data)
```

Aşağıda kullanılan parametrelerin açıklaması verilmiştir -

- **formül** , x ve y arasındaki ilişkiyi sunan bir semboldür.
- **veriler** , formülün uygulanacağı vektördür.

İlişki Modeli oluşturun ve Katsayıları alın

Canlı Demo

```
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

print(relation)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551          0.6746
```

İlişkinin Özetini Alın

Canlı Demo

```
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
```

```
# Apply the lm() function.
relation <- lm(y~x)

print(summary(relation))
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002  -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

tahmin () İşlevi

Sözdizimi

Doğrusal regresyonda tahmin () için temel sözdizimi -

```
predict(object, newdata)
```

Aşağıda kullanılan parametrelerin açıklaması verilmiştir -

- **nesne** , lm () işlevi kullanılarak önceden oluşturulmuş formüldür.
- **newdata** , tahmin değişkeni için yeni değeri içeren vektördür.

Yeni kişilerin ağırlığını tahmin edin

Canlı Demo

```
# The predictor vector.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# The resposne vector.
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
```

```
# Apply the lm() function.
relation <- lm(y~x)

# Find weight of a person with height 170.
a <- data.frame(x = 170)
result <-  predict(relation,a)
print(result)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
       1
76.22869
```

Regresyonu Grafik Olarak Görselleştirin

Canlı Demo

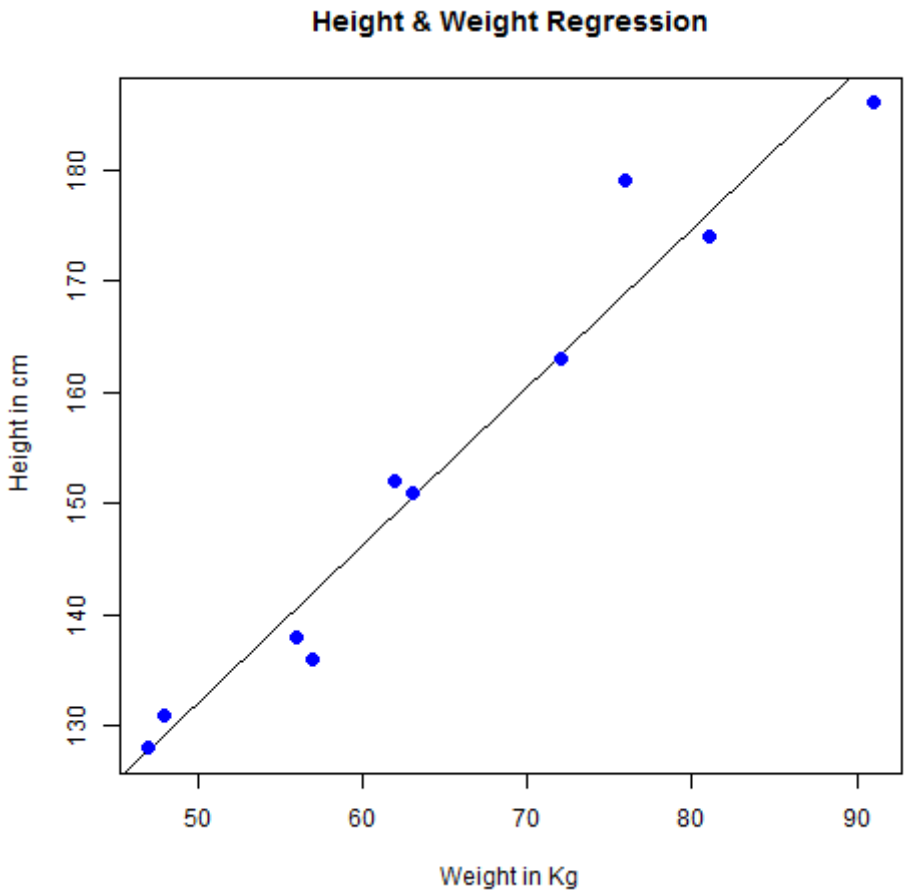
```
# Create the predictor and response variable.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation <- lm(y~x)

# Give the chart file a name.
png(file = "linearregression.png")

# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in cm")

# Save the file.
dev.off()
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -


Height & Weight Regression
Height in cm
130
140
150
160
170
180
50
60
70
80
90
Weight in Kg

R - Çoklu Regresyon

Çoklu regresyon, ikiden fazla değişken arasındaki ilişkiye doğrusal regresyonun bir uzantısıdır. Basit doğrusal ilişkide bir tahmin edicimiz ve bir yanıt değişkenimiz vardır, ancak çoklu regresyonda birden fazla tahmin değişkenimiz ve bir yanıt değişkenimiz vardır.

Çoklu regresyon için genel matematiksel denklem -

```
y = a + b1x1 + b2x2 +...bnxn
```

Aşağıda kullanılan parametrelerin açıklaması verilmiştir -

- **y** yanıt değişkenidir.
- **a, b1, b2 ... bn** katsayılarıdır.
- **x1, x2, ... xn** tahmin değişkenleridir.

R'de **lm ()** fonksiyonunu kullanarak regresyon modelini oluşturuyoruz. Model , giriş verilerini kullanarak katsayıların değerini belirler. Daha sonra, bu katsayıları kullanarak belirli bir yordayıcı değişken seti için yanıt değişkeninin değerini tahmin edebiliriz.

lm () İşlevi

Bu işlev, yordayıcı ve yanıt değişkeni arasındaki ilişki modelini oluşturur.

Sözdizimi

Çoklu regresyonda **lm ()** işlevinin temel sözdizimi -

```
lm(y ~ x1+x2+x3...,data)
```

Aşağıda kullanılan parametrelerin açıklaması verilmiştir -

- **formül** , yanıt değişkeni ile yordayıcı değişkenler arasındaki ilişkiyi sunan bir semboldür.
- **veriler** , formülün uygulanacağı vektördür.

Misal

Giriş Verileri

R ortamında bulunan "mtcars" veri setini düşünün. Galon başına kilometre (mpg), silindir deplasmanı ("disp"), beygir gücü ("hp"), arabanın ağırlığı ("wt") ve daha fazla parametre açısından farklı araba modelleri arasında bir karşılaştırma sağlar.

Modelin amacı, bir yanıt değişkeni olarak "mpg" ile tahmin değişkenleri olarak "disp", "hp" ve "wt" arasındaki ilişkiyi kurmaktır. Bu amaçla mtcars veri setinden bu değişkenlerin bir alt kümesini oluşturuyoruz.

Canlı Demo

```
input <- mtcars[,c("mpg","disp","hp","wt")]
print(head(input))
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
                  mpg   disp   hp    wt
Mazda RX4         21.0  160    110   2.620
Mazda RX4 Wag     21.0  160    110   2.875
Datsun 710        22.8  108     93   2.320
Hornet 4 Drive    21.4  258    110   3.215
Hornet Sportabout 18.7  360    175   3.440
Valiant           18.1  225    105   3.460
```

İlişki Modeli oluşturun ve Katsayıları alın

Canlı Demo

```
input <- mtcars[,c("mpg","disp","hp","wt")]

# Create the relationship model.
model <- lm(mpg~disp+hp+wt, data = input)

# Show the model.
print(model)

# Get the Intercept and coefficients as vector elements.
cat("# # # # The Coefficient Values # # # ","\n")

a <- coef(model)[1]
print(a)

Xdisp <- coef(model)[2]
Xhp <- coef(model)[3]
Xwt <- coef(model)[4]

print(Xdisp)
print(Xhp)
print(Xwt)
```

Yukarıdaki kodu çalıştırdığımızda aşağıdaki sonucu verir -

```
Call:
lm(formula = mpg ~ disp + hp + wt, data = input)
```

```
Coefficients:
(Intercept)         disp           hp           wt
  37.105505      -0.000937        -0.031157    -3.800891

# # # # The Coefficient Values # # #
(Intercept)
   37.10551
         disp
-0.0009370091
         hp
-0.03115655
       wt
-3.800891
```

Regresyon Modeli için Denklem Oluşturun

Yukarıdaki kesişme ve katsayı değerlerine dayanarak matematiksel denklemi oluşturuyoruz.

```
Y = a+Xdisp.x1+Xhp.x2+Xwt.x3
or
Y = 37.15+(-0.000937)*x1+(-0.0311)*x2+(-3.8008)*x3
```

Yeni Değerleri tahmin etmek için Denklem uygulayın

Yer değiştirme, beygir gücü ve ağırlık için yeni bir değerler kümesi sağlandığında kilometreyi tahmin etmek için yukarıda oluşturulan regresyon denklemini kullanabiliriz.

Disp = 221, hp = 102 ve wt = 2.91 olan bir otomobil için tahmini kilometre -

```
Y = 37.15+(-0.000937)*221+(-0.0311)*102+(-3.8008)*2.91 = 22.7104
```