# HÁZI DATA SCIENCE VERSENY

**Flyball\* csapat (Wild Runners Flyball Team – innentől csak WRFT) adatelemzés** feladat

*A keletkezett kód szándékosan nem tisztított. Benne hagytam még, a kísérleteket, befejezetlen, véglegesítendő, egyszerűsítendő részeket. Ennek oka az, hogy a verseny lezajlásával a projekt számomra még nem ért véget. A verseny szinte csak ürügy volt, hogy még komolyabban vegyem a projektet. De a fő motiváció, a tanulás, azaz Python kódolás, probléma megközelítés, elemzés, életszerű adatokon való gyakorlása. Az adatok megismerése, tisztítása az általam már jól ismert és használt Excel segítségével egyszerűbben is kivitelezhető lett volna, de a cél az volt hogy a projekt elejétől a végéig Python kóddal valósuljon meg. Kísérletezgessek a lehetőségekkel, próbálkozzak újrahasznosítható függvényekkel, beépített automatizmusokkal, több különböző adattisztítással lepróbálni az adatelemzést. Mindeközben építettem egy később is használható függvény és megoldástárat.*

Felteszem a kódot a Colabra is, a három Excel tábla feltöltése után tesztelhető az elejétől mostanáig.

2024.03.03

Wild\_Runners\_Flyball\_Team\_kutyak\_alapadatai.xlsx tísztítása "WRFT cleaning.ipynb" <https://drive.google.com/file/d/1FSNv9Hiv5BOUdhr_t_0agKpNQC_6rQ-c/view?usp=sharing>

A pontos végcél nem ismeretében, csak az alapvető tisztítások készültek el, ezek későbbi ismeretében az esetlegesen eldobott kiugró, nem valós értékek inkább korrekcióra is kerülhetnek. Kialakításra kerültek egyéni, és bináris kategória oszlopok is, az esetleges későbbi felhasználás céljából.

- Excel fájl beolvasása, itt nem kell variálni fülekkel, sor eldobással, plusz oszloppal  
- Hisztogramok, statisztikák listázása az oszlopokról, ellenőrzés céljából  
- "." -> "," csere az "ulna hossz" és "ugrási magasság" oszlopokban  
- "szuletesi datum" konvertálása dátum típusúvá  
- 'chip\_szam' típusának átalakítása objectre  
- Új oszlop létrehozása a (mai év - "szuletesi\_datum") értékkel  
- Új oszlop létrehozása ezzel a szabállyal: df\_WRFT["ulna\_ug\_mag"] = df\_WRFT["ugrasi\_magassag"] / df\_WRFT["ulna\_hossz"]  
- minden fajtát kicserélni a leghosszabb módon írt fajtára, amely az első néhány karakterével azonos.   
- Ha a leghosszabb fajtában "mix" vagy "keverék" szerepel, akkor az egészet "Keverék"-re cseréli.  
- kiugró értékek ellenőrzése, feltételezhető elírások javítása (helyiérték), Esetlegesen nem javítható sorok, eldobása.   
- Létrehozni bináris oszlopokat a DataFrame számára a megadott kategóriák alapján.  
- Írás CSV fájlba “df\_WRFT\_output.csv”  
- DataFrame beolvasása CSV fájlból ellenőrzés céljából.  
- Hisztogramok, statisztikák listázása az oszlopokról, ellenőrzés céljából

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 51 entries, 0 to 50

Data columns (total 30 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 felvezeto 51 non-null object

1 nev 51 non-null object

2 fajta 51 non-null object

3 nem 51 non-null object

4 ivar 51 non-null object

5 szuletesi\_datum 51 non-null datetime64[ns]

6 kora 51 non-null int32

7 chip\_szam 51 non-null object

8 ulna\_hossz 51 non-null float64

9 ugrasi\_magassag 51 non-null float64

10 fajta\_old 51 non-null object

11 ulna\_ug\_mag 51 non-null float64

12 fajta\_Kooikerhondje 51 non-null int64

13 fajta\_Border collie 51 non-null int64

14 fajta\_American staffordshire terrier 51 non-null int64

15 fajta\_Jack russell terrier 51 non-null int64

16 fajta\_Keverék 51 non-null int64

17 fajta\_Magyar vizsla 51 non-null int64

18 fajta\_Australian shepperd 51 non-null int64

19 fajta\_Parson russell terrier 51 non-null int64

20 fajta\_Small schnauzer 51 non-null int64

21 fajta\_Sheltie 51 non-null int64

22 fajta\_Belgian shepherd - malinois 51 non-null int64

23 fajta\_Patterdale terrier 51 non-null int64

24 fajta\_Mudi 51 non-null int64

25 fajta\_Whippet 51 non-null int64

26 nem\_kan 51 non-null int64

27 nem\_szuka 51 non-null int64

28 ivar\_ivartalan 51 non-null int64

29 ivar\_ivaros 51 non-null int64

dtypes: datetime64[ns](1), float64(3), int32(1), int64(18), object(7)

Egyedi értékek a(z) 'felvezeto' oszlopban:

Értékek Egyedi db

0 TM800418 1

1 GB900808 1

2 TU820403 1

3 KL970722 1

4 EH831207 1

5 LZ800202 1

6 OA821108 1

7 HM980701 1

8 EC940409 1

.. ... ...

42 UB870201 1

43 KL920415 1

44 BV810323 1

45 JV990222 1

46 AT991109 1

47 UZ940717 1

48 MT970826 1

49 SO990728 1

50 UH881130 1

[51 rows x 2 columns]

Egyedi értékek a(z) 'nev' oszlopban:

Értékek Egyedi db

0 Dexter 1

1 Sunny 1

2 Maggie 1

3 Them 1

4 Rocky 1

5 Dixie 1

6 Brix 1

7 Milo 1

8 Tyna 1

.. ... ...

42 Henry 1

43 Bailey 1

44 Enola 1

45 Punk 1

46 Sara 1

47 Pax 1

48 Rex 1

49 Dusty 1

50 Hank 1

[51 rows x 2 columns]

Egyedi értékek a(z) 'fajta' oszlopban:

Értékek Egyedi db

0 Border collie 26

1 Keverék 8

2 Australian shepperd 3

3 Belgian shepherd - malinois 3

4 Parson russell terrier 2

5 Kooikerhondje 1

6 American staffordshire terrier 1

7 Jack russell terrier 1

8 Magyar vizsla 1

9 Small schnauzer 1

10 Sheltie 1

11 Patterdale terrier 1

12 Mudi 1

13 Whippet 1

Egyedi értékek a(z) 'nem' oszlopban:

Értékek Egyedi db

0 kan 30

1 szuka 21

Egyedi értékek a(z) 'ivar' oszlopban:

Értékek Egyedi db

0 ivartalan 41

1 ivaros 10

Egyedi értékek a(z) 'szuletesi\_datum' oszlopban:

Értékek Egyedi db

0 2018-12-22 2

1 2021-01-04 2

2 2010-02-14 1

3 2020-04-20 1

4 2018-08-02 1

5 2018-09-02 1

6 2018-10-18 1

7 2019-01-19 1

8 2019-06-01 1

.. ... ...

40 2016-09-03 1

41 2017-04-15 1

42 2017-05-04 1

43 2017-05-31 1

44 2017-07-04 1

45 2017-08-25 1

46 2017-09-08 1

47 2017-11-17 1

48 2022-05-02 1

[49 rows x 2 columns]

Egyedi értékek a(z) 'chip\_szam' oszlopban:

Értékek Egyedi db

0 93900005238999 1

1 93900009650506 1

2 93900005172450 1

3 93900003871143 1

4 93900003818036 1

5 93900002523549 1

6 93900003669802 1

7 93900009646655 1

8 93900007664925 1

.. ... ...

42 93900008474491 1

43 93900004198045 1

44 93900002731726 1

45 93900007218892 1

46 93900003495780 1

47 93900004293075 1

48 93900004371167 1

49 93900005627226 1

50 93900008222471 1

[51 rows x 2 columns]

Egyedi értékek a(z) 'fajta\_old' oszlopban:

Értékek Egyedi db

0 Border Collie 15

1 Border 5

2 Mix 5

3 BorderCollie 3

4 border collie 3

5 Keverék 3

6 Belgian shepherd - Malinois 3

7 Australian Kelpie 2

8 Parson Russell terrier 2

9 Mudi 1

10 Patterdale terrier 1

11 Australian Shepperd 1

12 Kooikerhondje 1

13 Sheltie 1

14 Small schnauzer 1

15 Magyar Vizsla 1

16 Jack Russell Terrier 1

17 American Staffordshire Terrier 1

18 Whippet 1

EFC\_versenyadatok.xlsx "Flyball\_EFC\_cleaning\_240317.ipynb" <https://colab.research.google.com/drive/1cnsa37w3CkeCm-OLwBZ5RQI4_inSByTO?usp=sharing>

és

FLYBALLCZ\_versenyadatok.xlsx tisztítása "Flyball\_CZ\_cleaning\_240317.ipynb" <https://colab.research.google.com/drive/19qToYUXcPVjt9t-xGNmWNLu8-WyO6aQV?usp=sharing>

- Csak az első sor beolvasása az Excel fájlból, itt is látszik, hogy fejlécet csinál belőle.  
- Az első sor második cellájának értékének kinyerése  
- Excel fájl beolvasása, ahol a sorok 1 sorral át vannak ugorva, így már jól kezeli a fejléc mezőket.  
- Az első sor adatainak beszúrása minden sorba a DataFrame utolsó oszlopa utáni oszlopba. Így kiegészül az adat a helyszínnel is.  
- első oszlop átnevezéssel a dfs szótárt összemásolja df\_EFC dataframe-be.  
- Az első sor adatainak beszúrása minden sorba a DataFrame utolsó oszlopa utáni oszlopba  
- Fülek beolvasása, Fülek feldolgozása és kiíratása  
- Cserélje le minden fül 1. sorának eldobása után a maradó 1. oszlopának fejlécét "sorsz" -ra  
- Az összes DataFrame egyesítése  
- Fülek beolvasása és tárolása egy szótárban  
- Hisztogramok, statisztikák listázása az oszlopokról, ellenőrzés céljából minden terv szerint készült-e el  
- Írás CSV fájlba  
- DataFrame beolvasása CSV fájlból  
- Hisztogramok, statisztikák listázása az oszlopokról, ellenőrzés céljából

Mindkét fájlnál további tisztítás az összeillesztés után lesz elvégezve, a következő munkafázisban

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1176 entries, 0 to 1175

Data columns (total 21 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 sorsz 1176 non-null int64

1 Division 1176 non-null int64

2 when 1176 non-null datetime64[ns]

3 who 1176 non-null object

4 with who 1176 non-null object

5 total time 1176 non-null float64

6 W/L/T 1166 non-null object

7 Hurdles 1146 non-null float64

8 name 1149 non-null object

9 start 1176 non-null float64

10 1.dog 1176 non-null object

11 name.1 1141 non-null object

12 chng 1150 non-null object

13 2.dog 1168 non-null object

14 name.2 1146 non-null object

15 chng.1 1149 non-null object

16 3.dog 1166 non-null object

17 name.3 1137 non-null object

18 chng.2 1141 non-null object

19 4.dog 1157 non-null object

20 event\_place 1176 non-null object

dtypes: datetime64[ns](1), float64(3), int64(2), object(15)

memory usage: 193.1+ KB

==> df\_EFC\_output.describe() <=====================================

sorsz Division total time Hurdles start

count 1176.000000 1176.000000 1176.000000 1146.000000 1176.000000

mean 156.971939 6.916667 19.215536 22.323298 0.123605

std 96.841815 3.662093 4.775336 4.287228 0.475479

min 1.000000 1.000000 3.750000 15.000000 -0.720000

25% 75.750000 4.000000 16.617500 17.500000 0.050000

50% 149.000000 7.000000 18.255000 22.500000 0.090000

75% 231.000000 10.000000 19.952500 25.000000 0.140000

max 372.000000 13.000000 51.350000 30.000000 10.860000

<pandas.io.formats.style.Styler object at 0x78fd8bd1a020>

2024.03.15.

Flyball\_EFC\_CZ\_merging\_analyzing "Flyball\_EFC\_CZ\_merging\_analyzing\_240317.ipynb" <https://colab.research.google.com/drive/1vBq2-lo5Fu1rjElCYHjuv-3DHurwUN1x?usp=sharing>

* DataFrame beolvasása CSV fájlból "df\_EFC\_output.csv" és "df\_CZ\_output.csv"
* Oszlopnevek átnevezése, egységesítése
* a két dataframe egyesítése oszlopok mentén
* NaN értékek felkutatási módszerek tesztelése. Egyelőre nincs tisztítás
* Hisztogramok, statisztikák listázása az oszlopokról, ellenőrzés céljából
* Statisztikai kisérletezgetés, módosítás nélkül
* Írás CSV fájlba "df\_EFC\_CZ\_merged.csv"
* “DataFrame beolvasása CSV fájlból ellenőrzés céljából.
* Hisztogramok, statisztikák listázása az oszlopokról, ellenőrzés céljából

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 8573 entries, 0 to 8572

Data columns (total 21 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 sorsz 8571 non-null float64

1 Division 8571 non-null float64

2 when 8571 non-null object

3 who 8571 non-null object

4 with\_who 8571 non-null object

5 total\_time 8571 non-null float64

6 W/L/T 8464 non-null object

7 Hurdles 8530 non-null float64

8 name 8541 non-null object

9 start 8571 non-null float64

10 1\_dog 8571 non-null object

11 name\_1 8496 non-null object

12 chng 8295 non-null object

13 2\_dog 8517 non-null object

14 name\_2 8386 non-null object

15 chng\_1 8205 non-null object

16 3\_dog 8403 non-null object

17 name\_3 8323 non-null object

18 chng\_2 8133 non-null object

19 4\_dog 8316 non-null object

20 event\_place 8573 non-null object

dtypes: float64(5), object(16)

memory usage: 1.4+ MB

==> df\_EFC\_CZ.describe() <=====================================

sorsz Division total\_time Hurdles start

count 8571.000000 8571.000000 8571.000000 8530.000000 8571.000000

mean 305.887528 3.782406 19.752510 22.712778 0.117333

std 202.120140 2.626434 6.086355 4.499043 0.575569

min 1.000000 1.000000 0.000000 15.000000 -1.550000

25% 135.000000 2.000000 16.860000 20.000000 0.040000

50% 277.000000 3.000000 18.270000 22.500000 0.090000

75% 463.000000 5.000000 20.000000 27.500000 0.140000

max 824.000000 13.000000 86.030000 32.500000 25.580000

Kutyák alapadatai és versenyadatok összekapcsolása, szűrése "EFC\_CZ\_WRFT\_merging\_240317.ipynb" <https://drive.google.com/file/d/1eWFAMKOgS52vw7pXdMfRViZ1oqzeTLhx/view?usp=sharing>

* DataFrame beolvasása CSV fájlból "df\_EFC\_CZ\_merged.csv" és "df\_WRFT\_output.csv"
* leszűrjük a dataframe-ből a “wildrunners\_sorok” -at amikben a “who” oszlopok tartalmazzák a “WildRunners” töredéket és kiírjuk egy “wildrunners\_sorok” dataframe-be
* a “wildrunners\_sorok” dataframe-et
* új oszlopokat hozunk létre a "wildrunners\_sorok" DataFrame-ben "name\_desc" "name\_1\_desc" "name\_2\_desc" "name\_3\_desc" néven

"wildrunners\_sorok" DataFrame "name" "name\_1" "name\_2" "name\_3" oszlopában található nevek alapján keresünk a "df\_WRFT\_output" táblában,  
és a találatokat beletesszük a "name\_desc" oszlopokba a "wildrunners\_sorok" DataFrame-ben. De minek? Hogy 1 DF-ben legyen minden?

* Nem tudom, még, merre tovább, sötétben tapogatózom.
* Különféle lekérdezések próbálgatása, benne hagyom a kódban bár nem véglegesek.
* asszem a domain megértése hiányzik még.Nincs most több időm rá. Majdcsak jön az ihlet.
* Hisztogramok, statisztikák listázása az oszlopokról, ellenőrzés céljából

<class 'pandas.core.frame.DataFrame'>

Int64Index: 1128 entries, 1 to 8556

Data columns (total 25 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 sorsz 1128 non-null float64

1 Division 1128 non-null float64

2 when 1128 non-null datetime64[ns]

3 who 1128 non-null object

4 with\_who 1128 non-null object

5 total\_time 1128 non-null datetime64[ns]

6 W/L/T 1116 non-null object

7 Hurdles 1128 non-null float64

8 name 1128 non-null object

9 start 1128 non-null float64

10 1\_dog 1128 non-null object

11 name\_1 1125 non-null object

12 chng 1093 non-null object

13 2\_dog 1122 non-null object

14 name\_2 1119 non-null object

15 chng\_1 1101 non-null object

16 3\_dog 1119 non-null object

17 name\_3 1108 non-null object

18 chng\_2 1079 non-null object

19 4\_dog 1113 non-null object

20 event\_place 1128 non-null object

21 name\_desc 1128 non-null object

22 name\_1\_desc 1128 non-null object

23 name\_2\_desc 1128 non-null object

24 name\_3\_desc 1128 non-null object

dtypes: datetime64[ns](2), float64(4), object(19)

 [Flyball\_CZ\_cleaning\_xxxxxx.ipynb](https://drive.google.com/file/d/121q1Wq2nQXRSc6lBy3azRDG4NuYE64DH/view?usp=drive_web)

 [Flyball\_EFC\_cleaning\_xxxxxx.ipynb](https://drive.google.com/file/d/19OZcOJ25LkVZLtaU0uHyCIbaigexmz3O/view?usp=drive_web)

 [WRFT\_cleaning\_xxxxxx.ipynb](https://drive.google.com/file/d/1OohagrHhj9uRSdUUWMGzFYeKhAthFBK3/view?usp=drive_web)

 [EFC\_CZ\_WRFT\_merging\_xxxxxx.ipynb](https://drive.google.com/file/d/1d1rcGc0K-kdNTKrKEj9qkzX1QJ-Kwk-f/view?usp=drive_web)

További manipuláció tervek

* gyűjtsd ki az összes kutya nevet a Flyball\_EFC\_CZ dataframe-ból "name", "name\_1", "name\_2", "name\_3" oszlopokból egy új “minden-kutya” df-be az első oszloba, ugyaninnen a “who” oszlopból a második oszlopba “csapat” néven, aztán jöhetnek a verseny statisztikák, hány versenyen indult, hány versenyen volt mért ideje “total\_time”, hányon nyert, hányon vesztett, hány döntetlen, “W/L/T”, mekkora akadály, 1 futóként átlag, max, min, varinacia idők “1\_dog”, 2 futóként átlag, max, min, varinacia idők “2\_dog”,3 futóként átlag, max, min, varinacia idők “3\_dog”, 4 futóként átlag, max, min, varinacia idők “4\_dog”, start hiba átlag, max, min, varinacia idők, “change” átlag, max, min, varinacia idők, “change\_1” átlag, max, min, varinacia idők,“change\_2” átlag, max, min, varinacia idők., hány ok és OK átlag, max, min, varinacia % az indult futamokhoz.
* statisztikák az idő múlásával, tendenciák
* listázzuk a csapatneveket sorszámmal
* kérdezze meg a csapatnév sorszámát, listázza kutyákat (alap összes, 0 legyen wildrunners)
* hányszor futott egy kutya egy adott időszakban és abból hányszor nyert, vagy %
* verseny helyszínekből kinyerni a dátumokat

2024.03.31

Kérések:

* 1. szeretné tudni, hogy a kutyák futási teljesítményére szignifikáns hatással van-e az, hogy melyik színű pályán futnak
* 2. kíváncsi lenne arra is, hogy a kutyák futási teljesítményére szignifikáns hatással van-e az, hogy ivaros/ivartalan kanra/szukára vált rá a kutya
* 3. illetve szeretné az összes eddigi elemzést (a korábbi alap elemzéseket) versenyekhez (és helyszínekhez, és dátumokhoz) rendelve látni, mert úgy többet mondanának neki az információk

Ez a forduló nagyon megszorongat. Csak most vettem észre jó néhány aknát az adathalmazban. Gondolom gyakorlott szemmel ezt már az elején is kellett volna látni. Na nem, én inkább beleléptem, megtépázott, de még élek, rendezem soraimat. És újra elölről.

“Flyball\_EFC\_CZ\_merging\_analyzing”

* az egyesített verseny adatokat kell még tovább alakítgatni, hogy a kérdésekre választ lehessen kapni.
* először is ki kellett egészíteni a pálya adatokkal. A színek beolvasásával elment egy napom, de nem sikerült. Végül az időpont párosokkal sikerül megoldanom “track”. Ehhez kiegészült a df még “when\_date” és “when\_time” oszlopokkal. A dátum az “event\_place” oszlopból lett kinyerve.
* arra döntöttem, hogy ki kell dobálni az összes sort ami “NaN” értéket tartalmaz, meghagyva a lehetőséget kézi felülbírálásra, későbbiekben.

Ezzel el is készült a módosított egyesítése a versenyeknek:  
"df\_EFC\_CZ\_merged\_whentime\_track\_jó.csv"

Van még dolog vele, de szét kellett választanom, két részre, mert a gépem nem bírja kezelni itthon egyben.

“Flyball\_EFC\_CZ\_post\_analyzing”

A “df\_EFC\_CZ\_merged” df-et tovább alakítgatom:

* típus ellenőrzésnek vetem alá

when 7640 non-null object to Date  
chng 7640 non-null object to floa64  
chng\_1 7640 non-null object to floa64  
3\_dog 7640 non-null object to floa64

chng\_2 7640 non-null object to floa64

4\_dog 7640 non-null object to floa64

when\_date 7640 non-null object to date

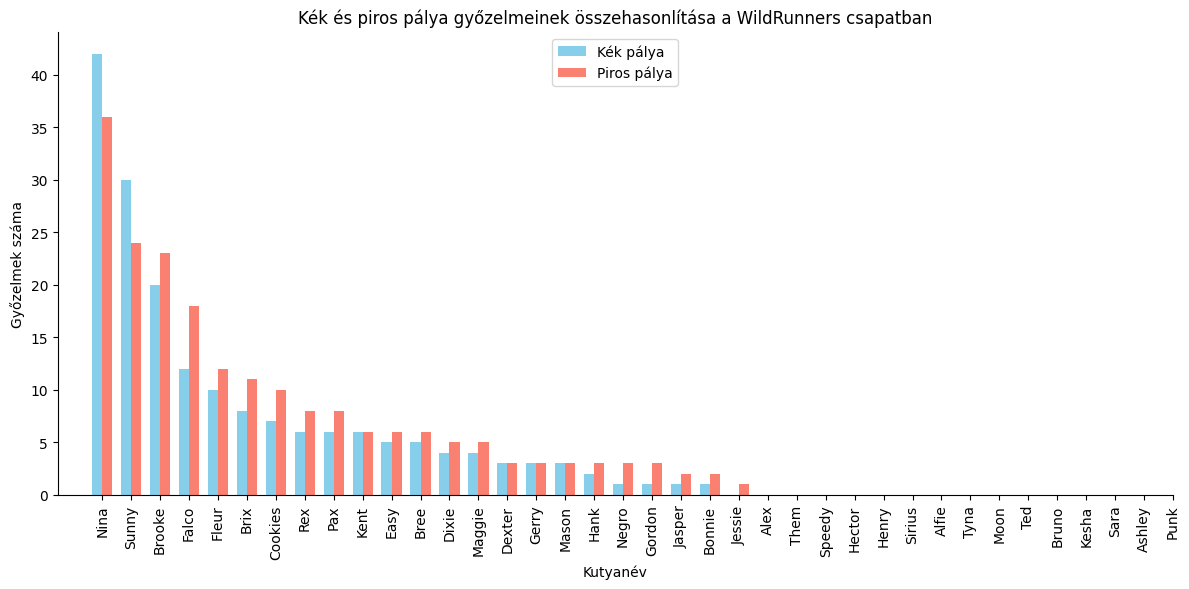
when\_time 7640 non-null object to time

* érték ellenőrzésnek vetem alá

szöveges mezők kezelése a “chng” oszlopokban

chyba= nagyon rossz? (érvénytelen)  
OK=nagyon jó ?  
ok=kicsit jó  
Ok= közepes  
error =nagyon rossz? (érvénytelen)  
vběhl= végzetes (érvénytelen)

* W/L/T oszlopot szétkategorizálni
* 1. szeretné tudni, hogy a kutyák futási teljesítményére szignifikáns hatással van-e az, hogy melyik színű pályán futnak



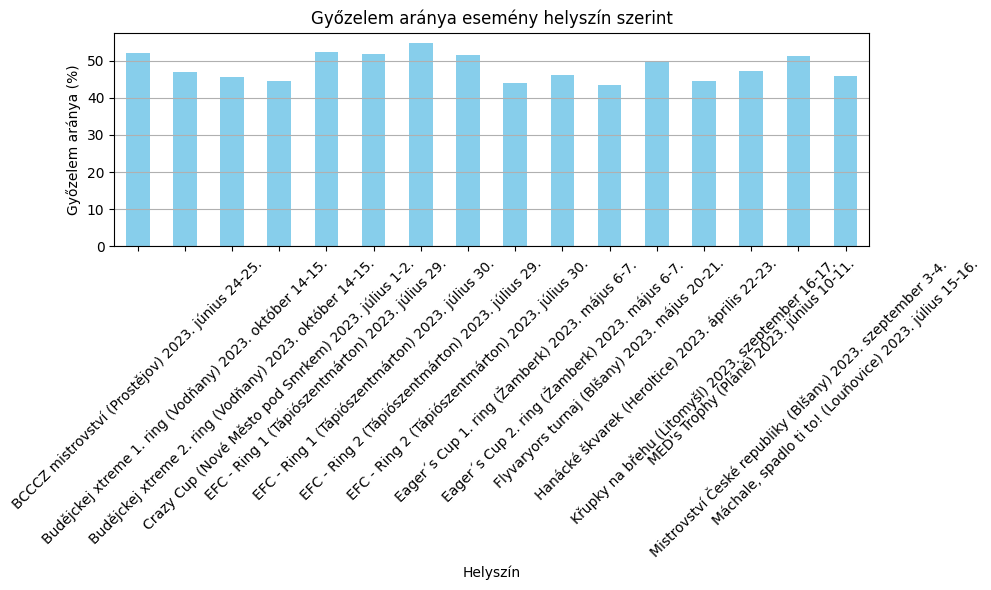
Az látszik, hogy igen van jelentős hatása van annak, hogy melyik színű pályán futnak. Az egy érdekes jelenség, hogy a legjobban teljesítő 2 kutya a kék pályán jelentősen sikeresebb, A kisebb teljesítményű kutyáknál azonban ez megfordul.

A statisztikai elemzés szerint:  
T-próba statisztika: 0.19587341020861246

p-value: 0.8459875752835451

Nincs szignifikáns különbség a két csoport között.

* 3. illetve szeretné az összes eddigi elemzést (a korábbi alap elemzéseket) versenyekhez (és helyszínekhez, és dátumokhoz) rendelve látni, mert úgy többet mondanának neki az információk



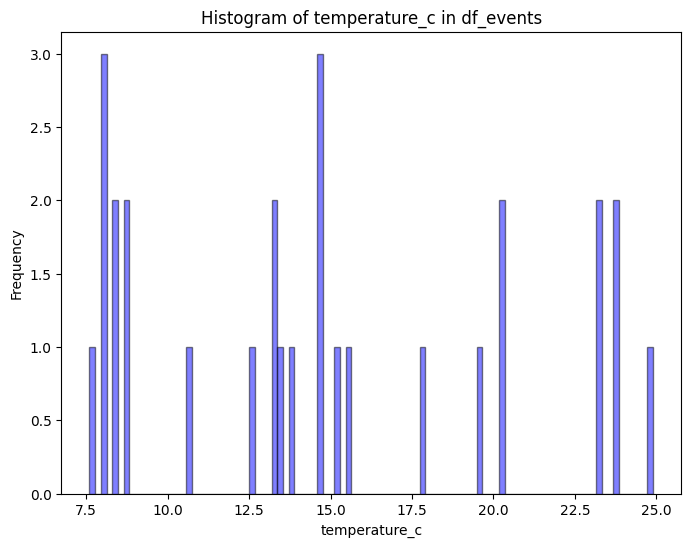
Verseny látogatás utáni tapasztalatok:

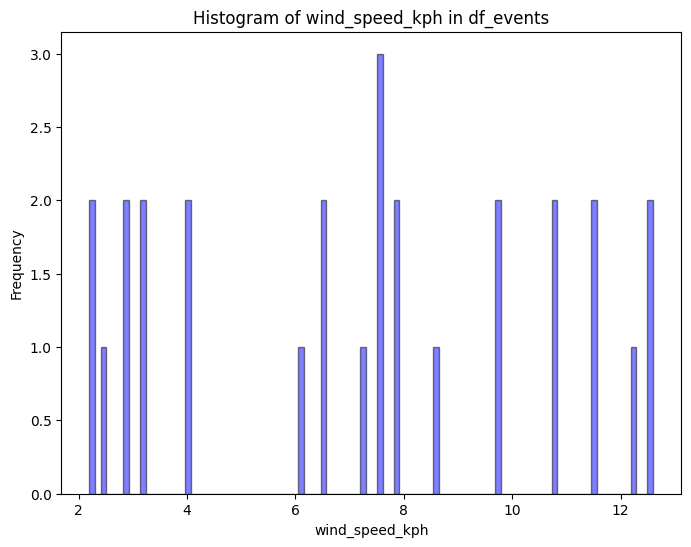
A kutyák teljesítményét a saját köridejük, és a versenyeken a stabilitásuk határozza meg. (hibák száma, egyenletes köridők, kisebb mértékben a másik pályán futó kutya, ulna/ugrási\_magasság, )  
Továbbá tesztelni lényegesen befolyásoló-e: neme, fajta, ivar, kora, felvezető személye, felvezető kora) Összehasonlítandó az összes csapat kutyáival.

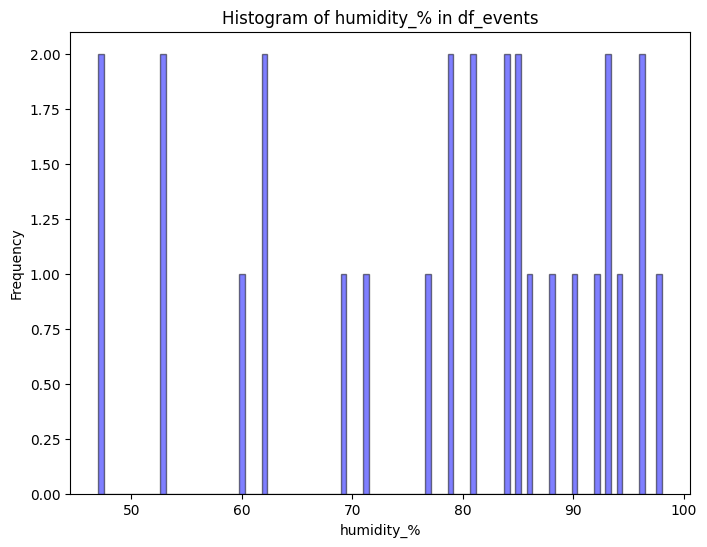
A csapatok teljesítményét több tényező is meghatározhatja. (a csapattagok, a csapattagok sorrendje, az indító és a labdaadogató személye)  
Volt hogy hibázott a labda adogató és mivel ő változhat és nincs infónk a kutya idejében, vagy hibáiban fog csak látszódni.  
A felvezető inkább állandó (általában gazdi), de ritkán mégis változhat, és erről sem ad infót az adathalmaz. Pedig a váltások az ő pontosságán múlnak leginkább.  
Továbbá érdemes tesztelni célszerű-e betenni egy nagyon alacsony kutyát a csapatba, a kisebb akadályok céljából. Ront-e a teljesítménye a kisebb láb miatt, vagy jócskán behozható a nagyobb kutyáknak az alacsonyabb akadályok miatt.

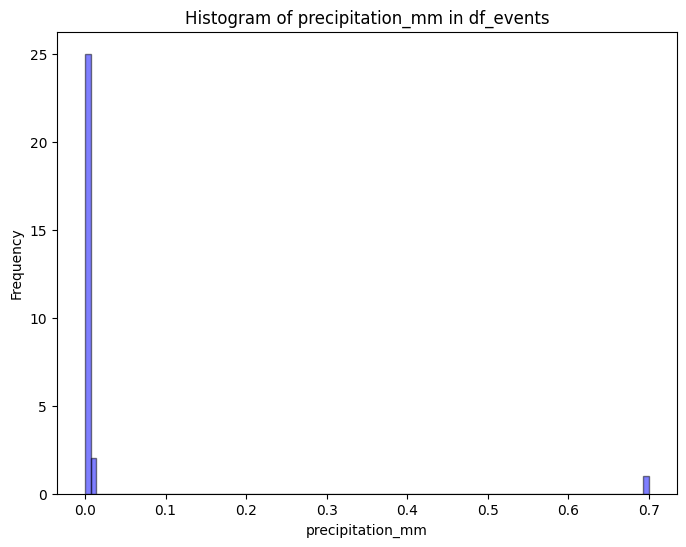
2024.04.21.

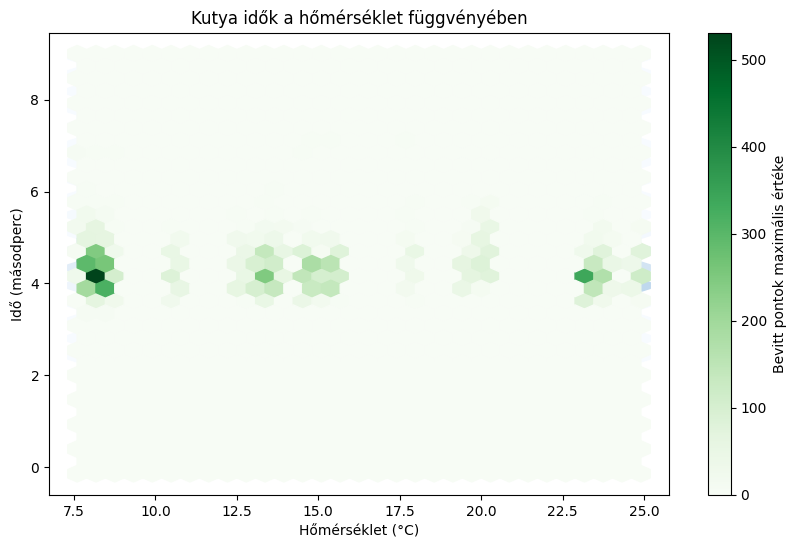
* először kidolgoztam Tomi bemutatója alapján az API lekérdezés mikéntjét félautomata módon. Utána függvény készült belőle, hogy majd végigiterálva az időpontokon, letöltse az összes szükséges adatot.
* ezután el kellett készíteni a dátumokhoz kapcsolódó helyszínek listáját.Rettenet mennyiségű próbálkozás volt automatikus Python kódot írni, ami hiba nélkül készít egy ezt tartalmazó dataframe-t. Nem sikerült az adatokban szereplő dupla napok valahogy mindig bezavartak a dátumkezelésbe. Nem szeretem az ilyesmit de itt fel kellett adnom, és kézi beavatkozással léptem túl rajta. Végül elkészült a “df\_events” dataframe.
* ennek segítségével és a “get\_weather\_history” függvénnyel lekérdeztem az összes időjárási adatot. Kiegészítettem vele a “df\_events”-et. Ez lehet majd a segítség bővíteni a verseny adatokat az időjárás adatokkal.
* Sajnos a verseny adatoknál újra előjön a kötőjelet tartalmazó dátum problémája, és az adatmennyiség miatt a kézi segítség se nagyon jöhet szóba. Lehet, hogy a kódolásba beletörik a bicskám. Van egy koncepcióm a napok szétválasztására, de egyelőre nem találom a hibát, miért nem működik jól. De mivel az értékelésnél elég jelentős súllyal számít az elemző gondolkodás, legalább leírom mi az elképzelés. A “when\_date” két egymás utáni nap is lehet. Talán az adatbázisban elfoglalt sorrendjük alapján eldönthető pontosan melyik nap. Ha a “when\_time” aktuális értéke kisebb mint az előző, akkor napot ugrottunk. Ekkor a “when\_date” értékét növelni kell eggyel. Kivéve ha az előző “when\_date” nem egyezik a mostanival. Ha az aktuális időpont nagyobb vagy egyenlő, mint az előző időpont, akkor az aktuális sor dátumát az előző sor dátumával egyenlővé teszi. ÉS SIKERÜLT, MŰKÖDIK.
* Így most jöhet “df\_EFC\_CZ” df kiegészítése időjárás adatokkal. Végigiterálok a sorokon és hozzá keresem az értékeket a “df\_events” df-ből.
* jöhetnek különféle, statisztikák, grafikonok.









Mo

Most még ilyenekkel kísérletezgezek….