Szakdolgozat

Angeli Imre Márton

2023

Budapesti Corvinus Egyetem

COVID-19 HALÁLOZÁSOK A TÁRSBETEGSÉGEK TÜKRÉBEN

Készítette: Angeli Imre Márton

Gazdaságinformatikus BSc Szak

2023

Konzulens: Molnár Géza

Tartalom

1 Bevezetés

A témaválasztás mellett elsősorban az motivált, hogy a digitálisan tárolt adatmennyiség robbanásszerű növekedésének hála napjainkban egyre inkább meghatározóvá váltak az adatokban rejlő értékek kiaknázására irányuló törekvések. A Big Data térnyerése számos adatfeldolgozáshoz kapcsolódó technológia megjelenését hozta magával, melyek segítségével nem csupán a sokszor strukturálatlan információ tömeg hatékony mozgatása, tisztítása, hanem az adatokban rejlő mintázatok felfedezése, előrejelzések készítése, általános konklúziók megalkotása is lehetővé vált.

A magasszintű programozási nyelvek, szoftverek, technológiák elterjedése a legkülönbözőbb tudományterületek számára kínál hatékony adatfeldolgozást, mely segítségével egy automatizált folyamat révén általános tendenciákat, trendeket fogalmazhatunk meg egy felvetett probléma kapcsán. Különböző vizualizációs eszközök, diagrammok, dashboardok, kimutatások segítségével érzékeltethetjük a változók közötti összefüggéseket, amelyek mind az értelmezést, mind a megalapozott döntéshozatalt elősegítik.

Az adatok másodlagos feldolgozására irányuló növekvő igény mind az orvostudományban, mind a demográfiában tetten érhető. Az információtechnológiai fejlesztések forradalmi változásokat hoztak mindkét területen. Az adatok gyűjtése, elemzése és értelmezése által az orvosok és kutatók új lehetőségekhez jutottak a kezelések optimalizálásában és a betegségek megértésében. A demográfiai változók vizsgálatához szintén jelentős eszköztár áll rendelkezésünkre a nyilvánosan hozzáférhető adatbázisok, Excel-táblák, illetve különböző strukturálatlan adatkészletek formájában.

Kutatásomban ennek a két tudományterületnek a közös metszetét veszem górcső alá. A két terület együttes vizsgálatának kulcsfontosságú szerep jutott a 2019-ben elinduló koronavírus világjárvány idején, melynek során kiemelt figyelem irányult a statisztikai módszertanok alkalmazására, melyek segítségével pontos képet kaphattunk a vírus által leginkább veszélyeztetett csoportok kórképi jellemzőiről, koreloszlásáról.

Az elmúlt két év alatt a COVID-19 világjárvány súlyos kihívás elé állította a világ egészségügyi rendszereit és társadalmait. A pandémia által leginkább érintettek között számos demográfiai és egészségügyi tényező játszott szerepet, befolyásolva a halálozási rátát és az egészségügyi rendszerek terhelését.

A kormány által közzétett hivatalos médiatartalmakban, valamint egyéb tudományos sajtótermékekben közölt álláspontok alapján leginkább az idős, egy vagy több krónikus

betegséggel rendelkező személyek alkották a legveszélyeztetettebb csoportot a koronavírushoz köthető halálozások tekintetében. (Hornyák, 2020)

Számos cikket publikáltak különböző internetes platformokon, melyek a magasvérnyomás, szívbetegség, cukorbetegség által érintett csoportokban fellelhető komorbiditást hangsúlyozták. (Müller: A magas vérnyomásban szenvedőknél súlyosabb a koronavírus, 2020)

Ezenkívül a kóros elhízás a fiatalok számára is komoly veszélyt jelentett a pandémia idején, és pozitív irányú kapcsolatot mutatott a koronavírusban való elhalálozás valószínűségével. (Paharia, 2023)

Kutatásom során kiemelt figyelmet szánok ezen betegségek többlethalálozást eredményező hatására a hazai koronavírus adatokat alapul véve.

Fontos ugyanakkor megjegyezni, hogy számos egyéb tényező is befolyásolta a halálozási adatok alakulását, többek között a szociális helyzet, a régió egészségügyi felszereltsége is szignifikáns hatással bírt a vírussal szembeni ellenállóképességre. (U.S. poor died at much higher rate from COVID than rich, report says, 2022)

Ezek az adatok ugyanakkor nehézkesen mérhetők kvantitatívan, illetve ilyen részletességgel nem találhatóak leírások a Covid 19-cel kapcsolatban elhunyt betegek statisztikájában. Ebből kifolyólag a kutatás nem tér ki ezen változók vizsgálatára.

A dolgozat célja a hazai COVID halálozások összefüggéseinek vizsgálata, a már említett tényezők befolyásoló szerepének valós, elérhető adatállományok feldolgozásának, analizálásának segítségével történő alátámasztása, illetve további kapcsolatok felderítése. Elsősorban a társbetegségek befolyásoló hatására tér ki a kutatás.

Az adatok analizálása ugyanakkor jelentős problémákat vet fel, hiszen strukturálatlan, gyakran helyesírási hibákat tartalmazó adatok állnak rendelkezésünkre, így számos esetben ugyanazon diagnózist egymástól különböző karakterláncok jelölik. A magasvérnyomás betegség például több mint 30 féle különböző formában fordul elő. Ezek az elnevezések az emberi gondolkodás számára könnyen értelmezhetők, az azonosságok hamar felismerhetők. Ugyanakkor a nagyméretű adatállományok automatizált feldolgozásához használt számítógépes technológiák nem ismerik fel ilyen egyszerűen a különböző formájú adatokban rejlő megegyező információtartalmat. Ebből kifolyólag a kutatás első részében az adatok tisztítása, klaszterezése, egységes formátumok kialakítása kerül előtérbe, melynek célja egy olyan

helyesírási hibáktól, redundanciától mentes adatállomány kialakítása, amely a további elemzések, modellezések alapjául szolgálhat.

Az egyes diagnózisok tényleges hatásának vizsgálatához ugyanakkor nem elég csupán a társbetegségként való előfordulási gyakoriságokat elemezni. Ahhoz, hogy pontos képet kaphassunk egy betegség veszélyességéről számításba kell vennünk a népességben vett előfordulását is. Ennek meghatározása ugyanakkor problémákat vet fel, hiszen nem érhető el olyan nyilvános, letölthető adatbázis, amely személyenként lebontva tárolná valamennyi magyar állampolgár kórképének részletes leírását. Annak érdekében, hogy közelítőleges becslést kaphassak egy betegség előfordulására, a diagnózis kezelésére alkalmazott éves hatóanyagforgalmat vizsgálom a kutatás során, és a megfelelő táblakapcsolatok alkalmazásával megjelenítem a becsült relatív gyakoriságukat a teljes népesség körében.

Miután a további transzformációk révén kialakított standardizált táblázatok is rendelkezésemre állak, elkészítem azon vizualizációkat és modelleket, melyek bemutatják a nemre, korra, társbetegségre vonatkozó kapcsolatokat, és összevetik azokat az előzetesen feltételezett összefüggésekkel.

2 Elméleti áttekintés

Ebben a fejezetben bemutatom a kutatás során érintett témakörök elméleti hátterét. Körbejárom az adatok importálása szempontjából releváns webbányászat témakörét, illetve az az adattisztítás megvalósítása során alkalmazott természetes nyelvfeldolgozás eszköztárát. Tárgyalásra kerül továbbá az adatszabványok kialakításához, az adatminőség javításához kapcsolódó Levenshtein távolság, illetve a hierarchikus klaszterezés is.

2.1 Webbányászat

Az elemzésem szempontjából releváns adatok az archivált kormányzati oldalról érhetőek el. Ahhoz azonban, hogy ezekből az adatokból analizálásra alkalmas struktúrát alakítsak ki, szükségem volt a webbányászat eszközeinek használatára.

A módszer lényege, hogy a HTML felépítésű weboldalak strukturálatlan formában megjelenő adataiból egy könnyen kezelhető adatkészletet kapjunk.

A folyamat során az adatbányász program közvetlenül hozzáfér a teljes világhálón elérhető tartalmakhoz a http protokollon keresztül, vagy egy webböngésző használatával. A program futtatása közben az adatfeldolgozás automatizáltan, nagy sebességgel történik, és a szoftver néhány másodperc alatt több gigabyte-nyi adatot képes a web-ről összegyűjteni, feldolgozni, és egy központi adattárházban eltárolni visszakereshető formában. (Gonda, 2009)

A folyamat magában foglalja a weboldal letöltését egy ún. request (kérés) elindításával, valamint a tartalom kinyerését az egyes HTML elemekből. Jellemzően egy-egy specifikus HTML tag belső tartalmára fókuszálnak további elemzések céljából. (Lawson, 2015)

Legelterjedtebb felhasználási területei a következők:

- webindexelés
- adatbányászat
- ár összehasonlítás
- ingatlanhirdetések összegyűjtése

Napjainkban, a big data térnyerésének köszönhetően virágkorukat élik a webbányászathoz köthető módszertanok, melyeknek szerepe a társadalomtudományokban is tetten érhető. A kvantitatív szövegelemzést a nemzetközi társadalomtudomány egyik leggyorsabban fejlődő húzóágazatának tartják. (Czinkóczi, 2018)

2.2 Természetes nyelvfeldolgozás

A természetes nyelvfeldolgozást (röviden NLP) egyaránt besorolhatjuk a nyelvészet, az informatika, valamint a mesterséges intelligencia részterületei közé. A folyamat lényege, hogy számítógép segítségével minél sikeresebben képesek legyünk kiaknázni a szavakban, mondatokban, szövegrészletekben, vagy akár az emberi beszédben rejlő információmennyiséget. (Dudás, 2011)

Ehhez elsősorban a számítógép, illetve az emberi nyelv interakciójának fejlesztésére van szükség, melynek segítségével a számítógép valós időben képes strukturálatlan felépítésű szövegszerkezeteket értelmezni, feldolgozni, illetve azokhoz jelentéstartalmat társítani. A természetes nyelvi feldolgozás nem csupán a szigorú nyelvtani szabályok ellenőrzésére korlátozódik, ennél jóval összetettebb folyamatról van szó. Ahhoz, hogy minél inkább az emberi gondolkodáshoz hasonló szövegértelmezési mechanizmusokat alkothassunk, legalább ugyanilyen fontos a szövegkörnyezet, kontextus felismertetése. (Copestake, 2004)

Esetünkben a szóhasonlóságok alapján történő betegség elírások kiküszöbölése volt a releváns feladat, amely az emberi gondolkodás alapján könnyen kivitelezhető lenne, azonban a gépi feldolgozás szempontjából számos kihívást rejt magában.

Az NLP célja tehát olyan algoritmusok kialakítása, melyek révén a számítógép képes feldolgozni mind az egyes nyelvi elemek, mind a teljes dokumentumok tatalmát. Mindezek révén elősegítjük az adatállományok összehasonlításának, kategorizálásának, illetve rendszerezésének automatizálását. A természetes nyelvi feldolgozás az interneten elérhető információmennyiség robbanásszerű növekedésének következtében virágkorát éli. Nélkülözhetetlen szerepet tölt be az adatelemzés területén belül, melynek oka, hogy napjainkban a digitálisan elérhető adatállomány 70-90%-a az explicit programozás számára értelmezhetetlen, strukturálatlan formában jelenik meg. (Barabás, 2013)

A mesterséges intelligenciának számos felhasználási területe ismert, melyek közül a legelterjedtebbek a következők:

- Szöveges besorolás
- Beszédfelismerés
- Chatbotok
- Gépi fordítás
- Automatikus szövegkiegészítés
- Helyesírás-ellenőrzés

- Spam-felismerés

A természetes nyelvi feldolgozás során legtöbbször különböző adattisztításhoz kapcsolódó metódust alkalmazunk a szabadszöveges inputokon annak érdekében, hogy lecsökkentsük az adatokban rejlő redundanciát. Ezek közül a módszerek közül a legelterjedtebbek a következők:

- szegmentálás:
  - Egy adott szöveget mondatok tömbjévé alakítunk
- tokenizálás:
  - A mondatból vagy szószerkezetből szavakat képzünk
- stop szavak:
  - A jelentéstartalom megállapításának szempontjából kevésbé releváns szavak, például névelők, kötőszavak kiszűrése
- lemmatizálás:
  - Egy szó alapformájának kinyerése

2.3 Levenshtein távolság

Pythonban a karakterláncok hasonlóságának leírására kiválóan alkalmas a Levenshtein távolság. A mutató értéke azt közli, hogy minimum hány karakter módosítást kell végeznünk ahhoz, hogy az egyik string tökéletesen egyezzen a másikkal. (Navarro, 2001)

Ezek a módosítások a következők lehetnek:

- karakter beszúrása
- karakter törlése
- karakter megváltoztatása

Az algoritmus lényege, hogy a két összehasonlítandó szóból egy $n \times m$-es mátrixot képzünk, ahol m az egyik, n a másik karakterlánc hossza. Az (x, y) cellában mérjük a Levenshtein távolságot az első szó x-edik karakteréig tartó része, illetve a második szó 0. és y. karaktere által körül határolt szöveg között. (Navarro, 2001)

A mátrix a bal felsőtől a jobb alsó sarkáig tölthető. Minden művelethez, amely minimálisan szükséges a két rész-kartakterlánc azonosságához, költséget rendelünk, amelynek értéke attól függően, hogy történik e módosítás 0, vagy 1 lehet. Az egyes módosításokkor (a vízszintes tengely szempontjából) a következő lépéseket alkalmazzuk:

1. Törlés: Vízszintesen jobbra ugrunk

2. Beszúrás: Függőlegesen lefelé haladunk
3. Csere: Átlósan jobbra lefelé mozgunk
4. Nincs változás: Átlósan jobbra lefelé mozgunk

	I	N	F	A	R	K	T	U	S
I	0	1	2	3	4	5	6	7	8
N	1	0	1	2	3	4	5	6	7
R	2	1	1	2	2	3	4	5	6
F	3	2	1	2	3	3	4	5	6
T	4	3	2	2	3	4	3	4	5
U	5	4	3	3	3	4	4	3	4
S	6	5	4	4	4	4	5	4	3

1. táblázat: Levenshtein távolság-saját szerkesztés

Az 1. táblázat alapján azt láthatjuk, hogy az első két karakter esetében semmilyen változás nem indokolt. Amikor azonban a 3. betűhöz érünk, már szükségünk van egy cserére. Az ezt követő két karakter bevonása következtében pedig már a törlés is elkerülhetetlen műveletté válik. Amint a „*K*” karakterhez iterálunk ismételten szükségessé válik a csere, ezt követően azonban jól látszik, hogy a költségek kumulált összege nem módosul, miközben továbbra is átlósan jobbra lefelé haladunk, tehát további transzformáció végrehajtása nem indokolt.

2.4 Hierarchikus klaszterezés

Az egyes betegségek hasonlóságát nem felügyelt gépi tanulási eljárással is ellenőrizhetjük. Azért célszerű ennek a módszernek az alkalmazása, mivel nem állnak rendelkezésünkre előre definiált címkék, ehelyett más megoldást kell alkalmaznunk a minél homogénebb csoportok kialakításához.

Ehhez a legcélszerűbb a hierarchikus elvű agglomeratív klaszterezés alkalmazása. Ez egy ún. bottom-up eljárás, melynek lényege, hogy kezdetben minden egyes elem külön-külön, saját klasztert alkot, majd minden egyes iteráció során összerendeljük a két legközelebbi klasztert. Első lépésben tehát létrehozzuk az első „összetett” klaszterünket, eggyel csökkentve ezáltal a csoportok számát. Ezt követően minden további iteráció során folytatódik ez a tendencia egyre nagyobb klasztereket létrehozva. (Nielsen, 8. Hierarchical Clustering, 2016)

A legfontosabb különbség a k közép algoritmushoz képest (amely szintén egy homogén csoportok kialakítását célzó nem felügyelt gépi tanulási algoritmus), hogy nem szükséges előre megadni a klaszterek (vagy centroidok) számát, hanem egy meghatározott klaszterszám elérését

adjuk meg leállási feltételként. Amennyiben ezt nem tesszük meg, az algoritmus akkor ér véget, amikor valamennyi megfigyelés egyetlen klaszterbe sorolódik. (Schubert, 2021)

Az eljárás során érdemes dendogramot használnunk. Ez egy vizualizációs eszköz, amely kiválóan szemlélteti az egyes csoportok összevonását. Ennek segítségével meg tudjuk állapítani, hogy mi az a klaszterszám, melynek esetében a kialakult csoportok már kellő mértékben elkülönülnek egymástól, így ez alapján tudunk optimális klaszterszámot választani. (Kovács, 2014)

A távolság mértékének alapja legtöbb esetben egy ún. közelségi mátrix. Ez egy szimmetrikus adatszerkezet, melyben mind a sorok, mind az oszlopok száma megegyezik a vizsgálandó megfigyelések számával. A klaszterek közötti különbséget számos módszerrel leírhatjuk. Ezek közül a legelterjedtebbek a következők:

- **Egyszerű kapcsolás módszere:**

  Ez a módszer a különböző klaszterekben található, egymáshoz legközelebb eső két pont alapján történő összevonást jelenti (Fülöp, 2019)

- **Összetett kapcsolás módszere:**
  A különböző klaszterekben található, egymástól legtávolabb eső pont pár között mért távolság alapján végzi az összevonásokat. (Fülöp, 2019)
- **Csoportátlag módszer**:
  A klaszter-távolságok meghatározásakor az egymástól különböző csoportokból vett összes pontpár távolságának átlagát veszi számításba (Fülöp, 2019)
- **Ward módszer:**
  A csoportátlaghoz hasonlóan a klasztereket itt is a középpontjukkal jellemezzük, ugyanakkor a belső négyzetes hiba minimalizálására törekszünk az összevonásokkor. (Fülöp, 2019)

2.5 Cramer-együttható

Kutatásom során megvizsgáltam a kétféle módszer segítségével kialakított, „tisztított” oszlopok közötti konzisztenciát. Annak érdekében, hogy számszerűen is meghatározzam a két minőségi ismérv közötti asszociációs kapcsolat szoroságát, kiszámoltam a Cramer-együtthatót.

A mutatószám érteke azt fejezi ki, hogy mennyire tekinthető függetlennek egymástól a két ismérv. 0 érték esetén a két változó független egymástól, míg 1 érték esetén függvényszerű a

kapcsolat közöttük. 0,3 alatt gyenge, 0,3, illetve 0,7 között közepes, felette pedig erős összefüggésről beszélhetünk. (Cramér, 1946)

A kiszámításához a teljes függetlenségnek megfelelő elvi relatív gyakoriságokat szükséges összevetni a tényleges relatív gyakoriságokkal. (Cramér, 1946)

$$\chi^2 = \sum \frac{(f_{ij} - f_{ij}^*)^2}{f_{ij}^*}$$

1. egyenlet: Cramer-együttható kiszámítása-forrás: (Cramér, 1946)

$$C = \sqrt{\frac{\chi^2}{N * \min\{(\mathrm{r}-1);(\mathrm{c}-1)\}}}$$

2. egyenlet: Cramer-együttható kiszámítása-forrás: (Cramér, 1946)

2.6 Hoover index

A betegségek gyakoriságának vizsgálata után kiszámítottam azok koncentrációjának mértéket. Ehhez a Hoover indexet határoztam meg.

Ez az egyenlőtlenségi mutató azt határozza meg, hogy az egyik ismérv mennyiségének hány százalékát kell átcsoportosítanunk ahhoz, hogy megoszlása a másik mutató megoszlásával azonos legyen. Értékkészlete 0 és 100 között változhat. 0 érték esetén a két változó eloszlása azonosnak tekinthető, míg 100 esetén teljes szegregációról beszélhetünk. (Sapolsky, 2005)

$$H = \frac{1}{2} * \frac{\sum_i |x_i - \bar{x}|}{\sum_i x_i}$$

3. egyenlet: Hoover index kiszámítása-forrás: (Hoover, 1936)

3 Eszközök, és módszerek

Ez a fejezet betekintést nyújt a kutatás során alkalmazott módszerek használatába, kitér azok előnyeire, sajátosságaira.

3.1 Alkalmazott módszerek

Kutatásom során az ETL folyamatokra jellemző lépéssorozatot hajtottam végre, amely mind az adatkinyerést (**E**xtract), mind az adatátalakítást (**T**ransform), mind a betöltést (**L**oad) magában foglalja. Az adatimportálás, illetve adattisztítás végrehajtásához a Python programozási nyelvet használtam.

Első lépésben (ld. 1. ábra) a kormány archivált, hivatalos weboldalán elérhető, páciensenként tagolt koronavírus adatokat importáltam CSV formátumba. Ehhez a részfolyamathoz a *BeauifulSoup* csomagot használtam fel, amely elsősorban HTML, illetve XML állományok feldolgozására alkalmas.

Az adattisztítás részfolyamathoz érkezve az *nltk* csomagból elérhető, természetes nyelvfeldolgozásra használt függvénykészletet használtam. A metódusok alkalmazásával lehetőségem nyílt az egyes társbetegség elnevezések egységes formai szabványának kialakítására, szótövezésére, tokenizálására, illetve a stop szavak (csekély információtartalommal bíró szavak pl. kötőszavak) szűrésére.

Az adattisztítás követően továbbra is fennálló, legtöbbször helyesírási hibák miatt kialakuló inkonzisztens elnevezések szűrésére, összevonására a Scikit Learn csomagot használtam fel, amely széleskörben alkalmazott felügyelt, és nem felügyelt gépi tanulási algoritmusokat tartalmaz. Kutatásom során a közelségi mátrixon alapuló csoportkialakításra alkalmazott hierarchikus (agglomeratív) klaszterezési módszertant alkalmaztam, hiszen a Levenshtein távolságok meghatározása nyomán kialakuló kétdimenziós tömb könnyen feldolgozható, optimális szerkezetű adatkészletet biztosított a művelet elvégzéséhez. Az algoritmus futtatását követően előállított dendogram által vizualizált összevonások alapján könnyen meghatározhatóvá vált a csoportok (ugyanazon betegséghez tartozó helyesírási variációk) száma.

Az adatátalakítások, táblaösszevonások, aggregálások révén kialakított standardizált adatok vizualizálására, további modell építésre az R programozási nyelvet használtam A társbetegség, kor, illetve nem változók felhasználásával megalkotott regressziós modellek felépítéséhez a beépített *lm* függvényt, míg az összefüggések vizualizálására a *ggplot* csomagot használtam,

3.2 A kutatás felépítése

Kutatásom során az igazoltan koronavírus fertőzés következtében elhunyt hazai népesség interneten elérhető adatállományát dolgoztam fel. Mivel ez az adatszerkezet jelentős mennyiségű rekordot tartalmaz, törekedtem az adatokban rejlő érték kiaknázására. A kormányzati oldalról elérhetőek ugyan az egyes társbetegségek, ugyanakkor eltérő helyesírásokkal jelennek meg. Ezek egységesítésére volt már korábban kutatási projekt, melynek során *regex* kifejezésekkel próbálták meg összevonni a különböző írásmódokat. (Ferenci, 2022)

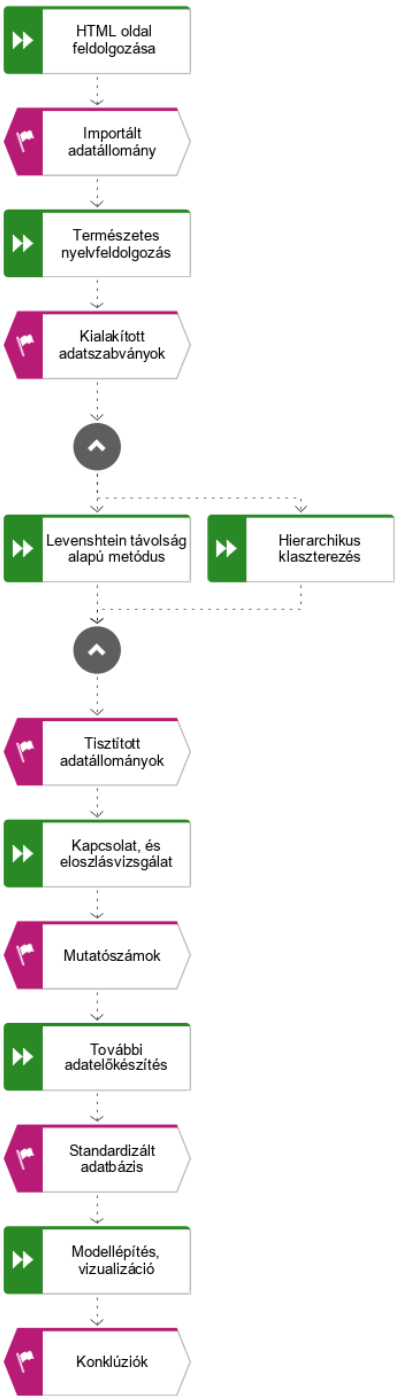
Ennek a módszernek azonban jelentős korlátja, hogy szükséges előre definiálni az egységesítés szabályait. Kutatásom során igyekeztem kiküszöbölni a hasonló jellegű problémákat, és szövegbányászati, valamint nem felügyelt gépi tanulási eljárással vizsgáltam meg az eltérő írásmódokból eredő variációk egységesítési lehetőségeit, és ezen egységesített társbetegség megadás alapján vizsgáltam, hogy a halálozások milyen összefüggést mutatnak a társbetegségekkel különböző dimenziókban.

Az első fejezetben az adatok importálását, illetve tisztítását végeztem el. A szükséges adatokat a hivatalos kormányzati weboldalról importáltam elemzésre alkalmas, feldolgozható állományba. Ezt követően a természetes nyelvi feldolgozás eszközeinek segítségével egységes formai szabályokat igyekeztem kialakítani, melyek segítségével, immár egy strukturált adatállományon lehetett elvégezni a további elemzéseket.

Annak érdekében, hogy a további elírásokat, helyesírási variációkat is egységesen kezelhessük, kétféle különböző klaszterezési eljárást alkalmaztam az adathalmazon a második fejezetben. Egy önállóan megírt metódus segítségével olyan csoportok kerültek kialakításra, melyekben legalább részleges hasonlóságot mutatnak az egyes diagnózisok (pl hasonló szavak fordulnak bennük elő). Egy másik megközelítés alkalmazása során nem felügyelt gépi tanulási eljárással igyekeztem elérni, hogy javarészt konzisztens elemekből álló csoportok alakulhassanak ki, csökkentve ezáltal a redundanciát.

Miután immár tisztított adatok álltak rendelkezésemre, kialakítottam azokat a táblázatokat, melyek a további elemzések alapjául szolgáltak. Többek között szükségem volt egy gyakorisági táblára, amely az egyes diagnózisok Covid társbetegség szerinti, illetve a teljes társadalomban mért becsült előfordulási gyakoriságát tartalmazza. Ennek kialakításához többféle adatforrás összekapcsolására volt szükségem.

Az utolsó fejezetben diagrammok, táblázatok, valamint statisztikai mutatók segítségével vizsgáltam meg a nem, kor, és diagnózisok összefüggéseit, illetve az egyes ismérvek közötti kapcsolatok szorosságát. A különböző vizualizációs eszközök (pontdiagrammok, oszlopdiagrammok, dobozábrák) révén számos következtetést vonhatunk le a rendelkezésünkre álló adatokból.



1. ábra: folyamatábra-saját szerkesztés

3.3 Eszköz választás

Az adatfeldolgozáshoz napjainkban többféle különböző technológia áll rendelkezésünkre, melyek hatékony megoldásokat kínálnak adatfeldolgozási, illetve adattisztítási folyamatok automatizálására.

Hasonló problémák kezelésére a legelterjedtebb eszközt az utóbbi évtizedekben a Microsoft Excel jelentette. Napjainkra a táblázatkezelő program az asztali alkalmazások egyik legnépszerűbb szoftverévé vált, csupán a szövegszerkesztők rendelkeznek szélesebb körű felhasználói bázissal. A program lehetővé teszi a felhasználó számára, hogy egyetlen munkafüzet használatával könnyen átlátható jelentéseket, formázott táblázatokat, diagrammokat készítsen. Ezenkívül az Excel magában foglalja mindazon matematikai, statisztikai képleteket, módszertanokat, melyeket különböző kutatások során széles körben alkalmaznak. Könnyű kezelhetősége, sokszínű eszköztára miatt az üzleti élet meghatározó szereplőjévé vált. (Berk & Carey, 2007)

Bár az Excel hatékony funkciókat nyújt, illetve VBA makrók alkalmazásával további automatizációs lehetőségeket biztosít, a vizualizálás, illetve az összetettebb, akár programozást igénylő feladatok terén korlátozott lehetőségeket kínál.

A strukturált formában elérhető adatok hatékony feldolgozására, elemzésére kiváló lehetőséget biztosít az SQL nyelv alkalmazása. A lekérdező nyelv segítségével könnyen kezelhetjük a relációs adatbázisrendszerekben (RDBMS) előforduló adatokat. A technológia egyaránt lehetőséget biztosít az adatok szelekciójára, projekciójára, továbbá az egyes rekordok módosítása, törlése, beszúrása is hatékonyan megoldható ezzel a technolósgiával (CRUD műveletek). Bár az SQL szabványosított nyelvi elemeket kínál az adatok lekérdezésére, illetve manipulálására, a különböző adatbázisrendszerek (pl. Oracle, Microsoft SQL Server, MySQL) a nyelv eltérő verzióit használják. (Hartigan, 1975)

Az RDBMS-ek mind az adatelemzés, mind a szoftverfejlesztés szempontjából széles felhasználói bázissal rendelkeznek, ugyanakkor jelentős kihívást eredményez az adatok formai kötöttsége (relációs adatokhoz tervezték), a drága skálázhatóság, továbbá a jelentős adattömeget érintő összetett lekérdezések megemelkedett erőforrásigénye. (McQuillan, 2015)

Napjainkban a nagyvállalatok esetén a Power BI a piacvezető adatelemző szoftverek közé tartozik. A platform széleskörű üzleti intelligencia megoldásokat kínál a felhasználók számára, melyek segítségével hatékonyan kivitelezhetők a különböző adattranszformációk, modellezések, illetve az akár interaktív vizualizációk is. További előnyt jelent, hogy könnyen

importálhatók adatok a legkülönbözőbb adatforrásokból is (pl. SQL server, Excel tábla, szövegfájlok), emellett az Excelhez hasonlóan egy könnyen kezelhető felhasználói felülettel rendelkezik. (Ferrari & Russo, 2016)

A Power BI-hoz hasonlóan a Tableau szintén széles körű támogatottsággal rendelkezik nagyvállalati környezetben. Segítségével komplex adatelemzési folyamatokat lehet hatékonyan kezelni, továbbá jelentős eszközkészletet biztosít dinamikus vizualizációk, dashboardok készítéséhez. (Gupta, Pinto, Sankhe-Savale, Gillet, & Cherven, 2022)

Összességében elmondható, hogy az előzőkben bemutatott mindkét szoftver (Power BI és Tableau) számos hasznos funkciót biztosít az adatok analizálásához, ugyanakkor kevésbé rugalmasak adattranszformációk területén a Python, illetve az R programozási nyelvekhez képest, melyeket valamennyi adatkezelési folyamathoz alkalmazhatunk, összetettségtől függetlenül.

A teljes adatfeldolgozási folyamat végrehajtásához a Python programozási nyelvet használtam, amely amellett, hogy egy könnyen olvasható és értelmezhető, felhasználóbarát szintaxissal bír, kiválóan alkalmas nagyméretű adatarchitektúrák transzformálására, tisztítására, továbbá gazdag könyvtárállománnyal rendelkezik a statisztikai számítások területén. (Boschetti & Massaron, 2018)

Munkám során törekedtem a nyelv által kínált funkciók széleskörű használatára, amely valamennyi részfolyamatban tetten érhető.

Pythonban számos összetett, könnyen kezelhető adatszerkezet áll rendelkezésre, melyek a legkülönbözőbb problémákra nyújtanak megoldást. A pandas könyvtárból elérhető DataFramek az adatokat táblázatoknak megfelelő formátumban tárolják, illetve lehetőséget biztosítanak az adatok szelekciójára, projekciójára, aggregálására, illetve különböző adattáblák összekapcsolására is. (Boschetti & Massaron, 2018)

Ezenkívül a társbetegség elnevezések analizálásakor több esetben használtam listákat, halmazokat, tömböket, szótárakat is, melyek tartalma szintén rugalmasan kezelhető, illetve számos metódus áll rendelkezésre különböző műveletek elvégzésére, adatok lekérdezésére.

A Python programozási nyelv számos könyvtárat kínál különböző adatelemzési folyamatokhoz. Kutatásom során a következő csomagokat használtam fel:

- Beautiful soup

- Levenshtein
- Scikit learn
- NLTK

A Beautiful soup csomagot széles körben használják különböző web- és szövegbányászatra épülő projektekben. A könyvtár használatának segítségével lehetőségem nyílt a szabályos, táblázatos struktúrában elérhető adatok gyors, és hatékony feldolgozására. Gyakori jelenség, hogy a *Selenium* csomaggal kombináltan használják a fejlesztők, illetve adatelemzők, ezáltal nem csupán a statikus weboldalak HTML állományát nyerik ki, hanem a különböző JavaScript események következtében létrejövő dinamikusan megjelenő tartalmakat is képesek nyomon követni, automatizáltan feldolgozni. Kutatásom során ugyanakkor nem volt szükséges ennek a technológiának az alkalmazása, a *Beatiful Soup* csomag önmagában való használata biztosította a különböző HTML tag-ek által körül határolt, számomra releváns tartalom importálását.

A Scikit learn csomag használatának segítségével könnyen felfedezhetők az adatokban rejlő mintázatok, összefüggések, illetve lehetőséget biztosít klaszterek meghatározására különböző változók függvényében. A könyvtár egyaránt tartalmaz felügyelt, illetve nem felügyelt gépi tanulási algoritmusokat.

Az NLTK könyvtár magában foglalja mindazon metódusokat, melyek a 2.2. fejezetben kerültek tárgyalásra.

A Python nyelv különböző metódusokat biztosít a szövegek közötti hasonlóságok vizsgálatára is. A legelterjedtebb algoritmusok közé tartozik a *Cosine Similarity,* amely két különböző dokumentum (karakterláncokból álló vektorok) közötti hasonlóságot mutatja meg, miután az egyes vektorelemekhez hozzárendeltük azok *TF-IDF* indexét. Ez a szám azt fejezi ki, hogy mennyire gyakori az adott szó a vektorban (TF), illetve mennyire ritka az előfordulása az összes dokumentumban (IDF). (Zhu, 2023)

Ez a megoldás ugyanakkor elsősorban több szóból álló listák közötti hasonlóság megállapítására használható. Ahhoz, hogy kutatásom során az egyes társbetegségekből klasztereket készíthessek a Levenshtein távolságot használtam, amely a szavakban előforduló karakterek eltéréséből számítja a megadott inputok közötti hasonlóságot a 2.3. fejezetben tárgyalt módszer alapján.

A modellépítéshez, illetve vizualizáció készítéshez az R programnyelvet használtam, amely a Pythonhoz hasonlóan egy nyílt forráskódú, adatelemzési folyamatokban széles körben elterjedt

környezet, továbbá számos külső könyvtárral rendelkezik statisztikai számítások elvégzéséhez, diagrammok készítéséhez.

A beépített *lm* függvény könnyen értelmezhető, olvasható objektum formájában mutatja be többek között a modellben használt magyarázó változók koefficiensét, próbafüggvényét, p-értékét, illetve az R^2 értéket is.

A *ggplot* csomag egyszerűen átlátható szintaxisának segítségével lehetővé teszi részletgazdag grafikonok készítését. A diagrammok elkészítéséhez egy könnyen olvasható, és értelmezhető réteges struktúrát használ, amely egyértelműen leírja az ábrázolni kívánt objektum valamennyi tulajdonságát (diagrammtípus, tengelyfeliratok, színek, jelmagyarázat, vonalvastagság stb.).

4 Gyakorlati kivitelezés

Ebben a fejezetben bemutatom az ETL folyamatok által definiált lépéssorozat végrehajtását, az adatok előkészítésétől kezdve a további elemzésekhez szükséges standardizált adatszerkezetek megalkotásáig. Ezenkívül bemutatásra kerülnek a következtetések levonásához szükséges vizualizációk, regressziós modellek is. Az egyes lépésekhez készített Python, illetve R forráskódok az alábbi linken érhetőek el: https://github.com/angeliimre/szakdolgozat

4.1 Adatok előkészítése

Ebben az alfejezetben tárgyalásra kerülnek a további elemzések szempontjából releváns importálások, adattranszformációk.

4.1.1 Adatok importálása

Kutatásom során a *beautiful soup* csomagot használtam fel a kormányzati weboldalon elérhető adatok kinyerésére, mely kiválóan alkalmas HTML fájlok feldolgozására.

Mivel az elhunytak adatai nem csupán egy, hanem több oldalon helyezkednek el, ezért nem volt elég egyetlen URL címet megadni *request*-ként. Ugyanakkor mivel az oldalak közötti különbséget csupán egy egyedi, növekvő sorrendiséget követő számozás jelenti, ezért egy *for* ciklus segítségével könnyen kinyerhető volt az összes link tartalma. Az egyes ciklusléptések során az aktuális ciklusváltozónak megfelelő URL címen elhelyezkedő tartalom került kiértékelésre. A HTML forrásban megfigyelhető, hogy a releváns tartalom <*table*> tagek között helyezkedik el, tehát az oldalak vázát egy-egy táblázat alkotja. Ebből adódik, hogy a feldolgozás szempontjából a táblasorokon (<*tr*>) belül elhelyezkedő <*td*> tagek közötti szöveges állomány volt a releváns. Egy sorban 4 <*td*> tag található, melyekben rendre a beteg egyedi azonosítója, neme, kora, illetve a tásbetegségei találhatóak. Utóbbiak egyetlen <*td*> tagben szerepelnek egy vesszővel, illetve egy szóközzel elválasztva.

Első körben a megfelelő *class* név alapján azonosítható táblázatot volt szükséges megtalálni a programnak a *find* metódus segítségével. Ezt követően az ebben szereplő egyes táblázatsorokon (*tr*) iteráltam végig, majd pedig egy belső ciklus segítségével, a sorokat alkotó *td*-k tartalmát dolgoztam fel.

A kutatás későbbi lépéseinek elvégzéséhez fontos volt, hogy az eredeti táblázatos formátumot meg tudjam őrizni a CSV állományban is, ezért szükséges volt megoldani, hogy az ugyanazon rekordhoz tartozó adatok a vesszővel tagolt fájlban is egy sort alkossanak. Éppen ezért amint az aktuális táblasorhoz ért a ciklus, létrehoztam egy *string* típusú változót, amelyhez pontosvesszővel folyamatosan hozzáfűztem a benne szereplő táblaadatok tartalmát. Ezt a

szöveges változót exportáltam a CSV fájlba, miután eltávolítottam az utolsó karaktert jelentő pontosvesszőt, és helyette megadtam a sortörést szimbolizáló „\n” karaktert, illetve az egyedi magánhangzó jelöléseket megfelelő karakterekre cseréltem. Előfordul ugyanis, hogy a magyar ékezetes karakterek nem a megszokott módon vannak jelölve és ez sok esetben az eltérő kódolás miatt problémákhoz vezethet.

Az immáron könnyebb feldolgozást lehetővé tevő, CSV formátumban elérhető adatokat a *pandas* csomag segítségével *DataFrame*-mé konvertáltam. Ezt követően ezt az összetett adatstruktúrát használtam fel a további elemzésekhez.

4.1.2 Adatminőség kezelése

A következő fejezetekben a negyedik oszlop adatait (társbetegségek) fogom elemezni a természetes nyelvfeldolgozás módszereinek segítségével. Kutatásom során többféle NLP eljárást is alkalmaztam az egyes társbetegségek egységes kezelésének kialakítására.

Első körben feltártam, hogy milyen egyedi adataink vannak, annak érdekében, hogy pontosabb képet kaphassak az értékkészletről. A countVectorizer metódus segítségével a tokenizált adatokból egy szótárat készítettem. A dictionary kulcs értékeit alapul véve egy olyan listát hoztam létre, amely minden lehetséges értéket pontosan egyszer tartalmaz. A tokenizer paraméter alapértelmezetten igen sokféle karaktert tekint szeparátornak, amely esetünkben nem minden esetben lehet optimális. Többek között azért sem, mivel a szóköz is egy elválasztó karakterként van értelmezve, ugyanakkor jó néhány, több szóból álló betegséget egységesen szeretnénk kezelni. Ebből adódóan ebben a szituációban célszerűbb egy olyan függvényt létrehozni, melynek segítségével saját magunk határozhatjuk meg a határoló stringeket, ezáltal egységesen kezelhetjük azokat a rekodokat is, melyek esetében egy-egy kötőszó jelenti a szeparátort.

A könnyebb értelmezhetőség, és a redundancia csökkentése érdekében fontos, hogy meghatározzuk az adatszabványokat, melyek révén az outputokat immáron egységes formai szabályok szerint kezelhetjük. A bemenő paraméterként megadott stringet (társbetegségek) első körben kisbetűssé alakítottam a metódus segítségével, így számos redundáns adatot kiszűrtem, amelyek a case szenzitivitásból adódnak. Ezt követően a magyar nyelvben előforduló kötőszavaktól is megtisztítottam az adatokat, majd az eltávolításuk után keletkező dupla szóközt immáron vesszőre cseréltem. Ezek után immáron egységesen tudtam kezelni a társbetegségek típusait, hiszen az elválasztó karakter minden esetben egy vessző-szóköz páros volt.

Ezt követően azokat a karaktereket is kiszűrtem, melyek az adatelemzés szempontjából nem relevánsak pl.!,:.

4.2 Társbetegség csoportok kialakítása

Ebben a fejezetben a Levenshtein távolságra alapuló eljárások kerülnek bemutatásra, melyek az egyes társbetegség kategóriák kialakítását célozzák. Az egyes módszerek hiányosságainak, szűk keresztmetszeteinek kiküszöbölése végett kétféle megközelítést is alkalmaztam, melyek két külön alfejezetben szerepelnek. Ezt követően a kapott outputokat összevetettem egymással hasonlóság szempontjából, illetve megvizsgáltam a kétféle algoritmus futtatását követően létrejövő adatszerkezetekben szereplő betegségelnevezések eloszlását.

4.2.1 Levenshetin távolság alkalmazása

A 2.3. fejezetben tárgyalt rekurzív algoritmust használtam fel az egyes betegség kifejezések közötti távolságok meghatározásához, az összevonások elvégzéséhez.

Első lépésben a *PorterStemmer* csomag *stem* metódusának segítségével a bemeneti karakterláncokat megtisztítottam a toldalékoktól. Ezt követően a Levenshtein távolságok figyelembevételével összehasonlítottam a bemeneti stringeket. Figyelni kellett ugyanakkor arra a jelenségre is, hogy jelentős hossz béli eltéréssel rendelkeznek az egyes szavak, így célszerű volt a szó hosszúságának megfelelően relatívan megadni a hibahatárokat, ebben az esetben a karakterlánc hosszának 25%-át adtam meg az eltérés maximális nagyságának.

Azért célszerű ezt a hibahatárt alkalmazni, mert tapasztalatokból kiindulva, nem tekinthető túlzottan szigorúnak, hiszen még a rövidebb elnevezésű, legalább négy karakterből álló betegségek esetén is megengedhetünk bizonyos mértékű hibahatárt, például a 4 betűs „sérv” szó esetén is 1 karakter hibát még elfogadunk. Az ennél rövidebb szavak esetén azonban nincs jelentősége ennek a hibahatárnak, csak a tökéletes egyezés az elfogadható. Ez logikus is, hiszen eléggé életszerűtlen az az eshetőség, hogy egy 3 betűs szót nagy mennyiségben hibásan, eltérően írunk le. Későbbiekben vizsgálni fogom ezt a vágási értéket konzisztencia szempontjából egy másik klaszterezési megoldás eredményével összevetve.

Amennyiben ez a feltétel teljesül a két betegséget azonosnak tekinthetjük pl „parkinsonkór”-„parkinson kór”

```
In [15]: azonos("parkinsonkór","parkinson kór")
Out[15]: True
```

2. ábra: Azonosságvizsgálat-saját szerkesztés

Következő lépésben tokenizáltam a szóközök mentén az egyes betegségeket hiszen, ha van hasonló szó a két szószerkezetben jó eséllyel hasonló jelentéssel bírnak. Ez a módszer az adatelőkészítés fejezetben alkalmazott szeparálásnál (melynek során az egyes betegségeket tekintettük külön-külön tömbelemnek) lényegesen egyszerűbb, hiszen ebben az esetben egységesen egy szóköz az elválasztó karakter, nem kell más tényezőket pl kötőszavakat figyelembe vennünk.

```
azonos("obstruktív tüdbetegség","krónikus tüdőbetegség")
```

```
True
```

3. ábra: Azonosságvizsgálat-saját szerkesztés

Ugyanakkor fontos tisztáznunk, hogy vannak olyan nagy számban előforduló szavak, melyek révén teljesen más jelentéstartalommal bíró szavak is hasonlítanak egymáshoz. Például az agyi érbetegség, illetve az agyi tumor bár ugyanannak a szervnek a rendellenességére utal, a két betegséghez mégis teljesen eltérő tünetek tartoznak, így hibás lenne őket azonosnak tekinteni. Továbbá gondot okozhatnak az olyan általános megnevezések, amelyek teljesen eltérő diagnózisokban is szerepelnek pl „kor”, „betegség”. Ebből kifolyólag a következő karakterláncokat kivontam a vizsgálatból:

- szervi megnevezések:
  - máj
  - tüdő
  - vese
  - agy
  - szív
- Nagy számban előforuló elnevezések
  - elégtelenség
  - kór
  - típusú
  - általános
  - megbetegedés
  - beteg
  - rendellenesség
  - zavar
- különböző szervekhez tartozó rendellenes jelenségek:

    - tumor
    - daganat
    - tágulat
- időre utaló megnevezések:
    - éve
    - hónapos

```
In [28]: azonos("agyi betegség","májbetegség")
Out[28]: False
```

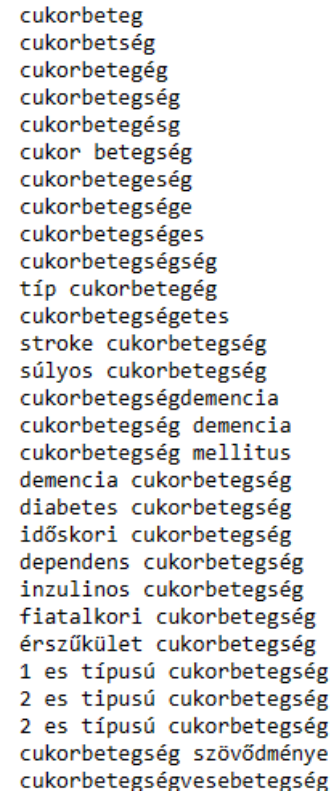
4. ábra: Azonosságvizsgálat-saját szerkesztés

A többi szóra ugyanakkor továbbra is a relatívan meghatározott Levenshtein távolság a mérvadó. Azonban ha ez nem teljesül (legtöbbször a szó rövidsége miatt), még mindig lehetséges az egy csoportba sorolás, amennyiben az egyik szó tartalmazza a másikat, pl „agyi”-„agy”. Ebben az esetben ugyanakkor fontos tisztáznunk, hogy a szó a másik szó elején (pl **szívbeteg-szívbeteg**ség) esetleg a végén szerepeljen (pl zsír**májbetegség-májbetegség**), ezáltal kiszűrjük az olyan hibákat, melyek szerint például az „**agy**”, illetve a „pajzsmirigy megn**agy**obbodás” hasonlónak minősülnek

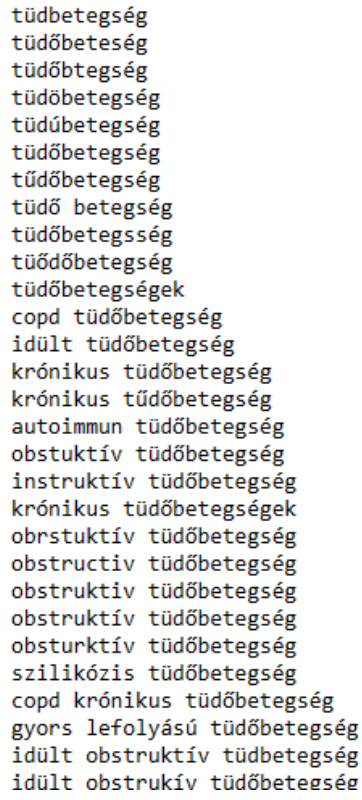
```
In [29]: azonos("agy","pajzsmirigy megnagyobbodás")
Out[29]: False
```

5. ábra: Azonosságvizsgálat-saját szerkesztés

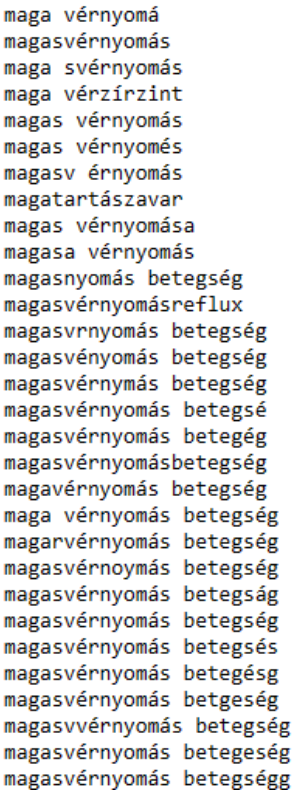
Miután az egyes szavak hasonlóságát a metódusunk segítségével már meg tudtam állapítani, megkezdtem a teljes adathalmaz szűkítését azon szavakra, melyek teljesen egyediek, és nem hasonlítanak egymáshoz. Első körben a szóhossz szempontjából módosítottam az eredeti lista sorrendjét, ezáltal az adott betegség legrövidebb változata került be az új, tisztított listába. Ennek a lépésnek a másodfajú hiba esélyének minimalizálása miatt volt jelentősége, hiszen rövidebb karakterláncok esetén kisebb eltérések megengedettek, ezáltal kisebb arányban kerülhettek különböző diagnózisok tévesen ugyanabba a csoportba. Egy *for* ciklus segítségével bejártam az eredeti listát, és csak abban az esetben adtam hozzá az egyes elemeit a szűkített listához, ha még nem szerepelt benne hozzá hasonló betegség. Ellenőrzésképp megnéztem, mely betegségek kerültek ugyanazon szűkített csoportba két egymásba ágyazott ciklus segítségével.



6. ábra: cukorbetegség variációi-saját szerkesztés



7. ábra: tüdőbetegség variációi-saját szerkesztés



8. ábra: magasvérnyomás variációi-saját szerkesztés

A csoportkiértékelés alapján az láthatjuk, hogy a legtöbb esetben az elvárásainknak megfelelően történt a klaszterek kialakítása.

Miután áttekintettem a létrejött összevonásokat egy metódus létrehozásával elősegítettem, hogy az eredeti *dataframe* társbetegség oszlopában immáron ugyanazzal a megnevezésekkel, egységesen szerepeljenek az összetartozó diagnózisok.

Ugyanakkor fontos tisztáznunk, hogy a csupán egyedi megnevezéseket tartalmazó szűkített lista a legrövidebb helyesírási variációt tartalmazza az adott betegség klaszterből. Megfigyelhetjük azonban, hogy nem minden esetben ez a verzió felel meg a nyelvtani szabályoknak. Ahhoz, hogy a legnagyobb eséllyel az elírástól mentes változatot kaphassam, az adott elem klaszterébe tartozó elnevezések közül a teljes adathalmazban legtöbbször előforduló elemet választottam egy metódus segítségével.

Az egyes rekordok tokenizálását követően a metódus segítségével minden felsorolt betegség „tisztított” változatát szerepeltetjük az egyes mezőkben, miután újra egymáshoz fűztük azokat.

	id	nem	kor	tersbetegseg	javitott
20	46246	Férfi	84.0	magasvérnyomás-betegség, iszkémiás szívbetegsé...	magasvérnyomás betegség, iszkémiás szívbetegsé...
21	46245	Férfi	93.0	iszkémiás szívbetegség, magasvérnyomás-betegség	iszkémiás szívbetegség, magasvérnyomás betegség
22	46244	Férfi	101.0	magasvérnyomás-betegség	magasvérnyomás betegség
23	46243	Férfi	90.0	cukorbetegség, magasvérnyomás-betegség, stroke...	cukorbetegség, magasvérnyomás betegség, stok, ...
24	46242	Nő	96.0	magasvérnyomás-betegség, szívbetegség, csontri...	magasvérnyomás betegség, iszkémiás szívbetegsé...
25	46241	Nő	87.0	érszűkület, magasvérnyomás-betegség, iszkémiás...	érszűkül, magasvérnyomás betegség, iszkémiás s...
26	46240	Férfi	71.0	magasvérnyomás-betegség, agyi infarktus, vesee...	magasvérnyomás betegség, agyi infarktus, króni...
27	46239	Nő	84.0	demencia, Alzheimer-kór, magasvérnyomás-betegség	demenc, alzheimer kór, magasvérnyomás betegség
28	46238	Nő	58.0	májbetegség, gyomorfekély, trombózis	májbetegség, gyomorfekély, mélyvénás trombózis
29	46237	Nő	74.0	magasvérnyomás-betegség, májbetegség	magasvérnyomás betegség, májbetegség
30	46236	Nő	72.0	magasvérnyomás-betegség, depresszió, cukorbete...	magasvérnyomás betegség, depresszió, cukorbete...
31	46235	Férfi	89.0	magasvérnyomás-betegség, krónikus lábszárfekély	magasvérnyomás betegség, gyomorfekély
32	46234	Nő	71.0	magasvérnyomás-betegség, idült obstruktív tüdő...	magasvérnyomás betegség, tüdőbetegség, mentáli...
33	46233	Nő	90.0	magasvérnyomás-betegség, vérszegénység	magasvérnyomás betegség, vérszegénység

9. ábra: Módosított DataFrame-saját szerkesztés

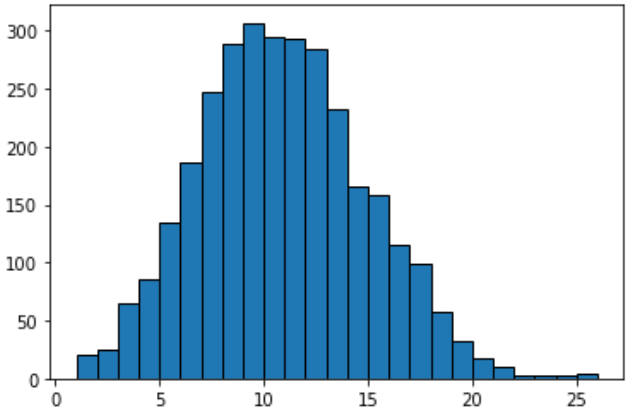
4.2.2 Hierarchikus klaszterezés alkalmazása

A 2.4. fejezetben bemutatott nem felügyelt gépi tanulási eljárást alkalmaztam a további információegyezőségek feltárásához.

A módszer alkalmazása előtt célszerű tisztáznunk, hogy ebben az esetben is a Levenshtein távolság mutató lesz az összehasonlítás alapja. Első körben létrehoztam egy mátrixot, amelynek szélessége és magassága is megegyezik az eredeti lista méretével, és amelynek minden sorában az éppen aktuális indexű listaelemtől vett távolságok szerepelnek, valamennyi listaelem esetében. Fontos megjegyezni azonban, hogy az így megadott távolságok torzíthatják a képet, hiszen a rövidebb szavak esetén jóval kisebb lesz az elírás differenciája, mint a hosszabbak esetében. Előfordulhat például, hogy míg kettő, csupán néhány karakterből álló szó még akkor is egy klaszterbe kerül, ha nincs bennük azonos karakter (pl. HIV, TBC). Annak érdekében, hogy ezt a problémát kiküszöbölhessük, kétféle megközelítést alkalmazhatunk.

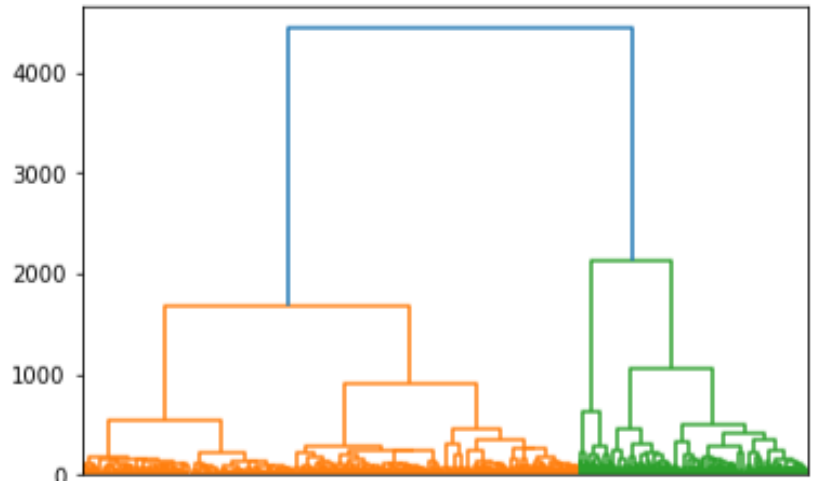
Egyfelől egy relatív távolságot adhatunk meg, amely az összehasonlítandó két szó közül a rövidebb hosszához viszonyítja a Levenshtein távolságot. Ennek a módszernek az a hátulütője, hogy túlságosan szigorúan veszi a szóhosszokat, és az esetek többségében kizárólag azonos hosszúságú karakterláncokat rendel egy csoportba. Esetünkben ugyanakkor számos elírás plusz egy karakter felesleges használatából vagy elhagyásából ered.

A második megközelítés lényege, hogy külön kezeljük azokat a szavakat, melyek egy meghatározott karakterszámot meghaladnak. A hosszúságok gyakoriságának eloszlását ábrázolva, egy normális eloszlást valószínűsítő haranggörbét kapunk. Jól látszik a hisztogramról, hogy a 10 hosszúságú karakterláncok, mint átlagos értékűek fordulnak elő a legnagyobb számban. Ebből kifolyólag ezt az értéket használtam vágási pontként.



10. ábra: szóhosszúságok eloszlása-saját szerkesztés

Következő lépésben az így létrehozott távolság mátrixok lesznek a klaszterezés alapjai. Az összevonás módszereként a „ward” eljárást alkalmaztam, hiszen ennek segítségével nagyjából hasonló méretű klasztereket kaphatunk, illetve maximalizálhatjuk a külső variacianciát. (Hartigan, 1975)



11. ábra: Dendogramm-saját szerkesztés

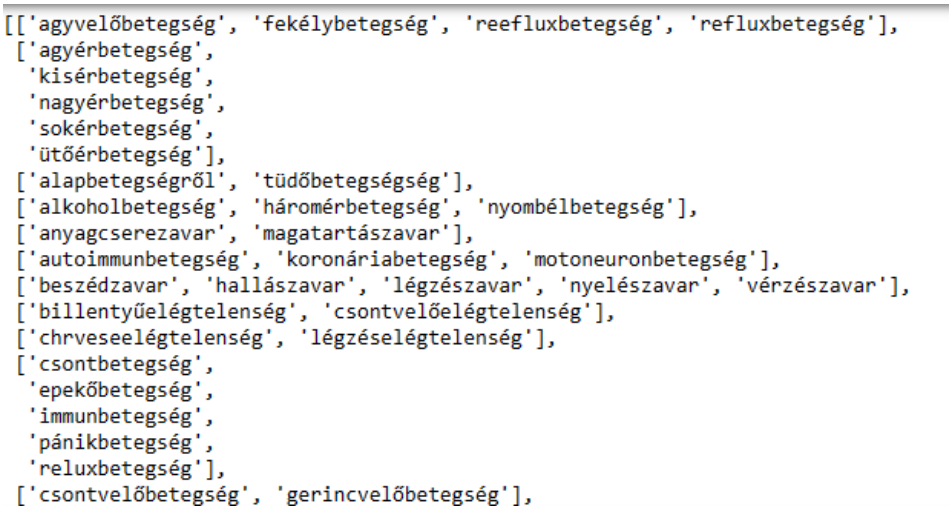
Az algoritmus futtatását követően minden lista elem kapott egy klaszterszámot. Ezeket felhasználva egy *for* ciklus segítségével összevontam az azonos klaszterszámhoz tartozó megfigyeléseket, ezáltal létrehoztam egy olyan listát melynek minden eleme egy adott klaszterbe tartozó betegségek listája.

```
array([292, 117, 117, 120,  28,  28,  45,  45, 102,  45,  45, 259,  16,
       120,  28,  45, 259,  28,  16,  16, 120,  35, 208,  16, 236, 236,
        35,  28,  28,  28, 102,  16,  16,  35,  16, 166,  35,  13,  45,
        13,  13,  16, 120,  13,  13, 236, 197, 236, 102, 236, 117, 236,
       292, 166, 292, 259, 117,  28,  28,  13,  13, 236, 236,  97,  18,
       117,  45, 170, 236, 229, 229, 228, 228,   4,  45,  21, 120, 120,
       175, 120, 120, 236,   4, 287,  18,   4,   4,  43,   4, 102, 170,
       170, 102, 102,  35,  21,  16,  16,  13,   4, 236, 228, 228,  53,
        45, 236,  16, 272,  21,  21,  34,  77, 236, 126, 126,  21,  18,
        43, 126,   8,  13,   4,  21,   4,   4, 102, 287,  35,  97, 287,
       117,   8,   8, 173, 157, 292,  53,  53, 157, 157,  35, 117, 117,
       120, 149, 208,  34,  27,  27,   8, 217, 217, 126,  82, 117,  89,
        97,  28,  28, 170, 102,  35, 117,   8, 157,  53, 166,  49,  61,
         4,  97,  49, 108,  49, 190, 190,  78,  82, 144,  20,  20,  61,
        28,  82,  49, 150,  58,  43, 115,  49,  61, 202, 202, 202,  61,
        52,  45,  45,  27,  89,  89, 102, 102,  89,  89, 102,  35, 157,
        16, 149,  35, 144, 298, 298, 157, 166, 257,  51, 101,  96,  96,
       166, 144,  28,  20,  82,  28,  45, 190,  28,  28, 175,  35,  20,
         6,  61,  97,  18,  28,  78, 129,  78,   4, 150,  58,  43,  43,
```

12. ábra: Klaszterszámok-saját szerkesztés

13. ábra: Betegségcsoportok a hierarchikus klaszterezés után-saját szerkesztés

A csoportokat figyelve szembetűnik, hogy számos esetben teljesen különböző diagnózisok kerültek ugyanazon klaszterbe, annak következtében, hogy hasonló végződéssel rendelkeznek (pl. szívbetegség és agybetegség vagy szívelégtelenség és veseelégtelenség).



14. ábra: Hibás klaszterek-saját szerkesztés

Annak érdekében, hogy ezeket a téves besorolásokat kiszűrjem, az így kialakult klaszterek elemeit egy újabb gépi tanulással csoportosítottam. Ez az eljárás csupán annyiban különbözik az előzőtől, hogy a távolságok alapja a szavak megtisztított (tehát a 2.1. fejezetben tárgyalt szórészletek eltávolítását követően kialakult) alakja.

```
------
reefluxbetegség
refluxbetegség
reluxbetegség
------
------
agyérbetegség
nagyérbetegség
------
kisérbetegség
sokérbetegség
ütőérbetegség
verőérbetegség
```

15. ábra: Javított klaszterek-saját szerkesztés

Ezt követően az eredeti társbetegség oszlopnak is elkészítettem a „klaszterezett” változatát. Ugyanúgy, mint az előző megoldásnál ebben az esetben is tokenizáltam az eredeti mezőket, majd egymáshoz fűztem az egyes betegségek klaszterében előforduló leggyakoribb elemeket, ennek az előző megoldással való konzisztencia szempontjából lesz jelentősége.

id	nem	kor	tersbetegseg	javitott	klaszterezett
46266	Nő	92.0	magasvérnyomás betegség	magasvérnyomás betegség	magasvérnyomás
46265	Nő	86.0	asztma, vastagbélgyulladás	asztma, vastagbélgyulladás	asztma, vastagbélgyulladás
46264	Férfi	80.0	agyi működés zavara,	pajzsmirigy túlműködés	agyi működés zavar
46263	Férfi	94.0	magasvérnyomás betegség, epilepszia, Parkinson...	magasvérnyomás betegség, epilepszia, parkinson...	magasvérnyomás, epilepszia, parkinson
46262	Nő	89.0	magasvérnyomás betegség, Parkinson kór	magasvérnyomás betegség, parkinson kór	magasvérnyomás, parkinson
...	...	...	...	...	...
5	Férfi	68.0	szív és érrendszeri	és érrendszeri megbetegedés	szív és érrendszeri
4	Férfi	79.0	szív és érrendszeri	és érrendszeri megbetegedés	szív és érrendszeri
3	Férfi	74.0	szív és érrendszeri	és érrendszeri megbetegedés	szív és érrendszeri
2	Nő	65.0	rosszindulatú daganat	tüdő rosszindulatú daganata	rosszindulatú daganat
1	Nő	76.0	szív és érrendszeri	és érrendszeri megbetegedés	szív és érrendszeri

16. ábra: Módosított DataFrame-saját szerkesztés

4.2.3 Kapcsolatvizsgálat

A szó legrövidebb változatának alkalmazásakor arra törekedtem, hogy a kétféle klaszterezési eljárás során hasonló outputokat kapjak. Amennyiben az újonnan létrehozott oszlopokat vizsgáljuk, azt láthatjuk, hogy a legtöbb rekord esetében gyakran megegyező eredményeket láthatunk.

Ahhoz azonban, hogy számszerűleg is tudjuk vizsgálni a két minőségi változó (oszlop) közötti asszociációs kapcsolat szorosságát, kiszámoltam a Cramer-együtthatót a *researchpy* csomag segítségével. A *crosstab* metódus futtatását követően az alábbi eredménytáblát kaptam:

	Chi-square test	results
0	Pearson Chi-square (7925659.0) =	1.495275e+07
1	p-value =	0.000000e+00
2	Cramer's V =	9.751000e-01

17. ábra: Cramer-együttható-saját szerkesztés

Az alsó sorban kiolvashatjuk a Cramer-együttható értékét, amely megközelítőleg 0,94. Ez alapján egy szoros (0,7-et meghaladó erősségű) kapcsolatról beszélhetünk.

Ez alapján tehát a két változó értékei nagyban befolyásolják egymást, a teljes függetlenségnek megfelelő relatív gyakoriságok, és a tényleges relatív gyakoriságok jelentősen eltérnek egymástól. Azt mondhatjuk tehát, hogy a két oszlop javarészt egymással konzisztens értékeket tartalmaz. Ebből kifolyólag az egyszerűség kedvéért a modellépítés, illetve a vizualizáció szempontjából csak az első eljárással előállított oszlop adatait dolgoztam fel. Jól látható ugyanis, hogy ebben az esetben valamelyest jobban körül határolható csoportokat, észszerűbb összevonásokat láthatunk, hiszen nem csupán a betegségelnevezések közötti Levenshtein távolságokkal dolgozunk (kizárva ezáltal azt a lehetőséget, hogy egy rövid, és egy relatív hosszú karakterlánc egy csoportot alkosson), hanem más faktorokat is figyelembe veszünk, pl. hasonló, vagy azonos szavak a betegség kifejezésekben, vagy az egyik betegség tartalmazza a másik betegség valamely karakterláncát.

4.2.4 Eloszlások vizsgálata

Annak érdekében, hogy megállapítsam, mekkora az egyes betegségek részesedése a Coviddal összefüggő halálozásokban, mindkét új oszlop esetében megvizsgáltam a relatív gyakoriságokat. Először a 2.1. fejezetben tárgyalt eljárás révén kialakított oszlophoz tartozó gyakorisági táblát hoztam létre.

Első körben az egyes rekordokat gyakoriságok alapján növekvő sorrendbe rendeztem.

Diagnózis	gyakorisag
2	0
agyban	0
vérben	0
zavara	0
megbetegedések	0
...	...
tüdőbetegség	4241
krónikus veseelégtelenség	4665
iszkémiás szívbetegség	9149
cukorbetegség	13354
magasvérnyomás betegség	30356

18. ábra: abszolút gyakoriságok-saját szerkesztés

Ezt követően az abszolút gyakoriságok mellett relatív gyakoriságokat is megadtam. Ahhoz azonban, hogy láthassuk, hogy az aktuális rekorddal bezárólag a megelőző elemeknek összesen mekkora a részesedésük a teljes sokaságból, a kumulált relatív gyakoriságokat is meghatároztam. Miután egy *for ciklus* segítségével megkaptam a kumulált relatív gyakoriságokat tartalmazó listát, beillesztettem a *DataFrame* új oszlopába.

	Diagnózis	gyakorisag	rel_gyak	kum_rel_gyak	rel_pos
szívizom vérellátásának zavara		313	0.002699	0.119686	0.945302
	agykárosodás	327	0.002820	0.122506	0.946704
	vastagbélgyulladás	372	0.003208	0.125714	0.948107
	mélyvénás trombózis	377	0.003251	0.128964	0.949509
	tüdőgyulladás	406	0.003501	0.132465	0.950912
	agyi érbetegség	428	0.003691	0.136156	0.952314
	magas vérzsírszint	445	0.003837	0.139993	0.953717
	depresszió	493	0.004251	0.144244	0.955119
	vesebetegség	500	0.004311	0.148556	0.956522
	alzheimer kór	502	0.004329	0.152884	0.957924
hasnyálmirigy gyulladás		512	0.004415	0.157299	0.959327

19. ábra: gyakoriságok-saját szerkesztés

A 21. ábra alapján jól látható az egyes betegségek egyenlőtlen részesedése a teljes adathalmazból. Az adatszerkezetet vizsgálva megfigyelhetjük például, hogy a gyakoriság szempontjából az alsó 95%-ba tartozó betegségek együttesen alig több mint 13%-os részesedéssel bírnak.

Diagnózis	gyakorisag	rel_gyak	kum_rel_gyak	rel_pos
szívelégtelenség	2364	0.020385	0.361809	0.987377
kóros elhízás	2697	0.023256	0.385065	0.988780
érelmeszesedés	2863	0.024687	0.409753	0.990182
demencia	3231	0.027861	0.437613	0.991585
daganatos megbetegedés	3455	0.029792	0.467405	0.992987
tüdőbetegség	4241	0.036570	0.503975	0.994390
krónikus veseelégtelenség	4665	0.040226	0.544201	0.995792
iszkémiás szívbetegség	9149	0.078891	0.623092	0.997195
cukorbetegség	13354	0.115150	0.738243	0.998597
magasvérnyomás betegség	30356	0.261757	1.000000	1.000000

20. ábra: gyakoriságok-saját szerkesztés

Az is szemet szúr, hogy a leggyakoribb 5 betegség (tüdőbetegség, veseelégtelenség, szívbetegség, cukorbetegség, magasvérnyomás) gyakorisága közösen az összes társbetegség majdnem felét lefedi.

Hasonlóan jártam el a hierarchikus klaszterezés segítségével előállított oszlop esetében is.

	Diagnózis	gyakorisag	rel_gyak	kum_rel_gyak	rel_pos
3039	bélgyulladás	29	0.000248	0.087725	0.949925
645	koszorúér meszesedés	29	0.000248	0.087973	0.950176
674	végstádiumú veseelégtelenség	30	0.000257	0.088229	0.950428
2320	adat felöltés	30	0.000257	0.088486	0.950679
3190	alultápláltság	30	0.000257	0.088742	0.950931

21. ábra: gyakoriságok-saját szerkesztés

Ebben az esetben azt láthatjuk, hogy a diagnózisok 95%-a még alacsonyabb (9% alatti) kumulált relatív gyakorisági szinttel rendelkezik, tehát relatíve több betegség esetén beszélhetünk ritka előfordulásról.

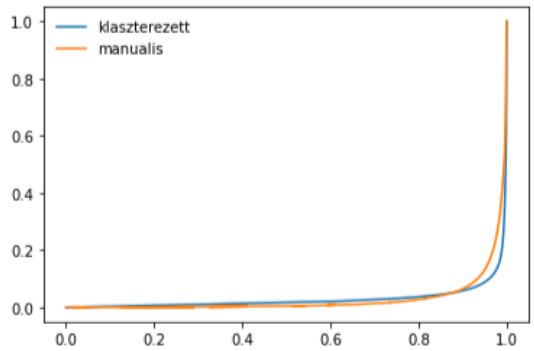
	Diagnózis	gyakorisag	rel_gyak	kum_rel_gyak	rel_pos
1603	ismert alapbetegség	2030	0.017360	0.467606	0.997735
2700	szívelégtelenség	2263	0.019353	0.486958	0.997987
324	krónikus veseelégtelenség	2315	0.019797	0.506756	0.998239
3753	daganatos megbetegedés	2858	0.024441	0.531197	0.998490
975	demencia	3073	0.026280	0.557477	0.998742
204	szívbetegség	3650	0.031214	0.588691	0.998993
3941	iszkémiás szívbetegség	4949	0.042323	0.631014	0.999245
1027	magas vérnyomás	8945	0.076496	0.707510	0.999497
1835	cukorbetegség	12960	0.110832	0.818342	0.999748
1510	magasvérnyomás	21242	0.181658	1.000000	1.000000

22. ábra: gyakoriságok-saját szerkesztés

A leggyakrabban előforduló betegségek javarészt megegyeznek az előző megoldásban látható diagnózisokkal. A leginkább szembetűnő különbség, hogy néhány esetben az algoritmus túlságosan érzékeny volt a karakterláncok közötti eltérésekre, ezáltal néhány, ugyanazon

betegséghez tartozó helyesírási variáció külön klaszterbe került (pl. magasvérnyomás-magas vérnyomás). Jól látszik ugyanakkor, hogy a leggyakoribb betegségek itt is 50% körüli kumulált relatív gyakorisággal rendelkeznek.

Ahhoz azonban, hogy pontosabb képet kapjak a betegségek koncentrációjáról Lorenz görbéket alkalmaztam. Ezeknek vizualizálásához a *matplotlib* csomagban elérhető vonaldiagrammokat használtam fel. Az x-tengelyen a relatív pozíciók, míg az y tengelyen a kumulált relatív gyakoriságok szerepelnek.



23. ábra: Lorenz görbék-saját szerkesztés

Megfigyelhetjük, hogy mindkét esetben igencsak megközelítik a görbék a négyzet oldalát, és jelentősen eltérnek a koncentráció hiányát jelző átlótól. Ugyanakkor a „klaszterezett" görbe közelebb van a teljes koncentrációt jelentő (1,0) koordinátához.

Ennél átfogóbb képet ad, ha kiszámítjuk a Hoover indexet mindkét esetben:

```
x=eloszlas["rel_gyak"]*100
f=1/len(eloszlas["rel_gyak"])
sum(abs(x-f))/2
```

```
49.641686302050594
```

```
x=eloszlas2["rel_gyak"]*100
f=1/len(eloszlas2["rel_gyak"])
sum(abs(x-f))/2
```

```
49.49999999999995
```

24. ábra: Hoover indexek-saját szerkesztés

A mutató értéke mindkét esetben megközelíti az 50%-ot, tehát az egyes betegségek közel felét kellene átcsoportosítanunk ahhoz, hogy az egyes diagnózisok egymással megegyező mértékben részesüljenek a teljes adathalmazból. Ez is jól szemlélteti az adatok egyenlőtlen eloszlását, és a nagymértékű koncentrációt.

4.3 További adatelőkészítések

Ebben a fejezetben bemutatom a későbbi modellezéshez, vizualizációhoz szükséges standardizált adatkészletek kialakításának főbb lépéseit.

4.3.1 Betegségek gyakorisága a társadalomban vs a halálozási adatokban

Ahhoz, hogy láthassuk egy-egy diagnózis relatív hatását a COVID halálozásokra, érdemes első körben megvizsgálni az előfordulási gyakoriságát a teljes népességben. Fontos ugyanakkor tisztáznunk, hogy olyan nyilvánosan elérhető adatbázis, amely hitelesen leírja valamennyi állampolgár betegségeit, tüneteit, nem létezik. Ahhoz, hogy minél pontosabb képet kaphassunk a diagnózisokról, érdemes az egyes gyógyszerhatóanyagok éves forgalmát vizsgálnunk, ezekről az adatokról számos megbízható és hiteles forrás érhető el. Ezzel a módszerrel ugyanakkor közel sem kapunk olyan pontos képet, mintha a betegségek direkt előfordulását vizsgálnánk. Egyfelől fennáll az esély az alul becslésre, hiszen csak a gyógyszerfogyasztókat vizsgáljuk, és nem az összes, tüneteket produkáló személyt. Másfelől egy hatóanyag több betegség enyhítésére is szolgálhat, amely felülbecslést eredményezhet, hiszen minden adott hatóanyagú gyógyszert szedőt a hatóanyag minden betegségénél figyelembe veszünk a felösszegzés során. Mindezek ellenére közelítőleges becslést tudunk adni a kórképek társadalomban vett előfordulására. Az adatokat a NEAK honlapjáról töltöttem le, az adatbázisban fellelhetőek a 2019-2020-as adatok a hatóanyag forgalomra vonatkozóan. A táblázatok access formátumban érhetőek el, számomra a „BETEGSZAMOK_TTT” adattábla volt a releváns, melyben többek között elérhetőek mind a gyógyszerelnevezések, azok hatóanyaga, kiszerelése, valamint a 2019 második felére, illetve 2020 első félévére vonatkozó forgalmuk is.

Ahhoz, hogy az adatokat importálni tudjam telepítettem a pyodbc nevű csomagot. A connection string megadása után pedig egy egyszerű sql lekérdezés segítségével az egész adattábla tartalmát betöltöttem egy DataFrame-be. Ezt követően létrehoztam egy újabb adatszerkezetet, melyben az adatok aggregálva szerepelnek. Ehhez egy groupby metódust használtam, hogy láthassam, hogy hatóanyagonként mennyi az forgalomba került mennyiség összege (sum).

	HATOANYAG	h_gyakorisag
617	J01CR02	1467320.0
14	A02BC02	1332523.0
130	A11CC05	889861.0
178	B01AC06	761307.0
377	C10AA07	693144.0
...	...	...
590	H01CC01	0.0
591	H01CC02	0.0
593	H02AB01	0.0
594	H02AB02	0.0
1303	V08BA01	0.0

25. ábra: Hatóanyagok gyakorisága-saját szerkesztés

Ahhoz, hogy ezt a gyakorisági táblát össze tudjam kapcsolni azzal az adatszerkezettel, melyben az egyes társdiagnózisok gyakoriságai láthatók, szükségem volt egy újabb táblára, mely egymáshoz rendeli a megfelelő betegségeket és a hatóanyagokat. Mivel hasonló tartalmú magyar nyelven fellelhető nyilvános adatbázis nem létezik, ezért angol, illetve más nyelveken elérhető dokumentumokban fellelhető információkra támaszkodtam. Kutatásom során egy spanyol nyelvű standardizáld excel tábla adatait dolgoztam fel, mely tartalmazza mind a betegségek részletes elnevezését, a hatóanyagok kódját, és besorolását is.

Annak érdekében, hogy ezt a táblát kapcsolóként alkalmazhassam a diagnózisok és a hatóanyagok egymáshoz rendeléséhez, szükségem volt egy másik táblára, amely a két tábla társbetegség elnevezéseit köti össze.

Ehhez első lépésben az eredeti eloszlás táblában egy új oszlopban megadtam a betegségek spanyol nyelvű fordítását. Ennek megvalósításához a Google fordító API-ját használtam fel. Annak érdekében, hogy a fordítási hibákat és az ezekből adódó esetleges információvesztést minél hatékonyabban kiküszöböljem, a több szóból álló karakterláncok esetében mind a teljes szószerkezet, mind az azt alkotó valamennyi karakterlánc külön-külön fordítását is megjelenítettem az új oszlopban.

Diagnózis	gyakorisag	diagnozis_esp
magasvérnyomás betegség	30356	hipertensión
cukorbetegség	13354	diabetes
iszkémiás szívbetegség	9149	cardiopatía, enfermedad isquémica del corazón
krónikus veseelégtelenség	4665	insuficiencia renal, falla renal cronica
tüdőbetegség	4241	enfermedad pulmonar

26. ábra: Betegségek, és fordításaik-saját szerkesztés

A kialakított oszlop ugyanakkor önmagában továbbra sem teszi lehetővé a kapcsolat kialakítását a két tábla között, hiszen a spanyol nyelvű adatbázis jóval részletesebb tünetleírásokat tartalmaz. Az eltérő adatszerkezetekből adódó hibák kiküszöbölésének céljából egy újabb *DataFrame*-et hoztam létre, amelynek első oszlopa az előző lépésben kialakított tükörfordításokat tartalmazza. A második oszlop ezekhez az általános megnevezésekhez rendeli hozzá a spanyol nyelvű adatbázisban fellelhető részletes leírásokat annak függvényében, hogy a tükörfordítás valamely változatának karakterláncait tartalmazza-e a részletes elnevezés.

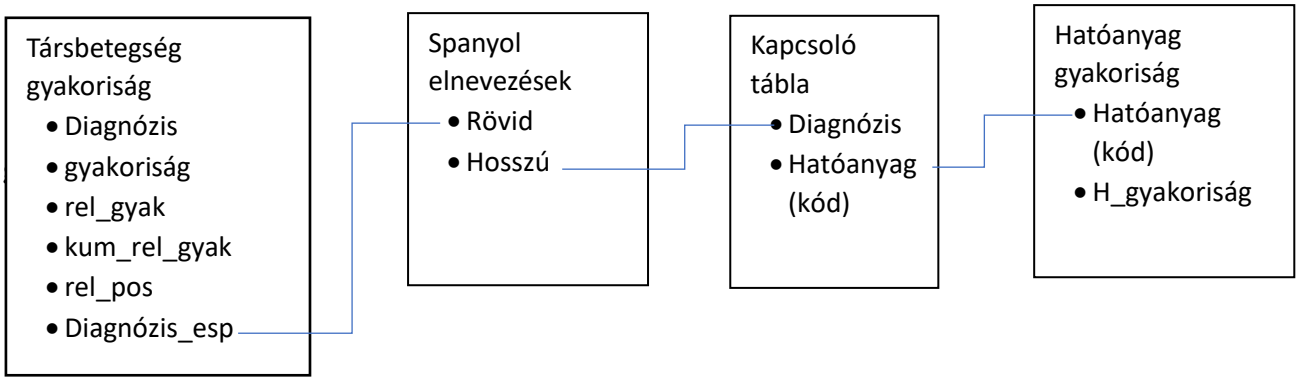
	rovid	hosszu
0	hipertensión	hipertensión secundaria, no especificada
1	hipertensión	hipertensión pulmonar primaria
2	hipertensión	hipertensión renovascular
3	hipertensión	hipertensión esencial (primaria)
4	hipertensión	preeclampsia superpuesta en hipertensión crónica

27. ábra: Spanyol nyelvű elnevezések-saját szerkesztés

Rendelkezésünkre áll tehát immár mind a 4 adattábla, mely az összehasonlítások alapjául szolgálhat:

- Társbetegség gyakorisági tábla:
  - Ez az adattábla tartalmazza a COVID-ban elhunyt szermélyek társbetegségeinek abszolút, relatív, kummulált relatív gyakoriságát a teljes COVID halálozás adatbázisban, valamit a spanyol nyelvű tükörfordításokat
- Spanyol elnevezéseket tartalmazó tábla
  - Ez az adattábla tartalmazza az egyszerűsített elnevezéseket, illetve a részletes tünetleírásokat.
- Kapcsoló tábla
  - Spanyol nyelvű, standardizált adatszerkezet, amely egymáshoz rendeli a megfelelő diagnózist, és hatóanyagokat.
- Hatóanyag gyakorisági tábla

- Ez a tábla aggregáltan tartalmazza, hatóanyag szerinti csoportosításban a 2019-2020-as évi gyógyszerforgalmat



Következő lépésben a társbetegségek, illetve hatóanyagok közötti many-to-many kapcsolat hatékony kezelése érdekében a nyilakkal ábrázolt idegen kulcs kényszerek mentén összekapcsoltam a négy DataFrame-et. Az így kialakított standardizált adattábla segítségével végeztem el az aggregálásokat.

Ezt követően a társbetegség, és a hatóanyag kód oszlopok szerint végeztem el a hatóanyagforgalom átlagolását. Ez a lépés azért volt szükségszerű, mivel az eredeti spanyol nyelvű adatbázisban a hatóanyagokat forgalmazó tábla számunkra redundáns módon az előállító laboratórium elnevezését is tartalmazza, így előfordulhat, hogy ugyanazon hatóanyag ugyanazon kórképhez többszörösen van hozzárendelve. Az így kialakított adatszerkezet segítségével végeztem el a társbetegség szerinti gyógyszerforgalom felösszegzését.

	Diagnózis	h_gyakorisag
210	tüdő és	21768394.0
249	és	21768394.0
239	vér és	21768394.0
194	szív és	21768394.0
38	betegségei	20511698.0
...	...	...
43	bélelzáródás	0.0
29	aorta szűkület	0.0
24	amnézia	0.0
22	alvási apnoe	0.0
254	üszkösödés	0.0

28. ábra: Hatóanyag gyakoriság-saját szerkesztés

Azt láthatjuk, hogy az adattábla több olyan rekordot is tartalmaz, melyek nem konkrét betegségelnevezések, hanem a helytelenül tagolt betegségfelsorolások (vessző hiba stb.) tokenizálásából fennmaradó csekély információtartalmú elnevezések, kötőszavak. Következő lépésben egy metódus segítségével kiszűrtem a táblából azokat a rekordokat, melyek kizárólag ezeket az önmagukban redundáns szavakat, illetve stop szavakat tartalmazzák.

	Diagnózis	h_gyakorisag	gyakorisag
156	nyelészavar	11146240.0	3
141	magasvérnyomás betegség	6820385.0	30356
140	magas vérnyomás	6820385.0	27
94	hypertonia	6820385.0	4
91	hipertónia	6820385.0	4
...	...	...	...
123	kóma	0.0	8
229	veseelégtelenség	0.0	4
125	kóros fogyás	0.0	3
85	hasfali sérv	0.0	3
132	laktózintolerancia	0.0	2

29. ábra: Hatóanyag gyakoriság vs. társbetegség gyakoriság-saját szerkesztés

Miután felülvizsgáltam az adatkészletet, és a hatóanyaggyakoriságnak megfelelően sorrendbe rendeztem a diagnózisokat, szembetűnt, hogy továbbra is találkozhatunk redundáns elnevezésekkel, melyek immár nem a hibás tagolásból, hanem a szinonimák alkalmazásából származnak. Csekély számban előfordulnak olyan karakterláncok, melyek nem a betegség hétköznapi leírását, hanem az idegen nyelvekből átvett elnevezésüket jelenítik meg (pl. magasvérnyomás-hipertónia). Ezek a rekordok abból a szempontból okoznak gondot, hogy jelentősen torzítják a képet, ha a betegségek komorbiditását (két vagy több különböző betegség együttes jelenléte ugyanazon személynél) a társadalomban vett becsült gyakoriságukhoz képest szeretnénk ábrázolni.

Annak érdekében, hogy ezeket az adatokat egységesen kezelhessem, a hatóanyaggyakoriság, ezen belül pedig a társbetegséggyakoriság szerint csökkenően rendezett adattábla diagnózisokat tartalmazó oszlopát a következők szerint módosítottam:

- Megvizsgáltam, hogy a megegyező becsült előfordulású betegségek fordításakor valóban ugyanazt a karakterláncot kapom-e
- Amennyiben az előző feltétel teljesül, az adott szakasz társbetegségként vett leggyakoribb elnevezését rendeltem hozzá az egyes szinonimákhoz

	Diagnózis	h_gyakorisag	gyakorisag	Diagnózis_egys
156	nyelészavar	11146240.0	3	nyelészavar
141	magasvérnyomás betegség	6820385.0	30356	magasvérnyomás betegség
140	magas vérnyomás	6820385.0	27	magasvérnyomás betegség
94	hypertonia	6820385.0	4	magasvérnyomás betegség
91	hipertónia	6820385.0	4	magasvérnyomás betegség
...	...	...	...	...
123	kóma	0.0	8	kóma
229	veseelégtelenség	0.0	4	veseelégtelenség
125	kóros fogyás	0.0	3	kóros fogyás
85	hasfali sérv	0.0	3	hasfali sérv
132	laktózintolerancia	0.0	2	laktózintolerancia

30. ábra: Egységesített diagnózisok-saját szerkesztés

Miután a módosított DataFrame-en elvégeztem a gyakoriságok aggregálását megkaptam azt az adattáblát, amelyet a későbbi vizualizáláshoz használtam fel:

	Diagnózis_egys	h_gyakorisag	gyakorisag
94	nyelészavar	11146240.0	3
84	magasvérnyomás betegség	6820385.0	30391
149	ízületi gyulladás	5958940.0	214
105	reflux	5804419.0	898
53	hörghurut	5322633.0	134
...	...	...	...
71	kóma	0.0	8
134	veseelégtelenség	0.0	4
46	hasfali sérv	0.0	3
73	kóros fogyás	0.0	3
77	laktózintolerancia	0.0	2

4.3.2 További adatelőkészítések a későbbi modellezéshez, vizualizációhoz

Következő lépésben létrehoztam egy olyan adatszerkezetet, amelynek segítségével könnyedén alátámaszthatóak a társbetegség, nem, illetve a kor közötti kapcsolatok, összefüggések. Ehhez elsőkörben módosítottam az eredeti, egységesített elnevezéseket tartalmazó adattáblát aszerint, hogy minden egyes társbetegség külön, külön sorban jelenjen meg. Jól látható, hogy ezzel a megoldással számos duplikáció jön létre, hiszen ahány társbetegség, annyi sorban szerepelnek az adott beteg adatai. Ugyanakkor ez az adatstruktúra már alkalmas arra, hogy ez alapján szemléltethessük az egyes társdiagnózisok korral, nemmel való összefüggéseit. Ennek előállításához az eredeti *DataFrame* 2.1. fejezetben tárgyalt eljárás révén létrehozott társbetegségeket tartalmazó oszlopán végigiteráltam egy *for* ciklus segítségével, és az aktuális elemet szeparáltam az elválasztó karakter mentén. Egy új listát immáron ezekkel a tokenizált elemekkel töltöttem fel egy belső ciklus segítségével. Ugyanebben a belső ciklusban (a ciklusszámnak megfelelő mennyiségben) feltöltöttem a korokat és a nemeket tartalmazó listát is a külső ciklus aktuális indexének megfelelő sorszámú eredeti DataFrame elemével.

	nem	bet	kor
0	Nő	magasvérnyomás betegség	92.0
1	Nő	asztma	86.0
2	Nő	vastagbélgyulladás	86.0
3	Férfi	pajzsmirigy túlműködés	80.0
4	Férfi	magasvérnyomás betegség	94.0
...	...	...	...
115974	Férfi	és érrendszeri megbetegedés	68.0
115975	Férfi	és érrendszeri megbetegedés	79.0
115976	Férfi	és érrendszeri megbetegedés	74.0
115977	Nő	tüdő rosszindulatú daganata	65.0
115978	Nő	és érrendszeri megbetegedés	76.0

31. ábra: gyakorisági tábla-saját szerkesztés

4.4 Modell építés és adatvizualizáció

Elemzésem során igyekeztem megtalálni az egyes változók (kor, nem, diagnózis, társbetegség gyakoriság, illetve a társadalomban vett gyakoriság) közötti összefüggéseket. Ehhez az R programnyelvet használtam fel, amely kiváló lehetőséget biztosít az adatvizualizációk létrehozásához, statisztikai számításokhoz, valamint modell építéshez is. Miután a szükséges adatok exportálásra kerültek a Python kód futtatását követően, beimportáltam őket a létrejött

Excel fájlokból. Megfigyelhetjük, hogy egyaránt találkozhatunk minőségi (társbetegség, nem), illetve mennyiségi ismérvekkel (kor, gyakoriság) is, így többféle megközelítést alkalmaztam a kapcsolatok feltárására.

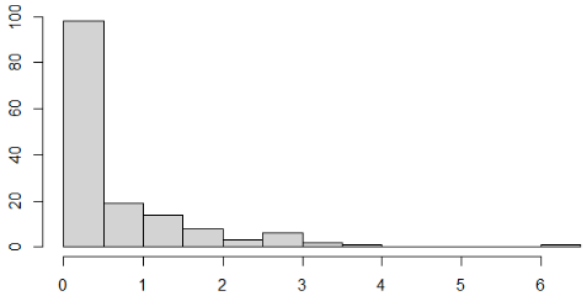
4.4.1 Társbetegségek gyakorisága a COVID halálozásokban és a társadalomban

Első körben két mennyiségi ismérvet vizsgáltam:

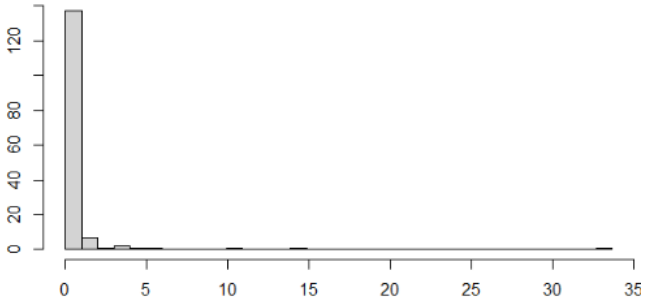
- A betegségek előfordulása COVID társbetegségként
- A diagnózisok társadalomban vett becsült előfordulása

4.4.1.1 Eloszlások vizsgálata

A kapcsolatvizsgálat előtt külön-külön elemeztem a két változó eloszlását. Ehhez egy-egy hisztogram segítségével vizualizáltam az adatokat.



33. ábra: Hisztogram a hatóanyaggyakoriság alapján-saját szerkesztés



32. ábra: Hisztogram a társbetegségek előfordulása alapján-saját szerkesztés

Megfigyelhető, hogy mindkét esetben jobbra elnyúló alakzatokat láthatunk, tehát viszonylag nagyszámú ritka, csekély részesedéssel bíró betegséget tartalmaz az adatbázis, míg a relatíve nagy hányaddal rendelkező betegségek esetén csak alig pár diagnózisról beszélhetünk. Ennek legvalószínűbb oka az lehet, hogy míg „felfelé" irányban korlátlan számú outlier értékkel találkozhatunk (pl a társbetegségként való előfordulás megközelíti a 25%-ot), addig az átlagot messze alulmúló értékeknek van egy alsó korlátja (0). Ezt a balra ferde eloszlást statisztikai mérőszámok segítségével is alátámasztottam. Ehhez mindkét változó esetén a summary metódust alkalmaztam:

Minimum	1. kvartilis	Medián	Átlag	3. kvartilis	Maximum
0.00222	0.00554	0.01827	0.65789	0.15893	33.65969

34. ábra: Statisztikai mutatók a társbetegségekhez-saját szerkesztés

Minimum	1. kvartilis	Medián	Átlag	3. kvartilis	Maximum
0	0.0412	0.2687	0.6579	0.8988	6.0251

35. ábra: Statisztikai mutatók a hatóanyaggyakoriságokhoz-saját szerkesztés

Mindkét esetben az láthatjuk, hogy a medián értéke közel 0-t vesz fel, tehát gyakoriság szempontjából a diagnózisok alsó 50%-a nem, vagy alig rendelkezik valamekkora részesedéssel. Ez a mutató jóval kevésbé érzékeny a kiugró értékekre, mint az átlag, nem meglepő tehát, hogy ennek értéke mindkét esetben szignifikánsan alacsonyabb az átlagénál.

Ezekből az értékekből is jól kirajzolódik tehát a jobbra elnyúlás. Ahhoz azonban, hogy ki tudjuk fejezni ennek mértékét, érdemes megjelenítenünk a ferdeség mutatót is. Ehhez a *psych* csomagot célszerű importálnunk, amely után a *describe* függvénnyel egyéb statisztikai mérőszámokat is meg tudunk vizsgálni.

```
> describe(adatok$h_rel_gyakorisag)
   vars   n mean   sd median trimmed  mad min  max range skew kurtosis   se
X1    1 152 0.66 0.92   0.27    0.47 0.39   0 6.03  6.03 2.37     7.59 0.07
> describe(adatok$rel_gyakorisag)
   vars   n mean   sd median trimmed  mad min   max range skew kurtosis   se
X1    1 152 0.66 3.14   0.02     0.1 0.02   0 33.66 33.66 8.48    80.81 0.25
```

36. ábra: Statisztikai mutatószámok-saját szerkesztés

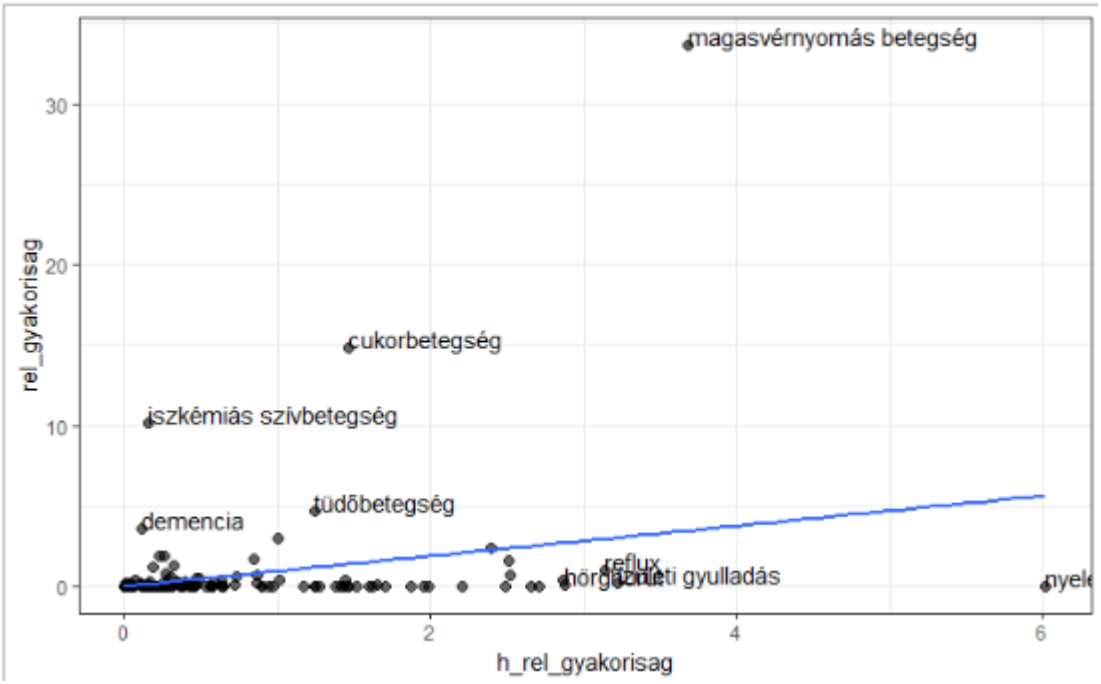
A kurtosis (ferdeség) paraméter mindkét esetben erősen eltér pozitív irányban a normális eloszlásra jellemző 0 értéktől. Megfigyelhetjük azonban azt is, hogy a társbetegségként való előfordulási gyakoriságok szempontjából a jobbra elnyúlás jóval markánsabban (7 feletti érték) tetten érhető, mint a társadalomban vett becsült részesedések esetében (2,37).

4.4.1.2 Korrelációs kapcsolat vizsgálat

Következő lépésben a két változó kapcsolatát vizsgáltam. Első körben a COVID társbetegségeket ábrázoltam egy pontdiagrammon, amelyet a *ggplot* csomag segítségével könnyen fel lehetett paraméterezni. Az *x* tengelyen szerepelnek a hatóanyagok forgalmából számolt társadalomban vett relatív gyakoriságok, míg az y tengelyen pedig a COVID társbetegségként való relatív gyakoriságokat ábrázoltam. Emellett pedig az egyes alakzatokat egy-egy felirattal is elláttam, amely az adott koordinátához tartozó társbetegség megnevezését tartalmazza.

Mivel több mint 800 diagnózist vizsgálunk, ezért nem érdemes mindegyik betegségelnevezést megjelenítenünk az ábrán, célszerű csupán a valamilyen szempontból (társbetegség gyakoriság, vagy hatóanyag gyakoriság) kilógónak tekinthető értékeket megjeleníteni. Ehhez készítettem egy metódust, amely abban az esetben adja vissza a diagnózis elnevezését, ha az társbetegség előfordulás szempontjából az 5 leggyakoribb betegség között szerepel, vagy a forgalmazott hatóanyagok tekintetében az első 5 hely valamelyikét foglalja el, egyéb esetben egy üres

karakterlánccal tér vissza. Ezt a függvényt futtattam le a *DataFrame* egy új oszlopán, ezt az oszlopot adtam meg a diagramm *label* paramétereként.



37. ábra: Korrelációs kapcsolat vizualizálása-saját szerkesztés

Amennyiben egy trendvonalat illesztünk az ábránkra, jól látható a pontfelhő nagymértékű szóródása az egyenes körül, amelyből arra következtethetünk, hogy viszonylag gyenge a két változó között fennálló lineáris kapcsolat

Ahhoz, hogy ezt mérőszámokkal is alátámasszam, a beépített *lm* metódus segítségével egy modellt futtattam, melynek eredmény változójaként a társbetegség relatív gyakoriságokat, míg magyarázóváltozójaként, a hatóanyagforgalmat adtam meg. A futtatást követően az alábbi együtthatókat kaptam:

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.006097   0.336581   0.018  0.98557
h_rel_gyakorisag 0.935572   0.285247   3.280  0.00132 **
```

38. ábra: Koefficiensek-saját szerkesztés

Ami már elsőre szemet szúr, hogy a hatóanyaggyakoriság bétájának p-értéke szignifikánsnak tekinthető minden szokásos alfa megválasztása mellett. A nullhipotézist tehát, mely szerint a mintán kívüli világban a betegség társadalomban vett előfordulása nem befolyásolja a COVID mortalitást minden észszerűen megválasztott szignifikanciaszinten elvetjük.

Ami a tengelymetszetet illeti, ennek az értékét könnyen értelmezhetjük. Ez alapján azt mondhatjuk, hogy azok a betegségek, melyeknek az adott évben nem volt megjeleníthető gyógyszerforgalmuk várhatóan a COVID társbetegségek 0,006%-át teszik ki. Ezenkívül a két változó összefüggését úgy írhatjuk le, hogy amennyiben 1%ponttal magasabb a betegségre szánt éves relatív hatóanyagforgalom várhatóan várhatóan 0,94%ponttal növekszik a COVID társbetegségként való relatív előfordulás értéke a halálozásokban. Ebben az esetben tehát egy pozitív irányú kapcsolatról beszélhetünk.

Amennyiben a kapcsolat szorosságát is számszerűsíteni szeretnénk érdemes leolvasnunk az R^2 értéket is, amely esetünkben mindössze 7,28% lesz. Tehát a mutató értéke alapján csekély mértékben magyarázza a társadalomban vett becsült relatív gyakoriság az adott diagnózis Covid társbetegségként való előfordulását a halálozásokban.

A pontdiagrammra visszatérve azt tapaszatalhatjuk, hogy több olyan betegség is szerepel a listában, melyek a COVID mortalitásban messze felülreprezentáltak, a társadalomban vett relatív előfordulási gyakoriságukhoz képest:

- magasvérnyomás betegség
- szívproblémák
- cukorbetegség
- demencia
- tüdőbetegség

Ezek tehát azok a diagnózisok, melyek a többi betegséghez képest jóval nagyobb valószínűséggel okoznak halálozást COVID társbetegségként. A koefficiensek alapján például kiszámíthatjuk, hogy a magasvérnyomásnak a koronavírushoz köthető halálozásokból vett relatív részesedésének becsült értéke 3,45%, ehhez képes a tényleges érték 33,66%. Elsősorban ezek a diagnózisok okozzák a gyenge összefüggést a COVID-19 halálozások és a gyógyszerforgalom alapján becsült relatív gyakoriságok között kilógó viselkedésük miatt.

Megfigyelhetünk ugyanakkor olyan betegségeket is melyek esetén szignifinánsan kisebb a mortalítás társbetegségként, amekkora a hatóanyagforgalom alapján elvárható lenne. Ezekre pédák a következő betegségek:

- reflux
- hörghurut
- izületi gyulladás

Az eredmény nem meglepő, hiszen ezek a betegségek a való életben sem tekinthetőek magas mortalitásúnak, inkább enyhe tüneteket okoznak. Logikus tehát, hogy a jelentős előfordulási gyakoriságuk ellenére nem eredményeznek jelentős többlethalálozást a COVID-dal összefüggésben sem.

A modellben szereplő együtthatókat felhasználva, és a reflux hatóanyaggyakoriságát behelyettesítve azt láthatjuk, hogy a becsült mortalitás társbetegségként 2,94%, ennek ellenére a tényleges halálozási relatív gyakoriság messze alulreprezentált, 1% alatti.

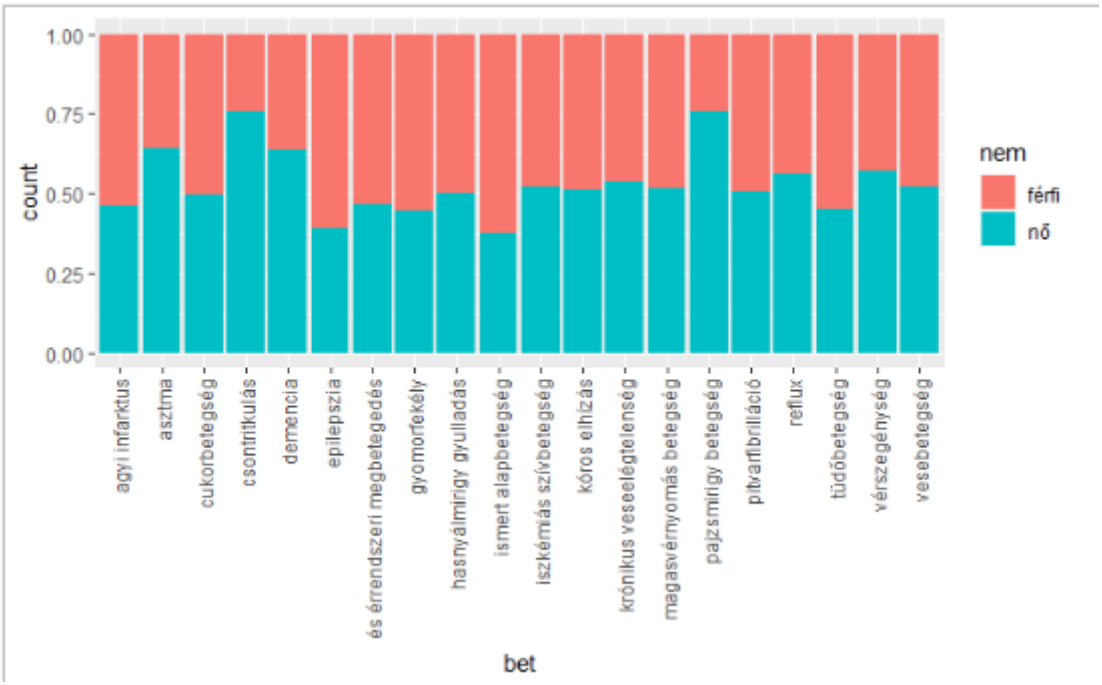
4.4.2 Kor, nem és társbetegség közötti összefüggések

Következő lépésben a COVID-19-ben elhunytak korát, nemét, és társbetegségeit egyaránt tartalmazó adattáblát töltöttem be. Ebben a táblázatban duplikált adatok is nagy mennyiségben előfordulnak, hiszen azok a betegek, akik esetében többféle társbetegséget is diagnosztizáltak, többször is szerepelnek a táblában.

4.4.2.1 Asszociációs kapcsolat a betegség és a nem között

Első körben a betegség, illetve a nem összefüggését vizsgáltam, mint két minőségi ismérv kapcsolatát. Ennek a vizualizációjához 100%-ig halmozott oszlopdiagrammokkal dolgoztam. Ahhoz azonban, hogy az ábra átlátható maradjon, csupán a 20 leggyakoribb diagnózist vettem figyelembe. Ezeket az adatokat könnyen megkaptam, miután az előző (betegségeket, és gyakoriságokat tartalmazó) táblázatot a társbetegség szerinti relatív előfordulási gyakoriság alapján csökkenő sorrendbe rendeztem, és ezt követően kiszűrtem az első 20 rekord betegség elnevezéseit. Ezután ezt a DataFrame-et is átalakítottam aszerint, hogy csak azokat a rekordokat tartalmazza, amelyek esetében a betegség elnevezés megegyezik az imént kiszűrt diagnozisok valamelyikével.

Ezeket az adatokat ábrázolva a következő diagrammot kaptam:



39. ábra: Asszociációs kapcsolat vizualizálása a betegség és nem között-saját szerkesztés

A 39. ábrán azt láthatjuk, hogy annak ellenére, hogy a teljes (duplikátumokat tartalmazó) adathalmazban a nők és férfiak nagyjából megegyező számban vannak jelen (nők aránya 50,75%, férfiak aránya 49,25%), ezek az arányok az egyes diagnózisok esetében jelentősen eltérnek egymástól. A legfeltűnőbb egyenlőtlenségek, a pajzsmiriggyel összefüggő betegségek esetében figyelhetőek meg nők javára, míg a férfiak leginkább a tüdőbetegséggel, illetve epilepsziával kapcsolatban álló tünetekben felülreprezentáltak az elhunytak között. Ami még szemet szúr, hogy a 3 leggyakoribb társdiagnózis esetében nagyjából fele-fele arányban oszlanak meg a halálozások a két nem között.

Amennyiben pontosabban, számszerűen is meg szeretnénk vizsgálni ezeket az arányokat érdemes a *prop.table* beépített metódust alkalmaznunk az adathalmazon. Ennek segítségével könnyedén előállítható egy relatív gyakorisági táblázat:

	férfi	nő
agyi infarktus	0.5371429	0.4628571
asztma	0.3609467	0.6390533
cukorbetegség	0.5062903	0.4937097
csontritkulás	0.2429752	0.7570248
demencia	0.3658310	0.6341690
epilepszia	0.6081794	0.3918206
és érrendszeri megbetegedés	0.5330357	0.4669643
gyomorfekély	0.5525926	0.4474074
hasnyálmirigy gyulladás	0.5000000	0.5000000
ismert alapbetegség	0.6272978	0.3727022
iszkémiás szívbetegség	0.4810362	0.5189638
kóros elhízás	0.4886911	0.5113089
krónikus veseelégtelenség	0.4653805	0.5346195
magasvérnyomás betegség	0.4830347	0.5169653
pajzsmirigy betegség	0.2409779	0.7590221
pitvarfibrilláció	0.4943820	0.5056180
reflux	0.4410112	0.5589888
tüdőbetegség	0.5484556	0.4515444
vérszegénység	0.4290587	0.5709413
vesebetegség	0.4800000	0.5200000

40. ábra: Betegségek nemek közti megoszlása-saját szerkesztés

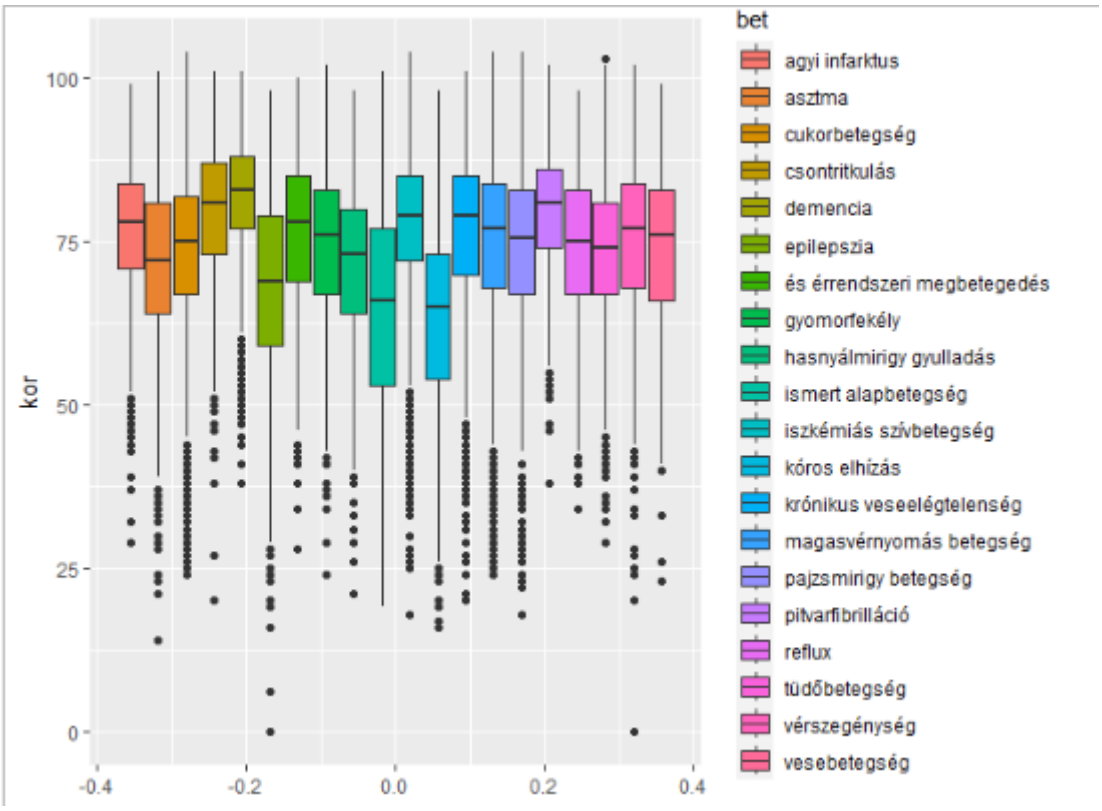
Az 40. ábra alapján is azt tapasztalhatjuk, hogy halálozások alapján vett 3 leggyakoribb betegség (iszkémiás szívbetegség, cukorbetegség, magasvérnyomás) esetén megközelítőleg a teljes adathalmaznak megfelelő nemarányokat láthatjuk.

Ahhoz, hogy láthassuk, hogy a két változó értékei mennyire befolyásolják egymást, érdemes az asszociációs kapcsolat szorosságát a Cramer-együtthatóval is kifejeznünk. Ennek megállapításához a *questionr* csomagot használtam. Az importálást követően a *cramer.v* metódust felhasználva azt láthatjuk, hogy egy viszonyleg gyenge (0,3-nál alacsonyabb) kapcsolatot kapunk, alig több mint 0,13-as értékkel.

Azt mondhatjuk tehát, hogy a két változó alig befolyásolja egymást, sokkal közelebb vagyunk a teljes függetlenséget jelentő 0-s értékhez, mint a függvényszerű kapcsolatot mutató 1-hez.

4.4.2.2 Vegyes kapcsolat a társbetegség, illetve kor között

Következő lépésben a tásbetegség, illetve kor közötti összefüggéseket vizsgáltam. Ennek vizualizálásához dobozábrákkal dolgoztam, melyeket egy diagramfelületre plotoltam ki a *ggplot* csomag segítségével. Az ábra y tengelyén a kor változó értékei láthatóak, míg az egyes alakzatok a jelmagyarázatban szereplő betegségek szerint vannak színezve (ebben az esetben is csak a 20 leggyakoribb társbetegséggel dolgozunk).



41. ábra: Vegyes kapcsolat vizualizálása a betegség, és kor között-saját szerkesztés

Jól látható, hogy mind a medián, mind az 1. illetve 3. kvartilis jelentősen eltér az egyes diagnózisok tekintetében. Ugyanígy szemet szúrnak az eltérő interkvartilis terjedelmek is. Megfigyelhetjük, hogy leginkább azon betegségek esetén szűkebb a középső 50%-ot jelentő intervallum, melyek az idősebb korosztályra jellemzőek pl. demencia. Ez észszerűnek is tűnik, hiszen ezek a betegségek a teljes társadalmat alapul véve is elsősorban az idősebb, általában 80. életévüket betöltő személyeknél alakulnak ki, és mivel ez a korosztály már önmagában véve is leginkább egy szűk (legtöbbször a magyar halálozási átlagéletkor meghaladó korcsoportba tartozó) társadalmi réteget alkot, ezért logikusnak tűnik, hogy a kor tekintetében az adatok kevésbé szóródnak.

Az ábráról az is jól kivehető, hogy a nagyobb szóródás, és egyben az alacsonyabb medián életkor a leginkább szembetűnően az elhízással összefüggő COVID halálozásokban jelentkezik. Ez is logikusnak tűnik, hiszen ez az egészségügyi probléma jóval elterjedtebb a fiatalabb korosztály esetében az össznépességet tekintve.

Azt is észrevehetjük, hogy kilógó értékek kizárólag lefelé irányban fordulnak elő. Ennek oka, hogy a legtöbb esetben a medián életkor megközelíti a magyar születéskor várható élettartamot, ezért felfelé jóval limitáltabb a lehetőség az anomáliák kialakulására.

Ahhoz, hogy számszerűen is megjelenítsem a statisztikai mérőszámokat betegségek szerinti bontásban, a *by* függvényt alkalmaztam.

```
stat3$bet: cukorbetegseg
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  24.00   67.00   75.00   73.64   82.00  104.00
------------------------------------------------
stat3$bet: iszkémiás szívbetegség
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00   72.00   79.00   77.94   85.00  104.00
------------------------------------------------
stat3$bet: magasvérnyomás betegség
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  24.00   68.00   77.00   75.42   84.00  104.00
```

42. ábra: Kormegoszlás betegségek szerinti bontásban-saját szerkesztés

A három leggyakoribb társbetegség alapján azt láthatjuk, hogy a kvartilisek, illetve az átlagok hasonló értékeket mutatnak.

Ahhoz, hogy a teljes adathalmazban megvizsgáljam a két ismérv kapcsolatát, kiszámítottam a szóráshányadost.

A kapott, 0,3-at meghaladó (0,33) értékről azt mondhatjuk, hogy ebben az esetben sem beszélhetünk erős, függvényszerű összefüggésről, ugyanakkor a korrelációs, és asszociációs kapcsolattól eltérően itt átlépjük a közepes szorosságú kapcsolat alsó határát. Azt mondhatjuk tehát, hogy még ha gyengén is, de befolyásolják a társbetegségek a COVID-dal összefüggő halálozási életkort.

4.4.2.3 Regressziós modell építése

Következő lépésben egy többváltozós lineáris regressziós modellt építettem, melyben az eredményváltozó a halálozási kor, magyarázóváltozók pedig a faktorrá alakított betegség, illetve nem voltak. A modell futtatását követően azt tapasztaltam, hogy számos redundáns változó keletkezett, a koefficiensek többségének p-értéke messze meghaladja a szokásos szignifikanciaszinteket, csupán néhány esetben beszélhetünk szignifikáns hatásokról. Ahhoz, hogy egy kényelmesebben kezelhető, könnyebben értelmezhető modellt kaphassak, csupán azokat a változókat tartottam meg, amelyek esetében a béták szignifikánsan különböznek 0-tól, legalább 10%-os alfa megválasztása mellett.

Ehhez szükségem volt egy új DataFrame-re, melyben eltároltam az eredeti koefficienseket (standard hibákkal, próbafüggvény, illetve p-értékekkel együtt). Ezt követően kiszelektáltam azokat a rekordokat, amelyek esetén a p-érték 0,1 alatt van.

Következő lépésben az indexekként szereplő betegségelnevezéseket megtisztítottam a „bet” előtagtól, ezáltal visszakaptam az eredeti diagnózisokat. Miután létrehoztam, és lefuttattam a metódust az oszlopon, az eredeti (a modellünk adatforrásaként felhasznált) táblázatban a „referencia” karakterláncra cseréltem azokat a társdiagnózisokat, amelyek nem szerepeltek a szignifikáns hatású betegségek között. A transzformációkat követően megalkottam az új modellt

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	72.50628	0.04854	1493.738	< 2e-16	***
nemnő	4.83050	0.06814	70.894	< 2e-16	***
betagyhártya daganat	-26.54205	3.86692	-6.864	6.74e-12	***
betautizmus	-43.76778	4.38471	-9.982	< 2e-16	***
bethiperandrogenizmus	-50.33677	11.60041	-4.339	1.43e-05	***
bethypercholesterinaemia	-35.50628	8.20280	-4.329	1.50e-05	***
betkoraszülöttség 4 hónapos	-74.11644	6.69754	-11.066	< 2e-16	***
betmájszteatózis	-33.33677	11.60041	-2.874	0.004057	**
betprosztata rendellenesség	-49.33677	11.60041	-4.253	2.11e-05	***
bettüdő primer tumor	-30.42152	8.20273	-3.709	0.000208	***
beturocystitis	-35.33677	11.60041	-3.046	0.002318	**
betvárandós	-43.33677	8.20280	-5.283	1.27e-07	***

43. ábra: Regressziós modell-saját szerkesztés

Az új, redundáns változóktól megtisztított modellben immár valamennyi béta szignifikánsan különbözik 0-tól még 1%-os alfa megválasztása esetén is. Ami még szemet szúr, hogy javarészt ritkán hallható, különleges elnevezésű diagnózisokról beszélhetünk (pl. hiperandrogenizmus, májszteatózis). Ez nem is meglepő, hiszen ezen dummy változók esetében az eredeti adathalmazban nincs elegendő adatpont ahhoz, hogy a modell pontosan meg tudja becsülni az ezekhez tartozó hatásokat, ezáltal az ilyen változók becsült hatásai eltúlzottak lehetnek, és szignifikánsnak tűnhetnek. Könnyen belátható esetükben, hogy a kis mértékű szóródás miatt a próbafüggvény nevezőjében szereplő standard hibák minimálisra csökkennek, azáltal pedig a t-eloszlásból számított p-értékek indokolatlanul alacsony értékeket vesznek fel.

A betegségek közül az autizmus, illetve a koraszülöttség koefficiense tűnik a legszignifikánsabbnak. Azt mondhatjuk, hogy változatlan nem mellett az adott személy várhatóan 44 évvel fiatalabban halálozik el a COVID, illetve az autizmus következtében, mintha valamilyen más, a referenciakategóriába tartozó társbetegséget diagnosztizáltak volna nála. Ami még szemet szúr, hogy a nem hatása is szignifikánsnak mondható (0-hoz közeli p-érték). A béta alapján kijelenthetjük, hogy azonos kórkép esetén a nők majdnem 5 évvel tovább élnek, mint a férfiak. Ez alapján elmondható tehát, hogy a nők ellenállóbbak a COVID-dal szemben, mint a hasonló egészségi állapottal rendelkező férfiak.

5 Összefoglalás

Kutatásom során jelentős mennyiségű adat állt rendelkezésemre ahhoz, hogy meg tudjam vizsgálni a COVID-19 halálozások hazai összefüggéseit. Célom az volt, hogy egy teljeskörű ETL folyamat által definiált lépéssorozat végrehajtását követően, egy tisztított, strukturált adatállomány alapján ábrázoljam a halálozási adatokban fellelhető összefüggéseket, és összevessem azokat az előzetes feltevésekkel.

Munkám során eltérő tartalmú, és szerkezetű állományokkal dolgoztam, melyeket különféle, önállóan megalkotott részfolyamatok segítségével töltöttem be a megfelelő, elemzésre alkalmas adatstruktúrákba. Miután többféle eszközzel is kísérletet tettem az egységes írásmódok, strukturált adatszerkezetek kialakítására, immár könnyen kezelhető adatforrásokat kaptam, melyeket felhasználva vizualizációk, és statisztikai mérőszámok segítségével ábrázoltam a COVID-19 lehetséges hazai összefüggéseit.

Jól látható azonban, hogy egyik módszer sem kínált teljesen hibamentes megoldást, részben az adatok korlátozott rendelkezésre állása (pl. betegségek gyakorisága a társadalmon belül), részben a sokszor felismerhetetlen betegség elírások miatt. A bizonytalanságok, adathiányok ellenére ugyanakkor elmondható, hogy számos következtetést vonhatunk le az egyes változók kapcsolataiból.

A kutatás során megtapasztalhattuk, miképpen függ a halálozási életkor az egyes társbetegségektől. Megfigyelhettük a diagnózisokon belüli arányokat is a nemek tekintetében. Láthattuk, hogy mely krónikus betegségben szenvedő egyének néztek szembe a legnagyobb veszéllyel a járvány során.

A vizualizációk, modellek, statisztikai mérőszámok igazolták az előzetesen ismert feltevések megalapozottságát. A magasvérnyomásban, cukorbetegségben, tüdő-, és szívbetegségben szenvedő személyek magas kockázattal néztek szembe a pandémia során, hiszen esetükben a halálozási arány szignifikánsan felülreprezentált a diagnózisok társadalomban vett becsült relatív gyakoriságához képest. Megfigyelhettük továbbá, hogy az elhízottság a fiatalabbak között is jelentős rizikófaktort jelentett, hiszen ezen betegség esetén jóval alacsonyabb átlag, illetve medián életkorról beszélhetünk a többi meghatározó kórképhez képest. Megtapasztalhattuk továbbá, hogy bár a halálozási okokban előforduló 3 leggyakoribb betegséget tekintve a nemek aránya közel megegyezik, a nők várhatóan majdnem 5 évvel tovább élnek, mint az azonos kórképpel rendelkező férfiak.

Fontos ugyanakkor tisztázni, hogy a kutatás során tárgyalt szövegbányászati eszközökön túlmenően számos egyéb megközelítés alkalmazható hasonló jellegű problémák megoldásához. További lehetőséget biztosít az egyezőségek feltárása az ún. *regex* kifejezések alkalmazása, melyeknek segítségével egy adott szövegminta alapján történik az adatok összevonása. Továbbá a hierarchikus klaszterezésen túlmenően is számos nem felügyelt gépi tanulási eljárás létezik hasonló szövegbányászati feladatokra, ezek közül a legelterjedtebb a k-közép algoritmus.

A szövegegyezőségek feltárásán kívül a modellezés terén is számos, a kutatás során nem tárgyalt megközelítést alkalmazhatunk további konklúziók alkotásához. Érdemes megvizsgálni a betegségek együttes előfordulásának hatását a halálozási korra, illetve megfigyelni annak a nemmel való kapcsolatát is.

6 Ábrajegyzék

7 Táblázatjegyzék

8 Egyenletjegyzés

9 Hivatkozások

Barabás, P. (2013). *TÉMA- ÉS NYELVADAPTÁLHATÓ TERMÉSZETES NYELVI VEZÉRLŐ KERETRENDSZER.* Miskolc.

Berk, K. N., & Carey, P. (2007). *Data Analysis with Microsoft Excel.*

Boschetti, A., & Massaron, L. (2018). *Python Data Science Essentials.* Birmingham: Packt Publishing.

Brudner, E. (2022. január 3). *22 Advantages & Disadvantages of Using Spreadsheets for Business*. Forrás: HotSpot: https://blog.hubspot.com/sales/dangers-of-using-spreadsheets-for-sales

Copestake, A. (2004). *Natural Language Processing.*

Cramér, H. (1946). *Mathematical Methods of Statistics.*

Czinkóczi, S. (2018). A szövegbányászat forradalmasíthatja a társdalomtudományokat. *444*.

Dudás, L. (2011). *Alkalmazott Mesterséges Intelligencia.*

Ferenci, T. (2022. február). *https://github.com/tamas-ferenci/C19MortalityLineListingHUN*.

Ferrari, A., & Russo, M. (2016). *Introducing Microsoft Power BI.*

Fülöp, A. (2019). 8. fejezet. In M. S. Pang-Ning Tan, *Bevezetés az adatbányászatba.* Budapest: Panem Könyvkiadó Kft.

Gonda, L. (2009). *Webbányászati módszerek alkalmazása a web.* Debrecen.

Gupta, S., Pinto, S., Sankhe-Savale, S., Gillet, J., & Cherven, K. M. (2022). *The Tableau Workshop.* Birmingham: Packt Publishing Ltd.

Halvorsen, H.-P. (2017). *Structured Query Language.*

Hartigan, J. A. (1975). *Clustering Algorithms.* New York: Wiley.

Hoover, E. M. (1936). *The Measurement of Industrial Localization.*

Hornyák, J. (2020. április). Megmutatjuk, hogy kikre nagyon veszélyes a koronavírus Magyarországon. *Portfolio*.

Jr, E. M. (1984). *An Introduction to Regional Economics.*

Kovács, E. (2014). *Többváltozós adatelemzés.*

Lawson, R. (2015). *Web Scraping with Python.* Packt Publishing.

McQuillan, M. (2015). *Introducing SQL Server.*

Müller: A magas vérnyomásban szenvedőknél súlyosabb a koronavírus. (2020. október). *HVG*.

Navarro, G. (2001). *A guided tour to approximate string matching.*

Nielsen, F. (2016). *8. Hierarchical Clustering.*

Nielsen, F. (2016). 8. Hierarchical Clustering. In F. Nielsen, *Introduction to HPC with MPI for Data Science* (old.: 195–211).

Paharia, P. T. (2023. április). Study finds positive association between obesity and COVID-19 mortality across 142 countries. *News Medical Life Sciences*.

Sapolsky, R. (2005). *Sick of Poverty.*

Schubert, E. (2021). *HACAM: Hierarchical Agglomerative Clustering Around Medoids – and its Limitations.* München.

U.S. poor died at much higher rate from COVID than rich, report says. (2022. április). *Reuters*.

Zhu, A. (2023. április 3). *Medium*. Forrás: Understanding TF-IDF and Cosine Similarity for Recommendation Engine: https://medium.com/geekculture/understanding-tf-idf-and-cosine-similarity-for-recommendation-engine-64d8b51aa9f9