Változók típusai

Diszkrét (kategorikus)	Véges vagy legfeljebb megszámlálhatóan végtelen pl.: nem (férfi/nő), vércsoport, iskolai végzettség, ... <u>Eloszlása</u> Mik a változó lehetséges értékei, és azokat milyen gyakorisággal veszi fel

Folytonos	Adott intervallumon végtelen sok érték
	Pl.: testmagasság, csapadékmennyiség, ...
	
	<u>Eloszlásának jellemzői</u>
	1. Középpont
	- mintaátlag (x'): várható érték becslése
	- medián (Q2): aminél az adatok fele ≤
	- módusz: leggyakoribb mintaelem
	
	|min---Q1---Q2---Q3---max|
	
	2. Szóródás
	- terjedelem = max - min
	- IQR = Q3 - Q1
	- szórás: átlagosan milyen távol vannak az adatok a mintaátlagtól
	3. Alak
	- szimmetrikus (átlag = medián)
	- jobbra ferde (átlag > medián)
	`skewness() > 0`
	- balra ferde (átlag < medián)
	`skewness() < 0`

T-próba

n = elemszám
x' = mintaátlag
c = vizsgált érték
SD = minta szórása
SE = standard hiba = SD / √n
alfa = szignifikancia szint

```
elemszam = 80
mintaatlag = 118
mintaszoras = 12
teszt = 110
alpha = 0.05
```

		H0-t elutasítjuk <=>
Próbastatisztika alapján	t = (x' - c) / SE $t_{alfa/2}$: n-1 szabadsági fokú t-eloszlás 1-(alfa/2) kvartilise	**|t| > $t_{alfa/2}$**

```
egymintas_t = function(mu, n, xvonas, sd) {
  (xvonas-mu)/(sd/sqrt(n))
}

t = egymintas_t(teszt, elemszam, mintaatlag, mintaszoras)
t_alpha = qt(1-alpha/2, df = elemszam-1)
```

Konfidencia-intervallum alapján	KI = x' ± $t_{alfa/2}$ * SE	**c ∉ KI**

```
# 95% megbízhatóságú konfidencia intervallum a várható értékre
konf.int_also = function(n, xvonas, sd, alpha) {
  xvonas-qt(1-alpha/2, df = n-1)*sd/sqrt(n)
}
konf.int_felso = function(n, xvonas, sd, alpha) {
  xvonas+qt(1-alpha/2, df = n-1)*sd/sqrt(n)
}

konf.int_also(elemszam, mintaatlag, mintaszoras, alpha)
konf.int_felso(elemszam, mintaatlag, mintaszoras, alpha)
```

p-érték alapján		**p < alfa**

```
p = pt(abs(t), df = elemszam-1, lower.tail = F)*2
```

Elsőfajú, másodfajú hiba

	H0-t "elfogadjuk"	H0-t elutasítjuk
H0 igaz	ok	P(**Elsőfajú hiba**) = alfa
H0 hamis	P(**Másodfajú hiba**) = béta	ok
próba ereje: 1 - béta = P(elutasítjuk H_0-t | H_A)		

R tips & tricks

Adat beolvasása X táblából az Y csomagból	`library(Y)`
	`input = data.frame(X)`
	`attach(input) # így elérjük a mezőit könnyen`

Adott állomány beolvasása	`input = read.table("salary.txt", header = TRUE)`
	`bank = read.csv2("bankloan.csv")`
	`kings = scan("http://.../kings.dat", skip = 3)`
	`admission = read.csv(url)`

Pontbecslések	`mean(data)    # átlag`
	`sd(data)      # szórás`
	`var(data)     # variancia`
	`median(data)  # medián`

Alaki jellemzők	`library(moments)`
	`skewness(data)    # ferdeség`
	`kurtosis(data)    # lapultság`

Kvartilisek, min, max, terjedelem, IQR	`quantile(data)`
	`quantile(data, probs = c(0.25, 0.5, 0.75))`
	`max(data)`
	`min(data)`
	`max(data) - min(data)`
	`IQR(data)`
	
	`summary(data)`

Üres elemek kiszűrése	`result = data[!is.na(data)]`
Üres elemek számossága	`sum(is.na(data))`

Új táblázat létrehozása meglévő adatok alapján	`result = table(data1, data2)`
	
	`# hasznosabb`
	`result = data.frame(data1, data2)`

Pontbecslések valami szerint csoportosítva (Bármilyen függvény alkalmazása a cellákra)	`tapply(data, group, function, na.rm = T)`
	
	`# például az életkorok átlaga nemek szerint`
	`tapply(Age, Sex, mean)`

Kiugró értékek	`boxplot.stats(data)$out`

Sorbrarendezés	`res = data[order(data$by), ]`
	
	`# vagy vektor esetén`
	`res = data[order(data)]`

Szűrés valamilyen feltétel alapján	`data[data$by == val, ]`
Standard normális eloszlás generálása	`set.seed(1234)` `minta = rnorm(1000)`

Standardizálás, QQ-ábra	`std_salary = scale(salary)`
	`qqnorm(std_salary)`
	`qqline(std_salary)`

Vizsgálatok és megvalósításuk R-ben

Várható érték konkrét szám **μ = c**	Egymintás t-próba	t.test(var, mu = val)

```
# 5 százalékos szignifikancia szinten teszteljük azt a nullhipotézist, hogy
# az átlagos pulzusszám 75-tel egyenlő.

t.test(Pulse, mu = 75)
```

Összefüggő minták (= 2) **μ1 = μ2**	Páros t-próba	t.test(var1, var2, paired = T) t.test(var1-var2, mu = 0)

```
# Hasonlítsuk össze a két kéz fesztávolságát. 5 százalékos szignifikancia szinten
# teszteljük azt a nullhipotézist, hogy
# a két kéz átlagos fesztávolsága megegyezik.

t.test(Wr.Hnd, NW.Hnd, paired = T)
t.test(Wr.Hnd-NW.Hnd, mu = 0)
```

Független minták (= 2) **μ1 = μ2**	Kétmintás t-próba (F-próba + t-próba)	*var.test(var ~ group, conf.level = 0.99)* t.test(var ~ group, var.equal = T, conf.level = 0.99) t.test(var1, var2, var.equal = T, conf.level = 0.99)

```
# 1 százalékos szignifikancia szinten teszteljük azt a nullhipotézist, hogy
# az átlagos pulzusszám azonos férfiak és nők esetén.
# Adjunk meg egy 99 százalék megbízhatóságú konfidencia intervallumot
# a két várható érték különbségére.

var.test(Pulse ~ Sex, conf.level = 0.99)
t.test(Pulse ~ Sex, var.equal = T, conf.level = 0.99)
```

Független minták (> 2) **μ1 = μ2 = ... = μn**	Egyszempontos ANOVA (Levene-próba + ANOVA)	*leveneTest(var ~ group, center = mean)* oneway.test(var ~ group, var.equal = TRUE) *factor(group)*

```
# Vizsgáljuk meg a pulzusszámokat a testmozgás gyakorisága (Exer változó) szerint.
# Teszteljük azt a nullhipotézist, hogy a pulzusszámok várható értéke
# minden csoportban ugyanakkora.

library(car)
leveneTest(Pulse ~ Exer, center = mean)
summary(aov(Pulse ~ Exer))

oneway.test(Pulse ~ Exer, var.equal = TRUE)
```

Egy változó **normális eloszlást követ-e?**	Egymintás Kolmogorov-Szmirnov-próba	ks.test(var, "pnorm", mean(var), sd(var))

```
# A Kolmogorov-Szmirnov-próba alkalmazásával teszteljük le
# 5 százalékos szignifikanciaszinten azt a nullhipotézist, hogy
# a 'salary' változó normális eloszlást követ.

ks.test(salary, "pnorm", mean(salary), sd(salary))

# p < 0.05 => salary nem normális eloszlású
```

Egymástól **független minták eloszlásának összehasonlítása**	Kétmintás Kolmogorov-Szmirnov-próba	ks.test(var[cond1], var[cond2])

```
# Kétmintás Kolmogorov-Szmirnov-próba segítségével teszteljük le
# azt a nullhipotézist, hogy a férfiak és a nők körében azonos
# a fizetés eloszlása.

ks.test(salary[gender == "Male"], salary[gender == "Female"])
```

A megfigyelt vs. várt gyakoriságok	khi[2]-próba binomiális-próba	chisq.test(gyak, p = valsz) binom.test(egyik, ossz, valsz)

```
# Egy dobókockával százszor egymás után dobva a következő gyakoriságokat kaptuk:
# 1: 15, 2: 16, 3: 14, 4: 15, 5: 20, 6: 20
# Teszteljük azt a nullhipotézist, hogy a dobókocka szabályos.

valsz = rep(1/6, 6)
gyak = c(15, 16, 14, 15, 20, 20)
chisq.test(gyak, p = valsz)

# p = 0.8323 > 0.05 => nem tudjuk elutasítani a H0-t, vagyis
# nem tudjuk igazolni, hogy a dobókocka nem szabályos

# =================================
# Egy felmérés során a megkérdezett 127 ember közül 51 az aktuális polgármestert
# támogatja, 76 pedig a másik jelöltet.
# Mondhatjuk-e, hogy azonos a két jelölt támogatottsága?

valsz = c(0.5, 0.5)
gyak = c(51, 76)
chisq.test(gyak, p = valsz)

# p = 0.02653 < 0.05 => elutasítjuk H0-t, vagyis
# a két jelölt támogatottsága szignifikánsan eltér

# binomiális próba (egyik, össz, valszin)
binom.test(51, 127, p = 1/2)

# =================================
# Az 'ertekek20' nevű vektor 20 kockadobás eredményét tartalmazza.
# Teszteljük azt a nullhipotézist, hogy a dobókocka szabályos.
# Milyen figyelmeztető üzenetet kapunk, és ennek mi az oka?

# Ha az eredmények vektora adott, akkor abból előbb egy táblázatot kell készíteni
ertekek20 = c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,5,6,6,6,6)
gyak20 = table(ertekek20)

valsz = rep(1/6, 6)
chisq.test(gyak20, p = valsz)

# nem elég nagy az elemszám
# van olyan gyakoriság, ami < 5
# túl alacsony a mintavétel, nem tudunk pontos eredményt adni

# 2 elem esetén lehet binomiális próbát alkalmazni!
```