

# EXCEL VBA PÁRBESZÉDABLAK FEJLESZTÉSE SHAPIRO-WILK PRÓBÁHOZ ROYSTON ALGORITMUSÁVAL

Fabulya Zoltán

**Abstract:** A matematikai statisztikában hipotézis kiértékelését végző próbák jelentős része csak normális eloszlású sokaságból származó mintán hajtható végre. A normalitás ellenőrzésére a legmegbízhatóbb Shapiro-Wilk próbát Royston kiterjesztése alapján kis és nagy elemszámú mintán is alkalmazhatjuk. Mivel kiértékelendő adataink táblázatkezelő programmal dolgozhatók fel kényelmesen, így nagy segítség a felhasználók számára, hogy a statisztikai számítások is elvégezhetők a táblázatkezelővel, nincs szükség statisztikai programokra és az adatok exportálására a további feldolgozásukhoz. Ezért volt szükséges a normalitás ellenőrzésével kiegészíteni a Microsoft Excel lehetőségeit. A számítások elvégzésének felhasználóbarát módja párbeszédablakon valósítható meg, melynek kialakítását a Visual Basic for Applications szolgáltatás teszi lehetővé. Eseményvezérelt programozási technika eredményezi a felhasználó tevékenységével harmóniában működő program elkészíthetőségét, a program kezelőfelületének megtervezését. Az elkészült program nem csak begépelésükkel támogatja az adatok megadását, kérhetjük a munkalap adatainak is a kiértékelését. Az eredményeket lehetőségünk van táblázatosan is megjeleníteni munkalapon.

**Abstract:** In mathematical statistics, a significant part of the hypothesis evaluation tests can only be performed on a sample from a normally distributed population. To check normality, the most reliable Shapiro-Wilk test based on Royston's extension can be applied to both small and large samples. Since our data to be evaluated can be conveniently processed with a spreadsheet program, it is a great help for users that statistical calculations can also be performed with the spreadsheet, there is no need for statistical programs and data export for further processing. That is why it was necessary to supplement the capabilities of Microsoft Excel with a normality check. A user-friendly way of performing the calculations can be realized in a dialog window, the design of which is made possible by the Visual Basic for Applications service. An event-driven programming technique enables the creation of a program that works in harmony with the user's activity, and the design of the program's user interface. The completed program does not only support data entry by typing them in, we can also request an evaluation of the worksheet data. We can also display the results in a table on a worksheet.

*Kulcsszavak:* Excel VBA, matematikai statisztika, normális eloszlás, Royston algoritmus

*Keywords:* Excel VBA, mathematical statistics, normal distribution, Royston algorithm

## 1. Bevezetés

A matematikai statisztikában a hipotézis vizsgálatok jelentős részét csak akkor alkalmazhatjuk, ha feltételezhető a mintáról, hogy normális eloszlású sokaságból származik. Emiatt a normalitás ellenőrzése a minta adatainak alapján sűrűn elvégzendő művelet. Több módszer közül a legalkalmasabb, kis elemszámú mintán is nagy megbízhatóságú a Shapiro-Wilk próba normális eloszlás tesztelésére (Thode, 2002). Viszont a felhasználhatóságát nehezíti, illetve korlátozza, hogy a számítások nehézkesen hajthatók végre táblázatból származó adatok miatt, melyek csak legfeljebb 50 elemszámú mintához érhetők el (Shapiro–Wilk, 1965). Ezeket a hátrányokat szünteti meg Royston kiegészítése a Shapiro-Wilk próbához, mellyel táblázatok adatai nélkül, akár 5000 elemszám esetén is elvégezhető a normális eloszlás ellenőrzése (Royston, 1995).

Fabulya (2022) kutatása bemutatta, hogyan valósítható meg Royston algoritmus Microsoft Excel környezetben úgy, hogy a munkafüzet szokásos használatával legyen megadható a kiértékelendő minta, s az eredmények a munkafüzetben jelenjenek meg formulákkal és munkalapfüggvényekkel számított módon. Jelen tanulmány azt a kutató munkát ismerteti, mely az Excel Visual Basic for Applications (VBA) szolgáltatásával kialakított párbeszédablakon biztosítja a Royston algoritmus végrehajtását, felhasználva a munkafüzetben kialakított számítási technikákat. A felhasználói felület kihasználja az eseményvezérelt programozási technika nyújtotta lehetőségekkel, hogy a használathoz alkalmazkodóan viselkedő, könnyen kezelhető legyen. Kiemelten fontos a gyakorlatban való alkalmazás során a könnyen kezelhetőség pl. piackutatási adatok feldolgozásánál (Zsótér–Kaliczka, 2014), de akár különféle modellek kialakításánál, továbbfejlesztésénél (Zsótér et al., 2020) is. Az adatok megadása begépeléssel is megvalósítható, de Excel munkalapot is kijelölhetünk adatforrásként. A normalitás vizsgálat eredményei szintén a párbeszédablakon jelennek meg, de szükség esetén, a munkalapon táblázatos formában is elérhetők.

## 2. Anyag és módszer

### 2.1. A Shapiro-Wilk próba és kiterjesztése Royston algoritmusával

A Shapiro-Wilk próba a legerősebb a sokaság normális eloszlását vizsgáló próbák körében. A minta adatait növekvő rendezettségben kell használni a próba statisztikai függvényének az (1) képlet szerinti kiszámításakor (Shapiro–Wilk, 1965).

$$W = \frac{\left(\sum_{i=1}^{n/2} a_i \cdot (x_{n+1-i} - x_i)\right)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (1)$$

ahol:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \quad (2)$$

$W$  – a Shapiro-Wilk próba statisztikai függvénye

$\bar{x}$  – a minta átlaga

$n$  – a minta elemszáma

$x_i$  – a minta elemei növekvő rendezettségben:  $x_i \leq x_j$  ( $i < j$ )

$a_i$  – együtthatók

Egyedüli hátránya a Shapiro-Wilk próbának, hogy az együtthatók, melyek értékét táblázat tartalmazza, legfeljebb 50 elemszámig adottak. Royston algoritmusával képletekkel számolhatók az együtthatók, s így 5000 elemszám a felső határa a kiszámíthatóságnak (Royston, 1982; Royston 1995). Az algoritmus képletei Fabulya (2022) cikkében megtalálhatók.

## 2.2. Az Excel VBA

A Microsoft Excel táblázatkezelő program Visual Basic for Applications (VBA) eszközével párbeszédablak alakítható ki programjaink kezelőfelületeként, számításaink elvégzéséhez. Eseményvezérelt programozással felhasználóbarát program készíthető, s így olyan programrészekből (eljárásokból, függvényekből) épül fel a teljes a program, melyek végrehajtását a felhasználó tevékenysége váltja ki. Leggyakrabban egy vezérlő értékének megváltozása, illetve egy parancsgombra klikkelés eseménye igényel programozást (Zimmerman, 1996).

Az alkalmazott vezérlők:

- Párbeszédablak.
- Több lap.
- Keret.
- Jelölőnégyzet.
- Beviteli mező.
- Lista.
- Kombinált lista.
- Címke.
- Léptetőnyíl.
- Parancsgomb.
- Kép.

Amikor a számítások a munkalapon történnek, de az adatokat és az eredményeket a párbeszédablak vezérlőivel szeretnénk kezelni, akkor elegendő csak olyan utasításokból álló eseményvezérelt eljárásokat készíteni, melyek a munkalap és a párbeszédablak közötti adatátvitelt biztosítják. Ezek értékadó utasítások, melyekkel a munkalap egy cellája vagy a párbeszédablak egy vezérlője kap értéket a kommunikáció irányától függően (Matteson, 1995). Az Excel és a VBA lehetőségei sok kutatásban megjelennek (Hampel, 2021), így a matematikai statisztikában használatos próbák kiértékelését segítő alkalmazások során is (Hampel, 2022).

## 3. Eredmények és értékelésük

### 3.1. A párbeszédablak megtervezése az adatbevitelhez

A készítendő program tervezésekor figyelembe kell venni a célokat, melyek a következők:

- A program minden funkciójának kezelése egyetlen párbeszédablakon megvalósítható legyen.
- A programot tartalmazó Excel fájl megnyitásakor jelenjen meg a párbeszédablak.
- A minta adatait begépeléssel, illetve a mintákat tartalmazó munkalap kiválasztásával is biztosítsa a felület alkalmas vezérlőkkel.
- Az adatbevitel és az eredmények megjelenítése külön lapon legyen. Az eredmények lapra lapozáskor automatikusan történjen meg az eredmények kiértékelése.

Az 1. ábrán látható a program kezelőfelülete az adatok beviteléhez kialakított lap vezérlőivel abban az esetben, amikor a minta adatai begépeléssel adhatók meg. Az *Adatok* és az *Eredmények* lapok közötti váltás klikkeléssel történhet. Jelölő négyzettel kérhető az *Adatsor begépelése* lehetőség, míg az elsőfajú hibaválósínúség megadását beviteli mező teszi lehetővé. A *Kilépés* parancsgombra klikkelve bezáródik a párbeszédablak.

1. ábra: Adatbevitel begépeléssel

Forrás: saját kutatás alapján a szerző szerkesztése.

Az adatsor begépelésekor a *Minta adatai* keretben lista vezérlő mutatja a már bevitt adatokat. Az *Új adat* címkénél beviteli mezőbe (TextBox1) gépelhetünk, s az Enter (KeyCode=13) leütésekor az adat a listába (ListBox1) kerül, egyúttal törlődik a beviteli mező tartalma, hogy máris gépelhessük a következő adatot. A VBA kód az alábbi:

```
Private Sub TextBox1_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    If KeyCode = 13 Then
```

```

ent = True
If IsNumeric(TextBox1.Text) Then
    ListBox1.AddItem TextBox1.Text
    TextBox1.Text = ""
    TextBox1.SetFocus
Else
    MsgBox "Nem szám", , "Hiba"
End If
End If
End Sub

```

Hibás adatot a listában kijelölése után egyetlen klikkeléssel törölhetünk a *Kijelölt adat törlése* parancsgommbal, melynek VBA kódja:

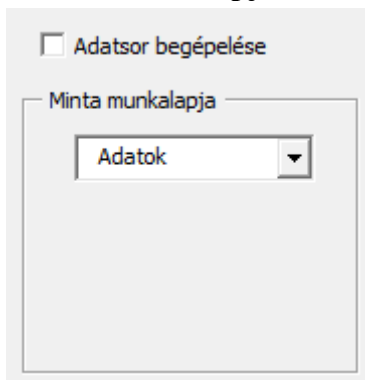
```

Private Sub CommandButton2_Click()
    If ListBox1.ListIndex = -1 Then
        MsgBox "Nincs kijelölt elem!", , "Hiba"
    Else
        ListBox1.RemoveItem (ListBox1.ListIndex)
        ListBox1.ListIndex = -1
    End If
End Sub

```

Amikor a minták adatai munkalapon tárolódnak, s emiatt nincs szükség adatsor begépelésére, akkor a 2. ábrán látható *Minta munkalapja* keretben kombinált lista vezérlővel választhatjuk ki a mintákat tároló munkalapot.

## 2. ábra: Minta munkalapjának megadása



Forrás: saját kutatás alapján a szerző szerkesztése.

Az *Adatsor begépelése* jelölőnégyzet adatának változásakor a program lecseréli a *Minta adatai* keretet (Frame1) a *Minta munkalapja* keretre (Frame2) a bennük lévő vezérlőkkel együtt:

```

If CheckBox1 Then
    Frame1.Visible = True
    Frame2.Visible = False
Else

```

```

Frame1.Visible = False
Frame2.Visible = True
End If

```

### 3.2. Az eredmények megjelenítése

Royston algoritmusát megvalósító számítások az Excel munkalapokon kialakított formulákkal történnek. Ezért a munkalapon az adat megváltozásakor automatikusan újraértékelődnek a formulákat tartalmazó cellák, s így keletkeznek az eredmények. Az eredmények megjelenítéséhez a párbeszédablakon csak annyi a teendőnk, hogy az adatokat a munkalagra helyezzük, majd az eredményeket a munkalapról a párbeszédablak vezérlőinek átadjuk. A szükséges VBA program végrehajtását az *Eredmények* lapra lapozás eseménye váltja ki. A 3. ábrán látható a párbeszédablak az eredmények megjelenítésekor.

3. ábra: Az eredmények megjelenítésének felülete

Shapiro-Wilk Royston próba

Adatok | Eredmények

$$W = \frac{\sum_{i=1}^n a_i \cdot (x_{n+1-i} - x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

Eredmények:

Statistikai függvény értéke (W)

Elsőfajú hibavalószínűség (ε)

Szignifikancia szint (p)

Normalitás

Forrás: saját kutatás adatai alapján a szerző szerkesztése.

Egy minta esetén, amikor az adatokat a párbeszédablakban a lista vezérlőnk tárolja, nincs szükség további vezérlőkre az eredmények kijelzéséhez, mert nem kell a minták egyikét kijelölni. Az alábbi VBA programrészlet helyezi az adatokat a lista vezérlőből (ListBox1) a munkalapra:

```
For i = 0 To ListBox1.ListCount - 1
    Sheets("Adatok").Cells(i+1, 33) = ListBox1.List(i)
Next i
```

A munkalapon az adatváltozás hatására kiszámítódnak az eredmények, melyeket a kiír VBA eljárás a vezérlők (TextBox) értékéhez rendeli az alábbi utasításokkal:

```
Public Sub kiir(m)
    TextBox8.Text = Sheets("Eredmények").Cells(16, m+1)
    TextBox9.Text = Sheets("Eredmények").Cells(2, 3)
    TextBox10.Text = Sheets("Eredmények").Cells(17, m+1)
    TextBox11.Text = Sheets("Eredmények").Cells(15, m+1)
End Sub
```

A fenti eljárásban az *m* argumentumra azért volt szükség, hogy több minta esetén a munkalap (Eredmények) megfelelő oszlopából származzon egy cella értéke.

Amikor több mintát kell kiértékelni, akkor a kijelzés annyiban módosul, hogy egy léptetőnyíl is szükséges a minta kiválasztásához. Ekkor a minta sorszámát is meg kell jeleníteni (4. ábra).

4. ábra: Eredmények több minta esetén

Forrás: saját kutatás adatai alapján a szerző szerkesztése.

A minta váltásakor a program lecseréli a kijelzett adatokat. Az alábbi VBA eljárást a léptetőnyíl (SpinButton1) eseménye aktiválja.

```
Private Sub SpinButton1_Change()
    Label28 = SpinButton1 & ". minta"
    Call kiiras
End Sub
```

A fenti kód biztosítja a minta sorszámának kijelzését a címke (Label28) vezérlővel, valamint a kiíras eljárás végrehajtásával a megfelelő eredményeket az alábbi VBA kóddal.

```
Public Sub kiiras()      'Eredmények lapról a Form-ra
    If CheckBox1 Then  'Van adatsor
        'Léptető elrejtése
        SpinButton1.Visible = False
        Label26.Visible = False
        Label27.Visible = False
        Label28.Visible = False
        'Kiírás a 34. oszlop adatai alapján
        Call kiir(33)
    Else
        'Léptető megjelenítése
        SpinButton1.Visible = True
        Label26.Visible = True
        Label27.Visible = True
        Label28.Visible = True
        Label28 = SpinButton1 & ". minta      "
        'Kiírás a léptető értéke alapján
        Call kiir(SpinButton1.Value)
    End If
End Sub
```

A fenti, kiíras eljárás ellenőrzi, hogy szükséges-e a léptetőnyíl megjelenítése több minta miatt, és ennek megfelelően a vezérlők láthatóságát (Visible) beállítja mielőtt a kiválasztott mintához tartozó eredményeket produkálná a kiir eljárás meghívásával.

#### 4. Következtetések

A Shapiro-Wilk próba Royston algoritmusával képletekkel teszi kiértékelhetővé a normalitás ellenőrzését a minta adatai alapján. A szükséges számítások automatikus végrehajtását a Microsoft Excel táblázatkezelő program munkalapjain formulák végzik. A Visual Basic for Applications teszi lehetővé, hogy felhasználóbarát kezelőfelületet kialakítva, eseményvezérelt programozási technikával egy párbeszédablakon a minták adatait rögzíthessük, és az eredményeket megjelenítsük. Így matematikai statisztikai programcsomag nélkül nagy elemszámú minták esetén is elvégezhető a próba.

#### Irodalomjegyzék

- Fabulya Z. (2022): Számítás automatizálása Shapiro-Wilk próbához Royston algoritmus szerint Excel számológéppel. *Jelenkori Társadalmi és Gazdasági Folyamatok*, 17 (3-4): 67–74. <https://doi.org/10.14232/jtgf.2022.3-4.67-74>
- Hampel Gy. (2021): Információfeldolgozás és döntéstámogatás gazdák számára Microsoft Excellel. In: Hampel Gy.; Kis K.; Monostori T. (szerk.): *Mezőgazdasági és vidékfejlesztési kutatások a*



- jövő szolgálatában 2.: Tudomány: iránytű az élhető jövőhöz.* Magyar Tudományos Akadémia Szegedi Akadémiai Bizottság Mezőgazdasági Szakbizottság, Szeged. 51–60.
- Hampel Gy. (2022): Alkalmazás fejlesztése egymintás u-próba kiértékelésére VBA vezérlőkkel Excel környezetben. *Jelenkori Társadalmi és Gazdasági Folyamatok*, 17 (1-2): 119–126. <https://doi.org/10.14232/jtgf.2022.1-2.119-126>
- Matteson B. L. (1995): *Microsoft Excel Visual Basic Programmer's Guide*. MicrosoftPress, Washington.
- Royston, J. P. (1982): Algorithm AS 181: The W Test for Normality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31 (2): 176–180. <https://doi.org/10.2307/2347986>
- Royston, P. (1995): Remark AS R94: A Remark on Algorithm AS 181: The W-test for Normality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44 (4): 547–551. <https://doi.org/10.2307/2986146>
- Shapiro, S. S., Wilk, M. B. (1965): An Analysis of Variance Test for Normality (Complete Samples). *Biometrika*, 52 (3/4): 591–611. <https://doi.org/10.2307/2333709>
- Thode, H. C. (2002): *Testing For Normality (1st ed.)*. CRC Press. <https://doi.org/10.1201/9780203910894>
- Zimmerman M. W. (1996): *Microsoft Office 97 Visual Basic Programmer's Guide*. MicrosoftPress, Washington.
- Zsótér B., Illés S., Simonyi P. (2020): Model of Local Economic Development in Hungarian Countryside. *European Countryside*, 12 (1): 85–98. <https://doi.org/10.2478/euco-2020-0005>
- Zsótér B., Kaliczka R. (2014): Examinations carried out in relation to the shopping habits and satisfaction of costumers in the shops of Coop Szeged Ltd. *Review of Faculty of Engineering Analecta Technica Szegedinensia*, 8 (1): 38–41.