

# **SZAKDOLGOZAT**

**Bajnóczi Andrea**

**2024**



**Magyar Agrár- és Élettudományi Egyetem**

**Szenti István Campus**

**Műszaki és Informatikai Intézet**

**Adattechnológus-adatelemző szakember szakirányú  
továbbképzési szak**

**SMART CITY RENDSZEREKBŐL SZÁRMAZÓ ADATOK ELEMZÉSE**

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

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

**Külső konzulens:** Eigler Tamás  
elektromos főmérnök

**Készítette:** **Bajnóczi Andrea (WUODCH)**

**Gödöllő**

**2024**

# Tartalomjegyzék

<b>1</b>	<b>Bevezetés és célkitűzések</b>	4
<b>2</b>	<b>Szakirodalmi áttekintés</b>	6
2.1	Fogalmak	6
2.1.1	Adat	6
2.1.2	Adattudomány	6
2.1.3	Adattárolás	7
2.1.4	Adatok megszerzése és előkészítése	7
2.1.5	Adatelemzés	8
2.1.6	Adatvizualizáció	9
2.2	Mi az a smart city, azaz okos város?	9
2.3	A legelterjedtebb smart city alrendszerek	12
2.3.1	Közlekedés	12
2.3.2	Parkolás	13
2.3.3	Közvilágítás	15
2.3.4	Nyílt wifi	17
2.3.5	Megfigyelő rendszerek	17
2.4	A világ számokban	18
2.4.1	Lakosság tekintetében	18
2.4.2	Smart city tekintetében	18
2.4.3	Intelligens forgalomirányító és parkolási rendszerek tekintetében	19
<b>3</b>	<b>Alkalmazott módszerek</b>	21
3.1	Az adatok forrása	21
3.2	Adatelemzési módszer	21
3.3	Eredmények értékelése	21
<b>4</b>	<b>Eredmények és értékelésük</b>	22
4.1	A python-kód	22
4.2	Python-kód alapján kapott eredmények listázása	36
4.2.1	Az adatok évenkénti összesítése	36
4.2.2	Az összesített adatok viszonyítása az előző évhez képest	37
4.3	Eredmények megjelenítése adatvizualizációval	38
4.3.1	Adatok évenkénti összesítésének megjelenítése diagramon	39
4.3.2	Előző évhez viszonyított éves adatok megjelenítése diagramon	40
4.3.3	Autóbalesetek száma Covid-hatás nélkül	41
<b>5</b>	<b>Következtetések és javaslatok</b>	42

5.1	Következtetések .....	42
5.2	Javaslatok .....	42
<b>6</b>	<b>Összefoglalás</b> .....	<b>43</b>
	Irodalomjegyzék .....	45

# 1 Bevezetés és célkitűzések

Világunkban egyre gyorsabban fejlődik az információtechnológia és ezzel együtt a vezeték nélküli eszközök, rendszerek hálózata is. Az okos eszközök ma már jelen vannak életünk minden pontján az otthonoktól kezdve az autókon át a termelés, a közlekedés, az egészségügy vagy akár az energiaszektor területéig. Az okos eszközök szinte bárhol elhelyezhetők, akár kültéren, akár beltéren. Sokféleségük révén rengeteg különbözőféle adatot képesek gyűjteni, mint például hőmérséklet, hang, kép, súly, energiafelhasználás, páratartalom, a lehetőségek szinte határtalanok. Ezeket az adatokat emberi beavatkozás nélkül képesek az Interneten keresztül továbbítani, gyűjteni, feldolgozni és visszacsatolásokat adni, ezáltal pedig egy teljes rendszert működtetni.

A szakdolgozatomban az okos rendszereket és azok működését mutatom be részletesen, ezen belül is elsősorban a smart city megoldásokat, mint például a közlekedés, a parkolás, a közvilágítás, a nyílt wifi és a megfigyelőrendszerek előnyös hatásait, illetve esetleges hátrányait.

Manapság már nagyon sok mindenre használják az okos kifejezést, aminek tulajdonképpen bármiféle funkciója is van. Ma már akár egy toll is lehet okos, hiszen írni lehet vele. De tulajdonképpen mitől is lesz valami okos? Egy jól megtervezett, felépített és beállított rendszer tényleg okos vagy csak végrehajtja a beprogramozott utasításokat? Joggal merülnek fel az emberekben a kérdések, hogy:

- ha egy lakásban okos fűtési rendszer van, de a tulajdonosnak kell azt bekapcsolnia és beállítania, hogy hány fokot szeretne, azaz folyamatosan foglalkoznia kell vele, akkor az a rendszer tényleg okos?
- a felhő alapú rendszerek, az IoT-technológia (Internet of Things: Internethez csatlakoztatott eszközök), a gépi tanulás és az adatfeldolgozás csak néhány éve kezdett széleskörben elterjedni a világban. Új területnek számít, ami miatt sokszor a tervezők sincsenek tisztában az okos rendszerek igényeivel. Felmerül a kérdés, hogy az oktatás képes-e alkalmazkodni az informatikai szempontból ennyire gyorsan fejlődő világhoz, képes-e megtanítani az embereket ezek tervezésére, kivitelezésére, használatára, vagy ezen a téren továbbra is az öntanulás lesz az irányadó még néhány évig, évtizedig? Nemrég dolgoztam

például egy lakóprojekten, amit elismert tervezők terveztek és sok éves tapasztalattal rendelkező kivitelezőcsapat épített fel, de a házak üzemeltetési rendszerei folyton leálltak, mert nem mindig volt Internet. Nem értettük a helyzetet, amíg rá nem jöttünk, hogy a lakók az okos dugaljakon keresztül tulajdonképpen ingyen hozzáférnek az Internethez és gyakorlatilag elszívják azt a ház rendszereitől. Így a projekt végén, plusz költséget jelentett, hogy le kellett cserélni az IT-kábeleket, hogy a problémát kiküszöböljük.

- vannak-e korlátok? Lehet-e elég és megfelelően szakképzett embert biztosítani, van-e elég alapanyag, van-e fejlesztési kapacitás vagy hajlandóság az országok, városok, vállalatok részéről a technológiák alkalmazására?
- mi az, amit érdemes egyáltalán okosítani? Például az jó-e, ha a hűtő minden nap megrendel egy doboz tejet a legközelebbi boltból. Persze, egyik oldalról egy kényelmes megoldást jelent, másik oldalról viszont felvetődik a gazdaságosság, a környezetvédelem vagy a karbonlábnyom kérdése.
- hol lehet szert tenni valós előnyökre? Miközben a smart home rendszerek előnye leginkább csak annyi, hogy kényelmesebbé teszik az emberek otthon töltött idejét, addig az energiaszektorban, távközlésben, egészségügyben hatalmas változásokat lehet elérni egy-egy okos megoldással. Például ha a mentőorvos már a mentőautóban tisztában van egy ember betegéletútjával vagy azzal, hogy milyen gyógyszereket szed, illetve azonnali adatokat tud szolgáltatni a beteg állapotára vonatkozóan a kórház részére, így mire a mentő beér és ott már egy felkészült team várja a beteget, azzal az emberi élet szempontjából értékes időt lehet nyerni.

A szakdolgozatom során célkitűzésem már megvalósult smart city rendszerek bemutatása, valamint az ezekből a rendszerekből származó adatok feldolgozási lehetőségeinek bemutatása. A rendszerek közül elsősorban az okos közlekedési rendszerből származó adatok feldolgozására kívánok fókuszálni. Állítólag a dugók 50 %-át leginkább az okozza, hogy az autósok parkolóhelyet keresve körbe-körbe keringenek, ezzel akadályozva a többi autós haladását. Egy 172.000 fős spanyol városban, Santanderben, a városvezetés állítása szerint 80 %-kal csökkentek a dugók az okos parkolási rendszer bevezetését követően. Vajon tényleg ekkora csodát tehet egy okos parkolási rendszer?

## 2 Szakirodalmi áttekintés

### 2.1 Fogalmak

#### 2.1.1 Adat

„Valaminek a megmagyarázására, megvilágítására, jellemzésére vagy kiegészítésére közölt tény, részlet, adalék.

Valamely dologra vagy tárgykörre vonatkozó, ismert, írásban nyilvántartott tény.

Jogtudományban olyan tények ismerete, tudása, amelyek valamely következtetéshez, valaminek a megoldásához szükségesek, amelyek valamely eredmény, döntés, határozat létrejöttét segítik; valamely tény, mint tényszerű bizonyíték.

Tudományos megközelítésben valamely kísérlet, mérés, vizsgálat eredményeképpen megállapított tény, tétel.” **(Bárczi & Ország, 1959-1962)**

„A számítástechnikában adatnak (angol nyelven data) nevezzük a számokkal leírható dolgokat, melyek számítástechnikai eszközökkel rögzíthetők, feldolgozhatók és megjeleníthetők. Az adat nagyon tág fogalom: gyakorlatilag bármilyen jel potenciálisan adatnak tekinthető. ... Az adatoknak önmagukban nincs jelentésük. (Például annak az adatnak, hogy 30 nincs önmagában jelentése.) Az adatok az értelmezéstől, azok feldolgozásának módjától, alkalmazásuktól nyernek értelmet, és válhatnak információvá, hasznos adatokká.” **(Wikipedia, 2023.)**”

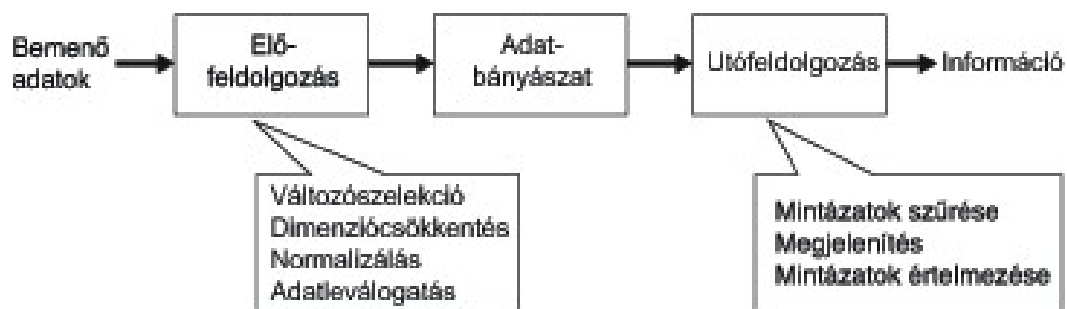
#### 2.1.2 Adattudomány

Az adattudomány vagy más néven adatbányászat olyan tudományágak egyesülésével jött létre, mint a statisztikához kapcsolódó mintavételezés, becslés, hipotézisvizsgálat, valamint a mesterséges intelligenciához, alakfelismeréshez és gépi tanuláshoz kapcsolódó keresőalgoritmusok, modellezési módszerek és tanulóeljárások, de magába foglalja az optimalizálást, az evolúciós számítástudományt, az információelméletet, a jelfeldolgozást, az adatvizualizációt és az információkeresést is.

Az adatbányászat az adatbázisokban végzett tudásfeltárás teljes folyamata az adatok előfeldolgozásától az eredmények utófeldolgozásáig, amelynek célja a nyers adatok hasznos információvá történő átalakítása. **(Tan, et al., 2006)**

### 1. ábra: Adatbányászat folyamata

(Forrás: Bevezetés az adatbányászatba (Tan, et al., 2006))



### 2.1.3 Adattárolás

Az adatok tárolása strukturált vagy stukturálatlan rendszerekben történhet. A strukturált adatok általában mennyiségi jellegűek, oszlopokba és sorokba vannak rendezve, így egyszerű keresést és feldolgozást tesznek lehetővé. Tárolásuk elsősorban táblázatokban (Microsoft Excel) vagy nagyobb mennyiségű adat esetén relációs adatbázisokban (SQL) történik. A strukturálatlan adatok olyan információk, amelyek semmilyen módon nem rendszerezhetők, így elsősorban minőségi adatelemzésre alkalmasak. A strukturálatlan adatok formátuma a Word-dokumentumoktól, .csv-fájloktól, .json-fájloktól, képektől és PDF-fájloktól kezdve a hang- és videofájlokig széles körben változhat. Ezek tárolása nemrelációs adatbázisokban és adattavakban történik. **(Microsoft, 2024)**

### 2.1.4 Adatok megszerzése és előkészítése

Az adatbányászat során az adatok elsődleges forrásai lehetnek például a kód- és felhőtárhelyek, a nyílt forráskódú adatbázisok, az adattárházak, az adattavak.

Az előfeldolgozás célja, hogy a nyers adatokat a vizsgálati céloknak megfelelő formátumba alakítsa át, amely magába foglalja a több forrásból származó adatok egyesítését, tisztítását, a kiugró adatok és az ismétlődések eltávolítását, a hiányzó értékek



kezelését (például a sor vagy oszlop törlésével vagy átlagos érték beírásával). Az adatfeldolgozás folyamatában ez a lépés a legidőigényesebb fázis. **(Tan, et al., 2006)**

### 2.1.5 Adatelemzés

Az adatelemzés az adatok gyűjtésére, feltárására, a vizsgálatok lefolytatására, az eredmények értelmezésére, a következtetések levonására és a döntéshozatal elősegítésére szolgál.

Az adatelemzés számos tevékenységet fed le, amelyek mindegyike saját fókusszal és célokkal rendelkezik. Ezek a tevékenységek leíró, diagnosztikai, előrejelző, előíró és kognitív elemzéseként vannak kategorizálva, amelyek külön-külön az alábbi kérdésekre adnak választ:

- leíró: mi történt?
- diagnosztikai: miért történt?
- előrejelző: mi fog történni?
- előíró: milyen műveleteket kell végrehajtani a fejlődés érdekében?
- kognitív: hogyan oldható meg a probléma a legjobban?

Az adatelemzés egy állandóan ismétlődő körforgás, amelynek lépései:

- összegyűjteni a kérdéseket, amelyekre választ várunk,
- adatok kinyerése, átalakítása és betöltése,
- adattisztítás,
- adatelemzés,
- eredmények közlése például jelentésben vagy adatvizualizációval,
- döntéshozatal,
- visszacsatolás és újabb kérdések felmerülése esetén az elemzés és a megoldások optimalizálása.

**(Microsoft, 2024)**

Az adatok elemzésének célja a vizsgált adatok sajátos jellemzőinek feltárása és megértése. Ennek legfőbb lépése a megfelelő modellezési módszer kiválasztása és adatokra illesztése. A különböző modellezési eljárások különböző célokat szolgálnak:

- a lineáris regresszió célja az előrejelzés, azaz hogy bizonyos feltételek fennállása esetén mi várható;

- a logisztikus regresszió a lineáris regresszió általánosítása, amelynek célja az osztálytagság valószínűségének előrejelzése, azaz egy bizonyos esemény bekövetkezésének valószínűségét becsüli meg;
- a multinominális és a Gauss-féle naiv-Bayes algoritmusok célja az osztályozás, azaz adatok előre definiált osztályokba sorolása;
- a K-közép módszer, azaz klaszterezés célja a kapcsolatok feltárása.

Az elemzés jósága nagyban függ az adatok minőségétől és mennyiségétől, a minél pontosabb előrejelzéshez több ezer adatra van szükség.

### 2.1.6 Adatvizualizáció

Az adatvizualizáció az adatelemzés eredményeinek rajzos vagy táblázatos megjelenítése mindenki számára könnyen értelmezhető formában. Az emberi agy alapvetően képes nagy mennyiségű képi információ gyors befogadására, feldolgozására, mintázatok felfedezésére és következtetések levonására, amelyet elősegít az adatelemzés eredményeinek különböző Power BI jelentések, gráfok és diagramok formájában történő megjelenítése. Ezen túlmenően egy szakember képes a saját szakterületéhez kapcsolódó lényeges és lényegtelen minták gyors kiszűrésére is, így teljes figyelmével a fontos elemekre tud összpontosítani. **(Tan, et al., 2006)**

## 2.2 Mi az a smart city, azaz okos város?

Az Európai Bizottság meghatározása szerint az okos város túlmutat a digitális technológiák használatán a jobb erőforrás-felhasználás és a kevesebb károsanyag-kibocsátás érdekében. Intelligensebb városi közlekedési hálózatokat, korszerűsített vízellátást és hulladéklerakó létesítményeket, valamint az épületek megvilágításának és fűtésének hatékonyabb módjait jelenti. Ez egyúttal interaktívabb és reagálóbb városvezetést, biztonságosabb köztereket és az idősödő lakosság igényeinek kielégítését is jelenti. **(Európai Bizottság, 2024)**

## 2. ábra: Okos város-konceptió

(Forrás: *Smart City Strategy, Foundation of Every Smart City (Isarsoft GmbH, 2023)*)



A különböző alrendszerek saját igényekre szabva, tetszés szerint kapcsolhatók egymáshoz, ezért nincs két egyforma okos város a világon. Közös jellemző azonban minden esetben, hogy az egyes alrendszerek infokommunikációs megoldásokkal kapcsolódnak egymáshoz, valamint kiegészítik, segítik egymás működését, amelynek végső célja, hogy jobbá tegyék a település lakosainak életminőségét.

A jól működő okos városok esetében:

- „A technológia alkalmazása beivódott a mindennapi életbe.
- A város által nyújtott szolgáltatások összekapcsoltak.
- Hatékony az erőforrás felhasználás, energiahatékonyságra törekszik.
- A folyamatos adatgyűjtés – és feldolgozás által valós időben képes a felmerülő problémák megoldására.
- Az okos város vezetése épít a lakosok észrevételeire és bevonja őket.
- Az adatok, információk áramlásának köszönhetően a pénzügyi források hatékonyabban menedzselhetők, így a városok több pénzt spórolhatnak.” (Szalai, 2018.)

Egy komplex okos város létrehozásának alapja a különböző smart city megoldások egymásba és egymáshoz integrálása, amely jelenleg még több fronton kihívásokba ütközik, így olyan világszintű fejlesztéseket tesz szükségessé, mint például:

- a generációs Internet-technológiákra (FI/ NGI) alapozott többcélú eszközök és szoftvermegoldások,
- a szenzorok fejlesztése és hálózatba kapcsolása,
- az Internet of Things (IoT) megoldások,
- kiberfizikai rendszerek (CPS) fejlesztése,
- az adattudományi módszerek (adatelemzés, adatbányászat, Big Data) felhasználása,
- a térinformatika (GIS),
- a felhőszolgáltatások (cloud computing & networking) fejlesztése,
- a nagymennyiségű adatra specializálódott elosztó keretrendszerek (például Hadoop) fejlesztése,
- a mesterséges intelligencia és önálló tanulási metódusok fejlesztése,
- a kiterjesztett és virtuálisvalóság-technológia (AR/ VR) lehetőségeinek felhasználása,
- az Internetbiztonság. (Sallai, et al., 2018)

### 3. ábra: Okos város alrendszerek

(Forrás: Az okos város koncepció és az Internet következő generációja. Okos város – okos közigazgatás technológiák (Sallai, 2018))



## 2.3 A legelterjedtebb smart city alrendszerek

### 2.3.1 Közlekedés

Az intelligens közlekedési és forgalomirányítási rendszerek integrált megközelítést biztosítanak a torlódások minimalizálására és a városi közlekedés biztonságának növelésére.

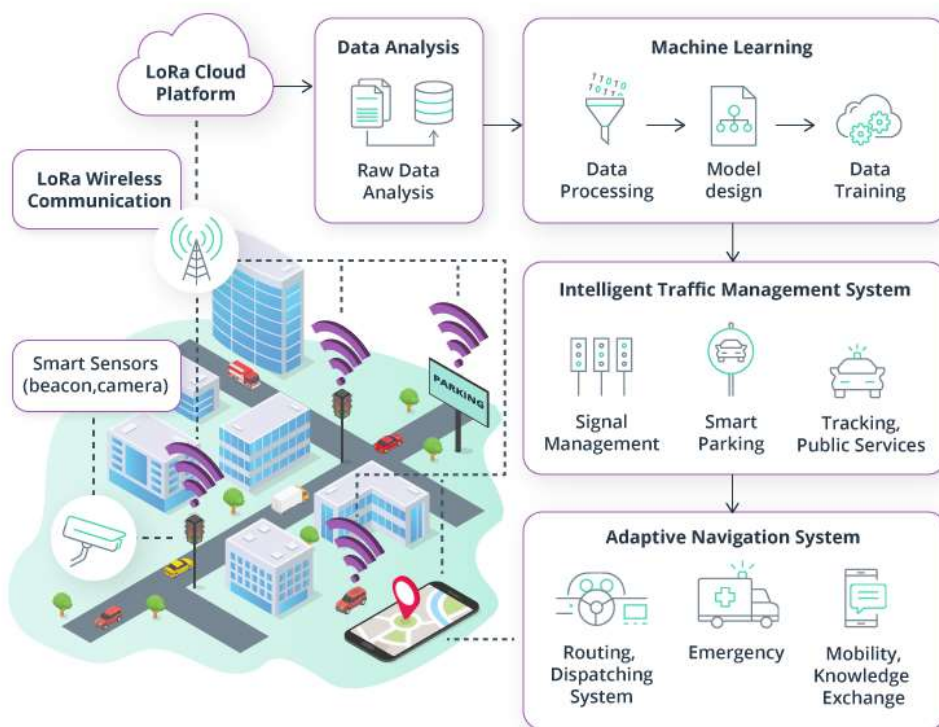
Az intelligens közlekedési és forgalomirányítási rendszerek főbb előnyei:

- a várható forgalom előrejelzése,
- fokozott biztonság, a balesetek számának csökkenése,
- a balesetek okozta anyagi károk miatti költségcsökkenés,
- vészhelyzetek esetén csökken a reagálási idő a leggyorsabb útvonalak kiválasztásával,
- az útvonalak optimalizálásával és a parkolási idő csökkentésével csökken a károsanyag-kibocsátás.

**(Haiston, 2023.)**

#### 4. ábra: LoRa és gépi tanulás alapú intelligens forgalomirányító rendszer vázlata

(Forrás: *Intelligent Traffic Management (Zavadko, 2022.)*)



## 1. táblázat: intelligens forgalomirányítási rendszerek kulcselemei

(Forrás: *Intelligent Traffic Management (Zavadko, 2022.)*)

Hardver	Szoftver
IoT szenzorok: <ul style="list-style-type: none"><li>- hőmérsékletérzékelők</li><li>- levegőminőség érzékelők</li><li>- adatgyűjtők</li></ul>	Felhő – és edge tárhely: <ul style="list-style-type: none"><li>- felhő alapú közlekedésellenőrző-rendszer</li><li>- adatgyűjtő platform vagy adattó</li><li>- térinformatika (GIS)</li></ul>
CCTV kamerák	Közlekedési appok támogatása
Közlekedési lámpák	Big data és prediktív analízis
Fizetőkapuk és fizetős útvonalak ellenőrző rendszerei	Gépi tanulás: <ul style="list-style-type: none"><li>- megerősítő tanulás</li><li>- optikai karakterfelismerés (OCR)</li><li>- számítógépes megjelenítés</li></ul>
Chipek az edge csomópontokban a gyorsabb adatátvitelért	Helyalapú szolgáltatások

### 2.3.2 Parkolás

Az okos parkolási rendszerekkel előre tervezhetővé válik a parkolás és kiiktatja az emberek életéből a szerencsén múló parkolóhely keresést. Alapesetben egy autós korán elindul, az ismert utakon megy végig, majd a célhoz érve, ha nem talál parkolóhelyet, akkor elkezd körbe-körbe autózni a közeli utcákban, egyre távolodva az eredeti célpontjától, amíg szabad helyet nem talál. Ugyanakkor intelligens parkolási rendszer használatával már az indulás előtt ellenőrizheti a szabad parkolóhelyek számát és akár le is foglalhatja a számára megfelelőt, amivel időt és üzemanyagköltséget takarít meg, csökkenti a felesleges CO<sub>2</sub> kibocsátást és a forgalmi dugók kialakulásának esélyét is. **(Cleverciti Systems GmbH, 2021)**

Az intelligens parkolási rendszerek valós idejű információkat biztosítanak a parkolási lehetőségekről, a szabad parkolóhelyekre irányítják a járművezetőket és általánosságban jobbá teszik a parkolási élményt. A rendszer részei:

- szenzorok és detektorok, amelyek a parkolóban érzékelik és nyomon követik a járművek mozgását és helyzetét, folyamatosan adatokat gyűjtenek és továbbítanak a központi egység részére;
- adatelemzés, melynek célja a szabad parkolóhelyek számának és helyzetének megállapítása;
- kommunikációs csatornák, amelyek összekapcsolják egyrészt az érzékelőket és a központi egységet, másrészt mobilalkalmazásokon vagy webhelyeken keresztül informálják a járművezetőket a szabad helyekről, amely felületek többnyire fizetési és foglalási lehetőségeket is biztosítanak a számukra. **(Sensors Dynamics, 2023.)**

Az intelligens parkolási rendszereknek négy elterjedt típusa van:

- padlószenzoros, amelynél az érzékelőt a padlóba építik, így amikor egy autó beáll az adott parkolóhelyre, az érzékelő radartechnológia segítségével foglalként jelzi azt a központi egység felé. A szabad parkolóhelyeket általában zöld, a foglaltakat piros fényvel jelzi.
- számláló, amely az épületbe behajtáskor parkolójegy váltását vagy kihajtáskor a kifizetett parkolójegy bemutatását követően nyitja fel a sorompót és az azon áthaladó járműveket számolja, valamint az épület kapubehajtójánál lévő táblán jelzi, hogy hány darab szabad parkolóhely van az épületben. Ennek hátránya, hogy csupán a darabszámot jelezni, de a szabad parkolóhelyek pontos beazonosítására nem alkalmas.
- mennyezetszenzoros, amelynek technológiája megegyezik a padlószenzorossal, annyi különbséggel, hogy az érzékelők és a jelzőlámpák nem padlószinten, hanem a födém alatt helyezkednek el, így messziről is jól láthatók.
- kamerás, amely elsősorban nyílt területeken, például utcán használatos, mivel egy kamera több parkolóhely foglaltságát is nyomon tudja követni, így nem kell minden parkolóhelyre érzékelőt telepíteni. **(Cleverciti Systems GmbH, 2021)**



## 5. ábra: Padlószenzoros parkolási rendszer

(Forrás: *What is a smart parking and how does it work (Cleverciti Systems GmbH, 2021)*)



### 2.3.3 Közvilágítás

Az intelligens közvilágítási rendszerek két szempontból is előnyösek. Egyrészt jelentős költségcsökkentés érhető el az energiafelhasználás kapcsán, másrészt pedig számos különféle érzékelővel szerelhetők fel, amely révén a közbiztonság is nagymértékben javítható.

Az intelligens közvilágítás rendszerek automatikusan szabályozzák az utcai lámpák fényerejét, külön vezérlési lehetőséget biztosítanak minden egyes lámpatestre vonatkozóan, valós időben rögzítik és továbbítják a közvilágítás állapotára vonatkozó adatokat a központba, így lehetővé teszik a megelőző karbantartást is. Különböző érzékelőkkel együtt további energia- és költségmegtakarítások érhetőek el, így például fényérzékelővel és mozgásérzékelővel felszerelve a fényerőt automatikusan tudja szabályozni minden egyes kandeláber az éppen szükséges szintnek megfelelően. A hagyományos izzókról a LED izzókra való átállással egy város a közvilágításból fakadó



energiaköltségei 60-80 %-át tudja megspórolni, ami a fent említett érzékelőkkel kiegészítve még további akár 20 %-os megtakarítást jelenthet. **(Tvilight Projects BV, 2013)**

Kihasználva a közvilágítási rendszer azon adottságát, hogy a kandeláberek mindenhol jelen vannak, így további érzékelők felszerelésével a közbiztonság, a parkolási rendszer és a forgalomirányítási rendszer javítására is pozitív hatással lehetnek:

- hangérzékelők révén alkalmassá válnak például emberi sikolyok vagy lövések azonosítására és riaszthatják a megfelelő hatóságokat;
- gyorsulásmérők révén képesek érzékelni az ütközéseket vagy az oszlop dőlését, így riaszthatják a megfelelő hatóságokat;
- szeizmikus szenzorok révén képesek azonosítani a földrengéseket és riasztani a lakosságot, valamint a hatóságokat;
- szélérősségmérő, páratartalommérő, UV-erősségmérő, esőérezékelő szenzorok révén képesek hang- és fényjelzésekkel figyelmeztetni a lakosságot és a hatóságokat;
- parkolószenzorok révén képesek azonosítani az üres parkolóhelyeket, így összekapcsolhatók az intelligens parkolási rendszerekkel is;
- egyéb szenzorokkal és kamerákkal felszerelve alkalmassá tehetők például az illegálisan parkoló járművek azonosítására, a forgalom folyamatos nyomon követésére, torlódások azonosítására, sebességmérésre, nyílt wifi biztosítására, elektromos eszközök töltésére, a lehetőségek tárháza szinte végtelen. **(Lewis, 2019.)**

Összefoglalva az intelligens közvilágítás megvalósításának előnyei:

- a jelentős energia- és költségcsökkentés,
- a karbantartási költségek csökkentése, mivel az utólagos karbantartás helyett lehetővé válik a megelőző karbantartás,
- a lámpatestek és izzók élettartamának növekedése,
- a közvilágítás teljeskörű ellenőrzése,
- a CO<sub>2</sub> kibocsátás és ezzel együtt a karbonlábnyom csökkentése,
- a fényszennyezés csökkentése,
- a közbiztonság és ezzel együtt az életminőség javítása,

- a valós adatokra és forgalmi mintákra alapozott tervezés lehetősége. **(Lewis, 2019.)**

#### 2.3.4 Nyílt wifi

A nyílt wifi vagy free wifi rendszerek ingyenes hozzáférést biztosítanak az Internethez mindenki részére. Az okos városok működését az egymásba integrált rendszerek, a valós idejű adatfeldolgozás és a folyamatos információáramlás biztosítja. A lakosok a megfelelő digitális platformok révén (webes applikációk, honlapok) bárhol, bármikor hozzáférhetnek az adatokhoz és megszerezhetik a számukra adott pillanatban releváns információt, legyen az például egy útvonal, egy nyitva tartás, egy menetrend vagy szabad parkolóhely foglalása. Ehhez általában két dologra van szükségük, Internetre és mobiltelefonra. Ingyenes, nyílt wifi-hálózatok kiépítésével biztosítani lehet az állandó kapcsolatot a várost működtető rendszerek és az azokat használó emberek között.

A nyílt wifi-hálózatok kiépítése minden okos város számára nagy kihívást jelent, mivel a szolgáltatást a felhasználók részére ingyenesen kell biztosítani, ugyanakkor egy teljes várost lefedő, mindenhol megfelelő kapacitást és jelerősséget biztosító, valamint adatbiztonságot garantáló rendszer kiépítése igencsak költséges, erőforrás- és emberigényes feladat. **(Nimbus Lighting Group Ltd., 2021)**

#### 2.3.5 Megfigyelő rendszerek

A korábbiakban ismertetettek szerint a kamerarendszerek tökéletes kiegészítői lehetnek az intelligens forgalomirányító, parkolási és közvilágítási rendszereknek, valamint hatékonyan javíthatják a közbiztonságot is. A kamerarendszerekből nap, mint nap rengeteg adat keletkezik, amelyet emberi erőforrásokkal képtelenség lenne átnézni, ezért ezen a téren egyre nagyobb szerepet kap a valós idejű adatfeldolgozás, adatelemzés és a mesterséges intelligencia. A videotartalom-elemzésre kifejlesztett szoftverek a felvételeken található objektumok azonosításával, osztályozásával és indexelésével (például autók, buszok, teherautók, férfiak, nők, gyerekek, állatok) gyorsan kereshetővé, használhatóvá és számszerűsíthetővé teszik az adatokat. A rendszerek előre beprogramozott igények szerint képesek meghatározott adatok gyűjtésére (például demográfiai összetétel) vagy akár

arcfelismerésre (gyanúsított vagy eltűnt személy), rendszámfelismerésre (útdíjfizetés), szabálytalanság (tilosban parkoló autó) vagy bűncselekmény felismerésére és ezekkel kapcsolatos automatikus riasztásokra, forgalom- vagy utasszámlálásra, de akár járványügyi szabályok betartásának monitorozására is (COVID járvány idején bevezetett maszkviselés és távolságtartási szabályok), valamint elősegítik a várostervezők munkáját is. **(Sheikh, 2023.)**

## 2.4 A világ számokban

### 2.4.1 Lakosság tekintetében

55 %: a világ lakosságának ekkora része él városokban az ENSZ Gazdasági és Szociális Tanácsának 2018-as felmérése szerint. **(ENSZ Gazdasági és Szociális Tanács, 2018.)**

68 %: a világ lakosságának ekkora része fog városokban élni 2050-re az ENSZ Gazdasági és Szociális Tanácsa által készített, a világ urbanizációs kilátásainak 2018-as felülvizsgálata alapján. **(ENSZ Gazdasági és Szociális Tanács, 2018.)**

9 darab: ennyi megaváros van a világban, amelyek a világ, jelenleg legnépesebb városainak minősülnek. Ezek Tokió 37 millió lakossal, Új-Delhi 29 millió lakossal, Sanghai 26 millió lakossal, Mexikóváros és Sao Paulo 22 millió lakossal, Kairó, Mumbai, Peking és Daka 20 millió lakossal. **(ENSZ Gazdasági és Szociális Tanács, 2018.)**

43 darab: ennyi megaváros lesz az előrejelzések szerint 2030-ra a világban. Tokió vezető szerepét várhatóan átveszi majd Új-Delhi és a világ minden 8. embere valamelyik megapoliszban fog élni. **(ENSZ Gazdasági és Szociális Tanács, 2018.)**

### 2.4.2 Smart city tekintetében

A fenti előrejelzések alapján látható, hogy néhány éven belül az urbanizáció egyaránt jelentős kihívás elé állítja a városokat, a várostervezőket és a kormányokat is. A városlakók számára a minél jobb életminőség elérése érdekében a társadalmi, környezeti és gazdasági akadályok leküzdésének egyik módja lehet az intelligens várostechológiák alkalmazása, amelyek erőforrás-takarékos módon kapcsolják össze a városokat, az embereket és a technológiát.

141 darab: világszerte körülbelül ennyi, okos városként számon tartott város létezik jelenleg. **(IMD Smart City Observatory, 2023)**

20 legokosabb város: az IMD 2023. évi Smart City Indexe alapján a világ legfejlettebb okos városai Zürich, Oslo, Canberra, Koppenhága, Lausanne, London, Szingapúr, Helsinki, Genf, Stockholm, Hamburg, Peking, Abu-Dhabi, Prága, Amszterdam, Szöul, Dubai, Sydney és Hongkong. A listán Budapest a 87. helyet foglalja el, miközben 2019-ben még a 64. helyen állt. **(IMD Smart City Observatory, 2023)**

### 2.4.3 Intelligens forgalomirányító és parkolási rendszerek tekintetében

36 óra: A Juniper Research 2022. május 16-án közzétett előrejelzése szerint az intelligens forgalomirányítási rendszerek révén 2027-ig évi 36 órával csökken autósoként a forgalomban töltött átlagos vezetési idő. **(Juniper Research, 2022.)**

205 millió tonna: az intelligens forgalomirányítási rendszerek világszinten, potenciálisan ennyivel csökkenthetik a CO<sub>2</sub>-kibocsátást 2027-ig, így jelentős szerepet játszanak a karbonlábnyom csökkentésében. **(Juniper Research, 2022.)**

41 %: ennyivel fog emelkedni 2022 és 2027 között az intelligens forgalomirányítási rendszereknek köszönhetően megtakarított CO<sub>2</sub>-kibocsátás mennyisége, amely 2022-ben 145,7 millió tonnát tett ki. **(Juniper Research, 2022.)**

4 % és 9 %: az intelligens forgalomirányítási rendszerek, azon belül a csúcsidőben a tömegközlekedést előnyben részesítő dinamikus programozású közlekedési lámpák révén Koppenhágában 4 %-kal nőtt a személyautók és 9 %-kal a buszok átlagsebessége, így ugyanennyivel csökkent a gépjárművek forgalomban töltött ideje és ezzel együtt a CO<sub>2</sub> kibocsátás is. **(Zavadko, 2022.)**

15-74 %: a forgalomban résztvevő autósok minimum 15 %-át, de csúcsforgalomban akár 74 %-át az éppen parkolóhelyet keresve, körbe-körbe autózók teszik ki, ezzel akadályozva vagy lassítva a forgalmat. **(Zavadko, 2022.)**

17-33 %: az intelligens forgalomirányító rendszereknek köszönhetően Göteborgban és Rómában 20 %-kal, Milánóban 17 %-kal, Londonban 30 %-kal, Stockholmban pedig 33 %-kal csökkent az utazási idő és ezzel együtt a forgalmi dugók aránya. **(Zavadko, 2022.)**

Több, mint 50%: a balesetek ekkora hányada következik be városok útkereszteződéseiben, amelyek áldozatai általában gyalogosok vagy kerékpárosok. **(Zavadko, 2022.)**

## 3 Alkalmazott módszerek

### 3.1 Az adatok forrása

Az adatok az Európai Unió, European data nevű, [www.data.europa.eu](http://www.data.europa.eu) webcímen elérhető, hivatalos honlapjáról származnak. **(Európai Unió, dátum nélk.)** Ugyanezeket az adatokat megtaláltam Barcelona város hivatalos honlapján is a <https://opendata-ajuntament.barcelona.cat> webcímen. A letöltött adatok Barcelona, mint okos közlekedési rendszerrel rendelkező város vizsgálatát célzó adatok a 2016-2023 közötti időszakra vonatkozóan az alábbiak szerinti csoportosításban:

- népesség,
- gépjárművek száma,
- autóbalesetek száma.

A feldolgozandó adatállomány összesen 22 fájlt és 101.776 sornyi adatot foglal magába.

### 3.2 Adatelemzési módszer

Az adatokat Microsoft Visual Studio Code szoftverrel és python kódnyelven megírt adatelemző programmal fogom elemezni.

A vizsgálat célja annak bizonyítása, hogy az okos közlekedési rendszer pozitív hatást gyakorol a közlekedésre, így csökkenthető a dugók mérete, a dugókban töltött idő hossza, valamint a közlekedési balesetek száma, ezáltal elérhető a közlekedés folytonosságának biztosítása és a városlakók életminőségének javítása.

### 3.3 Eredmények értékelése

A kapott eredményeket értékelni fogom mind szövegesen, mind pedig adatvizualizációs eszközökkel.

## 4 Eredmények és értékelésük

### 4.1 A python-kód

```
#      ---- FILE READER ----

import csv
import sys

def read_file(file_name):
    csv_list = []

    try:
        file = open(file_name, encoding='Latin1')
        try:
            file_content = csv.reader(file)

            for row in file_content:
                csv_list.append(row)
        finally:
            file.close()
    except OSError:
        sys.exit('Hiba a fájl kezelése közben!')
    else:
        return csv_list

#      ---- READ CONTENT FROM CSV ----

database_car_number_2016 = read_file('car_number_2016.csv')
database_car_number_2017 = read_file('car_number_2017.csv')
database_car_number_2018 = read_file('car_number_2018.csv')
database_car_number_2019 = read_file('car_number_2019.csv')
database_car_number_2020 = read_file('car_number_2020.csv')
database_car_number_2021 = read_file('car_number_2021.csv')

database_population_2016 = read_file('2016_pad_mdbas_sexe.csv')
database_population_2017 = read_file('2017_pad_mdbas_sexe.csv')
database_population_2018 = read_file('2018_pad_mdbas_sexe.csv')
database_population_2019 = read_file('2019_pad_mdbas_sexe.csv')
database_population_2020 = read_file('2020_pad_mdbas_sexe.csv')
database_population_2021 = read_file('2021_pad_mdbas_sexe.csv')
database_population_2022 = read_file('2022_pad_mdbas_sexe.csv')
database_population_2023 = read_file('2023_pad_mdbas_sexe.csv')

database_car_accidents_2016 =
read_file('2016_accidents_causa_conductor_gu_bcn_.csv')
```

```
database_car_accidents_2017 =
read_file('2017_accidents_causa_conductor_gu_bcn_.csv')
database_car_accidents_2018 =
read_file('2018_accidents_causa_conductor_gu_bcn_.csv')
database_car_accidents_2019 =
read_file('2019_accidents_causa_conductor_gu_bcn_.csv')
database_car_accidents_2020 =
read_file('2020_accidents_causa_conductor_gu_bcn_.csv')
database_car_accidents_2021 =
read_file('2021_accidents_causa_conductor_gu_bcn_.csv')
database_car_accidents_2022 =
read_file('2022_accidents_causa_conductor_gu_bcn_.csv')
database_car_accidents_2023 =
read_file('2023_accidents_causa_conductor_gu_bcn_.csv')

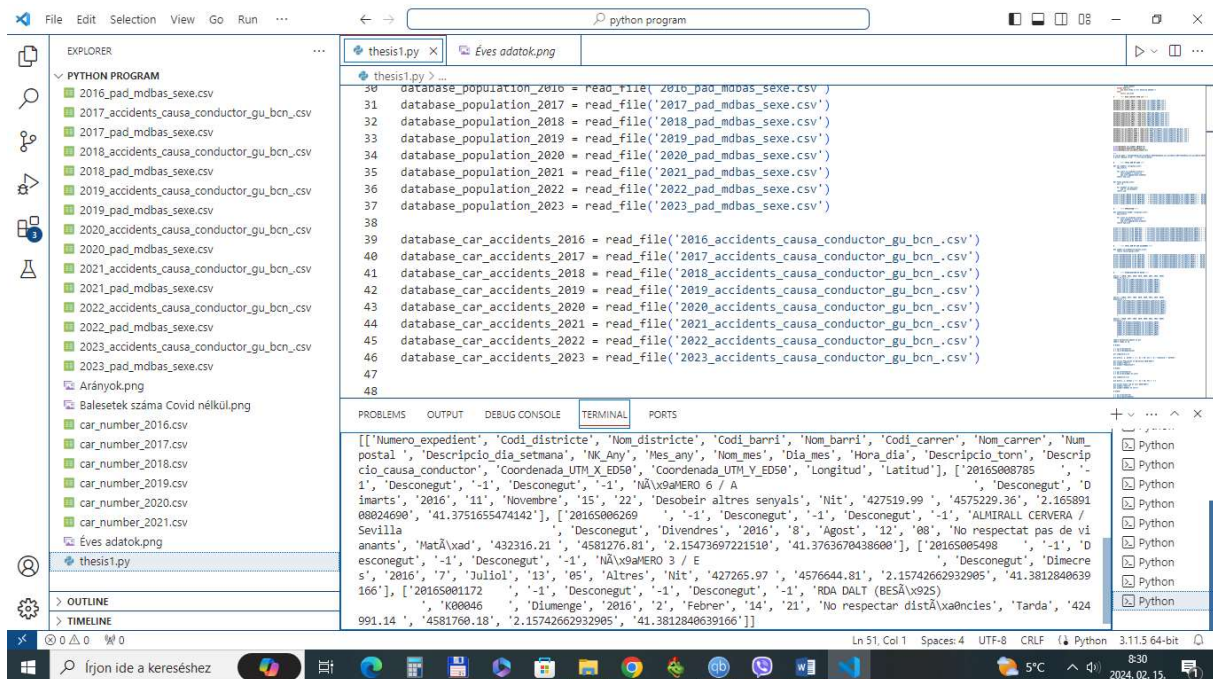
# print(database_car_number_2016[0:5])
# print(database_population_2016[0:5])
# print(database_car_accidents_2016[0:5])
```

A fenti kódrészlet az adatok beolvasásáért felelős. Az adatállomány összesen 22 fájlból áll. A kódrészlet megnyitja a fájlokat, kiolvassa és lementi a tartalmukat egy kezdeti csv\_list változóba, majd bezárja a fájlt. Amennyiben hiba történik a fájlok olvasása közben, úgy arról visszajelzést ad, ha minden rendben volt, akkor visszaadja a beolvasott adatokat tartalmazó listát. Ezt követően mind a 22 fájlra lefuttatásra kerül a beolvasó kód és minden fájl tartalmát lementi egy database kezdetű változóba, így a továbbiakban ezen változókra hivatkozva könnyen el lehet érni az adatokat. A fájlok beolvasása során az autóbalesetek adatait tartalmazó fájlok „UnicodeDecodeError: 'charmap' codec can't decode byte 0x81 in position 1964: character maps to <undefined>” hibát jeleztek. Utánanézzve rájöttem, hogy ezek az adatfájlok nem az általánosan elterjedt UTF-8-as kódolást használják, hanem a Latin-1 kódolást, így az open parancsot kiegészítettem az encoding = 'Latin1' részlettel, így már a beolvasásért felelős kódrészlet probléma nélkül lefutott. Próbaként minden adatállományból az első 5 sort kiíratam. Azért, hogy a program végső futásánál ez ne legyen zavaró, ugyanakkor bármikor felhasználható legyen, amikor ellenőrzésként szükség van rá, ezért kommentként benne hagytam a kódrészletben.



## 6. ábra: Adatok beolvasásának tesztfuttatása

(Forrás: saját munka)



```
thesis1.py
30 database_population_2016 = read_file('2016_pad_mdabas_sexe.csv')
31 database_population_2017 = read_file('2017_pad_mdabas_sexe.csv')
32 database_population_2018 = read_file('2018_pad_mdabas_sexe.csv')
33 database_population_2019 = read_file('2019_pad_mdabas_sexe.csv')
34 database_population_2020 = read_file('2020_pad_mdabas_sexe.csv')
35 database_population_2021 = read_file('2021_pad_mdabas_sexe.csv')
36 database_population_2022 = read_file('2022_pad_mdabas_sexe.csv')
37 database_population_2023 = read_file('2023_pad_mdabas_sexe.csv')
38
39 database_car_accidents_2016 = read_file('2016_accidents_causa_conductor_gu_bcn_csv')
40 database_car_accidents_2017 = read_file('2017_accidents_causa_conductor_gu_bcn_csv')
41 database_car_accidents_2018 = read_file('2018_accidents_causa_conductor_gu_bcn_csv')
42 database_car_accidents_2019 = read_file('2019_accidents_causa_conductor_gu_bcn_csv')
43 database_car_accidents_2020 = read_file('2020_accidents_causa_conductor_gu_bcn_csv')
44 database_car_accidents_2021 = read_file('2021_accidents_causa_conductor_gu_bcn_csv')
45 database_car_accidents_2022 = read_file('2022_accidents_causa_conductor_gu_bcn_csv')
46 database_car_accidents_2023 = read_file('2023_accidents_causa_conductor_gu_bcn_csv')
47
48

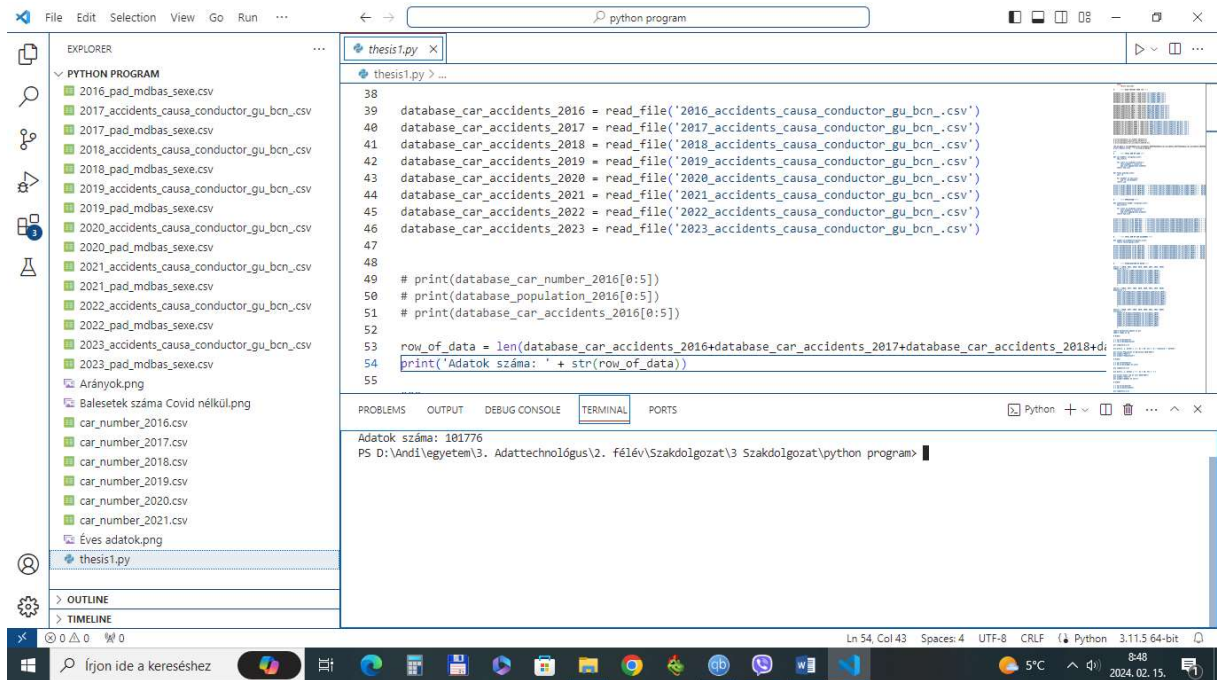
[[{'Numero_expedient': 'Codi_districte', 'Nom_districte', 'Codi_barri', 'Nom_barri', 'Codi_carren', 'Nom_carren', 'Num_postal', 'Descripcio_dia_setmana', 'NK_Any', 'Mes_any', 'Nom_mes', 'Dia_mes', 'Hona_dia', 'Descripcio_torn', 'Descripcio_causa_conductor', 'Coordenada_UTM_X_ED50', 'Coordenada_UTM_Y_ED50', 'Longitud', 'Latitud', ['20165008785', '-1', 'Desconegut', '-1', 'Desconegut', '-1', 'NÀ\x9aMERO 6 / A', 'Desconegut', 'D imants', '2016', '11', 'Novembre', '15', '22', 'Desobeir altres senyals', 'Nit', '427519.99', '4575229.36', '2.165891 08024690', '41.3751655474142'], ['20165006269', '-1', 'Desconegut', '-1', 'Desconegut', '-1', 'ALMIRALL CERVERA / Sevilla', 'Desconegut', 'Divendres', '2016', '8', 'Agost', '12', '08', 'No respectat pas de vi anants', 'Matà\xad', '432316.21', '4581276.81', '2.15473697221510', '41.3763670438600'], ['20165005498', '-1', 'D desconegut', '-1', 'Desconegut', '-1', 'NÀ\x9aMERO 3 / E', 'Desconegut', 'Dimecre s', '2016', '7', 'Juliol', '13', '05', 'Altres', 'Nit', '427265.97', '4576644.81', '2.15742662932995', '41.3812840639 166'], ['20165001172', '-1', 'Desconegut', '-1', 'Desconegut', '-1', 'RDA DALT (BESÀ\x92S)', 'K00046', 'Diumenge', '2016', '2', 'Febrer', '14', '21', 'No respectat distà\xabncies', 'Tarda', '424 991.14', '4581760.18', '2.15742662932995', '41.3812840639166']]]
```

```
# row_of_data =
len(database_car_accidents_2016+database_car_accidents_2017+database_car_accid
ents_2018+database_car_accidents_2019+database_car_accidents_2020+database_car
_accidents_2021+database_car_accidents_2022+database_car_accidents_2023+databa
se_car_number_2016+database_car_number_2017+database_car_number_2018+database_
car_number_2019+database_car_number_2020+database_car_number_2021+database_pop
ulation_2016+database_population_2017+database_population_2018+database_popula
tion_2019+database_population_2020+database_population_2021+database_populatio
n_2022+database_population_2023)
# print('Adatok száma: ' + str(row_of_data))
```

A fenti kódrészlet megszámolja és kiírja, hogy a 22 adatfájl összesen hány sor adatot tartalmaz. Mivel a program futása szempontjából nem fontos információ, csupán a szakdolgozat szempontjából szolgál érdekes adatként, ezért ezt is kommentként hagytam meg a kódban.

## 7. ábra: Adatsorok számának kiírása

(Forrás: saját munka)



```
38
39 database_car_accidents_2016 = read_file('2016_accidents_causa_conductor_gu_bcn_.csv')
40 database_car_accidents_2017 = read_file('2017_accidents_causa_conductor_gu_bcn_.csv')
41 database_car_accidents_2018 = read_file('2018_accidents_causa_conductor_gu_bcn_.csv')
42 database_car_accidents_2019 = read_file('2019_accidents_causa_conductor_gu_bcn_.csv')
43 database_car_accidents_2020 = read_file('2020_accidents_causa_conductor_gu_bcn_.csv')
44 database_car_accidents_2021 = read_file('2021_accidents_causa_conductor_gu_bcn_.csv')
45 database_car_accidents_2022 = read_file('2022_accidents_causa_conductor_gu_bcn_.csv')
46 database_car_accidents_2023 = read_file('2023_accidents_causa_conductor_gu_bcn_.csv')
47
48
49 # print(database_car_number_2016[0:5])
50 # print(database_population_2016[0:5])
51 # print(database_car_accidents_2016[0:5])
52
53 row_of_data = len(database_car_accidents_2016+database_car_accidents_2017+database_car_accidents_2018+d
54 print("Adatok száma: " + str(row_of_data))
55
```

Adatok száma: 101776  
PS D:\Andi\egyetem\3. Adattechnológus\2. félév\Szakdolgozat\3 Szakdolgozat\python program>

```
# ----- TOTAL SUM OF CARS -----
```

```
def car_numbers (original_list):
    new_list=[]

    for lists in original_list[1:]:
        last_element = lists[-1]
        new_list.append(last_element)
    return new_list
```

```
def total_sum(new_list):
    sum = 0

    for element in new_list:
        sum += int(element)
    return sum
```

```
print('A gépjárművek száma 2016-ban: ' +
str(total_sum(car_numbers(database_car_number_2016))) + ' darab.')
print('A gépjárművek száma 2017-ben: ' +
str(total_sum(car_numbers(database_car_number_2017))) + ' darab.')
print('A gépjárművek száma 2018-ban: ' +
str(total_sum(car_numbers(database_car_number_2018))) + ' darab.')
```

```

print('A gépjárművek száma 2019-ben: ' +
str(total_sum(car_numbers(database_car_number_2019))) + ' darab.')
print('A gépjárművek száma 2020-ban: ' +
str(total_sum(car_numbers(database_car_number_2020))) + ' darab.')
print('A gépjárművek száma 2021-ben: ' +
str(total_sum(car_numbers(database_car_number_2021))) + ' darab.')

```

A fenti kódrészlet első része (car\_numbers) végigmegy az adatbázison és minden sor utolsó elemét, amely a regisztrált gépjárművek számát tartalmazza, hozzáadja egy új listához. A kódrészlet második része (total\_sum) végigmegy az újonnan létrejött lista minden egyes elemén és összegzi az abban található számokat. A kódrészlet harmadik része pedig a gépjárművek számát tartalmazó adatbázisokon lefuttatja az előző két kódrészletet, majd évenként kiírja, hogy összesen hány darab gépjármű volt regisztrálva abban az évben Barcelona városában.

## 8. ábra: Gépjárművek számának kiírása

(Forrás: saját munka)

```

60 new_list=[]
61
62 for lists in original_list[1:]:
63     last_element = lists[-1]
64     new_list.append(last_element)
65 return new_list
66
67
68 def total_sum(new_list):
69     sum = 0
70
71     for element in new_list:
72         sum += int(element)
73     return sum
74
75 print('A gépjárművek száma 2016-ban: ' + str(total_sum(car_numbers(database_car_number_2016))) + ' darab')
76 print('A gépjárművek száma 2017-ben: ' + str(total_sum(car_numbers(database_car_number_2017))) + ' darab')
77 print('A gépjárművek száma 2018-ban: ' + str(total_sum(car_numbers(database_car_number_2018))) + ' darab')
78 print('A gépjárművek száma 2019-ben: ' + str(total_sum(car_numbers(database_car_number_2019))) + ' darab')
79 print('A gépjárművek száma 2020-ban: ' + str(total_sum(car_numbers(database_car_number_2020))) + ' darab')
80 print('A gépjárművek száma 2021-ben: ' + str(total_sum(car_numbers(database_car_number_2021))) + ' darab')
81

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

A gépjárművek száma 2016-ban: 816040 darab.
A gépjárművek száma 2017-ben: 820067 darab.
A gépjárművek száma 2018-ban: 827278 darab.
A gépjárművek száma 2019-ben: 824801 darab.
A gépjárművek száma 2020-ban: 822211 darab.
A gépjárművek száma 2021-ben: 811573 darab.
PS D:\Andi\egyetem\3. Adatechnológus\2. félév\Szakdolgozat\3 Szakdolgozat\python program>

```

Ln 82, Col 4 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit

```

#      ---- POPULATION ----

def inhabitants_number (original_list):
    new_list=[]

    for lists in original_list[1:]:
        last_element = lists[-2]
        new_list.append(last_element)
    return new_list

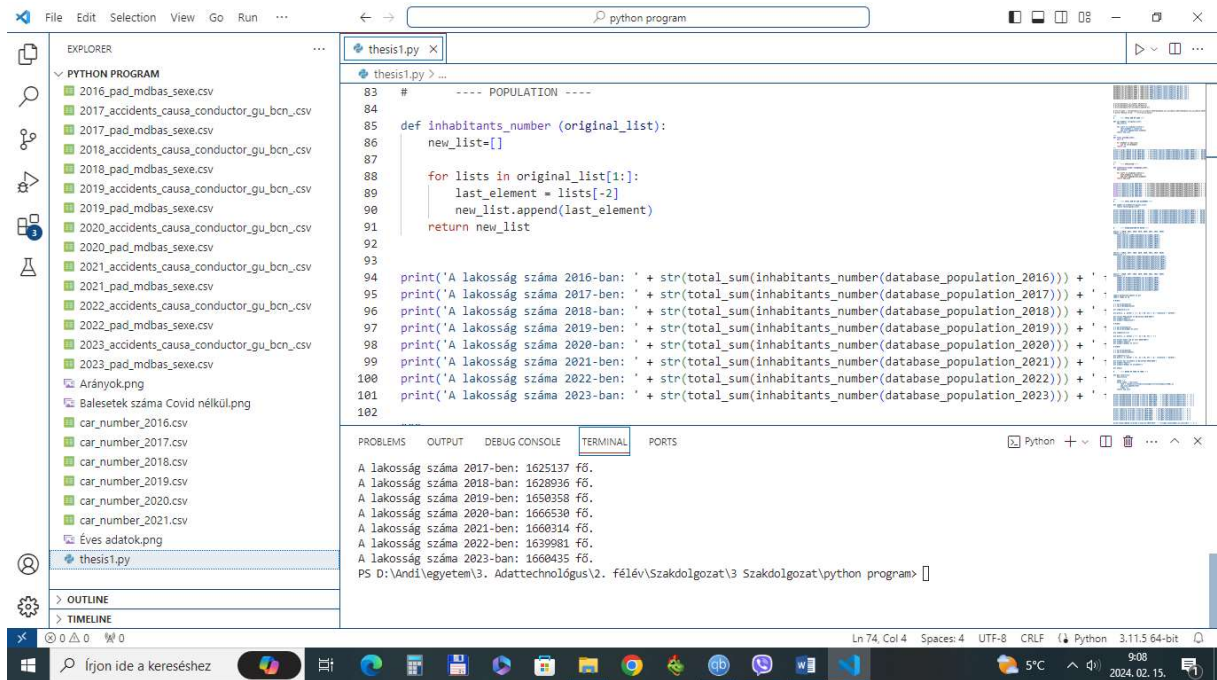
print('A lakosság száma 2016-ban: ' +
str(total_sum(inhabitants_number(database_population_2016))) + ' fő.')
print('A lakosság száma 2017-ben: ' +
str(total_sum(inhabitants_number(database_population_2017))) + ' fő.')
print('A lakosság száma 2018-ban: ' +
str(total_sum(inhabitants_number(database_population_2018))) + ' fő.')
print('A lakosság száma 2019-ben: ' +
str(total_sum(inhabitants_number(database_population_2019))) + ' fő.')
print('A lakosság száma 2020-ban: ' +
str(total_sum(inhabitants_number(database_population_2020))) + ' fő.')
print('A lakosság száma 2021-ben: ' +
str(total_sum(inhabitants_number(database_population_2021))) + ' fő.')
print('A lakosság száma 2022-ben: ' +
str(total_sum(inhabitants_number(database_population_2022))) + ' fő.')
print('A lakosság száma 2023-ban: ' +
str(total_sum(inhabitants_number(database_population_2023))) + ' fő.')

```

A fenti kódrészlet lényegében ugyanúgy működik, mint a gépjárművek száma esetén, azzal a különbséggel, hogy ebben az esetben az adatállomány sorainak utolsó előtti adata tartalmazta a számunkra releváns információt.

## 9. ábra: Lakosság számának kiírása

(Forrás: saját munka)



The screenshot shows a Python IDE with a file explorer on the left containing various CSV files. The main editor displays a Python script named 'thesis1.py'. The script defines a function 'inhabitants\_number' that takes a list and returns a new list with the last element removed. It then prints the total sum of inhabitants for each year from 2016 to 2023. The terminal window at the bottom shows the output of the script, displaying the population for each year.

```
83 # ----- POPULATION -----
84
85 def inhabitants_number (original_list):
86     new_list=[]
87
88     for lists in original_list[1:]:
89         last_element = lists[-2]
90         new_list.append(last_element)
91     return new_list
92
93
94 print('A lakosság száma 2016-ban: ' + str(total_sum(inhabitants_number(database_population_2016))) + ' ')
95 print('A lakosság száma 2017-ben: ' + str(total_sum(inhabitants_number(database_population_2017))) + ' ')
96 print('A lakosság száma 2018-ban: ' + str(total_sum(inhabitants_number(database_population_2018))) + ' ')
97 print('A lakosság száma 2019-ben: ' + str(total_sum(inhabitants_number(database_population_2019))) + ' ')
98 print('A lakosság száma 2020-ban: ' + str(total_sum(inhabitants_number(database_population_2020))) + ' ')
99 print('A lakosság száma 2021-ben: ' + str(total_sum(inhabitants_number(database_population_2021))) + ' ')
100 print('A lakosság száma 2022-ben: ' + str(total_sum(inhabitants_number(database_population_2022))) + ' ')
101 print('A lakosság száma 2023-ban: ' + str(total_sum(inhabitants_number(database_population_2023))) + ' ')
102
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - + + + +
A lakosság száma 2017-ben: 1625137 fő.
A lakosság száma 2018-ban: 1628936 fő.
A lakosság száma 2019-ben: 1659358 fő.
A lakosság száma 2020-ban: 1666530 fő.
A lakosság száma 2021-ben: 1660314 fő.
A lakosság száma 2022-ben: 1639981 fő.
A lakosság száma 2023-ban: 1660435 fő.
PS D:\Andi\egyetem\3. Adattechnológus\2. félév\Szakdolgozat\3 Szakdolgozat\python program> |
```

```
# ----- TOTAL SUM OF CAR ACCIDENTS -----
```

```
def number_of_elements(original_list):
    return len(original_list)
```

```
print('Autóbalesetek száma 2016-ban: ' +
str(number_of_elements(database_car_accidents_2016)) + ' darab.')
print('Autóbalesetek száma 2017-ben: ' +
str(number_of_elements(database_car_accidents_2017)) + ' darab.')
print('Autóbalesetek száma 2018-ban: ' +
str(number_of_elements(database_car_accidents_2018)) + ' darab.')
print('Autóbalesetek száma 2019-ben: ' +
str(number_of_elements(database_car_accidents_2019)) + ' darab.')
print('Autóbalesetek száma 2020-ban: ' +
str(number_of_elements(database_car_accidents_2020)) + ' darab.')
print('Autóbalesetek száma 2021-ben: ' +
str(number_of_elements(database_car_accidents_2021)) + ' darab.')
print('Autóbalesetek száma 2022-ben: ' +
str(number_of_elements(database_car_accidents_2022)) + ' darab.')
print('Autóbalesetek száma 2023-ban: ' +
str(number_of_elements(database_car_accidents_2023)) + ' darab.')
```

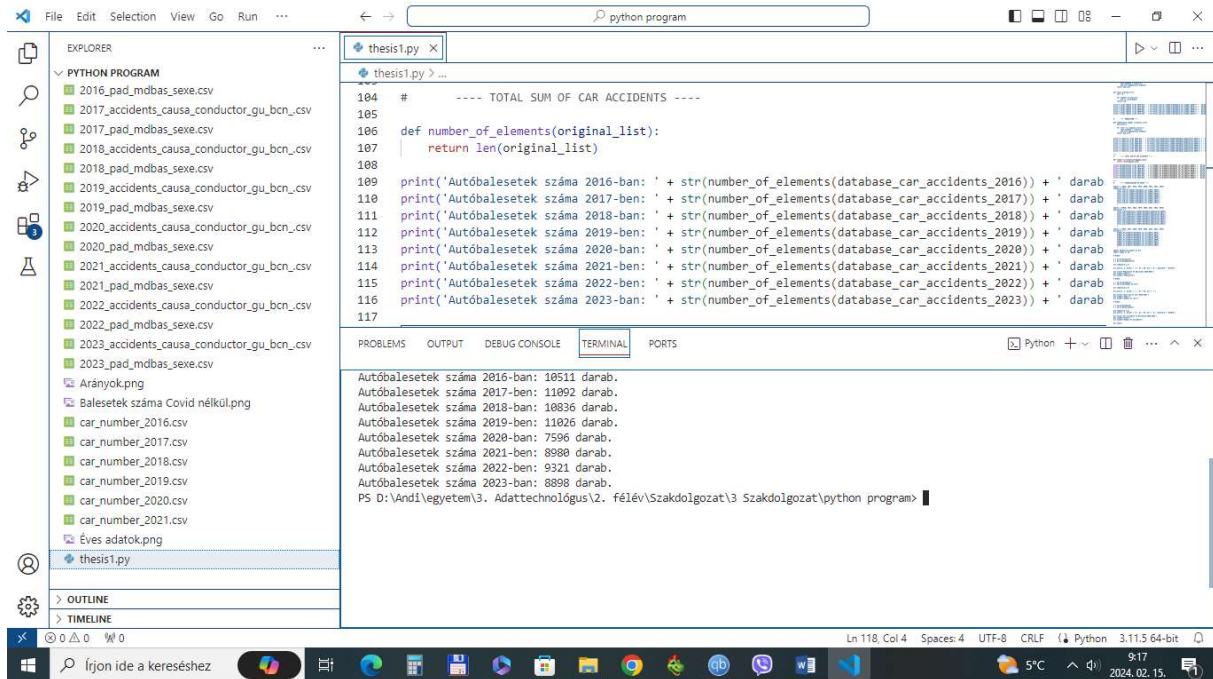
A fenti kódrészlet az autóbalesetek számát tartalmazó adatállományokon megy végig és megszámlolja, hogy hány sor adatot tartalmaznak. Ezekben az adatállományokban minden



sor egy-egy balesetet reprezentál. Ezt követően évenkénti bontásban kiírja, hogy melyik évben hány baleset történt Barcelona városában.

## 10. ábra: Autóbalesetek számának kiírása

(Forrás: saját munka)



```
104 # ----- TOTAL SUM OF CAR ACCIDENTS -----
105
106 def number_of_elements(original_list):
107     return len(original_list)
108
109 print('Autóbalesetek száma 2016-ban: ' + str(number_of_elements(database_car_accidents_2016)) + ' darab
110 print('Autóbalesetek száma 2017-ben: ' + str(number_of_elements(database_car_accidents_2017)) + ' darab
111 print('Autóbalesetek száma 2018-ban: ' + str(number_of_elements(database_car_accidents_2018)) + ' darab
112 print('Autóbalesetek száma 2019-ben: ' + str(number_of_elements(database_car_accidents_2019)) + ' darab
113 print('Autóbalesetek száma 2020-ban: ' + str(number_of_elements(database_car_accidents_2020)) + ' darab
114 print('Autóbalesetek száma 2021-ben: ' + str(number_of_elements(database_car_accidents_2021)) + ' darab
115 print('Autóbalesetek száma 2022-ben: ' + str(number_of_elements(database_car_accidents_2022)) + ' darab
116 print('Autóbalesetek száma 2023-ban: ' + str(number_of_elements(database_car_accidents_2023)) + ' darab
117
```

```
Autóbalesetek száma 2016-ban: 10511 darab.
Autóbalesetek száma 2017-ben: 11092 darab.
Autóbalesetek száma 2018-ban: 10836 darab.
Autóbalesetek száma 2019-ben: 11026 darab.
Autóbalesetek száma 2020-ban: 7596 darab.
Autóbalesetek száma 2021-ben: 8980 darab.
Autóbalesetek száma 2022-ben: 9321 darab.
Autóbalesetek száma 2023-ban: 8898 darab.
```

```
# ----- VISUALIZATION OF DATAS -----
```

```
years1 = [2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]
```

```
number_of_cars = [
    total_sum(car_numbers(database_car_number_2016)),
    total_sum(car_numbers(database_car_number_2017)),
    total_sum(car_numbers(database_car_number_2018)),
    total_sum(car_numbers(database_car_number_2019)),
    total_sum(car_numbers(database_car_number_2020)),
    total_sum(car_numbers(database_car_number_2021)),
    total_sum(car_numbers(database_car_number_2021)),
    total_sum(car_numbers(database_car_number_2021)),
]
```

```
years2 = [2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]
```

```
population = [
    total_sum(inhabitants_number(database_population_2016)),
    total_sum(inhabitants_number(database_population_2017)),
    total_sum(inhabitants_number(database_population_2018)),
    total_sum(inhabitants_number(database_population_2019)),
]
```

```
total_sum(inhabitants_number(database_population_2020)),
total_sum(inhabitants_number(database_population_2021)),
total_sum(inhabitants_number(database_population_2022)),
total_sum(inhabitants_number(database_population_2023))
]
```

```
years3 = [2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]
```

```
accidents = [
    number_of_elements(database_car_accidents_2016),
    number_of_elements(database_car_accidents_2017),
    number_of_elements(database_car_accidents_2018),
    number_of_elements(database_car_accidents_2019),
    number_of_elements(database_car_accidents_2020),
    number_of_elements(database_car_accidents_2021),
    number_of_elements(database_car_accidents_2022),
    number_of_elements(database_car_accidents_2023)
]
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
# PLOT1:
```

```
x = np.array(years2)
y = np.array(population)
```

```
plt.subplot(3,1,1)
```

```
plt.scatter(x, y, color = 'blue')
```

```
plt.title('Barcelona lakossága 2016-2023')
plt.xlabel("Év")
plt.ylabel("Lakosság szám")
```

```
# PLOT2:
```

```
x = np.array(years1)
y = np.array(number_of_cars)
```

```
plt.subplot(3,1,2)
```

```
plt.scatter(x, y, color = 'green')
```

```
plt.title('Gépjárművek száma 2016-2021')
plt.xlabel("Év")
plt.ylabel("Gépjárművek száma")
```

```
# PLOT3:
```

```

x = np.array(years3)
y = np.array(accidents)

plt.subplot(3,1,3)
plt.scatter(x, y, color = 'red')

plt.title('Autóbalesetek száma Barcelonában 2016-2023')
plt.xlabel("Év")
plt.ylabel("Balesetek száma")

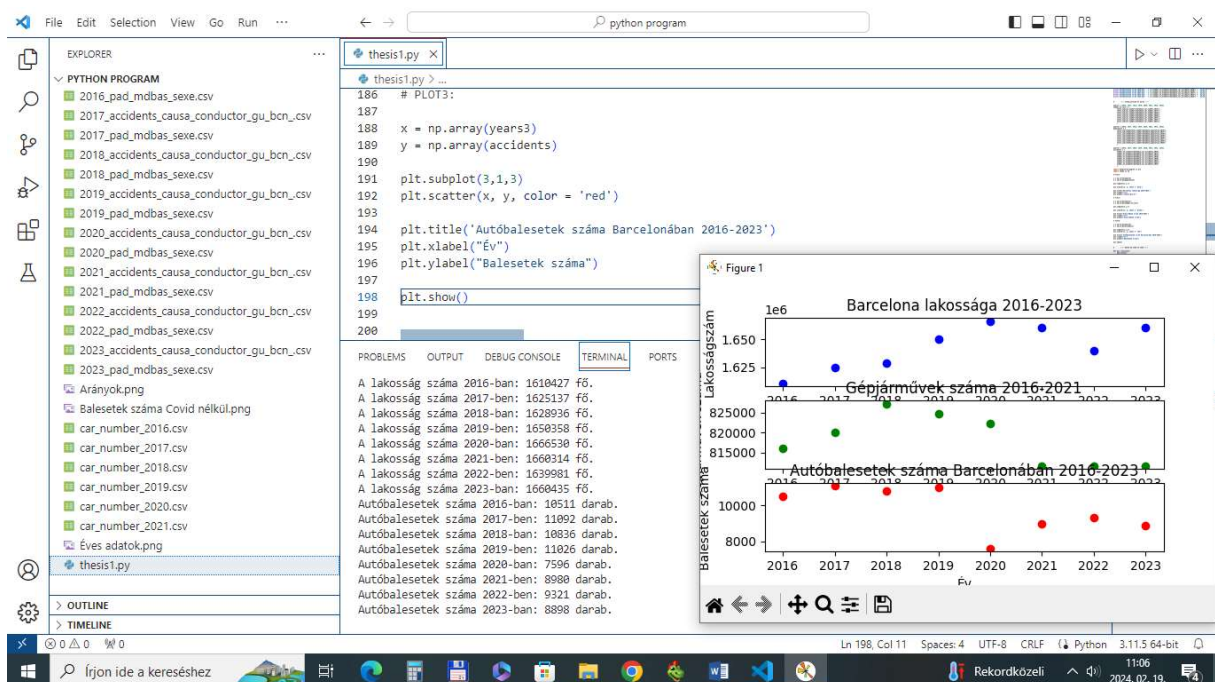
plt.show()

```

A fenti kódrészlet a korábban összesített és kiíratott gépjárművek számát, lakosság számát és autóbalesetek számát jeleníti meg diagramon. Mivel a gépjárművek száma 820.000 darab körül mozog, a lakosság száma 1.650.000 fő körül mozog, az autóbalesetek száma pedig 10.000 esetszám körül mozog, ezért a nagy dimenzióeltérés miatt egyetlen diagramon nem lehet ábrázolni a három adatot, mivel ebben az esetben csupán három vízszintes vonal jelenne meg, így az összehasonlíthatóság érdekében a legjobb megoldás, ha a három adatsort három egymás alatti diagramon jelenítjük meg.

## 11. ábra: Összesítő adatok megjelenítése diagramon

(Forrás: saját munka)





```

#      ---- RATES OF YEAR TO YEAR ----

def get_rate(list):
    new_list=[]

    index = 1
    while index < len(list):
        rate = round(((1.0-float(list[index-1]/list[index]))*100),2)
        new_list.append(rate)
        index +=1
    return new_list

print('Autóbalesetek számának változása 2016-2017: ' +
str(get_rate(accidents)[0]) + ' %.')
print('Autóbalesetek számának változása 2017-2018: ' +
str(get_rate(accidents)[1]) + ' %.')
print('Autóbalesetek számának változása 2018-2019: ' +
str(get_rate(accidents)[2]) + ' %.')
print('Autóbalesetek számának változása 2019-2020: ' +
str(get_rate(accidents)[3]) + ' %.')
print('Autóbalesetek számának változása 2020-2021: ' +
str(get_rate(accidents)[4]) + ' %.')
print('Autóbalesetek számának változása 2021-2022: ' +
str(get_rate(accidents)[5]) + ' %.')
print('Autóbalesetek számának változása 2022-2023: ' +
str(get_rate(accidents)[6]) + ' %.')

print('Lakosság számának változása 2016-2017: ' + str(get_rate(population)[0])
+ ' %.')
print('Lakosság számának változása 2017-2018: ' + str(get_rate(population)[1])
+ ' %.')
print('Lakosság számának változása 2018-2019: ' + str(get_rate(population)[2])
+ ' %.')
print('Lakosság számának változása 2019-2020: ' + str(get_rate(population)[3])
+ ' %.')
print('Lakosság számának változása 2020-2021: ' + str(get_rate(population)[4])
+ ' %.')
print('Lakosság számának változása 2021-2022: ' + str(get_rate(population)[5])
+ ' %.')
print('Lakosság számának változása 2022-2023: ' + str(get_rate(population)[6])
+ ' %.')

print('Gépjárművek számának változása 2016-2017: ' +
str(get_rate(number_of_cars)[0]) + ' %.')
print('Gépjárművek számának változása 2017-2018: ' +
str(get_rate(number_of_cars)[1]) + ' %.')
print('Gépjárművek számának változása 2018-2019: ' +
str(get_rate(number_of_cars)[2]) + ' %.')

```

```

print('Gépjárművek számának változása 2019-2020: ' +
str(get_rate(number_of_cars)[3]) + ' %.')
print('Gépjárművek számának változása 2020-2021: ' +
str(get_rate(number_of_cars)[4]) + ' %.')

```

A fenti kódrészlet első része (get\_rate) minden korábban kapott összesítő adat esetén kiszámolja az előző évhez viszonyított százalékos változást. A kódrészlet második része pedig szövegesen kilistázza a változásokat mind a gépjárművek számára, mind a lakosság számára, mind pedig az autóbalesetek számára vonatkozóan.

## 12. ábra: Előző évhez viszonyított százalékos változások kiírása

(Forrás: saját munka)

```

223 print('Lakosság számának változása 2017-2018: ' + str(get_rate(population)[1]) + ' %.')
224 print('Lakosság számának változása 2018-2019: ' + str(get_rate(population)[2]) + ' %.')
225 print('Lakosság számának változása 2019-2020: ' + str(get_rate(population)[3]) + ' %.')
226 print('Lakosság számának változása 2020-2021: ' + str(get_rate(population)[4]) + ' %.')
227 print('Lakosság számának változása 2021-2022: ' + str(get_rate(population)[5]) + ' %.')
228 print('Lakosság számának változása 2022-2023: ' + str(get_rate(population)[6]) + ' %.')
229
230 print('Gépjárművek számának változása 2016-2017: ' + str(get_rate(number_of_cars)[0]) + ' %.')
231 print('Gépjárművek számának változása 2017-2018: ' + str(get_rate(number_of_cars)[1]) + ' %.')
232 print('Gépjárművek számának változása 2018-2019: ' + str(get_rate(number_of_cars)[2]) + ' %.')
233 print('Gépjárművek számának változása 2019-2020: ' + str(get_rate(number_of_cars)[3]) + ' %.')
234 print('Gépjárművek számának változása 2020-2021: ' + str(get_rate(number_of_cars)[4]) + ' %.')
235

```

```

Autóbalesetek számának változása 2017-2018: -2.36 %.
Autóbalesetek számának változása 2018-2019: 1.72 %.
Autóbalesetek számának változása 2019-2020: -45.16 %.
Autóbalesetek számának változása 2020-2021: 15.41 %.
Autóbalesetek számának változása 2021-2022: 3.66 %.
Autóbalesetek számának változása 2022-2023: -4.75 %.
Lakosság számának változása 2016-2017: 0.91 %.
Lakosság számának változása 2017-2018: 0.23 %.
Lakosság számának változása 2018-2019: 1.3 %.
Lakosság számának változása 2019-2020: 0.97 %.
Lakosság számának változása 2020-2021: -0.37 %.
Lakosság számának változása 2021-2022: -1.24 %.
Lakosság számának változása 2022-2023: 1.23 %.
Gépjárművek számának változása 2016-2017: 0.49 %.
Gépjárművek számának változása 2017-2018: 0.87 %.
Gépjárművek számának változása 2018-2019: -0.3 %.
Gépjárművek számának változása 2019-2020: -0.32 %.
Gépjárművek számának változása 2020-2021: -1.3 %.

```

```
# ---- VISUALIZATION OF RATES ----
```

```

years1 = [2017, 2018, 2019, 2020, 2021, 2022, 2023]
rate_of_number_of_cars = [
    get_rate(number_of_cars)[0],
    get_rate(number_of_cars)[1],
    get_rate(number_of_cars)[2],
    get_rate(number_of_cars)[3],
    get_rate(number_of_cars)[4],
    0, # hiányzó adat
    0 # hiányzó adat
]

```

```
years2 = [2017, 2018, 2019, 2020, 2021, 2022, 2023]
rate_of_population = [
    get_rate(population)[0],
    get_rate(population)[1],
    get_rate(population)[2],
    get_rate(population)[3],
    get_rate(population)[4],
    get_rate(population)[5],
    get_rate(population)[6],
]
```

```
years3 = [2017, 2018, 2019, 2020, 2021, 2022, 2023]
rate_of_accidents = [
    get_rate(accidents)[0],
    get_rate(accidents)[1],
    get_rate(accidents)[2],
    get_rate(accidents)[3],
    get_rate(accidents)[4],
    get_rate(accidents)[5],
    get_rate(accidents)[6],
]
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x1 = np.array(years1)
y1 = np.array(rate_of_number_of_cars)
```

```
x2 = np.array(years2)
y2 = np.array(rate_of_population)
```

```
x3 = np.array(years3)
y3 = np.array(rate_of_accidents)
```

```
plt.scatter(x1,y1, color = 'green', label = 'Gépjárművek számának változása')
plt.scatter(x2,y2, color = 'blue', label = 'Lakosság számának változása')
plt.scatter(x3,y3, color = 'red', label = 'Balesetek számának változása')
```

```
plt.title('Változások 2016-2023')
plt.xlabel('Év')
plt.ylabel('Százalék')
```

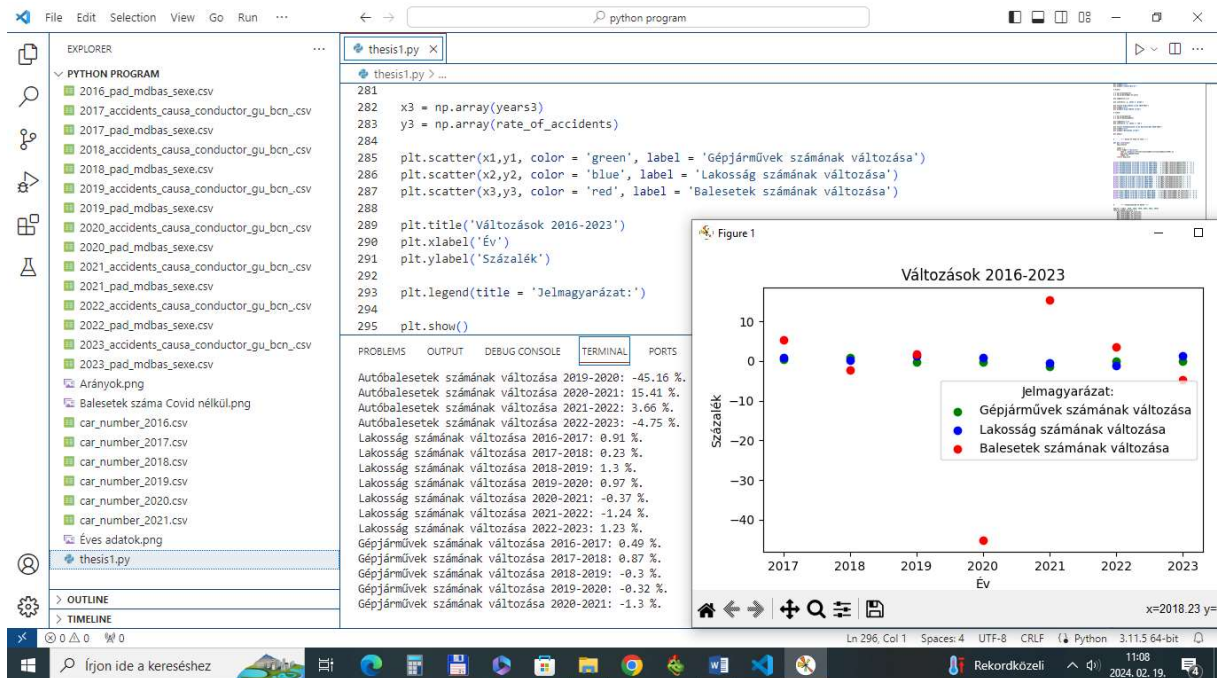
```
plt.legend(title = 'Jelmagyarázat:')
```

```
plt.show()
```

A fenti kódrészlet egy diagramon jeleníti meg mindhárom adatsorra vonatkozóan a százalékos arányok változását. Mivel a gépjárművek száma esetén nincs meg a 2022-2023-ra vonatkozó adat, ezért ott a változást 0-val feltételeztem, hogy az adatok összehasonlíthatók legyenek.

### 13. ábra: Százalékos változások megjelenítése diagramon

(Forrás: saját munka)



```
# ---- VISUALIZATION OF DATAS OF ACCIDENTS WITHOUT COVID-EFFECT ----
```

```
years_without_covid = [2016, 2017, 2018, 2019, 2022, 2023]
accidents_without_covid = [
    number_of_elements(database_car_accidents_2016),
    number_of_elements(database_car_accidents_2017),
    number_of_elements(database_car_accidents_2018),
    number_of_elements(database_car_accidents_2019),
    number_of_elements(database_car_accidents_2022),
    number_of_elements(database_car_accidents_2023)
]
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.array(years_without_covid)
y = np.array(accidents_without_covid)
```

```
plt.scatter(x, y, color = 'red')
```

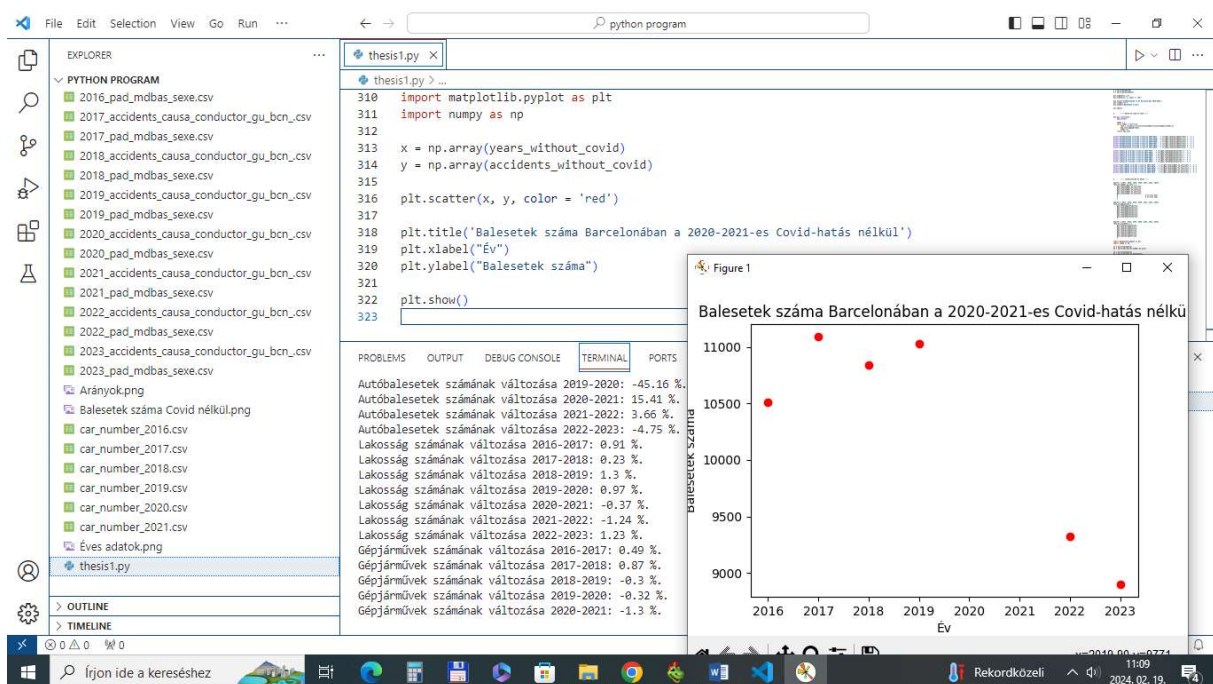
```
plt.title('Balesetek száma Barcelonában a 2020-2021-es Covid-hatás nélkül')  
plt.xlabel("Év")  
plt.ylabel("Balesetek száma")
```

```
plt.show()
```

A fenti kódrészlet ugyanúgy működik, mint az előző kódblokk, azzal a különbséggel, hogy nem tartalmazza a 2020-2021-es adatokat, mivel ott a Covid következményei miatt jelentős esés, majd pedig a járvány elmúltával jelentős emelkedés következett be a balesetek számában, így a két év adatai az elemzés során félrevezetőek.

## 14. ábra: Balesetek száma Covid-hatás nélkül

(Forrás: saját munka)



## 4.2 Python-kód alapján kapott eredmények listázása

### 4.2.1 Az adatok évenkénti összesítése

A gépjárművek száma 2016-ban: 816040 darab.

A gépjárművek száma 2017-ben: 820067 darab.

A gépjárművek száma 2018-ban: 827278 darab.

A gépjárművek száma 2019-ben: 824801 darab.

A gépjárművek száma 2020-ban: 822211 darab.

A gépjárművek száma 2021-ben: 811673 darab.

A lakosság száma 2016-ban: 1610427 fő.

A lakosság száma 2017-ben: 1625137 fő.

A lakosság száma 2018-ban: 1628936 fő.

A lakosság száma 2019-ben: 1650358 fő.

A lakosság száma 2020-ban: 1666530 fő.

A lakosság száma 2021-ben: 1660314 fő.

A lakosság száma 2022-ben: 1639981 fő.

A lakosság száma 2023-ban: 1660435 fő.

Autóbalesetek száma 2016-ban: 10511 darab.

Autóbalesetek száma 2017-ben: 11092 darab.

Autóbalesetek száma 2018-ban: 10836 darab.

Autóbalesetek száma 2019-ben: 11026 darab.

Autóbalesetek száma 2020-ban: 7596 darab.

Autóbalesetek száma 2021-ben: 8980 darab.

Autóbalesetek száma 2022-ben: 9321 darab.

Autóbalesetek száma 2023-ban: 8898 darab.

#### 4.2.2 Az összesített adatok viszonyítása az előző évhez képest

Autóbalesetek számának változása 2016-2017: 5.24 %.

Autóbalesetek számának változása 2017-2018: -2.36 %.

Autóbalesetek számának változása 2018-2019: 1.72 %.

Autóbalesetek számának változása 2019-2020: -45.16 %.

Autóbalesetek számának változása 2020-2021: 15.41 %.

Autóbalesetek számának változása 2021-2022: 3.66 %.

Autóbalesetek számának változása 2022-2023: -4.75 %.

Lakosság számának változása 2016-2017: 0.91 %.

Lakosság számának változása 2017-2018: 0.23 %.

Lakosság számának változása 2018-2019: 1.3 %.

Lakosság számának változása 2019-2020: 0.97 %.

Lakosság számának változása 2020-2021: -0.37 %.

Lakosság számának változása 2021-2022: -1.24 %.

Lakosság számának változása 2022-2023: 1.23 %.

Gépjárművek számának változása 2016-2017: 0.49 %.

Gépjárművek számának változása 2017-2018: 0.87 %.

Gépjárművek számának változása 2018-2019: -0.3 %.

Gépjárművek számának változása 2019-2020: -0.32 %.

Gépjárművek számának változása 2020-2021: -1.3 %.

### 4.3 Eredmények megjelenítése adatvizualizációval

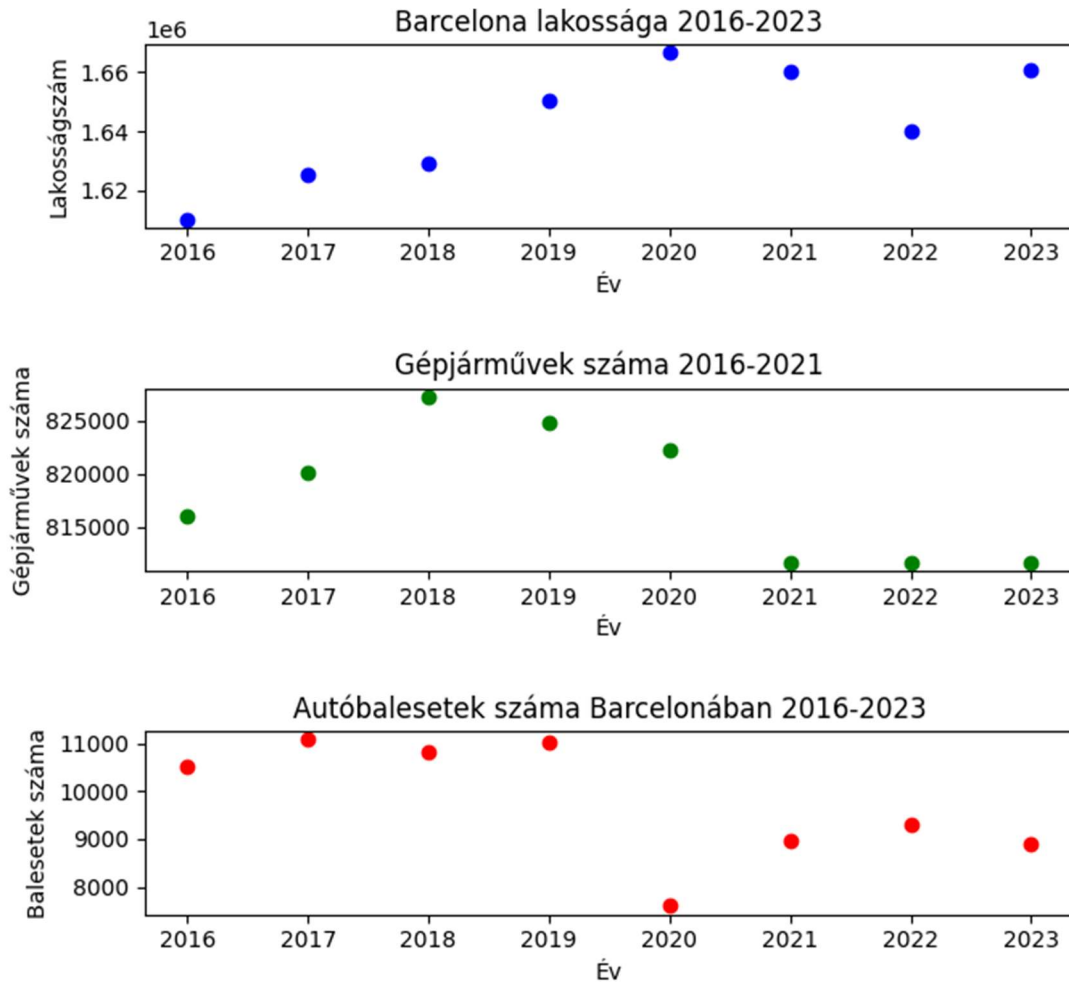
Eredetileg az eredményeket Power BI adatvizualizációval szerettem volna bemutatni, de mivel a Power BI lényege éppen az interaktivitás és ennek bemutatását jelen szakdolgozat keretei nem teszik lehetővé, így végül a matplotlib-diagramok mellett döntöttem.



### 4.3.1 Adatok évenkénti összesítésének megjelenítése diagramon

**15. ábra:** Barcelona népessége, autók száma és balesetek száma

(Forrás: saját munka)



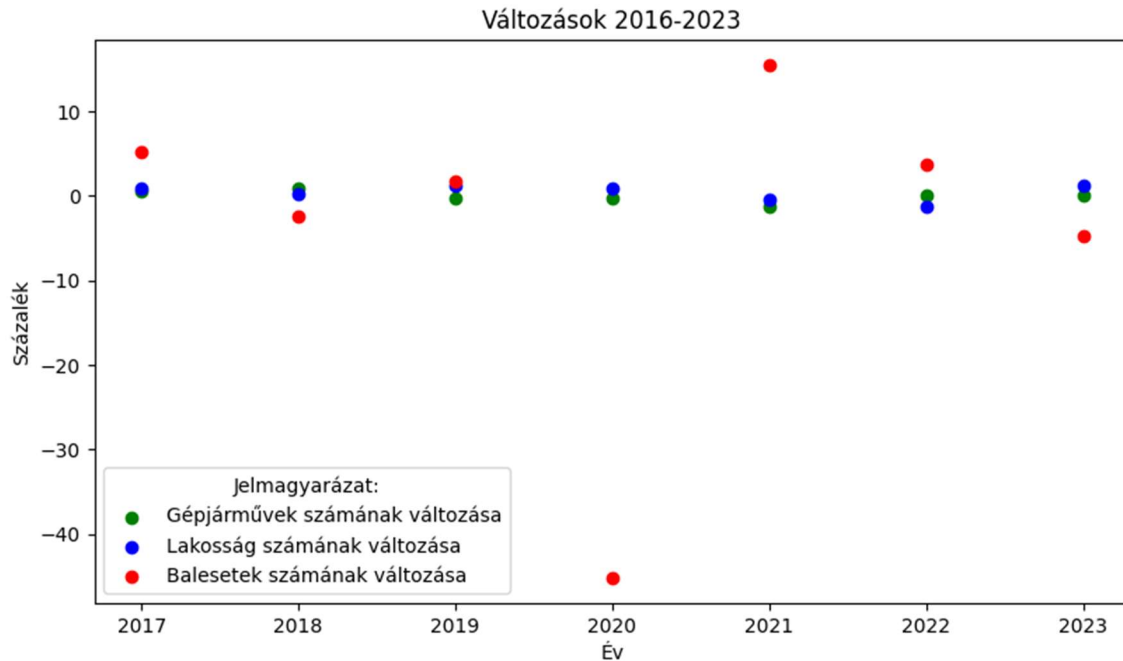
A diagramokból látható, hogy míg Barcelona lakossága minimális mértékben, de folyamatosan növekszik, addig a városban regisztrált gépjárművek száma 2018-ig növekedett, azóta viszont folyamatos csökkenést mutat. Az autóbalesetek száma 2019-ig lényegében stagnált, majd 2020-ban jelentősen lecsökkent, utána pedig kis mértékben visszakorrigált.



### 4.3.2 Előző évhez viszonyított éves adatok megjelenítése diagramon

#### 16. ábra: Éves adatok egymáshoz viszonyított változása

(Forrás: saját munka)

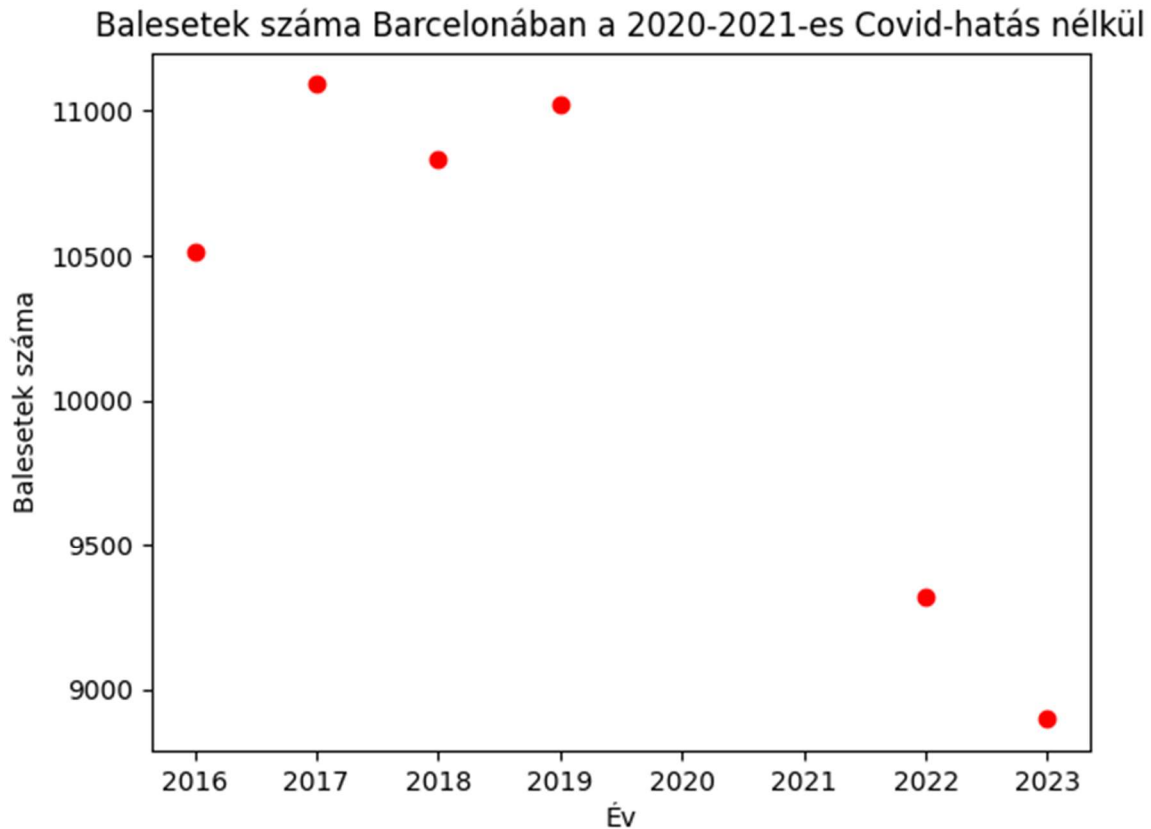


Amennyiben az éves összesített adatokat előző éves bázison egymáshoz viszonyítjuk, úgy a diagramból látható, hogy míg a népesség és a regisztrált gépjárművek száma nem változik számottevően, addig a balesetek számában erős csökkenés, majd pedig ugrásszerű növekedés következett be a 2020-2021-es időszakban. Véleményem szerint ez sokkal inkább a Covid-járvány miatt elrendelt kijárási tilalom, a „Maradj otthon”- kampány és a home office gyors terjedésének következtében valósulhatott meg, nem pedig a vezetési kultúra robbanásszerű fejlődésének vagy a smart rendszerekben rejlő lehetőség kiaknázásának következménye volt.

### 4.3.3 Autóbalesetek száma Covid-hatás nélkül

**17. ábra:** Autóbalesetek száma Barcelonában Covid-hatás nélkül

(Forrás: saját munka)



Amennyiben az autóbalesetek éves adataiból kivesszük a 2020-2021-es év adatait, úgy már jól látható, hogy a balesetek száma a 2019-et megelőző évi 11.000 esetről 2023-ra, folyamatos javulást mutatva, évi 9.000 eset alá csökkent, amely több, mint 18 %-os javulást jelent. Ez összhangban van azzal a Barcelona hivatalos weboldalán megjelent hírrel, mely szerint az autóbalesetek száma 2019 és 2022 között 22 %-kal, a halálos kimenetelű balesetek száma pedig, amelyek áldozatai elsősorban motorosok és gyalogosok voltak, 13 %-kal csökkent. **(Barcelona Városi Tanácsa, 2023.)**

A diagram alapján feltételezhető, hogy a 2019-ben elindult javulás egy folyamatos trendet határoz meg, így a következő években további csökkenés várható. Ehhez a csökkenő trendhez nagymértékben hozzájárult az, hogy Barcelona városa folyamatosan fejleszti okos rendszereit, ezzel kényelmesebb, biztonságosabb és élhetőbb körülményeket biztosítva a lakosok számára.

## 5 Következtetések és javaslatok

### 5.1 Következtetések

- míg 2009-ben világszerte csupán 24 publikáció jelent meg az okos városok kapcsán, addig 2018-ban már 6314 **(Winkowska, et al., 2019)**;
- míg 2009-ben az okos város fogalma éppen csak megszületően volt, addig 2023-ban az okos város index már 141 okos várost tartott nyilván világszerte **(IMD Smart City Observatory, 2023)**;
- a jelen szakdolgozatban elemzett Barcelonára vonatkozó közlekedési adatok a baleseti statisztikák néhány év alatt elért jelentős javulását mutatták;
- a nagy mennyiségű adat tárolására alkalmas rendszerek 2010 körüli megjelenése elindította az adatelemzés gyors fejlődését is, amelynek a közeljövőben újabb lökést ad majd a 2022-ben a nyilvánosság számára is elérhetővé tett mesterséges intelligenciában (ChatGPT, OpenAI) rejlő lehetőségek kiaknázása;

### 5.2 Javaslatok

- mivel az urbanizáció gyorsuló üteme egyre inkább a városokba tereli az embereket, az okos technológiákban rejlő lehetőségek kiaknázása biztosíthatja, hogy a létrejövő metropoliszok képesek legyenek lépést tartani a növekedéssel és élhető életteret biztosítsanak a lakosaik számára;
- a mesterséges intelligencia és az adatgyűjtés révén a világban tárolt adatok mennyisége átlagosan 6 havonta megduplázódik, ami szinte felfoghatatlan mennyiségű adatot és azokból gyorsan kinyerhető információt jelentenek, melynek kiaknázása létfontosságú az emberiség fenntartható fejlődésének biztosítása céljából;
- világszerte egyre több okos város jelenik meg, amelyek évről-évre jelentős javulást tudnak felmutatni az energiafelhasználástól kezdve a költségek csökkentésén keresztül a lakosok életminőségének javulásáig;

## 6 Összefoglalás

Jelen szakdolgozat célkitűzése az okos városok létrehozásának, valamint az okos rendszerekben rejlő lehetőségek kiaknázásának, illetve a rendszerekből származó adatok elemzési lehetőségeinek bemutatása volt.

A szakdolgozat első részében az adattudományhoz kapcsolódó fogalmak kerültek meghatározásra. Ezt követően az okos város fogalma, működése és a legelterjedtebb okos rendszerek kerültek bemutatásra.

A következő részben számokban bemutattam a világ közeljövőjében várható változásokat az urbanizáció és a világ népessége tekintetében, majd az okos városok tekintetében, végül pedig kifejezetten már megvalósult intelligens forgalomirányítási rendszerek előnyei kapcsán. Korábban olvastam egy 172.000 fős spanyol városról, Santanderről, ahol a városvezetés állítása szerint az okos parkolási rendszer bevezetésével 80 %-kal csökkent a dugók száma. Ez volt az a tény, amely kíváncsivá tett, ezért a szakdolgozatomban elsősorban az intelligens közlekedési megoldások adataira fókuszáltam.

Ezt követően az adatok forrása (22 fájl, összesen 101.776 sornyi adat), valamint azok pythonnal történő feldolgozása és az eredmények értékelési módszerei kerültek bemutatásra. Eredetileg az eredményeket Power BI adatvizualizációval szerettem volna bemutatni, de mivel a Power BI lényege az interaktivitás és ennek bemutatását jelen szakdolgozat keretei nem teszik lehetővé, így végül a matplotlib-diagramok mellett döntöttem.

A Barcelona városára vonatkozó népességi, gépjármű-nyilvántartási és közlekedési balesetek adatainak feldolgozása és az adatok vizuális megjelenítése alapján bizonyítható, hogy míg a lakosság és a gépjárművek száma 2016 és 2023 között lényegében állandó volt, addig a baleseti adatok 2019-től kezdve folyamatos és jelentős javulást mutattak (több, mint 18 %), amely várhatóan trendszerűen folytatódni fog a következő években is. A javulás háttérében az okos rendszerek használata, az okos város koncepció megvalósítása és folyamatos fejlesztése áll. Barcelona városa 2019-ben még 56. helyen állt az okos városok listájában (**IMD Smart City Observatory, 2023**). A verseny erősségét jelzi, hogy 2023-ban a folyamatos fejlesztések és jelentős eredmények ellenére is a 75. helyre csúszott vissza.

Míg 2009-ben az okos város csak egy futurisztikus elképzelés volt, addig a 2010-től kezdődően rohamos fejlődésnek indult adattárolási, adatelemzési technológiák, valamint a mesterséges intelligencia gyors fejlődésében rejlő lehetőségek kiaknázásával az okos rendszerek gazdaságosabbá, energiahatékonyabbá, kényelmesebbé, biztonságosabbá és élhetőbbé tehetik a világ egyre zsúfoltabbá váló településeit, miközben olyan előnyöket is nyújtanak, mint a CO<sub>2</sub>-kibocsátás, a karbonlábnyom, a fényszennyezés csökkentése vagy akár a bűncselekmények számának csökkenése.

## Irodalomjegyzék

- Barcelona Városi Tanácsa, 2023.. *Traffic accidents in the city down by 22%*. [Online]  
Available at: [https://www.barcelona.cat/infobarcelona/en/tema/security-and-prevention/traffic-accidents-in-the-city-down-by-22\\_1253561.html](https://www.barcelona.cat/infobarcelona/en/tema/security-and-prevention/traffic-accidents-in-the-city-down-by-22_1253561.html)  
[Hozzáférés dátuma: 13. 02. 2024.].
- Bárczi, G. & Országh, L., 1959-1962. *A magyar nyelv értelmező szótára*. első szerk. Budapest: Akadémiai Kiadó.
- Cleverciti Systems GmbH, 2021. *Smart Parking*. [Online]  
Available at: <https://www.cleverciti.com/en/smart-parking>  
[Hozzáférés dátuma: 07. 02. 2024.].
- ENSZ Gazdasági és Szociális Tanács, 2018.. *68% of the world population projected to live in urban areas by 2050, says UN*. [Online]  
Available at: <https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html>  
[Hozzáférés dátuma: 07. 02. 2024.].
- Európai Bizottság, 2024. *Smart Cities*. [Online]  
Available at: [https://commission.europa.eu/eu-regional-and-urban-development/topics/cities-and-urban-development/city-initiatives/smart-cities\\_en](https://commission.europa.eu/eu-regional-and-urban-development/topics/cities-and-urban-development/city-initiatives/smart-cities_en)  
[Hozzáférés dátuma: 07. 02. 2024.].
- Európai Unió, dátum nélk. *European data*. [Online]  
Available at: <https://data.europa.eu/data/datasets?locale=en&minScoring=0&page=4&limit=10>  
[Hozzáférés dátuma: 11. 11. 2023.].
- Haiston, J., 2023.. *What is a Smart Traffic Management System?*. [Online]  
Available at: <https://www.symmetryelectronics.com/blog/what-is-a-smart-traffic-management-system/>  
[Hozzáférés dátuma: 03. 02. 2024.].
- IMD Smart City Observatory, 2023. *Smart City Index 2023*, Svájc: ismeretlen szerző
- Isarsoft GmbH, 2023. *Smart City Strategy | The Foundation of Every Smart City*. [Online]  
Available at: <https://www.isarsoft.com/article/smart-city-strategy-the-foundation-of-every-smart-city>  
[Hozzáférés dátuma: 01. 02. 2024.].
- Juniper Research, 2022.. *Smart Traffic Management Systems to Save 205 Million Metric Tons of CO2 by 2027*. [Online]  
Available at: <https://www.juniperresearch.com/press/smart-traffic-management-systems-to-save-205-million-metric-tons-of-co2-by-2027/>  
[Hozzáférés dátuma: 03. 02. 2024.].
- Lewis, S., 2019.. *Smart Streetlight*. [Online]  
Available at: <https://www.techtarget.com/iotagenda/definition/smart-streetlight>  
[Hozzáférés dátuma: 08. 02. 2024.].
- Microsoft, 2024. *Understand-concepts-of-data-analytics*. [Online]  
Available at: <https://learn.microsoft.com/hu-hu/training/modules/understand-concepts-of-data->

[analytics/2-understand-data-analytics-types](#)

[Hozzáférés dátuma: 01. 02. 2024.].

Nimbus Lighting Group Ltd., 2021. *Why Should We Have Free Public Wi-Fi in Smart Cities?*. [Online]

Available at: <https://www.smartcity.co.nz/blog/why-should-we-have-free-public-wi-fi/>

[Hozzáférés dátuma: 08. 02. 2024.].

Sallai, G., 2018. *Az okos város koncepció és az internet következő generációja. Okos város – okos közigazgatás technológiák*. Budapest: ismeretlen szerző

Sallai, G., Kovács, K. & Jakab, L., 2018. A smart city megoldások technológiai háttere és fenntartható komplex modellje. *Magyar Tudomány*, 9.pp. 1304-1314.

Sensors Dynamics, 2023.. *What Is Smart Parking? How Does It Work?*. [Online]

Available at: <https://sensordynamics.com.au/what-is-smart-parking-how-does-it-work/>

[Hozzáférés dátuma: 07. 02. 2024.].

Sheikh, M., 2023.. *What is Smart City Surveillance?*. [Online]

Available at: <https://www.briefcam.com/resources/blog/what-is-smart-city-surveillance/>

[Hozzáférés dátuma: 08. 02. 2024.].

Szalai, Á., 2018.. *Az okos városok háttere*. [Online]

Available at: <https://geosci.u-szeged.hu/kozepiskolaknak/geojatek/2-cikk-okos-varosok>

[Hozzáférés dátuma: 02. 02. 2024.].

Tan, P.-N., Steinbach, M. & Kumar, V., 2006. *Introduction to Data Mining*. Harlow: Pearson Education Ltd.

Tvilight Projects BV, 2013. *What is a Smart Street Light?*. [Online]

Available at: <https://tvilight.com/what-is-a-smart-street-light/>

[Hozzáférés dátuma: 08. 02. 2024.].

Wikipedia, 2023.. *Adat (számítástechnika)*. [Online]

Available at: [https://hu.wikipedia.org/wiki/Adat\\_\(sz%C3%A1m%C3%ADt%C3%A1stechnika\)](https://hu.wikipedia.org/wiki/Adat_(sz%C3%A1m%C3%ADt%C3%A1stechnika))

[Hozzáférés dátuma: 01. 02. 2024.].

Winkowska, J., Szpilko, D. & Pejic, S., 2019. *Smart city concept in the light of literature review*, Lengyelország: Politechnika Bialostocka.

Zavadko, V., 2022.. *Intelligent traffic management*. [Online]

Available at: <https://intellias.com/intelligent-traffic-management/>

[Hozzáférés dátuma: 03. 02. 2024.].

<b>1. ábra:</b> Adatbányászat folyamata .....	7
<b>2. ábra:</b> Okos város-koncepció .....	10
<b>3. ábra:</b> Okos város alrendszerek .....	11
<b>4. ábra:</b> LoRa és gépi tanulás alapú intelligens forgalomirányító rendszer vázlata .....	12
<b>5. ábra:</b> Padlószenzoros parkolási rendszer .....	15
<b>6. ábra:</b> Adatok beolvasásának tesztfuttatása .....	24
<b>7. ábra:</b> Adatsorok számának kiírása.....	25
<b>8. ábra:</b> Gépjárművek számának kiírása .....	26
<b>9. ábra:</b> Lakosság számának kiírása.....	28
<b>10. ábra:</b> Autóbalesetek számának kiírása.....	29
<b>11. ábra:</b> Összesítő adatok megjelenítése diagramon .....	31
<b>12. ábra:</b> Előző évhez viszonyított százalékos változások kiírása .....	33
<b>13. ábra:</b> Százalékos változások megjelenítése diagramon .....	35
<b>14. ábra:</b> Balesetek száma Covid-hatás nélkül .....	36
<b>15. ábra:</b> Barcelona népessége, autók száma és balesetek száma .....	39
<b>16. ábra:</b> Éves adatok egymáshoz viszonyított változása .....	40
<b>17. ábra:</b> Autóbalesetek száma Barcelonában Covid-hatás nélkül .....	41



## Mellékletek

## NYILATKOZAT

### a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Bajnóczi Andrea  
A Hallgató Neptun kódja: WUODCH  
A dolgozat címe: Smart City rendszerekből származó 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 szellemi tulajdon-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 február hó 20. nap

*Bajnóczi Andrea*

Hallgató aláírása

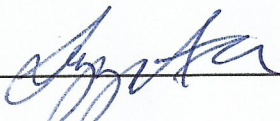
## NYILATKOZAT

Bajnóczi Andrea (név) (hallgató Neptun azonosítója: WUODCH) 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 védésre javaslom / nem javaslom<sup>1</sup>.

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

Kelt: Gödöllő, 2024. év február hó 20. nap

  
belső konzulens

---

<sup>1</sup> A megfelelő aláhúzendó.

<sup>2</sup> A megfelelő aláhúzendó.