

## Az R adatelemzési nyelv alapjai II.

### Gyakorlati feladat megoldása

#### Függvények

1. Írjon egy olyan függvényt, amely kiszámítja az argumentumban kapott szám faktoriálisát! Figyeljen az argumentum ellenőrzésére is.

```
f <- function( n )
{
  if( !is.numeric(n) )
    stop( "Nem szám." )

  stopifnot( n >= 0 )

  stopifnot( round(n) == n )

  if( n == 0 )
    return( 1 )

  result <- 1
  for( i in 1:n )
    result <- result * i

  result
}

# Példák a függvény meghívására
f(1)
## [1] 1

f(5)
## [1] 120

f(0)
## [1] 1
```

2. Írjon egy olyan függvényt, amely egy listában a hónapok neveivel indexelve visszaadja az argumentumban kapott évszámnak megfelelően az adott hónapba tartozó napok számát! Figyeljen a szökőévekre (4-el osztható, de 100-al nem osztható, kivéve a 400-al osztható évek).

```
m <- function(year)
{
  stopifnot( is.numeric(year), round(year) == year )
```

```
    leap <- F
    if((year %% 4) == 0) {
      if((year %% 100) == 0) {
        if((year %% 400) == 0) {
          leap <- T
        } else {
          leap <- F
        }
      } else {
        leap <- T
      }
    } else {
      leap <- F
    }
  }

  return( list(January=31, February=ifelse( leap, 29, 28), March=31) )
}
```

*# Példák a függvény meghívására*

`m(2000)`

```
## $January
## [1] 31
##
## $February
## [1] 29
##
## $March
## [1] 31
```

`m(2004)`

```
## $January
## [1] 31
##
## $February
## [1] 29
##
## $March
## [1] 31
```

`m(2100)`

```
## $January
## [1] 31
##
## $February
## [1] 28
##
## $March
## [1] 31
```

3. Írjon egy olyan függvényt, amely a bemenetül kapott adattábla (data.frame) alapján visszaad egy olyan új adattáblát, amely csak azokat a sorokat tartalmazza, amelyek nem fordul elő NA érték, illetve az argumentumban kapott oszlop szerint van sorrendezve. Függvény deklaráció: `complete.ordered.dataframe <- function( data = data, sort.by = sort.by )`

```
complete.ordered.dataframe <- function( data, sort.by )
{
  stopifnot( is.data.frame(data), is.character(sort.by), sort.by %in% colnames(data) )

  df <- data[complete.cases(data),]
  return( df[order(df[,sort.by]),] )
}
```

*# Példa a függvény meghívására*

```
complete.ordered.dataframe( data.frame(x=5:1,y=1:5), "x" )
```

```
##   x y
## 5 1 5
## 4 2 4
## 3 3 3
## 2 4 2
## 1 5 1
```

## Grafikonok létrehozása

```
df <- rbind( data.frame( Magassag = rnorm( n = 50, mean = 175, sd = 10 ),
                        Suly = rnorm( n = 50, mean = 80, sd = 10 ),
                        Nem = "Ferfi" ),
            data.frame( Magassag = rnorm( n = 50, mean = 165, sd = 8 ),
                        Suly = rnorm( n = 50, mean = 65, sd = 10 ),
                        Nem = "No" ) )

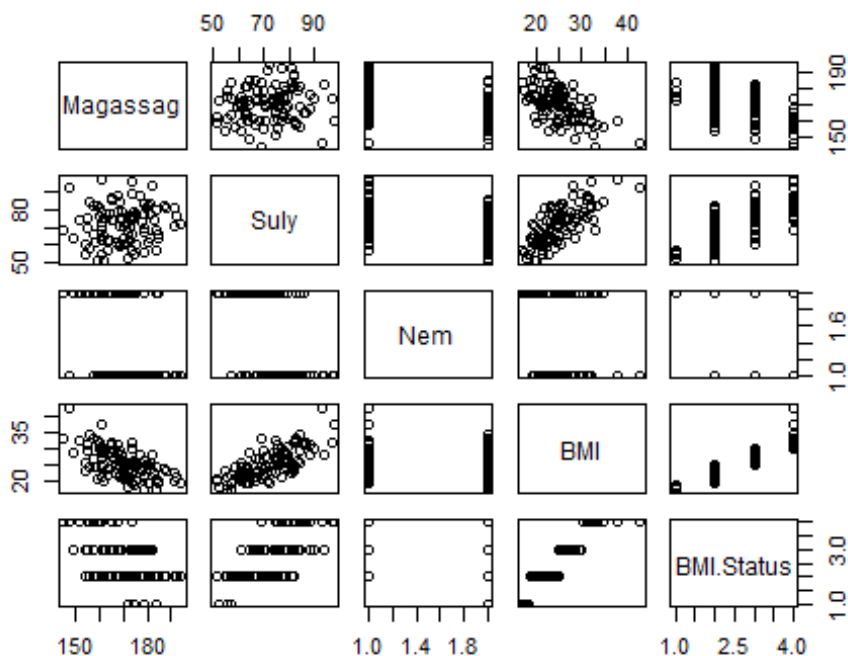
df$Nem <- as.factor( df$Nem )

df$BMI <- df$Suly / ( ( df$Magassag / 100 ) * ( df$Magassag / 100 ) )

df$BMI.Status <- cut( x = df$BMI,
                    breaks = c( 0, 18.5, 25, 30, Inf ),
                    labels = c( "Alultaplalt",
                                "Normalis",
                                "Tulsulyos",
                                "Elhizott" ) )
```

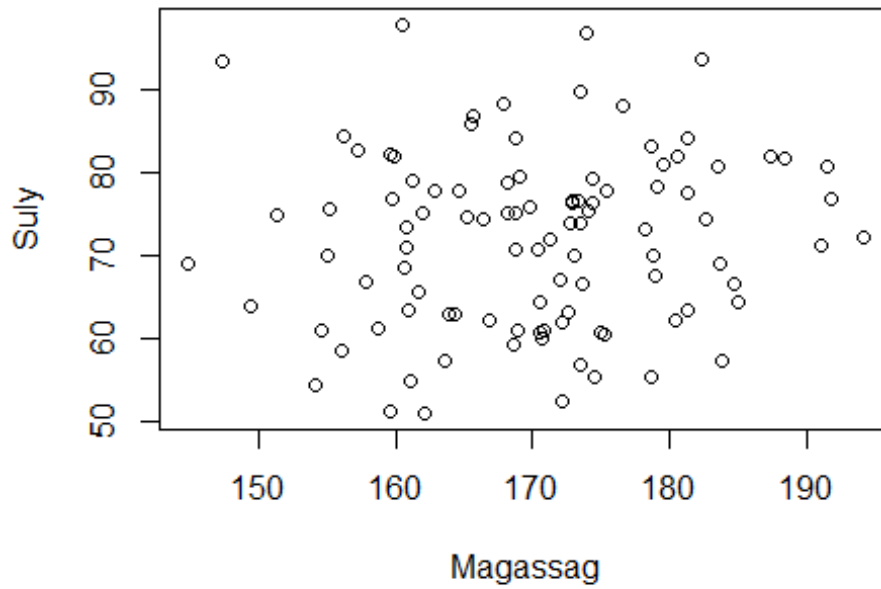
1. Ábrázolja az adattábla összes változóját az összes változó függvényében.

```
plot( df )
```



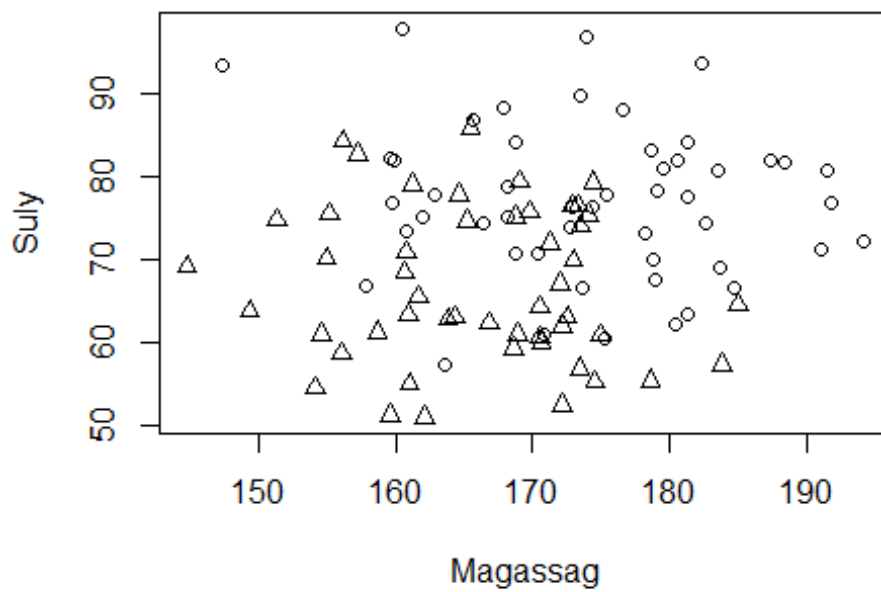
2. Ábrázolja a testsúlyt a testmagasság függvényében.

```
attach( df )
plot( Magassag, Suly )
```



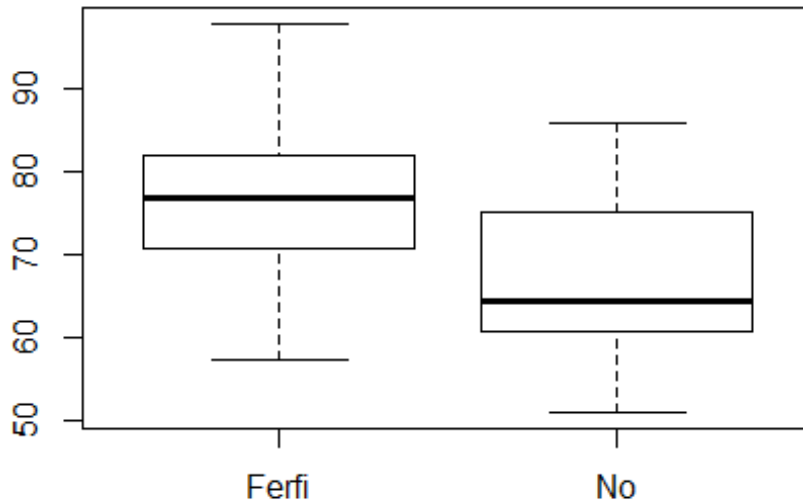
3. Ábrázolja a testsúlyt a testmagasság függvényében úgy, hogy nemeként különböző pont típussal ábrázolja az értékeket.

```
plot( Magassag, Suly, pch = as.numeric(Nem) )
```



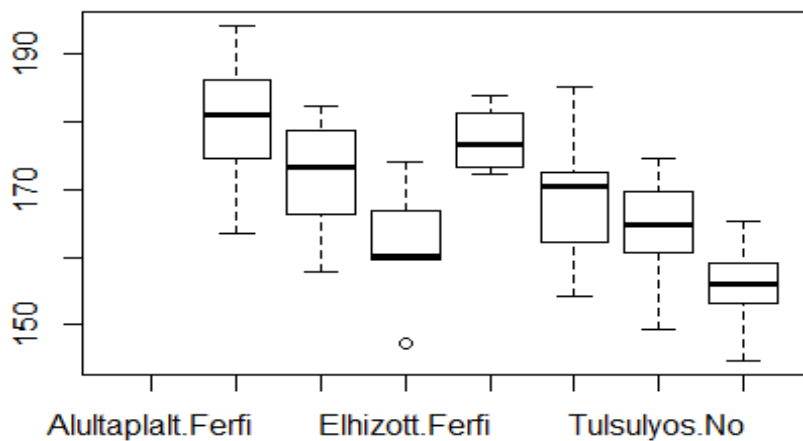
4. Ábrázolja dobozdiagramon a testsúly eloszlását nemenként.

```
boxplot( Suly ~ Nem )
```



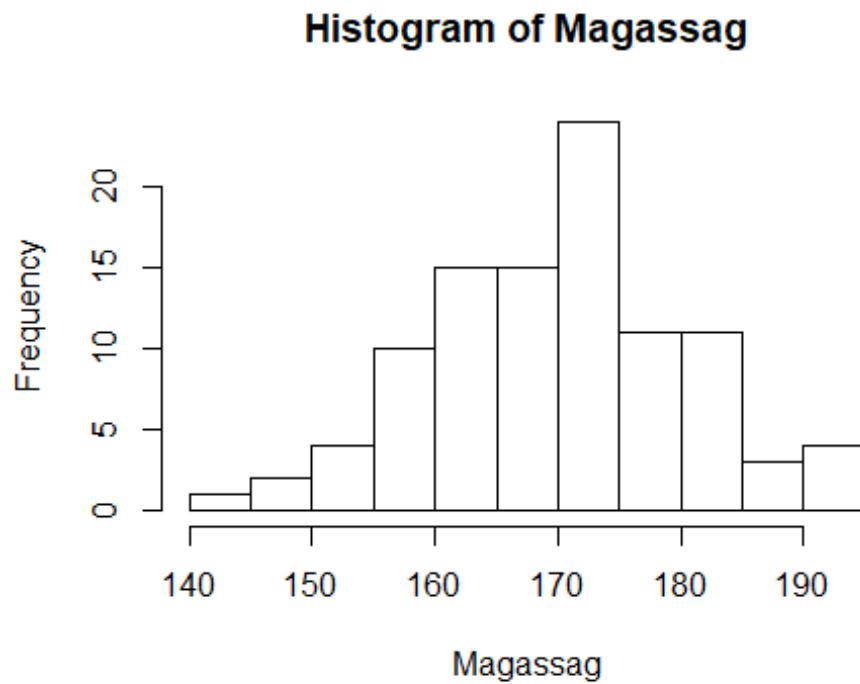
5. Ábrázolja dobozdiagramon a magasság eloszlását nemenként és BMI státuszonkénti bontásban.

```
boxplot( Magassag ~ BMI.Status + Nem )
```



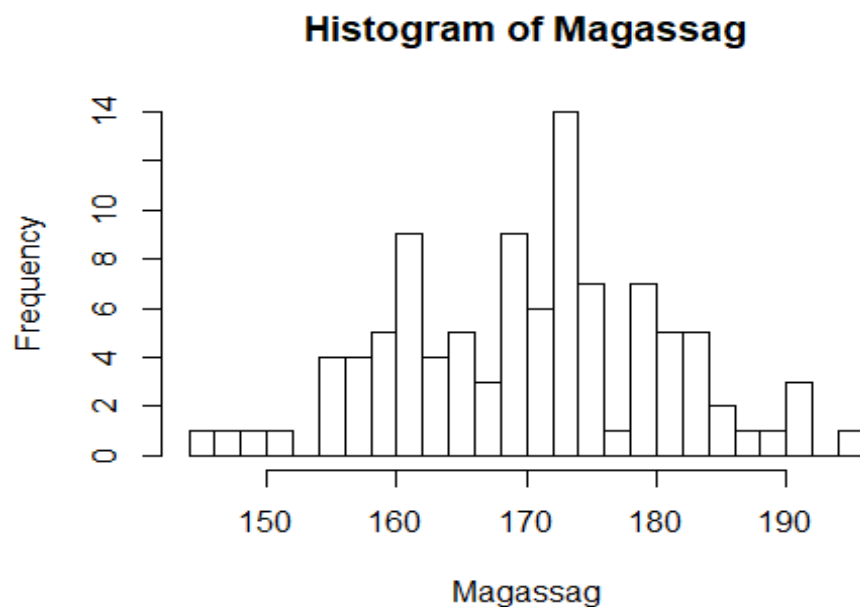
6. Ábrázolja a magasság eloszlását hisztogram segítségével.

```
hist( Magassag )
```



7. Ábrázolja a magasság eloszlását hisztogram segítségével úgy, hogy az oszlopok száma 20 legyen.

```
hist( Magassag, breaks=20 )
```



8. Ábrázolja a magasság és a testsúly eloszlását nemenként hisztogram segítségével úgy, hogy a 4 ábrát rendezze el egy 2\*2-es táblázatban. Figyeljen arra, hogy változónként azonos megjelenítési határokat alkalmazzon az x tengelyen.

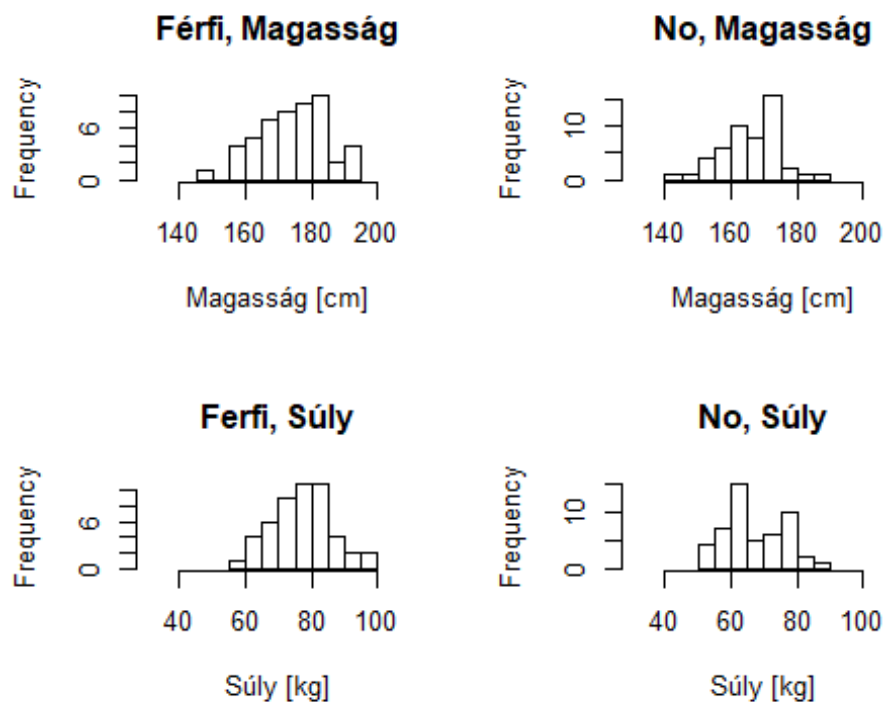
```
par(mfrow=c(2,2))
```

```
hist( Magassag[ Nem == "Ferfi"], xlim = c(130,210), main="Férfi, Magasság",  
xlab = "Magasság [cm]" )
```

```
hist( Magassag[ Nem == "No"], xlim = c(130,210), main="Nő, Magasság", xlab =  
"Magasság [cm]" )
```

```
hist( Suly[ Nem == "Ferfi"], xlim = c(30,110), main="Ferfi, Súly", xlab = "S  
úly [kg]" )
```

```
hist( Suly[ Nem == "No"], xlim = c(30,110), main="Nő, Súly", xlab = "Súly [kg  
]" )
```



```
par(mfrow=c(1,1))
```