Topic	Data Pre-Processing	
Class Description	**Students will review the data collected in the previous classes, download more data and merge the datasets into one.**	
Class	**C129**	
Class time	**45 mins**	
Goal	• Understanding and reviewing data • Merging multiple datasets and pre-processing the data	

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

Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**5 mins** **15 min** **15 min** **5 min**

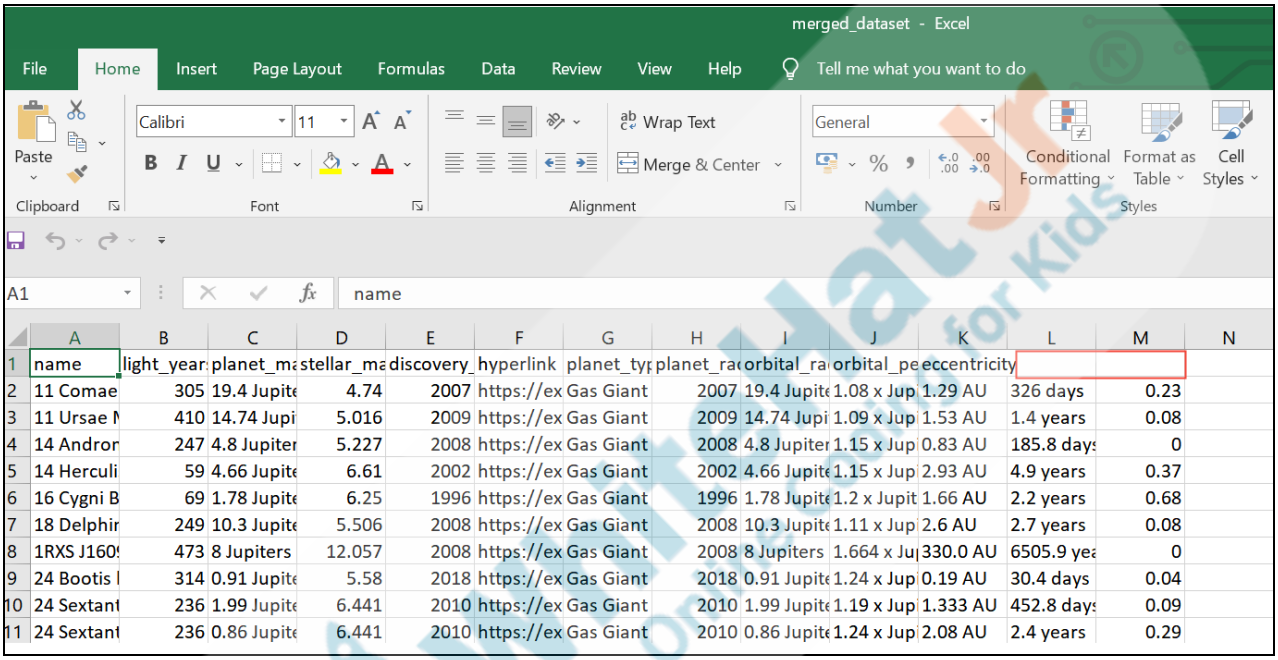
CONTEXT

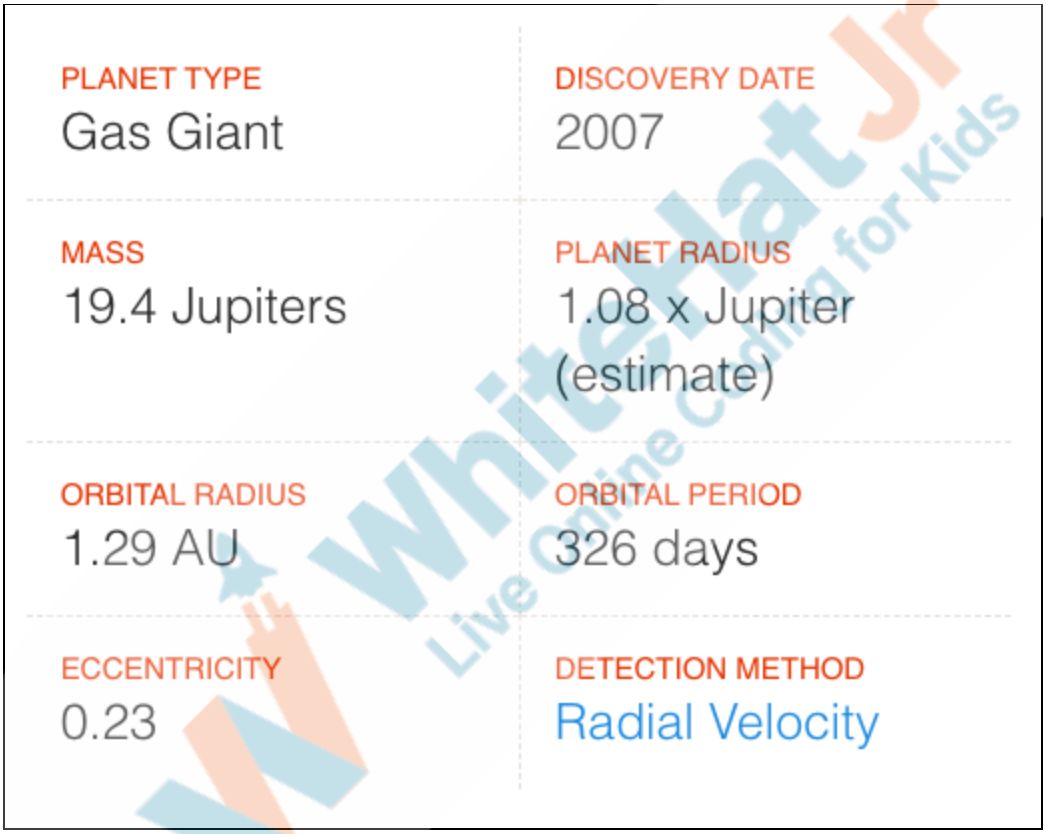
- **Review the concepts learned in the earlier classes**

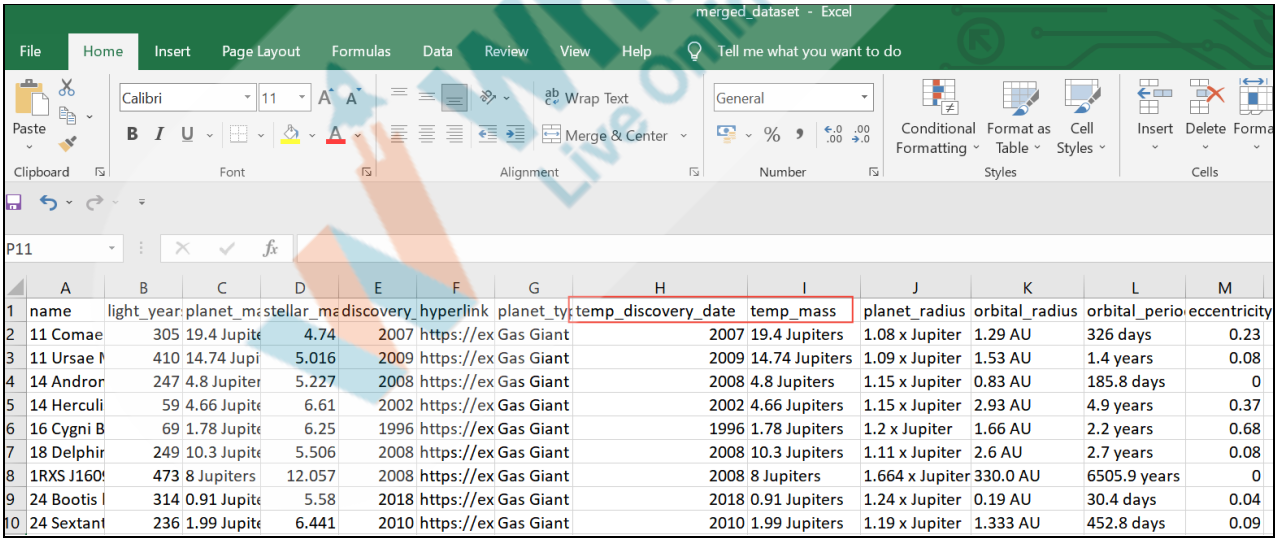
Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	Hi <Student Name>! In the last class, we completed scraping data from NASA's website. We scraped data from exo-planets. Do you know what exo-planets are?	**ESR:** Exo-planets are those planets that we have found outside our own solar system.

	That's right! Now in today's class, we will be looking at the data we just scraped in the last class. We will also download some more data from an existing website and finally we will merge the data as we pre-process it. Are you excited?	**ESR:** "Yes!"
	I have an exciting quiz question for you! Are you ready to answer this question? Teacher click on the  button on the bottom right corner of your screen to start the In-Class Quiz. A quiz will be visible to both you and the student. Encourage the student to answer the quiz question. The student may choose the wrong option, help the student to think correctly about the question and then answer again. After the student selects the correct option, the  button will start appearing on your screen. Click the End quiz to close the quiz pop-up and continue the class.	ESR: Yes

	Let's get started!	
Teacher Initiates Screen Share		
CHALLENGE • **Looking at the previous data** • **Downloading more data from the internet and then merge the data**		
Step 2: Teacher-led Activity (15 min)	*(Before beginning the class, please make the student download the CSV from the link below. This CSV is the final output of all the data scraped in the previous class.)* *<Teacher can download final.csv from Teacher Activity 1>* https://github.com/procodingclass/PRO-129-Datasets	*<Student can download from Student Activity 1>*
	Teacher asks the student to open the CSV that has all the data we scraped (The CSV was provided by us in the last class).	
	The first thing that we need to do is to cross verify our data and make sure it's right. It could lead to a lot of problems later on if the data that we have is not right. To start, let's just check the number of columns in the header of our csv and the number of columns in one of the rows in the csv.	*Student counts the number of items in the header and then one of the rows of the CSV.*

	How many items do we have in the header and how many in one of the rows?	**ESR:** We have 11 items in the header and 13 items in the rows.
	 A B C D E F G H I J K L M N 1 name light_year planet_ma stellar_ma discovery_ hyperlink planet_typ planet_ra orbital_ra orbital_pe eccentricity 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 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 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 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 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 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 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 9 24 Bootis I 314 0.91 Jupite 5.58 2018 https://ex Gas Giant 2018 0.91 Jupite 1.24 x Jup 0.19 AU 30.4 days 0.04 10 24 Sextant 236 1.99 Jupite 6.441 2010 https://ex Gas Giant 2010 1.99 Jupite 1.19 x Jup 1.333 AU 452.8 days 0.09 11 24 Sextant 236 0.86 Jupite 6.441 2010 https://ex Gas Giant 2010 0.86 Jupite 1.24 x Jup 2.08 AU 2.4 years 0.29	
	Okay! Now this data looks seriously wrong. The number of items in the header should be equal to the number of items in the rows. Let's cross check all the items in the header and the row to see what headers we are missing. Everything up until the hyperlink looks good, meaning we made a mistake when we scraped data from the hyperlinks.	

	Let's open the first hyperlink from the CSV.	*Student opens the first hyperlink:* https://exoplanets.nasa.gov/exoplanet-catalog/6988/11-comae-berenices-b/
		
	Great! Now if we look closely, our data tells us that we first scraped the ***Planet Type*** and then we scraped the ***Discovery Date***. If we look at our headers though, we are missing the discovery date header after planet type.	*Student adds the temporary header* ***temp_discovery_date*** *header after* ***planet_type****.*

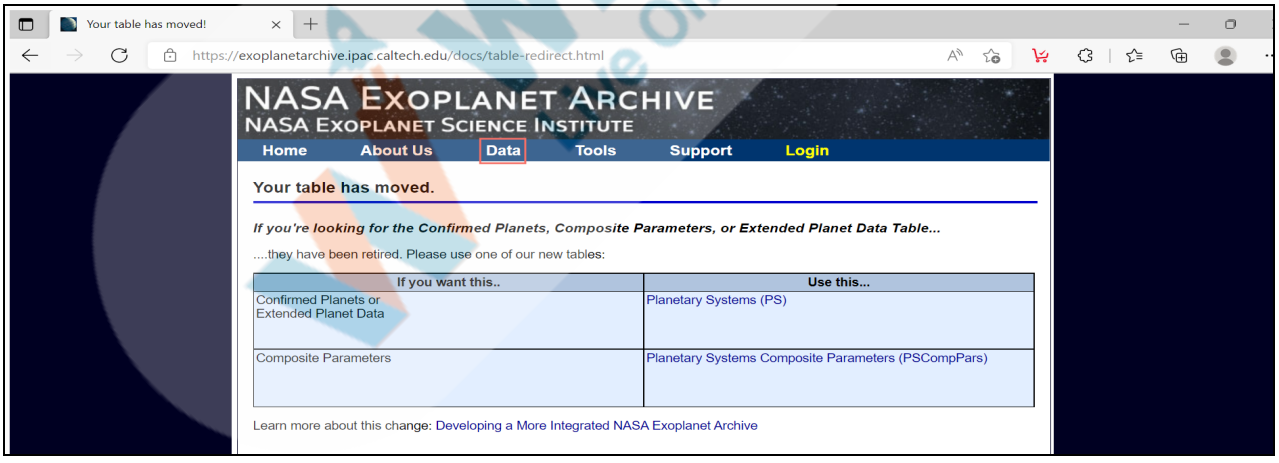
	We already scraped this data earlier, and that's why we didn't add this header, but we didn't filter it out either. For now, the best we can do is to add a temporary header in the CSV and we will remove this entire column when we do data cleaning in the next class.	
	Great, but we still have 13 items in our rows and 12 items in our headers. Can you find the next one and add a header to the CSV for it?	*Student finds the next missing header and adds it.* *(Right after the **temp_discovery_date** we just added, we need to add **temp_mass**)*
		
	Great! Now our data looks clean and we have 13 items in our header and 13 items in our rows as well.	

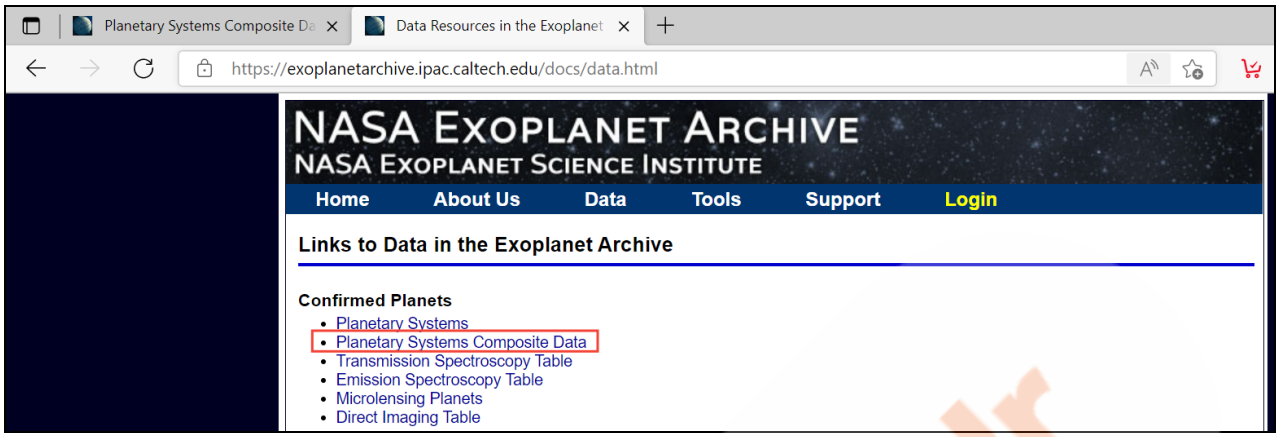
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	name	light_year	planet_ma	stellar_ma	discovery_	hyperlink	planet_ty	temp_discovery_date	temp_mass	planet_radius	orbital_radius	orbital_perio	eccentricity
2	11 Comae	305	19.4 Jupite	4.74	2007	https://ex	Gas Giant	2007	19.4 Jupiters	1.08 x Jupiter	1.29 AU	326 days	0.23
3	11 Ursae M	410	14.74 Jupi	5.016	2009	https://ex	Gas Giant	2009	14.74 Jupiters	1.09 x Jupiter	1.53 AU	1.4 years	0.08
4	14 Andror	247	4.8 Jupiter	5.227	2008	https://ex	Gas Giant	2008	4.8 Jupiters	1.15 x Jupiter	0.83 AU	185.8 days	0
5	14 Herculi	59	4.66 Jupite	6.61	2002	https://ex	Gas Giant	2002	4.66 Jupiters	1.15 x Jupiter	2.93 AU	4.9 years	0.37
6	16 Cygni B	69	1.78 Jupite	6.25	1996	https://ex	Gas Giant	1996	1.78 Jupiters	1.2 x Jupiter	1.66 AU	2.2 years	0.68
7	18 Delphir	249	10.3 Jupite	5.506	2008	https://ex	Gas Giant	2008	10.3 Jupiters	1.11 x Jupiter	2.6 AU	2.7 years	0.08
8	1RXS J160	473	8 Jupiters	12.057	2008	https://ex	Gas Giant	2008	8 Jupiters	1.664 x Jupiter	330.0 AU	6505.9 years	0
9	24 Bootis I	314	0.91 Jupite	5.58	2018	https://ex	Gas Giant	2018	0.91 Jupiters	1.24 x Jupiter	0.19 AU	30.4 days	0.04
10	24 Sextant	236	1.99 Jupite	6.441	2010	https://ex	Gas Giant	2010	1.99 Jupiters	1.19 x Jupiter	1.333 AU	452.8 days	0.09

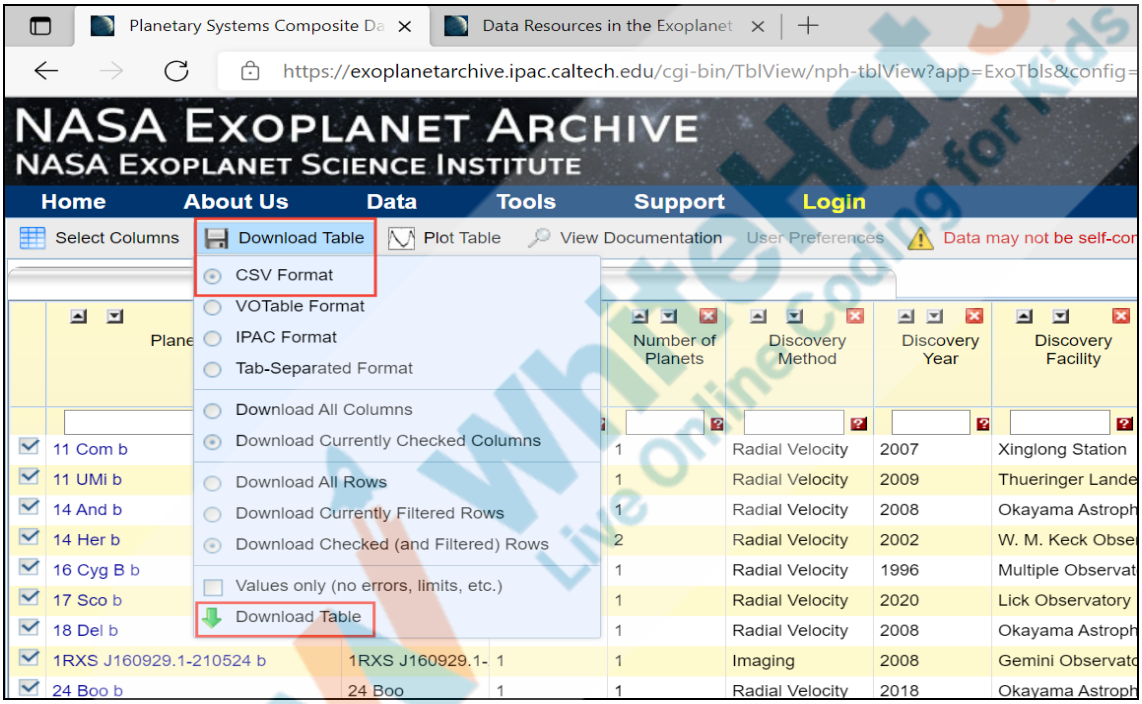
Now, we can see we have a total of 4,284 exo-planets' data. What we will do next is that there is another website that has even more data. We will download the data in CSV format from there and then we will merge the two data-sets.

For this, we will go to the following link:
https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=planets
<Teacher opens the link from Teacher Activity 2>

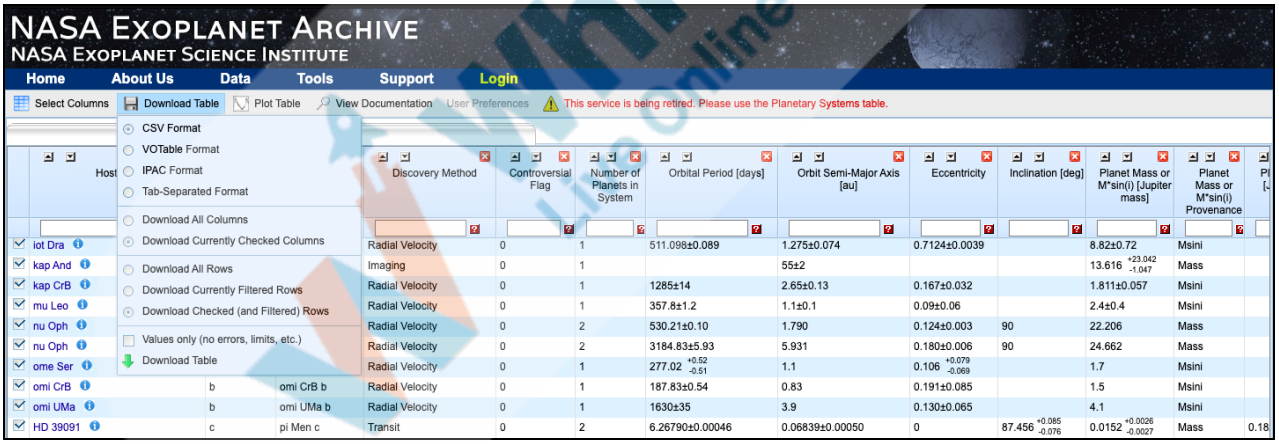
Follow the steps given below to download the dataset.







	Note: *The teacher can use the link directly to go to the table. The link may change due to the timely updation of data.* Here, we can again see at the bottom of the screen that this table	*Student opens the link from* ***Student Activity 2*** *and downloads the CSV.*

	has data of 5014 planets. This is exactly what we need! ***Note:*** *Number of planets may change as data is updated time to time.* This table has a lot of data that will help us analyze a lot of data about: our universe, which planets are habitable and which ones are not. Let's download this data! For downloading it, we will click on the Download Table button at the top of the table and select CSV type and then click on the Download Table option with a green arrow.	
		
	Great! Now, we need to merge the data.	
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

<table>
<tr><td colspan="3">

- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

</td></tr>
<tr><td colspan="3">

ACTIVITY

- **Student tries to find patterns between the data**
- **Student writes some code to merge the 2 data sets**

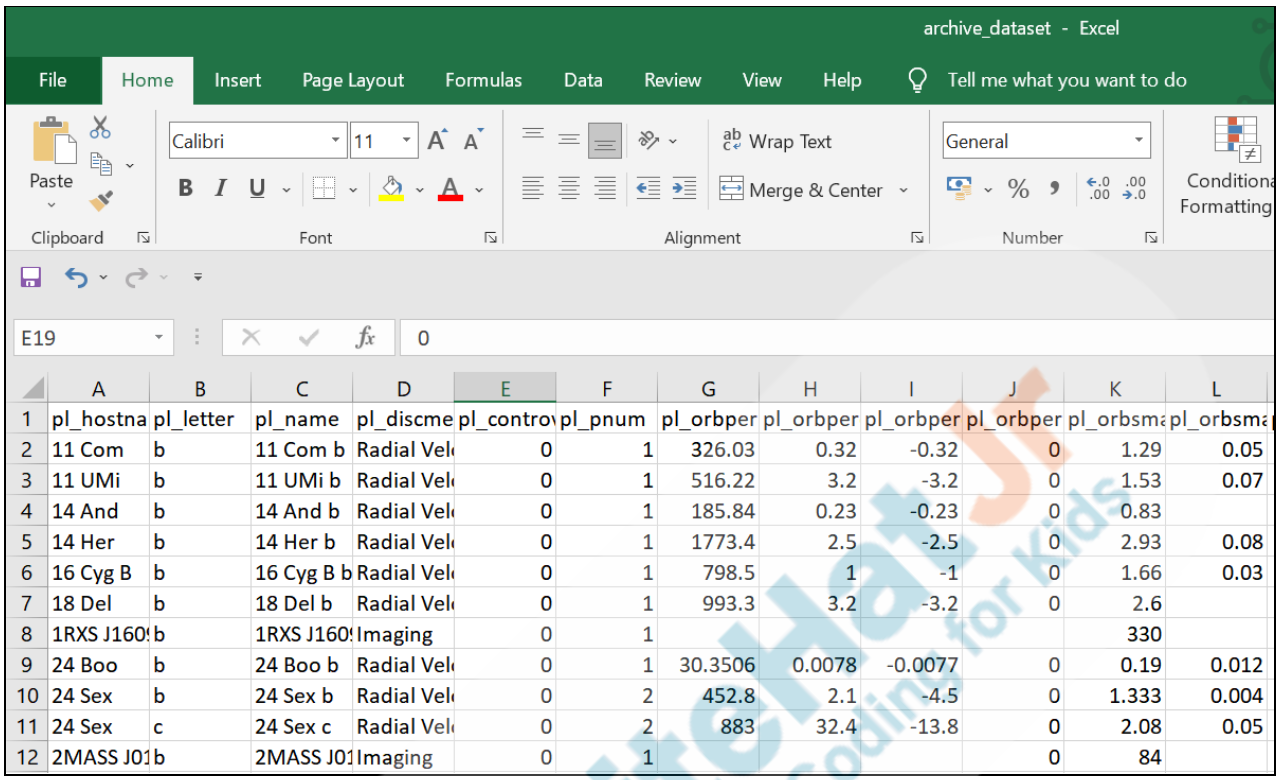
</td></tr>
<tr><td>Step 3:
Student-Led Activity
(15 min)</td><td><Open Student Activity 1 for Datasets. final.csv is the dataset from previous class and archive_dataset.csv is the dataset that we have downloaded from NASA's website>

https://github.com/procodingclass/PRO-129-Datasets

Okay, now the first thing that we have to do is that we need to look at the data and try to identify a pattern within the data.

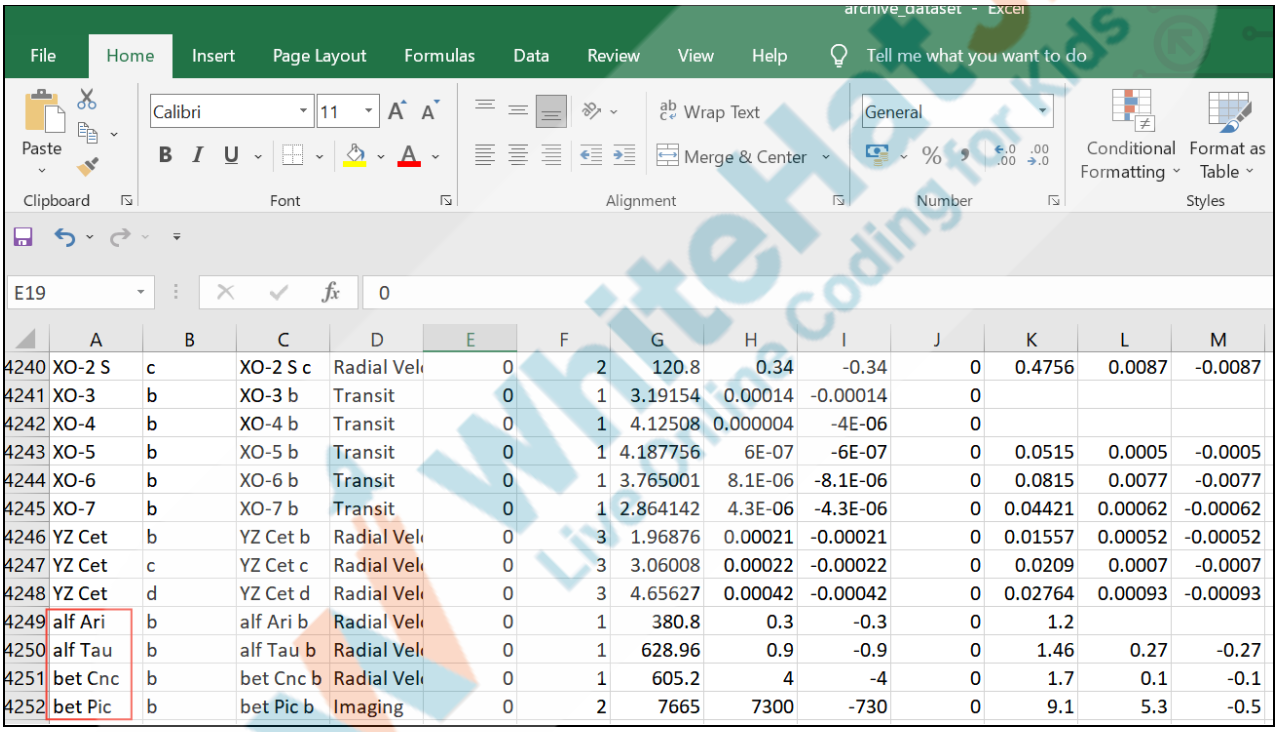
If we look at the names of these planets in both the data sets, we can see that the data we scraped earlier has a full version of the names while the planet data we downloaded has a shorter version of names.</td><td><i>Student looks carefully.</i></td></tr>
</table>

NAME ↑	LIGHT-YEARS FROM EARTH	PLANET MASS	STELLAR MAGNITUDE	DISCOVERY DATE
11 Comae Berenices b	305	19.4 Jupiters	4.74	2007
11 Ursae Minoris b	410	14.74 Jupiters	5.016	2009
14 Andromedae b	247	4.8 Jupiters	5.227	2008
14 Herculis b	59	4.66 Jupiters	6.61	2002
16 Cygni B b	69	1.78 Jupiters	6.25	1996
18 Delphini b	249	10.3 Jupiters	5.506	2008
1RXS J160929.1-210524 b	473	8 Jupiters	12.057	2008
24 Bootis b	314	0.91 Jupiters	5.58	2018
24 Sextantis b	236	1.99 Jupiters	6.441	2010
24 Sextantis c	236	0.86 Jupiters	6.441	2010



	A	B	C	D	E	F	G	H	I	J	K	L
1	pl_hostna	pl_letter	pl_name	pl_discme	pl_contro	pl_pnum	pl_orbper	pl_orbper	pl_orbper	pl_orbper	pl_orbsma	pl_orbsma
2	11 Com	b	11 Com b	Radial Vel	0	1	326.03	0.32	-0.32	0	1.29	0.05
3	11 UMi	b	11 UMi b	Radial Vel	0	1	516.22	3.2	-3.2	0	1.53	0.07
4	14 And	b	14 And b	Radial Vel	0	1	185.84	0.23	-0.23	0	0.83	
5	14 Her	b	14 Her b	Radial Vel	0	1	1773.4	2.5	-2.5	0	2.93	0.08
6	16 Cyg B	b	16 Cyg B b	Radial Vel	0	1	798.5	1	-1	0	1.66	0.03
7	18 Del	b	18 Del b	Radial Vel	0	1	993.3	3.2	-3.2	0	2.6	
8	1RXS J160	b	1RXS J160	Imaging	0	1					330	
9	24 Boo	b	24 Boo b	Radial Vel	0	1	30.3506	0.0078	-0.0077	0	0.19	0.012
10	24 Sex	b	24 Sex b	Radial Vel	0	2	452.8	2.1	-4.5	0	1.333	0.004
11	24 Sex	c	24 Sex c	Radial Vel	0	2	883	32.4	-13.8	0	2.08	0.05
12	2MASS J01	b	2MASS J01	Imaging	0	1				0	84	

	The name **11 Comae Berenices b** from the data we scraped is written as **11 Com b** in the data we downloaded. Similarly, all the names are different. Therefore we cannot use the names of these data points as a metric to merge the two. Can you think of any patterns that both of the datasets follow?	*Student tries to find a pattern for a couple of minutes.* **ESR:** The names are in alphabetical order.

	That's great! But there is one thing we need to be careful about. Both the datasets are in alphabetical order, however the second dataset (with tables) has the planet names that starts with a lower case alphabet in the bottom, segregated separately in alphabetical order while the data we scraped is in perfect alphabetical order.	*Student looks carefully.*
		
	Here, we can see the dataset has all the planet names until alphabet Y in alphabetical order, and then all the names with lower case alphabet are below it in alphabetical order as well. What do you think we need to do first to merge the two datasets?	**ESR:** We need to arrange the second dataset in alphabetical order irrespective of if the name is uppercase or lower case.

	A	B	C	D	E	F	G	H	I	J	K	L	M
4240	XO-2 S	c	XO-2 S c	Radial Vel	0	2	120.8	0.34	-0.34	0	0.4756	0.0087	-0.0087
4241	XO-3	b	XO-3 b	Transit	0	1	3.19154	0.00014	-0.00014	0			
4242	XO-4	b	XO-4 b	Transit	0	1	4.12508	0.000004	-4E-06	0			
4243	XO-5	b	XO-5 b	Transit	0	1	4.187756	6E-07	-6E-07	0	0.0515	0.0005	-0.0005
4244	XO-6	b	XO-6 b	Transit	0	1	3.765001	8.1E-06	-8.1E-06	0	0.0815	0.0077	-0.0077
4245	XO-7	b	XO-7 b	Transit	0	1	2.864142	4.3E-06	-4.3E-06	0	0.04421	0.00062	-0.00062
4246	YZ Cet	b	YZ Cet b	Radial Vel	0	3	1.96876	0.00021	-0.00021	0	0.01557	0.00052	-0.00052
4247	YZ Cet	c	YZ Cet c	Radial Vel	0	3	3.06008	0.00022	-0.00022	0	0.0209	0.0007	-0.0007
4248	YZ Cet	d	YZ Cet d	Radial Vel	0	3	4.65627	0.00042	-0.00042	0	0.02764	0.00093	-0.00093
4249	alf Ari	b	alf Ari b	Radial Vel	0	1	380.8	0.3	-0.3	0	1.2		
4250	alf Tau	b	alf Tau b	Radial Vel	0	1	628.96	0.9	-0.9	0	1.46	0.27	-0.27
4251	bet Cnc	b	bet Cnc b	Radial Vel	0	1	605.2	4	-4	0	1.7	0.1	-0.1
4252	bet Pic	b	bet Pic b	Imaging	0	2	7665	7300	-730	0	9.1	5.3	-0.5

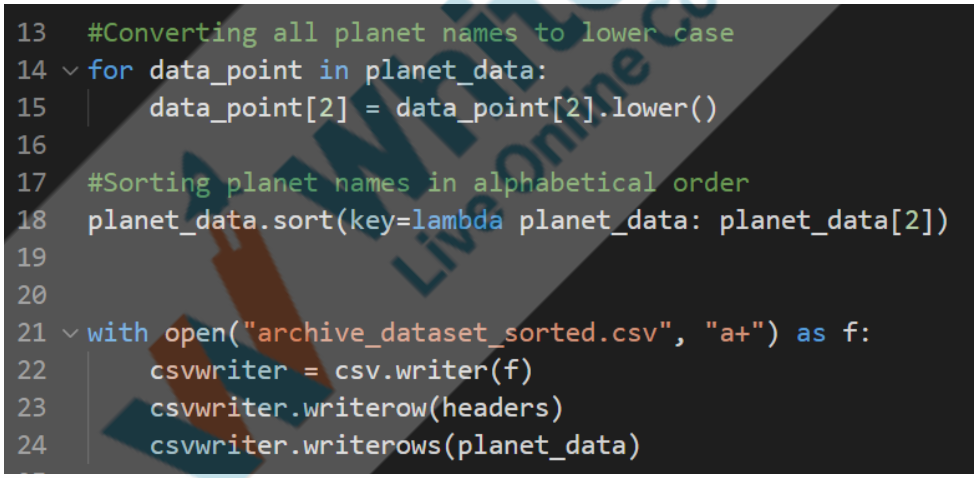
	That's right! Let's write a code that can quickly do that. Let's start by creating a virtual environment in a new directory: **python3.8 -m venv venv** Let's source the virtual environment: MACOS/UBUNTU:- **source venv/bin/activate** WINDOWS:- **venv\Scripts\activate.bat**	*Students create a virtual environment.*
	Great! Now we will move the CSV we just downloaded from the website to this directory, and we need to read this CSV in a python script. With python script, we want to create 2 variables where first one would only hold the headers and second one would hold all the planet data. Before we write a script, there is some extra data in the csv file we need to remove. All the lines starting with **#** is something that we don't need.	*Student moves the CSV to the new directory, removes the extra data and then creates a python script and reads the CSV data.*

	Once these lines are removed, we will write the script to sort the data in **acrchive_sorted.py** file.	

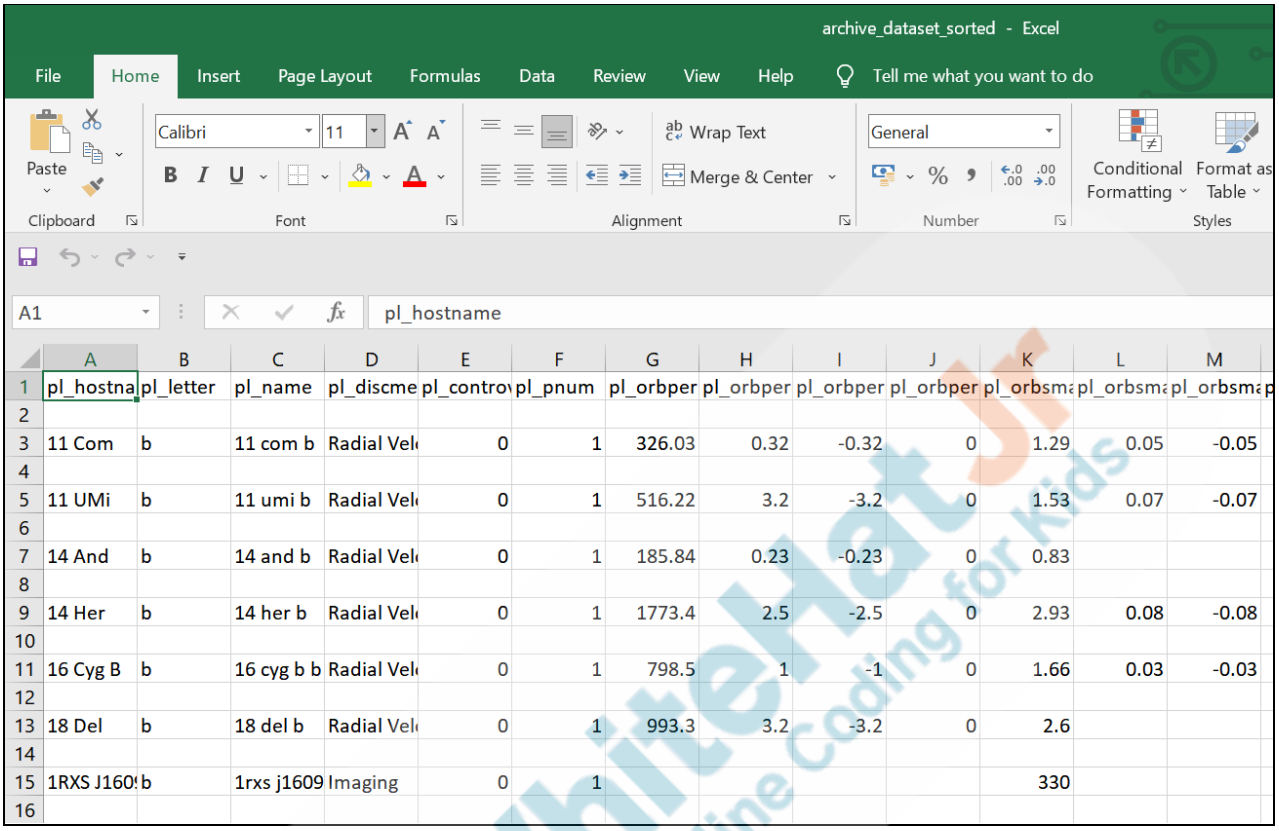
archive_sorted.py > ...

```
1  import csv
2
3  data = []
4
5  with open("archive_dataset.csv", "r") as f:
6      csvreader = csv.reader(f)
7      for row in csvreader:
8          data.append(row)
9
10 headers = data[0]
11 planet_data = data[1:]
```

	With this code, now we have all the headers in the variable **headers** and all the planet data in variable **planet_data**. Now, we just need to arrange this in alphabetical order and generate a new CSV! To arrange the data in alphabetical order, we will use the column **pl_name**, which is the third column. ***Fun Fact -*** All the lower case alphabets have a higher ascii value (lowercase a has an ascii of 97) and all the uppercase alphabets have a lower	**ESR:** We can convert all the planet names to either lowercase or uppercase and then sort the list in ascending order.

	ascii value (uppercase A has an ascii of 65). When comparing strings in python, strings get compared with their ASCII value. Using the fun-fact above, can we device a logic that can help us segregate this data in alphabetical order?	
	That's right! Let's quickly code it then. Here means that we want to sort the planet_data list, but since it is a list of lists, we want to sort it based on the 3rd element (at index 2) of the list.	*Student codes to arrange planet data in alphabetical order using the third column and then generates a new CSV.*
		
	Now, this should generate our new CSV. Let's compare the previous CSV that we created after scraping data with this one to see if their first and last rows are for the same exo-planet or not.	*Student compares the two.*

```
13  #Converting all planet names to lower case
14  for data_point in planet_data:
15      data_point[2] = data_point[2].lower()
16
17  #Sorting planet names in alphabetical order
18  planet_data.sort(key=lambda planet_data: planet_data[2])
19
20
21  with open("archive_dataset_sorted.csv", "a+") as f:
22      csvwriter = csv.writer(f)
23      csvwriter.writerow(headers)
24      csvwriter.writerows(planet_data)
```



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	pl_hostna	pl_letter	pl_name	pl_discme	pl_contro	pl_pnum	pl_orbper	pl_orbper	pl_orbper	pl_orbper	pl_orbsma	pl_orbsma	pl_orbsma
2													
3	11 Com	b	11 com b	Radial Vel	0	1	326.03	0.32	-0.32	0	1.29	0.05	-0.05
4													
5	11 UMi	b	11 umi b	Radial Vel	0	1	516.22	3.2	-3.2	0	1.53	0.07	-0.07
6													
7	14 And	b	14 and b	Radial Vel	0	1	185.84	0.23	-0.23	0	0.83		
8													
9	14 Her	b	14 her b	Radial Vel	0	1	1773.4	2.5	-2.5	0	2.93	0.08	-0.08
10													
11	16 Cyg B	b	16 cyg b b	Radial Vel	0	1	798.5	1	-1	0	1.66	0.03	-0.03
12													
13	18 Del	b	18 del b	Radial Vel	0	1	993.3	3.2	-3.2	0	2.6		
14													
15	1RXS J160	b	1rxs j1609	Imaging	0	1					330		
16													

Looks like they are in the same order! But there is an extra line added in the code. We have to remove this line and save it in another csv.

```
#remove blank lines
with
open('archive_dataset_sort
ed.csv') as input,
open('archive_dataset_sort
ed1.csv', 'w', newline='')
as output:
    writer =
csv.writer(output)
```

```
    for row in
csv.reader(input):
        if
any(field.strip() for
field in row):

writer.writerow(row)
```

```
#remove blank lines
with open('archive_dataset_sorted.csv') as input, open('archive_dataset_sorted1.csv', 'w', newline='')
    writer = csv.writer(output)
    for row in csv.reader(input):
        if any(field.strip() for field in row):
            writer.writerow(row)
```

Now we just have to write one more script where we can merge the two CSVs.

First we will have to move our previous data's CSV into this folder and then we will have to write a script. Let's do that quickly in a separate python file called **data_preprocessing.py**

Here, we'll use **final.csv**(data scraped and merged in the previous class).

Also, the preprocessed dataset

Student moves the previous CSV and then writes a script to merge the two.

```
import csv

dataset_1 = []
```

```
dataset_2 = []

with open("final.csv",
"r") as f:
    csvreader =
csv.reader(f)
    for row in csvreader:

dataset_1.append(row)

with
open("archive_dataset_sort
ed1.csv", "r") as f:
    csvreader =
csv.reader(f)
    for row in csvreader:

dataset_2.append(row)

headers_1 = dataset_1[0]
planet_data_1 =
dataset_1[1:]

headers_2 = dataset_2[0]
planet_data_2 =
dataset_2[1:]

headers = headers_1 +
headers_2
planet_data = []
```

```
for index, data_row in
enumerate(planet_data_1):

planet_data.append(planet_
data_1[index] +
planet_data_2[index])

with
open("merged_dataset.csv",
"a+") as f:
    csvwriter =
csv.writer(f)

csvwriter.writerow(headers
)

csvwriter.writerows(planet
_data)
```

data_preprocessing.py > ...

```python
import csv

dataset_1 = []
dataset_2 = []


with open("final.csv", "r") as f:
    csvreader = csv.reader(f)
    for row in csvreader:
        dataset_1.append(row)

with open("archive_dataset_sorted1.csv", "r") as f:
    csvreader = csv.reader(f)
    for row in csvreader:
        dataset_2.append(row)

headers_1 = dataset_1[0]
planet_data_1 = dataset_1[1:]

headers_2 = dataset_2[0]
planet_data_2 = dataset_2[1:]

headers = headers_1 + headers_2
planet_data = []
for index, data_row in enumerate(planet_data_1):
    planet_data.append(planet_data_1[index] + planet_data_2[index])

with open("merged_dataset.csv", "a+") as f:
    csvwriter = csv.writer(f)
    csvwriter.writerow(headers)
    csvwriter.writerows(planet_data)

```

	Great! We now have all the data merged in the merged_dataset.csv file.z	
Teacher Guides Student to Stop Screen Share		
FEEDBACK • **Appreciate the student for their efforts** • **Identify 2 strengths and 1 area of progress for the student**		
Step 4: Wrap-Up (5 min)	So in this class, we did some pre-processing of the data and corrected it to arrive on a final dataset. Collecting data and then pre-processing it are one of the most crucial things for any data scientist, before they perform any statistics on the data. One more thing we are yet to learn is data cleaning. In the next class, we will learn about how we can clean our data before we perform any statistics to it. Then our data will be prepared and we will find out all the planets that can become our next home?	
Teacher Clicks	✖ End Class	

Activity	Activity Name	Links
Teacher Activity 1	Data from previous class	https://github.com/procodingclass/PRO-129-Datasets
Teacher Activity 2	Exoplanets link	https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=planets
Teacher Activity 3	Solution link	https://github.com/procodingclass/PRO-129-RefCode
Student Activity 1	Dataset	https://github.com/procodingclass/PRO-129-Datasets
Student Activity 2	Exoplanet link	https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=planets