Második gyak

Monos Attila

2022-10-03

Nevezetes abszolút folytonos eloszlások

Egy X valószínűségi változó eloszlása (nagyjából) akkor abszolút folytonos, ha X több, mint megszámlálhatóan végtelen sok értéket vehet fel. Rendszerint ez azt jelenti, hogy X egy intervallumból veszi fel az értékeit, vagy bármilyen valós szám lehet. A hivatalos definíció az, hogy van olyan $f(x)$ függvény, melyre igaz, hogy minden valós x-re $F(x) = \int_{-\infty}^{x} f(t)\mathrm{d}t$.

Mi csak a nevezetes abszolút folytonos eloszlásokkal fogunk foglalkozni, hiszen ezeknek van gyakorlati hasznuk. Egy abszolút folytonos eloszlást a sűrűségfüggvényével (vagy az eloszlásfüggvényével) adunk meg. A sűrűségfüggvény és az eloszlásfüggvény között az alábbi kapcsolatok fontosak:

$$F(x) = P(X < x) = \int\limits_{-\infty}^{x} f(t)\mathrm{d}t,$$
$$f'(x) = F(x),$$
$$f, F\colon \mathbb{R} \to \mathbb{R},$$

vagyis mind a sűrűségfüggvényt, mind az eloszlásfüggvényt a teljes valós számegyenesen értelmezzük (természetesen ha X valamilyen értéket nem is vehet fel, akkor ott a sűrűségfüggvény 0, az eloszlásfüggvény viselkedését pedig ismerjük).

Egyenletes eloszlás

Legyen $a < b \in \mathbb{R}$, és válasszunk ki véletlenszerűen egy x pontot az $[a, b]$ intervallumból. Ekkor – a gimnáziumban megismert geometriai valószínűséghez hasonlóan – annak a valószínűsége, hogy a kiválasztott pont az $E \subseteq [a, b]$ halmazba esik:

$$P(x \in E) = \frac{|E|}{\Big|[a, b]\Big|},$$

ahol a $|.|$ jelzés az adott halmaz hosszát jelöli (ha van). Alternatívan a sűrűségfüggvénnyel is megadhatjuk ezt a valószínűséget:

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{ha } x \in [a, b] \\ 0, & \text{ha } x \notin [a, b] \end{cases}$$

Ekkor $P(x \in E) = \int_E f(x)\mathrm{d}x$ (vegyük észre, hogy ha E egy intervallum, akkor ez egy sima Riemann-integrál. Ha E intervallumokra bomlik, akkor több Riemann-integrál összege). A fentiek alapján (akár integrálva is) megkapható az egyenletes eloszlás, $E(a, b)$ eloszlásfüggvénye is:

$$F(X) = P(X < x) = \begin{cases} 0, & \text{ha } x < a \\ \frac{x-a}{b-a}, & \text{ha } x \in [a, b] \\ 1, & \text{ha } x > b \end{cases}$$

Abszolút folytonos valószínűségi változók esetén az eloszlásfüggvényt a következő paranccsal hívhatjuk meg: `p` + a valószínűségi változó neve, pl:

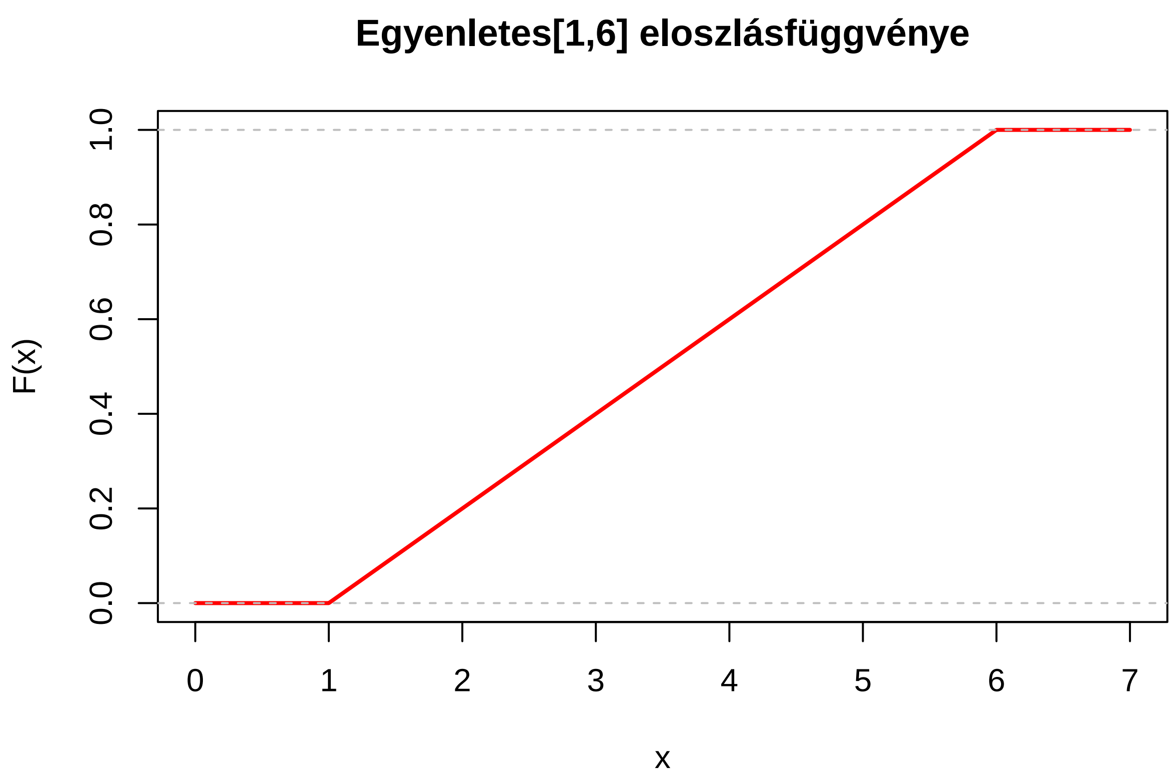
- `punif`
- `pnorm`
- `pt`

Természetesen ugyanígy vannak `dunif`…, `runif`…, `qunif`… parancsok is.

Legyen mostantól $X \sim E[1, 6]$.

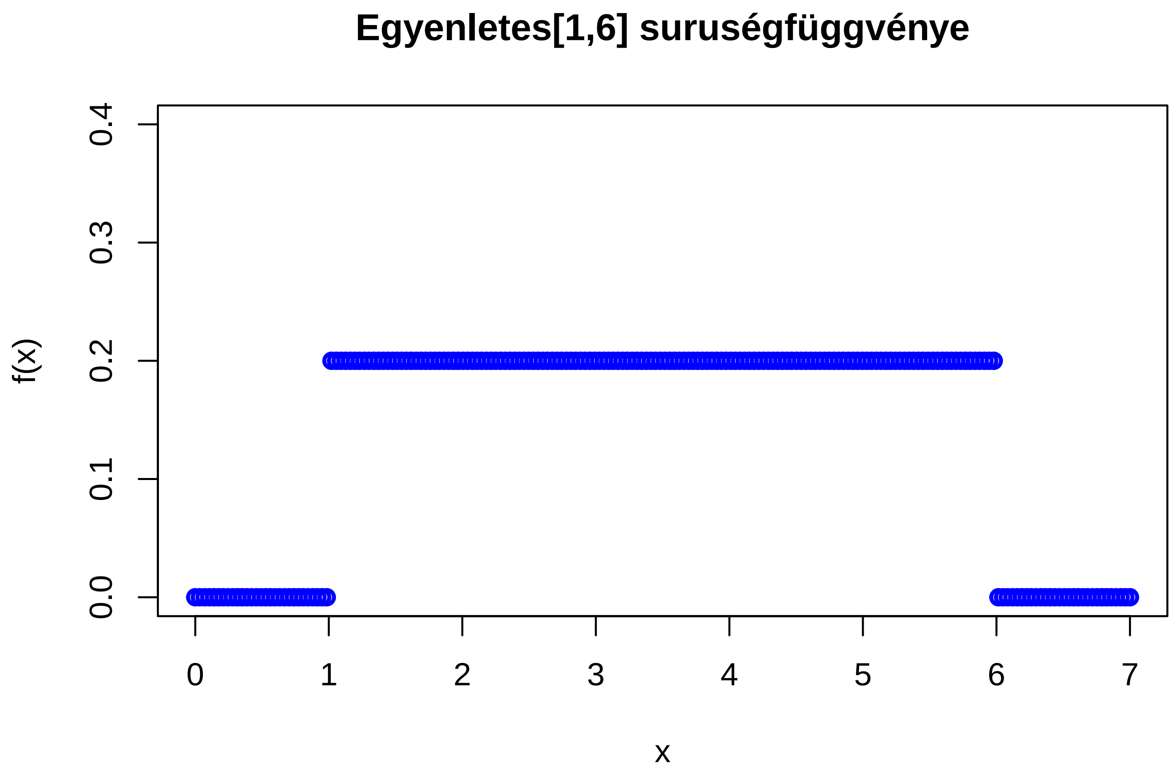
Eloszlásfüggvénye:

```
x_unif <- seq(0, 7, by = 1)
y_unif <- punif(x_unif, min = 1, max = 6)
plot(y_unif, type = "l", lwd = 2, col = "red", xaxt = "n", xlab = "x", ylab = "F(x)",
     main = "Egyenletes[1,6] eloszlásfüggvénye")
axis(1, at=1:8, labels=c(0:7))
abline(h = 0, col = "grey", lty = 2); abline(h = 1, col = "grey", lty = 2)
```



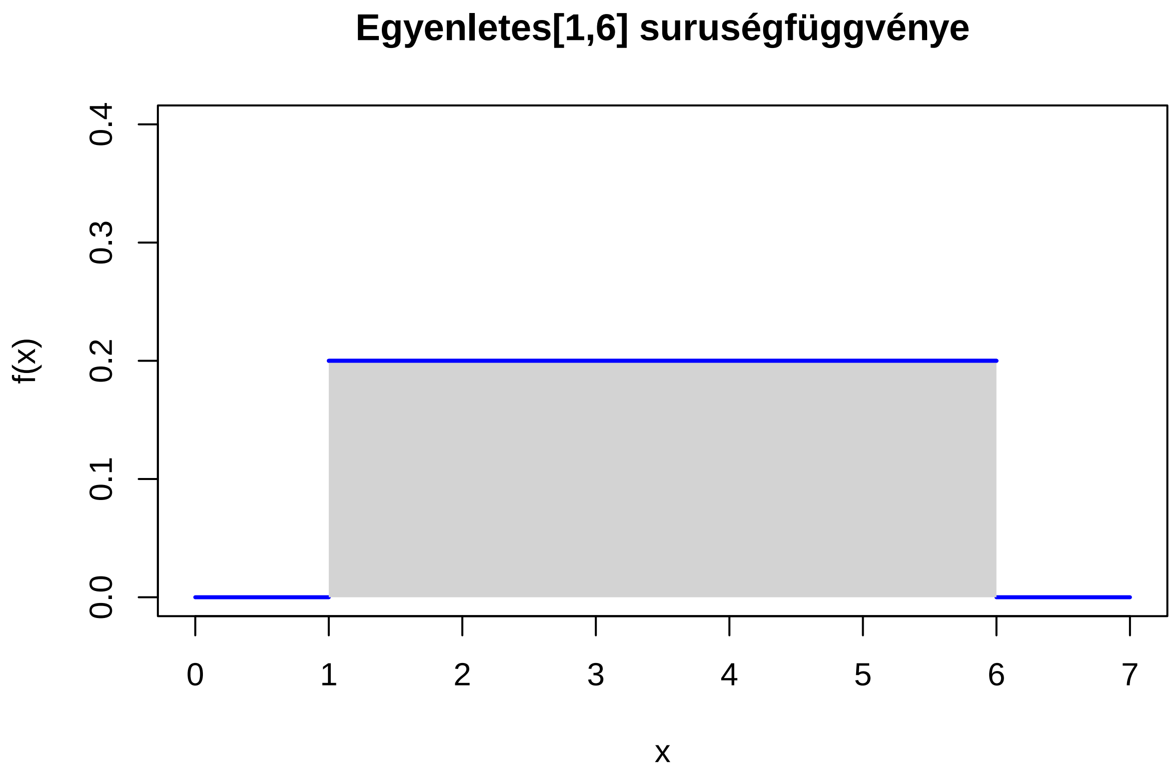
Sűrűségfüggvénye:

```
x <- seq(0, 7, length = 200)
y <- dunif(x, min = 1, max = 6)
plot(x, y, lwd = 2, col = "blue", ylim = c(0, 0.4), ylab = "f(x)",
     main = "Egyenletes[1,6] sűrűségfüggvénye")
```



Ugyanez szebb grafikával:

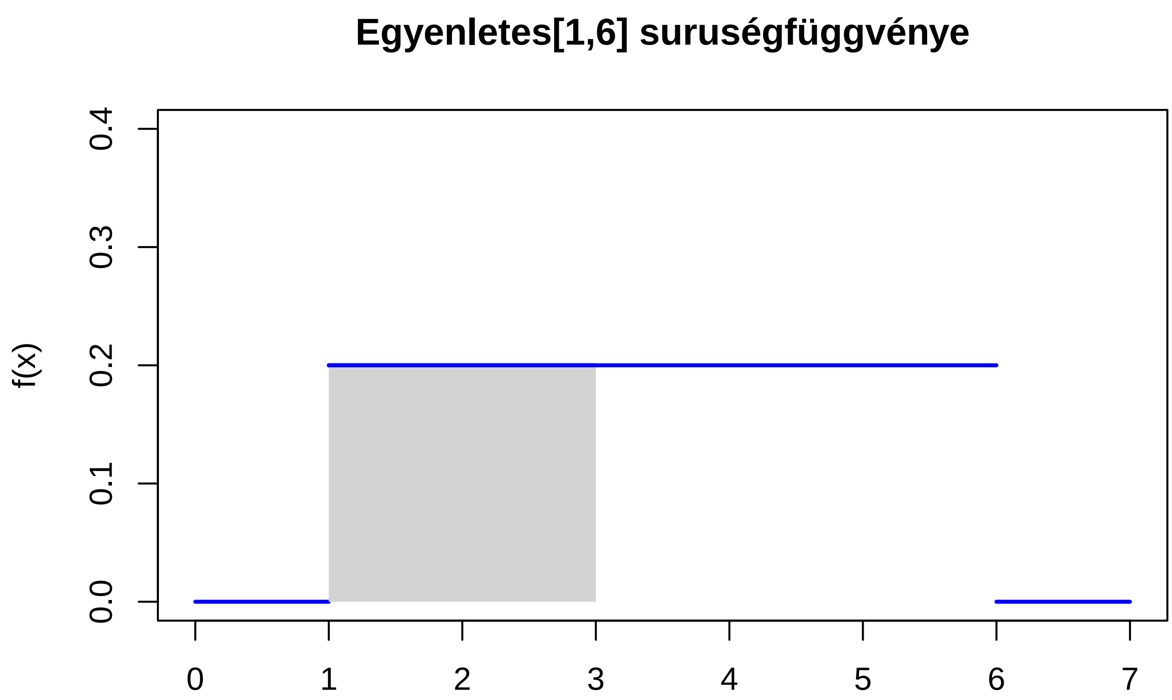
```
x <- seq(1, 6, length = 200)
y <- dunif(x, min = 1, max = 6)
plot(x, y, type = "l", lwd = 2, col = "blue", xlim = c(0,7), ylim = c(0, 0.4),
     ylab = "f(x)", main = "Egyenletes[1,6] sűrűségfüggvénye")

x_e <- c(seq(0, 1, length = 100))
y_e <- seq(0, 0, length = 100)
lines(x_e, y_e, type = "l", lwd = 2, col = "blue")

x_e <- c(seq(6, 7, length = 100))
y_e <- seq(0, 0, length = 100)
lines(x_e, y_e, type = "l", lwd = 2, col = "blue")

polygon(c(1,x,6),c(0,y,0),col = "lightgray", border = NA)
lines(x, y, type = "l", lwd = 2, col = "blue")
```



Egy függvényről úgy lehet ellenőrizni, hogy lehet-e sűrűségfüggvény, hogy megnézzük, hogy a függvény alatti terület 1 egység-e (azaz integrálunk). A függvény alatti területet ebben az esetben fent láthatjátok.

Ha $P(1 < x < 3)$-at szeretnénk kiszámolni, az is $f(x)$ integrálja, azaz egy függvény alatti terület:

```
x_poly = c(0, x, 7)
y_poly = c(0, y, 0.4)
plot(x_poly, y_poly, type = 'n', xlab = "", ylab = "f(x)", main = "Egyenletes[1,6] sűrűségfüggvénye")
polygon(c(1,x,6),c(0,y,0),col = "white", border = NA)
lines(x, y, type = "l", lwd = 2, col = "blue")

x_e <- c(seq(0, 1, length = 100))
y_e <- seq(0, 0, length = 100)
lines(x_e, y_e, type = "l", lwd = 2, col = "blue")

x_e <- c(seq(6, 7, length = 100))
y_e <- seq(0, 0, length = 100)
lines(x_e, y_e, type = "l", lwd = 2, col = "blue")

x_prob <- seq(1, 3, length = 100)
y_prob <- dunif(x_prob, min = 1, max = 6)
polygon(c(1, x_prob, 3), c(0, y_prob, 0), col = "lightgray", border = NA)
lines(x_prob, y_prob, type = "l", lwd = 2, col = "blue")
```



A szürke terület:

```
punif(3, min = 1, max = 6)
```

```
## [1] 0.4
```

1. Feladat:

A fenti kód alapján ábrázold a $P(3 < X < 4{,}5)$ valószínűségnek megfelelő területet, és a `punif` parancs segítségével számoljuk is ki!

```
#Insert R code here
```

A valószínűség:

```
punif(4.5, min = 1, max = 6) - punif(3, min = 1, max = 6)
```

```
## [1] 0.3
```

A `polygon` és `line` parancsok sajnos nem teljesen működnek PDF-be generálásnál, kell hozzá egy plot() parancs előtte. A fenti kódban van megoldás rá: egyszerűen a `type` argumentumba `'n'`-t írva kiplotolunk valami relevánsat, és meg van oldva a probléma.

##Exponenciális eloszlás Legtöbbször radioaktív részecskék bomlási idejét, élettartamot, vagy várakozási időt modelleznek vele. Paramétere $\lambda > 0$, sűrűségfüggvénye:

$$f(x) = \begin{cases} \lambda \cdot e^{-\lambda \cdot x}, & \text{ha } x > 0 \\ 0, & \text{ha } x \leq 0 \end{cases}$$

Így segíthet az R egy eloszlás megismerésében:

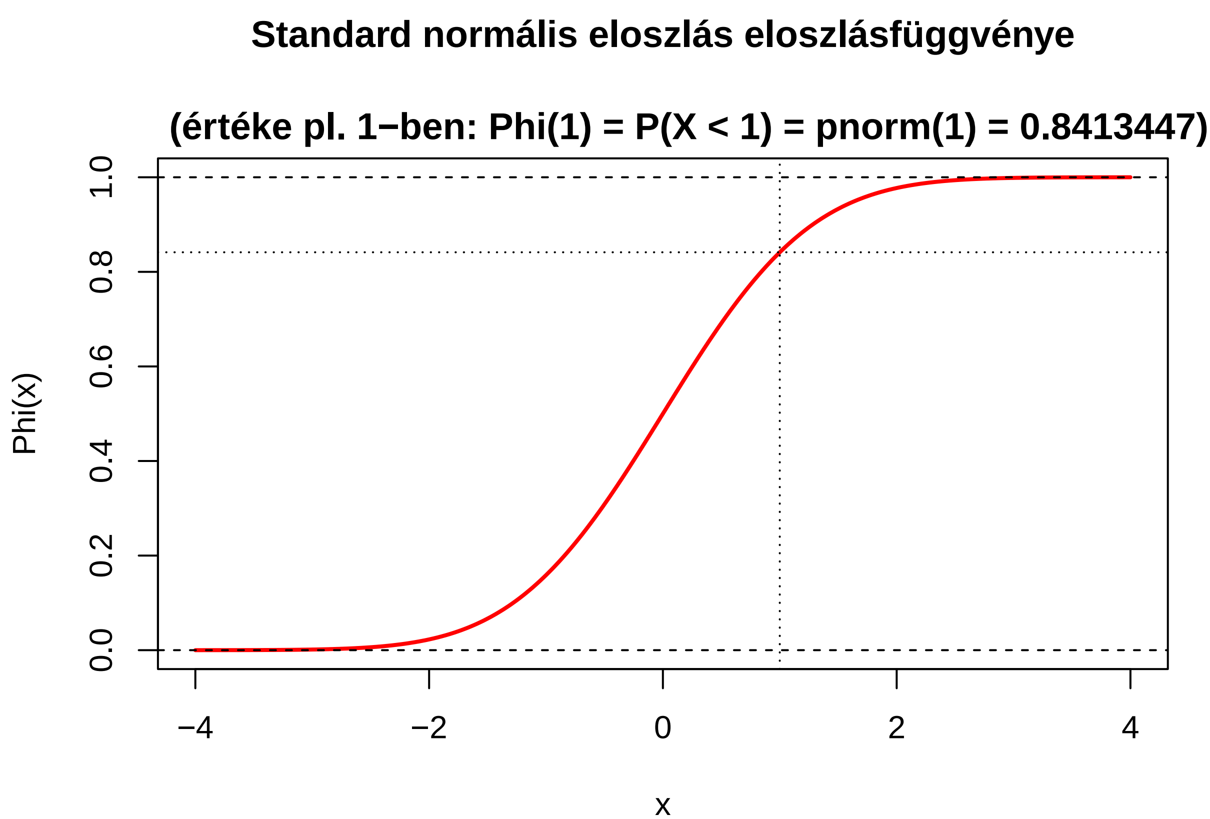
```
#?pexp
```

##Normális eloszlás Várható értékét m-el, szórását σ-val szoktuk jelölni. Ha visszaemlékszünk a CHT-re, akkor tulajdonképp nagy mintaszám esetén az összeget tekinthetjük normális eloszlásúnak. Sűrűségfüggvénye:

$$f(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(x-m)^2}{2\sigma^2}}$$

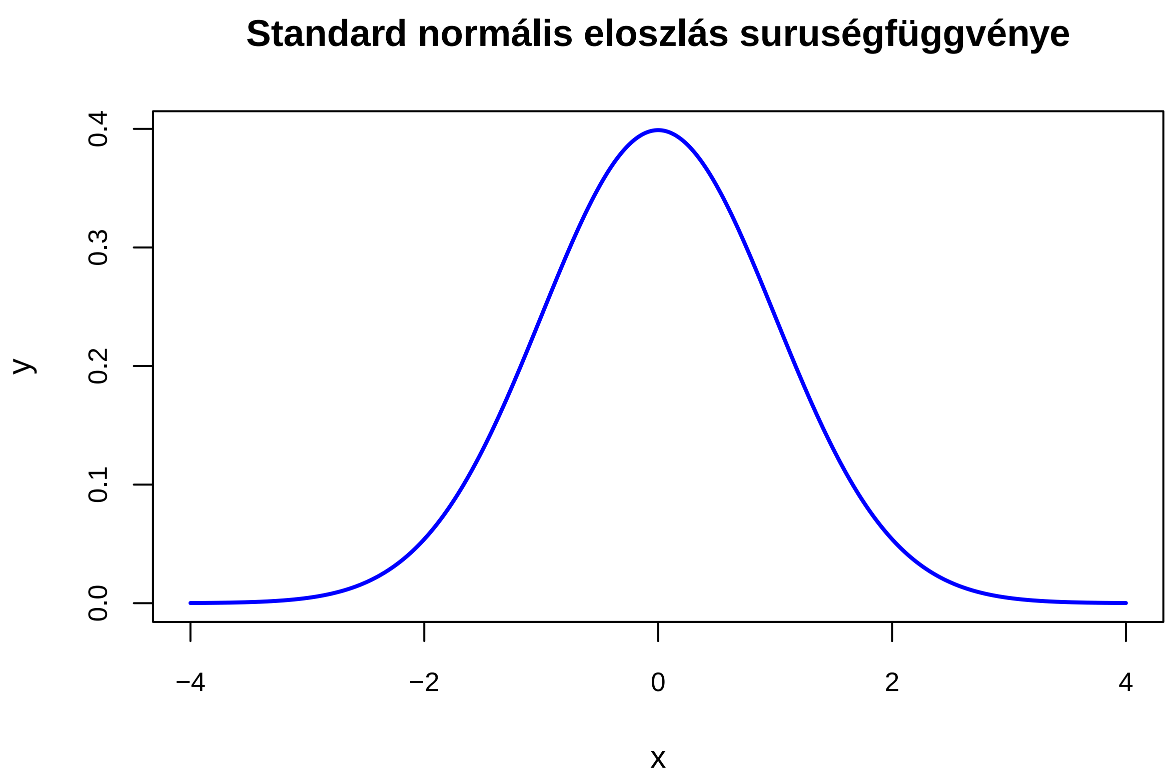
Testmértékeket, terméshozamot, vagy IQ-t szoktak vele modellezni (többek közt). Ha $m = 0, \sigma = 1$, akkor standard normális eloszlásról beszélünk: $X \sim N(0,1)$. Ennek az eloszlásfüggvénye, $\Phi(x) = \int_{-\infty}^{x} f(t)\mathrm{d}t$ nem elemi függvény, de van rá táblázat.

```
set.seed(2)
xseq <- seq(-4, 4, 0.01)
plot(xseq, pnorm(xseq, 0, 1), col = "red", type = "l", lwd = 2, xlab = "x",
     ylab = 'Phi(x)',
     main= "Standard normális eloszlás eloszlásfüggvénye\n
     (értéke pl. 1-ben: Phi(1) = P(X < 1) = pnorm(1) = 0.8413447)")
abline(h = c(0,1), lty = 2)
abline(v = 1, lty = 3)
abline(h = pnorm(1), lty = 3)
```



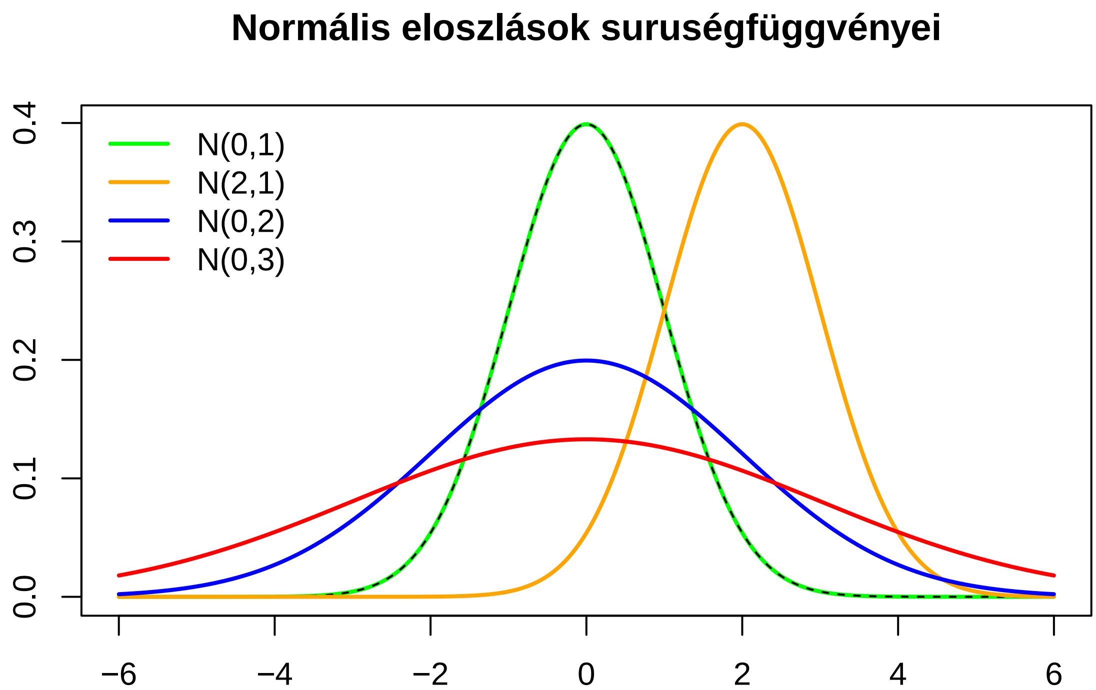
A standard normális eloszlás sűrűségfüggvénye:

```
density <- dnorm(xseq, 0, 1)
plot(xseq, density, type = "l", lwd = 2, col = "blue", cex.axis = 0.8,
     xlab = 'x', ylab = 'y',
     main = "Standard normális eloszlás sűrűségfüggvénye")
```



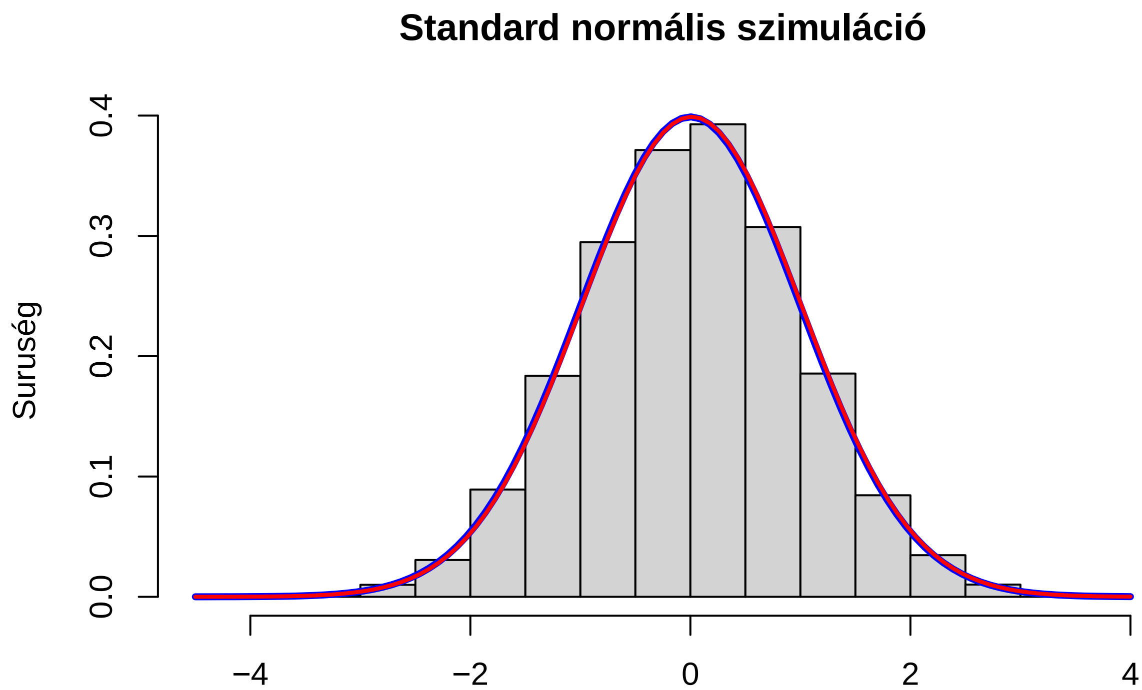
Nézzük most meg néhány nem standard normális eloszlás sűrűségfüggvényét (ér vele később játszadozni)!

```
x <- seq(-6, 6, 1/1000)
dnorm <- dnorm(x)

plot(x, dnorm, type = "l", col = "green", lwd = 2, ylab = '', xlab = '',
     main = "Normális eloszlások sűrűségfüggvényei")
legend(x = 'topleft', bty = 'n',
       legend = c("N(0,1)", "N(2,1)", "N(0,2)", "N(0,3)"),
       col = c("green", "orange", "blue", "red"), lwd = 2)

lines(x, dnorm, type = "l", col = "black", lty = 2)

dn2 <- dnorm(x, mean = 2, sd = 1)
lines(x, dn2, type = "l", col = "orange", lwd = 2)

dn3 <- dnorm(x, mean = 0, sd = 2)
lines(x, dn3, type = "l", col = "blue", lwd = 2)

dn4 <- dnorm(x, mean = 0, sd = 3)
lines(x, dn4, type = "l", col = "red", lwd = 2)
```



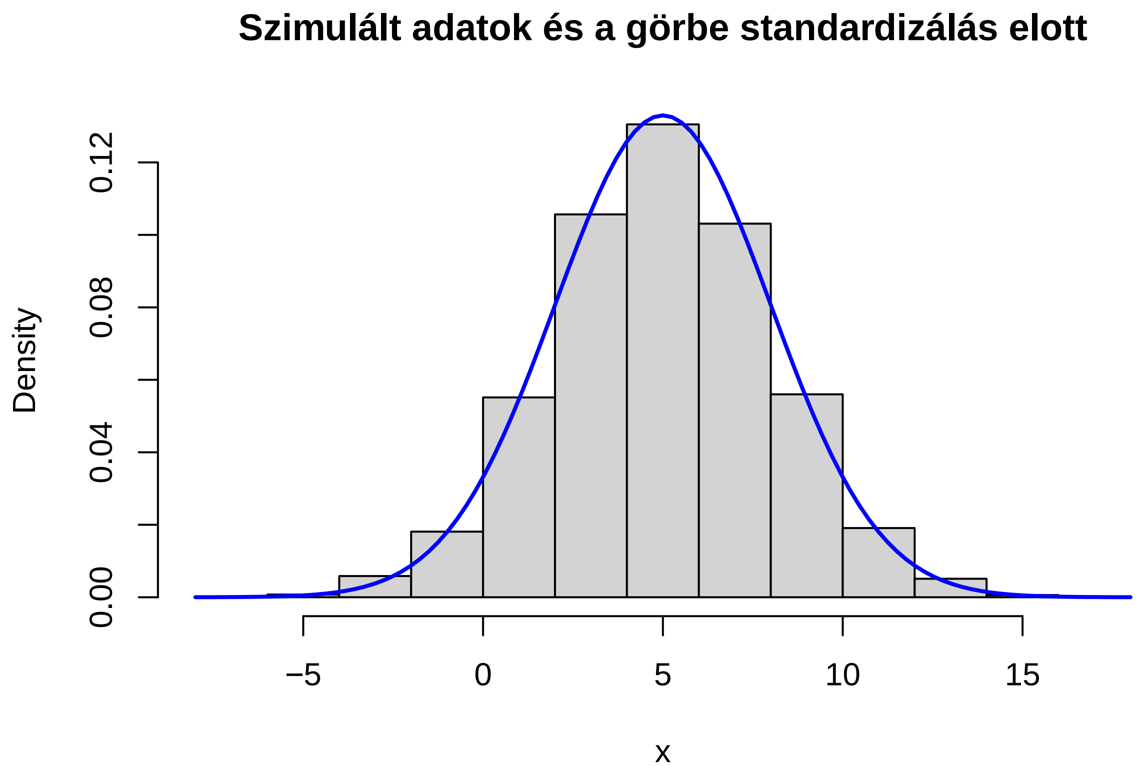
Szimulált adatok hisztogramja a sűrűségfüggvényhez képest:

```
sim <- rnorm(10000, 0, 1)
hist(sim, freq = FALSE, xlab = " ", ylab = "Sűrűség",
     main = "Standard normális szimuláció")
curve(dnorm(x, 0, 1), add = TRUE, col = "blue", lwd = 3.5)
curve(dnorm(x, mean(sim), sd(sim)), add = TRUE, col = "red", lwd = 2)
```



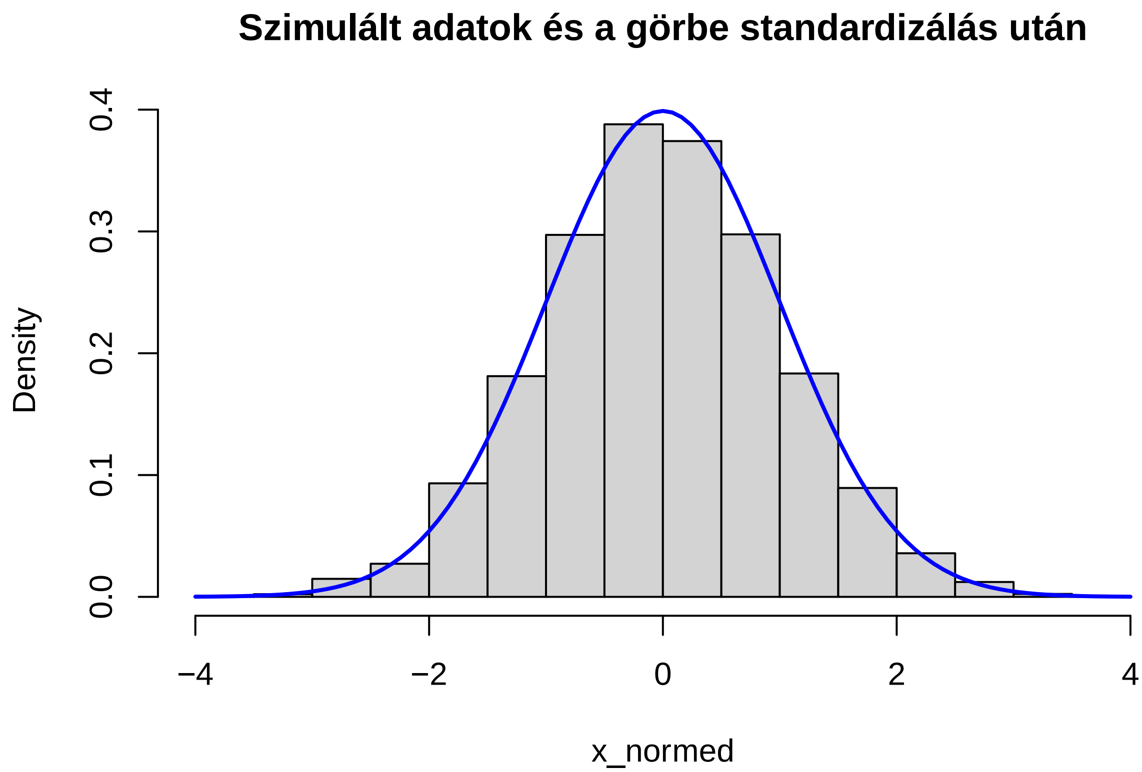
A normális valószínűségi változót szeretjük standardizálni, hiszen $N(0,1)$-nek a legkönnyebben számolható az eloszlásfüggvénye. Ez R-ben is így van:

```
x <- rnorm(10000, mean = 5, sd = 3)
hist(x, freq = FALSE, main = "Szimulált adatok és a görbe standardizálás előtt")
curve(dnorm(x, 5, 3), add = TRUE, col = "blue", lwd = 2)
```



Hasonlítsuk ezzel össze a standardizálás utáni eredményt (táblás gyakorlat alapján tudjuk, hogy a kettő ugyanaz kéne, hogy legyen)!

```
x_normed <- (x - 5) / 3
hist(x_normed, freq = FALSE, main = "Szimulált adatok és a görbe standardizálás után")
curve(dnorm(x, 0, 1), add = TRUE, col = "blue", lwd = 2)
```



2. Feladat:

Legyen $X \sim N(7, 3^2)$. Számoljuk ki $P(X > 8)$-at standardizálással és anélkül is!

```
#Insert R code here
```

3. Feladat:

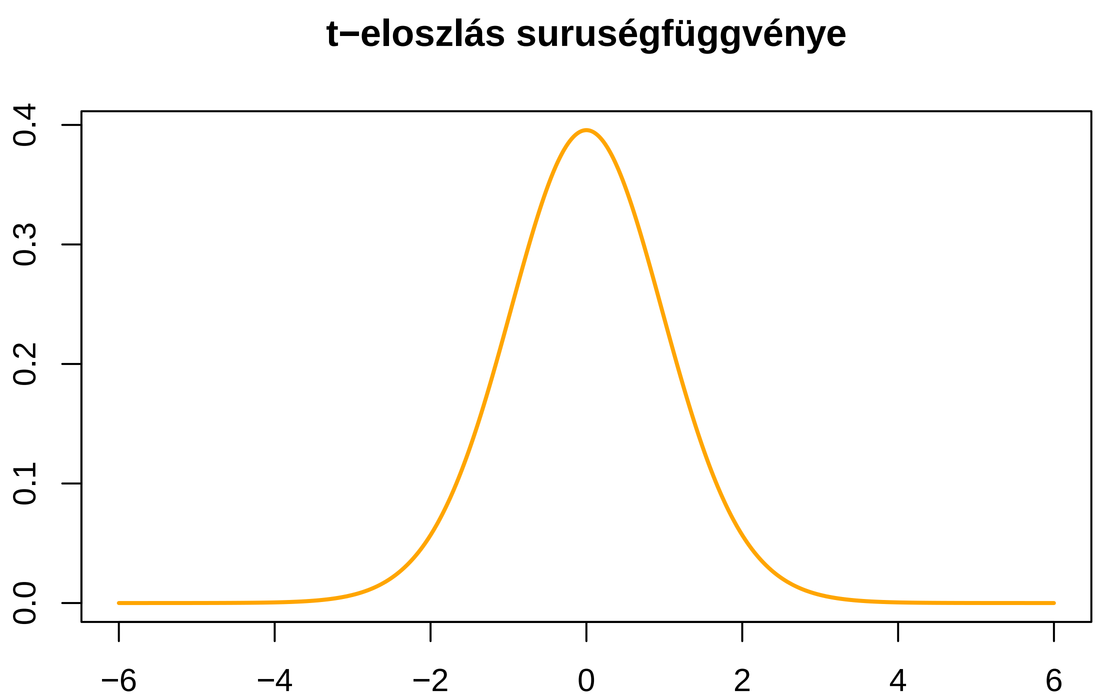
Az egyenletes eloszláshoz hasonlóan ábrázoljuk és számoljuk ki $P(-4 < X < -1)$-et, ha X standard normális eloszlású!

```
#x <- seq(-4, 4, length = 200)
#y <- dnorm(x, mean = 0, sd = 1)

### INSERT PLOT HERE

#x <- seq(-4, -1, lenght = 100)
#y <- dnorm(x, mean = 0, sd = 1)

### INSERT POLYGON HERE
```

t-eloszlás

Ismert még Student-eloszlásként is. Arra szoktuk használni, hogy egy normális eloszlású valószínűségi változó várható értékét becsüljük kis mintaszám és ismeretlen szórás esetén. A t-eloszlás fontosságáról később fogunk beszélni, táblás gyakorlatokon (remélem).

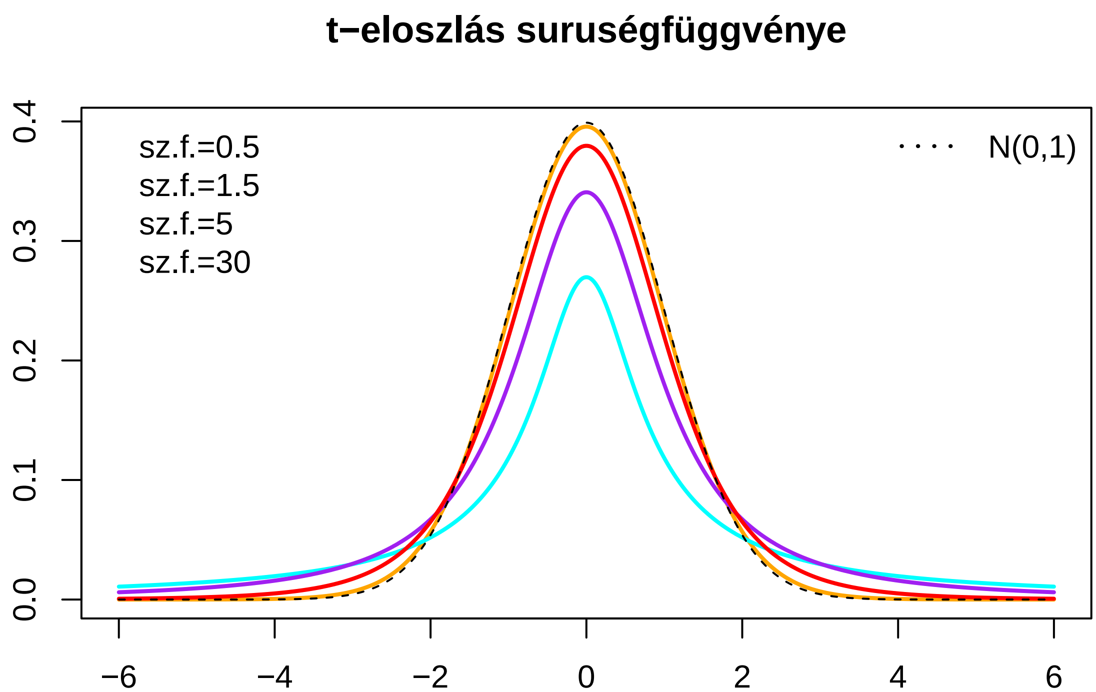
```
x <- seq(-6, 6, 1/1000)
dt4 <- dt(x, 30) #30: szabadsági fog, azaz degrees of freedom
plot(x, dt4, type = "l", col = "orange", lwd = 2, ylab = '', xlab = '',
     main = "t-eloszlás sűrűségfüggvénye")
```



Ez a götrbe meglepően hasonlít a normális eloszláséhoz. Nézzük meg különböző paraméterekkel a sűrűségfüggvényt, akkor is normális görbéhet hasonlót kapunk-e!

```
plot(x, dt4, type = "l", col = "orange", lwd = 2, ylab = '', xlab = '',
     main = "t-eloszlás sűrűségfüggvénye")

dt1 <- dt(x, 0.5)
lines(x, dt1, type = "l", col = "cyan", lwd = 2)

dt2 <- dt(x, 1.5)
lines(x, dt2, type = "l", col = "purple", lwd = 2)

dt3 <- dt(x, 5)
lines(x, dt3, type = "l", col = "red", lwd = 2)

normal <- dnorm(x)
lines(x, normal, type = "l", col = "black", lty = 2)

legend(x='topleft', bty='n',
       legend = c(paste('sz.f.', c(0.5, 1.5, 5., 30), sep='=')),
       #sz. f. jelentése: szabadsági fok
```

```
       col = c("cyan", "purple", "red", "orange"))

legend(x='topright', bty='n',
       legend = c(paste('N(0,1)', sep='=')),
       col = c("black"), lwd = 2, lty = 3)
```



Megfigyelhető, hogy ahogy növekszik a szabadsági fogk, úgy közelítünk a standard normálishoz.

4. Feladat:

A fentihez hasonlóan ábrázoljuk az exponenciális eloszlás sűrűségfüggvényét (`dexp` parancs segíthet ebben) $\lambda = 2.5, 2, 1, 0.5$ paraméterekkel! Mit figyelhetünk meg?

```
#Insert R code here
```

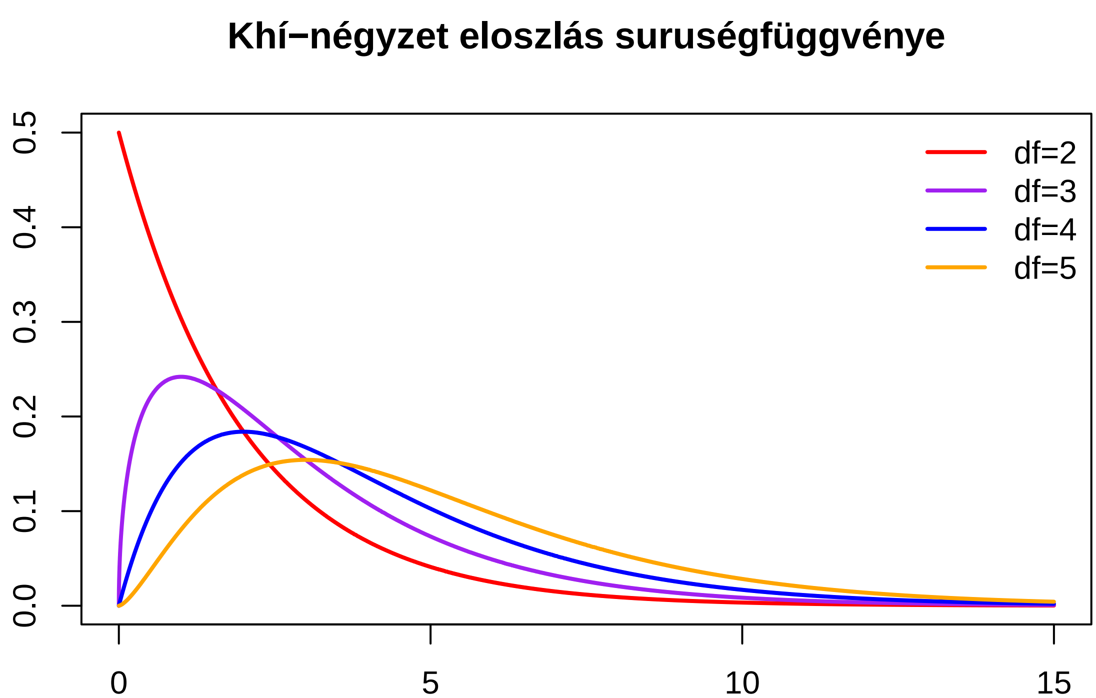
Khí-négyzet eloszlás

Ismert még Chi-squared néven is. Rövidebb jelölése : χ^2-eloszlás. Független standard normális eloszlások négyzetösszege – annyi darabé, amennyi a szabadsági foka. Rengeteg haszna van:

- Illeszkedés tesztelése
- Hipotézisvizsgálat
- Konfidenciaintervallumok megállapítása

Nézzük meg, különböző szabadsági fokokra hogy néz ki a χ^2-eloszlás!

```
x<- seq(0, 15, 1/1000)

dchisq1 <- dchisq(x, df = 2)
#df stands for degrees of freedom
plot(x, dchisq1, type = "l", col = "red", lwd = 2, ylab = '', xlab = '',
     main = "Khí-négyzet eloszlás sűrűségfüggvénye")

dchisq2 <- dchisq(x, df = 3)
lines(x, dchisq2, type="l", col = "purple", lwd = 2)

dchisq3 <- dchisq(x, df = 4)
lines(x, dchisq3, type="l", col = "blue", lwd = 2)

dchisq4 <- dchisq(x, df = 5)
lines(x, dchisq4, type="l", col = "orange", lwd = 2)

legend(x = 'topright', bty = 'n',
       legend = c(paste('df', c(2,3,4,5), sep = '=')),
       col = c("red", "purple", "blue", "orange"), lwd = 2)
```

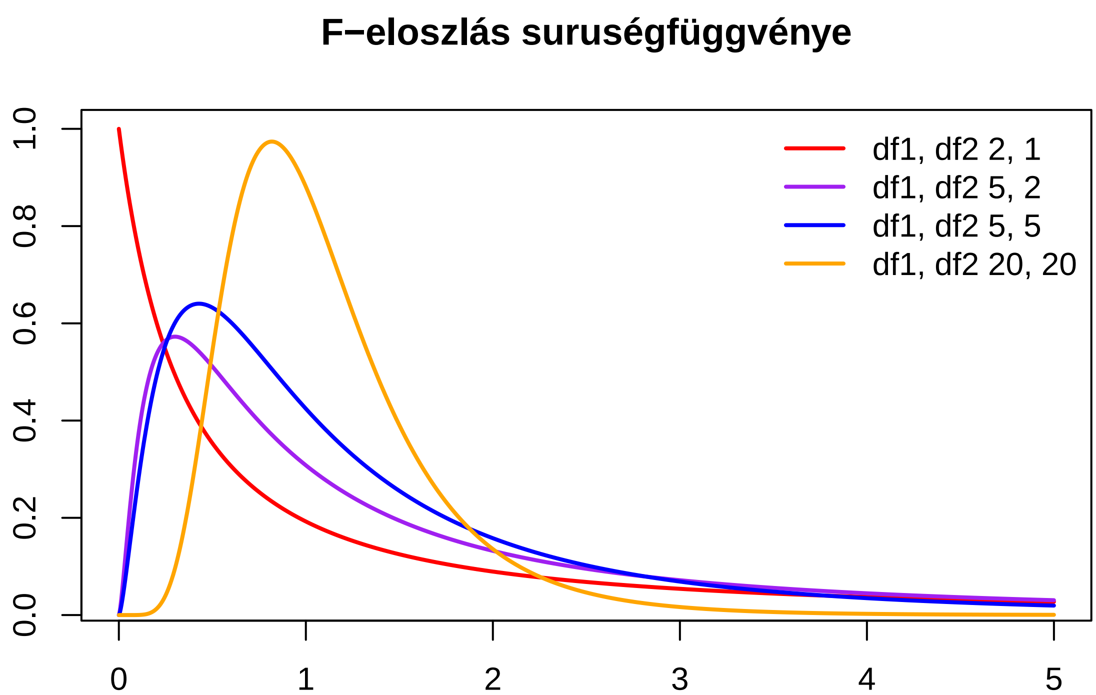


F-eloszlás

Két független χ^2-eloszlás hányadosából származik úgy, hogy a hányadosképzés előtt leosztjuk az eloszlásokat a szabadsági fokukkal. Legnagyobb haszna az, hogy két egymástól független, ismeretlen szórású normális eloszlásról meg tudjuk az F-eloszlással állapítani, hogy a szórásuk egyezik-e. Ha nem egyezik a két szórás, akkor nem nagyon tudunk mit tenni; ha pedig egyezik a szórás, akkor tudjuk vizsgálni a várható értékeket is, és utána már tudunk dolgozni az eloszlásokkal.

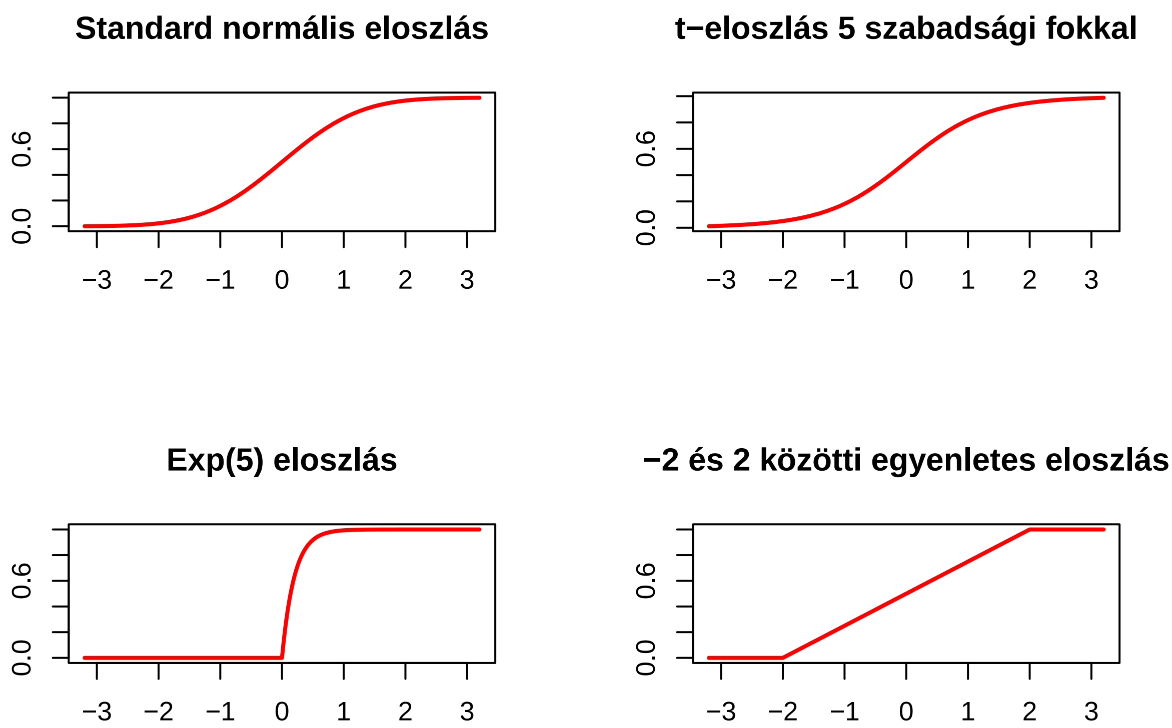
Az F-eloszlások (legalábbis néhány):

```
x <- seq(0, 5, 1/1000)

dF1 <- df(x, df1 = 2, df2 = 1)
plot(x, dF1, type = "l", col = "red", lwd = 2, ylab = '', xlab = '',
     main = "F-eloszlás sűrűségfüggvénye")

dF2 <- df(x, df1 = 5, df2 = 2)
lines(x, dF2, type = "l", col = "purple", lwd = 2)

dF3 <- df(x, df1 = 5, df2 = 5)
lines(x, dF3, type = "l", col = "blue", lwd = 2)

dF4 <- df(x, df1 = 20, df2 = 20)
lines(x, dF4, type = "l", col = "orange", lwd = 2)

legend(x = 'topright', bty = 'n',
       legend = c(paste('df1, df2', c("2, 1", "5, 2", "5, 5", "20, 20"))),
       col = c("red", "purple", "blue", "orange"), lwd = 2)
```



Néhány nevezetes abszolút folytonos eloszlásfüggvény

```
par(mfrow = c(2,2))
x_sample <- seq(-3.2, 3.2, 0.001)

plot(x_sample, pnorm(x_sample), type = "l", col = "red", lwd = 2,
     main = "Standard normális eloszlás", xlab = "", ylab ="")

plot(x_sample, pt(x_sample, 5), type = "l", col = "red", lwd = 2,
     main = "t-eloszlás 5 szabadsági fokkal", xlab = "", ylab ="")

plot(x_sample, pexp(x_sample, 5), type = "l", col = "red", lwd = 2,
     main = "Exp(5) eloszlás", xlab = "", ylab ="")

plot(x_sample, punif(x_sample, -2, 2), type = "l", col = "red", lwd = 2,
     main = "-2 és 2 közötti egyenletes eloszlás", xlab = "", ylab ="")
```

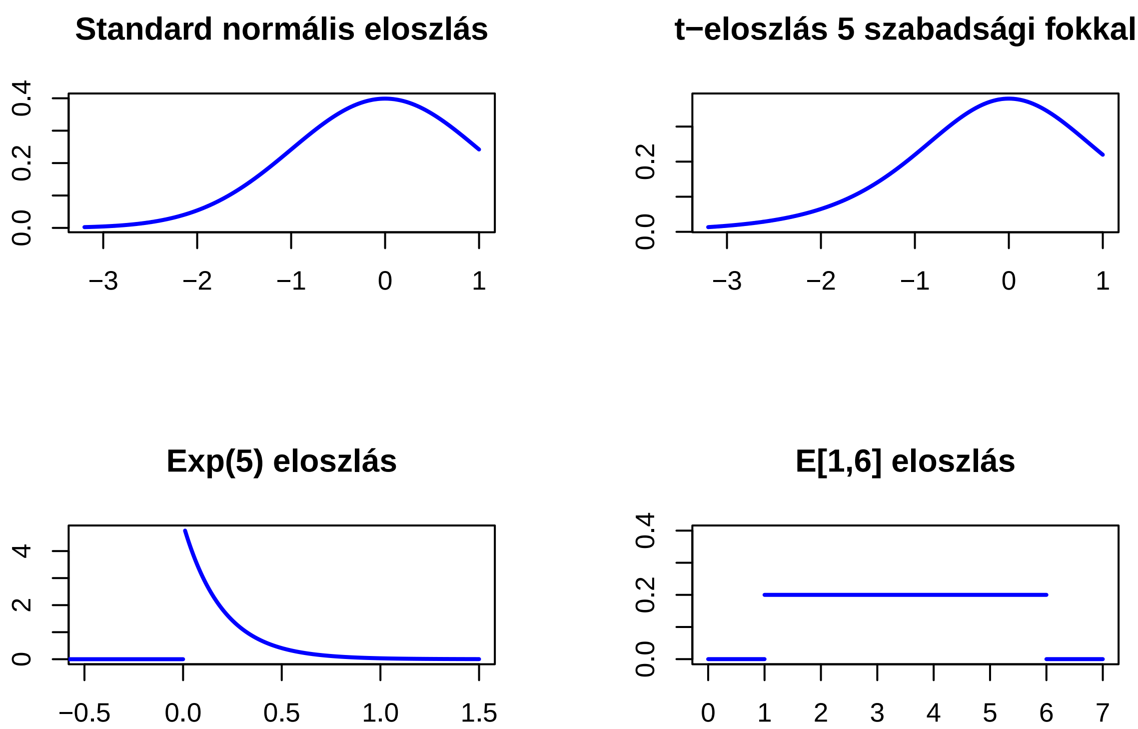


Néhány nevezetes abszolút folytonos sűrűségfüggvény

```
par(mfrow = c(2,2))
x_sample <- seq(-3.2, 3.-2, 0.001)

plot(x_sample, dnorm(x_sample), type = "l", col = "blue", lwd = 2,
     main = "Standard normális eloszlás", xlab = "", ylab = "")

plot(x_sample, dt(x_sample, 5), type = "l", col = "blue", lwd = 2,
     main = "t-eloszlás 5 szabadsági fokkal", xlab = "", ylab = "")

x_sample <- c(seq(0.01, 1.5, 0.001))
plot(x_sample, dexp(x_sample, 5), xlim = c(-0.5, 1.5), type = "l", col = "blue",
     lwd = 2, main = "Exp(5) eloszlás", xlab = "", ylab = "")
x_e <- c(seq(-3, 0, length = 100))
y_e <- seq(0,0, length = 100)
lines(x_e, y_e, type = "l", lwd = 2, col = "blue")

x <- seq(1, 6, lenght = 200)
```

```
## Warning: In seq.default(1, 6, lenght = 200) :
##  extra argument 'lenght' will be disregarded
```

```
y <- dunif(x, min = 1, max = 6)
plot(x, y, type = "l", lwd = 2, col = "blue", main = "E[1,6] eloszlás",
     xlim = c(0, 7), ylim = c(0, 0.4), ylab = '', xlab = '')
x_e <- c(seq(0,1, length = 100))
y_e <- seq(0, 0, length = 100)
lines(x_e, y_e, type = "l", lwd = 2, col = "blue")
x_e <- c(seq(6,7, length = 100))
y_e <- seq(0, 0, length = 100)
lines(x_e, y_e, type = "l", lwd = 2, col = "blue")
```




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

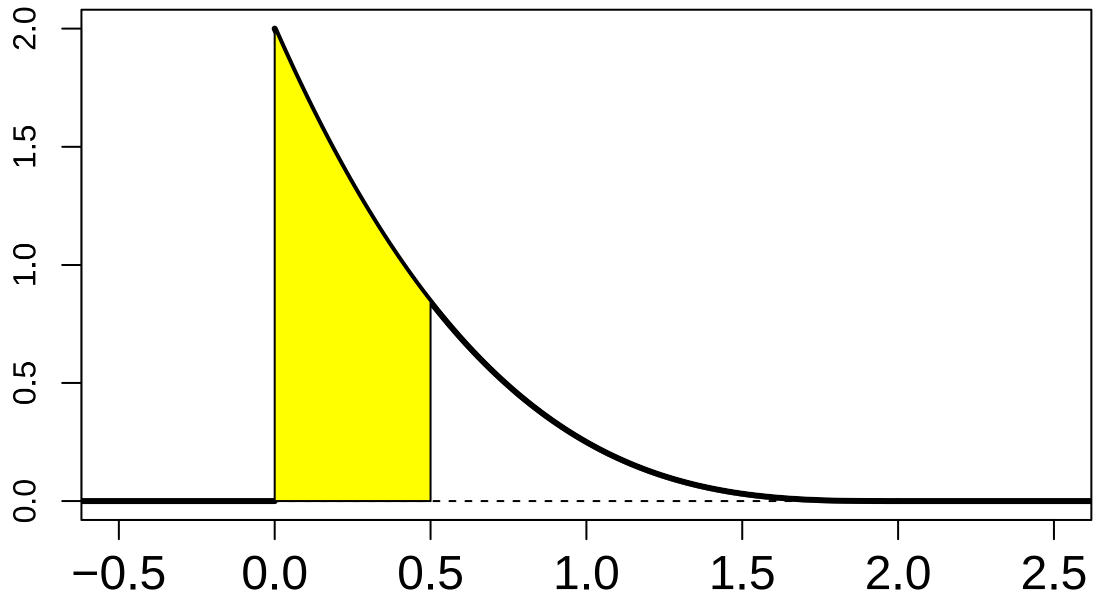
Érdemes az egyes eloszlások eloszlás-, és sűrűségfüggvényeit összehasonlítani!

5. Feladat:

Mit rajzol ki az alábbi program? Mutasd meg, hogy az alábbi függvény sűrűségfüggvény! Mit jelöl a besatírozott terület? Számold ki és ábrázold az eloszlásfüggvényt!

```
x <- seq(0, 2, 1/100) ;y <- 1/4*(2-x)^3

x_a <- seq(-1, 0, 1/100); y_a <- rep(0, length(x_a))
x_b <- seq(2, 3, 1/100); y_b <- rep(0, length(x_b))

plot(x, y, type = "l", col = "black", lwd = 3, ylab = '', xlab = '',
```

```
     xlim = c(-0.5, 2.5), xaxt = "n", main = " ")
axis(1, cex.axis = 1.5)
lines(x_b, y_b, type = "l", col = "black", lwd = 3)
lines(x_a, y_a, type = "l", col = "black", lwd = 3)
abline(h = 0, lty = 2)
x1 <- seq(0, 0.5, 1/100)
y1 <- 1/4*(2-x1)^3
polygon(c(0, x1, 0.5), c(0, y1, 0), col = "yellow")
```



Nagy számok törvénye

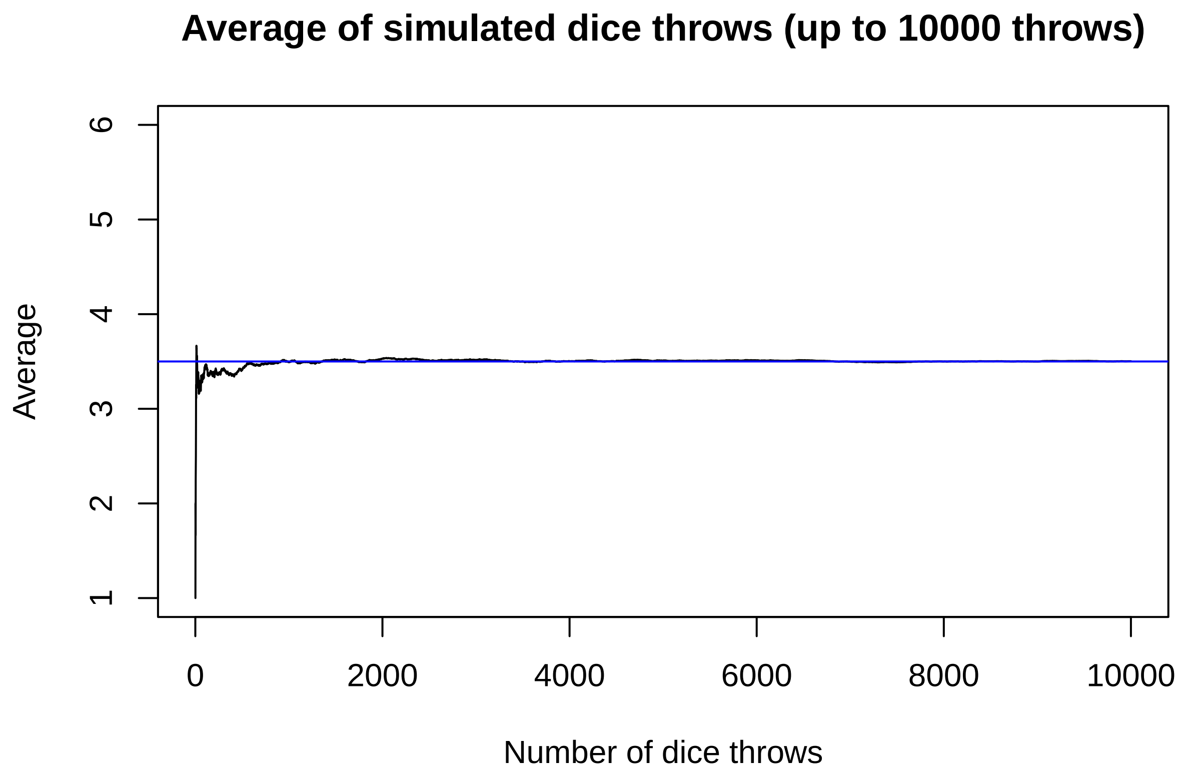
Ha $X_1, X_2, \ldots$ független azonos eloszlású valószínűségi változók és $EX_1 = m < \infty$, akkor

$$\frac{X_1 + \ldots + X_n}{n} \xrightarrow{n\to\infty} m \quad \text{1 valószínűséggel}$$

Kockadobással szemléltetve (érdemes n értékét változtatni):

```
dice <- 1:6
n <- 10000

throws <- sample(dice, size = n, replace = TRUE)
avgs <- cumsum(throws) / 1:n
plot(avgs, xlab = "Number of dice throws", ylab = "Average",
     ylim = c(1,6), type = "l",
```

```
    main = paste("Average of simulated dice throws (up to", n, "throws)"),)
abline(h = 3.5, col = "blue")
```



Itt tökéletesen láthatjuk, hogy a Nagy Számok Törvénye valóban teljesül: minél nagyobb mintából veszünk átlagot, annál közelebb leszünk a valódi várható értékhez. Későbbi tananyagban talán benne lesz, de a momentumbecslés nevű módszer pl. pont így becsli a vizsgált eloszlás várható értékét.

6. Feladat:

Demonstráljuk a Nagy Számok Törvényét tetszőlegesen paraméterezett biomiális és normális eloszlások esetén is (ezt úgy értem, hogy a paraméterek beállíthatóak általatok – ér is játszadozni vele, hogy paramétertől függetlenül működni fog)!

```
#Insert R code here for binomial distribution
```

```
#Insert R code here for normal distribution
```

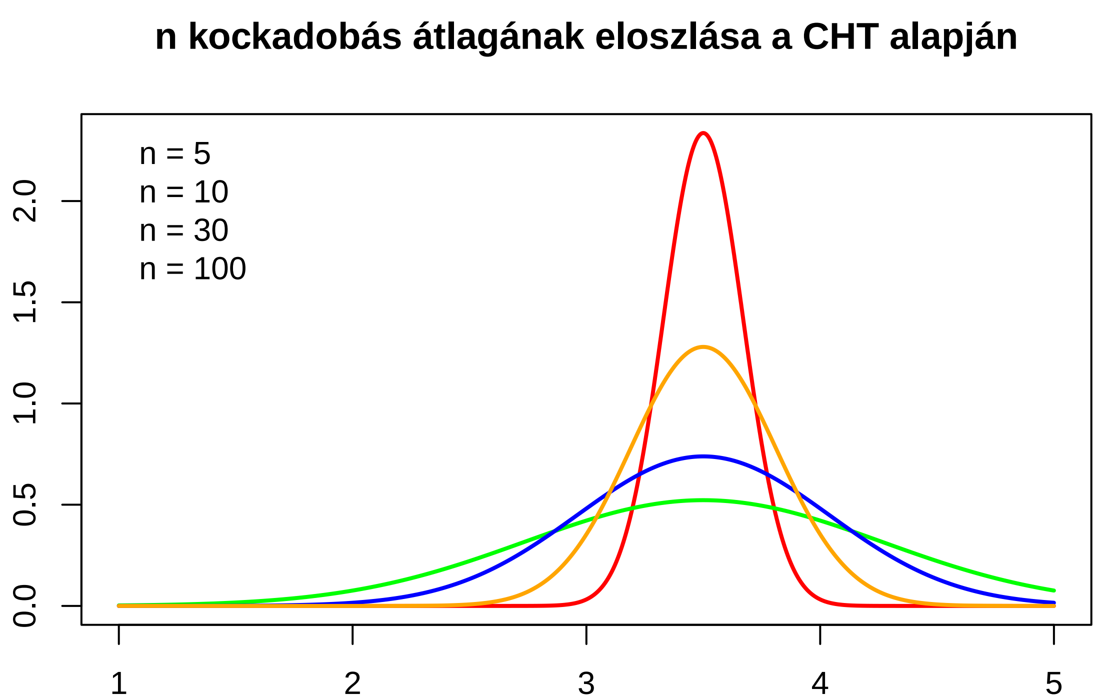
Centrális határeloszlás tétel

Ha $X_1, X_2, \dots$ független azonos eloszlású valószínűségi változók, $EX_1 = m$ és $D^2 X_1 = \sigma^2 < \infty$, akkor

$$P\left(\frac{X_1 + \ldots + X_n - n \cdot m}{\sqrt{n}\sigma} < x\right) \xrightarrow{n \to \infty} \Phi(x)$$

Gyakorlatban ez azt jelenti, hogy elég nagy minta esetén számolhatunk normális eloszlással. Demonstráljuk ezt a kockadobásos példával! CHT alapján n kockadobás átlagának eloszlása:

```
n <- 100
x_1 <- seq(1, 5, 0.001)
plot(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(n)),
     type = "l", col = "red", lwd = 2, ylab = "", xlab = "",
     main = "n kockadobás átlagának eloszlása a CHT alapján")
lines(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(5)),
      type = "l", col = "green", lwd = 2)
lines(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(10)),
      type = "l", col = "blue", lwd = 2)
lines(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(30)),
      type = "l", col = "orange", lwd = 2)
legend(x = 'topleft', bty = 'n',
       legend = c(paste(c("n = 5", "n = 10", "n = 30", "n = 100"), sep='=')),
       col = c("green", "blue", "orange", "red"))
```



Szimuláljuk most a kockadobásokat!

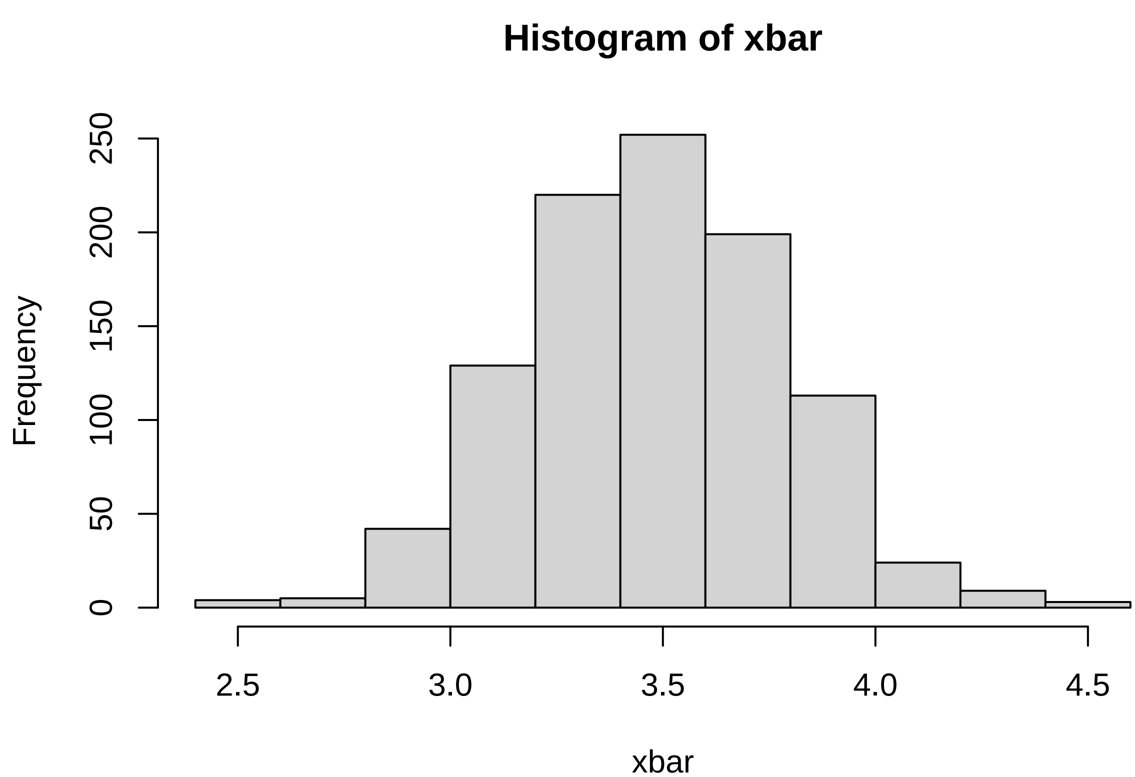
```
x <- 1:6
throw_num <- 30
rep <- 1000

A <- matrix(sample(x, throw_num*rep, replace = T), ncol = throw_num, byrow = TRUE)
xbar <- apply(A, 1, mean) #Az n dobás átlagát számolja ki

head(cbind(A, xbar))
```

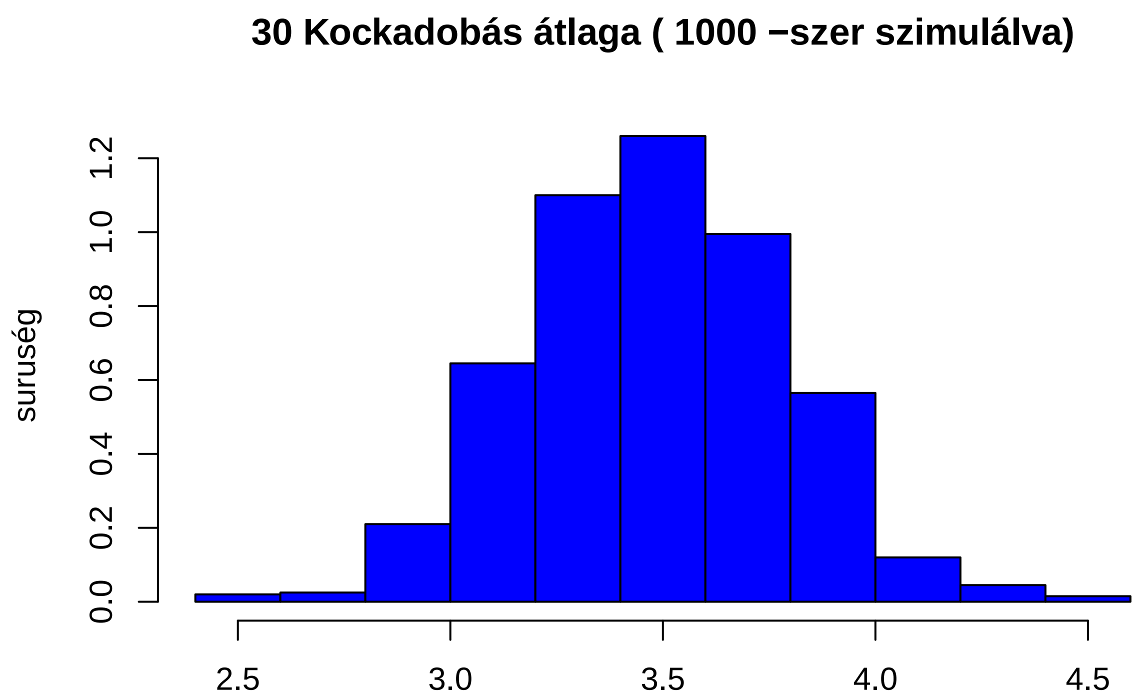
```
##                                                                       xbar
## [1,] 3 3 3 6 1 4 6 3 4 1 5 4 1 5 4 6 1 4 5 6 4 2 1 6 4 5 1 2 4 5 3.633333
## [2,] 6 6 5 5 4 1 5 5 4 6 3 5 4 4 4 1 2 6 6 4 1 4 1 4 1 2 2 4 5 3 3.766667
## [3,] 1 5 6 4 3 5 2 2 2 5 1 4 2 2 5 6 6 1 3 1 4 1 5 1 6 4 4 5 5 5 3.533333
## [4,] 4 3 6 1 1 4 2 3 6 3 3 3 6 5 4 3 3 6 1 1 5 5 6 2 4 6 6 3 1 4 3.666667
## [5,] 6 1 3 3 6 3 1 2 6 1 3 1 3 6 3 6 5 3 1 4 2 1 3 1 4 2 2 4 2 6 3.133333
## [6,] 4 1 4 4 1 2 5 1 2 4 6 6 2 5 6 5 6 1 6 2 5 6 3 2 1 4 2 1 5 5 3.566667
```

```
tail(cbind(A, xbar))
```

```
##                                                                          xbar
##   [995,] 2 6 1 6 3 2 6 4 4 3 2 3 3 4 1 3 1 1 5 1 3 2 5 3 3 4 4 5 1 3 3.133333
##   [996,] 2 6 5 3 3 1 4 6 2 6 1 3 6 6 3 2 1 3 5 1 5 5 3 3 2 6 3 2 4 6 3.600000
##   [997,] 2 2 1 5 2 4 2 2 4 6 4 1 1 5 3 5 3 3 1 5 2 5 3 3 2 1 5 1 6 4 3.100000
##   [998,] 4 5 2 5 1 5 3 4 5 2 1 3 4 2 2 6 2 5 1 4 1 4 4 1 2 5 6 5 1 4 3.300000
##   [999,] 6 6 4 5 5 1 3 2 1 2 5 1 4 4 2 5 1 4 3 1 4 2 1 4 4 3 6 1 1 5 3.200000
## [1000,] 4 5 1 1 2 1 2 5 1 3 2 3 4 5 3 2 4 2 6 6 3 2 3 4 3 1 1 2 2 6 2.966667
```

```
hist(xbar, col = "blue", freq = F, xlab = "", ylab = "sűrűség",
     ylim = range(0, max(hist(xbar)$density,
                         dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(throw_num)))),
     main = paste(throw_num, "Kockadobás átlaga (", rep, "-szer szimulálva)"))
```

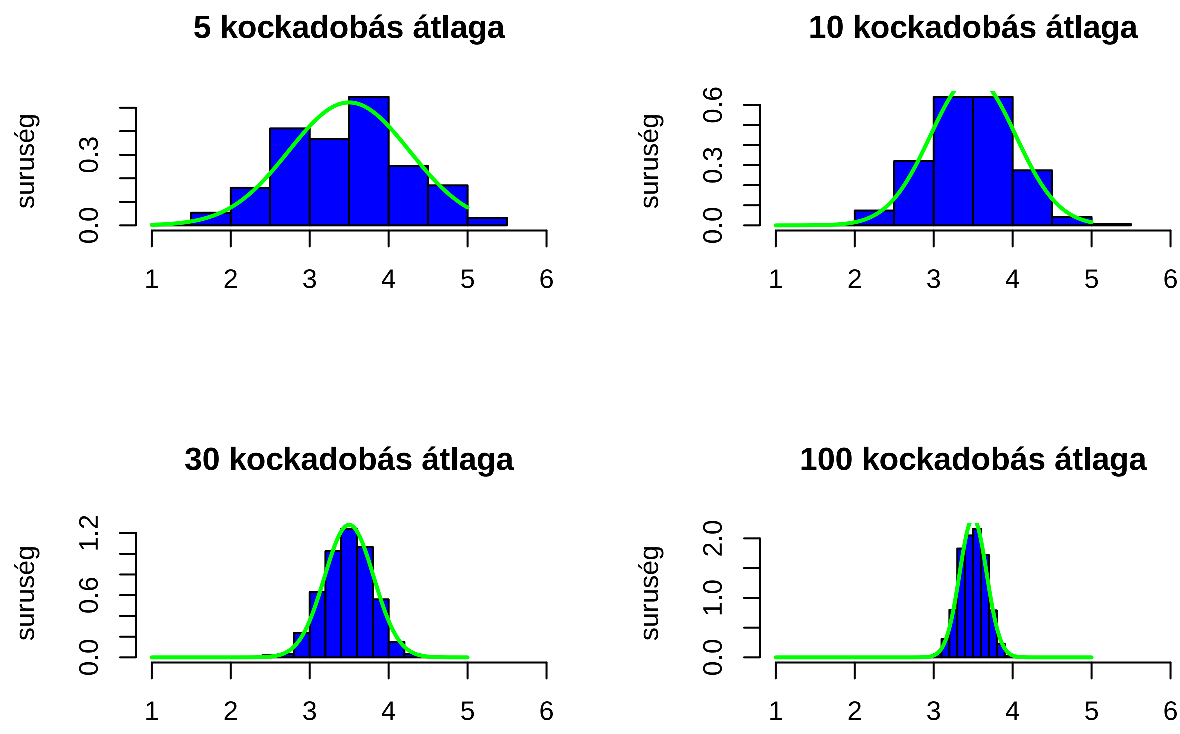




Érdemes a szimulált eredményeket összevetni a kapott normális eloszlásgörbével. Nézzük meg, a különböző értékekre a CHT adta normális görbe mennyire simul a hisztogramra!

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

hist(apply(matrix(sample(x, 5*rep, replace = T), ncol = 5, byrow = T),1, mean),
     col = "blue", freq = F, xlim = c(1,6), xlab = "", ylab = "sűrűség",
     main = "5 kockadobás átlaga")
lines(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(5)), type = "l",
      col = "green", lwd = 2)

hist(apply(matrix(sample(x, 10*rep, replace = T), ncol = 10, byrow = T),1, mean),
     col = "blue", freq = F, xlim = c(1,6), xlab = "", ylab = "sűrűség",
     main = "10 kockadobás átlaga")
lines(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(10)), type = "l",
      col = "green", lwd = 2)

hist(apply(matrix(sample(x, 30*rep, replace = T), ncol = 30, byrow = T),1, mean),
     col = "blue", freq = F, xlim = c(1,6), xlab = "", ylab = "sűrűség",
     main = "30 kockadobás átlaga")
lines(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(30)), type = "l",
      col = "green", lwd = 2)

hist(apply(matrix(sample(x, 100*rep, replace = T), ncol = 100, byrow = T),1, mean),
     col = "blue", freq = F, xlim = c(1,6), xlab = "", ylab = "sűrűség",
     main = "100 kockadobás átlaga")
```

```r
lines(x_1, dnorm(x_1, mean = 3.5, sd = 1.707825/sqrt(100)), type = "l",
      col = "green", lwd = 2)
```



Ha a `par` parancsot bárhol kiadjuk, akkor az egészen a munkamenet végéig felülírja az adott paramétereket – ezért érdemes odafigyelni arra, hogy visszaállítsuk a paramétereket az alapértelmezett érték(ek)re.

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

Figyeljük meg, hogy a konvergencia elég gyors – már 100 minta esetén is eléggé közel vannak a szimulált adatok a normális eloszlás görbéjéhez. Gondoljunk bele, 1000 vagy 10000 minta esetén számottevő lehet-e egyáltalán az eltérés.

7. Feladat: Számold ki R segítségével az alábbiakat, ha $Z \sim N(0,1)$!

- $P(Z < 1{,}645)$
- $P(Z < z) = 0{,}95$, mennyi z értéke? Hint: `?pnorm` segíthet.
- $P(Z < -1.645)$
- $P(Z < z) = 0{,}05$, mennyi z értéke?
- $P(Z > 1{,}96)$

Legyen most $X \sin N(25, 3^2)$. Számoljuk ki az alábbiakat!

- $P(X < 33)$

- $P(X < x) = 0{,}95$, mennyi x értéke?
- $P(X < 21)$
- $P(X < x) = 0{,}05$, mennyi x értéke?
- $P(X > 22)$
- $P(23 < X < 25)$

További táblás gyakorlat feladatok előfordulhatnak a jövőben, érdemes nézni, frissül-e ez az állomány!