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

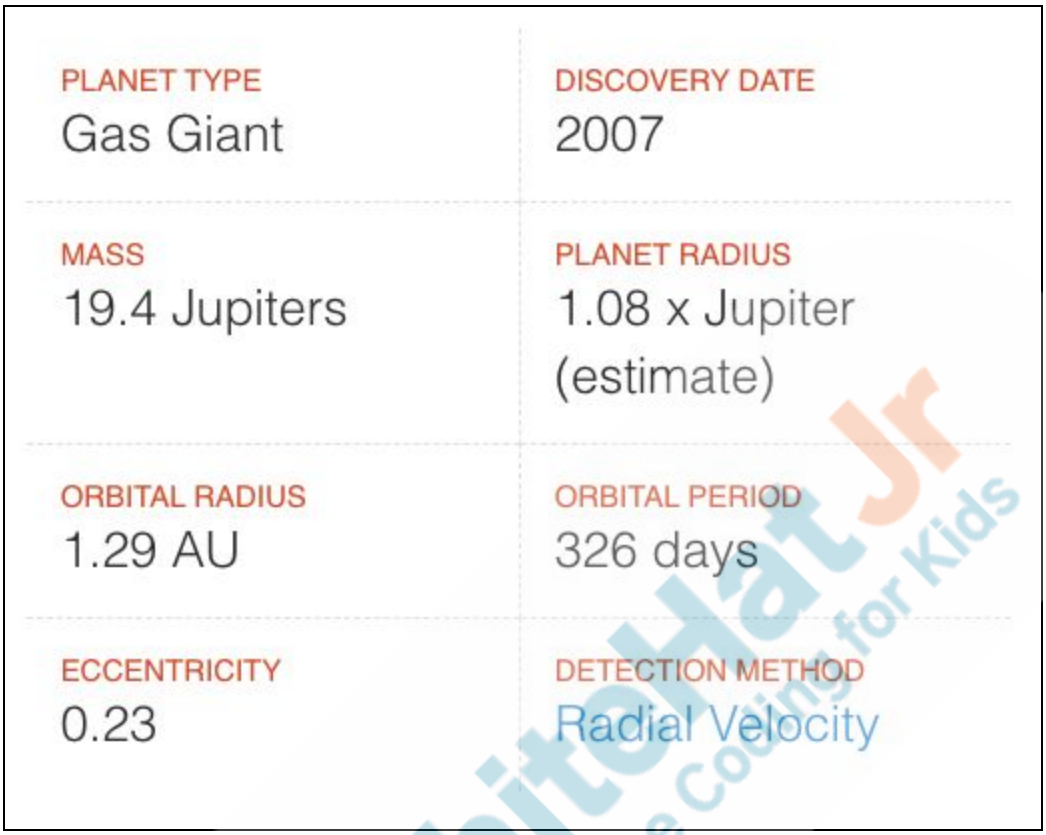
- **Review the concepts learned in the earlier classes**

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

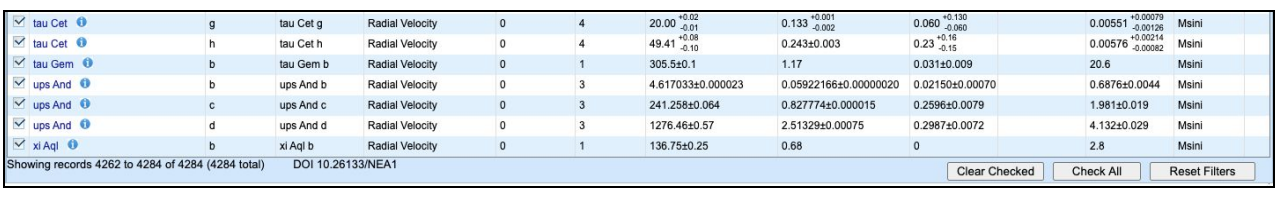
	Let's get started!	
Teacher Initiates Screen Share		
CHALLENGE • **Looking at the previous data** • **Downloading more data from the internet and then merge the data**		
Step 2: Teacher-led Activity (15 min)	*(Before beginning the class, please make the student download the CSV from the link below. This CSV is the final output of all the data scraped in the previous class.)* *<Teacher can download from Teacher Activity 1>* https://raw.githubusercontent.com/whitehatjr/web-scrapping-2/master/final.csv	*<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.
	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 are we 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/



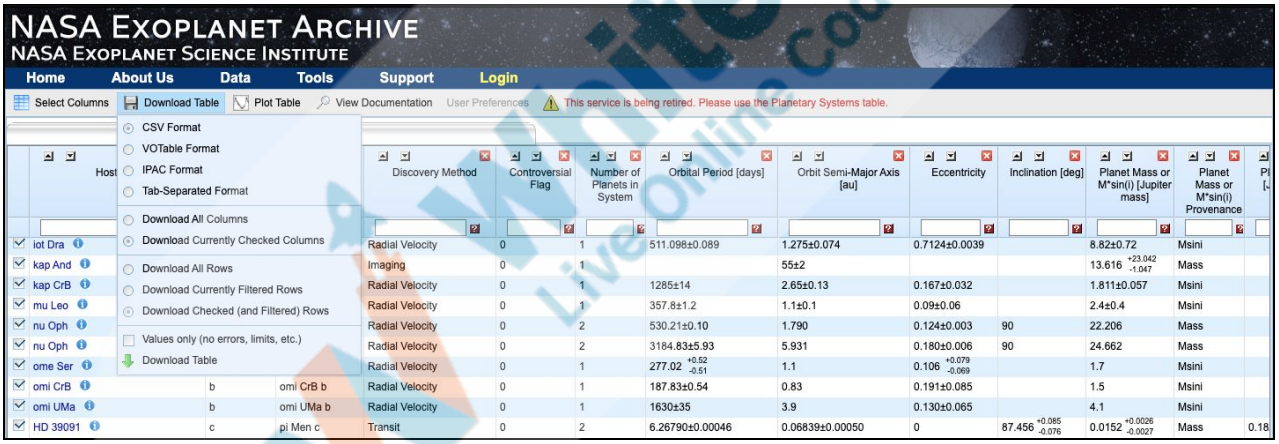
	Great! Now if we look closely, our data tells us that we first scraped the ***Planet Type*** and then we scraped the ***Discovery Date***. If we look at our headers though, we are missing the discovery date header after planet type. We already scraped this data earlier, and that's why we didn't add this 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	*Student adds the temporary header* ***temp_discovery_date*** *header after* ***planet_type****.*

	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. 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>*	



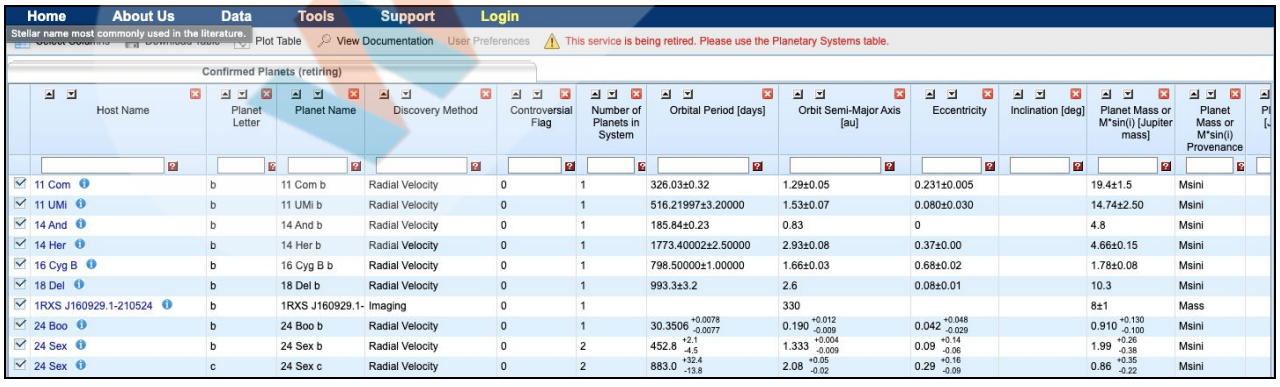
tau Cet	g	tau Cet g	Radial Velocity	0	4	$20.00^{+0.02}_{-0.01}$	$0.133^{+0.001}_{-0.002}$	$0.060^{+0.130}_{-0.060}$	$0.00551^{+0.00079}_{-0.00126}$	Msini
tau Cet	h	tau Cet h	Radial Velocity	0	4	$49.41^{+0.08}_{-0.10}$	0.243±0.003	$0.23^{+0.16}_{-0.15}$	$0.00576^{+0.00214}_{-0.00082}$	Msini
tau Gem	b	tau Gem b	Radial Velocity	0	1	305.5±0.1	1.17	0.031±0.009	20.6	Msini
ups And	b	ups And b	Radial Velocity	0	3	4.617033±0.000023	0.05922166±0.00000020	0.02150±0.00070	0.6876±0.0044	Msini
ups And	c	ups And c	Radial Velocity	0	3	241.258±0.064	0.827774±0.000015	0.2596±0.0079	1.981±0.019	Msini
ups And	d	ups And d	Radial Velocity	0	3	1276.46±0.57	2.51329±0.00075	0.2987±0.0072	4.132±0.029	Msini
xi Aql	b	xi Aql b	Radial Velocity	0	1	136.75±0.25	0.68	0	2.8	Msini

Showing records 4262 to 4284 of 4284 (4284 total) DOI 10.26133/NEA1 Clear Checked Check All Reset Filters

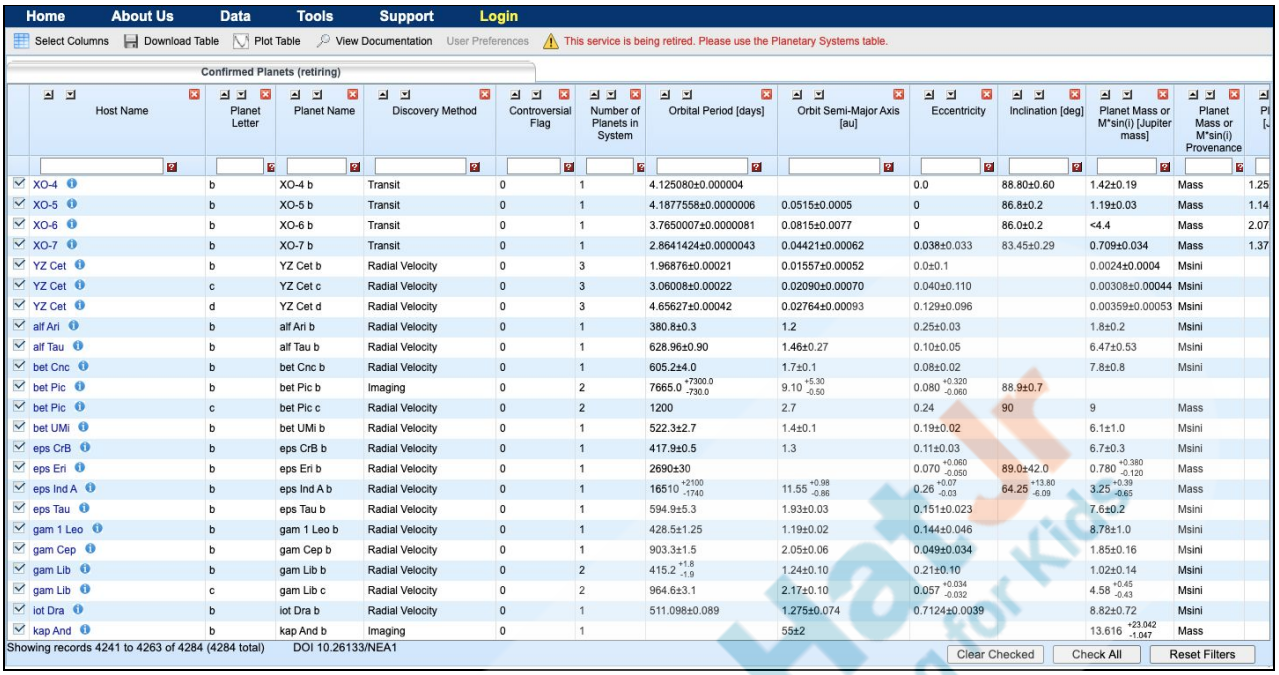
	Here, we can again see at the bottom of the screen that this table has data of 4,284. This is exactly what we need! This table has a lot of data that will help us analyze a lot of data about: our universe, which planets are habitable and which ones are not. Let's download this data! For downloading it, 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.	*Student opens the link from **Student Activity 2** and downloads the CSV.*
		
	Great! Now, we need to merge the data.	
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
• **Ask Student to press ESC key to come back to panel** • **Guide Student to start Screen Share**		

• **Teacher gets into Fullscreen**		
ACTIVITY • **Student tries to find patterns between the data** • **Student writes some code to merge the 2 data sets**		
Step 3: Student-Led Activity (15 min)	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.	*Student looks carefully.*

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



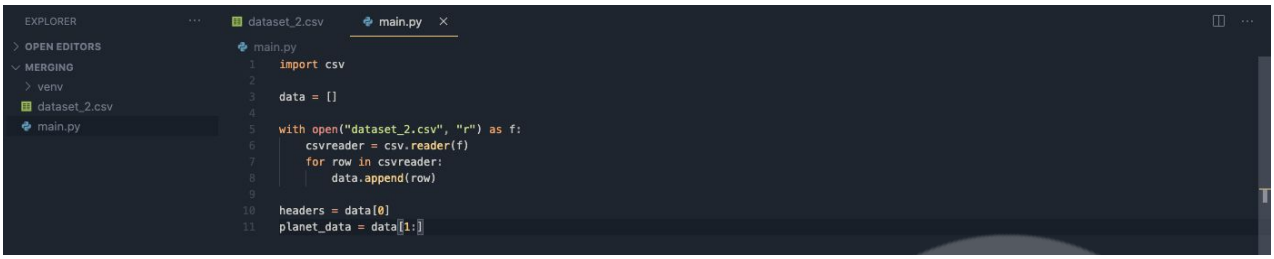
	The name **11 Comae Berenices b** from the data we scraped is written as **11 Com b** in the data we downloaded. Similarly, all the names are different. Therefore we cannot use the names of these data points as a metric to merge the two. Can you think of any patterns that both of the datasets follow?	*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.
	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:-	*Students create a virtual environment.*

	source venv/bin/activate WINDOWS:- **venv\Scripts\activate.bat**	

	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.
	
	Once these lines are removed, we will write the script:
	
	`import csv`
	
	`data = []`
	
	`with open("dataset_2.csv", "r") as f:`
	`  csvreader = csv.reader(f)`
	`  for row in csvreader:`
	`  data.append(row)`
	
	`headers = data[0]`
	`planet_data = data[1:]`
		Student moves the CSV to the new directory, removes the extra data and then creates a python script and reads the CSV data.

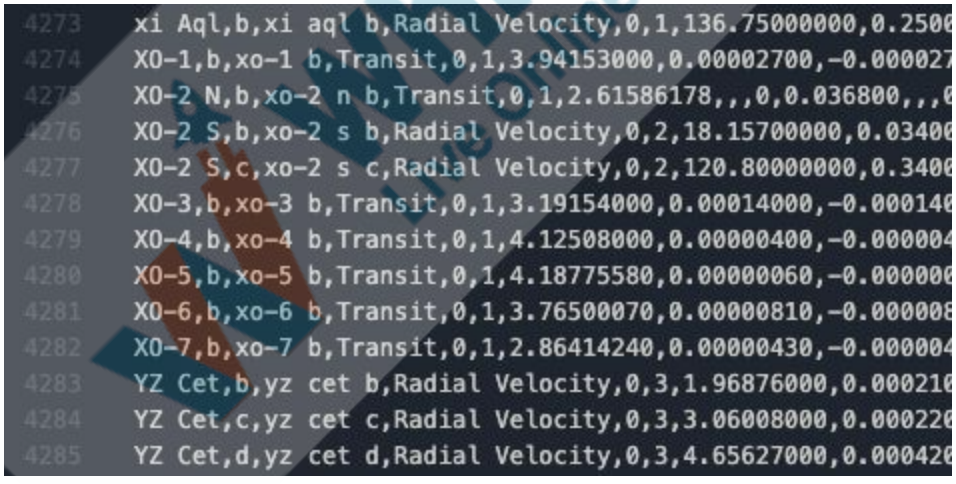


	With this code, now we have all the headers in the variable **headers** and all the planet data in variable **planet_data**. Now, we just need to arrange this in alphabetical order and generate a new CSV! To arrange the data in alphabetical order, we will use the column **pl_name**, which is the third 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 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?	**ESR:** We can convert all the planet names to either lowercase or uppercase and then sort the list in ascending order.

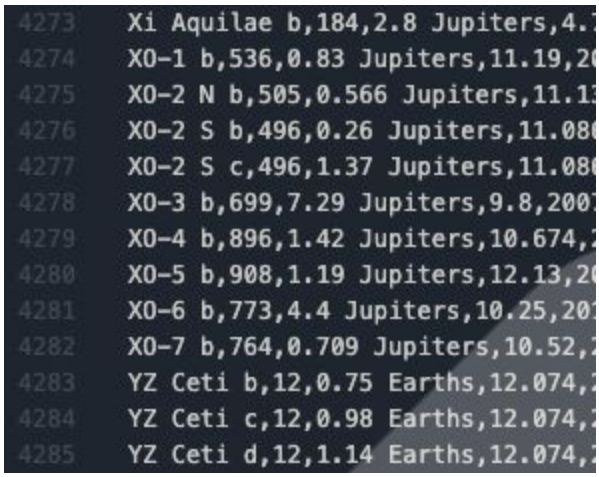
	That's right! Let's quickly code it then.
	
	`#Converting all planet names to lower case`
	`for data_point in planet_data:`
	`  data_point[2] = data_point[2].lower()`
	
	`#Sorting planet names in alphabetical order`
	`planet_data.sort(key=lambda planet_data: planet_data[2])`
	
	`with open("dataset_2_sorted.csv", "a+") as f:`
	`  csvwriter = csv.writer(f)`
	`  csvwriter.writerow(headers)`
	`  csvwriter.writerows(planet_data)`
	
	`planet_data.sort(key=lambda planet_data: planet_data[2])`
	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.

```
#Converting all planet names to lower case
for data_point in planet_data:
    data_point[2] = data_point[2].lower()

#Sorting planet names in alphabetical order
planet_data.sort(key=lambda planet_data: planet_data[2])


with open("dataset_2_sorted.csv", "a+") as f:
    csvwriter = csv.writer(f)
    csvwriter.writerow(headers)
    csvwriter.writerows(planet_data)
```

	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.*



```
4273    xi Aql,b,xi aql b,Radial Velocity,0,1,136.75000000,0.2500
4274    XO-1,b,xo-1 b,Transit,0,1,3.94153000,0.00002700,-0.000027
4275    XO-2 N,b,xo-2 n b,Transit,0,1,2.61586178,,,0,0.036800,,,0
4276    XO-2 S,b,xo-2 s b,Radial Velocity,0,2,18.15700000,0.03400
4277    XO-2 S,c,xo-2 s c,Radial Velocity,0,2,120.80000000,0.3400
4278    XO-3,b,xo-3 b,Transit,0,1,3.19154000,0.00014000,-0.000140
4279    XO-4,b,xo-4 b,Transit,0,1,4.12508000,0.00000400,-0.000004
4280    XO-5,b,xo-5 b,Transit,0,1,4.18775580,0.00000060,-0.000000
4281    XO-6,b,xo-6 b,Transit,0,1,3.76500070,0.00000810,-0.000008
4282    XO-7,b,xo-7 b,Transit,0,1,2.86414240,0.00000430,-0.000004
4283    YZ Cet,b,yz cet b,Radial Velocity,0,3,1.96876000,0.000210
4284    YZ Cet,c,yz cet c,Radial Velocity,0,3,3.06008000,0.000220
4285    YZ Cet,d,yz cet d,Radial Velocity,0,3,4.65627000,0.000420
```



Looks like they are in the same order! 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!

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

```
import csv

dataset_1 = []
dataset_2 = []

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

with open("dataset_2_sorted.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[inde
x] + planet_data_2[index])

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

	Great! We now have all the data merged!	
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://raw.githubusercontent.com/whitehatjr/web-scrapping-2/master/final.csv
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/whitehatjr/data-pre-processing
Student Activity 1	Data from previous class	https://raw.githubusercontent.com/whitehatjr/web-scrapping-2/master/final.csv
Student Activity 2	Exoplanet link	https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=planets