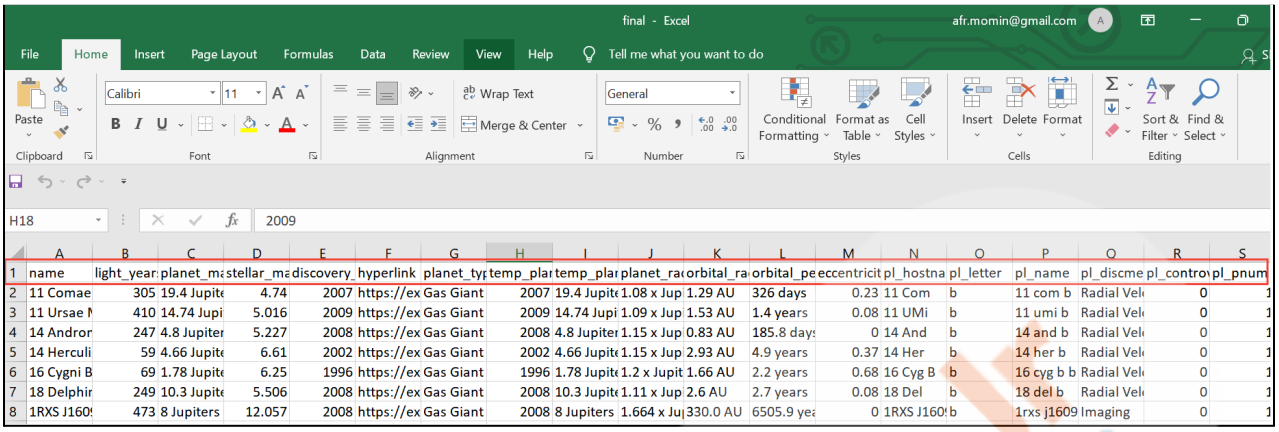
Topic	DATA CLEANING	
Class Description	**The student will clean the data that was retrieved in the previous classes.**	
Class	**PRO C130**	
Class time	**45 mins**	
Goal	• Understanding and reviewing data • To remove extra columns and rename the column headers	

Resources Required	• Teacher Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	  ○ Smartphone
	• Student Resources:
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen	

Class structure	**Warm-Up** **Teacher-Led Activity 1** **Student-Led Activity 1** **Wrap-Up**	**10 mins** **10 mins** **20 mins** **05 mins**
Credit & Permissions:	**Exoplanet Exploration by NASA** **Pandas by NumFOCUS**	
WARM-UP SESSION - 10 mins		
Teacher Starts Slideshow **Slide # to #** <**Note**: Only Applicable for Classes with VA> Refer to speaker notes and follow the instructions on each slide.		

Teacher Action	Student Action
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? **Following are the WARM-UP session deliverables:** • Greet the student. • Revision of previous class activities. • Quizzes.	**ESR**: Hi, thanks! Yes I am excited about it! Click on the slide show tab and present the slides.
WARM-UP QUIZ Click on In-Class Quiz	
Continue WARM-UP Session **Slide # to #** <**Note**: Only Applicable for Classes with VA>	
Activity Details **Following are the session deliverables:** • Appreciate the student. • Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.	
Teacher Action	**Student Action**
In the last class, we added more data and merged the two databases that we had. Can you recall the logic we used to merge the two CSVs? Great! Now in today's class, we will understand the meaning of all the columns that we have, in our CSV, and then we will see how we can clean our data. Cleaning the data means how we can remove all the unwanted data. It is an important step in	**ESR**: We first sorted the data from the second database in alphabetical order irrespective of whether it is uppercase or lowercase and then we merged the two!

data-preprocessing to make it easier for us to use it for analysis. In the previous class we removed the column using the CSV library but today we'll be using Pandas to clean the data. Are you excited?	**ESR:** Yes
Teacher Ends Slideshow 	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
ACTIVITY • **To make the student understand the meaning of all the columns** • **To clean the data using Pandas**	
Teacher Action	**Student Action**
In the last class, we have merged the two datasets and saved them in the **final.csv** file. ***Note:*** *Open Teacher Activity 2 and show the dataset to the student. The data is updated from time to time. Thus column names and the number of rows and columns may differ from previously processed data. Thus to give an idea about data cleaning* ***final.csv*** *is given with merged datasets.* Let's just start by understanding the meaning of all the columns one by one.	



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	name	light_year	planet_m	stellar_ma	discovery_	hyperlink	planet_typ	temp_plar	temp_plar	planet_ra	orbital_ra	orbital_pe	eccentricit	pl_hostna	pl_letter	pl_name	pl_discme	pl_controv	pl_pnum
2	11 Comae	305	19.4 Jupite	4.74	2007	https://ex	Gas Giant	2007	19.4 Jupite	1.08 x Jup	1.29 AU	326 days	0.23	11 Com	b	11 com b	Radial Vel	0	1
3	11 Ursae M	410	14.74 Jupi	5.016	2009	https://ex	Gas Giant	2009	14.74 Jupi	1.09 x Jup	1.53 AU	1.4 years	0.08	11 UMi	b	11 umi b	Radial Vel	0	1
4	14 Andror	247	4.8 Jupiter	5.227	2008	https://ex	Gas Giant	2008	4.8 Jupiter	1.15 x Jup	0.83 AU	185.8 days	0	14 And	b	14 and b	Radial Vel	0	1
5	14 Herculi	59	4.66 Jupite	6.61	2002	https://ex	Gas Giant	2002	4.66 Jupite	1.15 x Jup	2.93 AU	4.9 years	0.37	14 Her	b	14 her b	Radial Vel	0	1
6	16 Cygni B	69	1.78 Jupite	6.25	1996	https://ex	Gas Giant	1996	1.78 Jupite	1.2 x Jupit	1.66 AU	2.2 years	0.68	16 Cyg B	b	16 cyg b b	Radial Vel	0	1
7	18 Delphir	249	10.3 Jupite	5.506	2008	https://ex	Gas Giant	2008	10.3 Jupite	1.11 x Jup	2.6 AU	2.7 years	0.08	18 Del	b	18 del b	Radial Vel	0	1
8	1RXS J160	473	8 Jupiters	12.057	2008	https://ex	Gas Giant	2008	8 Jupiters	1.664 x Ju	330.0 AU	6505.9 yea	0	1RXS J160	b	1rxs j1609	Imaging	0	1

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. 1 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 planet (Gas Giant, Super-Earth, etc.).

planet_radius - This is the radius of the exoplanet 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, multiple solar systems contain many planets and sun(s).

orbital_period - This is the time it takes to complete one orbit of its 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 that 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 there 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. so we do calculations for it. Each calculation is based on observation such as here, which can have a margin of error in the actual value. Thus, all the columns with ***err1 and 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 exoplanet'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's 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**. Kepler Field Mission is a telescope designed to survey our galaxy Milky way to survey the exoplanets in the northern hemisphere of the sky.

pl_k2flag - This is a flag that tells if the solar system exhibits a planetary system based on the ***K2 Mission.*** Kepler Field Mission was renamed by NASA as K2 Mission which focused on hunting for exoplanets in both northern and southern hemispheres.

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. It is the measure of the brightness of a star or celestial body. For brighter objects, lower numbers are assigned as the optical

magnitude. ***st_optband -*** There are different bands in light. This is the band of the optical magnitude. ***gaia_gmag*** - Gaia is the mission to determine the position and velocity of a billion stars, creating the largest and most precise 3D map of the Milky Way. ***gaia_gmag*** is the photometric measure of a star's visibility. ***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. ***row_update -*** This is the date of the last update for this exo-planet. ***pl_facility -*** Facility at which the planet was discovered (Many facilities are observing and looking for new planets/stars in our galaxy).	
Great! Now we understand the meaning of all the rows. Let's dive into the data-cleaning part now! I will first import pandas as **pd** and then I will import **csv** to create the final output csv after cleaning this data. Yes. So we'll be accessing our data through Pandas by using DataFrames. 1. Get the data into Google Colab using the **git clone** command.	

```
!git clone https://github.com/procodingclass/PRO-NASA-Exoplanet-Processed-Data

Cloning into 'PRO-NASA-Exoplanet-Processed-Data'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
```

2. Import **pandas** library.
3. Read **final.csv** using the **read_csv()** method of pandas.
4. Print the shape of this DataFrame.This will return the number of rows and columns.

```
import pandas as pd

#read csv file
final_planets_df = pd.read_csv("/content/PRO-NASA-Exoplanet-Processed-Data/final.csv")

#check number of rows and columns
print(final_planets_df.shape)

(4284, 85)
```

As you can see here we have a total of 4284 rows and 85 columns. We have enormous data here. Let's discard unnecessary columns from this dataset.
You have to do it using pandas I'll be helping you through it.

Teacher Stops Screen Share

So now it's your turn.
Please share your screen with me.

Teacher Starts Slideshow
Slide # to #

<**Note**: Only Applicable for Classes with VA> Refer to speaker notes and follow the instructions on each slide.	
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.	**ESR:** Yes
Teacher Ends Slideshow 	
STUDENT-LED ACTIVITY - 20 mins	
• **Ask the student to press the ESC key to come back to the panel.** • **Guide the student to start Screen Share.** • **The teacher gets into Full Screen.**	
Student Initiates Screen Share	
ACTIVITY • **Student codes to remove unwanted columns** • **The student makes necessary changes to the data**	
Teacher Action	**Student Action**
Do you remember, in the previous class we tried to remove the column but extra rows were added while performing the operation? We can avoid it while we use Pandas to remove the column. Open Student Activity 1 to get the dataset link. Download it and save it in a directory. ***Note:*** *Guide the student to download the file final.csv and main.py. Save it in a directory and open it with VS Code.* First, let's list down all the columns that we want to remove. You can create a **text file** and copy-paste their names here for our convenience.	**ESR:** Yes

Note: *Guide the student to copy and paste all the column names in the text file. The one which is to be kept in our dataset can be removed from this list.*

hyperlink
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*	

col_remove.txt

```
1   hyperlink
2   temp_planet_date
3   temp_planet_mass
4   pl_letter
5   pl_name
6   pl_controvflag
7   pl_pnum
8   pl_orbper
9   pl_orbpererr1
10  pl_orbpererr2
11  pl_orbperlim
12  pl_orbsmax
13  pl_orbsmaxerr1
14  pl_orbsmaxerr2
15  pl_orbsmaxlim
```

That is a lot of columns to remove! All of it is the data that we do not require, hence we can delete it.

1. Use the **head()** method to print the DataFrame **final_planets_df.**

```
final_planets_df.head()
```

	name	light_years_from_earth	planet_mass	stellar_magnitude	discovery_date
0	11 Comae Berenices b	305.0	19.4 Jupiters	4.740	2007
1	11 Ursae Minoris b	410.0	14.74 Jupiters	5.016	2009
2	14 Andromedae b	247.0	4.8 Jupiters	5.227	2008
3	14 Herculis b	59.0	4.66 Jupiters	6.610	2002
4	16 Cygni B b	69.0	1.78 Jupiters	6.250	1996

5 rows × 85 columns

2. Print the column names of the DataFrame using the **columns()** method.

```
final_planets_df.columns
```

```
Index(['name', 'light_years_from_earth', 'planet_mass', 'stellar_magnitude',
       'discovery_date', 'hyperlink', 'planet_type', 'temp_planet_date',
       'temp_planet_mass', 'planet_radius', 'orbital_radius', 'orbital_period',
       'eccentricity', 'pl_hostname', 'pl_letter', 'pl_name', 'pl_discmethod',
       '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_orbincl', '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_dens', 'pl_denserr1',
       'pl_denserr2', 'pl_denslim', 'pl_ttvflag', 'pl_kepflag', 'pl_k2flag',
       'pl_nnotes', 'ra_str', 'ra', 'dec_str', '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_teff', 'st_tefferr1', 'st_tefferr2', 'st_tefflim',
       'st_mass', 'st_masserr1', 'st_masserr2', 'st_masslim', 'st_rad',
       'st_raderr1', 'st_raderr2', 'st_radlim', 'rowupdate', 'pl_facility'],
      dtype='object')
```

3. Delete all the columns that we do not want using the **drop()** method of Pandas.
4. Using **inplace=True** with the method will modify the current DataFrame.

```
final_planets_df.drop(columns=['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'], inplace=True)
```

5. Check the number of rows and columns using the **shape()** method.

```
print(final_planets_df.shape)
```

```
(4284, 19)
```

We deleted 66 columns!!

6. Check the DataFrame now. Here some column headers are present in the short form.

```
final_planets_df.head()
```

pl_hostname	pl_discmethod	pl_orbincl	pl_dens	ra_str	dec_str	st_teff	st_mass	st_rad
11 Com	Radial Velocity	NaN	NaN	12h20m43.03s	+17d47m34.3s	4742.0	2.70	19.00
11 UMi	Radial Velocity	NaN	NaN	15h17m05.89s	+71d49m26.0s	4213.0	2.78	29.79
14 And	Radial Velocity	NaN	NaN	23h31m17.42s	+39d14m10.3s	4813.0	2.20	11.00
14 Her	Radial Velocity	NaN	NaN	16h10m24.31s	+43d49m03.5s	5338.0	0.90	0.93
16 Cyg B	Radial Velocity	NaN	NaN	19h41m51.97s	+50d31m03.1s	5750.0	1.08	1.13

Let's change the name of these headers and make them more readable. Our headers look fine up until eccentricity. We will change the others with the following names:

No.	Column name	Replace with
1	pl_hostname'	solar_system_name
2	pl_discmethod	planet_discovery_method
3	pl_orbinc	planet_orbital_inclination
4	pl_dens	planet_density
5	ra_str	right_ascension
6	dec_str	declination
7	st_teff	host_temperature
8	st_mass	host_mass
9	st_rad	host_radius

We'll use the **rename()** method of Pandas. It takes the following parameters:

1. Dictionary with **key** as the **current column name** and **value** as the **new name**.
2. The **axis** of DataFrame defines the direction of operation to be performed on DataFrame. It can be either columns or rows.
3. After replacing the names of columns let's check the headers again.

```
#rename headers
final_planets_df = final_planets_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')
```

Save and run this file. To check the output use the **head()** method.

```
final_planets_df.head(50)
```

eccentricity	solar_system_name	planet_discovery_method	planet_orbital_inclination
0.23	11 Com	Radial Velocity	NaN
0.08	11 UMi	Radial Velocity	NaN
0.0	14 And	Radial Velocity	NaN
0.37	14 Her	Radial Velocity	NaN
0.68	16 Cyg B	Radial Velocity	NaN
0.08	18 Del	Radial Velocity	NaN

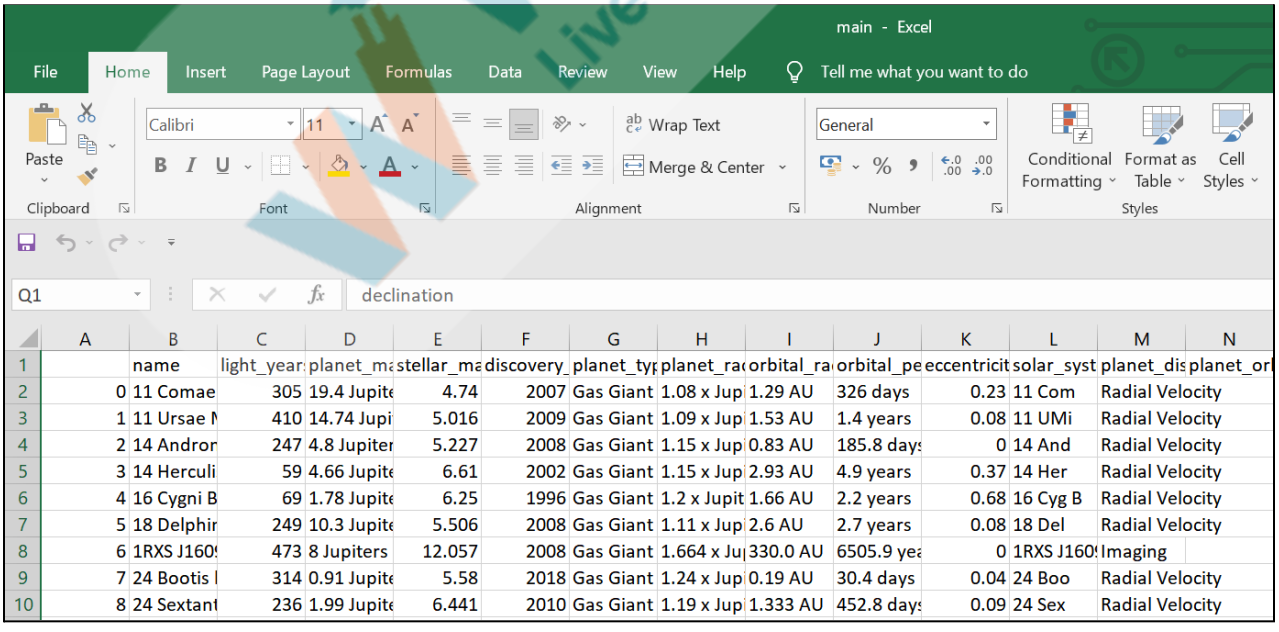
So the headers are replaced with new names. Now the data is in the desired form we can save this and use it for further processing. Let's save this in another CSV file. Since we are using Pandas, the **to_csv()** method does the job for us. In this method, we have to pass the name of the new CSV file. Let's call it **main.csv.**

Also, Use the **download()** method from **files** module to download the file.

```
#save to csv file
final_planets_df.to_csv('main.csv')

# Download CSV
from google.colab import files
files.download('main.csv')
```

Let's check the file whether we have the clean data or not. Go to the directory in which you saved the python file. Now, open main.csv to check the dataset.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		name	light_year	planet_ma	stellar_ma	discovery_	planet_typ	planet_ra	orbital_ra	orbital_pe	eccentricit	solar_syst	planet_dis	planet_or
2	0	11 Comae	305	19.4 Jupite	4.74	2007	Gas Giant	1.08 x Jup	1.29 AU	326 days	0.23	11 Com	Radial Velocity	
3	1	11 Ursae M	410	14.74 Jupi	5.016	2009	Gas Giant	1.09 x Jup	1.53 AU	1.4 years	0.08	11 UMi	Radial Velocity	
4	2	14 Andror	247	4.8 Jupiter	5.227	2008	Gas Giant	1.15 x Jup	0.83 AU	185.8 days	0	14 And	Radial Velocity	
5	3	14 Herculi	59	4.66 Jupite	6.61	2002	Gas Giant	1.15 x Jup	2.93 AU	4.9 years	0.37	14 Her	Radial Velocity	
6	4	16 Cygni B	69	1.78 Jupite	6.25	1996	Gas Giant	1.2 x Jupit	1.66 AU	2.2 years	0.68	16 Cyg B	Radial Velocity	
7	5	18 Delphir	249	10.3 Jupite	5.506	2008	Gas Giant	1.11 x Jup	2.6 AU	2.7 years	0.08	18 Del	Radial Velocity	
8	6	1RXS J160	473	8 Jupiters	12.057	2008	Gas Giant	1.664 x Ju	330.0 AU	6505.9 yea	0	1RXS J160	Imaging	
9	7	24 Bootis I	314	0.91 Jupite	5.58	2018	Gas Giant	1.24 x Jup	0.19 AU	30.4 days	0.04	24 Boo	Radial Velocity	
10	8	24 Sextant	236	1.99 Jupite	6.441	2010	Gas Giant	1.19 x Jup	1.333 AU	452.8 days	0.09	24 Sex	Radial Velocity	

Great! Our data looks much more clean and readable now. We have reduced the number of columns from 85 to 19 and we have made our headers much more readable. It looks like our data is ready to be used to perform statistics. We are done!	

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 05 mins

Teacher Starts Slideshow

Slide # to #
<**Note**: Only Applicable for Classes with VA>

Activity details

Following are the WRAP-UP session deliverables:

- Appreciate the student.
- Revise the current class activities.
- Discuss the quizzes.

WRAP-UP QUIZ
Click on In-Class Quiz

Continue WRAP-UP Session

Slide # to #
<**Note**: Only Applicable for Classes with VA>

Activity Details

Following are the session deliverables:

- Explain the facts and trivia
- Next class challenge
- Project for the day
- Additional Activity (Optional)

FEEDBACK

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
You get “hats-off” for your excellent work! In the next class, we will be applying some statistics to this data to see if we can find interesting facts about exoplanets.	*Make sure you have given at least 2 hats-off during the class for:* 

PROJECT OVERVIEW DISCUSSION

Refer the document below in Activity Links Sections

Teacher Clicks ✖ End Class

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Previous Class Code	https://colab.research.google.com/drive/1ilGwmFoTz44f8x1oyHwFJBPmoE2-S99K?usp=sharing
Teacher Activity 2	Dataset	https://github.com/procodingclass/PRO-C130-Dataset
Teacher Activity 3	Reference Code	https://colab.research.google.com/drive/1lsW0VslDq6wQ6fQz_o-gnEY-yOxdgJbQ?usp=sharing
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/df145145-f6f6-4a73-9c5d-7dc7f4030db2.pdf
Teacher Reference 2	Project Solution	https://colab.research.google.com/drive/1_Z6WYYOlTJ3XVWgmq0zKPfv8zUNOlalk?usp=sharing
Teacher Reference 3	Visual-Aid	Will be added after VA creation
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/6414d775-dfd4-4ece-95df-7698b4ab3da1.pdf
Student Activity 1	Boilerplate Code	https://github.com/procodingclass/PRO-C130-Boilerplate-Code