

# EGYMINTÁS T-PRÓBA PROGRAMOZHATÓ KIALAKÍTÁSA EXCEL VBA KÖRNYEZETBEN

Hampel György

**Abstract:** Az Excel számológépek egyszerűen használható lehetőséget biztosítanak számítások elvégzésére, s mindezt úgy, hogy adatváltozás esetén automatikus újraszámítás történik. Ezt kihasználva elegendő csak egyszer kialakítani a számológépet, amelyet így újból felhasználhatunk más bemenő adatok esetén. A Visual Basic for Application (VBA) szolgáltatással minimális programozói ismeret birtokában a bemenő adatok cseréjét programozottan is elvégezhetjük. Egymintás t-próba többszöri végrehajtását automatizálni tudjuk. Statisztikai számításokat biztosító programok használata során az újabb kiértékeléseket csak manuálisan tudjuk kezdeményezni. Nagy előnyt biztosít számunkra az Excel VBA ismétlődő kiértékelések során. Bemutatásra kerül a számítások automatizált elvégzéséhez szükséges felhasználóbarát kezelőfelület kialakítása az Excel táblázatkezelő program segítségével.

**Abstract:** Microsoft Excel spreadsheets provide the possibility to make calculations, and if data changes recalculations are done automatically. By taking advantage of this feature, a spreadsheet has to be created only once and it can be re-used with different input data. The replacement of the input data in a programmed way can be achieved with the use of Visual Basic for Application (VBA) with minimal programming knowledge. We can automate the multiple execution of the One-Sample T-test. With the use of statistical programs, the newer evaluations can only be initiated manually. Excel VBA gives a great advantage in repetitive evaluations. This paper describes the creation of a user-friendly interface in Microsoft Excel spreadsheet program for the automated execution of calculations required in a One-Sample T-test.

*Kulcsszavak:* Excel, VBA, statisztikai kiértékelés, egymintás t-próba

*Keywords:* Microsoft Excel, VBA, statistical evaluation, One-Sample T-test

## 1. Bevezetés

Statisztikai kiértékelések elvégzésére általában nem gondolunk arra, hogy egy táblázatkezelő programmal ezt hatékonyabban tehetnénk meg, mint a szokásos programokkal. Az Excel programozási lehetőséget is biztosít a Visual Basic for Application szolgáltatással. Kialakítva egy számológépet a szükséges statisztikai számítások elvégzésére oly módon, hogy az adatok megváltozása esetén az eredmények újraértékelődjenek, elkészítettük az alapját, hogy egy VBA programmal vezérelten automatizálni tudjuk a monoton ismétlődő kiértékeléseket. Természetesen statisztikai kalkulációk helyett pénzügyi (Zsótér, 2017), illetve egyéb komplex (Fabulya, 2017) feladatok automatizálását is hasonlóan biztosíthatjuk.

Az egyik leggyakrabban alkalmazott eszköz hipotéziseink ellenőrzésére az egymintás t-próba. Akkor alkalmazzuk, amikor egy numerikus adattípusú statisztikai sokaság (populáció) várhatóértékére vonatkozó állítás helyességét kell vizsgálnunk. Viszonylag egyszerű számítások végrehajtásával végezhetjük el a próbát, mely a sokaságból származó minta adatsor kiértékeléséből áll (Michaletzky–Mogyoródi, 1995).

## 2. Anyag és módszer

Az Excel 2010 verzióját választottuk munkánk során, hogy a Visual Basic for Application bővítményével programozási lehetőségünk legyen az egymintás t-próba automatizált többszöri végrehajtására. Azért is bizonyult ez jó döntésnek, mert sok esetben a kiértékelendő adatsorok is Excel fájlokban állnak rendelkezésünkre, ráadásul az elérhető munkalapfüggvényekkel könnyen kialakítható a számítások alapját biztosító munkalap (Wells–Harshberger, 1997). Mindemellett a legtöbb felhasználó rendelkezik a programmal, alapvető használatát ismeri, így költséghatékonyan alkalmazható.

### 2.1. Az Excel VBA alkalmazása

A táblázatkezelő programban automatikusan újraszámítódnak azok a cellák, melyekben más cellákra hivatkozó formulákat alakítottunk ki. Így kialakíthatjuk a kiértékelendő minta adatsorát és a számításokat is tartalmazó számolótáblát (Zimmerman, 1996). Mivel erre úgy van szükségünk, hogy az adatok változása esetén is megfelelő eredményt adjon, ezért nagyon körültekintőnek kell lennünk, minden lehetőséget figyelembe kell vennünk.

Amikor a számításokat automatizáltan, több mintán szeretnénk elvégezni, ahhoz a VBA programozási nyelv néhány utasítását használjuk:

- értékadó utasítás,
- ciklusszervező utasítás.

A program belső változói és a cellák közötti kétirányú adatmozgatást végezhetjük értékadó utasításokkal (Matteson, 1995). Ciklusszervezésre az ismételt tevékenységek miatt van szükségünk, egymás után több minta kiértékelésekor (Kovalcsik, 2005).

### 2.2. Az egymintás t-próba

Az egymintás t-próba segítségével egy statisztikai sokaság várhatóértékére vonatkozó hipotézis helyességét tesztelhetjük intervallum skálán mérhető numerikus adattípusú sokaságból származó minta alapján. Az alkalmazhatóságának feltétele, hogy normális eloszlásúnak legyen tekinthető a sokaság (Móri, 2011).

A próba végrehajtásakor először kiszámítjuk az alkalmazott statisztikai függvény (1) értékét ( $t$ ) a minta adatsorából.

$$t = \frac{\bar{x} - m_0}{\frac{s}{\sqrt{n}}} \quad s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (1)$$

ahol:

- $x_i$  – a minta  $i$ . eleme ( $i=1, \dots, n$ ),
- $s$  – a statisztikai sokaság becsült szórása,
- $\bar{x}$  – a mintaelemek átlaga, (számtani közép),
- $m_0$  – a hipotézisben vizsgált várhatóérték,
- $n$  – a minta elemszáma.

A döntést két technikával is meghozhatjuk:

- kritikus tartomány alapján,

- szignifikanciaszint ( $p$ ) kiszámításával.

Amikor a döntést a kritikus tartománnyal hozzuk meg, akkor a statisztikai függvény értékét hasonlítjuk össze a tartomány szélét mutató értékkel, mely a Student-féle t-eloszlás esetén a minta elemszámától ( $n$ ) és az elsőfajú hibavalószínűségtől ( $\varepsilon$ ) függő kritikus érték ( $t_{kr}$ ). Ha a kritikus tartományba esett a statisztikai függvény értéke, akkor szignifikáns, jelentős eltérést jelez a próba a hipotézis várhatóértékétől. Szintén ez adódik, ha a szignifikanciaszint kisebb az elsőfajú hibavalószínűségénél, azaz  $p < \varepsilon$  (Móri, 2011).

### 3. Eredmények és értékelésük

Célunk az Excelben egy olyan felület kialakítása, ahol a felhasználó csak a saját adatait kezelheti. Ezért úgy alakítjuk ki a szükséges objektumokat, hogy a számításokért felelős munkalap rejtett legyen, míg a feldolgozandó adatokhoz a saját Excel munkalapjain férjen hozzá. Így el kell készítenünk a számításokat végző munkalapot, valamint a kezelő felület munkalapját.

#### 3.1. Számítások munkalapja

A felhasználó számára rejtett a számítások munkalap, így nem szükséges az esztétikus, felhasználóbarát megjelenítésre nagy hangsúlyt fektetnünk. Az adatok a kezelő felület munkalapjáról úgy kerülnek át, hogy a kiértékelésben adatként értelmezhetetlen szöveges adatok kimaradjanak. Ez megoldható az Excel munkalap függvényeivel a (2) képlet szerint.

$$=HA(SZÁM(Adatok!A3);Adatok!A3;"")) \quad (2)$$

Tehát a  $HA()$  munkalapfüggvény ellenőrzi a  $SZÁM()$  függvénnyel, hogy az adatkezelő felület *Adatok* munkalapján az  $A3$  cella tartalma szám vagy sem, mert csak akkor kerül át a számítások munkalapra. E képlet másolatai biztosítják, hogy a rejtett munkalapra helyezzük az adatokat az  $A2$  cellától kezdődően. A számításokat ezeken az adatokon fogjuk elvégezni. Mivel ezek munkalapfüggvények, ezért minden adatváltozás automatikusan megjelenik itt is.

A számításokhoz szükséges bemenő adat még az elsőfajú hibavalószínűség ( $D3$  cella), valamint a vizsgálandó hipotézis típusa ( $D5$  cella) a várhatóértékkel ( $D4$  cella). Ezek ellenőrzés nélkül vehetők át az *Adatok* munkalapról, mert ott hibás értéket nem fogad el a program a felhasználótól. A számításokor a hipotézis típusa három féle lehet attól függően, hogy milyen irányú eltérést vizsgálunk a hipotézis várhatóértékétől:

- kétszélű ( $m \neq m_0$ ),
- balszélű ( $m < m_0$ ),
- jobbszélű ( $m > m_0$ ).

A felhasználó a vizsgálandó hipotézis megfogalmazásakor tetszőleges relációs jeleket alkalmazhat, de a számítás technikája a fenti típusú ellenhipotézisek (alternatív hipotézisek) egyikére vezet. Az *1. táblázatban* látható, hogy a lehetséges felhasználó által vizsgálandó hipotézis (H) típusok esetén a számításokor melyik alternatív hipotézist ( $H_A$ ) kell alkalmazni, és milyen döntést kell hozni H-ra, amikor

az alternatív hipotézist kell elfogadni, vagyis amikor a számított szignifikanciaszint kisebb az elsőfajú hibavalószínűségeknél ( $p < \varepsilon$ ).

**1. táblázat: Döntés az alkalmazott alternatív hipotézis elfogadásakor**

H	H <sub>A</sub>	Döntés H <sub>A</sub> elfogadásakor
$m = m_0$	$m \neq m_0$	H elutasítva
$m \neq m_0$	$m \neq m_0$	H elfogadva
$m < m_0$	$m < m_0$	H elfogadva
$m \leq m_0$	$m > m_0$	H elutasítva
$m > m_0$	$m > m_0$	H elfogadva
$m \geq m_0$	$m < m_0$	H elutasítva

Forrás: A szerző saját szerkesztése.

A számítások során egy-egy cellában a következő értékek alakulnak ki:

- a minta elemszáma ( $n$ ),
- a minta átlaga ( $\bar{x}$ ),
- a szórás becslt értéke ( $s$ ),
- a statisztikai függvény értéke ( $t$ ),
- a kritikus tartomány határa ( $t_{kr}$ ),
- a próba szignifikanciaszintje ( $p$ ).

Ezek mindegyike munkalapfüggvényekkel kapható meg:

- DARAB(),
- ÁTLAG(),
- SZÓRÁS(),
- T.INVERZ(),
- T.ELOSZL().

A kritikus tartomány határának, valamint a szignifikanciaszintnek a kiszámításakor azt is figyelembe kell venni, hogy melyik típusú ellenhipotézist alkalmazzuk.

A felhasználói felület felé a legfontosabb eredmény a szignifikanciaszint ( $p$ ), melyből könnyen adódik az eredmény a különböző típusú hipotézisek esetén.

### 3.2. Felhasználói felület kialakítása

Mivel csak Excel munkalap függvényeket alkalmaztunk a számítások munkalapon, ezért minden adatváltozásra a számítások aktualizálódnak automatikusan az *Adatok* munkalapon. Ez azt jelenti, hogy elegendő csak az *Adatok* munkalapot olyan felhasználói felületté alakítani, ahol az adatbevitel és az eredmény megjelenítése valósul meg. A kész felület látható az *1. ábrán*.

A próba eredményét három módon is megjelenítjük egy-egy bekeretezett részben:

- szövegesen,
- egy + vagy – jellel,
- a szignifikancia szinttel.

## 1. ábra: A próba felhasználói felülete

Egymintás t-próba		
Adatok	Elsőfajú hibavalószínűség ( $\epsilon$ )	0,05
22,1	Hipotézis várhatóértékre	m= 25
23,7	<b>A próba eredménye:</b>	
23,7	Az m=25 hipotézis elfogadható.	→ +
26	Szignifikancia szint:	0,13733
24,3		
26,4		
25,9		
24,4		
21,7		
23,3		

Forrás: A szerző saját szerkesztése.

A döntéskor az 1. táblázat szerint járunk el, tehát a  $H_A()$  függvénnyel a  $p < \epsilon$  feltételt kell csak ellenőrizni.

A felületen a világos háttérű cellák esetén van csak lehetősége a felhasználónak adatok módosítására. Megadható a minta adatsora, az elsőfajú hibavalószínűség, valamint a hipotézis várhatóértéke és típusa. Hogy a munkalap más celláit ne módosíthassa a felhasználó, azokat zárolttá kell állítani és bekapcsolni a munkalap védelmét. Viszont nem értelmezhető adatok ellen érvényesítési beállításokat alkalmazunk.

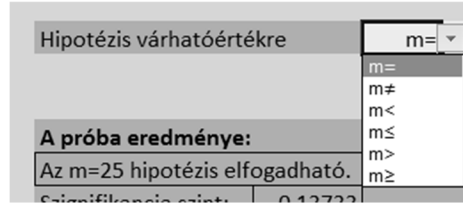
Az elsőfajú hibavalószínűség értéke leggyakrabban 0,01 vagy 0,05 szokott lenni. Itt a 2. ábrán látható érvényesítési beállítás miatt csak 0 és 0,2 közötti érték adható meg. Amikor ráállunk erre a cellára, megjelenik a tájékoztató figyelmeztetés, míg hibás adat esetén hibüzenet is látható.

## 2. ábra: Adatérvényesítés üzenetei

Forrás: A szerző saját szerkesztése.

A legördülő listás adatérvényesítés (3. ábra) biztosítja, hogy a hipotézis típusaként csak az értelmezhető relációk egyike legyen kiválasztható.

### 3. ábra: Legördülő listás adatérvényesítés



Forrás: A szerző saját szerkesztése.

#### 3.3. Programozott végrehajtás

Felhasználhatjuk arra is az Excel programot, hogy a VBA programozási lehetőségeit kihasználva az egymintás t-próbát több minta adatsorán automatikusan végrehajthassuk. Ehhez egy munkalapon (*Adatsorok*) egy-egy oszlopában szerepeltetve egy kiértékeléshez szükséges adatokat, az eredményeket is megkaphatjuk ugyanebben az oszlopban egy adatként, például + vagy – jelzés formájában. A programban ciklus szerkezetre van szükségünk. A ciklusban azt kell leprogramoznunk egy minta adatsor hipotézisének kiértékeléséhez, hogy az oszlop adatai kerüljenek át a felhasználói felület munkalapjára (*Adatok*), majd az ott adódó eredményt olvassuk ki, helyezzük a döntés eredményét az oszlop erre a célra fenntartott cellájába.

Az *Adatsorok* munkalap egy oszlopa a következő szerkezetű:

- az első cella az elsőfajú hibavalószínűség,
- a második cella a hipotézis típusa,
- a harmadik cella a hipotézis várhatóértéke,
- a negyedik cella az eredmény számára fenntartott cella,
- az ötödik cellától a minta adatsora következik.

A program a következő:

```
Public Sub kiertekeles()
    s1 = Sheets("Adatsorok")
    s2 = Sheets("Adatok")
    minta = 1
    Do Until s1.Cells(1, minta) = ""
        'Alapadatok feltöltése
        s2.Cells(2, 6) = s1.Cells(1, minta)
        s2.Cells(4, 6) = s1.Cells(2, minta)
        s2.Cells(4, 7) = s1.Cells(3, minta)
        'Törlés
        sor = 3
        Do Until s2.Cells(sor, 1) = ""
            s2.Cells(sor, 1) = ""
            sor = sor + 1
        Loop
        'Minta feltöltése
        sor = 3
```

```

Do Until s1.Cells(sor + 2, minta) = ""
    s2.Cells(sor, 1) = s1.Cells(sor + 2, minta)
    sor = sor + 1
Loop
'Eredmény kiolvasása
s1.Cells(4, minta) = s2.Cells(8, 7)
minta = minta + 1
End Sub

```

Az adatok áthelyezése az *s1* és *s2* változókkal elérhető *Adatsorok* és *Adatok* munkalapok cellái között történik. Az első oszloptól (*minta=1*) kezdődik a végrehajtás, mely akkor ér véget, mikor az aktuális oszlop üres cellával nem kezdődik (*do until s1.cells(1,minta)=""*). Egy cikluson belül először az alapadatokat helyezzük át, majd a minta adatsorát, de előbb törölni kell, mert bár egy újabb adatsor felülírja az előzőt, de ha az aktuális rövidebb, akkor a maradék adatok hamis eredményt okoznának. A ciklus végén kiolvassuk az eredményt és átállunk a következő minta oszlopára (*minta=minta+1*).

#### 4. Összegzés

Az egymintás t-próba számításainak végrehajtását automatizálni tudtuk. Kialakítottuk azt a felhasználóbarát felületet, melyen egyszerűen beállítható minden adat, s megjeleníti a próba eredményét. A újbóli felhasználást a Visual Basic for Application szolgáltatással programozott módon is megtehetjük. Ehhez rendelkezünk kell minimális szintű VBA programozói ismeretekkel.

#### Irodalomjegyzék

- Fabulya Z. (2017): Hőkezelési folyamatok összehangolása Excel VBA szolgáltatásokkal. *Jelenkori társadalmi és gazdasági folyamatok*, 12 (4): 19–25.
- Kovalcsik G. (2005): *Az Excel programozása*. Computerbooks, Budapest.
- Matteson B. L. (1995): *Microsoft Excel Visual Basic Programmer's Guide*. MicrosoftPress, Washington.
- Michaletzky Gy., Mogyóódi J. (1995): *Matematikai statisztika*. Nemzeti Tankönyvkiadó, Budapest.
- Móri T. (2011): *Statisztikai hipotézisvizsgálat*. Typotex Kft., Budapest.
- Wells E., Harshberger S. (1997): *Microsoft Excel 97 Developer's Handbook*. MicrosoftPress, Washington.
- Zimmerman M. W. (1996): *Microsoft Office 97 Visual Basic Programmer's Guide*. MicrosoftPress, Washington.
- Zsótér B. (2017): Financial planning in connection with accomodation development in a sport centre. *Quaestus multidisciplinary research journal*, 4 (11): 172–177.