

# **SZAKDOLGOZAT**

**Labátné Fercsik Katalin**

**2024**



**Magyar Agrár- és Élettudományi Egyetem**

**Szent István Campus**

**Műszaki és Informatikai Intézet**

**Adattechnológus-adatelemző szakember szakirányú  
továbbképzési szak**

**ONLINE ELÉRHETŐ ADATOK ELEMZÉSE**

**Belső konzulens:** Lágymányosi Attila Tibor  
egyetemi adjunktus

**Belső konzulens  
intézete/tanszéke:** **Mérnökinformatika**

**Külső konzulens:** Moór-Gergely Gabriella  
kontrolling referens

**Készítette:** **Labátné Fercsik Katalin  
(S9CF04)**

**Gödöllő**

**2024**

# Tartalomjegyzék

<b>1. Bevezetés és célkitűzések</b> .....	<b>1</b>
<b>2. Összehasonlított adatvizualizációs eszközök</b> .....	<b>3</b>
2.1. Microsoft Excel, mint adatkezelő eszköz.....	4
2.1.1. Adatok beemelése az MS Excelbe .....	5
2.1.2. Power Query, az adattisztítás eszköze .....	7
2.1.3. Power Pivot, mint a Power BI elősze .....	11
2.1.4. DAX .....	12
2.2. Power BI a legelterjedtebb.....	13
2.3. Python – Pandas, Matplotlib, Bokeh, Seaborn könyvtárak.....	19
<b>3. Online térből gyűjtött adatok elemzése</b> .....	<b>25</b>
3.1. Adatfeldolgozás és vizualizáció MS Excelben.....	25
3.1.1. Adatok beimportálása és adattisztítás Power Query-ben.....	25
3.1.2. Adatkapcsolatok kiépítése.....	27
3.1.3. Adatvizualizáció készítés Pivot Table alkalmazásával .....	28
3.2. Vizualizáció készítése Power BI felületen.....	30
3.2.1. Vizualizációs háttér kialakítása Canvában .....	31
3.2.2. Elemzés összeállítása, vizualizációs eszközök kiválasztása.....	33
3.3. Adatelemzés Python segítségével .....	35
3.3.1. Google Colab felület bemutatása .....	35
3.3.2. Könyvtárak és adatok importálása .....	36
3.3.3. DataFrame létrehozása, adattípusok lekérdezése .....	37
3.3.4. Adatok rendezése, vizualizáció elkészítése több könyvtár használatával.....	38
3.3.5. Az adatok összefüggésének vizsgálata statisztikai módszerekkel .....	42
<b>4. Összefoglalás</b> .....	<b>45</b>
<b>5. Felhasznált szakirodalom és források</b> .....	<b>46</b>
<b>Köszönetnyilvánítás</b> .....	<b>48</b>
<b>Ábrajegyzék</b> .....	<b>49</b>

## 1. Bevezetés és célkitűzések

Az információrobbanás korában, ahol a digitális világ fejlődésével a technológia egyszerűvé és olcsóvá tette az adatok gyűjtését, a modern vállalatok rengeteg adattal találkoznak. Napjainkban rendkívüli mennyiségű különféle adat áll rendelkezésre – általában feldolgozatlan formában. Azonban egyre több szakterület kezdi felismerni az adatok megfelelő feldolgozásának az előnyeit.

Egyre több területen vált elengedhetlenné az adatvezérelt döntéshozatal, azonban ezzel együtt jár a nagy mennyiségű, változatos adatok feldolgozásának és értelmezésének kihívása is. Az adatok gyors és folyamatos generálódása olyan dinamikus környezetet teremtett, ahol a vezetőknek is gyorsan kell döntéseket hozniuk. Az adatelemző szakembereknek pedig minél pontosabb információkat kell biztosítani a döntéshozatalhoz, számolva az idő rövidegével.

Ebben a dinamikus környezetben szükségessé vált olyan eszközök bevezetése, amelyek segítségével a nagy adatmennyiség könnyen áttekinthetővé válik, és a vezetőknek lehetőségük nyílik pontos értékítéletet alkotni a rendelkezésre álló információk alapján. Már nem elfogadható, hogy több száz jelentést kelljen ahhoz átnézni, hogy a döntéshozók átfogó képet kapjanak a vállalkozás minden folyamatáról. Egyre inkább elvárás egy olyan összetett információs halmaz, amely a vizualizációs eszközök segítségével betekintést enged az összesített adatok mellett azok részleteibe is.

Az adatokat számos módon fel lehet dolgozni, de a leghatékonyabb módszer a vizuális ábrázolás. A vizualizáció ereje abban rejlik, hogy az emberek gyorsan képesek nagy mennyiségű képi információt befogadni és mintázatokat felismerni. A legkülönbözőbb információkat egy könnyen érthető, szemléletes formában jeleníti meg. A vizuális látvány gyorsabban feldolgozható és emészthető, mint a hosszú táblázatok vagy szöveges adatok. Ezen kívül a különböző diagramok, grafikonok és egyéb vizualizációk segítik a vezetőket abban, hogy az adatok mögött rejlő összefüggéseket és trendeket könnyebben észrevegyék.

Láthatóvá válik, hogy az elemző szakemberek számára nem elegendő pusztán a szakterületük ismerete vagy egy egyszerű adattábla készítése. Az adatokat nem csak átlátni, hanem értelmezni és hatékonyan kommunikálni is szükséges a szakmai döntéshozatalhoz. Ebből kifolyólag a beszédesebb adatvizualizációk létrehozása elengedhetetlen.

Az informatikai tudás, különösen az adatkezelés terén, létfontosságúvá vált a gazdasági elemzők számára. Az adatfeldolgozási idő rövidebbé azonban megkövetelte olyan eszközök létrehozását, amelyek lehetőséget adnak arra, hogy az adatokat több szempontból is elemezzék, és jövőbeni trendeket, előrejelzéseket készítsenek. Ezáltal a szakértők nem csupán a múltat tudják értelmezni, hanem a jövőre vonatkozóan stratégiai döntések meghozatalát is támogathatják.

A szakdolgozatom célja az, hogy összehasonlítsak több Microsoft termék és a Python kódra épülő könyvtárak által nyújtotta lehetőségeket a vizualizáció terén. A vizsgálat során bemutatom az eszközök által biztosított funkciókat, elemzem felhasználhatóságukat és hatékonyságukat. Az adatokat az internetről több oldalról közvetlenül hivatkozom be, vagy onnan letöltött adatbázist használok. Céлом, hogy megállapítsam, melyik eszköz igényel kevesebb előképzettséget, melyik könnyebben elsajátítható, és melyik biztosítja a legnagyobb felhasználói élményt és legjobb eredményeket az adatelemzés terén.

## 2. Összehasonlított adatvizualizációs eszközök

Az adatvizualizáció az információk és adatok grafikus ábrázolása. Célja, hogy az egyébként nehezen felfogható számok és adatok soraiból és oszlopaiból értékeket kinyerjen, és azokat vizuálisan tetszetős grafikonokon ábrázolja. Ennek eredményeként az adatvizualizáció egy pillantással kulcsfontosságú betekintést nyújthat az adatokkal kapcsolatban, amire a nyers adatok, sőt a táblázatos formában elemzett adatok sem képesek.

Az adattudósok folyamatosan tökéletesítik az adatok vizualizálásának módszereit. Az alkalmazott vizualizációs eszközök és alkalmazások kiválasztását az adattartomány jellege határozza meg. Az adatok elemzésére szolgáló vizualizáció hatékonysága attól függ, mennyire képes hatékonyan közvetíteni az információkat. Ezért kiemelten fontos a megfelelő eszköz kiválasztása az adatok prezentálásához. Aranyszabályként fogalmazható meg, hogy meg kell találni az egyensúlyt a pontos adatok és a design között. (Balázs, és mtsai. 2013) Előfordulhat, hogy egy vizualizáció túlságosan komplex vagy különleges ahhoz, hogy az olvasó könnyen értelmezze, míg más esetekben egy túl egyszerű megközelítés nem éri el a várt eredményeket. (Guntuku és Shubhangi 2020) Ezek elkerülése érdekében fontos a gyakorlati tapasztalatok szerzése a különböző vizualizációs technikák értelmezésében, valamint azok előnyeinek és korlátainak megértésében.

A jó ábrák készítéséhez alaposan meg kell ismerni az adatokat, és érteni a mondanivaló lényegét. Nem elég csak egy számsort csoportosítani és grafikonba rendezni. Ismerni kell az adatok jelentését, a célokat az ábrával kapcsolatban, valamint a mögöttes folyamatokat és viszonyokat. Enélkül az ábra nem lesz hatékony.

Ha a cél már meghatározásra került, és a megjelenítendő információ is körvonalazódott, akkor már jó esély van megtalálni legmegfelelőbb ábrázolási módot. Érdeemes kísérletezni, hiszen az elsőre jónak tűnő megoldás nem feltétlenül a legjobb. Ugyanakkor az öncélú grafikák, amelyeknek nincs kapcsolata az adatokkal, feleslegesek, mert nem szolgálják az adatvizualizáció alapvető célját.

Adatvizualizáció készítésekor fontos, hogy az ábra minden jelentős információt közöljön a befogadóval. Meg kell adni a méretarányt vagy a megfelelő skálát. A színeknek is lényeges szerepük van, és figyelembe kell venni, hogy a színeknek különböző jelentése lehet kultúrák és szubkultúrák szerint. Az ábra legyen egyértelmű, és ne zavarja össze sem a szakértőt, sem a laikus olvasót, az üzenetet pedig hatékonyan közvetítse.

A vizualizáció legfontosabb szabályai (Po, és mtsai. 2020):

- Fontos ismerni a megcélozni kívánt közönséget. A vizuális elem tervezése a szándéktól kell, hogy vezérelt legyen.
- Legyen pontosan megfogalmazva a vizualizáció átadni kívánt üzenete.
- Alkalmazkodás a közvetítés eszközéhez.
- Nem szabad megfeledkezni a feliratokról.
- Az alapértelmezett beállításoktól célszerű eltérni.
- A színek adta lehetőségeket érdemes kihasználni.
- Nem célszerű eltérni a háttérben levő adatbázistól, mert félrevezető lehet az átadni kívánt üzenet.

## 2.1. Microsoft Excel, mint adatkezelő eszköz

A **MS Excel** az adatkezelés egyik legelterjedtebb eszköze, főként azért, mert jól strukturált eszközkészlete van. Mindezekon túl könnyedén importálhatunk vele adatokat különböző forrásokból. A jelentések létrehozása könnyű, szerkesztése és megosztása miatt széles körben elterjedt minden üzleti folyamatban. Funkciói lehetővé teszik a jelentések egyszerű létrehozását, preferált választás olyan szervezetekben, ahol a vállalati kultúrában nem kívánnak bonyolult informatikai eszközöket használni, és a felhasználók befogadóképessége szélsőséges.

A beépített funkciói segítségével könnyedén dolgozhat a felhasználó az adatokkal, készíthet grafikonokat vagy akár összehasonlító elemzéseket. Kiválóan kezeli az aránylag nagyobb adatmennyiségeket is, így nagy projektek vagy komplex adatelemzések esetén is hatékonyan használható. A beépített függvényei biztosítják az egyre jobb felhasználói

élményt. Összességében sokoldalú és hatékony eszköz az adatelemzéshez és adatkezeléshez, ami a felhasználók széles körének segítségére van a mindennapi munkában.

A program rugalmassága lévén lehetőségünk van egyszerűen adat bemásolásra weboldalakról. Nemcsak adatokat, hanem akár képeket is könnyedén átemelhetünk. Szinte minden adattípust és formátumot képes kezelni, ezért rendkívül sokoldalú eszköz mind kezdők, mind haladó felhasználók számára.

### 2.1.1. Adatok beemelése az MS Excelbe

Az adatok importálása történhet egyszerű másolással és beillesztéssel, de ezzel pont a frissítés funkcióját veszíti el az adathalmazunk. Ezért, hogy az **MS Excel** aktuális adatokat tartalmazzon, mindenképp érdemes az adatokat importálással beemelni.

Egy pénzügyi elemzés esetén egy vállalat elsődleges adatforrása a vállalat által vezetett nyilvántartás. A legtöbb nagy vállalatnál erre kialakított ERP<sup>1</sup> rendszerekben történik az adatok nyilvántartása. A legtökéletesebb adatelemzésre akkor kerülhet sor, ha az adatok közvetlenül az ERP rendszer adatbázisából kinyerhetőek. Az exportált adatokhoz, vagy nem elsődleges adatforráshoz való csatlakozás ugyanis az adatok torzulását eredményezheti, elveszítve így az adatintegritást, valamint a valós idejű adatszolgáltatást. (Olafusi 2024)

Az adatok importálása a Microsoft eszközök terén a következő lehetőségeket biztosítja:

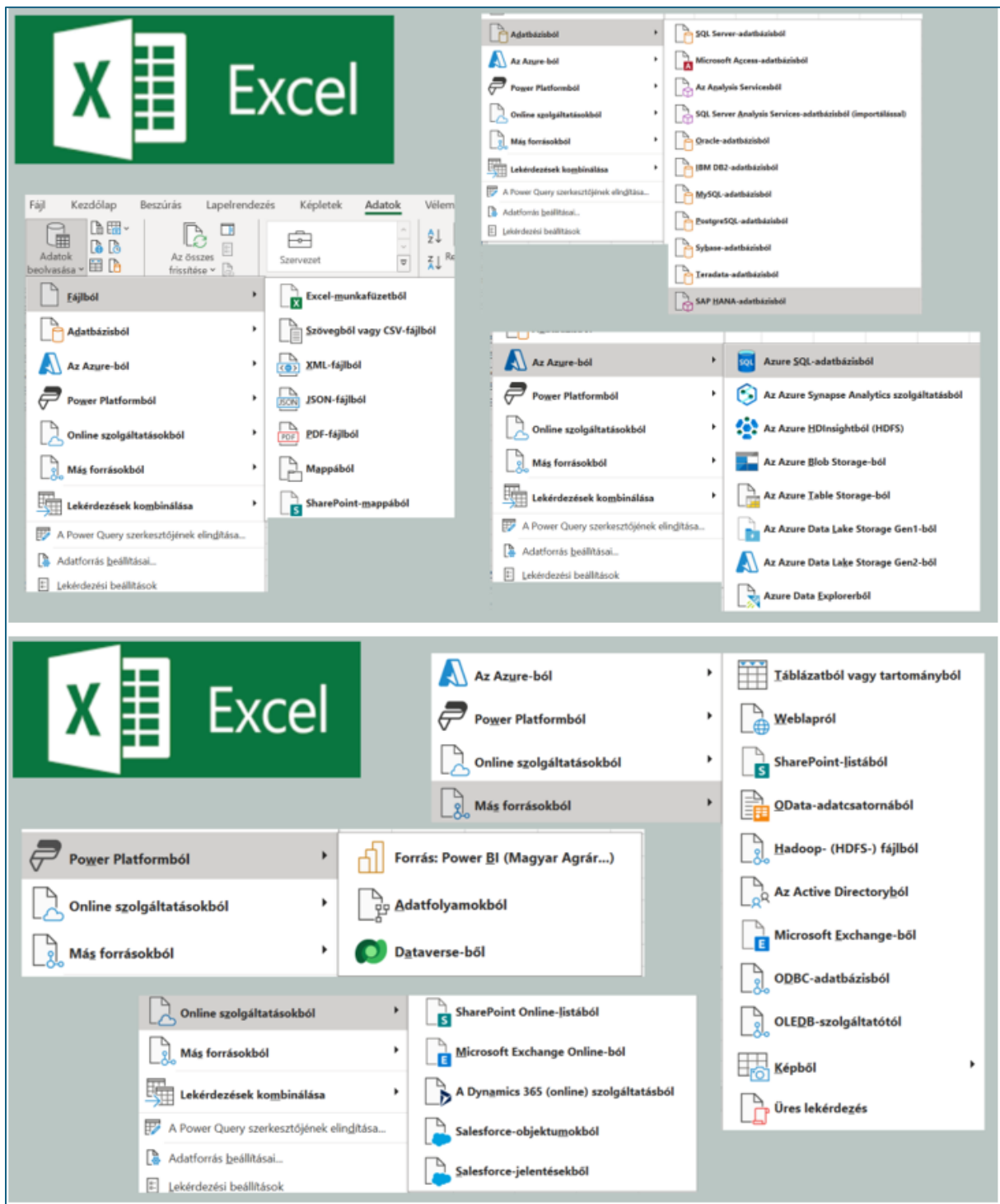
- fájlból,
- adatbázisból,
- Azure-ból,
- Power Platformból,
- online szolgáltatásokból,
- más forrásokból.

---

<sup>1</sup> ERP: Enterprise Resource Planning, azaz vállalatirányítási rendszer.



1. ábra: MS Excel adatbeviteli lehetőségek



Természetesen a sok lehetőség közül a legjobb adatforrást a jól megtervezett és felépített adatbázisokból nyert adatok jelentik. Különösen a relációs adatbázisok<sup>2</sup> a legmegbízhatóbb adatforrások, mivel strukturális elrendezésükben előre meghatározott adatvalidációs és

<sup>2</sup> Relációs adatbázis: olyan adatbázis, amelyben az adatok egymással kapcsolatban álló táblákban vannak tárolva.

adatfeldolgozási logikát alkalmaznak, ezáltal gátolva az ellentmondásos adatok használatát. A szabályozott adatstruktúra és a szigorú logikai rendszerek hatékonyan védenek az adatinkonzisztenciák ellen, javítva ezzel az adatok megbízhatóságát és konzisztenciáját. (Olafusi 2024)

Előfordulhat azonban, hogy az **MS Excel** nem tud összeköttetést létesíteni az elsődleges adatforrással, mert nem rendelkezik a csatlakozást biztosító szoftverrel, vagy biztonsági megfontolásból a szervezet nem engedélyezi az elsődleges adatforrás elérését. Illetve az is előfordulhat egy kisebb vállalat esetében, hogy az elsődleges adatforrások nem léteznek egyetlen vállalati adatbázisban. Ezek esetében érdemes a felhő alapú tárolás megoldása, így közvetlenül az **MS Excelben** biztosított a csatlakozás a fájlokhoz. (Olafusi 2024)

### **2.1.2. Power Query, az adattisztítás eszköze**

Gyakran vannak be adatelemzésre olyan adatokat, amelyeket más célra gyűjtöttek össze. Emiatt gyakran az adatok minősége nem teszi azonnal lehetővé az algoritmusok, elemzések felépítését. Az adatfeldolgozás nagy részét teszi ki ezért az adatok minőségének a javítása, azaz az adattisztítás. Ez a folyamat az, amely során az adatokat standardizálják és strukturálják, hogy megfeleljenek az adott felhasználási cél követelményeinek. Ez az eljárás fontos szerepet játszik az adatelemzésben és az információk kiértékelésében, mivel a tiszta és pontos adatok lehetővé teszik a megbízható következtetések levonását. Persze irreális elvárás az adatok tökéletessége, de cél a lehető legpontosabb eredmény elérése.

Az adattisztítás és formázás kulcsfontosságú lépés az adatelemzési folyamatban, mivel az adatok gyakran nem tökéletesek, és hibákat tartalmazhatnak. Emberi hibák, műszaki korlátok vagy az adatgyűjtési folyamatok hibái miatt előfordulhatnak problémák az adatokban. Lehetnek hiányzó értékek, hibás vagy duplikált adatok, valamint ellentmondások az adatokban.

Az adattisztítás során az adatokat ellenőrzik, hogy azok megfeleljenek az előre meghatározott szabályoknak és normáknak, és az esetleges hibákat vagy hiányosságokat

javítják. A formázás lényegében a megjelenés és szerkezet kialakítását jelenti az adatok számára, hogy könnyen értelmezhető és összehasonlítható legyenek. Ezen felül az adattisztítás és formázás segíti az adatok egységesítését a különböző forrásokból származó adatok esetén, ezáltal javítva azok egy nevezőre való hozását és megbízhatóságát. Az adattisztítás és formázás komplex folyamat, amely különféle technikákat és eszközöket igényel, és alapvető fontosságú a minőségi adatelemzés és döntéshozatal elősegítésében.

Az igazi **MS Excel** szakértők több eszközt is használnak a jó adathalmaz előállításához. Képletekkel vagy makrók<sup>3</sup> segítségével, a VBA<sup>4</sup> programozási nyelven rendezett utasítássorozattal képesek automatizálni a folyamatokat különböző feladatok végrehajtásához. Évekig gyakorlatilag ezek voltak az egyetlen eszközök, amelyek az adatok tisztításához és átalakításához segítséget nyújtottak. Bár hasznosságuk máig nem vitatott, két súlyos problémával is szembe kell nézni a használatukkor. A megoldások kialakítása és a technikák megtanulása is hosszú időt vesz igénybe. A haladó nyelvek elsajátítására évekre van szükség, majd további jelentős időre a megoldások kifejlesztéséhez, teszteléséhez és karbantartásához. A megoldások bonyolultságától függően a változásokra való rugalmas alkalmazkodás vagy egy másik forrás integrálása kihívást jelent. (Puls és Escobar 2021)

A Power Query az elmúlt évek egyik legnagyobb frissítése az MS Excelnek, amely a korábbi problémákat megoldja. Könnyen tanulható, egyszerű felhasználói felülettel rendelkezik, és lehetővé teszi a teljes automatizálást Dashboardok<sup>5</sup> segítségével, összetett képletek írása nélkül. A folyamat minden lépése könnyen karbantartható, és néhány kattintással frissíthető. Sok szempontból forradalmi változást hozott, például gyorsabb tanulási idővel és a bonyolult források könnyebb kezelésével a VBA-hoz képest. (Olafusi 2024)

---

<sup>3</sup> Makró: előre rögzített utasítássorozat, amely automatikusan futtatható MS Excel környezetben.

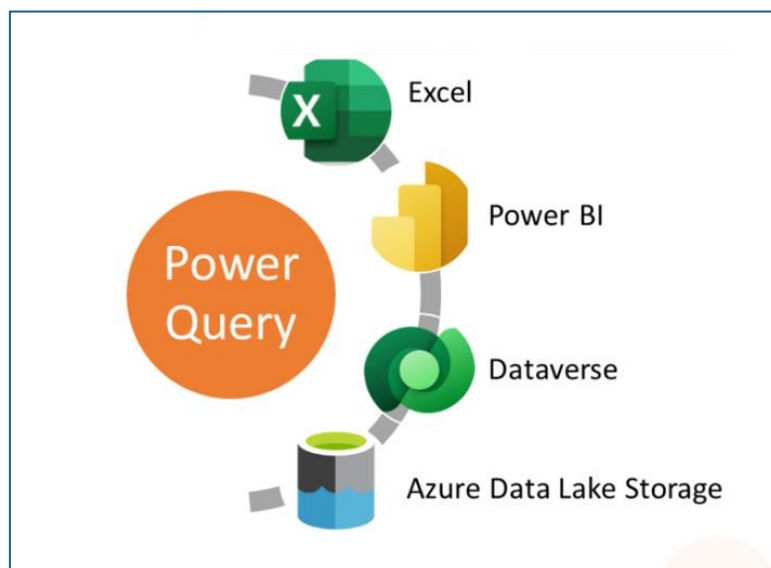
<sup>4</sup> VBA: Visual Basic for Applications

<sup>5</sup> Dashboard: irányítópult vagy műszerfal, egy olyan felhasználói felület, amely grafikusán, adatvizualizációkkal mutatja be egy szervezet legfontosabb mutatóit, egy oldalon.

Bár egy SQL<sup>6</sup>-szakember által írt lekérdezés általában hatékonyabb lehet, a Power Query egyszerű adatfolyamatokat és adattisztítást tesz lehetővé alacsonyabb idő- és erőforrás-befektetéssel. Emellett lehetővé teszi az adattranszformációs folyamatok kibővítését, hiszen ugyanazt a technológiát alkalmazza az **MS Excel**, a **Power BI Desktop**, a **Power Automate** és a **Power BI Dataflow** területén. (Puls és Escobar 2021)

Ennek eredményeként egy a **Power Query** segítségével kialakított megoldást könnyedén importálhatunk a **Power BI Desktopba** vagy másolhatunk a **Power BI Dataflowba**. Lényegében az **Power Query** egy ETL<sup>7</sup> eszköz. A Microsoft nem vitatott célja, hogy a **Power Query** legyen az elsődleges ETL motor minden adatelemző eszközében a nem informatikai aspektussal rendelkező felhasználók számára. (Olafusi 2024)

*2. ábra: Jelenlegi Microsoft alkalmazások, amelyek rendelkeznek Power Query-vel (Forrás: Building Interactive Dashboards in Microsoft 365 Excel, Szerző: Michael Olafusi)*



A **Power Query** használata jelentősen megkönnyíti az elemző szakemberek munkáját, ráadásul a felülete az **MS Excel** felületéhez hasonlóan van összeállítva. Egyszerre több különböző adatforráshoz képes csatlakozni anélkül, hogy az adatokat exportálni kellene, majd importálni az **MS Excel** felületre. Korábban minden frissített adatfájl újabb manuális adattisztítási folyamatokat kreált, azonban mindez már a múlté.

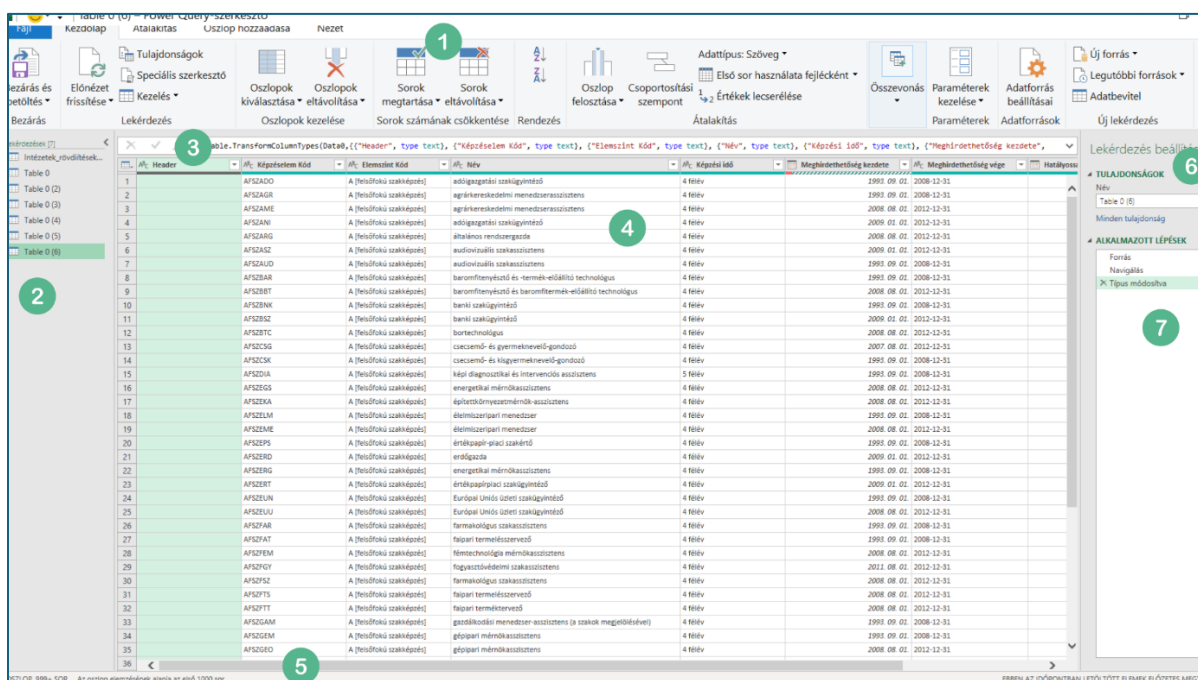
<sup>6</sup> SQL: Structured Query Language – egy programozási nyelv, amelyet relációs adatbázis-kezelő rendszerekben használnak az adatok lekérdezésére, módosítására és kezelésére.

<sup>7</sup> ETL: Extract, Transform and Load - egy adatintegrációs folyamat, amelynek célja az adatok gyűjtése, átalakítása és betöltése egy adattárházba vagy adatfolyamba.

Az adatok beemelése után **Power Query Editor**-ban van lehetőség az adatok feldolgozására, ami a következő részekből épül fel (Puls és Escobar 2021):

1. menüszalag,
2. navigációs panel,
3. képletsáv,
4. előnézeti ablak,
5. állapotsáv,
6. tulajdonságok,
7. végrehajtott lépések ablaka.

**3. ábra:** Power Query szerkesztő (Forrás: Ken Puls\_Miguel Escobar - Master Your Data with Power Query in Excel and Power BI alapján saját feldolgozásban)



Néhány kulcsfontosságú információ, amit érdemes megjegyezni erről a lenyűgöző technológiáról (Puls és Escobar 2021):

- Kapcsolódhat számos adatforráshoz.
- Rögzít minden lépést, létrehozva egy **Power Query** „scriptet”.
- Soha nem változtatja meg a forrás adatokat, lehetővé téve különböző parancsok alkalmazását és törlését.
- Később frissíthető, amikor az adatok változnak.

A **Power Query-ben** történő lekérdezések létrehozása lehet egységes vagy szétválasztott. Az **egységes lekérdezés** követi azt a minimalizálási koncepciót, mely más programozási nyelvekben is alkalmazandó, ahol a kódoptimalizálás az elsődleges cél, a felesleges részek kihagyásával. Számos program, például az SSIS<sup>8</sup> és az Azure Data Factory<sup>9</sup>, csak egyetlen lekérdezést támogat. Mindazonáltal van lehetőség a **lekérdezések szétválasztására** is, ami megkönnyíti az adatok kiválasztását. Ezáltal egyszerűbb lehet a változások követése, az új adatforrások beállítása és az adatokban történt módosítások nyomon követése, karbantartása. (Puls és Escobar 2021)

A **Power Query-ben** használt programozási nyelv az **M**, ami egy funkcionális nyelv. Mint minden programozási nyelv, saját szerkezettel és szintaxissal rendelkezik. Minden átalakítás egy M kódot hoz létre. (Olafusi 2024)

A **Power Query-ben** két értéktípust különböztetünk meg (Olafusi 2024):

- **Primitív** értékek azok, amelyek az alapvető értékeket kategorizálja (bináris, dátum, dátumidő, időzóna, időtartam, logikai, null, szám, szöveg, idő, típus).
- **Strukturált** értékek, mint például a táblák, a listák, a rekordok és a funkciók, amelyek primitív értékekből épülnek fel.

### 2.1.3. Power Pivot, mint a Power BI előszele

A kimutatásdiagram rendkívül fontos az adatelemzők munkájában, ami arra készítette a Microsoftot, hogy létrehozza a **Power Pivot-ot** a **Pivot Table** fejlettebb változataként.

A **Pivot Table** összefoglaló jelentések készítését teszi lehetővé, tulajdonképpen az adatok aggregált megjelenítésére szolgáló eszköz, amely már lehetővé teszi az adatok diagramokban történő ábrázolását is. Az **MS Excel** eszköztára rengeteg lehetőséget biztosít a megjelenítés módjára, adatok szűrésére, dinamikusan változó jelentések létrehozására. Az

---

<sup>8</sup> SSIS: SQL Server Integration Services, egy adatintegrációs és adatelőkészítési platform, amely az MS SQL Server része.

<sup>9</sup> Azure Data Factory: felhő alapú adatintegrációs szolgáltatás a Microsoft Azure platformon. Rugalmas adatintegrációt tesz lehetővé felhő környezetben.

elkészült jelentéseket és diagramokat lehetőségünk van összekapcsolni, amelyek segítségével egy szűrés hatására az összes kimutatás is leképezi az adatváltozást. (Olafusi 2024)

A **Power Pivot** már túlmutat a **Pivot Table** biztosította lehetőségeken. Képes több mint 1.048.576 sorból álló adathalmaz elemzésére, sőt több tábla adatait képes egyben kezelni a kimutatás táblázattal szemben. A beépített KPI<sup>10</sup> mutatók segítségével könnyen beépítheti azokat az automatizált jelentésekbe. (Olafusi 2024)

#### 2.1.4. DAX<sup>11</sup>

A **DAX** a **Power Pivot** és a **Power BI Desktop** táblázatos modelljeinek natív képlete és lekérdezési nyelve. (Deckler 2020) Függvényeket és operátorokat tartalmazó eszközkészlet, amely egy vagy több értéket visszaadó kifejezések létrehozására szolgál. (Horne 2020) Ezen eszköz segítségével számításokat és lekérdezéseket hajtanak végre a táblázatos adatmodellek kapcsolódó tábláiban és oszlopaiban lévő adatokon. (Microsoft 2023) Az **MS Excel** szemben cellák helyett táblázatokra, oszlopokra és egyéb képletekre hivatkozik. Lehetővé teszi számított oszlopok és mértékek létrehozását. (Olafusi 2024) Számos olyan **DAX-függvény** van, amelynek a neve megegyezik az **MS Excel függvények** nevével és funkcióival is, de vannak eltérő funkciókkal bíró, azonos nevű függvények is.

A **DAX-kód** írása nagyon hasonlít más programozási nyelven való kód írásához (Deckler 2020), de nem a hagyományos programozási nyelvekhez hasonlít. Tulajdonképpen a **Power Pivotban** a háttérben automatikusan létrejön egy **DAX-mérték**, hasonlóan az **MS Excel** makró felvevő funkciójához, amely nyomán létrejön a VBA kód. A **DAX** a BI-szakemberek számára lehetőséget ad arra, hogy mélyebb betekintést nyerjenek az adatokba. (Horne 2020) Bár az **MS Excelhez** hasonló, de mégis egy más gondolkodásmód elsajátítása szükséges a használatához.

---

<sup>10</sup> KPI: Key Performance Indicator, olyan kulcsfontosságú mutatók, amelyek segítenek mérni egy vállalat vagy egy projekt teljesítményét és sikerét.

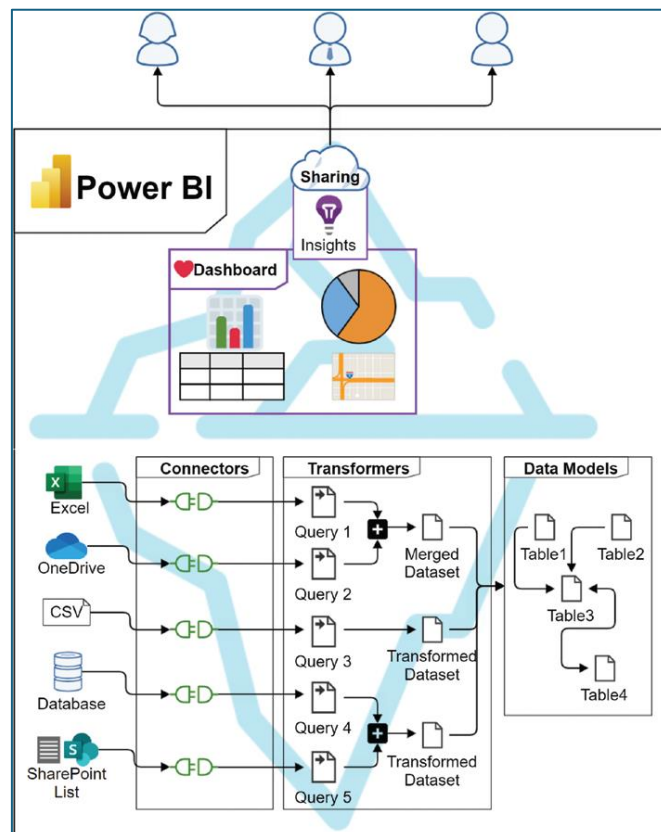
<sup>11</sup> DAX: Data Analysis Expression, egy Power Pivot vagy Power BI által használt funkcionális nyelv, amelyet kifejezetten az adatokkal való műveletekhez fejlesztettek ki.

## 2.2. Power BI a legelterjedtebb

Az adatvizualizációs, adatmodellező eszközök között az egyik legelterjedtebb termék a **Microsoft Power BI** eszköze. Célja az üzleti intelligencia és a vizualizáció támogatása. A **Power BI** lehetővé teszi összetett jelentések létrehozását, szinte bármilyen adatforrás felhasználásával. Az adatok importálási lehetősége az MS Excel-hez hasonló. Előnye, hogy gyorsan elsajátítható, és nem igényel programozói alaptudást. (Foulkes és Sparrow 2020)

Sokan úgy tekintenek a **Power BI-re**, mint egy kibővített MS Excelre, míg mások főként a vizualizációs eszközök aspektusára összpontosítanak. Ezek a nézetek nem helytelenek, csak éppen a felszínt karcolják. A legtöbben csak a **Power BI** jelentés vagy **Dashboard** oldalát látják, ugyanakkor kevesen vannak tisztában a háttérben levő kapcsolatokkal, átalakítással, modellezéssel és felhőszolgáltatásokkal. Pedig ereje talán inkább abban rejlik, amit az emberek nem látnak, mint abban, amit látnak. (Ding 2023)

4. ábra: Power BI folyamatai (Forrás: David Ding - Transitioning to Microsoft Power Platform)

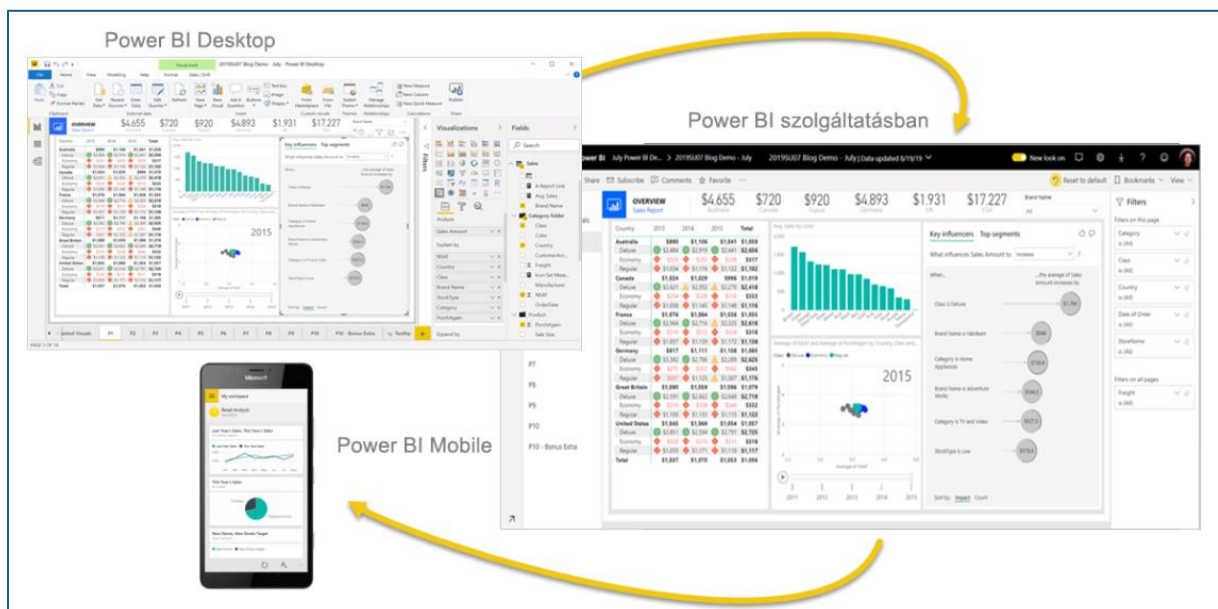




Elmondható, hogy a **Power BI** olyan szoftverszolgáltatások és alkalmazások összessége, amelyek egymással összehangoltan teszik lehetővé vizuálisan vonzó és interaktív elemzések készítését különböző, egymástól független adatforrásokból. (Hopkins 2022)

Elemi úgy lettek kialakítva, hogy egyszerűvé tegye elemzések létrehozását és megosztását. A **Power BI Desktop** a Windows asztali alkalmazása, a **Power BI Szolgáltatás** pedig az online szoftveres megoldás (SaaS<sup>12</sup>), és erre épülnek a **Power BI Mobile-alkalmazások**. (Iseminger, és mtsai. 2024)

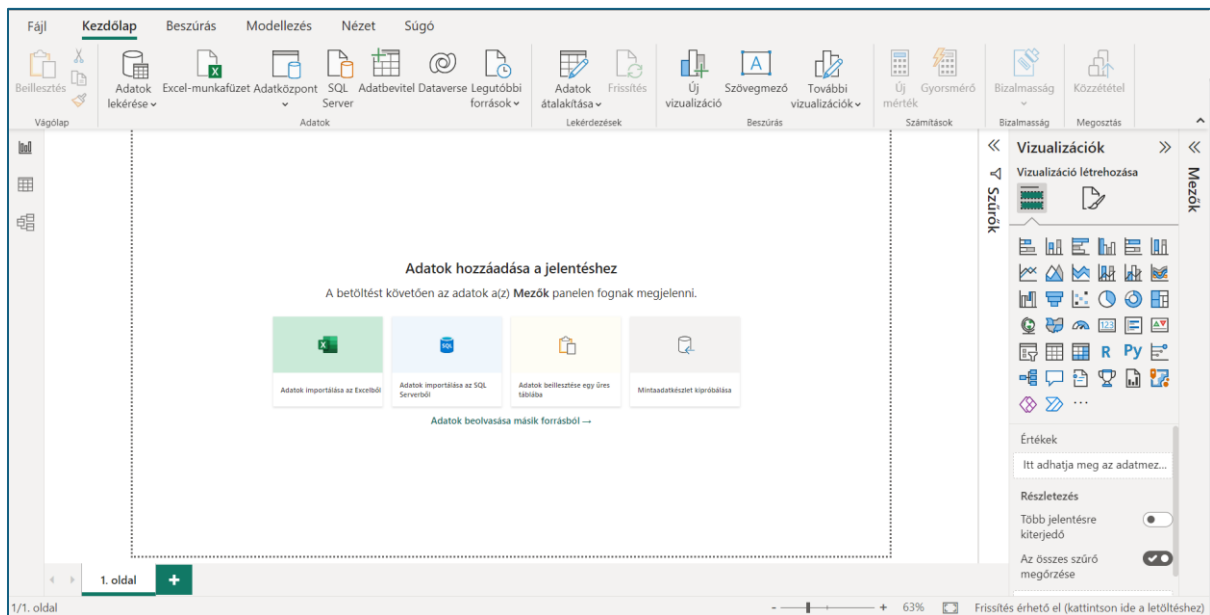
**5. ábra:** Power BI alkalmazások (Forrás: Dokumentáció a Power BI használatba vételéhez)



Az adatok lehetnek Excel-táblák, vagy felhőalapú adatbázisok és helyszíni hibrid adattárházak gyűjteményei. Segítségével egyszerűen csatlakozhat a különböző adatforrásokhoz, megjelenítheti és felfedezheti az egyén számára releváns információkat, és megoszthatja azokat más kompetens személyekkel.

<sup>12</sup> SaaS: Software as a Service – lekérhető szoftver, szoftverszolgáltatási módszer, ahol az adatok felhőben, az interneten vannak tárolva

6. ábra: Power BI Desktop adatbetöltő felülete

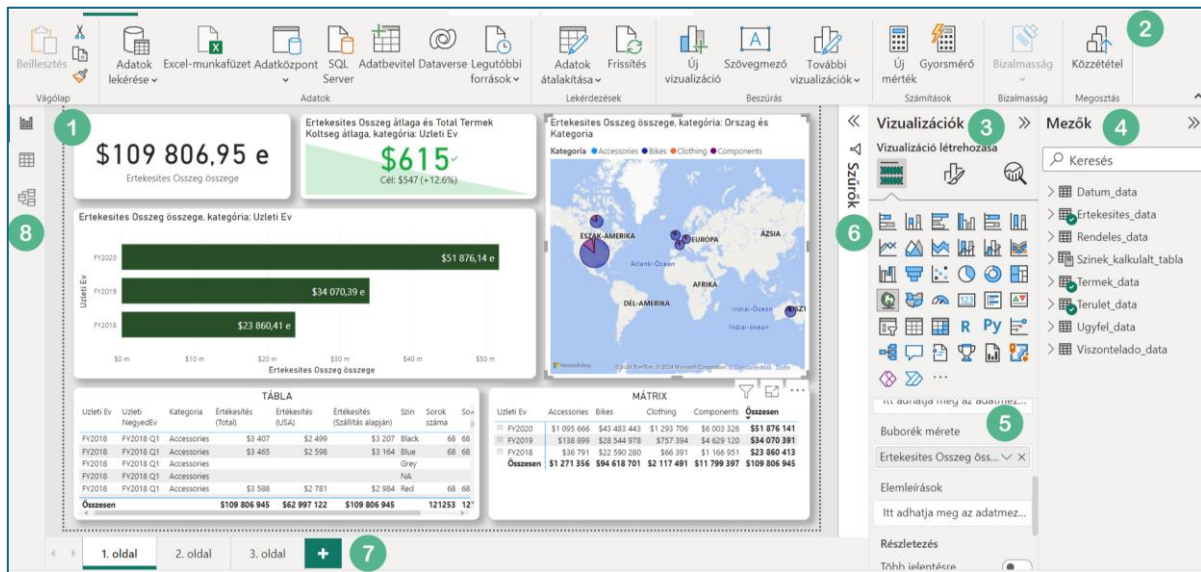


A **Power BI Desktop** alkalmazás egy ingyenesen elérhető eszköz az adatok vizualizációjának létrehozásához. Ezzel az alkalmazással különböző adatforrásokat könnyen kombinálhatunk egyetlen adatmodellben. Ez az adatmodell lehetővé teszi vizualizációk készítését, amelyek egységes struktúrába rendezve könnyen megoszthatók egy adott szervezet vagy vállalkozás belső környezetében.

A **Power BI Desktop** a következő elemekből épül fel (Ding 2023):

1. A vászon, amely a vizuális elemek létrehozásának és beállításának a területe.
2. Az eszköztár a jelentésekre és a vizuális elemekre vonatkozó közös műveleteket jeleníti meg.
3. A vizualizációs panel, az alapértelmezett vizualizáció kiválasztására.
4. Az adatmodellek adatmezői, amely a vizualizációk során meghívhatóak.
5. A vizualizáció formátum beállítását lehetővé tevő panel.
6. A szűrők hozzáadásának alapértelmezett területe a vizuális elemekhez.
7. A vizualizációs oldalak, amely több jelentést is tartalmazhat.
8. A jelentés, adat és modell nézetek közötti váltás felülete.

**7. ábra:** Power BI Desktop elemei (Forrás: David Ding - Transitioning to Microsoft Power Platform - ábrája alapján saját vizualizáció készítése, és beszámozása Canva program használatával)



Ez tehát az a felszín, amelyet a legtöbb felhasználó alkalmaz a vizualizáció során. Kezelése valóban nagyon hasonló a **MS Excel** programban rejlő lehetőségekhez. Most azonban szeretnék kitérni a program három kulcsfontosságú összetevőjére:

- **Power Query** az adattáblák betöltéséhez és átalakításához, adattisztítás eszköze, amelyet már az **MS Excel**nél részletesebben leírtam.
- **Adatmodellek**, az adatok közötti kapcsolatok létrehozásához.
- **DAX függvények**, a jelentésben mérők, oszlopok és táblázatok létrehozásához (Ding 2023)

A VBA-hoz képest a **Power Query M** egy vizuális adatfeldolgozást, adattranszformációt tesz lehetővé. A Power Query M egy adatorientált stílusú programozási nyelv. A Power Query-t becsomagolták a Power BI-ba az adatkészletek importálásához, átalakításához és kombinálásához. Ez az egyik fő ok, amiért a Power BI sok versenytársa előtt jár. (Ding 2023)

A **Power BI** lehetőséget ad arra, hogy könnyen váltson a felhasználó a háttér adatfeldolgozás valamint az előtéri adatmodellek és vizualizációk között. Ez az adatintegrációs képesség sok időt takarít meg a fejlesztési folyamat során, és kritikus része a jelentés automatizálásának.

A **DAX** a Power BI, Power Pivot (MS Excel adatmodellezés) és más analitikai eszközökben használt nyelv, amely dinamikus számításokat és lekérdezéseket tesz lehetővé táblázatos adatmodellekben. A **DAX-képleteket** négy fő területen alkalmazzák: mértékek, számított oszlopok, számított táblák és sorsintű biztonság. A mértékek dinamikus számítások, melyek jelentésekben, például Power BI vagy Excel-kimutatásokban használhatók. Számított oszlopok a táblákban dinamikusan kiszámolt adatokat tartalmaznak, számított táblák dinamikusan létrehozott táblákat reprezentálnak, míg sorsintű biztonság segítségével a DAX beállítható a felhasználói hozzáférés szabályozására. A **DAX-kifejezések eredményei** csak a jelentéskészítő ügyfél alkalmazásokban láthatók és értelmezhetők, mivel ezekben az alkalmazásokban az egyes cellák eredményeit környezetfüggő módon kiértékelik. (Horne 2020)

Egy érthető vizualizációnál érdemes több szempontból is megjeleníteni egy adathalmazt. Fontos létrehozni egy vizualizációs gyűjteményt, amely lehetővé teszi, hogy a jelentéseket többféle szemszögből is megvizsgálhassuk. Így gazdag jelentéseket hozhatunk létre, akár több forrásból származó adatokat is felhasználva egyetlen jelentésen belül.

A **Power BI Service** egy a Microsoft Azure felhőjében található felhőalapú Power BI szerver. A PBI Service tartalmazza a munkaterületeket, a közzétett jelentéseket és adatkészleteket is. 2022 májusában a **Power BI** három fő komponense – a Power Query, az adatmodell és a DAX – mind a felhőbe került. Jelentések közzététele és megosztása nem a jelentésfejlesztés legizgalmasabb része, de fejlesztők számára sok buktatót rejthet, és akár komoly adatvédelmi szabályszegéshez is vezethet. (Iseminger, és mtsai. 2024)

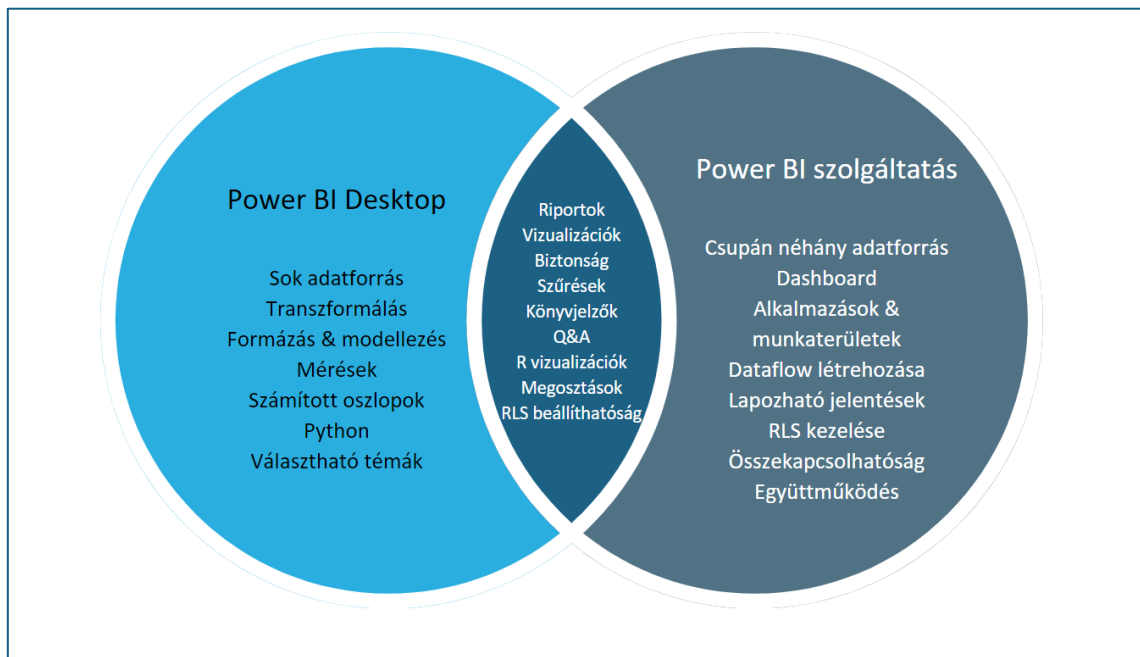
Egyik ilyen buktatója, hogy ha a megosztott feleknek nincs egyformán Power BI Pro előfizetése, bár a jelentéseket meg tudják nézni az ingyenes verzióval, de nem végezhetnek olyan interaktív műveleteket a jelentésben, amelyek valóban vonzóvá tenné az alkalmazás használatát a vezetők számára. Elveszíti azt a pluszt, amelyet egy ilyen vizualitást biztosító eszköznek tudnia kellene.

A Microsoft alapértelmezett megosztási módszere nem megfelelő a szigorú adatvédelmi politikával rendelkező szervezetek számára. A saját munkaterület (személyes munkaterület)

csak a fejlesztő számára elérhető. Lehetőséget nyújt jelentések és adatkészletek tárolására. Azonban, ha a fejlesztő elhagyja a szervezetet, senki sem férhet hozzá a személyes munkaterületen található jelentésekhez. A megosztási linkek lehetővé teszik, hogy azon keresztül bárki hozzáférhessen a jelentéshez. Ez felvet további adatvédelmi kockázatot. Minden egyes linket külön kell vizsgálni annak érdekében, hogy megértsük, ki fér hozzá a jelentéshez és az adatkészlethez, ami bonyolítja a folyamatot.

Az alábbi Venn-diagram a **Power BI Desktopot** és a **Power BI szolgáltatás** hasonlítja össze. A középső rész néhány olyan területet mutat, ahol átfedésben vannak. Néhány feladat a Power BI Desktopban vagy a szolgáltatásban is elvégezhető. A Venn-diagram két külső oldala az asztali alkalmazásra vagy a Power BI szolgáltatás egyedi funkciókra mutat. (Iseminger, és mtsai. 2024)

**8. ábra:** Venn-diagram a Power BI Desktop és Power BI szolgáltatás összehasonlítására (Forrás: Dokumentáció a Power BI használatba vételéhez ábrája alapján saját feldolgozásban)



Az automatizált jelentési folyamatoknak három fő előnye az idő- és az erőforrás-megtakarítás, valamint a humán hibák csökkentése és fenntarthatóság javítása. Miután a jelentést automatizálták, nincs szükség további erőforrásokra az ismétléshez, és a felszabaduló kapacitást fejlesztésre lehet fordítani. A bonyolult, manuális lépésekhez kötődő

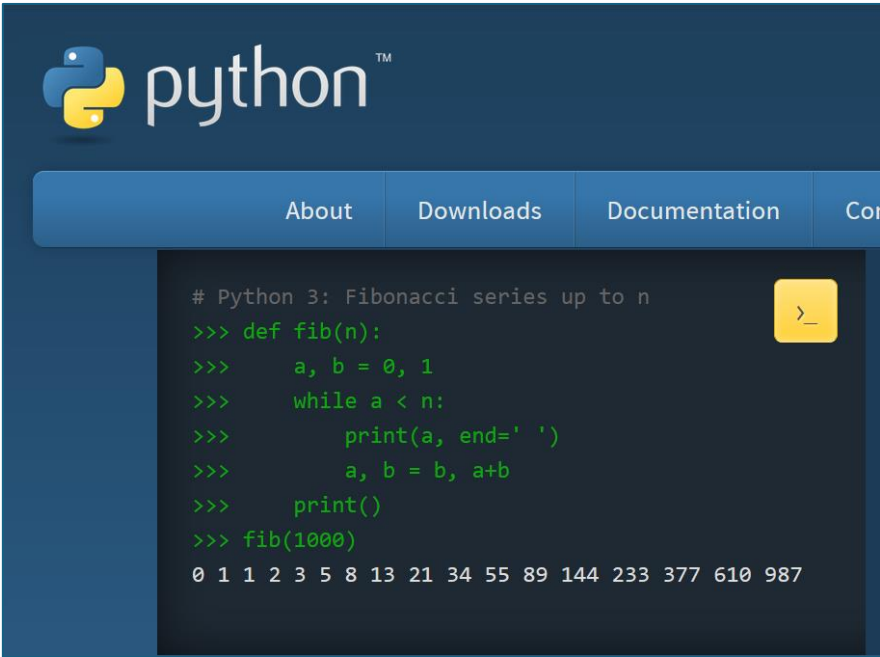
humán hibák minimalizálása és a folyamatok automatizálása az időigényes feladatok megkönnyítését eredményezi. A fenntarthatóság javítása érdekében az automatizált jelentési folyamatok segítenek a tudás átadásában és dokumentációjában, minimalizálva a folytonossági problémákat a csapatváltások során. Az automatizáció költség-haszon elemzése segíthet a legmegfelelőbb döntések meghozatalában a hosszú távú előnyök maximalizálása érdekében.

Az önálló üzleti intelligencia megoldások, mint például a **Power BI**, nagyban javították a felhasználói élményt az önálló adatfelderítés terén. A felhasználó az alapadatokat felhasználhatja további elemzésekhez.

### 2.3. Python – Pandas, Matplotlib, Bokeh, Seaborn könyvtárak

A **Python** az egyik legnépszerűbb programozási nyelv az adattudományban, amely eszközöket biztosít az adatok összegyűjtésére és feldolgozására. (So, és mtsai. 2020) Mivel nyílt forráskódú, az adattudomány könnyen elérhetővé válik, kihasználva a mások által írt könyvtárakat a gyakori adatfeladatok és problémák megoldására.

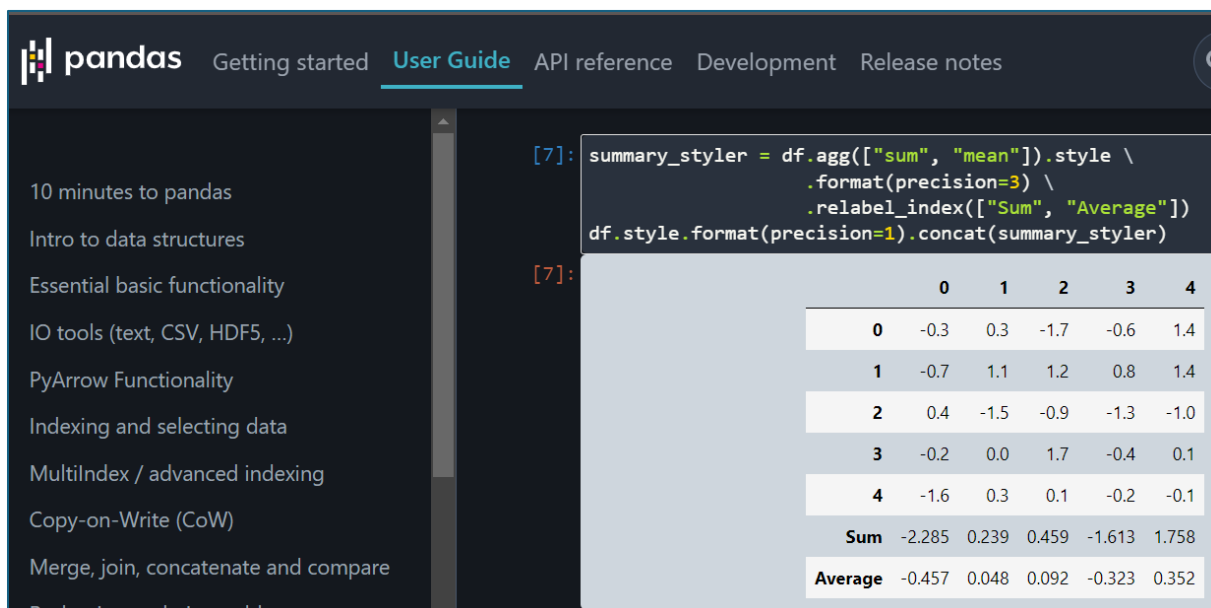
**9. ábra:** Python (Forrás: <https://www.python.org/downloads/>)



```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

A **Pandas** az egyik legelterjedtebb adatelemző könyvtára. Hihetetlen számú csomagot kínál az adatok tisztítására, további könyvtárak segítségével pedig különböző szintű interaktivitást és vizualizációs lehetőségeket kínál. Míg a **Pandas** szigorúan táblázatos formában dolgozik az adatokkal, az adatvizualizációra több más könyvtár áll rendelkezésre.

**10. ábra:** Pandas könyvtár (Forrás: [https://pandas.pydata.org/docs/user\\_guide/style.html](https://pandas.pydata.org/docs/user_guide/style.html))



The screenshot shows the Pandas User Guide interface. On the left is a navigation menu with items like '10 minutes to pandas', 'Intro to data structures', 'Essential basic functionality', 'IO tools (text, CSV, HDF5, ...)', 'PyArrow Functionality', 'Indexing and selecting data', 'MultiIndex / advanced indexing', 'Copy-on-Write (CoW)', 'Merge, join, concatenate and compare', and 'Packaging and pivot tables'. The main content area displays a code snippet and its output:

```
[7]: summary_styler = df.agg(["sum", "mean"]).style \
      .format(precision=3) \
      .relabel_index(["Sum", "Average"])
df.style.format(precision=1).concat(summary_styler)
```

```
[7]:
```

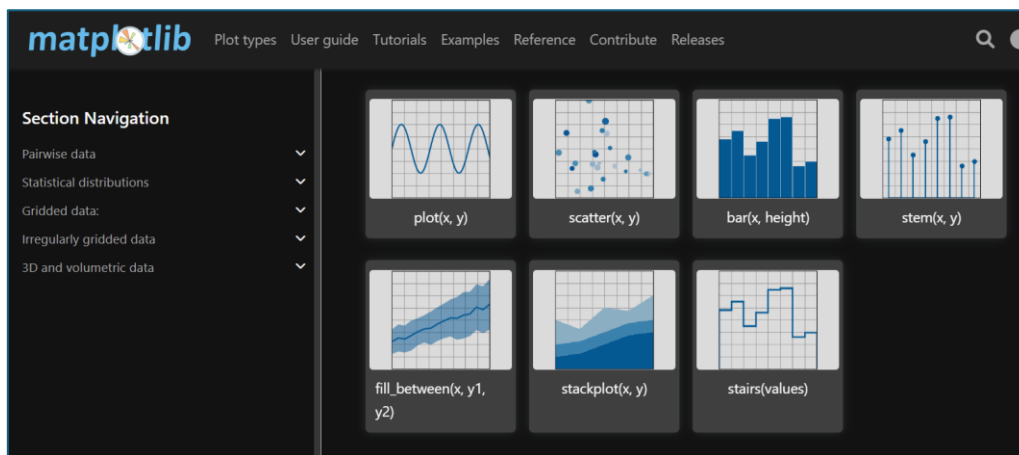
	0	1	2	3	4
0	-0.3	0.3	-1.7	-0.6	1.4
1	-0.7	1.1	1.2	0.8	1.4
2	0.4	-1.5	-0.9	-1.3	-1.0
3	-0.2	0.0	1.7	-0.4	0.1
4	-1.6	0.3	0.1	-0.2	-0.1
Sum	-2.285	0.239	0.459	-1.613	1.758
Average	-0.457	0.048	0.092	-0.323	0.352

A **Matplotlib** és a **Seaborn** könyvtárak a **Python** adatvizualizációban a statikus diagramok ábrázolását teszik lehetővé. Nagy jelentőségük van a feltáró adatelemzés során, mivel egyszerűen megvalósíthatóak, és nagyon gyorsan elkészíthetőek.

A **Matplotlib** a legrégebbi **Pythonos** adatvizualizációs könyvtár. (Balogh 2021) Objektorientált API<sup>13</sup>-t kínál, amely lehetőséget biztosít a diagramok beágyazására alkalmazásokba, általános célú GUI eszköztárak segítségével. Eredeti célja egy olyan ábrázoló eszköz kifejlesztése volt, amely vetélytársa lehet más szoftverkörnyezetekben található diagramkészítő eszközöknek, mint például a MATLAB, Octave, R, és így tovább. Ez a könyvtár lehetővé teszi statikus, közzétételi minőségű vizualizációk létrehozását.

<sup>13</sup> API: Application Programming Interface, azaz alkalmazásprogramozási felület.

11. ábra: Matplotlib (Forrás: [https://matplotlib.org/stable/plot\\_types/arrays/index.html](https://matplotlib.org/stable/plot_types/arrays/index.html))



**Matplotlib** csomag meglehetősen nagy, mivel elég sok funkciót tartalmaz. Szerencsére a legtöbb ábrázolási feladatunkhoz csak a **pyplot** modulra van szükségünk, amely biztosít egy MATLAB-szerű ábrázolási keretrendszert. Tudományos körökben a MATLAB nyílt forráskódú alternatívájaként emlegetik. (Molin 2021) Lehetőséget biztosít gyors és egyszerű **Python** alkalmazta vizualizációk készítésére, bár feltételez némi programozói tudást. Alapvető rajzoló funkciókat kínál, amelyekkel széles körű vizualizációkat hozhatunk létre. Egyedi, testre szabható ábrák alkothatóak a segítségével. Lehetővé teszi az MS Excel vagy a Power BI által kínált sablonoktól való elrugaszkodást és egyedi, kreatív alkotások elkészítését.

A **Matplotlib** széles körben alkalmazható különböző kontextusokban, beleértve a kutatási, adatelemzési és adattudományos adatvizualizációt. A jó vizualizáció készítésének a kulcsa, hogy jól kell tudni meghatározni minden elemet, ami a vizualizáció megjelenítésében szerepet játszik. Minden grafikon rengeteg egyszerű tulajdonsággal bír. Például a szín, minták, stílusok, effektek, amelyek beállításra az Microsoft termékek esetében külön panelek állnak rendelkezésre.

A Matplotlib tehát különféle vizualizációs lehetőséget kínál, beleértve a vonaldiagramokat **plot()**, szórásvonal diagramokat **scatter()**, oszlopdiagramokat **bar()**, hisztogramokat **hist()** és még sok más. Egyik kulcsfontosságú jellemzője a nagyfokú testreszabhatóság, amely lehetővé teszi a diagramok megjelenésének széles körű alakítását. A színek variálhatósága is számos lehetőséget biztosít. (Rajender 2023)



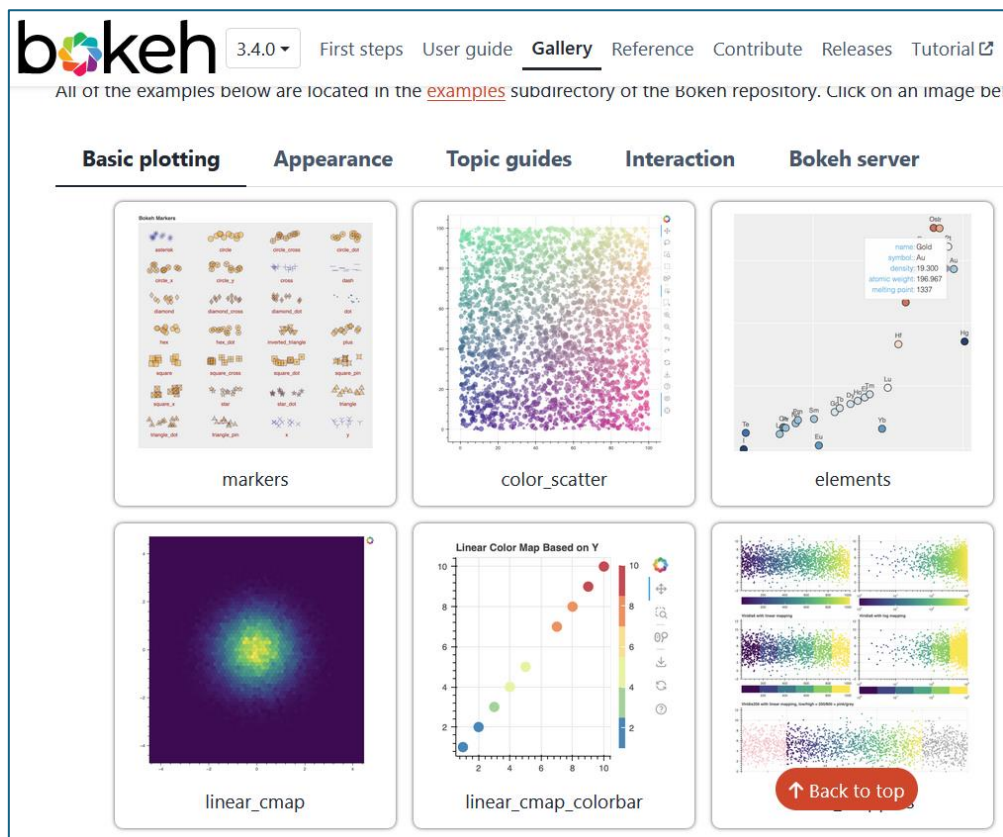
Az alapvető alakítási lehetőségeken túl a Matplotlib további fejlett formázási lehetőségeket is kínál, például különböző jelölők használatát, tengelyhatárok módosítása és jelmagyarázat hozzáadása. Ezeket arra lehet használni, hogy informatívabbá és vizuálisan vonzóbbá váljanak a diagramok. Továbbá számos előre definiált stílust is biztosít a vizuálisan vonzó diagramok létrehozásához. (Rajender 2023)

Összességében a Matplotlib hatékony eszköz lehet az adatokban rejlő összefüggések és mintázatok hatékony kommunikálására. Azonban, mint minden eszköz esetében, fontos, hogy helyesen használjuk, hogy a legtöbbet hozzuk ki belőle. Íme néhány dolog, amelyeket érdemes szem előtt tartani használatakor. (Rajender 2023):

- **Egyszerűség:** kerülendő a diagramok túlzásúfolása. A kevesebb néha több elvet érdemes megtartani, amelyek segíthetnek az üzenet átadásában.
- **Megfelelő diagramtípus:** különböző diagramtípusok jobban megfelelnek bizonyos típusú adatokhoz. Például egy vonaldiagram jobb a trendek megjelenítéséhez az idő függvényében, míg egy oszlopdiaagram jobb a kategóriák összehasonlításához.
- **Hatékony színhasználat:** a szín az egyik legjobb eszköz a fontos információk kiemelésére, de megfontoltan kell használni. Kerülendő a túl sok különböző szín használata.
- **Címkézett tengelyek:** az x és y tengelyeken az egységek és skálák egyértelműen legyenek tüntetve, hogy a címezett megértse az adatokat.
- **Cím és leírás:** használata segíthet megmagyarázni a diagram fő üzenetét.

Ha a Matplotlib ábrái nem lennének elég vonzóak, akkor a **Bokeh** lehetőséget biztosít arra, hogy interaktív és esztétikus vizualizációkat hozzunk létre, amelyek nem csupán statikus ábrák, hanem olyan diagramok, amelyekkel a felhasználók interakcióba léphetnek. Ez a könyvtár JavaScript-alapú adatvizualizáció készítésére alkalmas eszköz, amely **Python** kódot használ. Kisebb projektek esetében, üzleti környezetben ideális a formálhatósága miatt. Nagy hátránya azonban, hogy nehezebben tanulható, így több időt igényel a kiépítése. (Balogh 2021)

12. ábra: Bokeh (Forrás: <https://docs.bokeh.org/en/latest/docs/gallery.html>)



A **Bokeh** különösen hasznos nagy adathalmazok vizualizálására, mivel támogatja a streaminget és nagy adathalmazokat. Különösen hasznos webes alkalmazásokban és adatanalízisben, ahol interaktív és dinamikus vizualizációk szükségesek.

A következő néhány kulcsfontosságú definíció a **Bokeh-hez** kapcsolódik (Jolly 2018):

- **Alkalmazás:** Az alkalmazás egy a böngészőben futó Bokeh dokumentum, amely interaktív ábrákat és alkalmazásokat jelenít meg.
- **Glyphok:** A glyphok a Bokeh építőelemei, ezek a vonalak, körök, téglalapok és más formák, amelyeket egy Bokeh ábrán látható.
- **Szerver:** A Bokeh szerver hídként szolgál, amely összeköti a Python-t és a böngészőt, amely az alkalmazást tárolja.
- **Widgetek:** A widgetek interaktivitást biztosítanak a csúszkák, legördülő menük, a szövegdobozok és egyéb kis eszközök révén, amelyek beágyazhatóak az ábrába.

A **Seaborn** adatvizualizációs könyvtár, amely a **Matplotlib** könyvtárat alapul véve teszi lehetővé az esztétikusabb és informatívabb vizualizációk létrehozását. A **Seaborn** fő előnyei közé tartozik az egyszerűség és a gyorsaság, valamint a magasabb szintű absztrakció, amely lehetővé teszi a felhasználók számára, hogy kevesebb kódot használva érjenek el összetett diagramokat és adataik jobb megjelenítését.

**123. ábra:** Seaborn (Forrás: <https://seaborn.pydata.org/index.html>)



A **Seaborn** beépített témákat és stílusokat kínál, amelyek javítják a vizualizációk minőségét és könnyebbé teszik az adatok vizualizációját. Különösen hasznos nagy adathalmazok vizualizálására, mivel számos speciális rajzadási funkciót kínál, amelyek könnyen testre szabhatók.

### 3. Online térből gyűjtött adatok elemzése

Az adatokat az ötödik fejezetben megjelölt forrásként megjelölt magyar weboldalakról gyűjtöttem össze. Az adatbázisokban való kutatás az egyik legnagyobb kihívást jelentette számomra, mert nem volt egyszerű olyan adathalmazt találni, amelyre több elemzési módszert is ki tudtam próbálni. Ezért nemegyszer kezdtem újra a vizsgálatot. Néztem egészségügyi, illetve gazdasági adatokat különböző magyar és nemzetközi adatbázisokban megtalálható egészségügyi, illetve gazdasági adatokat. Végül mivel a felsőoktatási adatok elemzése – bár más aspektusban – mindennapi feladatom részét képezik, ezeket az adatokat választottam.

#### 3.1. Adatfeldolgozás és vizualizáció MS Excelben

##### 3.1.1. Adatok beimportálása és adattisztítás Power Query-ben

Az adatokat részben a közvetlenül a weboldalról hivatkoztam be, részben pedig ahol nagyobb adatbázis állt rendelkezésre, ott CSV<sup>14</sup> fájlként töltöttem le, majd így importáltam be az **MS Excel** felületről elindított **Power Query-be**, amiben az adatok tisztítását végeztem. Nagy segítség volt az oszlopok elemzésére szolgáló beépített eszköz, amely a fejléc alatt közvetlenül egy rövid összefoglalást biztosított, illetve előhívható volt egy alap statisztikai elemzéseket is tartalmazó oszlopprofil diagnosztika.

A következő változtatásokat végeztem el:

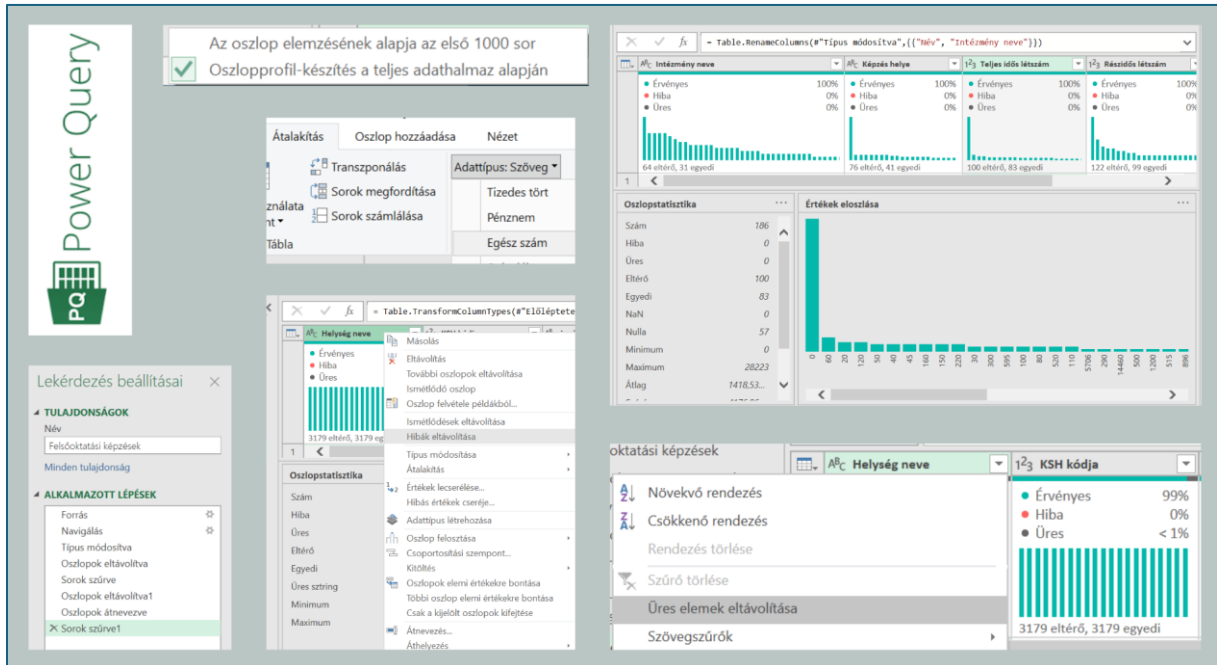
- Eltávolítottam a nem releváns adatokat tartalmazó oszlopokat.
- Eltávolítottam a hiányzó értékeket és a hibákat.
- Átneveztem az oszlopfejléceket, egységesítve azokat a különböző forrásból származó adatbázisokon.
- A redundanciák kiszűrésével a táblákból külön elemi adatokat tartalmazó táblákat hoztam létre.

---

<sup>14</sup> CSV: Comma-Separated Values: szövegfájlban tárolt adatok, amelyekben az értékek vesszővel vannak elválasztva

- A számként szereplő értékeket, amennyiben nem ismerte fel azt az eszköz, egész számmá alakítottam át.

13. ábra: Adattisztítási lépések bemutatása Power Query felületen (Saját forrásból)



Ezeknek a lépéseknek a szükségessége azért volt fontos, hogy a különböző adatforrásokból származó adatokat egységesítem mind formátumban, mind adattípus tekintetében, ami nagyon fontos az azt követő adatfeldolgozás és elemzés során, amely az adatintegritás javítását szolgálja, különösen nagy adathalmazok esetén.

A változtatások minden esetben megjelentek az alkalmazott műveletek végrehajtott lépések ablakában, lehetővé téve azok egyszerű nyomon követését és visszavonását szükség esetén. A folyamat során látható volt a táblázat szerkezete, és minden változás vizuálisan értelmezhető formában jelent meg. A szöveges módosítások rögzítése külön sorokban történt, és visszavonható volt, ha mégsem volt szükséges. Ez a lépés lehetővé tette a további munkafolyamat során a módosítások rugalmas visszavonását. Egy-egy újabb tisztítás – bár frissítést igényelt – többnyire zökkenőmentesen átvihető volt a már részben elkészített vizualizációba. Ugyanakkor az egymásra épülő módosítások esetén előfordulhattak problémák. Ezért volt érdemes a módosításokat elemi lépésenként rögzíteni.

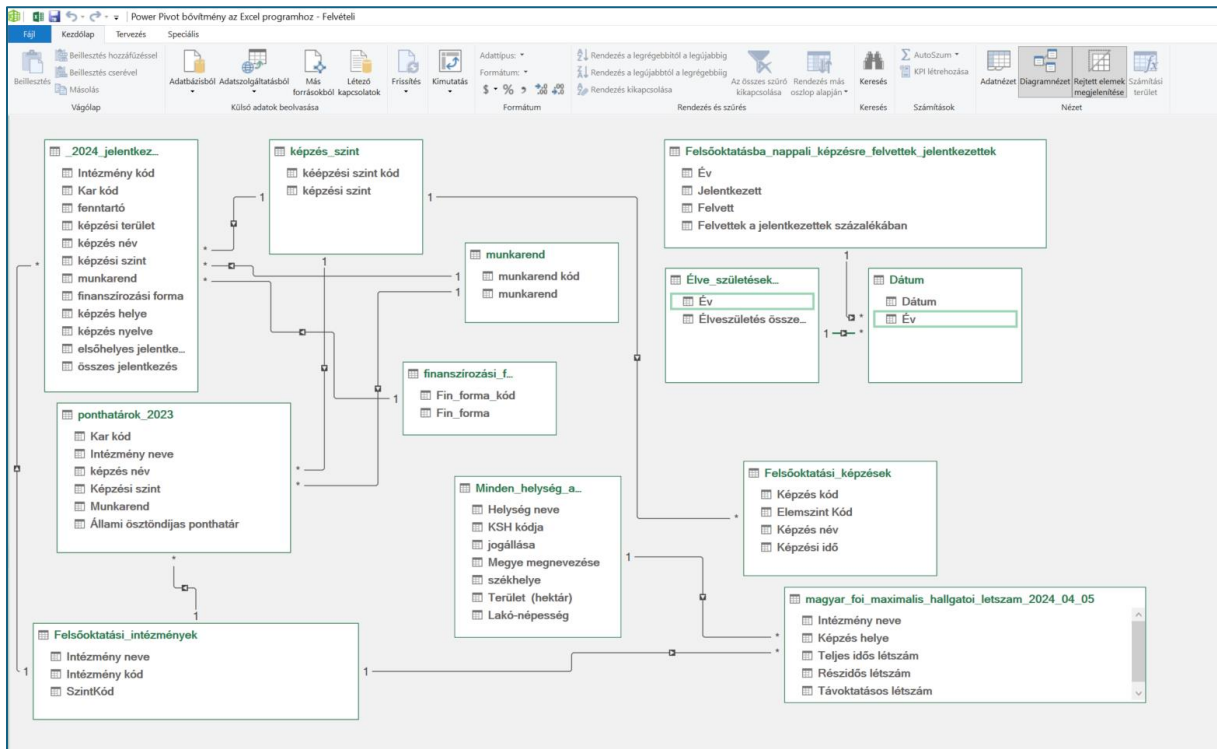
A példában statisztikai adatgyűjtésből származó adatokat használtam, ezért az adattisztításra a **Power Query** felületén kevesebb szükség volt. Azonban egy nagyobb adatbázisban jelentősen több időt igényelhet az adattisztítás, ha az nincs előfeldolgozva, vagy esetleg egy saját mérés nyomán került rögzítésre. Ez a folyamat mindig a legidőigényesebb részét képezi az elemző szakember munkájának. De nem csak azért, mert a hibás adat hibás döntéshez vezethet, hanem mert egy rosszul felépített tisztítási folyamat sem a megfelelő következtetéseket eredményezi. Néha előnyösebb lehet több órás munkát elengedni, és újra kezdeni az adatok feldolgozását, mintsem hibásan tisztított adatokkal dolgozni. Az ilyen folyamatok során az elemző egyre jobban megismerkedik az adatokkal és azok sajátosságaival, ami hosszú távon előnyös lehet.

### 3.1.2. Adatkapcsolatok kiépítése

Az adattisztítás után létrehoztam a kapcsolatokat a táblák között. Az összehasonlítás során azt tapasztaltam, hogy az **MS Excelben** történő kapcsolatépítés hasonlóan működik, mint a **Power BI** esetében. Azonban meg kell jegyezni, hogy ez utóbbi esetben az adattáblák, a kapcsolatok és a jelentések közötti váltás egyszerűbb volt a rendelkezésre álló eszközkészlet segítségével, addig az **MS Excelben** az adattárolás és a kapcsolatok kezelése a **Power Pivot** modulban történt, amely egy külön felület, és ez némiképp bonyolította az eljárást.

A kapcsolatokban két csoportot különítettem el. Két idősoros táblához egyszerű rögzítéssel hoztam létre egy összekötő dátum táblát. A többi tábla statikus adatokat tartalmazott, így azokhoz ennek megfelelő lekérdezéseket készítettem.

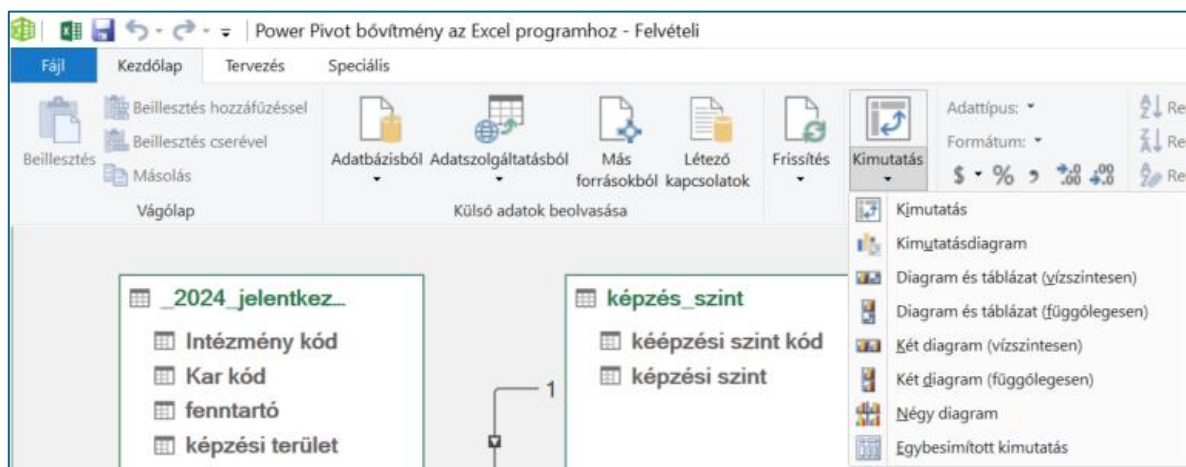
14. ábra: Power Pivotba beimportált és létrehozott összekötő táblák közötti kapcsolat háló (Saját forrás)



### 3.1.3. Adatvizualizáció készítés Pivot Table alkalmazásával

A **Power Pivot** felületről közvetlenül – akár a kapcsolati diagramnézetből – az eszköz tálcán levő lenyíló ikon segítségével többféle kimutatást tudtam készíteni.

15. ábra: Power Pivot kimutatás készítő eszköztár (Saját forrásból)



A kimutatások elkészítésekor a **Pivot Táblák** készítéséhez hasonló mezőlista elrendezéssel találkoztam. Az első két diagram esetében előre beállított formázást választottam.

A sáv diagram esetében a top tíz értékre szűrtem. Megállapítható a vizualizáció nyomán, hogy még mindig tartja első helyét az Eötvös Loránd Tudományegyetem, nagyon népszerű a jelentkezők körében.

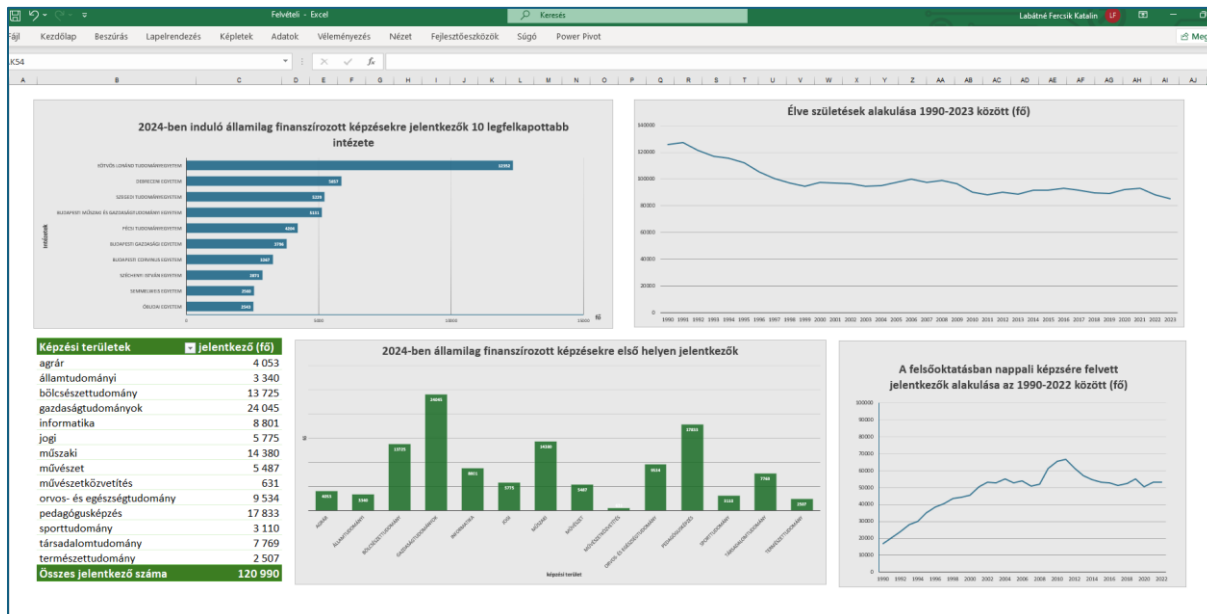
A vonaldiagram alapján látható, hogy az 1990-es évektől kezdve folyamatosan növekedett a felsőoktatási intézményekbe jelentkezők száma, melynek emelkedése 2003-ig lineáris és egyenletes volt. Ezután némi visszaesés következett, majd 2010-ben tetőzött a jelentkezők száma, azóta pedig csökkenő tendenciát mutat. Ezt a tendenciát számos tényező befolyásolhatta, melyek közül néhányat a következő hipotézisek is megvilágítanak:

- A korábban érettségizők egy ideig megnövelték a beiratkozók számát, mert újra vizsgázás nélkül be tudtak iratkozni az egyetemekre, de ez a tartalék egy idő után kimerült. Ez arra utalhat, hogy az emelkedő trendben részben a korábbi évfolyamok többletjelentkezése játszott szerepet, ami hosszú távon nem fenntartható.
- A születésszám változása nemcsak a gyerekvállalási kedv lassú csökkenése miatt alakult így, hanem demográfiai változások is befolyásolják. Például a Ratkó-korszakban az abortusztilalom miatt megugrott a gyereklétszám, majd amikor ezek a gyerekek felnőttek, újra volt egy demográfiai csúcs.
- Egyre többen választják a magyar egyetemek helyett más uniós országok egyetemeit. Ez a folyamat a különböző oktatási rendszerek közötti verseny következménye lehet, és hozzájárulhat a hazai jelentkezők számának csökkenéséhez.
- Szintén fontos tényező lehet, hogy szülőképes korú fiatalok külföldre vándorolnak, ott vállalnak munkát, ott alapítanak családot. Ez a demográfiai eltolódás csökkentheti a hazai felsőoktatási intézményekbe jelentkezők számát.

A képzések tekintetében a 2024-es évben első helyre jelentkezők között még mindig kiemelkedő a gazdasági képzési területre vágyók száma. Ami kifejezetten jó hír, hogy második helyen a pedagógus képzésre jelentkezők vannak. Harmadik helyen szerepelnek a műszaki képzések, ami remélhetőleg az innováció és kutatás terén segíti majd a tudományos területek fejlődését.



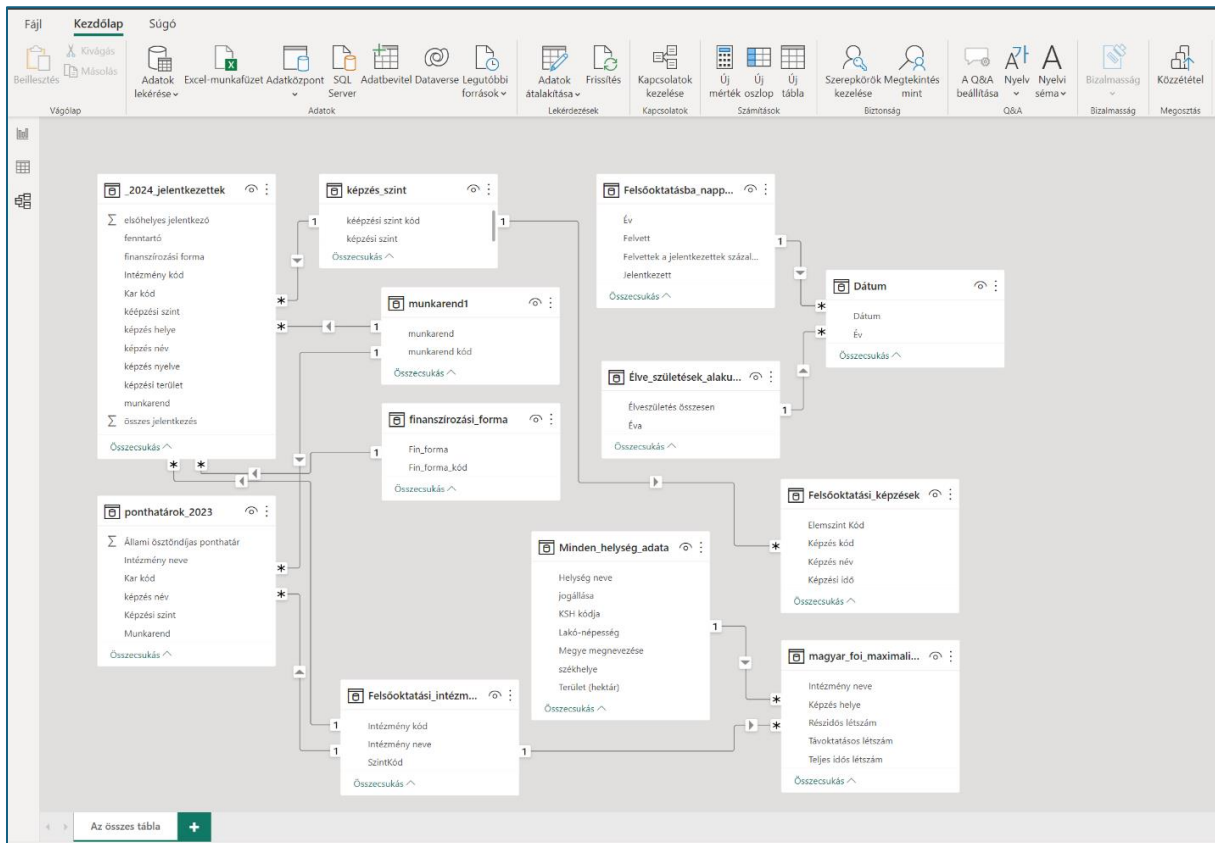
16. ábra: MS Excel és Power Pivot vizualizáció alkalmazása (Saját forrás)



### 3.2. Vizualizáció készítése Power BI felületen

Ezt követően nem végeztem már újra el az adattisztítási folyamatokat, lévén ugyanaz az eszköz használható **Power BI-ban** is némi apró eltéréssel. Már a kész szűrt adathalmazt emeltem be az **MS Excel** fájl segítségével, ami a webes hivatkozásokat épp úgy áthúzta, mint a meghajtón lévő fájlokra való hivatkozásokat, így a háttér tábláival együtt importáltam be a **Power BI** felületre. Több kapcsolatot fel is ismert a rendszer, de azért még szükséges volt jó pár beállításra és rendezésre, hogy a **Power Pivotban** készített kapcsolati ábrához hasonló szerkezet álljon elő.

### 17. ábra: Power BI kapcsolati tábla (Saját forrás)



#### 3.2.1. Vizualizációs háttér kialakítása Canvában

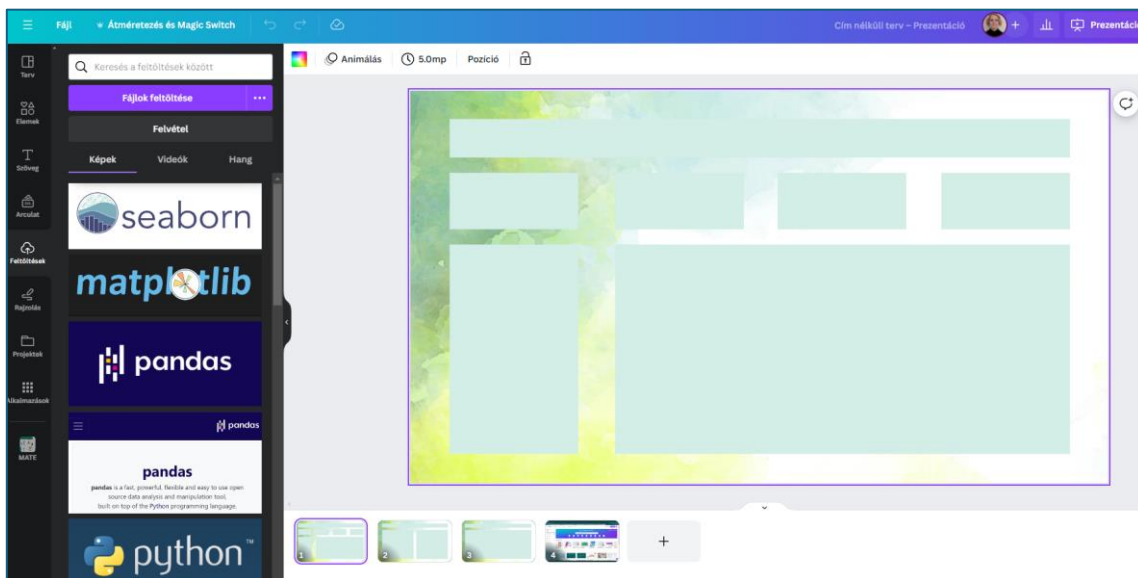
Amikor először megnyitottam a **Power BI-t**, egy üres vászon fogadott. Ahhoz, hogy egy látványos, mások számára is vonzó jelentést állítsak össze, szükség volt egy egyedi háttér beállítására. Egy jelentésnél ez lehet egy színes háttér, egy céglogó, vagy akár egy kép is, amely összhangban van a jelentés témájával. Azonban a „kevesebb néha több” elvet követve inkább egyszerű háttérrel érdemes alkalmazni, amely importálható külső forrásból, vagy elkészíthető valamilyen grafikus szerkesztőprogram használatával. Én is ezt az utóbbi módszert használtam, és megalkottam a **Canva** segítségével a jelentéseim háttérét.

A **Canva** rendkívül sokoldalú eszköz, amely lehetővé teszi széles körű grafikai tervezési feladatok elvégzését, beleértve a prezentációk, posztok, infografikák és szórólapok készítését is. A platform intuitív felülete és felhasználóbarát jellege lehetővé teszi a felhasználók számára, hogy gyorsan és hatékonyan hozzanak létre professzionális kinézetű grafikat, minimális szaktudás vagy tapasztalat nélkül is. Számos előre elkészített sablont,

grafikai elemet, betűtípust és egyéb eszközt kínál. Bár ingyenesen elérhető, több extra funkcióhoz, például speciális sablonokhoz vagy prémium képekhez való hozzáféréshez előfizetés szükséges. Összességében ideális eszköz lehet mind az amatőr, mind a profi grafikusok számára a gyors és hatékony grafikai tervezéshez. Felülete sokkal rugalmasabban kezelhető, mint a már sokak által használt és ismert **Power Point** prezentáció szerkesztő.

Amikor színeket választottam a jelentések, diagramok vagy vizualizációk tervezésekor, figyelembe vettem néhány alapelvet annak érdekében, hogy az adatok könnyen értelmezhetőek és vizuálisan vonzóak legyenek a célközönség számára. Bár egy professzionális jelentés esetén előnyösebb a monokróm, vagy triadikus színek használat, én inkább egy élénk és dinamikus megjelenítés mellett döntöttem, így analóg színeket használtam. Mivel a zöld szín hagyományosan a gazdasággal a jóléttel társított, gyakran használt gazdasági kimutatásokban vagy reklámkampányokban. Illetve mivel dolgozatomat nem kötötte egy vállalat által alkalmazott színpaletta, erre a színre fókuszálva kerestem hátteret a **Canvában**.

**18. ábra:** Elkészült háttér Canva program segítségével (Forrás: <https://www.canva.com/> helyen készített saját forrás)



Az elemzéshez használt hátteret prezentáció formában mentettem el, majd **Power Point** alkalmazásban megnyitva képként tároltam le. Így volt a legoptimálisabban beilleszthető a **Power BI** munkalapra. Ez a módszer lehetővé tette, hogy a prezentációban gondosan

kialakított háttérkép és tartalom strukturált formában legyen megjeleníthető a **Power BI** munkalapon, ezáltal a végeredmény meglehetősen megnyerő lett.

### 3.2.2. Elemzés összeállítása, vizualizációs eszközök kiválasztása

A Power BI felhasználói felületén több elemzői grafikát alkalmaztam. Az intézmények szűrési kritériumait függőleges listaként prezentáltam. Az adatok kis tábláinál a kártya ábrázolási módot használtam, és az eszköz szűrési feltételeit beállítottam a kívánt paraméterek alapján. Minden elemnél aktiváltam a több jelentésre kiterjedő szűrést, így a vizualizációs mezők beállítása dinamikusan változtathatóvá vált. Ennek eredményeként a sávdiagram interaktivitást biztosított az intézmények fókusz szerinti megtekintésére.

19. ábra: Power BI eszközzel készített elemzés dinamikusan változó adatokkal (Saját forrás)

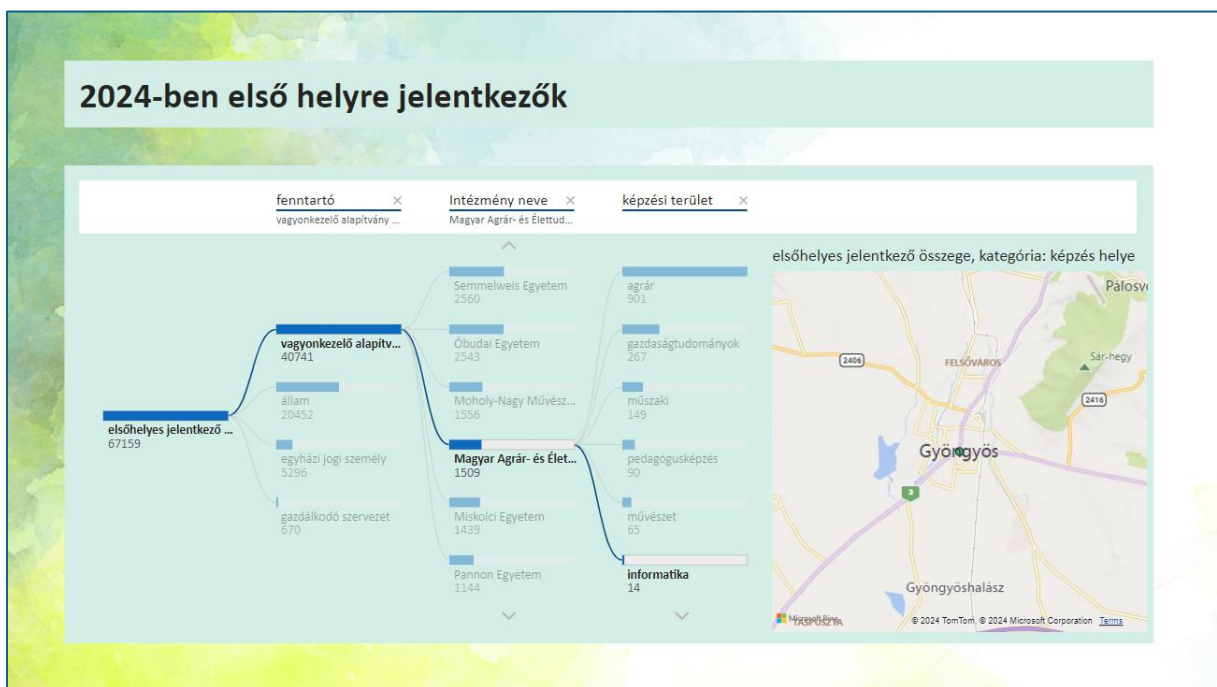


A 2024-es felvételi eljárás adatait felhasználva egy döntési fát, más néven felbontási fát hoztam létre az adatok szemléltetésére a következő ábrán. Az első helyre jelentkezők adatait kiindulópontként használva létrehoztam egy strukturát, amely segítségével lebontottam azokat a fenntartótól az intézményen át a képzési területig. Ebben a hierarchikus strukturában az egyes ágak végén megjelent az adott képzésre jelentkezők száma is. Ezután a térképes vizualizációban elhelyeztem az adott képzési területhez tartozó képzési helyeket

Magyarország közúti térképén. A két kimutatást összekapcsoltam, és így a vizualizációban a döntési szalon végig haladva, a képzésig eljutva megjelent annak térképes vizualizációja is.

Bár első pillantásra talán egy vezetőség számára haszontalannak tűnhet, a felvételiző diákok számára valóban hasznos információkat rejthet, amelyek segíthetnek az adott képzési hely pontos azonosításában. Például az Educatio Kiállításon egy interaktív információs pultnál nagyon hasznos lehet egy ilyen kitekintés, akár a múltbéli adatokra vonatkoztatva is, hiszen a diákok nem feltétlenül vannak tisztában ezekkel az adatokkal, ha már korán, akár tizedik osztályban érdeklődni kezdenek a jövőjükkel illetően.

**20. ábra:** Döntési fa és azzal együtt dinamikusan változó térkép Power BI felületen (Saját forrás)



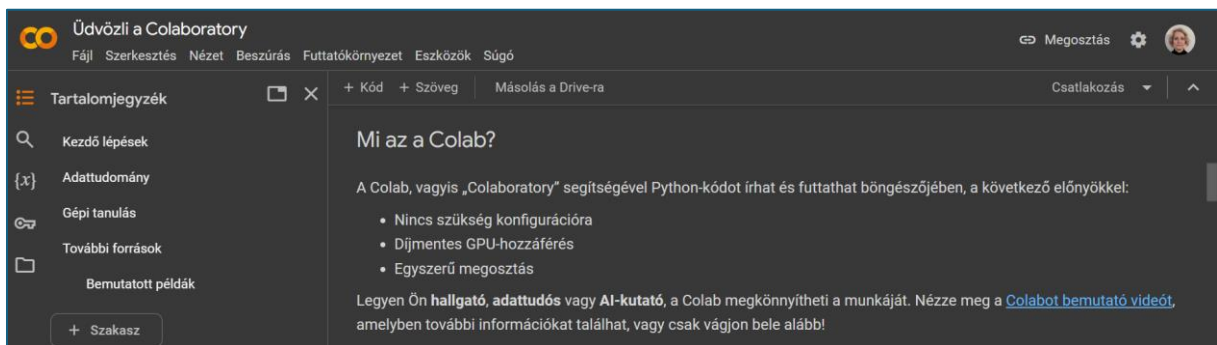
Számomra ez tűnik a legvonzóbb megoldásnak a dinamikus felülete miatt. Nagyon látványos, sok lehetőséget biztosít, és igazán jó felhasználói élményt nyújt mind a készítő, mind a felhasználó számára. Csupán annyi nehézség van a felület használatában, hogy nem konzekvensen van felépítve az egyes beállítási lehetőségek menürendszere. Egy-egy frissítés nyomán eltűnik, vagy átalakul egy már jól bejáratott vizualizáció, mert új helyre kerül vagy új elemmel egészül ki az eszköztára. Mindenképp érdemes figyelni a folyamatos fejlesztéseket, és megnézni az ezekhez készült dokumentációkat, hogy elkerülje a felhasználó a változás okozta kellemetlenségeket.

### 3.3. Adatelemzés Python segítségével

#### 3.3.1. Google Colab felület bemutatása

A **Python-os** elemzéshez az egyik legnépszerűbb online felületet használtam, a **Google Colabot**. Ez egy felhő alapú Jupyter Notebook szolgáltatás, amely ingyenesen elérhető. Nagyszerű lehetőséget biztosít a **Python** kód írásának a futtatására böngészőből. Az egyik legjelentősebb előnye az, hogy lehetőséget biztosít GPU<sup>15</sup> és TPU<sup>16</sup> erőforrások ingyenes használatára. Ez különösen hasznos, ha olyan számításigényes feladatokat kell végrehajtani, mint a mélytanulásban való modellképzés. Mindemellett kiváló platform a tanuláshoz, a kísérletezéshez, az adatelemzéshez és modellezéshez, mert rögtön az online felületen részenként lefuttathatók a kódok, és látható azok eredménye. Ezen felül számos könyvtár rendelkezésre áll, így nem szükséges azok telepítésére időt fordítani.

**21. ábra:** Google Colab felület (Forrás: <https://colab.research.google.com/>)



A Microsoft termékekhez hasonlóan a **Google Colab** felhasználói felületére is közvetlenül feltölthetők az elemzésben használt fájlok, ami kényelmes és hatékony módszert biztosít az adatelemzéshez és modellek készítéséhez. Ennek az előnye, hogy nincs szükség külső adatforrásokra vagy bonyolult fájlkezelési eljárásokra. A felhasználók egyszerűen hivatkozhatnak a feltöltött fájlokra a fájlkiválasztó segítségével, mert azok közvetlenül elérhetővé válnak a munkamenetben. Ez a gyors és egyszerű megoldás jelentősen csökkenti

<sup>15</sup> GPU: Graphics Processing Unit – egy olyan hardverkomponens, amelyet grafikus műveletek gyorsítására terveztek (videojátékok) de adatfeldolgozási képessége miatt nagy adatbázisok feldolgozására is használják.

<sup>16</sup> TPU: Tensor Processing Unit – egy olyan speciálisan kialakított hardver, amelyet a Google fejlesztett ki a gépi tanuláshoz. Többdimenziós adattáblák feldolgozására optimalizálták.

az időt és erőforrásokat, amelyeket az adatok importálására és előkészítésére kell fordítani. Ezenkívül a fájlok közvetlen feltöltése a **Google Colab** felületre lehetővé teszi a felhasználók számára, hogy könnyen megoszthassák a munkamenetüket másokkal, ami elősegíti az együttműködést és a tudásmegosztást a csapatban vagy a közösségben.

Az elemzést a 2024-es első helyre jelentkező diákok adathalmazán végeztem, némi statisztikai elemzést is alkalmazva. Amikor elakadtam, áttértem a **Spyder**<sup>17</sup> programba, ahol egyben futtattam le a kódokat, és a hibaüzeneteket elemeztettem a **ChatGPT**<sup>18</sup> felületen, vagy kódrészek magyarázatához kértem segítséget az ingyenesen elérhető AI felületen.

### 3.3.2. Könyvtárak és adatok importálása

A Python környezetben a könyvtárak importálásával kezdtem a feladatot. Mivel a csomagok telepítése a Colab felületen nem szükséges, csak az ellenőrzésre használt **Spyder** esetében kellett a csomagok telepítésével foglalkoznom. A csomagok installálásának parancsa a **pip install [csomagnév]**.

A Pythonban a csomagok importálására használt parancs **import könyvtár\_neve**. Ezzel a könyvtárba tartozó összes modul, osztály és függvény importálásra kerül a megadott könyvtárból. A könyvtár neve utáni **as** prepozícióval lehetőség van egy új, rövid név adására, így a könyvtár meghívásakor leegyszerűsödik az arra való hivatkozás.

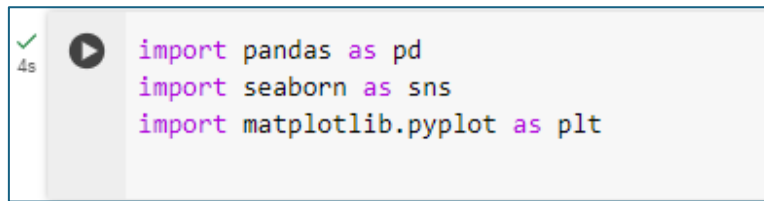
A **from könyvtár\_neve.valami import függvény** szintaxis használatakor a **könyvtár\_neve** általában egy Python modul nevét jelöli, míg a **valami** az adott modulban található almodul vagy osztályt jelöli, és a **függvény** pedig az adott almodulban vagy osztályban található függvényt vagy osztályt. A **könyvtár\_neve** után következő pont (.) jelzi, hogy a könyvtárnév alatt egy további részmodult vagy almodult hív meg a kód. Ezt a részmodul vagy almodul közvetlenül a könyvtár hierarchiájában található.

---

<sup>17</sup> Spyder: egy nyílt forráskódú Python IDE, azaz integrált fejlesztőkörnyezet

<sup>18</sup> ChatGPT: Chat Generative Pre-trained Transformer azaz Generatív előtanított transzformátor, egy olyan mesterséges intelligencia (MI) modell, amely a transzformátor architektúrára épül.

**22. ábra:** Könyvtárak importálása Google Colab felületen Python programozással



```
✓ 4s ▶ import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Az adatok beolvasása után a Colab felületre húztam a felhasználni kívánt adathalmazt. Ezzel tulajdonképpen feltöltöttem a fájlt a Google Colab virtuális környezetébe. Ez a virtuális környezet lehetővé tette, hogy a fájlt közvetlenül a Colab munkafüzetéből érjem el, használjam a kódban. Hátránya, hogy kijelentkezés és újra bejelentkezés esetén a fájlokat újra fel kellett tölteni.

### 3.3.3. DataFrame létrehozása, adattípusok lekérdezése

Ezt követően a kódban létrehozom a **DataFrame**-et, amely egy olyan adattáblázat, adathalmaz vagy adatbázis, amely elsősorban a **Pandas** könyvtárban használatos. Valójában a **DataFrame** egy kétdimenziós táblázatnak felel meg, amely sorokat és oszlopokat tartalmaz.

Az adatok betöltése után megvizsgáltam az adatok típusát, megszámláltam a sorok számát, és az elsőhelyes jelentkezők, valamint az összes jelentkező számára összegzést végeztem. Az elsőhelyes jelentkezők száma a valós jelentkezői létszámot mutatta, és ez a szám valóban helyes, ahogy az több hírportálon is megjelent, miszerint közel 121 ezer diák jelentkezett felsőoktatási intézményekbe a 2024-es szeptemberi felvételi eljárás keretén belül.



23. ábra: Adatok ellenőrzése Google Colab felületen (Forrás: <https://colab.research.google.com>)

```
[2] # Adatok betöltése
df = pd.read_excel("/content/24A_jelentkezo_kepzesenkent.xlsx")

# Adatok ellenőrzése és előzetes elemzése
print("Az adathalmaz típusai:")
print(df.dtypes)
print("Az adathalmaz sorainak száma:", len(df))
summary01 = df.describe()
total_first_correct = df['elsőhelyes jelentkező'].sum()
total_all_applications = df['összes jelentkezés'].sum()
print("Az első helyes jelentkezők összesen:", total_first_correct)
print("Az összes jelentkezés összesen:", total_all_applications)
print("Az adathalmaz első néhány sora:")
print(df.head())
```

Az adathalmaz típusai:

intezmeny	object
kar	object
fenntartó	object
képzési terület	object
meghirdetett képzés	object
képzési szint	object
munkarend	object
finanszírozási forma	object
képzés helye	object
képzés nyelve	object
elsőhelyes jelentkező	int64
összes jelentkezés	int64

dtype: object

Az adathalmaz sorainak száma: 8322

Az első helyes jelentkezők összesen: 120990

Az összes jelentkezés összesen: 443931

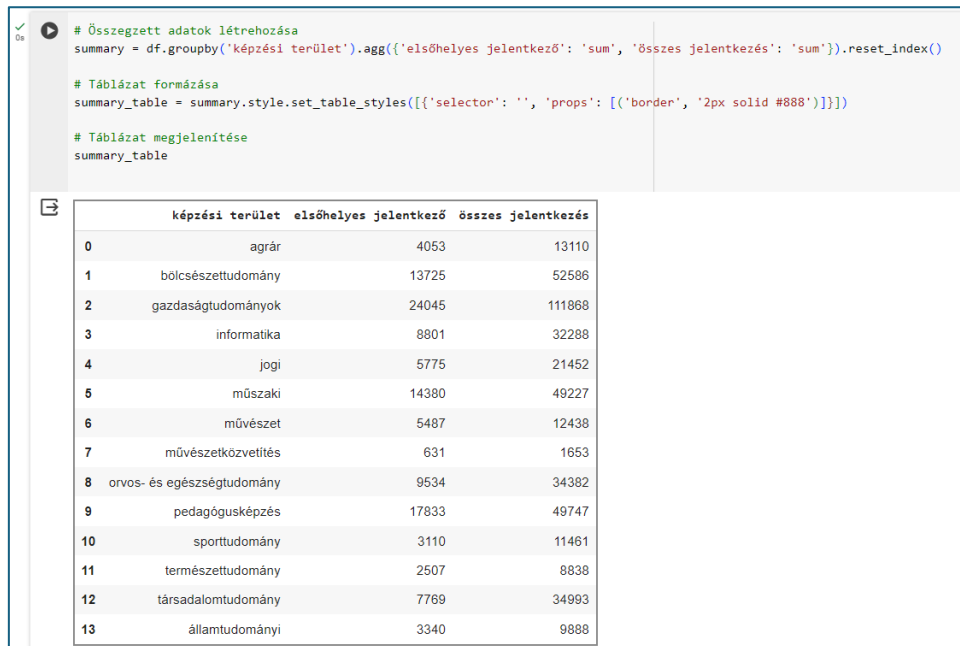
Az adathalmaz első néhány sora:

	intezmeny	kar	fenntartó	képzési terület	\
0	ANNYE	ANNYE	közalapítvány	államtudományi	
1	ANNYE	ANNYE	közalapítvány	államtudományi	
2	ANNYE	ANNYE	közalapítvány	bölcsészettudomány	
3	ANNYE	ANNYE	közalapítvány	bölcsészettudomány	
4	ANNYE	ANNYE	közalapítvány	gazdaságtudományok	

### 3.3.4. Adatok rendezése, vizualizáció elkészítése több könyvtár használatával

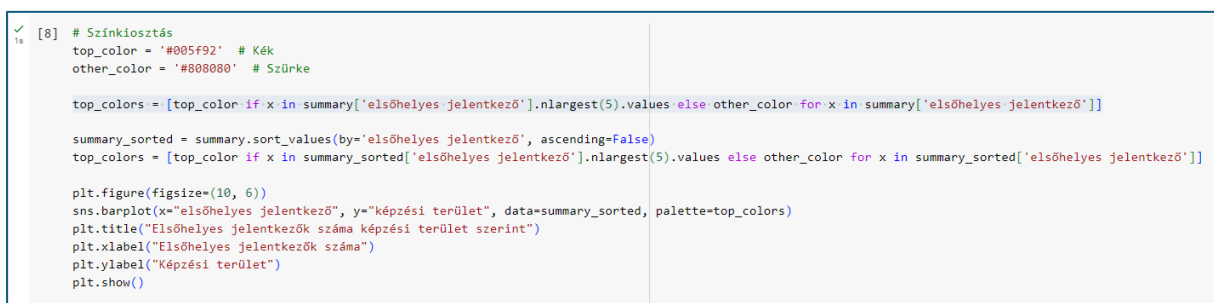
Az adatokat táblázatos formába rendeztem. A **groupby** függvény segítségével az adatokat csoportokba soroltam, és az integer típusú adatokat a **sum()** függvény segítségével összegeztem. A tábla formázására használt kódban az összegzett **DataFrame** objektumra egy megjelenítési stílust állítottam be, ahol **selector** üres string paraméterével az összes elemet formáztam, 2 pixel vastagságú sötét (hexadecimális kódja #888) keretezést használtam. Ez a vizuálisan rendezettebb megjelenítés segítette az adatok olvashatóságát és átláthatóságát.

**24. ábra:** Az összesített adatok megjelenítése képzési területenként az elsőhelyes jelentkezők száma szerint



A csoportosított adatokat használva az elsőhelyes jelentkezők alapján csökkenő sorrendbe állítottam az eredményeket a **sort\_values()** metódus segítségével, ahol az **ascending** paraméter értékének **False-ra** történő állításával a legnagyobb összeget helyeztem az első helyre. Ezt követően az elsőhelyes jelentkező adatait végig vizsgálva kiválasztottam az első öt legnagyobb értéket a **nlargest(5)** függvény segítségével. Amennyiben beleesett ebbe a kiválasztásba az érték, akkor a **top color** változóban tárolt színt rendeltem az értékhez, minden más értékhez pedig az **other color** változóárolt szín lett rendelve.

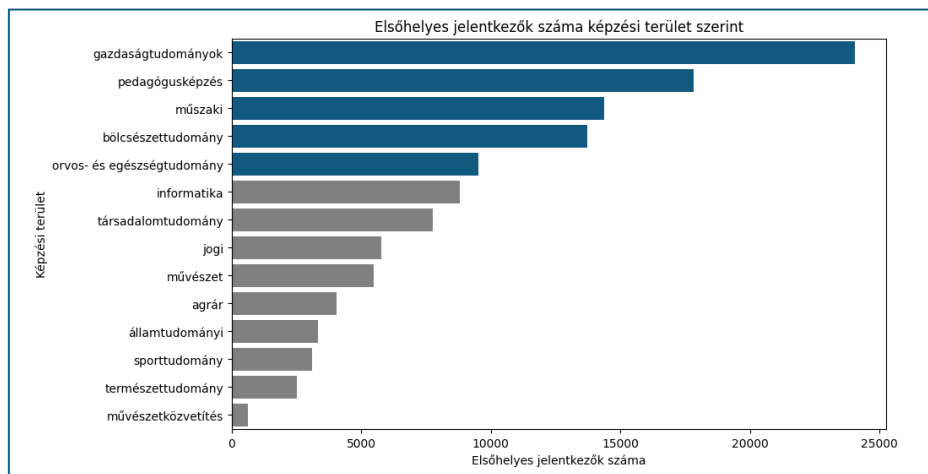
**25. ábra:** Sávdiaagram létrehozása és az első öt érték kiemelése az elsőhelyes jelentkezések tekintetében



Ezt követően a sávdiaagramot hoztam létre, amely az elsőhelyes jelentkezők számát képzési területek szerint mutatta. A **plt.figure(figsize=(10, 6))** egy új ábrát hozott létre a **matplotlib** segítségével, ahol a **figsize** paraméterrel beállítottam az ábra méretét hüvelykben. A

**sns.barplot()** kód egy sávdiaagramot rajzolt a **seaborn** könyvtár **barplot()** függvényével. Az **x** paraméter az oszlopok hosszát, a **y** paraméter pedig az oszlopok elhelyezkedését határozta meg. A **data** paraméterben megadtam az adatforrást, a **palette** paraméterben pedig a sávok színét állítottam be a **top\_colors** listából. A diagram cím megadása a **plt.title()**-el történt, a tengely feliratokat pedig a **plt.xlabel** és a **plt.ylabel()**-el állítottam be. Végül a **plt.show()** segítségével megjelenítettem a létrehozott ábrát.

**26. ábra:** A megjelenített sávdiaagram



Ezt követően a **Bokeh** könyvtárat használtam az adatok elemzésére, így betöltöttem az elemzéshez szükséges modulokat.

**27. ábra:** Bokeh könyvtárban levő almodulok meghívása

```

0s ✓ from bokeh.plotting import figure, output_file, show
    from bokeh.transform import factor_cmap
    from bokeh.palettes import Spectral10, Category20
    from bokeh.io import output_notebook
    from bokeh.transform import dodge
    from bokeh.models import ColumnDataSource, NumeralTickFormatter

```

A következő kódban először is az összegeztett DataFrame adatait növekvő sorrendbe rendeztem az összes jelentkezés szerint. Ezt az adatsort használta a sávdiaagram megjelenítéséhez a **Bokeh**.

28. ábra: Bokeh elemzés az összes jelentkezés és az elsős helyes jelentkezések tekintetében

```
✓ Os ▶ # DataFrame rendezése összes jelentkezés szerint csökkenő sorrendben
summary_sorted = summary.sort_values(by='összes jelentkezés', ascending=False)

# Bokeh figura létrehozása
p = figure(y_range=summary_sorted['képzési terület'], height=350, title="Elsős helyes jelentkezők és összes jelentkezések aránya",
          toolbar_location=None, tools="")

# Adatok forrásának létrehozása
source = ColumnDataSource(summary_sorted)

# Sávdigrammok hozzáadása
p.hbar(y='képzési terület', right='összes jelentkezés', height=0.4, color="#38ac55", legend_label="Összes jelentkezés", source=source)
p.hbar(y='képzési terület', right='elsős helyes jelentkező', height=0.4, color="#007548", legend_label="Elsős helyes jelentkező", source=source)

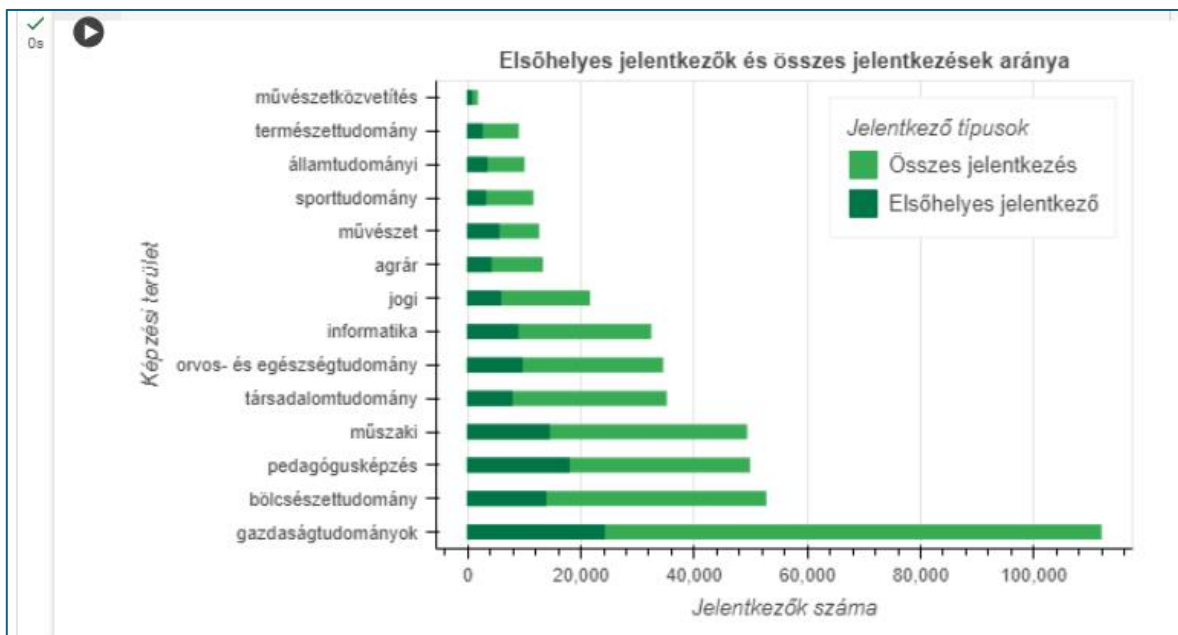
# Plot beállítások
p.ygrid.grid_line_color = None
p.xaxis.axis_label = "Jelentkezők száma"
p.yaxis.axis_label = "Képzési terület"
p.title.align = 'center'
p.legend.location = "top_right"
p.legend.title = "Jelentkező típusok"

# X tengely formázása
p.xaxis.formatter = NumeralTickFormatter(format="0,0")

# Plot megjelenítése
output_notebook()
show(p)
```

A diagram ábrázolta az első helyen jelentkezők és az összes jelentkezők számát a képzési területek szerint, két egymásra fektetett sáv segítségével, amelyek színe különböző, ezáltal könnyebb megkülönböztethető. A diagramot további beállításokkal láttam el, például a tengelyeket címkézéssel és címmel, majd a kész diagramot megjelenítettem.

29. ábra: Bokeh könyvtár felhasználásával készített sávdigram



### 3.3.5. Az adatok összefüggésének vizsgálata statisztikai módszerekkel

Ezt követően, hogy az összes jelentkezés és az elsős helyes jelentkező közötti összefüggéseket, kapcsolatokat megvizsgáljam, korrelációs analízist végeztem. Definiáltam a kiválasztott oszlopokat, majd kiválasztottam az adatokat a DataFrame-ből.

30. ábra: Korrelációs mátrix létrehozása

```
✓ 0s ▶ # Kiválasztjuk azokat az oszlopokat, amelyeket elemzés alá akarunk vetni
selected_columns = ['elsöhelyes jelentkező', 'összes jelentkezés']

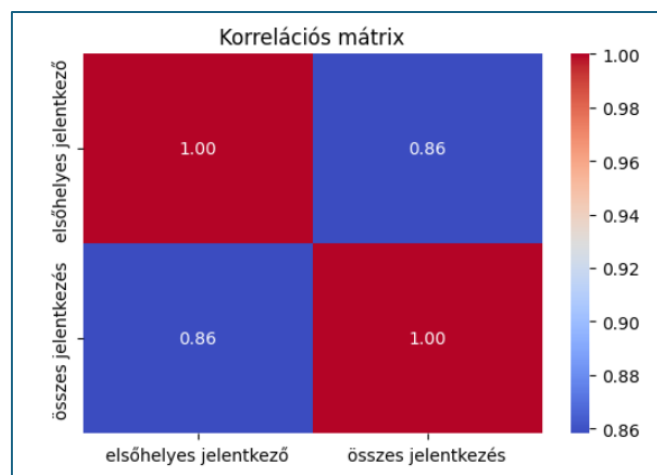
# Adatok kiválasztása
selected_data = df[selected_columns]

# Korrelációs mátrix létrehozása
correlation_matrix = selected_data.corr()

# Korrelációs mátrix megjelenítése
plt.figure(figsize=(6, 4))
sns.heatmap(corrrelation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Korrelációs mátrix')
plt.show()
```

A kód további részében egy korrelációs mátrixot hoztam létre, a **pandas** könyvtárat használva a **corr()** függvénnyel, amely megmutatta az egyes változók közötti korrelációs együtthatókat. Végül matplotlib és a seaborn könyvtárak használatával megjelenítettem a mátrixot egy hőterképben, ahol a színek és a skálázás segítségével ábrázoltam a korrelációs együtthatók értékeit. A hőterkép segítségével könnyen vizualizálható, hogy mely változók között van erősebb vagy gyengébb korreláció, és milyen irányú ez a kapcsolat.

31. ábra: korrelációs mátrix az összes jelentkezés és az elsős helyes jelentkezés arányában



Majd lineáris regresszió modellt hoztam létre egymásra illesztve az adatokat. Az elsőhelyes jelentkező és az összes jelentkezés értékeit az X és Y változókkal határoztam meg. Ezt követően hoztam létre a lineáris regressziós modellt **scikit-learn** könyvtárban található **sklearn.linear\_model** modulban levő **LinearRegression** osztály segítségével, majd az illesztést a **fit()** metódussal hajtottam végre az X és Y változókon.

A modell illesztésének eredményeként meghatároztam az illesztett egyenes paramétereit, a meredekséget és a tengelymetszetet. Ezután egy pontmegoszlást ábrázoltam (**scatter plot**), amelyen az X és Y változók értékei vannak elhelyezve. Ezt követően az illesztett egyenest is kirajoltam a **plotra**, amelyet a lineáris regressziós modell alapján számoltam ki. A kirajolt egyenes az illesztett modell által becsült összefüggést szemlélteti az elsőhelyes jelentkezők és az összes jelentkezések között. Az ábrázoláshoz a **matplotlib** könyvtárat használtam.

**32. ábra:** Lineáris regressziós egyenes egyenletének kiírása és a modell létrehozása

```
import numpy as np
from sklearn.linear_model import LinearRegression

# X és Y változók kiválasztása
X = df[['elsőhelyes jelentkező']]
Y = df['összes jelentkezés']

# Lineáris regressziós modell létrehozása és illesztése
model = LinearRegression()
model.fit(X, Y)

# Illesztett egyenes paramétereinek kinyerése
meredekség = model.coef_[0]
metszet = model.intercept_

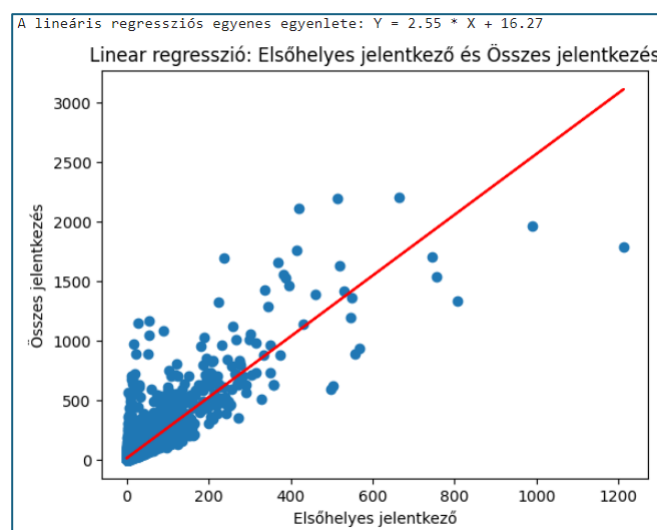
# Lineáris regressziós egyenes egyenletének megjelenítése
print(f"A lineáris regressziós egyenes egyenlete: Y = {meredekség:.2f} * X + {metszet:.2f}")

# Scatter plot létrehozása és illesztett egyenes kirajzolása
plt.scatter(X, Y)
plt.plot(X, meredekség*X + metszet, color='red')
plt.xlabel('Elsőhelyes jelentkező')
plt.ylabel('Összes jelentkezés')
plt.title('Lineár regresszió: Elsőhelyes jelentkező és Összes jelentkezés')
plt.show()
```

A **NumPy** lehetőséget biztosított az adatok elemzésében. Számos beépített funkciót kínál a tömbökön végzett műveletekhez, mint például matematikai műveletek, logikai műveletek, alakítási műveletek, rendezés, kiválasztás, alapszintű lineáris algebrai műveletek, alapszintű statisztikai műveletek, véletlen szimulációk és sok más. Ezért került a kód elején importálásra.

A létrejött lineáris regressziós egyenlet szerint a két változó között összefüggés van. A meredekség értéke azt mutatja, hogy az elsőhelyes jelentkező számának egységi növekedése az összes jelentkező értékét átlagosan 2.55 egységgel növeli. Amikor az elsőhelyes jelentkezők értéke 0, akkor az összes jelentkezés értéke 16.26, amely megmutatja, hogy az X tengely hol metszi az Y tengelyt. Tehát az egyenes egyenlete alapján megállapíthatjuk, hogy az elsőhelyes jelentkező és az összes jelentkezés között pozitív lineáris kapcsolat van.

**33. ábra:** Lineáris regressziós modell



Ezután kovariancia analízist végeztem, hogy a pozitív lineáris kapcsolatot szintén alátámasszam. Ennek az értéke 5040.21, ami azt jelzi, hogy van egy pozitív lineáris kapcsolat az elsőhelyes jelentkező és az összes jelentkezés változók között. Azaz általában, amikor az elsőhelyes jelentkező értéke nő, az összes jelentkezés értéke is nő, és fordítva. Azonban az abszolút értéke nagyon magas, ami azt jelenti, hogy ez a kapcsolat viszonylag erős.

**34. ábra.** Kovariancia kiszámítása

```
# Kovariancia számítása
covariance = df['elsőhelyes jelentkező'].cov(df['összes jelentkezés'])
print("Kovariancia az Elsőhelyes jelentkező és Összes jelentkezés között:", covariance)
```

Kovariancia az Elsőhelyes jelentkező és Összes jelentkezés között: 5040.213434672066

## 4. Összefoglalás

A szakdolgozatom célja az volt, hogy összehasonlítsak több Microsoft termék és a Python kódra épülő könyvtárak által nyújtotta lehetőségeket a vizualizáció terén. A vizsgálat során bemutattam az eszközök által biztosított funkciókat, elemeztem felhasználhatóságukat és hatékonyságukat. Megállapítottam, hogy a Microsoft eszközök kevesebb előképzettséget igényelnek és könnyebben elsajátíthatók. Megállapításom szerint a legjobb vizualizációt a Power BI felületen lehet elkészíteni, viszont az eszköz adta lehetőségek legjobb kihasználása végett szükségessé válhat egy előfizetői változatot alkalmazni. A felhasználók függvényében ez többletköltséget eredményezhet a vállalatok számára. A Python könyvtárak azonban ingyenesen elérhetőek. Eredményeként egyedibb vizualizációk hozhatóak létre, amelyek jobban felkeltik az alkalmazók figyelmét, nagyobb hatást érhetnek el.

Az adatelemzők jelentős részben az üzleti folyamatok támogatására specializálódtak. A MS Excel továbbra is rendkívül elterjedt eszköz az adatelemzés terén, felülmúlva a többi elérhető technológiai lehetőséget. Bár új, hatékony megoldások jelentek meg az adatfeldolgozásban, az MS Excel továbbra is meghatározó szereppel bír. Személyes tapasztalatok alapján még évek múltán is könnyen adaptálható, annak ellenére, hogy számos fejlesztést végeztek el rajta.

A vizuális megjelenítés, valamint a menürendszer, amelyet a Microsoft eszközök nyújtanak, véleményem szerint nem helyettesíthetőek. Azonban fontos kiemelni, hogy tanulmányaim során más lehetőségeket is megismertem és alkalmazok, nem kizárólag az említett platformot preferálva. Például az adatfeldolgozási folyamatok hatékonyságának növelése érdekében számos alkalommal használtam Python-t is.

A piac azonban elvárja, hogy egy adatelemző jártas legyen az Excelben, az SQL-ben és a BI<sup>19</sup> eszközökben. A központi eszközzé viszont az válik, amelyet a vezetők és felhasználók is használnak.

---

<sup>19</sup> BI: üzleti intelligencia



## 5. Felhasznált szakirodalom és források

- Balázs, Barbara, és mtsai. „Vizualizáció a Tudománykommunikációban.” *ELTE TTK*. 2013. [https://www.eltereader.hu/media/2014/05/Vizualizacio\\_READER.pdf](https://www.eltereader.hu/media/2014/05/Vizualizacio_READER.pdf) (hozzáférés dátuma: 2024. 03 19).
- Balogh, Nóra. *dmlab*. 2021. 09 10. <https://dmlab.hu/blog/a-legnepszerubb-adatvizualizacios-eszkozok/> (hozzáférés dátuma: 2024. 03 09).
- Deckler, Greg. *DAX Cookbook*. Birmingham, Egyesült Királyság: Packt Publishing Limited, 2020.
- Ding, David. *Transitioning to Microsoft Power Platform*. Sydney, NSW, Australia: Apress, 2023.
- Foulkes, Linda, és Warren Sparrow. *Learn Power Query*. Birmingham, United Kingdom: Packt Publishing Limited, 2020.
- Guntuku, Sharat Chandra, és Hora Shubhangi. *Interactive Data Visualization with Python*. Second Edition. Birmingham, Egyesült Királyság: Packt Publishing Limited, 2020.
- Hopkins, Wyn. *Power BI for the Excel Analyst*. Merritt Island: Holy Macro! Books, 2022.
- Horne, Ian. *Hands-On Business Intelligence with DAX*. Birmingham, Egyesült Királyság: Packt Publishing Limited, 2020.
- Iseminger, David, és mtsai. *Dokumentáció a Power BI használatba vételéhez*. 2024. 02 14. <https://learn.microsoft.com/hu-hu/power-bi/fundamentals/> (hozzáférés dátuma: 2024. 03 11).
- Jolly, Kevin. *Hands-On Data Visualization with Bokeh*. Birmingham, Egyesült Királyság: Packt Publishing Limited, 2018.
- Knight, Devin, Mitchell Pearson, Bradley Schacht, és Erin Ostrowsky. *Microsoft Power BI Quick Start Guide*. Birmingham, United Kingdom: Packt Publishing Limited, 2020.
- Microsoft. *Adatelemzési kifejezések (DAX) referenciája*. 2023. 10 20. <https://learn.microsoft.com/pdf?url=https%3A%2F%2Flearn.microsoft.com%2Fhu-hu%2Fdax%2Ftoc.json> (hozzáférés dátuma: 2024. 03 18).
- Molin, Stefanie. *Hands-On Data Analysis with Pandas*. Birmingham, Egyesült Királyság: Packt Publishing Limited, 2021.
- Olafusi, Michael. *Building Interactive Dashboards in Microsoft 365 Excel*. Birmingham, Egyesült Királyság: Packt Publishing Limited, 2024.
- Po, Laura, Nikos Bikakis, Frederico Deimoni, és George Papastefanatos. *Linked Data Visualization: Techniques, Tools, and Big Data*. Morgan & Claypool, 2020.

Puls, Ken, és Miguel Escobar. *Master Your Data with Power Query in Excel and Power BI*. Merritt Island: Holy Macro! Books, 2021.

Rajender, Kumar. *Mastering Data Analysis with Python*. United States of America: Jamba Academy, 2023.

Rougier, Nicolas P. *Scientific Visualization: Python & Matplotlib*. Lyon, France: HAL Open Science, 2021.

Schwabish, Jonathan. *Better data visualizations*. New York Chichester, West Sussex: Columbia University Press, 2021.

So, Anthony, Thomas V. Joseph, Robert John Thas, Andrew Worsle, és Samuel Dr. Asare. *The Data Science Workshop*. Second Edition kötet. Birmingham, Egyesült Királyság: Packt Publishing Limited, 2020.

### **Letöltött források:**

Központi Statisztikai Hivatal – Minden helység adata 2015. január 1-i adatbázisból

<https://www.ksh.hu/docs/hun/hnk/hnk2015.xls>

Letöltés ideje: 2024.04.05.

Legfrissebb jelentkezői adatok 2024. szeptemberben induló képzések.

[https://www.felvi.hu/felveteli/ponthatarok\\_statistikak](https://www.felvi.hu/felveteli/ponthatarok_statistikak)

Letöltés ideje: 2024.04.05.

Államilag elismert magyar felsőoktatási intézmények maximális hallgatói létszáma (kapacitás)

<https://firgraf.oh.gov.hu/tematikus-lista/magyar-foi-maximalis-hallgatoi-letszam/html>

Letöltés ideje: 2024.04.05.

Felvételi ponthatárok 2023. szeptemberben induló képzések

<https://www.felvi.hu/bin/content/vonal23a/>

Letöltés ideje: 2024.04.05.

### **Az adatelemzésbe közvetlenül behivatkozott adatbázisok:**

<https://firgraf.oh.gov.hu/felsooktatasi-kepzesek>

<https://firgraf.oh.gov.hu/intezmenyi-adatok>

[https://www.ksh.hu/stadat\\_files/okt/hu/okt0019.html](https://www.ksh.hu/stadat_files/okt/hu/okt0019.html)

[https://www.ksh.hu/stadat\\_files/nep/hu/nep0006.html](https://www.ksh.hu/stadat_files/nep/hu/nep0006.html)

### **Felhasznált segédprogram:**

<https://www.canva.com/>

## Köszönetnyilvánítás

Nagyon hálás vagyok Dr. Szalay Zsigmond Gábornak a biztatásért, amellyel a képzésre való beiratkozásra ösztönzött. Bár kezdetben nehezen szántam rá magamat arra, hogy újra iskolapadba üljek, tanár úr ösztönző szavai jelentős segítséget nyújtottak ebben a döntésben.

Lágymányosi Attilának, konzulensemnek külön köszönet jár az iránymutatásért a munka elkészülésében, valamint a további előrehaladásra buzdító és támogató szavakért.

Tóth Marianna vezetőm felé is hálás köszönetemet szeretném kifejezni a támogató hozzáállásáért, a biztató szavakért, és a dolgozatom lektorálásában nyújtott segítségért.

Rendkívül hálás vagyok kollégámnak és külső konzulensemnek, Moór-Gergely Gabriellának, akik szakmai támogatást nyújtott, segített az anyagok keresésében és az együtt gondolkozásban.

Nem feledkezhetek meg az összes oktatóról sem, akik munkájukkal hozzájárultak a tudásom fejlesztéséhez. Bár még messze vagyok attól, hogy mindent tudjak, elkötelezett vagyok amellet, hogy a megszerzett ismereteket valóban hasznosítsam a gyakorlatban, és tovább elmélyedjek azokban.

Végül pedig szeretném megköszönni családomnak, barátaimnak, hogy a tanulmányaim alatt szeretettel gondoltak rám, olykor helyettem dolgoztak, vagy imádságban hordoztak. Mindezért pedig Istené a dicsőség.

# Ábrajegyzék

1. ábra: MS Excel adatbeviteli lehetőségek	6
2. ábra: Jelenlegi Microsoft alkalmazások, amelyek rendelkeznek Power Query-vel (Forrás: Building Interactive Dashboards in Microsoft 365 Excel, Szerző: Michael Olafusi)	9
3. ábra: Power Query szerkesztő (Forrás: Ken Puls_Miguel Escobar - Master Your Data with Power Query in Excel and Power BI alapján saját feldolgozásban)	10
4. ábra: Power BI folyamatai (Forrás: David Ding - Transitioning to Microsoft Power Platform)	13
5. ábra: Power BI alkalmazások (Forrás: Dokumentáció a Power BI használatba vételéhez)	14
6. ábra: Power BI Desktop adatbetöltő felülete	15
7. ábra: Power BI Desktop elemei (Forrás: David Ding - Transitioning to Microsoft Power Platform - ábrája alapján saját vizualizáció készítése, és beszámozása Canva program használatával)	16
8. ábra: Venn-diagram a Power BI Desktop és Power BI szolgáltatás összehasonlítására (Forrás: Dokumentáció a Power BI használatba vételéhez ábrája alapján saját feldolgozásban)	18
9. ábra: Python (Forrás: <a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a> )	19
10. ábra: Pandas könyvtár (Forrás: <a href="https://pandas.pydata.org/docs/user_guide/style.html">https://pandas.pydata.org/docs/user_guide/style.html</a> )	20
11. ábra: Matplotlib (Forrás: <a href="https://matplotlib.org/stable/plot_types/arrays/index.html">https://matplotlib.org/stable/plot_types/arrays/index.html</a> )	21
13. ábra: Seaborn (Forrás: <a href="https://seaborn.pydata.org/index.html">https://seaborn.pydata.org/index.html</a> )	24
14. ábra: Adattisztítási lépések bemutatása Power Query felületen (Saját forrásból)	26
15. ábra: Power Pivotba beimportált és létrehozott összekötő táblák közötti kapcsolat háló (Saját forrás)	28
16. ábra: Power Pivot kimutatás készítő eszköztár (Saját forrásból)	28
17. ábra: MS Excel és Power Pivot vizualizáció alkalmazása (Saját forrás)	30
18. ábra: Power BI kapcsolati tábla (Saját forrás)	31
19. ábra: Elkészült háttér Canva program segítségével (Forrás: <a href="https://www.canva.com/">https://www.canva.com/</a> helyen készített saját forrás)	32
20. ábra: Power BI eszközzel készített elemzés dinamikusan változó adatokkal (Saját forrás)	33
21. ábra: Döntési fa és azzal együtt dinamikusan változó térkép Power BI felületen (Saját forrás)	34
22. ábra: Google Colab felület (Forrás: <a href="https://colab.research.google.com/">https://colab.research.google.com/</a> )	35
23. ábra: Könyvtárak importálása Google Colab felületen Python programozással	37
24. ábra: Adatok ellenőrzése Google Colab felületen (Forrás: <a href="https://colab.research.google.com">https://colab.research.google.com</a> )	38
25. ábra: Az összesített adatok megjelenítése képzési területenként az elsős helyes jelentkezők száma szerint	39
26. ábra: Sávdigram létrehozása és az első öt érték kiemelése az elsős helyes jelentkezők tekintetében	39
27. ábra: A megjelenített sávdigram	40
28. ábra: Bokeh könyvtárban levő almodulok meghívása	40
29. ábra: Bokeh elemzés az összes jelentkező és az elsős helyes jelentkezők tekintetében	41
30. ábra: Bokeh könyvtár felhasználásával készített sávdigram	41
31. ábra: Korrelációs mátrix létrehozása	42

32. ábra: korrelációs mátrix az összes jelentkezés és az elsőhelyes jelentkezés arányában	42
33. ábra: Lineáris regressziós egyenes egyenletének kiírása és a modell létrehozása	43
34. ábra: Lineáris regressziós modell	44
35. ábra. Kovariancia kiszámítása	44

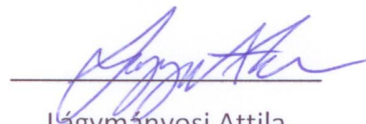
## NYILATKOZAT

Labátné Fercsik Katalin (hallgató Neptun azonosítója: S9CF04) konzulenseként nyilatkozom arról, hogy a szakdolgozatot áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A szakdolgozatot a záróvizsgán történő védeésre javaslom / nem javaslom<sup>1</sup>.

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem<sup>2</sup>

Kelt: Gödöllő 2024. év április hó 15. nap



Lágymányosi Attila

belső konzulens

---

<sup>1</sup> A megfelelő aláhúzendó.

<sup>2</sup> A megfelelő aláhúzendó.

## NYILATKOZAT

### a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Labátné Fercsik Katalin  
A Hallgató Neptun kódja: S9CF04  
A dolgozat címe: Online elérhető adatok elemzése  
A megjelenés éve: 2024  
A konzulens intézetének neve: Műszaki és Informatikai Intézet  
A konzulens tanszékének a neve: Mérnökinformatikai Tanszék

Kijelentem, hogy az általam benyújtott szakdolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, és az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlan állítottam, tudomásul veszem, hogy a záróvizsga-bizottság a záróvizsgából kizár és a záróvizsgát csak új dolgozat készítése után tehetek.

A leadott dolgozat, mely PDF dokumentum, szerkesztését nem, megtekintését és nyomtatását engedélyezem.

Tudomásul veszem, hogy az általam készített dolgozatra, mint szellemi alkotás felhasználására, hasznosítására a Magyar Agrár- és Élettudományi Egyetem mindenkori szellemitulajdon-kezelési szabályzatában megfogalmazottak érvényesek.

Tudomásul veszem, hogy dolgozatom elektronikus változata feltöltésre kerül a Magyar Agrár- és Élettudományi Egyetem könyvtári repozitori rendszerébe. Tudomásul veszem, hogy a megvédett és

- nem titkosított dolgozat a védést követően
- titkosításra engedélyezett dolgozat a benyújtásától számított 5 év eltelte után nyilvánosan elérhető és kereshető lesz az Egyetem könyvtári repozitori rendszerében.

Kelt: Gödöllő, 2024. év április hó 15. nap

Hallgató aláírása