Data Cleaning

What is our GOAL for this MODULE?

The goal of this module is to learn about how we clean our data to be easy to read and use.

What did we ACHIEVE in the class TODAY?

- We took the previous data, understood the meaning of all the columns and then deleted the columns that we did not need.
- We also renamed the columns to make our data more readable.

Which CONCEPTS/CODING BLOCKS did we cover today?

- Pandas (DataFrames)
- Python

How did we DO the activities?

1. Download the CSV that we achieved in the last class if you do not have it already https://github.com/whitehatjr/Data-cleaning/blob/master/final.csv

2. Let's understand the meaning of all the columns

   name **-** This is the name of the exo-planet.

   light_years_from_earth **-** This is the distance of this planet from Earth in light years. Light year is the distance light can travel in one year, and light is super fast. It can travel 9.461 Trillion km in 1 year.

   planet_mass **-** This is the mass of the planet with respect to Earth or Jupiter (Jupiter is the metric for Gas Giants while Earth is the metric for all other types of planets).

   stellar_magnitude **-** This is the brightness of the host star of the planet when observed from Earth (just as the sun is our host star).

   discovery_date **-** This is the year of discovery for the exo-planet.

   hyperlink **-** This is just the hyperlink that we scraped.

   planet_type **-** This is the type of the planet (Gas Giant, Super Earth, etc.).

   temp_planet_date **-** This is a duplicate.

   temp_planet_mass - This is another duplicate.

   planet_radius **-** This is the radius of the exo-planet with respect to Earth or Jupiter.

   orbital_radius **-** This is the average distance of this exo-planet from its sun. Just like our solar system has 1 sun, there are multiple solar systems that contain many planets and sun(s).

   orbital_period **-** This is the time it takes to complete one orbit of it's sun.

   eccentricity **-** This denotes how circular the orbit is. It might be oval in shape too. The lower the eccentricity, the more circular is the orbit.

   pl_hostname **-** The name of the host solar system.

   pl_letter **-** The letter given to this planet.

pl_name **-** The name of this planet (short version).

pl_discmethod **-** This is the discovery method which was used to find this exo-planet.

pl_controvflag **-** This is a boolean (0, 1) which says if the existence of this planet is questioned or not.

pl_pnum **-** This is the number of planets that are their in its solar system.

pl_orbper **-** This is again, the orbital period in days.

Now since we are collecting data for planets that exist so far away from us, there is no way for us to know the actual values of a planet, such as their orbital period, radius, etc. and we do calculations for it. Each calculation based on observation such as here, can have a margin of error in the actual value. Thus, all the columns with ***err1 & err2*** are the scope of errors, and we will ignore them.

pl_orbperlim **-** This is again the radius of the orbit of the planet.

pl_orbeccen **-** This is again the eccentricity of the planet.

pl_orbincl **-** This is the orbital inclination, which means that it is the tilt of the exo-planet's orbit when it revolves around its sun.

pl_bmassj **-** This is again the mass of the planet.

pl_bmassprov **-** This is the unit to calculate the mass.

pl_radj **-** This is again, the radius of the planet.

pl_dens **-** This is the density of the planet.

pl_ttvflag **-** This is a flag that indicates if this planet orbit exhibits any timing variations from other planets in the system.

pl_kepflag **-** This is a flag that tells if the solar system exhibits a planetary system (multiple planets) based on ***Kepler Field Mission.***

pl_k2flag **-** This is a flag that tells if the solar system exhibits a planetary system based on the ***K2 Mission.***

pl_nnotes **-** This is just the number of notes associated with the planet.

ra_str **-** This is the right ascension of the planetary system, which is the east-west

coordinate by which the position of this planet is measured.

dec_str **-** This is the north-south coordinate by which the position of the planet is measured.

st_dist **-** This is again the distance of the planet from Earth.

gaia_dist **-** This is again the distance of the planet from Earth in Gaia Parallax. Gaia Parallax is the coordinate that is calculated with Trigonometry.

st_optmag **-** This is the Optical magnitude (discussed earlier).

st_optband **-** There are different bands in light. This is the band of the optical magnitude.

gaia_gmag **-** This is the magnitude of the host star of the planet measured in G-Band.

st_teff **-** This is the temperature of the host star in Kelvin.

st_mass **-** This is the amount of mass contained in the host star.

st_rad **-** This is the radius of the host star.

rowupdate **-** This is the date of last update for this exo-planet.

pl_facility **-** Facility at which the planet was discovered (There are many facilities that are observing and looking for new planets/stars in our galaxy).

3. Now that we understand the meaning of these columns, we will create a new directory, create a virtual environment inside it and then we will source the virtual environment.
4. We first import pandas and then create a dataframe with our csv final.py.
5. Once the df is created, we will print the shape of the dataframe.

```
import pandas as pd
import csv

df = pd.read_csv("final.csv")
print(df.shape)
```

6. The output of this would be **(4284, 85).**

7. This means that we have 4,284 rows (the same number of rows as the number of exo-planets that we found) and we have 85 columns!

8. Below is the list of all the columns that we want to delete. These columns are either repeated, displaying the error values or is irrelevant data to the study we want to conduct:
   Hyperlink
   Temp_planet_date
   Temp_planet_mass
   Pl_letter
   Pl_name
   Pl_controvflag
   Pl_pnum
   Pl_orbper
   Pl_orbpererr1
   Pl_orbpererr2
   Pl_orbperlim
   Pl_orbsmax
   Pl_orbsmaxerr1
   Pl_orbsmaxerr2
   Pl_orbsmaxlim
   Pl_orbeccen
   Pl_orbeccenerr1
   Pl_orbeccenerr2
   Pl_orbeccenlim
   Pl_orbinclerr1
   Pl_orbinclerr2
   Pl_orbincllim
   Pl_bmassj
   Pl_bmassjerr1
   Pl_bmassjerr2
   Pl_bmassjlim
   Pl_bmassprov
   Pl_radj
   Pl_radjerr1
   Pl_radjerr2
   Pl_radjlim

Pl_denserr1
Pl_denserr2
Pl_denslim
Pl_ttvflag
Pl_kepflag
Pl_k2flag
Pl_nnotes
Ra
Dec
St_dist
St_disterr1
St_disterr2
St_distlim
Gaia_dist
Gaia_disterr1
Gaia_disterr2
Gaia_distlim
St_optmag
St_optmagerr
St_optmaglim
St_optband
Gaia_gmag
Gaia_gmagerr
Gaia_gmaglim
St_tefferr1
St_tefferr2
St_tefflim
St_masserr1
St_masserr2
St_masslim
St_raderr1
St_raderr2
St_radlim
Rowupdate
Pl_facility

9. To remove a column from the dataframe, we can write the following command:

```
del df["hyperlink"]
```

10. Similarly, to remove all the columns we listed above, we will do:

```
del df["hyperlink"]
del df["temp_planet_date"]
del df["temp_planet_mass"]
del df["pl_letter"]
del df["pl_name"]
del df["pl_controvflag"]
del df["pl_pnum"]
del df["pl_orbper"]
del df["pl_orbpererr1"]
del df["pl_orbpererr2"]
del df["pl_orbperlim"]
del df["pl_orbsmax"]
del df["pl_orbsmaxerr1"]
del df["pl_orbsmaxerr2"]
del df["pl_orbsmaxlim"]
del df["pl_orbeccen"]
del df["pl_orbeccenerr1"]
del df["pl_orbeccenerr2"]
del df["pl_orbeccenlim"]
del df["pl_orbinclerr1"]
del df["pl_orbinclerr2"]
del df["pl_orbincllim"]
del df["pl_bmassj"]
del df["pl_bmassjerr1"]
del df["pl_bmassjerr2"]
del df["pl_bmassjlim"]
del df["pl_bmassprov"]
del df["pl_radj"]
```

```
del df["pl_radjerr1"]
del df["pl_radjerr2"]
del df["pl_radjlim"]
del df["pl_denserr1"]
del df["pl_denserr2"]
del df["pl_denslim"]
del df["pl_ttvflag"]
del df["pl_kepflag"]
del df["pl_k2flag"]
del df["pl_nnotes"]
del df["ra"]
del df["dec"]
del df["st_dist"]
del df["st_disterr1"]
del df["st_disterr2"]
del df["st_distlim"]
del df["gaia_dist"]
del df["gaia_disterr1"]
del df["gaia_disterr2"]
del df["gaia_distlim"]
del df["st_optmag"]
del df["st_optmagerr"]
del df["st_optmaglim"]
del df["st_optband"]
del df["gaia_gmag"]
del df["gaia_gmagerr"]
del df["gaia_gmaglim"]
del df["st_tefferr1"]
del df["st_tefferr2"]
del df["st_tefflim"]
```

```
del df["st_masserr1"]
del df["st_masserr2"]
del df["st_masslim"]
del df["st_raderr1"]
del df["st_raderr2"]
del df["st_radlim"]
del df["rowupdate"]
del df["pl_facility"]
```

11. If we now print the shape of the df (**print(df.shape)**), we get:

```
(4284, 85)
(4284, 19)
```

12. We deleted a total of 66 Columns! Now, let's print the names of the columns with command:

```
print(list(df))
```

```
['name', 'light_years_from_earth', 'planet_mass', 'stellar_magnitude', 'discovery_date',
'planet_type', 'planet_radius', 'orbital_radius', 'orbital_period', 'eccentricity', 'pl_h
ostname', 'pl_discmethod', 'pl_orbincl', 'pl_dens', 'ra_str', 'dec_str', 'st_teff', 'st_m
ass', 'st_rad']
```

13. As we can see here, all the headers up until eccentricity looks good and descriptive, however all the headers after that are very abstract and hard to read. We will now change these headers by writing:

```
df = df.rename({
        'pl_hostname': "solar_system_name",
        'pl_discmethod': "planet_discovery_method",
```

```
            'pl_orbincl': "planet_orbital_inclination",
            'pl_dens': "planet_density",
            'ra_str': "right_ascension",
            'dec_str': "declination",
            "st_teff": "host_temperature",
            'st_mass': "host_mass",
            'st_rad': "host_radius"
        }, axis='columns')
```

14. Here, after changing all the headers, if we print the list of all the headers, we can see the following output:

```
['name', 'light_years_from_earth', 'planet_mass', 'stellar_magnitude', 'discovery_date',
'planet_type', 'planet_radius', 'orbital_radius', 'orbital_period', 'eccentricity', 'sola
r_system_name', 'planet_discovery_method', 'planet_orbital_inclination', 'planet_density'
, 'right_ascension', 'declination', 'host_temperature', 'host_mass', 'host_radius']
```

15. We will finally move this data with new headers and reduced columns into a new CSV, which will be the CSV that we use for all our data analysis and machine learning classifiers from here now.

```
df.to_csv('main.csv')
```

What's NEXT?

In the next class, we will begin with data analytics where we will plot different charts and try to find out insights about our data.

EXTEND YOUR KNOWLEDGE:

You can read the following blog on data cleaning to understand more:
https://medium.com/machine-intelligence-team/data-cleaning-with-python-d0ca811d6cdf